Design Considerations for Technical Interoperability in EOSC

Authorship community:

George Kakaletris[2,3] (0000-0002-2095-1220), Eva Sciacca[1,7,*] (0000-0002-5574-2787), Jean-Karim Hériché[2,8] (0000-0001-6867-9425), Wim Hugo[2,6] (0000-0002-0255-5101), Emanouil Atanassov[2,4] (0000-0002-7442-7096), Svetlozar Yordanov[2,5] (0000-0001-7671-6804)

[1] Co-Chair, EOSC Task Force on Technical Interoperability of Data and Services
[2] Member, EOSC Task Force on Technical Interoperability of Data and Services
[3] Communication & Information Technologies Experts SA (CITE), Greece
[4] Institute of Information and Communication Technologies (IICT-BAS), Bulgaria
[5] BAS, Bulgaria
[6] Data Archiving and Networked Services - Royal Netherlands Academy of Arts and Sciences (KNAW), Netherlands
[7] National Institute for Astrophysics (INAF), Italy
[8] European Molecular Biology Laboratory, Germany

* Corresponding author: eva.sciacca@inaf.it
** All TF members have had the opportunity to review and edit this document; invitations to be on the authorship list were open to all

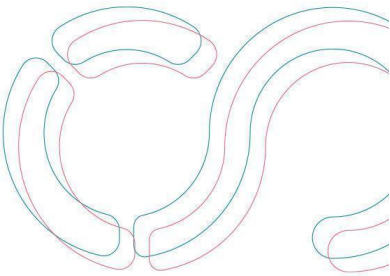
EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Table of contents

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

EOSC Association AISBL

Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931



Executive summary

This document is one of the deliverables of the EOSC Association (EOSC-A) Task Force on “Technical Interoperability of Data and Services”. Its main aim is to highlight a number of design principles and suggestions that increase interoperability and composability opportunities for data, software and services, and it is intended to complement the forthcoming Task Force deliverable entitled, "A landscape overview of the EOSC Interoperability Framework: Capabilities and Gaps".

The document introduces design-stage principles, both architectural and non-architectural, that are commonly met in software design and maintenance best practices, but are in this case approached from the perspective of EOSC’s technical interoperability. These 21 principles are additionally catalogued in Annex A in tabular form including the guidelines and recommendations of this report, together with the suggested criteria for satisfying the respective principles.

The document then proceeds to specify suggestions for the interoperability of EOSC services, addressing the pillars of Open Science, Cloud Computing and the User. The role of standards, as the cornerstone of system interoperability, is presented briefly, and Annex B contains an evidence-based enumeration of the relevant specifications. In the Annex, an indicative list of standard families, along with sources where common standards are drawn for EOSC services, is presented. In conclusion, a set of key points is summarised, that emphasises the need to adopt an interoperability strategy *at the time of design* of a given system.

EOSC Association AISBL

Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

1. Introduction

Interoperability is a term that lacks a concrete single definition, the most common one being "the ability of systems to exchange and make use of information"[1], which is altogether too vague to produce objectively measurable results. More often than not, in the field of information and communication technologies, interoperability is focused on the definition and adoption of common data standards and communication protocols which may address challenges at different levels (the lower ones being foundational, structural and semantic). This approach, which might have been the way to approach interoperability in earlier eras, when software was substantially simpler in its interactions and capabilities, does not suffice today. Having a comprehensible data format and protocol to exchange those data does not automatically assure that this can be integrated in the processes of a system unless special measures are taken at time of design. Moreover, interoperability may be a straightforward task when addressing a single protocol, but it becomes a serious challenge when software is required to interoperate with processes that are subject to change – with emerging and continuously evolving systems, or those assuming multiple complementary or competing protocols.

While interoperability is relevant in all areas of modern software design and implementation domains, for the European Open Science Cloud (EOSC) it is a foundational requirement. The reason is that EOSC envisages an ecosystem where barriers for connecting or repeating steps of scientific processes will be as low as possible to allow all the aspects of Open Science to flourish. In an ideal future, all processes could be interwoven or daisy-chained into more complex processes, or accommodate variable component inputs, able to discover and consume any relevant data form, cross any authoritative domain, utilise any virtualized resource that may be available to them, all while respecting the terms upon which those resources are exposed. However, methodologies, instruments, processes, dataforms, resources, etc., are literally unlimited and continuously evolving, each accompanied by different operational and technological requirements, which makes it impossible to have one interoperability solution that fits all. In this landscape, technical interoperability of systems – the focus of the Task Force (TF) –becomes an architectural challenge rather than a mere mandate to align specifications.

This document enumerates and describes a set of design principles that may increase the interoperability and composability of software and services relevant to EOSC. Whether or not these design principles are so far anticipated by the EOSC Interoperability Framework and related EOSC projects is the aim of a forthcoming deliverable of the TF entitled, "A landscape overview of the EOSC Interoperability Framework: Capabilities and Gaps".

[1] ETSI Interoperability Best Practices
https://www.etsi.org/images/files/Events/interoperability_best_practices_handbook.pdf

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Note that the present document does not intend to serve as an exhaustive listing of relevant principles, nor to set rules for the implementation of EOSC services. We recognize that architecture and software technologies are continuously evolving and that other, competing, factors may guide the choices of an implementation team. However, the deliverable presents a valid set of elements for a high-level design strategy to deliver new or enhance existing EOSC services.

Before jumping into the design principles, we briefly present the guiding principles of the work presented herein. The first one, with multiple benefits, is the **Openness** of the software. The concept of “openness”, itself being an overloaded term, has part of its origin in interoperability, and in that regard describes the capability of the system to exchange data with other systems. Another part of its origin is as a reference to a system’s capability to easily adapt to fulfil the challenge or case applied to it. The term expanded naturally to include the establishment of requirements for standards (requiring them to be open themselves), the ability to inspect and evolve software (leading to Open Source), and other relevant qualities, gradually making "open software" synonymous with software that can be used without substantial restrictions—although perspectives may vary on what a substantial restriction may be. Openness has a foundational role in EOSC, as presented in the respective Open Science alignment section.

Another major general principle that empowers composability and interoperability in the service-oriented era is the **Separation of Concerns**[2]. Initially coined for software (and vastly applied in object-oriented programming), it found its way into the services domain, leading to the concept of micro-services, i.e. services that address a single, relatively small set of responsibilities in a larger system. The **Single Responsibility Principle**[3], if applied as well, can further decompose software into separable entities or services. Those services, requiring signalling and data exchange with other services via typical communication channels, force software designers to define those interactions in a formal manner. Helpful paradigms, such as RESTfulness, come into play when crafting easily consumed APIs, as they impose expectations in the call structure, data and control flow among the engaged entities. The **Interface Segregation Principle**[4] is yet another that may add to the sustainability of interoperability among systems, minimising the interface where unneeded future conflicts may occur. Finally, the **Design over Performance** approach serves as an excellent framework for creating interoperable and open software. What's even more noteworthy, albeit not the primary focus of this document, is its ability to foster the development of high-quality

[2] TARR, Peri, et al. N degrees of separation: Multi-dimensional separation of concerns. In: Proceedings of the 21st international conference on Software engineering. 1999. p. 107-119.
[3] Robert C. Martin (2018). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall. ISBN 978-0-13-449416-6
[4] Martin, Robert (2002). Agile Software Development: Principles, Patterns, and Practices. Pearson Education.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

software, both in its entirety and in individual components. Performance should be tackled only after design objectives are met when performance is not the only critical factor for the service itself.

Although the effect of the principles presented here might be multi-dimensional, capturing other qualities of software and services, it must be noted that design principles for building performant, robust, scalable, Open Science-facing services are not addressed in this document, as they assume a different perspective. Complementary information in this direction may be found within the outcomes of the EOSC Association Task Force on "Infrastructure for Quality Research Software".

Finally, it has to be noted that Interoperability is strongly related to "compliance". This may be compliance to the terms of the agreement among two (interoperating) parties, or to more widely established terms (be it "standards" or other specifications' declarations). And this compliance may not be restricted to technical subjects that may impact interoperability designs. For instance, regulatory frameworks or service agreements may introduce non-technical prerequisites for the integration of systems and the exchange of data among them.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

2. Architectural choices for maximising Services Interoperability & Composability

In the context of this report, "interoperability" focuses on service interactions, i.e. signalling and data exchanges among software service instances. A service instance is a software component deployed on virtual or physical hardware, with network connectivity, and potentially enclosing its data/state, or communicating with another service to accommodate such needs, if present. In the EOSC ecosystem, the logical element that allows management of software and resources granted to it by an authoritative domain is the software service. Composing those services in larger structures, or selectively substituting them as needed, allows combination, validation, repetition, reproduction, comparison, extension and evolution of scientific processes, which together serve to address and empower a broad array of Open Science objectives. In the following paragraphs we present a non-exhaustive list of several common architectural principles that have a significant impact on interoperability and composability of software that, when adopted, help to empower software services as indicated above.

Although this document does not intend to serve as a ruleset for EOSC service implementers, nor to endorse the adoption of the herein mentioned principles by EOSC services, the authors consider that listed principles are quite relevant for the introduction of services to the EOSC ecosystem, as, probably sooner than later, a service will be found that will strongly benefit from their adoption.

The first architectural principle requires **Loose Coupling** – in which components are weakly associated with one another and thus changes in one component affect the existence or performance of at least one other component, and in which each of the architecture's components has or makes use of little to no knowledge of the definitions of the other components. Service-based architectures often aim to achieve such loose coupling.

Architecturally, there are two major service paradigms: **Service Oriented** and **Microservices** paradigms. The presence of API, a key concept to service interoperability, is orthogonal to architecture choice, however it is a common (though not mandatory) element in Microservices' architectures.

Both architectural paradigms offer the basis to address fundamental integration challenges; however, microservices' architectures often avoid being bound to a common Enterprise Service Bus (ESB) and all of the related technology stack. As such, the microservices' architecture is the architecture of choice for most cases today. Nevertheless, it is not a panacea, nor does it address *a priori* common needs and expectations of geographically distributed interoperable systems, unless specific actions are taken.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Among the interoperability and composability advantages of microservices' architectures, two are worth mentioning: the possibility to combine microservices into new arrangements with minimal assumptions regarding the technology for integration; and their use as interoperable components thanks to the fact that they tend to carry their own autonomous interfaces.

Following the API approach, the most prominent one is the REST API[5]. The **REST paradigm[6]** itself is not an API specification but an architectural approach, as it directs towards stateless interactions with services. That is, the client and the server need not communicate and establish a session and some state between them before exchanging messages that can then act on service resources. Under the REST paradigm, the API follows a formalisation that dictates that all actions (expressed via verbs) take effect on designated resources. This can lead some authors to avoid naming those interaction forms as APIs per se, so as to avoid confusion with the classic notion of service APIs.

Although REST APIs are natively supported by all major programming and runtime frameworks today, they are not a universal solution, and service implementers should balance the pros and cons in every case:

- Well-known standards pre-date the REST API paradigm and are not likely to follow it in many cases, thus the implementer should choose accordingly.
- Resource-oriented interactions do not come as natural in all cases, and may lead to complex multi-resource addressing calls. Although REST APIs can address this problem with composite resource designs, this may be overkill for very complex systems and over-design for extremely simple systems.
- Interactions such as queue messaging, common in scalable distributed systems, are not an area where REST excels, and different approaches are commonly utilised.

The predominant data types to use for exchange are JSON and XML, with binary data exchanged either encoded or as part of multipart messages. The REST paradigm is commonly considered tightly bound to JSON format because Single Page Applications (SPAs) are usually built on top of those APIs, where JSON comes as a natural choice. However, as APIs are not solely targeted against SPAs, and the compactness and modern tooling around JSON are not the only priorities in implementation, XML is also a well justified choice. This is despite the added complexity to process and persist it efficiently and the larger volume of exchanged traffic it typically generates. Reasons to support XML include its expressiveness (elements, attributes, namespaces, comments), strong foundations on supporting standards (xpath, xquery, xml schema, xsl, xslt), an arsenal of technologies to facilitate its processing, and its presence in numerous standards as baseline representation

[5] REST Principles and Architectural Constraints: https://restfulapi.net/
[6] FIELDING, Roy T.; TAYLOR, Richard N. Principled design of the modern web architecture. ACM Transactions on Internet Technology (TOIT), 2002, 2.2: 115-150.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

of their data structures. Having said that, one must note that JSON and XML are not comparable standards as they serve different needs; however, they do have a common area of applicability, which is data exchange among services.

Reuse don't Redo is a general approach to software design which has substantial impact on interoperability. Using existing software enhances the chances that existing benefits are realised, including those impacting positively interoperability, such as the support of other data types or well-known standards beyond the ones strictly targeted by a particular usage scenario. Reusable software tends also to have documentation and formal specifications, which both empower interoperability and composability. Additionally, reused software that is actively maintained offers good opportunities for future alignment to protocol changes or expansion into new ones if these can be gained for little "cost" (be it effort, risk, or direct monetary cost).

It has been previously mentioned that **compliance** is among the cornerstones of interoperability. **Compliance with technical standards** is one case, the most common in IT engineering, but it does not come without challenges. Navigating and selecting among those can be a challenge of its own for several reasons. For instance, a few domain standards may not be as "firm" as expected, offering several optional elements, or different profiles, and thereby leave it to the discretion of the designer to define the level of compliance and to account for its cost in effort. Reuse greatly helps to address this challenge. There are cases where there are no domain-specific standards either, in which case the designer has to either opt for some emerging specification or go for a custom (preferably REST) interface specification.

A common mistake in interoperability scenarios is to assume that implementing a single (open) standard or profile is adequate to address the interoperability expectations of service consumers exclusively. The principle "**One Size Does Not Fit All"**[7] is well suited for interoperability, since different consumers, managers or producers of information may require different flows or different data and interaction models. Under this principle, a service designer should carefully select the standards applicable in a domain, their flavours and alternative manifestations, and make informed implementation choices dependent on the consumers, managers or producers engaged in the designer's adoption roadmap. Great choices that usually come with minimal cost, when proper reusable software is utilised, give support to alternative data formats or alternative APIs, etc.

[7] "One Size Does Not Fit All in {DB} Systems", Andy Palmer, 25th Large Installation System Administration Conference (LISA 11), 2011.
"One size does not fit all: a grounded theory and online survey study of developer preferences for security warning types", A. Danilova et Al, ICSE '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, 2020.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Portability is not per se related to interoperability, however it strongly relates to composability, as it reduces restrictions that determine the ability of a service component to operate in an e-infrastructure. When designing portable software for EOSC, one thing that one must have in mind is not the Operating System independence, which is more or less guaranteed nowadays, with a few restrictions, but the e-infrastructure independence. Assuming that an e-infrastructure may provide a particular storage type or model, or some specific yet well-known API, may turn out to be a constraint when a component is called into a different integration scenario. Admittedly, a service needs to make baseline assumptions on its dependencies, thus the design principle here is to place those assumptions at the least restrictive level for portability and interoperability.

A choice that has to be made and respected in the long run is that of **API versioning**. There are good guidelines, embedded in standards or in paradigms (e.g. REST), that dictate how a service designer should handle API versioning so that software evolves without hindering the interoperability of older consumers. However, this does not come without a cost, as all APIs need to be maintained in code and, what is worse, when the model of a service evolves substantially, supporting an older API may not be economically feasible. Such a necessity eventually leads to changes that break backward compatibility with interoperating service consumers.

Pluggability/extensibility is a way to enhance opportunities to reuse software in an integration scenario without undo intervention. Design-stage adoption of a pluggability strategy appropriate for the technology at hand can give future adopters the ability to extend or revise how a system operates to accommodate certain integration scenarios. A holistic pluggability/extensibility approach that would allow deep adaptation of a software's capabilities and behaviour (spanning all of its layers and entities) would require considerable investment in the design stage, while targeted pluggability/extensibility, on the other hand, could address the needs of the software to adapt to very specific challenges that may arise in the future, and would require substantially less design-stage effort.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

3. Non-architectural interoperability choices

Several non-architectural choices may impact or support the interoperability of a service. In the context of EOSC, where services need to act on a set of both general and domain-specific regulatory/legal frameworks, serve the Open Science mandate and support cross-discipline collaboration in a trustworthy, dependable manner, the following options can serve as a starting point for further, case-specific consideration.

Aside from the compliance with technical standards already mentioned, **compliance with regulatory specifications** (which may themselves be *non-technical* standards) is also becoming common. The reasons for this are that (a) more and more human environments (social, enterprise, government, research, etc.) are now transferred almost completely into virtual environments, which have become the keepers of significant sensitive or confidential data, as well as enablers of crucial processes; and (b) service value chains have become a common way of doing business (in general, not specifically for profit) that needs to be regulated and validated in a number of ways. Without regulatory specifications, both data and processes are placed at risk, potentially impacting essential segments of human processes. Under this wider perspective, interoperability is not only limited by the technical ability of services to interoperate, but also by the ability of humans to agree on and comply with a common regulatory framework. Such frameworks may refer to data protection (e.g. compliance with the General Data Protection Regulation, GDPR, or enforcement of a specific data-at-rest protection or isolation policy), service-level or quality-of-service agreements (e.g. service operation at 99.9999% availability), financial terms (e.g. maximum rate of charge per CPU cycle), etc., and may come in various forms from various sources.

Regarding **API evolution**, as a service evolves, its APIs may need to drop compliance with prior specifications, which causes obvious problems for its adopters. To address this proactively, service managers should set policies for backward-compatibility maintenance, long-term support APIs, etc., that would allow a service client to follow its evolution without the constant need for maintenance. Nevertheless, any client will have to catch up after older API versions are rendered obsolete for the service, but this should happen at a slower rate. Here, the need for clear documentation of API evolution policy is required, alongside the necessary documentation of the API itself and its evolution.

Presence and quality of documentation is another facilitator of interoperability. Having up-to-date, trustworthy information, easily accessible by service integrators, greatly enhances the chance that a service will be included in an interoperability scenario, provided that it supports the features required by its producers, managers and consumers. Delivering the documentation in a feature-rich online repository and making it freely available is a must, and should be a minimal requirement within the Open Science ecosystem that this report targets. However, there are many more aspects to documentation that should be embedded in the

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

software production lifecycle. Documentation, ideally, should cover everything about the software in both human-readable and machine-actionable modes: policies, APIs, algorithms, static structure of the code, etc. Depending on the technology stack selected for software implementation and delivery, there might be more or fewer tools available to support a documentation strategy. Yet it is always a good strategy to start with "self-documented" elements. Writing code with meaningful elements (class, variable, method…), names and short-logic threads, conforming to conventions for both naming and design (e.g. applying patterns), always helps minimise and even eliminate the need for documentation of code. RESTful API conventions are a good example of a paradigm-specific strategy that minimises the documentation burden. Manually writing documentation for coding artefacts is not commonly favoured by developers since, apart from the effort it requires, such documentation easily drifts out of sync with actual code logic. It is easier, however, to document and maintain synchronisation with types and APIs, especially if conventions are respected, as the tools to support this are quite mature across platforms. Tools that reverse-engineer code to produce documentation can support the documentation process to a certain degree, while static code analysis can provide insight to how documentation and code structure strategies are carried out by an implementation team. A Test-Driven Development strategy is also an indirect method to document software, widely appreciated (but in some cases opposed) by implementation teams.

Licensing and access to software code itself can be a barrier (or a facilitator) of interoperability. Imposing restrictive access rights on the use of software, on the visibility and/or reuse of its source code or on the adoption of an interface/specification, can substantially limit the willingness of a consumer to utilise the software or the resulting service in their integration scenarios.

However, not all software may be licensed in a fully open/free manner for several reasons, including cascading intellectual property rights (IPR), a financial sustainability strategy or due to the objective costs incurred by its usage. Even when licensing restrictions apply, software acting in the Open Science ecosystem should be transparent about its internal workings, especially when it is responsible for translating scientific processes.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

4. Aligning Interoperability to EOSC priorities

Interoperability is an open field, and the approach taken is ultimately defined by the scope of the system. However, in the context of EOSC, there are a few priorities that technical service interoperability should address: Open Science, computing e-Infrastructures, and the user (including either the producer, manager or the consumer). Additionally, a service should address regulatory compliance with inscriptions of the respective EU framework, which will not be extensively covered in this document. In the following paragraphs we make short references to key areas that a system designer should consider for interoperability when delivering EOSC-enabled services.

Alignment with Open Science

The essence of EOSC is to empower Open Science practices and processes across the pantheon of modern e-Infrastructures, and thus it is fair to say that Open Science is more than just a priority for EOSC. EOSC poses a number of requirements on services and data, many related to interoperability. Interoperability is in general referring to complex and heterogeneous systems, yet in this document we mostly limit the scope to software services. This report has already touched on design choices that improve services' interoperability and enhance their suitability to empower Open Science processes; the focus of this section shifts to FAIR data's interoperability impact on design choices, as data FAIRness is one of the pillars of the practice of Open Science.

Provisioning for data to be exchanged in service interactions is an element of interoperability analysis, however it is common that interoperability of data is considered independently. The FAIR principle serves as more than mere data interoperability guidelines, and in one way or another, all four elements of FAIRness lead down to interoperability directives, despite only the "I" explicitly naming it as such. A non-exhaustive list of those areas where interoperability comes into play are provided below:

- Persistent identifiers (PIDs) forms, allocation, assignment and resolution
- Metadata schemata and their semantics
- Controlled vocabularies and vocabulary services
- Data formats
- Metadata registration
- Metadata consumption
- Machine actionable metadata
- Data registration
- Data consumption (domain or data-type specific protocols)
- Data discovery (search, harvesting, retrieval)
- Reproducibility and traceability (including provenance)

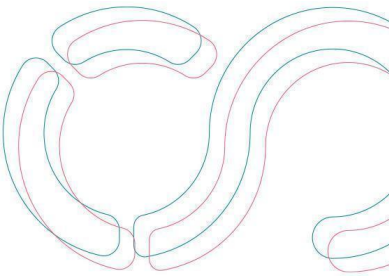
EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

- Linkages to the real world (e.g. to be able to reproduce experiments in labs)
- Licencing ("as open as possible, as closed as necessary") and control ("decentralised as possible, centralised as necessary")

Services acting in EOSC and dealing with data/metadata consumption/provisioning need to meet one or more of these challenges, and this should be accomplished through interoperability with existing systems. Here we must clarify that "data" in the area of Open Science can be very heterogeneous in nature: Sensing, observation, model and simulation data, to name a few, are commonly considered research data, while artefacts such as research publications are commonly treated in a similar way. Yet, educational resources, documents, and even software itself, can be considered data according to the FAIRness principles. Each of these require quite different mechanisms to capture their description, publication/announcement, discovery, reference, consumption, etc. Standards in the field are numerous and address several different aspects of the FAIR principles (SRU, OAI, OAI-PMH, Handle, DOI, DC, LOM, several ISO standards and RFCs, etc.). Adoption of one over another depends on the field and services at hand, especially when referring to domain-specific data and metadata formats, service endpoints, etc. However, there are some practices that greatly enhance interoperability or inclusion of a service in more complex flows, including, for example:

- Offering the means to locate and retrieve data directly from controlled catalogues and certified repositories (or gateways), and to consume them for any type of processing offered by the service. Using PID resolution or other methods for locating required data is a good practice, too.
- Offering the ability to directly deposit the products of a process onto appropriate catalogues and data hosting entities, taking care of the generation of respective metadata and issuing of PIDs. This approach can handle the automatic generation of all provenance information assuming it has a good knowledge of the inputs supplied and the process performed to deliver a data product.
- Offering metadata in simplified or alternative formats, which enhances the ability of repositories to list data products offered by a service. Such an example could be to offer DC metadata representation beside the richer domain-specific WCS for Earth Observation datasets, allowing them to be listed into global catalogues. Similarly, for instance, a service offering metadata in EDM alignment, over a more focused, elaborate or extended schema (e.g. LIDO, CARARE, CIDOC-CRM, etc.) would allow them to be listed in cultural heritage digital assets aggregators.
- Delivering data via simple protocols (e.g. https) and in simplified forms (e.g. a standard binary file). This makes it easier for service users to utilise the service's products in additional forms.

If the service is itself a repository, complying with the aforementioned standards for identification, depositing, retrieval and aggregation, as well as with the domain-specific



EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

standards, is essential for the repository's alignment and interoperability with other EOSC services.

Managing data licensing presents a significant challenge. To tackle this problem, employing machine-readable licenses that can be readily utilized and acted upon through clearly defined API services is crucial. However, incorporating and applying these provisions within automated workflows for generating derivatives, which involve multiple components beyond the original data licensing, such as funder rules, algorithms and software restrictions, supervisory entity regulations, and more, remains a complex and open-ended subject.

Licensing of the software itself, and the resulting services, is essential to empower interoperability from a design perspective, intensified as it is within the Open Science environment, since licensing is often a deciding factor for integration scenarios where interoperability comes into play.

Finally, one of the most crucial aspects of aligning with Open Science is the **openness** of software. This not only emphasises the software's interoperability capabilities but also ensures transparency in its functioning. Such transparency plays a vital role in integrating the software into dynamic, well-documented, and reproducible processes.

Alignment with Computing e-Infrastructures

Another pillar of EOSC is cloud-based infrastructure, and, for this, interoperability is less formalised through standards. Service integration, with a few exceptions, is mostly based on open specifications followed by several engaged systems with respect to network, storage and compute resource allocation for carrying out projects or processing tasks.

The ability of services to consume resources on the e-infrastructures, i.e., in a sense effectively interoperating with them, is essential to meet the demand for computational and storage requirements that may change over time. It is more of an architectural choice for service implementers to pick mechanisms, or even execution models, that do not hinder their portability capabilities, but rather allow them to interoperate with the platforms offered by Infrastructure Providers. Employing queues with workers (or other distributed processing models), workflow management systems or operating directly on a computational platform are a few examples of such choices.

Using a container engine (e.g. Docker, Kubernetes) and numerous supporting elements around it for the deployment and management of the services, networking, storage as well as response to load, has become the common practice for virtualizing resources and streamlining their utilisation. This elevates the relevant specifications to prominence, nicely serving the need for portability across e-infrastructures, even if they are not yet globally adopted.

Attempting to formalise this aspect, we can identify a number of standards/specifications:

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

- Containers images definition, such as Open Container Initiative (OCI) Image Specification and Docker Image Manifest (the two being very similar);
- expression of topologies or dependencies among components, such as TOSCA, pod definition, etc.

And technologies for:

- Container execution engines, e.g. containerised, Kubernetes, Docker and Singularity; Singularity, for example, suited bare-metal HPC clusters, which play a significant role in the EOSC ecosystem;
- technologies for microservices/applications deployment/management on containers or cloud infrastructure, such as Helm for Kubernetes, Ansible, etc.

Depending on the e-infrastructures at hand, a number of higher-level platforms to utilise resources may be in place, offered under the Platform as a Service model. Those may be workload management systems, such as Apache Hadoop and Spark; workflow management systems, such as Apache Airflow, Dagster, etc.; or lower-level distributed processing systems, such as Condor HPC. Those platforms raise their own interoperability requirements and present additional opportunities. An example is the ability of SparkSQL to interact with numerous types of data sources, offered almost for “free” to the users of the platform.

User alignment

Among the EOSC priorities that need to be considered from the interoperability perspective is that of “the user” (being either the information producer, manager or consumer of data and services). Interoperability, from the end-user perspective, boils down to allowing one to interact with services using the tools available on a given platform without missing out on the features of the service offering. This may be broken down into several elements that are nothing more than services and software interoperability:

- Interoperable subsystems to authenticate and authorise the users: independent of the client access platform (addressed by SSO standards), reaching, if possible/applicable to the level of a resource.
- Presentation layers capable of operating on any device: as addressed by common modern web UIs conformant with relevant W3C specifications.
- Integration at the client UI level: addressed by services (commonly standards-compliant ones) that deliver web accessible content and/or offer reusable web components for interaction with the content that can be included in client environments, such as dashboards.
- Delivery of data/content/metadata under protocols easy to consume by user applications applicable to a domain of interest: addressed by common data types.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

- Accessibility of content: addressing of accessibility concerns meeting the specifications for accessibility, for example complying with the ones offered by the W3C Accessibility Guidelines[8].

[8] W3C Accessibility Guidelines: https://www.w3.org/TR/wcag-3.0/

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

5. The Role of Standards

"Standards" are specifications that have acquired some sort of formality. They are generally separated into two classes:

- **Formal standards**, created through standardisation processes and organisations (Standards Development Organisations (SDOs)), which are formally maintained, independent of the degree of their adoption. It is not uncommon for formal standards to be accompanied by reference implementations supported directly or indirectly by the SDOs behind the standard.
- **De-facto standards** that have emerged from established conventions and practices, e.g. the adoption of APIs with (usually) proper specifications by large audiences. Such standards are proven to work, and stay around for a long time until the technology and needs silently surpass them. It is not uncommon that a formal standard emerges around a de-facto standard.

Those developing and deploying EOSC have already done a lot of work to strengthen and promote de-facto standards into formal ones. For instance, the Research Data Alliance (RDA) has been building synergies with SDOs and similar organisations so that RDA recommendations can either get a fast track to becoming standards, or can be incorporated into or contribute to standards, common goals and activities. Standards play a foundational role in the interoperability of services and must be taken into consideration at an early stage in the development of software and services, as their adoption may impact several design decisions. For instance, standards may be implicitly dictating technologies to be reused or software states to be maintained, affecting to varying degrees the technological stack employed to deliver a specific outcome. Furthermore, standards may not refer to specifications of interactions among software only, but also to the processes and operations around them, that may also have an impact on the ability of software services to collaborate. For instance, services not conforming to certain security standards may not be appropriate for interoperating in a given scenario, even though, technically, they are able to adequately interact.

Apart from standards, non-standard specifications have a significant role in integration scenarios. Among these there are:

- Open specifications: specifications that are made available without licensing restrictions as part of open source software and that usually conform to some commonly acceptable paradigm (e.g. REST API).
- Proprietary specifications: owned by particular entities and ruled by those entities, commonly exposed to the consumers of specific platforms.

Services should primarily implement formal and de facto standards, and opt for non-standard specifications, preferably open ones, if no applicable standards are available. Compliance

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

with standards may be achieved either by reuse or by implementation, with reuse being the preferred choice, especially when achieved via open, widely adopted solutions.

In **Annex B**, a short list of indicative standardisation bodies and standards families is presented. The annex does not intend to extensively cover all domains addressed by EOSC or software technologies, and should be seen as an example of how a potential lookup for standards could be shaped for all domains.

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

6. Regulatory Compliance

In the EOSC landscape it is essential that regulatory compliance is guaranteed under interoperability scenarios. A few examples of such compliance imperatives are mentioned below:

- GDPR[9] compliance requires that services engaged in a flow of data exchanges and processing are aligned with the directives of the European Law regarding Data Protection. This requires both legal statements as well as technological precautions for the handling of data.
- EU Data Act[10] additionally defines the rules to facilitate data sharing by companies, individuals and the public sector, and clarifies who can create value from data and under which conditions. The Data Act removes barriers to access data, for both the private and the public sector, while preserving incentives to invest in data generation by ensuring a balanced control over the data for its creators.
- EU Directive 2019/1024 on open data and reuse of public sector information: research data was explicitly put under the scope of this regulation. EOSC and its interoperability efforts will be critical for the implementation of this regulation. Publicly funded research data shall be re-usable both for commercial and non-commercial purposes equally.
- Utilisation of EOSC resources does not come openly for all uses. Academic resources are subject to specific regulatory and ethical restrictions that need to be respected by consumers. For instance, the generation of commercial profits by such e-infrastructures may be against the regulations of resource provisioning, which needs to be examined on a case-by-case basis as some usage scenarios are much more complex than others.
- Service-Level Agreements (SLAs) shape the requirements for utilising services, as depending on misbehaving services may hinder the quality of service delivery. EOSC requirements for TRL as well as provisions for service monitoring and support are two examples where SLAs can be deal-breakers in this context.

[9] Regulation (EU) 2016/679: https://eur-lex.europa.eu/eli/reg/2016/679
[10] The European Data Act: https://www.eu-data-act.com/

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

7. Conclusions

In summary, the following salient points are presented:

- Selecting a strategy for interoperability is essential for planning the breadth and depth of a service or system. The priorities of a system, driven by its domain and functional requirements and resource constraints, separate what must be done and what can be done, focusing on those practices that improve the chances for long-lasting interoperability. Reusing software, building on standards, following an extensibility approach and, in general, applying best-practices are always helpful in this respect.
- Interoperability needs to be considered at the design stage of a software system and not left entirely for the later stages in the system's lifecycle, as interoperability might become entangled with fundamental flows of a system, or it might require provisioning of substantial complexity (e.g. extensibility/pluggability mechanisms).
- The domain and the environment in which a system operates commonly raise special interoperability requirements. Domain- or challenge-specific standards families offered by established bodies should be prioritised where interactions with other systems are expected to take place. De facto standards should be considered depending on the domain and the presence of overlapping active standards. EOSC services are often found to share approaches for domain-specific interoperability, which is a good starting point for making choices around standards.
- Substantial challenges for interoperability may be introduced by applicable regulatory frameworks and SLAs that a service may need to comply or co-exist with. Those may lead to specific functional, architectural and technological choices that are best considered as early as possible in the design of software services.
- Interoperability of services in the EOSC ecosystem should serve EOSC's primary vision, which is to empower Open Science and the priorities it advocates (data FAIRness, research reproducibility and repeatability, etc.). On the other hand, interoperability needs to support the EOSC foundations, including the e-infrastructures that are going to form its network, and its end-users in the various roles one might have in this ecosystem (manager, researcher, publisher, consumer, etc.). A starting point for all services in the ecosystem is their interoperability with the EOSC Core services. More details on these aspects will be provided in the forthcoming deliverable of the Task Force entitled, "A landscape overview of the EOSC Interoperability Framework: Capabilities and Gaps".

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Acronyms and Abbreviations

Acronym	Name
API	Application Programming Interface
BDVA	Big Data Value Association
CIDOC-CRM	CIDOC - Conceptual Reference Model (CIDOC = International Committee for Documentation)
DC	Dublin Core
DOI	Digital Object Identifier
EC	European Commission
EDM	Europeana Data Model
EOSC	European Open Science Cloud
ESB	Enterprise Service Bus
ETSI	European Telecommunications Standards Institute
EU	European Union
GDPR	General Data Protection Regulation
HPC	High Performance Computing
IETF	Internet Engineering Task Force
ISO	International Standardisation Organisation
IT	Information Technology
IVOA	International Virtual Observatory Alliance
JSON	Javascript Object Notation
LIDO	Lightweight Information Describing Objects
LOM	Learning Object Metadata
OAI	Open Archives Initiative
OCI	Open Container Initiative
OGC	Open Geospatial Consortium
PID	Persistent Identifier
PMH	Protocol for Metadata Harvesting
RDA	Research Data Alliance
REST	Representational State Transfer
SDO	Standards Development Organisation
SLA	Service-Level Agreement
SPA	Single Page Application
SRU	Search Retrieve by URL
SSO	Single Sign On

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Acronym	Name
URL	Universal Resource Locator
W3C	World Wide Web Consortium
WCS	Web Coverage Service
XML	eXtensible Markup Language

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Annexes

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Annex A: Summary of Principles, Guidelines and Recommendations

The following table summarises the principles, guidelines and recommendations presented in this document. The first column is a numeric identifier, and sub-numbering indicates specific subcategories of the same aspect. The second and third column specify the aspect and a brief description, respectively. The last column indicates the typology, i.e. whether the aspect refers to a principle or a criterium, derived from the considerations of the TF members and presented in this document; and a rating (MAY, SHOULD or MUST) to indicate the urgency required for its adoption. In future versions of this document the table content will be further verified by community consensus or other external experts.

#	Aspect	Description	Typology
1	Openness	Open software is more interoperable	Principle, SHOULD
1.1	Open Standards	Open software uses open standards whenever possible	Criterium, SHOULD
1.2	Open Source	Ability to inspect and evolve the software	Criterium, SHOULD
1.3	Open Licence	Open software is available under one or more open licences	Criterium, MUST
2	Single Responsibility	Separation of Concerns – services that address a single relatively small set of responsibilities in a larger system	Principle, SHOULD
2.1	API Design	Easily consumed and well-documented APIs, based on standards such as REST and SmartAPI	Criterium, SHOULD
2.2	Formalisation	Define interactions between components in a formal manner	Criterium, SHOULD
3	Interface Segregation	Minimise opportunity for future conflict between components	Principle, SHOULD

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

#	Aspect	Description	Typology
4	Design over Performance	Performance should be tackled at a second level after design objectives are met, when performance is not the only critical factor for the service itself	Principle, SHOULD
5	Loosely Coupled Systems	Components can change without affecting others, and need not share state or internal data	Principle, SHOULD
5.1	REST APIs	Enables stateless interactions with services	Criterium, SHOULD
5.2	Exchange	Uses XML and JSON as a basis of exchange, with binary data encoded or in multipart messages	Criterium, SHOULD
6	Reuse don't Redo	• Reusing existing software enhances the chances that existing benefits are realised, some of those impacting positively interoperability • Reusable software tends also to have documentation and formal specifications, which both empower interoperability and composability	Principle, SHOULD
7	Compliance	Compliance with technical standards is a powerful case, the most common in IT engineering	Principle, SHOULD
8	One Size Does Not Fit All	Minimise proliferation and divergence of standards, but do not limit usability in doing so	Principle, SHOULD
9	Portability	Relates to composability, as it reduces restrictions that relate to the ability of a service component to operate in an infrastructure	Principle, SHOULD
9.1	Operating	Implemented via technologies such as	Criterium, SHOULD

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

#	Aspect	Description	Typology
	System Independence	virtualisation and containerisation	
9.2	Dependencies	Assign dependencies to the least restricting level or layer	Criterium, SHOULD
9.3	Infrastructure Independence	Modularisation, separation of concerns, and service orientation	Criterium, SHOULD
10	API Versioning	Dictate how a service designer should handle API versioning so that software evolves without hindering the interoperability of older consumers	Principle, SHOULD
10.1	REST API Versioning	Guidelines for version support in REST APIs	Criterium, SHOULD
11	Pluggability/ extensibility	A way to enhance the opportunities of a software being reused in an integration scenario without intervening with its implementation	Principle, SHOULD
11.1	Design pluggability	Design-stage adoption of one of the pluggability strategies appropriate for the technology at hand improves future extensibility	Criterium, SHOULD
12	Regulatory Compliance	Compliance with regulatory specifications (which themselves may be standards) is also becoming common	Principle, MUST
13	API evolution	Maintaining backward compatibility should be managed via a clear policy	Principle, SHOULD
14	Presence and quality of documentation	Having up-to-date, trustworthy information, easily accessible by service integrators, greatly enhances the chance that a service will be included in an	Principle, MUST

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

#	Aspect	Description	Typology
		interoperability scenario	
14.1	Documentation is available	Documentation is available and easily accessible	Criterium, SHOULD
14.2	Minimum documentation quality	Documentation meets minimum standards (e.g. OpenAPI/ SmartAPI) for API services	Criterium, SHOULD
15	As Open As Possible	Licencing and code access is a barrier or facilitator for interoperability	Principle, SHOULD
15.1	Use Well-Known Open Licences	Use one of a suite of well-known, widely adopted open software licences whenever possible that are machine-readable	Criterium, SHOULD
16	Specific Workflow Best Practices	Implement the best practices associated with service elements and workflow steps in the main research infrastructure use cases	Principle, SHOULD
16.1	Harvestable Catalogues and Registries	Offering the means to locate and retrieve data directly from catalogues and repositories (or gateways)	Criterium, SHOULD
16.2	Automated deposit	Offering the ability to directly deposit the products of a process on to appropriate catalogues and data hosting entities, taking care of generation of respective metadata and issuing the PIDs	Criterium, SHOULD
16.3	Simplified Metadata	Offering metadata in simplified formats	Criterium, SHOULD
16.4	Simple Data Protocols	Delivering data via simple protocols (e.g. https) and in simplified forms (e.g. a standard binary file)	Criterium, SHOULD

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

#	Aspect	Description	Typology
16.5	Domain-specific requirements	If the service is itself a repository, complying with the aforementioned standards for identification, depositing, retrieval and aggregation, along with domain specific ones, is essential for EOSC alignment	Criterium, SHOULD
17	Cloud-Based Orientation	Implement those standards and technologies that support widespread interoperability in the cloud	Principle, SHOULD
17.1	Containerisation	Use standard approaches such as Docker	Criterium, SHOULD
18	Platform as a Service Model	Use this model, with appropriate tools for workflow, workload, and utilisation management	Principle, SHOULD
19	User Focus	Implement approaches and standards that assist users	Principle, SHOULD
19.1	Authentication	• Interoperable methods to authenticate and authorise the users • Support single sign-on (SSO)	Criterium, SHOULD
19.2	Presentation	Presentation layers that may operate on any device: addressed by common modern web UIs conformant with relevant W3C specifications	Criterium, SHOULD
19.3	Interoperability	Delivery of data/content/metadata under protocols easy to consume by user applications applicable to a domain of interest: addressed by common data types	Criterium, SHOULD
19.4	Accessibility	Meeting the specifications for accessibility, for example those offered by W3C Accessibility Guidelines	Criterium, SHOULD

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

#	Aspect	Description	Typology
20	Existing standards	Apply existing standards whenever possible	Principle, SHOULD
20.1	Formal Standards	Use by preference: standards created by standardisation processes and organisations, which are formally maintained, independently of the degree of their adoption	Criterium, SHOULD
20.2	De Facto Standards	Use by preference: standards that emerged from e.g. adoption of APIs with (usually) proper specifications by large audiences	Criterium, SHOULD
20.3	Standards Formalisation	EOSC should promote the formalisation of de facto standards	Criterium, SHOULD
20.4	Open Specifications	Use if no standards are applicable: made available as part of open source software without licensing restrictions and usually conform to some generic specification (e.g. REST API)	Criterium, MAY
20.5	Proprietary Specifications	Use as a last resort: owned by particular entities and ruled by those entities, commonly exposed to the consumers of specific platforms	Criterium, MAY
21	Regulatory Compliance	EOSC services must comply with regulations and legal obligations in the EU	Principle, MUST
21.1	GDPR compliance	Requires that services engaged in a flow of data exchanges and processing are aligned to the directives of the European Law regarding Data Protection	Criterium, MUST
21.2	Fair Use	Academic resources are subject to specific regulatory and ethical	Criterium, MAY

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

#	Aspect	Description	Typology
		restrictions that need to be respected by their consumers	
21.3	Academic Use Only	Some resources are restricted to academic or research applications	Criterium, MUST
21.4	Service-Level Agreement	Service-Level Agreements shape the requirements for utilising services, as depending on misbehaving services may hinder the quality of delivery	Criterium, MUST
21.5	Maturity	Services are offered at TRL 8/9	Criterium, MUST

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Annex B: Indicative Standards

This appendix presents an indicative list of standards met in the EOSC ecosystem. By no means is the list exhaustive or complete. It will be updated following a best-effort principle to become increasingly inclusive, aiming to eventually become a comprehensive guide for the reader. The rest of it is structured as follows:

- Standards families: Lists a logical grouping of standards for further usage.
- Standardisation bodies: Lists entities acting on standards responsibilities.
- Standards catalogue: A proposal for listing prominent standards extensively used by actors in the EOSC ecosystem. This section is intentionally empty in the first version of this deliverable.

Standards families

Interoperability standards often exist within families that cater to the specific requirements of a particular technological domain or scientific interactions. Significant standards families commonly met in the EOSC landscape have to do with:

- Cross-Cutting Standards
  - Security (authentication / authorisation)
  - Usability and accessibility
  - Internet and web standards (linked open data, etc.)
  - Digital content and multimedia representation (MIME Types, type registries, etc.)
  - Others
- Standards Applicable to Research Data Management Systems and Services
  - Data and metadata exchange (OAI-PMH, D-CAT, etc.)
  - Workflow and notifications (Linked notification, etc.)
  - Deposit, curation, preservation, and dissemination (includes licences and access management, format preservation standards, etc.)
  - Support services (tool and type registries, etc.)
  - Others
- Application and domain-specific standards
  - Data, service, and metadata standards: domain-specific
  - Data, service, and metadata standards: format- and type-specific
  - Others
- Others

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Standardisation bodies

One or more entities, standardisation promoting groups or consortia, may be acting on a standards family or on a combination of standards to address a specific cause. Formal standardisation organisations, such as ETSI, ANSI and ISO, are legal entities established at national or international level that handle standards through stricter and more formal methods, often building on or being supported by work of standardisation consortia or groups. Those entities typically cover wide spectra of standards of very diverse domains.

Besides standardisation groups/consortia and standards organisations that handle the establishment of formal specifications of standards, other entities that cherry-pick standards and specifications and issue interoperability and compliance guidelines exist, such as OGC, IOVA, GA4GH, BDVA, etc. Among those are EOSC-A and RDA. All of these play a significant role in the adoption of a set of standards or the emergence of new de-facto standards. Under the umbrella of those groups, the development of new specifications is rather quick and allows for a rapid cycle of feedback and revisioning/versioning.

The following table summarises a few standardisation bodies, whose work directly or indirectly intersects with EOSC strategy and developments.

Acronym	Description	Domain
ANSI	The American National Standards Institute (ANSI) is a private, non-profit organization that administers and coordinates the U.S. voluntary standards and conformity assessment system. Founded in 1918, the Institute works in close collaboration with stakeholders from industry and government to identify and develop standards- and conformance-based solutions to national and global priorities.	General
ETSI	ETSI is a European Standards Organisation (ESO), a non-profit association set up in 1988 by the European Conference of Postal and Telecommunications Administrations (CEPT) in response to proposals from the European Commission.	General
GA4GH	The Global Alliance for Genomics and Health (GA4GH) is an international, non-profit alliance formed in 2013 to accelerate the potential of research and medicine to advance human health. Bringing together leading organizations working in	Health, Genomics

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Acronym	Description	Domain
	healthcare, research, patient advocacy, life science, and information technology, the GA4GH community is working together to create frameworks and standards to enable the responsible, voluntary, and secure sharing of genomic and health-related data.	
IETF	The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of Internet architecture and the smooth operation of the Internet. It was initially supported by the federal government of the United States but since 1993 has operated under the auspices of the Internet Society, an international non-profit organization.	Internet
IVOA	The International Virtual Observatory Alliance (IVOA) was formed in June 2002 with a mission to facilitate the international coordination and collaboration necessary for the development and deployment of the tools, systems, and organisational structures required to enable the global utilisation of astronomical archives as an integrated and interoperating virtual observatory.	Astrophysics, Astronomy
ISO	The International Organization for Standardization is the successor to the International Federation of the National Standardizing Associations (ISA), which operated from 1928 to 1942, publishing its first standard in 1951 and currently (as of 2021) numbering more than 24.000 standards.	General
OGC	Originated in 1994 the Open Geospatial Consortium (OGC) is an international consortium of more than 500 businesses, government agencies, research organizations, and universities driven to make geospatial (location) information and services FAIR - Findable, Accessible, Interoperable, and Reusable.	Earth Observation, Planetary body Observation
RDA	The Research Data Alliance (RDA) was launched as a community-driven initiative in 2013 by the European Commission, the United States	Research Data

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

Acronym	Description	Domain
	Government's National Science Foundation and National Institute of Standards and Technology, and the Australian Government's Department of Innovation with the goal of building the social and technical infrastructure to enable open sharing and re-use of data.	
W3C	World Wide Web Consortium (W3C) mission is to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web. It was founded in 1994 and led by Tim Berners-Lee and as of 21 March 2022, W3C had 459 members.	Internet

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931