



OGC Testbed-16
Call for Participation (CFP)

Version 1.10 - 6 February 2020

Table of Contents

1. Introduction	1
1.1. Background	1
1.2. OGC Innovation Program Initiative	1
1.3. Benefits of Participation	1
1.4. Master Schedule	2
2. Technical Architecture	3
2.1. Aviation	4
2.2. Machine Learning	11
2.3. Data Access and Processing API (DAPA) for Geospatial Data	21
2.4. Earth Observation Application Packages with Jupyter Notebooks	29
2.5. GeoPackage	36
2.6. Data Centric Security	39
2.7. Discrete Global Grid System (DGGS)	44
2.8. Analysis Ready Data (ARD)	48
2.9. Federated Security	51
2.10. Full Motion Video to Moving Features	52
2.11. OpenAPI	56
3. Deliverables Summary & Funding Status	59
4. Miscellaneous	61
Appendix A: Testbed Organization and Execution	63
A.1. Initiative Policies and Procedures	63
A.2. Initiative Roles	63
A.3. Types of Deliverables	63
A.4. Proposal Evaluation	64
A.5. Reporting	65
Appendix B: Proposal Submission	66
B.1. General Proposal Submission Guidelines	66
B.2. Questions and Clarifications	67
B.3. Proposal Submission Procedures	67
B.4. Tips for New Bidders	71
Appendix C: Abbreviations	74
Appendix D: Corrigenda & Clarifications	75

Chapter 1. Introduction

The Open Geospatial Consortium (OGC®) is releasing this **Call for Participation** ("CFP") to solicit proposals for the **OGC Testbed-16** (also called "Initiative" or just "Testbed"). The goal of the initiative is to evaluate the maturity of the Earth Observation Cloud Architecture that has been developed over the last two years as part of various OGC Innovation Program (IP) initiatives in a real world environment.



1.1. Background

The OGC Testbed is an annual research and development program that explores geospatial technology from various angles. It takes the **OGC Baseline** into account, though at the same time allows to explore selected aspects with a fresh pair of eyes. The Testbeds integrate requirements and ideas from a group of sponsors, which allows leveraging symbiotic effects and makes the overall initiative more attractive to both participants and sponsoring organizations.

1.2. OGC Innovation Program Initiative

This Initiative is being conducted under the **OGC Innovation Program**. The OGC Innovation Program provides a collaborative agile process for solving geospatial challenges. Organizations (sponsors and technology implementers) come together to solve problems, produce prototypes, develop demonstrations, provide best practices, and advance the future of standards. Since 1999 more than 100 initiatives have taken place.

1.3. Benefits of Participation

This initiative provides an outstanding opportunity to engage with the latest research on geospatial system design, concept development, and rapid prototyping. The initiative provides a business opportunity for stakeholders to mutually define, refine, and evolve service interfaces and protocols in the context of hands-on experience and feedback. The outcomes are expected to shape the future of geospatial software development and data publication. The Sponsors are supporting this vision with cost-sharing funds to partially offset the costs associated with development, engineering, and demonstration of these outcomes. This offers selected Participants a unique opportunity to recoup a portion of their initiative expenses.

1.4. Master Schedule

The following table details the major Initiative milestones and events. Dates are subject to change.

Milestone	Date	Event
M01	23 December 2019	Release of Call for Participation (CFP)
M02	21 January 2020	Questions for CFP Bidders Q&A Webinar Due
M03	28 January 2020	CFP Bidders Q&A Webinar. Register at https://attendee.gotowebinar.com/register/5009861363812363533 . The Webinar starts at 10am EST.
M04	09 February 2020	CFP Proposal Submission Deadline (11:59pm U.S. Pacific Time)
M05	31 March	All CFP Participation Agreements Signed
M06	6-8 April	Kickoff Workshop at US Geological Survey (USGS) National Center, 12201 Sunrise Valley Drive, Reston, Virginia 20192 (https://www2.usgs.gov/visitors/).
M07	31 May	Initial Engineering Reports (IERs)
M08	June	Interim Workshop at TC Meeting Montreal, Canada. Participation not mandatory but appreciated.
M09	30 September	TIE-Tested Component Implementations completed; Preliminary DERs complete & clean, ready for internal reviews
M10	31 October	Ad hoc TIE demonstrations (as requested during the month) & Demo Assets posted to Portal; Near-Final DERs posted to Pending & WG review requested
M11	November (specific date TBD)	Final DERs (incorporating WG feedback) posted to Pending to support WG & TC vote
M12	December (specific date TBD)	Final demonstration at TC meeting
M13	15 December	Participant Final Summary Reports due

Table 1. Master schedule

Chapter 2. Technical Architecture

This section provides the technical architecture and identifies all requirements and corresponding work items. It references the OGC standards baseline, i.e. the complete set of member approved **Abstract Specifications**, **Standards** including **Profiles** and **Extensions**, and **Community Standards** where necessary. Further information on the OGC standards baseline can be found [online](#).

NOTE

Please note that some documents referenced below may not have been released to the public yet. These reports require a login to the OGC portal. If you don't have a login, please contact OGC at techdesk@opengeospatial.org.

Testbed Threads

The Testbed is organized in a number of threads. Each thread combines a number of tasks that are further defined in the following chapters. The threads integrate both an architectural and a thematic view, which allows to keep related work items closely together and to remove dependencies across threads.

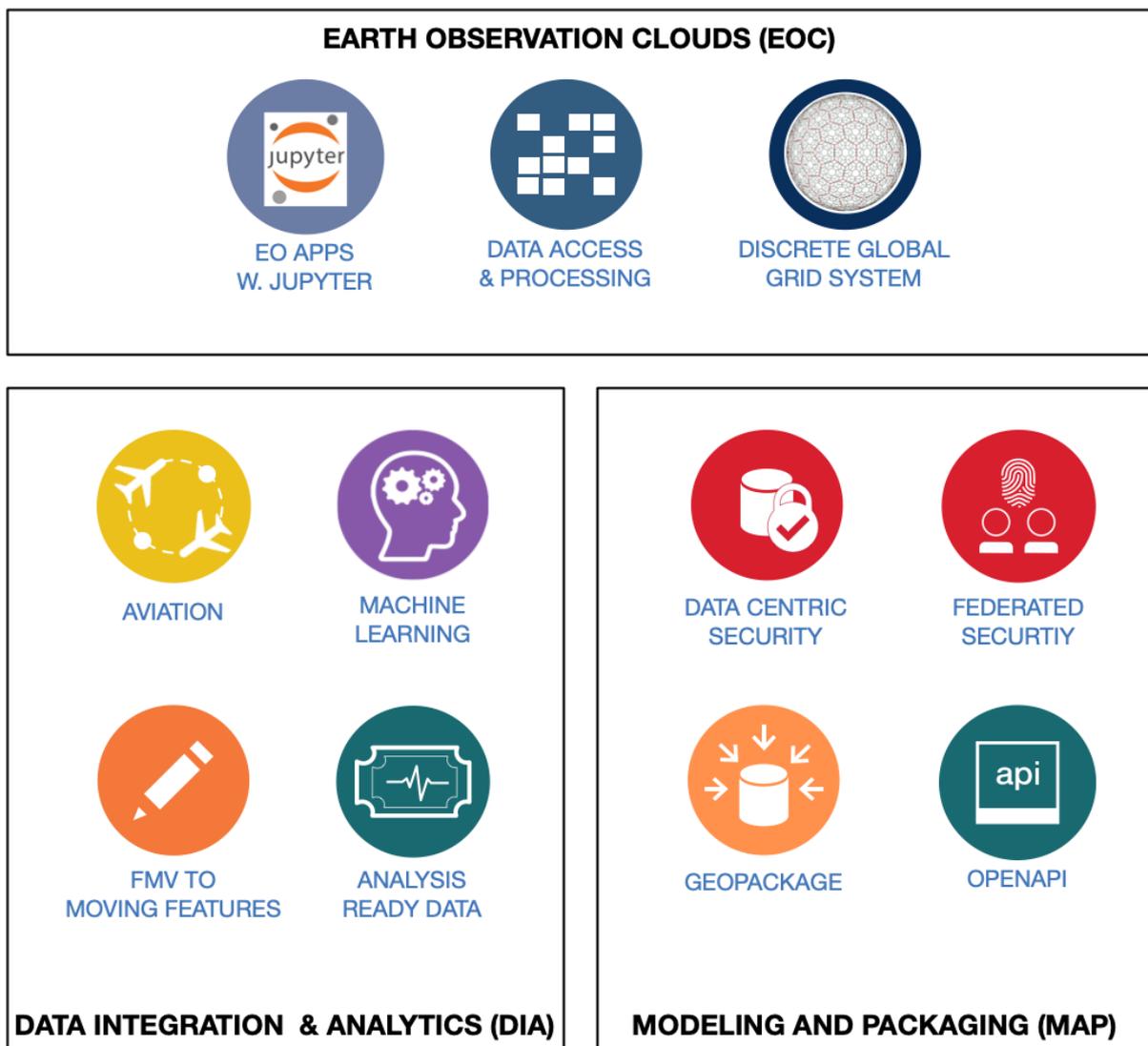


Figure 1. Testbed Threads

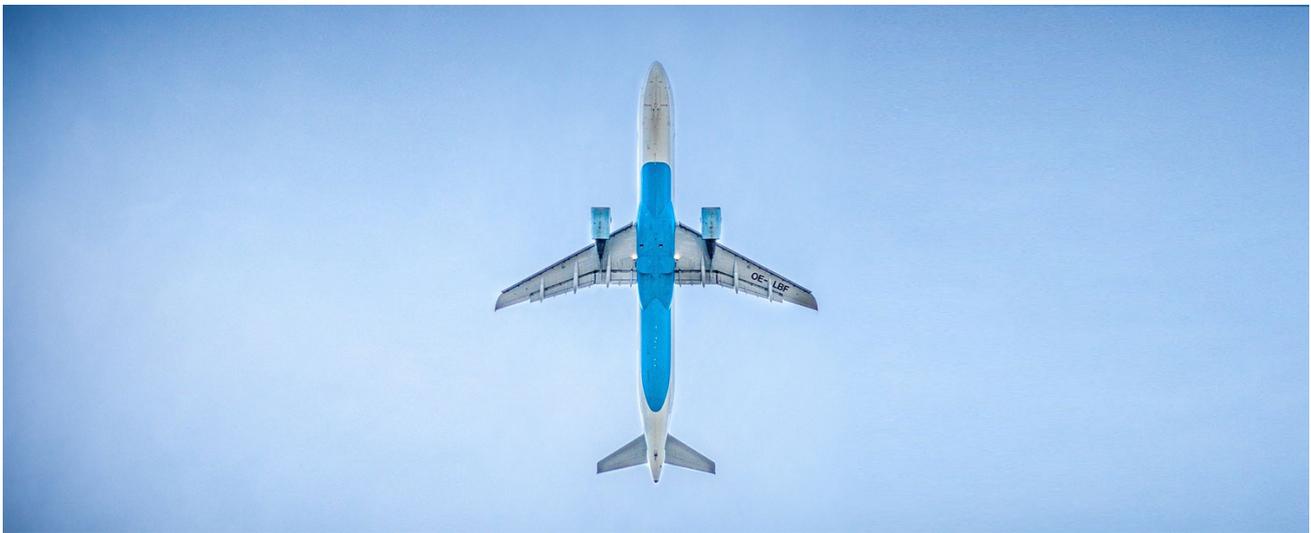
The threads include the following tasks

- Thread 1: **EARTH OBSERVATION CLOUDS (EOC)**
 - Earth Observation Application Packages with Jupyter Notebooks
 - Data Access and Processing API (DAPA) for Geospatial Data
 - Discrete Global Grid System (DGGS)
- Thread 2: **DATA INTEGRATION & ANALYTICS (DIA)**
 - Aviation
 - Machine Learning
 - Full Motion Video to Moving Features
 - Analysis Ready Data (ARD)
- Thread 3: **MODELING AND PACKAGING (MAP)**
 - Data Centric Security
 - Federated Security
 - GeoPackage
 - OpenAPI

2.1. Aviation

The goals of the Aviation task in Testbed-16 are: to evaluate emerging architectural and technological solutions for the **FAA SWIM** services, and to advance further the usage of Linked data in information integration in the SWIM context.

The API modernization work shall evaluate solutions for data distribution that complement those currently used by FAA SWIM. Particular emphasis in this context shall be on OpenAPI-based Web APIs. OGC is busy developing Web APIs for the various geospatial resource types such as features, coverages, maps, tiles, and processes among others. There are many documented benefits for using Web APIs in the context of geospatial data retrieval and processing, including faster time to market for products, more flexibility in deployment models, and straight forward upgrade paths as standards evolve.



The linked data aspect shall explore the use and benefit of semantic web technologies in the context of FAA SWIM services. Linked data shall help querying and accessing all required data for any given task and help with the heterogeneous semantics introduced by the various ontologies and taxonomies used within the aviation community. These include for example the SWIM Controlled Vocabulary ([SWIM CV](#)), the Web Service Description Ontological Model ([WSDOM](#)), or [semantics.aero](#).

Background

The System-Wide Information Management (SWIM) program, established and maintained by the FAA, supports the sharing of Air Traffic Management (ATM) information by providing communications infrastructure and architectural solutions for identifying, developing, provisioning, and operating a network of highly-distributed, interoperable, and reusable services.

As part of the SWIM architecture, data providers create services to access their data. For the FAA, these services are published in the NAS Services Registry/Repository (NSRR). NSRR is a catalog of all SWIM services and provides documentation on various aspects of each service, including its provider, functionality, quality characteristics, interface, and implementation.

One of the SWIM challenges is the handling of semantics across all participants. A diverse set of ontologies, controlled vocabularies, and taxonomies has been developed over the last decade. A good overview is provided in [OGC 18-035](#).

The SWIM Controlled Vocabulary ([SWIM CV](#)) provides to SWIM organizations, support contractors, vendors, and business partners a uniform understanding of terms employed in the SWIM environment. The CV contains a comprehensive list of terms with clear and unambiguous definitions. Each term is globally uniquely identified by a dereferenceable URI so that it can be related semantically to other terms, vocabularies, or resources. SWIM CV is now part of [semantics.aero](#).

[semantics.aero](#) is an open repository for use by the international aviation community to publish artifacts developed using Semantic Web technologies. Besides the SWIM CV, artifacts include taxonomies for classifying services by product type, availability, flight phase, ICAO region, etc. and are available in both human-readable (HTML) and machine-readable (RDF) versions.

The Web Service Description Ontological Model (**WSDOM**) is an ontology intended to be a basis for model-driven implementation of SOA-related artifacts. The ontology has been developed in Web Ontology Language (OWL) version 1.1; it consists of several files and is currently available in a single downloadable zip file. WSDOM can be considered an "RDF" realization of the Service Description Conceptual Model (SDCM). WSDOM standardizes information and metadata pertinent to describing SWIM services and facilitates interchange of service data between service providers and the FAA. The intent behind the ontology is to make service definitions clear, unambiguous, and discoverable by both humans and computer systems. WSDOM consists of ontology classes covering the key notions of service profile, service interface, service implementation, stakeholder, and document. WSDOM is patterned after the OWL-S semantic web services description ontology.

WSDOM was developed before SDCM and was written in OWL. For many people in the industry, OWL ontology was too technical to be understood by a large audience. To address this gap and make the service description more "readable", the Service Description Conceptual Model was created. That said, SDCM 2.0 is more recent than WSDOM 1.0, and therefore better reflects the current NSRR data structure. WSDOM 2.0 is currently being developed and not been aligned yet with the latest version of SDCM.

2.1.1. Problem Statement and Research Questions

FAA has invested in setting up **SWIM Feeds** that are accessible on a feed by feed basis. Each feed was designed as stand-alone. However, the value of data increases when it's combined with other data. In addition, real-world situations are often not related one to one with a SWIM feed (or a single data source for that matter). Therefore, Testbed-16 shall investigate data integration options based on semantic web technologies and analyze the current status of achievable interoperability. The latter is the basis for analytics, querying and visualization of data coming from distributed sources.

Research questions:

1. Should the existing SWIM architecture be modernized with resource-oriented Web APIs?
2. What role can OGC Web APIs play for a modernized SWIM services?
3. How can OGC APIs be used to address the heterogeneous semantic SWIM landscape?
4. What impact have linked data principles and requirements on OGC Web APIs?
5. How to deal with the various ontologies and taxonomies used in SWIM?
6. How to best enhance the various ontologies and how to build a scalable geospatial definition server?
7. How to best combine data from various SWIM data feeds to make it available for multi-source and linked data based analytics?

2.1.2. Aim

This Testbed-16 task aims at a better understanding on the value of modern Web APIs in the context of SWIM service integration and the potential of semantic web technologies to solve complex queries.

2.1.3. Previous Work

The topic of semantic-enablement of models used in the domain of aviation has been explored previously in OGC Testbeds (OGC 16-039, OGC 17-036). In past demonstrations, analyses recommended the use of run-time registries and complex use cases for service discovery and data taxonomy/ontology. However, much of the information exchanged within the System-Wide Information Management (SWIM) network is made up of various data models using XML schema encoding (such as AIXM), which addresses only the structure and syntax of information exchanged between systems, but not the semantic aspect of the model. Testbed-12 and -13 have made progress toward the semantic enablement of the controlled vocabularies using the Simple Knowledge Organization System (SKOS) encoding but these vocabularies were still referred from the XML document based on structure and syntax. This hybrid approach does not allow the usage of off-the-shelf solutions for Linked Data such as linking heterogeneous domain entities, deductive reasoning and unified access to information. Systems are currently built around specific data models and unable to communicate and link to each other causing duplication of information and making difficult to search and discover information that are relevant for users.

Testbed-14 formulated an approach to semantic-enable the different data models, taxonomy and service description that can incorporate semantic metadata. These metadata includes descriptive metadata, geospatial-temporal characteristics, quality information, and fitness-of-use information about services and data. This additional metadata enables the integration of information and services, improves search and discovery, and increases the level of possible automation (e.g. by reasoning or access and processing by agents).

Aviation activities have been part of several initiatives in the past. The following list provides an overview. Links to the relevant Engineering Reports are provided further below.

1. Testbed-15
 - a. OGC Testbed-15: Semantic Web Link Builder and Triple Generator
2. Testbed-14:
 - a. SWIM Information Registry
 - b. Semantically Enabled Aviation Data Models
3. Testbed-13:
 - a. Aviation Abstract Quality Model - Data Quality Specification
 - b. Quality Assessment Service
 - c. Geospatial Taxonomies

4. Testbed-12:
 - a. Aviation Semantics
 - b. Catalog Services for Aviation

The following Engineering Reports are relevant in the context of this task.

- [OGC 18-022r1](#), OGC Testbed-14: SWIM Information Registry Engineering Report
- **[OGC 18-035](#), OGC Testbed 14: Semantically Enabled Aviation Data Models Engineering Report**
- [OGC 17-036](#), OGC Testbed-13: Geospatial Taxonomies Engineering Report
- [OGC 17-032r2](#), OGC Testbed-13: Aviation Abstract Quality Model Engineering Report
- [OGC 16-018](#), OGC Testbed-12: Aviation Architecture Engineering Report
- [OGC 16-024r2](#), OGC Testbed-12: Catalog Services for Aviation Engineering Report
- [OGC 16-028r1](#), OGC Testbed-12: FIXM GML Engineering Report
- **[OGC 16-039r2](#), OGC Testbed-12 Aviation Semantics Engineering Report**
- [OGC 16-061](#), OGC Testbed-12 Aviation SBVR Engineering Report

Reports that addressed linked data and semantic web technologies include:

- OGC 19-021, OGC Testbed-15: Semantic Web Link Builder and Triple Generator (draft version available on [OGC portal](#) or upon request)
- [OGC 18-094r1](#), OGC Testbed-14: Characterization of RDF Application Profiles for Simple Linked Data Application and Complex Analytic Applications Engineering Report
- [OGC 18-032r2](#), OGC Testbed-14: Application Schema-based Ontology Development Engineering Report
- [OGC 17-018](#), OGC Testbed-13: Data Quality Specification Engineering Report
- [OGC 16-046r1](#), OGC Testbed-12 Semantic Enablement Engineering Report

2.1.4. Scenario & Requirements

[Figure 2](#) illustrates the current SWIM situation. A number of services are available that use different taxonomies, vocabularies, and ontologies. SWIM services are described in the service registry. All services are built on the Service Oriented Architecture (SOA).

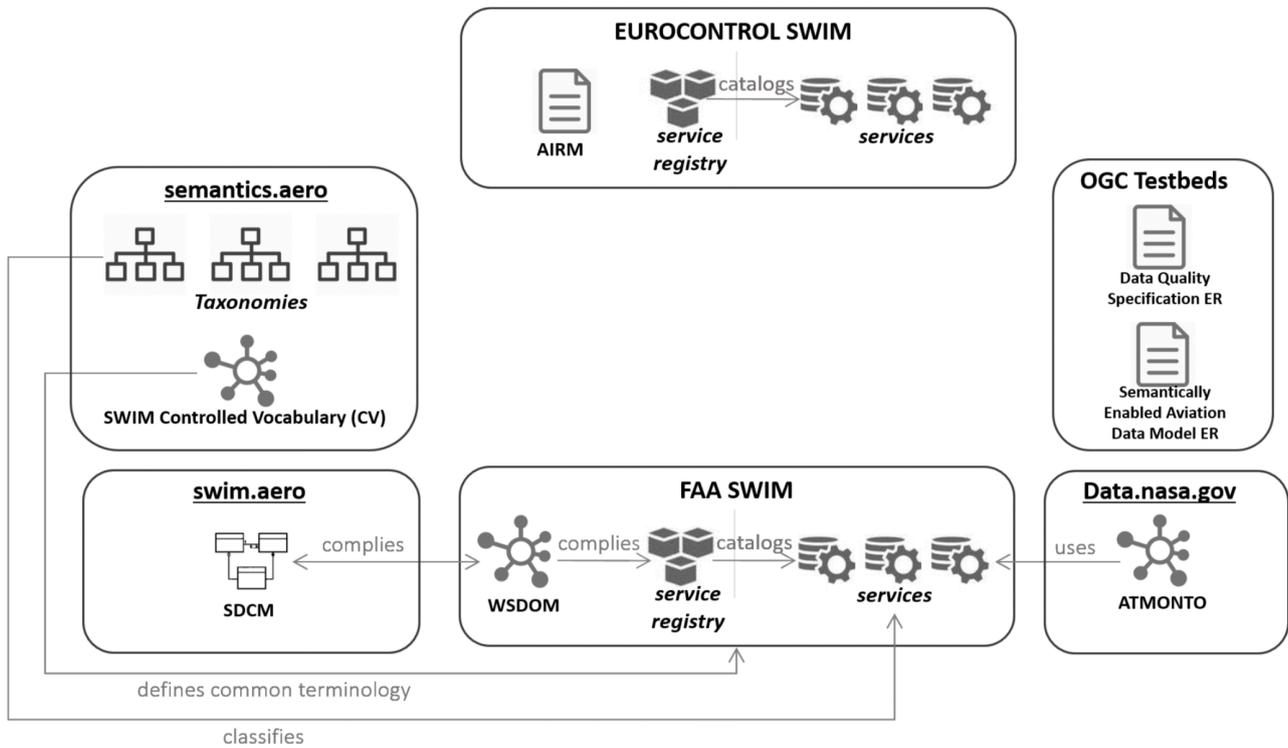


Figure 2. Aviation scenario, current situation

The current situation illustrated above shall be explored along two axes. First, the value and role of Web APIs and here more specifically OGC APIs for SWIM. Second, the handling of semantics and integration of data from various services, as illustrated in [Figure 3](#).

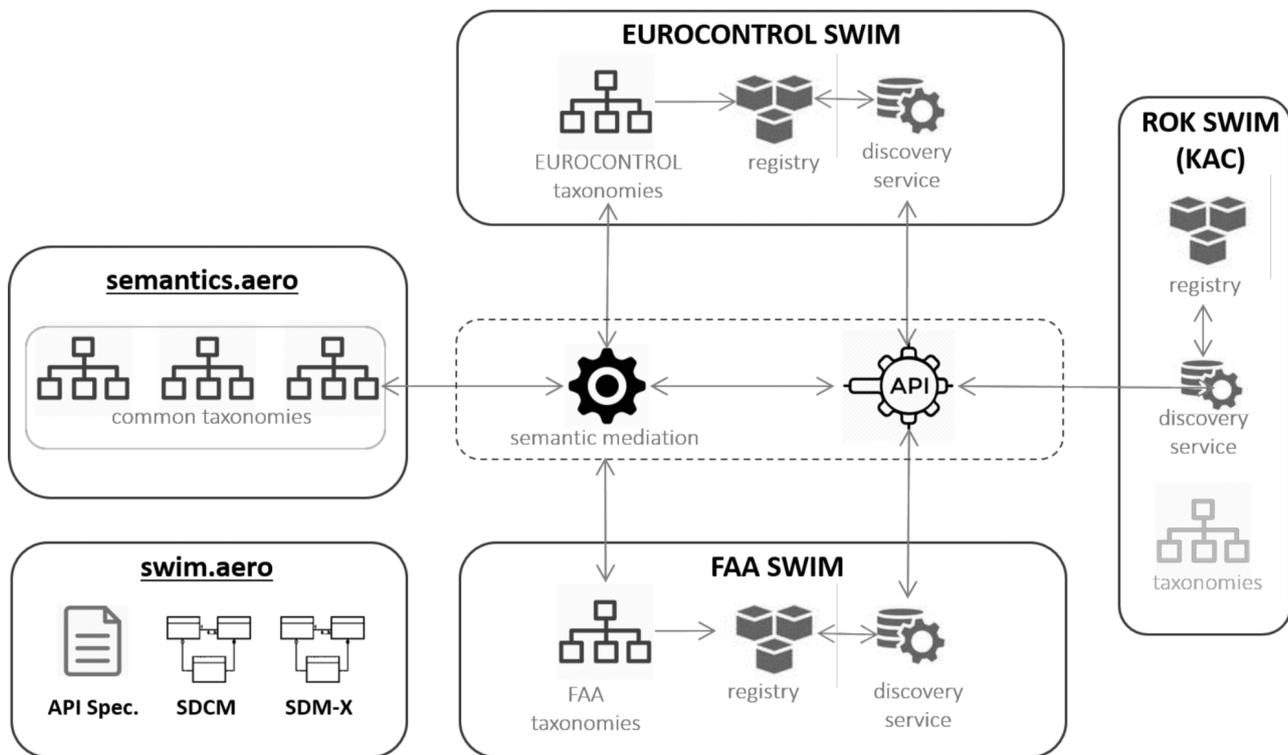


Figure 3. Example of deploying API and Semantic Technology in today's Global SWIM

[Figure 3](#) depicts a scenario where diversified SWIM initiatives use API and semantic mediation for integrating service meta-information collected by their respective registries.

The Testbed-16 aviation scenario will address the integration of SWIM data from various sources to answer complex queries, such as exemplarily:

- Which flights from IAD to any airport in Europe have not been subject of GDP Advisories in the last 2 hours?
- What is the closest airport in Florida to land a flight from IAD, given a Temporary Flight Restriction due to a hurricane?

2.1.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

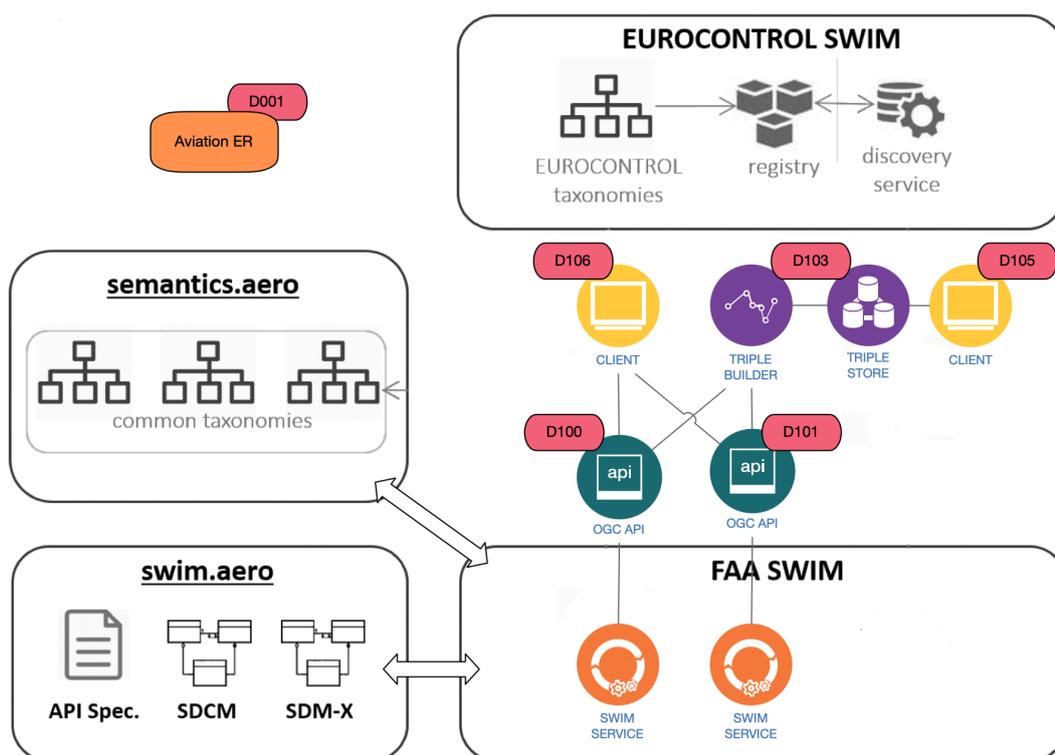


Figure 4. Aviation task architecture and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D001 Aviation Engineering Report** - Engineering Report capturing all results and experiences from this task. The report shall provide answers to all research questions and document implementations.

Components

- **D100 OGC API**- Service endpoint for a SWIM service. The endpoint shall support sufficient semantics that allow a triple builder to link data from various API endpoints.

- **D101 OGC API**- Similar to D100
- **D103 Triple Builder**- The triple builder shall use all OGC APIs to generate links between data and services. It shall support answers to complex queries such as exemplarily provided above. The triple builder shall store all triples in a triple store providing at least a GEOSPARQL interface. Additional entry paths between the triple store and the client D106 can be defined during the project.
- **D105 Semantic Web Client**- Client application to interact with the triple store to answer complex queries as provided above. Ideally, the client provides a graphical user interface that illustrates results on a map.
- **D106 SWIM OGC API Client**- Client application to interact with the OGC APIs that front-end SWIM services. Ideally, the client provides a graphical user interface that illustrates results on a map.

2.2. Machine Learning

The Machine Learning task focuses on understanding the potential of existing or new OGC standards for supporting Machine Learning (ML) applications in the context of wildland fire safety and response. In this context, the integration of ML models into standards-based data infrastructures, the handling of ML training data, and the integrated visualization of ML data with other source data shall be explored. Emphasis is on the integration of data from the [Canadian Geospatial Data Infrastructure \(CGDI\)](#), the handling of externally provided training data, and the provisioning of results to end-users without specialized software.



Figure 5. Photo by Matt Howard on Unsplash

Wildland fires are those that occur in [forests, shrublands and grasslands](#). While representing a natural component of forest ecosystems, wildland fires can present risks to human lives and infrastructure. Being able to properly plan for and respond to wildland fire events is thus a critical component of forestry management and emergency response.

Appropriate responses to wildland fire events benefit from planning activities undertaken before events occur. ML presents a new opportunity to advance wildland fire planning using diverse sets of geospatial information (e.g. satellite imagery, Light Detection and Ranging (LiDAR) data, land cover information, building footprints). As much of the required geospatial information is made available using OGC standards, a requirement exists to understand how well these standards can support ML in the context of wildland fire planning.

Testbed-16 will explore how to leverage ML, cloud deployment and execution, and geospatial information, provided through OGC standards, to improve planning approaches for wildland fire events. Findings of the work will also inform future improvement and/or development activities for OGC standards, leading to improved potential for the use of OGC compliant data within ML applications.

Advanced planning for wildland fire events can greatly improve the ability of first responders to address a situation. However, it is very difficult to account for the many variables (e.g. wind, dryness, fuel loads) and their combinations, that will be present at the exact time of an event. As such, there is an opportunity to evaluate how ML approaches, combined with geospatial information delivered using OGC standards, can improve response planning throughout the duration and aftermath of wildland fire occurrences.

Thus, in addition to planning related work, Testbed-16 shall explore how to leverage ML technologies for dynamic wildland fire response. It will also provide insight into how OGC standards can support wildland fire response activities in a dynamic context. Any identified limitations of existing OGC standards will be used to plan improvements to these frameworks. An opportunity also exists to explore how OGC standards may be able to support the upcoming [Canadian WildFireSat mission](#).

The [Canadian Wildland Fire Information System \(CWFIS\)](#) provides further information about wildland fires in Canada. It creates daily fire weather and fire behavior maps year-round and hot spot maps throughout the forest fire season, generally between May and September.

IMPORTANT

Though this task uses a wildland fire scenario, the emphasis is not on the quality of the modelled results, but on the integration of externally provided source and training data, the deployment of the ML model on remote clouds through a standardized interface, and the visualization of model output!

2.2.1. Problem Statements, Requirements, and Research Questions

Testbed-16 shall address the following three challenges.

1. Discovery and reusability of training data sets
2. Integration of ML models and training data into standards-based data infrastructures
3. Cost-effective visualization and data exploration technologies based on [Map Markup Language \(MapML\)](#)

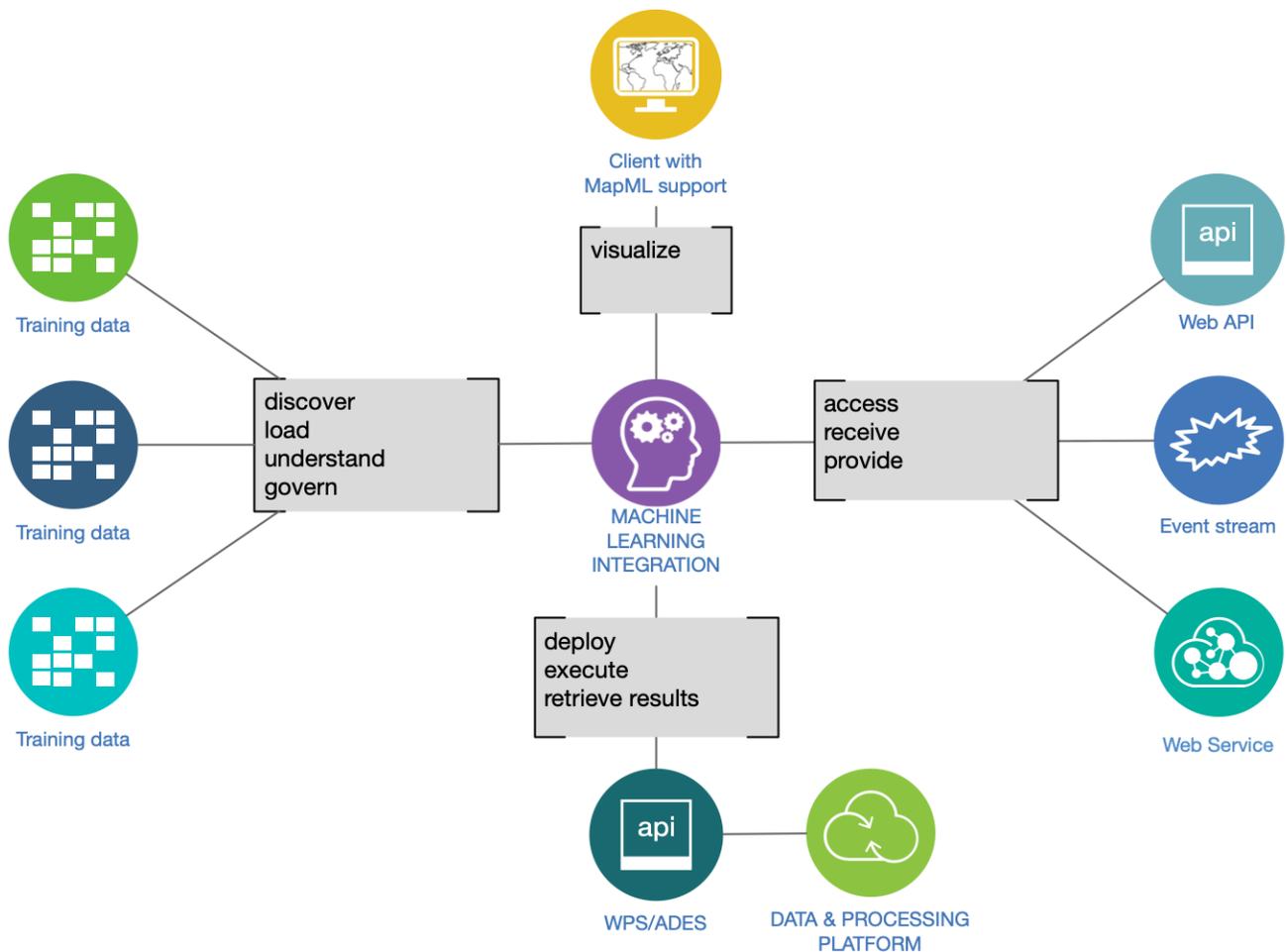


Figure 6. ML and EO integration challenges; various training data sets that need to be discovered, loaded, and interpreted (left); integration of (live) sources from Web APIs, event streams, or Web Services (right); and visualization (top)

Training Data Sets

We currently have unprecedented Earth Observation (EO) capabilities at hand. To combine these with the major advances in artificial intelligence in general and ML in particular, we need to close the gap between ML on one side and Earth observation data on the other. In this context, two aspects need to be addressed. First, the extremely limited discoverability and availability of training and test datasets, and second, interoperability challenges to allow ML systems to work with available data sources and live data feeds coming from a variety of systems and APIs.

In this context, training datasets are pairs of examples of labelled data (independent variable) and the corresponding EO data (dependent variables). Together, these two are used to train an ML model that is then used to make predictions of the target variable based on previously unseen EO data. Test data is a set of observations used to evaluate the performance of the model using some performance metric. In addition to the training and test data, a third set of observations, called a validation or hold-out set, is sometimes required. The validation set is used to tune variables called hyper parameters, which control how the model learns. In the following paragraphs, the set of training data, test data, and validation data are together referred to simply as Training Data Sets (TDS).

To address the general lack of training data discoverability, accessibility, and reusability,

Testbed-16 shall develop solutions that describe how training data sets shall be generated, structured, described, made available, and curated.

Integration of (Live) Data

The second aspect addresses the integration of data in ML model runs. This includes data available at Web APIs or Web services, and event streams.

Geospatial information required for wildland fire planning and response is commonly obtained from central data repositories. In Canada, large and well-known geospatial repositories, such as the [Earth Observation Data Management System \(EODMS\)](#), the Federal Geospatial Platform / [Open Maps](#) and [Canada's National Forest Information System \(NFIS\)](#) provide vast quantities and types of reputable geospatial data through OGC standards. However, these systems have generally not been designed to support advanced ML applications, especially within an emergency planning/response context. This component of the work aims to determine how well these systems can support ML applications in the context of OGC standards. It will also provide initial insight into the readiness of fundamental components of the Canadian Geospatial Data Infrastructure (CGDI) for supporting new technologies such as ML. It will give actionable recommendations as to how CGDI geospatial information repositories can be improved to better support ML applications. Potential improvements to OGC standards in the context of geospatial data repositories and extreme events will also be identified.

Deployment and Execution of ML Models

All machine learning models shall be deployed and executed on cloud platforms offering a specialized WPS interface known as Application Deployment and Execution Service (ADES). ADES has been developed in Testbed-13/14 and allows any type of application that is packaged as a Docker container and made available on a Docker Hub to be deployed and executed in a cloud environment. The WPS will be made available. To use that WPS, model providers need to package their model together with all necessary auxiliary data (training data, configuration files, etc.) in a Docker Container, create an Application Package description of that container, and submit it to the transactional ADES. Support for deployment and operation as well as all necessary cloud resources will be provided by Testbed-16 Sponsors. Further information on the ADES is provided in section *Previous Work*, [Earth Observation Cloud Processing](#).

Visualization of ML Results

With planning and response activities for wildland fire events, it is critical that stakeholders (e.g. planners, first responders, residents, policy makers) are able to visualize related geospatial information quickly and accurately. Currently, such visualization requires users to have access to specialized software and skills. These barriers shall be reduced using tools supporting MapML. When implemented, MapML allows for viewing and interaction with geospatial information within web browsers. With widespread availability of web browsers on multiple devices, intuitive user interfaces, and no cost constraints, making geospatial information available through MapML has the potential to revolutionize how we interact with geographic information.

This Testbed-16 task shall determine the utility of MapML for providing a geospatial information

visualization and interaction interface in the context of wildland fire planning and response. Findings will allow the sponsor to determine if MapML would provide a benefit to wildland fire stakeholders, including what improvements may be required. It will also aim to increase the visibility of MapML as a practical tool for geospatial visualization and interaction. Potential improvements to OGC standards to further leverage MapML capabilities will also be identified.

To be more precise, all geospatial information results from the wildland fire planning and response components shall be published, so that they incorporate MapML capability. The delivery of MapML has been explored within Testbed-13 and Testbed-14. The [Testbed-13 MapML Engineering Report \(OGC 17-019\)](#) recommends to use Web Map Service (WMS) or Web Map Tile Service (WMTS) as services that can be used to deliver MapML documents with small modifications. Other options arise from using OGC Web APIs as developed in Testbed-15. The sponsor requires MapML be implemented in a Web browser to publish results. Several open source web browser engines exist that can be leveraged for this work (e.g. [WebKit](#), [Gecko](#), and [Blink](#)). JavaScript implementations for MapML are **not** to be used for final result delivery.

Further on, Testbed-16 shall investigate the ability of MapML to operate on mobile devices in mobile environments.

A comparison of MapML to other visualization and interaction tools shall complement this work. Here, Testbed-16 shall compare and contrast the ability of MapML to act as an operational stakeholder geospatial information visualization and interaction tool with current approaches used within a wildland fire context. The sponsor is particularly interested in exploring the ability of MapML to federate between authoritative sources (e.g. municipal/territorial/Indigenous/provincial/state/federal governments, international organizations, etc.).

If applicable, Testbed-16 shall provide actionable recommendations for MapML improvement to better support extreme event advanced planning and active response.

Overview of Task Activities

The following diagram provides an overview of the main work items for this Testbed-16 task; with the training data at the bottom, existing platforms and corresponding APIs to the left, and Machine Learning models and visualization efforts to the right.

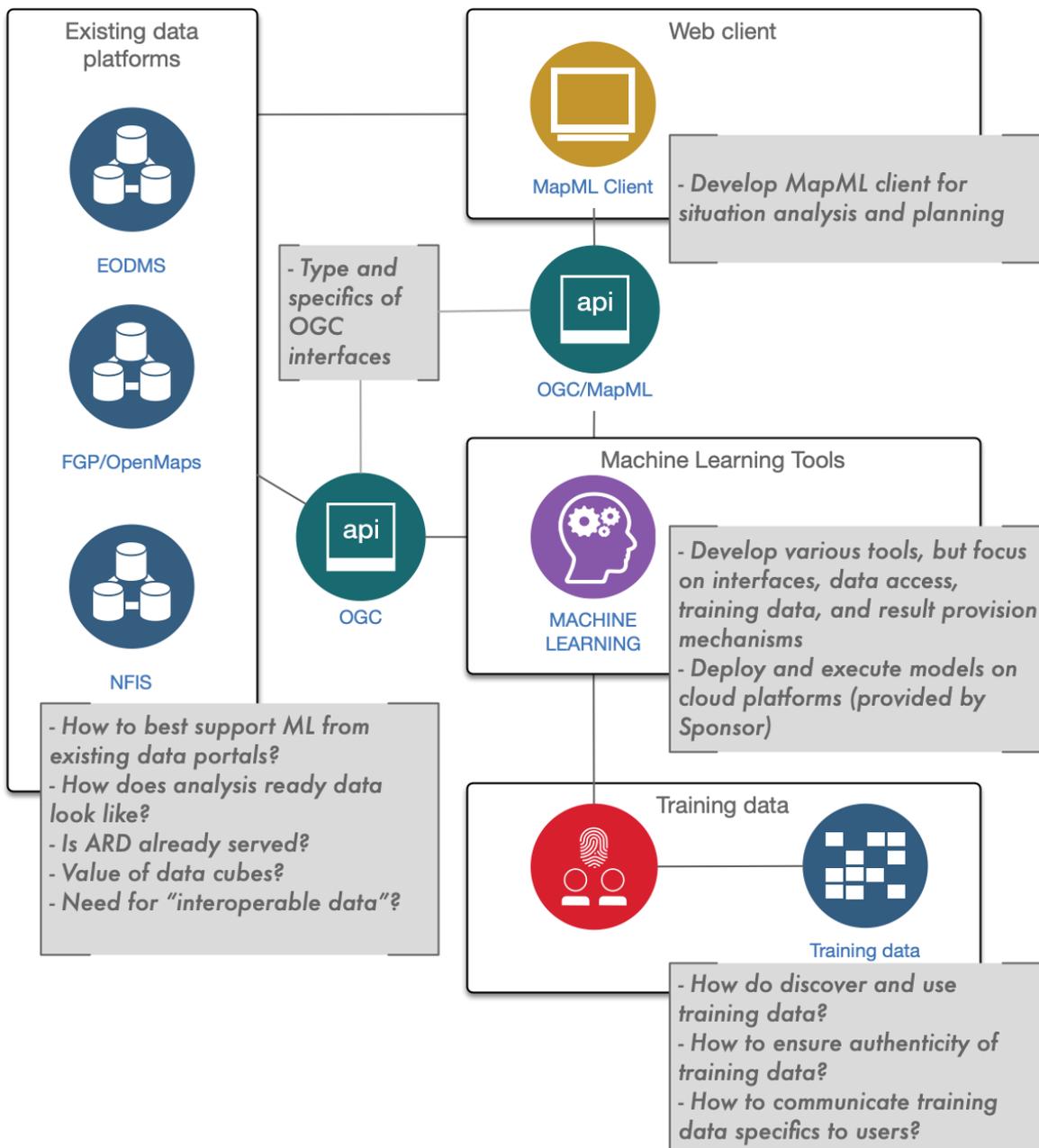


Figure 7. Major components and research aspects of the Machine Learning task

The platforms to the left in the **ML scenario** are all operational. EODMS supports multiple services including WCS, CSW, WFS and WPS. The **FAQ: Can I access EODMS using an API** provides further information. **FGP / Open Maps** supports WFS, WMS, and WMTS. It also supports ESRI REST. **NFIS** supports various interfaces; documentation for all platforms is accessible online.

Research Questions

The following overarching research questions shall further help to guide the work in this task:

- Does ML require "data interoperability"? Or can ML enable "data interoperability"? How do existing and emerging OGC standards contribute to a data architecture flow towards "data interoperability"?
- Where do trained datasets go and how can they be re-used?

- How can we ensure the authenticity of trained datasets?
- Is it necessary to have analysis ready data (ARD) for ML? Can ML help ARD development?
- What is the value of datacubes for ML?
- How do we address interoperability of distributed datacubes maintained by different organizations?
- What is the potential of MapML in the context of ML? Where does it need to be enhanced?
- How to discover and run an existing ML model?

2.2.2. Previous Work

The following reports serve as a baseline for this task:

- [OGC 18-023r1, OGC Testbed-14: MapML ER](#)
- [OGC 17-019, OGC Testbed-13: MapML Engineering Report](#)
- [OGC 18-038r2, OGC Testbed-14: Machine Learning Engineering Report](#)
- [OGC 18-045, OGC Testbed-14: Next Generation Web APIs - WFS 3.0 Engineering Report](#)
- [OGC 19-027, OGC Testbed-15: Machine Learning ER](#)
- [OGC 19-020, OGC Testbed-15: Catalogue and Discovery Engineering Report](#)
- [OGC 19-046, OGC Testbed-15: Quebec Model MapML ER Engineering Report](#)
- [OGC 18-049r1, OGC Testbed-14: Application Package Engineering Report](#)
- [OGC 18-050r1, OGC Testbed-14: ADES & EMS Results and Best Practices Engineering Report](#)
- [OGC 17-024, OGC Testbed-13: Application Deployment and Execution Service Engineering Report](#)

2.2.3. Scenario

The Machine Learning task scenario addresses two phases of wildland fire management, i.e. *Wildland Fire Planning* and *Wildland Fire Response*. For both phases, various steps of training and analysis data integration, processing and visualization shall be executed as outlined below. The scenario serves the purpose of guiding through the various steps in the two phases of wildland fire planning and response. It helps to ground all work in a real-world scenario. Focus shall remain on the requirements listed [above](#).

IMPORTANT

Though this task uses a wildland fire scenario, the emphasis is not on the quality of the modeled results, but on the integration of externally provided source and training data and the visualization of model output!

Annotated training data sets will be provided. Additional datasets can be provided during the

Testbed.

- RADARSAT-1 open Synthetic Aperture Radar (SAR) imagery through EODMS (see [this link](#) for more information).
- NRCan National Air Photo Library through EODMS (see [this link](#) for more information).
- Sample LiDAR datasets covering the Charles H. Herty Pines Nature Preserve in Statesboro, Georgia. Data contains 3D point files, present in ASCII format. Can be easily converted to other formats as needed.
- Additional LiDAR datasets to be provided by NRCan partners in New Brunswick.

The scenario includes the following major steps:

Wildland fire planning:

1. Investigate the application of different ML frameworks (e.g. [Mapbox RoboSat](#), [Azavea's Raster Vision](#), [NRCan's GeoDeepLearning](#)) to multiple types of remotely sensed information (i.e. synthetic aperture radar and optical satellite imagery, LiDAR), provided through OGC standards, to identify fuel availability within targeted forest regions.
2. Explore interoperability challenges of training data. Develop solutions that allow the wildland fire training data, test data, and validation data be structured, described, generated, discovered, accessed, and curated within data infrastructures.
3. Explore the interoperability and reusability of trained ML models to determine potential for applications using different types of geospatial information. Interoperability, reusability and discoverability are essential elements for cost-efficient ML. The structure and content of the trained ML models have to provide information about its purpose. Questions such as: "What is it trained to do?" or "What data was it trained on?" or "Where is it applicable?" need to be answered sufficiently. Interoperability of training data should be addressed equivalently.
4. Deep Learning (DL) architectures can use LiDAR data to classify field objects (e.g. buildings, low vegetation, etc.). These architectures mainly use the TIFF and ASCII image formats. Other DL architectures use 3D data stored in a rasterized or voxelized form. However, 3D voxels or rasterized forms may have many approximations that make classification and segmentation vulnerable to errors. Therefore, Testbed-16 shall apply advanced DL architectures directly to the raw point cloud to classify points and segments of individual items (e.g. trees, etc.). Participants shall use the [PointNET](#) architecture for this or propose different approaches. If different DL architectures are proposed, the sponsor will consider them as an alternative to PointNET. Sponsor approval will be required before a different architecture can be used.
5. Leverage outcomes from the previous steps to predict wildland fire behavior within a given area through ML. Incorporate training of ML using historical fire information and the Canadian Forest Fire Danger Rating system (fire weather index, fire behaviour prediction) leveraging weather, elevation models, fuels.
6. Using ML to discover and map suitably sized and shaped water bodies for water bombers and helicopters

7. Investigate the use of ML to develop smoke forecasts based on weather conditions, elevation models, vegetation/fuel and active fires (size) based on distributed data sources and datacubes using OGC standards.

Wildland fire response:

For the wildland fire response phase, the following aspects should be considered:

1. Explore ML methods for identifying active wildland fire locations through analysis of fire information data feeds (e.g. the Canadian Wildland Fire Information System, the United States Geological Survey LANDFIRE system) and aggregation methods. Explore the potential of MapML as an input to the ML process and the usefulness of a structured Web of geospatial data in this context.
2. Implement ML to identify potential risks to buildings and other infrastructure given identified fire locations. Potential for estimating damage costs.
3. Investigate how existing standards related to water resources (e.g. WaterML, Common Hydrology Features (CHyF), in conjunction with ML, can be used to locate potential water sources for wildland fire event response.
4. Develop evacuation and first responder routes based on ML predictions of active fire behaviour and real-time conditions (e.g. weather, environmental conditions).
5. Based on smoke forecasts and suitable water bodies, determine if suitable water bodies are accessible to water bombers and helicopters.
6. Explore the communication of evacuation and first responder routes, as well as other wildland fire information, through Publication/Subscription (Pub/Sub) messaging.
7. Examine how ML can be used to identify watersheds/water sources that will be more susceptible to degradation (e.g. flooding, erosion, poor water quality) after a fire has occurred.
8. Identify how OGC standards and ML may be able to support the goals of the upcoming [Canadian WildFireSat mission](#).

2.2.4. Work Items & Deliverables

The [following figure](#) illustrates the work items of this task and identifies deliverables.

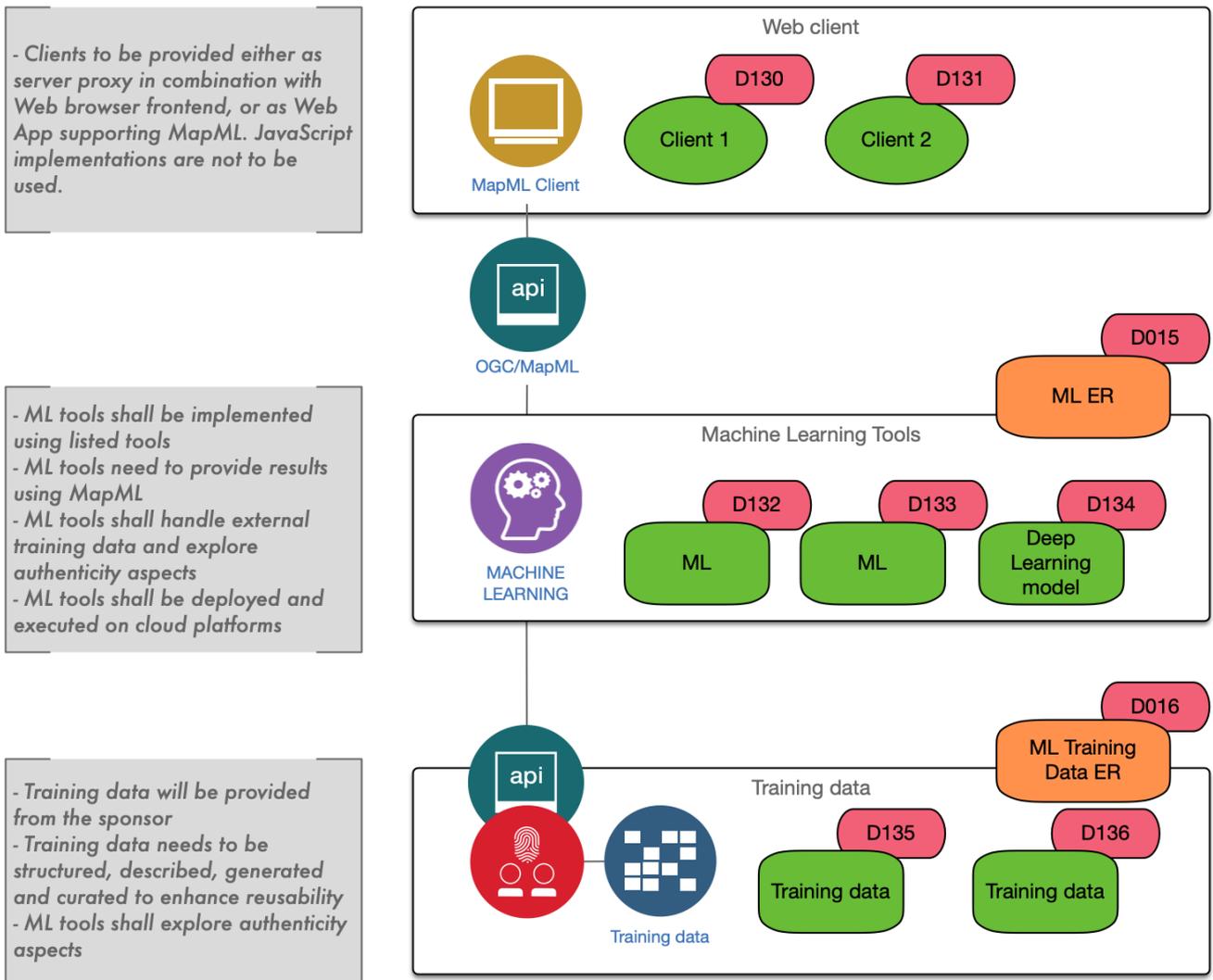


Figure 8. Machine Learning work items (client, machine learning tools, training data) and deliverables (green with numbered identifiers)

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions and to provide contributions to the Engineering Reports. Thread assignment and funding status are defined in section [Summary of Testbed Deliverables](#). Some work items are identical to facilitate Technology Integration Experiments (TIEs).

Engineering Reports

- **D015 Machine Learning Engineering Report** - Engineering Report capturing all results and experiences from this task. It shall respond to all requirements listed above. The Engineering Report shall contain a plain language executive summary to clearly outline the motivations, goals, and critical outcomes of this task, taking into account the mandates of the OGC and the sponsor(s).
- **D016 Machine Learning Training Data Engineering Report** - Engineering Report describing the training data metadata model, structure, file format, media type, and its integration into Spatial Data Infrastructures and SDI-based Machine Learning tools, which includes discovery, access, and authenticity evaluation.

Components

- **D130 MapML Client 1** - MapML client, to be provided either as server proxy in combination with Web browser frontend, or as Web App supporting MapML. JavaScript implementations are not to be used.
- **D131 MapML Client 2** - Similar to D130.
- **D132 Machine Learning Environment 1** - An ML framework such as e.g. [Mapbox RoboSat](#), [Azavea's Raster Vision](#), [NRCAN's GeoDeepLearning](#) with support for OGC Web Services or OGC Web APIs to retrieve externally provided data as described [above](#); and to provide results at OGC Web APIs or OGC Web service interfaces in a form that allows MapML clients to explore results. The model shall be deployed and executed on a cloud platform that provides a ADES interface for easy deployment and execution. The Preference will be given to contractors that will make the configured ML framework available to the sponsor at the end of the project. Ideally, this happens in the form of scripts that build a docker instance or process, which initializes the ML runs.
- **D133 Machine Learning Environment 2** - Similar to D132
- **D134 Deep Learning Environment** - A DL framework DL architecture based on PointNET or sponsor-approved equivalent. The model shall be deployed and executed on a cloud platform that provides a ADES interface for easy deployment and execution. Preference will be given to contractors that will make the configured Deep Learning Environment available to the sponsor at the end of the project. The environment shall be capable of:
 - Direct application to raw LiDAR point clouds.
 - Classifying points and segments for individual field objects (e.g. trees, etc.)
- **D135 Training Data Set 1** - Training data set including training data, test data, and validation data compliant with the model and definitions defined in D016. The data shall be made available at Web API endpoints.
- **D136 Training Data Set 2** - Similar to D135

2.3. Data Access and Processing API (DAPA) for Geospatial Data

The provider-centric view has defined data retrieval mechanisms for geospatial data in the past. As a result, many data access mechanisms are built around powerful web services with rich query languages or file-based download services. Both may result in a sub-optimal experience for the end user, who would often prefer an approach similar to a local function call on a dataset in memory. Testbed-16 shall explore this end-user centric perspective and develop proposals for future standardization work in the context of end-user centric data retrieval and processing APIs. For the end-user, a function call to calculate the minimum temperature value for a target area shall look similar independently of the data location, be it a local file, an in-memory structure (e.g. an xarray), or a remote data set stored on a cloud.



Several data encoding formats are currently in use in OGC, such as NetCDF, GeoTiff, HDF, GML/Observation and Measurements or **variations thereof**, or, increasingly often, JSON encoded data. The different formats exist for various reasons, such as efficiency enhancements for specific domains or use cases, interoperability efforts, or just historical reasons. JSON, for example, is the first format that comes to mind when thinking about sharing data in the internet context; simple to understand and code against, but it may probably be a bad idea to use it with medium sized matrices, since the size can be several times that of a comparable uncompressed GeoTIFF. Testbed-16 shall develop recommendations for data encoding formats that fit the use cases described below. Data encoding formats are used to exchange data. Data storage formats are out of scope for this Testbed. Data can be stored in any format on the provider side. The long term vision of the overall effort is to provide an API that shall be supported by all data providers. In this context, additional aspects such as cloud-friendly or cloud optimized formats (see for example cloud optimized geotiff, **COG**, or **Zarr**) need to be considered, but these are out of scope for this Testbed. A study on this topic is currently executed by NASA and ESA with results expected for early 2020. This study is looking at several EO data formats, both legacy (e.g., netCDF) and cloud-optimized (CoG, Zarr) to see what they offer with respect to supporting analysis in the cloud, as well as aspects of suitability as a storage format. Results from that study should be taken into account.

Though Testbed-16 definitely references the OGC baseline, there is freedom to explore data formats with a fresh pair of eyes. Recommended data formats do not need to be extremely generic and applicable to all possible situations, but shall be easy to handle for end-users and support the requirements defined by the environmental data retrieval use cases.

2.3.1. Problem Statement & Research Questions

This task shall address the following research questions:

1. How does a resource model look like that binds specific functions to specific data?
2. How to realize an end-user optimized data access and processing API (DAPA) that looks like local function calls?
3. What data encoding formats work best in which situation of data retrieval for the use cases defined below?

The first question addresses the need to bind specific data, for example weather data, to a specific set of functions, e.g. access and analytical functions. Not all data is equally suited for all

types of data processing. If the data does not support bands, all band algebra that is traditionally applied to multi-band satellite imagery is void. On the other side, nominally or ordinally scaled data may require different classification and interpolation techniques than rational data and does not allow the same set of mathematical operations. In short, the type of data needs to fit the set of operations. To achieve this need, each endpoint that provides access to data and analytics needs to express the specific combination in the list of data resources advertised by the endpoint. As Web API endpoints work on resources, the underlying resource data model needs to express valid combinations of data and operations.

The second question addresses the need to allow processing that is data storage agnostic. An end-user likes to make the same call for local data as for remote data. Figuratively speaking, an operation to calculate the minimum value within some data structure looks similar for *MyLocalInMemoryData.MIN()*, *MyLocalFile.MIN()*, and *http://your.data/subset.MIN()*. The DAPA shall make full use of the capabilities of the OpenAPI specification, which allows e.g. for CoverageJSON to provide a fully implemented schema of the returned data (less so for large formats, such as NetCDF, or binary formats). That helps first to automate code generation, and second to validate if the returned data follows a specific model.

The third research question addresses the need to better understand how to encode data for transport and exchange. Given that there is no universal answer to this question, the goal is to discuss encodings in the context of different scenarios and user groups.

2.3.2. Aim

The aim of this task is to simplify access to environmental and earth observation data by providing a Data Access and Processing API (DAPA). The API shall be evaluated by scientists and tested with Jupyter Notebook implementations, which can serve as examples for further research work.

2.3.3. Background & Previous Work

The Data Access and Processing API development takes into account several developments and existing best practices for both APIs and data encodings.

On the API side, there are for example [openEO](#), [GeoTrellis](#), or [GeoAPI](#). The European Commission funded research project [openEO](#) is currently developing an open API to connect R, python, javascript and other clients to big Earth observation cloud back-ends in a simple and unified way. [GeoTrellis](#) is a geographic data processing engine for high performance applications. It is implemented as a Scala library and framework that uses Apache Spark to work with raster data and supports many Map Algebra operations as well as vector to raster or raster to vector operations. The [OGC GeoAPI Implementation Standard](#) defines the GeoAPI library. The Java language API includes a set of types and methods that can be used for the manipulation of geographic information that is represented according to ISO and OGC standards.

These APIs are complemented by a set of emerging OGC API standards to handle geospatial data and processes. The OGC API family of (mostly emerging) standards is organized by resource type. So far, [OGC API - Features](#) has been released as a standard that specifies the fundamental API

building blocks for interacting with features. The spatial data community uses the term 'feature' for things in the real world that are of interest. OGC API standards define modular API building blocks to spatially enable Web APIs in a consistent way. The OpenAPI specification is used to define the API building blocks.

On the data encoding side, there are several existing standards that are frequently being used for Earth observation, environmental, ecological, or climate data. These include NetCDF, GeoTiff, HDF, GML/Observation and Measurements or **variations thereof**, or increasingly often JSON encoded data. Testbed-16 shall explore existing solutions as well as emerging specifications and provide recommendations with focus on the end-user, i.e. data or earth scientist.

The **OGC-ESIP Coverage and Processing API Sprint at ESIP** winter meeting in January 2020 performs an analysis on coverages beyond the current WCS capabilities. This effort takes into account various elements that need to be developed for an API approach based on the abstract specifications for Coverages and Processing as well as OPeNDAP, GeoXarray/ZARR, R-spatial and other modern software development environments. The **Geospatial Coverages Data Cube Community Practice document** describes community practices for Geospatial Coverage Data Cubes as implemented by multiple communities and running as operational systems.

2.3.4. Scenario & Requirements

Testbed-16 shall address three different use cases. All use-cases shall be implemented and executed in Jupyter Notebooks that interact with OGC Web APIs as illustrated in figure X below.



Figure 9. DAPA architecture

The use cases describe different data retrieval requests from end-user's point of view. The end-user wants to execute a Jupyter Notebook, which executes a function call on Web API. The Web API shall then interact with the actual data platform, though Testbed-16 abstracts away the last step and concentrates on the Jupyter Notebook and Web API instead.

Use-Case 1: Data Retrieval

The user wants to access geospatial data for a specific area in a simple function call. The function call shall identify the data and allow to define the discrete sampling geometry. Valid geometries shall include point locations (x, y, and optional z), bounding-box, and polygon. All geometries shall be provided either in-line or by reference, as exemplarily shown below:

- In-line: <https://web.api/collections/temperature?location=50.11159,8.68248>

- By reference: <https://web.api/collections/temperature?location=https://other.api/collections/cities/items/Frankfurt>

The latter shall allow to request data for a specific sampling geometry by providing a call to an OGC API - Features endpoint. The various encoding options of sampling geometries provided by OGC API - Features instances shall be discussed.

Users shall be enabled to access original data. The word “original” is a bit tricky in this context, as data often undergoes some process on its way from original reading to final product. As an example, imagine a digital temperature sensor. The actual reading performed in the sensor is some form of electricity, but the value provided at the sensor interface is 21°Cel. Thus, some form of calibration curve has been applied to the original reading, which might not be available at all. In this case, the value 21°Cel can be considered as “original”. The same principles apply to satellite data. The original raw data readings are often not accessible. Instead, the data underwent some correction process before it is made available. Higher product levels may include orthorectification or re-gridding processes. In any case, data providers shall provide a description of the performed processing together with the actual data. In addition, data should be available as raw as possible.

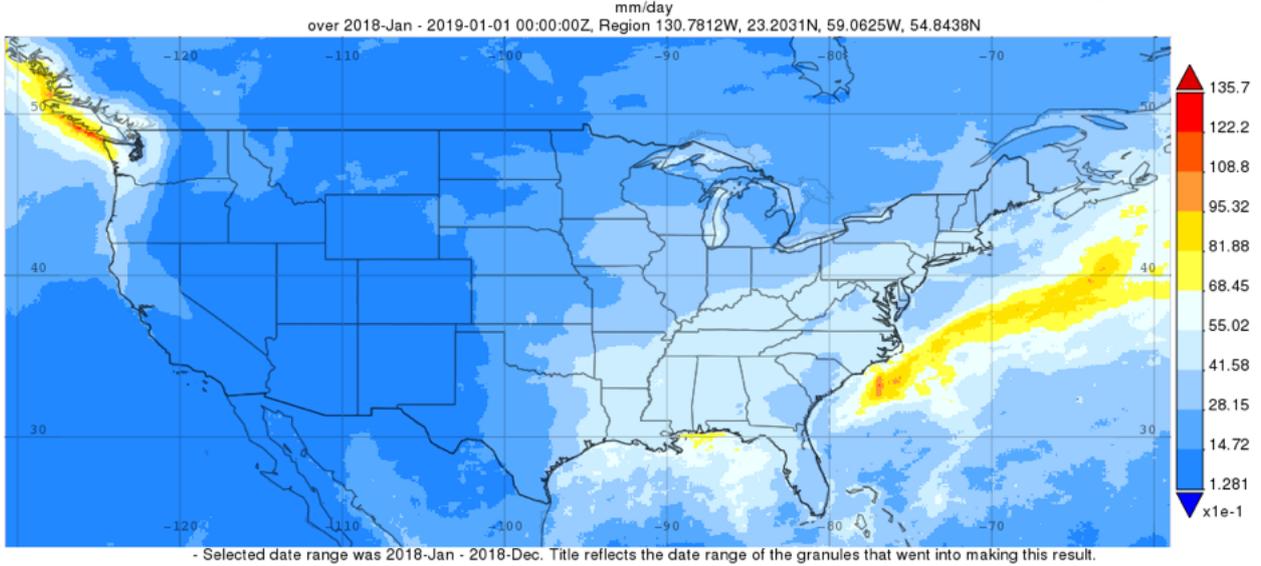
End-users want to retrieve all data that exists within the provided target geometry. In the case of a polygon geometry, the end-user shall receive all data that is located in that polygon. In the case of a point geometry, the end-user shall retrieve the value exactly at that point.

In addition, end-users shall have the option to define the (interpolation) method for value generation. If no option is selected, the Web API shall indicate how a given value was produced. Testbed-16 shall develop a set of frequently used production options, including for example “original value”, “interpolation method”, “re-gridding”, or any combination thereof.

This use case differentiates the following different data requests.

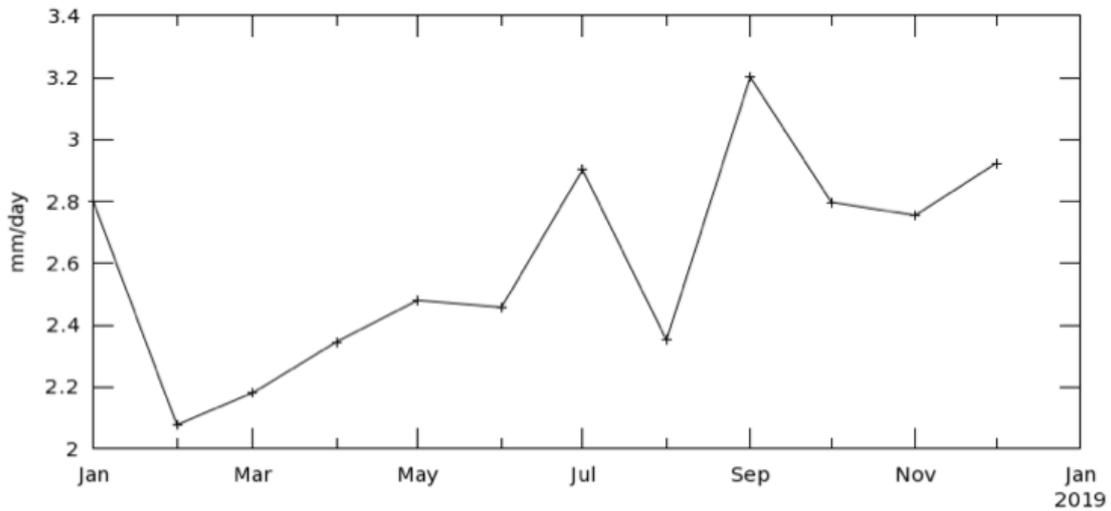
1. **Synopsis/Time-Averaged Map:** The end-user wants to retrieve data for a single point in time or as an average value over a time period. The figure below is an example of visualized time-averaged data for a number of sampling locations.

Time Averaged Map of Merged satellite-gauge precipitation estimate - Final Run (recommended for general use) monthly 0.1 deg. [GPM GPM_3IMERGM v06]

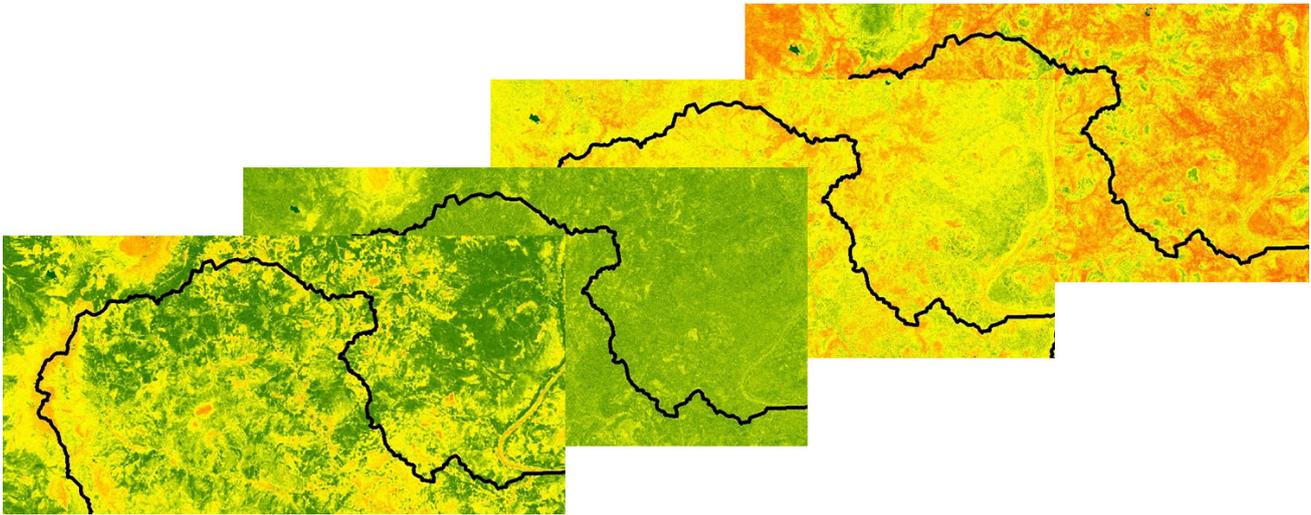


1. **Area-Averaged Time Series:** The end-user wants to retrieve a single value that averages all data in the target geometry for each time step. The figure below is an example of visualized area-averaged data for a number of time steps.

Time Series, Area-Averaged of Merged satellite-gauge precipitation estimate - Final Run (recommended for general use) monthly 0.1 deg. [GPM GPM_3IMERGM v06] mm/day over 2018-Jan - 2019-01-01 00:00:00Z, Region 130.7812W, 23.2031N, 59.0625W, 54.8438N



1. **Time Series:** The end-user wants to retrieve the full time series for each data point. The figure below is an example of visualized full time series data set that includes a number of time steps.



Testbed-16 shall explore these use-cases in combination with additional processing steps. For example, the end-user requests synoptic, map, or time series data, that is interpolated to a grid.

Testbed-16 does not address the data storage side. The stored data can be point cloud, gridded data set, datacube, or anything else.

Use-Case 2: Data Processing

Testbed-16 shall explore simple data processing functions. These include to calculate the minimum, maximum, and average values for any given data retrieval subset as accessible in the Data Retrieval use cases.

Use-Case 3: API Evaluation

The third use-case is orthogonal to the first two. It does not add any additional requirements on the API itself, but evaluates the API from an end-user point of view. This third use case will be implemented in the form of a full day workshop with several data scientists or earth scientists being invited to evaluate the API regarding

- Learning curve to use the API
- Richness of accessible functionality
- Amount of code needed to execute some common analyses

It is currently planned to organize the workshop in conjunction with another major event, such as e.g. ESIP summer meeting, July 2020 in Burlington, Vermont, or the OGC Technical Committee meeting in Montreal, Canada, June 2020. API developers are not required to attend the workshop physically. Remote participation will be provided. It is emphasized that workshop in June or July requires early design and implementation work to be finished.

The workshop shall allow the API developers and endpoint providers to further refine the EDR API and increase ease of use based on the feedback provided by the scientists. It is therefore expected that early versions of the API and corresponding implementations are available in time for the mid-term evaluation workshop.

2.3.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

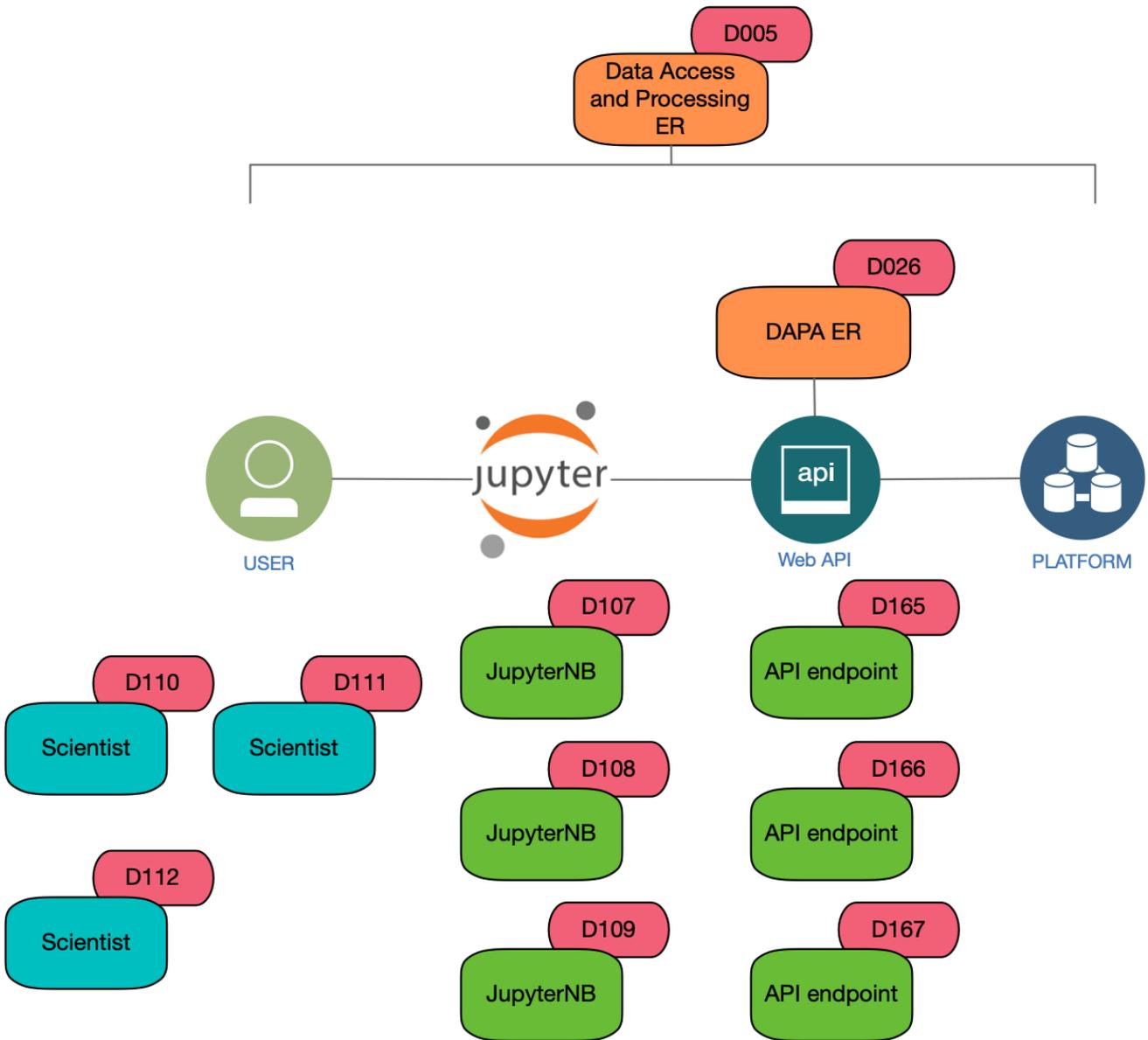


Figure 10. Deliverables of the Environmental Data Retrieval and Processing task

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D005 Data Access and Processing Engineering Report** - Engineering Report capturing all results and experiences from this task, including descriptions of all implementations and feedback from scientists.
- **D026 Data Access and Processing API Engineering Report** - Engineering Report describing the DAPA. The report shall be complemented by an OpenAPI example available on Swagger Hub.

Components

- **D107 Jupyter Notebook** - Jupyter Notebook that interacts with the API endpoints D165-167. The concrete data retrieval and processing scenarios shall be defined at the kick-off meeting. At minimum, the use cases **Data Retrieval** and **DataProcessing** shall be supported. The Jupyter Notebook can use any supported programming language.
- **D108 Jupyter Notebook** - similar to D107.
- **D109 Jupyter Notebook** - similar to D107.
- **D165 API Endpoint**- API endpoint implementation that provides the frontend to some data store. Any data store can be used, but operational data stores that are publicly available are preferred. The API endpoint shall implement the API defined in D026.
- **D166 API Endpoint**- similar to D165.
- **D167 API Endpoint**- similar to D165.
- **D110 Expert**- Data scientists or earth scientist helping to evaluate the API as described in the **Evaluation Use-case**. Experts are expected to provide their recommendations in written format that can be integrated into D005.
- **D111 Expert**- similar to D110.
- **D112 Expert**- similar to D110.

2.4. Earth Observation Application Packages with Jupyter Notebooks

Testbeds 13, 14, and 15 developed an architecture that allows deploying and executing arbitrary applications next to the physical location of the data to be processed. The architecture builds on specialized WPS interfaces that allow to submit so called Application Packages to Exploitation Platforms. An Exploitation Platform is a cloud based virtual environment that provides users with access to Earth observation data and tools for their manipulation. An Application Package contains all information, data, and software required to execute the application packaged inside a Docker container on a remote platform. The architecture is described and currently under evaluation in the current **OGC Earth Observation Applications Pilot**. Testbed-16 shall now complement this approach with applications based on **Project Jupyter**. The goal of Project Jupyter is to improve the workflows of researchers, educators, scientists, and other practitioners of scientific computing.



Testbed-16 shall explore how programming code developed by a scientist can be shared with other scientists in an efficient and secure way based on Jupyter Notebooks and related technology. The actual processing shall take place on exploitation platforms, which are data platforms that provide additional processing capacities. One of the key challenges in this context is retrieval of the data that shall be analyzed and processed. This data is stored in a variety of storage formats, such as individual files on clouds, datacubes, or databases, and needs to be made available to the Jupyter kernel. To facilitate data access, the data storages are ideally frontended with a standardized data access and processing API. The Testbed-16 task **Data Access and Processing API (DAPA) for Geospatial Data** has the goal to develop such an API and is therefore closely related. Whilst sharing Jupyter Notebooks is the focus of this task, both tasks need to merge towards the end of the Testbed to allow for shared TIEs, Technology Interoperability Experiments.

Testbeds 13-15 have explored various mechanisms to link individual applications into a chain of applications. Here, the output of one application serves as input for the next one. Though Testbed-13 favored BPMN as the preferred approach, Testbed-14 identified CWL as a simpler and more appropriate approach. Other initiatives favor process graphs expressed in JSON with JSON Schema for graph validation. In any case and independently of terminology, there are several aspects that have not been addressed in full detail yet and need further research. This includes in particular the error handling in federated cloud environments combined with appropriate roll-back and clean up mechanisms in case that some part of a chain fails.

2.4.1. Problem Statement and Research Questions

Jupyter Notebooks shall be shared with other scientists. However, the notebook documents are JSON documents that contain text, source code, rich media output, and metadata. Each segment of the document is stored in a cell. Sharing these JSON files is one option for exchange, but not ideal, given the envisioned automated deployment and execution on exploitation platforms.

Simple copying of notebooks between jupyter installations allows some interactive usage. On Exploitation Platforms, the goal is to go further and support orchestration and execution of notebooks in batch mode as part of workflows (see Testbed-14 results) and notebook discovery supported by catalogue technology (see Testbed-15 results). Also, it includes code inspection from authorized users and on the fly code changes to exploit the full potential of Jupyter notebooks.

Testbed-16 shall develop recommendations on how to handle Jupyter Notebooks similar to Application Packages; or expressed differently: how to include Jupyter Notebooks into Application Packages, so that they can be executed securely on exploitation platforms with support for code change and workflow orchestration and execution.

There are several elements to be considered in this context:

- It is possible to **create, list and load GitHub Gists** from notebook documents. Gists are a way to share code because it allows to share single files, parts of files, or full applications.
- With **jupyterhub**, it is possible to spawn, manage, and proxy multiple instances of a single-

user Jupyter notebook server. In other words, it's a platform for hosting notebooks on a server with multiple users. That allows providing notebooks to other scientists.

- **Binder** and **tmpnb** allow to get temporary environments to reproduce notebook execution, but are now superseded by jupyterhub.
- [<https://nbconvert.readthedocs.io/en/latest/Nbconvert>] or **papermill** allow to execute a JPNB from command line.
- Tools such as **nbviewer** allow to render notebooks as static web pages.
- **Jupyter_dashboards** allow to display notebooks as interactive dashboards, though this functionality is now superseded by Voilà.
- A big challenge with sharing notebooks is the security model. How to offer the interactivity of a notebook, making use of e.g. Jupyter widgets, without allowing arbitrary code execution by the end-user?
- Jupyter Notebooks shall be shared with non-technical persons. In particular, the browser based read-eval-print loop (REPL) notebook shall be presentable as a web application that hides all the programming code fields.
- **Voilà** supports interactive widgets, including roundtrips to the kernel. It does not permit arbitrary code execution by consumers of dashboards, is built on Jupyter standard protocols and file formats, and includes a template system to produce rich application layouts.
- How to chain a set of specific processes in a processing graph? Testbeds 13-15 experimented with **CWL** and **BPMN**, but other approaches exist such as **openEO process graphs**. Testbed-16 shall provide recommendations on the preferred solution and compare the advantages and disadvantages of the various solutions.

2.4.2. Aim

The aim of this task is to extend the Earth Observation Applications architecture developed in Testbeds 13-15 and further evaluated in the OGC Earth Observation Applications Pilot with support for shared and remotely executed Jupyter Notebooks. The notebooks shall make use of the Data Access and Processing API (DAPA) developed in the **Data Access and Processing API (DAPA) for Geospatial Data** task and tested in joint Technical Interoperability Experiments.

2.4.3. Previous Work

OGC Testbed activities in Testbed-13, Testbed-14, and the ongoing Testbed-15 have developed an architecture that allows the ad-hoc deployment and execution of applications close to the physical location of the source data. The goal is to minimize data transfer between data repositories and application processes. The following Engineering Reports describe the work accomplished in the Testbed-13/14:

- OGC Testbed-14: Application Package Engineering Report ([18-049r1](#))
- OGC Testbed-14: ADES & EMS Results and Best Practices Engineering Report ([18-050r1](#))

- OGC Testbed-14: Authorisation, Authentication, & Billing Engineering Report ([18-057](#))
- OGC Testbed-14: Next Generation Web APIs - WFS 3.0 Engineering Report ([18-045](#))
- OGC Testbed-13: EP Application Package Engineering Report ([17-023](#))
- OGC Testbed-13: Application Deployment and Execution Service Engineering Report ([17-024](#))
- OGC Testbed-13: Cloud Engineering Report ([17-035](#))

Testbed-13 reports are referenced to provide the background of this work and design decisions in context, but they are mostly superseded by Testbed-14 reports.

Testbed-15 has explored the discovery aspect of processes, applications, and data. The results will be made publicly available in OGC Testbed-15: Catalogue and Discovery Engineering Report ([OGC 19-020r1](#)).

A good summary of the current architecture is provided in the [OGC Earth Observation Applications Pilot](#) call for participation. The goal of the pilot is to evaluate the maturity of the Earth Observation Applications-to-the-Data specifications that has been developed over the last two years as part of various OGC Innovation Program (IP) initiatives in a real world environment. ‘Real world’ includes integration of the architecture in an environment requiring authenticated user identity, access controls and billing for resources consumed.

At the same time, significant progress towards more Web-oriented interfaces has been made in OGC with the emerging OGC APIs -Core, -Features, -Coverages, and -Processes. All of these APIs are using OpenAPI. These changes have not been fully explored in the current architecture, which provides additional ground for experimentation.

2.4.4. Scenario & Requirements

Testbed-16 envisions multiple scenarios that build on each other. All are fictitious and can be modified during the Testbed as long as the basic characteristics, i.e. data access, processing, and chaining, remain conserved.

The first scenario explores the handling of Jupyter Notebooks and the interaction with data and processing capacities through the Data Access and Processing API, DAPA. As said before, the scenario is fictitious and can be replaced by any other scenario as long as the individual steps, i.e. discovery, exploration, data requests and processing for both raster and vector data, and result representation remain conserved.

1. Notebook cell searches a catalog via OpenSearch with bounding box, time range and keywords .Notebook receives a list of data collections with WMS or OGC API endpoints for sample browse and displays them to the user
2. Notebook makes GetMap requests to WMS endpoints retrieves maps from OGC API endpoints to see sample pictures of the data collection over the bounding box and shows results on a map in Notebook

3. User selects one data collection for deeper exploration
4. Notebook queries catalog via OpenSearch for selected collection's data items* given bounding box and time range
5. Notebook receives an initial page of resource items with WCS/WFS or other data providing endpoints such as a Data Access and Processing API (DAPA) endpoint
6. Notebook selects one specific data item and requests data
7. Notebook receives data for desired space and time, and computes a time-averaged map and an area-averaged time series and displays them
8. Notebook continues to select, query, and process other data. This step needs to be repeated to illustrate how different types of data (vector and raster at different endpoints and serialized in different formats including data from DAPA) can be used with Jupyter notebooks
9. Notebook queries shapes from one endpoint to be used at another endpoints for area of interest selection
10. Notebook integrates various data sets and represents results.

The second scenario extends the first one by making the Jupyter Notebook or web application available to other users on an Earth Observation exploitation platform. The notebook shall be available in both batch mode (the full notebook is executed and only final results are provided) and interactive mode, where the user interacts with the notebook. The scenario includes the following components.

Component	Definition
EO Exploitation Platform (EP)	A cloud based virtual environment that provides users with access to EO data and tools for their manipulation
Thematic Exploitation Platform (TEP)	An EO Exploitation Platform focused on a specific theme
ENV-TEP	Fictitious TEP focused on environmental topics
User Application hosting service	A platform service that allows users to upload their application (that implements a data processing algorithm) on the platform. The upload works in the form of an ApplicationPackage that is then automatically integrated into the platform. It is assumed that this service is based on the results of OGC Testbed-14, Application Deployment and Execution Services and Execution Management Service respectively
Application execution service	A Platform service that allows users to execute applications available on the platform. It is assumed that this service is based on the results of OGC Testbed-14
Data Hosting service	A Platform service that allows users to store data on the platform
Data Access service	A Platform service that allows users to access the data available on the platform

Table 2. Scenario components

ENV-TEP provides the following platform level services:

- User Application hosting
- Application execution
- Data hosting
- Data access

Alice is a user of ENV-Platform and wants to code an algorithm that calculates an environmental quality index (EQIX) for major cities around the world. This quality index is based on a variety of data, e.g. average wind speed, annual rainfall, solar radiation data, population density, and air quality. The data is provided partly on the ENV-TEP, partly by other platforms and accessible via the Data Access and Processing API (DAPA).

Alice creates a Jupyter Notebook based application that is deployed on the platform using the User Application hosting service. It is executed by the Application execution service, with data being accessed through DAPA.

The algorithm is made available to others in two forms. First, as a Jupyter Notebook with source code available for step-wise execution and manipulation. Second, as a web application that only requires an area of interest as input and provides the quality index as output.

This scenario shall explore and develop recommendations for notebooks that can be used both in interactive step-wise execution mode and batch mode. The interactive step-wise execution mode displays intermediate results to the user, who can make decisions that influence the next steps (e.g. by selecting one out of many offered data sets). In batch mode, some cells that are used for visualization or user interaction need to be handled differently, as the user expects a final result exclusively.

The third scenario then continues with an application chaining element being added. In order to do so, Bob is introduced, who creates an application that takes EQIX results from several cities in an area and creates a liveable area index (LAIX). For that purpose, Bob runs the EQIX application for several cities, then combines results for selected cities that fall within a target area, adds additional data such as e.g. annual mean temperature for that area, and eventually produces a liveable area index. Applications to be chained in this scenario should be a combination of Jupyter notebook based applications and Application Packages that link Docker Containers with arbitrary applications. Bob should have read access to the EQIX application. He should not be allowed to delete the EQIX application.

2.4.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

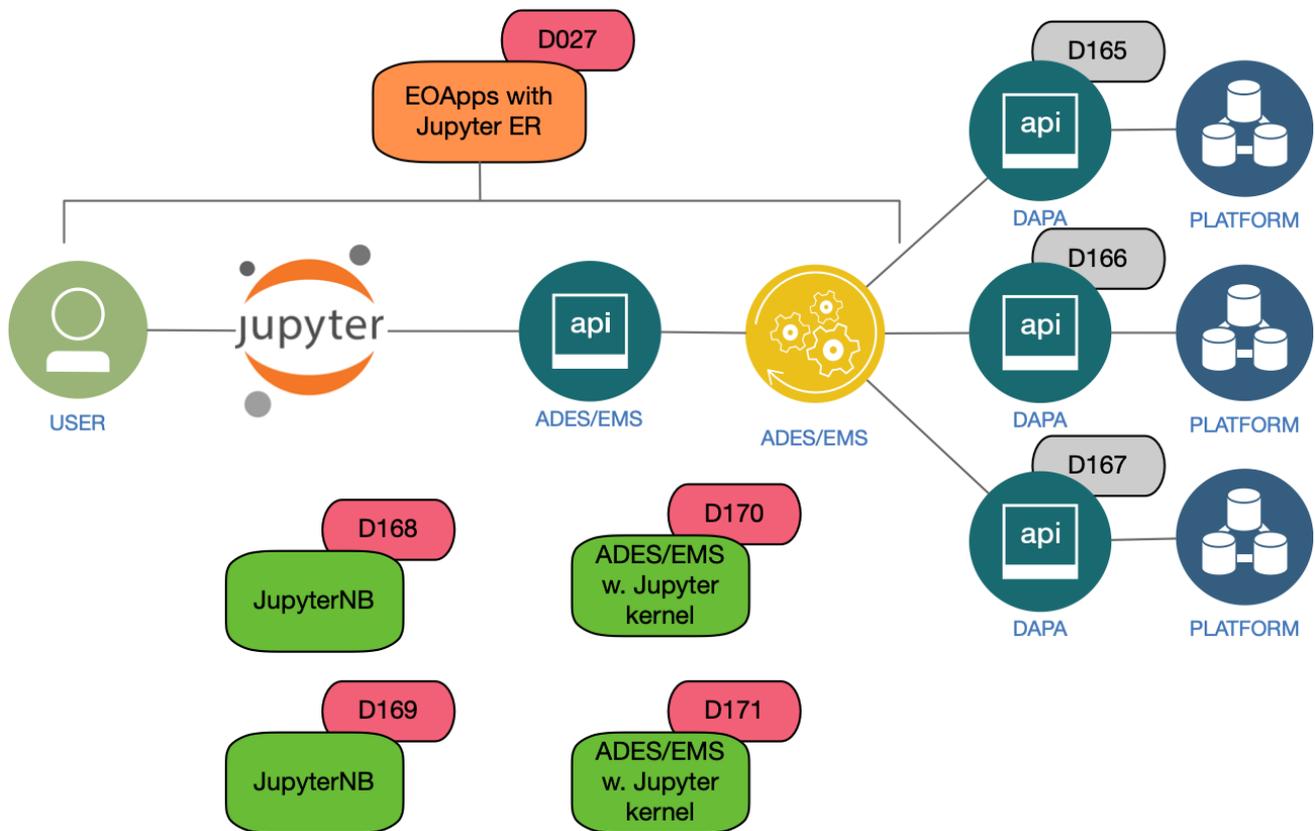


Figure 11. Earth Observation Application Processing with Jupyter Notebooks task architecture and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D027 EOApps with Jupyter Engineering Report** - Engineering Report capturing all results and experiences from this task. The report shall provide answers to all research questions and document implementations.

Components

- **D168 Jupyter NB** - Jupyter notebook (ideally *and* web application) that makes a complex application available for chaining at ADES/EMS.
- **D169 Jupyter NB** - similar to D168
- **D170 ADES/EMS w. Jupyter kernel** - ADES/EMS implementation with Jupyter support that allows registration of D168/169 and supports chaining. The platform shall interact with Data Access and Processing APIs as provided by the [Data Access and Processing API \(DAPA\) for Geospatial Data task](#).
- **D171 ADES/EMS w. Jupyter kernel** - similar to D170

2.5. GeoPackage

GeoPackage is an OGC standard that has grown substantially in popularity amongst the Geospatial community. The GeoPackage Encoding Standard was developed for the purpose of providing an open standard based platform for transferring geospatial information which is independent, portable, self-describing and compact.

Goal of this testbed is to advance the discoverability of the contents of a GeoPackage through utilising the concept of metadata profiles; and improved efficiency of using large scale vector datasets in GeoPackage.



2.5.1. Problem Statement and Research Questions

GeoPackage Encoding Standard has proven to be an effective "container" mechanism for bundling and sharing geospatial data for a variety of operational use cases. GeoPackage is an open standard based platform-independent format for transferring geospatial information.

The work in this testbed is focusing on improvements in GeoPackage through two use cases:

(1) Metadata Profiles

Discoverability of what is in a GeoPackage is important to enable a developer to quickly assess the type of data contained in a GeoPackage and to determine how it should be processed effectively. There is currently no agreement on the meaning and significance of metadata in GeoPackage or how that metadata should be used to serve any particular purpose. Manually opening a GeoPackage provides no way of recognising if the file has any particular type of metadata in it, without having to access the row entries in GeoPackage tables. The OGC document 'Proposed OGC GeoPackage Enhancements' proposes the concept of Metadata Profiles for GeoPackage and proposes this in two parts:

- Creating a new extension that defines a new extension "scope" (i.e., the `gpkg_extensions.scope` column) of Metadata.
- Creating an extension for each metadata profile that describes the meaning and significance of a particular type of metadata.

Metadata profiles will improve the functionality in GeoPackage and will be implemented for

Vector, Raster and Imagery datasets.

(2) Large Vector Datasets

Previous work has suggested that GeoPackage performance is poor when handling large vector datasets. This work is to examine and demonstrate the improved efficiency and effectiveness of storing large vector datasets in a GeoPackage. This may be through proposed extension(s) to GeoPackage which may include recommending open source tooling and software libraries for optimising SQLite, which is the underlying database that GeoPackage utilises. This approach may consider indexing, and Geohash could be considered to improve indexing functionality in GeoPackage.

“Geohash is a public domain geocode system invented in 2008 by Gustavo Niemeyer, which encodes a geographic location into a short string of letters and digits. It is a hierarchical spatial data structure which subdivides space into buckets of grid shape, which is one of the many applications of what is known as a Z-order curve, and generally space-filling curves.” (Wikipedia)

Diagram below shows GeoPackage interactions between client and server (GeoPackage Builder).

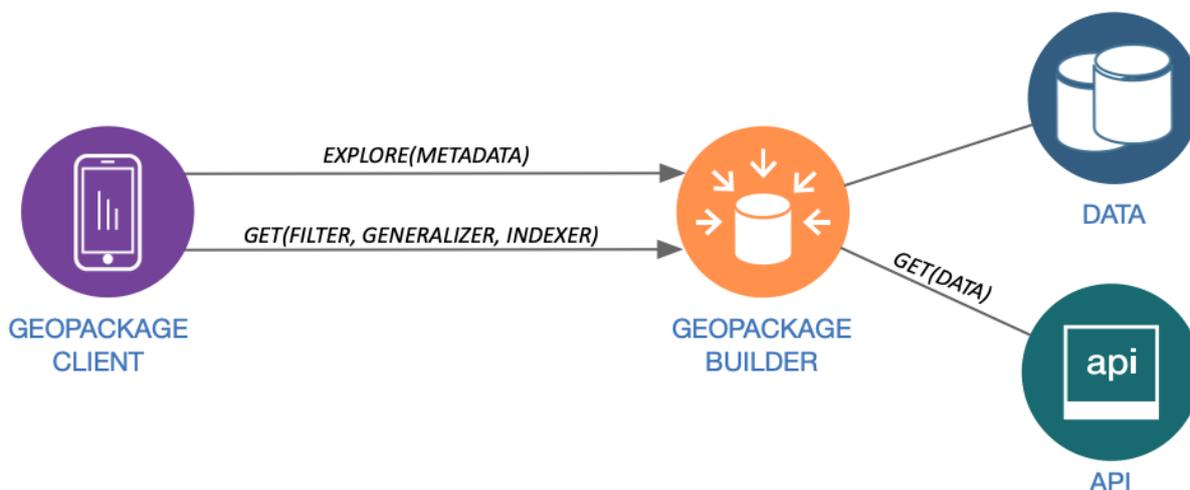


Figure 12. Overview of the GeoPackage task

2.5.2. Aim

Client and server implementations should demonstrate improved discoverability of content in GeoPackage through the implementation of Metadata Profiles and improved efficiency when distributing large vector datasets.

2.5.3. Previous Work

- [OGC 12-128r15](#), OGC® GeoPackage Encoding Standard Version 1.2.1
- [OGC 18-000](#), OGC® GeoPackage Related Tables Extension
- [OGC 19-047](#), OGC® Proposed OGC GeoPackage Enhancements

2.5.4. Scenario & Requirements

The following requirements shall be met:

- Develop a client and server implementation, which demonstrates the discoverability of the contents of a GeoPackage through Metadata Profiles. Metadata profiles should be developed for different types of data that are stored in a GeoPackage, such as Raster, Vector, Image and elevation data. Metadata Profiles should include the use, size, data type and provenance of the GeoPackage.
- Investigate potential causes of poor GeoPackage efficiency when used with Large Vector Datasets to establish whether the problem is with the format, or with the software tools being used. Develop a client and server implementation, which demonstrates the improved efficiency and effectiveness of storing large vector datasets in a GeoPackage. Open source tooling and libraries may be considered as an approach to achieve this. This work should document the approach and make recommendations for a GeoPackage extension or profile for large vector datasets. Portrayal of Vector Data should reference work in Open Portrayal Framework in Testbed- 15.

The implementation should focus on client and server implementations. All results and lessons learned shall be captured in an Engineering Report.

2.5.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

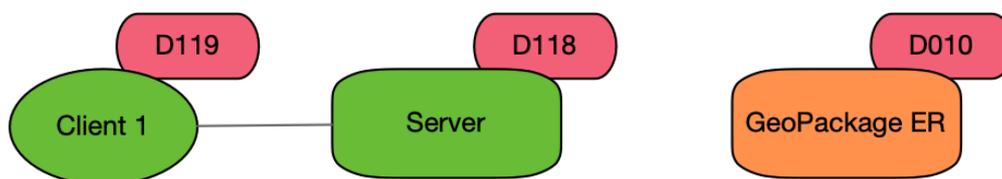


Figure 13. Deliverables of the GeoPackage task

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D010 GeoPackage Engineering Report** - Engineering Report capturing all results and experiences from this task. It should also capture and make recommendations for OGC standardisation and extensions.

Components

- **D119 GeoPackage Client** - Client implementation that supports the GeoPackage scenario documented above. The client can be implemented as a desktop application, mobile application, or browser based. Implementation is to be based upon Open source where

possible and a build script for working implementation in Docker or VM environment should be provided.

- **D118 GeoPackage Server**- Server implementation building GeoPackage with support for all requirements as defined above. Implementation is to be based upon Open source where possible and a build script for working implementation in Docker or VM environment should be provided.

2.6. Data Centric Security

What is Data Centric Security?

Data Centric Security is an approach that emphasizes the security of the data itself rather than the security of networks, servers, or applications, the approach achieves this by the means of:

- Encrypting the data at all times unless it is being manipulated by an authorized entity.
- Having a robust Attribute Based Access Control (ABAC) mechanism in place to provide attribution of all entities that wish to access the data
- A Policy Engine that determines whether or not access to a data element by an entity is permissible based upon policies stored within the Policy Engine.
- This results in:
 - The integrity of the data being verifiable.
 - Entities that produce/edit the data are verifiable and authorized.
 - Security measures required by/applicable to the data element(s) are attached to the data and remains with the data element(s) whilst it is either being stored or in transit.
 - Security measures related to confidentiality are greatly reduced within the systems within which the data is transferred or the services that the data passes through. Measures related to the delivery of the Integrity and Availability requirements of the systems being used are unchanged.



Why is Data Centric Security important?

The Data Centric Security approach delivers a number of benefits, such as:

- Data may reside on a service that the author of the data does not control (i.e. Cloud Storage).
- Data may transit through networks and be proxied by services that the author of the data does not control.
- Only an authorized entity may access the data element(s), thus providing the data owner with confidence that the data is appropriately protected.
- The use of Cryptographic techniques to bind the Metadata to the data provides a mechanism to verify that the data is authentic, has not been tampered with and is provided by the author they believe is providing the data.

2.6.1. Problem Statement and Research Questions

A fundamental requirement for Data Centric Security is that the data is always in an encrypted form until an authorized entity makes use of the data. An entity may be either a human or a system entity within the processing system.

As the data could pass through systems that don't belong to the data consumer or producer, the data must remain encrypted throughout the geospatial environment. The geospatial environment includes all infrastructure that touches the geospatial data (services, networks, storage, clients, etc.). When looking at utilizing OGC standards such as OGC API - Features in a data centric security scenario, standards need to include ways to classify the security requirements around data access. This classification can exist as additional metadata fields. The requirement stems from the need to limit different entities to a different subset(s) of the data. Additional requirements include the need for representation of the source of the information as well as an assurance that the information has not been tampered with.

The Data Centric Security work in Testbed-15 illustrated that it is possible to support Data Centric Security within the OGC API family of standards. Testbed-15 explored three scenarios, where a security proxy intercepted requests from a client. Data was then filtered, encrypted, and signed in three different ways. All three scenarios are described in detail in the Testbed-15 Data Centric Security Engineering Report.

1. In the first scenario, the security proxy forwards the request and modifies the response from a vanilla OGC API Features service to a STANAG 4774 and 4778 output format.
2. In the second scenario, the security proxy performs temporal decisions and spatial filtering additionally. To do so, the security proxy contains a geospatial policy of classified and unclassified areas.
3. In the third scenario, the security proxy forwards the request and response from an OGC API Features service that understands the STANAG 4774 and 4778 output format (and has access to data in these formats). In this scenario, the OGC API Features service returns a feature collection with STANAG 4774/4778 encoded feature objects.

In the Testbed-15 Engineering Report, the following topics were noted for future work:

- Testbed-15 used the STANAG 4774 and 4778 format, which used XML. Other encoding

formats exist and some applications, particularly in a commercial business, may not be as keen to support XML. The report proposed that the DCS solution should not be constrained to the use of XML and that an alternative, JSON-based container format should be investigated.

- A key management scenario that stores the keys in a key management service and requires the client to fetch the key via a key identifier stored in the metadata should also be implemented.

2.6.2. Aim

Further develop a Data Centric Security implementation in the OGC API family of standards, including a Data Centric Security JSON implementation.

Demonstrate a JSON Data Centric Security implementation using STANAG 4774 and 4778. NATO STANAG 4774 defines the Confidentiality Metadata Label Syntax for Data Centric Security. NATO STANAG 4778 is the Metadata Binding Mechanism for Joint Coalition Information Sharing.

An additional implementation should demonstrate a key management scenario, where keys are stored in a key store..

2.6.3. Previous Work

- [OGC 19-016](#), OGC Testbed-15 Data Centric Engineering Report
- STANAG 4774 – Metadata Confidentiality Label Syntax
- STANAG 4778 – Metadata Binding Mechanism

2.6.4. Scenario & Requirements

The overall goal is to continue the work of Testbed-15 and implement in this Testbed several of the recommendations identified in the Engineering Report.

Develop a JSON client and server implementation based on OGC API - Features and the Data Centric Security work in OGC Testbed-15. This work will reference STANAG 4774 – Metadata Confidentiality Label Syntax and STANAG 4778 – Metadata Binding Mechanism and implement this in a JSON language encoding. The Engineering Report should address questions such as: What do we learn from this implementation and what are the implications to Data Centric Security?

Develop a client and a server implementation in a key management scenario that stores encryption keys in a key management service. The client fetches the key via a key identifier, which is stored in the metadata. Any Key Management approach proposed should provide for the federation of multiple systems and the ability to transfer encryption key material, if so required. These implementation(s) should evaluate [IdAM](#) and [ABAC](#) Policy Engine Protocols in a client and server Data Centric Security implementation(s). Axiomatics Abbreviation Language for Authorization ([ALFA](#)) should be used as the Policy Engine Protocols in any implementation(s) as

well.

Address the content negotiation issue where container content formats need to be specified in addition to the container formats themselves. The STANAG 4778 output format is a container format that contains encrypted portions of sensitive data and associated metadata. This causes an extra challenge for the OGC API family of standards. Given the nested structure, the API standards need a way to specify both the container encoding and the format of the data in the container. Once standards such as OGC API - Features support the documentation of containers and data and get agreement by the implementing community, then interoperability is possible. However this may not be the only factor in interoperability. STANAG 4778 may not be an appropriate output format, especially when there may be a variety of different DCS formats in the future. One of the issues that different DCS formats may expose in the future is how to express a feature collection where items could be of different DCS formats. This would be caused by different authors contributing to the feature collection.

The OGC API – Records implementation allows geospatial data to be discovered. In a Data Centric Security implementation this is challenging. Work in this testbed is to explore how Geospatial data can be discovered in a Data Centric Security implementation.

In summary, OGC API family implementations shall be developed, which use a Data Centric Security approach based upon features and metadata:

- JSON Implementation of STANAG 4774 and 4778 should focus on client and server
- Key Management Scenario which stores encrypted keys in a store for the key management service
- Resolve the content negotiation issue with nested formats
- Explore how Geospatial data can be discovered in a Data Centric Security implementation.
- Consider the implication of Data Centric Security in feature and metadata implementation as an increased data burden on the network.
- Deliver implementation OGC API code
- Implementation is to be based upon Open source where possible
- Build script for working implementation on Docker or Virtual Machine (VM) environment
- Capture in an Engineering Report the implementation and make recommendations for OGC standardization

Desktop/client/server and cell-phone implementation scenarios

The various features shall be explored in two different scenarios. The first scenario assumes reliable and high speed internet access. The second scenario addresses cell-phone based data centric security.

Desktop/client/server scenario

The desktop/client/server does not put any specific constraints on the implementation setup. The DCS server can be provided as a single or distributed physical instance serving data and providing key management support. Connectivity between servers and clients can always be assumed.

Cell-phone scenario

A Sargent in the U.S. National Guard has been deployed on a disaster recovery mission. He carries with him a smart phone which contains sensitive data. When meeting with first responders, how does he share critical information with them without compromising sensitive information? How does internet connectivity affect that scenario?

Hypothesis: Use of the Data Centric Security techniques developed in Testbed-16 could address this problem. All sensitive data is encapsulated in a Data Centric Security package. Security policies are defined using GeoXACML. A Policy Enforcement Point (PEP) applet only allows access that data allowed under the currently active security policy. Authorized users can set the active security policy.

End State: The Sargent selects the security policy appropriate for the intended audience. He can now access data on his smart phone without worries about exposing sensitive information.

Task: Validate or invalidate this hypothesis. Demonstrate what is possible.

2.6.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

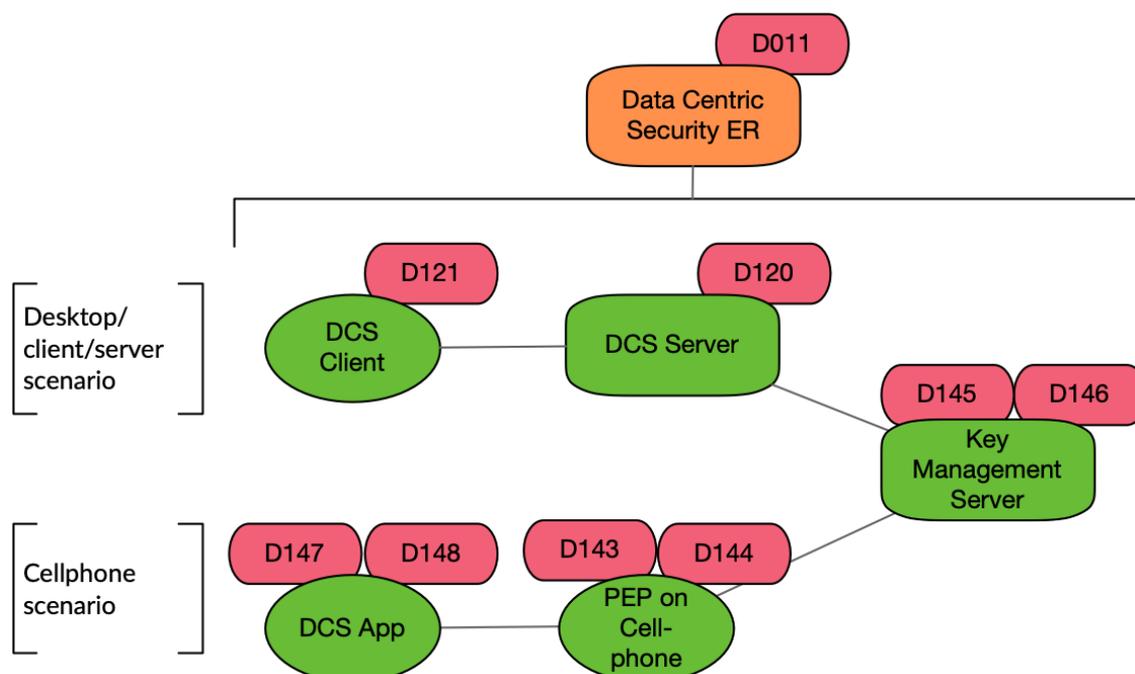


Figure 14. Data Centric Security task work items and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread

assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

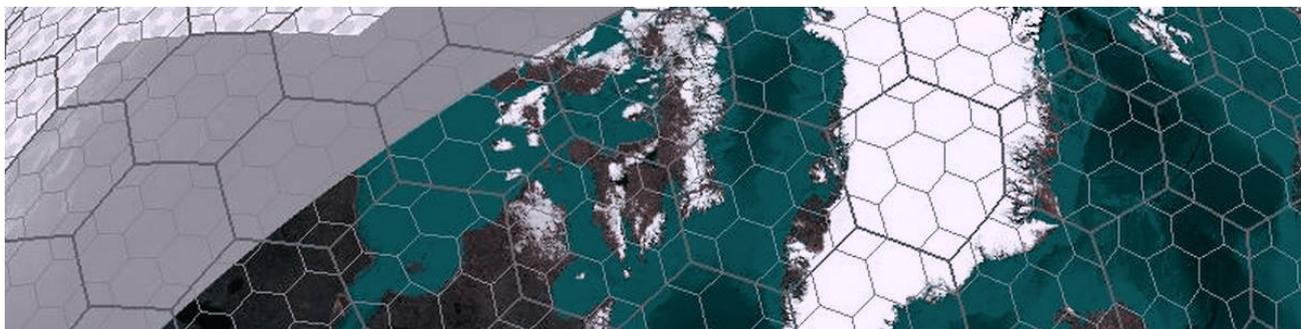
- **D011 Data Centric Security Engineering Report** - Engineering Report capturing all results and experiences from this task, including the JSON DCS specification. It should also make recommendations for OGC standardization and extensions.

Components

- **D121 Data Centric Security Client** - Client implementation that supports the Data Centric Security desktop/client/server scenario documented above. Ideally, implementation is based upon open source where possible and build script for working implementation in Docker or VM environment should be delivered.
- **D120 Data Centric Security Server**- Server implementation with support for data centric security desktop/client/server scenario. The server shall support key management. Implementation is to be based upon Open source where possible and build script for working implementation in Docker or VM environment should be delivered.
- **D143 PEP on Cellphone** - PEP implementation on cell-phone that supports cellphone scenario documented above.
- **D144 PEP on Cellphone** - Similar to D143
- **D145 Key Management Server** - Implementation of the key management server as required in both scenarios. Ideally, implementation is based upon open source where possible and build script for working implementation in Docker or VM environment should be delivered.
- **D146 Key Management Server** - Similar to D145
- **D147 DCS App** - Implementation of cellphone application to explore cellphone scenario as described above.
- **D148 DCS App** - Similar to D147

2.7. Discrete Global Grid System (DGGS)

A Discrete Global Grid System (DGGS) represents a spherical partitioning of the Earth's surface into a grid of cells ([Wikipedia](#)). The OGC maintains an Abstract Specification ([OGC 15-104r5](#)) that captures the foundational concepts for DGGS. This Testbed task aims to begin the process to move towards an OGC Implementation Standard for DGGS through the creation of an open-source based DGGS reference implementation. Testbed-16 represents the initial effort of what is considered a multi-initiatives process.



Discrete Global Grid Systems (DGGS) offer a new way for geospatial information to be stored, visualized, and analyzed. Based on a partitioning of the Earth's surface into a spherical grid, DGGS allows geospatial information to be represented in a way that more intuitively reflects relationships between data and the Earth's surface. With DGGS, providers and consumers of geospatial information can eliminate many of the uncertainties and distortions inherently present with traditional coordinate systems. To fully realize the benefits of DGGS, standard-compliant implementations are required to allow cell-id management across DGGS with varying structure and alignment.

2.7.1. Problem Statement and Research Questions

DGGS presents an opportunity for the geospatial community to implement a representation of Earth that is vastly different from traditional coordinate system approaches. DGGS has the potential to enable storage, analysis and visualization of geospatial information in a way that more accurately reflects the relationship between data and the Earth. While the OGC abstract specification captures fundamental DGGS concepts, there is a need to more concretely demonstrate DGGS to drive its adoption. Testbed-16 shall contribute to this advancement through development of a DGGS reference implementation.

Key questions for this work include the following:

- What DGGS structure would be best for developing a reference implementation? e.g. Uber's [Hexagonal Hierarchical Spatial Index](#), Open Equal Area Global Grid ([OpenEAGGR](#))
- What is a simple application that could be used to demonstrate the value of the reference implementation?
- What should be considered for future work oriented towards operational implementation of DGGS?

It is expected that results from this task will form the basis for future initiatives to fully enable DGGS through an OGC Implementation Standard.

2.7.2. Aim

This task aims at getting server-side DGGS implementation work started that supports a DGGS API. The API shall support two core functions, i.e. geographic location to cell-ID and cell-ID to geographic location, and optionally cell-ID to cell-ID conversion to support multiple DGGSs. The API shall be in-line with OGC OpenAPI family of standards.

2.7.3. Previous Work

- [OGC 15-104r5](#) - OGC DGGs Abstract Specification

2.7.4. Scenario & Requirements

The goal is to develop a reference DGGs implementation with a demonstrable use-case to highlight the potential and value of DGGs. The following requirements shall be met:

- The server-side implementation must be fully open source.
- The server-side implementation shall include a library that encapsulates the actual DGGs functionality. That library should be usable for existing tools and services and shall support the geographic location to cell-ID(s) and reverse conversion.
- All server-side development must be completed using open source tools, with all outputs made available in a public, freely accessible format.
- All aspects of the implementation (e.g. underlying code) must be made available through an open license. An example is the Government of Canada's [Open Government License](#). Other licenses will be considered by the sponsor if they contain similar characteristics.
- The client side demonstration application shall support features that highlight DGGs aspects and demonstrate the advantages a DGGs provides to the non-geospatial expert consumer type of user. The client shall be available as a browser-based solution. Open source is preferred for the client. The client shall visualize the DGGs at various zoom levels and interact with DGGs-enabled data services, i.e. OGC API endpoints that understand cell-IDs as spatial filters. Though appreciated, a globe-like visualization is not required. Simple visualizations (i.e. 2 dimensional) that demonstrate the capabilities of DGGs will be welcome.

The following figure illustrates possible implementation scenarios.

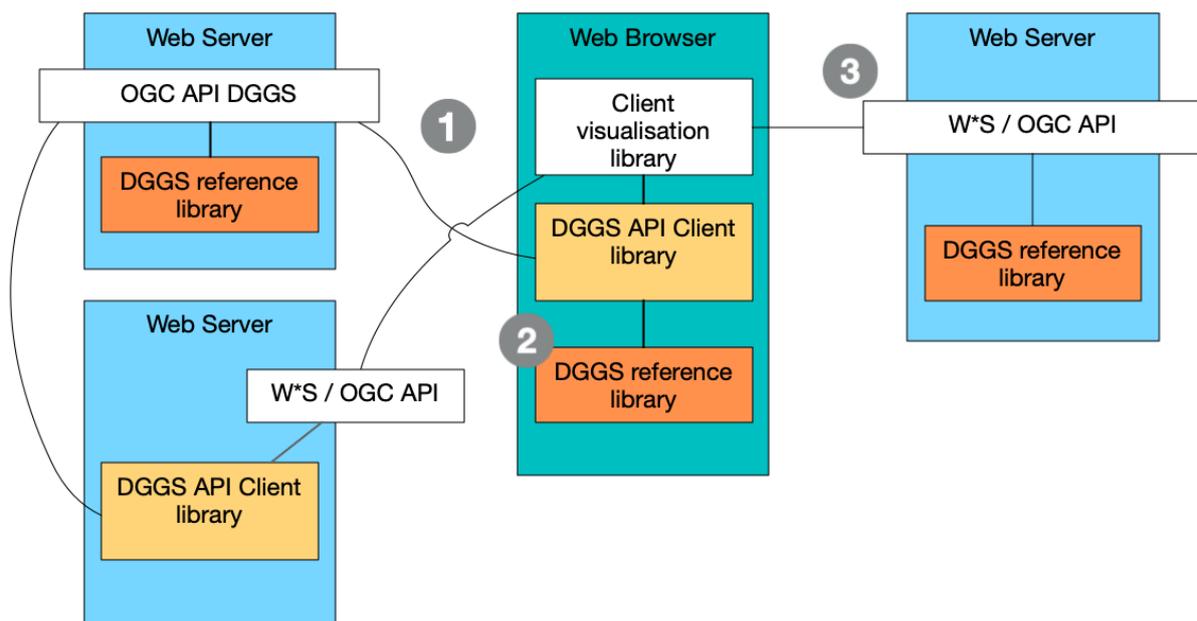


Figure 15. Possible DGGs implementation scenarios

In the first case (1), the *web browser* in the middle connects to the *DGGS server* through its *DGGS API Client library* to learn about DGGS cells provided by the *DGGS reference library*. The client illustrates cells by overlay with appropriate background material. The user selects cell-IDs and runs a query against a data server (OGC API or OGC W*S). The data service, not yet equipped with the DGGS reference library, queries the DGGS server to convert cell-IDs to geographic locations before returning data back to the client.

In the second case (2), the client can provide all visualization itself because it has direct access to the DGGS library. In this case, the user again selects cell-IDs and follows either the same path as described in (1), or queries a data server that is already equipped with the DGGS reference library. In this third case (3), the data server "understands" cell-IDs as spatial filters and can provide data in return without the detour to the DGGS server.

Other implementation scenarios are certainly possible.

2.7.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

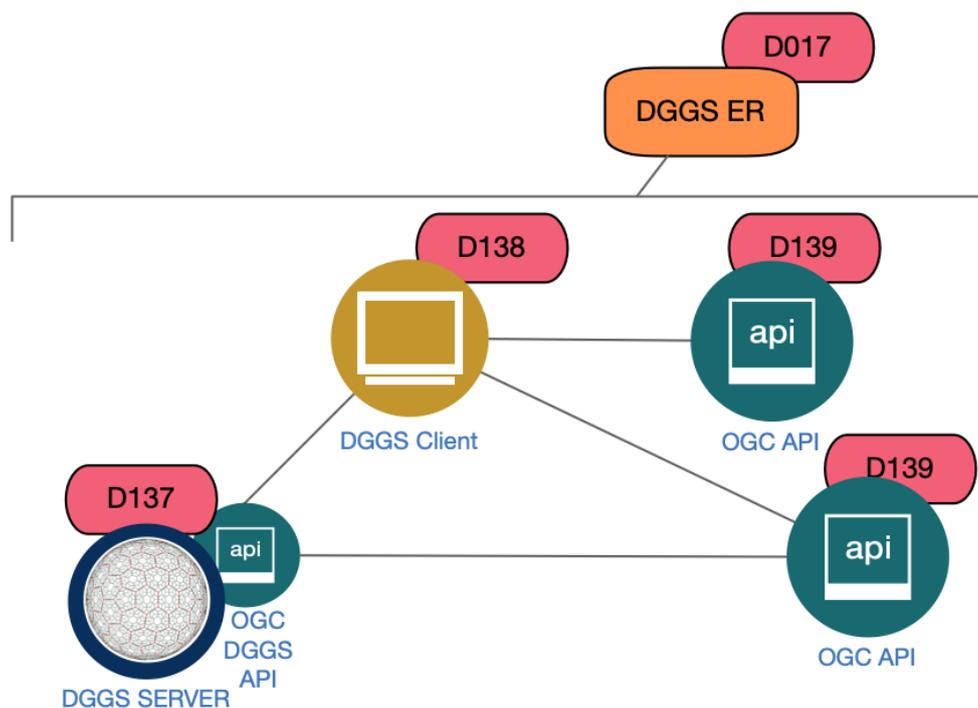


Figure 16. DGGS task work items and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D017 DGGS and DGGS API Engineering Report** - Engineering Report capturing all results and experiences from this task. The report shall include the documentation of the DGGS API. It shall describe the analysis performed, design of the proof-of-concept, and validation of the

results. It shall also make recommendations for OGC standardization and extensions. The Engineering Report shall contain a plain language executive summary to clearly outline the motivations, goals, and critical outcomes of this task, taking into account the mandates of the OGC and the sponsor.

Components

- **D137 DGGs Server Implementation** - Open source server implementation with support for DGGs API. The implementation shall be delivered as Docker container or VM with all source code. As a sub-component, the server-side implementation shall deliver a library that encapsulates the actual DGGs functionality. That library shall be available for existing tools and services and shall support the geographic location to cell-ID(s) and reverse conversion.
- **D138 DGGs Demo Client** - Client application with DGGs API support and capable of demonstrating DGGs capabilities. The client shall visualize the DGGs at various zoom levels and interact with D139 DGGs-enabled data services, i.e. OGC API endpoints that understand cell-IDs as spatial filters.
- **D139 DGGs Enabled Data Services** - DGGs-enabled data services, i.e. either OGC API endpoints or OGC W*S services that understand cell-IDs as spatial filters. The services can use the DGGs reference implementation library from D137 to convert cell-IDs to geographic filters; or implement the same DGGs to ensure consistent cell-IDs across D137, D138, and D139. Alternatively, the instances can make use of the D137 service instance for cell-ID(s) to geographic location conversion.

2.8. Analysis Ready Data (ARD)

Analysis Ready Data (ARD) is a powerful concept which lacks a common definition. The ARD concept has been explored most extensively by CEOS, the Committee on Earth Observation Satellites. The [CEOS Analysis Ready Data Strategy](#), released October 2019, provides the best introduction to the concept.

Other efforts have tried to extend and define the concept further, e.g. as “time-series stacks of overhead imagery that are prepared for a user to analyze without having to pre-process the imagery themselves” ([Holmes a, b](#)). The European Space Agency’s (ESA) Thematic Exploitation Platforms ([TEP](#)) can be considered being an ARD environment (although the term is not used explicitly), which is not limited to imagery. The National Aeronautics and Space Administration (NASA) and US Geological Survey ([USGS](#)) also provide ARD implementations, both aligned with the CEOS strategy.



Testbed-16 shall explore to which extent the ARD definition can be stretched meaningfully and if the proposed definition “*automated identification and pre-processing of all data needed to execute an analytic task regardless of source, format, or location*” can pass. Therefore, Testbed-16 is tasked to develop an architecture for Federated ARD. This architecture shall have two parts;

- a. define the as-is leveraging existing best practices from industry, academia, and government;
- b. provide a future vision of what ARD can be.

It is the intent that this architecture will serve as an organizing principle for ARD discussions and become a framework for maturation of ARD capabilities.

2.8.1. Problem Statement and Research Questions

ARD is a loosely defined concept, though appealing in its potential to facilitate interaction with data from an end-user’s perspective. This task shall help understanding ARD concepts and implementation architectures and serve as a guide for further ARD implementations. Goal is to define an architecture that works across cloud platforms and services operated by a variety of data providers.

2.8.2. Aim

Testbed-16 shall develop an architecture that allows maturing ARD concepts and discussions.

2.8.3. Previous Work

Part of the Analysis Ready Data concept and architectures have been part of previous OGC Innovation Program activities, but no existing Engineering Report is targeting the concept explicitly.

It is therefore recommended to be familiar with ARD as it is currently discussed. The following links are starting points for further research.

- [CEOS ARD Strategy](#)
- [OGC Testbed-15: Federated Clouds Security Engineering Report](#)
- [OGC Testbed-15: Federated Clouds Analytics Engineering Report](#)
- [Big Geospatial Data – an OGC White Paper](#)
- [The USGS Landsat Analysis Ready Data \(ARD\) Program and U.S. Landsat Analysis Ready Data](#)
- [ESIP Winter meeting 2020 session on Datacubes for Analysis Ready Data](#)
- [Analysis Ready Data Defined by Chris Holmes](#)

2.8.4. Scenario & Requirements

The ARD architecture should include the following properties:

- Federated: There is no single cloud. Every organization has its own infrastructure, analytics, and data. Federation provides a loosely coupled solution for performing automatic analysis (ARD) across these multiple clouds.
- Heterogeneity: Variety is one of the three “V”s of big data. A Federated ARD architecture must support a broad variety of data, analytics, and analytic missions. While remote sensing is arguably the most difficult case, the architecture should not be limited to that field.
- Move Analytics to the Data: Volume is another one of the three “V”s of big data. This is particularly true for Remote Sensing systems. The content is far too massive to ship around the world. Better to send the analytics to the data, process in place, and then return the results. A Federated ARD architecture should identify the protocols and packaging standards for enabling deployable analytics.
- Protect the Crown Jewels: “Information wants to be free!” unless it’s the access codes to your bank account. Not all resources can be made public. That does not mean they cannot be accessible. Federated identify, proxy services, executable security policies, and a number of other techniques can allow a remote party to use a resource without actually having access to it. The Federated Security thread will be particularly relevant to this principle.
- Event Driven: Most current ARD concepts are based on the request-response access pattern. This limits the degree to which the ARD processing can be hidden from the end user. In contrast, consider an event-driven architecture. Under an event-driven approach an analytic executes, not when it is told to, but when sufficient pre-conditions have been satisfied for it to do its job. For example, a change detection would automatically execute when new cloud free images have been collected over the area of interest. The user is then notified of the results, without any need to search the libraries or execute the analytic.

2.8.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

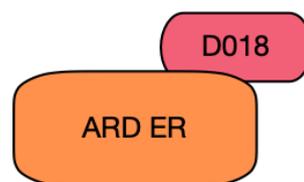


Figure 17. ARD task work items and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D018 Analysis Ready Data Engineering Report** - Engineering Report capturing all results and experiences from this task. The Engineering Report shall document the as-is and to-be Federated ARD architectures.

2.9. Federated Security

NIST and IEEE have developed an architecture for maintaining security in a federated cloud environment, see the [NIST Cloud Computing Collaboration Site](#) for general information and the [NIST Federated Cloud Reference Architecture](#) for details.



The architecture is based on the concept of *Virtual Administrative Domains*, i.e., federations that are hosted across one or more *Federation Managers* (FM). The OGC has performed experiments in Testbed 14 and 15, which proves out the concepts. However, if a FM were to be compromised, all of the federations being supported by that FM might be compromised. Hence, a thorough analysis of the security and trust requirements for FM implementations must be done.

2.9.1. Problem Statement and Research Questions

The OGC has performed experiments in Testbeds 14 and 15, which proves out the concepts. Testbed-16 shall perform a trust analysis of the architecture. In particular, Testbed-16 shall examine the protocols, encodings, and key management techniques required to implement this architecture for a high-security environment.

The potential topics and concepts to investigate may include, but are not limited to:

- Data-centric security approaches,
- Blockchain.
- Attribute-Based Encryption, and
- Zero Trust Architectures.

Though participation is open to all security experts, designated participants ideally include distributed security experts from federally funded research and development centers (FFRDCs) and other academic or scientific institutions. These institutions may include, but are not limited to Internet2, InCommon, the Open Science Grid, eduGAIN, and the European Open Science Cloud.

2.9.2. Aim

Examine and evaluate the NIST Federated Cloud Reference Architecture for a high-security environment in terms of protocols, encodings, and key management techniques.

2.9.3. Previous Work

- [OGC Testbed-15: Federated Clouds Security Engineering Report](#)
- [OGC Testbed-15: Federated Clouds Analytics Engineering Report](#)
- [OGC Testbed-14 Federated Clouds Engineering Report](#)
- [OGC Testbed-14: Security Engineering Report](#)

2.9.4. Scenario & Requirements

Bidders are requested to propose focus for analysis and possible scenarios. Both will be refined at the kick-off meeting.

2.9.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

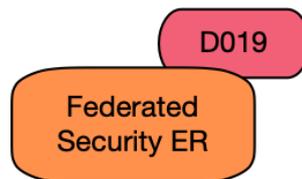


Figure 18. Federated Security task work items and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D019 Federated Security Engineering Report** - Engineering Report capturing all results and experiences from this task.

2.10. Full Motion Video to Moving Features

Unmanned Aerial Vehicles (UAVs) are increasingly used in military and civil applications. A video camera is often the primary sensor on a UAV, providing a torrent of imagery of the overflowed ground. From this imagery, it is possible to detect the presence of moving objects on the ground, such as vehicles and people, which can be of strategic importance.

Though, Full Motion Video (FMV) brings to mind watching endless hours of video. Yet, modern

Motion Imagery platforms are capable of extracting a considerable body of information from the raw collection and streaming that information in-band with the video stream. However, this information comes in a form which is not readily exploitable except by specialized systems.



This proposal aims to better integrate information derived from Motion Imagery into the data standards used by geospatial analysts.

2.10.1. Problem Statement and Research Questions

This task shall experiment with the generation of Moving Features from FMV. This work shall evaluate existing OGC Standards such as **OGC Moving Features** together with its new **JSON encoding**, SensorML, O&M, and Sensor Observation Service as well as NATO STANAG 4609 for suitability in this context.

STANAG 4609

STANAG 4609 describes an exchange format for motion imagery. It is the official format for motion imagery (video data, image sequences, FMV - full motion videos) exchange within the NATO nations. Motion imagery is defined by MISB to be video of at least 1 Hz image frequency together with metadata. STANAG 4609 describes the encoding of the video and the metadata (geographical data) for different usages. This includes the supported video codecs, bit rates, frame rates, container formats, metadata content, metadata encoding and hardware to distribute the motion imagery.

The standards which make television and cable networks possible are established and maintained by the Society of Motion Imagery and Television Engineers (SMPTE).

SMPTE ST 336 defines a byte-level data encoding protocol for which can be multiplexed with a video stream. Synchronization between the key-value pairs and the associated video frames is maintained using the same mechanism as is used to synchronize the audio and video streams. SMPTE also defines a small set of Keys in SMPTE ST 335.

The Motion Imagery Standards Board (MISB) has built on those standards to address additional requirements identified by the Defense and Intelligence communities. Those standards are codified in the Motion Imagery Standards Profile (MISP) and STANAG 4609. The MISB standards most relevant to this effort are MISB 0601 and MISB 0903. MISB 0601 defines the basic set of keys

for use by UAS systems. MISB 0903 defines additional keys for Video Moving Target Indicators (VMTI). Moving Target Indicators are reports on detections of objects in a FMV frame which appear to be moving, along with any additional descriptive information that the detector can provide.

OGC SensorML

Sensor Model Language (SensorML) provides a robust and semantically-tied means of defining processes and processing components associated with the measurement and post-measurement transformation of observations. This includes sensors and actuators as well as computational processes applied pre- and post-measurement. The main objective is to enable interoperability, first at the syntactic level and later at the semantic level (by using ontologies and semantic mediation), so that sensors and processes can be better understood by machines, utilized automatically in complex workflows, and easily shared between intelligent sensor web nodes.

OGC Observations and Measurements (O&M)

The **Observations and Measurements (O&M)** Standard defines a conceptual schema for sensor observations, and sampling features produces when making observations. These provide models for the exchange of information describing observation acts and their results, both within and between different scientific and technical communities. Observations commonly involve sampling of an ultimate feature of interest. This International Standard defines a common set of sampling feature types classified primarily by topological dimension, as well as samples for ex-situ observations. The schema includes relationships between sampling features (sub-sampling, derived samples).

OGC Sensor Observation Service (SOS)

The **Sensor Observation Service (SOS)** standard defines a Web service interface which allows querying observations, sensor metadata, as well as representations of observed features. Further, this standard defines means to register new sensors and to remove existing ones. Also, it defines operations to insert new sensor observations.

OGC Moving Features

This **OGC Standard** specifies standard encoding representations of movement of geographic features. The primary use case is information exchange.

A feature is an abstraction of a real-world phenomenon. It is a geographic feature if it is associated with a location relative to the Earth. ISO 19141:2008 represents moving features by defining a geometry for a feature that moves as a rigid body. This allows moving and stationary features to be analyzed and exploited using the same algorithms and software.

2.10.2. Aim

Generate Moving Features, serialized using O&M and SensorML and accessible via Sensor Observation Service, from Full Motion Video.

2.10.3. Previous Work

- SWE Implementation Maturity Engineering Report
- OWS-8 Tracking: Moving Target Indicator Process, Workflows and Implementation Results Engineering Report
- OWS-8 Analysis of OGC Standards for Supporting Mobile Object Processing Implementation Engineering Report
- OWS-8 Information Model for Moving Target Indicators and Moving Object Bookmarks

2.10.4. Scenario & Requirements

Visual Moving Target Indicator (VMTI) video analysis systems automatically detect and quantify the movement of vehicles and people from a continuously moving video taken from an air vehicle. In this scenario, participants will explore VMTI in the context of geospatial analysis and standards. More precisely, participants in this effort will perform four tasks:

1. Generation of Moving Features from VMTI detections. The first step is to determine if the temporal axis of moving features is sufficient for our needs and address any deficiencies. Then an approach to generate a moving feature from a VMTI key-value stream will be developed.
2. Apply the lessons learned from Moving Features generation to explore the generation of O&M Observations from a VMTI key-value stream.
3. Explore the generation of SensorML 2.0 records describing the collection platform and relevant telemetry information from the key-value stream (0601 and 0903 content).
4. Demonstrate the generation of moving features, Observations, and SensorML documents from sample Full Motion Video streams. Verify that the generated artifacts are faithful representations of the FMV stream contents.

2.10.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.



Figure 19. Full Motion Video to Moving Features task work items and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

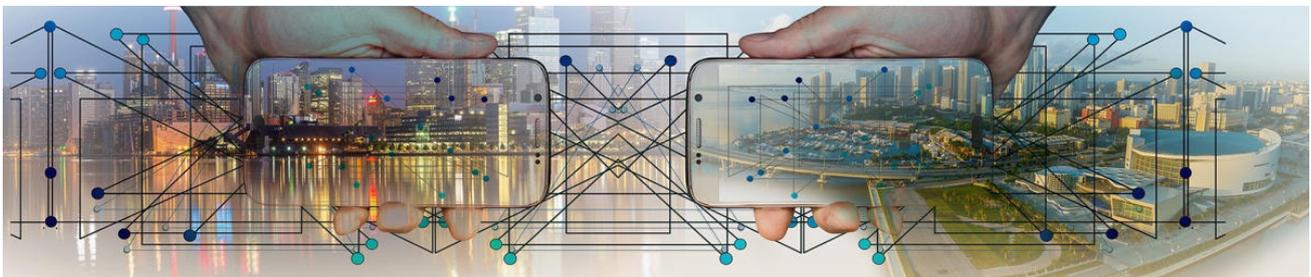
- **D021 Full Motion Video to Moving Features Engineering Report** - Engineering Report capturing all results and experiences from this task. It shall describe the analysis performed, design of the proof-of-concept, and validation of the results. It should also make recommendations for OGC standardization and extensions.

Components

- **D141 Full Motion Video to Moving Features Demonstrator** - Proof of concept software which generates Moving Features, Observations, and SensorML documents from FMV input.

2.11. OpenAPI

OGC APIs replace the “Capability Documents” provided by OGC Web Services with API descriptions using OpenAPI. However, an OpenAPI document, which fully describes an API and its supported resources, can become very complex. A process and supporting tools for managing the definition of Web APIs will greatly facilitate the consistent implementation of these APIs.



Tools such as **ShapeChange** exist that already addresses many elements of this process. This task will investigate and prototype enhancements to existing tools to support API management through the automated generation of OpenAPI documents based on a governing UML model and the selected conformance classes.

2.11.1. Problem Statement and Research Questions

ISO 19109:2015 Geographic information – Rules for application schema Clause 2.3 UML application schema, defines a Unified Modeling Language (UML) meta-schema for feature-based data. That General Feature Model (GFM) serves as the key underpinning for a family of ISO standards employed by national and international communities to define the structure and content of geospatial data.

ISO 19136:2007 Geographic information – Geography Markup Language (GML) Annex E UML-to-GML application schema encoding rules, defines a set of encoding rules that may be used to transform a GFM-conformant UML concept model to a corresponding XML Schema (XSD) based on the structure and requirements of GML. The resulting XSD file(s) may be used to define the structure of, and validate the content of, XML instance documents used in the exchange of geospatial data among open-source, commercial, consortia, and government systems. This methodology for deriving technology-specific encodings of GFM-conformant UML concept models based on formalized sets of encoding rules has been successfully applied in other contexts than GML (e.g., technologies based on Resource Description Framework (RDF)

encodings).

This task shall enable existing tool [ShapeChange](#) to support API management through the automated generation of OpenAPI documents based on a governing UML model and the selected conformance classes. The task is open to all tools, but favors open source implementations that become available at the end of the Testbed.

2.11.2. Aim

This task will investigate and prototype enhancements to existing tools to support API management through the automated generation of OpenAPI documents based on a governing UML model and the selected conformance classes.

2.11.3. Previous Work

- [UML-to-GML Application Schema \(UGAS\) Pilot \(UGAS-2019\) – Summary Engineering Report](#)
- [OGC Testbed-14: Application Schema-based Ontology Development Engineering Report](#)
- [OGC Testbed-14: Application Schemas and JSON Technologies Engineering Report](#)
- [OGC Testbed-14: Next Generation Web APIs - WFS 3.0 Engineering Report](#)

2.11.4. Scenario & Requirements

The OpenAPI definitions used in OGC APIs include the following sections:

- info: service metadata
- servers: base URL of servers supporting the API
- paths: resources and their HTTP methods
- tags: tags to group path entries
- security: security requirements and controls
- parameters: path and query parameters
- schemas: schemas used in the API

The schemas section describes schemas for the request and response messages used by this API. These schemas may be complex, such as in the case of Features and Feature Collections. While useful, even essential to the client, these schemas can be very difficult to create and maintain. Therefore, automated generation of OpenAPI schemas is the highest priority for this task.

The paths and parameters sections define the URL paths available to access resources on an API. These values are largely determined by the OGC API standard which is being implemented. Automated generation of these sections should be supported.

The info, servers, and security sections describe the current ownership and configuration of the

API. These values can change during operations without impacting the underlying resources or their access. These may not be viable candidates for automated generation.

2.11.5. Work Items & Deliverables

The following figure illustrates the work items and deliverables of this task.

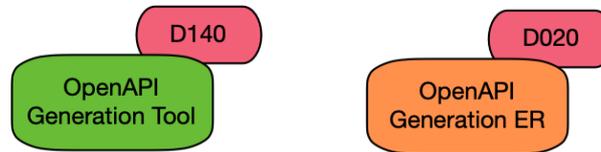


Figure 20. OpenAPI task work items and deliverables

The following list identifies all deliverables that are part of this task. Detailed requirements are stated above. All participants are required to participate in all technical discussions. Thread assignment and funding status are defined in section [Deliverables Summary & Funding Status](#).

Engineering Reports

- **D020 OpenAPI Generation Engineering Report** - Engineering Report capturing all results and experiences from this task. The report shall describe the changes made to existing tools such as ShapeChange, the design, and the rationale behind that design.

Components

- **D140 OpenAPI Generation Tool** - ShapeChange implementation that supports the generation of OpenAPI documents as described above. Implementation is to be based upon open source where possible and build script for working implementation in Docker or VM environment should be delivered.

Chapter 3. Deliverables Summary & Funding Status

The following table summarizes the full set of Initiative deliverables. Technical details can be found in section [Technical Architecture](#). Items with "Funding Status pending" identify work items that are most likely to be funded, but the sponsor contract has not been signed yet. Items marked with an asterisk identify work items where cost sharing funds can only be requested by entities from ESA member states, Canada, and Slovenia.

ID	Document / Component	Funding Status
D001	ER	funded
D005	ER	funded
D010	ER	funded
D011	ER	funded
D015	ER	funded*
D016	ER	funded*
D017	ER	pending
D018	ER	pending
D019	ER	pending
D020	ER	pending
D021	ER	pending
D026	ER	funded*
D027	ER	funded*
D100	Component	funded
D101	Component	funded
D103	Component	funded
D105	Component	funded
D106	Component	funded
D107	Component	funded
D108	Component	funded
D109	Component	funded
D110	Component	funded
D111	Component	funded
D112	Component	funded
D118	Component	funded
D119	Component	funded
D120	Component	funded
D121	Component	funded
D130	Component	funded
D131	Component	funded
D132	Component	funded
D133	Component	funded
D134	Component	funded

ID	Document / Component	Funding Status
D135	Component	funded*
D136	Component	funded*
D137	Component	pending
D138	Component	pending
D139	Component	pending
D140	Component	pending
D141	Component	pending
D143	Component	pending
D144	Component	pending
D145	Component	pending
D146	Component	pending
D147	Component	pending
D148	Component	pending
D165	Component	funded*
D166	Component	funded*
D167	Component	funded*
D168	Component	funded*
D169	Component	funded*
D170	Component	funded*
D171	Component	funded*

Table 3. CFP Deliverables and Funding Status

Chapter 4. Miscellaneous

Call for Participation

The CFP consists of stakeholder role descriptions, proposal submission instructions and evaluation criteria, a master schedule and other project management artifacts, Sponsor requirements, and an initiative architecture. The responses should include the proposing organization's technical solution, its cost-sharing requests for funding, and its proposed in-kind contributions to the initiative.

Once the original CFP has been published, ongoing authoritative updates and answers to questions can be tracked by monitoring the [CFP Corrigenda Table](#) and the [CFP Clarifications Table](#).

Participant Selection and Agreements:

Bidders may submit questions via timely submission of email(s) to the OGC Technology Desk. Question submitters will remain anonymous, and answers will be regularly compiled and published in the CFP Clarifications page.

OGC may also choose to conduct a Bidder's question-and-answer webinar to review the clarifications and invite follow-on questions.

Following the closing date for submission of proposals, OGC will evaluate received proposals, review recommendations with the Sponsor, and negotiate Participation Agreement (PA) contracts, including statements of work (SOWs), with selected Bidders. Participant selection will be complete once PA contracts have been signed with all Participants.

Kickoff: The Kickoff is a physical meeting where Participants, guided by the Initiative Architect, will refine the Initiative architecture and settle upon specific use cases and interface models to be used as a baseline for prototype component interoperability. Participants will be required to attend the Kickoff, including breakout sessions, and will be expected to use these breakouts to collaborate with other Participants and confirm intended Component Interface Designs.

Regular Teleconference and Interim Meetings After the Kickoff, participants will meet on a frequent basis remotely via web meetings and teleconferences.

Development of Engineering Reports, Change Requests, and Other Document Deliverables: Development of Engineering Reports (ERs), Change Requests (CRs) and other document deliverables will commence during or immediately after Kickoff.

Under the Participation Agreement (PA) contracts to be formed with selected Bidders, ALL Participants will be responsible for contributing content to the ERs. But the ER Editor role will assume the duty of being the primary ER author.

Final Summary Reports, Demonstration Event and Other Stakeholder Meetings: Participant Final Summary Reports will constitute the close of funded activity. Further development work

might take place to prepare and refine assets to be shown at demonstration events and other meetings.

Assurance of Service Availability: Participants selected to implement service components must maintain availability for a period of no less than six months after the Participant Final Summary Reports milestone. OGC might be willing to entertain exceptions to this requirement on a case-by-case basis.

Appendix A: Testbed Organization and Execution

A.1. Initiative Policies and Procedures

This initiative will be conducted within the [policy framework](#) of OGC's Bylaws and Intellectual Property Rights Policy ("IPR Policy"), as agreed to in the OGC Membership Agreement, and in accordance with the OGC Innovation Program Policies and Procedures and the OGC Principles of Conduct, the latter governing all related personal and public interactions.

Several key requirements are summarized below for ready reference:

- Each selected Participant will agree to notify OGC staff if it is aware of any claims under any issued patents (or patent applications) which would likely impact an implementation of the specification or other work product which is the subject of the initiative. Participant need not be the inventor of such patent (or patent application) in order to provide notice, nor will Participant be held responsible for expressing a belief which turns out to be inaccurate. Specific requirements are described under the "Necessary Claims" clause of the IPR Policy.
- Each selected Participant will agree to refrain from making any public representations that draft Engineering Report (ER) content has been endorsed by OGC before the ER has been approved in an OGC Technical Committee (TC) vote.
- Each selected Participant will agree to provide more detailed requirements for its assigned deliverables, and to coordinate with other initiative Participants, at the Kickoff event.

A.2. Initiative Roles

The roles generally played in any OGC Innovation Program initiative include Sponsors, Bidders, Participants, Observers, and the Innovation Program Team ("IP Team"). Explanations of the roles are provided in [Tips for New Bidders](#).

The IP Team for this Initiative will include an Initiative Director and an Initiative Architect. Unless otherwise stated, the Initiative Director will serve as the primary point of contact (POC) for the OGC.

The Initiative Architect will work with Participants and Sponsors to ensure that Initiative activities and deliverables are properly assigned and performed. They are responsible for scope and schedule control, and will provide timely escalation to the Initiative Director regarding any high-impact issues or risks that might arise during execution.

A.3. Types of Deliverables

All activities in this testbed will result in a Deliverable. These Deliverables can take the form of Documents or Implementations.

A.3.1. Documents

Engineering Reports (ER) and **Change Requests** (CR) will be prepared in accordance with OGC published templates. Engineering Reports will be delivered by posting on the (members-only) OGC Pending directory when complete and the document has achieved a satisfactory level of consensus among interested participants, contributors and editors. Engineering Reports are the formal mechanism used to deliver results of the Innovation Program to Sponsors and to the **OGC Standards Program** for consideration by way of **Standards Working Groups** and **Domain Working Groups**.

A.3.2. Implementations

Services, Clients, Datasets and Tools will be provided by methods suitable to its type and stated requirements. For example, services and components (e.g. a WPS instance) are delivered by deployment of the service or component for use in the Initiative via an accessible URL. A Client software application or component may be used during the Initiative to exercise services and components to test and demonstrate interoperability; however, it is most often not delivered as a license for follow-on usage. Implementations of services, clients and data instances will be developed and deployed in all threads for integration and interoperability testing in support of the agreed-up thread scenario(s) and technical architecture. The services, clients, and tools may be invoked for cross-thread scenarios in demonstration events.

A.4. Proposal Evaluation

Proposals are expected to be short and precisely addressing the work items a bidder is interested in. A proposal template will be made available. Details on the proposal submission process are provided under the **General Proposal Submission Guidelines**.

Proposals will be evaluated based on criteria in two areas: technical and management/cost.

A.4.1. Technical Evaluation Criteria

- Concise description of each proposed solution and how it contributes to achievement of the particular deliverable requirements described the Technical Architecture,
- Overall quality and suitability of each proposed solution, and
- Where applicable, each proposed solution is OGC-compliant.

A.4.2. Management/Cost Evaluation Criteria

- Willingness to share information and work in a collaborative environment,
- Contribution toward Sponsor goals of enhancing availability of standards-based offerings in the marketplace,
- Feasibility of each proposed solution using proposed resources, and

- Proposed in-kind contribution in relation to proposed cost-share funding request.

Note that all Participants are required to provide at least some level of in-kind contribution (i.e., activities or deliverables offered that do not request cost-share compensation). As a rough guideline, a proposal should include at least one dollar of in-kind contribution for every dollar of cost-sharing compensation requested. All else being equal, higher levels of in-kind contributions will be considered more favorably during evaluation. Participation may also be purely in-kind (no cost-share request at all).

Once the proposals have been evaluated and cost-share funding decisions have been made, the IP Team will begin to notify Bidders of their selection to enter negotiations for potentially becoming initiative Participants. The IP Team will develop for each selected bidder a **Participation Agreement** (PA), which will include a **Statement of Work** (SOW) describing the assigned deliverables.

A.5. Reporting

Initiative participant business/contract representatives are required (per the terms in the Participation Agreement contract) to report the progress and status of the participant's work. Detailed requirements for this reporting will be provided during contract negotiation. Initiative accounting requirements (e.g., invoicing) will also be described in the contract.

The IP Team will provide monthly progress reports to Sponsors. Ad hoc notifications may also occasionally be provided for urgent matters. To support this reporting, each testbed participant must submit (1) a Monthly Technical Report and (2) a Monthly Business Report by the first working day on or after the 3rd of each month. Templates and instructions for both of these report types will be provided.

The purpose of the Monthly Business Report is to provide initiative management with a quick indicator of project health from the perspective of each participant. The IP Team will review action item status on a weekly basis with the Initiative participants assigned to complete those actions. Initiative participants must be available for these contacts to be made.

Appendix B: Proposal Submission

B.1. General Proposal Submission Guidelines

This section presents general guidelines for submitting a CFP proposal. Detailed instructions for submitting a response proposal using the *Bid Submission Form* web page can be found in [Proposal Submission Procedures](#) further below.

Proposals must be submitted before the appropriate response due date indicated in the [Master Schedule](#).

Bidders responding to this CFP must be OGC members and must be familiar with the OGC mission, organization, and process. Proposals from non-members will be considered *provided that* a completed application for OGC membership (or a letter of intent to become a member) is submitted prior to (or with) the proposal.

Information submitted in response to this CFP will be accessible to OGC and Sponsor staff members. This information will remain in the control of these stakeholders and will not be used for other purposes without prior written consent of the Bidder. Once a Bidder has agreed to become a Participant, they will be required to release proposal content (excluding financial information) to all initiative stakeholders. Commercial confidential information should not be submitted in any proposal and should generally not be disclosed during initiative execution.

Bidders will be selected to receive cost sharing funds on the basis of adherence to the requirements stipulated in this CFP and the overall quality of their proposal. The general initiative objective is for the work to inform future OGC standards development with findings and recommendations surrounding potential new specifications. Bidders are asked to formulate a path for producing executable interoperable prototype implementations that meet the stated CFP requirements, and for documenting the findings and recommendations arising from those implementations. Bidders not selected for cost sharing funds may still be able to participate in addressing the stated CFP requirements on a purely in-kind basis.

However, to help maintain a manageable process, Bidders are advised to avoid attempts to use the initiative as a platform for introducing new requirements not included in [Technical Architecture](#). Any additional in-kind scope should be offered outside the formal bidding process, where an independent determination can be made as to whether it should be included in initiative scope or not. Items deemed out-of-initiative-scope might be more appropriate for inclusion in another OGC-IP initiative.

Each selected Participant (including purely in-kind Participants) will be required to enter into a Participation Agreement contract ("PA") with the OGC. The reason this requirement applies to pure in-kind Participants is that other Participants will be relying upon their delivery to show component interoperability across organizational boundaries.

Each PA will include a Statement of Work ("SOW") identifying Participant roles and responsibilities. The purpose of the PAs is to encourage and enable Participants to work together

to realize initiative goals for the benefit of the broader OGC community.

B.2. Questions and Clarifications

Once the original CFP has been published, ongoing authoritative updates and answers to questions can be tracked by monitoring the [CFP Corrigenda Table](#) and the [CFP Clarifications Table](#)

Bidders may submit questions via timely submission of email(s) to the OGC Technology Desk (techdesk@opengeospatial.org). Question submitters will remain anonymous, and answers will be regularly compiled and published in the clarifications.

A Bidders Q&A Webinar is scheduled as defined in the [Master Schedule](#). The webinar is open to the public, but anyone wishing to attend must register using the provided link.

The webinar will review any clarifications received to date and will provide an opportunity for follow-on questions.

B.3. Proposal Submission Procedures

The process for a Bidder to complete a proposal is essentially embodied in the online [Bid Submission Form](#). Once this site is fully prepared to receive submissions (soon after the CFP release), it will include a series of web forms, one for each deliverable of interest. A summary is provided here for the reader's convenience.

An account will need to be created before a bidder can use the form. Once an account has been created, the user will be taken to a home page indicating the "Status of Your Proposal." If any defects in the form are discovered, this page includes a link for notifying OGC. The user can return to this page at any time by clicking the *OGC logo* in the upper left corner.

IMPORTANT

Because the *Bid Submission Form* is still relatively new, it might contain some areas that are still brittle or in need of repair. Please notify OGC of any discovered defects. Periodic version updates will be provided as needed.

Please consider making **backup local copies** of all inputs in case any of them need to be re-entered.

Please also note that once this form goes "live", any submitted bids will be treated as earnest submissions, even those submitted well before the response deadline. Be certain that you intend to submit your proposal before you click the *Submit* button on the *Review* page.

B.3.1. High-Level Overview

Clicking on the *Propose* link will navigate to the Bid Submission Form. The first time through, the user should complete all the organizational information fields and click *Update and Continue*.

This will navigate to an "Add Deliverable" page that will resemble the following:

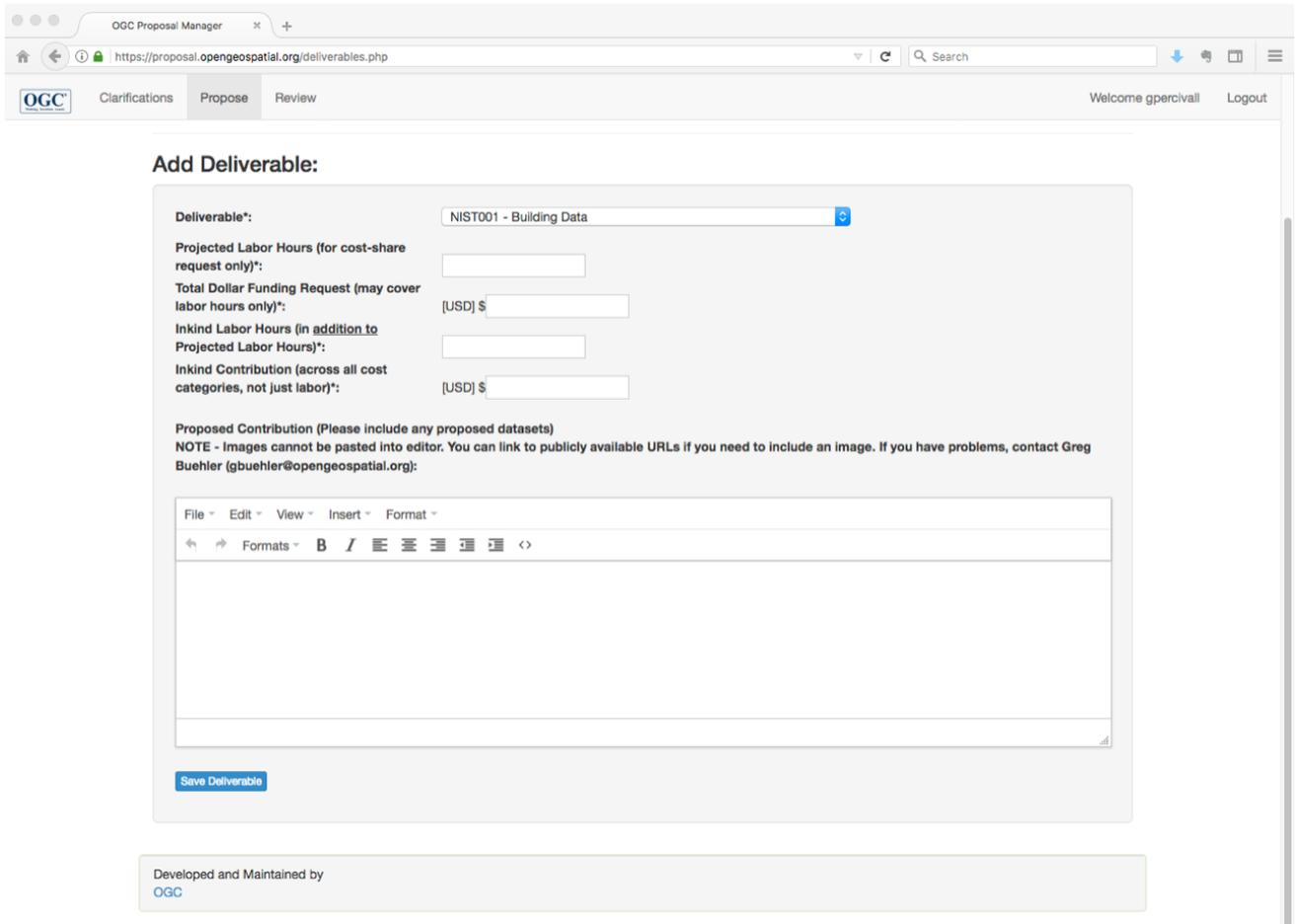


Figure 21. Sample "Add Deliverables" Page

The user should complete this form for **each** proposed deliverable.

On the far right, the *Review* link navigates to a page summarizing all the deliverables the Bidder is proposing. Note that this Review tab won't appear until the user has actually submitted at least one deliverable under the Propose tab first.

TIP

Consider regularly creating printed output copies of this *Review* page at various checkpoints during proposal creation in case an input error is made later.

Once a submission has been received, the system will send an email to the bidder and to OGC staff.

TIP

In general, up until the time that the user clicks this *Submit* button, the proposal may be edited as many times as the user wishes. However, this initial version of the form contains no "undo" capability, so please use caution in over-writing existing information.

The user is afforded an opportunity under *Done Adding Deliverables* at the bottom of this page to attach an optional *Attached Document* of explanation. If this attachment is provided, it is limited to one per proposal and there is a **5Mb file size limitation**.

This document could conceivably contain any specialized information that wasn't suitable for entry into a *Proposed Contribution* field under an individual deliverable. It should be noted, however, that this additional documentation will only be read on a **best-effort basis**. There is no guarantee it will be used during evaluation to make selection decisions; rather, it could optionally be examined if the evaluation team feels that it might help in understanding any specialized (and particularly promising) contributions.

B.3.2. Detailed Instructions

The *Propose* link takes the user to the first page of the proposal entry form. This form contains fields to be completed once per proposal such as names and contact information.

It also contains an optional *Organizational Background* field where Bidders (particularly those with no experience participating in an OGC initiative) may provide a description of their organization. It also contains a click-through check box where each Bidder will be required (before entering any data for individual deliverables) to acknowledge its understanding and acceptance of the requirements described in this appendix.

Clicking the *Update and Continue* button then navigates to the form for submitting deliverable-by-deliverable bids. On this page, existing deliverable bids can be modified or deleted by clicking the appropriate icon next to the deliverable name. Any attempt to delete a proposed deliverable will require scrolling down to click a *Confirm Deletion* button.

To add a new deliverable, the user would scroll down to the *Add Deliverable* section and click the *Deliverable* drop-down list to select the particular deliverable of interest.

The user would then enter the required information for each of the following fields (*for this deliverable only*). Required fields are indicated by an asterisk ("*"):

- Estimated *Projected Labor Hours** for this deliverable
- *Funding Request**: total U.S. dollar cost-share amount being requested for this deliverable (to cover burdened labor only)
- Estimated *In-kind Labor Hours** to be contributed for this deliverable
- Estimated *In-Kind Contribution*: total U.S. dollar estimate of the in-kind amount to be contributed for this deliverable (including all cost categories)

Cost-sharing funds may **only** be used for the purpose of offsetting burdened labor costs of development, engineering, and demonstration of initiative outcomes related to the Participant's assigned deliverables. By contrast, the costs used to formulate the Bidder's in-kind contribution may be much broader, including supporting labor, travel, software licenses, data, IT infrastructure, etc.

Theoretically there is no limit on the size of the *Proposed Contribution* for each deliverable (beyond the raw capacity of the underlying hardware and software). But bidders are encouraged to incorporate content by reference where possible (rather than inline copying and pasting) to avoid overloading the amount of material to be read in each proposal. There is also a textbox on a separate page of the submission form for inclusion of *Organizational Background* information, so there is no need to provide this information for each deliverable.

IMPORTANT

A breakdown (by cost category) of the "Inkind Contribution" may be included in the *Proposed Contribution* text box for each deliverable. However, please note that the content of this text box will be accessible to all Stakeholders and should contain no confidential information such as labor rates. If a Bidder wishes to provide confidential information that will only be accessible to OGC staff, it can be included in the *Attached Documentation* of the submission. Only one *Attached Document* may be submitted for the entire proposal (there is no way to submit a separate attachment for each individual deliverable).

A Bidder proposing to deliver a *Service Component Implementation* should use the *Proposed Contribution (Please include any proposed datasets)* field to identify what suitable **datasets** would be contributed (or what data should be acquired from another identified source) to support the proposed service.

This field *Proposed Contribution (Please include any proposed datasets)* could also be used to provide a succinct description of what the Bidder intends to deliver for this work item to meet the requirements expressed in the **Technical Architecture**. This language could potentially include a brief elaboration on how the proposed deliverable will contribute to advancing the **OGC standards baseline**, or how implementations enabled by the specification embodied in this deliverable could add specific value to end-user experiences.

The statements of work (SOWs) in the *Participation Agreement (PA)* contracts will ultimately require performance during initiative *execution* against the deliverables as described in the Technical Architecture plus any subsequent *Corrigenda*.

A single bid may propose deliverables arising from any number of threads or tasks. To ensure that the full set of sponsored deliverables are made, OGC might negotiate with individual Bidders to drop and/or add certain deliverables from their proposals.

B.4. Tips for New Bidders

Bidders who are new to OGC initiatives are encouraged to review the following tips:

- In general, the term "activity" is used as a verb describing work to be performed in an initiative, and the term "deliverable" is used as a noun describing artifacts to be developed and delivered for inspection and use.
- The roles generally played in any OGC Innovation Program initiative are defined in the OGC Innovation Program Policies and Procedures, from which the following definitions are derived and extended:
 - Sponsors are OGC member organizations that contribute financial resources to steer Initiative requirements toward rapid development and delivery of proven candidate specifications to the OGC Standards Program. These requirements take the form of the deliverables described herein. Sponsors representatives help serve as "customers" during Initiative execution, helping ensure that requirements are being addressed and broader OGC interests are being served.
 - Bidders are organizations who submit proposals in response to this CFP. A Bidder selected to participate will become a Participant through the execution of a Participation Agreement contract with OGC. Most Bidders are expected to propose a combination of cost-sharing request and in-kind contribution (though solely in-kind contributions are also welcomed).
 - Participants are selected OGC member organizations that generate empirical information through the definition of interfaces, implementation of prototype components, and documentation of all related findings and recommendations in Engineering Reports, Change Requests and other artifacts. They might be receiving cost-share funding, but they can also make purely in-kind contributions. Participants assign business and technical representatives to represent their interests throughout Initiative execution.
 - Observers are individuals from OGC member organizations that have agreed to OGC intellectual property requirements in exchange for the privilege to access Initiative communications and intermediate work products. They may contribute recommendations and comments, but the IP Team has the authority to table any of these contributions if there's a risk of interfering with any primary Initiative activities.
 - Supporters are OGC member organizations who make in-kind contributions aside from the technical deliverables. For example, a member could donate the use of their facility for the Kickoff event.
 - The Innovation Program Team (IP Team) is the management team that will oversee and coordinate the Initiative. This team is comprised of OGC staff, representatives from member organizations, and OGC consultants. The IP Team communicates with Participants and other stakeholders during Initiative execution, provides Initiative scope and schedule control, and assists stakeholders in understanding OGC policies and procedures.

- The term Stakeholders is a generic label that encompasses all Initiative actors, including representatives of Sponsors, Participants, and Observers, as well as the IP Team. Initiative-wide email broadcasts will often be addressed to "Stakeholders".
 - Suppliers are organizations (not necessarily OGC members) who have offered to supply specialized resources such as capital or cloud credits. OGCs role is to assist in identifying an initial alignment of interests and performing introductions of potential consumers to these suppliers. Subsequent discussions would then take place directly between the parties.
- Proposals from non-members will be considered provided that a completed application for OGC membership (or a letter of intent to become a member if selected) is submitted prior to or along with the proposal.
 - Any individual wishing to gain access to the Initiative's intermediate work products in the restricted area of the Portal (or attend private working meetings / telecons) must be a member-approved user of the OGC Portal system. Intermediate work products that are intended to be shared publicly will be made available as draft ER content in a public GitHub repository.
 - Individuals from any OGC member organization that does not become an Initiative Sponsor or Participant may still (as a benefit of membership) quietly observe all Initiative activities by registering as an Observer.
 - Prior initiative participation is not a direct bid evaluation criterion. However, prior participation could accelerate and deepen a Bidder's understanding of the information presented in the CFP.
 - All else being equal, preference will be given to proposals that include a larger proportion of in-kind contribution.
 - All else being equal, preference will be given to proposed components that are certified OGC-compliant.
 - All else being equal, a proposal addressing all of a deliverable's requirements will be favored over one addressing only a subset. Each Bidder is at liberty to control its own proposal, of course. But if it does choose to propose only a subset for any particular deliverable, it might help if the Bidder prominently and unambiguously states precisely what subset of the deliverable requirements are being proposed.
 - The Sponsor(s) will be given an opportunity to review selection results and offer advice, but ultimately the Participation Agreement (PA) contracts will be formed bilaterally between OGC and each Participant organization. No multilateral contracts will be formed. Beyond this, there are no restrictions regarding how a Participant chooses to accomplish its deliverable obligations so long as the Participant's obligations are met in a timely manner (e.g., with or without contributions from third-party subcontractors).
 - In general, only one organization will be selected to receive cost-share funding per deliverable, and that organization will become the Assigned Participant upon which other Participants will rely for delivery. Optional in-kind contributions may be made provided that they don't disrupt delivery of the required, reliable contributions from assigned Participants.

- A Bidder may propose against any or all deliverables. Participants in past initiatives have often been assigned to make only a single deliverable. On the other hand, several Participants in prior initiatives were selected to make multiple deliverables.
- In general, the Participant Agreements will not require delivery any component source code to OGC.
 - What is delivered instead is the behavior of the component installed on the Participant's machine, and the corresponding documentation of findings, recommendations, and technical artifacts as contributions to the initiative's Engineering Report(s).
 - In some instances, a Sponsor might expressly require a component to be developed under open-source licensing, in which case the source code would become publicly accessible outside the Initiative as a by-product of implementation.
- Results of other recent OGC initiatives can be found in the [OGC Public Engineering Report Repository](#).
- A Bidders Q&A Webinar will likely be conducted soon after CFP issuance. The webinar will be open to the public, but prior registration will be required.

Appendix C: Abbreviations

The following table lists all abbreviations used in this CFP.

<i>CFP</i>	Call for Participation
<i>CR</i>	Change Request
<i>DER</i>	Draft Engineering Report
<i>DWG</i>	Domain Working Group
<i>ER</i>	Engineering Report
<i>GPKG</i>	GeoPackage
<i>IP</i>	Innovation Program
<i>OGC</i>	Open Geospatial Consortium
<i>ORM</i>	OGC Reference Model
<i>OWS</i>	OGC Web Services
<i>PA</i>	Participation Agreement
<i>POC</i>	Point of Contact
<i>Q&A</i>	Questions and Answers
<i>RM-ODP</i>	Reference Model for Open Distributed Processing
<i>SOW</i>	Statement of Work
<i>SWG</i>	Standards Working Group
<i>TBD</i>	To Be Determined
<i>TC</i>	OGC Technical Committee
<i>TEM</i>	Technical Evaluation Meeting
<i>TIE</i>	Technology Integration / Technical Interoperability Experiment
<i>URL</i>	Uniform Resource Locator
<i>WFS</i>	Web Feature Service
<i>WPS</i>	Web Processing Service
<i>WG</i>	Working Group (SWG or DWG)

Appendix D: Corrigenda & Clarifications

The following table identifies all corrections that have been applied to this CFP compared to the original release. Minor editorial changes (spelling, grammar, etc.) are not included.

Section	Description
B.2. Questions and Clarifications	Repaired link to Clarifications
—	—

The following table identifies all clarifications that have been provided in response to questions received from organizations interested in this CFP.

Question	Clarification
— 9 January —	—
<i>In prior testbeds, cost-share funding from the European Space Agency (ESA) was offered under a separate Invitation to Tender solicitation (followed by separate Participation Agreement contracts). Will any of the Testbed-16 deliverables be handled in this manner?</i>	No, all deliverables in Testbed-16 will fall under this single CFP solicitation. Participation Agreements will be executed with OGC, with all cost-share funding to be provided in US dollars (USD). All currency amounts entered into the Bid Submission Form should be in USD.
— 10 January —	—
<i>We are interested in participating separately in the GeoPackage portion of Testbed 16 (section 2.5). But we would only be interested in the metadata extension portion of this item. The way this section is structured, it seems difficult to separate the metadata portion from the performance goals. How might we go about submitting for only the metadata extension improvements?</i>	Please submit a proposal for D010 but emphasize that you want to concentrate on the metadata extension portion exclusively. We will then try to find matching participants covering other parts of D010.
<i>Section A.4, second sentence states “A proposal template will be made available”. Where is the template, and will it contain additional guidance?</i>	As was the case for Testbed-15, the Testbed-16 template will be an online Bid Submission Form . There are no Word document or Excel spreadsheet templates. Bidders will enter data directly into fields on the form. The form should be available no later than Thursday, 16 January. In the meantime, Detailed Instructions can be found earlier in the CFP.
— 21 January —	—
<i>Are Wildland fire planning and Wildland fire response two consecutive Phases of the project or will they be developed in parallel?</i>	They will be developed in parallel.
<i>On section 2.2.3. Scenario, there is some work to do with "evacuation and first responder routes", which involves either deploying a routing engine or working with an external one. Is there a preference or a predefined routing engine?</i>	There is no preference and no predefined engine.

Question	Clarification
<p><i>Regarding the three deliverables on Machine Learning tools, D132/133/134, what are the interactions between them? Is D134 assigned to different tasks than D132/133 (for example, D134 for LiDAR data and D132/133 for satellite images)? Are D132 and D133 assigned to do the same tasks and expected to deliver similar results?</i></p>	<p>There are interactions between them. It is expected that different models will be explored. Bidders shall identify what their model requires and provides. Proposals can propose D132/133 for arbitrary models, but D134 shall be a deep learning environment as described.</p>
<p><i>Even though there are some specific ML frameworks mentioned for D132/133 (Mapbox RoboSat, Azavea's Raster Vision, NRCan's GeoDeepLearning) the Deliverable does not need to be restrained to just one of these frameworks but it could be built up combining several frameworks, libraries and ML technologies if deemed necessary. Is this correct?</i></p>	<p>Yes, this is correct.</p>
<p><i>Are D135 and D136 assigned to do the same tasks and expected to deliver similar results?</i></p>	<p>Both are expected to be training data sets. There is no need to deliver similar results or address the same data.</p>
<p>— 22 January —</p>	<p>—</p>
<p><i>The CFP under section "2.7.4. Scenario & Requirements" speaks of a persona described as "the non-geospatial expert consumer type of user". Could we be provided more detail about this persona, their needs and expectations?</i></p>	<p>The persona we have in mind has no understanding of CRS or projections. The person does not know about OGC APIs, but wants to identify some area on the globe to e.g. provide a reference to a city, region or similar. The persona wants to share cell IDs and is not in need to provide polygons with holes etc. Vice versa, the persona might have received some cell IDs and wants to know where these cells are located.</p>
<p><i>For a use case we are considering that highlights some of the advantages of DGGGS, a web connected mobile client app would be the best demonstration. Would such an approach be considered in lieu of or in addition to a browser based client?</i></p>	<p>Yes, that is a valid alternative.</p>
<p><i>Alternatively, would the use of a HTML5 webapp that can run offline be an acceptable approach to support a use case where web access is intermittent at best?</i></p>	<p>Yes, that is a valid alternative.</p>
<p><i>The storage of DGGGS data is mentioned in section "2.7.1. Problem Statement and Research Questions" and other places. Is the provision of a format that stores DGGGS defined features desired in this testbed? Or would this best be covered under a future work section of the engineering report?</i></p>	<p>The data we have in mind for D139 is not DGGGS data, but data that is provided by an interface that supports cell-IDs as filters. The data itself might remain untouched from what exists. Alternatively, data is annotated with cell-IDs instead of or in addition to its current spatial elements. The goal of the Testbed is to better understand what is required, what needs to change, and to identify a working solution. If you refer to format as in "container format", then no, this is not required for this Testbed.</p>

Question	Clarification
<p><i>Is this testbed expected to interact with Eurocontrol SWIM taxonomy, registry and discovery services? Only FAA is specified in the text, but the figure includes Eurocontrol.</i></p>	<p>Please see sections for 2.1.1 and 2.1.2. The goal is to answer the research questions listed there. This does not mean systems other than FAA must be part of the investigation, though it might not be possible to ignore them fully in order to develop a holistic view on the problem space and possible solutions. Figure 4 in section 2.1.5 show this by not having any arrows between the work items and Eurocontrol SWIM.</p>
<p><i>What is ROK SWIM (KAC)? Does it have any specific relevance for the scope of this testbed?</i></p>	<p>This refers to the Korean aviation control. It is listed here to show that the world is bigger than FAA and Eurocontrol, which might cause additional options and challenges when thinking about solutions using semantic web technology and principles.</p>
<p>— 23 January —</p>	<p>—</p>
<p><i>CFP2.7.1 proposes the research question of determining “what DGGs structure would be best for developing a reference implementation”. This question is closely aligned with one of the core goals of the OGC DGGs Registry. Given that the OGC DGGs Registry work has been part of a separate piece of work to the IP program, will it be appropriate to propose the OGC DGGs Registry be further developed and leveraged by TestBed 16 as a DGGs configuration/profile library to support D138 and D139?</i></p>	<p>Yes, that is appropriate.</p>
<p><i>CFP 2.7.3 DGGs – Previous work - The data model for DGGs is under revision as ISO 19170-1, which will be brought back as an update to OGC 15-104r5 OGC DGGs Abstract Specification. Is it appropriate to base D017, D137, D138, & D139 on the new 19170-1 data model that is available in the ISO Harmonised Model?</i></p>	<p>Yes.</p>
<p><i>Is it appropriate to propose DGGs based solutions to other parts of the Testbed, either as actual components or as an auxiliary ER chapter proposing a future alternative DGGs architecture? For example: CFP 2.3 Data Access and Processing API (DAPA) for Geospatial Data, CFP 2.5.1 (2) use of DGGs IDs and structures for Large Vector Datasets in GeoPackage, CFP 2.8 Analysis Ready Data (ARD), CFP 2.11 OpenAPI use of ShapeChange to configure OGC DGGs API?</i></p>	<p>Yes, it is appropriate.</p>
<p><i>Timetable - CFP 1.4 when is the earliest date that OGC expects to be able to confirm that CFP Proposal Submissions have been accepted?</i></p>	<p>Submitters will receive an immediate confirmation from the Bid Submission website that their proposal has been received. We are hoping to begin sending out preliminary offers to conduct negotiations around Wednesday, 19 February.</p>

Question	Clarification
<p><i>Kickoff and Interim Workshop - Given the current Climate Crisis, the distance to the US from other continents, and the significant contributions that flying makes to carbon footprints, what support will OGC and USGS as hosts be providing to maximise the benefit for virtual attendance at the Kickoff and Interim workshops? Will virtual attendance be considered as acceptable for the Kickoff workshop?</i></p>	<p>As indicated in Miscellaneous, Participants are required to attend the Kickoff, including breakout sessions, and will be expected to use these breakouts to collaborate with other Participants and confirm intended Component Interface Designs. In fact, cost-share Participants often include Kickoff travel costs in formulating their required in-kind contribution.</p> <p>Exceptions might be considered for extraordinary cases. For example, if an individual is unable to obtain a visa to travel to the U.S., and no alternative arrangement can be made (for example, someone who could stand in as a substitute), we might consider relaxing this requirement.</p> <p>But even in this case, the quality of participation isn't ideal. The Kickoff is a workshop intended to operate as a face-to-face meeting. The setup is similar to TC Meeting breakout sessions, attended by many individuals spread around the meeting room. The quality of remote participation is almost always suboptimal.</p> <p>The requirements for the Interim Workshop are much less stringent, however. As indicated in the Master Schedule, this workshop will be held concurrent with the TC Meeting. Participation is not mandatory but will be appreciated. It is left to each Participant to decide whether to attend or not. However, like the Kickoff, the Interim Workshop is intended to operate as a face-to-face meeting, and the quality of remote participation is likely to be lower.</p>
<p>— 24 January —</p>	<p>—</p>
<p><i>One of the paragraphs under General Proposal Submission Guidelines indicates that "Information submitted in response to this CFP will be accessible to OGC and Sponsor staff members." Does this mean that sponsors will be able to view bidder cost information?</i></p>	<p>Sponsors will have access to all bidder inputs to the "Proposed Contribution" box. OGC recommends that bidders AVOID including any cost information in this box. The following bidder cost inputs are NOT accessible to sponsors: "Projected Labor Hours", "Total Dollar Funding Request", "Inkind Labor Hours", and "Inkind Contribution".</p>
<p>— 29 January (from Bidders Q&A Webinar) —</p>	<p>—</p>
<p><i>Are both D147 and D148 both mobile clients with different mobile operating systems?</i></p>	<p>Both are defined as mobile phone applications. They can be built on any mobile phone operating system.</p>
<p><i>Has the funding status of D019 changed?</i></p>	<p>No, unfortunately we still have a number of work items with no funding commitment. We are working hard to release an updated list before the deadline (see Deliverables Summary & Funding Status for details).</p>

Question	Clarification
<i>For D132/D133 a diagram in the CFP states "ML tools need to provide results in ML" but then the description states that the ML models shall provide results at OGC Web APIs or OGC Web service interfaces in a form that allows MapML clients to explore results. It is assumed the Client must consume MapML, but how is the output of the ML environment converted to MapML? Is this part of the scope of the API?</i>	ML tools shall provide results using MapML (ideal) or at an OGC Web API that then needs to be consumed by a MapML engine that is then part of a D130/131 client/server proxy. If you can only provide one or the other, it depends on the type of client proposals what we can match.
<i>How does the in-kind contribution apply to the universities?</i>	Please put in in-kind contribution details as supported by your university. A short paragraph on university policy helps.
<i>Could you give a rough estimate of the desired word count for the "succinct description" in the Proposed Contribution?</i>	That depends on the work item. We don't expect more than 1 text page per work item. If you need to add graphics, please provide the references in the description and attach an extra file.
<i>Is there a target date for releasing an updated CfP before the submission deadline?</i>	There is no clear target date. We depend on incoming messages/contracts from sponsoring organizations.
<i>Should D121 be something like QGIS? I know in the last testbed it was a command line client. I assume we want something more this time.</i>	Ideally yes, but key is that the necessary functionality is supported.
<i>For deliverables that are 'similar to', e.g. D165/D166/D167, we understand that this is similar deliverables that would be distributed to different parties. How are proposals to be submitted, individually for each or are they grouped in the proposal form?</i>	You just need to propose for one of them. We will assign your single proposal to one of them.
<i>What are reasonable amounts of cost sharing money that one can request?</i>	This depends on the work item. We cannot provide any details here.
<i>Are you allowed to disclose the sponsors for the funded tasks? If so, is this listed somewhere?</i>	It is not listed anywhere and we don't have full authorization to release all names, therefore we prefer at this time to keep that list private.
<i>Will FMV data be made available as part of the testbed?</i>	Yes, it will be made available by the sponsor.
<i>In the CFP (Machine Learning), information about Training Data is provided under Section 2.2.3. While information about top 2 datasets is available, no information is available about LiDAR datasets. So, would they be provided at a later stage to the selected bidders?</i>	We don't have further details about the LiDAR data.

Question	Clarification
<p>wrt virtual attendance to the Kickoff - I'm going to respond to your 'grumpy professor' comment, with a grumpy 'climate professor' type comment. The climate crisis is real, and our response to it requires new approaches to doing things .. my question was in full recognition of the values and mode of operation of traditional kickoff meetings. So I was asking whether OGC is prepared to be innovative in the way it conducts the workshop to find ways to resolve the issues associated with virtual attendance. As a global organisation driving sustainability - this is a very real question for participants from the far side of the globe. I accept that this may be too late to bring such innovation to this testbed, but I will push for further innovation in this space.</p>	<p>Yes, we are fully aware of the situation and are working on a solution for future projects!</p>
<p>Can deliverables be tackled by a team of different organizations and if so, would it be ok, to cross-reference the different bids in the description, by the participating organizations?</p>	<p>Yes, that is possible. Please provide a clear message on who does what.</p>
<p>How can we find answers to any additional questions?</p>	<p>Send an email to the OGC techdesk (techdesk@opengeospatial.org).</p>
<p>— 30 January —</p>	<p>—</p>
<p>It's understood that OGC will issue a Patent Call at the Kickoff Workshop. Have any patents or applications been identified so far?</p>	<p>Yes, an application with status "Pending" is active under US20170293635A1 at the U.S. Patent and Trademark Office. The title is "Methods and systems for selecting and analyzing geospatial data on a discrete global grid system".</p> <p>If a patent is awarded, it's possible that one of its claims could potentially read onto a "D138 DGGS Demo Client" component implementation as described under 2.7.5 Work Items & Deliverables, particularly a client that integrates data from multiple DGGS servers.</p> <p>OGC is making no statement regarding the validity of any such assertion of overlap. The purpose of this Clarification is simply to provide an early notification to potential bidders of the existence of the patent application.</p> <p>It should be noted that none of the patent application claims would cover the Topic 21: Discrete Global Grid Systems Abstract Specification standard itself.</p>
<p>— 6 February —</p>	<p>—</p>

Question	Clarification
<i>In the past, there was a place to detail our global in-kind contribution, where we included cost of licenses, travels, hardware, etc. Is there a place somewhere to detail that part? I only see in-kind contribution linked to a specific deliverable.</i>	There's no separate text box to enter a global in-kind contribution. Instead, please provide an approximate estimate on a per-deliverable basis. Any additional global explanation can optionally be included your "Document of Explanation", which can be attached before making the submission. Please keep in mind, though, that labor rates and other sensitive information should NOT be included in the Document of Explanation since it will be shared with others besides OGC staff.