ESGF Future Architecture Report

Version: 1.1

Philip Kershaw[1], Ghaleb Abdulla[2], Sasha Ames[2], Ben Evans[3]

1 Centre for Environmental Data Analysis, RAL Space, STFC, UK
2 Center for Applied Scientific Computing, LLNL/DOE, USA
3 NCI, Australian National University, Acton, Australia

Version	Date	Author(s)	Changes
0.1	3 March 2020	Philip Kershaw (STFC)	Initial version for review
0.2	11 March 2020	Philip Kershaw, Ben Evans (NCI), Ghaleb Abdulla (LLNL)	Incorporation of review input
0.3	25 March 2020	Philip Kershaw (STFC)	Changes ready for sharing for ESGF meeting and IS-ENES3 General Assembly
0.4	3 June 2020	Philip Kershaw (STFC)	Added Executive Summary and reformatted ready for official release as ESGF Report and for incorporation with M5.1 milestone for IS-ENES3 project.
1.0	24 June 2020	Philip Kershaw (STFC)	Final revisions for release with M5.1 milestone for IS-ENES3
1.1	1 July 2020	Philip Kershaw (STFC)	Added explicit acknowledgements section. Release ready for Zenodo upload

1 Executive Summary

This document reports the findings of the Earth System Grid Federation Future Architecture Face-to-Face Meeting held in the UK 5-7 November 2019, attended by technical representatives from the ESGF partner organisations. ESGF is an International federated archive that provides access to Earth science data. It is fundamentally a distributed file management system with its main function being data dissemination through the provision of services for efficient replication and search. It has been in operation for over ten years and has undergone continuous improvements and enhancements over this time span.

The Future Architecture is a major review of the software system including a fundamental reassessment of its requirements with a view to develop a new overall system architecture. This has been initiated in the light of challenges around the maintenance of the existing software legacy and a changing technology landscape and practices in the communities it serves. The work is being carried out under the oversight of the ESGF Executive Committee[1,2] and is coordinated activity between ESGF and the European IS-ENES3 project.

The findings of the report can be summarised:

- Adopt a modular container-based application architecture as the single supported deployment system for ESGF
- Where possible, adopt community standards and solutions in order to improve the ability for ESGF to integrate with other similar systems
- Take advantage of Public (commercial cloud) technology to centralise and simplify the operation of services. These include search and identity management
- Improve the search service by better linking and integration with other services in ESGF such as ES-Doc[3], PID[4] and Errata[5]
- Completely replace the data publishing and user web frontend services

A roadmap has been agreed in order to prioritise the goals taking into account the available resources from the participating organisations. The short-term goals are to redevelop the deployment system using Docker[6] containers; replace the legacy identity management system and develop new software for data publishing and the user web frontend. An initial milestone of June 2020 has been set for the first release of a container-based release.

---

[1] https://esgf.llnl.gov/governance.html
[2] https://esgf.llnl.gov/committee.html
[3] https://es-doc.org/
[4] https://esgf-data.dkrz.de/projects/esgf-dkrz/pid
[5] https://es-doc.github.io/esdoc-errata-client/
[6] https://www.docker.com

2 Acknowledgements

The authors wish to acknowledge the support of ESGF Executive Committee in the organisation of Future Architecture meeting and the co-ordination and planning activities in the run up to the event. In addition, we want to express our thanks to the technical representatives from the participating institutions for their individual contributions at the meeting and in the subsequent work to bring together this document. Finally, special thanks should go to the IS-ENES3 project management team for their support and agreement to fund this event. Hosting a face-to-face meeting has been critical in enabling the collaboration to assess the current ESGF system together and establish a consensus to shape proposals for its future development.

The IS-ENES3 project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084.

3 Contents

4 Introduction

The review of the architecture was first initiated in September 2018. A first document was prepared to gather input from members of the ESGF community involved with the technical operation and development of the system. This was presented at the Face-to-Face meeting in Washington DC, December 2018. Following from this, the Executive Committee put plans in place for a dedicated meeting in 2019 to take forward proposals for a new architecture. This was held at Milton Hill House, Steventon in the UK in November 2019.

The meeting was organised along these guiding principles: keep to a small technically focused meeting (around twenty invited representatives from the community participating); flexible agenda with no planned presentations in order to maximise time to discuss issues and make decisions; hold a series of pre-meeting telcos in advance to prepare the groundwork. These centred on four high-level topics:

1) User experiences
2) Data repository and management
3) Compute on data
4) Platforms and system administration

This document along the lines of the topics discussed in the meeting starting with a high-level look at the objectives and requirements, a review of the landscape in terms of other comparable initiatives and wider technology developments relevant to the future evolution of ESGF. The four topics are also looked at in turn analysing the current system, requirements for a future system and proposals to meet those requirements.

5 High-level objectives for the infrastructure

1) To provide access to climate data primarily *MIP (Model Intercomparison Project) activities. ESGF may support access to other Earth science data but the core community to support is *MIP.
2) Address the challenge of hosting and access to large volume climate datasets through the provision of a distributed system of collaborating data provider organisations linked together in a federation.

6 High-Level System Requirements and Aspirations

6.1 Actors, stakeholders

We define the following actors and stakeholders in the infrastructure:

- Data providers
    a. Modelling centres
    b. Satellite-based data services
    c. Data workflows
    d. Sensor/observational data
    e. Other data services
- Analysis users
    a. Big data users: require a large computing facility;
    b. Medium data users: will benefit from a compute node on a modest cluster that they can access;
    c. Small data users: can do the work on their laptop
    d. Software application developers
- Infrastructure/data service providers and operators
    a. Sysadmin / network admin / infrastructure
    b. ESGF stack administrators
- Publishers - publish data into ESGF

6.2 Requirements

1. Provide access to large volume *MIP data
2. Federate access to data across participating data providers in a federation
3. Data providers (see definition in following section) retain control of data (not obligated to publish to whole federation)
4. Infrastructure/data service providers and operators must maintain an agreed level of operational service in order to integrate with the federation
5. Prioritise the provision of a very robust core of functionality for the infrastructure which satisfies a basic set of functions and services well
6. Provide a stable API to facilitate developers and third parties integrating their applications and services with the ESGF infrastructure.

7. Provide a good design which will inspire others to adopt it
8. Adopt community standards e.g. cataloguing, metadata to enable easy integration with third party applications, systems and infrastructure
9. Provide a baseline reference web-facing user interface to facilitate access to data for the user community
10. Support Open access and FAIR Data principles[7]

We recognise the need to gather more requirements downstream from science community (e.g. IPCC): are we focusing on the right areas (e.g., data analytics, compute)? Can we achieve those requirements?

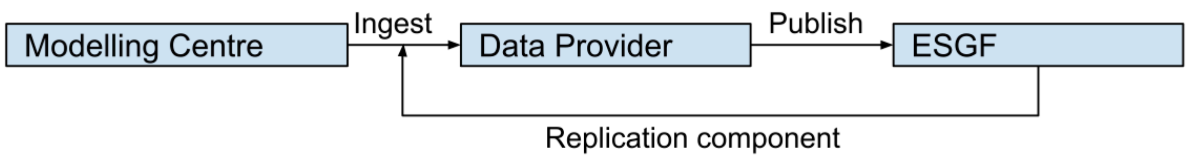

Figure 1: publishing model for ESGF. When data has been published at one site it can be replicated to other sites.

7 Review Landscape

In this section we provide a brief overview of the existing architecture and its unique aspects in the context of other similar initiatives in the Earth sciences and examples from other research domains.

7.1 Existing Architecture and Unique Aspects of ESGF

ESGF has successfully operated a globally distributed data infrastructure over ten years. A key strength has been the implementation of an effective system for federated search and replication of datasets amongst participating sites. The use of controlled vocabularies - the Data Reference Syntax (DRS) and application of strict standards for data ingestion particularly for CMIP5 (5th Coupled Model Intercomparison Project) - has assisted with indexing of data for search and the standardisation of services. It supports multiple entry points into the federation and as its use has grown, governance of data and operations have become more challenging.

The existing architecture is shown in Figure 2. The system is broken down into high-level components called *nodes* any one of which may be deployed at a given participating site in the federation. The *Data Node* hosts services for data publication and access. Additionally, access control filters enable access to data to be restricted based on authorisation policies. The *Index Node* hosts services to support data discovery and stores metadata content indexed from source data files. Federation is expressed in two core aspects of the overall functionality:

[7] https://doi.org/10.1038/sdata.2016.18

search and access control - IdEA (Identity, Entitlement and Access management) as it is referred to in ESGF:

- Search content in Index nodes can be linked such that users may search any one Index node and potentially discover data hosted at multiple data nodes across the federation
- IdEA: the federation supports single sign-on and authorisation is federated through a system of virtual organisation (VO)-level attribute services. A user can register for a VO-level attribute which represents an access entitlement honoured and enforced across all nodes in the federation. For, example if a user is registered for “CMIP5 Research” attribute they are granted access rights to CMIP5 data hosted at any Data node in the federation. Authorisation services use SAML[8] for their interfaces. Individual sites in the federation may host their own Identity Provider (IdP) to provide single sign-on for their users. OpenID 2.0[9] and MyProxyCA[10] were adopted as the technologies for web and command line client-based authentication respectively.

The *Compute Node* forms the third node type. This was first implemented using Live Access Server (LAS)[11]. Development has subsequently focused on implementations based on the OGC Web Processing Service standard[12].

[8] https://www.oasis-open.org/standards#samlv2.0
[9] https://openid.net/specs/openid-authentication-2_0.html
[10] http://grid.ncsa.illinois.edu/myproxy/ca/
[11] https://ferret.pmel.noaa.gov/LAS/
[12] https://www.ogc.org/standards/wps

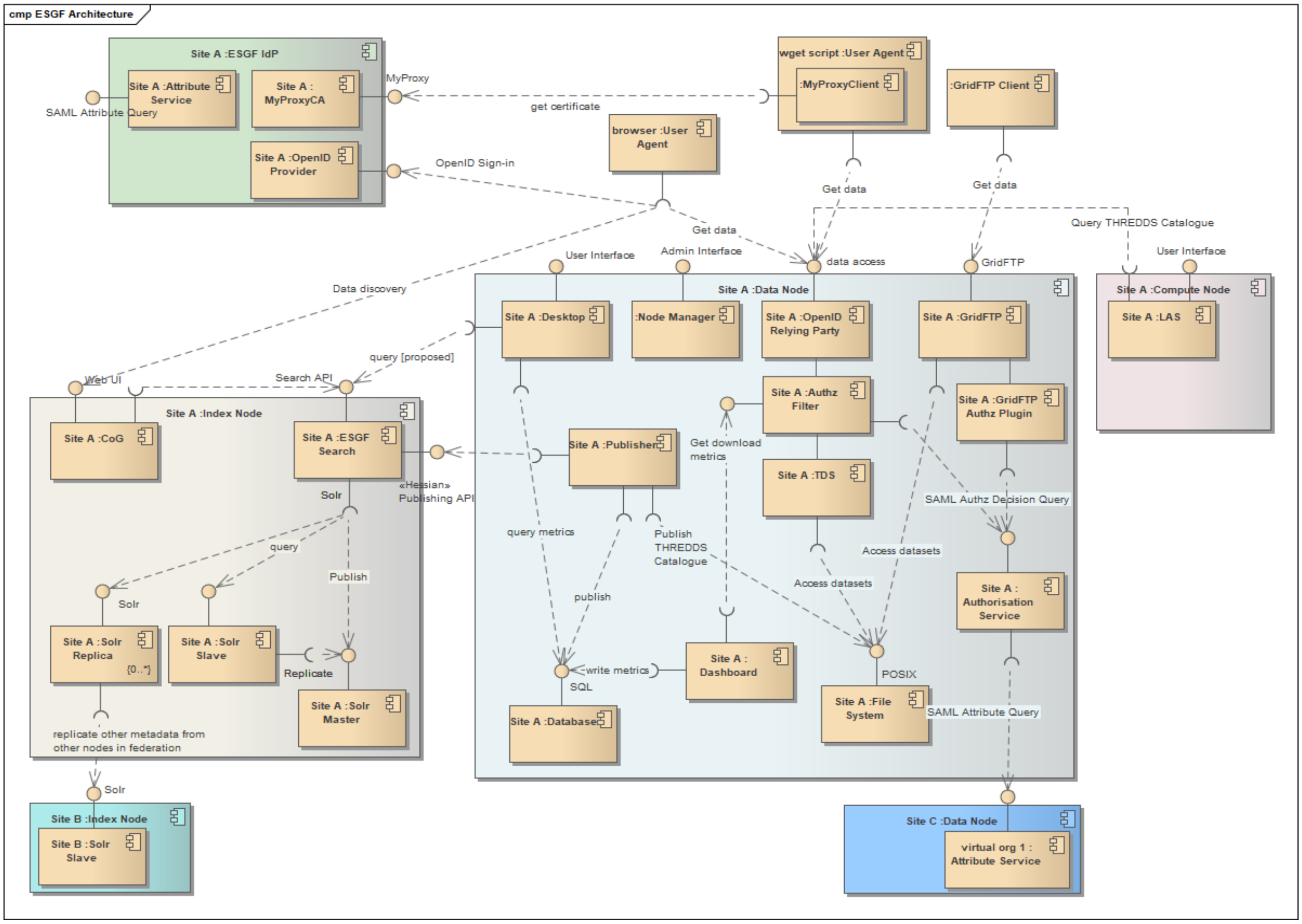

Figure 2: Existing ESGF Architecture

7.2 Similar systems and initiatives

Pangeo[13] is a project funded through the NSF EarthCube programme initiated in 2017. It has sought to build a community in the Geosciences around a common set of open source software for analysing large datasets. This software stack utilises cloud and container technologies, DevOps tooling and scientific packages. These include JupyterHub[14], Dask[15] and xarray[16] and Zarr[17]. xarray and Dask in particular facilitate analysis of large multidimensional datasets. Through grant funding an instance of Pangeo is running on Google Cloud together with a subset of CMIP6 data serialised with Zarr so as to optimise it for use with object storage which is more cost-effective when used on public cloud.

[13] http://pangeo.io/
[14] https://jupyter.org/hub
[15] https://dask.org/
[16] http://xarray.pydata.org/
[17] https://zarr.readthedocs.io/

The European Space Agency has funded a number of initiatives relevant to ESGF. The **HMA** (Heterogeneous Missions Accessibility)[18] project seeks to harmonise and standardise ground segment (Earth observation satellite data processing and archiving facilities) services and interfaces. FedEO[19] builds on the work of HMA and provides a brokered access to a distributed set of catalogues for EO data. A series of projects has sought to exploit cloud computing to address the challenges of Big Data - the SSEP (SuperSites Exploitation Platform), TEPs (**Thematic Exploitation Platforms**)[20] and **EOEPCA** (EO Exploitation Platform Common Architecture)[21]. The latter is currently underway and seeks to develop a common reference architecture for federating Earth observation platforms.

WLCG (Worldwide LHC Computing Grid)[22] is a distributed computing infrastructure to support the processing and analysis for experimental data from the LHC (Large Hadron Collider). WLCG differs from ESGF in its scale, sources of data (four main experiments from the LHC) and nature - it is largely a computational rather than data access infrastructure. It is predicated largely on a model of moving data to computing resources. It uses traditional Grid technologies originally developed in many cases around the needs of the particle physics community. Membership of the collaboration is via negotiation and signing of an MoU (Memorandum of Understanding).

ELIXIR[23] is a European collaboration to support life sciences research. It is divided into communities - domains in the life sciences - and platforms: tools, interoperability, data, compute and training. The compute platform includes a system for federated AAI (Authentication and Authorisation Infrastructure) across nodes, plans for data distribution and replication of *reference* datasets between participating nodes in the federation; in addition, cloud computing resources integrating with EGI (European Grid Infrastructure)[24].

18 https://earth.esa.int/hma/
19 http://wiki.services.eoportal.org/tiki-index.php?page=FEDEO
20 https://eo4society.esa.int/thematic-exploitation-platforms-overview/
21 https://eoepca.github.io/
22 https://wlcg.web.cern.ch/
23 https://elixir-europe.org/
24 https://www.egi.eu/

8 Technology Survey

A brainstorming session was held in the future architecture workshop to examine a range of technologies and their possible application to ESGF:

Technology	Applicability to ESGF	Notes
Intake[25], Intake-esm[26]	Publishing, metadata catalogue, search	YAML serialisation but also supports ElasticSearch[27] plugin
STAC[28]	Search interface standard	SpatioTemporal Asset Catalogue - a recently developed specification for geospatial search gaining popularity in the EO community
OpenSearch[29]	Search interface standard	Similar to the current ESGF search API but is an established standard used across the EO community. Self-describing interface
xarray[30], zarr[31], object storage	Data access, Data Node, data aggregation (TDS OPeNDAP[32]), User experience	Convenient interface for access, take up and interest amongst organisations and communities working with climate data. Has implications for file management (objects not stored as netCDF).
s3-netcdf-python[33], object storage	Data access, Data Node, data aggregation (TDS OPeNDAP), User experience	Similar to xarray/zarr/object store but objects stored formatted as netCDF. Technology still under development and testing.
CF-Python[34]	Data access, Data Node, data aggregation (TDS OPeNDAP), User experience	Provides a means for aggregating files together using a netCDF metafile.
Public (commercial) cloud hosting[35]	Potential to enable centralisation of identity and search services through resilient hosting	Cost and policy implications to consider
git-lfs[36]	Provenance and version management	Unlikely to scale for large file management; existing PID service performs this function

25 https://intake.readthedocs.io/en/latest/
26 https://intake-esm.readthedocs.io/en/latest/
27 https://www.elastic.co/
28 https://stacspec.org/
29 http://ceos.org/document_management/Working_Groups/WGISS/Projects/OpenSearch/CEOS_OpenSearch_Best_Practice_Doc-v.1.0.1_Jun2015.pdf
30 http://xarray.pydata.org/
31 https://zarr.readthedocs.io/
32 https://www.opendap.org/
33 https://github.com/cedadev/S3-netcdf-python
34 https://pypi.org/project/cf-python/
35 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
36 https://git-lfs.github.com/

Technology	Applicability to ESGF	Notes
git-annexe[37]	Provenance and version management	Unlikely to scale for large file management; existing PID service performs this function
PROV-O[38]	Provenance and version management	Already used by existing PID service
Blockchain	Provenance and version management	Provide a means of keeping a cryptographically secure ledger of transactions/state information
Prometheus[39] and InfluxDb[40]	Metrics, Dashboard	Have become industry-standard tools for metrics collection
Serverless	Installation, deployment	Serverless computing, builds on the concept of containers whereby application writers can deploy a snippet of code without the need to be concerned about the details of its operating environment (operating system etc.) Knative[41] - open standard for Serverless

Table 1: technology survey

The following diagram - Figure 3 - highlights some of the challenges around the adoption of new technologies for hosting and access to data when compared with more traditional models.

[37] https://git-annex.branchable.com/
[38] https://www.w3.org/2001/sw/wiki/PROV
[39] https://prometheus.io/
[40] https://www.influxdata.com/
[41] https://knative.dev/

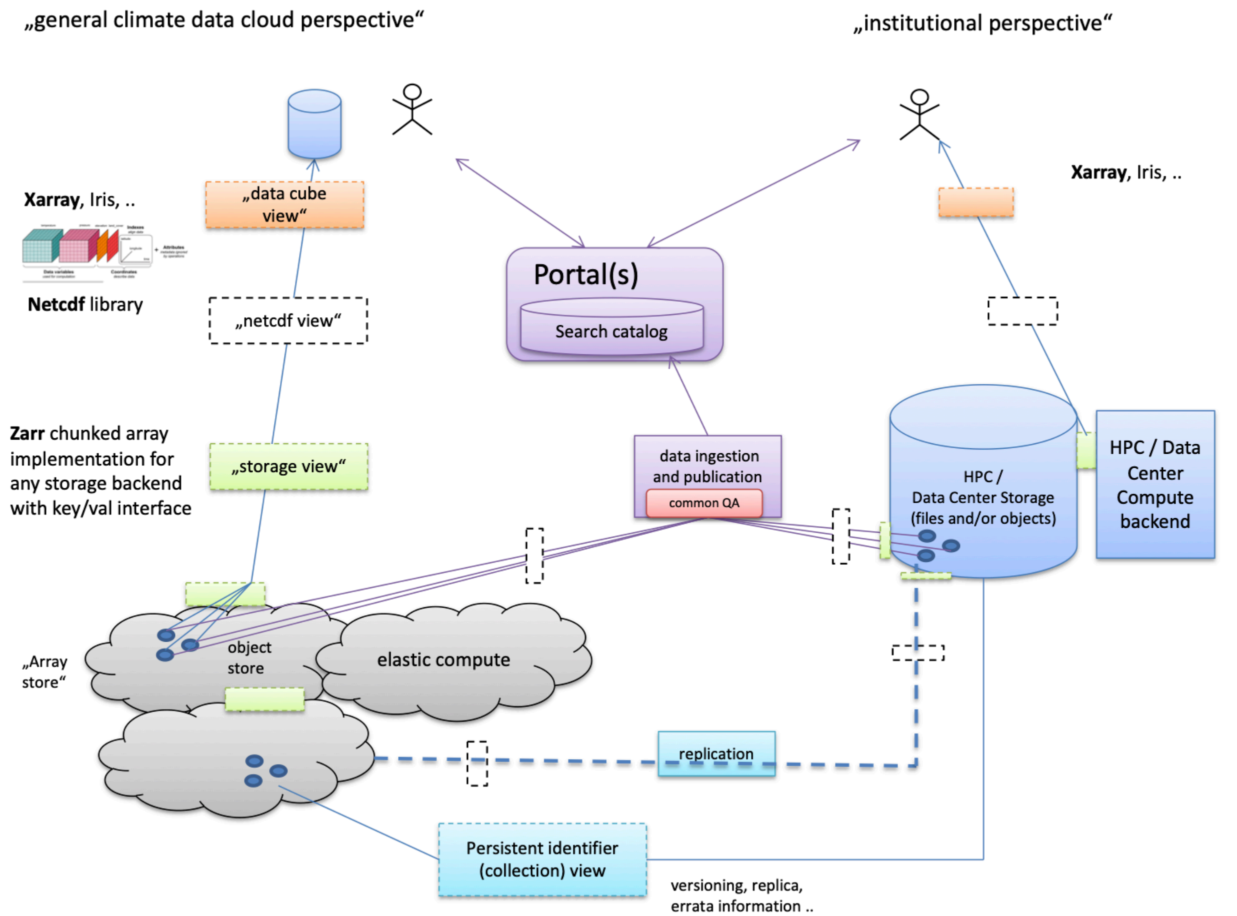

Figure 3: Contrasting perspectives on data hosting and access - institutional data hosting and data analysis from a cloud

Referring to the diagram:

- Alternative storage views are shown on each side of the diagram – cloud and institutional perspectives
- Data ingestion and publication in the middle of the diagram
- Replication pipeline between data centres and cloud - some centres are deploying to cloud
- Cloud could be public (commercial) or private/community (ie. operated by an ESGF partner institution)
- Persistent Identifier view – could track data that is cloud-hosted. The PID service is currently under-utilized
- QA (Quality Assurance) in the middle - needs to consider CVs (Controlled Vocabularies) and versions

9 User experience

9.1 Analysis

Here we consider users as defined in the previous section 6.1.

ESGF provides a system to dynamically generate scripts which download a set of files using wget[42] based on a user selection from the browser. This has been problematic in relation to error handling typically when one or more data nodes from which data is being retrieved is down, and with the integration of access control. The use of X.509[43] user certificates with the wget scripts has made them unnecessarily complicated. There have been improvements in user experience as a result of efforts to make the operational infrastructure more robust. For CMIP6, the removal of access control for wget scripts has made the system easier to use.

Developments in JavaScript provide the potential to greatly enhance the functionality for users. This includes the possibility of a new means of implementing bulk download. In this scenario, the user selects multiple files (based on search criteria) and selects download. Files are automatically downloaded to the user's client one by one. This is effectively a halfway house between the existing wget script capability and individual file download interactively from the browser.

9.2 Usability Requirements of Existing System

1. There is a common need for command line interface for data download and cache management.
2. Bulk data download from web-based user interface exists (wget throughout and Globus in a few deployments) but this needs improvement.
3. Tutorials and help documentation are needed to improve usability.

9.3 Technical Proposals

1. Make a new standalone web user interface for search. This should follow the principle of small modular components ie. it should only perform the function of search. It is likely to be implemented as a JavaScript app
2. Implement multi-download feature for search web user interface. User selects set of files for download, browser automatically initiates download of each file one after the other

[42] https://www.gnu.org/software/wget/
[43] https://tools.ietf.org/html/rfc5280

10 Data repository and management

10.1 Identity management

10.1.1 Analysis

Technology has advanced considerably in the last 10 years and there is a critical need to update the solution for ESGF to improve usability. At the time of its development, Shibboleth[44] (based on SAML 2.0[45]) was the de facto standard in the research community for single sign-on. However, the lack of support for a flow for non-browser-based interaction was a serious limitation for the application of Shibboleth for ESGF. Likewise, OpenID 2.0[46] was developed for browser-based interactions. It was gaining traction in the commercial sector but has since been widely deprecated in favour of newer standards OAuth 2.0[47] and OpenID Connect (OIDC)[48].

OAuth 2.0 fully supports use cases for delegation of access rights for users, important for more complex workflows and interactions that ESGF needs to support. It also supports non-browser-based interactions. OIDC builds on OAuth 2.0 to provide a full system for single sign-on. OIDC has had widespread adoption in the commercial sector and is being taken up as the default choice for research federations[49].

In the research community the Authentication and Authorisation for Research and Collaboration (AARC)[50] and AARC2 projects have done important work to establish baseline patterns for addressing use cases for research federations. These are expressed in the AARC architectural blueprint[51].

10.1.2 Requirements

1. A future system should seek to use widely adopted standards and off-the-shelf/open source where possible so as to avoid ESGF needing to build and maintain its own solutions.
2. The application of access control should be policy-based deriving from the needs of a given project hosted within the federation. For example, CMIP6 does not require access control but some resources e.g. computational may require registration and restrictions applied to access.

[44] https://www.shibboleth.net/
[45] https://www.oasis-open.org/standards#samlv2.0
[46] https://openid.net/specs/openid-authentication-2_0.html
[47] https://oauth.net/2/
[48] https://openid.net/connect/
[49] https://openid.net/wg/rande/
[50] https://aarc-project.eu
[51] https://aarc-project.eu/architecture/

10.1.3 Proposals

1. Build on work already completed to use OAuth 2.0 in the federation to support delegation use cases.
2. Augment OAuth 2.0 with OIDC and urgently deprecate OpenID 2.0 since it is no longer supported.
3. Address the barriers to more widespread adoption of OIDC in the federation namely a) bootstrapping of trust between IdPs (Identity Providers) and Relying Parties and b) integration of user management at IdP sites with new software supporting OIDC and OAuth 2.0
4. Adopt AARC blueprint pattern of a single Identity Provider proxy in order to address the many-to-many problem of multiple IdPs and Relying Parties and the necessary trust bootstrapping. Leverage public cloud hosting for the central proxy in order to provide the necessary operational resilience required for this service.
5. Review security policy implications for central IdP solution: GDPR[52], data sovereignty and user privacy.

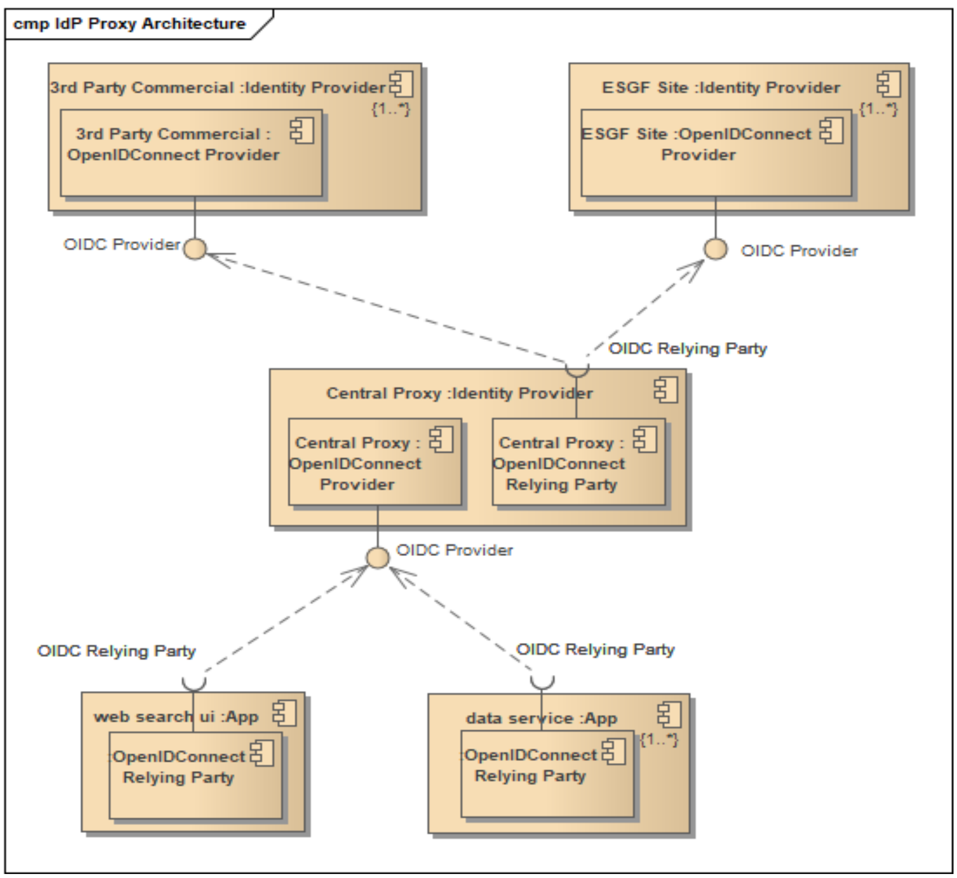

Figure 4: IdP Proxy concept from AARC Blueprint applied to ESGF. A central proxy mediates trust and identity management functionality between dependent ESGF applications (at the bottom) and other IdPs at the top. The IdPs at the top could include IdPs hosted at ESGF sites, external commercial IdPs or ones provided by NRENs or other research federations.

52 https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

10.2 Data services

10.2.1 Analysis

Our file format (CF-netCDF) has historically been driven by CMIP requirements. The format contains a basic set of usage metadata along with the climate model data itself. It is expected that this will continue to be the de facto standard for the climate community that ESGF supports.

Data access has been provided by ESGF through THREDDS Data Server and GridFTP. THREDDS has provided the core implementation for HTTP-based download and yet the future of this application is uncertain (e.g. version 5.0 used by ESGF has been in beta for some considerable time). It supports OPeNDAP but this interface has been used less in the first instance with CMIP5 because of the organisation of the data. THREDDS provides the ability to aggregate netCDF data over the time axis but performance can be limited for queries dependent on the volume of data requested and the number of time steps.

More recent innovations such as the xarray and zarr Python libraries move the aggregation management from server to client-side. A given configuration specifies a range of data objects held on a regular file system or object store. The latter has gained traction with the increased adoption of public cloud for serving scientific datasets. Object store is widely available on public cloud and provides a more cost-effective storage media than traditional POSIX file systems on cloud. A drawback of zarr is that when a given netCDF file is serialised to object store it is split into fragments whose configuration is managed on the client side. If this configuration is lost, it is difficult to rebuild the file from the fragments.

GridFTP provides a more efficient means for bulk data download than HTTP but in the early development of ESGF was difficult to integrate. Globus data transfer however is provided as SaaS (Software-as-a-Service) and also includes client-side tools for the major desktop operating systems. Further work could be prioritised to integrate it.

Multiple file download is also supported through a system of wget scripts. See User Experience section 9.

10.2.2 Requirements

1. Meet the needs for *MIPs
2. Continue to standardise on CF-netCDF
3. Data services must integrate easily with other such services at data centres deploying ESGF. This would be enabled by the adoption of common standards and solutions from across similar communities: Earth sciences, standards for geospatial data
4. Data replication needs to be presented to the user in a way that is clear and transparent
5. Versioning of data needs to be recorded and there needs to be better management of synchronisation, state and consistency

10.2.3 Proposals

1. Drop the use of THREDDS for file serving. Instead serve directly through Nginx. This will require the re-engineering of access control filters to integrate with Nginx.
2. Retain THREDDS for OPeNDAP on single files
3. Use client-side mechanisms for aggregation - Figure 5. Possibilities are xarray and zarr; CFPython netCDF metafiles; S3-netcdf-python library. See Figure 3.

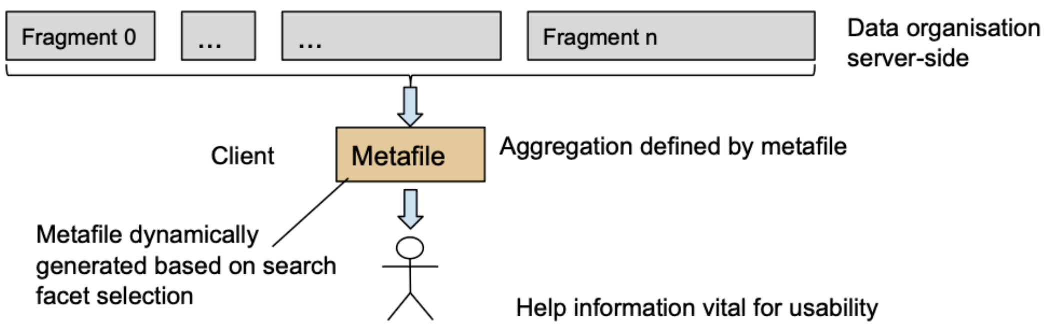

Figure 5: a model is desired whereby custom aggregations can be created on the client side to allow arbitrary querying of data along a time series or other axis. Aggregation definitions could be defined by a given search criteria.

10.3 Search API

10.3.1 Analysis

The Search API has provided a linchpin for federation. Clients may query any given Index node and obtain search hits for data hosted across a range of different sites around the world. The search system is built out of the NoSQL database application Apache Solr. Using Solr's sharding capability it has been possible to replicate search index content between nodes. This has provided resilience such that if one Index Node becomes unavailable another can be used in its place. More recently, a single public cloud hosted 'super Index Node' has been created containing the content from all the search indexes across the federation. This gives the advantage of having a single location for search queries and management of search content. Public cloud's capabilities for resilience and scaling mean that this could be considered as an alternative to hosting multiple copies of search content at different sites in the federation.

A problem with the current search API is that it is a bespoke one providing a thin facade over the underlying Solr API. This has meant that integrating ESGF data holdings with data repositories for other similar earth science and environmental sciences data can be challenging with some centres effectively running parallel infrastructures, one for ESGF and one for other datasets. ESGF would benefit from adopting a standard for its search API in order to facilitate better integration with other systems. However, there are a range of standards in use including OGC CSW, OpenSearch, STAC and ESM collection developed by NCAR. The latter has been implemented as a plugin to the open source Intake Python package used by the Pangeo project[53]. An ElasticSearch backend may also be available. ElasticSearch is an alternative search technology to Solr both of which are based on Lucene[54]. CEDA has utilised ElasticSearch for indexing of its data archive and this has shown to scale well for an index of ~250m files.

10.3.2 Requirements

1. Provide a means for users to search for data holdings by project across all the federation
2. Support DRS (set of controlled vocabularies) per project in order to facilitate faceted search.
3. Adopt community standards and existing solutions to support interoperability and reduce duplication of effort.
4. Enhance search capability for users by indexing from other sources beside source file metadata, for example ES-Doc and the PID system.
5. Support search by time interval.
6. Support geospatial search - query by region over a range of datasets - some datasets might have that region missing.
7. Support querying where information is provided about data gaps.
8. Keep data publishing and QA separate.
9. Develop a controlled Vocabulary of grid types to allow search by grid type.

10.3.3 Proposals

1. Evaluate community standards and initiatives and make a decision on which standard to adopt for the API. Candidates include: OpenSearch, STAC, ESM-Collection (NCAR). Implementations: CEDA OpenSearch implementation[55], Intake-ESM.
2. Pilot single super-index node for the deployment architecture (Figure 6): set-up super-index node, perform integration tests, evaluate and make deployment decision i.e. single super-index node or persist federated index node model, or some other hybrid approach.
3. Re-engineer publishing so as to index relevant content for ES-Doc so as to enhance ESGF Search capability.
4. Add support for search by time interval.
5. Add support for geospatial search (historically this has been a lower priority given the nature of the majority of data held in the federation – global projections data).

[53] https://pangeo-data.github.io/pangeo-datastore/
[54] https://lucene.apache.org
[55] https://github.com/cedadev/archive-opensearch

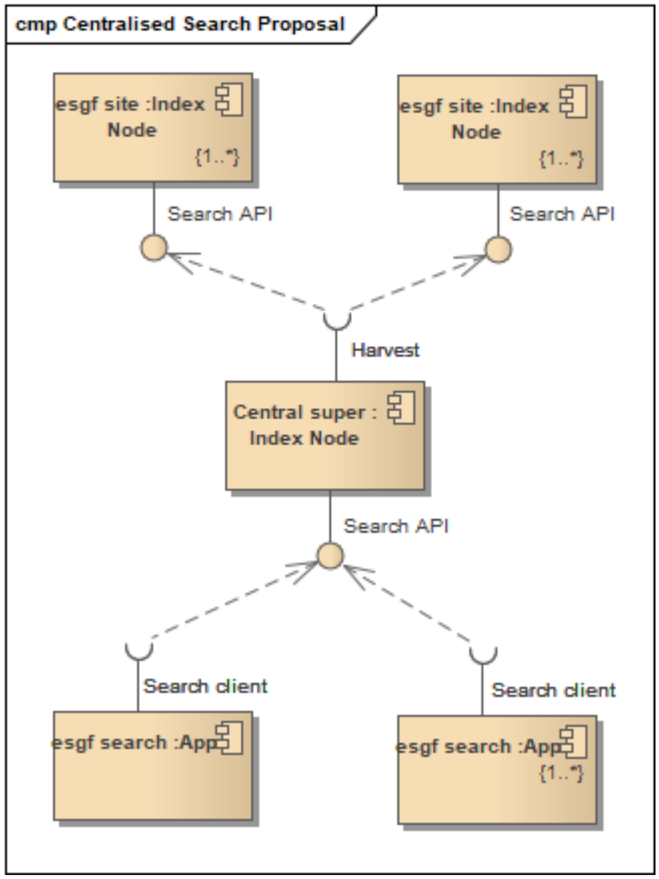

Figure 6: alternative architecture with single central search index and search service

10.4 Metadata Catalogues

10.4.1 Analysis

The Data Reference Syntax (DRS), a set of controlled vocabularies was first developed for CMIP5 and categorises datasets and facilitates search by providing search facets from which a user can narrow down their query based on characteristics of the data. Data is tagged using the DRS terms during the publishing process by extracting content from netCDF header files.

With more projects using ESGF, the DRS has been re-used, extended or modified to suit the new data's characteristics. However, there can be inconsistency in the use of tags and client applications such as CoG[56] need to be explicitly configured with a given set of facet names: there is no central consistent source for DRS vocabulary terms which is utilised by consuming clients. Within the environmental sciences community services have been created to maintain controlled vocabularies such as the NERC Vocabulary Service (NVS)[57]. For the ESA Climate Change Initiative Open Data Portal a vocabulary service[58] was created for serving a DRS

[56] https://www.earthsystemcog.org/projects/cog/
[57] https://www.bodc.ac.uk/resources/products/web_services/vocab/
[58] http://vocab.ceda.ac.uk/ontology/cci/cci-content/index.html

created for the data this project serves. This and the NVS take advantage of SKOS[59] to define terms. SKOS, based on Linked Data technology enables terms to be represented in a standardised way but also allows the definition of relationships between terms. Within ES-Doc, the pyessv[60] package also provides a way to define controlled vocabularies.

ES-Doc provides a central catalogue of information *about* climate models (Figure 7). Where the ESGF Search index sources content from netCDF data files, ES-Doc contains information directly from the modelling centres.

In the run up to the architecture meeting, a set of telcos were set up to discuss different aspects of ESGF in order to prepare. One area that was identified in the telco on user experience was the ability for users to make more sophisticated searches based on aspects of the model data. It was clear from this use case that there could be benefit from better integration between ESGF search and model information catalogued in ES-Doc: when using the former relevant content from the latter would not be found because it is not indexed into ESGF Search. An improved model is proposed in Figure 8.

---

[59] https://www.w3.org/2004/02/skos/
[60] https://pypi.org/project/pyessv/

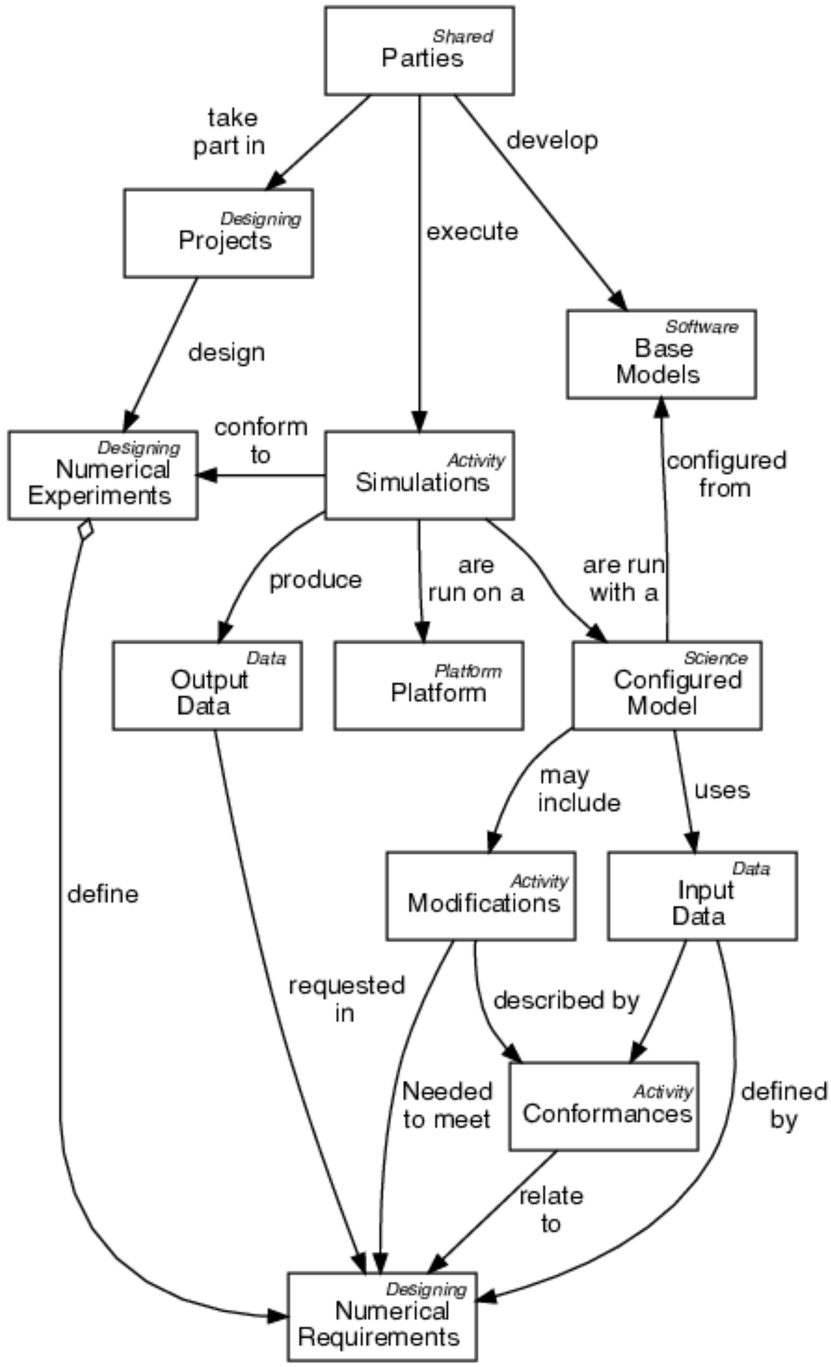

Figure 7: ES-Doc data model. Nearly all this information will be provided by some modelling groups to provide CMIP6 data provenance. Nearly every box represents a different information item with its own life cycle. Data object is not currently implemented

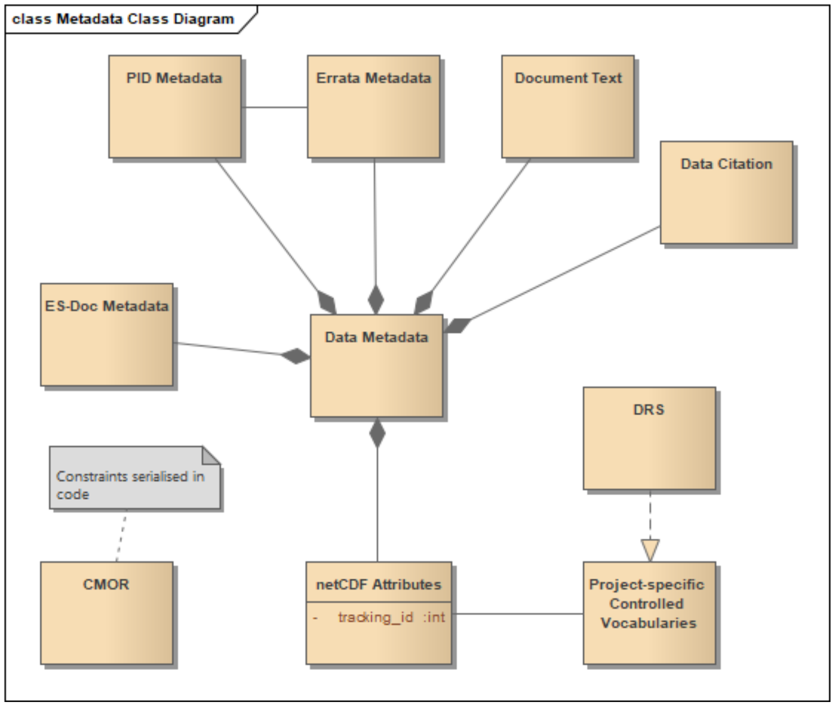

Figure 8: analysis of classes of metadata in ESGF and their relationship

10.4.2 Requirements

1. Provide a vocabulary service to provide the definitions of controlled vocabularies used.
2. It should be possible to support controlled vocabularies which are specific to a given project within ESGF (e.g. CMIP6, CORDEX) but also it should be possible to share controlled vocabularies across projects where appropriate or otherwise useful.
3. Clients and services dependent on these controlled vocabularies should use the vocabulary service as a single authoritative definition of these terms to ensure consistent use across a given project.
4. Extend Search results metadata as shown in the table below Table 2:

Existing System	Proposed System
Dataset id	Dataset version
DRS terms (search facets)	DRS terms (search facets)
File size	File size
Tracking id	Tracking id (PID)
PID (CMIP6)	
Checksum	Checksum
Access Methods	Access methods
File Version	

Existing System	Proposed System
Not present	Schema.org tags
Not present	ES-Doc content
Not present	date/time object for temporal search
Not present	Bounding box (geospatial constraints)
Not present	DOI (Citation) 1..n
Not present	Errata information (post-publishing)

Table 2: shows proposed changes for search results metadata for the search service

10.4.3 Proposals

1. Develop vocabulary service and integrate dependent services such as search and publishing with it.
2. Better integrate ES-Doc with the rest of ESGF services, especially search so that users can more readily find information about model data. Add other additional search metadata content as shown in the table above

10.5 Publishing, Replication and Versioning

10.5.1 Analysis

The ESGF Publisher enables data providers to publish CF-netCDF data into the ESGF federation. In practice this entails checking the validity of metadata content, the creation of THREDDS catalogues to configure what data is exposed and through what services. In addition, a key component is the process of indexing source files and copying key metadata content into Apache Solr[61], the NoSQL database used by ESGF. The sharding features of Solr are exploited to enable copies of search metadata to be replicated between different Index Nodes in the federation thus providing resilience such that if a given node is unavailable others can serve the same search queries from clients. However, this sharding approach can also be difficult to maintain. Operationally, upgrades have to be carefully managed so that interacting nodes have compatible versions.

The publishing software has evolved over many years, notably to integrate with Solr but also improvements to memory management have enabled it to better scale for large volumes of files associated for example with Earth observation data for the ESA CCI Open Data Portal project[62]. However, the software could benefit further with the ability to process data in parallel across multiple nodes.

[61] https://lucene.apache.org/solr/
[62] http://doi.org/10.5334/dsj-2020-016

ESGF also provides a system for replicating data between participating sites. Extensive operational and testing work has been carried out to achieve this. Together with the search system it enables ESGF to be a distributed information management system where data can be searched for from multiple entry points but also downloaded from multiple locations by virtue of the replicas. However, this system can be challenging to maintain operationally. There are policy questions about the service level that different sites are capable of sustaining, the uptime and resilience needed to maintain availability and the consistency of data and services.

Replication provides a valuable capability in terms of resilience of the service but it can cause confusion in the way that it is offered to users. Maintaining consistency of versioning between replicas has been an issue with the system historically but which has been addressed. Even so, the system would benefit from better communication and lifecycle representations (e.g. creation, deletion). A Git-style functionality for data versioning is desirable but this is difficult to implement practically given the size of datasets and files concerned.

10.5.2 Requirements

1. Publishing software should be modular and extensible to support other formats and metadata.
2. It should be possible to scale out publishing such that it can run in parallel.
3. It should be possible to replicate data between centres.
4. Separate publishing from QA process.
5. Enhance search capability for users by indexing from other sources beside source file metadata, for example ES-Doc and PID system[63].

10.5.3 Proposals

In the revised system we propose the following changes (Figure 9):

- Checking of data and indexing processes are separate
- Indexing of content from ES-Doc and PID service is done along with indexing of source data files
- Checking of data references project or federation-wide vocabulary services. The content of these services is version controlled and managed using a system of community governance.

[63] Nb. also listed as a requirement in preceding Search section

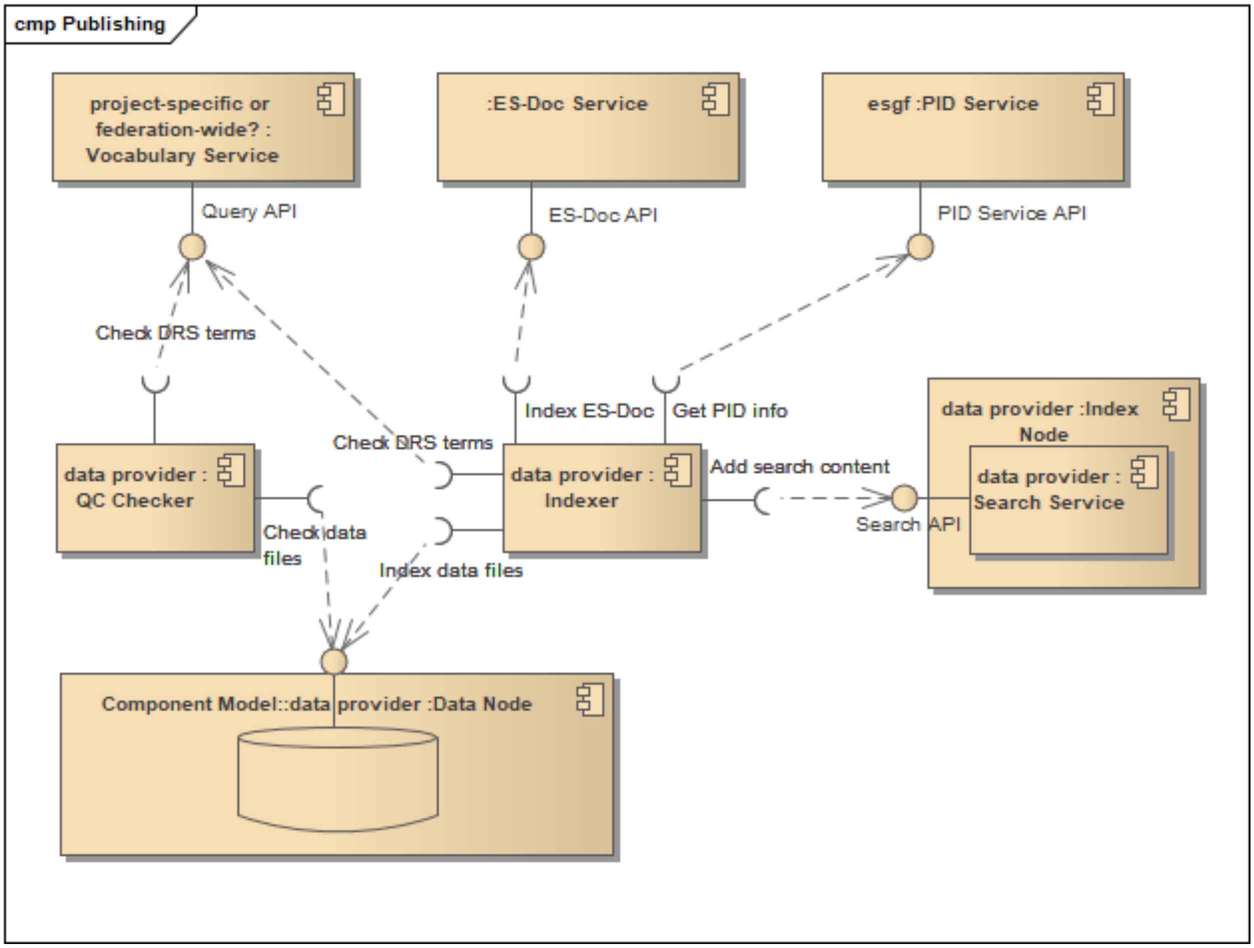

Figure 9: Proposals for revised publishing system

11 Compute on data

11.1 Analysis

The performance for data movement has historically been a limiting factor for moving data to compute. This section *Compute on Data* looks at the alternative paradigm of moving compute capability to where data resides at data nodes. This in itself presents challenges. For example, we need reproducibility and consistency for algorithms and code such that users have confidence that a given algorithm run at one location will produce results at another that are comparable. There are also considerations around access control since finite computing resources at any given site will need restriction. User delegation is implicit in compute workflows considering cases where a processing service may need to access other secured services on behalf of a user. There is some capability and experience in the community demonstrated with work done with the ENES Climate4Impacts Portal[64] and compute services developed by PCMDI which employ OAuth 2.0 to achieve this functionality.

A case could be made for a simple set of robust remote processing algorithms with focus on data reduction. The most desired compute functions being:

- Sub-setting - straightforward
- Averaging, min and max
- Re-gridding

The latter, re-gridding is very difficult to standardise. Different problems need different solutions. There is a need for knowledge of how to best apply and avoid users inadvertently using an inappropriate algorithm or settings for a given problem.

Such services may need to consider the semantics of matching an algorithm with suitable data and vice versa. These services should also be integrated with search services in ESGF so that they can be readily discovered and utilised.

Another consideration for compute services is their potential use in conjunction with Jupyter Notebooks for data analysis. One pattern could be to provide Python APIs to compute services available through notebooks. ESGF Tier 1 sites[65] could for example commit to provide JupyterHub instances for the user community.

Overarching all these considerations are some fundamental governance and resource management issues. If one site hosts compute services, who has access to these, how are computing resources accounted for and how do we ensure equitable share? Sustainability for processing services is also a concern. For all these reasons project-based initiatives are seen as being more likely in the short term rather than ESGF-wide development.

[64] http://www.climate4impact.eu/
[65] https://esgf.llnl.gov/esgf-media/2016-F2F/7-12-2016/community_software/F2F-2016-Tier1_Tier2.pdf

11.2 Requirements

Set the next section - requirements are specific to the needs of individual projects utilising ESGF.

11.3 Proposals

It was decided that compute on data work for ESGF is best pursued via project-based activities linked with ESGF. These activities will be tracked to see if there are suitable outputs for wider spread adoption across the federation.

12 Platforms and system administration

12.1 Analysis

Historically the system has been too monolithic. This makes it complex and difficult to change in an easy and controlled manner. This in turn risks stifling improvement and new innovation.

1. A more modular approach to service configuration and deployment is required as these are currently difficult to disentangle.
2. As a principle, configuration of services needs to be separated from deployment. - There are distinct requirements for the baseline infrastructure and applications which run in the infrastructure.
3. Investment of time should be made in long-term tracking and adoption of systems management best practices.
4. ESGF should take advantage of Infrastructure-as-Code approach for deployment practice.
5. Existing work with Docker containers, Kubernetes container orchestration and Ansible should be built upon. These all enable more standardization and flexibility with packaging and deployment. However, there is a need to avoid a split of deployment approaches: currently two tracks are being developed (Docker and Kubernetes based vs. Ansible based).

12.1.1 Role of public (commercial) cloud

Public cloud offers the potential for running services with a high level of resilience. The ability to replicate between regions can achieve this whilst also enabling geographic proximity to be exploited for access. ESGF has traditionally leveraged federation to provide resilience e.g. if one node is down, users can access replicated data at other nodes. Public cloud offers the potential to centralize services and avoid the need for sites to run duplicate services and reduce some of the operational burden. This has already been explored for ESGF with some test deployments including the EU Copernicus C3S 34a Lot 1 project which ran an index node on Google Cloud and a second example, a super index node run on AWS deployed by Luca Cinquini of NASA JPL. We list some of the other considerations for public cloud:

1. Data hosting is comparatively more expensive on public cloud vs. on-premise. Therefore, services with a smaller storage footprint promise a better cost/benefit.
2. Following the argument from the previous point, Index and IdP services would provide good candidates for centralised hosting with public cloud.
3. Public cloud hosting requires cost and policy questions to be addressed. For example, will this model work for all participating organizations and can this outsourcing model be trusted? Funding resources can influence an institution's decision to move to cloud (e.g., NASA is doing it because supporting monolithic compute doesn’t make sense in

their budget). Funding models change with moving to the cloud - does one site take responsibility or do all sites contribute?

12.2 Requirements

1. Where possible use widely adopted industry standards and technologies to ensure a maintainable interoperable solution.
2. Adopt a modular approach to service deployment.
3. Keep deployment configuration and application configuration separate.
4. Favour cloud native technologies for deployment where possible to engender portability between hosting environments be they on-premise traditional infrastructure, private/community (on-premise) cloud or public (commercial) cloud.

12.3 Proposals

1. Adopt Docker containers as the standard component to package applications. Provide dual supported methods for deployment a) Kubernetes container orchestration system b) Ansible deployment of Docker containers. This strategy takes into account the needs and capabilities of hosting sites.
2. Develop a modular deployment architecture such that individual applications can be deployed in isolation as required.

13 High-level Architecture

Based on the analysis from the previous sections, a preliminary new architecture can be sketched out. However, there are key areas where further exploration and discussion will be required. Referring to Figure 10:

- Top-left: search is centralised and cloud-hosted as a single index. However, it may aggregate content from individual nodes at participating sites in the federation
- Top-right: IdP proxy architecture with centralised IdP hosted on public cloud. This supports OpenID Connect for single sign-on and OAuth 2.0 for delegated access with tokens.
- Middle: Data Node architecture using Nginx for file serving and reserving THREDDS Data Server only for the purpose of providing OPeNDAP.
- Middle right: access control filters integrated directly with Nginx. This allows convenient integration with Kubernetes which can use Nginx as an Ingress Controller. Since all applications are hosted behind the Ingress Controller, they can all delegate access control to this entry point thus avoiding the need for application-specific access control functionality.
- Bottom: publishing system showing integration of content from additional sources into the search index – ES-Doc and PID system

Further areas to be described:

- Alternative data storage and access interfaces e.g. object storage / S3 interface. The nature of the object store interface is still to be determined e.g. xarray/zarr or other (Section 10.2.3).
- Additional metadata services (e.g. vocabulary services, citation, …).
- Metrics and logging system.

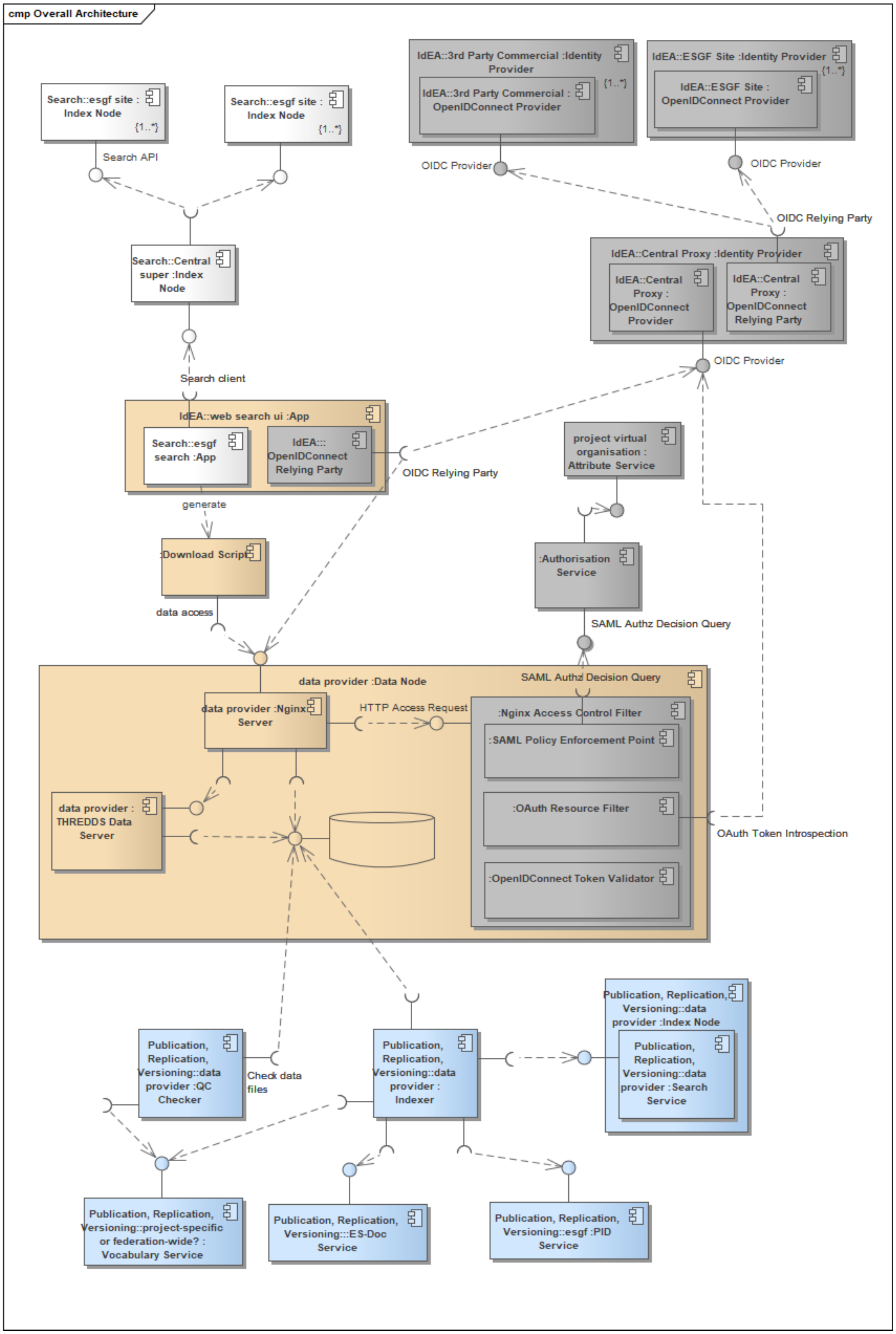

Figure 10: Overall architecture view. Nb. that this concentrates on data serving, identity management, search and publication aspects only

14 Prioritisation and Roadmap

1) There was a strong consensus to progress installation and deployment architecture:
    a) Make the core very robust, do the basic functions and services well.
    b) Adopt a modular approach to enable applications to be integrated easily.
    c) Keep to strict APIs to ensure clear contract and clean separation between interacting applications.

2) Identity Management:
    a) Centralise Index and IdP Proxy services to simplify operations and maintenance by taking advantage of public cloud. Federation has been historically favoured as a means to obtain operational resilience. Public cloud with its potential to offer high uptime now makes centralisation a viable option.
    b) Index node: a centralised index node has already been demonstrated by JPL (Luca Cinquini)
    c) IdP Proxy: apply the AARC architectural blueprint for identity federations proposes a central identity provider proxy service. This simplifies the integration of services since they only need to integrate with a single central IdP. It also offloads IdP hosting from participating sites. However, there is also the flexibility for sites to run their own IdP services integrating them with the IdP proxy

3) Platforms and system administration:
    a) Adopt Docker container as basic unit of deployment for all applications
    b) Together with the adoption of containers, support two deployment models
        i) Kubernetes
        ii) Ansible-deployed Docker containers on to regular hosts

4) Search services:
    a) Review Intake as a replacement for publishing system and possibly search API (Intake support for climate profile to STAC).
    b) Improve search service by indexing content from ES-Doc and from the PID Service. Use later to link with replicas.
    c) Make a new web-based search application
        i) For faceted search
        ii) Investigate multi-download capability with JavaScript.

5) Data services: Investigate client-side aggregation mechanisms
    a) Over traditional access mechanisms such as client-side interface to multiple OPeNDAP endpoints or netCDF direct download.
    b) Investigate and evaluate new systems such xarray and zarr over object store
    c) Address and improve support for bulk data download at sites
        i) Globus integration or other tools.

6) Compute Services: There was no consensus for federation wide supported Compute Node services. There is a recognition however that individual projects using ESGF may want to deploy web processing services.

7) Investigate toolset for industry-based metrics collection
    a) Prometheus endpoints
    b) Grafana or others for visualization

At the workshop a list of tasks was set out and put into priority order[66]:

ID	Task
1.	Keycloak investigation for single sign-on support
2.	Sort out dynamic registration of relying parties, plan engineering/transition to cloud-hosted IDP
3.	Transition to OpenID Connect
4.	Relying Party for Data Service
5.	Retire OpenID 2.0
6.	Cloud-hosted single IDP
7.	Group / VO registration service / attribute service
8.	Transition CoG (ESGF web portal) to new version of Django
9.	Get container images back to feature parity with ESGF's existing Ansible installer
10.	Containerization - Ansible-based
11.	Containerization - Kubernetes-based
12.	Investigate Intake, determine whether it can replace current publisher
13.	Investigate new search backend, e.g. ElasticSearch
14.	Investigate new search API, e.g. OpenSearch-based
15.	Make a new search user interface - needs to be able to query a service to get vocabulary info (needs its own design effort)
16.	Remove THREDDS catalogues (and database) from publishing - dependency on publisher work and on Intake investigation
17.	Implement new search interface to support multi-file download via simple web browser
18.	Refactor wget for access tokens (see remove X.509, below)
19.	Controlled Vocabulary service (see above lines)
20.	Modify Publisher to reference controlled vocabulary service directly

66 https://docs.google.com/spreadsheets/d/1QCxlfJVR2Xvx0HaTeABPn16yi3x6TB15tI25I50-ubE/edit?usp=sharing

ID	Task
21.	Other tools that reference controlled vocabulary service: QA tools, other tools. Needs a design effort.
22.	Re-examine facets and metadata, and crisply/succinctly identify the meaning of each
23.	Investigate the model of aggregation on the client side rather than server side Lazy aggregation - only move the data that's part of the aggregate...client fetches the data it needs and only the data it needs
24.	Transition Globus from X.509 to OAuth at the Globus Connect Server level
25.	Fix Globus config for downloads
26.	Retire X.509 user certificates for client authentication
27.	Improve Globus integration with CoG
28.	Cloud-hosted single index node
29.	Operational coordination between node managers for upgrades
30.	Investigate toolset for industry-based metrics collection - Prometheus endpoints - Perhaps Grafana for visualization? Need to investigate at the Federation Level
31.	Implement new search API
32.	Data usage metrics
33.	System monitoring/management metrics
34.	Re-architect catalogue investigation
35.	Re-organise index to be a flat index of files or retain model of an index of datasets and then index of files for each dataset
36.	Update publisher for new catalog architecture
37.	Index textual information from ES-Doc
38.	Harmonize file and index metadata for versions
39.	Add schema.org support to enable Google to index our content
40.	Add errata information to index
41.	Add DOIs to catalogue (part of re-architect catalogue)
42.	New publisher API
43.	Add ES-Doc terms to search capability
44.	Support for geospatial search
45.	Support for temporal search
46.	Sub-setting by time
47.	Sub-setting by space
48.	Add spatial resolution to search capability (including ranges, etc)

ID	Task
49.	Add search across N models, with checks for grid matching
50.	Make replica access transparent to users
51.	Improve replica semantics / management in file records
52.	Investigate integration of PID service into ESGF
53.	Modernize password hashing scheme - blocking factor for Jupyter and related services
54.	xarray / zarr pilot
55.	User migration
56.	Need to consider granularity of atomic items in index with move to object store model
57.	Investigate additional data service APIs (e.g. S3, HTTP, etc.)
58.	Investigate alternative to THREDDS for HTTP downloads (e.g. nginx)
59.	Investigate using alternative implementation for OPeNDAP (instead of THREDDS), eg Hyrax
60.	Make a better system for administrators to manage access policies (e.g. a service rather than a text editor)
61.	Develop a consistency/integrity service (e.g. crawler to check consistency of replicas, file checksums, etc)
62.	Investigate index record migration vs. re-publishing
63.	User-facing Synda[67] app, including documentation
64.	Modular Data QA checker - configurable via YAML
65.	Implement versioning / checkpointing for central index service
66.	Controlled vocabularies need to have version numbers
67.	Configurable QA in support of data publication
68.	Move PID service forward into new architecture
69.	Investigate cloud-hosted PID service
70.	Integrate PID service into new publisher
71.	What should a PID resolve to? PID system points to download URI not another service e.g. WMS
72.	Retire CoG

The two highest priority areas are identity management (adopt new solutions and retire legacy technologies) and containerisation (update container-based installation so that it is on a par with the ESGF Ansible installer).

[67] https://github.com/Prodiguer/synda

14.1 Steps November to June

- November - publish summary of this meeting, laying out the architecture
- November to March - intermediate tasks to demonstrate progress (see spreadsheet for assignments)
    - Search and Metadata Catalogues
        - Investigate Intake's viability for ESGF
    - Identity Management and Access Entitlement
        - Test identity management with access tokens
        - WPS pilot of Keycloak
    - Installation
        - Ansible containerization parity and installer proof of concept
    - Stress test the index node that's running in the cloud (including software requirements and cost analyses)
    - Project planning
        - Explore resource availability amongst partners
        - Perform critical path analysis on tasks to ensure delivery of goals
    - ESGF Executive Committee provides oversight
    - Arrange follow-up telecon with this meeting's participants (those who are able to attend) to review architecture documentation
- March (ESGF Face to Face meeting in Toulouse) - present roadmap to Steering Committee and IS-ENES3 General Assembly[68]
    - Announce container transition date (June 2020)
- June - deliver architecture document to IS-ENES

[68] Nb. As of writing, the Face to Face and IS-ENES3 General Assembly were re-organised as virtual meeting.

15 Summary

There is a strong consensus to progress the redevelopment of the ESGF software stack. There is a desire to focus on core services and implement a simple robust baseline. In this, modularity is critical and must be integrated as a principle throughout from design through to the implementation. The importance of cross-community standards is recognised and the need to better support data providers operating ESGF infrastructure alongside data and services for other domains.

New technologies provide new opportunities and capability for the future system. This can be seen across the four areas identified. System administration and platform management have been revolutionised across the IT sector with the adoption of cloud computing, infrastructure-as-code approaches and DevOps. ESGF can benefit utilising container technology to make a highly flexible and portable deployment system. Cloud computing itself allows new patterns to be adopted for federation including the centralisation of search and identity services.

For data repository and management there are discrete improvements that can be made to the search capability. New paradigms for data access have a fundamental impact for data providers on how data is stored and analysed. Further exploratory work and pilots will be needed in order to find the best solutions for wider adoption. The volumes of data concerned have been a motivator for new models for analysis, moving the compute to the data. Even so, there is currently no consensus on the development of such services that are standardised federation-wide. A number of projects leveraging ESGF are however building compute services. There is a need to track and coordinate between these activities and see where there are emerging solutions which could be adopted more widely.

From the prioritisation work a roadmap of development activity has been established. An initial release of ESGF is in preparation for June 2020. This addresses the fundamentals of the deployment system with the adoption of a more modular architecture built on container technologies. A set of further consultations is planned to address medium term development priorities taking into account sustainability of the system and the resourcing needed to support the work required.