



OGC Testbed 17

Making Location Information
Findable, Accessible,
Interoperable, and Reusable

ogc.org/testbed17

OGC Testbed-17 *Call for Participation (CFP)*

Version 1.20 - 22 March 2021

Table of Contents

1. Introduction	2
1.1. Background	2
1.2. OGC Innovation Program Initiative	2
1.3. Benefits of Participation	2
1.4. Master Schedule	3
2. Technical Architecture	4
2.1. Sensor Integration	5
2.2. Moving Features from Digital Motion Imagery	7
2.3. Web API for Aviation	11
2.4. Data Centric Security Across OGC APIs and Federated Clouds	16
2.5. Federated Cloud Analytics	20
2.6. Model Driven Standards	22
2.7. COG & Zarr: Specification & Evaluation	25
2.8. Attracting Developers: Lowering the entry hurdle for OGC Web API experiments	28
2.9. OGC Features and Geometries JSON	34
2.10. Geo Data Cubes	38
2.11. Compliance Interoperability & Testing Evaluation (CITE)	44
3. Deliverables Summary	47
4. Miscellaneous	52
Appendix A: Testbed Organization and Execution	54
A.1. Initiative Policies and Procedures	54
A.2. Initiative Roles	54
A.3. Types of Deliverables	54
A.4. Proposal Evaluation	56
A.5. Reporting	57
Appendix B: Proposal Submission	58
B.1. General Proposal Submission Guidelines	58
B.2. Questions and Clarifications	60
B.3. Proposal Submission Procedures	60
B.4. Tips for New Bidders	64
Appendix C: Abbreviations	67
Appendix D: Corrigenda & Clarifications	68
D.1. Corrigenda Table	68
D.2. Clarifications Table	68
D.3. End of Clarifications Table (convenience link)	77

Chapter 1. Introduction

The Open Geospatial Consortium (OGC) is releasing this *Call for Participation* (CFP) to solicit proposals for the **OGC Testbed-17**. The Testbed-17 initiative will explore eleven tasks, including Application Programming Interface (API) development, Features and Geometries JSON, Aviation, CITE, Geo Data Cubes, Sensor Integration, Moving Features from motion imagery, Security, Federated Cloud Analytics, COG & Zarr, as well as Model Driven Standards.



1.1. Background

OGC testbeds are an annual research and development initiative that explores geospatial technology from various angles. They take the [OGC Baseline](#) into account, and at the same time explore selected aspects with a fresh pair of eyes. 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](#). This 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.

Table 1. Master schedule

Milestone	Date	Event
M01	07 December 2020	Release of Call for Participation (CFP)
M02	06 January 2021 12 January 2021	Questions for CFP Bidders Q&A Webinar Due
M03	13 January 2021	CFP Bidders Q&A Webinar to be held 10:00-11:00 EST.
M04	24 January 2021	CFP Proposal Submission Deadline (11:59pm EST)
M05	31 March	All testbed Participation Agreements Signed. OGC will start sending preliminary offers to conduct negotiations no later than early March (and perhaps as soon as mid-February).
M06	6-8 April	Kickoff Workshop to be held in the Washington, DC area or organized as a virtual meeting. Details will be posted (when available) on the Testbed-17 Initiative Page .
M07	31 May	Initial Engineering Reports (IERs)
M08	June (specific date TBD)	IER presentations at Member Meeting
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 meet 3-Week-Rule for December publication vote
M12	December (specific date TBD)	Outreach presentations at Member Meeting
M13	15 December	Participant Final Summary Reports due

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](#).

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 using the *Additional Message* textbox in the [OGC Innovation Program Contact Form](#).

Testbed Threads

The Testbed is organized in a number of tasks. For organizational purposes, these tasks are further organized in threads. Each thread combines a number of tasks that usually share architectural or thematic aspects. Threads allow us to keep related work items closely together.

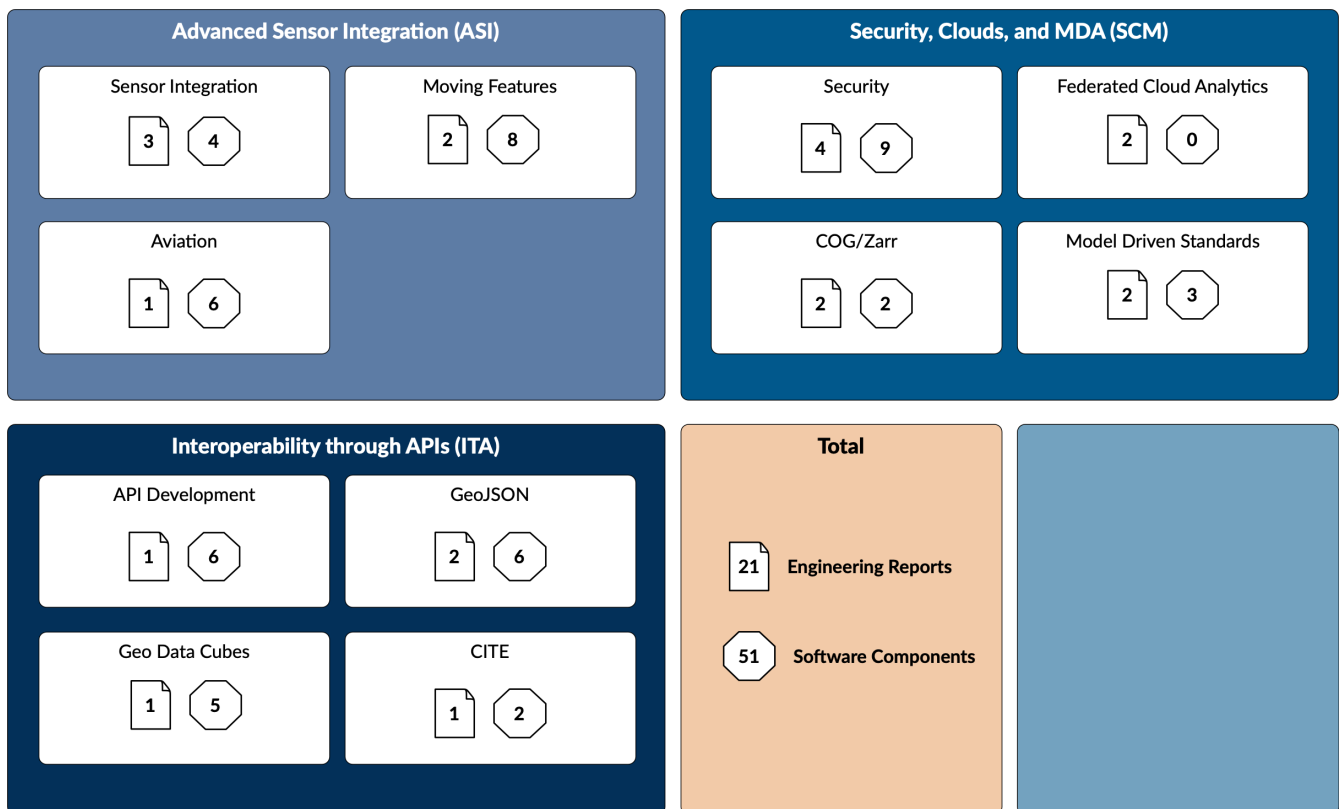


Figure 1. Testbed-17 Threads and Tasks

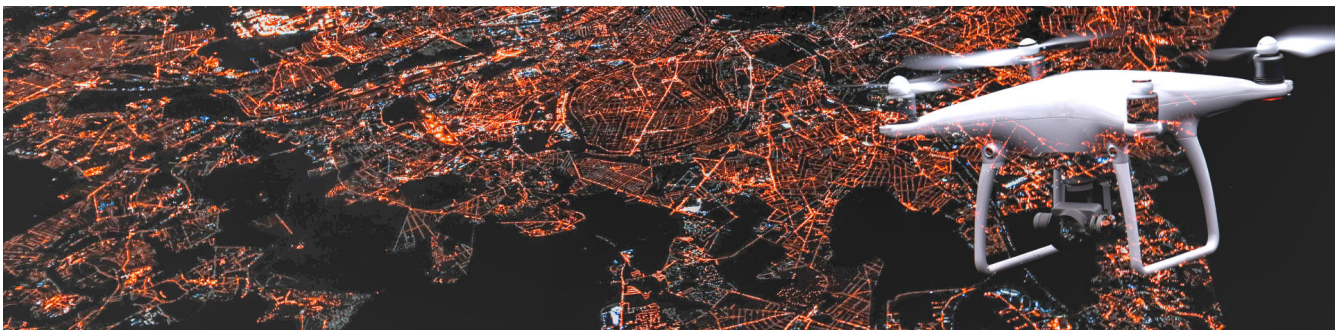
The threads include the following tasks

- Thread 1: **ADVANCED SENSOR INTEGRATION (ASI)**
 - [Sensor Integration](#)
 - [Moving Features from Digital Motion Imagery](#)
 - [Web API for Aviation](#)
- Thread 2: **SECURITY, CLOUDS, AND MDA (SCM)**
 - [Data Centric Security Across OGC APIs and Federated Clouds](#)

- [Federated Cloud Analytics](#)
- [Model Driven Standards](#)
- [COG & Zarr: Specification & Evaluation](#)
- Thread 3: **INTEROPERABILITY THROUGH APIS (ITA)**
 - [Attracting Developers: Lowering the entry hurdle for OGC Web API experiments](#)
 - [OGC Features and Geometries JSON](#)
 - [Geo Data Cubes](#)
 - [Compliance Interoperability & Testing Evaluation \(CITE\)](#)

2.1. Sensor Integration

Sensor systems are built using many different standards, formats, and protocols. This is a significant barrier to sensor integration. Yet there are good reasons why this should be so. Sensors must be deployed where they will best measure a physical phenomenon that imposes non-negotiable constraints on their size, weight, power, and communications capabilities. These constraints limit the options available to the designer. They must select the best option that will work within their technical constraints.



Sensor standards must embrace this diversity. What is required is not a one-size-fits-all solution, but a framework of standards which enable integration of sensors regardless of the technical constraints.

2.1.1. Aim

Evaluate previous work in this area and propose a way forward toward a standards-based solution.

2.1.2. Previous Work

The [Sensor Integration Framework](#) The Sensor Integration Framework (SIF) provides a standards framework for the integration of sensor systems regardless of their technical constraints. The SIF proposes a set of Technical Views (TV). Each Technical View is representative of a typical deployment environment and its associated constraints. A TV defines the standards and provides guidance for the development of sensor systems within the constraints of that deployment environment. The set of TVs forms a taxonomy of the known deployment environments. The designer of a sensor system can select the TV appropriate to their requirements, and design to the standards and guidance provided by that TV.

In addition to the Technical Views, the SIF provides a framework for information flow between the Technical Views. This framework defines a mediation service (or services) which assures that a sensor, no matter where deployed, is accessible to the whole federation.

Mediation between TVs takes place on three levels; protocol, format, and information.

- Protocol: There are many options for connecting to a sensor. They range from a RS232 serial cable to a gigabit Internet connection. However, protocol mediation is an old problem and the techniques are mature.
- Format: Format mediation is also well understood as long as both formats adhere to the same information model. Difficulties arise where there is not a clean mapping between data elements.
- Information: Information mediation is an evolving field. This is complicated by the fact that most formats combine semantics and syntax. The SIF addresses this issue by leveraging SWE Common, a semantics-free syntax that is part of the Sensor Web Enablement (SWE) suite of standards. Prototype software has demonstrated transformation of messages over Denied, Degraded, Intermittent, or Limited Bandwidth (DDIL) networks into SWE-Common messages. However, the semantic mediation capabilities are still in early stages of development.

MASBUS

The Measures and Signatures Intelligence Enterprise Service Bus (MASBUS) is a pilot implementation of the SIF developed by the U.S. Defense MASINT Office (DMO). It provides a Sensor Observation Service which is capable of mediating content from tactical sensors. MASBUS has been exercised and validated in a number of Enterprise Challenge events. The DMO plans to make MASBUS available to Testbed-17 participants as an Open Source project.

SIF Semantic Model

The SIF Semantic Model addresses the information mediation requirement. It is based on the Semantic Sensor Network Ontology developed by the W3C and OGC. Additional concepts are identified and defined through examination of the data models of existing sensor systems and standards. While it resembles an ontology, some deviations from strict orthodoxy have been taken in the interest of accuracy and usability. This is a work in progress that shall be supported with this Testbed-17 task.

2.1.3. Scenario & Requirements

The following sub-work items will be performed under the Sensor Integration Work Item:

- Review and comment on the SIF. Assess its suitability as a starting point for an OGC SIF effort.
- Review and optionally enhance the MASBUS software. Assess its suitability as a SIF reference implementation.
- Develop and test integration of OGC SensorThings API implementations into MASBUS using the MQTT protocol.
- Review the SIF Semantic Model. Analyze its potential to grow into a comprehensive semantic model for sensor data. Provide recommendations on if and how it should evolve into an OGC resource.

2.1.4. Work Items & Deliverables

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

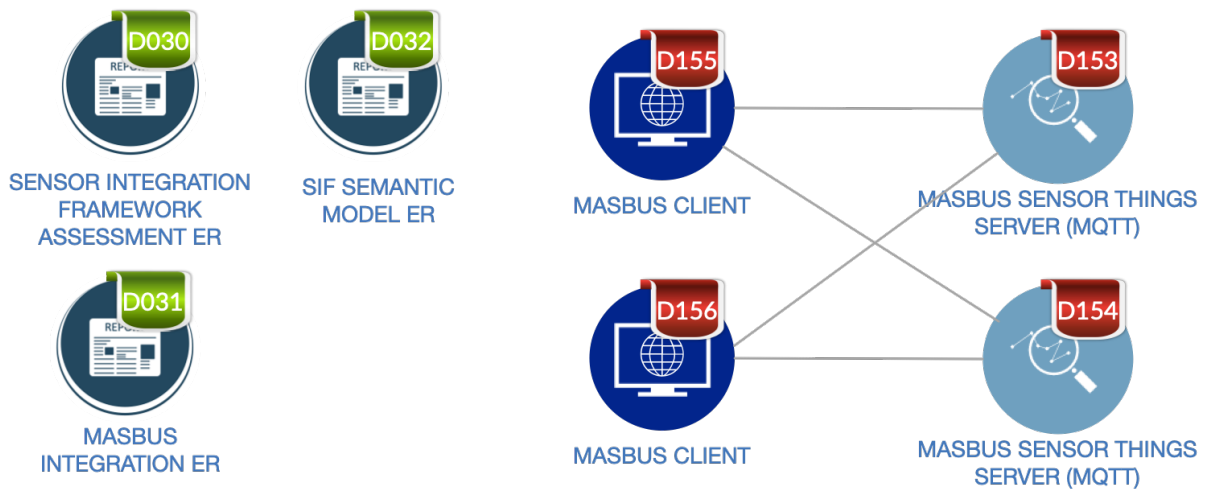


Figure 2. Sensor Integration task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D030 Sensor Integration Framework Assessment ER** – An Engineering Report which documents analysis performed, the conclusions, and recommendations resulting from the analysis of the Sensor Integration Framework.
- **D031 MASBUS Integration ER** – An Engineering Report on the analysis of the MASBUS pilot software and the results of efforts to integrate with OGC SensorThings API resources.
- **D032 SIF Semantic Model ER** – An Engineering Report on the SIF Semantic Model including recommendations on how to leverage this work to mature the sensor semantic resources provided by the OGC.

Components

- **D153 MASBUS Server** - Server implementation that demonstrates the Sensor Things MQTT extension of the MASBUS software.
- **D154 MASBUS Server** - similar to D153
- **D155 MASBUS Client** - Client implementation that demonstrates the Sensor Things MQTT extension of the MASBUS software.
- **D156 MASBUS Client** - similar to D155

2.2. Moving Features from Digital Motion Imagery

OGC Moving Features play an essential role in many scenarios. The growing availability of digital motion imagery and advancements in machine learning technology will further accelerate widespread use and deployment of moving feature detection and analysis systems. This task

accounts for these developments by addressing exchange of detections, shared processing of detections for correlation and analysis, and visualization of moving objects within common operational pictures. This task will explore and develop an architecture for collaborative distributed object detection and analysis of multi-source motion imagery. It is the goal to define a powerful API for discovery, access, and exchange of moving features and their corresponding tracks and to exercise this API in a near real-time scenario.



2.2.1. Problem Statement and Research Questions

There are a number of ways that systems detect and report on moving objects. These systems exist in “stovepipes of excellence”. As a result, users of these systems do not have access to information generated through other means. The ability to combine multiple sources of moving object data would greatly improve the quality of the data and the analytics which could be applied.

2.2.2. Aim

To identify an architecture framework and corresponding standards which will allow multiple sources of moving object detections to be integrated into a common analytic environment.

2.2.3. Previous Work

Testbed 16 explored technologies to transform detections of moving objects reported using motion imagery standards ([MISB Std. 0903](#)) into OGC Moving Features ([OGC 18-075](#)).

That work suggests a notional workflow:

1. Extract moving object detections from the motion imagery stream
2. Encode the detections as moving features
3. Correlate the detection of moving features into track moving features
4. Perform analytics to enrich and exploit the tracks of moving features

This work is documented in the Testbed-16 Full Motion Video to Moving Features Engineering Report ([OGC 20-036](#)).

The OGC Moving Features Standards Working Group (SWG) is currently planning to add a new activity related to OGC APIs to the working group charter. It is expected that this process will be completed at the time of Testbed-17 kick-off. The IP team will watch this process closely to ensure

both activities are aligned properly. In any case, Testbed-17 participants will be required to work closely with the SWG and coordinate all efforts.

2.2.4. Scenario & Requirements

Testbed 16 demonstrated that Motion Imagery derived Video Moving Target Indicators (VMTI) can be extracted from an MPEG-2 motion imagery stream and represented as OGC Moving Features. This Sub-Work Item shall formalize an architecture for integrating moving object detections, propose standards for the required APIs and content encodings, expand the sources of moving object detection that can be supported, and explore exploitation and enhancement capabilities which would leverage the resulting store of moving features.

The architecture shall include the following components:

1. **Detection ingest:** This component will ingest data from a moving object detection system, extract detections and partial tracks (tracklets), and export the detections and tracklets as OGC Moving Features.
2. **Tracker:** This component ingests detections and tracklets as OGC Moving Features, then correlates them into longer tracks. Those tracks are then exported as OGC Moving Features
3. **Data Store:** provides persistent storage of the Moving Feature tracks.
4. **Machine Analytics:** software which enriches the existing tracks and/or generates derived information from the tracks
5. **Human Analytics:** software and tools to help users exploit the Motion Imagery tracks and corresponding detections or correlated tracks. For example, a common operational picture showing both static and dynamic features.

This list of components and their definitions serve as a starting point. Participants in this task are free to modify them as conditions require. This work shall be demonstrated using a real-time situational awareness scenario. Ideally, participants experiment with both subscription models as well as data streams to trigger prompt updates in the analytics components based on Moving Feature behavior. Two scenario suggestions are:

1. **Wild Fire response:** Provide a real-time situational map of the response to a wild fire. In addition to static map data, the situation map would include:
 1. The current extent and movement of the fire represented as a moving feature which a polygon geometry.
 2. Fire fighters (individuals) and their equipment.
 3. Routes of ingress and egress of all aircraft over the fire zone as well as their current position.
 4. Unidentified movers in the fire zone – people who shouldn't be there.
2. **Counter Poaching:** In the Masai Mara game parks only the rangers are allowed out after dark. If you are not a ranger or an animal then you must be a poacher. Presume an overhead surveillance system that:
 1. Identifies the current location and movement of all rangers and their vehicles.
 2. Identifies the current location, movement, and type of all large animals. An AI analytic

which classifies animals by their movement would be useful here.

3. Identifies poachers (any mover which is not a ranger or animal)

Participants are free to devise their own scenario if necessary.

2.2.5. Work Items & Deliverables

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

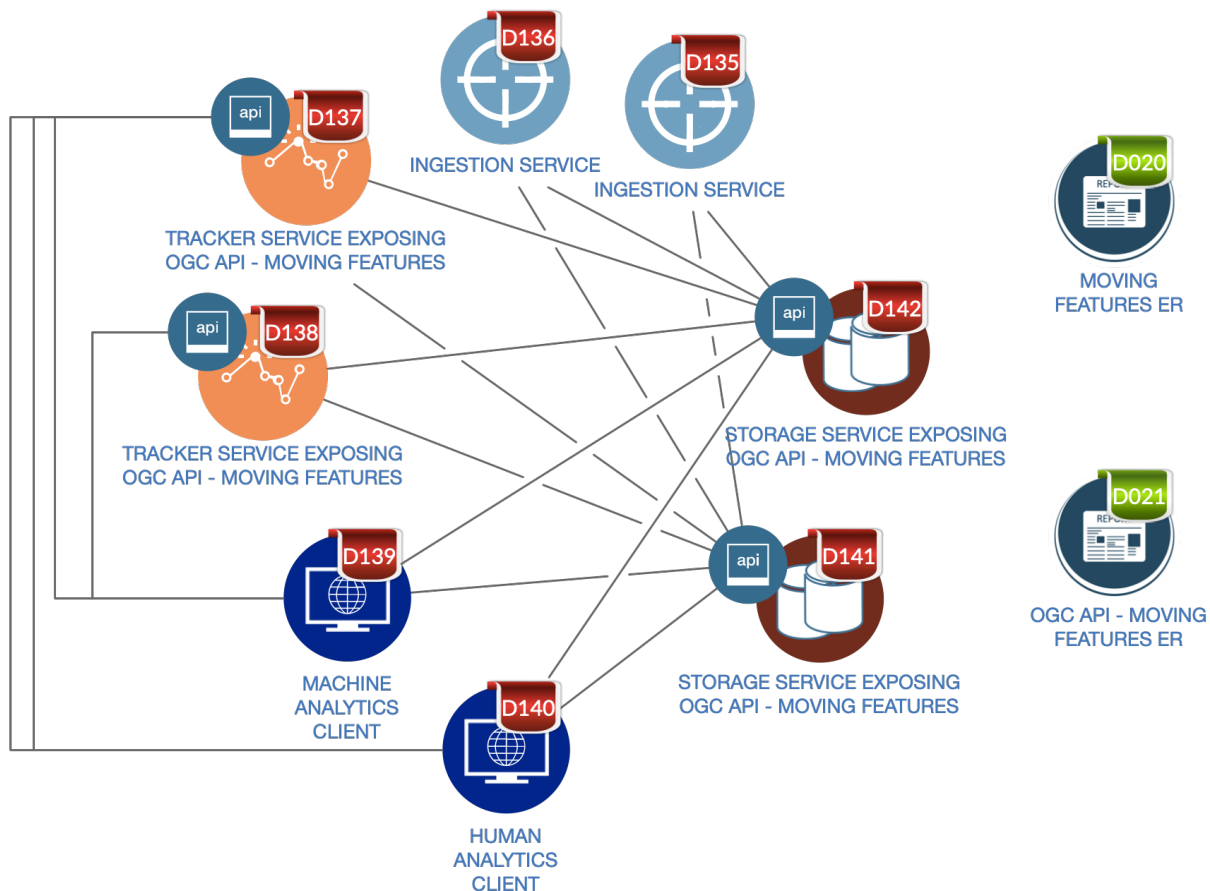


Figure 3. Moving Features task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D020 Moving Features Engineering Report** - Engineering Report that captures the proposed architecture, identifies the necessary standards, describes all developed components, reports on the results of all TIE activities, provides an executive summary and a description of recommended future work items.
- **D021 OGC API - Moving Features Engineering Report** - An Engineering Report that delivers an initial draft specification of the new OGC Web API.

Components

- **D135 Ingestion Service** - Software component that ingests data from a moving object detection

system, extracts detections and partial tracks (tracklets), and exports the detections and tracklets as OGC Moving Features to the Storage Service via the new OGC API - Moving Features. The component provider shall make the data set available that has been used for object detection to other participants in this task. If no source data is found for the final use cases, OGC and sponsors will help finding appropriate video material. The component can be implemented as a microservice or client.

- **D136 Ingestion Service** - component similar to D135.
- **D137 Tracking Service** - Service component that correlates detections and tracklets into longer tracks. Those tracks are then exported as OGC Moving Features to the Storage Service via the new OGC API - Moving Features. In addition, the service shall expose the new OGC API - Moving Features to allow other software components to discover and access tracks directly. The Tracking Service can work on its own detection system, but shall access detections and tracklets from the Storage Service. Ideally, the service supports subscriptions.
- **D138 Tracking Service** - similar to D137
- **D139 Machine Analytics Client** - Client component that provides OGC Moving Feature analytics and annotation. The client shall enrich existing tracks and/or generate derived information from the tracks. The software shall demonstrate the value added of multi-source track data. Enriched OGC Moving Features shall be stored in the Storage Service. In contrast to the Client D140, this client focuses on the analytics. It accesses external or uses internally available additional data sources, e.g. road and hiking path network data, to annotate detected moving objects in the scenarios.
- **D140 - Human Analytics Client** - Client software and tools to help users exploit the multi-source track data. For example, a common operational picture showing both static and dynamic features. In contrast to the Machine Analytics Client, focus is here on graphical representation of OGC Moving Features, detected and annotated from multiple source systems, in a common operational picture.
- **D141 Storage Service** - Service component that stores OGC Moving Features. The service exposes the new OGC API - Moving Features to discover, access, and upload OGC Moving Feature resources. The storage service shall have the potential to serve tracks in near real time.
- **D142 Storage Service** - similar to D141

2.3. Web API for Aviation

The [System Wide Information Management](#) (SWIM) Program is a National Airspace System (NAS)-wide information system that supports Next Generation Air Transportation System (NextGen) goals. Testbed-17 will experiment with the emerging OGC Web API to elevate the existing SWIM service-based architecture to Web API level.



Testbed-17 will experiment with convenience APIs, i.e. APIs that are optimally tailored to the specific needs of the SWIM data consumer community. The resulting OGC API -Aviation will be made from the OGC API building blocks and ensure a high level of interoperability across all SWIM data providers.

2.3.1. Problem Statement and Research Questions

The OGC API family of standards are being developed to make it easy for anyone to provide geospatial data to the web. These standards define resource-centric APIs that take advantage of modern web development practices. The OGC APIs are being constructed as "building blocks" that can be used to assemble novel APIs for web access to geospatial content. The building blocks are defined not only by the requirements of the specific standards, but also through interoperability prototyping and testing in OGC's Innovation Program.

The emerging OGC API standards allow developers to fully leverage web technologies and principles for setting up service instances. Leveraging the OpenAPI specification, the OGC APIs will enable cost-efficient setup of new services. Instead of months, new service instances can become operational within weeks. This effect is primarily based on a fundamental revision of the customer/provider relationship. The previous generation of OGC Web Services was based on the principle of providing full explorative support to the data consumer (customer/end-user). In consequence, the service provider offered all data at the interface and supported a rich query language. It was up to the data consumer to build the correct queries to access required subsets. This approach was cumbersome to both parties. Instead of offering customer tailored products, the data provider had to support a rich and complex query language that allowed the data consumer to "build their own products". The data consumer had to build complex filter statements to get to the required data. Often enough, it was difficult for the data consumer to find out what query options actually existed.

The new OGC API-based approach now turns this principle around. The service provider now builds tailored products. These products are accessible by calling a specific URL that takes the consumer to a general landing page in the simplest case. From there, the customer can follow links to specific products. Thus, instead of building complex filter queries, the customer can explore all offerings and learn about available products, similar to following hyperlinks on websites. The data consumer only filters if further refinement is required, with support for building filter queries being part of the API definition. The new API paradigm fully leverages web technologies and is built entirely on HTTP operations and principles, which has the great advantage that web developers can code against the new OGC APIs like any other resource-oriented Web API.

The new OGC API principles have enormous potential in the context of data integration across

several data providers. Because the data is served in a custom-made fashion, data fusion can be implemented quickly and cost efficiently. The OpenAPI-based Web API definitions and the HTML-encoded landing pages allow customers to understand what is offered at a service instance. The support for multiple encodings (JSON, JSON-LD, XML, RDF flavors, etc.) and the encoding negotiation at runtime allow customers to access data in their preferred model and encoding. The general OGC API support for links allows for integration of strong semantics, which is essential to ensure a common understanding of all data offered at an interface. It is an essential prerequisite for data fusion, where several data sets are combined in order to generate higher level knowledge and information.

2.3.2. Aim

Explore the potential of OGC Web APIs in the context of SWIM. Develop an OGC API -Aviation (Aviation API) based on OGC Web API building blocks that enables convenient access to existing SWIM services. Then, demonstrate the capabilities of the new Aviation API in a data fusion scenario.

2.3.3. Previous Work

The following non-exhaustive list identifies previous work that shall be considered in the context of this activity:

- [OGC 20-020, OGC Testbed-16: Aviation Engineering Report](#)
- [OGC: OGC 17-069, OGC API - Features - Part 1: Core \(2019\)](#)
- [OGC Testbed-14: SWIM Information Registry Engineering Report \(18-022r1\)](#)
- [OGC Testbed-14: Semantically Enabled Aviation Data Models Engineering Report \(18-035\)](#)
- [OGC: OGC 11-052r4, OGC GeoSPARQL - A Geographic Query Language for RDF Data \(2012\)](#)
- [EUROCONTROL, FAA, NGA: AIXM 5.1 Specification \(2010\)](#)
- [EUROCONTROL, FAA: FIXM Core v4.2.0](#)
- [IETF: RFC-7946 The GeoJSON Format \(2016\)](#)
- [Facebook: GraphQL Specification \(2018\)](#)
- [W3C: RDF Schema 1.1 \(2014\)](#)
- [W3C: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax \(2012\)](#)

2.3.4. Scenario & Requirements

The aviation scenario is illustrated in [Figure 4](#). Testbed-17 will exercise the Aviation API with traditional SWIM services ([Figure 4: 4a/b](#)) that will be made accessible via the new Aviation API by means of a façade service ([Figure 4: 3a/b](#)). Data from these services will be fused and offered as a new product at the fusion service ([Figure 4: 2](#)). Two clients shall be developed ([Figure 4: 1a/b](#)) as part of this scenario. One with focus on ease-of-use for end-users, one with focus on functionality for developers.

The first client shall be optimized for use by domain experts, but not developers. This client focuses

on discovery, access, and visualization of SWIM and fusion resources. The client shall convey a sense of an end-user friendly client that interacts with future Web API-enabled SWIM services. The client shall support both raw and fusion products. The client may hide complexity behind the graphical user interface as necessary. As an example, the end-user in the scenario described below shall be able to see airports that are sensitive to bad weather with some statistics in an attractive way, but certainly not as JSON or other raw format.

The second client shall be developed from a developer’s perspective. Here, the focus is less on visualization and ease-of-use, but on functionality and experimentation. The client shall allow exploring all capabilities of Aviation-API service instances, shall support following links, and help the developer with creation of queries and result analysis. In this case, delivering JSON to the user is a valid option, though other forms of quick result inspection shall be supported as well.

Both clients shall demonstrate the use of the Aviation API with both raw as well as fused SWIM data.

One important element in the context of Aviation data is semantic interoperability. Therefore, Testbed-17 shall build on and stress-test previous Testbed results for operational use. The following diagram shows logical setup for the envisioned Testbed-17 scenario.

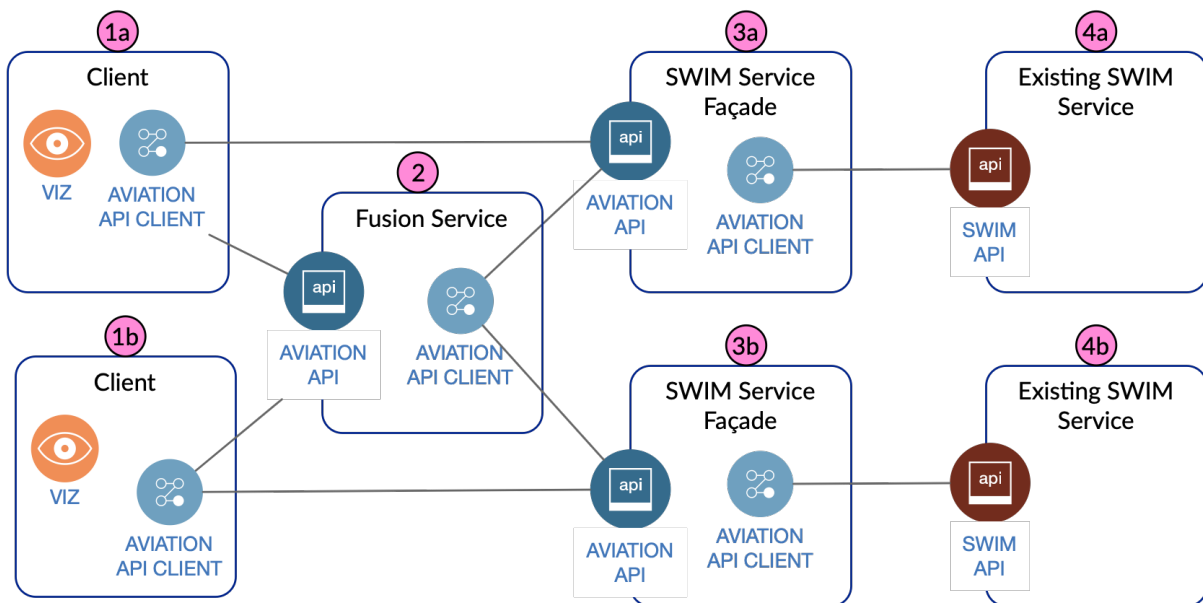


Figure 4. Logical Aviation Scenario with main components. Important: Two fusion services will be developed and need to be supported by both clients. The second fusion service is not illustrated to enhance readability.

In more detail, Testbed-17 will develop two clients, two fusion services, and two SWIM service façades. The fusion services use two or more other services to produce a higher level product. They shall not only orchestrate data, but add value to the data and the resulting product. A possible use case is two services, one for weather data and one for arrival/departure time data that are fused. Using the data from both services, the fusion process identifies airports that are sensitive to "bad weather". That is, the fusion service identifies airports that experience more delays than others on average in bad weather situations. The fusion service needs to load data from the two services, but needs to add intelligence to make sure it produces meaningful results. Alternative scenarios are possible and depend on both service/data availability and possible extensions as suggested by Testbed participants. Bidders are invited to suggest concrete use cases. The final decision on the use

cases will be made at the kick-off meeting. It is required that the capabilities of the new Aviation API are demonstrated for both fused and raw data.

The Aviation API will be developed similar to the other OGC Web APIs. It shall adhere to the requirements set forth by the OGC Standards Program and shall make use of OpenAPI and SwaggerHub. An OpenAPI definition shall be delivered together with the API Engineering Report.

2.3.5. Work Items & Deliverables

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

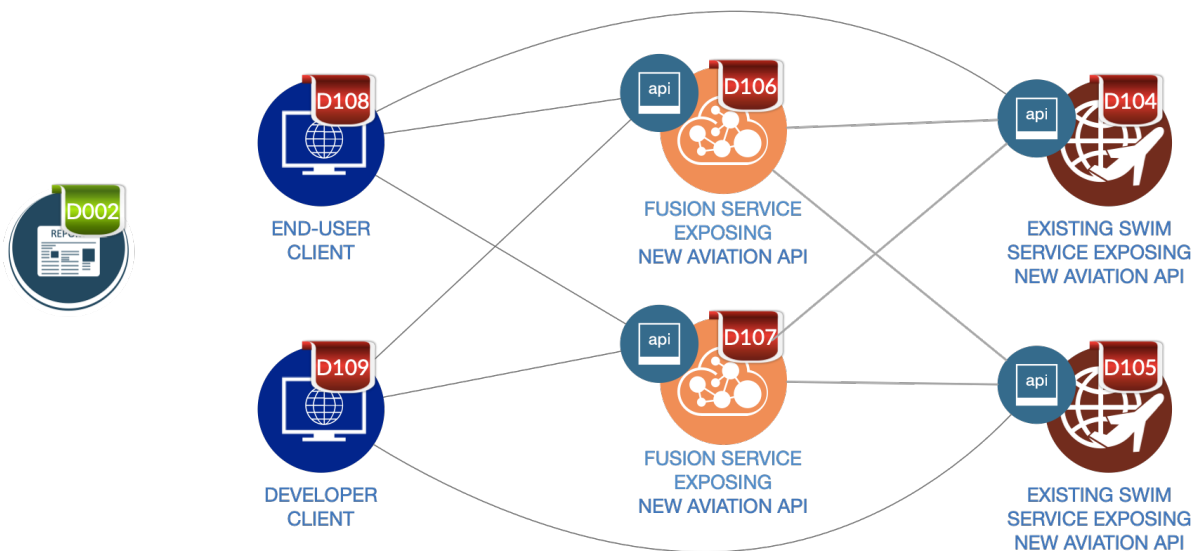


Figure 5. Aviation task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D002 Aviation API Engineering Report** - Engineering Report capturing all analysis performed, results, recommendations, and experiences from this task, including a detailed description of the new Aviation API in a format compliant to the requirements for OGC Standards Program. The API description may be delivered in a separate document to simplify the handling of both parts.

Components

- **D104 Existing SWIM Service Exposing new Aviation API** - A service component that enables access to an existing SWIM service via the new Aviation API. The component may act as a façade or is natively built on top of SWIM data.
- **D105 Existing SWIM Service Exposing new Aviation API** - similar to D104.
- **D106 Fusion Service Exposing new Aviation API** - A service component that supports fusion of data accessed via the Aviation API from both D104 and D105. The service in turn offers the fusion products at an Aviation API.
- **D107 Fusion Service Exposing new Aviation API** - similar to D106.

- **D108 End-user Client** - Client application that demonstrates the full scenario visually. The client shall access both fusion and raw data services. The client shall support the end-user perspective described above.
- **D109 Aviation Client** - Client application that demonstrates the full scenario visually. The client shall access both fusion and raw data services. The client shall support the developer perspective described above.

2.4. Data Centric Security Across OGC APIs and Federated Clouds

Data and software resources are typically protected through security controls which are provided by the hosting information technology infrastructure. This approach has limits which we are beginning to breach. Large scale distributed systems are not limited to a single platform. Software now travels between “clouds” to bring processing to the data. Internet of Things (IoT) devices reside on unreliable networks which may or may not allow access to security services. Data wanders far from home as it is passed from one recipient to another.



The OGC Innovation Program has shown that in a world that desperately needs better security models and infrastructure, there are potential ways to provide robust and secure solutions. These solutions implement the concepts of Data Centric Security (DCS) that keep data secure while at rest and in motion.

Data Centric Security implementations have been developed through work in Testbed-15 and Testbed 16. Initially using NATO STANAG 4774 and 4778 XML based standards to encrypt and sign geospatial feature data into secure containers for storage and transport in Testbed-15, OGC expanded into JSON based structures during Testbed-16, along with adding Key Management. While not being able to prototype an exact user scenario due to proprietary corporate or National Security interests, we have been able to show what such scenarios might look like and how DCS can add in the securing and proper sharing of geospatial information. During the previous Testbeds, only *OGC API-Features - Part 1: Core* was exercised. In Testbed-17, additional OGC APIs (family) are now in focus.

2.4.1. Problem Statement and Research Questions

This Work Item shall investigate and prototype techniques to protect resources in a globally distributed environment ranging from federated clouds down to the smallest IoT device. The work shall integrate previous efforts executed in the parallel tasks *Federated Security* and *Data Centric Security* into a single common security infrastructure. It shall further mature the federated

infrastructure required to support security controls regardless of where a resource resides. It shall mature the data centric security technologies and integrate them into the federated security infrastructure.

DCS has been demonstrated in the past with vector data exclusively. Now, the challenge is to apply DCS concepts to any type of binary data. The goal is to analyse DCS in the context of OGC APIs that deliver binary resource representations with confidentiality and integrity built in. The client that has been used in previous Testbeds does not provide full support for all emerging OGC APIs. Testbed-17 shall develop such a client or the necessary extensions to existing clients respectively. These clients shall support at least OGC API - Maps/Tiles/Coverages and GeoPackage. Further on, it shall be explored how DCS can be achieved using different formats such as the Trusted Data Format (TDF) instead of STANAG.

2.4.2. Aim

Integrate the Federated Security and Data Centric Security efforts into a single framework and advance the maturity of that framework. Demonstrate Data Centric Security for various binary data objects that are served by (emerging) OGC APIs with a set of client-server implementations. Ideally, a wide variety of data objects is supported, such as:

- Maps
- Tiles
- Coverages
- GeoPackage
- GeoTIFF
- GMLJP2

The list of exercised OGC APIs should not be limited to the corresponding APIs such as OGC API – Maps, - Tiles, or -Coverages, but shall include additional APIs such as EDR ([OGC API - Environmental Data Retrieval](#)) or the emerging DAPA (OGC API - Data Access and Processing, see [Previous Work](#)).

2.4.3. Previous Work

- OGC 20-021, OGC Testbed-16: Data Centric Security Engineering Report
- OGC 20-027, OGC Testbed-16: Federated Security Engineering Report
- [OGC 19-016r1, Testbed-15: Data Centric Security](#)
- OGC 20-025, OGC Testbed-16: Data Access and Processing API Engineering Report

2.4.4. Scenario & Requirements

[Two scenarios](#) have been used previously in both Testbed-15 and -16 to advance understanding of DCS. The *Desktop/Client/Server scenario* assumes connectivity between servers and clients at all times and does not put any specific constraints on the implementation setup. The *Cellphone scenario* on the other side foresees intermittent connectivity and support for strong encryption, as keys need to be taken on a mission of unknown duration.

Testbed-17 shall look at the application of DCS within Federated Cloud Environments and demonstrate that a DCS package created in one federation still enforces proper security controls when used in another federation. Participants are welcome to suggest alternative use cases to better represent the work within Testbed-17. On a high level, the Testbed-17 scenario(s) shall address the following aspects:

Federated Security

1. Gateway-based Zero Trust: Mature the zero-trust federation work performed in Testbed-16.
2. KMS API: Develop a specification for a Key Management Service (KMS) API.
3. Demonstrate that the KMS API performs its intended function for cloud federation and provides support for DCS.

Data Centric Security

1. Prove Client and Server interaction across a range of OGC API family of (emerging) standards as described above.
2. Explore how DCS may be implemented within GeoPackage
3. Extend the existing DCS specifications to support the full suite of CRUD (create, read, update, delete) operations.
4. Complete identification and definition of DCS related resources. This includes every data structure needed to support DCS including keys, tokens, labels, and policies.
5. Extend the DCS packaging specifications to support all DCS related resources identified above.
6. Engineering Report should consider DCS Standardization and consider what a draft specification should look like. Any DCS should be applicable across the OGC API family of (emerging) standards.

2.4.5. Work Items & Deliverables

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

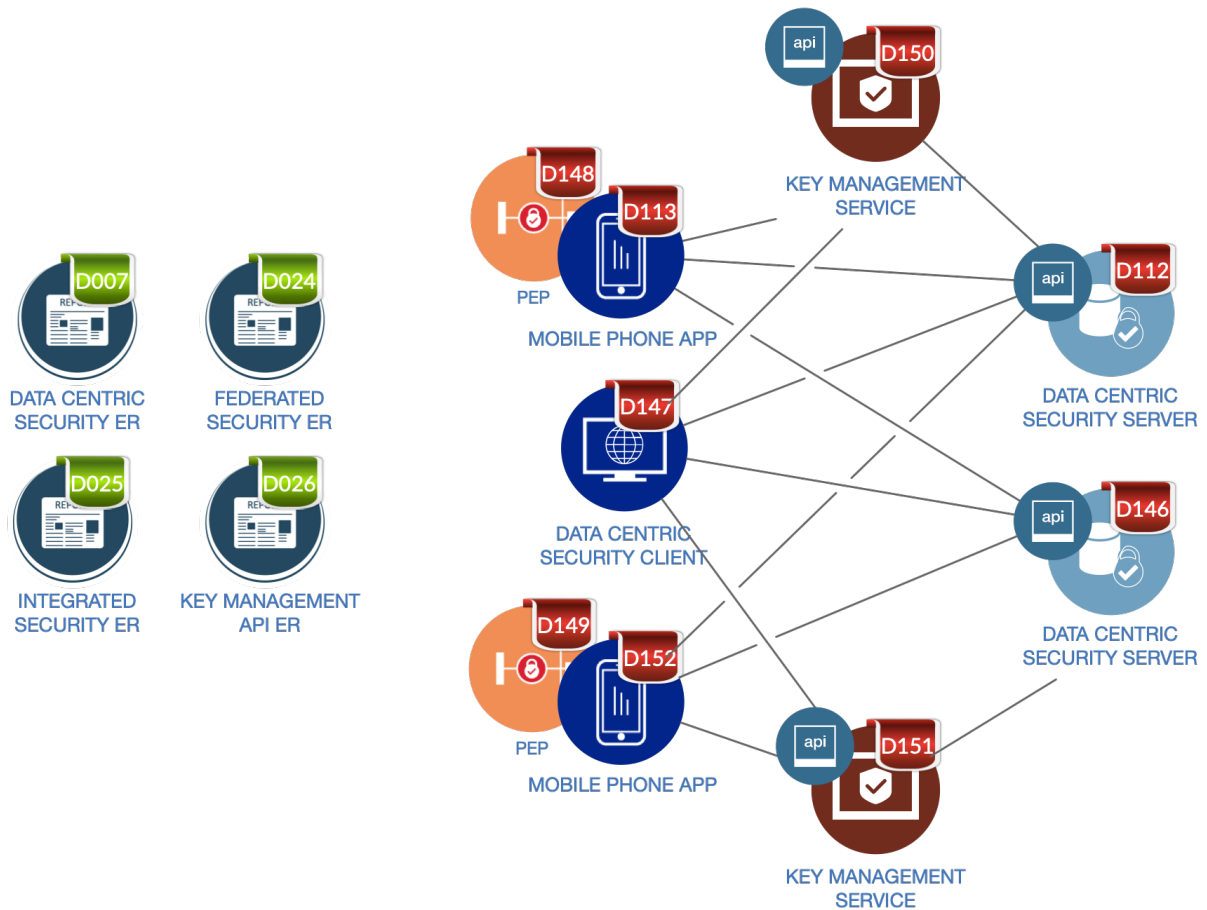


Figure 6. Data Centric Security Across OGC APIs task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D007 Data Centric Security ER** - Engineering Report capturing all results and experiences from this task. The Engineering Report shall be complemented with documentation for the OGC Standards Program activities that is best suited for direct consumption, i.e. use the appropriate templates and formats.
- **D024 Federated Security ER** – An Engineering Report that captures the work done to improve and validate the Federated Security suite of specifications.
- **D025 Integrated Security ER** – An Engineering Report that describes the integration of DCS into the Federated Security infrastructure and the results of all TIEs.
- **D026 Key Management Server API ER** – A draft specification for a key management server based on the integration of DCS and Federated Security.

Components

- **D112 Data Centric Security Server** - Server implementation with support for data centric security scenario. The server provider shall make use of the Key Management Services D148/149. The server shall support several OGC Web APIs and deliver geospatial data in different formats (features, maps, tiles, coverages, GeoPackages).

- **D146 Data Centric Security Server** - similar to D112
- **D113 Mobile Phone App** - Client implementation as an Android or iOS mobile phone application that supports the Cell Phone Data Centric Security scenario documented above. Client needs to support multiple OGC APIs and GeoPackage.
- **D152 Mobile Phone App** - similar to D113
- **D147 Data Centric Security Client** - Client implementation that supports the Data Centric Security desktop/client/server scenario. This delivery shall also include a build script for a working implementation in a Docker or VM environment.
- **D148 PEP** - Implementation of a Policy Enforcement Point (PEP) on cell-phone. That PEP shall support the cellphone scenario.
- **D149 PEP** - similar to D148
- **D150 Key Management Service** - Implementation of a key management server as required in both scenarios with support for the new Key Management Server API. This delivery also includes a script to build a working implementation of the KMS in a Docker or VM environment.
- **D151 Key Management Service** - similar to D150

2.5. Federated Cloud Analytics

The volume of Earth observation data has grown so large that moving it to a local machine for processing is no longer feasible. Instead of moving the data to local processing, the trend is now to store large repositories of Earth observation data on the cloud, on compute back-ends, and move the processing to the data.



To move processing to the data requires that a single analytic can be deployed on any cloud and that it can efficiently access the data stored on that cloud. This requires standards for data access that will work on any cloud.

2.5.1. Problem Statement and Research Questions

The OGC explored the "Applications-to-the-Data" problem in the Data Access and Processing thread of Testbed-16. This work was part of a larger, well-coordinated body of work which includes prior OGC initiatives such as previous Testbeds and in particular the OGC Earth Observations Applications Pilot.

Another Work Item performed under Testbed-16 explored the concept of Analysis Ready Data. Analysis Ready Data is "data that have been processed to a minimum set of requirements and

organized into a form that allows immediate analysis”. Software analytics need more than data if they are to do “immediate analysis”. They need to exchange information. This requires common formats (syntax) as well as a common understanding of what the data elements mean (semantics). Analysis Ready Data defines discipline-specific data standards which address both data syntax and semantics. Within limits, they provide the information exchange required when processing moves to the data.

This Work Item shall integrate Analysis Ready Data into the “processing to the data” workflows and evaluate:

1. What, if any, value does Analysis Ready Data provide to users of processing to the data technologies?
2. What adjustments need to be made to existing processing to the data technologies and standards to better incorporate Analysis Ready Data?

2.5.2. Aim

Mature the federated cloud analytics model to include “processing to the data” design patterns and content normalization based on Analysis Ready Data (the term content normalization combines the various normalization techniques applied to data). Validate that this addition simplifies the business logic underlying the Testbed-16 DAPA API and makes the results more useful for the analytic.

2.5.3. Previous Work

The following documents describe previous work relevant to this Work Item. This list is not comprehensive. All of these efforts built on previous work which is identified in the respective documents. Participants shall consider that prior art as well.

- [OGC Earth Observations Applications Pilot](#)
 - OGC 20-073: OGC Earth Observation Applications Pilot: Summary Engineering Report
 - OGC 20-045: OGC Earth Observation Applications Pilot: CRIM Engineering Report
 - OGC 20-043: OGC Earth Observation Applications Pilot: EOX-Sinergise-DLR-UVT-Terrasigna Engineering Report
 - OGC 20-037: OGC Earth Observation Applications Pilot: Pixalytics Engineering Report
 - OGC 20-038: OGC Earth Observation Applications Pilot: European Union Satellite Centre Engineering Report
 - OGC 20-034: OGC Earth Observation Applications Pilot: Spacebel Engineering Report
 - OGC 20-042: OGC Earth Observations Applications Pilot: Terradue Engineering Report
- [OGC Testbed-16](#)
 - OGC Testbed-16: Data Access and Processing Engineering Report (OGC 20-016)
 - OGC Testbed-16: Data Access and Processing API Engineering Report (OGC 20-025)
 - OGC Testbed-16: Analysis Ready Data Engineering Report (OGC 20-041)
 - OGC Testbed-14: Application Package Engineering Report (OGC 18-049)

- OGC Testbed-14: ADES and EMS Results and Best Practices Engineering Report (OGC 18-050)
- External Initiatives
 - OGC openEO: <https://openeo.org/>
 - OGC GeoAPI: <https://www.ogc.org/standards/geoapi>
 - OGC CEOS Analysis Ready Data (CARD4L): <http://ceos.org/ard/>

2.5.4. Scenario & Requirements

1. Review the prior work and devise a notional architecture for federated “processing to the data” systems. This architecture shall identify the technologies and standards required to realize the architecture and also define the roles they would play.
2. Add Analysis Ready Data to the notional architecture. Identify what changes and additions would be needed to fully integrate the whole.
3. Develop the plan for an Interoperability Experiment to test the notional architecture.

2.5.5. Work Items & Deliverables

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



Figure 7. Federated Cloud Analytics 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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D028 Federated Analytics Notional Architecture ER** – An Engineering Report which documents the notional architecture, describes the alternatives considered for integrating Analysis Ready Data, and describes the rationale for the alternative chosen.
- **D029 Federated Analytics Interoperability Experiment Plan ER** – An Engineering Report that defines the plan for an Interoperability Experiment to exercise the notional architecture, validate it where possible, and identify improvements needed.

Components

none

2.6. Model Driven Standards

Model-driven architecture (MDA) is a software design approach for the development of software

systems. The OGC Innovation Program has addressed model transformation for more than a decade.



This task shall provide a state-of-the-art report by analysing a set of existing tools with their capabilities and limits. The task shall give clear recommendations how model-driven design can be fully exploited in the context of rich data model and API design efforts.

2.6.1. Problem Statement and Research Questions

In recent years the OGC has seen the emergence of new encoding formats, data models, and service architectures. The result has been a proliferation of standards which should form a coherent body of work. The tools required to achieve and guarantee this coherence are lacking. This Work Item shall investigate application of Model Driven Architecture (MDA) tools and techniques to manage the development and maintenance of a portfolio of related standards.

2.6.2. Aim

This subtask shall evaluate and prototype MDA as both a framework and toolset for managing OGC standards.

2.6.3. Previous Work

Model Driven Architecture (MDA) was introduced by the Object Management Group (OMG) in 2001. MDA techniques have been used sporadically to develop OGC and ISO TC211 standards. The ShapeChange open source toolset (<https://shapechange.net/>) has been instrumental to those efforts. Participants in this Work Item shall build on this body of work. Efforts of particular relevance include:

1. UGAS initiatives
2. Testbed-16 OpenAPI Work Item
3. CityGML 3.0

Engineering Reports documenting OGC ShapeChange initiatives can be found at <https://shapechange.net/about/background-documents/>

2.6.4. Scenario & Requirements

MDA defines two types of models: Platform Independent Models (PIM) and Platform-Specific Models (PSMs). A PIM provides the definition of a data and/or computing capability which is independent of the implementing technology. A PSM defines how the PIM capability is realized using a specific implementing technology. The corresponding OGC and ISO terms are:

Table 2. Mapping of terms for MDA and formats typically used in OGC contexts

MDA	OGC/ISO	Model Formats
Platform Independent Model (PIM)	Conceptual Model	UML
Platform-Specific Model (PSM)	Implementation Specification	XML Schema, JSON Schema, SQL, OWL/RDF, others.

Participants in this Work Item shall explore MDA to manage both service and data standards. Of particular interest are the following formats and OGC Standards or elements:

Table 3. Transformation Exercises

PIM	PSM
CityGML 3.0	XML, JSON, OWL/RDF
OWS Capabilities	JSON
OpenAPI	JSON
UML Data Dictionary	Text

Specific Sub-Work Items are:

1. Evaluate existing MDA tools potentially suitable for use with Geospatial standards. Identify tools with promise. Identify requirements for which there does not appear to be a suitable tool (gaps).
2. Develop UML models for use in exercising tools. Modify existing models if they are not suitable for the intended use. All participants in this task need to support the UML model development for the [transformation exercises listed above](#).
3. Exercise identified tools. Document their capabilities and limits. Make recommendations on the future use of each tool by the OGC. Those recommendations shall fall into one of three categories: don't use, adopt, extend.
4. Develop prototypes which explore solutions to identified gaps.

Develop a draft Best Practices document for representing an OGC Platform Independent Model in UML. This document provides guidance to the developers of OGC Conceptual Model Standards so that those models can be used to generate valid Implementation Specifications using MDA tools.

2.6.5. Work Items & Deliverables

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



Figure 8. Model Driven Standards 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 and support the development of the conceptual models as well as the Engineering Report(s) with their contributions.

Engineering Reports

- **D022 Model Driven Standards ER** - Engineering Report which captures the analysis performed, prototypes developed, conclusions, and recommendations for further work.
- **D023 OGC UML Modeling Best Practices** - A document that captures the best practices that should be used when developing Platform Independent Models for OGC standards.

Components

- **D143 MDA Tool** - Model Driven Architecture software tool capable of exercising UML model to platform specific model transformation for the work items identified in the *Transformation Exercises* table. The first goal is to analyze and document the capabilities and limits of the tool. Eventually, the tool shall be further extended to prototypically support the transformation exercises. Any prototype tools developed under this Work Item shall be delivered to the OGC as publicly releasable software. All implementation specifications generated using existing and prototype MDA capabilities shall be delivered in their platform specific target format. These are NOT intended to be draft versions of OGC standards. Rather they serve as evidence of the quality of MDA produced specifications.
- **D144 MDA Tool** - similar to D143.
- **D145 MDA Tool** - similar to D143.

2.7. COG & Zarr: Specification & Evaluation

A [Cloud Optimized GeoTIFF](#) (COG) is a regular GeoTIFF file, aimed at being hosted on a HTTP file server, with an internal organization that enables more efficient workflows on the cloud. It does this by leveraging the ability of clients issuing HTTP GET range requests to ask for just the parts of a file they need. The OGC GeoTIFF SWG has accepted COG as a new task that will be added to the charter soon in order to become an OGC Standard. The currently available [COG specification](#) is rather a starting point than a real specification. Thus, it is the goal of this task to develop a draft specification that can be further developed by the OGC Standards Program to an OGC Standard.



[Zarr](#) is a generic data container format for the storage of chunked, compressed, multi-dimensional arrays. The Zarr development began in 2016 in the genomics community as a storage library for the scientific Python ecosystem. It integrates closely with Python software packages Numpy, Xarray, and Dask. The current version of Zarr, [version 2](#), has been proposed as an OGC Community Standard (see the [Zarr Community Standard Work Item Justification](#)). The development of version 3 is in a very early phase right now and appreciates input from the geospatial community. The December OGC Member Meeting will decide about Zarr becoming a new work item within OGC. Expecting a positive vote, Testbed-17 shall explore both COG and Zarr, compare their capabilities, and evaluate their potential and path towards an OGC Common Object Container. For this reason, metadata needs to be aligned and support for modern Coordinate Reference Systems expressed in WKTCRS2 ensured.

2.7.1. Problem Statement and Research Questions

The COG specification is not fully developed. Zarr has been developed as a generic container and needs to be evaluated for its capabilities to store geospatial data. Other container formats, such as CIS, NetCDF, or HDF5 can often be used interchangeably. A common metadata model is missing. In this context, the following research questions shall be addressed:

- How does the COG specification get defined in a manner that fulfills all requirements set by the OGC Standards Program for an OGC Standard.
- What role can Zarr play in the OGC context?
- How does Zarr perform in relation to other generic container formats such as HDF5 or NetCDF?
- Is it possible to develop a common metadata model that allows understanding what is in a container and how it can be accessed?
- What role does STAC play in this context?
- How good do COG and Zarr perform with Well Known Text (WKT) representation of Coordinate Reference Systems in WKTCRS2?
- How well do COG and Zarr perform with modern CRS?

2.7.2. Aim

Develop a COG specification that complies with the requirements set by the OGC Standards Program. Develop a specification that can be directly considered by the GeoTIFF SWG to be put forward as an OGC Standard.

Compare COG with other solutions for multi-dimensional data in the cloud context with focus on

Zarr. Support the Zarr integration into the OGC as a Community Standard as much as possible and provide feedback to the Zarr community for the next version of the specification.

2.7.3. Previous Work

The current versions of COG and Zarr are documented online. OGC did investigate Common Object Container solutions some time back. This work may be revitalized in this context with the goal to align more than just the metadata structures.

- OGC 20-052, [Zarr Community Standard Justification](#)
- [Zarr Storage Specification version 2](#)
- OGC 17-004, [Common Object Container Draft Charter](#)
- [Common Object Container presentation](#)

2.7.4. Scenario & Requirements

This task shall develop the COG specification from its current version into a draft OGC specification that complies with the OGC Standards Program requirements for standards. The work shall take the current discussion within the OGC GeoTIFF SWG as a starting point, which includes examination of COG, BigTIFF and STAC for common metadata.

The COG spec shall then be evaluated as a container specification in the context of other containers with focus on Zarr. The list of additional container formats that should be part of this task includes NetCDF, HDF5, and OGC CIS.

Both the COG and Zarr implementations shall be demonstrated within a typical scenario. The scenario shall highlight commonalities and differences with respect to metadata, discovery, and understanding of container contents.

2.7.5. Work Items & Deliverables

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



Figure 9. COG & Zarr: Specification & Evaluation 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 and support the development of the Engineering Report(s) with their contributions. All participants will jointly

develop the use cases and demonstration scenarios.

Engineering Reports

- **D046 COG Specification ER** - An Engineering Report that delivers an initial draft specification of the emerging COG standard.
- **D047 COG/ZARR Evaluation ER** - Engineering Report that captures all results and lessons learned, describes any proposed architecture, identifies necessary standards and their roles, describes all developed components, reports on the results of all TIE activities, and provides an executive summary and a description of recommended future work items.

Components

- **D180 COG Implementation** - Software component for experimenting with COG. Given that COG shall be evaluated in the context described above, participants may need to provide additional components necessary to perform all evaluation work. The implementation may be satisfied by experimentation against existing tools and drivers.
- **D181 Zarr Implementation** - Software component for experimenting with Zarr. Given that COG shall be evaluated in the context described above, participants may need to provide additional components necessary to perform all evaluation work. The implementation may be satisfied by experimentation against existing drivers.

2.8. Attracting Developers: Lowering the entry hurdle for OGC Web API experiments

OGC standards for Web APIs currently focus on the server-side definition of API elements and interaction patterns. The standards themselves need to have a certain level of formality to ensure high levels of interoperability. Being meticulously defined to reduce room for interpretation, these standards are considered tedious to read. Web developers often favor learning-by-example approaches. Rather than direct examination of a standard, developers prefer to start with implementation guidelines, sample code, and best practice documentation.



To better meet these requirements by Web developers, this task will provide a set of example code for both server- and client-side software, installation scripts, and best practices on how to start with OGC APIs. The installation scripts serve a dual purpose. By providing scripts that illustrate the deployment and operation of API instances not only on local machines but in different cloud environments, it makes the often challenging mapping of software components to cloud infrastructure a smooth experience.

2.8.1. Problem Statement and Research Questions

It is hard to attract new developers to implement standards that are written in a specific language that meticulously defines all formal elements. Though being necessary to minimize room for interpretation and in consequence maximize interoperability between two components implementing the same standard, official standards are rarely a good read. Many developers favor a different approach, which starts with simple documentation and examples. From there, additional features are explored stepwise, with the actual standard often the last resource being consulted. This task shall accommodate for developers who prefer the starting-and-learning-by-example approach. This task shall serve as a starting point for further development and exploration. It shall deliver all knowledge necessary to develop, deploy, and execute standards-based Web APIs to Web developers. The task shall follow a "How-To" philosophy with lots of hands-on experiments, examples, and instructions, rather than lecturing about the standards.

2.8.2. Aim

This task shall lower the entry hurdle to OGC Web APIs by providing a set tools and guidelines on how to develop, deploy, and operate an OGC Web API-based service instance in a modern cloud environment. Three items are in focus:

1. A client software library that simplifies interaction with the service;
2. A service instance with sample code for Web API support, control unit that mediates between the front-end API and the backend data stores, and support for various cloud-native data backends
3. Installation and deployment scripts that illustrate the deployment of both the service instance as well as the data backends in various cloud environments

2.8.3. Previous Work

This task shall take the latest versions of OGC API-Features as well as the Environmental Data Retrieval (EDR) API into account. The EDR API is specified in the OGC API – Environmental Data Retrieval candidate standard.

2.8.4. Scenario & Requirements

OGC Web API instances are usually implemented as microservices with an standards-based front-end API and a custom-made back-end. The microservice core acts as a controller and implements a set of functions that translate between Web API calls and data backend. These translations can have any level of complexity. The figure below illustrates the envisioned Testbed-17 scenario. A client with a dedicated Web API interaction software library (1) interacts with a data service that exposes an OGC Web API (2) at the front-end and includes a cloud native data storage software library (3) at the backend that interacts with cloud native storage, e.g., cloud databases, cloud object storage, or existing OGC Web service instances such as WFS.

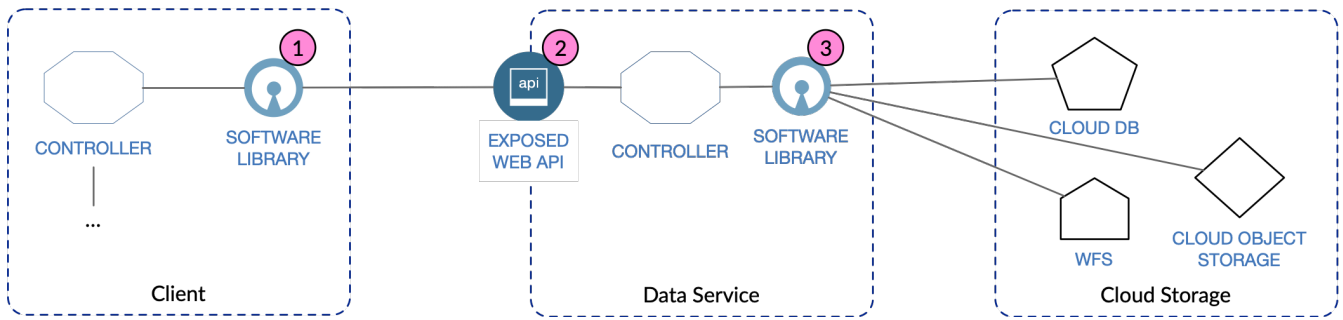


Figure 10. Testbed-17 scenario for API How-To's

Challenges often arise when a server instance shall be deployed in a cloud environment, where the developer is confronted with a number of additional features and design decisions. Therefore, all elements developed in this task can be deployed on infrastructure provided by various cloud providers. The necessary setup and installation scripts are explicit work items that will allow anyone to experiment with OGC API instances in the cloud. The figure below illustrates the envisioned deployment scenario. Each data service will be deployed twice in different cloud environments (A, B). The client application can be deployed anywhere.

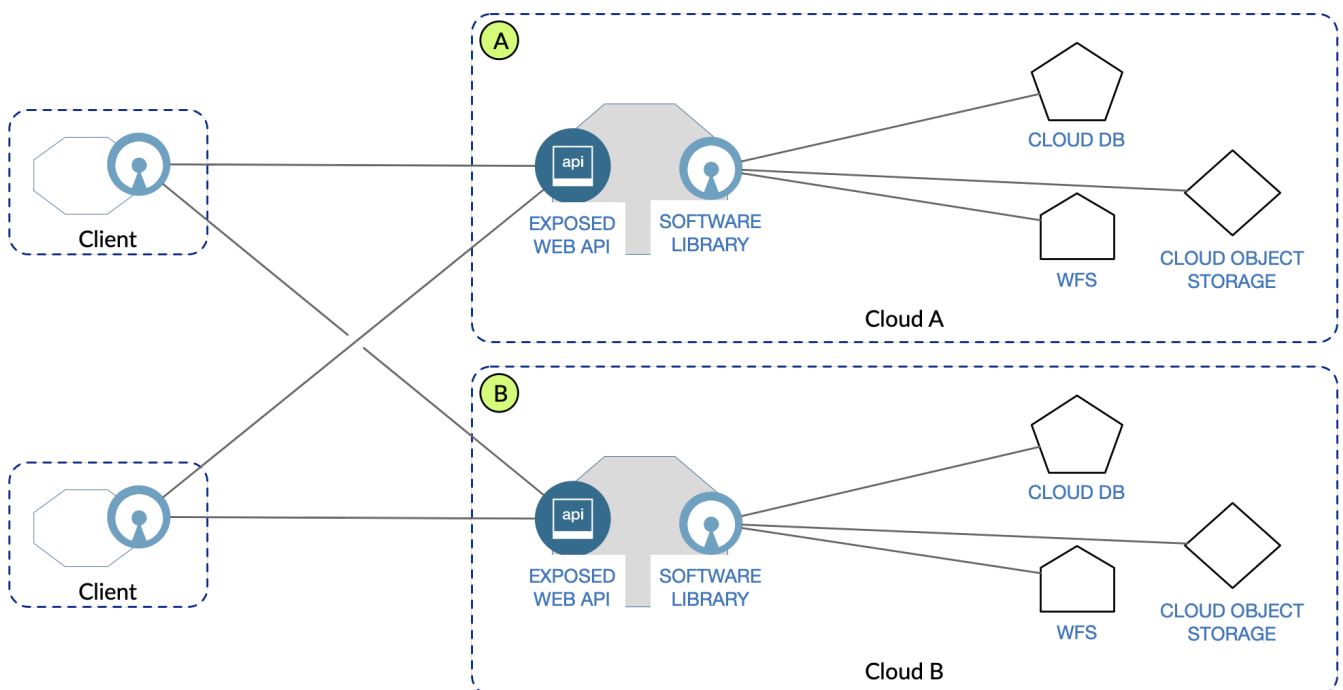


Figure 11. Deployment scenario

This task shall develop implementations of the OGC APIs -Features and -EDR (Environmental Data Retrieval) with backend access to data stored in cloud databases, cloud object storage, and existing OGC Web Feature Service (WFS) instances. Participants are welcome to reuse existing WFS instances, independently of their location.

The full implementation scenario is illustrated further below. It is emphasized that this task has the goal to deliver guiding material and best practices for server- and client-side API development and deployment. As such, all implementations shall support typical user scenarios and shall be documented in detail. The functionality and richness of the implementation as well as data products will be mutually agreed between participants, sponsors, and the IP-Team in the first phase of the testbed. No development needs to support every possible feature, but a reasonable subset as mutually agreed upon at the kick-off meeting. To test all components and guidance material, the work in this task shall be shared by participants acting as 'client developers', 'server developers', or

‘deployment testers and data providers’:

- **Client developers:** Organization O1 shall implement and deliver a client-side software library in Python for both OGC API -Features and -EDR. Organization O2 shall deliver a similar library in Javascript (alternatively Typescript). Both organizations shall demonstrate their library in a client application.
- **Server developers:** Organization O3 shall implement and deliver a server-side software library in Python for both OGC API -Features and -EDR. The library shall allow access and exploration of both OGC API -Feature and -EDR resources and support data hosted on cloud object storage, cloud databases, and OGC WFS instances (with constraint functionality). O3 shall provide a deployable instance together with deployment and execution documentation that allows third parties to test the deliverable(s) as a microservice in different cloud environments. Organization O4 shall deliver a similar library in Javascript (alternatively Typescript).
- **Deployment testers and data providers:** Organization O5 shall deploy a microservice based on deliverables provided by O3 and O4 in at least two different cloud environments (to the extent possible, as some data backends are cloud specific, others work across cloud environments or are fully cloud independent). In addition, O5 shall provide data in a database, cloud object storage, and as a Web Feature Service instance to test data access (reuse of existing WFS and/or Web Coverage Service (WCS) instances, even outside of the target cloud environment, is admitted). All data sources shall be available to all organizations in this task and freely accessible (i.e. object storage and WFS/WCS shall be fully accessible, cloud native databases as much as possible). Organization O6 shall perform similar to O5.

In summary, organizations O1 and O2 develop a client demo and client API implementation, organizations O3 and O4 develop a server demo and API implementation together with usage and deployment information and scripts, and organizations O5 and O6 test the deployment in various cloud environments with their own data (which is made available to others for testing as well). As such, organizations O5 and O6 act as external players that want to start exploring OGC Web APIs in action for their own data.

The focus of the task is on repeatable steps and tangible results. All organizations are required to document the usage of their components as detailed as necessary to allow external parties to start their own experiments with the provided implementations and deployments. Thus, focus is not on full support for each API and data backend detail, but on practical examples that showcase typical use cases of data discovery and access.

The data service will be deployed on various cloud environments using a set of scripts and user guides to make this process a smooth experience. For deployment, various options shall be supported. These include deployment and operation on a cloud-based virtual machine or deployment as a container-based solution (e.g. Docker, Kubernetes, or OpenShift).

All bidders shall define their preferred cloud environment as well as supported functionalities and deployment strategies in their proposals. Bidders may bid for multiple roles as described above (i.e., client developer, service developer, or deployment tester and data provider). Bidders doing so are required to allow selection of individual roles. The IP Team is working with various cloud providers to obtain free cloud resources.

DevOps and CI/CD

As the three roles described above have dependencies between each other, all participants in this task are required to follow a DevOps approach. DevOps as used here is about establishing a culture of continuous learning and experimentation so that all participants can start their work early and continuously refine and extend both functionality and quality during the Testbed-17 implementation phase. One of the most common blockers of the value stream from engineering to end-users is inter-team dependencies. If a team is unable to independently release their software, whether to internal or external customers, they will lose motivation and the ability to react to user feedback. Thus, sufficient team autonomy is one of the cornerstones of this task. This does not mean that teams will not be aligned. Rather, it requires a very clear definition of intermediate and final goals and responsibilities early in the process, so that all teams are able to pull in the same direction. To support this work, a Continuous Integration (CI) and Continuous Delivery/Deployment (CD) model will be developed at the kickoff meeting and refined as necessary. As a key principle of continuous delivery, all software and scripts shall be kept in a deployable state continuously.

Milestones

The following timeline broadly defines key milestones for this task. The milestones serve as orientation points to allow bidders to estimate required resources at the time of proposal development.

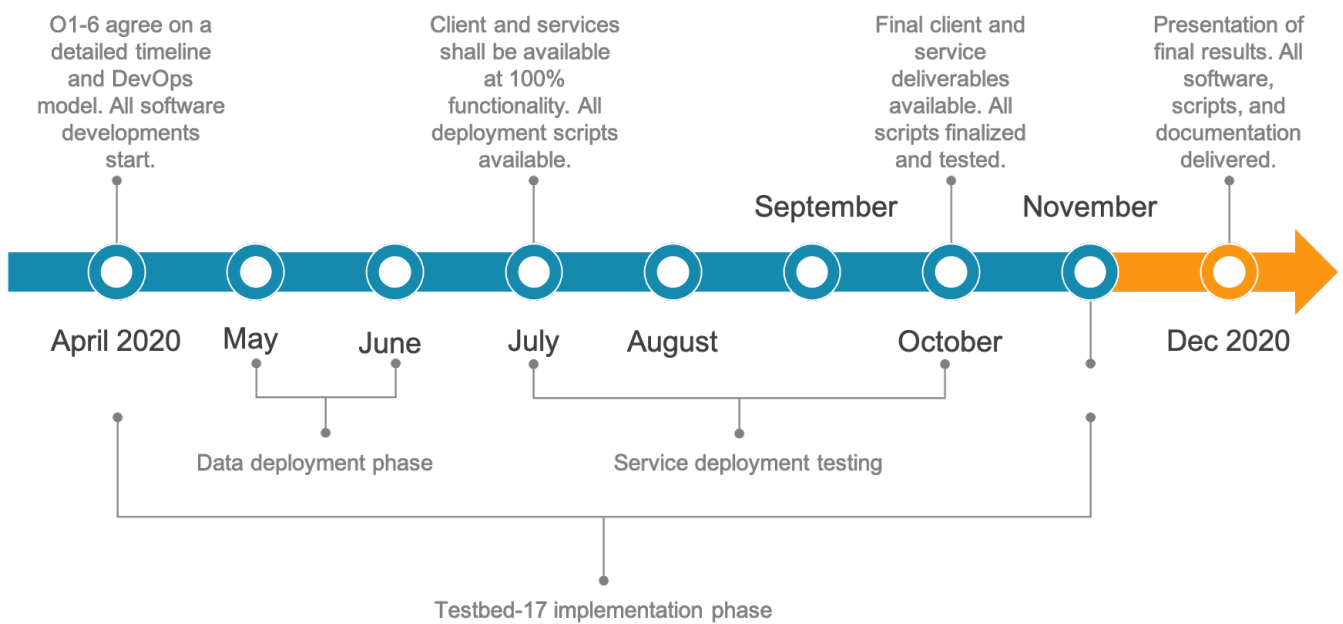


Figure 12. High level milestones for this task

2.8.5. Work Items & Deliverables

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

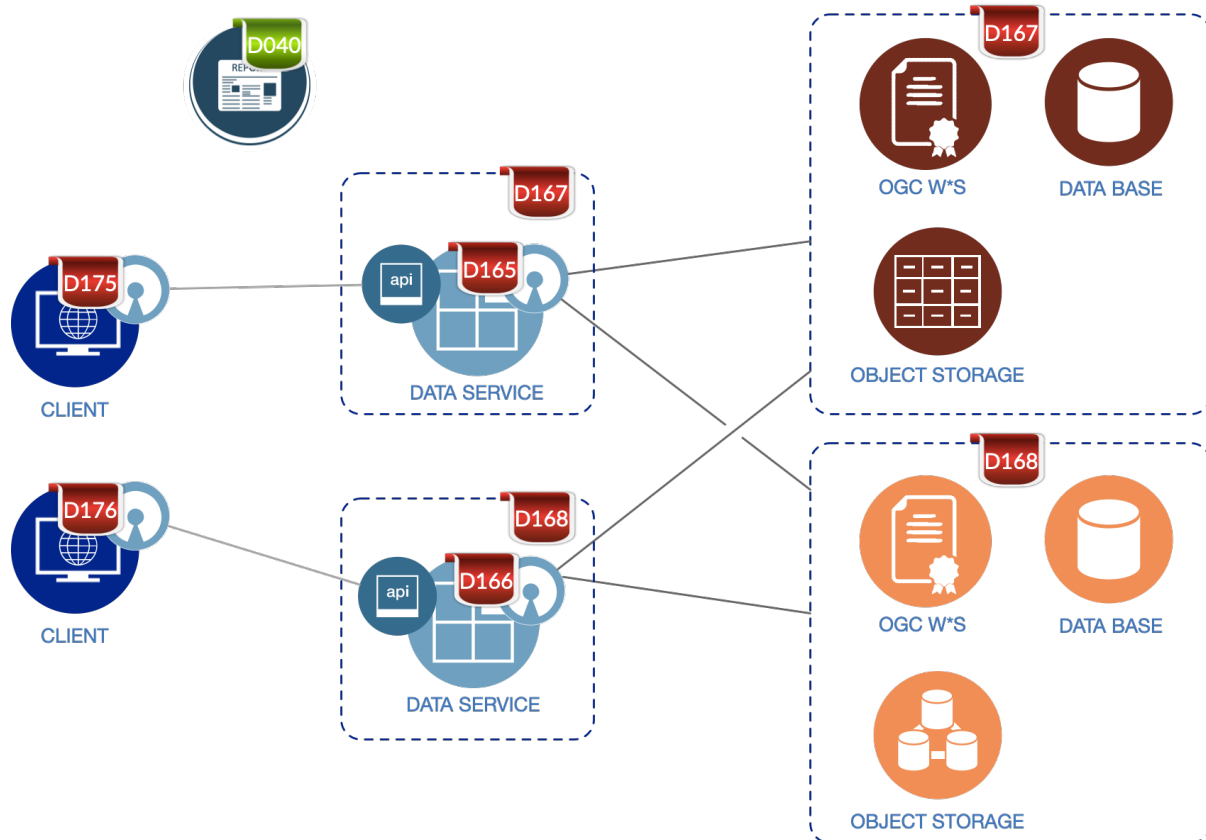


Figure 13. API How-To's task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions. For all work items and as emphasized above, the term “document” shall be interpreted more in the sense of “how-to”, i.e. more instructive, less documentary; components shall be implemented and documented accordingly.

Engineering Reports

- **D040 API Experiments ER** - Engineering Report providing all usage, deployment and installation instructions, guidelines and other documentation required to get started with OGC APIs within cloud environments. The report shall address design, development, and deployment of OGC Web APIs and provide best practices on how to use OGC APIs **with including** Jupyter Notebooks for use cases and scenarios to be defined at/after Testbed-17 Kick-off. The Engineering Report captures the proposed architecture, identifies the necessary standards, describes all developed components, reports on the results of all Technical Interoperability Experiments (TIE) activities, provides an executive summary and a description of recommended future work items

Components

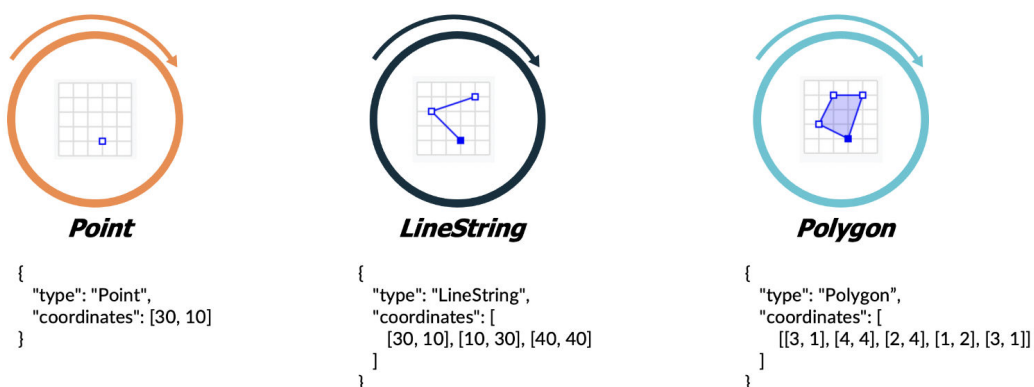
- **D175 API Experiments Client** - Client with software library as described above. The client library shall be sufficiently documented. Code examples and detailed usage documentation shall be provided to illustrate the usage of the API. Programming language: Python
- **D176 API Experiments Client** - Similar to D175. Programming language: Javascript (alternatively: Typescript)

- **D165 API Experiments Server** - Software component that can be deployed as a microservice as described above. The server front-end and back-end library shall be sufficiently documented. Code examples and detailed usage documentation shall be provided to illustrate the usage of the API. Programming language: Python
- **D166 API Experiments Server** - Similar to D166. Programming language: Javascript (alternatively: Typescript)
- **D167 Data Backend and Deployment** - Data being made available in a specific cloud environment, hosted in database(s), cloud object storage, and Web Feature Service. Additional data storages may be added. The provider of the data storages shall deploy the server component as developed, documented and delivered by D165 and D166 and communicate all experiences and lessons learned back to the server provider to help optimizing documentation and scripts.
- **D168 Data Backend and Deployment** - Similar to D167.

Important: As the goal of this task is to provide a robust and easy to understand starting point for Web developers to explore standards-based APIs, all software and installation scripts shall be delivered under a [permissive software license](#) (e.g. GNU All-permissive License or MIT License). The various work items may depend on commercial software as long as the essential functionality is properly documented and available as part of the open source software libraries.

2.9. OGC Features and Geometries JSON

In 2015 Internet Engineering Task Force (IETF) founded the Geographic JSON working Group. This group of developers released GeoJSON as [RFC 7946](#) in August 2016 as the latest specification. RFC 7946 replaced the GeoJSON 2008 specification. GeoJSON is based on JavaScript Object Notation (JSON), which is an open standard file format published in its current format in 2017 as [RFC 8259](#) by IETF. GeoJSON is an open standard format designed for representing geographical features along with their non-binary attributes.



In previous versions of GeoJSON the use of alternative CRS was specified. However, in the RFC 7946 the use of alternative CRS was removed because the use of different CRS had proved to have interoperability issues. The perceived use case for GeoJSON is as a processing software and was not expected to have access to CRS database or to have network access to CRS transformation parameters.

GeoJSON supports primitive geometry types (Point, LineString, Polygon), multipart geometries such as Multipoint, MultiLineString, MultiPolygon, as well as collections thereof (GeometryCollection).

Geometric objects with additional properties are Feature objects. Sets of features are contained by FeatureCollection objects. Though these different geometry types and the flexibility with features allow to address a wide variety of use cases, several communities assess GeoJSON as not fully qualified to serve their needs. The multipart geometries are often not needed and add unnecessary complexity to implementations, while insufficient support for coordinate reference systems (CRS) prevents the usage of GeoJSON in communities where geographic precision is an essential element. The GeoJSON specification does allow for the use of different CRS with prior agreement, but there is no way within the specification to advertise which CRS is being used. This work is seeking to eliminate any risk, as different CRS are used regularly within the Aviation or Defence and Intelligence communities.

Many applications benefit from using GeoJSON as a data format. If data integration from various sources plays a role, different CRS are often used. Then, the flexible handling and clear definition which CRS is being used becomes an essential feature.

The OGC Standards Program is currently building a new Standard Working Group (SWG) with the goal to eliminate these deficiencies. The [SWG charter](#) is currently under discussion. It is expected that the SWG will start its activities soon. The Testbed-17 task needs to be executed in close coordination with the SWG. Participants of the task shall be active members of the SWG. It is envisioned that this task will accelerate the work of the SWG by providing additional resources for model design and implementation. The provisioning of cost-share resources for implementations shall allow rapid prototyping and evaluation of the emerging extended GeoJSON specification. At the same time, the tight cooperation between the SWG and this Innovation Program activity ensures that all use-cases defined in Testbed-17 are sufficiently considered.

2.9.1. Problem Statement and Research Questions

With JSON often being favored over XML these days, the demand for GeoJSON with support for coordinate reference systems (CRS) is growing. Given that GeoJSON in its current form does not provide sufficient CRS support, Testbed-17 shall investigate the development of a CRS extension or within a profile of GeoJSON that could take the form of an OGC-version of GeoJSON.

Further on, the definition of lightweight profiles of GeoJSON shall be possible to allow communities to constrain allowed geometry types as well as feature attributes and values in a machine-readable form. This part shall focus on the aviation domain, where participants shall investigate challenges associated with the implementation of GeoJSON in emerging aviation, defense, and other applicable use-cases (including, but not limited to Unmanned Aerial Systems (UAS) / UAS Traffic Management (UTM)). Common needs across multiple use-cases shall be highlighted before developing a set of draft recommendations for core enhancements or within a profile of GeoJSON to address those common needs. In addition, it shall be explored what is possible in terms of extensibility and work with other domain-specific standards organizations to develop and test a set of extensions off the core in order to accommodate additional unique implementation requirements and reduce adoption risk / interoperability issues.

2.9.2. Aim

Investigate and define a solution that enables GeoJSON to support different Coordinate Reference Systems (CRS) as an extension or within a profile of GeoJSON. Further on, allow for communities to build and formally define profiles of the fully CRS-enabled GeoJSON with limited sets of supported

geometry types and with clear constraints for feature type definitions.

2.9.3. Previous Work

- <https://geojson.org/>
- <https://tools.ietf.org/html/rfc7946>
- [OGC Features and Geometries JSON Standards Working Group Charter](#)

2.9.4. Scenario & Requirements

The scenario requires the ability to use data from many different sources. Data may include material from different Earth Observation sources as well as aviation data (e.g., from UAVs), which may be using different CRSs, provided as GeoJSON metadata. The scenario requires CRS-enabled data to be loaded in offline data containers, as internet connectivity is only available intermittently. The scenario shall evaluate aviation specific requirements such as geometry constraints and profiles. The scenario may be broken into a number of sub-scenarios, but in total shall support the following requirements:

- Client and server implementation of GeoJSON, which allows different CRSs to be defined.
- Support usage of Features and Geometries JSON in offline containers
- Support profiles of the Features and Geometries JSON with reduced geometry options and value constraints for resource properties (aviation scenario).

This work is seeking to deliver and define a specification to GeoJSON that allows different CRSs to be defined and profiles to be generated that reduce the possible serialization options (e.g., by limiting the number of allowed geometry types). This work should include client and server implementations across the OGC API family of (emerging) standards.

2.9.5. Work Items & Deliverables

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

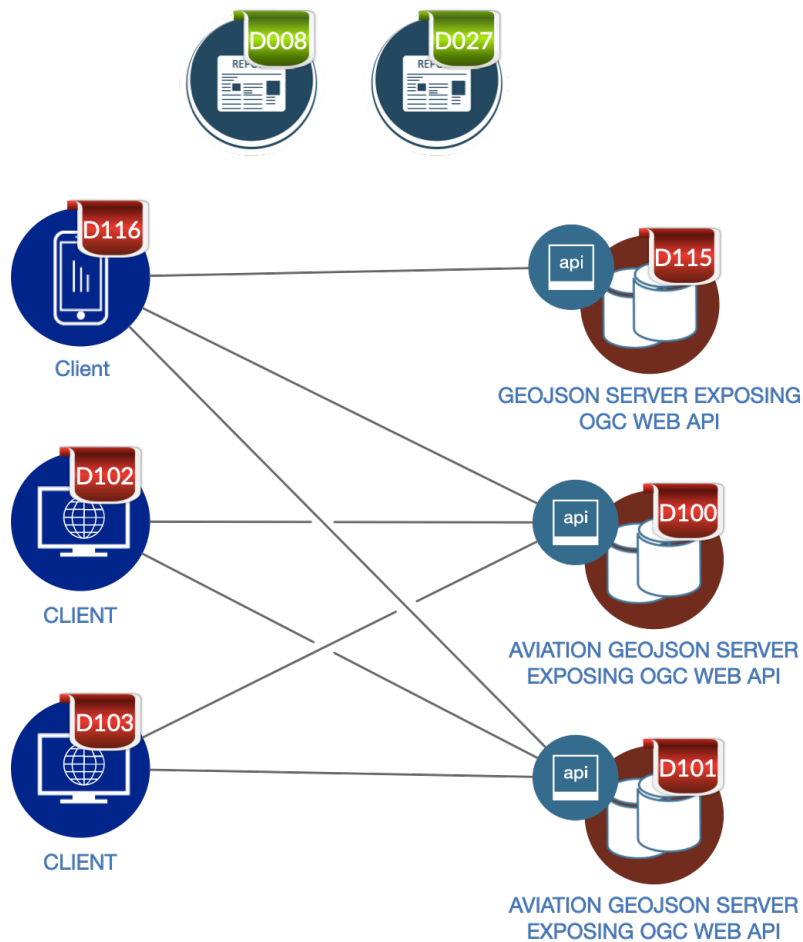


Figure 14. OGC Features and Geometries JSON task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D008 Features and Geometries JSON Engineering Report** - Engineering Report capturing all results and experiences from this task, including material for direct consumption by the emerging [OGC Features and Geometries JSON Standards Working Group](#). These can be delivered in the form of direct contributions to the SWG work, i.e. in the form of "pull requests" against the SWG Github repository. The Engineering Report shall summarize all results including experiences and lessons learned from the implementations in this task.
- **D027 Features and Geometries JSON CRS Analysis of Alternatives Engineering Report** - Engineering Report that documents the alternatives examined to generate the proposed extension and the rationale for the selected approach as defined in D008.

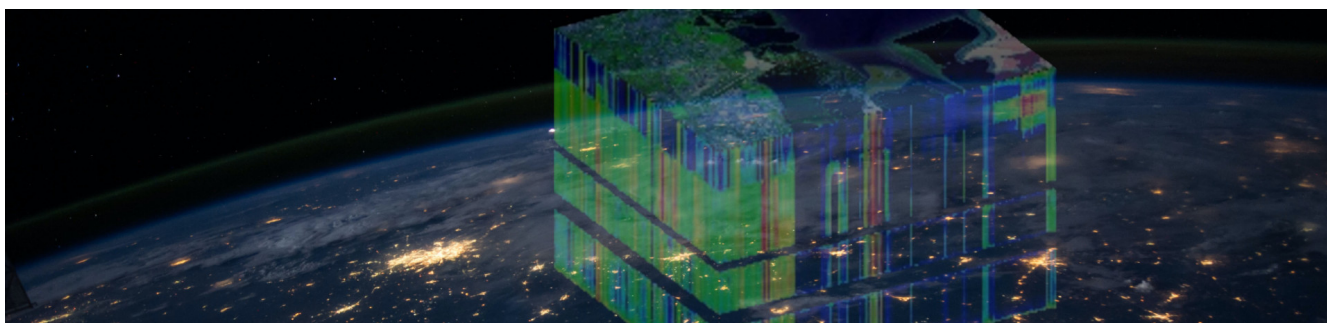
Components

- **D100 Features and Geometries JSON Server for Aviation** - A server instance that fully supports the modified and profiled GeoJSON within an aviation scenario. Ideally, the service supports the new Aviation API as defined in section [Web API for Aviation](#). Alternatively, the server shall support OGC API-Features.
- **D101 Features and Geometries JSON Server for Aviation** - similar to D100.

- **D102 Features and Geometries JSON Client for Aviation** - A client application that fully supports the modified and profiled GeoJSON within an aviation scenario. Ideally, the client supports the new Aviation API as defined in section [Web API for Aviation](#). Alternatively, the client shall support OGC API-Features.
- **D103 Features and Geometries JSON Client for Aviation** - similar to D102.
- **D115 Features and Geometries JSON Server** - OGC APIs instance with support for Features and Geometries JSON resources.
- **D116 Features and Geometries JSON Client** - Desktop, Web, or mobile client application that supports the Features and Geometries JSON server instances.

2.10. Geo Data Cubes

The Earth Observation area is facing a major transformation, with the emergence of big EO data host services and big EO data analytic services beyond the traditional providers. As is perhaps inevitable in situations of rapid change, there are currently many self-organized initiatives. Testbed-17 will identify, develop, and promote emergent consensus: towards a neutral definition of Geo Data Cubes (GDCs) and a Web API for convenient, interoperable Geo Data Cube access and exploitation. Through the API, any GDC that supports the standardized interface can be accessed, allowing applications and users to make use of GDC data in an efficient, consistent way. An API can also allow users to understand capabilities of a GDC, for example its spatial resolution or temporal dimensions and granularity.



Many organizations have invested significant resources in Geospatial Data Cubes: infrastructure that supports storage and use of multidimensional geospatial data in a structured way. While advances have been made to develop GDCs to support specific needs, challenges remain with enabling wide access to them, limiting their ability to support interoperability.

2.10.1. Problem Statement and Research Questions

Testbed-17 shall explore development of an API that is able to support access to and evaluation of GDCs in a uniform, standardized way from multiple environments (e.g., other GDCs, platforms, various file systems, etc.). The support of discovery, access, sharing, and use of GDC data should enable workflows involving distributed computing resources and algorithms. This effort will specifically explore the use of OGC APIs as building blocks to support GDC interoperability and evaluation. It is expected that outcomes of the project will support multiple uses within regional to international spatial data infrastructures. Therefore, additional objectives from this project include:

- Gain immediate insight into the potential of existing OGC APIs for supporting GDC

interoperability and evaluation.

- Identify potential improvements to or the creation of new geospatial standards to better support GDC interoperability and evaluation.
- Identify strategies to improve the ability of spatial data infrastructures to support new geospatial technologies such as GDCs.
- Provide guidance on how governmental organizations can leverage GDCs to help meet national and international priorities.

The following overarching research questions shall further help guide the work for the GDC task and shall be discussed during the Testbed-17 implementation phase:

- Are there emergent commonalities from Geo Data Cube implementations that can be agreed as open consensus standards?
- Are there specific interoperability requirements for GDCs that differ from those for other geospatial data types and systems?
- How can design of a GDC API support interaction with other standards-based API frameworks (e.g., supporting OGC API-Processes to complete analysis using GDC data)?
- How can a GDC API's design ensure timely performance under high and low data volume use cases?
- How can a GDC API support interoperability with other structured and unstructured geospatial information storage infrastructure (e.g., data lakes)?
- How can a GDC API support use, discovery, and interoperability of multiple standardized data formats (e.g., Spatial Temporal Asset Catalog (STAC), Cloud Optimized GeoTiff (COG), OGC Web Services (W*S), OGC APIs)?
- How can a GDC API support use, discovery, and interoperability of existing systems (e.g. registries, catalogues)?
- How can a GDC API support use of advanced technology tools, such as machine learning?
- What opportunities exist to link/leverage efforts from other related interoperability efforts (e.g. outcomes from the [OGC's Earth Observation Applications Pilot](#) and/or [OGC Testbed-16's Data Access and Processing API for Geospatial Data](#))?
- How will a GDC API be able to support multi-jurisdiction requirements in order to support GDC interoperability between organizations and governments?
- Can foundational principles be defined to support Geo Data Cube standards?

The activities executed in Testbed-17 will be part of the larger OGC innovation and standardization activities umbrella. As such, they will inform future OGC activities. All work on OGC Web APIs shall be executed in close cooperation between the Innovation Program and the Standards Program to ensure consistent and coordinated development of all Web APIs and Web API building blocks. It is expected that results from this task will form the basis for future initiatives to fully enable interoperable GDCs through an OGC Implementation Standard. All developments shall be based on existing OGC Web API building blocks as much as possible.

2.10.2. Aim

Develop an OGC API - Geospatial Data Cubes draft specification with the goal to standardize use, discovery, and access to GDCs. This task shall demonstrate and evaluate the capabilities of GDCs in various use-cases. The API shall be accessed by consumer clients as well as machine learning models to evaluate GDC usability in the context of machine learning.

2.10.3. Previous Work

- Geospatial Coverages Data Cube Community Practice ([OGC 18-095r7](#), Approved Sep. 2020)
- OGC Testbed-16: Data Access and Processing API Engineering Report (OGC 20-025)
- [OGC Earth Observation Applications Pilot](#) (several [detailed ERs](#) with [Summary ER](#))
- [OGC Web APIs](#)
- OGC Testbed-16: Analysis Ready Data Engineering Report (OGC 20-041)
- OGC Testbed-16: Machine Learning Engineering Report(OGC 20-015)
- OGC Testbed-16: Machine Learning Training Data Engineering Report(OGC 20-018)
- The [OGC Geospatial Coverage Abstract Standard](#), also now issued by ISO, defines a data representation to increase interoperability of raster encoding formats. The Coverages Abstract Spec has a solid research basis reflected initially in an NCGIA report and later publications:
 - [The Ontology of Fields](#) - Peuquet, Smith, and Brogaard, NCGIA, 1998
 - [Towards a general theory of geographic representation in GIS](#) - Goodchild, Yuan & Cova 2007
 - [Towards standardised dynamic and 3D field representations](#) - H. Ledoux, 2008
 - The Geospatial Coverages Standard was a basis for OGC Web Coverage Services (WCS) and OGC Coverage Implementation Schema (CIS). WCS provided a web services implementation that is used operationally in Geospatial Coverage Data Cube implementations. Now what is needed is to step back from WCS and look at modeling spatial information as fields.

2.10.4. Scenario & Requirements

The task shall operate in the context of terrestrial and marine elevation, and forestry data scenarios. At a high level, the current scenario envisions a client application to request result sets of Digital Elevation Model (DEM) operations such as elevation profile, viewshed analysis, and computing of elevation statistics (min/max/mean values for elevation, slope, aspect) inside a user-defined polygon. In addition, a forestry scenario shall be developed that illustrates usage of GDCs in the forestry domain, specifically to enable access, processing, fusion, and extraction of diverse forestry datasets. Detailed scenarios will be developed during the first stage of the Testbed-17 implementation phase. Participants are invited to suggest appropriate use-cases. Ideally, these include Canadian territory and waters.

To support project completion, the following data sets will be provided (participants are invited to add additional data sets):

- Access to the GeoBase terrestrial elevation cloud instance hosted on Amazon Web Services.
- Access to marine elevation datasets hosted on Open Maps (e.g., Canadian Hydrographic Service

[Non-Navigational bathymetric data](#)) and on other environments.

- Access to forestry datasets including [Wetlands 84-16](#) data from the National Forest Information System.

All implementations of the proposed API shall be designed to meet needs and requirements for GDC elevation information interoperability between the land and marine domains. The key outcome from this task shall be the creation of a draft GDC API specification for using existing OGC APIs (as-is or through extensions), or for developing a new OGC API to enable GDC interoperability and evaluation. The draft specification shall be sufficiently complete such as to enable timely integration with the OGC's Standards Program for further development of the API to an OGC standard.

This task shall also highlight how a GDC API can support use and interoperability of multiple standardized geospatial data formats such as STAC, COG, W*S, and OGC APIs and systems such as existing registries and catalogues.

Cloud-Based Access, Analytics, and Processing, and Relationships to Other File Systems

Development of the GDC API shall also consider how to support access to GDCs through cloud-based infrastructure. This task shall also explore cross-cloud interoperability of GDCs through the API in order to ensure GDC information can be shared between clouds, and that GDC access can be maintained if changes to a GDC cloud-service are required (e.g. changing to a new cloud provider).

This cloud-oriented task shall also investigate implementing analysis and processing capabilities for GDC data through the API framework. Potential linkages with recent, similar efforts completed as part of the [OGC's Earth Observation Applications Pilot](#) and [OGC Testbed-16's Data Access and Processing API for Geospatial Data](#) shall be explored and leveraged to the greatest extent possible to maximize API use and interoperability.

While cloud-based interoperability is a key consideration for this effort, the work shall also consider how the GDC API can support interoperability with other forms of infrastructure (e.g. data lakes, workstation-based file systems, high performance computing systems, etc.). Demonstrating an ability to link with non-GDC systems represents an important aspect of ensuring GDCs can make use of and support different forms of infrastructure.

Overall, the following major work items have been identified:

1. Develop an OGC API to support GDC interoperability. This API shall use either
 - a. a current OGC API structure,
 - b. an extension to an existing OGC API, or
 - c. propose a new OGC API to meet GDC interoperability requirements.
2. Demonstrate access to and analysis/processing of GDC information through the proposed API via a cloud environment. The draft API shall be available at all stages to sponsors to independently test the functionality of the API itself.
3. Demonstrate that the API enables evaluation of a GDCs capabilities, including its dimensions and granularity from a spatial and temporal perspective.
4. Demonstrate that the API enables use, discovery, and interoperability of diverse standardized

geospatial data formats, including STAC, COG, W*S, and OGC APIs and systems such as registries and catalogues.

5. Demonstrate that the proposed API solution enables cross-cloud interoperability for GDCs, and that interoperability can be maintained if the GDC cloud environment experienced significant change (e.g. transitioning to a different cloud provider).
6. Demonstrate that the proposed API solution is able to meet the needs of a use case workflow in the context of GDC interoperability for terrestrial and marine elevation, and forestry information. If all sponsor's needs cannot be met through the solution, provide suggestions for future work that will enable requirements to be met.
7. Demonstrate that the proposed API supports interoperability with GDCs in multiple contexts (e.g., between GDCs, between GDCs and workstation-based file systems, etc.).
8. Demonstrate that the proposed API is able to support interoperability between terrestrial and marine elevation information (e.g., marine elevation data from one GDC and terrestrial elevation data from a different GDC can be accessed and integrated through the API).
9. Demonstrate that the proposed API is able to support interoperability between different types of datasets (e.g., forestry data and elevation data). As an example, in the context of forestry information, the proposed API should be able to:
 - a. Process and extract information from the forestry imagery data cubes using state-of-the art image compositing techniques.
 - b. Address the interoperability of different forestry data formats (such as LiDAR point cloud, images, etc.) to facilitate the fusion of these data.
 - c. Predict changes of forestry attributes and provide meaningful insights through temporal-spatial data products
10. Demonstrate that the proposed API can enable advanced technologies (e.g. machine learning) to leverage GDC environments and information.
11. Develop an implementation plan allowing the proposed API approach to be incorporated into the OGC standards program.

With respect to machine learning (ML), this task shall explore and demonstrate how ML can interact with a data cube through the API and how this may differ from how a human would interact with the API. This could lead to design considerations (e.g., how can the API be optimized to best enable ML and human interactions with a data cube?). The use case may be simple and ideally relates to the elevation or forestry use cases. As an example, it could be explored how ML could access forestry datasets to complete analysis related to harvest planning.

2.10.5. Work Items & Deliverables

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

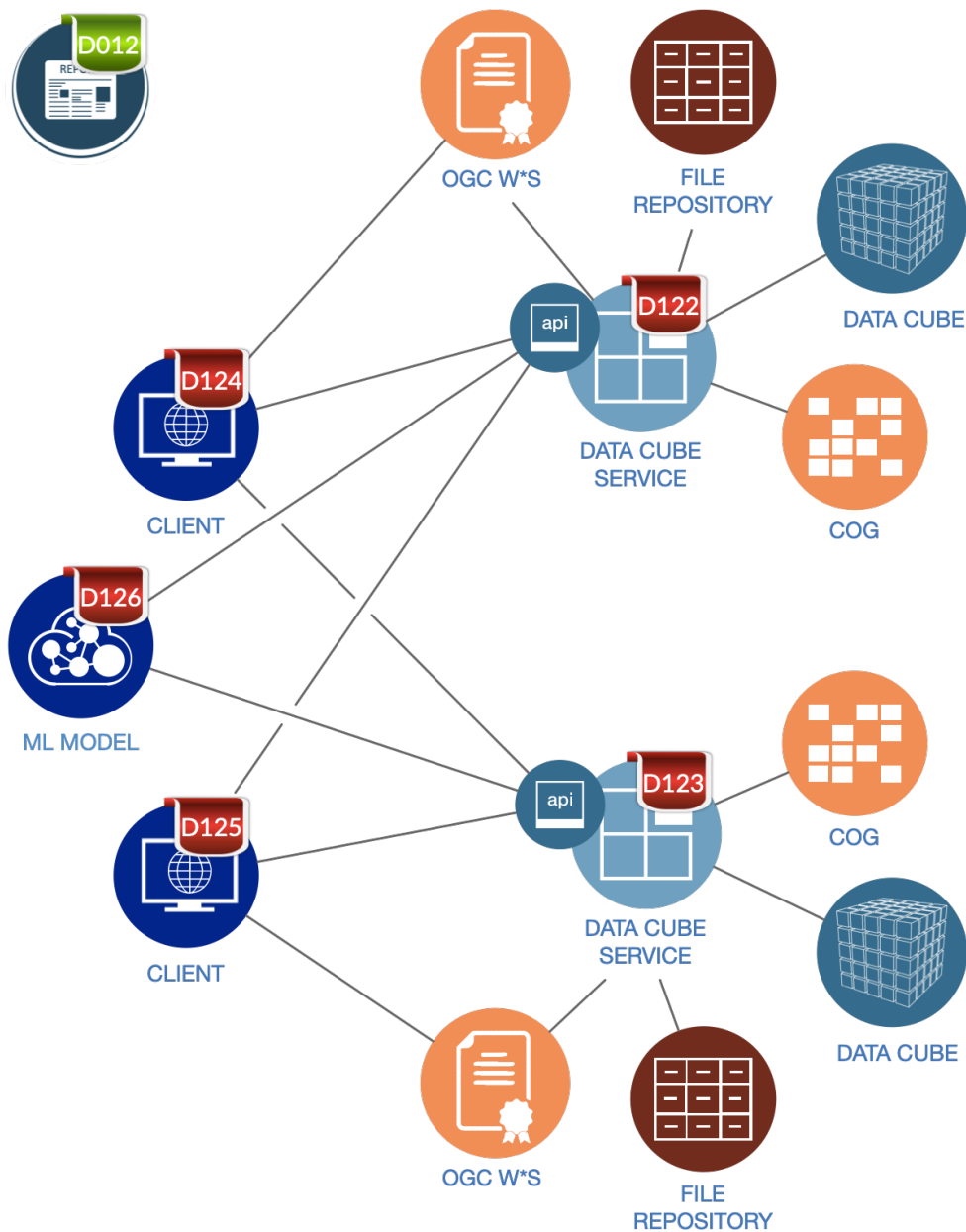


Figure 15. Geo Data Cubes task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D012 Geo Data Cube API ER** - Engineering Report specifying the Geo Data Cube API. The API shall be defined according to the requirements set by the OGC Standards program for OGC Web APIs. The Engineering Report shall further capture the analysis performed, prototypes developed, conclusions, and recommendations for further work.

Components

- **D122 Geo Data Cube Service** - Server implementation with support for OGC GDC API. As a sub-component, the server-side implementation shall deliver a driver / library that encapsulates the actual GDC interface functionality. That driver / library shall be delivered under a permissive software license (e.g. GNU All-permissive License or MIT License) that allows integration into

existing OGC tools and services. The service shall demonstrate access to various data sources and existing OGC web service instances. Ideally, the service is deployed twice. Once in a cloud environment, and second on a local physical machine. Both instances shall demonstrate access to a variety of data sources as illustrated in the figure above. Service providers shall support access to the data sets defined above as well as additional data as needed for the final demonstration scenario.

- **D123 Geo Data Cube Service** - similar to D122
- **D124 Geo Data Cube API Client** - A client application with OGC GDC API support and capable of demonstrating GDC interoperability capabilities (e.g. access, analysis, and processing through a cloud-enabled GDC) in the context of the elevation data scenario.
- **D125 Geo Data Cube API Client** - similar to D124
- **D126 ML Model** – Machine Learning model with GDC API support demonstrating how GDCs can be used by Machine Learning models.

2.11. Compliance Interoperability & Testing Evaluation (CITE)

The OGC Compliance Program is a certification process that ensures organizations' solutions are compliant with OGC Standards. It is a universal credential that allows agencies, industry, and academia to better integrate their solutions. OGC compliance provides confidence that a product will seamlessly integrate with other compliant solutions regardless of the vendor that created them.



As such, the OGC Compliance Program plays an essential role in the context of standards-based interoperability, as it allows evaluating the level of compliance of any product to the implemented standard(s).

2.11.1. Problem Statement and Research Questions

Traditionally, OGC CITE uses *TEAM Engine* (Test, Evaluation, And Measurement Engine), a Java-based application for testing web services and other information resources. It executes test suites developed using the TestNG framework, OGC Compliance Test Language (CTL) scripts, and possibly other JVM-friendly languages. TEAM Engine can be used to test almost any type of service or information resource. It is the official test harness used by the Open Geospatial Consortium's (OGC) compliance program.

With the changing landscape moving away from the Service Oriented Architecture with its remote procedure calls over HTTP towards a modern, resource-based Web architecture, the question is if

TEAM Engine is still the best tool for the OGC Compliance Program. To test this aspect and to deliver an additional test suite, Testbed-17 will develop tests for OGC API-Processes. The tests will be implemented for TEAM Engine and for an additional, yet to be identified testing environment.

The task shall address the following research questions:

- What does a TEAM Engine test for OGC API-Processes look like?
- What alternative test environment(s) should be used in the future and why?
- How do tests look like for this new test environment?
- Is it possible to automatically generate tests or from the latest generation of OGC specifications? If it is possible, then what level of automatization is possible? Does a high level of automatization require a change to the standard format?

2.11.2. Aim

Develop a test suite for OGC API-Processes for TEAM Engine. Explore alternative test environments and provide recommendations for the way forward. Develop a test suite for OGC API-Processes with the recommended test environment and evaluate to which extent tests may be auto-generated from the standards directly.

2.11.3. Previous Work

- [OGC Compliance Testing](#)
- [TEAM Engine](#)
- [CITE](#)
- [Guidelines on Development on Tests](#)
- [GitHub repository of initial draft executable test suite of OGC API-Processes](#)

2.11.4. Scenario & Requirements

Develop a test suite for OGC API-Processes for TEAM Engine and an alternative test environment. Compare both broadly and provide recommendations for the future.

2.11.5. Work Items & Deliverables

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

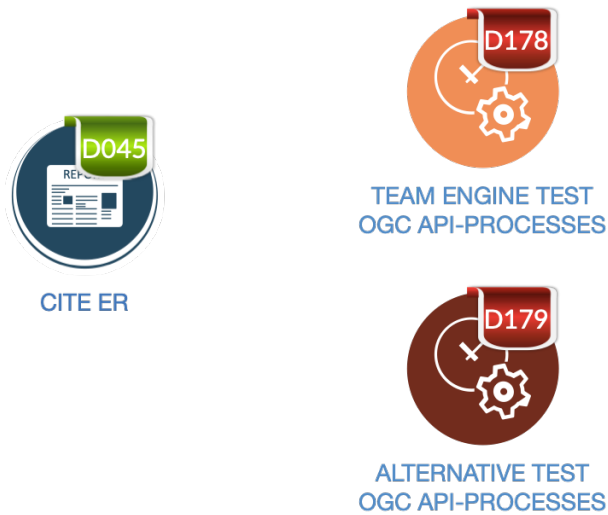


Figure 16. CITE task work items and deliverables

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 and support the development of the Engineering Report(s) with their contributions.

Engineering Reports

- **D045 CITE ER** - Engineering Report capturing all results and experiences from this task, including both test suites and their comparison.

Components

- **D178 TEAM Engine Test OGC API-Processes** - Test suite for OGC API-Processes to be executed on TEAM engine
- **D179 Alternative Test OGC API-Processes** - Alternative test environment with test suite for OGC API-Processes

Chapter 3. Deliverables Summary

The following tables summarize the full set of Initiative deliverables. Technical details can be found in section [Technical Architecture](#).

NOTE Please note that for several indicated work items, cost sharing funds may only be requested by entities from ESA member states, Canada, and Slovenia.

Please also note that not all work items were supported by sponsor funding at time of CFP publication. Negotiations with sponsors are ongoing, but there is no guarantee that every item will ultimately be funded.

IMPORTANT Bidders are invited to submit proposals on all items of interest under the assumption that funding will eventually become available. Timely notifications of changes in funding status will be provided in the updated [CFP Corrigenda Table](#).

Table 4. CFP Deliverables - Grouped by Task

Task	ID and Name (Bold: open only to ESA members, Canada, Slovenia)
<i>Data Centric Security Across OGC APIs and Federated Clouds</i>	<ul style="list-style-type: none"> • D007 Data Centric Security ER • D024 Federated Security ER • D025 Integrated Security ER • D026 Key Management Server API ER • D112, D146 Data Centric Security Server (2 instances) • D113, D152 Mobile Phone App (2 instances) • D147 Data Centric Security Client • D148, D149 PEP (2 instances) • D150, D151 Key Management Service (2 instances)
<i>OGC Features and Geometries JSON</i>	<ul style="list-style-type: none"> • D008 Features and Geometries JSON ER • D027 Features and Geometries JSON CRS Analysis of Alternatives ER • D100, D101 Features and Geometries JSON Server for Aviation (2 instances) • D102, D103 Features and Geometries JSON Client for Aviation (2 instances) • D115 Features and Geometries JSON Server • D116 Features and Geometries JSON Client

<p><i>Moving Features from Digital Motion Imagery</i></p>	<ul style="list-style-type: none"> • D020 Moving Features ER • D021 OGC API - Moving Features ER • D135, D136 Ingestion Service (2 instances) • D137, D138 Tracking Service (2 instances) • D139 Machine Analytics Client • D140 - Human Analytics Client • D141, D142 Storage Service (2 instances)
<p><i>Model Driven Standards</i></p>	<ul style="list-style-type: none"> • D022 Model Driven Standards ER • D023 OGC UML Modeling Best Practices • D143, D144, D145 MDA Tool (3 instances)
<p><i>Federated Cloud Analytics</i></p>	<ul style="list-style-type: none"> • D028 Federated Analytics Notional Architecture ER • D029 Federated Analytics Interoperability Experiment Plan ER
<p><i>Sensor Integration</i></p>	<ul style="list-style-type: none"> • D030 Sensor Integration Framework Assessment ER • D031 MASBUS Integration ER • D032 SIF Semantic Model ER • D153, D154 MASBUS Server (2 instances) • D155, D156 MASBUS Client (2 instances)
<p><i>Web API for Aviation</i></p>	<ul style="list-style-type: none"> • D002 Aviation API ER • D104, D105 Existing SWIM Service Exposing new Aviation API (2 instances) • D106, D107 Fusion Service Exposing new Aviation API (2 instances) • D108 End-user Client • D109 Aviation Client
<p><i>Attracting Developers: Lowering the entry hurdle for OGC Web API experiments</i></p>	<ul style="list-style-type: none"> • D040 API Experiments ER • D175 API Experiments Client (Python) [ESA members, Canada, Slovenia] • D176 API Experiments Client (Javascript or Typescript) [ESA members, Canada, Slovenia] • D165 API Experiments Server (Python) • D166 API Experiments Server (Javascript or Typescript) • D167, D168 Data Backend and Deployment (2 instances)

<i>Geo Data Cubes</i>	<ul style="list-style-type: none"> • D012 Geo Data Cube API ER • D122, D123 Geo Data Cube Service (2 instances) • D124, D125 Geo Data Cube API Client (2 instances) • D126 ML Model
<i>COG & Zarr: Specification & Evaluation</i>	<ul style="list-style-type: none"> • D046 COG Specification ER [ESA members, Canada, Slovenia] • D047 COG/ZARR Evaluation ER [ESA members, Canada, Slovenia] • D180 COG Implementation [ESA members, Canada, Slovenia] • D181 Zarr Implementation [ESA members, Canada, Slovenia]
<i>Compliance Interoperability & Testing Evaluation (CITE)</i>	<ul style="list-style-type: none"> • D045 CITE ER [ESA members, Canada, Slovenia] • D178 TEAM Engine Test OGC API-Processes [ESA members, Canada, Slovenia] • D179 Alternative Test OGC API-Processes [ESA members, Canada, Slovenia]

Table 5. CFP Deliverables - ESA Members, Canada, Slovenia - Sorted by ID

<ul style="list-style-type: none"> • D045 CITE ER • D046 COG Specification ER • D047 COG/ZARR Evaluation ER • D175 API Experiments Client (Python) • D176 API Experiments Client (Javascript or Typescript) • D178 TEAM Engine Test OGC API-Processes • D179 Alternative Test OGC API-Processes • D180 COG Implementation • D181 Zarr Implementation
--

Table 6. CFP Deliverables - ERs Open to All Bidders - Sorted by ID

- D002 Aviation API ER
- D007 Data Centric Security ER
- D008 Features and Geometries JSON ER
- D012 Geo Data Cube API ER
- D020 Moving Features ER
- D021 OGC API - Moving Features ER
- D022 Model Driven Standards ER
- D023 OGC UML Modeling Best Practices
- D024 Federated Security ER
- D025 Integrated Security ER
- D026 Key Management Server API ER
- D027 Features and Geometries JSON CRS Analysis of Alternatives ER
- D028 Federated Analytics Notional Architecture ER
- D029 Federated Analytics Interoperability Experiment Plan ER
- D030 Sensor Integration Framework Assessment ER
- D031 MASBUS Integration ER
- D032 SIF Semantic Model ER
- D040 API Experiments ER

Table 7. CFP Deliverables - Components Open to All Bidders - Sorted by ID

- D100, D101 Features and Geometries JSON Server for Aviation (2 instances)
- D102, D103 Features and Geometries JSON Client for Aviation (2 instances)
- D104, D105 Existing SWIM Service Exposing new Aviation API (2 instances)
- D106, D107 Fusion Service Exposing new Aviation API (2 instances)
- D108 End-user Client
- D109 Aviation Client
- D112, D146 Data Centric Security Server (2 instances)
- D113, D152 Mobile Phone App (2 instances)
- D115 Features and Geometries JSON Server
- D116 Features and Geometries JSON Client
- D122, D123 Geo Data Cube Service (2 instances)
- D124, D125 Geo Data Cube API Client (2 instances)
- D126 ML Model
- D135, D136 Ingestion Service (2 instances)
- D137, D138 Tracking Service (2 instances)
- D139 Machine Analytics Client
- D140 - Human Analytics Client
- D141, D142 Storage Service (2 instances)
- D143, D144, D145 MDA Tool (3 instances)
- D147 Data Centric Security Client
- D148, D149 PEP (2 instances)
- D150, D151 Key Management Service (2 instances)
- D153, D154 MASBUS Server (2 instances)
- D155, D156 MASBUS Client (2 instances)
- D165 API Experiments Server (Python)
- D166 API Experiments Server (Javascript or Typescript)
- D167, D168 Data Backend and Deployment (2 instances)

Chapter 4. Miscellaneous

Call for Participation (CFP): The CFP includes of a description of [deliverables](#) against which bidders may submit proposals. Several deliverables are more technical in nature, such as documents and component implementations. Others are more administrative, such as monthly reports and meeting attendance. The arrangement of deliverables on the timeline is presented in the [Master Schedule](#).

Each proposal in response to the CFP should include the bidder's technical solution(s), its cost-sharing request(s) for funding, and its proposed in-kind contribution(s) to the initiative. These inputs should all be entered on a [per-deliverable basis](#), and proposal evaluations will take place on the same basis.

Once the original CFP has been published, ongoing updates and answers to questions can be tracked by monitoring the [CFP Corrigenda Table](#) and the [CFP Clarifications Table](#). The HTML version of the CFP will be updated automatically and stored at the same URL as the original version. The PDF version will have to be re-downloaded with each revision.

Bidders may submit questions using the *Additional Message* textbox in the [OGC Innovation Program Contact Form](#). Question submitters will remain anonymous, and answers will be regularly compiled and published in the CFP clarifications.

A Bidders Q&A Webinar will be held on the date listed in the [Master Schedule](#). The webinar is open to the public, but anyone wishing to attend must register using the provided link. Questions are due on the date listed in the [Master Schedule](#).

Participant Selection and Agreements: Following the submission deadline, OGC will evaluate received proposals, review recommendations with Sponsors, and negotiate Participation Agreement (PA) contracts, including statements of work (SOWs). Participant selection will be complete once PA contracts have been signed with all Participants.

Kickoff: The Kickoff is a 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 Telecons and Meetings After the Kickoff, participants will meet frequently via weekly telecons and quarterly [OGC Member Meetings](#).

Development of Deliverables: Development of Components, Engineering Reports, Change Requests, and other deliverables will commence during or immediately after Kickoff.

Under the Participation Agreement contracts, **ALL** Participants will be responsible for contributing content to the ERs, particularly regarding their component implementation experiences, findings, and future recommendations. But the ER Editor will be the primary author on the shared sections such as the *Executive Summary*.

More detailed deliverable descriptions appear under [Types of Deliverables](#).

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 webinars, 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 Report milestone.

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 generally take the form of Documents or Component 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 common [ER Template](#) will be used as the starting point for each document. Various template files will contain requirements such as the following (from the *1-summary.adoc* file):

TIP

The Executive Summary shall contain a business value statement that should describe the value of this Engineering Report to improve interoperability, advance location-based technologies or realize innovations.

Ideas for meeting this particular requirement can be found in the [CFP Background](#) as well as in previous ER content such as the business case in the [SELFIE Executive Summary](#).

Document content should follow this [OGC Document Editorial Guidance](#) (scroll down to view PDF file content). File names for documents posted to Pending should follow this pattern (replacing the document name and deliverable ID): *OGC Testbed-17: Aviation Engineering Report (D001)*. For ERs, the words *Engineering Report* should be spelled out in full.

A.3.2. Component Implementations

Component Implementations include services, clients, datasets, and tools. A service component is typically delivered by deploying an endpoint via an accessible URL. A client component typically exercises a service interface to demonstrate interoperability. Implementations should be developed and deployed in all threads for integration testing in support of the technical architecture.

IMPORTANT

Under the Participation Agreement contracts, **ALL** Participants will be responsible for contributing content to the ERs, particularly regarding their component implementation experiences, findings, and future recommendations. But the ER Editor will be the primary author on the shared sections such as the *Executive Summary*.

Component implementations are often used as part of outreach demonstrations near the end of the timeline. To support these demos, component implementations are required to include *Demo Assets*. For clients, the most common approach to meet this requirement is to create a video recording of a user interaction with the client. These video recordings may optionally be included in a new YouTube Playlist such as [this one for Testbed-15](#).

Videos to be included in the new YouTube Playlist should follow these instructions:

TIP

- Upload the video recording to the designated Portal directory (to be provided), and
- Include the following metadata in the Description field of the upload dialog box:
 - A Title that starts with "OGC Testbed-17:", keeping in mind that there is a 100-character limit [if no title is provided, we'll insert the file name],
 - Abstract: [1-2 sentence high-level description of the content],
 - Author(s): [organization and/or individuals], and
 - Keywords: [for example, OGC, Testbed-17, machine learning, analysis ready data, etc.].

Since server components often do not have end-user interfaces, participants may instead support outreach by delivering static UML diagrams, wiring diagrams, screenshots, etc. In many cases, the images created for an ER will be sufficient as long as they are suitable for showing in outreach activities such as Member Meetings and public presentations. A server implementer may still choose to create a video recording to feature their organization more prominently in the new YouTube playlist. Another reason to record a video might be to show interactions with a "developer user" (since these interactions might not appear in a client recording for an "end user").

TIP

Demo-asset deliverables are slightly different from [TIE testing deliverables](#). The latter don't necessarily need to be recorded (though they often appear in a recording if the TIE testing is demonstrated as part of one of the recorded weekly telecons).

A.4. Proposal Evaluation

Proposals are expected to be brief, broken down by deliverable and precisely addressing the work items of interest to the bidder. Details of 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, whether the 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 (costs for which no cost-share compensation has been requested). As a rough guideline, a proposal should include at least one dollar of in-kind contribution for every dollar of cost-share compensation requested. All else being equal, higher levels of in-kind contributions will be considered more favorably during evaluation. Participation may also take place by purely in-kind contributions (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 notifying Bidders of their selection to enter negotiations to become and initiative Participant. Each selected bidder will enter into a **Participation Agreement (PA)**, which will include a **Statement of Work (SOW)** describing the assigned deliverables.

A.5. Reporting

Participants will be required to report the progress and status of their work; details will be provided during contract negotiation. Additional administrative details such as invoicing procedures will also be included in the contract.

A.5.1. Monthly Reporting

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 each participant's perspective. The IP Team will review action item status on a weekly basis with assigned participants. Initiative participants must remain available for the duration of the timeline so these contacts can be made.

A.5.2. Participant Final Summary Reports

Each Participant should submit a Final Summary Report by the milestone indicated in the [Master Schedule](#). These reports should include the following information:

1. Briefly summarize Participant's overall contribution to the testbed (for an executive audience),
2. Describe, in detail, the work completed to fulfill the Participation Agreement Statement of Work (SOW) items (for a more technical audience), and
3. Present recommendations on how we can better manage future OGC Innovation Program initiatives.

This report may be in the form of email text or a more formal attachment (at the Participant's discretion).

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 the [Step-by-Step Instructions](#) below.

IMPORTANT

Please note that the content of the "Proposed Contribution" text box in the Bid Submission Form will be accessible to all Stakeholders and should contain **no confidential information** such as labor rates.

Similarly, no sensitive information should be included in the [Attached Document of Explanation](#).

Proposals must be submitted before the deadline indicated in the [Master Schedule](#).

Bidders responding to this CFP must be organizational OGC members familiar with the OGC mission, organization, and process.

Proposals from non-members or individual members will be considered *provided that* a completed application for organizational membership (or a letter of intent) is submitted prior to or with the proposal.

TIP

Non-members or individual members should make a note regarding their intent to join OGC on the **Organizational Background** page of the *Bid Submission Form* and include their actual **Letter of Intent** as part of an [Attached Document of Explanation](#).

The following screenshot shows the **Organizational Background** page:

Organization Name*:	<input type="text"/>
First Name (Given) of Technical Contact*:	<input type="text"/>
Last Name (Family) of Technical Contact*:	<input type="text"/>
Email Address of Technical Contact*:	<input type="text"/>
Telephone of Technical Contact*:	<input type="text" value="+1 201-555-0123"/>
First Name (Given) of Business Contact*:	<input type="text"/>
Last Name (Family) of Business Contact*:	<input type="text"/>
Email Address of Business Contact*:	<input type="text"/>
Telephone of Business Contact*:	<input type="text" value="+1 201-555-0123"/>
Organizational Background:	
	<div style="border: 1px solid #ccc; padding: 5px;"> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> File Edit View Insert Format Table Help </div> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> ← → Formats B <i>I</i> ☰ ☰ ☰ ↔ ☰ ☰ ☰ ☰ ☰ ☰ ☰ <> 🗑️ 👁️ ? </div> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; min-height: 100px;"> </div> <div style="text-align: right; padding-right: 5px;"> 0 WORDS </div> </div>
<input type="checkbox"/>	By checking this box, Bidder acknowledges its understanding and acceptance of all Participant obligations described in CFP Appendix A: Management Requirements.
Update and Continue	

Figure 17. Sample Organizational Background Page

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. Sensitive information other than labor-hour and cost-share estimates should not be submitted.

Bidders will be selected for cost share funds on the basis of adherence to the CFP requirements and the overall proposal quality. The general testbed objective is to inform future OGC standards development with findings and recommendations surrounding potential new specifications. Each proposed deliverable should formulate a path for (1) producing executable interoperable prototype implementations meeting the stated CFP requirements and (2) documenting the associated findings and recommendations. Bidders not selected for cost share funds may still request to participate on a purely in-kind basis.

Bidders should 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. Out-of-scope items could potentially be included in another OGC IP initiative.

Each selected Participant (even one not requesting any funding) will be required to enter into a

Participation Agreement contract ("PA") with the OGC. The reason this requirement applies to purely in-kind Participants is that other Participants will likely be relying upon their delivery. Each PA will include a Statement of Work ("SOW") identifying specific Participant roles and responsibilities.

B.2. Questions and Clarifications

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

Bidders may submit questions using the *Additional Message* textbox in the [OGC Innovation Program Contact Form](#). Question submitters will remain anonymous, and answers will be regularly compiled and published in the CFP clarifications.

A Bidders Q&A Webinar will be held on the date listed in the [Master Schedule](#). The webinar is open to the public, but anyone wishing to attend must register using the provided link. Questions are due on the date listed in the [Master Schedule](#).

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.

For any individual who has not used this form in the past, a new account will need to be created first. 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.

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.

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 updates will be provided as needed.

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

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 provide organizational information on the [Organizational Background Page](#) and click *Update and Continue*.

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

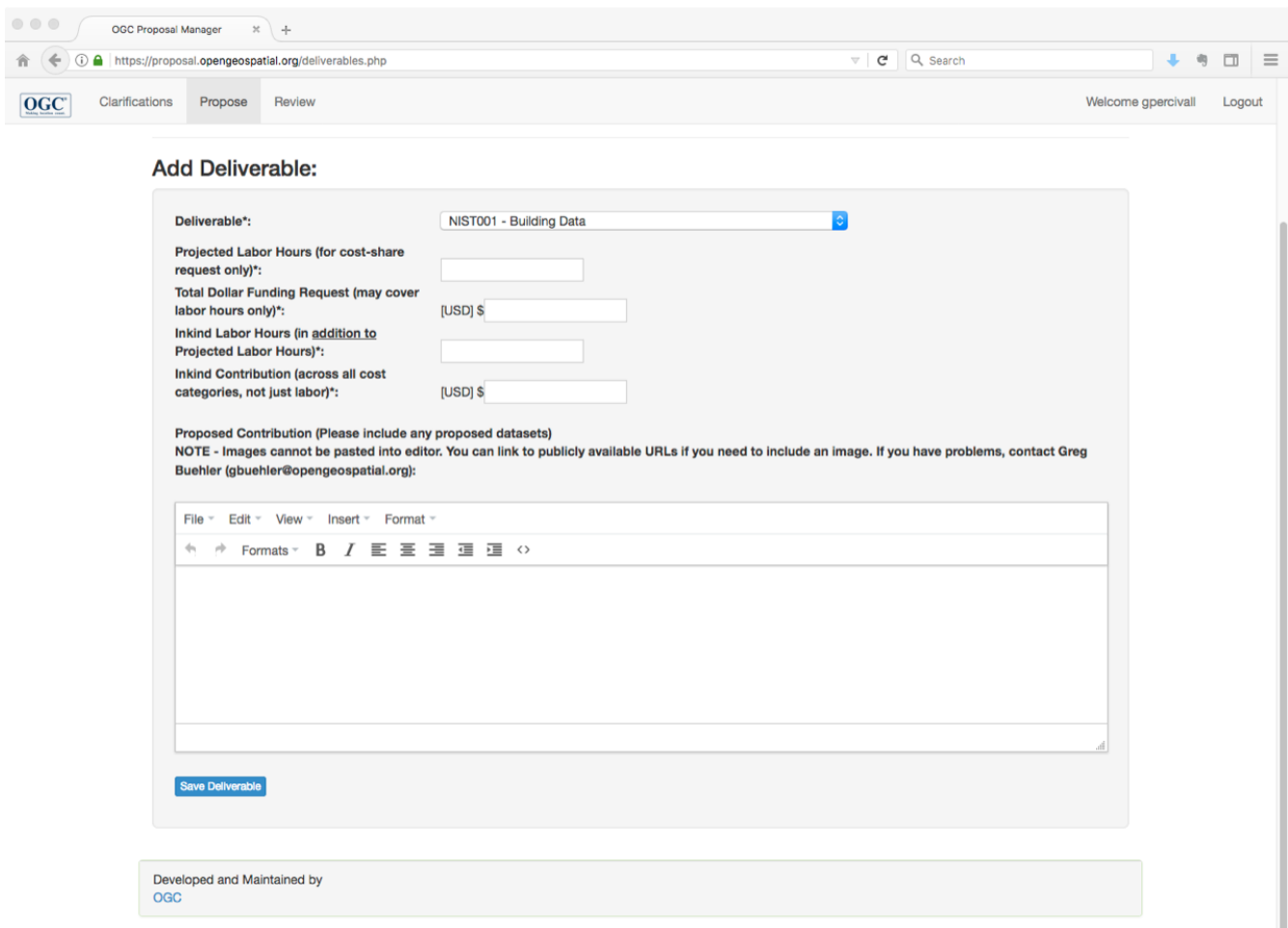


Figure 18. Sample "Add Deliverables" Page

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

TIP

For component implementations having multiple identical instances of the same deliverable, the bidder only needs to propose just one instance. For simplicity, each bidder should just submit against the lowest-numbered deliverable ID. OGC will assign a unique deliverable ID to each selected Participant later (during negotiations).

On the far right, the *Review* link navigates to a page summarizing all the deliverables the Bidder is proposing. 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 points during proposal creation.

Once the *Submit* button is clicked, the user will receive an immediate confirmation on the website that their proposal has been received. The system will also 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*.

Done Adding Deliverables:

Do you need to attach a document of explanation*:

Yes No

Deliverables Completed - Continue

Please do not include any sensitive information (such as labor rates) in this documentation.

Select a file to upload.

Choose File No file chosen

Only .DOC, .DOCX, .ODF, .PDF, .TAR, .TXT, or .ZIP files with maximum size of 5 MB is allowed.

Upload

Deliverables Completed - Continue

Figure 19. Sample Dialog for an "Attached Document of Explanation"

IMPORTANT

No sensitive information (such as labor rates) should be included in the *Attached Document of Explanation*.

If this attachment is provided, it is limited to one per proposal and must be **less than 5Mb**.

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. Step-by-Step 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 item.

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, and
- Estimated *In-Kind Contribution*: total U.S. dollar estimate of the in-kind amount to be contributed for this deliverable (including all cost categories).

TIP

There's no separate text box to enter a global in-kind contribution. Instead, please provide an approximate estimate on a per-deliverable basis.

Cost-sharing funds may **only** be used for the purpose of offsetting burdened labor costs of development, engineering, documentation, and demonstration 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, and so on.

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 repeat 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.

Similarly, no sensitive information should be included in the [Attached Document of Explanation](#).

This field *Proposed Contribution (Please include any proposed datasets)* should 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.

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

In general, please try to limit the length of each *Proposed Contribution* to about one text page per deliverable.

TIP

Note that images cannot be pasted into the field *Proposed Contribution* textbox. Bidders should instead provide a link to a publicly available image.

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 selected 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.
- Suppliers are organizations (not necessarily OGC members) who have offered to supply specialized resources such as 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 or individual members will be considered *provided that* a completed application for organizational membership (or a letter of intent) is submitted prior to or with the proposal.
 - Non-members or individual members should make a note regarding their intent to join OGC on the **Organizational Background** page of the *Bid Submission Form* and include their actual **Letter of Intent** as part of an [Attached Document of Explanation](#).
- 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.
- Individuals from any OGC member organization that does not become an initiative Sponsor or Participant may still (as a benefit of membership) observe 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 these obligations are met in a timely manner (whether a 3rd-party subcontractor provides assistance is up to the Participant).
- 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 required, reliable contributions from the 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 of any component source code to OGC.
 - What is delivered to OGC is the behavior of the component installed on the Participant's machine, and the corresponding documentation of findings, recommendations, and technical artifacts contributed to 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](#).

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

D.1. Corrigenda Table

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.

Table 8. Corrigenda Table

Section	Description
2.9.1 OGC Features and Geometries JSON Problem Statement and Research Questions	Paragraph added at the end
1.4 Master Schedule	Extend deadline to submit questions for the Bidders Q&A Webinar from 6th to 12th January.
2.8.5 (Attracting Developers) Work Items and Deliverables	Delete "with" and insert "including".
B.1 General Proposal Submission Guidelines	Add instructions for non-members to submit Letters of Intent.
2.9 OGC Features and Geometries JSON	Change task name to <i>OGC Features and Geometries JSON</i> .

D.2. Clarifications Table

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

Please us this [convenience link](#) to navigate to the end of the table.

Table 9. Clarifications Table

Question	Clarification
-- Pre-Release --	
<i>How can we submit additional questions?</i>	Submit an inquiry using the <i>Additional Message</i> textbox in the OGC Innovation Program Contact Form .
<i>When are additional questions due if we want them addressed as part of the Bidders Q&A Webinar?</i>	Questions are due on the date listed in the Master Schedule .
-- 17 December --	

Question	Clarification
<p><i>Q: The 2.1.2 Sensor Integration Previous Work section contains a link to a Git repository for the Sensor Integration Framework (SIF). The UML diagrams in this repo are XML files. Providing instructions on how to view the diagrams would be helpful.</i></p>	<p>A: These are XMI files which include UML diagrams that can be imported for viewing. The files in the SIF repo UML_Model directory contain several hundred such diagrams altogether.</p> <p>There are several available tools that can import and view these diagrams. Examples of proprietary tools are Enterprise Architect and Altova UModel. Examples of open source tools such as Modelio are listed in Wikipedia. Import capabilities might also be available in IDEs such as Eclipse. A web search on "XMI viewer" might also be instructive.</p>
<p>-- 8 January --</p>	
<p><i>Q: Regarding the aviation aspects of 2.9 OGC Features and Geometries JSON, are there any relevant materials on what the unique challenges are for using GeoJson to encode aviation data? The call mentions that "The scenario shall evaluate aviation specific requirements such as geometry constraints and profiles." Is it possible to share some document that discusses the specific requirements?</i></p>	<p>A: The challenge with GeoJSON for aviation is primarily to further reduce the number of options, in particular to express geometries. Two particularly relevant questions are what geometries are required by the aviation community and how to implement a profiled version of GeoJSON? Developing the specific use cases themselves will be part of the testbed activity. Section 2.3 Web API for Aviation gives some ideas on how these might look. Each Bidder should also suggest use cases as part of their proposal.</p>
<p>-- 12 January --</p>	
<p><i>What sort of budget is there for the whole testbed, and what are the distributions of funds for different threads and work items?</i></p>	<p>The whole Testbed is roughly 2M USD in funding. The funds are distributed depending on complexity of the work items.</p>
<p><i>What organizations may apply? Can an individual who is a member of an organization submit a proposal?</i></p>	<p>Anyone may apply, but membership is required. Cost share funding can only be made available to organizations, not to individual members.</p>
<p><i>What's the interest level of the sponsors for augmented reality (AR) features in the clients? Is visualization in AR valuable for these sponsors?</i></p>	<p>No specific AR requirements have been mentioned, but AR features are certainly a great additional value to any client.</p>

Question	Clarification
<p><i>In the 2.2 Moving Features from Digital Motion Imagery task, is it possible to add use cases that have urban data? There is a possibility that one or more partners could contribute such data and even location(s) with moving image sensors.</i></p>	<p>Yes, that is possible. The use cases have not been fully defined. To a good extent, they depend on the available data.</p>
<p><i>The SensorHub was developed and demonstrated in Testbed 16 (Demonstrator 1 of the ER 20-036). In the engineering report 20-036 OGC SensorThings Sensing API, there is a class diagram (figure 20) that has location, but not orientation of the "sensor thing". Is it within scope for the Testbed 17 2.2 Moving Features from Digital Motion Imagery task to propose addition of GeoPose for sensors to the architecture and to demonstrate its implementation?</i></p>	<p>Yes.</p>
<p><i>Under the 2.10 Geo Data Cubes task, how are Geo Data Cubes different or similar to Open Data Cubes?</i></p>	<p>Open Data Cube is one implementation of a data cube. A geo data cube is a data cube with geospatial data.</p>
<p><i>Under the 2.10 Geo Data Cubes task, could the discovery of Geo Data Cubes through an organization's Spatial Discovery Service be added as a software component (e.g., similar to D122 Geo Data Cube Service) but aimed at in-situ/on-site discovery use-cases?</i></p>	<p>Yes, we are open to such an offering.</p>
<p><i>Under the 2.10 Geo Data Cubes task, if Geo Data Cubes are associated with DGGS (discrete global grid systems), would it be of interest to implement a solution that assigns a GeoPose calculated a priori from DGGS coordinates? Our organization could potentially demonstrate how Geo Data Cubes could be discoverable through our Spatial Discovery Services by annotating a record for a data cube with GeoPose.</i></p>	<p>The focus of Geo Data Cubes is on the API. In this context, discovery is an important topic. Adding GeoPose is an interesting extension, but we need to make sure that the key requirements are addressed first.</p>

Question	Clarification
<i>Under the 2.10 Geo Data Cubes task, is there a possibility to have an extension to the Geo Data Cube scenario to include a real time component (e.g., the real time capture of reality to (a) populate and (b) have real time access to/to "consume" a portion or all of a Geo Data Cube in real time?)</i>	Yes, though this is not the primary focus in Testbed-17.
-- 13 January --	
<i>Will the recording be made available for download/viewing?</i>	Yes, the link will be on the Testbed-17 initiative page (and it will also be emailed to registered webinar attendees).
<i>Will the PowerPoint deck used in this presentation be available for download?</i>	Yes, the link is on the Testbed-17 initiative page (as are the links to the CFP in both HTML and PDF formats).
<i>Can you please provide the URL to the Tips and Tricks for New bidders?</i>	Tips for New Bidders.
<i>Are you suggesting that members should skip the "Organizational Background" question (of the Bid Submission Form)?</i>	No, please provide the organizational background in any case, even for existing OGC members.
<i>Did not finish all information on first page but hit "update and continue". How do I get back to finish adding info on page 1?</i>	Scott will address this in a written clarification later.
<i>Are there any important changes in the submission process from that which was required for Testbed-16?</i>	No, the process is nearly identical to that used in Testbed-16.
<i>Just looking to confirm that members, if unable to participate or not chosen for particular deliverables, will still be able to sign the Observer Agreement and still engage in the thread/tasks?</i>	Yes, every member can join the Testbed as an observer or fully in-kind contributor. A link to the Testbed-17 Observer Agreement is on the Portal page at https://portal.ogc.org/?m=public&orderby=default&tab=7 .
<i>Under the 2.4 Data Centric Security Across OGC APIs and Federated Clouds task, can the deliverable for D148/D113 be a GUI that offers a command line and a text box with the results instead of a full GIS App? Can D147 be a command line executable?</i>	Yes, as a minimum, though it would great to see functionality embedded inside a slightly richer client. Yes, D147 can be delivered as a command line executable as a minimum, though a graphical client is preferred.

Question	Clarification
<p><i>Under the 2.4 Data Centric Security Across OGC APIs and Federated Clouds task, since D148/D113 are a mobile app, should a zip or mobile archive format be the deliverable instead of a docker (as dockers don't run on mobile devices)?</i></p>	<p>Yes, that is the more appropriate delivery format in this case.</p>
<p><i>Under the 2.4 Data Centric Security Across OGC APIs and Federated Clouds task, if a GUI application is delivered in D147 does the GUI application need to run in a docker?</i></p>	<p>Ideally yes, but alternatively it can be delivered in a format that allows exploring it on machines operated by sponsors.</p>
<p><i>(See updated Q&A below under 21 January.) Under the 2.1 Sensor Integration task, must D153 be implemented as an extension of the MASBUS software or can another implementation of SIF be developed?</i></p>	<p>We need the MQTT part as an extension to MASBUS. We will check with the sponsor to which extent this applies to the SIF.</p>
<p><i>Under the 2.3 Web API for Aviation task, deliverable D104, is there an existing SWIM service that is openly available and that could be used in the SWIM service facade? Or links to public documentation of the technical details of such services and the data?</i></p>	<p>The Testbed-16 Aviation task made use of several SWIM data sources in building an API - Features capability. For details, please consult Chapter 8. SWIM Data Relay API of the Testbed-16 Aviation ER.</p>
<p><i>Under the 2.2 Moving Features from Digital Motion Imagery task, what is difference between deliverable 141, 142 or 135, 136?</i></p>	<p>D141 and D142 are identical, so are 135 and 136. The first two serve as storage services, whereas the latter two are ingesting the tracklets and detections.</p>
<p><i>Under the 2.2 Moving Features from Digital Motion Imagery task, it mentioned similar tasks for 141 and 142. We want to submit proposals for each, but I do not know the difference between these two deliverables. What is the difference between these two deliverables?</i></p>	<p>There is no difference, please submit your proposal for 141 only. You will be automatically considered for both.</p>
<p><i>Under the 2.2 Moving Features from Digital Motion Imagery task, does D135 include object detection task? If not, what are inputs of the ingestion service (for example VMTI or specific detection format)?</i></p>	<p>Ideally, the ingestion system uses a VMTI source, but the component can be based on/produce fully synthesized data. In the latter case, it is important that the ingestion service provides meaningful detections and sequences thereof.</p>

Question	Clarification
<p><i>Under the 2.2 Moving Features from Digital Motion Imagery task, deliverable D139 Machine Analytics Client, can we propose a service to handle enrichment or we are forced to provide the solution completely on client side?</i></p>	<p>The component may be implemented as a service.</p>
<p>-- 14 January --</p>	
<p><i>Where can the Testbed-16 Engineering Reports be found?</i></p>	<p>Published reports are now available under the Testbed-16 section of the OGC Public Engineering Report Repository. There are also draft versions of the Data Centric Security ER (OGC 20-021) and the Machine Learning ER (OGC 20-015r2).</p>
<p>-- 18 January --</p>	
<p><i>Under the 2.8 Attracting Developers: Lowering the entry hurdle for OGC Web API experiments task, D165 - API Experiments: Are we only limited to Python/Javascript developed servers? What about using Spring Boot or Quarkus (for Java)?</i></p>	<p>The goal is to have Python and Javascript in focus for Testbed-17. Bidders are free to offer additional ideas. Given that the general goal is to provide entry points for microservice architectures and the fact that new environments such as Spring Boot and Quarkus bring Java back into play, there are certainly alternatives to the required Python/Javascript. Bidders are invited to provide arguments for alternatives.</p>
<p><i>Under the 2.8 Attracting Developers: Lowering the entry hurdle for OGC Web API experiments task, D040 - API Experiments: the ER seems to have a focus on Jupyter notebooks, is the Javascript API client also included?</i></p>	<p>No, focus is not in Jupyter Notebooks, but should include these. The ER definition in 2.8.5 (Attracting Developers) Work Items and Deliverables was misleading. It has been corrected to read: "Engineering Report providing all usage, deployment and installation instructions, guidelines and other documentation required to get started with OGC APIs within cloud environments. The report shall address design, development, and deployment of OGC Web APIs and provide best practices on how to use OGC APIs with including Jupyter Notebooks for use cases and scenarios to be defined at/after Testbed-17 Kick-off. The Engineering Report captures the proposed architecture, identifies the necessary standards, describes all developed components, reports on the results of all Technical Interoperability Experiments (TIE) activities, provides an executive summary and a description of recommended future work items."</p>

Question	Clarification
<p><i>Under the 2.8 Attracting Developers: Lowering the entry hurdle for OGC Web API experiments task, D167 - API Experiments: the data backend seems to be OGC W*S, database and object storage at the same time (all three). Is that correct? Object storage is used for raster data and ? Should we set up COGs?</i></p>	<p>The idea is to have different kinds of data repositories being explored. Having both vector and coverage data would certainly be helpful. Another usage of object storage is ad-hoc provisioning of taped data. The tape itself is not available to the user, but data from the tape will be made available in an object storage for further processing.</p>
<p><i>Under the 2.7 COG & Zarr: Specification & Evaluation task, D180 - COG & ZARR: only one software component for COG and one for ZARR? What does it do? Read? Write? Service? Display?</i></p>	<p>The goal of the task is to develop a draft specification for COG that complies with OGC Standards Program P&Ps, and compare it with the expected ZARR Community Standard as well as with other container formats as described in section 2.7.1. Bidders are invited to come forward with a scenario that allows to respond to the task requirements and in particular the research questions provided in section 2.7.1. The number of funded implementations for D180/181 depends on the received bids, but bidders shall assume that they need to showcase a full scenario. Bidders only providing parts of the scenario shall define what additional components are required.</p>
<p><i>Is the "Library GPL" considered a permissive license (The library GeoTools uses this license)?</i></p>	<p>A permissive open source license is a non-copyleft open source license that guarantees the freedom to use, modify, and redistribute, while also permitting proprietary derivative works.</p>
<p><i>D122 - What is the structure/format and the interface between the data cube service and the "data cube" mentioned in the 2.10 Geo Data Cubes task?</i></p>	<p>This interface is not defined. It depends on the data cube implementation you chose. The bidder is free to use any type of interface. The usage of commonly used data cube implementations is recommended but not required.</p>
<p><i>Under the 2.9 OGC Features and Geometries JSON task, D115 - Features and Geometries JSON server: what data is served? Is it simple features?</i></p>	<p>We expect simple features, but the exact type of data is not defined at this stage. Bidders are invited to make suggestions, ideally with corresponding data being available and usable in the context of Testbed-17.</p>

Question	Clarification
<p><i>Topic 2.3 Web API for Aviation calls for development of an Aviation API. Is this Aviation API intended to be all inclusive of all aspects of Aviation data including Aeronautical (AIXM), Flight (FIXM), Weather (WXXM), surveillance, maintenance, flow, etc.? Would you expect there to be separate Aviation APIs for each data type?</i></p>	<p>We don't expect a complete API developed in Testbed-17 that includes everything. The API should demonstrate Web API functionality for the aviation domain in principle. If that results in a single or multiple APIs shall be one result of the testbed.</p>
<p><i>Under the 2.8 Attracting Developers: Lowering the entry hurdle for OGC Web API experiments task, ESA funding only covers Client deliverables. Why is there not ESA funding for the Server? We would be interested in proposing to a server deliverable if ESA funding is available. While it's clear that ESA members can bid on the server component, can ESA funding be considered for the Server component?</i></p>	<p>Task 2.8 has funding available for the components listed in section 2.8.5. The server components D165-168 are funded by a different sponsor. There are no constraints on these four items (anybody may bid).</p>
<p><i>Under the 2.2 Moving Features from Digital Motion Imagery task, there are several pairs of similar deliverables, for example, D135 and D136, D137 and D138, and D141 and D142. What is the difference between these deliverables?</i></p>	<p>There is no difference. Please submit your proposal for the lower-numbered deliverable in each pair, and you will be automatically considered for both.</p>
<p>-- 19 January --</p>	
<p><i>Does the 2.2 Moving Features from Digital Motion Imagery task include creating a data model for Moving Features API? Or can we propose to use other OGC APIs to handle storage service tasks (e.g. STA)?</i></p>	<p>Yes, the Moving Features API requires a data model to exchange moving features, tracklets, and tracks.</p>

Question	Clarification
<p><i>2.11 Compliance Interoperability & Testing Evaluation (CITE) is missing a reference implementation. If the test is developed and there is no one to test it, it will be hard to assess if the tests are sound. Getting a purely informal in-kind contribution from the community will slow down the process to assess the testing and the new environment. I suggest adding a reference implementation component or explain what will be the envision process for testing the test.</i></p>	<p>Goal of Task 2.11 is mainly to explore alternatives to TEAM engine and to experiment with the most promising solution. The experiments shall use OGC API Processes. We expect the bidder to provide an implementation. This implementation does not need to be "the reference" implementation, but it should cover the essential elements of the API. Once we have agreement on the future testing environment, we will start with reference implementations. That said, Testbed-17 participants may even develop an implementation that can be considered a reference.</p>
<p><i>Under the 2.8 Attracting Developers: Lowering the entry hurdle for OGC Web API experiments task, if we implement D167 do we also need to implement D165? It says in the D167 text "The provider of the data storages shall deploy the server component as developed, documented and delivered by D165 and D166".</i></p>	<p>No, the participant doing D167 shall test the results delivered by D165 and D166 by deploying it according to the delivered documentation. The participant doing D167/168 is the stress tester for whatever is delivered by the other participants.</p>
<p>-- 21 January --</p>	
<p><i>Under the 2.1 Sensor Integration task, must D153 be implemented as an extension of the MASBUS software or can another implementation of SIF be developed?</i></p>	<p>We need the MQTT part as an extension to MASBUS. Bidders are invited to make suggestions regarding the extent to which they think this should apply to the SIF.</p>
<p>-- 22 January --</p>	
<p><i>Under the COG & Zarr: Specification & Evaluation task, component D180 is a very general component. For example, it can be a read or write implementation. But in the submission form there is no way to submit two components. Once you have submitted the first one, the form will not show the deliverable number to perform a second submission. Can you please fix or advise?</i></p>	<p>If a bidder wishes to provide two component implementations as part of this single deliverable, they should simply describe these two as part of their submission for the single D180 deliverable.</p>

Question	Clarification
<p><i>Our team would like to participate and we did not do it before. Any support to start off is highly appreciated.</i></p>	<p>This would be a perfect case for the new How-To video that was recently published. The link is in the 3rd paragraph of the initiative page. But here a copy of the language for convenience: "For step-by-step instructions on how to submit a proposal (in addition to the instructions in the CFP itself), please download the 1-hour video (about 320MB) How to submit a Testbed-17 CFP Proposal. It covers much of the same content as was covered in the Bidders Q&A Webinar."</p>
<p><i>-- n January --</i></p>	
<p>—</p>	<p>—</p>

D.3. End of Clarifications Table (convenience link)