ECMWF Copernicus Procurement

Invitation to Tender

Copernicus Climate Change Service

Volume II

Data Stores Service – Platform Enhancement

ITT Ref: CJS2_211
ISSUED BY: ECMWF Administration Department Procurement Section
Date: July 2025
Version: Final

Funded by the European Union

Implemented by ECMWF

Contents

Table of Figures

1 Introduction

1.1 Overview

The Data Stores Service (DSS) encompass the technical layers underpinning the Data Access platforms supporting the implementation of Copernicus Services by ECMWF on behalf of the European Commission (EC) - Climate Change (C3S), Atmosphere Monitoring (CAMS) and contributes to Emergency Management (CEMS). DSS expose public user-facing interfaces for those Services - Climate Data Store (CDS) for C3S, the Atmosphere Data Store (ADS) for CAMS and the Early Warning Data Store (EWDS) for CEMS. These instantiated by-Service interfaces address a diverse community of users, including policymakers, businesses and scientists. More detailed information about ECMWF-Copernicus services can be found in the following links:

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/
Emergency Management Service (CEMS)	https://emergency.copernicus.eu/
Early Warning Data Store (EWDS)	https://ewds.climate.copernicus.eu/

DSS is designed as a distributed system and open framework encompassing different architectural layers (data, software, interfaces, cloud, external) which present strong technical and functional interdependencies between them. The platform (software layer) underpins interactive web portals and programmatic API-based interfaces (interfaces layer) for a broad catalogue of heterogenous datasets (data layer), applications and other digital assets. The use of Open-Source and Standards allow to serve users directly but also to federate with external platforms which are powered by the catalogued content and functional capabilities offered within the DSS portfolio. Data Stores Service, together with other services such as the European Weather Cloud (EWC), deploys and run on the ECMWF in-house Common Cloud Infrastructure (CCI) – (cloud layer) which is physically hosted at ECMWF Data Centre in Bologna (Italy).

A recent DSS modernization process resulted on the public opening of a fully refurbished Service platform on the second part of 2024. This modernized DSS inherited the vision, architectural principles and functional scope of the former CDS platform whilst relying on open-source, cloud-oriented and cutting-edge technologies. Modernization was cornerstone to overcome the obsolescence of former components and strengthening the alignment with the ECMWF Software Engine (ESEE). All different architectural layers of the Service were modernized enabling the DSS to continue evolving organically -vertically and horizontally - and to continue offering direct users and federated platforms improved capabilities and tools for data access and visualization.

Scope of the tender: The tendered service aims to cover the following objectives:
1) Provide technical support for the operational management of the Data Stores Service (DSS) platform assuring its reliability and correct functioning.
2) Enhance the Data Stores Service (DSS) platform collaborating in the design, development, and implementation of software components.

1.2 Architectural layers

The architecture of the DSS encompassed different layers which have strong technical and operational interdependencies between them:

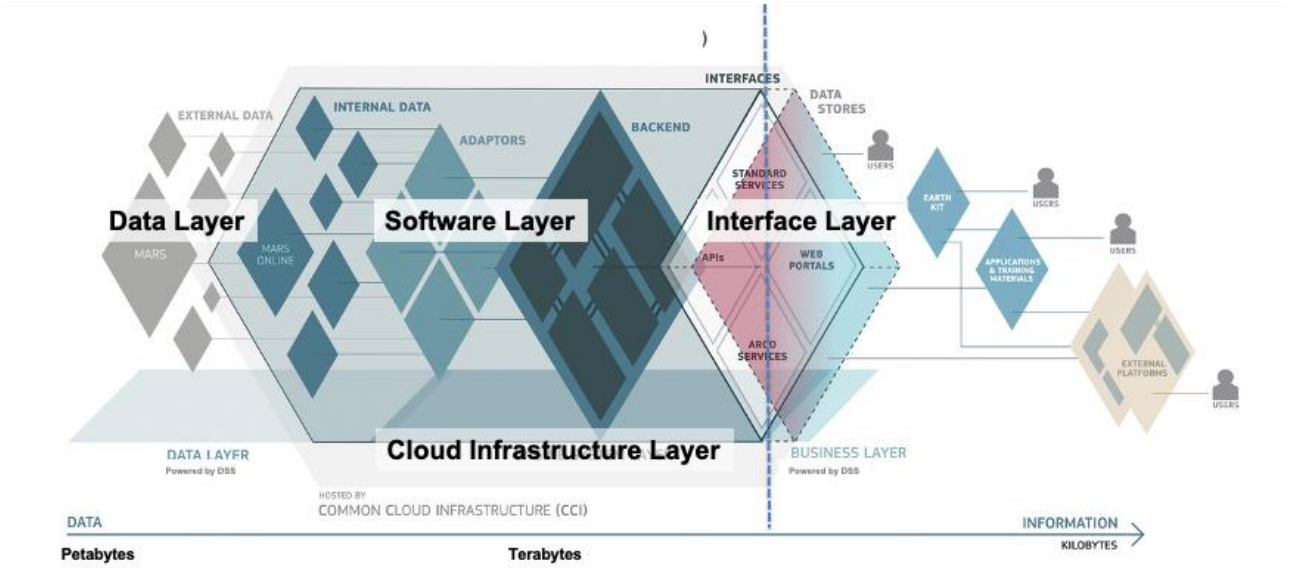

Figure 1: Diagram of the different layers that integrate the Data Stores Service (DSS)

- Data layer: The foundational pillar of the Service. Depending on where data repositories are physically hosted in reference to DSS, these are considered:
  - External Data repositories (hosted outside the CCI), ranging from federated and brokered datasets supported by third party organizations to the ECMWF MARS Archive, where flagship datasets, such as Global Reanalysis, are primarily hosted.
  - Internal Data (hosted within CCI). The most representative is the on-line MARS where a subset of the most requested variables from the core ECMWF MARS Archive are regularly uploaded. It also includes the Modernized Observations repository which was set-up as part of the modernization, other small on-disk datasets, and an experimental ARCO (Analysis Ready, Cloud Optimized) Data Lake.

- Software layer (core engine): Set of Software components which deploy and run together to support a seamless user journey for searching, discovering, sub-setting and retrieving data via interactive and programmatic interfaces. Core software layer components integrate within the ECMWF Software EnginE (ESEE). Functional scope covers the following:
  - Oversees the full life cycle of incoming data requests.
  - Hosts the catalogues, including datasets metadata and configuration.
  - Implements quality of service rules (QoS) to regulate user traffic and system resources.
  - Dispatch requests to the Data Layer via dedicated adaptors.
  - Exposes interactive and programmatic interfaces (APIs) highly configurable and compliant with standards.

- Interface Layer (dedicated service instances): Combination of operational deployments of the Software Layer with branded by-service configuration and catalogue content. Currently the DSS exposes the following service instances:

Copernicus Service	Data Store Instance	Link

C3S	Climate Data Store (CDS)	https://cds.climate.copernicus.eu/
CAMS	Atmosphere Data Store (ADS)	https://ads.atmosphere.copernicus.eu/
CEMS	Early Warning Data Store (EWDS)	https://ewds.climate.copernicus.eu/

The deployment of a running cluster of the DSS Service (eg. Production, Test or Development) is performed on top of Kubernetes as the container orchestration engine. The Kubernetes setup supports dynamic scaling of cluster components, allowing them to scale up or down as needed. Each DSS cluster consists of a shared backend that supports multiple environments and web portals which are defined at configuration level. Software code, catalogued content and configuration are hosted and managed in shared repositories on a version control platform, facilitating collaborative work between ECMWF and different partners and supporting automatic deployments.

On the periphery of the core layers, but closely interlinked with them, seats other components which are "powered by" and/or “complementary” to DSS. Of relevance are the following:

- Earthkit: ECMWF open-source Python code repository offering a broad set of expert libraries optimised to work with ECMWF and DSS resources.

Service	Link
Earthkit’s documentation	https://earthkit.readthedocs.io/en/latest/
Earthkit open code on Github	https://github.com/ecmwf/earthkit

- Visual Interactive Content (VICs): this includes a broad set of user-oriented applications and training material that showcase data and services making combined use of the full range of resources and capabilities of the DSS Data Layer and earthkit.

Service	Link
Available applications	https://cds.climate.copernicus.eu/applications
Example application: ERA5 Explorer	https://era-explorer.climate.copernicus.eu/

- External platforms: Platforms and infrastructures "powered by" the content and services provided by the DSS. These platforms interact with or consume data resources via the publicly exposed interfaces, embed technical components or integrate VICs as part of their portfolio. The WEkEO DIAS Platform, a partnership of ECMWF, EUMETSAT, Mercator Ocean and EUMETSAT and EEA, can be referred as a representative example of an external platform "powered by" the Data Stores.

The scope of this tender focus on the software layer of the Data Stores Service but not only. Due to the strong architectural and functional interdependencies across components in all different layers, tenderers should therefore consider the broader architectural context when submitting their proposals.

1.3 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
ADS	Atmosphere Data Store
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.
ARCO	Analysis Ready, Cloud Optimized
CCI	Common Cloud Infrastructure. The ECMWF in-house Cloud hosting the Data Stores Service.
CDS	Climate Data Store
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
DIAS	Data Information and Access Service
DSS	Data Stores Service
EQC	Evaluation & Quality Control
ESEE	ECMWF Software Engine
EWC	European Weather Cloud
Expert	The type of CDS Toolbox users writing CDS Toolbox workflows to build applications.
Metadata	Descriptive information about the data
OGC	Open Geospatial Consortium. https://www.ogc.org
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
SDLC	Software Development Life Cycle
SOA	Service-Oriented Architecture
Tool	A software that performs computations on some input data or products and produces some output data or products.

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.

1.4 Technical background

This chapter provides contextual technical background of the platforms and systems that conform the operational environment of the Data Stores Service and provides a high-level view on how these interrelate. An overall understanding of this environment is essential for assessing the scope and dependencies of the work to be carried out under this contract.

1.4.1 Modernized Data Store

The Data Stores platform consists of the following components:

- **Web portal:** This is a key component of the platform, offering the user-facing interface for discovery and access datasets and related services. It is developed as a JavaScript application and framework, leveraging server-side rendering (SSR) for enhanced performance and search engine optimization (SEO), while supporting full client-side interactivity.

  The Web Portal is highly configurable, with its structure and appearance defined by external configuration files managed in version control repositories (Git). This configuration-based approach allows multiple web portals to be deployed on top of a shared backend. Each web portal support tailored catalogues with interactive capabilities for data discovery, sub setting and retrieval which are automatically generated.

  Web portal integrates with a range of backend services and APIs to enable its functionality. Some of those are:

  - OGC APIs - Common (Catalogue, Faceted Search) and Processes
  - User Profile API
  - Statistics API
  - EQC (Evaluation and Quality Control) API

  For accessing data and services offered in the Data Stores, users must be registered and accept the terms and conditions governing the use of the Services. User authentication and session management are integrated with the ECMWF Identity Provider (keycloak).

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

- **Backend:** This is the core engine responsible for managing data requests throughout their life cycle on the system. It also handles the interactions with the underlying data layer. Backend provides the logic and

services needed to catalogue, filter, retrieve, and deliver data efficiently and securely. The backend encompasses the following components:

- **Catalogue:** Relational view of the catalogue metadata records optimised to be queried via all APIs. The structure of the database model is optimized to serve queries coming from different APIs and services. The Catalogue manager contains a series of utilities that keep synchronised the database with the configuration sources.
- **User profiles:** Service for managing information from user. User profiles are a combination of authentication information handle by ECMWF Authentication Services (Keycloak) and extended User Profile Metadata which is required during the registration process when first visiting any of the Web Portal instances of the Data Stores. User profiles service receives requests from the API server for checking acceptance of legal terms and from the Web portal.
- **API Server:** API entry point to the data, tools and applications catalogue.
  - **Catalogue API**: Handle queries about catalogued resources from clients. Download forms and forms constraints are also available through this API.
  - **Processes API:** This service receives data requests from clients, validate them, forward them to the Compute component and provides request management functionalities to clients.
  - **Adaptors:** Translates user requests into normalised Data Store requests before being dispatched to the different data repositories. This component receives validated requests from the Processes API.

- **Compute:** Component that receive and execute tasks submitted by the backend, handle the cache of the results, and implement the Quality of Service (QoS) rules. The compute encompasses the following components:
  - **Broker:** Service that receives requests from the Backend, implements the Quality of Service (QoS) rules that decide about limits, priorities and permissions and pass tasks to the scheduler. The Broker knows (through the request database) the status of the requests. The Quality of Service (QoS) rules can be dynamically changed to accommodate system resources, traffic and user demands. The Broker exposes an API that implements the OGC processes API.
  - **Request database:** Database that contains the status of all the user requests that reach the system with the information needed by the Broker to apply the QoS rules. The request database holds all the information need it to restore the system in case of unexpected disruptions.
  - **Scheduler:** Receive tasks from the Broker and submit them to the Compute workers. Scheduler is aware of the available resources of the system and knows the status of the running requests.
  - **Compute worker:** Execute the requests. The number of workers can be seamlessly increased and decreased across different Clouds. Requests processed by a worker write in the cache or return an error to the broker. All the operations are asynchronous. Workers are spawned by Kubernetes on demand.
  - **Results cache:** Object Storage that contains the cache of the system. Only Compute workers can write on it, and it is accessible from outside for user downloads.

- **Back-Office (CIM-EQC):** Component designed for managing the collaborative creation of Quality Assurance Reports (QARs) as support of the Evaluation and Quality Control function (EQC). QARs are complex documents containing dozens of different fields structured in groups. The structure of QARs is managed by Quality Assurance Templates (QATs) which defines document syntax (field data types, data validations, field relations). Moreover, different workflows manage the different steps in the QAR lifecycle and the interactions between different actors.

- EQC Workflow Manager: This component is responsible for publishing an API and a user interface for managing the creation, edition and publication of QARs. It implements a User Interface (UI), a backend server component with customizable role-based permissions as well as an API to facilitate integration with other components.

- Metrics: Log and monitoring service in near-real-time collecting metrics and KPIs from the operation of different components.
  - Metrics Database: Stores real-time metrics of the system. Operational logs and real-time metrics are ingested in a self-managed time series database. The stored information is available for consumption for a period that is configurable (days/weeks).
  - Metrics component: Offers administrators a near real-time view of the operational status of the system. It exposes a simplified API to fetch metrics which are integrated with ECMWF monitoring and observability tools and configured to trigger alerts when some metrics go above a certain threshold.
  - KPI database: Long-term storage for summarized KPI information.
  - KPI component: Computes and offers summarized KPI information. A logging standard (based on syslog protocol) across components allows uniform KPI information computation. KPI Information is exposed via API.

- Deployment: The deployment of running instances of the Data Store Service is performed on top of Kubernetes as the container orchestration engine and Kubespray for cluster deployment and configuration. The Kubernetes cluster is configured to be able to scale up and down components such as the workers. A working instance deploys as follow:
  - Kubernetes manifest syntax stored on a version control system (Git) to have a better control on the update and rollout of new configurations.
  - Flux performs the application deployment (CD) whenever a new change is triggered from the Git repository.

  The various assets are deployed along two complementary axes: Kubernetes namespaces, which group resources belonging to a specific system, and object labels, which distinguish between different deployment versions within a given namespace. This logical structure improves naming isolation and enables fine-grained control over resource configurations, such as version rollouts, autoscaling policies, and environment-specific settings.

  Deployments leverage horizontal Pod autoscaling (i.e. Workers), that together with the Cluster Auto Scaling allows for a better utilization of the available resources.

- Observations repository: This component is integrated into the Data Layer of the DSS to provide in-situ observations data and related functionalities. It works in coordination with other software components, such as data adaptors, to ensure seamless access and integration of these observational datasets.

  - Observations storage: host the files in CDM format and expose them with an S3 compatible API. It complies to the CDM-OBS data model (as currently defined at github.com/glamod/common_data_model and to a well-established data format (NetCDF4) that allows fast searches and is easily readable with xarray.
  - Observations catalogue host metadata tables pointing to the objects in the object storage. This component also hosts the data licenses and the versions. Metadata storage and the object storage are required to be synchronized.
  - Observations API Server: Serves metadata on observations files.

- **Administration and publication tools:** Set of command line capabilities to publish and administer Observations repository content into the DSS catalogue.

Currently the Observations repository serve a series of datasets available via the Data Stores catalogue. Some examples are:

Data Set	Catalogue link
In situ temperature, relative humidity and wind profiles from 2006 to March 2020 from the GRUAN reference network	https://cds.climate.copernicus.eu/datasets/insitu-observations-gruan-reference-network?tab=overview
In situ observations of meteorological variables from the Integrated Global Radiosounding Archive and the Radiosounding Harmonization dataset from 1978 onward	https://cds.climate.copernicus.eu/datasets/insitu-observations-igra-baseline-network?tab=overview

The following table summarize the different technologies which are used on each of the components:

Component	Technologies	Others
Web Portal	React, TypeScript, Next.js	file(s) formats: JSON, YAML, HTML.
Backend	Python, PostgreSQL, FastAPI, pygeoapi, OSWLib, SQLAlchemy, Geoserver.	OGC API – Common, OGC API – Processes, OGC API – Environmental Data Retrieval (EDR)
Compute	Python, PostgreSQL, Dask, Openstack OS.	
Back-Office (EQC)	Java (Spring Boot, Spring Web, Spring REST, Jooq, Flowable), React, Apache Lucene and PostgreSQL	
Metrics	Prometheus, OpenMetrics, FastAPI, React, Kibana-Elastic, PosgreSQL	Splunk, Opsview
Deployment	Kubernetes, Flux, Openstack	Kubespray
Observations repository	Openstack Ceph, S3, PostgreSQL, Python, FastAPI	CDM-Obs Data Model

Figure 2 Summary of technologies used on the Data Stores Service

The successful tenderer will be granted access to detailed technical documentation; development platforms and source code associated with of the Data Stores Service after the signature of the contract. In addition, a structured knowledge transfer and handover period with the outgoing contractor is envisaged to ensure continuity and a smooth transition.

1.4.2 ECMWF Software EnginE (ESEE)

ECMWF has long been at the forefront of global weather prediction and atmospheric research, with a strong tradition of developing and maintaining high-performance computing systems and specialized software for atmospheric and climate modelling. In this context ECMWF Software EnginE (ESEE) encompass the broad family of software products and libraries design, developed and managed by ECMWF. ESEE has evolved driven by community requirements, growing data volumes, increasing model complexity, and the emergence of new paradigms such as Machine Learning and high-throughput data workflows.

The Data Stores software platform has been designed and developed based on ECMWF's extensive in-house knowledge and experience running operational systems and developing expert software. Looking ahead, evolution of the Data Stores Service aims to gradually strength its convergence towards ESEE to ensure operational compatibility, reusability of code and software components and alignment with ECMWF architectural guidelines and principles for delivering a more unified, maintainable, and future-ready platform for serving Copernicus data to users and to facilitate federation with other EU platforms such as WEkEo, DestinE or emerging Data Spaces.

As part of the ECMWF Software EnginE (ESEE), earthkit plays a key role in supporting the Data Stores Service (DSS). This open-source Python project, led by ECMWF, is of relevance to DSS as its libraries are used across multiple components of the system to facilitate data access, transformation, and integration workflows.

earthkit	https://earthkit.readthedocs.io/en/latest/

earthkit is designed to accelerate weather and climate science workflows by simplifying tasks such as data access, processing, analysis, and visualisation. It offers a modular suite of interoperable components built on trusted Python libraries such as NumPy, Pandas, and Matplotlib, while also integrating smoothly with ECMWF's operational software stack, including ecCodes, FDB, and other core infrastructure used within DSS.

Its high-level, user-friendly interface makes earthkit suitable for a wide range of users across ECMWF and the wider community. With scalability and operational readiness in mind, it helps bridge the gap between research and production, enabling the Data Stores Service to provide more flexible and efficient tools for accessing and using data.

The project is currently in BETA, with several components available through PyPI and GitHub. As development continues, interfaces and features may evolve to support emerging needs and deeper integration with DSS and visual interactive content (VICs) such user applications and viewers.

The successful tenderer will be required, whenever its suites functional purposes, to use **earthkit** libraries and even to actively contribute to its open-source code development in alignment with ECMWF software practices.

1.4.3 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 the partner EUMETSAT and their suppliers. The service is now live on the Internet.
The delivery platform allows end-users to access a substantial portfolio of original Copernicus Programme and Sentinel satellite data using a harmonised data access mechanism and common data catalogue. Users can 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 portfolio of CDS, ADS and CEMS datasets is available through the WEkEO platform to allow bespoke processing of that data in unison with all other Copernicus data using a common data access mechanism.

WEkEO is a federated platform that regularly harvest Data Stores catalogues metadata via the programmatic standards interfaces, reuse configuration files to recreate data subsetting functionalities, forms and constraints, and access source data via APIs with dedicated Quality of Service rules. WEkEO also integrates earthkit as data access, manipulation and visualization tools

The successful tenderer will be expected to support the integration of the Data Stores Service with WEkEO and other federated platforms, through the enhancement, implementation and maintenance of the programmatic interfaces and operational monitoring and management capabilities.

2 Contract Definition

2.1 Vision and Objectives

Vision: 1) Provide technical support and operational maintenance for the Data Stores Service platform, enabling reliable service delivery and continuous system availability. 2) Contribute to all phases in the software life cycle to the evolved functional capabilities of the Data Stores Service platform.

This Invitation to Tender (ITT) is expected to result in the award of a single Framework Agreement implemented via a single Service Contract with a duration of 36 months for the provision of services supporting both the operational maintenance and the continued enhancement of the Data Stores Service platform throughout contributions to the different phases in the Software Development Life Cycle (SDLC) which may cover planning, design, development, implementation, test, deployment and maintenance.

The areas of work covered by this contract can be then summarized as:

- Provide operational third-line support for troubleshooting and resolution of technical incidents.
- Contribute to the software development life cycle of different DSS system components.

2.2 Scope of work

- Support and technical maintenance of the Data Stores Service (DSS) platform, including all components and technologies outlined in **Chapter 1.4.1**, to ensure operational reliability, availability, performance and quality of service to users. Key activities include:
    - Provide third-line support for the investigation and resolution of technical incidents or service disruptions affecting the platform.
    - Perform tuning and configuration of components to optimise performance and ensure correct system behaviour.
    - Apply corrective actions, including hotfixes, patches, and configuration updates, in coordination with ECMWF staff.
    - Ensure the high availability, stability, and reliability of the Data Stores Service in operational environments.
    - Carry out software version upgrades and manage component updates as part of regular maintenance or incident resolution workflows.

- Contribute to the software life cycle activities following ECMWF architectural principles, quality standards, and long-term strategic objectives for the Data Stores and ECMWF Software Engine.
    - Participating in all phases of the software development lifecycle (SDLC), including planning, design, implementation, testing, deployment, and documentation.
    - Evolve existing software components of the DSS platform to support evolving user and operational requirements.
    - Refactoring of existing software components to improve maintainability, performance, or compliance with ECMWF standards.
    - Ensure the architectural integration of components with ECMWF software engine and with other external platforms to which DSS may federate.
    - Working in an Agile development environment, using tools such as GitLab, Jira, and Confluence, and adhering to ECMWF's coding, documentation, and quality assurance practices.

The contractor is expected to work collaboratively with ECMWF teams and be flexible to adapt evolving requirements throughout the duration of the contract.

2.3 Work methodology

The contractor is expected to follow an Agile software development methodology aiming to produce incremental releases of working software. This approach also aims to ensure close collaboration with ECMWF, continuous feedback, and regular delivery of functionalities that can be evaluated, integrated, and improved over time. Each iterative development cycle is expected to cover:

- **Planning:** Definition of requirements and objectives, task breakdown, and prioritisation of activities in close collaboration with ECMWF.
- **Design:** Technical analysis and solution design, ensuring alignment with the overall system architecture.
- **Development:** Development of clean, modular, and maintainable code that meets the acceptance criteria defined during planning.
- **Testing:** Integration of automated and manual testing practices, including unit, integration, and system tests, to ensure robustness and performance.
- **Deployment:** Delivery of tested components into staging or production environments using CI/CD pipelines, with appropriate rollback and release documentation.
- **Review and Retrospective:** Presentation of completed work, collection of feedback, and identification of improvements for future sprints.

Progress will be tracked using ECMWF tools, including GitLab for version control and CI/CD, Jira for sprint planning and task tracking, and Confluence for documentation.

This structured yet flexible approach is intended to promote collaborative work, continuous delivery, knowledge sharing, and ensure the resulting software components integrate smoothly into Data Stores Service (DSS).

2.4 Team Structure and Skills

The contractor is expected to propose a multidisciplinary team with the appropriate mix of roles and expertise to support maintenance and development activities across all different components of the Data Stores platform, covering software development, platform deployment, quality assurance, and data processing capabilities.

The following are typologies of profiles expected to be submitted:

- **Software Engineer / Developer**: Experienced in designing, implementing, and maintaining large-scale, software systems. Proficient in Python (Rust as alternative) and familiar with REST APIs, FastAPI, SQLAlchemy, PostgreSQL, and cloud-native patterns (e.g. microservices, containers). Knowledge of ECMWF standards (e.g. ecCodes, OGC APIs) is a plus. Proficiency in Rust is also desirable.
- **DevOps Engineer**: Skilled in managing CI/CD pipelines, infrastructure as code (e.g. using Kubernetes, Flux, or Kubespray), and automated deployment in cloud environments (e.g. OpenStack). Strong understanding of monitoring, logging (e.g. Prometheus, Kibana), and platform reliability engineering.
- **Software Quality Engineer**: Responsible for setting up and maintaining test strategies, including unit, integration, and system-level tests. Experienced with automated testing frameworks and quality assurance tools integrated into Agile workflows.
- **Frontend Developer / UX Expert**: Experienced in developing modern, standards-compliant user interfaces using frameworks such as React, TypeScript, and Next.js. Able to design and implement intuitive, performant, and accessible interfaces for data access and visualization.

- Data Engineer / Domain Specialist *(Desirable)*: Familiar with environmental or meteorological data formats (e.g. GRIB, NetCDF) and access frameworks (e.g. FDB, MARS, CDS API). Capable of supporting semantic data access, indexing, and optimisation of I/O performance.
- AI/ML/LLM Specialist *(Desirable)*: Knowledge of model training, dataset preparation, and optimisation for GPUs is expected. Experience with large language models (LLMs) and their integration into data access or support services (e.g. via conversational interfaces, retrieval-augmented generation) is considered a strong asset.

Tenderers should clearly identify which individuals will fill these or equivalent roles and provide evidence of relevant experience, qualifications, and successful delivery in similar projects. CVs or concise profiles should be included in the proposal, highlighting key skills and past contributions to comparable services or platforms.

2.5 Service delivery

The contractor will be responsible for delivering services that ensure the operational reliability, technical evolution, and long-term maintainability of the Data Stores Service (DSS). The Service delivery process is expected to follow Agile methodology. In the technical side service delivery will align with ECMWF software engineering and development practices, and meet quality, security, and delivery standards.

- Operational and Development Responsibilities:
    - Operational Support: Provide third-line technical support, troubleshooting and incidents resolution in line with agreed SLAs. Final aim of the operational support is to ensure the reliability, stability and scalability of the DSS platform. The operational support is expected to cover working hours and extended on-call support.
    - Software Development: Contribute to the development of new features and to the enhancement of existing components, covering the full software development lifecycle—planning, implementation, testing, deployment, and documentation. This activity will be based on incoming requirements that will be analysed jointly with ECMWF and converted into tasks following the Agile methodology.

- Quality Assurance and Documentation: The contractor is expected to implement and maintain robust quality assurance processes to ensure the functionality, reliability and maintainability of software deliverables. This includes:
    - Applying automated and manual testing strategies (unit, integration, regression testing).
    - Conducting code reviews and acceptance processes following ECMWF release processes and coding standards.
    - Delivering comprehensive documentation, including system architecture, interface control documents, users and administration documentation.

- Knowledge Transfer and Collaboration: To ensure the continuity of the service and effective team collaboration, the contractor must:
    - Actively engage in knowledge-sharing activities, including handovers, documentation, and proactive participation in meetings, reviews or demonstrations.
    - Contribute to and keep regularly updated training material and operational documentation for internal teams and external stakeholders.
    - Work collaboratively using ECMWF tools (e.g. GitLab, Jira, Confluence).

- Security and Compliance: Given the operational and highly expose nature of ECMWF and the DSS, the contractor must:

- Follow secure software development lifecycle practices throughout all phases of work.
- Ensure proper handling of credentials, tokens, and user data in accordance with EU data protection regulations (e.g. GDPR).
- Address security vulnerabilities through code scanning, threat modelling, and the application of patches or mitigations as needed.

- **Service Level Agreements:** Service delivery will be governed by a set of Service Level Agreements (SLAs) previously agreed with ECMWF. SLAs will be formalised during contract negotiations but are expected to be aligned with the proposal in the following table:

Severity Level	Description	Resolution Time	Implementation
Critical	Complete service outage or major degradation impacting final users	4 hours	Hot-fix release
High	Significant functionality loss or degraded performance affecting service delivery	1 day	Hot-fix release
Medium	Minor issues, limited user impact, non-urgent bugs	Within current iterative development cycle	Major release
Low	Cosmetic issues, general questions, or non-urgent improvement requests	Based on define priorities	Major release

Figure 3 Service Level Agreements (SLAs)

The envisaged levels of support split in:

- **Normal Support:** With in working hours to be defined based on the geographical location of the tenderer and the working calendar to be applied.
- **On-Call Support** (just for Critical incidents): Optional or defined based on ECMWF requirements and operational role.

2.6 Summary of requirements

To guide the tenderer on drafting its proposal, the following structure for their response is proposed:

- Tenderer profile:
  - **General credentials:** Provide legal, organisational, and financial details as requested in Volume IIIA **Template for Tenderers – Administrative Information** (Sections 1–2). Include any relevant certifications (e.g. ISO 9001, ISO/IEC 27001) and a brief description of you quality assurance processes, including how these are maintained and audited.
  - **Track record and references:** Using the format in Section 4 of Volume IIIA, present up to three relevant projects delivered in the past 5 years, ideally describing the involvement of individuals in the proposed team. Include contact details for reference checks. ECMWF may contact referees at any stage. (cf. further details in section 4.2.2 of this document).
  -

- Technical capabilities and resources:

- Technological expertise: Demonstrate proficiency in the technologies and frameworks described in this ITT. Provide examples of applying these technologies in similar projects.
- Team profiles: Describe the team structure and proposed roles. Include CVs of experts, highlighting relevant experience, technical qualifications, and certifications
- Innovation and problem solving: Provide examples of how your team has contributed to innovative solutions, highlighting projects in similar context of applicability—e.g. reengineering software code, optimising architectures, introducing state-of-the-art technologies, or resolving unexpected challenges.
- Documentation and training: Explain your approach to technical and user documentation, as well as knowledge transfer. ECMWF uses Atlassian Confluence for internal documentation and Read The Docs for user-facing materials.

- Management and organisation of work:
  - Working Methodology: Describe your project management approach, including planning, delivery, risk management, and scope control. (cf. further details and requirements related to management and coordination work package in section 3.2.3 of this document). ECMWF aim to follow an Agile methodology, but tenderer is expected to detail and propose ways of implementation.
  - Support and Maintenance: Outline your ability to provide operational support, including error handling, software updates, and bug fixing. Indicate availability (e.g. business hours or extended), response times, and escalation paths.
  - Scalability and Resilience: Demonstrate your ability to scale resources and adapt to changing requirements or workloads. Provide examples of flexibility in past contracts.
  - Progress Reporting and Handover: Describe your reporting mechanisms and tools (e.g. Confluence, GitHub, JIRA). Explain how you ensure transparency, traceability, and a clean handover at project completion.

3 Contract Management

3.1 Contract Approach

As stated in previous chapters the final objectives of this contract are:

- Provide operational third-line support for troubleshooting and resolution of technical incidents for the current operational DSS.
- Contribute to the software development life cycle of DSS system components.

The contract approach will then align with the following objectives:

- Technical solutions will be designed, developed, implemented and deployed following ECMWF vision, strategy and architectural principles for the future of the Data Stores Service.
- Experience from the operational management of the current system will be the baseline for incoming requirements and technical specifications.
- Prioritization and final scope of incoming requirements will be driven by ECMWF but contractor is expected to actively contribute with proposals and ideas.
- The evolution of the service is expected to happen gradually avoiding system disruptions and user impact.
- FAIR (Findable, Accessible Interoperable, Reusable) principles are driven at all levels where these are applicable (metadata, data, …).
- Existing code and components are primarily considered for being reutilized or reengineered.
- Project roadmap and deliverables will be aligned with the requirements and goals of ECMWF and Copernicus Services.
- Transition between systems components is done in a smooth and seamless way.

To achieve the above-mentioned objectives an approach to the contract in two separate areas of work is proposed:

- Operational support: Provision of operational support for the troubleshooting and resolution of technical incidents. Based on a backlog of operational bugs and incidents.
- Platform enhancement: Activities within the development life cycle of software components within the Data Stores Service platform.

Work and activities expected to be covered under each of these phases will be addressed by different Work Packages as described in chapter 3.2 of this document.

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.

3.2 Work Packages

3.2.1 Work Package 1: Maintenance and support

As part of the work tendered, the contractor is expected to take over the operational maintenance of the current Data Stores Service infrastructure, including the different components described on chapter 1.4.1, This activity is envisaged to be addressed in different steps:

- **Handover:** this is expected to be carried out 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 a series of sessions to become confident to take over the maintenance task. This phase is planned to have a duration up to 2 months.
- **Support:** The contractor will take full responsibility for the operational maintenance of the current Data Store Service Infrastructure. This activity will be triggered once the handover activities are completed.

One formally accepted and integrated within the operational platform; any delivered software will be covered by the support activity.

The scope of this work package covers the following objectives and tasks:

- Hand-over and knowledge transfer activities with former contractor.
- Third-line support activities based on the agreed Service Level Agreements (SLAs).
- 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 Work Package 2: Software platform enhancement

This work package focuses on the technical and functional evolution of the software components and architectural layers that conform the Data Stores Service (DSS) platform. Activities will address the development of new features, the refinement of existing components, and the alignment of the platform with evolving user and system requirements.

These activities are likely to address among others the following major topics, as well as many other minor topics:

- Redesign of the broker/queueing system and task distribution
- Support for ECMWF's feature extraction stack
- Improvements to the STAC cataloguing integrations
- Implement a collaborative Catalogue Management Platform
- Evolution of User Interfaces and Visual Interactive Content (VICs)
- Adaption of ML/LLM for advanced capabilities of components

The different activities are expected to be carried out jointly by ECMWF staff and the contractor team, using an Agile or equivalent iterative methodology. Where appropriate, the analysis and design of solutions may

require interaction with third parties, particularly those concerning integration with external platforms or services.

The final objectives and tasks of this work package are the following:

- **Preliminary phase:** this phase will be conducted just after the kick-off of the contract and will address the following objectives:

  - Introduce the project teams.
  - Define overall vision and expectations for the evolution of the platform.
  - Define the work methodology and configure the work environment and tools.
  - Define the initial set of requirements, populate the backlog and define priorities for the first release.

- **Development phase:** This phase will start just after the preliminary phase has produced the expected outputs. Objectives of the development phased are the following:

  - Implement CI/CD development approach.
  - Conduct continuous iteration and requirements analysis with key stakeholders.
  - Produce software code that address the expected requirements under the agreed terms and functional specifications.
  - Conduct component, unit, system, and integration testing.
  - Deliver incremental working units of software for acceptance tests at ECMWF.
  - Deliver all required documentation.

Internal organization of this work package can be split in different sub-packages if this serves the project to organize and manage resources more efficiently.

All final deliverables as result of this phase must be approved by ECMWF as conditional requirement for final integration on the operational platform.

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

Scope	Participants	Duration	Objectives	Location
Initial workshop (core activity of the preliminary phase)	ECMWF, Contractor	To be agreed	• Introduce the project teams. • Define overall vision and expectations for the evolution of the platform. • Define the Work Methodology and configure the different supporting tools. • Define the initial backlog and priorities for the first release.	ECMWF premises (location to be confirmed)

Regular work meetings (Agile)	ECMWF, contractor, third parties (optional)	To be agreed based on the final work methodology	• Backlog review. • Planning	Flexible
Technical Meetings	ECMWF, contractor, third parties (optional)	To be agreed. On-demand	• Deep dives into specific requirements. • Clarification of requirements. • Regular catchups.	Virtual

Reference material for this phase will be:

- Work methodology proposed by the contractor as response of this tender.
- Current Data Store Service platform infrastructure, code repositories and available documentation.
- Inputs and requirements collected from key stakeholders.

3.2.3 Work Package 0: Management and Coordination

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 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.

The Tenderer is requested to include management and coordination activities within a dedicated work package (WP0). This work package shall include the overall responsibility for day-to-day service management and coordination.

In particular, the following management aspects shall be briefly described in the technical proposal:

Meetings:

- A kick-off meeting (by videoconference).
- ECMWF will organise regular progress review meetings (by videoconference).
- ECMWF organises annual C3S and CAMS General Assemblies. The Contractor (with maximum 2 team members) is expected to attend to at least one of these meetings on an annual basis and contribute to discussions related to the topic of this ITT. A travel budget for this purpose shall be foreseen and included in Volume IIIA as part of tenderer's response.
- Tenderers can propose additional project internal meetings, as they deem needed, as part of their response.

Quality assurance and control: The timely delivery as well as final quality check of the deliverables shall be ensured by the prime contractor (in terms of content, use of ECMWF reporting templates for deliverables and reports (Microsoft Word), format, deliverable numbering and naming, spelling and typos…); all reports and deliverables in this project shall be submitted in English. Unless otherwise specified the specific contract

Deliverables shall be made available to ECMWF in electronic format, via the relevant deliverable repository system.

Communication management (incl. external and internal communication). Any external communication activity must be agreed with the ECMWF Copernicus Communication team in advance. This includes, but not exhaustively, communication planning, branding and visual style, media outreach, website and social media activity, externally facing text and graphical content and events. Such agreed activity would also need to be evaluated and reported on, once complete, so that success measures and KPIs could be provided to the European Commission (cf. Clause 2.4.6 of the Framework Agreement).

Set of Key Performance Indicators (KPIs) suitable for monitoring various aspects of contract and service performance, including (but not limited to):

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

The proposed KPIs shall be SMART (specific, measurable, actionable, realistic and time bound). All KPIs shall be regularly reviewed and updated together with ECMWF, during the contract. The KPIs shall be designed to quantify different aspects of quality of service against the requirements described in this document. If needed, these initial specifications may be refined together with ECMWF during the negotiation of the contract. The Contractor shall report to ECMWF on these KPIs as part of the quarterly progress review meetings, as well as part of the Quarterly and Annual Implementation 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. The template to be used by the Tenderers to describe the KPIs is included in Volume IIIB of the ITT "Template for Tenderers".

Risk Management: The proposal shall include a risk register that describes identified risks for each work package, along with a mitigation strategy for each of the identified risks. This mitigation strategy shall be composed by both preventive and corrective measures. The risk register shall be updated regularly by the Contractor, and any update (related to new risks, likelihood or impact) shall be reported during the progress review meeting, as well as part of the quarterly and annual implementation reports.

Resources planning and tracking using the appropriate tools.

Subcontractor management, **including conflict resolution**, e.g. the prime contractor is responsible for settling disagreements, although advice/approval from ECMWF may be sought on the subject. If relevant, a list of subcontractors describing their contribution and key personnel shall be provided, as well as backup names for all key positions in the contract. Tenderers shall describe how the Framework Agreement; in particular Clause 2.9 on Sub-contracting has been flowed down to all their subcontractors.

Management of personal data and how this meets the requirements of Clause 2.8 on Personal Data Protection and Annex 6 of the Framework Agreement.

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**, as 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.

List of minimum deliverables and milestones required as part of WP0, covering the contractual and financial **reporting obligations** towards ECMWF in line with the Terms and Conditions of the Framework Agreement (cf. Clause 2.3 and Annex 5):

Required Deliverables under WP0		
Deliverable #	Title	Due

D211.bis.0.1.1-YYYY.QQ	Quarterly Implementation Report QQ YYYY (QQ YYYY being the previous quarter)	On 15/04, 15/07 and 15/10
D211.bis.0.1.2-YYYY	Annual Implementation Report Part 1 YYYY (YYYY being the Year n-1) This includes: 1) Quarterly implementation Report for the previous quarter Q4 YYYY 2) Preliminary financial form YYYY (YYYY being the Year n-1)	Annually on 15/01
D211.bis.0.1.3-YYYY	Annual Implementation Report Part 2 YYYY (YYYY being the Year n-1)	Annually on 28/02
D211.bis.0.1.4	Final implementation report	end of the contract
D211.bis.0.2.1-YYYY	Annual Implementation Plan YYYY (YYYY being the Year n+1)	Annually on 30/09
D211.bis.0.3.1-YYYY	Copy of prime contractor's general financial statements and audit report YYYY (YYYY being the Year n-1)	Annually (no-cost associated)

Required Milestones under WP0			
Milestone #	Title	Means of verification	Due
M211.bis.0.1.1.QX	Progress review meetings with ECMWF	Minutes of meeting	Quarterly
M211.bis.0.1.2	Kick-off meeting	Minutes of meeting	By M1

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 technical 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.
- Proposed roles and assignments within the proposed work methodology.
- 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 Copernicus Program, Services and related Data platforms such as the ECMWF Data Stores Service (DSS) for Climate and Atmosphere Monitoring Services or shared federated platforms such as WEkEO.

The Successful Tenderer's 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 to oversee the development and deliverables. The Product Owner will be the point of contact for the Tenderer. The Product Owner will:

- Monitor the Successful Tenderer's work execution.
- Review Successful Tenderer's specifications and architectures to ensure that they are "fit for purpose".
- Be the focal point to provide the Successful 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 as agreed on the work methodology.
- Progress reports covering the various tasks included under the work package description.
- 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, including:
    - 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 Summary of tasks to be performed

The successful Tenderer is required to:

- Implement the work packages described in this ITT. (cf. detailed information in sections 3.2 of this document)
- Deliver related WP deliveries following the DSS processes and requirements. (cf. detailed information in sections 3.2 and 3.4 of this document)
- Carry out training of ECMWF personnel if required (transfer of knowledge). (cf. detailed information in sections 2.5 and 3.2 of this document)

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.

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	http://trac.osgeo.org/geotiff/

	information to be embedded within a Tagged Image File Format (TIFF) 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 Metadata Harvesting	https://www.openarchives.org/pmh/

OAUTH	open protocol to allow secure authorizatio n 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 Graphics file	http://www.libpng.org/pub/png/

	(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 representation of water observations data	http://www.opengeospatial.org/standards/waterml

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/