International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 6
ISSN: 2088-5334

Semantic Interoperability in Multi-Cloud Platforms: A Reference Architecture Utilizing an Ontology-Based Approach

Norazian M Hamdan [a,*], Novia Admodisastro [a], Hafeez Bin Osman [a], Muhammad Sufri Bin Muhammad [a]

[a] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia*
**Corresponding author: gs64632@student.upm.edu.my*

***Abstract*—The rapid expansion of cloud computing has necessitated the development of multi-cloud strategies, which leverage the strengths of multiple cloud service providers to mitigate risks such as vendor lock-in and enhance performance and reliability. Nevertheless, semantic interoperability remains a critical challenge in multi-cloud platforms, where diverse cloud services need to communicate and function seamlessly. Current solutions lack a unified semantic-based representation within reference architectures in multi-cloud platforms and mainly focus on the independent interoperability of a service model, i.e., SaaS, PaaS, and IaaS. This study addresses the critical issue of semantic interoperability in multi-cloud platforms, where the heterogeneity of proprietary cloud solutions impedes seamless integration and communication. Thus, we proposed a reference architecture utilizing an ontology-based approach to facilitate semantic interoperability across diverse cloud platforms. The reference architecture is based on five semantic interoperability requirements identified in our previous study. This paper presents the design and development of a reference architecture that includes high-level and low-level components supported by a taxonomy of semantic interoperability in multi-cloud platforms. Expected outcomes of this study include a standardized framework using an ontology-based approach for semantic mapping and integration of cloud services, which will significantly enhance interoperability and efficiency in multi-cloud platforms. The significance of this research lies in its potential to advance the state of knowledge and practice in multi-cloud computing, enabling more robust and flexible cloud service ecosystems.**

***Keywords*—Multi-cloud platforms; ontology; reference architecture; taxonomy; semantic interoperability.**

Manuscript received 11 Mar. 2024; revised 18 Jun. 2021; accepted 12 Sep. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

Cloud computing has revolutionized the IT landscape, providing scalable and flexible resources over the Internet. As businesses expand, the limitations of a single-cloud strategy have become apparent, leading to adopting multi-cloud strategies. A multi-cloud strategy involves utilizing multiple independent cloud architectures that operate together as a unified cloud system, distributing applications across these clouds in separate components [1]. These strategies leverage multiple cloud service providers to mitigate risks like vendor lock-in and enhance performance and reliability [2].

The primary issue in multi-cloud platforms is enabling semantic interoperability, allowing different cloud services to understand and utilize shared data effectively. Nevertheless, this issue arises because cloud providers offer proprietary solutions, thus resulting in the heterogeneity of cloud services [3]. As a result, cloud consumers cannot easily switch from one provider to another, causing the consumers to be fixed to a single provider (vendor lock-in). This complicates or prevents interoperability between services offered by different cloud providers or deployed on various clouds [4].

Current research efforts on semantic interoperability present solutions in the form of a reference architecture [5], frameworks [6], [7], [8], microservices architecture [9], toolkit [10], [11], and a few semantic models utilizing ontologies [12], [13], [14], [15], [16], [17]. However, these solutions lack a standardized representation of semantic-based solutions within reference architectures. Aside from that, most solutions focus on interoperability independently across the three cloud service models, like [8], [10], [11], [14], [15], [17] for Software-as-a-Service (SaaS), [13] for Platform-as-a-Service (PaaS), and [6], [9] for Infrastructure-as-a-Service (IaaS)). Studies by [7] and [16] built their solutions to address the PaaS and IaaS service models, and only the works by [5] and [12] have addressed semantic interoperability across all three service models. Limited efforts have been dedicated to achieving interoperability across these three service models due to the complexity posed by cloud heterogeneity, which makes it challenging to ensure seamless integration between different cloud environments [4]. Consequently, semantic interoperability remains a significant issue in multi-cloud platforms, hindering seamless integration and communication across various cloud platforms. Over the years, semantic ontology has been a well-recognized approach for enabling seamless interoperability by utilizing ontologies and their defined rules. This paper discusses insights into an ontology-based approach to enhance semantic interoperability and achieve seamless integration and communication across diverse cloud platforms. We then present the design and development of a reference architecture utilizing this ontology-based approach based on five semantic interoperability requirements identified in our previous study.

The paper's organization is as follows: Section II discusses the background of cloud interoperability, semantic ontology, multi-cloud interoperability, and semantic ontology in multi-cloud interoperability. This section also presents the ontology development process employed in the proposed reference architecture. Section III presents our proposed solution by describing it as a high-level reference architecture and low-level reference architecture and producing a taxonomy of semantic interoperability in multi-cloud platforms and its results. Finally, we conclude our paper in Section IV.

II. Materials and Method

In the field of multi-cloud computing, cloud interoperability presents significant challenges. These challenges encompass syntactic, semantic, and other types of interoperability [18]. Additionally, common issues arise from different cloud providers employing distinct virtualization technologies, service descriptions, pricing models, and Service Level Agreements (SLAs), as well as varying Application Programming Interfaces (APIs) and non-standardized authentication and authorization methods [19], [20]. Research in this field has focused on developing standardized protocols, middleware solutions, and interoperability frameworks to address these issues. The need to facilitate interoperability across various platforms develops as the number of cloud platforms supporting research rises [21]. Moreover, interoperability across multi-cloud platforms and other cloud services is essential for efficient operation [22].

Nevertheless, the cloud community sees semantic interoperability between heterogeneous clouds as extremely important. Researchers have explored semantic interoperability approaches to enable seamless communication and integration among cloud services in multi-cloud platforms. Firstly, the standard-based approach uses the common standards, protocols, and data formats produced by the standardization bodies. These standards are used across the development, management, security, and other related aspects of cloud platforms [23]. However, the problem with this approach is that no standard has been universally accepted to solve the multi-cloud interoperability issues [24].

Secondly, the model-based approach involves the development and deployment of shared models. The limitation of this approach is that it cannot transition between models and real-world implementation semantics [25]. Third is the open libraries and open services, which are a collection of software libraries and software services. They are publicly available and can be freely used, modified, and distributed [23]. However, this approach has a few limitations, like poor end-user documentation, lack of long-term support in the open-source community, and the fact that open-source solutions are usually presented as they are [26].

Finally, the semantic approach refers to understanding, organizing, and interpreting information based on its meaning and relationships rather than just its syntax or structure. This approach employs the Semantic Web technologies like ontologies, semantic annotations, and such [27]. In contrast to the three methods described earlier, the semantic approach provides a more robust and scalable solution for semantic interoperability [28]. For example, using ontologies, knowledge within a domain can be represented formally, including the entities, attributes, and relationships. This allows for a shared understanding that can be universally applied across different systems. This standardized framework ensures data interpretation and service integration consistency, addressing the heterogeneity inherent in multi-cloud platforms.

Therefore, by leveraging semantic ontologies and rules defined within these ontologies, cloud services can achieve precise data mapping, transformation, and integration, facilitating seamless interoperability. This approach mitigates the risks of vendor lock-in and enhances the flexibility and scalability of cloud services, making it particularly suitable for the dynamic and diverse landscape of multi-cloud platforms. The following subsections discuss the main background works of the study.

A. Semantic Ontology

Ontologies are formal representations of concepts and their relationships within a specific domain [29]. They are used to reflect domain knowledge, where the ontology classes are typically depicted using graphical models, as models are considered to have explicit meanings [30]. In cloud computing, ontologies describe common cloud functionalities, enabling a shared terminology that supports interoperability [12]. As a result, cloud providers can achieve interoperability between their platforms, enabling cost-effective data migration and facilitating the brokerage of data transfers [31]. Due to their ability to overcome the problems caused by semantic heterogeneity, ontologies have emerged as the core element of semantic solutions [4], [24], [32], [33].

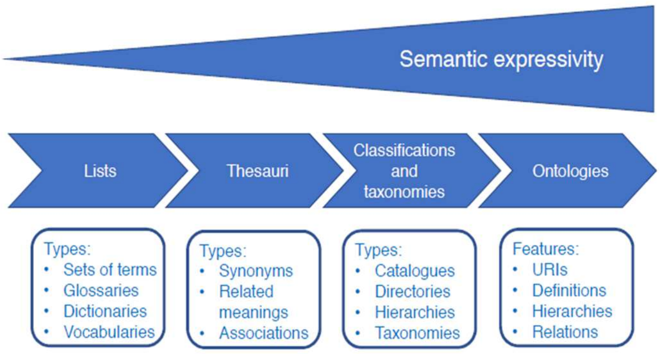

Fig. 1 Semantic Expressivity Spectrum for Knowledge Management Resources [34]

Since ontologies are represented formally, they surpass the capabilities of lists, thesauri, and taxonomies. The semantic expressivity spectrum in Fig. 1 highlights various knowledge management resources ranging from simple lists to complex ontologies. Ontologies are at the higher end of the spectrum, providing a higher level of expressivity by incorporating features such as Uniform Resource Identifiers (URIs), definitions, hierarchical structures, and relationships among concepts. This allows for the formalization of semantics, which facilitates machine processability, reasoning, knowledge production, and automatic identification of inconsistencies, resulting in a robust framework for managing and integrating complex knowledge domains [34]. Nevertheless, lower-level spectrums, specifically taxonomies, are equally significant because they are part of the ontology development process [35]. The taxonomies organize the concepts in a hierarchical structure, like a superclass-subclass hierarchy [36].

According to Al-Sayed et al. [36], an ontology consists of three core elements: *classes* (or concepts), *objects* (or instances), and *properties* (or relations). Similarly, Reyes-Peña & Tovar-Vidal [37] stated that an ontology is typically constructed from the three core elements mentioned in [36] and the other two elements, functions and *axioms*. *Classes* are used to define a collection of instances that share similar properties. *Objects* are individuals or specific examples of classes. *Properties* indicate relationships between instances (e.g., object properties) or between instances and data (e.g., data-type properties). A *function* is an element that uses other elements to calculate information. Lastly, *axioms* impose limitations, rules, and logic on the relationship between ontology elements.

Various approaches exist for representing ontologies like Unified Modelling Language (UML) diagrams, natural language descriptions, knowledge graphs, and more. However, the most popular approach is using semantic web languages. According to the World Wide Web Consortium (W3C), Semantic Web languages are designed to represent structured information on the web, enabling machines to understand and process data meaningfully. The primary languages used in the Semantic Web include Resource Description Framework (RDF), RDF Schema (RDFS), and Web Ontology Language (OWL).

RDF provides a basic framework for representing resource information in subject-predicate-object triples [6]. RDFS extends RDF by adding mechanisms for defining vocabularies, including classes and properties, to create a basic ontology structure [38]. OWL builds on RDF and RDFS, offering more advanced features for expressing complex relationships, constraints, and rules within ontologies [39]. These languages collectively facilitate data integration, sharing, and reuse across diverse domains and applications, driving interoperability and richer semantic understanding on the web.

B. Multi-Cloud Interoperability

According to ISO/IEC 22123-1:2023, cloud interoperability involves a cloud consumer's system exchanging information with a cloud service or a cloud service communicating with other cloud services in a predefined way to achieve the intended outcomes [40]. Hence, multi-cloud interoperability generally refers to the seamless integration and operation of cloud services from different providers within a cohesive multi-cloud strategy. This strategy leverages various multi-cloud platforms, such as Amazon Web Services (AWS), Google Cloud, and Microsoft Azure, to optimize performance, cost, and compliance while avoiding vendor lock-in. Organizations can move data and applications freely between these platforms by ensuring cloud interoperability and enhancing flexibility and resilience.

Multi-cloud interoperability ensures seamless interaction and integration across different cloud service models, including SaaS, PaaS, and IaaS. Therefore, cloud interoperability is based on three levels as follows:

1) SaaS Level Interoperability: At the SaaS level, interoperability ensures that various software applications from different cloud providers can seamlessly integrate, allowing for a unified user experience and efficient data exchange. [41]. For example, assuming a company is using multiple SaaS applications for its operations: Salesforce on AWS for customer relationship management (CRM), Microsoft Teams on Azure for team communication, and QuickBooks on Google Cloud for accounting. These applications are on different cloud providers yet must exchange data seamlessly.

When a sales representative updates a customer's information in Salesforce, that update should automatically be reflected in QuickBooks to ensure the accounting records are up to date. Similarly, if there's a team discussion on Microsoft Teams about a particular client's billing issue, relevant data from QuickBooks should be easily accessible within Microsoft Teams. The functional features for SaaS interoperability may include CRM, Enterprise Resource Planning (ERP), email applications for communications, office tools for productivity, and user interfaces [36].

2) PaaS Level Interoperability: For PaaS, interoperability is vital for enabling developers to build, deploy, and manage applications across multiple cloud platforms without facing compatibility issues [41]. This flexibility supports innovation and agility in application development. For example, a software development company wants to create a web application that leverages the strengths of different cloud platforms to ensure optimal performance, flexibility, and cost-effectiveness. The company employs the following PaaS applications: Visual Studio Code on Azure for development and testing, Google App Engine on Google Cloud for deployment, and Azure SQL Database on Azure for managing relational databases. PaaS functional features can be categorized into design, modeling, development, and testing [36].

3) IaaS Level Interoperability: At the IaaS level, interoperability allows for the efficient orchestration and management of computing resources, storage, and networking across different cloud infrastructures [41]. This capability is essential for optimizing performance, reducing costs, and ensuring reliability. Although infrastructure is considered at IaaS level interoperability, it is almost impossible to create a system independent of infrastructure details [42]. For example, a large enterprise must run a high-performance web application that demands efficient resource management, high availability, and cost optimization. The company utilizes the following IaaS services: Amazon EC2 on AWS for its main web servers (i.e., computing resources), Google Cloud Storage on Google Cloud for its storage solutions, and Azure Virtual Network on Azure for connectivity between different components (i.e., networking). IaaS functional features may include access mechanisms, virtual resources, storage, network, security, service-level agreement, and others [41].

Several works have been found to address multi-cloud interoperability based on these levels. Bouzerzour et al. [14] proposed a model that acts as a transformation mediator based on a Generic Cloud Service Description (GCSD). The proposed solution employs mapping rules to convert diverse cloud service descriptions into a standardized GCSD format, enabling interoperability. The authors further developed a Model-as-a-Service (MaaS) Cloud Interoperability Pivot Model (CIPiMo) to support interoperability by converting the cloud service description languages into a GCSD model to standardize them [15].

Benhssayen and Ettalbi [6] designed the framework for addressing semantic interoperability among IaaS resources. The proposed framework allows cloud consumers to request specific cloud resources and receive a list of available resources from several providers that meet their needs. Next, they extended their framework to enable semantic interoperability to retrieve IaaS resources and offer PaaS services in a multi-cloud platform [7].

Anglano et al. [10], [11] created EasyCloud, a toolkit that facilitates the construction and use of multi-cloud systems. They claimed that the tool is interoperable across multiple platforms, platform-independent, efficient in resource provisioning, and simple to use.

Lastly, Mane et al. [17] introduced the Middleware for Data-as-a-Service (DaaS)/Database-as-a-Service (DBaaS) and SaaS called MIDAS middleware and a Domain Specific Modelling Language (DSML) to ensure semantic interoperability and data integration between DaaS/DBaaS providers. In their paper, the authors demonstrated the adaptability of MIDAS across various cloud platforms and addressed interoperability challenges, reducing the effort required to overcome vendor lock-in issues.

C. Semantic Ontology in Multi-Cloud Interoperability

Ontology plays a crucial role in facilitating semantic interoperability in multi-cloud platforms. It defines and standardizes knowledge required for various domains to ensure consistent interpretation and use of data across diverse systems [43]. Other than that, ontologies can bridge the gap between different cloud services by providing a unified framework or layer that masks the heterogeneity of diverse cloud services [3]. This approach is significant in multi-cloud interoperability, where developers and programmers can use ontologies to establish interoperability between various cloud providers and their offerings [44], [45]. Consequently, the systematic literature review by [46] reveals that ontologies are frequently mentioned as an approach for multi-cloud interoperability.

There are a few recent works on ontology-based approaches for multi-cloud interoperability. Firstly, an IEEE multiagent Foundation for Intelligent Physical Agent (FIPA) compliant reference architecture for cloud service discovery and selection using cloud ontology aims to mitigate vendor lock-in issues and enhance portability and interoperability in cloud computing [47]. Secondly, an ontology built on OWL formally represents the interoperability between SaaS and DaaS layers called MIDAS-OWL [48]. Third is a cloud ERP API ontology aimed at addressing the problem of vendor lock-in by enabling interoperability among different cloud ERP systems [12]. And lastly, an expressive OWL ontology-based medical data model to facilitate effective semantic interoperability [49].

Based on these existing works mentioned above, OWL has been used within their work to address the multi-cloud interoperability issues. OWL is considered the ideal candidate for formally describing ontologies due to its flexibility in navigating ontologies with query languages and visual aids, as well as its capacity to represent complicated relationships. It constitutes 13% of cloud ontology studies on cloud interoperability and standardization, which helps address vendor lock-in issues [36].

In summary, the multi-cloud interoperability concept is critical in multi-cloud platforms, which organizations use to avoid vendor lock-in, enhance reliability, and leverage the best services from each provider. Despite advancements, full cloud interoperability remains a complex goal, requiring ongoing research and collaboration among cloud stakeholders. The following section presents our attempt to develop the proposed solution.

We proposed a reference architecture utilizing an ontology-based approach for facilitating multi-cloud interoperability. The incorporated ontologies can provide a shared, formalized understanding of the concepts, relationships, and data structures across different cloud platforms. We chose OWL to represent the ontologies for multi-cloud interoperability since it has the capacity to express and formalize the semantics of various cloud services. The proposed solution is also based on our work [50], a compiled set of requirements for semantic interoperability in multi-cloud platforms. The general requirements are described as follows:

1) REQ-01: A reference architecture that utilizes role-based and layer-based architectures to depict the cloud actors (stakeholders), components, and their relationships in a multi-cloud platform.

2) REQ-02: A semantic interoperability layer-based architecture that utilizes the ontology-based approach.

3) REQ-03: A taxonomy of semantic interoperability in multi-cloud platforms that represents interoperability between cloud services.

4) REQ-04: A repository of cloud ontologies that capture the semantics of multiple cloud services and are reusable across different platforms. In this paper, we focus on the OWL as the primary language to develop our ontologies.

5) REQ-05: Semantic rules to map provider-specific terms to the specified terms in our OWL ontologies.

Meanwhile, the ontology development process is shown in Fig. 2. Firstly, we identify the core concepts and classes representing the main entities in the multi-cloud platforms, such as CloudProvider, CloudService, CloudResource, and others. Next, we define the properties and relationships between these classes (concepts). This includes defining object properties (relationships between two individuals) and data properties (relationships between individuals and data values). This is followed by detailing the hierarchical relationships between the superclass and its subclasses. At this stage, we develop the taxonomies along with the ontologies.

Then, we create instances of these classes to represent specific cloud providers, services, and resources. After that, we add the data properties to specify attributes of these instances, like provider names, service names, or resource types. The next process includes defining rules and axioms. OWL allows for reasoning and inference, deducing new knowledge from existing data. For example, we could define a rule to infer that a user utilizing a computer service provided by AWS is also an AWS user. In the final process, we include SPARQL Protocol and RDF Query Language (SPARQL) queries to retrieve and integrate data across different cloud services. Ultimately, at the end of the ontology development process, we plan to produce the taxonomies and several OWL ontologies that represent the semantics of cloud services.

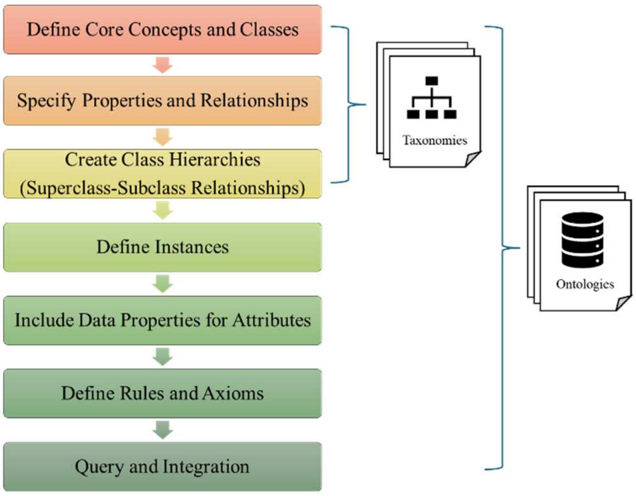

Fig. 2 Ontology Development Process

III. Results and Discussion

In this section, three results are presented and discussed: a high-level reference architecture, a low-level reference architecture, and a taxonomy of semantic interoperability in multi-cloud platforms. We presented our proposed solution using a reference architecture because it may serve as a blueprint or template for the design, deployment, and management of a specific domain, such as cloud computing. It is a high-level, abstract representation of its components, relationships, and interactions required to support a particular set of capabilities. Additionally, a reference architecture provides a systematic approach to achieve interoperability between different cloud services and providers [51]. The details are described in the following subsections.

A. High-Level View Reference Architecture

In this section, we aim to achieve REQ-01 as stated in Section III. Fig. 3 shows our proposed high-level view reference architecture for facilitating semantic interoperability in multi-cloud platforms. The key components include two core cloud actors (i.e., *Cloud Consumer* and *Cloud Provider*) that interact through the Semantic Interoperability Layer, with the Security and Privacy Layer and Governance Layer spanning across the reference architecture.

We refer to our work based on the two prominent Cloud Computing Reference Architecture (CCRA) from the National Institute of Standards and Technology (NIST) [52] and the International Business Machines Corporation (IBM) [53]. The NIST CCRA outlines five prominent cloud actors: Cloud Consumer, Cloud Provider, Cloud Auditor, Cloud Broker, and Cloud Carrier [52]. In contrast, the IBM CCRA highlights Cloud Consumer, Cloud Provider, and Cloud Creator as their major actors [53]. Since our study aims to address semantic interoperability issues in multi-cloud platforms, we limit our focus to the core cloud actors in the cloud service interactions to achieve simplicity in designing our reference architecture. Therefore, we identify *Cloud Consumer* and *Cloud Provider* as our proposed reference architecture's two core cloud actors. The actors are described as follows:

1) Cloud Consumer: An individual or organization that utilizes cloud computing services provided by cloud providers. Based on their needs, consumers access, manage, and use cloud resources, such as applications, storage, and processing power [52]. As shown in Fig. 3, there are three categories of cloud consumers. Firstly, a SaaS consumer uses software applications hosted and managed by a cloud provider. These applications are accessed online, typically through a web browser like Microsoft 365 or Google Workspace. Next, a PaaS consumer uses a platform provided by the cloud provider to develop, deploy, and manage applications. The platform includes tools, frameworks, and infrastructure for development. For example, a cloud developer is one of the roles of a cloud consumer who designs and builds applications using cloud services in any level of service model. Finally, an IaaS consumer utilizes the cloud provider's basic computing resources (e.g., virtual machines, storage, and networks). They have control over the operating system, storage, and deployed applications. Examples of platforms include Amazon EC2 or Google Compute Engine.

2) Cloud Provider: A company or organization offering consumers cloud computing services. Cloud providers own and manage the software, platforms, and infrastructure that deliver cloud services. Additionally, they offer various service models like SaaS, PaaS, and IaaS [52]. Examples of service providers include AWS, which provides computing power (EC2) and storage (S3); Microsoft Azure, which offers virtual machines and databases; IBM Cloud, which offers hybrid cloud solutions and AI-powered services; and Google Cloud Platform, which is known for its data analytics and machine learning solutions.

The *Semantic Interoperability Layer* functions as an intermediary between *Cloud Consumers and Providers.* It semantically transforms data and services, enabling the provision of cloud services across multiple platforms. The following subsection provides a detailed description of this layer.

Lastly, the two cross-cutting aspects (*Security and Privacy*, and *Governance*) that span across the reference architecture contain the capabilities that require consistent implementation within a cloud computing system and cooperation across roles. *Security and Privacy* are crucial for data protection and *Governance* is for ensuring consistent policy enforcement [54].

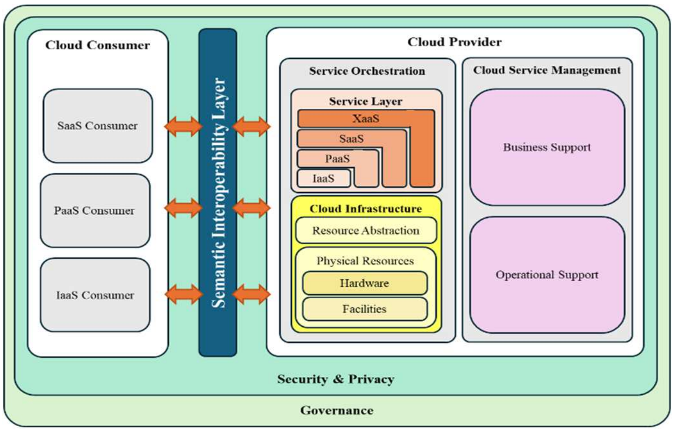

Fig. 3 High-Level View of a Semantically Interoperable Reference Architecture

B. Low-Level View Reference Architecture

We intend to accomplish REQ-02, from the list of requirements in Section III, by developing a low-level view reference architecture with three main layers (refer Fig. 4):

1) Cloud Consumer Layer: In this layer, the Application Layer holds the data and service components. This layer utilizes semantically mapped data provided by the *Semantic Interoperability Layer* to offer various services to cloud consumers.

2) Cloud Provider Layer: This layer consists of various cloud services (e.g.: A, B, … *n*) provided by different cloud providers.

3) Semantic Interoperability Layer: This layer plays a crucial role in achieving semantic interoperability by using OWL ontologies to translate and map data between different cloud services.

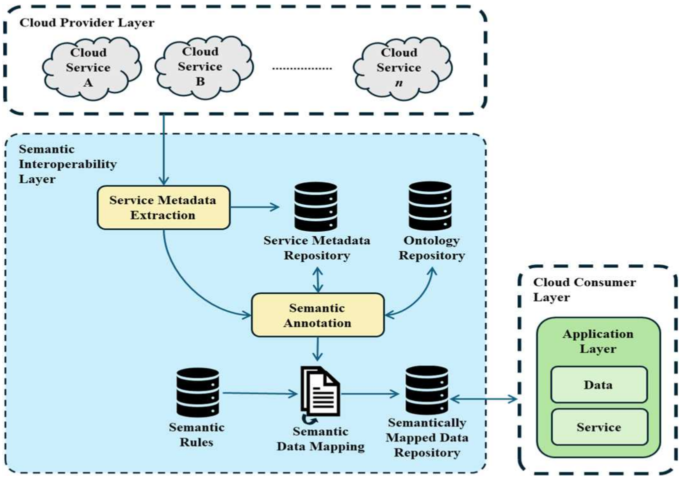

Fig. 4 Low-Level View of a Semantically Interoperable Reference Architecture

The *Semantic Interoperability Layer* is crucial in enabling seamless data exchange and integration across multi-cloud platforms. This layer includes components such as *Service Metadata Extraction*, *Semantic Annotation*, *Semantic Data Mapping*, *Service Metadata Repository*, *Ontology Repository*, *Semantically Mapped Data Repository*, and *Semantic Rules*. The roles of each component are as follows:

1) Service Metadata Extraction: This component extracts metadata from various cloud services. Metadata includes information about the service's functionality, input and output parameters, performance characteristics, and more. The extracted metadata is normalized and then stored in the *Service Metadata Repository*.

2) Semantic Annotation: This component adds semantic tags and labels to the extracted metadata, providing a meaningful context to the data. OWL ontologies from the *Ontology Repository* will be used to annotate metadata with standardized terms and concepts.

3) Semantic Data Mapping: This component maps data from different cloud services to a common semantic model, ensuring that data can be understood and processed consistently across platforms.

4) Service Metadata Repository: This repository stores the extracted and semantically annotated metadata, making it accessible for further processing and querying.

5) Ontology Repository: This is a central repository for storing and managing our OWL ontologies, which are used for semantic annotation and data mapping.

6) Semantically Mapped Data Repository: This repository stores semantically mapped data, ensuring consistency and interoperability across different cloud platforms.

7) Semantic Rules: Semantic rules define how data and services should be interpreted, transformed, and integrated based on the ontologies and semantic annotations. These rules are used to map provider-specific terms to the standardized terms that we defined in our OWL ontologies.

To develop the OWL ontology, we produced a taxonomy of semantic interoperability in multi-cloud platforms, which is described in detail in the next subsection.

C. A Taxonomy of Semantic Interoperability in Multi-Cloud Platforms

Taxonomy is the scientific field dedicated to classifying living organisms [55]. This systematic approach to categorizing life forms extends beyond biology and finds relevance in various other domains, including computer science. In computer science, taxonomy classifies and organizes data, algorithms, and information into a structured hierarchy. This structure facilitates easier navigation and retrieval of information and supports effective knowledge management by creating transparent, standardized relationships between data elements [56]. Hence, using an ontology-based approach, we presented a taxonomy of semantic interoperability in multi-cloud platforms to organize the essential components for a semantic interoperable cloud. This part of the study is to fulfill REQ-03 listed in Section III.

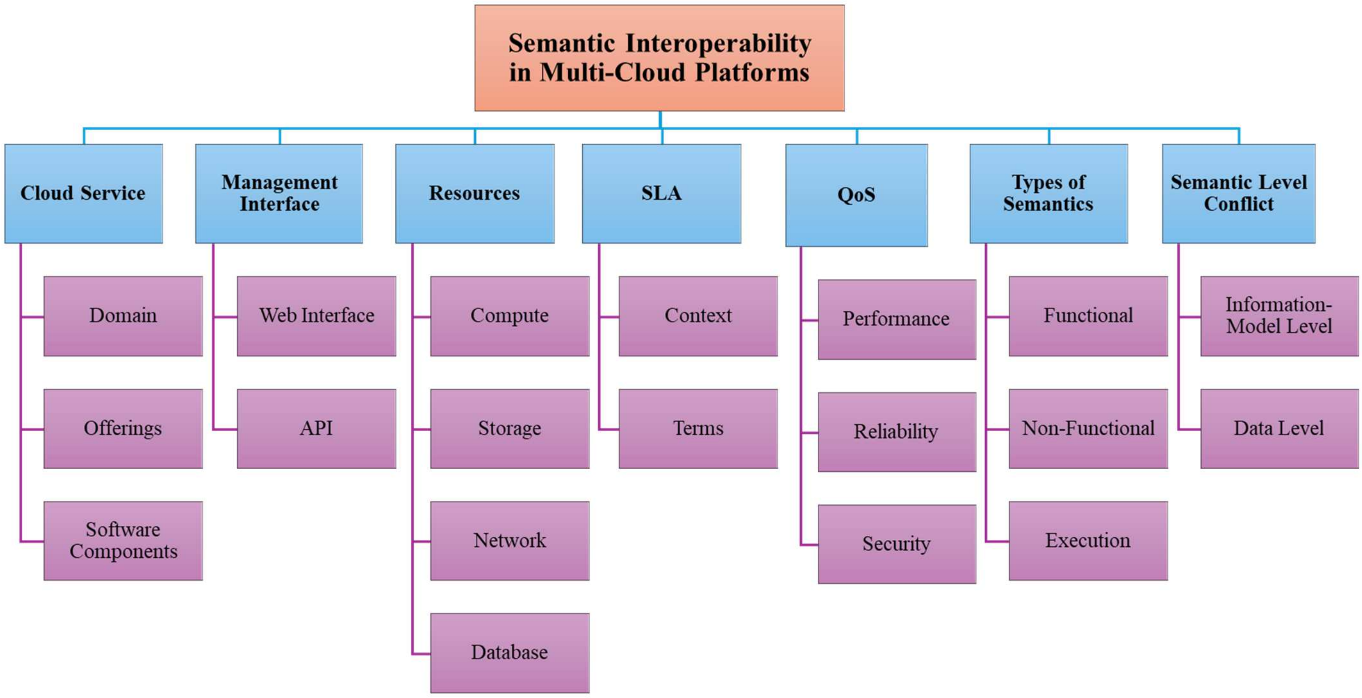

Fig. 5 A Taxonomy of Semantic Interoperability in Multi-Cloud Platforms

The taxonomy includes seven key components crucial for ensuring consistent semantics and functionality across different cloud platforms (refer to Fig. 5). We discuss these key components as follows:

1) Cloud Service: This component refers to the on-demand services delivered to companies and customers over the Internet. It is typically divided into three main categories: SaaS, PaaS, and IaaS [52]. The sub-components include the specific area of cloud services being addressed (*Domain*), the types of services provided (*Offerings*), and a collection of supplementary features that can enhance the basic functionality of a cloud offering (*Software Components*) [57].

2) Management Interface: This component provides a centralized point of interaction for managing various aspects of the cloud platform, including infrastructure, applications, and services. It consists of a *web interface to enable users to interact* with cloud resources and services and an *API* to allow different software components to communicate [57].

3) Resources: This refers to the various components and assets that are provided and managed in a cloud platform to support applications and services. The resource types include *Compute* (e.g., virtual machines and containers), *Storage* (e.g., block storage, object storage, and file storage), *Network* (e.g., virtual network, load balancer, and Domain Name Systems (DNS)), and *Database* (e.g., relational database, and data warehouse) [36].

4) SLA: This component refers to the terms of the agreement, which includes *Context* (i.e., parties involved, agreed services, and the duration of the agreement) and *Terms* (i.e., service terms and guarantee terms). These terms define the expected service quality and commitments between cloud providers and consumers [58].

5) QoS: This component covers quality of service metrics such as *Performance*, *Reliability*, and *Security*. *Performance* includes response time, throughput, latency, execution time, and bandwidth. *Reliability* contains indicators like availability, uptime, failure rate, Mean Time Between Failures (MTBF), and Mean Time to Repair (MTTR). Additionally, *Security* aspects are addressed, including encryption, authentication, authorization, compliance, and data integrity [3].

6) Types of Semantics: This component refers to the types of semantics that have occurred. It relates to the operations and capabilities of cloud services (*Functional*), encompasses QoS metrics and security aspects (*Non-Functional*), and focuses on runtime aspects (*Execution*) [57].

7) Semantic Level Conflict: This component indicates the level at which conflicts are identified, which can occur at the *Information-Model Level* and *Data Level*. At the *Information-Model Level*, this relates to differences in logical structures, data structures, inconsistent metadata, interactions between different data structures, and restrictions on access to specific data items. Meanwhile, *Data Level* describes the data variations resulting from different representations and interpretations of the same or comparable data [57].

Concerning semantic interoperability in multi-cloud platforms, this taxonomy is vital for standardizing and categorizing various aspects of cloud services. Providing a common framework for understanding and organizing different components in a hierarchical structure helps ensure that different cloud services can interoperate seamlessly. This is useful for future work, where we will use this taxonomy to develop our cloud ontology for semantic interoperable clouds.

IV. Conclusion

Achieving seamless multi-cloud interoperability requires ongoing research and collaboration among cloud stakeholders. This study presents a significant step towards this goal by introducing an ontology-based reference architecture and a taxonomy of semantic interoperability in multi-cloud platforms. The proposed solutions address key challenges such as service heterogeneity and vendor lock-in, offering a robust framework for semantic interoperability in multi-cloud platforms. The reference architecture serves as a blueprint for designing, deploying, and managing cloud services, providing a systematic approach to achieving interoperability across diverse cloud platforms. Additionally, taxonomy is a reference for managing and developing ontologies on similar domains. Due to its scalability and flexibility, the taxonomy offers options for expansion.

Our future work will focus on expanding the taxonomy with more detailed attributes and relationships. This is significant as it will enhance the precision and depth of the taxonomy, facilitating more accurate and comprehensive semantic mappings. Ultimately, the improved taxonomy will serve as a guideline for developing OWL ontologies for our study, leading to more robust and interoperable cloud service solutions in multi-cloud platforms. Furthermore, we plan to achieve REQ-04 and REQ-05 (from Section III) in our future work.

References

[1] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu, “Cloud Computing Architecture: A Critical Analysis,” *2018 18th International Conference on Computational Science and Applications (ICCSA)*, pp. 1–7, Jul. 2018, doi: 10.1109/iccsa.2018.8439638.

[2] M. Reece, T. E. Lander, M. Stoffolano, A. Sampson, J. Dykstra, S. Mittal et al., "Systemic Risk and Vulnerability Analysis of Multi-cloud Environments", *arXiv preprint*, 2023.

[3] M. M. Al-Sayed, H. A. Hassan, and F. A. Omara, “CloudFNF: An ontology structure for functional and non-functional features of cloud services,” *Journal of Parallel and Distributed Computing*, vol. 141, pp. 143–173, Jul. 2020, doi: 10.1016/j.jpdc.2020.03.019.

[4] N. E. H. Bouzerzour, S. Ghazouani, and Y. Slimani, “A survey on the service interoperability in cloud computing: Client-centric and provider-centric perspectives,” *Software: Practice and Experience*, vol. 50, no. 7, pp. 1025–1060, Jan. 2020, doi: 10.1002/spe.2794.

[5] K. Sana, N. Hassina, and B.-B. Kadda, “Towards a Reference Architecture for Interoperable Clouds,” *2021 8th International Conference on Electrical and Electronics Engineering (ICEEE)*, pp. 229–233, Apr. 2021, doi: 10.1109/iceee52452.2021.9415944.

[6] K. Benhssayen and A. Ettalbi, “Semantic Interoperability Framework for IAAS Resources in Multi-Cloud Environment,” *International Journal of Computer Science and Network Security*, vol. 21, no. 2, Feb. 2021, doi: 10.22937/ijcsns.2021.21.2.1.

[7] K. Benhssayen and A. Ettalbi, “An Extended Framework for Semantic Interoperability in PaaS and IaaS Multi-cloud,” *Digital Technologies and Applications*, pp. 415–424, 2022, doi: 10.1007/978-3-031-02447-4_43.

[8] S. Challita, F. Zalila, and P. Merle, “Specifying Semantic Interoperability between Heterogeneous Cloud Resources with the FCLOUDS Formal Language,” *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 367–374, Jul. 2018, doi: 10.1109/cloud.2018.00053.

[9] J. Oliveira de Carvalho, F. Trinta, and D. Vieira, “PacificClouds: A Flexible MicroServices based Architecture for Interoperability in Multi-Cloud Environments,” *Proceedings of the 8th International Conference on Cloud Computing and Services Science*, pp. 448–455, 2018, doi: 10.5220/0006705604480455.

[10] C. Anglano, M. Canonico, and M. Guazzone, “EasyCloud: a Rule based Toolkit for Multi-platform Cloud/Edge Service Management,” *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 188–195, Apr. 2020, doi:10.1109/fmec49853.2020.9144821.

[11] C. Anglano, M. Canonico, and M. Guazzone, “EasyCloud: Multi-clouds made easy,” *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 526–531, Jul. 2021, doi:10.1109/compsac51774.2021.00078.

[12] D. Androcec and R. Picek, “Cloud ERP API Ontology,” *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–5, Jul. 2022, doi:10.1109/icecet55527.2022.9873020.

[13] N. Bassiliades, M. Symeonidis, P. Gouvas, E. Kontopoulos, G. Meditskos, and I. Vlahavas, “PaaSport semantic model: An ontology for a platform-as-a-service semantically interoperable marketplace,” *Data & Knowledge Engineering*, vol. 113, pp. 81–115, Jan. 2018, doi:10.1016/j.datak.2017.11.001.

[14] N. E. H. Bouzerzour, S. Ghazouani, and Y. Slimani, “Cloud interoperability based on a generic cloud service description: Mapping OWL-S to GCSD,” *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 70–75, Sep. 2020, doi:10.1109/wetice49692.2020.00022.

[15] N. E. H. Bouzerzour and Y. Slimani, “Towards a MaaS Service for Cloud Service Interoperability,” *Proceedings of the 10th International Conference on Model-Driven Engineering and Software Development*, pp. 72–83, 2022, doi: 10.5220/0010911400003119.

[16] G. G. Castañé, H. Xiong, D. Dong, and J. P. Morrison, “An ontology for heterogeneous resources management interoperability and HPC in the cloud,” *Future Generation Computer Systems*, vol. 88, pp. 373–384, Nov. 2018, doi: 10.1016/j.future.2018.05.086.

[17] B. Mane, A. Magalhaes, G. Quinteiro, R. Maciel, and D. Claro, “A Domain Specific Language to Provide Middleware for Interoperability among SaaS and DaaS/DBaaS through a Metamodel Approach,” *Proceedings of the 23rd International Conference on Enterprise Information Systems*, pp. 83–94, 2021, doi:10.5220/0010459600830094.

[18] A. Zeid, S. Sundaram, M. Moghaddam, S. Kamarthi, and T. Marion, “Interoperability in Smart Manufacturing: Research Challenges,” *Machines*, vol. 7, no. 2, p. 21, Apr. 2019, doi:10.3390/machines7020021.

[19] G. T. Ayem, S. G. Thandekkattu, and N. R. Vajjhala, “Review of Interoperability Issues Influencing Acceptance and Adoption of Cloud Computing Technology by Consumers,” *Intelligent Systems and Sustainable Computing*, pp. 49–58, 2022, doi: 10.1007/978-981-19-0011-2_5.

[20] B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions,” *Future Generation Computer Systems*, vol. 79, pp. 849–861, Feb. 2018, doi: 10.1016/j.future.2017.09.020.

[21] R. L. Grossman et al., “A Framework for the Interoperability of Cloud Platforms: Towards FAIR Data in SAFE Environments,” *Scientific Data*, vol. 11, no. 1, Feb. 2024, doi: 10.1038/s41597-024-03041-5.

[22] C. Ramalingam and P. Mohan, “An Efficient Applications Cloud Interoperability Framework Using I-Anfis,” *Symmetry*, vol. 13, no. 2, p. 268, Feb. 2021, doi: 10.3390/sym13020268.

[23] K. Kaur, DR. S. Sharma, and DR. K. S. Kahlon, “Interoperability and Portability Approaches in Inter-Connected Clouds,” *ACM Computing Surveys*, vol. 50, no. 4, pp. 1–40, Oct. 2017, doi: 10.1145/3092698.

[24] C. Ramalingam and P. Mohan, “Addressing Semantics Standards for Cloud Portability and Interoperability in Multi Cloud Environment,” *Symmetry*, vol. 13, no. 2, p. 317, Feb. 2021, doi:10.3390/sym13020317.

[25] G. Zacharewicz et al., “Model-based approaches for interoperability of next generation enterprise information systems: state of the art and future challenges,” *Information Systems and e-Business Management*, vol. 15, no. 2, pp. 229–256, May 2016, doi: 10.1007/s10257-016-0317-8.

[26] M. J. Heron, V. L. Hanson, and I. Ricketts, “Open Source and Accessibility: Advantages and Limitations,” *Journal of Interaction Science*, vol. 1, no. 1, p. 2, 2013, doi: 10.1186/2194-0827-1-2.

[27] H. Brabra, A. Mtibaa, L. Sliman, W. Gaaloul, and F. Gargouri, “Semantic Web Technologies in Cloud Computing: A Systematic Literature Review,” *2016 IEEE International Conference on Services Computing (SCC)*, pp. 744–751, Jun. 2016, doi:10.1109/scc.2016.102.

[28] Z. A. Adhoni and D. Lal, “Framework, semantic and standard approaches in multi-clouds to achieve interoperability: A survey,” *Journal of Integrated Science and Technology*, vol. 10, no. 2, 2022.

[29] E. Femi Aminu, I. O. Oyefolahan, M. Bashir Abdullahi, and M. T. Salaudeen, “A Review on Ontology Development Methodologies for

Developing Ontological Knowledge Representation Systems for various Domains*,” International Journal of Information Engineering and Electronic Business*, vol. 12, no. 2, pp. 28–39, Apr. 2020, doi:10.5815/ijieeb.2020.02.05.

[30] J. Agbaegbu, O. T. Arogundade, S. Misra, and R. Damaševičius, “Ontologies in Cloud Computing—Review and Future Directions,” *Future Internet*, vol. 13, no. 12, p. 302, Nov. 2021, doi:10.3390/fi13120302.

[31] Y. Serhane, A. Sekkaki, K. Benzidane, and M. Abid, “Cost Effective Cloud Storage Interoperability Between Public Cloud Platforms,” *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 12, no. 3, Apr. 2022, doi:10.17762/ijcnis.v12i3.4822.

[32] A. Patel and S. Jain, “Present and future of semantic web technologies: a research statement,” *International Journal of Computers and Applications*, vol. 43, no. 5, pp. 413–422, Jan. 2019, doi:10.1080/1206212x.2019.1570666.

[33] A. Rejeb et al., “Charting Past, Present, and Future Research in the Semantic Web and Interoperability,” *Future Internet*, vol. 14, no. 6, p. 161, May 2022, doi: 10.3390/fi14060161.

[34] I. Harrow et al., “Ontology mapping for semantically enabled applications,” *Drug Discovery Today*, vol. 24, no. 10, pp. 2068–2075, Oct. 2019, doi: 10.1016/j.drudis.2019.05.020.

[35] N. Noy and D. Mcguinness, “Ontology Development 101: A Guide to Creating Your First Ontology,” *Knowledge Systems Laboratory*, vol. 32, Jul. 2001.

[36] M. M. Al-Sayed, H. A. Hassan, and F. A. Omara, “Towards evaluation of cloud ontologies,” *Journal of Parallel and Distributed Computing*, vol. 126, pp. 82–106, Apr. 2019, doi: 10.1016/j.jpdc.2018.12.005.

[37] C. Reyes-Peña and M. Tovar-Vidal, “Ontology: Components and Evaluation, a Review,” *Research in Computing Science*, vol. 148, no. 3, pp. 257–265, Dec. 2019, doi: 10.13053/rcs-148-3-21.

[38] A. Hogan, “RDF Schema and Semantics,” *The Web of Data*, pp. 111–183, 2020, doi: 10.1007/978-3-030-51580-5_4.

[39] D. Allemang, J. Hendler, and F. Gandon, “Semantic Web for the Working Ontologist,” Jul. 2020, doi: 10.1145/3382097.

[40] BSI Group, “BS ISO/IEC 22123-1:2021 Information technology. Cloud computing - Vocabulary”, *Knowledge BSI Group,* 28 Feb 2021, doi:10.3403/30408969.

[41] M. Kostoska, M. Gusev, and S. Ristov, “An Overview of Cloud Interoperability,” *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, vol. 8, pp. 873–876, Oct. 2016, doi: 10.15439/2016f463.

[42] Z. A. Adhoni and N. Dayanand Lal, “Taxonomy for Classification of Cloud Service (Paas) Interoperability,” *Information Systems for Intelligent Systems*, pp. 69–75, 2023, doi:10.1007/978-981-19-7447-2_7.

[43] M. Bauer et al., “Towards Semantic Interoperability Standards based on Ontologies,” *AIOTI White Paper*, Oct. 2019, doi:10.13140/rg.2.2.26825.29282.

[44] E. B. Gürsel and A. Tarek, “Analysis Of Interoperability In Cloud Computing,” *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, pp. 189–192, Apr. 2019, doi:10.1145/3323933.3324089.

[45] B. D. Martino, A. Esposito, and G. Cretella, “From business process models to the cloud: a semantic approach,” *International Journal of High Performance Computing and Networking*, vol. 12, no. 4, p. 368, 2018, doi: 10.1504/ijhpcn.2018.096718.

[46] J. Alonso et al., “Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review,” *Journal of Cloud Computing*, vol. 12, no. 1, Jan. 2023, doi:10.1186/s13677-022-00367-6.

[47] G. Abbas, A. Mehmood, J. Lloret, M. S. Raza, and M. Ibrahim, “FIPA-based reference architecture for efficient discovery and selection of appropriate cloud service using cloud ontology,” *International Journal of Communication Systems*, vol. 33, no. 14, Jul. 2020, doi:10.1002/dac.4504.

[48] E. L. F. Ribeiro, M. Souza, and D. B. Claro, “MIDAS-OWL: An Ontology for Interoperability between Data and Service Cloud Layers,” *XVII Brazilian Symposium on Information Systems*, pp. 1–8, Jun. 2021, doi: 10.1145/3466933.3466953.

[49] A. Banu and A. Ameen, “Transforming Medical Data into Ontologies for Improving Semantic Interoperability,” *Emerging IT/ICT and AI Technologies Affecting Society*, pp. 71–85, Aug. 2022, doi:10.1007/978-981-19-2940-3_5.

[50] N. M. Hamdan and N. Admodisastro, “Towards a Reference Architecture for Semantic Interoperability in Multi-Cloud Platforms,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 12, 2023, doi:10.14569/ijacsa.2023.0141254.

[51] J. Soldatos, E. Troiano, P. Kranas, and A. Mamelli, “A Reference Architecture Model for Big Data Systems in the Finance Sector,” *Big Data and Artificial Intelligence in Digital Finance*, pp. 3–28, 2022, doi:10.1007/978-3-030-94590-9_1.

[52] F. Liu et al., “NIST cloud computing reference architecture,” *National Institute of Standards and Technology*, 2011. doi: 10.6028/nist.sp.500-292.

[53] L. Coyne et al., “IBM Private, Public, and Hybrid Cloud Storage Solutions”, 5th ed. IBM Redbooks, 2018.

[54] BSI Group, “BS ISO/IEC 22123-2:2023 Information technology. Cloud computing - Concepts,” *Knowledge BSI Group*, doi:10.3403/30408972u.

[55] R. Rosselló-Móra, “Taxonomy,” *Encyclopedia of Astrobiology*, pp. 1648–1649, 2011, doi: 10.1007/978-3-642-11274-4_1562.

[56] K. Aalijah, R. Irfan, U. Umar, and S. Nayab, “Automatic Taxonomy Generation and Incremental Evolution on Apache Spark Parallelization Framework,” *KIET Journal of Computing and Information Sciences*, vol. 5, no. 1, pp. 15–36, Feb. 2022, doi: 10.51153/kjcis.v5i1.83.

[57] N. Loutas, E. Kamateri, and K. Tarabanis, “A Semantic Interoperability Framework for Cloud Platform as a Service,” *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, pp. 280–287, Nov. 2011, doi: 10.1109/cloudcom.2011.45.

[58] T. Labidi, Z. Sakhrawi, A. Sellami, and A. Mtibaa, “An Ontology-Based Approach for Preventing Incompatibility Problems of Quality Requirements During Cloud SLA Establishment,” *Computational Collective Intelligence*, pp. 663–675, 2019, doi: 10.1007/978-3-030-28377-3_55.