Accepted at: 7^{th} IEEE International Conference on Fog and Edge Computing (ICFEC 2023)

Data Sovereignty at the Edge of the Network

Vasileios Karagiannis
Center for Digital Safety & Security
Austrian Institute of Technology
Vienna, Austria
vasileios.karagiannis@ait.ac.at

Astrid Al-Akrawi
Center for Digital Safety & Security
Austrian Institute of Technology
Vienna, Austria
astrid.al-akrawi@ait.ac.at

Oliver Hödl
Center for Digital Safety & Security
Austrian Institute of Technology
Cooperative Systems Research Group
University of Vienna
Vienna, Austria
oliver.hoedl@ait.ac.at

***Abstract*—Data sovereignty upholds that data is subject to the rules of the data provider and the applicable laws of the country where the data is collected. To achieve data sovereignty, various approaches have been proposed which rely on cloud computing resources to provide users with the necessary functionality to exchange sovereign data. However, when users are located far away from the cloud, using cloud resources can cause delays and potential network bandwidth bottlenecks. To address such issues, we propose an architecture that relies on edge computing resources. In this architecture, the components that handle the transfer of sovereign data are placed on-premise, allowing users to exchange data without using the cloud. To evaluate our approach, we conduct various experiments in a real-world setting using a system that follows the proposed architecture and a baseline that relies on the cloud. The results show that the proposed approach provides significant benefits, such as a ∼20% latency reduction and increased network bandwidth.**

***Index Terms*—Data Sovereignty, Edge Computing, Cloud Computing, Dataspace Connector, Data Spaces, Sovereign Data**

I. Introduction

Data Sovereignty is a novel concept in the distributed systems community, which refers to the distribution and usage of data [1]. According to data sovereignty, data is subject to the laws of the country in which it is collected, and the constraints of the data provider who may define how the data can be used, in what context, and by whom, among others. This concept is becoming increasingly important in modern societies because governments realize that due to cloud computing, data collected internally (e.g., from local citizens) may be exported abroad where it is processed and exploited disregarding the privacy regulations of the origin country [2], [3]. In addition, data that is hosted outside the origin country becomes subject to the laws of the host country. This can lead to the exposure of sensitive information to the host country's government [2]. Such reasons motivate lawmakers to create preventive laws such as the European GDPR [4], or the USA ban on Huawei [1]. Furthermore, countries (such as Canada) issue white papers and guidelines to raise awareness about data sovereignty and protect the privacy of the citizens [5].

To comply with the concept of data sovereignty, various initiatives have been formed consisting of both academic and industrial partners [1], [6], [7]. Their prime goal is to design architectures and mechanisms that enable the exchange of sovereign data, i.e., data that maintain its sovereignty while being transferred [1]. Examples of such initiatives are the International Data Spaces Association (IDSA) and Gaia-X, which drive the adoption of sovereign data with software, documentation and events [8].

Despite the broad support, data sovereignty is still in its infancy. Therefore, the existing approaches for transferring sovereign data are at an early stage and might be further improved. In current architectures, for example, the data is usually not stored in the cloud due to sovereignty concerns. However, other components may be placed in the cloud [9]. Such components can be related to, e.g., authentication mechanisms, or system monitoring [10], [11]. Another component that may be placed in the cloud is the connector which handles the transfer of the data [12]. While there can be incentives for placing components in the cloud (as discussed later on in Section IV-A), communication with a cloud may result in high latency and network bandwidth bottlenecks [13]. Such side effects of the cloud may reduce the quality of experience of the users and hinder the adoption of data sovereignty systems.

To tackle these problems, in this paper we propose an architecture that relies on edge computing resources for hosting both the data and the connector. In addition, we implement a system that is based on the proposed architecture and enables the transfer of sovereign data between users without accessing the cloud. To evaluate our approach, we deploy our system in a real-world setting with users that reside in different countries, and we conduct various experiments. The results show significant performance benefits such as reduced communication latency and increased network bandwidth compared to a baseline that uses cloud computing resources.

The remainder of this paper is organized as follows: In Section II, we present related work. Then, in Section III, we describe a system model for sovereign data exchange. Afterward, Section IV presents the proposed approach, and Section V provides an empirical evaluation. Finally, Section VI concludes this paper and suggests future research directions.

II. Related Work

Related work on data sovereignty can be found in conceptual and review papers as well as white papers and re-

This work has been co-funded by the European Union's Horizon Innovation Actions under grant agreement No. 101069510, EDDIE—European Distributed Data Infrastructure for Energy.

ports [10], [14]. Geisler et al. [15] provide a comprehensive view of data ecosystems and analyze relevant requirements and challenges related to handling data when considering privacy and sovereignty aspects. Kotka et al. [16] analyze the legal aspect of moving sensitive data to the cloud while taking into account data sovereignty and applicable data protection facets. Firdausy et al. [12] discuss the applicability of data sovereignty to enterprises and organizations. Solmaz et al. [17] discuss data spaces, i.e., controlled environments in which sovereign data is exchanged, and focus on motivation, technical developments and challenges related to data interoperability. Finally, Brost et al. [18] point out the value of data in the context of industrial data spaces, and focus on security aspects and usage control, i.e., techniques for ensuring that data is used as dictated by the data provider. While these works do provide significant contributions to the field of data sovereignty, they do not produce quantitative results from an actual implementation which is the goal of the work at hand.

There is also related work from contributions with implementations and results, which is, however, not as extensive. Qarawlus et al. [19] and also Nast et al. [20] address the problem of handling sovereign data from devices with limited hardware resources. Qarawlus et al. focus on messaging schemes, while Nast et al. aim at designing a specialized connector that exposes sovereign data based on a standardized API. Sarabia-Jácome et al. [21] propose a system for sovereign data exchange that targets a seaport use case in which data from the port terminal is shared with a port authority using the FIWARE platform [22]. Liang et al. [23] propose a system for sharing personal health data which considers privacy and sovereignty. In the proposed system, health data from wearable devices is uploaded to the cloud. To access this data, a user needs to give explicit consent, e.g., to an insurance company. Notably, such approaches do not discuss the difference between placing the connector in the cloud and placing it at the edge. To the best of our knowledge, this paper is one of the first works that provide empirical results regarding the latency of transferring sovereign data using edge resources.

III. System Model

In this section, we present a system model for facilitating the exchange of sovereign data. Our model is based on widely-accepted reference architectures from the literature and aligns with the principles of data sovereignty [6], [10], [24]. To make this system model comprehensible, first, we introduce the utilized terminology in Section III-A. Afterward, in Section III-B, we describe the essential components of the system along with the basic interactions among these components.

A. Nomenclature

In a system for sovereign data exchange, there are data consumers and data providers. A *consumer* is an individual, organization, or application that wants to acquire sovereign data, and use it according to its terms of use. A *provider* is an individual or organization that owns data and wants to share this data with others that plan to use it according to its terms of use. Sovereign data, or more specifically, a *sovereign dataset* is a dataset which is accompanied by terms of use that have been created by the data provider and comply with local laws and regulations. The terms of use dictate how the data can be used. Specifically, the terms of use may specify the type of allowed processing (e.g., algorithms related to statistics, machine learning, etc.), the duration of allowed processing (e.g., for one month, or other), redistribution rules, and other prohibitions and constraints. The collection of all the terms of use is referred to as the dataset's *policy*. Policies can be formulated using ODRL (Open Digital Rights Language) which provides an expression of rights based on json or xml.

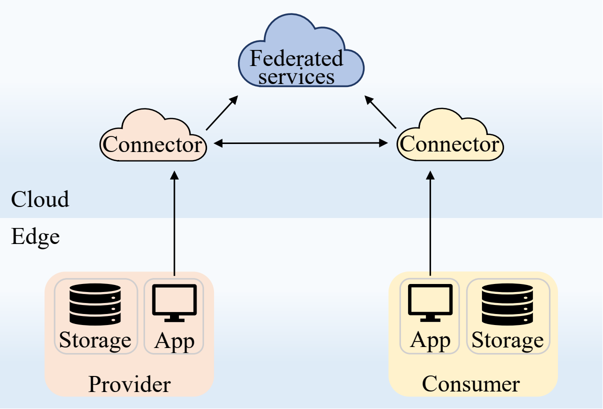

Fig. 1: System architecture for sovereign data exchange.

To handle the acquisition and acceptance of policies, a connector can be used [25]. The *connector* is a software application that handles the basic interactions for exchanging sovereign data, e.g., requesting a dataset, agreeing to the policy, and transferring the dataset from the provider to the consumer. Thus, the connector also aids in achieving interoperability between data providers and data consumers.

In addition to the connector, a system for sovereign data exchange may include federated services provided by a trusted organization [24]. The *federated services* are software applications that can be used by all participants, i.e., both providers and consumers. Two important federated services are the trusted authority, and the clearing house [10]. The *trusted authority* validates the identity of the connectors and ensures trust among providers and consumers. The *clearing house* logs all the interactions that take place within the system, and aids in resolving conflicts (e.g., if a consumer claims that the agreed dataset has not been transferred, but the provider claims otherwise). These two federated services aim at ensuring that for any sovereign data transfer, there are transcripts regarding the identity of the participants and the agreed policy.

B. Architecture

A system that facilitates sovereign data consists of various components that can be deployed in the cloud (e.g., using a commercial cloud provider such as Google) and/or at the edge (e.g., using private on-premise computing resources), as shown in Fig. 1. Even though such a system might comprise multiple data providers and data consumers, Fig. 1 presents a basic depiction with one provider and one consumer. As shown in Fig. 1, the provider has local storage which hosts the sovereign data. Similarly, the consumer has local storage for hosting the data once acquired by the provider. In addition, both the provider and the consumer have a local app (i.e., a front-end application) that provides a user interface to the system. For example, through the app, the provider can interact with the provider connector. Similarly, the consumer can interact with the consumer connector. Both connectors are typically deployed in the cloud (this is further discussed in Section IV-A). The federated services which aim at being used by all participants are also placed in the cloud [10], [24].

To exchange sovereign data, initially, the provider registers a dataset at the provider connector. To do so, the provider submits to the connector the policy of the dataset, and the means to pull the dataset from the provider's storage (e.g., via a communication protocol such as HTTP). When a dataset is registered at the provider connector, consumers may request it. To do that, a consumer (via the consumer app) makes a call to the consumer connector to request a registered dataset from the provider connector [6]. The sequence of steps that follow to request and transfer a sovereign dataset is shown in Fig. 2.

In Step 1 of Fig. 2, the consumer connector requests a dataset from the provider connector. The provider connector responds with the policy of the dataset (Step 2). The consumer examines this policy and signs a contract to abide by the dataset's terms of use (Step 3). The provider connector accepts and responds with a contract agreement (Step 4). The consumer connector then requests a transfer of the agreed dataset (Step 5). In this request, the consumer connector includes a file path to the consumer's storage (and additional credentials to access this storage if needed). Finally, the provider connector pulls the dataset from the storage of the provider (Steps 6 and 7) and pushes the dataset to the storage of the consumer (Steps 8 and 9). This concludes the transfer (Step 10).

While Fig. 2 shows interactions among the basic components of the system, more interactions may be necessary, e.g., with the federated services. For example, a consumer may need to run queries and search for useful sovereign datasets offered by any provider. Such queries can be performed to a federated service that acts as a central catalog for all providers to advertise their data (e.g., using metadata or self-descriptions [24]). By searching this catalog, the consumer can discover the addresses of provider connectors with useful datasets and proceed to request any of these datasets using the consumer connector (as shown in Fig. 2).

IV. Sovereign Data with Edge Computing

In this section, we propose an architecture for enabling sovereign data, which leverages edge resources to provide additional functionality and improved performance. To this end, in Section IV-A we provide some key observations regarding the location of the system components and potential benefits that may derive from the use of edge computing resources. Then, we describe the proposed architecture in Section IV-B.

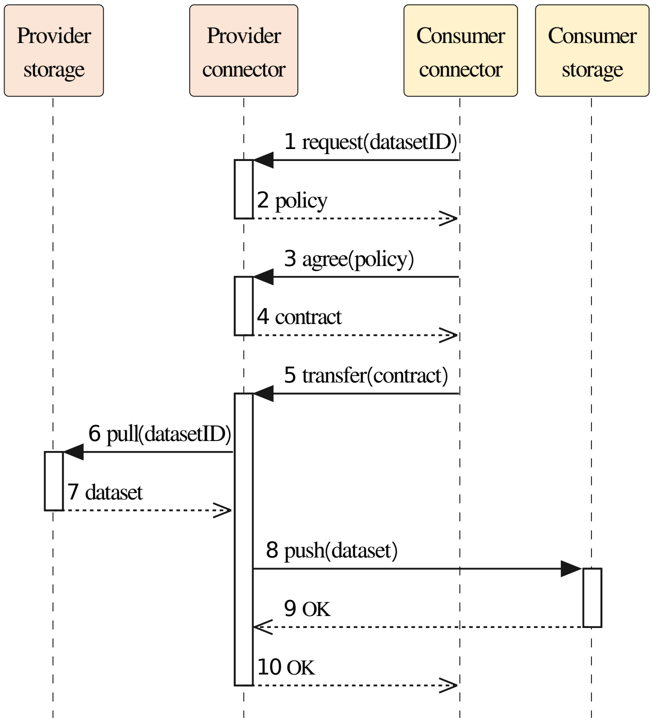

Fig. 2: Transfer of a sovereign dataset.

A. Observations

As shown in Fig. 1, both the provider and the consumer use on-premise storage for hosting sovereign datasets. This is important because third-party storage (e.g., using cloud services) can go against the principles of data sovereignty [3]. For example, a sovereign dataset must be used according to its policy. However, storage providers do not typically provide policy guarantees. On the contrary, it is usually the users of the storage that must agree to the terms of use of the storage provider, which may include anonymized processing of stored data without considering each dataset's policy [3]. For this reason, on-premise storage is preferred for sovereign data.

The connectors, on the other hand, might be deployed either in the cloud or at the edge [12], [26]. Nevertheless, early deployments seem to focus on the cloud [27], [28]. There may be various reasons for this. The cloud provides general-purpose scalable resources that can be used at a low price without the need to buy or maintain any equipment [29]. Thus, using the cloud for the connectors provides ease, flexibility and scalability [30]. The connectors do store information regarding datasets and policies, but they do not store the actual data. Even during the transfer, the provider connector pulls a stream of data from the provider storage and pushes it to the consumer storage without storing it (as discussed in Section III-B). Thus, deploying the connectors in the cloud does not breach the concept of sovereignty, as long as the location of the cloud is not in a prohibited area.

Interestingly, the possibility to run connectors in the cloud has created novel business opportunities for platform-as-a-service approaches [28], which are also referred to as Data Sovereignty-as-a-Service or Connector-as-a-Service [31]. In such approaches, the service provider deploys an instance of a connector in the cloud and handles all the necessary maintenance and interactions with federated services. By using this service, users can make full use of a sovereign data exchange system without running and maintaining a connector. Instead, users only need to register datasets and define policies.

While deploying the connectors in the cloud can be justified by such valid reasons, it may also have some side effects. When the provider connector pulls the data from the source (i.e., the provider storage) and pushes it to the destination (i.e., the consumer storage), there is no consideration for the network path that the data follows, the latency to transfer the data or the bandwidth utilization in the underlying network. Thus, there may be cases when the source and the destination reside nearby, but the data is transferred through a remote cloud thereby utilizing additional network resources and increasing the latency of the transfer [32]. While this specific problem has not been widely researched in the context of data sovereignty yet, similar problems regarding transferring data through a remote cloud have been addressed in the context of edge computing and the IoT [33]–[38]. For this reason, in the next section, we propose an architecture for sovereign data exchange that is inspired by edge computing, and we advocate the potential benefits.

B. Proposed System Architecture

To derive the proposed system architecture, first, we consider a typical case of sovereign data transfer, as shown in Fig. 1. When the provider connector pulls the data from the provider storage and pushes it to the consumer storage, the communication latency of the transfer includes the latency of the network path from the provider to the cloud ($Lat_{Pro \to Cloud}$) and the latency from the cloud to the consumer ($Lat_{Cloud \to Con}$). This means that the latency of the transfer through the cloud, i.e., $LatCloud_{Pro \to Con}$, can be formulated as:

$$LatCloud_{Pro \to Con} = Lat_{Pro \to Cloud} + Lat_{Cloud \to Con} \quad (1)$$

Thus, the cloud can be considered as a detour on the network path between the provider and the consumer. Detours through the cloud are likely to consume a high amount of network resources, while also increasing the latency [32]. For this reason, we propose to place the connectors on-premise, as shown in Fig. 3. In this case, the provider connector and the consumer connector are placed at the provider site and the consumer site, respectively. The connectors can still be accessed by users through the user's app. Also, the connectors can still communicate with the federated services that are deployed in the cloud. However, since both of the connectors are now at the edge of the network, they might be able to communicate with each other without the need to go through a remote cloud. This may be able to provide performance benefits and additional functionality that are associated with the use of edge resources (as discussed below).

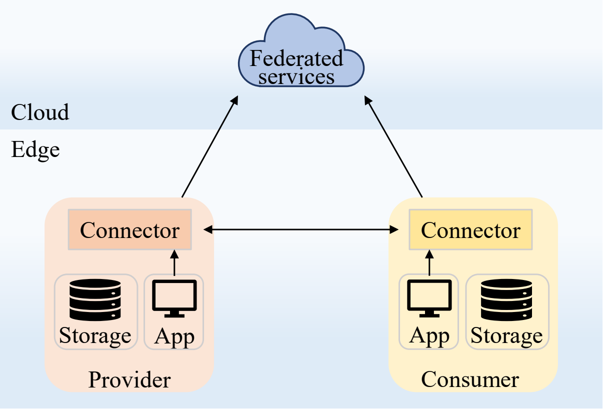

Fig. 3: System architecture with on-premise connectors.

Regarding functionality, since the data is transferred directly from the provider to the consumer, there is no risk of moving the data through a cloud that is located in an undesignated region. For example, in case both the provider and the consumer reside in Austria, there may be a requirement that the data does not leave the country. However, if the provider connector runs in a cloud that is located in another country (e.g., in Switzerland), this requirement is not met because the data travels through the provider connector's location. This can be avoided in the proposed architecture (whereby the provider connector is placed on the premise of the provider) because the data is sent from the provider site directly to the consumer site without going through the cloud.

Regarding performance, the communication latency of the transfer in the proposed architecture includes the latency of the provider connector to read the data from the storage and the latency to send the data to the consumer ($Lat_{Pro \to Con}$). The former can be assumed to approximate zero when the provider connector is at the provider site because in this case there is no communication latency (only reading a file from the filesystem). Thus, the latency of the transfer when the connectors are placed at the edge, i.e., $LatEdge_{Pro \to Con}$, is:

$$LatEdge_{Pro \to Con} = 0 + Lat_{Pro \to Con} \quad (2)$$

By comparing equation 1 with equation 2, we note that in both latencies the source and destination are the same. Nevertheless, equation 1 includes a detour through the cloud. In case the cloud is on the path from the source to the destination, the network path of equation 1 will be similar to the path of equation 2 [39]. Thus, the two latencies will also be similar, i.e., $LatEdge_{Pro \to Con} \approx LatCloud_{Pro \to Con}$ (3). However, in case the cloud is not on the path between source and destination, the network path of equation 1 will be longer (due to the detour) and consequently, the latency will be higher [32]. Thus, $LatEdge_{Pro \to Con} < LatCloud_{Pro \to Con}$ (4). By uniting equations 3 and 4 applies that:

$$LatEdge_{Pro \to Con} \lesssim LatCloud_{Pro \to Con} \quad (5)$$

Therefore, the proposed architecture is expected to provide similar or lower communication latency for transferring sovereign data. Presumably, the use of edge computing resources may lead to other benefits as well, such as improved user experience. This can occur due to reduced response times when the user interacts with the connector (through the app) because the connector is now deployed closer to the app for both the provider and the consumer, as shown in Fig. 3. Nevertheless, in this paper, and also in our evaluation in Section V, we focus on benefits related to the transfer of the data which is the main goal of data sovereignty.

V. Evaluation

To evaluate our approach, we build a system that follows the proposed edge architecture (shown in Fig. 3). In addition, we implement a system that follows the typical cloud architecture (shown in Fig. 1) which is used as a baseline. The goal of our evaluation is to compare the two approaches regarding communication latency and network bandwidth. To this end, we deploy connectors in the cloud using the Google Cloud Platform, which are used by the baseline. In addition, we deploy connectors at the user sites, i.e., in the different countries, which are used by the proposed approach. The user sites also include a custom Python-based storage application that is used for hosting the data. Fig. 4 shows the location of the cloud in Germany and the user sites in Spain, England, France, Belgium, Netherlands, Italy, and Poland, which are all located within Europe, close and farther away from the cloud. We consider this to be an appropriate area for our experiments because there are already many European countries involved with building testbeds and trying out sovereign data approaches [40]. The connectors we use are based on the Eclipse Dataspace Connector which is an open-source software application based on Java that is supported by the Eclipse foundation [25]. Some additional modifications are performed on this connector to make it compatible with our system. All components are connected through the Internet.

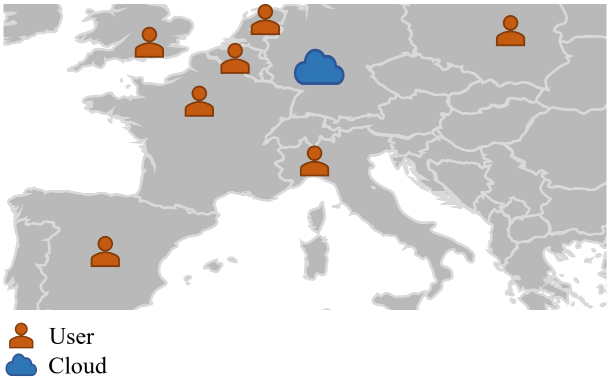

Fig. 4: Location of the cloud and the users in Europe.

For this evaluation, we consider a smart energy use case because smart energy applications typically include interactions with the user, e.g., for alerts or real-time analytics, and rely on low latency for offering prompt response times and high user experience [32], [41], [42]. Since low latency is the goal of our approach, we consider this to be an appropriate use case. In our experiments, we use data from a publicly available dataset with real measurements from smart meters [43]. We use measurements of different data sizes to represent different applications. The data sizes we use are 85 Bytes (B) corresponding to 1 measurement, 7 Kilobytes (KB) for measurements of 1 day, 224 KB for 1 month, 1 Megabyte (MB) for 4.5 months, 2 MB for 9 months and 4 MB for 18 months. For example, the data size of 1 measurement can represent a safety application in which the provider is a household that sends sovereign data to an anomaly-detection service (consumer) which aims at detecting potential hazards in real time (such as gas leaks, or malfunctions). Even though for this particular case, the transfer of smaller data sizes may be considered more relevant e.g., to transfer every new measurement as soon as it is available, applications that use larger data sizes, e.g., for real-time analytics, can also require low latency. For this reason, larger data sizes are also taken into account.

A. Results

To produce valuable results which represent the general case, we run 2,520 experiments. In the experiments of the baseline, the provider connector–which is deployed in the cloud–follows the process of Fig. 2 to transfer data from the provider site to the consumer site through the cloud. In the experiments of the proposed approach, the provider connector–which now resides at the provider–follows the same process to transfer data from the site of the provider to the site of the consumer directly. For both approaches, every user acts as a data provider that sends sovereign data of different data sizes to all the other users (that act as consumers in different countries) multiple times. For each time, we measure the communication latency and the hop count of the transfer.

To visualize the latency measurements, we present Table I and Fig. 5. Specifically, Fig. 5 shows the latency of each data size based on the baseline and the proposed approach. Table I shows the numerical values of the average, the standard deviation, and the reduction in the average that stems from the use of the proposed approach. As shown in Fig. 5, the communication latency of each approach is similar for very small data sizes and increases when the data size becomes larger. The proposed approach has a similar standard deviation with the baseline, i.e., the distribution of the values is similar. However, the average latency of our approach is about 20% lower than the baseline for all data sizes (also shown in Table I). Moreover, we note that the upper quartile of the

TABLE I: Communication latency in milliseconds (ms) of each data size and the percentages of reduction.

	Baseline		Proposed		Reduction
	avg	st dev	avg	st dev	avg (%)
85 B	63	15	51	16	19
7 KB	63	17	51	17	19
224 KB	173	51	133	52	23
1 MB	249	73	198	74	20
2 MB	266	79	219	82	18
4 MB	313	84	258	89	18

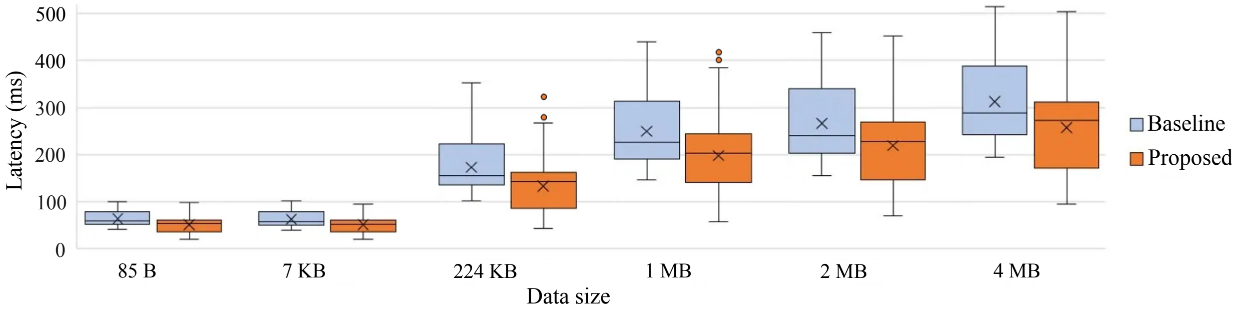

Fig. 5: Communication latency (in ms) of each data size based on the baseline and the proposed approach.

proposed approach is always similar to or less than the average of the baseline. This means that, in our approach, 75% of the values are similar to or less than the average value of the baseline. Furthermore, the minimum values of every data size in the proposed approach, i.e., the lower whiskers in Fig. 5, are always at least 50% lower than the corresponding minimum values of the baseline.

Notably, such a significant reduction in latency (i.e., 50% or more) occurs for data transfers between users that reside close to each other, e.g., in Belgium and the Netherlands, because in this case, transferring the data through the cloud forms a large detour (as discussed in Section IV-B). We also note that the maximum values of all data sizes are similar in both approaches. This happens when the network path of both approaches is similar, i.e., when the cloud is on the path from the provider to the consumer. In our experiments, the maximum latency occurs for transfers between users in Spain and Poland. In this case, transferring the data through the cloud does not form a noticeable detour, as shown in Fig. 4. Finally, we observe that while the percentage of reduction in the average latency is rather steady (about 20% as shown in Table I), the actual reduction in ms grows when the data size increases. The average latency reduction starts at 12 ms for data of 85 B and grows to 55 ms for data of 4 MB. Considering that the examined use case aims at detecting hazards potentially in real time, 20% latency reduction that can also reach 50% may be considered a significant improvement. Overall, we note that the results comply with the latency analysis in Section IV-B, and show that the proposed edge architecture tends to reduce the communication latency compared to the cloud architecture.

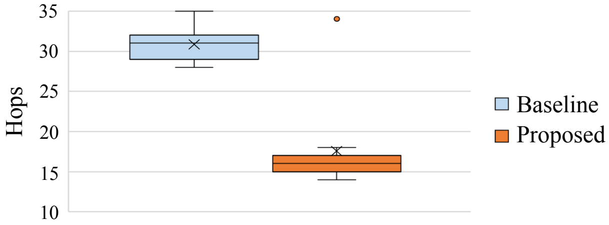

Fig. 6: Hop count of the data transfers based on the baseline and the proposed approach.

To further investigate the exchange of sovereign data, we measure the hop count of the data transfers in both approaches, because a high number of hops can be associated with delay and bandwidth bottlenecks [13], [44]. Thus, data transfers over shorter network paths (i.e., with low hop count) are likely to have more available bandwidth. In our experiments, the number of hops between a provider and a consumer is not affected by the data size, i.e., the same values apply to all data sizes, so we do not plot the hops of each data size separately. Fig. 6 shows the hop count of the two examined approaches. The baseline has an average of 31 hops (with a standard deviation of 2) and the proposed approach has an average of 18 hops (with a standard deviation of 5). This accounts for a 42% reduction in the average with all values of the proposed approach being less than the baseline, apart from the outliers. Thus, overall the proposed approach uses shorter paths than the baseline, which means that the transfer of sovereign data using our approach is likely to have more available bandwidth.

Regarding the utilization of computational resources, we note that the two examined approaches perform similarly, e.g., CPU utilization in both approaches remains below 5% with occasional spikes. This happens because the same number of data transfers is performed in both cases. The main difference between the two approaches is that the baseline requires two connectors (a provider connector and a consumer connector) to be deployed in the cloud and serve all the users. The proposed approach, on the other hand, requires two connectors (a provider connector and a consumer connector) to be deployed on-premise for each user. The former results in the connectors operating continuously to serve all the users, whereas the latter results in some connectors being idle when they do not participate in a data transfer.

VI. Conclusion

In this paper, we build a system for sovereign data exchange that relies on edge computing resources to transfer the data from a provider to a consumer. We also deploy this system in a real-world setting and we show that the proposed approach provides reduced latency and increased bandwidth, compared to a baseline that relies on the cloud. Due to the promising results, we consider that interesting future work may include investigating usage control techniques that can be implemented at the edge of the network, and examining scalability aspects.

References

[1] A. Braud, G. Fromentoux, B. Radier, and O. Le Grand, "The road to European digital sovereignty with Gaia-X and IDSA," *IEEE Network*, vol. 35, no. 2, pp. 4–5, 2021.

[2] "Government of Canada white paper: Data sovereignty and public cloud," in *https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/cloud-services/gc-white-paper-data-sovereignty-public-cloud.html*. Accessed: January 2023.

[3] C. R. Baudoin, "The impact of data residency on cloud computing," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 430–435, IEEE, 2018.

[4] K. Singi, S. G. Choudhury, V. Kaulgud, R. J. C. Bose, S. Podder, and A. P. Burden, "Data sovereignty governance framework," in *Proceedings of the International Conference on Software Engineering Workshops (ICSE)*, pp. 303–306, ACM, 2020.

[5] M. Lukings and A. Habibi Lashkari, "Data sovereignty," in *Understanding Cybersecurity Law in Data Sovereignty and Digital Governance*, pp. 1–38, Springer, 2022.

[6] "IDS reference architecture model (version 3.0)," pp. 1–118, International Data Spaces Association, 2019.

[7] F. Lauf, S. Scheider, J. Bartsch, P. Herrmann, M. Radic, M. Rebbert, A. T. Nemat, C. Schlueter Langdon, R. Konrad, A. Sunyaev, and S. Meister, "Linking data sovereignty and data economy: arising areas of tension," *Wirtschaftsinformatik Proceedings 19*, 2022.

[8] B. Otto, "A federated infrastructure for European data spaces," *Communications of the ACM*, vol. 65, no. 4, pp. 44–45, 2022.

[9] T. Usländer, M. Baumann, S. Boschert, R. Rosen, O. Sauer, L. Stojanovic, and J. C. Wehrstedt, "Symbiotic evolution of digital twin systems and dataspaces," *Automation*, vol. 3, no. 3, pp. 378–399, 2022.

[10] "Position paper: GAIA-X and IDS," pp. 1–33, International Data Spaces Association, 2021.

[11] N. Naik and P. Jenkins, "Self-sovereign identity specifications: Govern your identity through your digital wallet using blockchain technology," in *Proceedings of the International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 90–95, IEEE, 2020.

[12] D. R. Firdausy, P. D. A. Silva, M. Van Sinderen, and M.-E. Iacob, "Towards a reference enterprise architecture to enforce digital sovereignty in international data spaces," in *Proceedings of the Conference on Business Informatics (CBI)*, vol. 1, pp. 117–125, IEEE, 2022.

[13] M. Satyanarayanan, "How we created edge computing," *Nature Electronics*, vol. 2, no. 1, pp. 42–42, 2019.

[14] "White paper: Edge computing in the EuProGigant project," pp. 1–20, Institute for Production Management, Technology and Machine Tools (PTW), Technical University of Darmstadt, 2021.

[15] S. Geisler, M.-E. Vidal, C. Cappiello, B. F. Lóscio, A. Gal, M. Jarke, M. Lenzerini, P. Missier, B. Otto, E. Paja, *et al.*, "Knowledge-driven data ecosystems toward data transparency," *ACM Journal of Data and Information Quality (JDIQ)*, vol. 14, no. 1, pp. 1–12, 2021.

[16] T. Kotka, L. Kask, K. Raudsepp, T. Storch, R. Radloff, and I. Liiv, "Policy and legal environment analysis for e-government services migration to the public cloud," in *Proceedings of the International Conference on Theory and Practice of Electronic Governance (ICEGOV)*, pp. 103–108, ACM, 2016.

[17] G. Solmaz, F. Cirillo, J. Fürst, T. Jacobs, M. Bauer, E. Kovacs, J. R. Santana, and L. Sánchez, "Enabling data spaces: existing developments and challenges," in *Proceedings of the International Workshop on Data Economy (DE)*, pp. 42–48, ACM, 2022.

[18] G. S. Brost, M. Huber, M. Weiß, M. Protsenko, J. Schütte, and S. Wessel, "An ecosystem and IoT device architecture for building trust in the industrial data space," in *Proceedings of the Workshop on Cyber-Physical System Security (CPSS)*, pp. 39–50, ACM, 2018.

[19] H. Qarawlus, M. Hellmeier, J. Pieperbeck, R. Quensel, S. Biehs, and M. Peschke, "Sovereign data exchange in cloud-connected IoT using international data spaces," in *Proceedings of the Cloud Summit*, pp. 13–18, IEEE, 2021.

[20] M. Nast, B. Rother, F. Golatowski, D. Timmermann, J. Leveling, C. Olms, and C. Nissen, "Work-in-progress: Towards an international data spaces connector for the internet of things," in *Proceedings of the International Conference on Factory Communication Systems (WFCS)*, pp. 1–4, IEEE, 2020.

[21] D. Sarabia-Jácome, I. Lacalle, C. E. Palau, and M. Esteve, "Enabling industrial data space architecture for seaport scenario," in *Proceedings of the World Forum on Internet of Things (WF-IoT)*, pp. 101–106, IEEE, 2019.

[22] V. Araujo, K. Mitra, S. Saguna, and C. Åhlund, "Performance evaluation of FIWARE: A cloud-based iot platform for smart cities," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 250–261, 2019.

[23] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proceedings of the International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, IEEE, 2017.

[24] "GAIA-X: Technical architecture," pp. 1–56, Federal Ministry for Economic Affairs and Climate Action, 2020.

[25] "Eclipse dataspace components," in *https://projects.eclipse.org/projects/technology.edc*. Accessed: January 2023.

[26] W. Holfelder, A. Mayer, and T. Baumgart, "Sovereign cloud technologies for scalable data spaces," *Designing Data Spaces*, p. 419, 2022.

[27] A. Sakaino, "International collaboration between data spaces and carrier networks," in *Designing Data Spaces*, pp. 471–483, Springer, 2022.

[28] C. S. Langdon and K. Schweichhart, "Data spaces: first applications in mobility and industry," *Designing Data Spaces*, p. 493, 2022.

[29] D. Bermbach, A. Chandra, C. Krintz, A. Gokhale, A. Slominski, L. Thamsen, E. Cavalcante, T. Guo, I. Brandic, and R. Wolski, "On the future of cloud engineering," in *Proceedings of the International Conference on Cloud Engineering (IC2E)*, pp. 264–275, IEEE, 2021.

[30] Á. Alonso, A. Pozo, J. M. Cantera, F. De la Vega, and J. J. Hierro, "Industrial data space architecture implementation using FIWARE," *Sensors*, vol. 18, no. 7, pp. 1–18, 2018.

[31] "Data sovereignty as a service (DSaaS) by sovity," in *https://sovity.de*. Accessed: January 2023.

[32] V. Karagiannis, P. A. Frangoudis, S. Dustdar, and S. Schulte, "Context-aware routing in fog computing systems," *IEEE Transactions on Cloud Computing*, 2021.

[33] B. Charyyev, E. Arslan, and M. H. Gunes, "Latency comparison of cloud datacenters and edge servers," in *Proceedings of the Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2020.

[34] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *ICAS Transaction on IoT and Cloud Computing*, vol. 3, no. 1, pp. 11–17, 2015.

[35] M. Xu, Z. Fu, X. Ma, L. Zhang, Y. Li, F. Qian, S. Wang, K. Li, J. Yang, and X. Liu, "From cloud to edge: a first look at public edge platforms," in *Proceedings of the Internet Measurement Conference (IMC)*, pp. 37–53, ACM, 2021.

[36] D. Loghin, L. Ramapantulu, and Y. M. Teo, "Towards analyzing the performance of hybrid edge-cloud processing," in *Proceedings of the International Conference on Edge Computing (EDGE)*, pp. 87–94, IEEE, 2019.

[37] V. Karagiannis and S. Schulte, "Distributed algorithms based on proximity for self-organizing fog computing systems," *Pervasive and Mobile Computing*, vol. 71, p. 101316, 2021.

[38] M. Barzegaran, N. Desai, J. Qian, and P. Pop, "Electric drives as fog nodes in a fog computing-based industrial use case," *The Journal of Engineering*, vol. 2021, no. 12, pp. 745–761, 2021.

[39] V. Karagiannis and S. Schulte, "edgerouting: Using compute nodes in proximity to route IoT data," *IEEE access*, vol. 9, pp. 105841–105858, 2021.

[40] "Gaia-x hubs," in *https://gaia-x.eu/who-we-are/hubs/*. Accessed: January 2023.

[41] R. Mathumitha, P. Rathika, and K. Manimala, "Big data analytics and visualization of residential electricity consumption behavior based on smart meter data," in *Proceedings of the International Conference on Breakthrough in Heuristics And Reciprocation of Advanced Technologies (BHARAT)*, pp. 166–171, IEEE, 2022.

[42] V. Karagiannis, "Area limitations on smart grid computer networks," *International Journal of Wireless and Microwave Technologies (IJWMT)*, vol. 6, no. 3, pp. 71–78, 2016.

[43] "Refit data," in *https://repository.lboro.ac.uk/articles/dataset/REFIT_Smart_Home_dataset/2070091/1*. Accessed: January 2023.

[44] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *Proceedings of the Global Internet of Things Summit (GIoTS)*, pp. 1–6, IEEE, 2017.