Demand Response systems distributed on the cloud, offering a security & privacy framework for data flow

Ahmed Alobaidi[1], SeyedEbrahim Dashti[1], and Rawad Salah Hadi[2]

[1]Islamic Azad University Shiraz
[2]Islamic Azad University Jahrom Branch

March 31, 2023

Abstract

Demand Response (DR) is quickly becoming a critical component of the contemporary energy industry, notably in EU energy markets. As a result, substantial work has gone into standardizing demand response data models. As a result, an increasing number of demand response concepts are based on these standards. As a result, an increasing number of demand response concepts are based on these standards. These approaches, however, are often centralized, and those that rely on cloud solutions employ the cloud as a centralized data repository, assuming that the data is already homogenised when saved, i.e. all data has the same structure and type. In practice, however, DR plans rely on a number of components that deliver data in a variety of forms and types. Furthermore, the various DR standards establish models for various data formats, which impede data sharing between different DR systems. This article introduces CIM, a generic technology that allows current disaster recovery systems to disperse their components in the cloud while providing a robust security and privacy foundation for data interaction. Furthermore, the CIM includes a semantic interoperability layer capable of transforming data into a normalised form when transferred, allowing it to be consumed transparently by DR components. Experiments support the CIM as a solution for DR systems to decentralize their designs and share heterogeneous data with other DR systems that adhere to other DR standards.

SeyedEbrahim DashtiDepartment of Computer Engineering, Jahrom Branch, Islamic Azad University Sayed.dashty@gmail.com (* corresponding author)Ahmed Rahman AbdulzahraDepartment of Computer Engineering, shiraz Branch, Islamic Azad University Rowad Salah HadiDepartment of Computer Engineering, shiraz Branch, Islamic Azad University

ABSTRACT

Demand Response (DR) is quickly becoming a critical component of the contemporary energy industry, notably in EU energy markets. As a result, substantial work has gone into standardizing demand response data models. As a result, an increasing number of demand response concepts are based on these standards. As a result, an increasing number of demand response concepts are based on these standards. These approaches, however, are often centralized, and those that rely on cloud solutions employ the cloud as a centralized data repository, assuming that the data is already homogenised when saved, i.e. all data has the same structure and type. In practice, however, DR plans rely on a number of components that deliver data in a variety of forms and types. Furthermore, the various DR standards establish models for various data formats, which impede data sharing between different DR systems. This article introduces CIM, a generic technology that allows current disaster recovery systems to disperse their components in the cloud while providing a robust security and privacy foundation for data interaction. Furthermore, the CIM includes a semantic interoperability layer capable of transforming data into a normalised form when transferred, allowing it to

be consumed transparently by DR components. Experiments support the CIM as a solution for DR systems to decentralize their designs and share heterogeneous data with other DR systems that adhere to other DR standards.

Keywords: demand, cloud, security, data flow.

I. INTRODUCTION

Demand response (DR) programs have emerged as one of the most important ways for energy grid operators to reduce energy shortages or excesses and, as a result, enhance system dependability. 1 More crucially, with the increasing prevalence of Renewable Energy Sources (RES),2 Energy Storage Systems (ESS [1]), and even Electric Vehicles (EV3), extra flexible assets are easily accessible to provide new avenues for profitability, maximize current ones, and reduce overall risks. The economic benefits of this new era are not restricted to essential energy players such as grid operators and merchants. End-customers, or individual units, and the different types of Distributed Energy Resources (DERs) placed on their premises are, in fact, playing an increasingly important role, which may be linked to the development of new business models centered on aggregation and virtual power plants (VPP4). Aside from the aforementioned economic benefits, coordinating such a diverse landscape of DERs enables exploiting the underutilized flexibility accessible at lower scales. Indeed, residential and tertiary customers have been recognized as substantial sources of flexibility [2, 3], made even more so by the arrival of prosumers, or consumers who also create energy. For example, in 2016, the EU member states produced more than 33GW

of home solar photovoltaics (PV), 53% of which was transferred to the grid [3]. This is likely to rise as a result of the EU's strong green energy policy, which aim to attain a 32% proportion of RES by 2030. 5 However, because to the lack of a scalable Information and Communications Technology (ICT) infrastructure capable of managing the sheer bulk of small to medium-scale clients, disaster recovery (DR) programs are currently primarily supplied to big industrial customers. In this context, DR proposals are typically designed to work in a closed world in which new data sources are not expected to appear, and thus they do not consider the necessity of integrating new data sources that rely on different formats, models, or protocols to exchange and consume data with them. Numerous DR data models have been suggested, but only a handful are ontologies that allow for the provision of a semantic interoperable layer for data sharing [4]. Similarly, proposals rely on non-semantic models [5-12], while there has recently been a movement toward building proposals using ontologies [13-18]. However, the majority of these ideas lack natural means for integrating additional data sources, necessitating a significant data harmonisation effort to incorporate new data sources. Furthermore, the DR plans rely on a diverse set of protocols, some of which need infrastructures to openly give their data (HTTP) or broadcast their information in low-security environments (MQTT); in fact, security is a component that is frequently overlooked or ignored in most proposals. The CIM middleware is introduced in this article to handle the practical issues that arise in real- world DR systems. The CIM's major purpose is to create a private and secure peer-to-peer cloud network enabling disaster recovery systems and data infrastructures to transparently interchange and consume data, despite the fact that the systems and infrastructures use various formats or models. To that purpose, the CIM employs semantic interoperability modules, which enable bidirectional data translation mechanisms to convert data represented in disparate forms or models into a semantically compatible version based on RDF

[19] and an ontology [20], and vice versa. The CIM tool was created in the framework of the European DELTA project, in which semantic interoperable data adheres to the DELTA ontology [21], although it may be used with any ontology. Furthermore, anytime a data payload is transferred, the CIM performs semantic validation of data based on W3C SHACL shapes [22], verifying its validity and coherence with the ontology. The CIM is an Open Source tool6 that implements its semantic interoperability layer [23] using well-established IoT techniques that have been adapted to the DR and the decentralized cloud context. However, the CIM may be utilized in a variety of application domains; it is not limited to ontologies relating to the energy sector. The CIM has been utilized as middleware in the context of DELTA for communications

across DR systems and data infrastructure using an edge-cloud architecture. The CIM, on the other hand, goes beyond encouraging the decentralization of DR systems and the integration of distributed data sources by providing them with a distributed, scalable, secure, and end-to-end privacy- preserving peer-to-peer (P2P) network that can be hosted across multiple cloud providers and ensures service liveness even in the face of failures. The CIM enables the unification of discrete real-world corporate DR systems without needing them to disclose any data through public channels. A careful experiment was put out to demonstrate the features of the CIM. First, the accuracy of the CIMs' semantic compatibility has been established. The capacity of the CIMs to respond to concurrent requests while exchanging ordinary payloads or payloads that must be translated in order to be semantically compatible. Finally, the ability of CIMs to exchange larger payloads has been proven.

The remainder of this article is organized as follows: Section 2 analyzes current DR proposals in the literature, Section 3 elicits practical challenges that DR proposals must address, Section 4 reports the CIM tool, providing an insight into its architecture and functionalities, Section 5 explains the experiments performed to validate the CIM, as well as the results obtained, and Section 6 summarizes our findings and conclusions.

Related work

This section is broken into two parts: the first provides an overview of existing data models and standards used in the context of disaster recovery (Section 2.1), and the second reviews current DR systems in terms of security (Section 2.2) and semantic interoperability features (Section 2.3). (Section 2.3). Table 1 summarizes the various characteristics of the DR systems studied.

DR data models and standards

Several DR efforts have been presented that provide support for various features of data in terms of formats and models in the DR domain. Some standards have been developed that give XML data scheme models. Energy Market Information Exchange (eMIX [24]) is a pricing and product standard. The Universal Smart Energy Framework (USEF [25]) encourages the commoditization of flexible energy consumption through flexible markets. The Ope- nADR standard [26] is another well-known endeavor that conceptualizes DR through a data and communication definition.

Other DR attempts, such as EN 50090-1 [27] and the IEC family of standards, provide generic data models in UML without requiring a specific data format (CIM [28], 62056 COSEM [29], 62746 [30]). Furthermore, the Smart Grid Architecture Model (SGAM [31]) outlines the architectural design of smart grid use cases, with five levels representing business objectives and processes, functions, information exchange and models, communication protocols, and components.

Finally, several DR projects give data models in the form of ontologies. These ontologies are mostly concerned with modeling energy-related data, while some, such as the OpenADR ontology [4] and the DELTA ontology [21], also deal with DR. On the one hand, several of these ontologies are concerned with modeling measurements (OntoEnergy [32]) or measurements and equipment (CIM ONTOLOGY,7 MAS2TERING [33], ThinkHome [34]). There are, on the other hand, ontologies that include other topics, such as events (MI- RABEL [35]) or geolocation (BOnSAI [36], SEMANCO,8 SESAME9).

Finally, other ontologies such as SAREF4ENER,10 MAS2TERING [37], EEPSA [38], RESPOND,11 SARGON [39], and OEMA12 are intended for stakeholders involved in smart grids and energy-related enterprises.

As a result, the DR has a diverse set of standards that rely on various types of data models and accompanying data formats, resulting in a heterogeneous data environment. When data is transmitted between DR systems that adhere to various standards, this circumstance displays an obvious technological barrier. This challenge is exacerbated by the fact that not all standards can model the same type of data, and hence relying on several standards may become necessary. To address this issue, a semantically compatible DR solution

that enables transparent data consumption independent of the model and format in which it is expressed is required.

Table 1 Comparison of existing DR proposals.

Proposal	Data exchange					Semantic	Interop.
	Architecture	Protocol	Security	Format	Model	Uplift	Downlif
Hossein et al. [5]	Edge–cloud	Undefined	Undefined	Undefined	Undefined	–	–
Wang et al. [6]	PPP	HTTP	Undefined	Custom	Custom	–	–
Deng et al. [7]	PPP	HTTP	Undefined	Custom	Custom	–	–
Chen et al. [8]	PPP	HTTP	Undefined	Custom	Custom	–	–
Kaur et al. [9]	Edge–cloud	Software defined networking	Undefined	Custom	Custom	–	–
Zhang et al. [10]	Edge–cloud	Undefined	Undefined	Tabular	Custom	–	–
Frincu and Draghici [11]	PPP	HTTP	Undefined	Tuple	Custom	–	–
Galkin et al. [12]	PPP	OCPP, IEC61850, XMPP	Undefined	XML, JSON	OpenADR, OCPP, GOOSE	–	–
Kim et al. [16]	Pub/Sub	MQTT	Undefined	Any	Any	–	–
Zhou et al. [15]	PPP	HTTP	Undefined	RDF	Custom		
MAS2TERING [33]	PPP	FIPA [17]	Undefined	RDF	MAS2TERING1		
CoSSMic [13]	PPP	HTTP	None	RDF	SEAS, SOSA		
RESPOND [14]	Pub/Sub	MQTT	Undefined	RDF	RESPOND		
SHAR-Q	P2P	HTTP/XMPP	TLS, SASL	RDF	SHAR-Q		
Wicaksono et al. [18]	PPP	HTTP(s)	Undefined	RDF	Any		
CIM	P2P	HTTP/XMPP	JWT, TLS, SASL	RDF	Any		

Data exchange and security in DR

DR may be carried out using several data interchange architectures that employ various protocols and security methods. Many approaches rely on point-to-point (PPP) designs based on HTTP [6- 8,11,13,15,18,40] or other protocols [12]. In this type of architecture, exchanging data needs knowing ahead of time which endpoints to share data with, which normally necessitates discovery capabilities and characterizing the systems in order to make them discoverable. There are various security mechanisms for point-to-point architectures, which impedes interoperability across systems that must know not just the endpoints, but also

the security mechanisms that they implement and then support.

Some DR systems rely on publish/subscribe (Pub/Sub) architectures, which are implemented with MQTT, to tackle the discovery problem. Data is exchanged through a broker in these systems under a topic where a client may post data and others can subscribe. MQTT, on the other hand, has known security concerns that may limit its applicability in real-world circumstances where security and privacy are critical.

Finally, additional disaster recovery systems depend on peer-to-peer (P2P) or edge-cloud designs that use a range of protocols. Despite the fact that there are several security techniques accessible for these architectures, none of the previous ideas based on edge-cloud address which is preferable [5,9,10]. Only SHAR-Q, which is based on peer-to-peer communication, defines the use of SASL in an XMPP cloud. It is worth noting that these designs do not have the discovery concerns that the point-to-point architecture has.

When transferring data, the CIM provides two degrees of protection. JWT tokens are used for communication between the CIM and the local infrastructure, although any authentication method might be used. CIMs communicate with one another in a peer-to-peer architecture using an XMPP cloud. To connect to the cloud, a CIM requires a set of credentials in the form of a certificate (SASL); moreover, the CIMs utilize a distinct certificate to encrypt communications (TLS). In addition, the CIMs use a white access control list system, which requires nodes from the XMPP network to be defined in order to communicate data.

Semantic interoperability and data validation in DR

Interoperability is defined as two information systems' capacity to share and consume data in a transparent manner [41]. This interoperability is known as semantic interoperability when the data being shared is represented using Semantic Web technologies. To that goal, semantic interoperable systems agree on the use of RDF data presented according to a specified ontology. One of the primary benefits of adopting ontologies is that RDF data may be consumed by systems that rely on distinct ontologies, as long as these systems follow a set of equivalence criteria between these ontologies [20]. Instead, when a system is not RDF-based, a data translation is required to translate from a heterogeneous format and model into a semantic interoperable version (uplift); and vice versa, in order for the system to receive a semantic interoperable payload and translate it into an understandable format and model (downlift).

There are various DR approaches that do not rely on ontologies and so do not provide semantic interoperability. Hossein et al. [5] concentrate on the protocol plane and how their solution outperforms previous DR protocols in terms of data exchange performance. Their concept is not tied to any certain data format or paradigm.

The proposal by Wang et al. [6] focuses on executing DR in virtual machines hosted on cloud services such as Amazon. Their approach examines demand response needs at tenants' infrastructure and, as a consequence, reduces the number of virtual machines needed. Although their proposal contains tenant infrastructures that can be extremely varied, it is built on a bespoke model for data interchange, therefore integrating new infrastructures needs a developer to convert them to be compatible with the proposal model.

Deng et al. [7] offer a cloud-based method to maximize profitability in a tailored DR system. Their proposal specifies a specific structure and model for the algorithm to use while performing computations on the cloud. As a consequence, orders are delivered to the client's location.

Chen et al. [8] propose a cloud DR system for electric cars that is based on bespoke DR signals that adhere to a certain model and format. Kaur et al. [9] describe a similar DR method for electric cars. The concept also makes use of a proprietary data model and data format for DR signals, which are utilized to communicate with cars and other stakeholders.

Zhang et al. [10] offer a method for DR training and application of a reinforcement learning algorithm. To that purpose, the authors choose for an edge-cloud architecture, however they do not identify the protocol employed. The edge nodes in this architecture supply data from various sensors, and the algorithm is taught

and employed at the cloud level. The data is presented in a tabular style, and the model was created on the fly for this project.

The DR system proposed by Francu and Draghici [11] is based on cloud services that gather data from certain smart home sensors.

These sensors provide data to the cloud, where it is stored as a tuple, with each location indicating the measurement of a distinct sensor. The DR activities are then calculated on the cloud level, and commands are delivered back to the smart home actuators.

Galkin et al. [12] offer an architecture for protocol layer interoperability. Their idea focuses on modifying communication depending on several protocols (IEC 61850 GOOSE, OpenADR, OCPP and UDP). However, the authors consider how to expand their idea to create an automated translation layer to adjust heterogeneous data at the aggregator level, but they do not give such automatic translation tools.

Finally, Kim et al. [16] present a thorough examination of the advantages of employing publish/subscribe and topic-based group designs in DR rather than master-slave architectures. Their approach does not emphasize the use of a single model or format, nor does it address compatibility.

It should be noted that the aforementioned approaches must deal with heterogeneous systems for which no interoperability method is provided. Furthermore, the fact that the majority of these approaches establish a specific data model or even employ a custom format severely limits the interoperability of these DR systems when incorporating new infrastructures as data sources or interfacing with other current DR systems. On the contrary, several DR systems have embraced ontologies and standards, which facilitates interoperability with other systems or infrastructures that use the same ontologies or standards.

Zhou et al. [15] propose an ontology-based DR system for electric cars. In this system, a set of existing systems provide data that is compliant with the ontology; if new systems are included, they must natively

support data expressed in the custom ontology; i.e., the system does not provide generic mechanisms for translating heterogeneous data into semantically interoperable data (uplift).

Similarly, MAS2TERING [33] provides a semantically compatible DR system with other systems based on various ontologies. The MAS2TERING system is built on the MAS2TERING ontology [37], which incorporates many standard ontologies to enable interoperability. MAS2TERING, on the other hand, lacks tools for dealing with diverse data (uplift).

COSSMic [13] provides a DR system that merges smart house consumption data with meteorological data, both in CSV format. The proposal proposes an ad hoc system for translating these data files into RDF, which is then published on the Web for consumption.

Similarly, RESPOND [14] uses an ad hoc approach to convert data from many data sources into RDF since these sources are known ahead of time. The data is then kept in a third service, where tools and services are planned to give measurement-driven recommendations to end users for energy demand reduction and impact their behavior. Furthermore, end users and stakeholders are constantly informed via a mobile app [42].

Finally, two DR systems provide an uplift mechanism based on adapters that execute heterogeneous data source translation into semantically interoperable data. Wicaksono et al. [18] combine a wide range of data sources, creating a semantically compatible layer on top of which ML algorithms may be fed. SHAR-Q13 combines data from many data sources and offers a semantic interoperable layer on top of which value- added services are deployed; they employ data from prior sources to create value in various forms (ML predictions, marketplaces, etc.).

It is worth noting that the majority of ideas in the literature do not examine the translation of heterogeneous data from disparate sources into semantically interoperable data (uplift), and none really consider the reverse operation (downlift). Furthermore, verifying data that has been transmitted is a critical problem to guarantee

that the data is not only compliant with the ontology, but also correct and valid (e.g., a DR signal does not increase the load above certain dangerous thresholds).

The CIM includes methods for both uplift and downlift, as well as a bidirectional translation mechanism. It should be noted that this is critical in order for non-ontology-based systems to consume the data being transmitted. Furthermore, as previously stated, all communications take place in a secure XMPP network. These are, to the best of the writers' knowledge, unique and original elements of the CIM.

Challenges

The CIM was designed and implemented to address a number of practical challenges derived from the authors' collective experiences participating, on the one hand, in research projects of various TRLs and, on the other, in private collaborations with real-world stakeholders spanning multiple application domains. Figure 1 depicts a high-level view of the CIM and its capabilities. The difficulties and how the CIM addresses them are presented in the following subsections: decentralising DR systems, security and privacy, data validation, and semantic interoperability.

Fig. 1. CIM architecture.

Decentralising DR systems

DR systems rely on many components, which can be broadly classed as services or devices. The former often rely on data supplied by the latter to calculate some findings that are turned into actions to be conducted, such as anticipating the load of an infrastructure or providing DR signals for the components. Devices are typically, but not always, IoT devices that gather client data and conduct actions on the energy infrastructure; for example, when a DR signal is received, a device's behavior can be modified.

Because of scalability concerns, the rising prevalence of IoT devices, which are important data sources in this context, renders DR designs that rely on centralized servers for data collection unsuitable in practice. Furthermore, in modern systems, devices play a more active role, as they must respond to signals received by other architectural components, such as DERs receiving control signals from utilities.

As IoT devices do not have public endpoints, there is a clear demand for the construction of a distributed and scalable communication layer that supports duplex message exchanges. Furthermore, the communication layer must offer service liveness, which simply means that it is fault-tolerant to failures such as servers failing. Finally, the communication layer should provide enterprise-grade security while simultaneously adhering to data protection standards such as GDPR.

Cloud computing enables a diverse set of services, meeting the needs of both devices and services [43]. It enables the distribution of devices or services among large groups of networked distant servers, using the processing power required by these components in centralized designs. Because a huge quantity of data is exchanged in the context of DR, the ideal cloud solution is the use of networks based on the eXtensible Messaging and Presence Protocol (XMPP protocol) [44].

The fundamental benefit of using XMPP as the default communication protocol is that it is a well- established and standardised protocol developed for real-time data streams, with several open-source apps for clients and servers and support for a variety of operating systems [45]. Furthermore, cloud networks based on this protocol have various security features in place to secure communications, including as SASL for authentication and TLS for data encryption.

However, transitioning from a centralized to a decentralized architecture based on XMPP networks is not an easy operation. Adopting this strategy necessitates the extension of centralized DR systems' technical

stacks in order to first construct the XMPP network and, second, exchange data across the network. To do this, the CIM may be used as a middleware to exchange data across an XMPP cloud. Existing systems do

not need to adopt a new technical stack when using CIM.

Security and privacy

For data consumption, most DR ideas rely on the cloud. These approaches have numerous disadvantages in terms of security and privacy. On the one hand, the various components of a DR system must publicly expose their data exchange end points, which exposes possible access opportunities for attackers. Furthermore, because the cloud is employed as a vast data repository, the data contained therein is also accessible. However, because all data is housed in a third-party data store that is not the original system that created the data, the data owner also becomes the platform where it is kept. Furthermore, because all data is centralized, it becomes impossible to maintain the data.

Decentralized XMPP networks, on the other hand, provide numerous security levels for both joining the network and transferring data. Furthermore, because the components that engage in such networks are hidden behind XMPP clients, the only method to communicate with them is through a nonpublic network. As a result, these networks are an excellent alternative for decentralizing disaster recovery systems that automatically become more secure against external attackers.

Different XMPP clients may have their own decentralized access control list policies in order to protect their privacy, while XMPP servers can cluster the clients and establish extra access control list policies for transferring data. Furthermore, any data sent via the XMPP network may be encrypted end-to-end.

The CIM enables for the connection to an XMPP cloud, and to join the CIMs, an SASL certificate provided by a network administrator is required. Furthermore, the CIM encrypts all communication over the XMPP network with a TLS certificate. As an extra privacy layer, the CIM employs a white access control list (ACL) technique to prohibit requests from other CIMs in the XMPP cloud from being handled.

Finally, another critical security issue is the CIM's connectivity with local data sources, namely the DR components. Although this communication takes place in a trusted local infrastructure, the CIM requires these infrastructures to employ an authentication method to engage with it, enhancing security not just in the cloud but also in the local infrastructure.

Semantic interoperability

As previously stated, semantic interoperability is defined as systems' ability to transparently exchange data and have a shared understanding of it [41], which results in the ability to transparently consume the transferred data. Once the data interchange layer, in this example XMPP, has been established, the common understanding must be established. A frequent way to this purpose is to develop a common ontology and communicate data represented in accordance with it [23]. The usage of ontologies provides several benefits in terms of interoperability; however, the data shared must be in RDF format.

There are several information sources in the context of DR, ranging from devices (IoT) to data given via ad hoc DR components and/or databases. The heterogeneity of the DR standards landscape has already been described in the related work; however, from a practical standpoint, this heterogeneity is even greater due to the various devices that rely on a wide range of formats and models established by their vendors, which are not even DR standards.

An active challenge is to have non-RDF data sources (DR systems, IoT devices, etc.) and provide a transparent mechanism for translating their data into an equivalent RDF version modelled according to a common ontology so that the data is understandable by other actors involved in the data exchange who expect data to be in RDF using the agreed ontology. Uplift refers to the process of converting data on the fly into an equivalent RDF version.

Furthermore, there is the inverse issue. When data is re-queried, a semantically compatible version is returned. However, because DR systems rely on diverse formats and models, the interoperable data must be converted back into a different format and model that the system supports. Downlift refers to the process of converting data from RDF to an equivalent non-RDF form.

To that end, the CIM includes Uplifting and Downlifting mechanisms, which allow DR systems to translate their data before sending it to the XMPP cloud so that third-party entities can understand it, as well as adapt data from the XMPP network to their own formats and models so that it can be consumed transparently.

Data validation

Data in real-world systems is rarely limited within the tight confines in which it is recorded. As previously said, contemporary systems require interactions among different components that share data in order to execute various functions. When a data payload is received, the initial responsibility of each component is to validate the message received.

Validation is classified into two types: (1) syntactic validation, which enforces the right syntax of the data (e.g., assuring the correct JSON-LD syntax14); and (2) semantic validation, which validates that the data is consistent and meets a set of requirements.

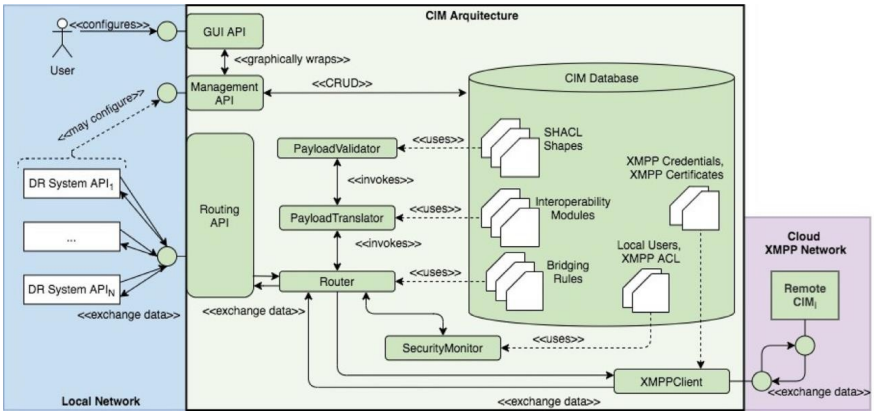

Fig. 2. CIM architecture.

Data validation is often not handled at the network level in cloud systems since it is more likely to be spread by network clients. Furthermore, because to a lack of ontologies and semantic technologies connected to the DR domain, these clients simply provide syntactic validation, which ensures that the data syntax is proper. Semantic validation, on the other hand, detects discrepancies in the data and guarantees that the data contains only relevant information.

The lack of semantic validation in data interchange reduces the data's dependability. To solve this issue, whenever a payload is transferred, the CIM conducts semantic validation using SHACL shapes [22]. This validation allows developers to describe not only the structure of the payload but also the limitations that the data must meet and validate them without requiring specific interventions or code. It should be noted that the SHACL shapes language is a W3C standard.

The CIM tool

The CIM was meant to be a general tool that could be implemented with any specific collection of technologies; its architecture is seen in Fig. 2. Its primary functions are mentioned below.

CIM Configuration: The CIM has two primary components for configuring its features: the GUI API and the Management API. Both components publish two APIs on a DR system's local network; however, the GUI API is merely a graphical interface for users to engage with the Management API. As a result, the second API is the one that provides the primary configuration features. Several items may be created, read, updated, or deleted (CRUD) using this API, including:

Bridging Rules: These are user-supplied rules that translate XMPP queries to local APIs and vice versa. Furthermore, when the payloads transferred via the specified APIs need to be translated, these rules define the name of an interoperability module.

-A sample Bridging Rule is {"XMPP API":"/dr/reports", "local_ API":"http://localhost:900/reports/power", "method": "GET", "module":"oadrReport.module"}, entailing that any request received through the XMPP network to /dr/reports will be sent to the local API localhost:900/reports/power, and the payload provided by such endpoint will be translated with the oadrReport.module.

Interoperability modules: These modules include translation mappings that may be used with any RDF serialisation approach [46,47]. The mappings include rules for converting data from various forms to RDF and vice versa. Any interoperability module must convert non-RDF data into RDF serialisation JSON-LD specified using an ontology.

SHACL shapes : is a collection of RDF documents containing SHACL shapes that are used to perform semantic and syntactic validation on the exchanged payloads. These shapes guarantee that all data, whether translated or not, is represented in accordance with the ontology used for data interchange with CIMs.

-Local users: When given, the CIM permits to attach JWT tokens [48] that must be utilized by the local APIs to deliver data over the CIM.

XMPP ACL: The XMPP access control list (ACL) is a list of CIMs to which the current CIM is authorized to transmit data, as well as a list of CIMs to which the local CIM is allowed to send data.

-XMPP credentials: These are the XMPP username and password used to access the XMPP network.

-XMPP certificates: The CIM is protected by two certificates: an X.509 encryption certificate and a mutual authentication certificate. The former encrypts all data transferred, whereas the latter establishes double-authenticated channels with other CIMs before exchanging data. The XMPP network administrator must offer these certificates.

-XMPP connection: The CIM relies on the XMPPClient component to connect with an XMPP server and join an XMPP network. This component has two aims. On the one hand, it connects and authenticates in an XMPP network using XMPP credentials and an X.509 mutual authentication certificate within the XMPP certificates. This component, on the other hand, enables the Router to send and receive requests with other CIMs connected to the XMPP cloud network, which encrypts all data transmitted using an X.509 encryption certificate within the XMPP certificates.

-Semantic interoperability: The PayloadTranslator, which is in charge of homogenizing the payloads to be transferred when necessary, implements the CIM's semantic interoperability. This component is called by the Router, which gives the payload as well as the name of the Interoperability Module that must be used to translate it. The PayloadTranslator then fetches such an Interoperability Module and translates it. Finally, regardless of whether translation happened, the PayloadTranslator invokes the PayloadValidator and provides the payload (the translated or the original if no translation was required) to the Router, along with the validation report produced by the PayloadValidator.

-Semantic validation: The PayloadValidator is the CIM component in charge of this validation. It is important to note that semantically verifying a payload also requires syntactic checking. The PayloadTranslator, which offers a JSON-LD payload, calls this component. The PayloadValidator then validates the payload using SHACL shapes and delivers a validation report to the PayloadTranslator.

-Privacy and security: The SecurityMonitor handles privacy and security in CIMs; however, keep in mind that the XMPPClient also implements privacy and security features. The Router activates this component and supplies the distant CIM XMPP username whenever a request must be delivered across the XMPP network or is received by such network. The SecurityMonitor then examines the CIM XMPP username in the XMPP ACL to see if the local CIM is authorized to send data to the remote CIM or if the local CIM

is allowed to receive data from the remote CIM. As a result, this component returns to the Router a status code indicating whether or not the request may be executed.

When a local API, on the other hand, makes a request, the Router activates the SecurityMonitor. In certain circumstances, this component determines if the request contains a JWT token and whether the token is genuine, implying that it has not expired and is associated with a local user. Similarly, the SecurityMonitor sends a status code to the Router indicating whether the request may be handled or is not authorized.

Finally, XMPP networks have additional security layers for data exchange, such as Transport Layer Security (TLS) for transport security,15 Simple Authentication and Security Layer (SASL) for authentication16 and End-to-End signing and object encryption for the extensible messaging and presence protocol. 17 XMPP networks, in example, enable you to organize various CIMs into clusters and then establish access control list restrictions on the clusters.

-CIM data exchange: The data interchange occurs in the CIM, which includes various components that execute complicated activities as previously mentioned. The Router component handles data interchange in the CIM; it must be distinguished whether a local API from the DR system delivers a request to the XMPP network or when a request is received from the XMPP network.

When the APIs of a DR system on the local network want to transmit a request over the XMPP network, they submit it to the Routing API. The request must include the distant CIM XMPP username from whom the response is expected, as well as a valid JWT token previously issued by the CIM.

The Routing API then routes this request to the Router component. The Router calls the PayloadTranslator, which returns a translation of the payload being transferred and a validation report, if necessary. If the

payload is invalid, the Router responds with an invalid status code using the Routing API. If the payload is valid, the Router validates whether the request is legitimate by invoking the SecurityMontior, and it provides an unauthorised response code to the API through the Routing API. If the request is legitimate, it is packaged as an XMPP request and sent to the XMPPClient. Finally, the XMPPClient routes the XMPP request across the XMPP network to the destination CIM, which processes the request and responds.

When a request arrives over the XMPP network, the XMPPClient accepts it and transmits it to the Router component, which unpacks the XMPP request into a local request. The Router then uses the SecurityMonitor to confirm that the request is legitimate, and if it is not, it returns an unauthorised response code to the remote CIM through the XMPPClient. Otherwise, the Router executes the PayloadTranslator, and if the given payload is invalid, the Router returns an invalid response code to the remote CIM. Finally, assuming the request was legitimate and authorized, the Router routes the local request through the Routing API to an existing API.

Decentralisation and integration of DR systems

One of the CIM's primary aims is to enable DR systems that are centralized in a local infrastructure to distribute their components in an XMPP network and communicate data with other DR systems. There are various advantages of using the CIM to decentralized DR systems: (A) Distributing DR components across infrastructures utilizes the processing power required by these components; (B) Security in XMPP data sharing using an encryption mechanism and JWT tokens in local data exchange; (C) Decentralisation improves privacy because data access is controlled locally by the different CIMs based on their ACL, making it easier to address issues such as GDPR; (D) The DR system architecture becomes modular, making it easier to extend and scale these components; and (E) Integrating components from different DR systems requires only the development of Interoperability Modules as additional effort. It should be noted that the CIM-enabled advantages address the issues raised in Section 3.

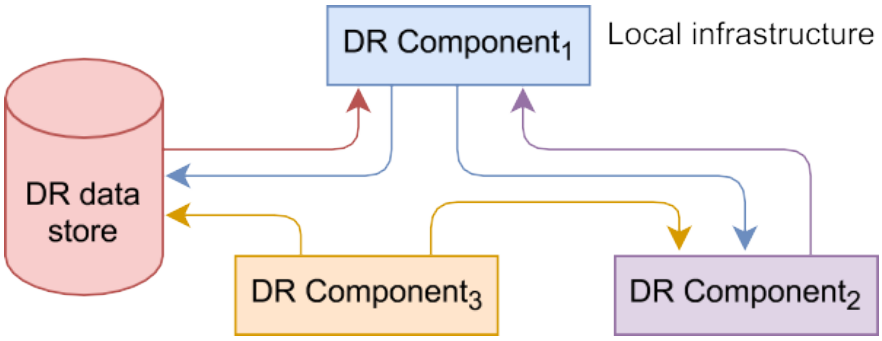

Fig. 3. A centralised DR system.

Assume the DR system described in Figure 3. It is obvious that a local infrastructure is required to supply processing power for several components; moreover, there is no encryption between the communication and privacy methods. The various components communicate data over public endpoints, which, despite considerable protection, are vulnerable to attackers. Also, keep in mind that adding new components or scaling existing ones may necessitate more labor and processing capacity Finally, communicating with a different DR system is not possible in this architecture because there is no semantic interoperability layer to translate the payloads, and this other DR system may use different communication protocols, complicating communications even if both DR systems use the same standard.

The DR system depicted in Fig. 3 may be decentralised in various local infrastructures using the CIM, as depicted in Fig. 4. The processing power required by each DR component in this new design is also dispersed among several local infrastructures, making the overall DR system more efficient.

Fig. 4. A decentralised DR system using the CIM.

Furthermore, all DR components are only accessible via the XMPP network. Joining this network necessitates the creation of authentic XMPP credentials and certificates by the XMPP network administrator. As a result, the various components are better shielded from possible threats. Furthermore, all conversations are encrypted from beginning to end. It is also worth noting that communication between the CIM and the local DR components necessitates the use of a JWT token to authenticate the requests. Although this communication does not necessitate a robust security strategy because local infrastructures are not publicly accessible and are also trustworthy. The CIM's present implementation relies on JWT authentication; however, this might be changed by alternative methods such as OAuth or basic authentication.

In terms of data protection, the CIMs have their own access control list. This implies that even if an attacker could join the network, the attacker's CIM would be unable to share data owing to the CIMs' distributed access control lists. Furthermore, the XMPP server may set access control list policies across CIM groups.

The design of the DR system represented in Fig. 4 is modular, and so its components may be scaled. When a component has a high workload, it might be duplicated in the XMPP network to balance the former component's total workload. Furthermore, adding new components just necessitates deploying them through the CIM and configuring the necessary security and privacy parameters of the other CIMs.

From the standpoint of semantic interoperability, integrating a component that follows a different DR standard, and thus has a different format and model, requires only the development of the necessary Interoperability Modules to translate the payloads that are expected to be sent and those that are expected to be received by such a component via the CIM.

As a result, the CIM allows DR systems to exchange data with components designed according to multiple DR standards. Furthermore, the CIM provides a good security framework for exchanging data with the CIM

locally using JWT tokens or remotely via the XMPP network utilizing authentication mechanisms, end-to-end encryption, certificates, and the access control lists that each CIM maintains.

Security in P2P networks is determined by whether the network is centralised, hybridised, or decentralized. A centralised or hybrid network's security provides a single point of failure: the centralised servers. An assault on one of these servers might compromise the security of the entire network. A rogue node in a decentralized P2P network can corrupt a portion of the network, but it is doubtful that a single bad node could control the entire network. As a result, decentralized networks are less vulnerable to assaults than centralised or hybrid networks, although these latter two types of networks are more suited to monitoring,

making attack detection and network recovery easier. The following are examples of attacks that can be launched against a P2P network.

Attack against eavesdropping. Created at the network layer. By collecting tiny packets from the network, attackers can obtain access to data and eavesdrop on communication. TLS, as specified in the standard, is used to protect the stream from eavesdropping.

Sybil assault. Consists of generating a huge number of bogus identities and utilizing them to gain significant influence in the network, causing disruption or preparing for future assaults. Configuring TLS and SASL, as specified in the standard, helps to secure the client's server from direct assault or identification by third parties.

Attacks on buffer overflow. The attacker overwrites memory components, altering network functioning and potentially destroying or exposing data. As stated in the standard, utilizing base64 in SASL helps to prevent against buffer overflow attacks and other implementation-based vulnerabilities.

A denial of service attack has occurred. The most typical DoS attack is a single node flooding the network with fake packets, which prevents or slows network activity. When two or more nodes are participating in an assault, it is referred to as distributed DoS. The XML stanzas18, as indicated in the standard, assist defend the client's server against DDoS assaults after TLS and SASL are implemented.

Other attack kinds are conceivable. Certain KPIs can be configured to minimize these assaults. 19 The following are the most important KPIs:

Keep track of the overall number of requests. Examine how many requests are being processed on the network.

Keep an eye on the nodes. Determine the number of nodes in the network, whether it drops or rises, and whether the quantity is adequate.

Requests should be monitored per node. Monitor how many requests each node receives and sends: whether some nodes send fraudulent requests or whether a node behaves differently at various times.

Keep track of the requests by IP address. Keep track of how many requests are received and issued by each IP address, and use this information to determine their geographical location.

The Node software. Examine which software versions are being utilized by the nodes and whether these versions have known security flaws.

These KPIs may be seen in the CIMs since they are linked to an XMPP broker that offers them. As a consequence, they may be watched for prospective assaults.

Experiments

Three tests were designed and carried out to verify the CIM tool. The first experiment involves confirming that the semantic interoperability layer supplied by the CIM offers accurate payload translations. The second step is to examine the CIM efficiency when the number of parallel requests is increased without affecting

the size of the payloads being transferred. The third experiment examines the CIM efficiency when the size of the payload being transferred is increased. All of the findings gathered and presented are the mean of the results obtained after repeating each experiment ten times. The JSON-LD payloads are expressed using the DELTA ontology 20. All of the above studies were carried out on two computers that had the following characteristics: Intel E5-2678v3 processor, 16GB RAM, and 500GB storage. In addition, an

OpenFire XMPP broker was installed on a server with two Intel E5-2680v3 processors, 32 GB of RAM, and one terabyte of storage.

The varied request flows employed throughout the studies are depicted in Figs. 5 and 6. It is worth noting that both GET and POST requests are evaluated, as well as JSON-LD payloads expressed using the DELTA ontology and XML payloads represented using the OpenADR model. All payloads shared and utilized in the various experiments may be found in the experimentation folder of the CIM repository. 21

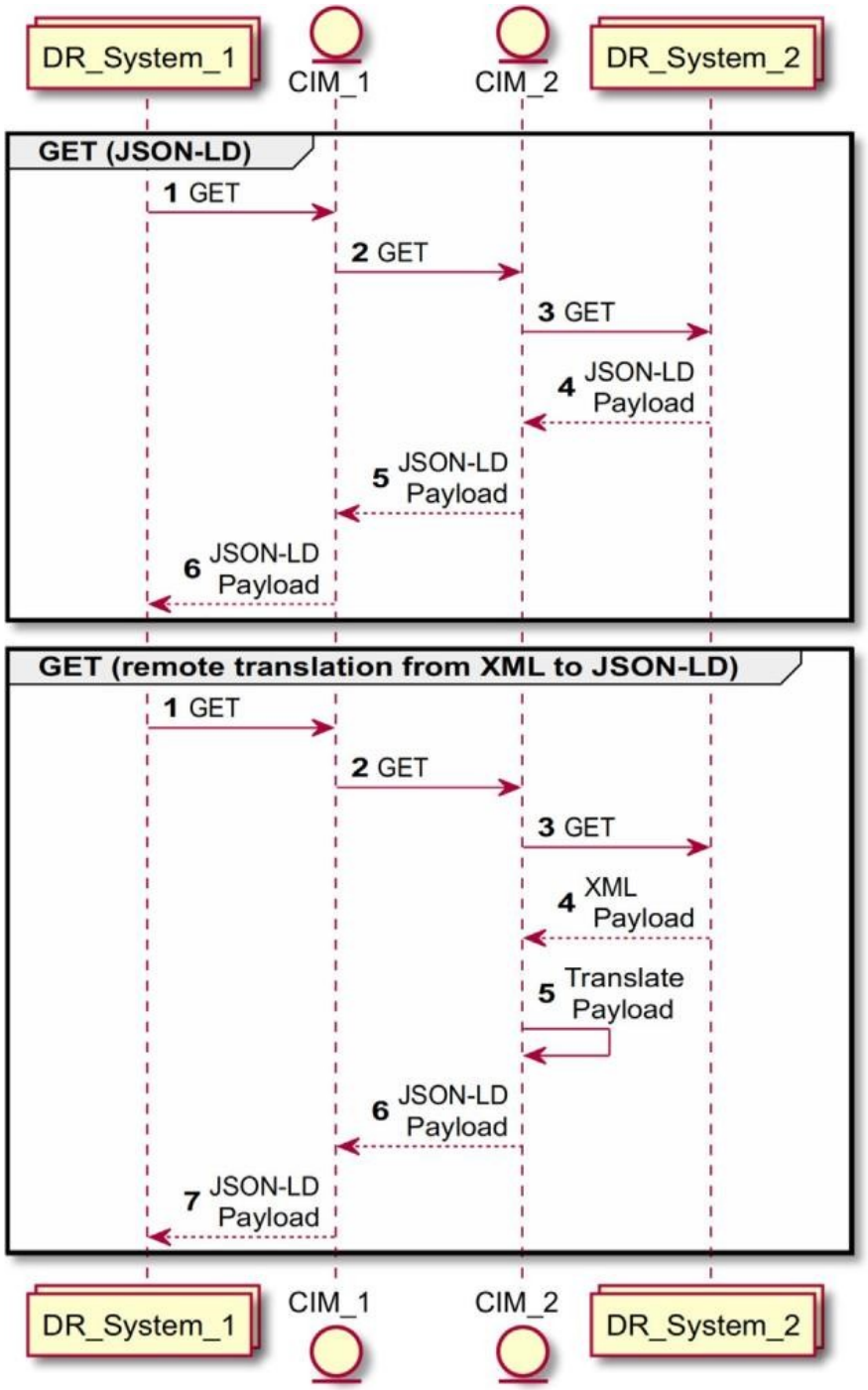

Fig. 5. Flows in experiment 1 for GET requests.Fig. 6. Flows in experiment 1 for POST requests.

Experiment 1: semantic interoperability

The SGAM framework was previously used to demonstrate how a semantic interoperability implementation may be assessed and quantified in the context of DR schemes [49]. The semantic interoperability of the

CIM was specifically evaluated to demonstrate how payloads defined in RDF, independent of serialization, and using alternative ontologies (SAREF [51] or SAREF4ENER22) could be successfully and accurately translated into JSON-LD using the DELTA ontology. Nonetheless, such results did not demonstrate how the CIM can also cope with payloads written in formats other than RDF and with models that are not ontologies.

The collection of test scenarios presented in the previous study was expanded in this experiment to evaluate the CIM semantic compatibility in those uncovered circumstances. To that goal, two payloads were specified in XML using the OpenADR XML format. The payloads were then shared using CIMs, which transformed them into their corresponding JSON-LD format. The GET (xml to json-ld), POST (json-ld to xml), and POST (xml to json-ld) messages are depicted in Figs. 5 and 6.

The DELTA ontology SHACL shapes were used to evaluate the payload converted to JSON-LD. 23 A manual validation was also undertaken to guarantee that the translation was done appropriately. Table 2 shows the outcomes of this exploration after delivering 100 payloads in each test case (GET (xml to json-ld), POST (json-ld to xml), and POST (xml to json-ld)).

Table 2 :Results of the semantic interoperability experiments.

Number of requests	Test case	Test case description	Verdict
100	GET (xml to json-ld)	Translate payload from OpenADR standard (XML) to DELTA ontology (JSON-LD)	PASS
100	POST (json-ld to xml)	Translate payload from DELTA ontology (JSON-LD) to OpenADR standard (XML)	PASS
100	POST (xml to json-ld)	Translate payload from OpenADR standard (XML) to DELTA ontology (JSON-LD)	PASS

Table 2 demonstrates that the CIM can successfully translate payloads from non-RDF formats into JSON-LD payloads modeled by the DELTA ontology. Although the XML format and the OpenADR schema were tested, additional interoperability modules can be installed to allow the CIM to convert payloads from other DR standards with alternative formats and models.

Experiment 2: scaling parallel requests

All requests represented in Figs. 5 and 6 are routed through the CIMs in this experiment. The time (in seconds) necessary to acquire the findings after transmitting these payloads, including the translation time involved in the data exchange, is the performance statistic for this experiment. These findings are detailed in Table 3 and illustrated in Fig. 7. From 1 to 600, the number of concurrent requests increases by 50 thread ticks. In order to compare the results, the averaged reaction times were converted to a logarithmic scale.

Finally, the CIM limits the number of parallel requests to 650 for security concerns; as a result, the maximum number of parallel requests for this experiment was set at 600.

Based on the statistics in Table 3, it appears that certain queries have extremely comparable response times. The Iman- Davenport test [52] was used to determine whether there are significant statistical differences between all of these requests, both GET and POST, with and without translation, at a confidence level of 95%, in order to determine whether the translations cause overhead during data transmission.

As a consequence, the test finds no statistically significant differences between GET (json-ld), GET (xml to json-ld), POST (json-ld), and POST (json-ld to xml), however there are differences with POST (xml to

json- ld). This difference is plainly seen in Fig. 7, where the time required to respond to these queries is substantially longer than the others.

Conclusion: This experiment demonstrated that sharing data across DR systems utilizing CIMs is a quick process with a linear trend, either when parallel requests are received, as seen in Fig. 7. It is worth noting that when there is no translation, 600 parallel requests are responded in less than half a second (<500 ms).

Table 3 Averaged response times for parallel requests.

Parallel requests	Response time (ms)				
	Get json-ld	GET xml to json-ld	POST json-ld	POST xml to json-ld	POST json-ld to xml
50	48	228	34	44	34,925
100	71	243	42	46	70,673
150	102	252	58	61	113,525
200	129	261	70	74	141,480
250	161	288	85	89	179,393
300	177	302	107	107	254,351
350	214	317	124	122	252,578
400	242	342	142	148	354,394
450	274	340	165	167	403,611
500	305	347	183	181	449,728
550	332	377	223	199	403,513
600	359	384	222	227	440,371

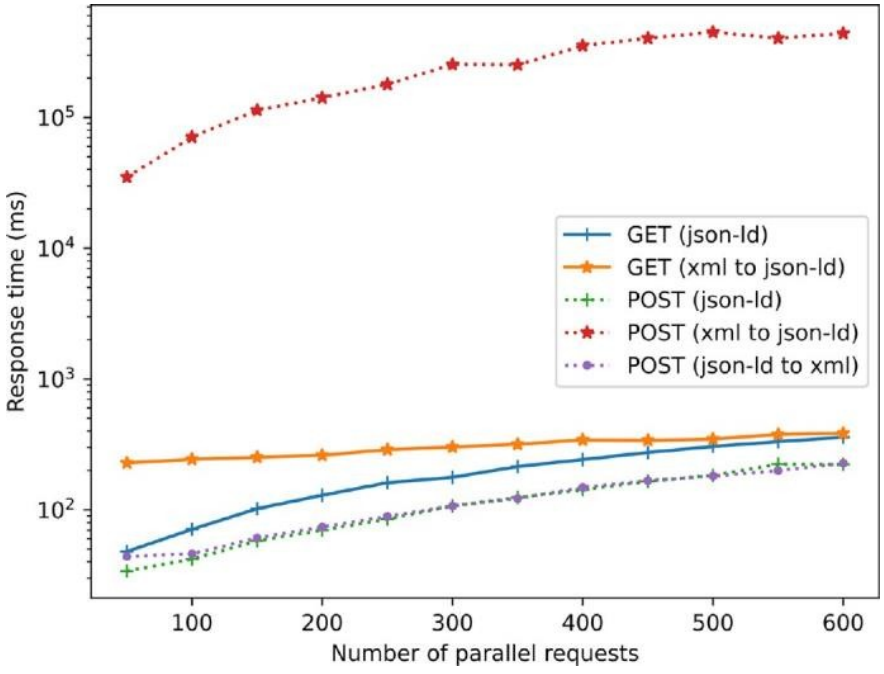

Fig. 7. Response times for scaling parallel requests.

Furthermore, with the exception of POST (xml to json-ld) in the trials, the CIM is capable of converting payloads without significant statistical difference in the majority of situations. This exception happens when

the payload is translated by the first CIM; while the remote CIM is translating, this exception does not occur; as a result of this discovery, multithreading in the first CIM is most likely impeding data translation.

Experiment 3: scaling a payload size

The GET (json-ld) and POST (json-ld) flows are shown in Figs. 5 and 6, respectively. In addition, a 1 Mb payload was synthetically enlarged to 9 Mb by 0.25 Mb pieces. Fig. 8 shows the times it took the CIMs to complete the GET and POST requests, depending on the size of the payload. The time it took to complete both queries in seconds is the performance statistic in this experiment. Finally, for security considerations, the CIM limits the size of the payload that may be exchanged to 9Mb; as a result, the maximum size in the experiment is the aforementioned.

Based on the data shown in Fig. 8, it can be determined that the CIM takes less than a half-minute to handle requests involving 9 Mb utilizing GET or POST requests; nevertheless, payloads involving 1 Mb require around 1 s to solve.

Conclusion: The CIM can handle payloads up to 9 Mb in a fair amount of time, as seen in Fig. 8. Note that there is a security constraint in place to prevent DoS attacks and server congestion caused by message overflow. Consider that most DR systems do not require huge volumes of data to be sent. at least not rapidly; instead, they must generally exchange tiny messages relatively quickly. As a result, even if the CIM takes seconds to respond to huge payloads, it can be said that it meets the data interchange criteria for DR systems in terms of size [49].

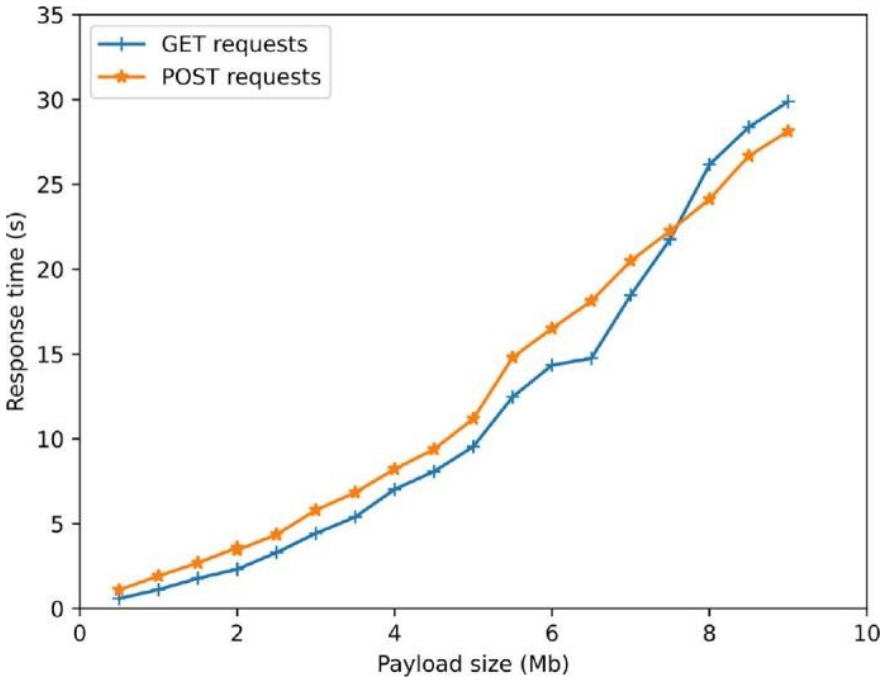

Fig. 8. Response times for scaling a payload size.

Limitations and lessons learnt

The use of the CIM in the context of the DELTA project demonstrated to the authors that semantic interoperability modules appear to be a viable way for performing uplifting and down-lifting instead of adapters (which are non-reusable pieces of software). However, the authors have observed that developers frequently create ad hoc solutions for Uplifting while ignoring Downlifting. As a result, while interoperability modules

appear to be a reasonable technology solution for achieving semantic interoperability, greater distribution is required, and developers must be educated to design and utilize them.

From the experimental part and the DELTA pilots' implementation. The authors discovered that in some circumstances, particularly in high-latency settings, the time necessary to complete a data exchange was insufficient. The authors investigated the cause of this disadvantage and determined that XMPP networks are not the quickest alternative. However, this is not a significant drawback because it is fast enough for most DR circumstances. Nonetheless, for other cases, such as DR with electric vehicles, new peer-to-peer systems should be investigated.

Furthermore, whereas DR often necessitates live data and signal transmission, payloads are typically small. However, certain DELTA project pilots needed the transmission of previous data. These huge payloads are unsuitable for XMPP networks. For this specific situation, the answer in the project was chunking the data; nevertheless, in circumstances where there is a definite requirement to exchange huge payloads, XMPP is not an appropriate technological choice.

CONCLUSIONS

The CIM, a cloud distributed semantically interoperable data exchange platform based on semantic technologies, is presented in this work. This technology advances current suggestions for data interchange in disaster recovery by addressing security, scalability, and interoperability concerns. To the best of the authors' knowledge, it is the only technology available to provide secure cloud communication across systems that

use diverse DR techniques. Furthermore, the CIM tool is platform-independent and may be used by DR systems to decentralize its components or communicate with other DR systems built with other standards. DR systems that rely on ontology-based standards, such as those indicated in Section 2, can automatically incorporate CIM by supplying the SHACL shapes that go with them. DR systems that do not rely on ontologies will need to provide interoperability modules to facilitate the conversion of their payload format and model into JSON-LD models based on any ontology.

Technically, the trials conducted have demonstrated that the CIM is an effective instrument for data interchange. Furthermore, investigations reveal that, in general, payload translation does not result in a considerable increase in reaction times. Nonetheless, it must be admitted that in one instance, the translation did result in an overhead. Finally, tests have proven that the CIM is capable of delivering big payloads up to 9Mb in size.

Future work will focus on improving translation times for those marginal circumstances where there is an overhead. To that purpose, new payload translation libraries and algorithms will be integrated and evaluated with the CIM. In addition, the CIM will be used in additional use cases to examine its acceptance in domains other than DR.

As a last note, keep in mind that CIM response times will increase as network technology advances. Its connection with 5G will be tested in the future to see if this new technology improves data sharing significantly.

REFERENCES

[1] L. Dusonchet, S. Favuzza, F. Massaro, E. Telaretti, G. Zizzo, Technological and legislative status point of stationary energy storages in the EU, Renew. Sustain. Energy Rev. 101 (2019) 158–167. [2] F.P. Sioshansi, Consumer, Prosumer, Prosumager: How Service Innovations Will Disrupt the Utility Business Model, Academic Press, 2019. [3] M. dos Santos Silva, Study on "residential prosumers in the European energy union", 2017. [4] A. Fernández-Izquierdo, A. Cimmino, C. Patsonakis, A.C. Tsolakis, R. García- Castro, D. Ioannidis, D. Tzovaras, OpenADR ontology: Semantic enrichment of demand response strategies in smart grids, in:

Proceedings of the 2020 International Conference on Smart Energy Systems and Technologies, Istanbul, Turkey, 7-9 September 2020, IEEE, 2020, pp. 1–6. [5] M.H. Yaghmaee, A. Leon-Garcia, M. Moghaddassian, On the performance of distributed and cloud-based demand response in smart grid, IEEE Trans. Smart Grid 9 (5) (2018) 5403–5417, http://dx.doi.org/10.1109/TSG.2017. 2688486. [6] C. Wang, N. Nasiriani, G. Kesidis, B. Urgaonkar, Q. Wang, L.Y. Chen, A. Gupta, R. Birke, Recouping energy costs from cloud tenants: Tenant demand response aware pricing design, in: Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems, Bangalore, India, July 14-17, 2015, ACM, 2015, pp. 141–150. [7] T. Deng, J. Yao, H. Guan, Maximizing profit of cloud service brokerage with economic demand response, in: Proceedings of the IEEE Conference on Computer Communications, Honolulu, HI, USA, April 16-19, 2018, IEEE, 2018, pp. 1907–1915, http://dx.doi.org/10.1109/INFOCOM.2018.8486412. [8] Y. Chen, J.M. Chang, Fair demand response with electric vehicles for the cloud based energy management service, IEEE Trans. Smart Grid 9 (1) (2018) 458–468, http://dx.doi.org/10.1109/TSG.2016.2609738. [9] K. Kaur, S. Garg, G. Kaddoum, S.H. Ahmed, F. Gagnon, M. Atiquzza- man, Demand-response management using a fleet of electric vehicles: An opportunistic-SDN-based edge-cloud framework for smart grids, IEEE Netw. 33 (5) (2019) 46–53, http://dx.doi.org/10.1109/MNET.001.1800496. [10] X. Zhang, D. Biagioni, M. Cai, P. Graf, S. Rahman, An edge-cloud integrated solution for buildings demand response using reinforcement learning, IEEE Trans. Smart Grid 12 (1) (2021) 420–431, http://dx.doi.org/10.1109/TSG. 2020.3014055. [11] M. Frincu, R. Draghici, Towards a scalable cloud enabled smart home automation architecture for demand response, in: Proceedings of the 2016 IEEE PES Innovative Smart Grid Technologies Conference Europe, Ljubljana, Slovenia, October 9-12, 2016, IEEE, 2016, pp. 1–6, http://dx.doi.org/10. 1109/ISGTEurope.2016.7856235. [12] N. Galkin, C.-W. Yang, L. Nordström, V. Vyatkin, Prototyping multi-protocol communication to enable semantic interoperability for demand response services, in: 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2021, pp. 15–20. [13] J. Wu, F. Orlandi, T. AlSkaif, D. O'Sullivan, S. Dev, A semantic web approach to uplift decentralized household energy data, Sustain. Energy Grids Netw. 32 (2022) 100891, http://dx.doi.org/10.1016/j.segan.2022.100891, URL: https://www.sciencedirect.com/science/article/pii/S2352467722001497. [14] I. Esnaola-Gonzalez, F.J. Díez, L. Berbakov, N. Tomasevic, P. Štorek, M. Cruz, P. Kirketerp, Semantic interoperability for demand-response programs: Respond project's use case, in: 2018 Global Internet of Things Summit (GIoTS), IEEE, 2018, pp. 1–6. [15] Q. Zhou, S. Natarajan, Y. Simmhan, V. Prasanna, Semantic information mod- eling for emerging applications in smart grid, in: 2012 Ninth International Conference on Information Technology-New Generations, IEEE, 2012, pp. 775–782. [16] Hongseok Kim, Y. Kim, K. Yang, M. Thottan, Cloud-based demand response for smart grid: Architecture and distributed algorithms, in: Proceedings of the 2011 IEEE International Conference on Smart Grid Communications, Brussels, Belgium, October 17-20, 2011, IEEE, pp. 398–403, http://dx.doi. org/10.1109/SmartGridComm.2011.6102355. [17] F. Bellifemine, A. Poggi, G. Rimassa, JADE–a FIPA-compliant agent framework, in: Proceedings of PAAM, London, 1999, p. 33. [18] H. Wicaksono, T. Boroukhian, A. Bashyal, A demand-response system for sustainable manufacturing using linked data and machine learning, in: Dynamics in Logistics, Springer, Cham, 2021, pp. 155–181. [19] A.T. Schreiber, Y. Raimond, RDF 1.1 primer, 2014. [20] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider, S. Rudolph, OWL 2 web ontology language primer, W3C Recomm. (2009). [21] A. Fernández-Izquierdo, A. Cimmino, R. García-Castro, Supporting demand- response strategies with the DELTA ontology, in: 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA), IEEE, 2021, pp. 1–8. [22] H. Knublauch, D. Kontokostas, Shapes constraint language (SHACL), W3C Recomm. (2017) URL: https://www.w3.org/TR/shacl/. [23] A. Cimmino, A. Fernández-Izquierdo, M. Poveda-Villalón, R. García-Castro, Ontologies for IoT semantic interoperability, in: C. Zivkovic, Y. Guan, C. Grimm (Eds.), IoT Platforms, Use Cases, Privacy, and Business Models: With Hands-on Examples Based on the VICINITY Platform, Springer International Publishing, Cham, 2021, pp. 99–123. [24] Energy Market Information Exchange (EMIX) Version 1.0, OASIS Consor- tium, 2012. [25] An Introduction to the Universal Smart Energy Framework, Smart Energy, 2019. [26] OpenADR 2.0 Profile Specification, B Profile, OpenADR Alliance, 2015. [27] EN 50090-1 Home and Building Electronic Systems (HBES), CENELEC, 2011. [28] IEC 61970-301:2020 Energy Management System Application Program Interface (EMS-API) - Part 301: Common Information Model (CIM) base, International Electrotechnical Commission, 2020. [29] IEC 62056 Electricity

Metering Data Exchange - The DLMS/COSEM Suite, International Electrotechnical Commission, 2016. [30] IEC 62746-10-1:2018 Systems Interface Between Customer Energy Man- agement System and the Power Management System, International Electrotechnical Commission, 2018. [31] C. Neureiter, Introduction to the SGAM toolbox, in: Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences, Tech. Rep, 2013. [32] T. Linnenberg., A.W. Mueller., L. Christiansen., C. Seitz., A. Fay., OntoENERGY – a lightweight ontology for supporting energy-efficiency tasks - enabling generic evaluation of energy efficiency in the engineering phase of automated manufacturing plants, in: Proceedings of the International Con- ference on Knowledge Engineering and Ontology Development, Algarve, Portugal, September 19-22, 2013, Vol. 1, 2013, pp. 337–344. [33] J.L. Hippolyte, S. Howell, B. Yuce, M. Mourshed, H.A. Sleiman, M. Vinyals, L. Vanhee, Ontology-based demand-side flexibility management in smart grids using a multi-agent system, in: Proceedings of the 2016 IEEE Interna- tional Smart Cities Conference, Trento, Italy, September 12-15, 2016, 2016, pp. 1–7. [34] M.J. Kofler, C. Reinisch, W. Kastner, A semantic representation of energy- related information in future smart homes, Energy Build. 47 (2012) 169–179. [35] J. Verhoosel, D. Rothengatter, F. Rumph, M. Konsman, An ontology for modeling flexibility in smart grid energy management, in: Proceedings of the EWork and EBusiness in Architecture, Engineering and Construction - European Conference on Product and Process Modelling, Reykjavik, Iceland, 25-27 July 2012, CRC Press, 2012, pp. 931–938. [36] T.G. Stavropoulos, D. Vrakas, D. Vlachava, N. Bassiliades, Bonsai: a smart building ontology for ambient intelligence, in: Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, Craiova, Romania, June 13-15, 2012, ACM, 2012, pp. 1–12. [37] J.-L. Hippolyte, S. Howell, B. Yuce, M. Mourshed, H.A. Sleiman, M. Vinyals, L. Vanhée, Ontology-based demand-side flexibility management in smart grids using a multi-agent system, in: 2016 IEEE International Smart Cities Conference (ISC2), IEEE, 2016, pp. 1–7. [38] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez, A. Arnaiz, EEPSA as a core ontology for energy efficiency and thermal comfort in buildings, Appl. Ontol. 16 (2) (2021) 193–228. [39] M. Haghgoo, I. Sychev, A. Monti, F.H. Fitzek, SARGON–smart energy domain ontology, IET Smart Cities 2 (4) (2020) 191–198. [40] M.-F. Robbe, M. Vinyals, S. Lodeweyckx, J.M. Espeche, P.-E. Brun, S.V. Costa, M. Mourshed, A. Kavgić, T. Loureiro, Putting residential flexibility management into action with pilot sites in europe: From mas2tering to drive projects, Multidiscip. Digit. Publ. Inst. Proc. 2 (15) (2018) 1130. [41] A.M. Ouksel, A. Sheth, Semantic interoperability in global information systems, ACM SIGMOD Rec. 28 (1) (1999) 5–12. [42] I. Esnaola-Gonzalez, F.J. Diez, Integrating building and iot data in demand response solutions, in: Proceedings of the 7th Linked Data in Architecture and Construction Workshop (LDAC 2019), 2389, CEUR, 2019, pp. 92–105. [43] A.E. Youssef, Exploring cloud computing services and applications, J. Emerg. Trends Comput. Inform. Sci. 3 (6) (2012) 838–847. [44] A. Hornsby, R. Walsh, From instant messaging to cloud computing, an XMPP review, in: Proceedings of the IEEE International Symposium on Consumer Electronics , Braunschweig, Germany, June 7-10, 2010, IEEE, 2010, pp. 1–6. [45] M. Kirsche, R. Klauck, Unify to bridge gaps: Bringing XMPP into the internet of things, in: Proceedings of the 2012 IEEE International Conference on Per- vasive Computing and Communications Workshops, Lugano, Switzerland, March 19-23, 2012, IEEE, 2012, pp. 455–458. [46] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, in: Journal on Data Semantics X, Springer, 2008, pp. 133–173. [47] C.R. Rivero, A. Schultz, C. Bizer, D. Ruiz Cortés, Benchmarking the perfor- mance of linked data translation systems, in: Proceedings of the Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012, CEUR-WS, 2012. [48] O. Ethelbert, F.F. Moghaddam, P. Wieder, R. Yahyapour, A JSON token-based authentication and access management schema for cloud saas applications, in: Proceedings of the 5th IEEE International Conference on Future Internet of Things and Cloud, Prague, Czech Republic, August 21-23, 2017, IEEE, 2017, pp. 47–53. [49] A. Cimmino, N. Andreadou, A. Fernández-Izquierdo, C. Patsonakis, A.C. Tsolakis, A. Lucas, D. Ioannidis, E. Kotsakis, D. Tzovaras, R. García-Castro, Semantic interoperability for DR schemes employing the SGAM framework, in: Proceedings of the 2020 International Conference on Smart Energy Systems and Technologies, Istanbul, Turkey, 7-9 September 2020, IEEE, 2020, pp. 1–6. [50]A. Cimmino a,J. Cano-Benito a, A. Fernández-Izquierdo a, C. Patsonakis b, A. C. Tsolakis b, R. García-Castro a, D. Ioannidis b, D. Tzovaras b : A scalable, secure, and semantically interoperable client for cloud-enabled Demand Response, Future Generation Computer Systems 141 (2023) 54–66 . [51] L. Daniele, F. den Hartog, J. Roes, Created in close interaction with the industry:

the smart appliances reference (SAREF) ontology, in: Proceedings of the 7th International Workshop Formal Ontologies Meet Industries, Berlin, Germany, August 5, 2015, Springer, 2015, pp. 100–112. [52] D.G. Pereira, A. Afonso, F.M. Medeiros, Overview of Friedman's test and post-hoc analysis, Comm. Statist. Simulation Comput. 44 (10) (2015) 2636–2653.