

ECMWF Copernicus Procurement

Invitation to Tender



Data and Information Access Services (DIAS) Platform

Processing, Data Access & Tools Software

Volume II

| |
|---|
| ITT Ref: COP_043 |
| ISSUED BY: ECMWF Administration Department Procurement Section |
| Date: 23 July 2018 |
| Version: Final (1.0.0) |



Funded by the European Union

Implemented by



DATA AND INFORMATION ACCESS SERVICES (DIAS) PLATFORM

PROCESSING, DATA ACCESS & TOOLS SOFTWARE

COP_043 (ITT-2)

STATEMENT OF WORK

TABLE OF CONTENTS

| | | |
|-------------|--|-----------|
| 1 | INTRODUCTION | 7 |
| 1.1 | Background to WEKEO (DIAS) | 7 |
| 1.2 | Nomenclature | 9 |
| 1.3 | Reference Documentation | 9 |
| 1.3.1 | European Commission Generic Requirements for the DIAS | 9 |
| 1.4 | The requirement for Flexibility | 9 |
| 1.5 | Publication of The Technical Component of The Successful Tender | 10 |
| 1.6 | High-level Summary of Work Packages | 10 |
| 1.7 | Free & Open Source Software (FOSS), Community Editions Software (CE) & Proprietary Software – Requirements & Guidance | 12 |
| 1.8 | Explicit Agreement to Licence and Pay for Proprietary Software Packages or Tools | 14 |
| 1.9 | Notes on the Use of Named Products | 14 |
| 1.10 | Customer (ECMWF) Furnished Items | 15 |
| 1.11 | General Remarks on Support and Usage of Products Specified in this ITT-2 | 16 |
| 1.12 | General Data Protection Regulation (GDPR) | 17 |
| 1.13 | Information Security (InfoSec) – Policies & Procedures | 17 |
| 2 | DIAS ITT-1 ITERATION V0 – SYSTEM DESCRIPTION..... | 19 |
| 3 | WORK PACKAGES | 20 |
| 3.1 | (WP1) – Common Data Access API and Searchable Catalogue | 21 |
| 3.1.1 | European Commission Generic Requirements | 21 |
| 3.1.2 | Background to the Data | 22 |
| 3.1.3 | Data Access Requirements | 24 |
| 3.1.3.1 | Minimal Copying..... | 24 |
| 3.1.3.2 | License Enforcement & Access Control – Free Data..... | 24 |
| 3.1.3.3 | License Enforcement & Access Control – Commercial Data | 25 |
| 3.1.3.4 | Traceability & Provenance..... | 25 |
| 3.1.3.5 | European Commission Requirement - Usage Auditing | 25 |
| 3.1.3.6 | Caching | 25 |
| 3.1.3.7 | API & Service Software – Brokered Location Hiding – Browsable/Searchable Data Catalogue . | 26 |
| 3.1.3.8 | Framework & Adapters | 26 |
| 3.1.3.9 | API General Characteristics | 27 |

| | | |
|------------|--|-----------|
| 3.2 | (WP2) - Virtual Machine / Container Images & Browsable/Searchable Image Catalogue | 29 |
| 3.2.1 | Compute Node Images | 29 |
| 3.2.1.1 | General Requirements | 30 |
| 3.2.1.2 | Meteorological & Geographic Processing Software Packages | 31 |
| 3.2.1.3 | Languages, Editors & IDEs | 32 |
| 3.2.1.4 | Python & R Essential Libraries | 32 |
| 3.2.1.5 | Machine Learning / Deep Learning / Artificial Intelligence | 33 |
| 3.2.1.6 | Operating Systems..... | 33 |
| 3.2.1.7 | Security..... | 33 |
| 3.2.1.8 | Network Integration | 35 |
| 3.3 | (WP3) - Big Data Analytics Platform (Cluster Node Images) | 36 |
| 3.3.1 | The Lambda Architecture | 36 |
| 3.3.2 | The Processing Components | 38 |
| 3.3.3 | The Storage Components | 40 |
| 3.3.4 | Security | 40 |
| 3.4 | (WP4) – Build Automation (DevOps & CI/CD) | 43 |
| 3.4.1 | DevOps Toolchain | 43 |
| 3.4.2 | Shared Archive – Original Artefacts | 43 |
| 3.4.3 | Shared Archive – Pre-existing Artefacts | 44 |
| 3.4.4 | Shared Version Control System | 44 |
| 3.4.5 | DevOps CI/CD Client Tools | 44 |
| 3.4.6 | Simple Site Replication | 44 |
| 3.4.7 | Technology Options | 44 |
| 3.4.7.1 | Planning..... | 45 |
| 3.4.7.2 | Code Creation | 45 |
| 3.4.7.3 | Code Review | 45 |
| 3.4.7.4 | Documentation..... | 45 |
| 3.4.7.5 | Build..... | 45 |
| 3.4.7.6 | Testing | 46 |
| 3.4.7.7 | Binary Large Object Archiving | 46 |
| 3.4.7.8 | Image Creation & Manipulation | 46 |
| 3.4.7.9 | Infrastructure as Code | 46 |
| 3.4.7.10 | Portable Work Environments | 46 |
| 3.4.7.11 | Configuration Management | 46 |
| 3.5 | (WP5) – Testing | 48 |
| 3.5.1 | Testing Requirements | 48 |
| 3.5.1.1 | Unit Tests..... | 48 |
| 3.5.1.2 | Integration Tests..... | 48 |
| 3.5.1.3 | Functional Tests..... | 48 |
| 3.5.1.4 | Testing Framework | 48 |
| 3.5.1.5 | Simple Testing Schedule | 48 |
| 3.5.1.6 | Easy Extension | 48 |
| 3.5.1.7 | Results Feedback | 48 |
| 3.5.1.8 | Multi-site Replication | 48 |
| 3.5.2 | Tests & Testing Framework Timescales & Purpose | 49 |
| 3.6 | (WP6) – Documentation | 50 |
| 3.6.1 | General Notes on Production & Delivery of Documentation | 50 |

| | | |
|------------|--|-----------|
| 3.6.2 | General Description of Approach | 50 |
| 3.6.3 | Tenderer Developed Software Packages & Tools | 50 |
| 3.6.3.1 | High-Level Description..... | 50 |
| 3.6.3.2 | Detailed Description | 51 |
| 3.6.3.3 | Installation & Configuration Guide..... | 51 |
| 3.6.3.4 | User Guide | 51 |
| 3.6.3.5 | Build Guide | 51 |
| 3.6.3.6 | Testing Guide..... | 51 |
| 3.6.3.7 | Source Repository | 51 |
| 3.6.3.8 | Security Hashes | 51 |
| 3.6.3.9 | Rights Assignment | 51 |
| 3.6.4 | Open Source Software Packages & Tools | 51 |
| 3.6.4.1 | High-Level Description..... | 51 |
| 3.6.4.2 | Installation & Configuration Guide..... | 52 |
| 3.6.4.3 | User Guide | 52 |
| 3.6.4.4 | Build Guide | 52 |
| 3.6.4.5 | Testing Guide..... | 52 |
| 3.6.4.6 | Source Repository | 52 |
| 3.6.4.7 | Attributions | 52 |
| 3.6.4.8 | Security Hashes | 52 |
| 3.6.4.9 | Licenses | 52 |
| 3.6.5 | Proprietary Software Packages & Tools | 52 |
| 3.6.5.1 | High-Level Description..... | 52 |
| 3.6.5.2 | OSM Documentation..... | 53 |
| 3.6.5.3 | Implementation Document | 53 |
| 3.6.5.4 | Licence Assignment(s) | 53 |
| 3.6.5.5 | Warranty Certificate(s) | 53 |
| 3.6.5.6 | Online Licensing & Support Credentials | 53 |
| 3.6.5.7 | Firm-Fixed-Price Per-Use or Per-User..... | 53 |
| 3.7 | (WP7) – Training | 54 |
| 3.7.1 | General Training Materials Requirements | 54 |
| 3.7.2 | General Description of Approach | 54 |
| 3.7.3 | Specific Training Materials Requirements | 54 |
| 3.7.3.1 | Use of the data access API..... | 54 |
| 3.7.3.2 | Use of end-user Virtual Machine Images and/or Container Images. | 54 |
| 3.7.3.3 | Use of the pre-loaded tools on the various Compute Node Virtual Machine Images. | 54 |
| 3.7.3.4 | Use of the pre-loaded big-data tools on the various Cluster Virtual Machine Images. | 55 |
| 3.8 | (WP8) – Support | 56 |
| 3.8.1 | Period of Support | 56 |
| 3.8.2 | General Requirements | 56 |
| 3.8.3 | Integrated Support Operation | 56 |
| 3.8.4 | Support Requirements | 56 |
| 3.8.4.1 | Triage System | 56 |
| 3.8.4.2 | Customer Access to Triage System..... | 56 |
| 3.8.4.3 | Re-Prioritisation..... | 56 |
| 3.8.4.4 | Triage System Changes..... | 56 |
| 3.8.4.5 | Initial Reporting to Users..... | 57 |
| 3.8.4.6 | Full Reporting to Users | 57 |
| 3.8.4.7 | Workarounds..... | 57 |

| | | |
|------------|--|-----------|
| 3.8.4.8 | Upstream Reporting | 57 |
| 3.8.4.9 | Vendor Reporting | 57 |
| 3.8.4.10 | Test Framework Extension | 57 |
| 3.8.4.11 | User Satisfaction Data | 57 |
| 3.8.4.12 | Skills Transfer to Customer..... | 58 |
| 3.8.4.13 | Time Zone | 58 |
| 3.8.4.14 | Physical Presence | 58 |
| 4 | (WP9) - MANAGEMENT & IMPLEMENTATION | 59 |
| 4.1 | Effective Date of Contract (EDC) and Date of Final System Acceptance (FSA) | 59 |
| 4.2 | Implementation Plan | 59 |
| 4.3 | Tenderer Proposed Team Structure | 61 |
| 4.4 | Customer Project Team | 61 |
| 4.5 | Detailed Design | 61 |
| 5 | CONSOLIDATED REQUIREMENTS MATRICES..... | 63 |
| 5.1 | Consolidated Requirements Matrix – Section 1 - Introduction | 64 |
| 5.2 | Consolidated Requirements Matrix – Section 2 – DIAS Iteration V0 - Summary | 69 |
| 5.3 | Consolidated Requirements Matrices – Section 3 – Work Packages | 70 |
| 5.3.1 | Consolidated Requirements Matrix – Section 3.1 – Common Data Access API and Searchable Catalogue | 70 |
| 5.3.2 | Consolidated Requirements Matrix – Section 3.2 – Virtual Machine / Container Images & Browsable/Searchable Image Catalogue | 74 |
| 5.3.3 | Consolidated Requirements Matrix – Section 3.3 – Big Data Analytics Platform | 80 |
| 5.3.4 | Consolidated Requirements Matrix – Section 3.4 – Build Automation (DevOps & CI/CD) | 83 |
| 5.3.5 | Consolidated Requirements Matrix – Section 3.5 - Testing | 87 |
| 5.3.6 | Consolidated Requirements Matrix – Section 3.6 - Documentation | 89 |
| 5.3.7 | Consolidated Requirements Matrix – Section 3.7 - Training | 92 |
| 5.3.8 | Consolidated Requirements Matrix – Section 3.8 - Support | 94 |
| 5.4 | Consolidated Requirements Matrix – Section 4 –Management & Implementation | 96 |

1 Introduction

1.1 Background to WEkEO (DIAS)

As part of the European Union's Copernicus Programme under the auspices of the European Commission, the European Centre for Medium-Range Weather Forecasts (ECMWF) together with the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) and Mercator Ocean International, have joined forces to implement a Data and Information Access Services (DIAS) Platform called WEkEO. From here onwards this platform will be referred to as WEkEO (DIAS) or simply DIAS.

The WEkEO (DIAS) platform will consist of a heterogeneous cloud-based infrastructure that will make all Copernicus data and information available to its users. It will also provide the tools and processing resources needed to process the data, including big-data analysis tools.

The physical sources of the data are spread across a wide geographical area, and the quantity of data is very large – and certainly too large to replicate to many locations on a continuous basis. Accordingly, the WEkEO (DIAS) will *hide* the physical location of the data; or in other words, it will provide Data as a Service (DaaS). In addition, processing will usually be carried out as physically close to the data as possible, thereby avoiding unnecessary use of wide-area networking resources. These are key principles of the system architecture and will be discussed further in this ITT-2.

Within the WEkEO (DIAS) platform there are 4 procurement areas, each of which will have its own ITT stream. These areas are:

- Services and Operations (EUMETSAT ITT, otherwise known as ITT-1)
- **Processing, Data Access, and Tools Software (This ECMWF issued ITT, otherwise known as ITT-2)**
- System Engineering and Validation (EUMETSAT ITT, otherwise known as ITT-3)
- User Uptake, Marketing, and User Support (MERCATOR ITT, otherwise known as ITT-4)

At the time of writing, there are expected to be four iterations of the platform as follows:

- Iteration V0 - is available to end users and is intended as a proof-of-concept. It is expected that Tenderers will familiarise themselves with all available information regarding this proof-of-concept platform so as to target their tenders as accurately as possible. Tenderers are required to provide in their tenders a high-level summary of no more than one page of their understanding of this proof-of-concept platform.
- **Iteration V1 - will be a production grade platform based on the experiences of building Iteration V0, and on adapting the Iteration V0 APIs, and is due for delivery into operation by the end of Quarter 1 2019. Users are expected to migrate from V0 to V1 during the Q2 2019.**
- Iterations V2 and V3 - are each foreseen as adding additional functionality to create a comprehensive production quality big-data heterogeneous cloud infrastructure. The specifications for these iterations will be based on user feedback from Iterations V0 and V1.

This ITT-2 requires the Successful Tenderer to design, develop, implement, test, and deliver the **Processing, Data Access, and Tools Software** for Iteration **V1** of the DIAS. It also requires the Successful Tenderer to prepare and deliver related documentation, prepare and deliver related training materials, and provide related lifecycle support. Together with the output from ITT-1 Iteration V1, this will then form the entire Iteration V1 DIAS platform.

The V1 DIAS platform is required to be operational by the end of Q1 2019 and to be supported through to end of Q4 2019. An illustrated timeline for the DIAS can be found in Figure 1.1.1.

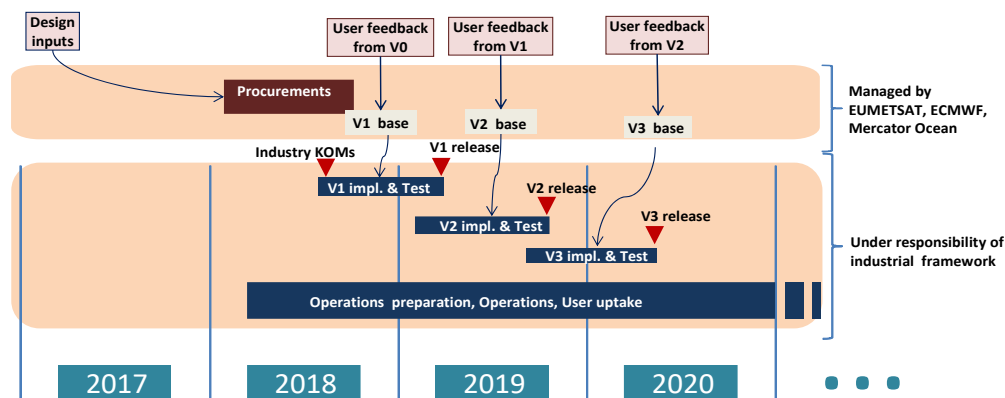


Figure 1.1.1: DIAS Project Illustrated Timeline – All Partners

The partnership's EC delegated division of responsibilities is illustrated in Figure 1.1.2.

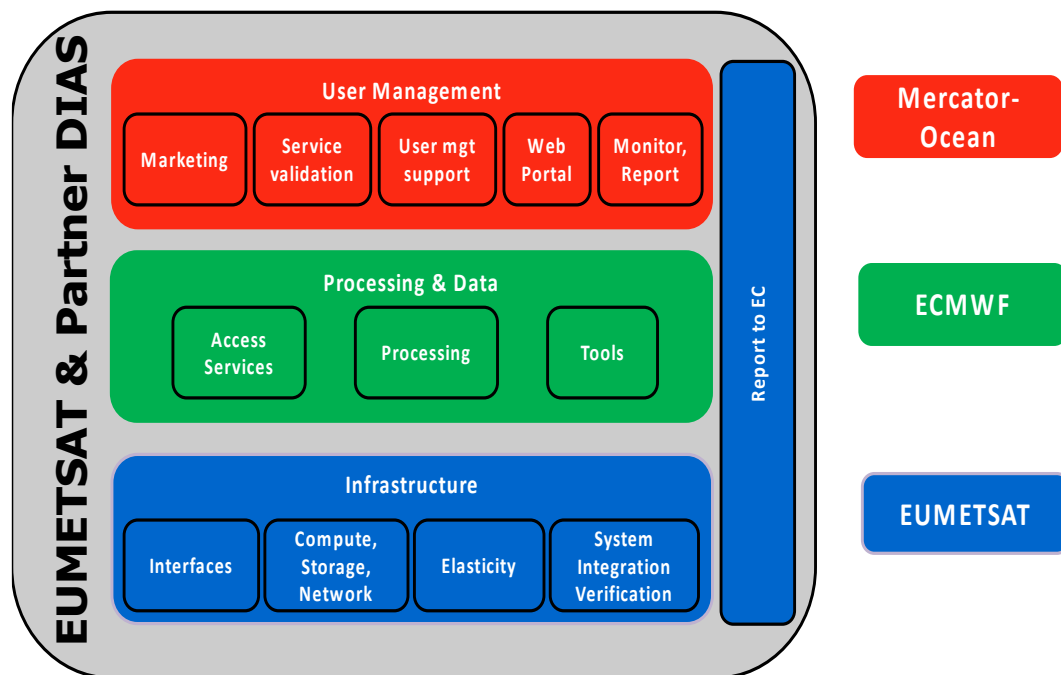


Figure 1.1.2: Partnership Illustration

In summary, the Successful Tenderer is required to deliver the following:

- the design and implementation of a Common Data Access and Catalogue API for all data and information available in the WEKEO (DIAS) platform, hiding from the users the details of the access methods and locations of each dataset,
- the design and implementation of Virtual Machine and/or Container Images that are preloaded with a set of software and tools to allow users to access and process the DIAS data and information,
- the design and implementation of a Big-Data platform that will allow users to efficiently perform analytics and machine learning (AI) on very large volume of data and information available in the DIAS, and
- the maintenance and support of the software delivered during the V1 operations period which is expected to last until the end of Q4 2019.

1.2 Nomenclature

Throughout this document the use of the term:

- **Customer** shall refer to the European Centre for Medium-Range Weather Forecasts (ECMWF).
- **Successful Tenderer** shall refer to the tenderer whose tender has been selected at evaluation and is under contract with the Customer.

Requirements expressed as “The Successful Tenderer is required to “ shall mean delivery is due at an agreed time **after** contract award.

- **Tenderers** shall refer to all tenderers submitting a tender to ECMWF for evaluation.

Requirements expressed as “Tenderers are required to “ shall mean delivery is due **with tenders**.

1.3 Reference Documentation

1.3.1 European Commission Generic Requirements for the DIAS

The European Commission has published a set of generic requirements for the DIAS in the document mentioned and linked to below. Tenderers are required to familiarise themselves with the contents of this document and certify that they have done so in their tenders.

[“Functional Requirements for the Copernicus Distribution Services and the Data and Information Access Services \(DIAS\)”](#)

http://copernicus.eu/sites/default/files/documents/News/Data_Access_Functional_Requirements_Dec2016.pdf

1.4 The requirement for Flexibility

The time available for the design, development, implementation, testing and delivery of software, preparation and delivery of documentation, preparation and delivery of related training materials,

and final system acceptance is very short – being approximately up to six (6) calendar months. Furthermore, as user experience builds using the products of V0 of the DIAS platform the requirements are likely to require some small adjustments.

For example, if the project is to gain widespread user acceptance, which is a key European Commission (EC) objective and requirement, it will be very important to adjust priorities according to feedback from subject matter experts, software development experts and platform administrators, so as to deliver the highest-impact components at the earliest opportunity.

In addition, it is to be expected that some new requirements will emerge that are sufficiently contained in functionality that they can be brought into the scope of Iteration V1 rather than waiting for the procurement of Iteration V2.

Tenderers are required to demonstrate in their proposals how they intend to satisfy this requirement for flexibility generally, and specifically how they will do this without the need for extended negotiations of Contract Changes. Tenderers are required to limit their description to no more than one page.

1.5 Publication of The Technical Component of The Successful Tender

There are a number of points of interaction between the requirements of this ITT-2 and the requirements of the ITTs for **Services & Operations, System Engineering & Validation, User Uptake, Marketing and User Support** (See Section 1.1).

The Successful Tenderer will be required to agree to full publication of the technical components of its tender response for the purpose of achieving seamless integration with the Successful Tenderers of the related ITTs. Furthermore, Tenderers are required to explicitly stipulate their consent to this requirement in their tenders (subject to them winning the tender evaluation).

1.6 High-level Summary of Work Packages

There follows a high-level description of the products of the eight (8) technical work packages required to be delivered by the Successful Tenderer: The full descriptions can be found in Section 3.

- (WP1) – *Common Data Access API & Browsable/Searchable Data Catalogue*

The design, development, implementation, testing and delivery, and final system acceptance of:

- a Common Data Access API through which all data and information available in the WEkEO (DIAS) platform may be accessed in a uniform manner, hiding from the users the details of the actual access methods and physical storage locations used for each dataset,
- a Browsable and Searchable Catalogue of all available Data together with sample views where appropriate,

NOTE: The Successful Tenderer is required to begin its work from the API, Metadata, and adapter specifications of DIAS Iteration V0, and to work in close collaboration with the Customer and with EUMETSAT when and if additions, deletions or other changes are

believed to be required. This will ensure continuity of the DIAS from Iteration V0 to Iteration V1.

- *(WP2) – Virtual Machine / Container Images & Browsable/Searchable Image Catalogue*

The design, development, implementation, testing and delivery, and final system acceptance of:

- Virtual Machine and/or Container Images preloaded with selected sets of software and tools to allow users to process the DIAS data and information accessed through the Common Data Access API.
- A Browsable and Searchable Catalogue of all available images, their intended communities of interest, and their inventories of installed software.

NOTE: The Iteration V1 DIAS infrastructure will be based on OpenStack with a Morpheus user interface. The base-level store from which images will be launched will be Glance with its limited metadata facilities. The image catalogue will provide richer metadata including communities of interest and software inventories and will point to the unique identifier for the relevant image in Glance. This catalogue will be exported via a RESTful API that can be accessed by custom code or a command line tool or by a modified GUI such as Horizon or Morpheus.

- *(WP3) – Big-Data Analytics Platform*

(WP3A) - The design, development, implementation, testing and delivery, and final system acceptance of a Big-Data platform that will allow users to efficiently perform analytics on the very large volume of data and information available through the Common Access API of the DIAS.

(WP3B **Option**) - The enhancement of the abovementioned big-data platform to accommodate machine learning (AI – artificial intelligence) techniques across the entire DIAS data and information sets.

- *(WP4) – Testing*

The development and delivery of a set of unit tests, integration tests and system tests for checking the proper functioning of the DIAS Common Access API, the Virtual Machine & Container Images, and the Big-Data Analytics Platform. The tests are to be integrated into a simple but sufficient testing framework. They are also to be extensible by subject matter experts, software development experts and platform administrators. They will also form the basis for Final System Acceptance (FSA).

- *(WP5) – Build Automation (DevOps)*

The design, development, implementation, testing and delivery, and final system acceptance of automation capabilities for (re-)constructing the artefacts developed under work packages WP1 through WP4 and, if applicable, this WP5 (i.e. bootstrapping steps).

This capability must be suitable for basic use, but customisation by end-users must also be possible.

- *(WP6) – Documentation*

The development and delivery, and final system acceptance of on-line wiki-based electronic format documentation describing the products of work packages WP1 through WP5. Appropriate indexing and re-use of existing open source documentation is encouraged. There are several target audiences; these are detailed in Section 3.6.

- *(WP7) – Training*

The development and delivery, and final system acceptance of on-line wiki-based electronic format training materials introducing beginners to the products of work packages WP1 through WP5. It may be assumed that intermediate- or higher-level users will make use of documentation resources instead. Appropriate indexing and re-use of existing open source training materials is encouraged. There are several target audiences, these are detailed in Section 3.7.

- *(WP8) – Support*

The maintenance and support of the products of work packages WP1 through WP7 during the DIAS Iteration V1 operational lifecycle which will last until the end of Q4 2019. An **option** to extend this support period by a further two (2) periods of three (3) months each may be requested by the Customer.

- *(WP9) – Management and Implementation*

The contract management principles, project implementation plan and schedule, milestone/deliverables and documentation requirements.

1.7 Free & Open Source Software (FOSS), Community Editions Software (CE) & Proprietary Software – Requirements & Guidance

Tenderers may propose the use of software from all reputable sources including:

- Free Open Source Software (FOSS) products,
- Community Editions Software (CE) of commercially supported Enterprise Edition (EE) products,
- Proprietary products, or
- Other, such as their own developments.

If a proposed solution requires the modification of Free Open Source Software:

- Tenderers are required to inform Customer of all such cases in their tenders and to fully describe and justify the proposed changes. Tenderers should be mindful that the

Customer is most concerned to avoid modified software that may become obsolete and unsupported/unsupportable over time.

- Tenderers are required to give a binding undertaking in their tenders that they will submit their changes upstream to the official maintainers should they become the Successful Tenderer.
- The Successful Tenderer is required to submit its changes upstream at the earliest opportunity and to fully support those changes until such time as they are either accepted upstream and released as a new package or the lifecycle support period has come to an end.
- When upstream changes are accepted and a new package is released the Successful Tenderer is required to update the DIAS platform to include the newly released package.

Tenderers are required to prefer Free Open Source Software (FOSS) solutions where they are available and of a sufficient quality and supportability.

Tenderers are required to justify their decisions in all cases where they choose an Enterprise Edition (EE) of a product over the corresponding Community Edition (CE), including the consequential costs impact.

Where a paid for software product is called for in a Tenderer's design his tender is required to show:

- the entire cost of ownership (including but not limited to license fees, support, updates and/or upgrades) and use over the full lifecycle of the DIAS expressed as a firm fixed price, and
- the technical reasons for specifying the product.

Tenderers are required to provide:

- a complete inventory of all Free Open Source Software (FOSS) used in their High-Level Designs whether aimed at the DIAS system or at the end-user components. (Customer recognises that this inventory cannot be considered definitive at the tendering stage.)
- the corresponding licence type for each item of FOSS.
- a full copy of each licence type.

Successful Tenderer will be required to:

- maintain up-to-date the above inventory throughout the entire lifecycle of iteration V1 of the DIAS.

Tenderers are required to provide:

- a complete inventory of all Community Editions Software and Proprietary Software.

Successful Tenderer will be required to maintain throughout the support period:

- a complete inventory of all Community Editions Software and Proprietary Software for general perusal by the Customer and by users.

1.8 Explicit Agreement to Licence and Pay for Proprietary Software Packages or Tools

Tenderers may propose the integration of proprietary software products in their designs whether aimed at the DIAS system itself or at the end-user components.

In all cases, Tenderers are required to:

- fully describe the type of licence needed (e.g. per-site, per-use, per-user, per-time-period or other),
- fully determine the cost of a licence,
- fully describe all licence management and administration requirements.

Some proprietary licensing may prevent the aggregate cost of a product's use being known at Final System Acceptance (FSA). For example, where cost is per-user and the number of users is unknown. To mitigate this

Tenderers are required to acknowledge and Successful Tenderer is required to ensure the following restrictions:

- **users shall never be deemed to have accepted any licence terms by default, and**
- **users shall never be deemed to have agreed to pay any licence fee by default.**

Tenderers are required to provide a High-Level Design for, and the Successful Tenderers is required to implement:

- **a mechanism whereby users can be made aware of the costs involved and the terms of use of any software package or tool they are proposing to use.**
They shall always be given an opportunity to make a fully informed decision whether they wish to proceed with licensing the use of the package or tool in question.
- **a mechanism whereby users can pay for and obtain a licence in the event that they wish to do so.**
- **an inventory of purchased/issued licences and corresponding users.**

Tenderers are required to provide a High-Level Design for, and Successful Tenderers is required to implement and deliver:

- **a mechanism whereby Customer can observe any licence restrictions that may apply to the DIAS system itself – for example, where a service is licensed by number of authenticated users of that service (e.g. Gitlab EE)**

1.9 Notes on the Use of Named Products

A number of software products are mentioned by name throughout this ITT-2. Some are commercial-Off-The-Shelf software (COTS), some are Free Open Source Software (FOSS) and some may fall elsewhere on the licensing spectrum.

These products are mentioned as Customer preferences, but are not mandatory. Tenderers may therefore use other products that they see as best fit for an overall successful system integration of their chosen design, understanding that these Customer preferences provide clarity as to the sort of functionality that is being sought through this ITT-2

1.10 Customer (ECMWF) Furnished Items

The Customer will provide the following equipment, services and/or access (collectively to be known as the Customer Furnished Items or CFIs) for use by Successful Tenderer during the execution of the contract:

| CFI # | Short Name | Description |
|-------|--|---|
| 1 | Access to and use of the DIAS Infrastructure | <p>The DIAS infrastructure is being procured through ITT-1 under the procurement management of EUMETSAT for deployment at three sites</p> <p>Once the first site becomes available for use, the ITT-2 contractor shall be given unprivileged access through a single tenancy for initial deployment and verification work.</p> <p>Further unprivileged tenancies shall be provided upon production of a reasonable justification.</p> <p>Privileged access to the infrastructure shall be provided under supervision upon production of a reasonable justification.</p> <p>In due course the other DIAS infrastructure sites shall become available and the ITT-2 contractor shall repeat his deployment and verification work as required to achieve Final System Acceptance (FSA).</p> <p>NOTE: Successful Tenderer is required to provide his own facilities during the design, development and testing phases of the project. This shall include any testing at scale that he deems necessary to ensure successful delivery to Customer.</p> |
| 2 | Sample Data | <p>Customer will provide best efforts support to the ITT-2 contractor to ensure that he has access to a representative sample set of data for development and testing.</p> <p>NOTE: Successful Tenderer is required to report to Customer in a timely fashion any potential or actual delays to his work occasioned by a lack of such sample data.</p> |
| 3 | Full Data Access | <p>Customer will co-ordinate information flow between the ITT-2 contractor, EUMETSAT and their contractor, and Mercator Ocean International and their contractors to ensure that he is able to gain full access to all available data for final system commissioning and Final System Acceptance (FSA).</p> <p>NOTE: Successful Tenderer is required to report to Customer in a timely fashion any potential or actual delays to his work occasioned by a lack of such co-ordination.</p> |

| | | |
|---|---|---|
| 4 | Ongoing Access for Support | <p>Customer will ensure that the ITT-2 contractor secures and retains appropriately privileged access to the ITT-1 infrastructure during the support period.</p> <p>NOTE: As a part of Final System Acceptance (FSA) Successful Tenderer is required to provide Customer with a comprehensive list of his support access requirements.</p> |
| 5 | Integration with Mercator Ocean User Support Function | <p>Customer will ensure that the ITT-2 contractor is given sufficient access to Mercator Ocean International to ensure that their support operation is fully integrated with the user helpdesk.</p> <p>NOTE: Successful Tenderer is required to report to Customer in a timely fashion any impediments he finds to establishing his support operation.</p> |
| 6 | Full Access to DIAS V0 Components and Specifications | <p>The Successful Tenderer is required to begin its work from the API, Metadata, and adapter specifications of DIAS Iteration V0, and to work in close collaboration with the Customer and with EUMETSAT when and if additions, deletions or other changes are believed to be required. This will ensure continuity of the DIAS from Iteration V0 to Iteration V1. Customer will ensure that the Successful Tenderer is supplied with all relevant information to fulfil this requirement.</p> <p>NOTE: Successful Tenderer is required to report to Customer in a timely fashion any impediments he finds to fulfilling this requirement.</p> |

1.11 General Remarks on Support and Usage of Products Specified in this ITT-2

This ITT-2 calls for:

- a data access Application Programmer's Interface (API) and dataset Catalogue to provide a single logical point of search, selection and access to a varied array of data sources. It also calls for this API to be extensible so that it can accommodate future unforeseen requirements.

Tenderers must expect that the vast majority of users will simply use the provided functionality and will have little or no use for the extensibility features; indeed, the Customer is likely to be the only user of the extensibility features after completion of the Support Phase.

Accordingly, Tenderers should seek to minimise support calls by providing training and documentation that is focussed on end-users and basic usage.

- a suite of virtual machine and/or container images and an image catalogue sufficient to cover all the likely common use cases. It also calls for virtual machine and/or container images to be user specifiable or modifiable to allow full customisation.

Tenderers must expect that the vast majority of users will simply use the provided virtual machine or container images unchanged and will have little or no use for the ability to customise.

Accordingly, Tenderers should seek to minimise support calls by providing training and documentation that is focussed on end-users and basic usage.

In summary, tenderers must use their best judgement to achieve a balance between engineering, training, delivery and documentation to ensure that their support operation is feasible at the scale expected of the DIAS in full production.

Tenderers are required to explain how their designs are the result of such best judgement.

Note: Testing is required to cover all components and types of usage irrespective of volume of use.

1.12 General Data Protection Regulation (GDPR)

In this section, the use of the word Tenderers shall also be taken to include the eventual Successful Tenderer.

Tenderers are required to comply with Regulation EU 679/2016 of 27 April 2016, the General Data Protection Regulations, and other national data protection laws as applicable to them in all aspects of their designs (of whatever level) for Iteration V1 of the DIAS.

Customer is currently unable to fully anticipate in this ITT-2 all situations in which personal data may be required to be processed by the platform; accordingly, Tenderers are required to audit their designs (of whatever level) for all uses of personal data and ensure that any processing complies with all applicable law.

Tenderers are required to provide the Customer with a full statement of compliance showing where personal data is used and how that use is within the limits of the law. Such a statement of compliance is required in tenders and for the Successful Tenderer again at Final System Acceptance (FSA).

Example situation: Tenderers may produce a design for managing user access to proprietary software that requires the storage i.e. processing of personal data. Tenderers must show how this will be implemented and how it will comply with GDPR law, such as e.g. by showing that no personal data will be processed outside the EU and EEA, unless an adequate level of protection is warranted (e.g. on the basis of a “privacy shield” arrangement).

1.13 Information Security (InfoSec) – Policies & Procedures

Successful Tenderer is required to protect the Customer’s information assets in accordance with the Customer’s relevant Policies & Procedures. In particular:

Successful Tenderers is required to:

- comply with the Customer’s Information Security Policies & Procedures and with all the underlying applicable policies (as included in this ITT-2 tendering package)
- be accountable for the information security aspects of the services and systems defined in this ITT-2

Tenderers are required to:

- provide a list of roles and responsibilities within their organisations relevant to the information security aspects of this ITT-2,
- provide a description of their secure development procedures and practices, examples being: secure code analysis and systems vulnerability assessment,
- provide a description of their Vulnerability Management Process.

A preliminary evaluation of each Tenderer's compliance with the Customer's information security requirements will be performed by Customer as part of tender evaluation. This will be based solely on the material submitted by the Tenderer in its tender.

The Customer reserves the right to require the Successful Tenderer to supply audit and/or penetration test reports to determine whether the Successful Tenderer has implemented the required Customer's Information Security Policies and Procedures.

2 DIAS ITT-1 Iteration V0 – System Description

The Successful Tenderer is required to begin its work from the API, Metadata, and adapter specifications of DIAS Iteration V0, and to work in close collaboration with the Customer and with EUMETSAT when and if additions, deletions or other changes are believed to be required.

Tenderers are advised to make themselves aware of the details of DIAS Iteration V0 using the following document as a minimum which has been supplied to them as a part of this ITT-2 tendering package:

DIAS V0 System Description

Doc.No. : EUM/GSI/TEN/17/951373 Issue : v1C

Date : 18 June 2018

WBS/DBS :

© EUMETSAT

The copyright of this document is the property of EUMETSAT.

EUMETSAT Eumetsat-Allee 1, D-64295 Darmstadt, Germany

Please refer to the document “COP_043_Volume II Annex 1_DIAS_V0_System_Description.pdf”.

3 Work Packages

This chapter provides the details of each of the eight (8) technical work packages briefly described in Section 1.6.

3.1 (WP1) – Common Data Access API and Searchable Catalogue

A Common Data Access API and Catalogue of all available DIAS data and information (and all other available contributed data) is to be High-Level Designed by all Tenderers and to be Detail Designed, developed and delivered by the Successful Tenderer.

3.1.1 European Commission Generic Requirements

The reference to the generic requirements documentation specified by the European Commission for the DIAS service can be found in Section 1.3.1 and contains, among others, the table below.

Tenderers are advised to take account of the full contents of this document, where applicable, in their designs.

| Req # | Copernicus Distribution Service High Level Requirements |
|--|--|
| <i>Distribution Service Main Functionalities</i> | |
| DIST- 1-01 | The Copernicus distribution services shall give access to all Copernicus data and information. This includes all data and information listed in [RD2], [RD6] and in the Catalogues of the Copernicus Services (Marine, Atmosphere, Climate Change, Land, Emergency, security). |
| DIST- 1-02 | All data and information shall be available to the users within the timeliness requirements as specified in [RD3], [RD6] and the Service Catalogues of the Copernicus Services |
| DIST- 1-03 | The Copernicus distribution services shall have the following functionalities for all Copernicus data and information (i.e. including Long Term Archives): (Customer Note: Not expected at Iteration V1.) <ul style="list-style-type: none"> - Discovery (Catalogue with Search functions); - View (Customer Note: This will be implemented within the catalogue, where possible, on georeferential data or as a sample where full view is not possible.) - Download - Restricted Access for specific user and data and information limited to specific cases - User management and support utility. |
| DIST- 1-04 | Data and information shall be described in searchable metadata files |
| DIST- 1-05 | The Copernicus distribution services catalogue shall, as a minimum, allow to search data or information based on the following criteria: area of interest, application field, type and date stamp |
| DIST- 1-06 | Common ontologies shall be adopted and semantic search enabled. (Customer Note: Not expected at Iteration V1.) |
| DIST- 1-07 | The Copernicus distribution services catalogue shall be user-friendly and search results shall be refined based on the user's behaviour |
| DIST- 1-08 | The Copernicus distribution services shall be documented, in particular its interfaces |
| <i>Interoperability</i> | |
| DIST- 1-09 | Services or functionalities covered by INSPIRE, including metadata content, shall be compliant with the legal obligations of the INSPIRE Directive and conformant with INSPIRE technical guidelines and good practices. If conformance with technical guidelines is shown to be incompatible with the quality of the services/functionalities required, the data or information providers and relevant stakeholders (including the INSPIRE community) shall come to an agreement of standards or implementation practices that can guarantee interoperability between the different services/platforms. |
| DIST- 1-10 | Copernicus Data and Information compliance with INSPIRE metadata requirements will ensure their discoverability by European Open Data Portal |
| DIST- 1-11 | The interfaces of the Copernicus distribution services shall be based on standards |
| DIST- 1-12 | The Copernicus distribution services catalogues shall be interoperable amongst each other and provide complete catalogue information to the DIAS. |
| Data identification, authentication and integrity | |

| | |
|---|---|
| DIST-1-13 | It shall be possible to identify unequivocally, authenticate and check the integrity of Copernicus data and information distributed by the Copernicus Entrusted Entities via the Copernicus distribution services |
| User management and registration (Customer Note: Not expected in this ITT stream.) | |
| DIST- 1-14 | User management and registration shall be coordinated across the Copernicus distribution services |
| DIST- 1-15 | No registration shall be required for the consultation of the catalogues and for viewing of data and metadata. |
| Distribution service connectivity | |
| DIST- 1-16 | The Sentinel data and service information distribution services shall connect to the GÉANT network as additional measure to increase connectivity and further serve the research community |
| Distribution Services Reporting | |
| DIST- 1-17 | Regarding the Copernicus distribution services, each Entrusted Entity shall report to the European Commission on: <ul style="list-style-type: none"> - Availability performance of the Distribution Functions - Availability performance of the Distribution Services |
| DIST- 1-18 | The components of the distribution services shall report quarterly on the service usage providing meaningful statistics on the basis of information about users, user profiles, usage patterns, etc. The reports shall cover user queries through the helpdesk. |
| DIST- 1-19 | The components of the distribution services shall report quarterly on service performance including data and information availability. The reports shall provide statistical information averaged over one month <ul style="list-style-type: none"> - Performance: availability, timeliness, anomalies - Distribution service information availability: volume, completeness, integrity, timeliness - Performance of helpdesk function |
| Legal requirements | |
| DIST- 1-20 | The distribution service must comply with all the requirements of the Copernicus data and information policy [AD1 & AD2] |
| DIST- 1-21 | All users shall have free, full and open access to the Copernicus data and information distributed by the Copernicus distribution services |
| DIST- 1-22 | Access to Copernicus data and information shall be equal to all and non-discriminatory |

3.1.2 Background to the Data

The data in question is very large, is stored in different geographical locations, is in a number of different formats, and is derived from a range of different sensors. There is a substantial amount of historical data and daily additions are also large.

Figure 3.1.1 gives an idea of the types of data available, the various formats it is to be found in and the approximate volumes of data. This table is intended for illustration purposes only. Tenderers are required to rely on their own survey information when preparing tenders.

| Category | Type | Format | Approximate Volume |
|------------------------------------|-------------------|---|--|
| Atmosphere / Climate / Meteorology | CAMS | GRIB, NETCDF, XML metadata | 22 PB (2021) |
| | C3S | PNG, Txt, MP4 | 71 PB (2021) |
| | Sentinel-2 | SAFE/JPEG2000 (INSPIRE) | 10,5 PB (2021) |
| | Sentinel-3 | NETCDF4 | 10,5 PB (2021) |
| | Sentinel-4 | NETCDF | TBD |
| | Sentinel-5 | NETCDF | TBD |
| | Sentinel-5P | NETCDF | 1,65 PB (2021) |
| | Meteosat services | NETCDF, HDF5, BSQ, TIFF/GEOTIFF, HRIT, JPEG, PNG, IDCS | 1 PB/year |
| | MetOp | | |
| Marine | Polar TEP | | |
| | CMEMS | NetCDF 3.0 | 300 TB |
| | Coastal TEP | | |
| | Sentinel-1 | SAFE/TIFF | ~ 2,5 TB (OCN products, 2018/04) |
| | Sentinel-2 | | 10,5 PB (2021) |
| | Sentinel-3 | NetCDF 4 | 10,5 PB (2021) |
| | Hydrology TEP | | |
| | Polar TEP | | |
| Land | Pleiades | DIMAP V2 | |
| | Proba-V | HDF-t GeoTIFF | 1 PB (2021) |
| | CLMS | Shape Files, XML, raster | 16 TB (global 2017) |
| | Forestry-TEP | SAFE, JP2000, NETCDF4 | (150 TB per user VM) |
| | Sentinel-1 | SAFE/TIFF | 1,3 PB (GRD products, 2018/04) 1,2 PB (SLC products, 2018/04) |
| | Sentinel-2 | SAFE/JPEG2000 (INSPIRE) | 10,5 PB (2021) |
| | Sentinel-3 | NetCDF 4 | 10,5 PB (2021) |
| Emergency & Security | Pleiades | DIMAP V2 | |
| | EMS | netCDF, GRIB, PCraster, WaterML, geotiff, geopdf, jpeg, shapefiles, kml/kmz | 1,3 PB (2021) |
| | Sentinel-1 | SAFE/TIFF | |
| | Sentinel-2 | SAFE/JPEG2000 (INSPIRE) | 10,5 PB (2021) |
| | Sentinel-3 | NetCDF 4 | 10,5 PB (2021) |
| | Geohazards TEP | | |
| | Pleiades | DIMAP V2 | |

Figure 3.1.1 – Types, Formats & **Approximate** Volumes of Data

It is planned to operate more than 20 satellites in orbit at the same time with 11 different instrument families across the constellation. Note that as more Sentinel satellites are commissioned the daily data downlink volume will increase and is estimated to reach around 10 TB/day. Tenderer's proposed solutions are required to take account of this growth and its consequences for scaling requirements.

Tenderers are required to:

- provide a brief (no more than one page) summary of how they intend to account in their designs for the amount of data, the daily rate of increase of data, and the consequential need for scalability.

There are four main access points for Sentinel satellite data. They are:

- At ESA:
 - o SciHub (DHUS) (diashub.copernicus.eu, diashub2.copernicus.eu and diashub3.copernicus.eu)

- Copernicus Space Component Data Access (CSCDA) (status uncertain)
- At EUMETSAT:
 - EUMETCast (for near real-time data) (Sentinel 3 Marine only)
 - Copernicus Online Data Access (coda.eumetsat.int) (Sentinel 3 Marine only)
 - Online Data Access (OLDA) (olda.eumetsat.int)

The following public cloud providers distribute Sentinel 2 data only:

- Amazon AWS
- Google Cloud Platform GCP

Finally, the technical mechanisms for retrieving data from the above sources varies across:

- Manual Download
- Automatic Download
- Search & Download APIs
- Download Through Pre-packaged Libraries

The information above about access points is intended for illustration purposes only. Tenderers are required to rely on their own information gathering when preparing tenders.

3.1.3 Data Access Requirements

3.1.3.1 *Minimal Copying*

The volume of data is very large and growing and mirroring between the three sites has been ruled out as infeasible and unaffordable. Accordingly, it will be necessary to employ selective and intelligent caching and/or streaming in order to make best use of wide-area networking capacity and optimal use of local high-speed storage. Virtual machines and/or Containers will be created where the majority of data access will be local, but some wide-area access requiring caching is to be expected.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.

3.1.3.2 *License Enforcement & Access Control – Free Data*

The data may in most cases be available free of charge; however, end-users are still required to accept the licence terms attached to the data. Accordingly, it will be necessary to electronically collect licence acceptances from clearly identified end-users, to record these acceptances and to update the user's access credentials as required.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.

3.1.3.3 License Enforcement & Access Control – Commercial Data

Some of the data may be contributed on a commercial basis; it will be necessary to incorporate licence acceptance and/or access rights checking. It is expected that users subscribing to commercial data will pay the provider directly whereupon the user's credentials will be appropriately updated. Customer does not foresee the requirement for any payment mechanisms within the DIAS platform in this Iteration V1.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.

3.1.3.4 Traceability & Provenance

The data selected for processing by an end-user together with the tools he uses to process it must be recorded for the purpose of asserting data provenance (but with user personal data anonymised).

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.

3.1.3.5 European Commission Requirement - Usage Auditing

The European Commission requires to be able to analyse usage (uptake) patterns of the datasets and of the DIAS platform as a whole. Usage statistics will be required for all assets delivered under this ITT-2.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.

3.1.3.6 Caching

Data is stored at differing levels of accessibility. Some data is on tape and not in a tape silo, in other words, it is offline. Some data is on tape and in a tape silo – and is therefore nearline. And some data is on random access devices – and therefore online.

It will be necessary to have a notion of asynchronous access; but, Tenderers are expected to decide their own design and describe it in their High-Level Design.

For example (when accessed data is remote):

- online data might be locally cached and almost instantaneously available (subject to a transfer over the WAN) and remain available for a set period before being discarded locally
- nearline data might be locally cached and available with some delay (subject to an auto-load of a tape in the silo, a tape read, and a transfer over the WAN) and remain available for a set period before being discarded locally
- offline data might be locally cached with a scheduled delay (subject to a manual tape load to drive, a tape read, and a transfer over the WAN) and remain available for a set period before being discarded locally

It is expected that the caching mechanism will be sufficiently intelligent to only locally cache one copy of those items of data that are requested by multiple users. In such circumstances, its local destruction would be delayed to the end of the service guarantee period for the most recent requester. However, individual users would still be restricted to the window of access assigned to them. Tenderers may assume that sufficient local storage will be allocated to allow the cache to behave reasonably, and not to thrash.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how all the above caching requirements will be met.

3.1.3.7 API & Service Software – Brokered Location Hiding – Browsable/Searchable Data Catalogue

The principle components of the Data Access API are described below:

- The API and the service software implementing it will be published at three sites – ECMWF, EUMETSAT & Mercator Ocean International. It may also be published elsewhere in due course as additional data providers join the DIAS.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how they will structure their API service taking full account of the evolutionary requirement that they begin with the V0 API design.

- The Broker is a key component of the service software. This component is responsible for hiding the location of the data and the access protocol from the end-user; it also will implement Quality-of-Service (QoS) through the use of a queue.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how the Broker requirements will be met.

- The Catalogue is a key component of the service software. This component is responsible for providing a comprehensive inventory of available datasets that can be browsed and/or searched.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how the Catalogue requirements will be met.

NOTE: The Successful Tenderer is required to begin its work from the API, Metadata, and adapter specifications of DIAS Iteration V0, and to work in close collaboration with the Customer and with EUMETSAT when and if additions, deletions or other changes are believed to be required.

3.1.3.8 Framework & Adapters

There are a number of different formats in use among the various datasets that are to form the DIAS. All those currently in use are to be accommodated, and it must be relatively simple to add new ones. Accordingly, a framework-and-adapters approach is suggested, defined as:

- a single software framework that handles all of the generic services (cataloguing, caching, access control, etc ...)

- an internal API for inserting adapters that adapt the framework to a particular type of dataset, and
- an (extensible) suite of adapters that implement a number of different dataset types.

Tenderers should examine the Iteration V0 adapters code and consider whether it is suitable for re-use.

Tenderers may wish to consider the extract-transform-load (ETL) class of tools for some parts of this requirement – for example, NiFi and Talend. The choice of tool(s) remains entirely for Tenderers to decide.

Successful Tenderer is required to deliver:

- a framework as sketched above
- an internal API as sketched above, and
- a suite of data adapters that fit into their chosen framework and transform data at rest into their chosen on-the-wire transfer format. (Tenderers may wish to make use of the Iteration V0 code; this is entirely a matter for their discretion.)

With such a large collection of data from so many different sources, it is extremely important to have a rigorous naming, cataloguing and searching capability.

Successful Tenderer is required to deliver:

- a single namespace for all datasets (this namespace must be extensible to new data sources and/or types),
- a single vocabulary for sub-setting datasets by attributes (this vocabulary must be based on existing ones, in particular, [CF Conventions & Metadata](#)),
- a catalogue with complete and accurate descriptions of each and every dataset using INSPIRE ISO 19139, and
- a search facility providing high-quality search results based on user specified criteria for any piece of metadata associated with any dataset.

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how their framework-and-adapters design will be implemented.

3.1.3.9 *API General Characteristics*

Successful Tenderer is required to deliver:

- a RESTful API design and implementation of the Data Access API & Browseable/Searchable Catalogue,
- a facility in their API to allow subsets of data to be delivered based on JSON templates,

- a Quality-of-Service (QoS) capability in their API together with high-availability and elastic scaling to prevent service outages and/or degradation under heavy user demand,
- a Secure API based on the HTTPS protocol for all requests (data transfers may use HTTP subject to licence restrictions).

Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how the API General Characteristics requirements will be met.

3.2 (WP2) - Virtual Machine / Container Images & *Browsable/Searchable Image Catalogue*

This section describes in detail the Virtual Machine Images (VMIs) and/or Container Images (Cis), the associated browse/search catalogue and the instantiate/terminate capability that the Successful Tenderer is required to deliver. For the sake of brevity, and where it makes sense, the terms VMIs or Virtual Machine Images shall also include Cis or Container Images.

Successful Tenderer is required to deliver two distinct types of Virtual Machine Images;

- The first type is intended for direct use by end users (possibly as a virtual desktop or via SSH and the command line) – these will be referred to as Compute Node Images.
- The second type is intended to aid the construction of special purpose clusters that will be used indirectly by end users (i.e. they will not usually routinely log-in to such running images). These will be referred to as Cluster Node Images. This section deals with Compute Node Images. See Section 3.3 for further information about Cluster Node Images.

Successful Tenderer is required to deliver:

- a browsable and searchable catalogue of available Virtual Machine Images and/or Container Images,

The Iteration V1 DIAS infrastructure will be based on OpenStack with a Morpheus user interface. The base-level store from which images will be launched will be Glance with its limited metadata facilities. The image catalogue will provide richer metadata including communities of interest and software inventories and will point to the unique identifier for the relevant image in Glance. This catalogue will be exported via a RESTful API that can be accessed by custom code or a command line tool or by a modified GUI such as Horizon or Morpheus.

- a method for instantiating instances of Virtual Machine and/or Container Images, and
- a method for terminating instances of Virtual Machine and/or Container Images.

Successful Tenderer is required to deliver the browse, search, instantiate and terminate capabilities as a web service accessible through both an API, through a browser, and via a command line tool.

Successful Tenderer is required to incorporate in his Detailed Design the ability for the Customer and/or end-users to create their own Virtual Machine and/or Container Images and to register them with the Catalogue with possibly limited scope (for example, user organisation).

Tenderers are required to demonstrate in their High-Level Designs how they will meet all the requirements of this section.

3.2.1 Compute Node Images

The following list of requirements is extensive, and images that fulfil all of the requirements may be much larger than is necessary for the target communities of interest and/or disciplines.

While all software listed in Section 3.2.1.2 through Section 3.2.1.5 is mandatory, this does not mean that every image must contain every piece of software.

Tenderers are required to:

- propose appropriate subsets in their tenders that collectively form a tailored suite,
- provide a rationale for each choice,
- provide a cross-reference demonstrating the inclusion of every piece of software in at least one image,
- provide a list of known-at-the-time conflicts that may prevent certain combinations of software and/or Linux distributions.

Prior to implementation by Successful Tenderer, the precise content of each image will be agreed with the Customer who may require adjustments to be made.

Successful Tenderer is required to size each image, provide appropriate instance flavour settings and automatically enter this combined information into a consolidated Catalogue of available images.

Figure 3.2.1.1 gives an example illustrated view of the top-level requirements for Compute Node Images. It is advisory only; tenderers are responsible for deciding what makes best sense when designing their individual Virtual Machine Images (VMIs).

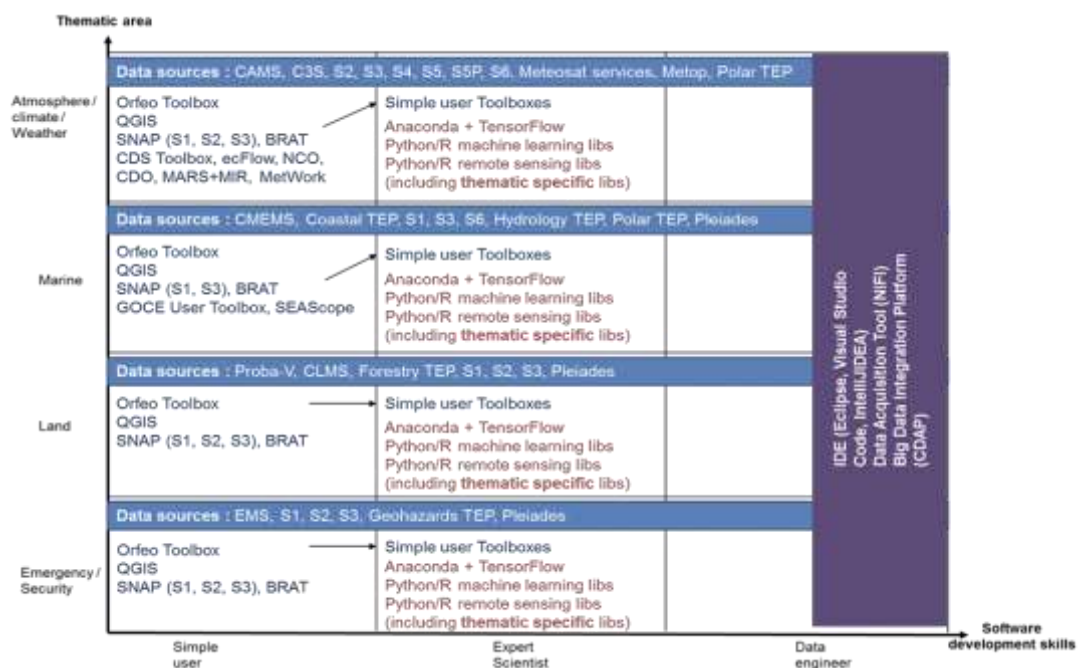


Figure 3.2.1.1 – VMI – Thematic Area versus Software Development Skills

3.2.1.1 General Requirements

Successful Tenderer is required to:

- design, develop, implement, test and deliver a suite of images for Catalogue publication globally across the WEkEO (DIAS) cloud. These images shall contain standard suites of software sufficient to address the needs of three broad groups of users, namely:
 - domain experts across the spectrum of DIAS applicability
 - software developers across the spectrum of sophistication
 - platform administrators and DevOps engineers
- ensure that it is possible for the Customer and/or end-users to customise these images to their own individual requirements and to re-publish them suitably renamed either to their local scope within the catalogue or to the global catalogue scope subject to appropriate authorisation from platform administrators
- ensure that where end-users wish to create their own custom images from scratch, they are able to make use of any tools that Successful Tenderer uses in the construction of his images. Again, publication either to the local scope or to the global scope subject to authorisation shall be possible
- make appropriate use of automation in the construction of Virtual Machine Images. This shall be sufficient to ensure freedom from manual keying errors and avoidance of unnecessary repetition. See Section 3.4 for more details
- install the Data Access API client software if appropriate on all images irrespective of their target user group or discipline
- ensure that upon instantiation of a virtual machine and/or container the client API is configured with the user's credentials so that the user does not have to login to the API in addition to logging in to the instance
- ensure that where possible and appropriate pre-installed software packages are modified and/or extended to be aware of the Data Access API, so that the user does not need to configure anything

Tenderers are required to demonstrate in their High-Level Designs how they will meet all the above General Requirements.

3.2.1.2 Meteorological & Geographic Processing Software Packages

Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):

- all Science Toolbox Exploitation Platform (STEP) SNAP toolboxes for the Sentinel missions
- the Orfeo Toolbox (OTB) and by extension the ITK Image Segmentation & Registration Toolkit
- the Broadview Altimetry Toolbox (BRAT)

- Panopoly
- ToolsUI
- the QGIS Geographic Information System software
- EO Workbench (EOPaaS)

3.2.1.3 Languages, Editors & IDEs

Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):

- the Python and R programming language tools. Tenderers may add additional languages as required by his design. Both Python 2.7 and Python 3 are to be provided.
- Interactive versions of Python (Jupyter) and R (RStudio) through the Anaconda (Python Distribution)
- the Apache Zeppelin web-based notebook (tenderer may use the docker image within a VMI/CI if he so wishes)
- the following IDEs / Editors:
 - Eclipse, IntelliJ IDEA (Community Edition), Visual Studio Code, Atom Editor and Brackets Editor

3.2.1.4 Python & R Essential Libraries

Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):

NOTE: Where other packages are required as dependencies, or where Tenderers believe there are important gaps, these lists should be extended by the Tenderers, and the additions made explicit in their Tenders.

- essential Python libraries and tools for data manipulation, optimized computation and visualization:
 - SciPy, Numpy, Pandas, IPython, Matplotlib, Seaborn, Bokeh and SciKit-learn
- essential R libraries and tools for data manipulation, optimized computation and visualization:
 - Caret, Kernlab, randomForest, nnet, e1071, klaR, tree, rpart, MICE, Lasso2, Lars, Gbm, BayesTree
- Python remote sensing libraries:
 - GDAL, Georasters, GeoPanda, CLIMAF, Pcdi_metrics, Fiona, Fmask, Shapely, Cartopy, Rtree, PySal, NanSat, SatPy
- R remote sensing libraries:

- Sp, Rgdal, Fields, Maptools, Rstoolbox, Sf, Rasters, Gdistance, Geosphere
- Copernicus / Sentinel libraries and utilities:
 - Sentinelsat, Sentinel2ProductIngestor, Sat-downloader, Sat-api, Sentinel-hub, Peps-download

3.2.1.5 Machine Learning / Deep Learning / Artificial Intelligence

Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):

- Include the following deep learning package:
 - TensorFlow

3.2.1.6 Operating Systems

Successful Tenderer is required to:

- Implement **ALL** the installed/configured software requirements on **EACH** of the following Linux distributions:

NOTE: If during implementation a particular Virtual Machine Image proves impossible to implement on a particular Linux distribution, Successful Tenderer shall inform Customer and agree a compromise or exemption.

- RedHat 7.5-1804 / CentOS 7.5-1804 (or latest stable release at time of development)
- Ubuntu 16.04 (Xenial Xerus) / Ubuntu 18.04 (Bionic Beaver) (or latest stable release at time of development)
- Debian 8 (Jessie) / Debian 9 (Stretch) (or latest stable release at time of development)
- OpenSUSE 15.0 (Leap) (or latest stable release at time of development)
- Scientific Linux 7.5 (or latest stable release at time of development)
- Include a standard set of Linux system administration tools of the kind typically found on a Linux Desktop Machine or on a Linux Server Machine
- Include appropriate desktop environment options according to the Linux distribution in use
- ensure that all package management client software is loaded by default and configured to use the local artefacts store (see Section 3.4)

3.2.1.7 Security

Successful Tenderer is required to:

- ensure that the only method of access to images is through a user's SSH keys (the public key for which is pre-loaded to the Cloud). Username and password access is forbidden except where GUI-based access mandates otherwise. (This is a pragmatic exemption that may change in the future.)
- ensure that user credentials for the Data Access API & Catalogue are seamlessly added to each virtual machine upon instantiation
- ensure that all system and/or vendor accounts (including default accounts and public keys) that are not required are removed or disabled where removal is infeasible.
- ensure that all system and/or vendor accounts that are required have their passwords changed to non-default values.
- ensure that all software applications that can function without root access do so.
- ensure that there are no plaintext passwords or other forms of access credentials hard-coded into applications, programs or scripts or stored in filesystems, databases or version control systems
- ensure that only required software is installed and/or enabled in all cases **except** where a Virtual Machine Image is intended for general-purpose use (for example, a general-purpose scientific or administrative workstation image)
- ensure that any standalone asset shall be capable of enforcing and maintaining an adequate access control policy (for example, a Linux VM with no external directory service relying entirely on local permission management),
- ensure that images are configured to give instance owners sudo rights. (Where an image is normally only accessible through SSH keys, the user password required to exercise sudo rights shall be transmitted to the user by the DIAS platform **once** and **once** only. User shall then be obliged to change said password on first use.)
- ensure that the principle of least privilege and need-to-know is implemented by default in all images using an appropriate firewall capability within the image and a corresponding cloud Infrastructure Security Group. Where individual tools or libraries call for less strict security, this shall be pre-configured within the image firewall and an appropriate cloud Security Group created and associated with the image.
- ensure that an automated suite of settings is applied to all Linux images during creation to ensure that they conform to current best security practice. At minimum this suite shall:
 - remove all unnecessary software and data
 - remove unnecessary access credentials
 - disable unnecessary services
 - deactivate unused features
 - disable unused network ports
 - forbid anonymous access

As an example, Tenderers may wish to use a tool that applies Security Technical Implementation Guide (STIG) to all images such as [Ansible Hardening](#) or OpenSCAP that is an implementation of the Security Content Automation Protocol (SCAP). Tenderers are required to make their own judgements about the best tool(s) to use for their design and should describe their choice(s) in their tenders.

NOTE: It is not required to repeat this process at instance creation or at any time thereafter.

3.2.1.8 Network Integration

Successful Tenderer is required to:

- ensure that the Virtual Machine Images delivered use appropriate methods from the underlying Cloud infrastructure or user's environment to configure DNS, NTP and authentication (e.g. cloud-init, Morpheus and OpenStack or user overrides)

3.3 (WP3) - Big Data Analytics Platform (Cluster Node Images)

Aggregate data volumes across the partner organisations alone are substantial and growing fast. At the end of 2016 there was a total of 9.6 Petabytes. This is expected to grow to a total of 65 Petabytes by the end of 2018 and to 184 Petabytes by the end of 2021.

While no single processing requirement is ever likely to span all data, it is expected that some processing requirements will operate on at least tens of Terabytes of data.

In addition, some data sources are offline (i.e. on tape or other passive media), some are nearline (i.e. tape silos), some are online (i.e. disks and SSDs) and some are or may eventually be real-time. The products of WP1 will disguise this distinction using a Common Data API; however, it cannot hide the differential bandwidth and/or latency with which the various data types become available.

At this Iteration V1 of the DIAS it is unclear precisely what sorts of processing requirements may arise among the expected user community and across the different types of data. For this reason, Customer wishes to put in place sufficient big-data basic building blocks to allow users to start working in the big-data knowledge space. This in turn will promote a better understanding of the requirements, leading to small platform changes during the lifecycle of Iteration V1 or larger platform changes during the lifecycles of Iterations V2 and V3.

3.3.1 The Lambda Architecture

The standard approach to simultaneous big-data processing of offline, online and real-time data is to use the so-called [Lambda Architecture](#) (see Wikipedia).

The Lambda Architecture is:

- a batch layer for processing pre-stored, non-real-time data plus a speed layer for processing newly incoming real-time data, plus a serving layer for combining results for user consumption

where:

- processing inaccuracies in the speed layer are eventually corrected by subsequent processing of the same data by the batch layer, and inaccurate or incomplete results in the serving layer are replaced by accurate and complete results.

A simple diagram showing the batch and speed processing branches and typical technologies used in those branches is shown in Figure 3.3.1.1.

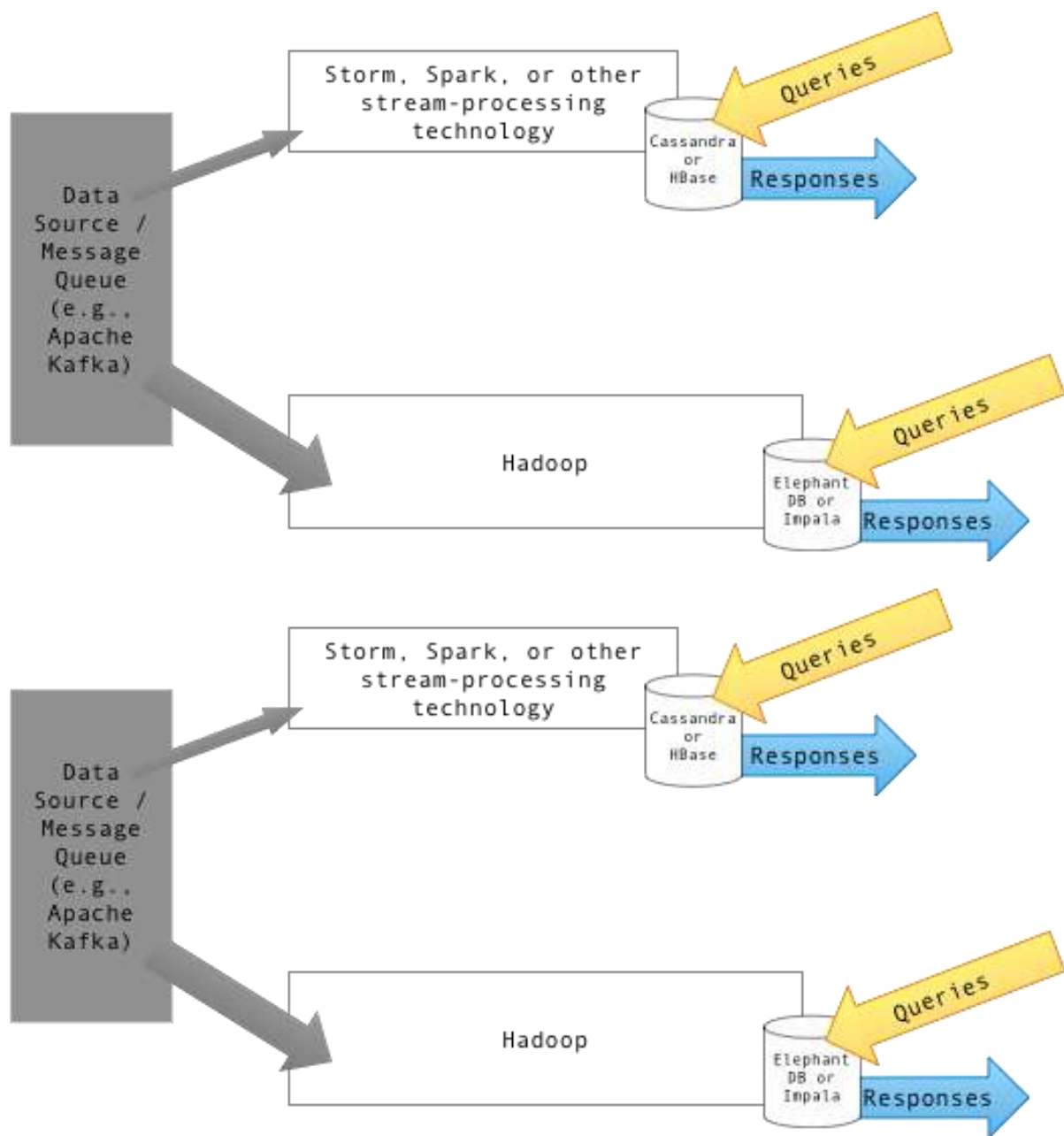


Figure 3.3.1.1: Example Lambda Architecture Configuration

By Textractor - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=34963987>

A second diagram in Figure 3.3.1.2 also shows the serving layer.

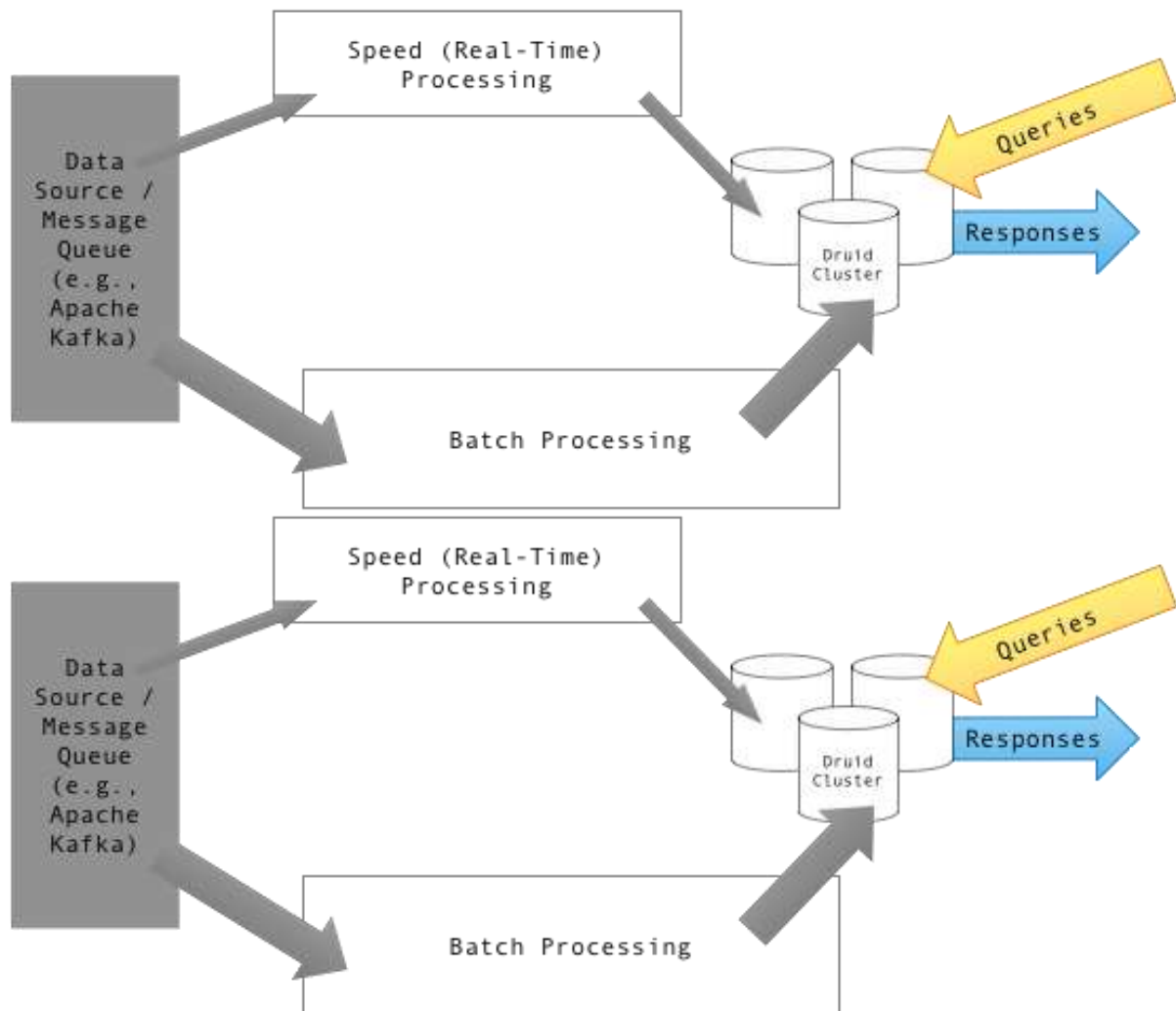


Figure 3.3.1.2: Example Lambda Architecture Configuration with Serving Layer

By Textractor - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=34963987>

3.3.2 The Processing Components

The core processing functionality that the Customer requires to have on the platform is Apache Spark.

Tenderers are required to deliver:

- a recommended distribution or set of distributions of Apache Spark. This may be a single distribution or a combination of several distributions. It is important to illustrate in tenders the benefits and limitations of each distribution in regard to, but not limited to, such matters as:
 - cost – what are the total lifecycle cost including any per-user component(s)?
 - support – how will support work from our end-user’s viewpoint?
 - functionality – what other tools and sub-systems are integrated with the product?
 - technical specifications – how complete is the product?

- currency of distribution relative to upstream open source version – is the vendor a key participant in the upstream development process and does his product stay close to the upstream functionality? (In answering this, tenderers should be aware that Customer is trying to assess the extent to which an offering has become or is becoming proprietary.)

Tenderers should note that the Customer's purpose of calling for several distributions is to allow the tenderer to propose:

- least cost (possibly free) distributions for users who do not wish to pay for commercial support and/or features
- leading-edge distributions for big data researchers developing new approaches to DIAS data processing that require the latest Apache Spark features
- commercially supported and featured distributions for users who require them such as those responsible for producing production quality operational data

Successful Tenderer is required to deliver:

- a suite of Virtual Machine Images and/or container images designed and implemented to provide the above distributions and suitable for building the following environments:
 - single node developer environments
 - small-scale multi-node clustered developer environments – these are intended for clustered software testing and not for big data handling. They are not likely to be interfaced to any massive-scale NoSQL databases or clustered filesystems.
 - Medium-scale multi-node clustered data science environments – these are intended for data scientist to begin familiarisation with big data processing with Apache Spark. They may require a shared big data filesystem or database to be present.
 - **(Optional)** Large-scale multi-node clustered data science environments intended for production of operational data. In accordance with our basic building blocks principle, we are unable at this time to fully specify the needs of such an environment.

The images are to be designed and implemented using the approach described in Section 3.2.

- a set of automation mechanisms for creating clusters (where appropriate) using the above Virtual Machine Images and/or Container Images for the single, small-scale and medium-scale cases and **(Optional)** for the large-scale case. These automations are to be integrated into the Build Automation (DevOps & CI/CD) system described in Section 3.4.

Tenderers are required to use:

- HashiCorp Terraform together with his choice of configuration management mechanism to achieve this for Virtual Machine Images
- Docker or Kubernetes together with his choice of configuration management mechanism to achieve this for Container Images

... and to augment these with appropriate configuration management tools where appropriate.

3.3.3 The Storage Components

Customer does not propose to establish at this stage a cloud-wide big-data filesystem or NoSQL data store, since it is unclear whether this is either wise or useful. However, Customer does want to gain experience with the technologies that are used to build such systems.

Also, Customer is unable at the time to be certain of whether there will be server capacity off-cloud, or on-cloud in passthrough mode, to build, for example, a Ceph or Cassandra cluster. And so, Customer proposes to gain experience using the facilities that it knows it can rely on having, namely compute instances, large attachable volumes of virtual disk, and high-speed private and public networks. This will not provide an environment suitable for performance measurement or capacity planning; however, it will allow architecture planning and functionality testing.

As a final restriction on scope, and in the interests of de-risking the platform, Customer does not propose at this time to permit the construction of large-scale cluster that are shared any more widely than a cloud tenancy. This may be relaxed once the security and provisioning mechanisms for cross-tenancy working are better understood; but that is a matter for Iteration V2 or later.

With the above limitations in mind, Successful Tenderer is required to provide:

- distributions of Ceph and Cassandra in free open source software (FOSS) form only
- Ceph and Cassandra build mechanisms suitable for generating the following environments:
 - o minimal environments that present the full product API but run on minimum resource (e.g. a laptop, a desktop or a virtual desktop)
 - o small-scale clusters that present the full product API but run on a sufficient number of resources to enable cluster functions to be investigated and tested
 - o **(Optional)** large-scale clusters if there are sufficient differences from the small-scale cluster case above

Tenderers are required to provide:

- a High-Level Design for their proposed Ceph and Cassandra environments.

3.3.4 Security

This section differs only slightly from Section 3.2.1.7; however, it is included so that the Work Package can stand alone.

Successful Tenderer is required to:

- ensure that the only method of access to images is through a user's SSH keys (the public key for which is pre-loaded to the Cloud). Username and password access is forbidden.

- ensure that all system and/or vendor accounts (including default accounts and public keys) that are not required are removed or disabled where removal is infeasible.
- ensure that all system and/or vendor accounts that are required have their passwords changed to non-default values.
- ensure that all software applications that can function without root access do so.
- ensure that there are no plaintext passwords or other forms of access credentials hard-coded into applications, programs or scripts or stored in filesystems, databases or version control systems
- ensure that only required software is installed and/or enabled in all cases **except** where a Virtual Machine Image is intended for general explorative use (for example, a general purpose scientific workstation image)
- ensure that any standalone asset shall be capable of enforcing and maintaining an adequate access control policy (for example, a Linux VM with no external directory service relying entirely on local permission management), (This situation may very well arise in the case of Virtual Machine Images designed as nodes for Ceph, Cassandra, Kubernetes or any other clustered system)
- ensure that images are configured to give instance owners sudo rights; however, a mechanism must be put in place to ensure that the user sets/resets the password within a short time from creation of the VM,
- ensure that the principle of least privilege and need-to-know is implemented by default in all images using an appropriate firewall capability. Corresponding Cloud Infrastructure Security Groups shall be delivered to further enforce these restrictions within the cloud infrastructure. Where individual tools or libraries call for less strict security, this will be controlled through the automated application of appropriate firewall and security group rules.
- ensure that an automated suite of settings is applied to all Linux images to ensure that they conform to current best security practice. At minimum this suite shall:
 - remove all unnecessary software and data
 - remove unnecessary access credentials
 - disable unnecessary services
 - deactivate unused features
 - disable unused network ports
 - forbid anonymous access

As an example, tenderers may wish to use a tool that applies Security Technical Implementation Guide (STIG) to all images such as [Ansible Hardening](#) or OpenSCAP that is an implementation of the Security Content Automation Protocol (SCAP). Tenderers are required to make their own judgements about the best tool(s) to use for their design and should describe their choice(s) in their tenders.

3.4 (WP4) – Build Automation (DevOps & CI/CD)

This ITT-2 calls for the delivery of a large number of software products brought together into a coherent whole for publication to potentially thousands of users.

The three **essential architectural pillars** of the processing tools of the DIAS are:

- **Repeatability** – the guarantee of a consistent user experience across similar workflows and across time,
- **Scalability** – the guarantee of a consistent quality of service and containable cost of support irrespective of the eventual size of the user community,
- **Archiving** - given the broad scope of free open source products that are envisaged as forming the constituent parts of the platform, it is vital that any and all versions of such products that have been used remain available for the lifecycle of the platform – otherwise repeatability is lost.

In the following requirements:

- original artefacts are any pieces of work that are created and delivered to the platform by the Successful Tenderer and not otherwise obtainable by, for example, download from the Internet.
- pre-existing artefacts are any pieces of work that have been created by third-parties and delivered to the platform by the Successful Tenderer and are, in normal circumstances, obtainable by, for example, download from the Internet.

3.4.1 DevOps Toolchain

Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:

- a DevOps toolchain that enables continuous integration and continuous delivery of all platform artefacts in a manner that is repeatable and scales well to tens of thousands of users

3.4.2 Shared Archive – Original Artefacts

Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:

- a shared archive of all original artefacts generated and required to (re-)build the platform from scratch (this also includes any user-created products that have been centrally registered with the DevOps toolchain)

(The following is an incomplete list of examples of such artefacts: compiled object files, archive files, image files, and other binary large objects (BLOBs) such as test data.)

3.4.3 Shared Archive – Pre-existing Artefacts

Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:

- a shared archive of all pre-existing (internet downloaded and vendor produced) artefacts required to (re-)build the platform from scratch including any products that are dependencies of centrally registered user-created products

(The following is an incomplete list of examples of such artefacts: ISO images, Java packages, Python packages, baseline cloud images.)

3.4.4 Shared Version Control System

Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:

- a shared version control system for storage of all production versions of original artefacts in source format (code and related documentation such as READMEs and INSTALLs)

(Storage of large objects under version control is strongly discouraged.)

3.4.5 DevOps CI/CD Client Tools

Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:

- a simple automated mechanism whereby a user can add all the DevOps CI/CD client tools to his own environment whether it be a VM from Section 3.2 or a local workstation/laptop environment running an appropriate version of Linux. This is primarily intended to automate away user errors during initial setup – thereby saving on support cases. Tenderers may find it sufficient to achieve this with a simple idempotent script.

3.4.6 Simple Site Replication

Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:

- A simple mechanism to ensure that the three primary DIAS sites receive all changes to the DevOps CI/CD content systems (i.e. version control and archive). This may be achieved on an eventually consistent basis as long as catalogues are updated only after content is fully consistent. Tenderers may find it sufficient to achieve this with very simple rsync-based replication provided the target can never be observed in an inconsistent state.

3.4.7 Technology Options

There follows a list of technologies and tools that exemplify possible solutions for the requirements described above. As discussed in Section 1.9, it is not to be taken as a prescriptive list. Any suitable technologies and tools may be submitted for evaluation. The purpose of this section is simply to clarify

the sort of technologies and tools that this ITT-2 is referring to in each area of the above requirements list.

3.4.7.1 Planning

A tool such as Atlassian Jira is very popular across a wide spectrum of user types. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it.

It should be noted that Customer uses an enterprise-wide implementation of Jira.

3.4.7.2 Code Creation

As source code is created it should be submitted to version control often and in small chunks; holding new work in local repositories over long periods without pushing it to a central repository is strongly discouraged.

If Git were to be proposed then a tool such as GitLab (for on premises) or GitHub (for off premises) is commonly used as a sharing repository. Whichever tool Tenderer choose, they are required to describe the capabilities of the tool and how they intend to make use of it.

It should be noted that Customer uses GitLab on some of its project and has plans to publish some of its works using GitHub.

3.4.7.3 Code Review

There are a number of options in this area and Tenderers may propose any suitable candidate. The features the Customer would prefer to see include: integration with source code control, source code browsing, code annotation by reviewers, integration with e-mail and/or messaging.

Static security analysis of code might be done with tools such tools as AppScan Source, BlackDuck, Fortify, Veracode, Snyk and Sonatype Checkmarx.

Static security analysis of code (mainly Java and/or C++) might be done with such tools as Coverity, PMD, JDepend, Checkstyle, Findbugs, Emma, Cobertura, SonarQube and Klocwork.

Whichever tools Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it.

3.4.7.4 Documentation

Wiki-based documentation is strongly preferred (it is not necessary to produce book quality printed versions of the documentation). Commonly, this is achieved using commercial offerings such as Atlassian Confluence or open source offerings such as MediaWiki. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it.

It should be noted that the Customer makes enterprise-wide extensive use of Atlassian Confluence.

3.4.7.5 Build

An open source Jenkins solution might typically be chosen along with appropriate plugins and local customisation. Whichever tool the Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it.

Customer has a strong preference for an simple solution rather than a comprehensive solution.

3.4.7.6 *Testing*

There are a number of options for testing such as Mockito, Gatling, Spock, Arquillian, wiremock, Cucumber, JMeter, JUnit and PowerMock (to some extent depending on language to be covered).

Whichever tools Tenderers choose, they are required to describe the capabilities of the tools and how and for which languages they plan to use them.

3.4.7.7 *Binary Large Object Archiving*

A commonly chosen commercial offering in this area is Artefactory. A community edition exists; however, it is severely limited in the BLOB types it understands. An alternative would be Sonatype Nexus OSS which has less stringent limitations. Whichever tool the Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it.

The essential purpose of this capability is to make the system self-contained so that all artefacts can be re-created without reference to Internet sources, i.e. it serves a part of the archive requirement.

This ITT-2 requires the delivery of a number of customised cloud images and cluster configurations, and it is to be expected that a number of automation tools will be used at the various different stages of preparing these images. There follow some examples of the kind of tools expected to be found in tenders. These examples are **not** prescriptive; Tenderers are free to propose their own choices. What is vital though is that the end result does not involve extensive use of command line actions or lengthy GUI interactions to (re-)create any software artefacts delivered under this ITT-2 – repeatability through automation is paramount.

3.4.7.8 *Image Creation & Manipulation*

HashiCorp Packer – a multi-platform image creation tool. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it.

3.4.7.9 *Infrastructure as Code*

HashiCorp Terraform – a multi-provider Infrastructure-as-Code (IaC) provisioning tool. Whichever Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it.

3.4.7.10 *Portable Work Environments*

HashiCorp Vagrant – a multi-virtualisation-platform orchestrator for small virtual machine clusters ideal for mobile developer and demonstrator work environments. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it.

3.4.7.11 *Configuration Management*

There are a number of tools in this area. In alphabetical order, the most well-known are: Ansible, Chef, Puppet and Salt. Again, the tenderer must decide what works best for his proposed design. Tenderer should **note** though that Customer has a large installed base

of Puppet configuration management capabilities and significant experience of developing and using Puppet built up over many years; while the tenderer is free to choose a tool other than Puppet, he will need to present strong arguments for doing so. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it.

It is essential that Tenderers avail themselves of all design information regarding the underlying infrastructure being delivered through ITT-1 (EUMETSAT). This will ensure that there is a seamless integration between any tools delivered under this work package and the capabilities of the infrastructure being delivered under ITT-1. Tenderers are required to accept full responsibility for delivering this integration on the ITT-2 side – save that Customer will facilitate communications with EUMETSAT and MERCATOR and their contractors – see CFI-1 in Section 1.10.

3.5 (WP5) – Testing

Testing of the DIAS shall have two equally important purposes:

- it will be used as the basis for Final System Acceptance (FSA), and
- it will be used for ongoing verification of the DIAS for the entire duration of its support to operational use (See Section 3.8).

3.5.1 Testing Requirements

3.5.1.1 Unit Tests

The Successful Tenderer is required to develop a suite of unit tests sufficient to cover all units of software developed by him as a deliverable under this ITT-2.

3.5.1.2 Integration Tests

The Successful Tenderer is required to develop a suite of integration tests sufficient to cover all software integrations developed by him as a deliverable under this ITT-2.

3.5.1.3 Functional Tests

The Successful Tenderer is required to develop a suite of functional tests sufficient to demonstrate correct operation of a suite of typical user and administrator use cases.

3.5.1.4 Testing Framework

The Successful Tenderer is required to deliver all tests fully integrated into a testing framework. This test framework and the tests shall be updated during the support period to ensure that there is no regression of corrected faults. A simple framework should be chosen sufficient to meet these requirements; use of the framework should not require continuous maintenance or support.

3.5.1.5 Simple Testing Schedule

The Successful Tenderer is required to ensure that a full run of all tests is initiated by the DevOps CI/CD system described in Section 3.4 on at least a weekly schedule. A sophisticated testing schedule may be required at Iteration V2 or later; however, at Iteration V1 this is not needed.

3.5.1.6 Easy Extension

The Successful Tenderer is required to ensure that platform developers may submit their own test cases at will.

3.5.1.7 Results Feedback

The Successful Tenderer is required to ensure that testing feedback is provided to platform developers as quickly as possible to facilitate speedy corrections.

3.5.1.8 Multi-site Replication

The Successful Tenderer is required to replicate the test environment to the three production DIAS sites to be delivered by EUMETSAT under ITT-1.

Tenderers are required to show in their High-Level Designs how they will implement the above replication requirement.

Tenderers are required to acknowledge in their tenders their agreement that Final System Acceptance testing will be conducted on all three of the production sites.

The Successful Tenderer is required to conduct Customer witnessed Final System Acceptance testing on all three production sites and to deliver to Customer a test report in each case.

The Successful Tenderer is required to support a campaign of site testing during the support phase in the case that new sites become available.

It is not anticipated that end-users will make use of the testing framework unless they are developing a tool or library that is expected to become a standard part of the DIAS platform. Instead, they will make their own arrangements for testing. This should limit the amount of testing traffic generated by the user community; note though that they will not be precluded from submitting tests.

3.5.2 Tests & Testing Framework Timescales & Purpose

Successful Tenderer is required to complete the specification and development of their testing framework and associated tests and submit them to Customer for approval by Milestone WP5M1.

Customer will review and accept/reject the work products by Milestone WP5M2. Tenderer will then have 10 working days to make good any deficiencies and re-submit the work products.

Final System Acceptance and delivery of test reports will take place at Milestone WP5M3 and shall last up to 5 working days.

Once Customer issues a certificate of successful Final System Acceptance (FSA) Successful Tenderer is required to enter the support period and the testing framework and tests shall become part of the operational environment and shall be maintained and used to validate changes and prevent regression.

Tenderers are required to include the testing framework in their High-Level Design which is required to be delivered as a part of their tender.

3.6 (WP6) – Documentation

The DIAS platform calls for the development of an API and server capability, the integration of various open source software packages, the possible integration of various proprietary software packages and the development of various Virtual Machine Images and/or Container Images. Accordingly, a hybrid approach to the delivery of documentation should be employed.

3.6.1 General Notes on Production & Delivery of Documentation

The Successful Tenderer is required to:

- present all newly generated documentation as Wiki pages and in PDF and/or Word formats
- re-use open source documentation where possible – but only in the form of Wiki pages and PDF/Word formats where available
 - o external links are strongly discouraged
 - o other content types are discouraged but not prohibited (i.e. .doc, .xls, .pdf, .epub)
- re-use vendor documentation where possible – but only in the form of Wiki pages and PDF/Word formats where available
 - o external links are strongly discouraged
 - o other content types are discouraged but not prohibited (i.e. .doc, .xls, .pdf, .epub)
- PDF/Word formats should be supplied where available via links within appropriate Wiki pages

3.6.2 General Description of Approach

Tenderers are required to:

- Provide a description of their approach to the preparation of documentation covering:
 - o Tools to be used
 - o Processes to be used for quality assurance
 - o Handling of the various original sources of documentation
 - o Deployment to the DIAS platform

3.6.3 Tenderer Developed Software Packages & Tools

The following requirements apply per software package or tool.

3.6.3.1 High-Level Description

Successful Tenderer is required to deliver a high-level description of the software product. This guide shall describe at a high-level what it does and briefly how it does it,

3.6.3.2 Detailed Description

Successful Tenderer is required to deliver a detailed description of the theory of operation of the software product together with a guide to how the functionality is spread across the various source code components of the product, (for the sake of clarity, this requirement only applies to Successful Tenderers' own developed code modules and code that he writes to integrate with other products)

3.6.3.3 Installation & Configuration Guide

Successful Tenderer is required to deliver an installation and configuration guide. This guide shall be sufficient for a system administrator or end-user (as appropriate) to install the software package and configure it for use,

3.6.3.4 User Guide

Successful Tenderer is required to deliver a user guide. This guide shall be limited to how to use the software (prior installation and configuration of said software is assumed),

3.6.3.5 Build Guide

Successful Tenderer is required to deliver a build guide. This guide shall show how to generate an installable binary of the software product from its corresponding source code,

3.6.3.6 Testing Guide

Successful Tenderer is required to deliver a testing guide. This guide shall show how to test the software product,

3.6.3.7 Source Repository

Successful Tenderer is required to deliver a source repository with all code needed to build the software product – dependencies to be dealt with as separate software products – tools to be installed separately as indicated by the build guide,

3.6.3.8 Security Hashes

Successful Tenderer is required to deliver hashes as appropriate to ensure the integrity of the software product,

3.6.3.9 Rights Assignment

Successful Tenderer should be aware of the intellectual property regime under which Customer (i.e. ECMWF) is operating. All rights in the software have to be assigned to Customer (i.e. ECMWF) and the European Commission.

3.6.4 Open Source Software Packages & Tools

The following requirements apply per software package or tool.

3.6.4.1 High-Level Description

Successful Tenderer is required to deliver a high-level description of the software product. This guide shall describe at a high-level what it does and briefly how it does it,

3.6.4.2 Installation & Configuration Guide

Successful Tenderer is required to deliver an installation and configuration guide. This guide shall be sufficient for a system administrator or end-user (as appropriate) to install the software package and configure it for use,

3.6.4.3 User Guide

Successful Tenderer is required to deliver a user guide. This guide shall be limited to how to use the software (prior installation and configuration of said software is assumed),

3.6.4.4 Build Guide

Successful Tenderer is required to deliver a build guide. This guide shall show how to generate an installable binary of the software product from its corresponding source code,

3.6.4.5 Testing Guide

Successful Tenderer is required to deliver a testing guide. This guide shall show how to test the software product,

3.6.4.6 Source Repository

Successful Tenderer is required to deliver a source repository with all code needed to build the software product – dependencies to be dealt with as separate software products – tools to be installed separately as indicated by the build guide,

3.6.4.7 Attributions

Successful Tenderer is required to deliver appropriate attribution where the work of others has been used including links to the original web site(s),

3.6.4.8 Security Hashes

Successful Tenderer is required to deliver hashes as appropriate to ensure the integrity of the software product.

3.6.4.9 Licenses

Successful Tenderer must demonstrate that all open source software packages and tools will not conflict with the intellectual property ownership rights' assignment of the software. Customer (i.e. ECMWF) will reject any software the license terms of which might be harmful to the overall intellectual property rights as required to be obtained by ECMWF and the European Commission.

3.6.5 Proprietary Software Packages & Tools

The following requirements apply per software package or tool.

3.6.5.1 High-Level Description

Successful Tenderer is required to deliver a high-level description of the software product. This guide shall describe at a high-level what it does and briefly how it does it,

3.6.5.2 OSM Documentation

Successful Tenderer is required to deliver all documentation without exception that he has received from the original software manufacturer whether on paper or electronic,

3.6.5.3 Implementation Document

Successful Tenderer is required to deliver an implementation document showing how he has integrated the software into the overall platform, and detailing all configuration information needed to re-implement his integration,

3.6.5.4 Licence Assignment(s)

Successful Tenderer is required to assign all licences as described in the terms and conditions.

3.6.5.5 Warranty Certificate(s)

Successful Tenderer is required to provide certificate(s) of warranty and support (where applicable) to the Customer with the rights to such warranty and support assigned to the Customer,

3.6.5.6 Online Licensing & Support Credentials

Successful Tenderer is required to provide access credentials to all online licensing and support associated with the software package to the Customer, and provide proof that he has notified the original software manufacturer of his actions and obtained their agreement to such access being provided,

3.6.5.7 Firm-Fixed-Price Per-Use or Per-User

Successful Tenderer is required to provide a firm-fixed-price Price Sheet, valid for the duration of Iteration V1, showing the price per-use or user as appropriate, for software packages that are so licensed. This shall also be provided online and prominently displayed when a user first tries to access the package.

3.7 (WP7) – Training

3.7.1 General Training Materials Requirements

Successful Tenderer is required to deliver all training materials in accordance with the following criteria:

- suitable for self-study,
- indexed from a central training catalogue page.

3.7.2 General Description of Approach

Tenderers are required to:

- Provide a description of their approach to the preparation of training materials covering:
 - o Tools to be used
 - o Processes to be used for quality assurance
 - o Deployment to the DIAS platform

3.7.3 Specific Training Materials Requirements

3.7.3.1 *Use of the data access API.*

Successful Tenderer is required to produce a guide that shall give an introduction to the catalogue of data, to searching and to obtaining data with simple HTTPS/HTTP requests

Successful Tenderer is required to produce a guide that shall also take the user through a small number of examples of accessing the catalogue, selecting a small number of data sets, and consuming those data sets with curl/wget/Jupyter and in a simple application coded in both Python and R

3.7.3.2 *Use of end-user Virtual Machine Images and/or Container Images.*

Successful Tenderer is required to produce a guide that shall give an introduction to the catalogue of images, to the associated flavours, security groups and required data volumes.

Successful Tenderer is required to produce a guide that shall also take the user through a small number of examples of accessing the catalogue, selecting a virtual machine image, selecting a flavour, selecting a security group, selecting any necessary data volumes and initiating a virtual machine.

Successful Tenderer is required to produce a guide that shall then take the user through basic user access to the new virtual machine using secure shell (SSH) and the tenderers chosen secure GUI access method (examples being, VNC through a tunnel or HPE RDS).

3.7.3.3 *Use of the pre-loaded tools on the various Compute Node Virtual Machine Images.*

Successful Tenderer is required to produce a guide that shall give a basic introduction to the main tools available in each of the virtual machine images.

Successful Tenderer is required to produce a guide that shall then take the user through basic use of each of those tools using pre-loaded data sets where appropriate.

3.7.3.4 Use of the pre-loaded big-data tools on the various Cluster Virtual Machine Images.

Successful Tenderer is required to produce a guide that shall give a basic introduction to the Apache Spark toolset.

Successful Tenderer is required to produce a guide that shall then take the user through basic use of Apache Spark on a single node and on a small cluster including installing and configuring the software (as necessary).

Successful Tenderer is required to produce a guide that shall also introduce the user to customising the supplied Virtual Machine Images and/or Container Images for achieving more sophisticated automatic deployments.

3.8 (WP8) – Support

3.8.1 Period of Support

The Successful Tenderer is required to maintain and support the products of work packages WP1 through WP7 during the Iteration V1 operational lifecycle which will last until the end of Q4 2019.

The tenderer is required to offer an option to extend this support period by a further two (2) periods of three (3) months each if subsequently requested by ECMWF.

3.8.2 General Requirements

Tenderers are required to describe in their tenders how they would construct a Support Operation satisfying the requirements of this work package.

3.8.3 Integrated Support Operation

Our partner, Mercator Ocean International has primary responsibility for user support and the helpdesk function. However, the Successful Tenderer is required to provide 2nd and 3rd line support where tickets raised at Mercator Ocean International helpdesk cannot be resolved there.

It is expected that Mercator Ocean will have the necessary filtering in place to separate tickets according to whether they relate to the infrastructure (ITT-1), the Common Data Access API, the Virtual Machine Images or the Big-Data Analytics component (ITT-2). However, it must not be assumed that they can perfectly perform this function or that there will not be cross-ITT fault cases. In cases where tickets are wrongly routed, the Successful Tenderer is required to co-operate with the ITT-1 contractor to achieve a satisfactory resolution of the user's problem.

3.8.4 Support Requirements

3.8.4.1 Triage System

Successful Tenderer is required to manage his support resources using a triage system, taking full account at minimum of numbers of users affected, time sensitivity of affected services, and quality of service, but in any case, to provide an initial response within four (4) working hours as per Section 3.8.4.13 (Time Zone).

3.8.4.2 Customer Access to Triage System

Successful Tenderer is required to provide Customer with access to his triage system for the purpose of evaluating its effectiveness,

3.8.4.3 Re-Prioritisation

Successful Tenderer is required to respond to reasonable requests for re-prioritisation of tickets by Customer,

3.8.4.4 Triage System Changes

Successful Tenderer is required to respond to reasonable requests for change to the triage system by Customer,

3.8.4.5 *Initial Reporting to Users*

Successful Tenderer is required to provide to the user an initial assessment of the cause of a fault together with an intended timeframe for resolution,

3.8.4.6 *Full Reporting to Users*

Successful Tenderer is required to inform the user when a timeframe is going to slip and why

and / or

inform the user when the fault is fixed, and also inform him of any actions that he must take to benefit from the fix such as re-building his own components,

3.8.4.7 *Workarounds*

Successful Tenderer is required to wherever possible provide the user with a workaround that will allow him to carry on with his work pending full resolution of his fault,

3.8.4.8 *Upstream Reporting*

Successful Tenderer is required to escalate fault resolutions to 'upstream' channels in the case of open source software. This is to ensure that faults found and fixes implemented remain in place after package updates. It is understood that there are a number of different regimes for reporting faults and proposing fixes and that it can take a while to have a fix accepted upstream,

3.8.4.9 *Vendor Reporting*

Successful Tenderer is required to escalate fault reports to vendors in the case of proprietary software and act as a go-between to update the user of progress on a timely basis – the user is never to be directly referred to the vendor. A support contract from a vendor does not excuse the ITT-2 contractor from his contractual responsibility to support the DIAS user community,

3.8.4.10 *Test Framework Extension*

Successful Tenderer is required to extend the scope of tests in the testing framework in Section 3.5 to ensure that to the greatest possible extent no regression takes place,

3.8.4.11 *User Satisfaction Data*

Successful Tenderer is required to collect user satisfaction feedback and together with all other available support data prepare an analysis, on a monthly basis, of support effectiveness. Tenderer is free to propose his own regime for this; however, the following are some suggested metrics:

- faults reported
- faults fixed
- faults deemed unfixable
- faults fixed and submitted upstream (open source software)
- faults fixed by vendors (proprietary software)
- time to fix statistics
- fault hotspots

- user community fault reporting statistics (anonymised to GDPR acceptable standards)

3.8.4.12 Skills Transfer to Customer

Successful Tenderer is required:

- to provide required training, documentation and support to two (2) Customer members of its staff or Contractors nominated by Customer, and
- to ensure that sufficient training on all aspects of the DIAS Iteration V1 Platform has been given to those staff to permit them to support and upgrade the platform as necessary from handover to Customer onward

3.8.4.13 Time Zone

Successful Tenderer will operate its support operation during normal business hours (08:30 to 17:00) Central European Time (CET).

3.8.4.14 Physical Presence

Successful Tenderer is required to establish the physical presence of the support function at his own premises for the duration of Iteration V1 support; Customer shall not provide premises, physical assets or personnel.

4 (WP9) - Management & Implementation

4.1 Effective Date of Contract (EDC) and Date of Final System Acceptance (FSA)

High-Level Planning:

- EDC - Customer plans to reach contract award / signature for ITT-2 no later than October 2018. However, Customer in its sole discretion, does not guarantee to award a contract by this date or at all. The date given is solely to guide project planning.,
- FSA – Customer wishes to achieve Final System Acceptance (FSA) by 31st March 2019,
- The support phase will cover the period from FSA until the end of 2019.

4.2 Implementation Plan

Tenderers are required to provide a detailed implementation plan of proposed activities for the duration of the Framework Agreement.

The Framework Agreement will consist of two separate Service Contracts, SC1 and SC2, as follows:

- SC1 will address all the development activities leading up to the Iteration V1 system delivery and Final System Acceptance by 31st March 2019, i.e, Work Packages 1 to 7.
- SC2 will address the operational support of the Iteration V1 operational lifecycle (Work Package 8), from Iteration V1 delivery to the end of 2019. The implementation plan shall define relevant milestones, designed as markers of demonstrable progress in development and/or quality of service delivery. Each milestone shall be accompanied by enough descriptive text to ensure that it is understood what is being delivered, where and when. Any milestones requiring feedback from Customer (for example, design and plan reviews) shall be clearly described as requiring such feedback. Milestone deliverables should be consistent with the technical requirements specified in Sections 3 and 5 of this ITT.

The following management aspects shall be described in the proposed implementation plan:

- tracking of planning,
- resources and performance (incl. Key Performance Indicators (KPIs)),
- quality assurance and control,
- risk planning and mitigation plan,
- communication management between ECMWF, WEkEO partners, stakeholders,
- conflict resolution,
- subcontractor management, and
- personal data management.

A list of subcontractors describing their contribution and key personnel, legal name and address shall be provided. The Tenderer shall describe how the Framework Agreement, in particular Clause 2.9 has been flowed down to all their subcontractors.

As part of the general project management description, the Tenderer shall include the following elements:

- Contractual obligations as described in the Framework Agreement Clauses 2.1, 2.3 and 2.4.
- Kick-off meeting at ECMWF in Reading, U.K.
- Monthly teleconferences with ECMWF and contributions to interaction with EUMETSAT-ECMWF-Mercator Ocean coordination meetings and other WEKEO contractor(s).
- Proposal for payment milestones following the template payment plan as specified in Volume V (cf. Annex 3C) and according to following payment schemes:
 - o SC1: Fixed Price
 - o SC2: Fee-based Fixed Price

Tenderers are required to propose the level of resources that they consider adequate to implement SC2. Customer shall have the option to adapt the level of resources before or during the execution of SC2, and based on fixed price agreed fees

| WP9 Contractual Obligations Template | | | | |
|---|--------------------|---------------|---|--|
| # | Responsible | Nature | Title | Due |
| D0.y.z-YYYYQQ | Tenderer | Report | Quarterly Implementation Report QQ YYYY <i>QQ YYYY being the previous quarter</i> | Quarterly on 15/01, 15/04, 15/07 and 15/10 |
| D0.y.z-YYYY | Tenderer | Report | Annual Implementation Report YYYY <i>YYYY being the Year n-1</i> | 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 <i>YYYY being the Year n-1</i> | Annually on 15/01 |
| D0.y.z-YYYY | Tenderer | Report | Draft Implementation plan YYYY <i>YYYY being the Year n+1</i> | Annually on 28/02 |
| D0.y.z-YYYY | Tenderer | Report | Finalised Implementation plan YYYY <i>YYYY being the Year n+1</i> | Annually on 31/10 |
| D0.y.z-YYYY | Tenderer | Other | Copy of prime contractor's general financial statements and audit report YYYY <i>YYYY being the Year n-1</i> | Annually |
| D0.y.z-YYYY | Tenderer | Other | Letter from auditor specific to C3S contract YYYY <i>YYYY being the Year n-1</i> | End of the contract |
| D0.y.z-YYYY | Tenderer | Other | Risk Analysis and Contingency Plan | |

| WP9 Milestones Template | | | | |
|-------------------------|-------------|--|-----------------------|-----------------------------------|
| # | Responsible | Title | Means of verification | Due |
| M0.y.z | Tenderer | Kick-Off meeting | Minutes of meeting | Month 1 |
| M0.y.z | Tenderer | Progress review meetings with ECMWF / Payment milestones | Minutes of meeting | per dates defined in payment plan |

4.3 Tenderer Proposed Team Structure

Customer wishes to leave tenderers as free as possible to structure their teams as they see fit. However, there follows a brief list of requirements that must be followed.

Tenderer are required to:

- appoint a Project Manager who has sole responsibility for the delivery of all deliverables required under this contract taking full account of agreed timescales, quality requirements and price
- appoint a Technical Leader who has responsibility together with the Project Manager for understanding and where necessary clarifying the requirements and producing a technical design for negotiation and agreement with Customer
- describe his proposed Project Team as follows:
 - o roles or functions
 - o minimum experience required and actual experience offered
 - o numbers of individuals in each role or function and duration / level of effort
 - o provide a summary organisation chart

4.4 Customer Project Team

Customer shall appoint a Technical Leader and a Contract Management Officer.

All formal communications with Customer concerning the contract shall be with the Contract Management Officer including formal deliverables.

All other communications with Customer concerning matters such as day-to-day project progress, refinement of technical specifications, facilitation requests, progress reports shall be with the Technical Leader with copies to the Contract Management Officer.

It is well understood that the ITT-2 contractor shall require efficient information flow with the ITT-1 contractor and with Mercator Ocean. Customer shall make all reasonable efforts to facilitate this information flow so as to ensure that the ITT-2 contractor remains able to make progress.

4.5 Detailed Design

Successful Tenderers is required to deliver to Customer by EDC + 2 weeks a Detailed Design showing for each deliverable (as applicable):

- o an architecture

- a component specification
- all required pre-cursors (i.e. libraries and tools to be used)
- delivery format of deliverable

Upon delivery of the Detailed Design to Customer there shall be a period of 2 weeks of review during which Customer shall have the right to require the Tenderer to make all reasonable efforts to accommodate modifications and/or additions.

5 Consolidated Requirements Matrices

This section contains a set of Consolidated Matrices of all the Requirements described in this ITT – one for each of Sections 1, 2 & 4 and one per work package for Section 3.

Tenderers are required to respond for each requirement in this matrix set with a simple YES or NO to indicate whether they consider that their tender complies with the requirement. They are also required to insert a reference to the appropriate section of their tenders.

Tenderers may wish to add their own notes to further elaborate on their YES or NO response. Such notes may take the form of text, documents, links to web pages or other suitably relevant and brief materials. Where the material is unclear, irrelevant or unduly lengthy it may be ignored during evaluation at the Customer's sole discretion.

Tenderers should not use their notes as a substitute for a properly reasoned flow of discussion in the main body of their tenders.

Unless otherwise marked to the contrary, every requirement is a mandatory requirement.

Where a requirement is considered to be potentially too challenging for inclusion in Iteration V1 Customer will have marked the requirement as Highly Desirable in the Consolidated Requirements Matrix. This will be shown in the form of the letters (HD) after the requirement number.

Tenders showing a NO (non-compliance) for a mandatory requirement will be deemed non-compliant as a whole.

Tenders showing a NO (non-compliance) for a highly-desirable requirement will be further evaluated.

IMPORTANT NOTES:

- **Some deliverables are required as part of the tender response. These are denoted by wording of the form "Tenderers are required to"**
- **Some deliverables are to be developed and delivered after Effective Date of Contract (EDC) and before or at Final System Acceptance (FSA). These are denoted by wording of the form "Successful Tenderer is required to"**
- **It is essential that Tenderers supply sufficient detail to show that their tenders are based on a sound high-level design, they should insert references to appropriate sections of their tenders in the Compliance Matrix column marked "Tender Cross-Reference & Notes Re: Compliance".**
- **Tenderers by indicating YES to compliance are warranting that their high-level designs define a product which when implemented will lead to a compliant deliverable.**
- **Customer reserves the right to clarify elements of Tenderer's responses during the evaluation process**

5.1 Consolidated Requirements Matrix – Section 1 - Introduction

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 1-1 | 1.1 | Iteration V0 - is available to end users and is intended as a proof-of-concept. It is expected that Tenderers will familiarise themselves with all available information regarding this proof-of-concept platform so as to target their tenders as accurately as possible. Tenderers are required to provide in their tenders a high-level summary of no more than one page of their understanding of this proof-of-concept platform. | | |
| 1-2 | 1.1 | This ITT-2 requires the Successful Tenderer to design, develop, implement, test, and deliver the Processing, Data Access, and Tools Software for Iteration V1 of the DIAS. It also requires the Successful Tenderer to prepare and deliver related documentation, prepare and deliver related training materials, and provide related lifecycle support. Together with the output from ITT-1 Iteration V1, this will then form the entire Iteration V1 DIAS platform. | | |
| 1-3 | 1.1 | The V1 DIAS platform is required to be operational by the end of Q1 2019 and to be supported through to end of Q4 2019. An illustrated timeline for the DIAS can be found in Figure 1.1.1. | | |
| 1-4 | 1.1 | In summary, the Successful Tenderer is required to deliver the following: <ul style="list-style-type: none"> - the design and implementation of a Common Data Access and Catalogue API for all data and information available in the WEkEO (DIAS) platform, hiding from the users the details of the access methods and locations of each dataset, - the design and implementation of Virtual Machine and/or Container Images that are preloaded with a set of software and tools to allow users to access and process the DIAS data and information, - the design and implementation of a Big-Data platform that will allow users to efficiently perform analytics and machine learning (AI) on very large volume of data and information available in the DIAS, and - the maintenance and support of the software delivered during the V1 operations period which is expected to last until the end of Q4 2019. | | |
| 1-5 | 1.3.1 | The European Commission has published a set of generic requirements for the DIAS in the document mentioned and linked to below. Tenderers are required to familiarise themselves with the contents of this document and certify that they have done so in their tenders. | | |
| 1-6 | 1.4 | Tenderers are required to demonstrate in their proposals how they intend to satisfy this requirement for flexibility generally, and specifically how they will do this without the need for extended negotiations of Contract Changes. Tenderers are required to limit their description to no more than one page. | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 1-7 | 1.5 | The Successful Tenderer will be required to agree to full publication of the technical components of its tender response for the purpose of achieving seamless integration with the Successful Tenderers of the related ITTs. Furthermore, Tenderers are required to explicitly stipulate their consent to this requirement in their tenders (subject to them winning the tender evaluation). | | |
| 1-8 | 1.6 | There follows a high-level description of the products of the eight (8) technical work packages required to be delivered by the Successful Tenderer : The full descriptions can be found in Section 3. ... and all further text in Section 1.6. | | |
| 1-9 | 1.7 | If a proposed solution requires the modification of Free Open Source Software: <ul style="list-style-type: none"> - Tenderers are required to inform Customer of all such cases in their tenders and to fully describe and justify the proposed changes. Tenderers should be mindful that the Customer is most concerned to avoid modified software that may become obsolete and unsupported/unsupportable over time. - Tenderers are required to give a binding undertaking in their tenders that they will submit their changes upstream to the official maintainers should they become the Successful Tenderer. - The Successful Tenderer is required to submit its changes upstream at the earliest opportunity and to fully support those changes until such time as they are either accepted upstream and released as a new package or the lifecycle support period has come to an end. - When upstream changes are accepted and a new package is released the Successful Tenderer is required to update the DIAS platform to include the newly released package. | | |
| 1-10 | 1.7 | Tenderers are required to prefer Free Open Source Software (FOSS) solutions where they are available and of a sufficient quality and supportability. | | |
| 1-11 | 1.7 | Tenderers are required to justify their decisions in all cases where they choose an Enterprise Edition (EE) of a product over the corresponding Community Edition (CE), including the consequential costs impact. | | |
| 1-12 | 1.7 | Where a paid for software product is called for in a Tenderer's design his tender is required to show: <ul style="list-style-type: none"> - the entire cost of ownership (including but not limited to license fees, support, updates and/or upgrades) and use over the full lifecycle of the DIAS expressed as a firm fixed price, and - the technical reasons for specifying the product. (NOTE: By answering YES tenderer is asserting that he has complied with these requirements for each and every piece of paid for software in use in his design.) | | |
| 1-13 | 1.7 | Tenderers are required to provide: <ul style="list-style-type: none"> - a complete inventory of all Free Open Source Software (FOSS) used in their High-Level Designs whether aimed at the DIAS system or at the end-user components. (Customer recognises that this inventory cannot be considered definitive at the tendering stage.) - the corresponding licence type for each item of FOSS. - a full copy of each licence type. | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 1-14 | 1.7 | Successful Tenderer will be required to: <ul style="list-style-type: none"> - maintain up-to-date the above inventory throughout the entire lifecycle of iteration V1 of the DIAS. | | |
| 1-15 | 1.7 | Tenderers are required to provide: <ul style="list-style-type: none"> - a complete inventory of all Community Editions Software and Proprietary Software. | | |
| 1-16 | 1.7 | Successful Tenderer will be required to maintain throughout the support period: <ul style="list-style-type: none"> - a complete inventory of all Community Editions Software and Proprietary Software for general perusal by the Customer and by users. | | |
| 1-17 | 1.8 | In all cases, Tenderers are required to: <ul style="list-style-type: none"> - fully describe the type of licence needed (e.g. per-site, per-use, per-user, per-time-period or other), - fully determine the cost of a licence, - fully describe all licence management and administration requirements. | | |
| 1-18 | 1.8 | Tenderers are required to acknowledge and Successful Tenderer is required to ensure the following restrictions: <ul style="list-style-type: none"> - users shall never be deemed to have accepted any licence terms by default, and - users shall never be deemed to have agreed to pay any licence fee by default. | | |
| 1-19 | 1.8 | Tenderers are required to provide a High-Level Design for, and the Successful Tenderers is required to implement: <ul style="list-style-type: none"> - a mechanism whereby users can be made aware of the costs involved and the terms of use of any software package or tool they are proposing to use. They shall always be given an opportunity to make a fully informed decision whether they wish to proceed with licensing the use of the package or tool in question. <ul style="list-style-type: none"> - a mechanism whereby users can pay for and obtain a licence in the event that they wish to do so. - an inventory of purchased/issued licences and corresponding users. | | |
| 1-20 | 1.8 | Tenderers are required to provide a High-Level Design for, and Successful Tenderers is required to implement and deliver: <ul style="list-style-type: none"> - a mechanism whereby Customer can observe any licence restrictions that may apply to the DIAS system itself – for example, where a service is licensed by number of authenticated users of that service (e.g. Gitlab EE) | | |
| 1-21 | 1.9 | These products are mentioned as Customer preferences, but are not mandatory. Tenderers may therefore use other products that they see as best fit for an overall successful system integration of their chosen design, understanding that these Customer preferences provide clarity as to the sort of functionality that is being sought through this ITT-2 | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|--------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | (NOTE: Tendere is required to confirm that he has made choices freely on the basis of the best fit for his design.) | | |
| 1-22 | 1.10 – CFI 1 | Tenderer acknowledges the following requirement: NOTE: Successful Tenderer is required to provide his own facilities during the design, development and testing phases of the project. This shall include any testing at scale that he deems necessary to ensure successful delivery to Customer. | | |
| 1-23 | 1.10 – CFI 2 | Tenderer acknowledges the following requirement: NOTE: Successful Tenderer is required to report to Customer in a timely fashion any potential or actual delays to his work occasioned by a lack of such sample data. | | |
| 1-24 | 1.10 – CFI 3 | Tenderer acknowledges the following requirement: NOTE: Successful Tenderer is required to report to Customer in a timely fashion any potential or actual delays to his work occasioned by a lack of such co-ordination.. | | |
| 1-25 | 1.10 – CFI 4 | Tenderer acknowledges the following requirement: NOTE: As a part of Final System Acceptance (FSA) Successful Tenderer is required to provide Customer with a comprehensive list of his support access requirements. | | |
| 1-26 | 1.10 – CFI 5 | Tenderer acknowledges the following requirement: NOTE: Successful Tenderer is required to report to Customer in a timely fashion any impediments he finds to establishing his support operation. | | |
| 1-27 | 1.10 – CFI 6 | Tenderer acknowledges the following requirement: NOTE: Successful Tenderer is required to report to Customer in a timely fashion any impediments he finds to fulfilling this requirement. | | |
| 1-28 | 1.11 | In summary, tenderers must use their best judgement to achieve a balance between engineering, training, delivery and documentation to ensure that their support operation is feasible at the scale expected of the DIAS in full production. Tenderers are required to explain how their designs are the result of such best judgement. | | |
| 1-29 | 1.12 | Tenderers are required to provide Customer with a full statement of compliance showing where personal data is used and how that use is within the limits of the law. | | |
| 1-30 | 1.13 | Successful Tenderer is required to protect the Customer's information assets in accordance with the Customer's relevant Policies & Procedures. In particular: | | |
| 1-31 | 1.13 | Successful Tenderers is required to: - comply with the Customer's Information Security Policies & Procedures and with all the underlying applicable policies (as included in this ITT-2 tendering package) | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> - be accountable for the information security aspects of the services and systems defined in this ITT-2 | | |
| 1-32 | 1.13 | <p>Tenderers are required to:</p> <ul style="list-style-type: none"> - provide a list of roles and responsibilities within their organisations relevant to the information security aspects of this ITT-2, - provide a description of their secure development procedures and practices, examples being: secure code analysis and systems vulnerability assessment, - provide a description of their Vulnerability Management Process. | | |
| 1-33 | 1.13 | <p>A preliminary evaluation of each Tenderer's compliance with the Customer's information security requirements will be performed by Customer as part of tender evaluation. This will be based solely on the material submitted by the Tenderer in its tender.</p> <p>Tenderers are required to explicitly indicate their acceptance of this stipulation.</p> | | |
| 1-34 | 1.13 | <p>The Customer reserves the right to require the Successful Tenderer to supply audit and/or penetration test reports to determine whether the Successful Tenderer has implemented the required Customer's Information Security Policies and Procedures.</p> <p>Tenderers are required to explicitly indicate their acceptance of this stipulation.</p> | | |

5.2 Consolidated Requirements Matrix – Section 2 – DIAS Iteration V0 - Summary

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 2-1 | 2 | The Successful Tenderer is required to begin its work from the API, Metadata, and adapter specifications of DIAS Iteration V0, and to work in close collaboration with the Customer and with EUMETSAT when and if additions, deletions or other changes are believed to be required. | | |
| 2-2 | 2 | <p>Tenderers are advised to make themselves aware of the details of DIAS Iteration V0 using the following document as a minimum which has been supplied to them as a part of this ITT-2 tendering package:</p> <p><i>DIAS V0 System Description</i> Doc.No. : EUM/GSI/TEN/17/951373 Issue : v1C Date : 18 June 2018 WBS/DBS : © EUMETSAT The copyright of this document is the property of EUMETSAT. EUMETSAT Eumetsat-Allee 1, D-64295 Darmstadt, Germany</p> <p>Tenderer acknowledges that it has been so advised.</p> | | |
| 2.3 | 2 | <p>Any Tenderer finding this document missing from their tendering package must advise Customer immediately.</p> <p>Tenderer confirms that it has received this document.</p> | | |

5.3 Consolidated Requirements Matrices – Section 3 – Work Packages

5.3.1 Consolidated Requirements Matrix – Section 3.1 – Common Data Access API and Searchable Catalogue

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.1-1 | 3.1.1 | The reference to the generic requirements documentation specified by the European Commission for the DIAS service can be found in Section 1.3.1 and contains, among others, the table below. Tenderers are required to take account of the full contents of this document, where applicable, in their designs. | | |
| 3.1-2 | 3.1.2 | Figure 3.1.1 gives an idea of the types of data available, the various formats it is to be found in and the approximate volumes of data. This table is intended for illustration purposes only. Tenderers are required to rely on their own survey information when preparing tenders. | | |
| 3.1-3 | 3.1.2 | It is planned to operate more than 20 satellites in orbit at the same time with 11 different instrument families across the constellation. Note that as more Sentinel satellites are commissioned the daily data downlink volume will increase and is estimated to reach around 10 TB/day. Tenderer's proposed solutions are required to take account of this growth and its consequences for scaling requirements. Tenderers are required to: - provide a brief (no more than one page) summary of how they intend to account in their designs for the amount of data, the daily rate of increase of data, and the consequential need for scalability. | | |
| 3.1-4 | 3.1.2 | The information above about access points is intended for illustration purposes only. Tenderers are required to rely on their own information gathering when preparing tenders. | | |
| 3.1-5 | 3.1.3.1 | The volume of data is very large and growing and mirroring between the three sites has been ruled out as infeasible and unaffordable. Accordingly, it will be necessary to employ selective and intelligent caching and/or streaming in order to make best use of wide-area networking capacity and optimal use of local high-speed storage. Virtual machines and/or Containers will be created where the majority of data access will be local, but some wide-area access requiring caching is to be expected. Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met. | | |
| 3.1-6 | 3.1.3.2 | The data may in most cases be available free of charge; however, end-users are still required to accept the licence terms attached to the data. Accordingly, it will be necessary to electronically collect licence acceptances from clearly identified end-users, to record these acceptances and to update the user's access credentials as required. Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met. | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.1-7 | 3.1.3.3 | <p>Some of the data may be contributed on a commercial basis; it will be necessary to incorporate licence acceptance and/or access rights checking. It is expected that users subscribing to commercial data will pay the provider directly whereupon the user's credentials will be appropriately updated. Customer does not foresee the requirement for any payment mechanisms within the DIAS platform in this Iteration V1.</p> <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.</p> | | |
| 3.1-8 | 3.1.3.4 | <p>The data selected for processing by an end-user together with the tools he uses to process it must be recorded for the purpose of asserting data provenance (but with user personal data anonymised).</p> <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.</p> | | |
| 3.1-9 | 3.1.3.5 | <p>The European Commission requires to be able to analyse usage (uptake) patterns of the datasets and of the DIAS platform as a whole. Usage statistics will be required for all assets delivered under this ITT-2.</p> <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how this requirement is met.</p> | | |
| 3.1-10 | 3.1.3.6 | <p>Data is stored at differing levels of accessibility. Some data is on tape and not in a tape silo, in other words, it is offline. Some data is on tape and in a tape silo – and is therefore nearline. And some data is on random access devices – and therefore online.</p> <p>It will be necessary to have a notion of asynchronous access; but, Tenderers are expected to decide their own design and describe it in their High-Level Design.</p> <p>For example (when accessed data is remote):</p> <ul style="list-style-type: none"> - online data might be locally cached and almost instantaneously available (subject to a transfer over the WAN) and remain available for a set period before being discarded locally - nearline data might be locally cached and available with some delay (subject to an auto-load of a tape in the silo, a tape read, and a transfer over the WAN) and remain available for a set period before being discarded locally - offline data might be locally cached with a scheduled delay (subject to a manual tape load to drive, a tape read, and a transfer over the WAN) and remain available for a set period before being discarded locally <p>It is expected that the caching mechanism will be sufficiently intelligent to only locally cache one copy of those items of data that are requested by multiple users. In such circumstances, its local destruction would be delayed to the end of the service guarantee period for the most recent requester. However, individual users would still be restricted to the window of access assigned to them. Tenderers may assume that sufficient local storage will be allocated to allow the cache to behave reasonably, and not to thrash.</p> <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how all the above caching requirements will be met.</p> | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.1-11 | 3.1.3.7 | <ul style="list-style-type: none"> The API and the service software implementing it will be published at three sites – ECMWF, EUMETSAT & Mercator Ocean International. It may also be published elsewhere in due course as additional data providers join the DIAS. <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how they will structure their API service taking full account of the evolutionary requirement that they begin with the V0 API design.</p> | | |
| 3.1-12 | 3.1.3.7 | <ul style="list-style-type: none"> The Broker is a key component of the service software. This component is responsible for hiding the location of the data and the access protocol from the end-user; it also will implement Quality-of-Service (QoS) through the use of a queue. <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how the Broker requirements will be met.</p> | | |
| 3.1-13 | 3.1.3.7 | <ul style="list-style-type: none"> The Catalogue is a key component of the service software. This component is responsible for providing a comprehensive inventory of available datasets that can be browsed and/or searched. <p>Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how the Catalogue requirements will be met.</p> | | |
| 3.1-14 | 3.1.3.7 | The Successful Tenderer is required to begin its work from the API, Metadata, and adapter specifications of DIAS Iteration V0, and to work in close collaboration with the Customer and with EUMETSAT when and if additions, deletions or other changes are believed to be required. | | |
| 3.1-15 | 3.1.3.8 | <p>There are a number of different formats in use among the various datasets that are to form the DIAS. All those currently in use are to be accommodated, and it must be relatively simple to add new ones. Accordingly, a framework-and-adapters approach is suggested, defined as:</p> <ul style="list-style-type: none"> a single software framework that handles all of the generic services (cataloguing, caching, access control, etc ...) an internal API for inserting adapters that adapt the framework to a particular type of dataset, and an (extensible) suite of adapters that implement a number of different dataset types. <p>Tenderers should examine the Iteration V0 adapters code and consider whether it is suitable for re-use.</p> <p>Tenderers may wish to consider the extract-transform-load (ETL) class of tools for some parts of this requirement – for example, NiFi and Talend. The choice of tool(s) remains entirely for Tenderers to decide.</p> <p>Successful Tenderer is required to deliver:</p> <ul style="list-style-type: none"> a framework as sketched above an internal API as sketched above, and | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> - a suite of data adapters that fit into their chosen framework and transform data at rest into their chosen on-the-wire transfer format. (Tenderers may wish to make use of the Iteration V0 code; this is entirely a matter for their discretion.) | | |
| 3.1-16 | 3.1.3.8 | <p>With such a large collection of data from so many different sources, it is extremely important to have a rigorous naming, cataloguing and searching capability.</p> <p>Successful Tenderer is required to deliver:</p> <ul style="list-style-type: none"> - a single namespace for all datasets (this namespace must be extensible to new data sources and/or types), - a single vocabulary for sub-setting datasets by attributes (this vocabulary must be based on existing ones, in particular, CF Conventions & Metadata), - a catalogue with complete and accurate descriptions of each and every dataset using INSPIRE ISO 19139, and - a search facility providing high-quality search results based on user specified criteria for any piece of metadata associated with any dataset. | | |
| 3.1-17 | 3.1.3.8 | Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how their framework-and-adapters design will be implemented. | | |
| 3.1-18 | 3.1.3.9 | <p>Successful Tenderer is required to deliver:</p> <ul style="list-style-type: none"> - a RESTful API design and implementation of the Data Access API & Browsable/Searchable Catalogue, - a facility in their API to allow subsets of data to be delivered based on JSON templates, - a Quality-of-Service (QoS) capability in their API together with high-availability and elastic scaling to prevent service outages and/or degradation under heavy user demand, - a Secure API based on the HTTPS protocol for all requests (data transfers may use HTTP subject to licence restrictions). | | |
| 3.1-19 | 3.1.3.9 | Tenderers are required to demonstrate in their High-Level Designs and Successful Tenderer is required to elaborate in its Detailed Design how the API General Characteristics requirements will be met. | | |

5.3.2 Consolidated Requirements Matrix – Section 3.2 – Virtual Machine / Container Images & Browsable/Searchable Image Catalogue

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.2-1 | 3.2 | <p>Successful Tenderer is required to deliver two distinct types of Virtual Machine Images;</p> <ul style="list-style-type: none"> - The first type is intended for direct use by end users (possibly as a virtual desktop or via SSH and the command line) – these will be referred to as Compute Node Images. - The second type is intended to aid the construction of special purpose clusters that will be used indirectly by end users (i.e. they will not usually routinely log-in to such running images). These will be referred to as Cluster Node Images. This section deals with Compute Node Images. See Section 3.3 for further information about Cluster Node Images. | | |
| 3.2-2 | 3.2 | <p>Successful Tenderer is required to deliver:</p> <ul style="list-style-type: none"> - a browsable and searchable catalogue of available Virtual Machine Images and/or Container Images, <p>The Iteration V1 DIAS infrastructure will be based on OpenStack with a Morpheus user interface. The base-level store from which images will be launched will be Glance with its limited metadata facilities. The image catalogue will provide richer metadata including communities of interest and software inventories and will point to the unique identifier for the relevant image in Glance. This catalogue will be exported via a RESTful API that can be accessed by custom code or a command line tool or by a modified GUI such as Horizon or Morpheus.</p> <ul style="list-style-type: none"> - A search taxonomy that includes, but is not limited to: <ul style="list-style-type: none"> o User organisation o User experience level o User discipline (e.g. oceanography, climate) o Available tools - a method for instantiating instances of Virtual Machine and/or Container Images, and - a method for terminating instances of Virtual Machine and/or Container Images. | | |
| 3.2-3 | 3.2 | Successful Tenderer is required to deliver the browse, search, instantiate and terminate capabilities as a web service accessible through both an API, through a browser, and via a command line tool. | | |
| 3.2-4 | 3.2 | Successful Tenderer is required to incorporate in his Detailed Design the ability for the Customer and/or end-users to create their own Virtual Machine and/or Container Images and to register them with the Catalogue with possibly limited scope (for example, user organisation). | | |
| 3.2-5 | 3.2 | Tenderers are required to demonstrate in their High-Level Designs how they will meet all the requirements of this section. | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.2-6 | 3.2.1 | While all software listed in Section 3.2.1.2 through Section 3.2.1.5 is mandatory, this does <u>not</u> mean that every image must contain every piece of software. Tenderer acknowledges this stipulation. | | |
| 3.2-7 | 3.2.1 | Tenderers are required to: <ul style="list-style-type: none"> - propose appropriate subsets in their tenders that collectively form a tailored suite, - provide a rationale for each choice, - provide a cross-reference demonstrating the inclusion of every piece of software in at least one image, - provide a list of known-at-the-time conflicts that may prevent certain combinations of software and/or Linux distributions. | | |
| 3.2-8 | 3.2.1 | Prior to implementation by Successful Tenderer, the precise content of each image will be agreed with the Customer who may require adjustments to be made. Tenderer acknowledges this stipulation. | | |
| 3.2-9 | 3.2.1 | Successful Tenderer is required to size each image, provide appropriate instance flavour settings and automatically enter this combined information into a consolidated Catalogue of available images. | | |
| 3.2-10 | 3.2.1.1 | Successful Tenderer is required to: <ul style="list-style-type: none"> - design, develop, implement, test and deliver a suite of images for Catalogue publication globally across the WEKEO (DIAS) cloud. These images shall contain standard suites of software sufficient to address the needs of three broad groups of users, namely: <ul style="list-style-type: none"> o domain experts across the spectrum of DIAS applicability o software developers across the spectrum of sophistication o platform administrators and DevOps engineers - ensure that it is possible for the Customer and/or end-users to customise these images to their own individual requirements and to re-publish them suitably renamed either to their local scope within the catalogue or to the global catalogue scope subject to appropriate authorisation from platform administrators - ensure that where end-users wish to create their own custom images from scratch, they are able to make use of any tools that Successful Tenderer uses in the construction of his images. Again, publication either to the local scope or to the global scope subject to authorisation shall be possible | | |

| For use by Customer | | | For use by Tenderer | |
|---|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> - make appropriate use of automation in the construction of Virtual Machine Images. This shall be sufficient to ensure freedom from manual keying errors and avoidance of unnecessary repetition. See Section 3.4 for more details - install the Data Access API client software if appropriate on all images irrespective of their target user group or discipline - ensure that upon instantiation of a virtual machine and/or container the client API is configured with the user's credentials so that the user does not have to login to the API in addition to logging in to the instance - ensure that where possible and appropriate pre-installed software packages are modified and/or extended to be aware of the Data Access API, so that the user does not need to configure anything | | |
| <p>Requirements 3.2-11 to 3.2-14 are not required to be fulfilled by every virtual machine image (VMI) or container image (CI). The exact contents of each image are a matter for tenderers (as per text in Section 3.2.1.1). However, it is required that each item appears in at least one virtual machine image (VMI) or container image (CI) and that the overall combinations of software are well explained and justified. Tenderers may wish to use a table for their responses to this group of requirements so as to make clear which images contain which software; in this case, they may leave requirements 3.2-4 through 3.2-7 blank in this matrix. Tenderers should find a suggested spreadsheet attached to the tendering package in the document "COP_043_Volume II Annex 3_Software_Packages_Compliance_Matrix.xlsx".</p> | | | | |
| 3.2-11 | 3.2.1.2 | <p>Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):</p> <ul style="list-style-type: none"> - all Science Toolbox Exploitation Platform (STEP) SNAP toolboxes for the Sentinel missions - the Orfeo Toolbox (OTB) and by extension the ITK Image Segmentation & Registration Toolkit - the Broadview Altimetry Toolbox (BRAT) - Panopoly - ToolsUI - the QGIS Geographic Information System software - EO Workbench (EOPaaS) | | |
| 3.2-12 | 3.2.1.3 | <p>Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):</p> <ul style="list-style-type: none"> - the Python and R programming language tools. Tenderers may add additional languages as required by his design. Both Python 2.7 and Python 3 are to be provided. - Interactive versions of Python (Jupyter) and R (RStudio) through the Anaconda (Python Distribution) - the Apache Zeppelin web-based notebook (tenderer may use the docker image within a VMI/CI if he so wishes) | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> - the following IDEs / Editors: Eclipse, IntelliJ IDEA (Community Edition), Visual Studio Code, Atom Editor and Brackets Editor | | |
| 3.2-13 | 3.2.1.4 | <p>Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):</p> <p>NOTE: Where other packages are required as dependencies, or where Tenderers believe there are important gaps, these lists should be extended by the Tenderers, and the additions made explicit in their Tenders.</p> <ul style="list-style-type: none"> - essential Python libraries and tools for data manipulation, optimized computation and visualization: <ul style="list-style-type: none"> o SciPy, Numpy, Pandas, IPython, Matplotlib, Seaborn, Bokeh and SciKit-learn - essential R libraries and tools for data manipulation, optimized computation and visualization: <ul style="list-style-type: none"> o Caret, Kernlab, randomForest, nnet, e1071, klaR, tree, rpart, MICE, Lasso2, Lars, Gbm, BayesTree - Python remote sensing libraries: <ul style="list-style-type: none"> o GDAL, Georasters, GeoPanda, CLIMAF, Pcdi_metrics, Fiona, Fmask, Shapely, Cartopy, Rtree, PySAL, NanSat, SatPy - R remote sensing libraries: <ul style="list-style-type: none"> o Sp, Rgdal, Fields, Maptools, Rstoolbox, Sf, Rasters, Gdistance, Geosphere - Copernicus / Sentinel libraries and utilities: <ul style="list-style-type: none"> o Sentinelsat, Sentinel2ProductIngestor, Sat-downloader, Sat-api, Sentinel-hub, Peps-download | | |
| 3.2-14 | 3.2.1.5 | <p>Successful Tenderer is required to include in at least one image (consistent with the primary purpose of the image):</p> <ul style="list-style-type: none"> - Include the following deep learning package: <ul style="list-style-type: none"> o TensorFlow | | |
| 3.2-15 | 3.2.1.6 | <p>Successful Tenderer is required to:</p> <ul style="list-style-type: none"> - Implement ALL the installed/configured software requirements on EACH of the following Linux distributions: <p>NOTE: If during implementation a particular Virtual Machine Image proves impossible to implement on a particular Linux distribution, Successful Tenderer shall inform Customer and agree a compromise or exemption.</p> <ul style="list-style-type: none"> o RedHat 7.5-1804 / CentOS 7.5-1804 (or latest stable release at time of development) | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|------------------------|--|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> ○ Ubuntu 16.04 (Xenial Xerus) / Ubuntu 18.04 (Bionic Beaver) (or latest stable release at time of development) ○ Debian 8 (Jessie) / Debian 9 (Stretch) (or latest stable release at time of development) ○ OpenSUSE 15.0 (Leap) (or latest stable release at time of development) ○ Scientific Linux 7.5 (or latest stable release at time of development) - Include a standard set of Linux system administration tools of the kind typically found on a Linux Desktop Machine or on a Linux Server Machine - Include appropriate desktop environment options according to the Linux distribution in use - ensure that all package management client software is loaded by default and configured to use the local artefacts store (see Section 3.4) | | |
| 3.2-16 | 3.2.1.7 | <p>Successful Tenderer is required to:</p> <ul style="list-style-type: none"> - ensure that the only method of access to images is through a user's SSH keys (the public key for which is pre-loaded to the Cloud). Username and password access is forbidden except where GUI-based access mandates otherwise. (This is a pragmatic exemption that may change in the future.) - ensure that user credentials for the Data Access API & Catalogue are seamlessly added to each virtual machine upon instantiation - ensure that all system and/or vendor accounts (including default accounts and public keys) that are not required are removed or disabled where removal is infeasible. - ensure that all system and/or vendor accounts that are required have their passwords changed to non-default values. - ensure that all software applications that can function without root access do so. - ensure that there are no plaintext passwords or other forms of access credentials hard-coded into applications, programs or scripts or stored in filesystems, databases or version control systems - ensure that only required software is installed and/or enabled in all cases except where a Virtual Machine Image is intended for general-purpose use (for example, a general-purpose scientific or administrative workstation image) - ensure that any standalone asset shall be capable of enforcing and maintaining an adequate access control policy (for example, a Linux VM with no external directory service relying entirely on local permission management), - ensure that images are configured to give instance owners sudo rights. (Where an image is normally only accessible through SSH keys, the user password required to exercise sudo rights | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|------------------------|--|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <p>shall be transmitted to the user by the DIAS platform once and once only. User shall then be obliged to change said password on first use.)</p> <ul style="list-style-type: none"> - ensure that the principle of least privilege and need-to-know is implemented by default in all images using an appropriate firewall capability within the image and a corresponding cloud Infrastructure Security Group. Where individual tools or libraries call for less strict security, this shall be pre-configured within the image firewall and an appropriate cloud Security Group created and associated with the image. - ensure that an automated suite of settings is applied to all Linux images during creation to ensure that they conform to current best security practice. At minimum this suite shall: <ul style="list-style-type: none"> o remove all unnecessary software and data o remove unnecessary access credentials o disable unnecessary services o deactivate unused features o disable unused network ports o forbid anonymous access <p>As an example, Tenderers may wish to use a tool that applies Security Technical Implementation Guide (STIG) to all images such as Ansible Hardening or OpenSCAP that is an implementation of the Security Content Automation Protocol (SCAP). Tenderers are required to make their own judgements about the best tool(s) to use for their design and should describe their choice(s) in their tenders.</p> <p>NOTE: It is not required to repeat this process at instance creation or at any time thereafter.</p> | | |
| 3.2-17 | 3.2.1.8 | <p>Successful Tenderer is required to:</p> <p>ensure that appropriate methods are in place to configure DNS, NTP and authentication via the appropriate cloud initialisation mechanism (e.g. cloud-init)</p> | | |

5.3.3 Consolidated Requirements Matrix – Section 3.3 – Big Data Analytics Platform

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.3-1 | 3.3.2 | The core processing functionality that the Customer requires to have on the platform is Apache Spark. | | |
| 3.3-2 | 3.3.2 | <p>Tenderers are required to deliver:</p> <ul style="list-style-type: none"> - a recommended distribution or set of distributions of Apache Spark. This may be a single distribution or a combination of several distributions. It is important to illustrate in tenders the benefits and limitations of each distribution in regard to, but not limited to, such matters as: <ul style="list-style-type: none"> o cost – what are the total lifecycle cost including any per-user component(s)? o support – how will support work from our end-user’s viewpoint? o functionality – what other tools and sub-systems are integrated with the product? o technical specifications – how complete is the product? o currency of distribution relative to upstream open source version – is the vendor a key participant in the upstream development process and does his product stay close to the upstream functionality? (In answering this, tenderers should be aware that Customer is trying to assess the extent to which an offering has become or is becoming proprietary.) <p>Tenderers should note that the Customer’s purpose of calling for several distributions is to allow the tenderer to propose:</p> <ul style="list-style-type: none"> o least cost (possibly free) distributions for users who do not wish to pay for commercial support and/or features o leading-edge distributions for big data researchers developing new approaches to DIAS data processing that require the latest Apache Spark features o commercially supported and featured distributions for users who require them such as those responsible for producing production quality operational data | | |
| 3.3-3 | 3.3.2 | <p>Successful Tenderer is required to deliver:</p> <ul style="list-style-type: none"> - a suite of Virtual Machine Images and/or container images designed and implemented to provide the above distributions and suitable for building the following environments: <ul style="list-style-type: none"> o single node developer environments o small-scale multi-node clustered developer environments – these are intended for clustered software testing and not for big data handling. They are not likely to be interfaced to any massive-scale NoSQL databases or clustered filesystems. | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|------------------------|--|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> ○ Medium-scale multi-node clustered data science environments – these are intended for data scientist to begin familiarisation with big data processing with Apache Spark. They may require a shared big data filesystem or database to be present. ○ (Optional) Large-scale multi-node clustered data science environments intended for production of operational data. In accordance with our basic building blocks principle, we are unable at this time to fully specify the needs of such an environment. <p>The images are to be designed and implemented using the approach described in Section 3.2.</p> <ul style="list-style-type: none"> - a set of automation mechanisms for creating clusters (where appropriate) using the above Virtual Machine Images and/or Container Images for the single, small-scale and medium-scale cases and (Optional) for the large-scale case. These automations are to be integrated into the Build Automation (DevOps & CI/CD) system described in Section 3.4. <p>Tenderers are required to use:</p> <ul style="list-style-type: none"> ○ HashiCorp Terraform together with his choice of configuration management mechanism to achieve this for Virtual Machine Images ○ Docker or Kubernetes together with his choice of configuration management mechanism to achieve this for Container Images <p>... and to augment these with appropriate configuration management tools where appropriate.</p> | | |
| 3.3-4 | 3.3.3 | <p>With the above limitations in mind, Successful Tenderer is required to provide:</p> <ul style="list-style-type: none"> - distributions of Ceph and Cassandra in free open source software (FOSS) form only - Ceph and Cassandra build automation mechanisms suitable for generating the following environments: <ul style="list-style-type: none"> ○ minimal environments that present the full product API but run on minimum resource (e.g. a laptop, a desktop or a virtual desktop) ○ small-scale clusters that present the full product API but run on a sufficient number of resources to enable cluster functions to be investigated and tested ○ (Optional) large-scale clusters if there are sufficient differences from the small-scale cluster case above | | |
| 3.3-5 | 3.3.3 | <p>Tenderers are required to provide:</p> <ul style="list-style-type: none"> - a High-Level Design for their proposed Ceph and Cassandra environments. | | |
| 3.3-6 | 3.3.4 | Successful Tenderer is required to: | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> - ensure that the only method of access to images is through a user's SSH keys (the public key for which is pre-loaded to the Cloud). Username and password access is forbidden. - ensure that all system and/or vendor accounts (including default accounts and public keys) that are not required are removed or disabled where removal is infeasible. - ensure that all system and/or vendor accounts that are required have their passwords changed to non-default values. - ensure that all software applications that can function without root access do so. - ensure that there are no plaintext passwords or other forms of access credentials hard-coded into applications, programs or scripts or stored in filesystems, databases or version control systems - ensure that only required software is installed and/or enabled in all cases except where a Virtual Machine Image is intended for general explorative use (for example, a general purpose scientific workstation image) - ensure that any standalone asset shall be capable of enforcing and maintaining an adequate access control policy (for example, a Linux VM with no external directory service relying entirely on local permission management), (This situation may very well arise in the case of Virtual Machine Images designed as nodes for Ceph, Cassandra, Kubernetes or any other clustered system) - ensure that images are configured to give instance owners sudo rights; however, a mechanism must be put in place to ensure that the user sets/resets the password within a short time from creation of the VM, - ensure that the principle of least privilege and need-to-know is implemented by default in all images using an appropriate firewall capability. Corresponding Cloud Infrastructure Security Groups shall be delivered to further enforce these restrictions within the cloud infrastructure. Where individual tools or libraries call for less strict security, this will be controlled through the automated application of appropriate firewall and security group rules. - ensure that an automated suite of settings is applied to all Linux images to ensure that they conform to current best security practice. At minimum this suite shall: <ul style="list-style-type: none"> o remove all unnecessary software and data o remove unnecessary access credentials o disable unnecessary services o deactivate unused features o disable unused network ports o forbid anonymous access | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | As an example, tenderers may wish to use a tool that applies Security Technical Implementation Guide (STIG) to all images such as Ansible Hardening or OpenSCAP that is an implementation of the Security Content Automation Protocol (SCAP). Tenderers are required to make their own judgements about the best tool(s) to use for their design and should describe their choice(s) in their tenders. | | |

5.3.4 Consolidated Requirements Matrix – Section 3.4 – Build Automation (DevOps & CI/CD)

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.4-1 | 3.4.1 | Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work: <ul style="list-style-type: none"> a DevOps toolchain that enables continuous integration and continuous delivery of all platform artefacts in a manner that is <u>repeatable</u> and <u>scales</u> well to tens of thousands of users | | |
| 3.4-2 | 3.4.2 | Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work: <ul style="list-style-type: none"> a shared <u>archive</u> of all original artefacts generated and required to (re-)build the platform from scratch (this also includes any user-created products that have been centrally registered with the DevOps toolchain) <i>(The following is an incomplete list of examples of such artefacts: compiled object files, archive files, image files, and other binary large objects (BLOBs) such as test data.)</i> | | |
| 3.4-3 | 3.4.3 | Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work: <ul style="list-style-type: none"> a shared <u>archive</u> of all pre-existing (internet downloaded and vendor produced) artefacts required to (re-)build the platform from scratch including any products that are dependencies of centrally registered user-created products <i>(The following is an incomplete list of examples of such artefacts: ISO images, Java packages, Python packages, baseline cloud images.)</i> | | |
| 3.4-4 | 3.4.4 | Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work: | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <ul style="list-style-type: none"> - a shared version control system for storage of all production versions of original artefacts in source format (code and related documentation such as READMEs and INSTALLs) <i>(Storage of large objects under version control is strongly discouraged.)</i> | | |
| 3.4-5 | 3.4.5 | <p>Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:</p> <ul style="list-style-type: none"> - a simple automated mechanism whereby a user can add all the DevOps CI/CD client tools to his own environment whether it be a VM from Section 3.2 or a local workstation/laptop environment running an appropriate version of Linux. This is primarily intended to automate away user errors during initial setup – thereby saving on support cases. Tenderers may find it sufficient to achieve this with a simple idempotent script. | | |
| 3.4-6 | 3.4.6 | <p>Tenderers are required to High-Level Design and the Successful Tenderer is required to detailed design, implement, test, deliver and use to the fullest extent possible in his other related work:</p> <ul style="list-style-type: none"> - A simple mechanism to ensure that the three primary DIAS sites receive all changes to the DevOps CI/CD content systems (i.e. version control and archive). This may be achieved on an eventually consistent basis as long as catalogues are updated only after content is fully consistent. Tenderers may find it sufficient to achieve this with very simple rsync-based replication provided the target can never be observed in an inconsistent state. | | |
| 3.4-7 | 3.4.7.1 | <p>A tool such as Atlassian Jira is very popular across a wide spectrum of user types. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it.</p> <p>It should be noted that Customer uses an enterprise-wide implementation of Jira.</p> | | |
| 3.4-8 | 3.4.7.2 | <p>As source code is created it should be submitted to version control often and in small chunks; holding new work in local repositories over long periods without pushing it to a central repository is strongly discouraged.</p> <p>If Git were to be proposed then a tool such as GitLab (for on premises) or GitHub (for off premises) is commonly used as a sharing repository. Whichever tool Tenderer choose, they are required to describe the capabilities of the tool and how they intend to make use of it.</p> <p>It should be noted that Customer uses GitLab on some of its project and has plans to publish some of its works using GitHub.</p> | | |
| 3.4-9 | 3.4.7.3 | <p>There are a number of options in this area and Tenderers may propose any suitable candidate. The features the Customer would prefer to see include: integration with source code control, source code browsing, code annotation by reviewers, integration with e-mail and/or messaging.</p> <p>Static security analysis of code might be done with tools such tools as AppScan Source, BlackDuck, Fortify, Veracode, Snyk and Sonatype Checkmarx.</p> | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | Static security analysis of code (mainly Java and/or C++) might be done with such tools as Coverity, PMD, JDepend, Checkstyle, Findbugs, Emma, Cobertura, SonarQube and Klocwork. Whichever tools Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it. | | |
| 3.4-10 | 3.4.7.4 | Wiki-based documentation is strongly preferred (it is not necessary to produce book quality printed versions of the documentation). Commonly, this is achieved using commercial offerings such as Atlassian Confluence or open source offerings such as MediaWiki. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it. It should be noted that the Customer makes enterprise-wide extensive use of Atlassian Confluence. | | |
| 3.4-11 | 3.4.7.5 | An open source Jenkins solution might typically be chosen along with appropriate plugins and local customisation. Whichever tool the Tenderers choose, they are required to describe the capabilities of the tool and how they intend to make use of it. Customer has a strong preference for an simple solution rather than a comprehensive solution. | | |
| 3.4-12 | 3.4.7.6 | There are a number of options for testing such as Mockito, Gatling, Spock, Arquillian, wiremock, Cucumber, JMeter, JUnit and PowerMock (to some extent depending on language to be covered). Whichever tools Tenderers choose, they are required to describe the capabilities of the tools and how and for which languages they plan to use them. | | |
| 3.4-13 | 3.4.7.7 | A commonly chosen commercial offering in this area is Artefactory. A community edition exists; however, it is severely limited in the BLOB types it understands. An alternative would be Sonatype Nexus OSS which has less stringent limitations. Whichever tool the Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it. The essential purpose of this capability is to make the system self-contained so that all artefacts can be re-created without reference to Internet sources, i.e. it serves a part of the <u>archive requirement</u> . | | |
| 3.4-14 | 3.4.7.8 | HashiCorp Packer – a multi-platform image creation tool. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it. | | |
| 3.4-15 | 3.4.7.9 | HashiCorp Terraform – a multi-provider Infrastructure-as-Code (IaC) provisioning tool. Whichever Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it. | | |
| 3.4-16 | 3.4.7.10 | HashiCorp Vagrant – a multi-virtualisation-platform orchestrator for small virtual machine clusters ideal for mobile developer and demonstrator work environments. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it. | | |
| 3.4-17 | 3.4.7.11 | There are a number of tools in this area. In alphabetical order, the most well-known are: Ansible, Chef, Puppet and Salt. Again, the tenderer must decide what works best for his proposed design. Tenderer should note though that | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | Customer has a large installed base of Puppet configuration management capabilities and significant experience of developing and using Puppet built up over many years; while the tenderer is free to choose a tool other than Puppet, he will need to present strong arguments for doing so. Whichever tool Tenderers choose, they are required to describe the capabilities of the tool and how and for which artefacts they plan to use it. | | |
| 3.4-18 | 3.4.7 | It is essential that Tenderers avail themselves of all design information regarding the underlying infrastructure being delivered through ITT-1 (EUMETSAT). This will ensure that there is a seamless integration between any tools delivered under this work package and the capabilities of the infrastructure being delivered under ITT-1. Tenderers are required to accept full responsibility for delivering this integration on the ITT-2 side – save that Customer will facilitate communications with EUMETSAT and MERCATOR and their contractors – see CFI-1 in Section 1.10. | | |

5.3.5 Consolidated Requirements Matrix – Section 3.5 - Testing

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.5-1 | 3.5.1.1 | The Successful Tenderer is required to develop a suite of unit tests sufficient to cover all units of software developed by him as a deliverable under this ITT-2. | | |
| 3.5-2 | 3.5.1.2 | The Successful Tenderer is required to develop a suite of integration tests sufficient to cover all software integrations developed by him as a deliverable under this ITT-2. | | |
| 3.5-3 | 3.5.1.3 | The Successful Tenderer is required to develop a suite of functional tests sufficient to demonstrate correct operation of a suite of typical user and administrator use cases. | | |
| 3.5-4 | 3.5.1.4 | The Successful Tenderer is required to deliver all tests fully integrated into a testing framework. This test framework and the tests shall be updated during the support period to ensure that there is no regression of corrected faults. A simple framework should be chosen sufficient to meet these requirements; use of the framework should not require continuous maintenance or support. | | |
| 3.5-5 | 3.5.1.5 | The Successful Tenderer is required to ensure that a full run of all tests is initiated by the DevOps CI/CD system described in Section 3.4 on at least a weekly schedule. A sophisticated testing schedule may be required at Iteration V2 or later; however, at Iteration V1 this is not needed. | | |
| 3.5-6 | 3.5.1.6 | The Successful Tenderer is required to ensure that platform developers may submit their own test cases at will. | | |
| 3.5-7 | 3.5.1.7 | The Successful Tenderer is required to ensure that testing feedback is provided to platform developers as quickly as possible to facilitate speedy corrections. | | |
| 3.5-8 | 3.5.1.8 | <p>The Successful Tenderer is required to replicate the test environment to the three production DIAS sites to be delivered by EUMETSAT under ITT-1.</p> <p>Tenderers are required to show in their High-Level Designs how they will implement the above replication requirement.</p> <p>Tenderers are required to acknowledge in their tenders their agreement that Final System Acceptance testing will be conducted on all three of the production sites.</p> <p>The Successful Tenderer is required to conduct Customer witnessed Final System Acceptance testing on all three production sites and to deliver to Customer a test report in each case.</p> <p>The Successful Tenderer is required to support a campaign of site testing during the support phase in the case that new sites become available.</p> | | |
| 3.5-9 | 3.5.2 | Successful Tenderer is required to complete the specification and development of their testing framework and associated tests and submit them to Customer for approval by Milestone WP5M1. | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <p>Customer will review and accept/reject the work products by Milestone WP5M2. Tenderer will then have 10 working days to make good any deficiencies and re-submit the work products.</p> <p>Final System Acceptance and delivery of test reports will take place at Milestone WP5M3 and shall last up to 5 working days.</p> <p>Once Customer issues a certificate of successful Final System Acceptance (FSA) Successful Tenderer is required to enter the support period and the testing framework and tests shall become part of the operational environment and shall be maintained and used to validate changes and prevent regression.</p> <p>Tenderers are required to include the testing framework in their High-Level Design which is required to be delivered as a part of their tender.</p> | | |

5.3.6 Consolidated Requirements Matrix – Section 3.6 - Documentation

| For use by Customer | | | For use by Tenderer | |
|--|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.6-1 | 3.6.1 | <p>The Successful Tenderer is required to:</p> <ul style="list-style-type: none"> - present all newly generated documentation as Wiki pages and in PDF and/or Word formats - re-use open source documentation where possible – but only in the form of Wiki pages and PDF/Word formats where available <ul style="list-style-type: none"> o external links are strongly discouraged o other content types are discouraged but not prohibited (i.e. .doc, .xls, .pdf, .epub) - re-use vendor documentation where possible – but only in the form of Wiki pages and PDF/Word formats where available <ul style="list-style-type: none"> o external links are strongly discouraged o other content types are discouraged but not prohibited (i.e. .doc, .xls, .pdf, .epub) - PDF/Word formats should be supplied where available via links within appropriate Wiki pages | | |
| 3.6-2 | 3.6.2 | <p>Tenderers are required to:</p> <ul style="list-style-type: none"> - Provide a description of their approach to the preparation of documentation covering: <ul style="list-style-type: none"> o Tools to be used o Processes to be used for quality assurance o Handling of the various original sources of documentation o Deployment to the DIAS platform | | |
| <p>The following items (3.6-3 to 3.6-11) are required per package or tool. Tenderers are free to submit their compliance responses in the form of a table with one package per row and one requirement per column. For example, an Excel spreadsheet with a YES or NO entry per compliance item together with cell comments where required is acceptable. Customer has included a sample spreadsheet with this tendering package in the document “COP_043_Volume II Annex 2_Software_Documentation_Compliance_Matrix.xlsx”.</p> | | | | |
| 3.6-3 | 3.6.3.1 | Successful Tenderer is required to deliver a high-level description of the software product. This guide shall describe at a high-level what it does and briefly how it does it, | | |
| 3.6-4 | 3.6.3.2 | Successful Tenderer is required to deliver a detailed description of the theory of operation of the software product together with a guide to how the functionality is spread across the various source code components of the product, (for the sake of clarity, this requirement only applies to Successful Tenderers’ own developed code modules and code that he writes to integrate with other products) | | |
| 3.6-5 | 3.6.3.3 | Successful Tenderer is required to deliver an installation and configuration guide. This guide shall be sufficient for a system administrator or end-user (as appropriate) to install the software package and configure it for use, | | |

| For use by Customer | | | For use by Tenderer | |
|---|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.6-6 | 3.6.3.4 | Successful Tenderer is required to deliver a user guide. This guide shall be limited to how to use the software (prior installation and configuration of said software is assumed), | | |
| 3.6-7 | 3.6.3.5 | Successful Tenderer is required to deliver a build guide. This guide shall show how to generate an installable binary of the software product from its corresponding source code, | | |
| 3.6-8 | 3.6.3.6 | Successful Tenderer is required to deliver a testing guide. This guide shall show how to test the software product, | | |
| 3.6-9 | 3.6.3.7 | Successful Tenderer is required to deliver a source repository with all code needed to build the software product – dependencies to be dealt with as separate software products – tools to be installed separately as indicated by the build guide, | | |
| 3.6-10 | 3.6.3.8 | Successful Tenderer is required to deliver hashes as appropriate to ensure the integrity of the software product, | | |
| 3.6-11 | 3.6.3.9 | Successful Tenderer should be aware of the intellectual property regime under which Customer (i.e. ECMWF) is operating. All rights in the software have to be assigned to Customer (i.e. ECMWF) and the European Commission. | | |
| The following items (3.6-12 to 3.6-20) are required per package or tool. Tenderers are free to submit their compliance indications in the form of a table with one package per row and one requirement per column. For example, an Excel spreadsheet with a YES or NO entry per compliance item together with cell comments where required is acceptable. Customer has included a sample spreadsheet with this tendering package in the document "COP_043_Volume II Annex 2_Software_Documentation_Compliance_Matrix.xlsx". | | | | |
| 3.6-12 | 3.6.4.1 | Successful Tenderer is required to deliver a high-level description of the software product. This guide shall describe at a high-level what it does and briefly how it does it, | | |
| 3.6-13 | 3.6.4.2 | Successful Tenderer is required to deliver an installation and configuration guide. This guide shall be sufficient for a system administrator or end-user (as appropriate) to install the software package and configure it for use, | | |
| 3.6-14 | 3.6.4.3 | Successful Tenderer is required to deliver a user guide. This guide shall be limited to how to use the software (prior installation and configuration of said software is assumed), | | |
| 3.6-15 | 3.6.4.4 | Successful Tenderer is required to deliver a build guide. This guide shall show how to generate an installable binary of the software product from its corresponding source code, | | |
| 3.6-16 | 3.6.4.5 | Successful Tenderer is required to deliver a testing guide. This guide shall show how to test the software product, | | |
| 3.6-17 | 3.6.4.6 | Successful Tenderer is required to deliver a source repository with all code needed to build the software product – dependencies to be dealt with as separate software products – tools to be installed separately as indicated by the build guide, | | |
| 3.6-18 | 3.6.4.7 | Successful Tenderer is required to deliver appropriate attribution where the work of others has been used including links to the original web site(s), | | |
| 3.6-19 | 3.6.4.8 | Successful Tenderer is required to deliver hashes as appropriate to ensure the integrity of the software product. | | |

| For use by Customer | | | For use by Tenderer | |
|---|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.6-20 | 3.6.4.9 | Successful Tenderer must demonstrate that all open source software packages and tools will not conflict with the intellectual property ownership rights' assignment of the software. Customer (i.e. ECMWF) will reject any software the license terms of which might be harmful to the overall intellectual property rights as required to be obtained by ECMWF and the European Commission. | | |
| The following items (3.6-21 to 3.6-27) are required per package or tool. Tenderers are free to submit their compliance indications in the form of a table with one package per row and one requirement per column. For example, an Excel spreadsheet with a YES or NO entry per compliance item together with cell comments where required is acceptable. Customer has included a sample spreadsheet with this tendering package in the document "COP_043_Volume II Annex 2_Software_Documentation_Compliance_Matrix.xlsx". | | | | |
| 3.6-21 | 3.6.5.1 | Successful Tenderer is required to deliver a high-level description of the software product This guide shall describe at a high-level what it does and briefly how it does it, | | |
| 3.6-22 | 3.6.5.2 | Successful Tenderer is required to deliver all documentation without exception that he has received from the original software manufacturer whether on paper or electronic, | | |
| 3.6-23 | 3.6.5.3 | Successful Tenderer is required to deliver an implementation document showing how he has integrated the software into the overall platform, and detailing all configuration information needed to re-implement his integration, | | |
| 3.6-24 | 3.6.5.4 | Successful Tenderer is required to assign all licences as described in the terms and conditions. | | |
| 3.6-25 | 3.6.5.5 | Successful Tenderer is required to provide certificate(s) of warranty and support (where applicable) to the Customer with the rights to such warranty and support assigned to the Customer, | | |
| 3.6-26 | 3.6.5.6 | Successful Tenderer is required to provide access credentials to all online licensing and support associated with the software package to the Customer, and provide proof that he has notified the original software manufacturer of his actions and obtained their agreement to such access being provided, | | |
| 3.6-27 | 3.6.5.7 | Successful Tenderer is required to provide a firm-fixed-price Price Sheet, valid for the duration of Iteration V1, showing the price per-use or user as appropriate, for software packages that are so licensed. This shall also be provided online and prominently displayed when a user first tries to access the package. | | |

5.3.7 Consolidated Requirements Matrix – Section 3.7 - Training

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.7-1 | 3.7.1 | <p>Successful Tenderer is required to deliver all training materials in accordance with the following criteria:</p> <ul style="list-style-type: none"> - suitable for self-study, - indexed from a central training catalogue page. | | |
| 3.7-2 | 3.7.2 | <p>Tenderers are required to:</p> <ul style="list-style-type: none"> - Provide a description of their approach to the preparation of training materials covering: <ul style="list-style-type: none"> o Tools to be used o Processes to be used for quality assurance o Deployment to the DIAS platform | | |
| 3.7-3 | 3.7.3.1 | <p>Successful Tenderer is required to produce a guide that shall give an introduction to the catalogue of data, to searching and to obtaining data with simple HTTPS/HTTP requests</p> <p>Successful Tenderer is required to produce a guide that shall also take the user through a small number of examples of accessing the catalogue, selecting a small number of data sets, and consuming those data sets with curl/wget/Jupyter and in a simple application coded in both Python and R</p> | | |
| 3.7-4 | 3.7.3.2 | <p>Successful Tenderer is required to produce a guide that shall give an introduction to the catalogue of images, to the associated flavours, security groups and required data volumes.</p> <p>Successful Tenderer is required to produce a guide that shall also take the user through a small number of examples of accessing the catalogue, selecting a virtual machine image, selecting a flavour, selecting a security group, selecting any necessary data volumes and initiating a virtual machine.</p> <p>Successful Tenderer is required to produce a guide that shall then take the user through basic user access to the new virtual machine using secure shell (SSH) and the tenderers chosen secure GUI access method (examples being, VNC through a tunnel or HPE RDS).</p> | | |
| 3.7-5 | 3.7.3.3 | <p>Successful Tenderer is required to produce a guide that shall give a basic introduction to the main tools available in each of the virtual machine images.</p> <p>Successful Tenderer is required to produce a guide that shall then take the user through basic use of each of those tools using pre-loaded data sets where appropriate.</p> | | |
| 3.7-6 | 3.7.3.4 | <p>Successful Tenderer is required to produce a guide that shall give a basic introduction to the Apache Spark toolset.</p> <p>Successful Tenderer is required to produce a guide that shall then take the user through basic use of Apache Spark on a single node and on a small cluster including installing and configuring the software (as necessary).</p> | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | Successful Tenderer is required to produce a guide that shall also introduce the user to customising the supplied Virtual Machine Images and/or Container Images for achieving more sophisticated automatic deployments. | | |

5.3.8 Consolidated Requirements Matrix – Section 3.8 - Support

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 3.8-1 | 3.8.1 | The Successful Tenderer is required to maintain and support the products of work packages WP1 through WP7 during the Iteration V1 operational lifecycle which will last until the end of Q4 2019. The tenderer is required to offer an option to extend this support period by a further two (2) periods of three (3) months each if subsequently requested by ECMWF. | | |
| 3.8-2 | 3.8.2 | Tenderers are required to describe in their tenders how they would construct a Support Operation satisfying the requirements of this work package. | | |
| 3.8-3 | 3.8.3 | Our partner, Mercator Ocean International has primary responsibility for user support and the helpdesk function. However, the Successful Tenderer is required to provide 2 nd and 3 rd line support where tickets raised at Mercator Ocean International helpdesk cannot be resolved there. It is expected that Mercator Ocean will have the necessary filtering in place to separate tickets according to whether they relate to the infrastructure (ITT-1), the Common Data Access API, the Virtual Machine Images or the Big-Data Analytics component (ITT-2). However, it must not be assumed that they can perfectly perform this function or that there will not be cross-ITT fault cases. In cases where tickets are wrongly routed, the Successful Tenderer is required to co-operate with the ITT-1 contractor to achieve a satisfactory resolution of the user's problem. | | |
| 3.8-4 | 3.8.4.1 | Successful Tenderer is required to manage his support resources using a triage system, taking full account at minimum of numbers of users affected, time sensitivity of affected services, and quality of service, but in any case, to provide an initial response within four (4) working hours as per Section 3.8.4.13 (Time Zone). | | |
| 3.8-5 | 3.8.4.2 | Successful Tenderer is required to provide Customer with access to his triage system for the purpose of evaluating its effectiveness, | | |
| 3.8-6 | 3.8.4.3 | Successful Tenderer is required to respond to reasonable requests for re-prioritisation of tickets by Customer, | | |
| 3.8-7 | 3.8.4.4 | Successful Tenderer is required to respond to reasonable requests for change to the triage system by Customer, | | |
| 3.8-8 | 3.8.4.5 | Successful Tenderer is required to provide to the user an initial assessment of the cause of a fault together with an intended timeframe for resolution, | | |
| 3.8-9 | 3.8.4.6 | Successful Tenderer is required to inform the user when a timeframe is going to slip and why and / or inform the user when the fault is fixed, and also inform him of any actions that he must take to benefit from the fix such as re-building his own components, | | |

| | | | | |
|--------|----------|---|--|--|
| 3.8-10 | 3.8.4.7 | Successful Tenderer is required to wherever possible provide the user with a workaround that will allow him to carry on with his work pending full resolution of his fault, | | |
| 3.8-11 | 3.8.4.8 | Successful Tenderer is required to escalate fault resolutions to 'upstream' channels in the case of open source software. This is to ensure that faults found and fixes implemented remain in place after package updates. It is understood that there are a number of different regimes for reporting faults and proposing fixes and that it can take a while to have a fix accepted upstream, | | |
| 3.8-12 | 3.8.4.9 | Successful Tenderer is required to escalate fault reports to vendors in the case of proprietary software and act as a go-between to update the user of progress on a timely basis – the user is never to be directly referred to the vendor. A support contract from a vendor does not excuse the ITT-2 contractor from his contractual responsibility to support the DIAS user community, | | |
| 3.8-13 | 3.8.4.10 | Successful Tenderer is required to extend the scope of tests in the testing framework in Section 3.5 to ensure that to the greatest possible extent no regression takes place, | | |
| 3.8-14 | 3.8.4.11 | Successful Tenderer is required to collect user satisfaction feedback and together with all other available support data prepare an analysis, on a monthly basis, of support effectiveness. Tenderer is free to propose his own regime for this; however, the following are some suggested metrics: <ul style="list-style-type: none"> ○ faults reported ○ faults fixed ○ faults deemed unfixable ○ faults fixed and submitted upstream (open source software) ○ faults fixed by vendors (proprietary software) ○ time to fix statistics ○ fault hotspots ○ user community fault reporting statistics (anonymised to GDPR acceptable standards) | | |
| 3.8-15 | 3.8.4.12 | Successful Tenderer is required: <ul style="list-style-type: none"> ○ to provide required training, documentation and support to two (2) Customer members of its staff or Contractors nominated by Customer, and ○ to ensure that sufficient training on all aspects of the DIAS Iteration V1 Platform has been given to those staff to permit them to support and upgrade the platform as necessary from handover to Customer onward | | |
| 3.8-16 | 3.8.4.13 | Successful Tenderer will operate its support operation during normal business hours (08:30 to 17:00) Central European Time (CET). | | |
| 3.8-17 | 3.8.4.14 | Successful Tenderer is required to establish the physical presence of the support function at his own premises for the duration of Iteration V1 support; Customer shall not provide premises, physical assets or personnel. | | |

5.4 Consolidated Requirements Matrix – Section 4 –Management & Implementation

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|---|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| 4-1 | 4.1 | <p>High-Level Planning:</p> <ul style="list-style-type: none"> - EDC - Customer plans to reach contract award / signature for ITT-2 no later than 30th September 2018. However, Customer in its sole discretion, does not guarantee to award a contract by this date or at all. The date given is solely to guide project planning., - FSA – Customer wishes to achieve Final System Acceptance (FSA) by 31st March 2019., - The support phase will cover the period from FSA until the end of 2019. <p>Tenderer acknowledges and accepts the above objectives.</p> | | |
| 4-2 | 4.2 | Tenderers are required to provide a detailed implementation plan of proposed activities for the duration of the Framework Agreement. | | |
| 4-3 | 4.2 | Tenderers are required to propose the level of resources that they consider adequate to implement SC2. Customer shall have the option to adapt the level of resources before or during the execution of SC2, and based on fixed price agreed fees | | |
| 4-4 | 4.3 | <p>Tenderer are required to:</p> <ul style="list-style-type: none"> - appoint a Project Manager who has sole responsibility for the delivery of all deliverables required under this contract taking full account of agreed timescales, quality requirements and price - appoint a Technical Leader who has responsibility together with the Project Manager for understanding and where necessary clarifying the requirements and producing a technical design for negotiation and agreement with Customer - describe his proposed Project Team as follows: <ul style="list-style-type: none"> o roles or functions o minimum experience required and actual experience offered o numbers of individuals in each role or function and duration / level of effort o provide a summary organisation chart | | |
| 4-5 | 4.4 | <p>Customer shall appoint a Technical Leader and a Contract Management Officer.</p> <p>All formal communications with Customer concerning the contract shall be with the Contract Management Officer including formal deliverables.</p> <p>All other communications with Customer concerning matters such as day-to-day project progress, refinement of technical specifications, facilitation requests, progress reports shall be with the Technical Leader with copies to the Contract Management Officer.</p> | | |

| For use by Customer | | | For use by Tenderer | |
|---------------------|-------------|--|---------------------|---|
| Req # | Section Ref | Description or Excerpt from ITT Body | Compliant YES or NO | Tender Cross-Reference & Notes Re: Compliance |
| | | <p>It is well understood that the ITT-2 contractor shall require efficient information flow with the ITT-1 contractor and with Mercator Ocean. Customer shall make all reasonable efforts to facilitate this information flow so as to ensure that the ITT-2 contractor remains able to make progress.</p> <p>Tenderers acknowledge the above.</p> | | |
| 4-6 | 4.5 | <p>Successful Tenderers is required to deliver to Customer by EDC + 2 weeks a Detailed Design showing for each deliverable (as applicable):</p> <ul style="list-style-type: none"> ○ an architecture ○ a component specification ○ all required pre-cursors (i.e. libraries and tools to be used) ○ delivery format of deliverable <p>Upon delivery of the Detailed Design to Customer there shall be a period of 2 weeks of review during which Customer shall have the right to require the Tenderer to make all reasonable efforts to accommodate modifications and/or additions.</p> | | |

END OF DOCUMENT