Interoperable Infrastructure for Universities, Enterprises and the Public Sector

Hagen Echzell[1] and Terje Sylvarnes[2]

[1] USIT, University of Oslo
hagene@uio.no
[2] The Norwegian Digitalisation Agency
terje.sylvarnes@digdir.no

Abstract. Many organizational processes cross the boundaries of individual organizations. This is especially true for the academic sector where mobility programs like "Erasmus+" enable students to be enrolled with different universities throughout their studies. However, the technical means to enable the necessary collaboration between different institutions and their administrations are often not adequate. Many tools for group messaging, file exchange, or domain-specific applications do not work or lack functionality as soon as users or data from outside an organization are involved, such that E-Mail is a widely used fall-back. Based on terminology from the recent *Interoperable Europe Act* (IEA) [7] we will present technical and semantic *interoperability solutions* to address these issues.
On the technical side, we present two existing protocols: OpenCloudMesh is an open file sharing protocol implemented by open-source solutions like Nextcloud and ownCloud. The Matrix protocol enables secure, decentralized group and instant messaging, but might also support further use cases like video conferences in the future.
On the semantic side, we present a case study where we, based on existing vocabularies, model a university course catalog in the *Resource Description Framework* (RDF) serialization format JSON-LD. The RDF representation of the course data introduces a semantic context to the course data, making the exchange of the catalog data independent of specific APIs, thus enabling interoperability across heterogeneous data structures and university-specific systems.
The introduced technical and semantic interoperability solutions enable public sector bodies like universities and enterprises to improve their cross-organizational collaboration without giving up their digital sovereignty to a centralized, commercial actor.

Keywords: Interoperability · Federation · OpenCloudMesh · Matrix · RDF · JSON-LD

1 Introduction

Work, research, and administrative processes are rarely fully contained within a specific company or institution: Globalized value chains require cooperation among national and international companies. Research, to a substantial extent, happens in multi-institutional collaborations, often crossing the borders of countries and continents. Also, public administrations and universities must be able to deal with foreign residents and exchange students in their daily work, through cooperation and information exchange with institutions from other countries.

For these purposes, there exists a variety of juridical and organizational frameworks from different domains. As an example, when dealing with a foreign national, tax authorities can often rely on double taxation treaties between countries. In the academic sector, the "Erasmus+" mobility program provides an organizational framework for enrollment into and recognition of courses from other institutions.

On the technical level, however, even basic tasks like sending an instant message or sharing a file can be challenging in practice. It is not a new problem that technical infrastructure can turn into a number of information silos, i.e. separated systems that do not interact or that lack compatibility with each other. The prevalent situation around instant group messaging services serves as a good example of such a silo architecture: Institutions can choose to subscribe to a commercial provider in this domain, like Slack or Microsoft Teams, or they could choose to run a sovereign service on their own infrastructure. Available open-source solutions like Mattermost make setting up such systems relatively easy. Nevertheless, it is often not possible to send a simple instant message to a colleague from another institution, especially so when both institutions have chosen different solutions. There seem to be two practical solutions to this problem: Either, the users fall back to E-Mail or physical mail to communicate. Even though these two modes of communication are almost universally accepted as communication standards, they lack many features of a modern group messaging service, in particular instant messaging. The other option seems to be to agree on a common commercial provider. Both Microsoft Teams and Slack – under the keywords "shared channels" [15] and "Slack Connect" [20] – provide modes to establish group chats across organizational boundaries. Our problem seems to be easily solvable if only every institution would agree on a single commercial provider implementing such cross-institutional functionality as a de facto standard.

This second option, however, implies several mostly non-technical problems as it creates a strong dependence, or "vendor lock-in" on a single gatekeeper entity. As a result, this gatekeeper would obtain a monopoly status, shifting the economic negotiating position to the disadvantage of the organizations that would depend on access to the commercially provided communication channel. On top of that, the companies with a dominant market position are often headquartered outside of Europe, for example in the USA. The circumstances under which the transfer of personal data to the USA complies with European data protection regulations have been a matter of active dispute at the latest since the Schrems II [4] ruling, while the legal situation has changed several times in recent years as a result of new agreements and rulings.

We claim that the two options outlined above – Fallback on older technologies or modern, cross-institutional collaboration at the cost of increasing dependence on commercial gatekeepers – present a false dichotomy. We will back up this claim by introducing the concept of *interoperability* that allows for both sovereign infrastructure operation and collaboration with external companies, universities, or administrations. Interoperability is even becoming legally relevant through the recent *Interoperable Europe Act* (IEA) [7] that came into force in the European Union in July 2024. The IEA obliges public sector bodies that decide on new or substantially changed digital public services with trans-European relevance to carry out an "interoperability assessment" (Art. 3 IEA). The IEA's understanding of interoperability is further specified by the four concepts of *legal, organizational, semantic, and technical interoperability* and implemented through so-called *interoperability solutions*. In this paper, we will present technical specifications, standards, and reference implementations that qualify as such interoperability solutions. In the first sections, we will focus on the basic tasks of file sharing and group messaging. Finally, we will present a case study on the semantic and technical interoperability of universities' course catalogs. For that, we will use the *Resource Description Framework* (RDF) [27], a W3C standard for graph-structured data interchange and description.

Our research was originally motivated by questions about the common IT infrastructure of a European coalition of universities, as well as the IEA requirements for trans-European public sector services. However, the interoperability solutions and concepts we present would also be applicable as a blueprint in almost any other cross-organizational collaboration context, for example where private sector organizations cooperate or where different municipalities need to exchange information.

2 A basic structure for interoperability

It makes sense to distinguish between two concepts of interoperability, that is *horizontal* and *vertical interoperability* that we outline in Fig. 1. In our context, vertical interoperability can be defined as the possibility of using different client programs to access a given service. This property is often achieved through a *Client-Server-API* or protocol. Examples are POP or IMAP which provide a standard way of communication between E-Mail clients and servers, thereby decoupling the choice of a local E-Mail program from the provider of the service. We will however have our main focus on the other concept which is horizontal

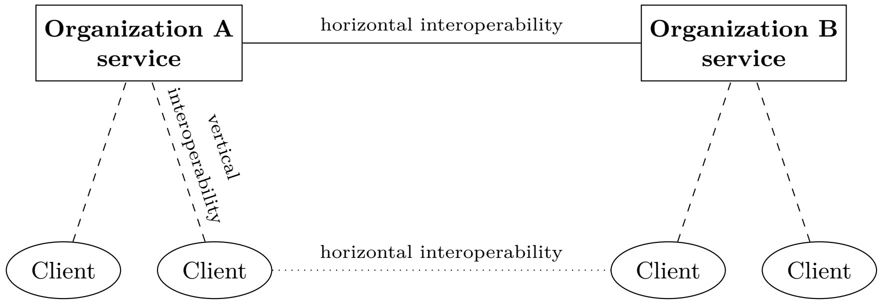

Fig. 1. Organizations A and B are each running a service. We can define vertical interoperability such that it enables their users to use a client program of their choice. Horizontal interoperability, on the other hand, makes two similar services work with each other, even across domains and organizations. Horizontal interoperability – in the technical sense – can be achieved through a Server-Server-API or protocol. Some encrypted forms of horizontal interoperability require the clients to be interoperable with each other as well. We will cover this topic in Section 4.1.

interoperability. Again, E-Mail serves as an example: The SMTP protocol allows E-Mail servers from different vendors to "talk to each other", which enables sending E-Mail across separate domains, i.e. between different institutions and companies. Such a standard way of interfacing between two or more services can be achieved through a defined *Server-Server-API* or protocol. A network of many such horizontally interoperating systems can be called a *federation*.

Fig. 2 shows a generic example of such a federation. Users Alice, Bob, Carol, and Dan can use a client that implements a specific Client-Server-API to connect to their respective servers or service providers. The servers – here "matrix.org", "ergaster.org" and "example.com" – exchange data according to some Server-Server-API. Such a federated architecture would allow Alice and Bob, who might be users from different organizations, to exchange information without relying on a centralized service provider that would act as a gatekeeper.

3 Federated file exchange

One basic use case of cross-organizational collaboration is the exchange of data in the form of files. While in many sectors it is common practice to send files as an E-Mail attachment, this approach has a couple of downsides: E-Mail does not fulfill many modern security and privacy standards by default, like fine-grained access controls or defined retention periods. Additionally, collaborating on a file like a document requires sending intermediate versions back and forth between the collaborating parties and merging any changes manually. There are tools like Google Docs and the like that have the potential for a more streamlined and structured way of collaborating on files. However, these are often centralized

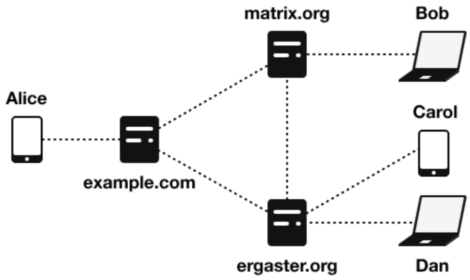

Fig. 2. Users use client programs that implement a specific Client-Server-API to connect to a server. Their respective servers communicate to each other via a Server-Server-API. Both APIs are represented by dotted lines. The image comes from matrix.org [23] where it is used to describe the Matrix standard, but it is general enough to illustrate a generic federated network.

services that do not allow horizontal interoperability. Open-source solutions like Nextcloud [1] or ownCloud [2] provide similar functionality to share files with other users and keep them in sync. The companies behind these projects advertise their products with a focus on data sovereignty: As the core products are open-source, companies or public sector bodies can deploy them in their own data centers without any vendor lock-in or reliance on commercial platforms. However, such a setup – even if it is based on open-source components – does not automatically provide interoperability. Nevertheless, for this purpose, there is the OpenCloudMesh protocol.

3.1 OpenCloudMesh as a federated file sharing protocol

In 2015, ownCloud, Inc. unveiled the *OpenCloudMesh* (OCM) project under the umbrella of GÉANT, an association that coordinates European National Research and Education Networks. An initial version of OCM was released in 2020 as an open standard, while the current version is v1.2.0 [8]. Co-founder Frank Karlitschek announced: “OpenCloudMesh gives each organization private cloud file sync and share, while Federated Cloud sharing, also known as server-to-server sharing, enables safe sharing between those clouds.” [3] In more technical words, OCM is a Server-Server-API that defines a sending and a receiving server. The sending server creates a shared resource, usually a WebDAV share which contains a number of files. It then notifies the receiving server, which in turn can provide access to these files to one or some of their users. From the users’ perspective, this process is hidden behind a comprehensible interface. If a user wants to give read or write access to a folder to a user from another organization, they need to ask the receiving user for their “OCM Address” – which Nextcloud calls “Federated Cloud ID” – in the format `username@example-organization.com`.

In the Nextcloud user interface, the process of sharing a file with such an OCM Address is almost identical to creating a regular share with another local user.

OCM is a Server-Server-API that provides for horizontal interoperability. This means, that as long as organizations each run a service that implements and enables OCM – and notably even if these are services made by different vendors, like ownCloud, Nextcloud, or SeaFile [19] –, their users will be able to share files and folders with each other, across the boundaries of their own organization. OCM can be seen as a technical interoperability solution as defined by the IEA.

4 Federated group messaging

Parallel to file sharing, users may have a need to send instant messages to users of other institutions or to create a common channel for group messaging. We already covered this use case and potential solutions and downsides as an illustrative example in our introduction. To repeat the most important points, institutions that use Microsoft Teams or Slack have the option to enable group chats across organizations, at least where both organizations use the exact same service from the same provider. One could argue that these services could therefore be seen as interoperability solutions. However, we already outlined the downsides of this approach: As long as vendors refuse to open their closed ecosystems and publish an open, documented Server-Server-API that allows for federation, any vendor with a sufficient market share in the European Market becomes a gatekeeper with a dominant negotiation position. In the hypothetical case where all public institutions converge on a single such service provider, this provider would become a monopolist who might abuse their position.

With the *Digital Markets Act* (DMA) that came into force in 2022, [5] the European Union has a modern digital antitrust law that is intended to weaken the dominant position of such gatekeepers. In particular, Art. 7 of the DMA introduces horizontal interoperability obligations for several gatekeepers that operate so-called *number-independent interpersonal communications services* (NIICS) – that is, broadly speaking, instant and group messaging services like WhatsApp and Facebook Messenger. The DMA opens up an interesting discussion on technical and semantic interoperability: Which kind of standard would enable horizontal interoperability between different messaging services? A promising candidate for a suitable interoperability solution is the Matrix protocol which not only allows modern group and instant messaging in a federated manner but additionally provides for End-to-End-Encryption (E2EE) between users.

4.1 Using the Matrix protocol for federated messaging

According to the Matrix protocol specification, "Matrix defines a set of open APIs for decentralised communication, suitable for securely publishing, persisting and subscribing to data over a global open federation of servers with no single point of control" [24]. This description is very broad, and in fact the specification mentions different use cases, from messaging over *Voice over IP* (VoIP) to Internet of Things applications. In this chapter, we will focus on the messaging use case. Unlike the previously mentioned OCM, the Matrix specification also defines a Client-Server-API in addition to a Server-Server-API. Both these APIs are mainly based on JSON objects that are exchanged via TLS-encrypted and signed HTTP. Matrix is quite a complex standard, and we will only outline the basic concepts: Matrix defines the concept of *rooms* that one or many users can join and send *events* like messages, files, or media in. In the context of these rooms, two or more Matrix servers with users in the same room exchange persistent data with each other, like messages, metadata about a room's members, the room's name, etc. Using a defined state resolution algorithm, servers merge these data to a graph-representation of the room data. Additionally, Matrix servers exchange non-persistent ephemeral data like typing notifications. An interesting feature of Matrix is that it provides for optional *End-to-End-Encryption* (E2EE) inside these rooms. This means that if a user enables E2EE upon the creation of a room, servers never have access to the clear-text messages, even if they become compromised. The encryption algorithm is called *Olm* and is based on the double ratchet algorithm made popular by the encrypted messenger "Signal". For scalability reasons, the specification includes an extended algorithm called *Megaolm* for large rooms. E2EE makes Matrix especially suitable for use cases with higher confidentiality requirements. This functionality also explains why Matrix must include a Client-Server-API: To enable E2EE-capable horizontal interoperability, we need the clients to behave in a particular way since essential cryptographic functionality is delegated to the client.

For both Matrix servers and Matrix clients, there exists a diverse ecosystem of open-source software. The most popular reference applications are the Python-based "Synapse" on the server side, as well as "Element" on the client side. [12] Element can be used from the web browser or as a native application for the most popular operating systems. An organization that would like to use Matrix would have to host a Synapse server – or some alternative implementation – and allow their users to log in through a client app of their choice. After exchanging their "Matrix User ID" in the format `@username:example-organization.com`, users can exchange instant messages or start group chats with users from other organizations. Except for a slightly different username syntax, the general user experience is very similar to that of popular applications like Slack.
Moreover, the Matrix specification already covers basic VoIP use cases like telephony and video calls between individual users. Further VoIP functionality, e.g. an extension to the specification that would enable video conferences [14] is currently being worked on.

For completeness, we will mention several alternative approaches: With *Message Layer Security* (MLS) [9], as well as *More Instant Messaging Interoperability* (MIMI) [6], the *Internet Engineering Task Force* (IETF) develops a set of standards with similar goals as Matrix. At the same time, Nextcloud very recently announced support for federated messaging and calls [16]. We could however not find any open API or protocol describing how to implement this functionality outside of Nextcloud. One more noteworthy standard is the *Extensible Messaging and Presence Protocol* (XMPP) [28], a federated IETF standard dating back to 2002 that, through a vast number of extensions, in fact, provides much of the same functionality as Matrix. Nowadays, we do not see XMPP being used a lot by public or private sector bodies. While there are speculations that the many optional extensions contributed to a rather fragmented ecosystem, we can not conclude with certainty why XMPP seems to be rarely used. The reason we specifically present Matrix as an interoperability solution is that there is an active ecosystem of universities (see Fig. A.1) and public sector bodies in Europe that use Matrix [10,11,13].

5 Case study: Interoperable course catalogs

As we mentioned, our research was motivated by questions about the common IT infrastructure of a European university alliance. In this context, several European universities would like to present their courses as part of a common course program that students can freely choose from. Two possible solutions to make the common course data available are integrating each other's courses into the universities' own catalogs, or a common course catalog platform that presents details about all the courses in one central place. However, existing course catalog systems have often grown over time, were adapted to local needs, and usually use different database schemas and endpoints. One university might present a course as a server-generated HTML page (Listing A.1), another university might have a homegrown JSON-based API, perhaps even with key names in a non-English language (Listing A.2). This heterogeneity poses a problem for federated catalogs or a common catalog platform.

On a first look, it seems like establishing interoperability would require all the participating universities to agree on a common Server-Server-API like in the examples before. Such an API enables technical horizontal interoperability but comes with its own set of problems. Implementing the API would require a far-reaching refactoring of existing infrastructure and front-ends that depend on the affected endpoints. Alternatively, universities could build and maintain the common API in addition to their existing endpoint, but this can be costly and comes with an additional maintenance burden. In addition, agreeing on a common API, which should be stable and backward compatible in the long term, might not be feasible.

At this point, we introduce the concept of *semantic interoperability*: the required course information is already publicly available for students and administrative staff to parse, understand and include in their respective organizational processes. Therefore, we do not necessarily need a new API, but rather a method to make the existing information machine-readable and understandable across universities' contexts. For our use case, we present a solution that is based on the *Resource Description Framework* (RDF) [27].

RDF is a W3C standard that defines a framework for representing machine-readable information on the Web. RDF data consists of statements, so-called triples, in the form of a Subject-Predicate-Object. Seen as a graph, the subject and object are the nodes, while the predicate is the edge between these nodes. The entities in the triple are either primitives like integers or strings, called literals in RDF, or global and unique URIs pointing to other entities. Using RDF, the semantics, i.e. the meaning of the data is embedded in the data structure itself, increasing interoperability between systems that communicate based on the same RDF vocabularies. Such an RDF vocabulary is a data model consisting of classes, properties, and their relations that we can use to describe our data. JSON-LD [17] is one format used to express RDF data. Since JSON-LD is simply JSON data with a semantic context, it gives us a suitable way of adding meaning to the JSON course data without disrupting the existing API. Another option is to use RDFa [21] to embed structured data in existing HTML documents.

With these concepts in mind, we can annotate the JSON data (Listing 1) and HTML templates (Listing A.3) that generate course sites, even without impacting compatibility with existing front-ends and other infrastructure. We have based the annotations on the `Course` type from the *Schema.org* vocabulary [18], which provides several properties we can use to describe the course data, such as name, description, number of credits given, etc; in principle, any vocabulary which covers the necessary concepts for this use case would work. A prerequisite is that all universities agree on the same vocabulary, or use vocabularies that can be mapped to each other.

When universities' endpoints start serving semantically annotated data, as seen in the examples, it is quite easy to query course links to extract and merge data from these formerly heterogeneous data sources. The data exchange is now independent of specific APIs, and any additional universities who want to participate in the data exchange of course data need only to publish their course data according to the common vocabulary. In this sense, the data is now interoperable. If we add even more annotated data on provenance, licenses, etc., and use predefined codes and controlled vocabularies for the description of the data itself, the data will become even more reusable for external systems. But there is one open question: how do consumers of the data, such as a common course catalog platform, know all the relevant course URIs or IDs without a costly and potentially incomplete crawling of the other universities' websites?

For the purpose of findability, we present the W3C standard *Data Catalog Vocabulary* (DCAT), an RDF-based framework to create interoperability for whole data catalogs. We can treat each collection of course data as a DCAT dataset and generate a DCAT catalog listing of all the courses and where to access and download the course data. For the course "Semantic Technologies" given

```
{
    "@context": {
        "schema": "https://schema.org/",
        "kd": "schema:courseCode",
        "vekting": {
            "@id": "schema:numberOfCredits",
            "@type": "schema:Integer"
        },
        "navn_en": {
            "@id": "schema:name",
            "@language": "en"
        },
        "noName": {
            "@id": "schema:name",
            "@language": "no"
        },
        "dsc": {
            "@id": "schema:description",
            "@container": "@language"
        },
        "prreqs": "schema:coursePrerequisites"
    },
    "@type": "schema:Course",
    "@id": "https://www.uis.no/en/course/EDAT502_1",
    "kd": "E-DAT502_1",
    "vekting": 5,
    "navn_en": "Semantic technologies",
    "noName": "Semantiske teknologier",
    "dsc": {
        "en": "The course gives an introduction to the fundamentals of semantic technologies [...]",
        "no": "Kurset gir en innføring i grunnleggende semantiske teknologier [...]"
    },
    "prreqs": "Diskret matematikkGrunnleggende programming"
}
```

Listing 1: With help of the JSON-LD `@context` property we can map the existing JSON keys to `Schema.org/Course` properties, adding semantic meaning to each key. Note that the original keys are still in place for compatibility with existing front-ends. You can test the snippet with the JSON-LD Playground (https://json-ld.org/playground/) to see the expanded data structure.

at the University of Stavanger, this could look like Listing A.4. The property `dcat:downloadURL` points to the endpoint that serves the actual course data, in our example, this would be the URL which is the ID of the Semantic Technologies course. With `dcterms:format` and `dcterms:conformsTo` we specify the format as JSON-LD and that the data model conforms to the `Course` type from Schema.org. This enables a common course platform to distinguish the different formats and handle them correctly before presenting the data to users in a streamlined way. When the university includes its publicly available courses as part of its DCAT catalog, a common course platform would now only need to know about the endpoints serving the data catalog and use the information retrieved from there to access all course data.

5.1 Interoperable Transcript of Records

The *Transcript of Records* is a common document in the academic sector. Such a document is issued by a university and shows – not necessarily complete – information about the particular courses a student has passed. To have passed courses recognized at a student's home university, a transcript might be issued by the foreign university after a stay abroad. A Transcript that includes all the passed courses must also be issued when a student finishes their degree. In the context of upcoming university alliances, it will be much more common that universities need to merge such Transcript data from different sources, to create a final complete Transcript that includes all the courses a student has passed at other universities. It should therefore be possible to merge the relevant data by automated means, which requires interoperable data. Based on the principles outlined in this chapter, it is quite easy to create an adequate data structure or to semantically annotate an existing endpoint. Listing A.5 shows a machine-readable example Transcript.

5.2 Considerations

Privacy, trust, and security: As for any data exchange, privacy, trust, and security must be considered. Student data is personal data. It must therefore be handled according to the GDPR. Each consumer must be able to trust that the resources received from another institution are valid and up-to-date. In addition, the data exchange must be done securely. Since the RDF-based interoperability solutions we present in this paper are based on the web stack, systems can leverage existing approaches for maintaining trust and security.

6 Conclusion

This paper about interoperability in the private and public sector was motivated by the insight that many organizational processes cross the boundaries of individual organizations, not only, but especially so in the academic sector. However,

while there exist many organizational and juridical frameworks to facilitate cooperation between separate organizations, whether they are from the same or different countries, there is a need for modern technical and semantic means of collaboration. With the Interoperable Europe Act (IEA) [7], the European Union coins the concept of an interoperability solution. One of the most universally implemented technological interoperability solutions seems to be E-Mail: As E-Mail is based on the SMTP protocol, different organizations can choose a service provider of their choice and send messages to others, independently of their selection of provider.
Nevertheless, for many use cases, E-Mail is not an adequate tool. With platforms like Microsoft Teams, Slack, Google Docs and many more, modern collaboration tools have emerged. These, however often lack interoperable qualities: Organizations that want to collaborate need to choose the same service provider for these tools as their partners.

To illustrate ways to relieve this seeming dichotomy, in Section 2 we introduced the notion of horizontal interoperability: A concept that allows services made by different vendors to work together. Where many such interoperable services form a network, we can speak of a federation.

In Sections 3 and 4 we presented the standards OpenCloudMesh (OCM) and Matrix, two technical interoperability solutions that enable horizontal interoperability for their respective use cases. OpenCloudMesh allows two or more organizations to share and sync files with each other, without relying on a central authority. There are different implementations for this protocol: ownCloud, Nextcloud, and Seafile each can create and join OCM shares [19]. Our section on Matrix focuses on the use cases of group and instant messaging. Organizations that base their messaging on a Matrix-compliant service can easily and securely send instant messages to collaboration partners from other organizations, or create a common message "room". Again, there is no need for a common service provider.

For both OCM and Matrix, there are open-source reference implementations. This is particularly relevant as Article 4 of the IEA [7] expects public sector bodies to prioritize interoperability solutions that are available under an open-source license.

In our last section, we presented a case study: Universities that join a common university alliance would like to exchange structured data about the courses they offer. While the information they hold about these courses is similar in principle, heterogeneous APIs, database schemas and interfaces pose a problem for interoperability. Even though we could have presented a technical solution by defining a particular API, we decided to leverage the concept of semantic interoperability. Universities can semantically annotate their existing APIs and websites with respect to a common RDF vocabulary. On this conceptual basis, data from different sources can be flexibly extended, linked, and merged, and be repurposed for further use cases like a Transcript of Records.

The presented interoperability solutions are by no means limited to the academic sector. Municipalities, small and large enterprises, and governments, all of them need to communicate with other public and private sector bodies and exchange structured or unstructured data with each other. We therefore recommend taking the aspect of horizontal interoperability into account when discussing new or to-be-updated IT solutions, whether an organization is subject to the IEA or not.

Any of our interoperability solutions have one thing in common: Collaborating organizations should agree on common standards, protocols, or vocabularies. Even though it is often possible to mediate, or "bridge" [22], between different, but similar protocols or vocabularies, common standards will lead to less fragmentation in the ecosystem and to easier adoption of new features through versioned and updated protocols or vocabularies. How to create consensus among public sector bodies or across industrial sectors is not a technical question, but a political and organizational one. It is of course possible to arrange formal committees to create such a consensus. But there are also notable efforts like "openDesk" [13], a sovereign collaborative workplace platform for the German public administration. Projects like this one, led by the Federal Ministry of the Interior and Community might create a certain momentum and therefore contribute to an informal consensus on protocols for certain use cases, also in other countries. openDesk has a specific focus on interoperability, its basic version includes Nextcloud and Element, and it therefore supports both OCM and Matrix.

6.1 Further research

There are many relevant use cases for interoperability that we could not cover in this paper. Having federated calendars is one of these. With CalDAV, there exists a protocol for vertical calendar interoperability. Just like OCM extends WebDAV to a standard that makes user-friendly horizontal interoperability possible, we would like to see a similar prototype based on CalDAV to share calendars across services.

A topic that would justify its own paper would be a survey about federated identity services: How can a user from one organization get access or login to some generic resource or service from another organization using their existing user account? There are many different real-world approaches and standards that are relevant to this question.

One issue that blurs the boundaries of vertical and horizontal interoperability is the question of data formats. File exchange only provides value where all parties are capable of opening and modifying the received files. Sending a file or document in a proprietary file format might pose operating system or software requirements on the receiver that are contrary to the objective of interoperability. Mapping the needs and solutions for interoperable file formats – possibly beyond the common document, table, and presentation files and formats – would be a relevant next step for the public sector and, in a domain-specific manner, for private sector bodies from different industries. Regarding domain-specific data formats, one could consider a semantic approach and define or survey specific RDF-vocabularies for use in various industries and sectors.

References

1. GitHub – Nextcloud – A safe home for all your data, https://github.com/nextcloud, accessed: 2024-09-21
2. GitHub – ownCloud, https://github.com/owncloud, accessed: 2024-09-21
3. ownCloud Initiates Global Interconnected Private Clouds for Universities and Researchers (Jan 2015), https://owncloud.com/news/owncloud-initiates-global-interconnected-private-clouds-universities-researchers/, accessed: 2024-09-21
4. Judgment of the Court (Grand Chamber) of 16 July 2020. Data Protection Commissioner v Facebook Ireland Limited and Maximillian Schrems. No. Case C-311/18 (Jul 2020), https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62018CJ0311
5. Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act) (Text with EEA relevance) (Sep 2022), https://eur-lex.europa.eu/eli/reg/2022/1925/oj
6. More Instant Messaging Interoperability (mimi) – Charter for Working Group. Tech. Rep. charter-ietf-mimi-01, Internet Engineering Task Force (Feb 2023), https://datatracker.ietf.org/wg/mimi/about/, accessed: 2024-09-20
7. Regulation (EU) 2024/903 of the European Parliament and of the Council of 13 March 2024 laying down measures for a high level of public sector interoperability across the Union (Interoperable Europe Act) (Mar 2024), http://data.europa.eu/eli/reg/2024/903/oj/eng
8. GitHub – OpenCloudMesh API (Mar 2025), https://github.com/cs3org/OCM-API, accessed: 2025-04-14
9. Barnes, R., Beurdouche, B., Robert, R., Millican, J., Omara, E., Cohn-Gordon, K.: The Messaging Layer Security (MLS) Protocol. RFC 9420 (Jul 2023). https://doi.org/10.17487/RFC9420
10. Element (New Vector Ltd): Element – Sectors, https://element.io/sectors, accessed: 2024-09-20
11. Element (New Vector Ltd): Element – Universities collaborate on Matrix, https://element.io/matrix-in-germany/projects/universities, accessed: 2024-09-20
12. Element (New Vector Ltd): GitHub – Element, https://github.com/element-hq, accessed: 2024-09-20
13. German Federal Ministry of the Interior and Community (BMI): Open CoDE – openDesk (May 2024), https://gitlab.opencode.de/bmi/opendesk/info/-/blob/main/README_EN.md#basic-version, accessed: 2024-09-25
14. Hodgson, M.: GitHub – MSC3401: Native Group VoIP Signalling, https://github.com/matrix-org/matrix-spec-proposals/pull/3401, accessed: 2024-09-21
15. jacktremper, MikePlumleyMSFT, denisebmsft, v-mathavale, alekyaj, msbemba, v-dihans, chrisda, robmazz, chrfox: Plan external collaboration

with channel conversations, file collaboration, and shared apps (Dec 2023), https://learn.microsoft.com/en-us/microsoft-365/solutions/plan-external-collaboration?view=o365-worldwide, accessed: 2024-07-31
16. Korotaev, M.: Be connected with Nextcloud Hub 9 (Sep 2024), https://nextcloud.com/blog/nextcloud-hub9/, accessed: 2024-09-20
17. Longley, D., Kellogg, G., Champin, P.A.: JSON-LD 1.1. W3C recommendation, W3C (Jul 2020), https://www.w3.org/TR/2020/REC-json-ld11-20200716/
18. Schema.org: Course – A Schema.org Type (Jul 2024), https://schema.org/Course, accessed: 2024-09-14
19. Schießle, B.: OCM - from the initial design to a core component of Nextcloud Hub (Jan 2021), https://indico.cern.ch/event/970232/contributions/4184483/, accessed: 2024-09-21
20. Slack Technologies: Slack Connect – Communicate and Collaborate Externally with Clients, https://slack.com/connect, accessed: 2024-07-31
21. Sporny, M., Adida, B., Herman, I., Birbeck, M.: RDFa 1.1 Primer - Third Edition. W3C note, W3C (Mar 2015), https://www.w3.org/TR/2015/NOTE-rdfa-primer-20150317/
22. The Matrix.org Foundation C.I.C.: matrix – Bridges, https://matrix.org/ecosystem/bridges/, accessed: 2024-09-25
23. The Matrix.org Foundation C.I.C.: Elements of Matrix (Feb 2023), https://matrix.org/docs/matrix-concepts/elements-of-matrix/, accessed: 2024-09-21
24. The Matrix.org Foundation C.I.C.: Matrix specification – version v1.14 (Mar 2025), https://spec.matrix.org/v1.14/, accessed: 2025-04-14
25. University of Oslo: IN3060 – Semantiske Teknologier (2024), https://www.uio.no/studier/emner/matnat/ifi/IN3060/index.html, accessed: 2024-09-14
26. University of Stavanger: Semantiske teknologier (2023), https://www6.uis.no/Stine/EMNE/2023/H%C3%98ST/Bokm%C3%A51/E-DAT502_1.pdf, accessed: 2024-09-14
27. Wood, D., Lanthaler, M., Cyganiak, R.: RDF 1.1 Concepts and Abstract Syntax. W3C recommendation, W3C (Feb 2014), https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/
28. XMPP Standards Foundation: XMPP – Specifications, https://xmpp.org/extensions/, accessed: 2024-09-21

A Appendix

Figures

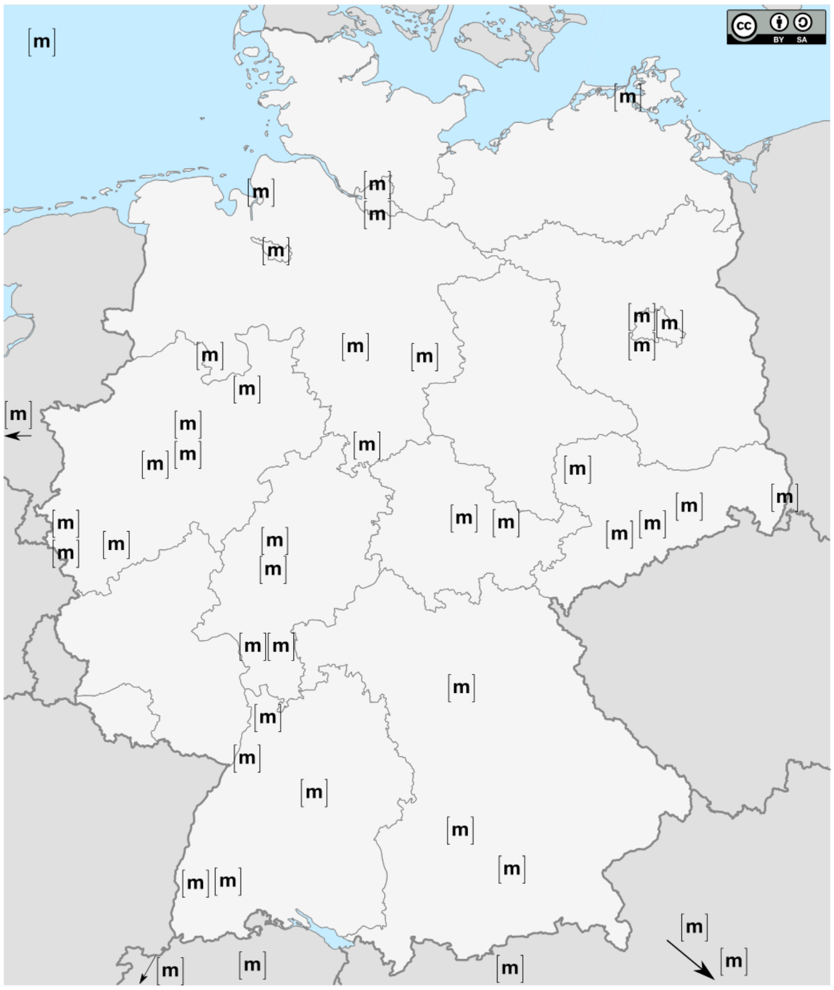

Fig. A.1. Matrix is used in many European universities [11].
CC BY-SA 4.0 TU Dresden
https://creativecommons.org/licenses/by-sa/4.0/legalcode

Listings

```html
<body>
 <main>
  <h1>IN3060 - Semantiske teknologier</h1>
  <h2>Kort om emnet</h2>
  <p>"Semantic Web" (SW) er en spennende ny utvikling for fremtidens WWW
  ↪  [...]</p>
  <h2>Hva lærer du?</h2>
  <p>Etter å ha tatt IN3060:</p>
  <ul>
   <li>har du oversikt over grunnleggende standarder (RDF, RDFS, OWL,
   ↪  SPARQL) [...]</li>
  </ul>
  <h2>Obligatoriske forkunnskaper</h2>
  <p><a
  ↪  href="https://www.uio.no/studier/emner/matnat/ifi/IN1010/index.html">
    IN1010 - Objektorientert programmering</a>.</p>
  <h2>Fakta om emnet</h2>
  <dl>
   <dt>Nivå</dt> <dd>Bachelor</dd>
   <dt>Studiepoeng</dt> <dd>10</dd>
  </dl>
 </main>
</body>
```

Listing A.1: An HTML document about the course “IN3060 – Semantiske teknologier” [25] at the University of Oslo. The HTML was taken from the University of Oslo’s website and edited by us for readability.

```json
{
    "kd": "E-DAT502_1",
    "vekting": 5,
    "navn_en": "Semantic technologies",
    "noName": "Semantiske teknologier",
    "dsc": {
        "en": "The course gives an introduction to the fundamentals of
        ↪  semantic technologies [...]",
        "no": "Kurset gir en innføring i grunnleggende semantiske
        ↪  teknologier [...]"
    },
    "prreqs": "Diskret matematikkGrunnleggende programming"
}
```

Listing A.2: A JSON representation of the course “Semantic Technologies” [26] at the University of Stavanger. While the course data is taken from the University of Stavanger, we fabricated the JSON scheme for illustrative purposes.

```html
<body vocab="https://schema.org/">
 <main typeof="Course">
  <h1 property="name"><span property="courseCode">IN3060</span> - Semantiske teknologier</h1>
  <h2>Timeplan, pensum og eksamensdato</h2>
   <a property="hasCourseInstance" typeof="CourseInstance" href="https://www.uio.no/studier/emner/matnat/ifi/IN3060/v24/">Vår 2024</a>
  <h2>Kort om emnet</h2>
  <p property="description">"Semantic Web" (SW) er en spennende ny utvikling for fremtidens WWW [...]</p>
  <h2>Hva lærer du?</h2>
  <p>Etter å ha tatt IN3060:</p>
  <ul>
   <li>har du <span property="teaches">oversikt over grunnleggende standarder (RDF, RDFS, OWL, SPARQL) [...]</span></li>
  </ul>
  <h2>Obligatoriske forkunnskaper</h2>
  <p><a property="coursePrerequisites" typeof="Course" href="https://www.uio.no/studier/emner/matnat/ifi/IN1010/">IN1010 - Objektorientert programmering</a>.</p>
  <h2>Fakta om emnet</h2>
  <dl>
   <dt>Nivå</dt> <dd property="educationalLevel">Bachelor</dd>
   <dt>Studiepoeng</dt> <dd property="numberOfCredits">10</dd>
  </dl>
 </main>
</body>
```

Listing A.3: HTML: After slightly adapting the HTML course template, web-crawlers with access to the course page can extract structured, semantic data about the course. Notice the added "property" attributes, which use predefined properties from `Schema.org/Course`. You can test the snippet with the Schema.org validator (https://validator.schema.org/) to see the structured data.

```json
{
  "@context": {
    "dcterms": "http://purl.org/dc/terms/",
    "dcat": "http://www.w3.org/ns/dcat#",
    "title": {
      "@id": "dcterms:title",
      "@container": "@language"
    },
    "catalog": {
      "@reverse": "dcat:dataset"
    }
  },
  "@id": "https://data.uis.no/course/E-DAT502_1",
  "catalog": {
    "@id": "https://www.uis.no/course-catalog"
  },
  "title": {
    "nb": "Semantiske teknologier",
    "en": "Semantic technologies"
  },
  "dcterms:publisher": "Universitetet i Stavanger",
  "dcat:distribution": {
    "@type": "dcat:Distribution",
    "dcat:downloadURL": "https://www.uis.no/course/E-DAT502_1",
    "dcterms:format": "http://publications.europa.eu/resource/authority/file-type/JSON_LD",
    "dcterms:conformsTo": "https://schema.org/Course"
  }
}
```

Listing A.4: Dataset description of the course data, using DCAT, and in the JSON-LD format

```
{
  "@context": {
    "schema": "https://schema.org/"
  },
  "@type": "schema:Person",
  "@id": "https://www.uis.no/en/students/xxxxxxxxx",
  "schema:name": "Jane Doe",
  "schema:hasCredential": [
    {
      "@type": "schema:EducationalOccupationalCredential",
      "@id": "https://data.uis.no/course/E-DAT502_1"
    }
  ]
}
```

Listing A.5: An example Transcript of Records in JSON-LD format. Note that the course "Semantic technologies (E-DAT502_1)" at the University of Stavanger is identified by its URI. With access to the data catalog from Listing A.4, we can automatically retrieve and merge all relevant information in a semantic format. Again, we fabricated the JSON scheme for illustrative purposes.