ECMWF Copernicus Procurement

Invitation to Tender

Copernicus Climate Change Service Volume II

Copernicus Data Stores Modernisation

ITT Ref: CJS2_211
ISSUED BY: ECMWF Administration Department Procurement Section
Date: 25 May 2021
Version: Final

Funded by the European Union

Implemented by ECMWF

Contents

Table of Figures

1 Introduction

1.1 Overview

The Climate (CDS) and Atmosphere (ADS) Data Stores are the core infrastructure supporting the implementation of the Climate Change (C3S) and Atmosphere Monitoring (CAMS) Services which have been developed under the auspices of Copernicus, the European Union's Earth Observation Programme. C3S and CAMS services are implemented by the European Centre for Medium-Range Weather Forecasts (ECMWF) on behalf of the European Commission (EC). Both are highly visible web-based services addressing a diverse community of users, including policymakers, businesses and scientists, and helping them investigate and tackle climate change and atmosphere monitoring challenges. The following links provide more detailed information about the scope and current offerings of both services.

Service	**Link**
Copernicus Climate Change Service (C3S)	https://climate.copernicus.eu/
Climate Data Store (CDS)	https://cds.climate.copernicus.eu/
Copernicus Atmosphere Monitoring Service (CAMS)	https://atmosphere.copernicus.eu/
Atmosphere Data Store (ADS)	https://ads.atmosphere.copernicus.eu/

CDS and ADS are instances of the same underlying core infrastructure (generically referred as CDS infrastructure). Designed as a distributed system and an open framework, this shared infrastructure provides web-based and API-based retrieve facilities to a wide catalogue of datasets, applications and other digital information. It also provides a development platform (Toolbox) which allows the creation of web-based applications operating on the datasets and products available in the catalogues. These applications can subsequently be made available to end-users. All Toolbox computations are executed within the infrastructure in a distributed, service-oriented architecture (SOA). The data used by the applications does not leave the Data Stores, and only the results are made available to the users with the ability to download these results to their local system.

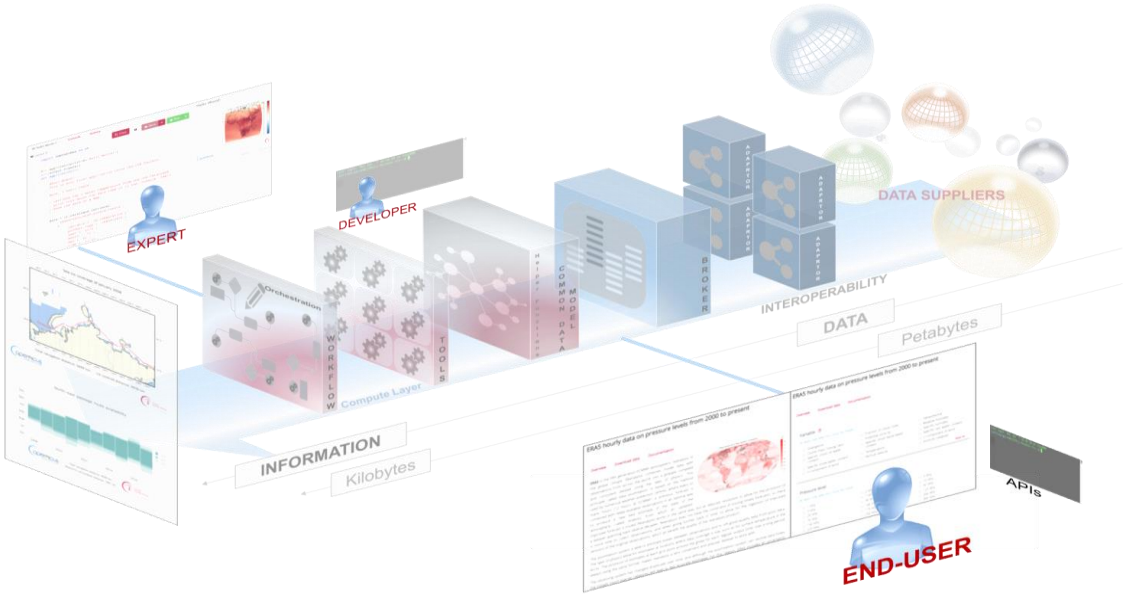

Fig. 1: Data Store conceptual framework infrastructure diagram

The CDS infrastructure is hosted in an on-premises Cloud physically located within the ECMWF Bologna Data Centre and implemented using a collection of virtual machines, networks and large data volumes within an OpenStack + Ceph cluster all orchestrated using Heat and automated with Puppet. The cloud storage is split on a mix of hard disks (HDD) and solid-state disks (SSD) distributed to maximise the benefit of the available SSD capacity. Elasticity of resources and automated deployment capabilities provide flexibility to deploy multiple instances of the Data Store infrastructure and dedicated servers for different purposes.

Scope of the tender: In the next Copernicus phase, Data Stores will remain the core building blocks of C3S and CAMS Services. The tendered service aims to modernize the underlying infrastructure by taking onboard operational experience, user feedback, lessons learned, know-how and advanced technologies. The main aims are to make the current services more accessible, take full advantage of the open-source scientific software that has become available since the lifetime of the current infrastructure, ensure compatibility with state-of-the-art solutions such as machine learning, data cubes and interactive notebooks and avail of the capabilities offered by the underlaying Cloud infrastructure as containerization. In summary, the goal is to evolve the current infrastructure into a modern, more usable, robust and interoperable one that will allow to strengthen synergies with related platforms, such as WEkEO, maximizing shared capabilities and resources to engage with a broader user community.

1.2 Glossary

In this document the following definitions shall apply for the purpose of this ITT. Where there is a difference between the definitions below and the definitions in Volume I of the ITT documents (Instructions and Conditions), the definitions below shall take precedence.

Name	Definitions
API	Application Programming Interface
Application	An interactive web page that displays maps, graphs and/or textual information that are the results of computations performed on the data and products of the Data Stores.
Broker	A middleware software component which will schedule and forward requests from the web portal to remote repositories.
CDS	Climate Data Store
CopDS	An acronym for Copernicus Data Store (which incorporates both the ADS and CDS) used in this document as to refer to the envisaged infrastructure covered by this tender.
ADS	Atmosphere Data Store
CDM	Common Data Model
CIM	Content Integration Manager
Data	The raw data collected.
Data Repository	A generic name for a system that holds data and/or products. This can be a database, a collection of files, etc.
Data Supplier	An organisation that makes its data and products available through the Climate Data Store
Developer	The type of CDS users adding tools to the CDS Toolbox framework
EQC	Evaluation & Quality Control
ESGF	Earth System Grid Federation
Expert	The type of CDS Toolbox users writing CDS Toolbox workflows to build applications.
Metadata	Descriptive information about the data
Product	A derived, valued added piece of information, usually generated from raw data.

Product Catalogue	A list of available products
Proof of Concept	The realization of a certain method or idea to demonstrate its feasibility or a demonstration in principle, whose purpose is to verify that some concept or theory has the potential of being used.
QAR	Quality Assurance Report
QAT	Quality Assurance Template
QoS	Quality of Service
SOA	Service-Oriented Architecture
Tool	A software that performs computations on some input data or products and produces some output data or products.
Toolbox	The set of available tools, workflows and applications within the CDS .
URL	Uniform Resource Locator.
Users	The internal and external Users of the C3S infrastructure.
UX	User eXperience
VLE	An acronym for Virtual Learning Environment.
Web portal	The web interface of the CDS that enables the users to view information, access/perform tasks on the Product Catalogue, Toolbox and CMS.
Workflow	A series of invocations of software tools whereby the output of the preceding tool becomes the input for the one which follows it until the required processing chain is completed.

1.3 Background

1.3.1 Current Infrastructure

The current infrastructure is designed as a distributed system and open framework, providing improved access to a wide range of datasets via a powerful service-oriented architecture. It offers seamless web-based and API-based search and retrieve facilities to access C3S and CAMS data and information.

The CDS/ADS contains a wide variety of datasets. These datasets are distributed and located at different data suppliers and are accessible via adaptors using a range of protocols.

The current infrastructure consists of the following components:

- **Web portal:** a single point of entry for the discovery and manipulation of data and products available in the catalogue. It provides access to content pages and different links to help desk, FAQs, user forum. It also has embedded functional content from third parties, e.g. quality assessment tab.

Reference	Link
Climate Data Store	https://cds.climate.copernicus.eu/
Atmosphere Data Store	https://ads.atmosphere.copernicus.eu/

- **Broker/scheduler**: schedules and forwards incoming requests to the remote distributed repositories. These requests can be either data retrieval requests or computation requests. This component also orchestrates more complex workflows and guarantees Quality of Service (QoS) by implementing queues and configurable rules.

Reference	Link
Status of the queues in the Broker	https://cds.climate.copernicus.eu/live/queue
Status of the limits implemented on QoS	https://cds.climate.copernicus.eu/live/limits

- **Compute layer:** performs computations on a combination of data retrieved from several distributed data repositories. Computation capabilities within the system are limited to those provided by the tools from the toolbox.
- **Adaptors:** these components guarantee interoperability between the broker and the various kinds of data repositories translating data requests into a format that is understood by the data source.
- **Results Cache:** this holds the results of computations ready to be downloaded by the user; all data and computation results are held by the cache area.
- **Monitoring/Metrics**: these elements collect statistics on the whole system to monitor operations and measure KPIs respectively.

Reference	Link
CDS Live	https://cds.climate.copernicus.eu/live/
CDS EQC KPIs Dashboard	https://eqc.climate.copernicus.eu/monitoring#highlevel_view;all

- **Web API**: A web based Representational State Transfer (REST) application programming interface (API) that allows users to automate their interactions with the CDS.

Reference	Link
API documentation	https://cds.climate.copernicus.eu/api-how-to
Toolbox API reference	https://cds.climate.copernicus.eu/toolbox/doc/api.html

- **Toolbox:** Toolbox concept and infrastructure is described in more details in the next section.

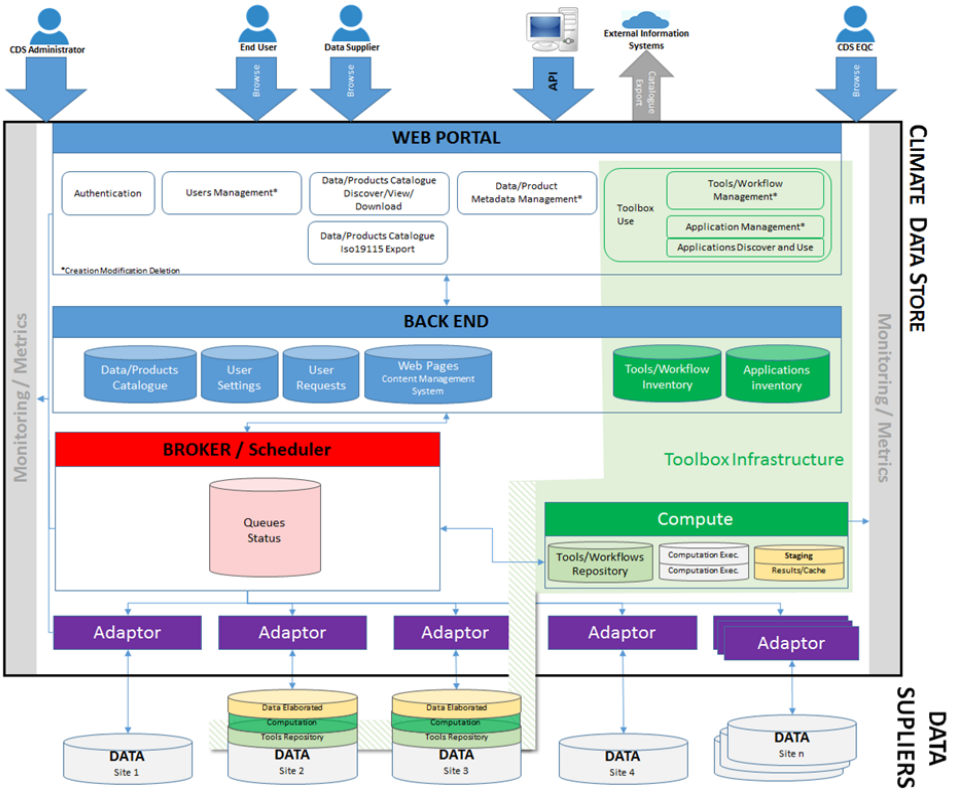

Fig. 2: Climate Data Store infrastructure components

By means of its components the CDS provides access to Data Repositories, which vary greatly in size, nature and formats. It should be noted that some of the repositories contain many petabytes of data and several billion products. Data repositories are expected to grow substantially in the coming future.

The CDS infrastructure is hosted in on-premises OpenStack cloud services and implemented using a collection of virtual machines, networks and large data volumes within an OpenStack Pike + Ceph Luminous cluster all orchestrated using Heat and automated with Puppet.

1.3.2 Toolbox

The CDS Toolbox is a platform that can be used by developers to create web-based applications that use the datasets and products available in the CDS catalogue. These applications are subsequently made available to end-users. Users are given some control over the applications by interacting with web form elements. For instance, enabling selection of a range of dates or a geographical area of interest, which are then used to parameterise the application.

Reference	**Link**
Toolbox editor	https://cds.climate.copernicus.eu/cdsapp#!/toolbox
Toolbox documentation	https://cds.climate.copernicus.eu/toolbox/doc/index.html
Example of CDS public application	https://cds.climate.copernicus.eu/cdsapp#!/search?type=application
Example of CDS data viewer	https://cds.climate.copernicus.eu/cdsapp#!/dataset/sis-marine-properties?tab=app

All computations are executed within the CDS infrastructure in a distributed, service-oriented architecture (SOA). The data used by the applications does not leave the CDS, and only the results are made available to

the users. These results are typically in the form of tables, maps and graphs on the CDS data portal. Users are also offered the ability to download these results to their local system.

Available datasets are encoded in files using various formats, such as WMO GRIB or NetCDF-CF. The data types range from single point observations or time series at a given location, to multi-dimensional global fields. The variety of data types, formats and structures make their combined use highly challenging. The Toolbox provides a set of high-level utilities that allow developers to implement applications without the need to know about the specifics of the different data encodings used.

The Toolbox also hides the physical location of the datasets, their access methods, formats, units, etc. from those who are developing the applications. Developers are presented with an abstract view of the data based on the Toolbox Common Data Model (CDM), which aims to be capable of representing all the datasets available in the CDS.

The Toolbox also provides a series of tools to perform basic operations on the datasets, such as averaging, calculating differences, sub-setting, etc. For performance reasons, some tools are specialised for certain datasets (e.g. sub-setting GRIB data and sub-setting NetCDF-CF data) and can be executed next to the data. This aspect is also hidden from the application developer, who is only provided with a single view of a tool (e.g. only one sub-setting operation can be selected). All tools are registered in a database and documented within the CDS.

Application developers can create workflows using Python and the tools and datasets available to them on the CDS (see Figure 3). They can associate the workflow with a web page containing input widgets (check boxes, drop down menus, etc.) used to parameterise the workflow, as well as output elements (tables, maps or graphs, etc.) generated by the parameterised workflow when it is executed (see Figure 4). The output elements are then dynamically updated as end-users interact with the widgets.

Public applications and data viewers currently available on the CDS will have to be migrated and adapted to run in the new infrastructure. The Tenderer will be required as part of this contract to facilitate and support this migration.

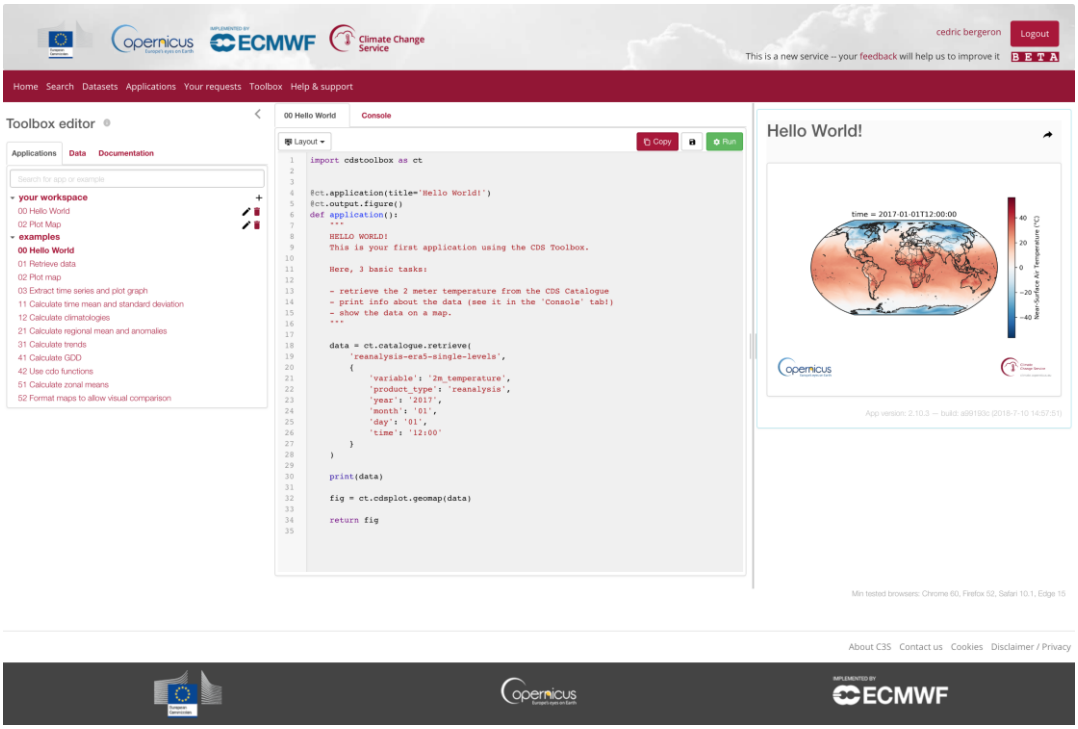

Fig. 3: The CDS Toolbox editor allows users to develop workflows and applications. They can also access examples and documentation.

The Toolbox is nevertheless a constrained environment which limits the options in terms of development tools and data that users can integrate on workflows and applications to those made available within the CDS.

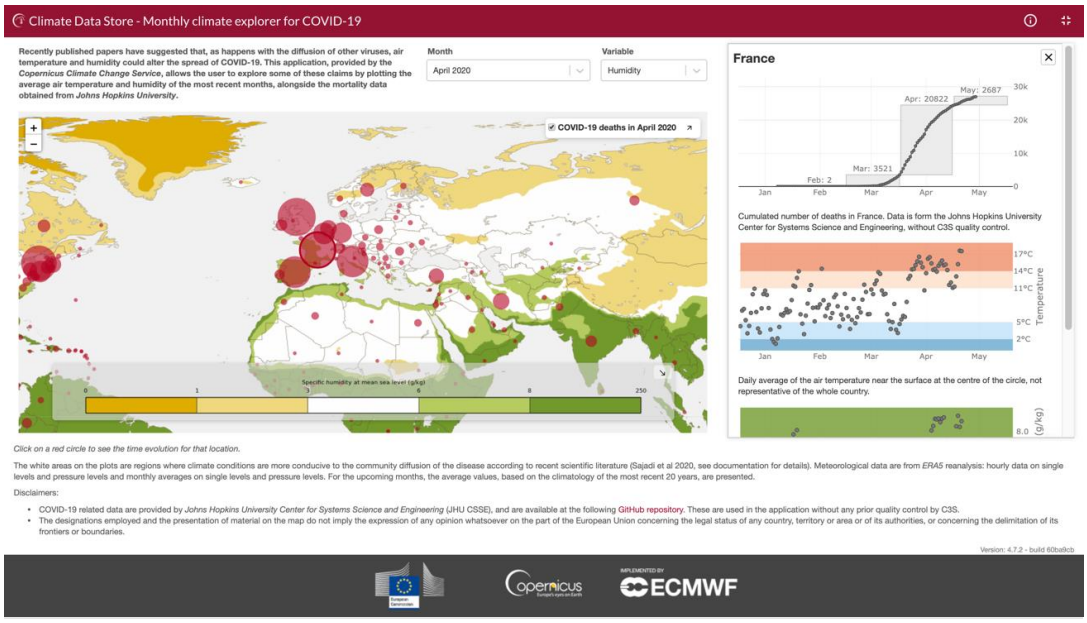

Fig. 4 : Example of a CDS Toolbox application.

1.3.3 WEkEO

As part of the European Union's Copernicus Programme, ECMWF, together with EUMETSAT and Mercator Ocean International, have joined forces to implement a Data and Information Access Services (DIAS) Platform called WEkEO.

WEkEO DIAS platform	https://www.wekeo.eu/

WEkEO is a distributed cloud-computing infrastructure used to process and make the data generated by the Copernicus Environmental Services (CMEMS, CAMS, C3S and CLMS) accessible to users. It also provides privileged access to derived products from these services and to all satellite data from the Copernicus Sentinels operated by EUMETSAT and ESA.

Within the partnership, ECMWF is responsible for the procurement of the software to implement Data Access Services, Processing and Tools. ECMWF developed the requirements for the software and awarded and managed a contract for the implementation of those requirements to a European provider of software engineering services. The services have been integrated with the delivery platform by our partner EUMETSAT and their suppliers. The service is now live on the Internet.

The delivery platform uses OpenStack + Ceph. End-users are able to access a substantial portfolio of original Copernicus Programme and Sentinel satellite data using a harmonised data access mechanism and common data catalogue. They are able to harness compute resources to process that data without the networking and storage costs associated with public cloud offerings. They have access to a wide range of tools and technologies in the areas of DevOps, including data access, artificial intelligence, machine learning, workflow, compilation, build and more. The CDS/ADS data is also available through the WEkEO DIAS to allow bespoke processing of that data in unison with all other DIAS data using a common data access mechanism.

The Tenderer is envisaged to provide as part of its proposal references detailing how the proposed components will allow to strengthen synergies with the WEkEO platform.

1.3.4 Workload and Usage Statistics

Copernicus Services are looking to modernize the current CDS infrastructure as to allow to effectively accommodate a continuous evolving demand in terms of:

- Number of users and their diversity.
- Volume of data being stored and delivered.
- Computing resources.
- Diversity of products being produced and served.
- Distributed nature of the accessed data repositories.
- Integration and interoperability with open development environments.

At the end of Q1 2021, global CDS statistics were:

- > 80.000 Registered users with an average of 120 new user registrations every day.
- > 65.000 active users with an average of +-1500 active users per day.
- 65 PBs delivered during Q1 with a daily average of +-72 TBs in the form of +- 300.000 requests per day.
- 91 catalogued datasets and 23 public applications, plus 37 embedded on the European Climate Data Explorer.

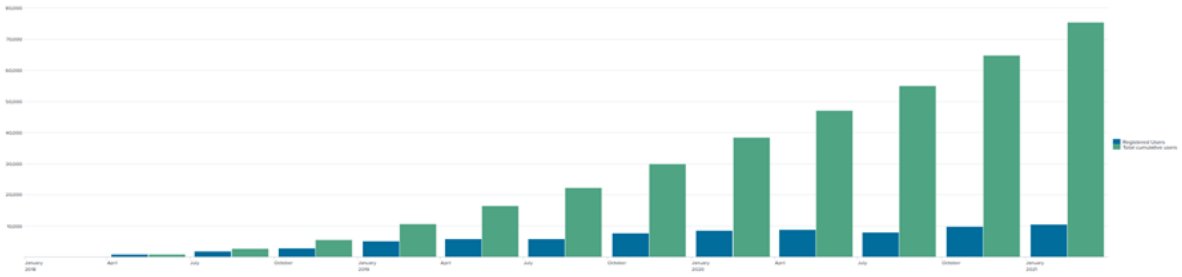

Fig: User registration evolution per quarter (Q1)

Reference	**Link**
CDS Live statistics	https://cds.climate.copernicus.eu/live

2 Contract Definition

2.1 Vision and Objectives

Vision: to enhance and upgrade C3S and CAMS services with a modern, more usable, scalable and interoperable Data Store & Toolbox infrastructure *(referred in this document as CopDS)* that will engage with a broader user community and will facilitate interoperability and synergies with external projects and platforms.

The objectives of this contract can be summarized as:

- Capitalize operational experience, user feedback, lessons learned and know-how from existing Climate (CDS) and Atmosphere (ADS) Data Stores to move into a modern, more reliable and interoperable platform.
- Uptake modernised technologies which have evolved since the initial implementation of the current infrastructure.
- Embrace open-source scientific software and ensure compatibility with state-of-the-art solutions such as machine learning, data cubes and interactive notebook.
- Strengthen synergies with DIAS WEkEO platform and improve the capacity, efficiency, interoperability and reliability of shared interfaces and resources.
- Provide improved and flexible access to data and toolbox capabilities from multiple development platforms.
- Evolve the system architecture as to take full advantage of cloud computing technologies and underlaying cloud infrastructure.
- Consolidate the integration between different software systems that integrate C3S and CAMS.
- Take over the technical maintenance of the current CDS Infrastructure covering the Climate (CDS) and Atmosphere (ADS) operational systems and to assure a smooth transition of content and operation to the new developed infrastructure.

2.2 Statement of Requirements

2.2.1 General Requirements

The following sections detail the project requirements, summarise the expected architectural requirements and functional scope of the system split by the main envisaged components. In each section, a brief description and a table of baseline requirements has been provided for the Tenderer to consider. Listed components and associated table of requirements does not indicate any limitation with regards to implementation.

In some instances, alternative solutions may exist to address these requirements. ECMWF would like the Tenderer to submit details of such alternative options as part of their response to this ITT.

All enumerated requirements describe a baseline of how ECMWF envisages the potential solution to be implemented. They are by no means a constraint for the solution to be proposed by the Tenderer but shall serve as a guide to define the scope of the proposal.

Tenderers are expected to refer on their responses to the existing functional components and describe how their proposal is envisaged to improve the current functionality.

Being at the core of the implementation of the services, *CopDS* has strong interdependencies with other C3S and CAMS areas of work and related EU funded projects. During the execution of this project interaction with third parties will be required. Envisaged areas of interaction are mentioned in the following chapters.

Tenderers are expected to propose a flexible workplan methodology which will allow interaction with third parties and integration of new requirements throughout the duration of the project.

CopDS will be hosted in an In-house Cloud Infrastructure physically located in ECMWF premises in Bologna (Italy). Cloud system is otherwise functionally identical to the existing one (see chapter 1.3.1) save for the addition of OpenStack's container (Zun), container networking (Kuryr) and container orchestration (Magnum) capabilities.

2.2.2 Architectural Requirements

The architectural requirements must be developed to support the services provided to users and data suppliers. In addition, they should be sustainable and flexible in design to allow for technological advances. The final architecture must be extensible (plug-in architecture). This can be achieved by ensuring that the software developed acts as a software framework.

The architecture of the system will be service oriented, open and extendable in terms of the number of concurrent users, handled data volumes and processing capacity for workflows, tools and applications.

Architectural components shall be scalable and allow for distributed computing. They must be designed as to take full advantage of the underlying cloud infrastructure. The architecture is expected to use containers (microservices) managed by and orchestrator software for deploying and managing those containers at scale.

The modernized system infrastructure is expected to be re-engineered as to maximize the capabilities of the underlying cloud infrastructure moving into a hyper-scalable container-based platform.

Computations that involve very large amounts of data (e.g. more than 1 TB) will have to be performed at the data repositories. Computation involving smaller amounts of data can be considered to be performed centrally, or using cloud services.

The proposed system architecture will address the following requirements:

Reference	Requirement	Remarks
1.	Must be developed to support the services provided to users and data suppliers.	
2.	Architecture is expected to use containers (microservices) managed by an orchestrator software.	Docker, Kubernetes.
3.	Should be sustainable and flexible in design to allow for technological advances.	
4.	Must be extensible (plug-in architecture).	Adaptors, protocols, tools.
5.	Must ensure that the software developed acts as a software framework.	
6.	Must be scalable, interoperable and open by design to enable interaction and integration with other platforms and services.	Learning virtual environment (LVE), User support, etc.

7.	Design must allow for growth in volume of data, number and size of queries, and the potential increase in the number of users.	
8.	Should not rely on specific hardware infrastructure so it can be migrated to other environments.	
9.	Must run on a Linux/UNIX operating system.	
10.	All necessary documentation for the administration of the system must be provided.	
11.	The deployment of the system must be fully automated. It should be possible to create a fully functional CDS from scratch (e.g. a test system) , as well as deploying incremental software releases without interruption of service.	The automation must also manage the content of any databases (e.g. the Catalogue).
12.	Must implement automated tests suites for software delivery and integration.	

Tenderers should include in their response how their proposed design ensures future scalability.

Modernized infrastructure shall allow for automatic deployment of changes but also for the deployment from scratch of fully working instances of the system, including content. Instances shall support dedicated configuration.

2.2.2.1 Interoperability

Interoperability is key for the evolution of the system as it facilitates the integration and communication at different levels, from data providers to external platforms and services.

The interoperability shall be achieved:

- Internally, between the infrastructure and the distributed data repositories.
- Externally, with information systems such as the INSPIRE Geoportal, the WMO Information System (WIS), the Global Earth Observation System of Systems (GEOSS) or the WEkEO platform.

In order to achieve interoperability, the *CopDS* must implement agreed international standards (see Appendix 5.1)

2.2.2.1.1 Internal Interoperability

For information purposes please note that data suppliers will make their outputs available to the users via the *CopDS*. They can do so by either of the following methods:

(a) uploading their data and products to a designated server (“push”) within the dedicated Cloud Infrastructure (e.g. when volumes are small and no infrastructure in place at their site).

(b) providing them via web services (e.g. when volumes are large or infrastructure is in place).

In the case of (a), suppliers will only use data formats agreed by ECMWF. ECMWF only accepts data in formats that follow internationally recognised standards. Such standards must be open (e.g. non-proprietary), managed by recognised international standardisation bodies (e.g. ISO, WMO, OGC, etc.), or any de-facto standard Open-Source software should also exist that can read and write files of these standards.

Serialisation formats (e.g. NetCDF, XML, JSON) should be supported by standard schemas and conventions. All text-based formats should be encoded in UTF-8. ECMWF implements tools to check the compliance of the provided data and products to the agreed standards before they are added to the Data Stores as part of the Evaluation and Quality Control component.

Examples of case (a) are data uploaded to the *CopDS* in WMO GRIB edition 1 and 2, NetCDF files conforming to CF-1.6, or greater.

In the case of (b), suppliers will only use protocols agreed by ECMWF to invoke the web services. ECMWF will only accept protocols that follow internationally recognised standards. Such standards must be open (e.g. non-proprietary), managed by a recognised international standardisation process (e.g. ISO, WMO, OGC, etc.), or be a de-facto standard such as OpenDAP. ECMWF will consider using bespoke web-based APIs to access the data and products if they implement very simple protocols (e.g. REST), and the results returned by these APIs are compatible with (a). It should be noted that requests for these web services will mostly originate from *CopDS* itself, as part of a workflow run on behalf of an end-user; ECMWF will therefore need to have the necessary credentials to invoke these services. ECMWF will not provide information on the end user's identity when invoking the web services. ECMWF will nevertheless collect usage statistics for all aspects of the *CopDS* operation.

Examples of case (b) are OGC standards (WMS, WCF, WFS, etc.), OpenDAP, etc. Other protocols could be considered as the system evolves.

Every dataset and/or service provided will have to be documented using the appropriate metadata standards (ISO19115).

Data providers that make very large remote datasets available (for example, >1Tb) are asked (when appropriate) to implement specific web services-based data manipulation facilities. These will allow some agreed reduction and/or analysis algorithms to be run directly on the data and products located on the suppliers' systems and return the results of said algorithms.

CopDS should ensure that these services are invoked in a controlled fashion, to prevent any misuse of the system. These web services will be implemented with either OGC's WPS standard, or based on simple web based REST API or equivalent. The results returned by these services will have to be in formats compatible with (a) or (b).

2.2.2.1.2 External Interoperability

The most important standards required are recommended by the INSPIRE Directive (see Appendix 5.1) which will allow interoperability with other European platforms:

- When applicable, users will be given the ability to view the data and products from the Product Catalogue (INSPIRE view service, using the relevant OGC standards).
- When applicable, users will be given the ability to download the data and products (INSPIRE download service, using the relevant OGC standards).
- It should be possible to query the Product Catalogue programmatically (INSPIRE discovery service, using the relevant OGC standards).

CDS Catalogue (CSW)	https://cds.climate.copernicus.eu/geonetwork/srv/eng/csw?SERVICE=CSW&VERSION=2.0.2&REQUEST=GetCapabilities

Other INSPIRE services will be implemented at a later stage.

The *CopDS* will also have to be interoperable with:

- WMO Information System (WIS).

- WEkEO DIAS reference service for environmental data.
- Global Earth Observation System of Systems (GEOSS).
- Global Framework for Climate Services (GFCS).
- World Climate Research Programme (WCRP).

The proposed system architecture will address the following requirements:

Reference	Requirement	Remarks
13.	The Tenderer must ensure interoperability as defined in the text above and ensure compatibility with standards in Appendix 5.1.	

Tenderers should include in their response which standards need to be baselined to ensure interoperability at the mentioned levels.

2.2.2.2 Open Source Software

The Tenderer shall propose software solutions which are considered to be the most practical and cost effective for ECMWF's intended use and future evolution and maintenance.

ECMWF expects the Tenderer to propose open-source software, whenever it is available and can provide or contribute to a practical solution. A considerable amount of open-source software exists, which the Tenderer shall consider as opportunities for use, this includes but is not limited to that software listed in Appendix 5.2

As part of the proposal for each component, the Tenderer must provide a list of:

- Open-Source software considered.
- Open-Source software proposed for use.
- Components being developed.
- Programming languages that will be used.
- Any proprietary or non-Open-Source item that may be used.

If a proposed solution would require the modification of open-source software, ECMWF and the Tenderer will first consult about the permissions or restrictions imposed by the relevant open-source licence and the risk that the modified software will become obsolete and unsupported over time.

Ownership of all rights in the software solutions, which the Tenderer develops for ECMWF, shall be assigned, through ECMWF, to the European Commission (EC).

The result of the work tendered will itself be released under an open-source licence.

Technological choices will be based on existing open-source software, Python being the preferable programming language.

2.2.2.3 Data Licensing and Data Policy

Data and products in the *CopDS* will be subject to Terms and Conditions of Use. Moreover, licences may be necessary to access certain data and products.

Users will be asked to accept the Terms and Conditions prior to accessing the data and products. The Terms and Conditions may vary for each set of data and products. Under certain circumstances ECMWF might consider to grant access to selected data and services as Open Data for unregistered users.

The fact that the user has accepted the *Terms and Conditions*, or has been granted a licence for the use of these data and products, will be stored in the user profile.

CDS Licence example	https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels-monthly-means?tab=overview

Certain data and products may in addition require specific licences to be granted by the data or product owner. The process by which these licences are issued is out of scope.

The proposed system architecture will address the following requirements:

Reference	Requirement	Remarks
14.	Ability for the product catalogue to hold references to the Terms and Conditions and/or licences.	
15.	It should be possible to associate the metadata record in the product catalogue to a given set of Terms and Conditions and/or licences.	Versioning of Terms and Conditions and/or licences shall be considered.
16.	If there are Terms and Conditions and/or licences attached to a data or products, the user will only have to accept them once to be granted access for each request for these data or products.	A catalogued product might cover different datasets underneath assigned to different licences.
17.	Ability to update user profile to record acceptance of Terms and Conditions and/or licences.	
18.	Ability to grant access to data to unregistered users under certain circumstances as open data.	

2.2.3 System Requirements

As part of the proposal, the Tenderer is expected to present a component-based software architecture.

Interaction between different components shall be ruled by clearly stated Interface Control Documents (ICD).

Development of each component shall allow to be run independently.

Architectural components are expected to be implemented using containers (microservices).

Enumerated components describe a baseline of how ECMWF envisages the potential solution and serve as criteria to enumerate requirements. They are by no means an indication of the final ones but might serve as a guide for the tenderer to articulate its proposal.

Project milestones will demonstrate functional use cases which shall involve integration of different components.

Per each of the components the Tenderer is expected to define how this is expected to be developed, implemented and interface within the architecture.

Project deliverables will be benchmarked with the current CDS.

2.2.3.1 Web Portal

The web portal will provide a single entry-point for users to discover and interact with available content. This will allow users to browse and search the catalogue, manage data requests, perform computations on these data and display the outputs in the form of interactive applications. The portal will allow customized content to be displayed. The portal will also provide access to the Content Integration Manager (CIM) information and the Evaluation and Quality Control (EQC) (see Backend specifications, chapter 2.2.3.2).

Various functionalities must be enabled, these include:

- Access to content pages: documentation, on-line help, best practice & standards, events and news, doi and how-to-cite.
- Access to the products catalogue.
- Access to Evaluation and Quality Control (EQC) information.
- Access to the Content Integration Manager (CIM) information.
- Access to material provided by the Learning Virtual Environment (LVE).
- Access to the toolbox functionality.
- Invocation of tools on selected data and products as part of the request process.
- Monitoring of the status for data retrieval and computation requests by users.
- Visualisation of results data and products, as well as computations results, when applicable.
- User support components, help desk logging request, FAQs etc.
- KPIs and statistics.
- User account management.

Strong focus on the usability and user experience of the web portal and seamless integration between the different integrated functionalities is paramount.

In their response, the Tenderer is expected to provide references to previous work done on public web portals and experience on UX.

The web portal should conform to W3C web content accessibility (Conformance Level “Triple -A”).

The style will be provided by ECMWF, based on the EC Copernicus requirements.

The Web Portal will cover the following baseline functionality:

Reference	Requirement	Remarks
19.	User management facility with the ability to set up and register new users with different profiles (end-user, supplier, EQC member, web portal content providers, public user, etc.), modify user information and delete users.	User management should be compliant to the GDPR (https://gdpr.eu/compliance/)
20.	Users can self-register by providing a valid email address (OAuth based authentication can be considered) and administer their account and personal information including deletion of their accounts.	
21.	Users will be able to subscribe/unsubscribe to mailing lists.	
22.	Seamless integration and user experience across the portal content and functionalities (catalogue, toolbox editor, EQC, etc.)	This might require further interaction with third parties.

23.	Users can browse/search for available content (data, applications, documentation, other) without being logged in.	Content discoverability is a key feature for CopDS. Faceted, matching text, tags.
24.	Ability to display configurable/dynamic web alerts and messages for catalogued products.	Quality rating, labels, alerts, …
25.	Ability to display configurable/dynamic web alerts and messages at portal level.	Informative banners, advertisements, newsletters.
26.	Ability to define relations and interdependencies between catalogued products.	Tags, facets.
27.	Users can view the content and detailed information of catalogued products.	User friendly interface for metadata.
28.	Ability to administer quality information (EQC).	Final scope might require interaction with third parties.
29.	Ability to display quality information (EQC) linked to catalogued products.	Final scope might require interaction with third parties.
30.	Ability to administer Content Integration information (CIM).	Final scope might require interaction with third parties.
31.	Ability to display learning resources provided by the LVE linked to catalogued products.	Final scope and interfaces with LVE might require interaction with third parties.
32.	Users can download data and products through interactive forms, only when logged in.	Widgets and constraints. Special rules and options might be considered for public users.
33.	Ability for the product catalogue to hold references to the Terms and Conditions and/or Licences.	
34.	Ability to configure interactive forms based on a set of widgets and dynamic constraints.	Widget can be evolved and adapted. Harmonized across components.
35.	Users can invoke tools from the toolbox on selected data and products as extended capability for the download functionality, only when logged in.	
36.	Ability for users to tag content and to save, display and manage their searches, requests and preferences, only when logged in.	
37.	Ability to display interactive widgets to collect feedback from users.	Triggered by navigation events. Statistics and KPIs.

38.	Ability for the users to track and manage their requests: status, download, cancel requests, only when logged in.	
39.	Ability to display information about overall status of the system.	Queues, limits, statistics.
40.	Users can raise help desk request and access user support functionalities.	Final scope might require interaction with third parties. Integration with a Jira ticketing system
41.	Ability to administrate and customize the layout of the web portal interface and the navigation options.	Background, logos, menus, images, links.
42.	Ability for the data supplier to update, create, modify and delete metadata records as part of the integration process.	Final scope might require interaction with third parties.
43.	Ability to define, hold and display content defined on a standard markup language.	html, markdown.
44.	Ability to make portal and content visible and discoverable by search engines.	Google datasets.
45.	Ability to track and report website traffic and user journeys.	

Web portal load speed and responsiveness are key factors to be considered.

2.2.3.2 Backend

The backend component will consist of a relational data model supporting all the required information for feeding the web portal and other components. It will also be used for indexing the search engine. The backend will hold the following blocks of information:

- **User Services** maintains the user profile information that captures the details of each user, including their privileges and priorities regarding access to data/processing resources. Users will self-register and manage their profiles. *CopDS* administrators will be able to manage user profiles including user-specific rules regarding quality of service. Required API keys and management of accepted licences. It also contains information supporting dedicated user functionality available through the portal (tags, preferences, subscriptions, etc). The Tenderer shall describe in the Proposal the management of personal data and how this meets the requirements of Clause 2.8 and Annex 6 of the Volume V Framework Agreement.
- **Catalogue** holds the inventory of all content and products available through the portal. The Catalogue functions are supported by a dedicated search engine and index that provides fast faceted search capabilities. The attributes of each catalogued product are described in sufficient detail to support all user queries and to establish the applicability of tools and services. This includes detailing the capabilities of the data provider and the methods for accessing and manipulating the data. The catalogue might also contain extended metadata or any other additional information about data and products.

  Integration and deployment of catalogued content must support automatic updates from a source repository which is backed up and dedicated per operational instance of the system. This activity is expected to be carried out by the tenderer in close collaboration with ECMWF and third parties.

The backend will cover the following baseline functionality:

Reference	Requirement	Remarks
46.	Ability to consult and administer user information, preferences and settings	Describe in the Proposal the management of personal data and how this meets the requirements of Clause 2.8 and Annex 6 of the Volume V Framework Agreement.
47.	Ability to maintain the user profile information that captures the details of each user	User profile and information shall be aligned to fulfil EC reporting obligations (sector, country, …). This will be detailed at a later stage.
48.	Ability to manage information about data and products available in the CopDS, including support for Digital Object Identifier (DOI), with the view of creating a browsable/searchable catalogue (title, abstract, spatial and temporal extents, etc.).	
49.	Ability for the data and products to hold references to the Terms and Conditions and/or Licences.	
50.	Ability to support additional information about data and products available in the CopDS, in order to implement data retrieval and computations (physical location, data format, sub-setting information, etc.)	
51.	Ability to link product catalogue records and to define interrelations and dependencies between holdings in the backend.	
52.	Ability to hold interlinked entities with the Content Integration Manager (CIM) extended data model and functionality.	See chapter 2.2.3.9 for detailed information about CIM
53.	Ability to hold interlinked entities with the Evaluation and Quality Control (EQC) extended data model and functionality.	See chapter 2.2.3.10 for detailed information about EQC
54.	Ability to hold interlinked entities with the Virtual Learning Environment (VLE).	Subject to an Interface Control Document to be agreed with the LVE provider.
55.	Ability to manage details on the contents of the tools available in the toolbox, relating to the data and products.	
56.	Ability to manage the requests: monitor progress, cancel and download and/or view results. Past results should be available for a few days.	
57.	Ability to expose catalogue content as interoperable metadata services which are compliant with standards	ISO19115, INSPIRE, OGC
58.	Ability to manage information on the validity, data quality, and to make this available to the users.	Final definition might require further interaction with the EQC team.
59.	Ability to host users' list of past and current data retrieval and computation requests, with their status, and the ability to view/download their results.	

60.	Ability to log activities carried out on the product catalogue	
61.	Ability to keep a glossary of terms	Tagging system, thesaurus

2.2.3.3 Broker/Scheduler

The broker/scheduler component schedules and forwards requests from the web portal to the remote distributed repositories. These requests can be either data retrieval requests or computation requests. This component will also orchestrate more complex computations and guarantee Quality of Service (QoS) by implementing queues.

The broker will be a centralised system responsible for:

- Managing a queue of incoming data retrieval and computation requests.
- Perform sanity checks and scheduling requests according to limits, priorities and permissions (QoS).
- Dispatch requests to the appropriate data repository (or the compute layer) via a set of adaptors.
- Orchestrate workflows of requests, if required.
- Manage intermediate results.
- Provide status information to the users.
- Cancel users' requests if required.

The broker will implement quality of service (QoS) by queuing data retrieval and computation requests and scheduling their execution according to a series of rules, taking into account various parameters such as the user profile, the type of requests, the expected request cost (volume of data, CPU usage, etc.)

- QoS will also protect the system by implementing limits, for example:
    - Overall limits (e.g. maximum number of simultaneous requests).
    - Per user limits (e.g. maximum number of simultaneous requests per user).
    - Per destination limit (e.g. maximum number of simultaneous requests for a given data repository).
    - Per cost limit (e.g. maximum number of requests accessing more than a number of TBs).
- QoS will also protect the system by implementing permissions, e.g. users are not allowed to request more than a certain number of GBs of data in a single request.
- QoS will also implement priorities, for example:
    - Small requests over large requests.
    - Requests from the web portal over requests from the API.
    - Requests for VIP users.

Reference	**Link**
CDS QoS status	https://cds.climate.copernicus.eu/live/limits
CDS QoS grammar example	*user "Default per-user limit" (user ~ ".*") : 2;* *priority "Long jobs" (estimatedTime > hour(1)) : -hour(2);* *limit "The maximum number of request per dataset A is $value"* *(method ~ ".*satellite.*") : 6 ;*

It must be possible to configure the limits, priorities and permissions dynamically without interruption of service.

Please note that the lists of examples provided are not comprehensive. The Tenderer will work with ECMWF during the analysis phase to define a closer list of criteria to be addressed by the QoS.

The broker component is at the core of the system so it should be designed to support a heavy rate of transactions and escalate accordingly.

The broker will cover the following baseline functionality:

Reference	Requirement	Remarks
62.	Implement QoS based around the criteria described above.	Limits, priorities, permissions.
63.	Ability to configure the QoS rules dynamically without interruption of service.	Automatic configuration align with system workload status.
64.	Ability to forward requests to the relevant services (distributed data repositories, compute layer…).	
65.	Ability to schedule complex workflows composed of a combination of several requests.	
66.	Ability to maintain status information for all the requests being processed (e.g. queued, active, complete, aborted, …)	Offline and online processing of requests.
67.	Ability to report meaningful information in case of request failures.	Log management.
68.	Ability to be invoked from both the Web Portal (user interface) and programmatically as part of the Application Programming Interface (API).	
69.	Ability to be escalated and reconfigured according to workload requirements without interruption of service.	QoS, runners, compute nodes.
70.	Ability to support asynchronous requests.	
71.	Ability to scale by adding more broker instances.	

The Tenderer shall have proven experience in QoS development and large operational systems implementation.

2.2.3.4 Compute

The compute layer is a combination of facilities at the data provider location, when available, and a dedicated set of internal servers to run computations when data providers cannot support them on their infrastructure.

Computations are limited to those provided by the system as tools or dedicated services. They can be performed at the data repositories, at a centralised location, or in a cloud environment.

For the dedicated set of servers, the Tenderer is invited to discuss with ECMWF the best technical solution to maximize the capabilities of cloud computing.

The compute layer will generate temporary results of the computation and data retrievals. Once the results have been downloaded by the users, they must be deleted from this layer.

A dedicated cache area will hold the results of computation ready to be downloaded by the user.

When data retrieval and/or compute requests generate results from one data supplier, it should be possible for the results to be downloaded directly by the user.

The compute layer will cover the following baseline functionality:

Reference	Requirement	Remarks
72.	Ability to perform computation on a set of dedicated servers.	Number of servers can be scaled up to accommodate load.
73.	Ability to store temporary results.	
74.	Ability for the system to remove temporary results once the user has downloaded them or after a given time period.	Cache area and management.
75.	Ability to transfer directly the results from the data location to the end-user when applicable.	

2.2.3.5 Adaptors

Adaptor components will guarantee interoperability between the broker and the various kinds of distributed data repositories, possibly translating data retrieval, or computation requests. The content of the Data Stores varies in size (from megabytes to petabytes), type (flat files to relational databases) and formats. Adaptors will evolve over time so they should work as services.

A set of core adaptors supporting access to existing data sources are expected to be delivered as part of the project.

Each supplier will be expected to provide access to its data and products and computation facilities according to some agreed standards, such as:

Standard	Remark
HTTP/ HTTPS	Currently implemented on Data Store.
OGC's WMS, WCS, WFS, WPS	Currently implemented on Data Store.
OpenDAP	
THREDDS	
SFTP - not in standards	

The adaptors will cover the following baseline functionality:

Reference	Requirement	Remarks
76.	Ability to combine multiple adaptors for a single catalogued product.	
77.	Ability to register new adaptors as services and safely deploy updates.	
78.	Ability to configure different instances of the adaptor to accommodate requirements from different data providers.	

2.2.3.6 API

There must be a web-based API that will allow users to build programs to interact with the *CopDS*. The API will allow remote access to the *CopDS* functionality, such as data and product retrieval and invocation of the Toolbox.

The API should:

- Be RESTful.
- Be language agnostic (users can write clients in any programming language).
- Support asynchronous requests.

Registered users will receive an authentication token (e.g. an API key), that will be used to authenticate their requests.

More detailed information about the current CDS API can be found in the following links:

Reference	Link
CDS API in Swagger	https://cds.climate.copernicus.eu/modules/custom/cds_apikeys/app/apidocs.html
CDS API in Github	https://github.com/ecmwf/cdsapi

The API will cover the following baseline functionality:

Reference	Requirement	Remarks
79.	Ability to query the catalogue of data and products via the API.	
80.	Ability to query the content of the toolbox via the API.	
81.	Ability to perform data retrieval request and computation requests via the API.	Allow cancellation.
82.	Ability to download results via the API.	Download checks (checksum or similar).
83.	Ability to query the status of the broker.	Queues status, limits.
84.	Ability to customize informative messages return to users via API.	Alerts, error messages.

2.2.3.7 Testing/Monitoring/Metrics

The proposed infrastructure should have the ability to:

- Log all system activities, for diagnostics and trouble shooting, to be used by analysts and as support of capacity planning (ECMWF currently uses Splunk for log indexing and analysis).
- Monitor and report system issues to the operators. This system will need to integrate a set of checks into a monitoring tool, showing the health of the system (ECMWF currently uses Opsview).
- Measure system activity in terms of number of users, data downloads, user satisfaction and to be able to feed this into ECMWF Key Performance Indicators (KPI).
- Generate statistics about the use and access to the system on a pre-defined list of 'user characteristics' to address EC reporting obligations as stated in the Contribution Agreement and to allow informing decision making.

This information will be used when reporting to the EC, and to enable continuous improvements in the service (capacity planning).

Monitoring outputs will be suitable for further usage as inputs for user-oriented functionalities and dynamic configuration (dynamic QoS, rankings, usage statistics, others).

This component will cover the following baseline functionality:

Reference	Requirement	Remarks
85.	Ability to create health checks of the system components to the Operators.	To be integrated with Opsview.
86.	Ability to generate system and application logs.	To be integrated with Splunk.
87.	Ability to conduct audits.	
88.	Ability to manage faults and record errors.	Self-explanatory and human readable messages.
89.	Ability to minimize downtime during upgrades.	
90.	Ability to provide system metrics based on KPIs.	
91.	Ability to get statistics and include them on web portal capabilities.	Rankings, preferred products, user activity, system live status report.
92.	The system should have the ability to monitor all activities.	Data downloads, system down time, file size, transmission speeds.
93.	Ability to use monitoring and statistics as input for further functional capabilities.	Dynamic configuration, user-oriented functionalities, others.
94.	Implements unit and functional tests on delivered software components.	

2.2.3.8 Toolbox

The *CopDS* Toolbox is a suite of scientific software tools for interacting with data from the *CopDS* catalogue. It provides users with a single software package for retrieving, analysing and visualising Copernicus datasets, and presenting their results as interactive web applications.

Applications developed in the toolbox are made publicly available to end users, providing graphical interfaces for interacting with and parametrising toolbox scripts. High quality applications are one of the main mechanisms through which *CopDS* data is accessed and explored by a wide variety of users. These applications range from simple 'data viewers' which purely visualise a raw dataset, to highly detailed 'data explorers' providing interactive maps and complex statistics driven by user input.

The *CopDS* toolbox implementation will support two different functioning modes:

- **Integrated toolbox**: an 'online' development environment seamlessly integrated within the *CopDS* web portal, providing fast and efficient access to catalogued data by taking full advantage of available computation resources and functionalities provided by the *CopDS* infrastructure.
- **Segregated toolbox**: a standalone version of the toolbox which allows users to install and run toolbox software locally, with limited access to certain *CopDS* infrastructure capabilities.

Tenderer is required to provide information about how consistency and integrity between the two functioning modes will be achieved.

The foundation of the toolbox implementation will be a suite of quality-assured, open source Python libraries for performing scientific analysis and visualisation, ensuring compatibility with a broader range of Python tools already familiar to the scientific community. Whenever possible, data retrieval and processing will be executed within *CopDS* infrastructure to allow users to work with very large datasets without having to download them locally.

The main objectives of the modernized toolbox are:

- Develop a suite of scientific Python tools for interacting with the *CopDS*, with a focus on:
    - **Retrieval tools** for accessing the wide range of datasets available in the *CopDS* catalogue.
    - **Analysis tools** for performing scientific analysis and processing of large geo-temporal datasets.
    - **Visualisation tools** for producing high quality interactive data visualisations such as statistical graphs and geospatial maps.
    - Tools for producing highly configurable dashboard-like web applications which present the output of toolbox scripts as user-friendly and interactive user interfaces.
- Maintain high levels of quality assurance and documentation, with open source Python software hosted through a version-controlled repository hosting service (e.g. GitHub) to encourage and manage peer-reviewed contributions.
- Maximise underlying cloud infrastructure resources to deliver high-performance processing of very large geo-temporal datasets.
- Provide software and support for enforcing data compliance with a Common Data Model (CDM), ensuring compatibility between catalogued datasets and available tools.
- Accessible documentation and training material to help users to understand and use available tools and functionalities.

The toolbox will cover the following baseline functionality:

Reference	Requirement	Remarks
95.	Ability to support integrated and segregated implementations of the toolbox packages.	
96.	Ability to run tools as containerized microservices	
97.	Ability for tools to produce non-scientific outputs which can be leveraged in the development of applications.	
98.	Ability to develop python notebooks.	Development environment seamlessly integrated within the web portal
99.	Ability to invoke retrieval, computation and visualization tools via an API interface	Outputs from API calls shall maximize compatibility with scientific software and packages
100.	Ability for converting catalogued data into a common data format which maximises compatibility with tools via a Common Data Model (CDM).	
101.	Ability to inject interactive input widgets tools to be leveraged in web applications.	Dropdown boxes, buttons, date/time selections, text inputs.

102.	Ability to create dashboard-like layouts as final outputs for users without being exposed to the underlying Python code.	
103.	Ability for multiple users to work together on the same workflow	Share workflow with other users
104.	Ability to produce and include WMS layers on geospatial maps.	Geoserver. INSPIRE compliance.
105.	Ability to integrate with training notebooks produced by the Learning platform.	

Reference	**Link**
CDS Applications with input widget layout	https://cds.climate.copernicus.eu/cdsapp#!/software/app-c3s-daily-era5-statistics?tab=app
CDS Applications with dashboard-like layout	https://cds.climate.copernicus.eu/cdsapp#!/software/app-era5-explorer?tab=app

As for the development and implementation of the toolbox, the following technologies are proposed:

Purpose	**Technologies**
Editor	Jupyter notebooks (Python environment with possibility of further implementation of R)
Python packages	- xarray - numpy - netCDF4 - Matplotlib - Plotly - cartopy - Scipy - CliMetLab - rpy2
Additional software support via CDS-Toolbox python package(s)	- CDO
Web-mapping tools	- Geoserver - Leaflet - OpenStreetMap
Supporting tools	- GitHub - ReadTheDocs

The Tenderer is requested to provide a proposal for a modernized toolbox that will embrace open-source scientific software and will ensure compatibility with state-of-the-art solutions such as machine learning, data cubes and interactive notebooks.

2.2.3.9 Content Integration Manager (CIM)

Products aimed to be accessible through the *CopDS* catalogue will follow an iterative integration process consisting of different steps and involving different actors (Provider, Contract Management, Technical Officers, Document reviewers, the CopDS team, User Support, EQC etc). The Content Integration Manager (CIM) component will be the entry point for the integration of new datasets and applications into the system facilitating the collection and management of deliverables provided by third parties. Products are provided together with (high-level) discovery metadata, full user documentation, independent quality assessment, outreach material, licenses and references. Provided material will serve as bases for the final content exposed to users, as well as for internal contractual reporting. Roughly, the metadata and deliveries associated with a dataset splits into the following categories, given with some examples:

- Contractual metadata: delivery date. delivery type (dataset, document, code).
- Technical (data) metadata: data record ID; main variable(s); related variable(s); temporal/spatial resolution(s).
- Descriptive metadata: abstract text; abstract Image/Application; applicable license; producing party; main variable(s); related variable(s); associated user documentation.
- User documentation: document URL; document version; associated data record(s)/delivery(ies).
- Service messages: User facing notices of (expected) product updates/extension. Expected/Reported degradation; Any delay, cancelation, etc of provision.
- Other deliverables.

The CIM component will function as *an entry point and central repository for metadata and documents* associated with datasets and applications acquired by ECMWF for provision through the *CopDS*. Principal aim of this component is to facilitate information gathering and sharing between different actors (Data Provider, Contract Management, Technical Officers, Document reviewers, the CopDS team, User Support, EQC etc) throughout the datasets life cycle on the *CopDS*.

The main functional objectives of CIM can be summarized as follows:

- Serve as single entry point for contractual deliverables concerning datasets aimed to be published on the *CopDS* catalogue.
- Streamline the acquisition, integration and publication process of datasets in the CopDS catalogue facilitating the collaborative interaction between different actors (add, edit, maintain, view).
- Keep track of contractual deliveries and deliverables all along the acquisition, integration and publication process: status (expected/delivered/approved; priority for integration; associated values.
- Define input templates for collecting and presenting information about product definition, quality and usability (referred as Quality Assurance Templates QATs).
- Define relationships and interdependencies between various (groups of) deliveries/deliverables.
- Provide version control for submitted metadata and documentation along with different tiers of access rights, functions and tasks depending on role.
- Provide the tools and infrastructure for aiding harmonisation and cataloguing on the level of data record metadata.
- Allow flexible extraction of content to populate other CopDS components.
- Maximise machine operability in the acquisition/ingestion workflows (automated checks for data presence/integrity at delivery; harvest metadata,…).

Reference	Requirement	Remarks
106.	Ability to define roles, permissions.	Based on the core CopDS Users Services component.

107.	Ability for actors to manage content based on assigned permissions.	Add, Edit, Read, Delete.
108.	Ability to keep track of submitted content.	Status, versions, changes.
109.	Ability to define workflows with assigned tasks to different roles.	
110.	Ability to define relations and interdependencies between submitted content.	Groups of deliverables.
111.	Ability to manage content through an easy-to-use interface.	
112.	Ability to interoperate content with other CopDS components.	Web Portal, Catalogue, EQC.
113.	Ability to define different QATs for collection and presentation of products definition, quality and usability.	Multiple types of QATs can be defined for different categories of data products.
114.	Extracts from QATs can be displayed on the web as part of the data product information.	QATs are the baseline for EQC QARs once filled and published.
115.	Ability to define shared vocabularies and common practices to ensure consistency of QATs.	International standards and best practices related to quality control and/or quality assessment.

2.2.3.10 Evaluation and Quality Control (EQC)

The Evaluation and Quality Control (EQC) function will offer a homogeneous user-led service for the quality assessment of datasets, applications and other facilities available to manipulate them such as tools and standards. The *CopDS* EQC component will provide the technical support for an established and continued EQC function seamless integrated with the rest of *CopDS* infrastructure components.

Reference	**Link**
Quality assurance (EQC) tab for ERA5	https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=eqc

CopDS EQC will be built on top of content accessible via *CopDS* CIM function and will ease the collaborative development and content management of Quality Assurance Reports (QARs) built on top of the underlaying QATs. The resulting QARs will be stored, documented and published as part of the *CopDS EQC* functionality.

Quality Assurance Reports (QARs) will provide users with comprehensive information about the technical and scientific quality of the datasets registered via CIM to be subsequently published in the CopDS catalogue. The different sections of the QARs will be seamlessly made accessible to the users in the *CopDS* web portal through a synthesis table. The synthesis table is devised as a functionality to organize and homogenize the EQC QARs information to be linked and published as extension of catalogued products.

The EQC framework is a multi-actor process that requires collaborative iterations with different stakeholders (data provider, EQC evaluator and reviewer, C3S approver and observer) each of them with different roles and responsibilities during the assessing information cycle. CopDS EQC function will facilitate this iterative evaluation process allowing scores and traced comments between actors to be added to sections of the QAR.

The CopDS web portal interface will allow that QARs developed within the EQC function can be made available to users alongside with the descriptive information of the corresponding dataset product.

The EQC assessing information cycle to be supported by the EQC function is oriented (driven) to (by):

- A modular, flexible and scalable data model able to support new data/information sources and new actors involved.
- Automatization as much as possible of the information management flows in order to reduce human errors and make the system sustainable in the long-term.
- An iterative and reproducible approach suitable for continuous improvement.
- A user-friendly presentation of the EQC quality information, clustered to facilitate its consultation.
- Homogeneous and consistent support for the *CopDS* data quality information, recognizing the existence of inherent differences across the dataset categories.
- Transparency and traceability of the quality assessments flow.
- FAIR (Wilkinson et al. 2016) and TRUST principles (Lin et al. 2020) and ISO 19157:2013.
- Service management practices to perform the EQC activities according to the best practices in an operational environment.

Reference	Requirement	Remarks
116.	Ability to define different user roles and permissions to perform different functions.	Based on the core *CopDS* Users Services component.
117.	Ability to create Quality Assurance Reports (QARs) and manage its content in collaborative between different actors.	
118.	Ability to store, document and publish QARs.	
119.	Ability to add scores and traced comments between actors to sections of the QAR.	Iterative process.

Content Integration Manager (CIM) and Evaluation and Quality control (EQC) extended data models and functions will require of further interaction and definition of requirements based on existing prototypes and involving third parties during the analysis phase.

Tenderers are expected to propose a flexible workplan methodology which will allow to address this work.

2.2.3.11 Observations Repository

CopDS will host centrally managed non-gridded observation repositories that will facilitate the collection of and access to these data within the system. These repositories will host data delivered by different providers as from the COP1 C3S_311a, C3C_311c and the forthcoming COP2 C3S2_311 contracts The final design of the technical solutions for such repositories will have to be defined in close collaboration with ECMWF experts and the external data provider Goal is to provide a generic solution that can be used by the external observation providers in a standardized way to manage the backend of their datasets.

All non-gridded observations in the CopDS follow a common data model for observations (CDM-OBS) that standardizes their logical organization, units of geophysical quantities, their naming and the way in which additional information (metadata) is stored. This model can be implemented in a number of data formats, the following of which will be used:

- NetCDF.
- CSV.
- and potentially the ODB2 format that is used at ECMWF to store observations.

Repositories shall be flexible to ingest timely updates in an efficient manner.

ECMWF will provide more detailed information during the Analysis phase.

The technical solution shall cater for an adequate way to serve the observations to *CopDS* users in an efficient manner. Different users have different requirements. As a minimum the *CopDS* shall be able to handle the delivery of the following two use cases:

- *1)* A user that requires all observations available in a particular *CopDS* catalogue entry for one particular date.
- *2)* A user that is interested in observations in a small area, or from a particular station for the entire length of the available data records.

In addition, the bidder shall implement observation-specific generic functionalities in the *CopDS*. These include:

- Instant estimates of data volumes that are contained in particular form requests before the data is accessed.
- Feasibility to search observations on station-identifier level.

Functionality of tailor-made user queries in the *CopDS* API that allow for sophisticated data requests.

The observation repository will cover the following baseline functionality:

Reference	Requirement	Remarks
120.	Ability to manage versions.	
121.	Ability to automatize the collection and integration of data deliveries.	Flexible to ingest timely updates in an efficient manner.
122.	Define a harmonized structure suitable to accommodate data from different providers sources.	
123.	Ability for the hosted data to hold references to the Terms and Conditions and/or Licences.	Versions.
124.	Ability to back up the data.	
125.	Ability to define the interactive forms and constraints presented in the web portal.	
126.	Provision of generic solutions that allow access to observations in an efficient manner for a range of use cases.	Tailor-made user queries.
127.	Implementation of observation-specific functionalities in the *CopDS.*	

Reference	Link
GRUAN, an example of an existing non-gridded observational dataset	https://cds.climate.copernicus.eu/cdsapp#!/dataset/insitu-observations-gruan-reference-network

Long time series of compressed csv files of observations shall rarely exceed a few TB. For larger datasets to be hosted further considerations shall be considered and discussed directly with ECMWF.

3 Contract Management

3.1 Contract Approach

As stated in previous chapters the final objective of this contract is to modernize and further improve operational services which nowadays are performing at high quality standards and serving a large community of users. In consequence the approach proposed for this contract aims to assure the following:

- ECMWF vision and architectural principles are aligned with those of the successful tender.
- Knowhow and experience from the current system are taken onboard by the new system in the form of functional requirements and technical specifications.
- Requirements and expectations of main stakeholders are properly addressed.
- Functional integration and interfaces with interlinked platforms and projects are properly defined.
- Existing code and components are considered for being reutilized or reengineered.
- Project roadmap is optimized and aligned with main milestones of Copernicus services.
- Current content is fully migrated at minor impact and risk.
- Transition between systems is done in a smooth and seamless way.

In order to achieve the above-mentioned objectives an approach to the contract by phases is being proposed.

- **Analysis phase:** During this phase the Tenderer will interact with ECMWF and other parties to define the system scope and architecture, the technologies to be used and a detailed development and implementation strategy and roadmap. The Tenderer will also be requested during this phase to propose and execute a plan for knowledge transfer and handover of the current infrastructure with the former contractor as to take over the operational maintenance of the current CDS infrastructure until this is fully replaced by the operational implementation of the modernized infrastructure.
- **Transitional maintenance:** Upon completion of the handover of the CDS infrastructure from the previous contractor, the Tenderer will take over the operational maintenance of the CDS infrastructure. This phase will terminate once operations are fully overtaken by the new infrastructure.
- **Development and implementation phase:** After agreement and approval of deliverables from the Analysis phase, the contract will trigger a development phase following the agreed terms and roadmap. Based on status and outputs from discussions in the analysis phase, it could be decided to trigger the development split by blocks. A priori the following development blocks are envisaged:
    - **Core components:** Those supporting core use case functionalities such as discovery, transformation and retrieval of data.
    - **Extended components:** Those supporting extended functionality with dependencies from third party inputs (EQC and CIM)
    - **Toolbox:** Those supporting development use case functionalities which allow to perform operations on top of available data, from simple transformations to complex web-based applications.
    - **Observation repository:** Centrally managed repository for observations data which allow for a close and efficient integration with the rest of *CopDS* functionalities.
- **Content migration:** As described on stated requirements current content of the CDS and ADS catalogues, datasets and applications, has to be migrated into the new infrastructure as part of the work covered by this tender. Migrated content is expected to perform equivalent or improved functionality. This activity is expected to be carried out by the tenderer in close collaboration with ECMWF and designated third parties.

Overall work and activities expected to be covered under each of these phases will be contractually addressed by different Work Packages as described in chapter 2.2 of this document.

Work at all phases and stages in the project will be done in close collaboration with ECMWF. The Development phase will follow an agile methodology or equivalent, implementing a CI/CD (Continuous Integration and Continuous Development) approach. The selected contractor will be required to discuss and iterate with ECMWF and third parties throughout the whole development process.

ECMWF proposes the Agile methodology reported in Annex 4 of the Framework Agreement template included in Volume V of this ITT, but welcomes suggestions from the Tenderer on what methodology they propose to apply for the different phases of the project based on their knowledge and experience with projects of a similar nature. The Tenderer must provide examples of how they have applied this approach in similar projects they have previously worked on.

The methodologies proposed by the Tenderer must ensure that final deliverables are fit for purpose, aligned with the project vision and remain within project cost and schedule.

ECMWF intends to award a single framework agreement (for a maximum duration of 48 months including the maintenance) which shall be implemented with two Service Contracts, the first one to cover maximum the initial 6 months (Analysis and Handover phase).

3.2 Work Packages

3.2.1 Transitional Maintenance

As part of the work tendered, the contractor is expected to take over the maintenance of the current CDS infrastructure, including CDS/ADS and Toolbox (See chapter 1.3), the current EQC CMS and the prototype for the Content Integration Manager (CIM). This activity is envisaged to be addressed in two different stages:

- **Handover:** this is expected to be carried out as part of the preliminary analysis in a series of interactions between current and former contractors supervised and facilitated by ECMWF. During this stage, the contractor is expected to collect and review all the existing documentation and run all the required sessions as to become confident to take over the maintenance task. This phase is planned to have a duration up to 6 months.
- **Maintenance:** In a second stage the contractor will take full responsibility for the operational maintenance of the current CDS Infrastructure until operation is covered in full by the new infrastructure.

The scope of this work package covers the following objectives:

- Corrective maintenance and support to guarantee operational continuity.
- Generate incident reports and root cause diagnosis.
- Perform preventive maintenance actions to avoid failures or service degradation.
- Support integration and software updates.
- Advice on the reusability of existing components.
- Facilitate the transition and migration to the new infrastructure.

3.2.2 Preliminary Analysis

This work package aims to carry out a detailed functional and technical analysis of the CopDS system to be implemented prior to the development phase. This phase is expected to be run jointly between ECMWF staff and the contractor team in a series of iterations to be agreed. For certain components, interaction with third parties will be required.

The final objectives of this phase are the following:

- Define project goals and drivers.
- Define the architecture principles and functional scope of the final system.
- Produce user stories/use cases, to define structure and functionality of the proposed solution.
- Perform a gap analysis between the current and final systems.
- Define the software packages and technologies to be implemented.
- Define the development strategy and implementation roadmap.
- Define the ICDs that will drive the integration between infrastructure components.
- Define the work methodology and set up the required tools.
- Identify key stakeholders, roles and responsibilities.
- Review synergies and interdependencies with existing projects.
- Define communication flows and channels.
- Set up the working tools and environments.
- Define roadmap and core objectives for the next project milestones.
- Collect and review existing documentation.

Dedicated objectives for the transitional maintenance of the current CDS infrastructure:

- Knowledge transfer and handover of the current system architecture.
- Run the required handover sessions jointly with the former CDS contractor.

All final deliverables as result of this phase must be approved by ECMWF as conditional requirement before triggering the development and transitional maintenance phases.

During this phase it is envisaged a close and iterative collaboration between ECMWF, the contractor of this tender and designated third parties through a series of events as described in the following table:

Scope	Participants	Duration	Objectives	Location
Initial workshop	ECMWF, Contractor	To be agreed	• Introduce the project teams. • Define the requirements and expectations of the project. • Perform a preliminary exchange of knowledge and ideas. • Define a first draft of architectural components (WPs) • Plan further actions.	ECMWF premises (Virtual in case restrictions persist)

Work meetings	ECMWF, contractor, third parties	To be agreed	• Follow up the business and technical analysis. • Plan further actions	Flexible
Regular follow-up meetings	ECMWF, contractor, third parties	To be agreed (short)	Short regular interactions to catch-up about status of deliverables, progress of work, solve questions and plan further actions.	Virtual
Presentation of results	ECMWF, Contractor	To be agreed	• Present the outputs of the work as described in the objectives described above. • Plan further action	ECMWF premises

Final scope and agenda for mentioned events shall be agreed at a later stage during the contract negotiation.

Expected deliverables for this phase are:

Nature	**Title**
Report	Business Analysis & System Architecture
Report	UX analysis and mock-up of the web portal
Report	Interface Control Documents (ICD)
Report	Work plan and implementation roadmap
Report	Financial report

Reference material for this phase will be:

- Solution proposed by the contractor as response of this tender.
- Current Climate Data Store infrastructure and available documentation.
- Inputs and requirements collected from key stakeholders.

3.2.3 Development and Implementation

The final approach and scope of this work package will be driven by the outputs and agreements produced by the preliminary analysis work package.

The scope of work under this package covers the development and implementation of a working system covering requirements as described in chapter 2.2 of this document.

The final objectives of this work package are the following:

- Implement an iterative/continuous improvement development approach.
- Conduct continuous iteration and requirements analysis with key stakeholders.
- Produce a prototype to demonstrate structure and functionality (proof of concept).
- Produce the components of the pre-operational phase.
- Conduct component, unit, system, and integration testing.
- Deliver the system and perform acceptance tests at ECMWF.
- Deliver all required documentation.

Internal organization of this work package can be considered to be split in different sub-packages if this serves the project to shorten the roadmap or make a more efficient use of resources.

Given the developmental nature of this phase, ECMWF anticipates that the adoption of a CI/CD (Continuous Integration and Continuous Development) approach will be beneficial to ensure its successful delivery. This would typically involve a design framework comprising of:

- Requirements gathering.
- Prototyping.
- Visual designs.
- Functionality.

Such a project delivery framework will enable effective development for the project in a "time-box" approach.

The development strategy and implementation roadmap will define a detailed schedule of the activities to be carried out within the contract with two main milestones: "proof of concept" and "pre-operational" phases (see table below), including proposed reviews, milestones, and validation and verification points.

It is expected that "proof of concept" and "pre-operational" phases will provide access to a working piece of the infrastructure which combines deliverables and functionalities from different components to fulfil a predefined set of functional use cases (search, discover, download) to be agreed with ECMWF per each phase.

The Kanban approach is strongly recommended but ECMWF welcomes suggestions from the Tenderer on what agile methodology they propose to apply based on their knowledge and experience with software projects of a similar nature.

The Tenderer must provide examples of how they have applied this approach in similar contracts they have previously worked on. The development methodology proposed by the Tenderer must ensure that the final delivery is fit for purpose and remains within the contract cost and schedule.

The selected contractor will be required to discuss and iterate on the design/architecture with ECMWF throughout the lifetime of the development process. Due to the dynamic nature of the Copernicus program and Services incoming requirements are expected during the project.

3.2.4 Content Migration

This work package is aimed to address the migration of all catalogued content on the current CDS infrastructure to be operationally hosted, managed and run by the modernized components of the *CopDS* infrastructure. The starting point of this work package along the contract roadmap shall be kept open to be triggered under mutual agreement between ECMWF and the Tenderer depending on the status of discussions and developments.

During the Analysis phase the Tenderer is expected to get a detailed view about the current set up of the CDS content in order to define a more accurate migration strategy aligned with the activities and roadmap of other work packages under this contract.

As for the execution of this work package the Tenderer is expected to work in close collaboration with ECMWF and designated third parties.

Content currently available through the catalogue covers:

- Datasets: descriptive metadata, download forms and constraints, adaptors, licences, data viewers, QoS.
- Applications: workflows, Python source code, web dashboards.

Reference	Link
Datasets	https://cds.climate.copernicus.eu/cdsapp#!/search?type=dataset
Applications	https://cds.climate.copernicus.eu/cdsapp#!/search?type=application

3.3 Project Team

3.3.1 Tenderer

The Tenderer shall demonstrate the availability of expertise as required for the implementation of the services in line with the components and requirements stated on this document.

The Tenderer shall demonstrate for itself, and for any proposed subcontractors that they have participated in national or international research and/or private sector software development projects in the last 5 years for the activities for which this Tender is proposed. ECMWF may ask for evidence of performance in the form of certificates issued or countersigned by the competent authority.

The Tenderer will appoint a Project Manager, responsible for the delivery of the system, to oversee the progress of this project.

The Tenderer must outline the project team. The outline must contain the following:

- Relevant experience of key staff and management personnel.
- Names of project manager, and main technical contact and number of work hours dedicated to the project team for the duration of the project.
- The Tenderer must state if there are any sub-contractors, and define their roles.
- An indication of how many staff will be part of the project team and at what level.

The project team assigned to this contract is expected to:

- Have solid knowledge in:
  - Web technologies and UX.
  - Service Oriented Architectures (SOA).
  - Open source.
  - Containerization and orchestration technologies.
  - Geospatial data handling within a Linux and Python environment.
  - Advanced Python.
  - Data structures and algorithms.
  - NetCDF and GRIB formats.
  - OGC standards (CSW, WMS).
  - Cloud technologies.
  - UML and Business architectures.
- Demonstrated experience on running large scale systems … preferably related with the domain (Meteorological, Climate, …)
- Have an adequate understanding of the current CDS Infrastructure and the CDS Toolbox environments.

The Tenderer project team is expected to work very closely with the ECMWF team for the duration of the project.

3.3.2 ECMWF

ECMWF will appoint a Product Owner and a technical lead per each component to oversee the development and deliverables. The Product Owner will be the point of contact for the Tenderer. The Product Owner will:

- Monitor the Tenderer work execution.
- Review Tenderer's specifications and architectures to ensure that they are "fit for purpose".
- Be the focal point to provide the Tenderer with the ECMWF inputs required at each stage.
- Validate and prioritise the requirements list.
- Agree time boxing priorities.
- Be the focal point to support the incremental testing of each iterative phase.
- Sign off key milestones and deliverables.

The ECMWF team will attend project meetings as deemed necessary for the monitoring of the Tenderer's activities, and will be granted unrestricted access by prior agreement to the Tenderer's facilities where the work is being carried out.

3.4 Deliverables

The expected top-level deliverables are outlined in section 3.2. These can be in the form of documents or reports, data sets or databases, services, and user support. Requirements for each type are described in the following subsections.

3.4.1 Documents and Reports

All project reports and documentation for this ITT shall be produced in English. The quality of reports and deliverables shall be equivalent to the standard of peer-reviewed publications and practice. Unless otherwise specified in the specific contract, deliverables shall be made available to ECMWF in electronic format (PDF/Microsoft Word/Microsoft Excel or HTML) via the Copernicus Deliverables Repository portal. the details will be agreed at the negotiation stage.

A high-level project management plan must be delivered as part of this ITT.

The following documents shall be delivered as part of the contract:

- An updated project management plan including milestones.
- Progress reports.
- Work package documentation.
    - Requirements specification.
    - Design, development, test plan, test case tests, test reports and test scripts.
- Detailed description of the test cases used for the internal validation of the software including test results.
    - Functional tests.
    - Performance tests.
    - Availability tests.

- Interface descriptions of all modules.
- Software quality assurance plan.
- Risk register.

- Sign offs.
- Training plans.
- Documentation (system, software, source code).
- Release notes.
- User guide, data supplier guide, administration guide, installation guide.

3.4.2 Data and IPR

It is a condition of EU funding for C3S/CAMS that ownership of any datasets/software developed with C3S/CAMS funding passes from the suppliers to the European Union via ECMWF. Ownership will pass from the date of creation of the datasets/software. Suppliers will be granted a non-exclusive licence to use the datasets/software which they have provided to C3S/CAMS for any purpose.

All software and products used by the successful Tenderer to produce the C3S/CAMS datasets/software will remain the property of the successful Tenderer, except for those components which are acquired or created specifically for C3S/CAMS purposes, with C3S/CAMS funding, and which are separable and useable in isolation from the rest of the successful Tenderers' production system. The identity and ownership of such exceptional components will be passed to the European Union annually. The successful Tenderer will be granted a non-exclusive licence to use them for any purpose.

3.5 Tasks to be performed

The successful Tenderer is required to:

- Implement the work packages described in this ITT.
- Deliver related WP deliveries following the CDS processes and requirements.
- Deliver all required documentation (in English).
- Carry out training of ECMWF personnel if required (transfer of knowledge).

3.6 Key Performance Indicators

As part of the bid, the Tenderer shall specify a proposed set of Key Performance Indicators (KPIs) appropriate for the service. The KPIs shall be designed to quantify different aspects of quality of service against the requirements described in this document. These initial specifications shall be refined together with ECMWF during the negotiation of the contract. Contractors shall report to ECMWF on a set of SMART (specific, measurable, actionable, realistic and time bound) KPIs suitable for monitoring various aspect of service performance, including (but not limited to):

- Code quality (performance, output etc.)
- Service delivery
- Contract management
- User support

The KPIs will be reported in the Quarterly and Annual reports. At the end of each year, a service readiness review shall take place that will include assessment of performance against the set of KPIs.

4 Tender Format and Content

General guidelines for the Tender are described in Volume IIIB. Specific requirements for this particular ITT are described in the next sub-sections.

4.1 Page Limits

As a guideline, it is expected that individual sections of the Tenderer's response do not exceed the page limits listed below. These are advisory limits and should be followed wherever possible, to avoid excessive or wordy responses.

Table 1 Page limits per section

Section	*Page Limit*
Executive Summary	2
Track Record	2 (for general) and 2 per entity
Quality of Resources Applied	2 (excl. Table 1 in Volume IIIB and CV's with a maximum length of 2 pages each)
Technical Solution Proposed	30 pages in total for the technical solution and Work Packages (Table 2 in Volume IIIB, the section on reference, publications, patents and any pre-existing IPR are excluded from the page limit and have no page limit)
Management and Implementation	10 (excl. Table 4 and Table 5 in Volume IIIB) + 2 per each Work package template (Table 3 in Volume IIIB)
Pricing table	No limitation

4.2 Specific additional instructions for the Tenderer's response

The following is a guide to the minimum content expected to be included in each section, additional to the content described in the general guidelines of Volume IIIB. This is not an exhaustive description and additional information may be necessary depending on the Tenderer's response.

4.2.1 Executive Summary

The Tenderer shall provide an executive summary of the proposal, describing the objectives, team and service level.

4.2.2 Track Record

The Tenderer shall demonstrate the availability of expertise as required for the implementation of the services in line with the work package descriptions.

The Tenderer shall demonstrate for itself, and for any proposed subcontractors that they have participated in national or international research and private sector software development projects in the last 5 years for the activities for which this Tender is proposed. ECMWF may ask for evidence of performance in the form of certificates issued or countersigned by the competent authority. The Tenderer shall in particular demonstrate their experience in:

- Kubernetes
- UX, Front-end development
- Python
- Open source
- Data Base administration
- Geospatial standards (OGC, ISO, INSPIRE)

- AGILE development methodology
- Implementation of large operational systems

4.2.3 Quality of Resources applied

The Tenderer shall propose a team that meets at least the following requirements:

- A Service Manager with more than 5 years of experience in managing activities related to an ITT of this size, with experience in the appropriate delivery methodology proposed in Section 3.
- A technical project team with more than 5 years of experience on performing activities related to the various aspects of this ITT.

The CVs, proven track record and certification of key individuals is required, including a brief description of the role these individuals will play in the contract.

4.2.4 Management and Implementation plan

The Tenderer shall provide a detailed implementation plan of proposed activities for the duration of the framework agreement. Deliverables should be consistent with requirements specified in section 2.2.

The Tenderer is requested to include management and implementation activities within a dedicated work package (WP0). The number of milestones is not restricted, but they should be designed as markers of demonstrable progress in service development and/or quality of service delivery. Adjustments to the proposed implementation plan can be made on an annual basis depending on needs for service evolution, changed user requirements, or other requirements as agreed between the European Commission and ECMWF.

As part of the general project management description the Tenderer shall consider the following elements (this is not an exhaustive list):

- Reporting obligations shall be provided in accordance with the Framework Agreement Article 2.3 and Annex 5.
- Monthly teleconferences with ECMWF and a proposal for involvement of ECMWF in major project reviews shall be provided as part of the management plan.
- A proposed payment plan shall be provided as part of the proposal in Volume IIIA.
- The following management aspects shall be described: task and resources planning and tracking, quality assurance and control, communication management (ECMWF, stakeholders, internal communication), conflict resolution, subcontractor management, personal data management (i.e. how this meets the requirements of Clause 2.8 and Annex 6 of the Volume V Framework Agreement) and risk assessment and mitigation plans.
- If relevant, a list of sub-contractors and details of their contribution, key personnel, legal names and addresses shall be provided. The Tenderer shall describe how the Framework Agreement, in particular Clause 2.9, has been communicated down to all their sub-contractors.

Tenderers shall complete the relevant table in Volume IIIA as part of their bid, which shall include the deliverables and milestones for this work package already indicated in the tables below. Volume IIIA will be used by the Tenderer to describe the complete list of deliverables, milestones and schedules for each work package. All milestones and deliverables shall be numbered as indicated. All document deliverables shall be periodically updated and versioned as described in the tables.

WP0 Contractual Obligations Template				
#	*Responsible*	*Nature*	*Title*	*Due*
D0.y.z-YYYYQQ	Tenderer	Report	Quarterly Implementation Report QQ YYYY *QQ YYYY being the previous quarter*	Quarterly on 15/04, 15/07 and 15/10
D0.y.z-YYYY	Tenderer	Report	Annual Implementation Report YYYY *YYYY being the Year n-1*	Annually on 28/02
D0.y.z	Tenderer	Report	Final report	60 days after end of contract
D0.y.z-YYYY	Tenderer	Other	Preliminary financial information YYYY *YYYY being the Year n-1*	Annually on 15/01
D0.y.z-YYYY	Tenderer	Report	Annual Implementation plan YYYY *YYYY being the Year n+1*	Annually on 31/09
D0.y.z-YYYY	Tenderer	Other	Copy of prime contractor's general financial statements and audit report YYYY *YYYY being the Year n-1*	Annually

5 Appendices

5.1 Standards, Protocols and APIs

"AAA"/"Triple A" Accessibility	title for compliancy with Priority 1, 2 and 3 of the Web Content Accessibility Guidelines 1.0 (WCAG 1.0)	http://www.w3.org/TR/WCAG10/
CF	Climate and Forecast metadata conventions	http://cfconventions.org/
CSV	Comma Separated Value	
DataCite	Digital citations to find, access and reuse data	https://www.datacite.org/
DOI	Digital Object Identifier system	http://www.doi.org/
ECMWF-ODB	ECMWF's Observations Database	
FTP	File Transfer Protocol	http://www.w3.org/Protocols/rfc959/
GEMINI	UK Discovery Metadata Standard	http://www.agi.org.uk/join-us/agi-groups/standards-committee/uk-gemini
GeoJSON	a format for encoding a variety of geographic data structures	http://geojson.org/
GeoTIFF	file standard which allows geo-referencing information to be embedded within a Tagged Image File Format (TIFF)	http://trac.osgeo.org/geotiff/

	file (inc. Animations)	
GRIB 1	General Regularly-distributed Information in Binary form Version 1	https://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/Introduction_GRIB1-GRIB2.pdf
GRIB 2	General Regularly-distributed Information in Binary form Version 2	https://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/Introduction_GRIB1-GRIB2.pdf
GridFTP	high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks	http://toolkit.globus.org/toolkit/docs/latest-stable/gridftp/
HDF	Hierarchical Data Format	https://www.hdfgroup.org/
HTTP	Hypertext Transfer Protocol	http://www.w3.org/Protocols/
INSPIRE	Infrastructure for Spatial Information in the European Community	http://inspire.ec.europa.eu/
ISO19115	Defines the schema required for describing geographic information and services by means of metadata.	http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=53798

ISO19119	Identifies and defines the architecture patterns for service interfaces used for geographic information	http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=39890
ISO19139	defines Geographic MetaData XML (gmd) encoding, an XML Schema implementation derived from ISO 19115	http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=32557
JSON	JavaScript Object Notation	http://json.org/
KML	Keyhole Markup Language	http://www.opengeospatial.org/standards/kml/
LAS	Live Access Server is a web server to provide flexible access to scientific data	http://ferret.pmel.noaa.gov/LAS
MARS	Meteorological Archival and Retrieval System	http://www.ecmwf.int/en/what-mars
ncBrowse	Java application to visualise netCDF files	http://www.epic.noaa.gov/java/ncBrowse/
NetCDF	Network Common Data Form	http://www.unidata.ucar.edu/software/netcdf/
OAI-PMH	Open Archives Initiative Protocol for	https://www.openarchives.org/pmh/

	Metadata Harvesting	
OAUTH	open protocol to allow secure authorization in a simple and standard method	http://oauth.net/2/
OGC - CSWWIS	WMO Information System	http://www.wmo.int/pages/prog/www/WIS/
OGC - SOS	OGC Sensor Observation Service	http://www.opengeospatial.org/standards/sos
OGC - WCPS	OGC Web Coverage Processing Service	http://www.opengeospatial.org/standards/wcps
OGC - WCS	OGC Web Coverage Service	http://www.opengeospatial.org/standards/wcs
OGC - WFS	OGC Web Feature Service	http://www.opengeospatial.org/standards/wfs
OGC - WMS	OGC Web Map Service	http://www.opengeospatial.org/standards/wms
OGC - WPS	OGC Web Processing Service	http://www.opengeospatial.org/standards/wps
OGC-WCTS	OGC Web Coverage Tile Service	http://www.opengeospatial.org/
OGC-WMTS	OGC Web Map Tiling Service	http://www.opengeospatial.org/
OpenDAP	Open Source Project for a Network Data Access Protocol	http://www.opendap.org/
OpenLayers	Open Source JavaScript library for displaying map data in web browsers	http://openlayers.org/
PNG	Portable Network	http://www.libpng.org/pub/png/

	Graphics file (inc Animations)	
Rasdaman	enables Web-based geo data offerings and Big Data Analytics on multi-dimensional raster ("array") data of unlimited size	http://www.rasdaman.com/
REST	Representational State Transfer	
SensorML	OGC standard encoding for describing sensors and measurement processes	http://www.ogcnetwork.net/SensorML
SFTP	Secure File Transfer Protocol	http://www.w3.org/Protocols/rfc959/3_DataTransfer.html
THREDDS Data Server	Thematic Real time Environmental Distributed Data Services	https://www.unidata.ucar.edu/software/thredds/current/tds/
Timeseries ML	OGC encoding standard for the representation of time series observations data	https://portal.opengeospatial.org/files/60856
UV-CDAT	Ultrascale Visualization Climate Data Analysis Tools	http://uvcdat.llnl.gov/
WaterML	OGC standard encoding for the	http://www.opengeospatial.org/standards/waterml

	representation of water observations data	

5.2 Software

Apache Open Climate Workbench	software that performs climate model evaluation using model outputs from a variety of different sources	https://climate.apache.org/
Cartopy	Python package for advanced map generation with a simple matplotlib interface	http://scitools.org.uk/cartopy/index.html
CDO	Climate Data Operators	https://code.zmaw.de/projects/cdo
ecCodes	Package developed by ECMWF which provides an application programming interface and a set of tools for decoding and encoding messages	
GDAL	translator library for raster and vector geospatial data formats	http://www.gdal.org/
GI-Axe	Brokering framework	http://essi-lab.eu/do/view/GIaxe/WebHome
GI-Cat	Broker catalogue service	http://essi-lab.eu/do/view/GIcat
GrADS	Grid Analysis and Display System	http://iges.org/grads/
IRIS	Python package for analysing and visualising meteorological and oceanographic data sets	http://scitools.org.uk/iris/index.html
java-netcdf	java netcdf library	https://www.unidata.ucar.edu/netcdf-java
Leaflet	Open Source JavaScript library used to build web mapping applications	http://leafletjs.com/
Magics	ECMWF's Meteorological plotting software	https://software.ecmwf.int/wiki/display/MAGP/Magics
Matplotlib	a python 2D plotting library	http://matplotlib.org/
Metview	ECMWF's Meteorological workstation application	https://software.ecmwf.int/wiki/display/METV/Metview
MIR	computer display server for the Linux operating system	http://unity.ubuntu.com/mir/

ncBrowse	Java application to visualise netCDF files	http://www.epic.noaa.gov/java/ncBrowse/
NCO	netCDF Operators	http://nco.sourceforge.net/
NumPy	NumPy is the fundamental package for scientific computing with Python	http://www.numpy.org/
ECMWF_odb_api	API to the ODB	
OpenLayers	Open Source JavaScript library for displaying map data in web browsers	http://openlayers.org/
Rasdaman	enables Web-based geo data offerings and Big Data Analytics on multi-dimensional raster ("array") data of unlimited size	http://www.rasdaman.com/
Scipy	Python-based ecosystem of open-source software for mathematics, science, and engineering.	http://www.scipy.org/