

OGC Testbed-13
Fit-for-Purpose Engineering Report

Table of Contents

1. Summary	4
1.1. Demonstration Scenario	4
1.2. Key Requirements	8
1.3. Component Implementation Design	10
1.4. Prior-After Comparison	14
1.5. Technology Integration Experiments	15
1.5.1. Filters and Encodings	15
1.5.2. Services and Data	16
1.5.3. Client Applications	19
1.6. Key Findings	37
1.7. What does this ER mean for the Working Group and OGC in general	38
1.8. Document contributor contact points	38
1.9. Future Work	39
1.10. Foreword	39
2. References	40
3. Terms and Definitions	41
3.1. Abbreviated Terms	42
4. Overview	44
5. Experiments	45
5.1. Demonstration Scenario	45
5.2. Key Requirements	49
5.3. Component Implementation Design	51
5.4. Prior-After Comparison	55
5.4.1. WFS and Stored Query Support Assessment	55
5.4.2. OWS Context Documents Assessment	56
5.4.3. CQL_Filter on WFS, WMS, WCS and WPS Assessment	59
5.4.4. Role of a Client as a Mediator for Filters and Encodings	59
5.4.5. Role of the Workflow and WPS	60
5.4.6. OGC Filtering in GeoServer	60
5.4.7. Assessment on Best Method to Meet Filter Requirement	61
5.5. Technology Integration Experiments	61
5.5.1. Filters and Encodings	62
5.5.2. Services and Data	64
5.5.3. Client Applications	82
6. Findings	101
Appendix A: Context Encodings	102
Appendix B: Revision History	118
Appendix C: Bibliography	119

Publication Date: 2018-01-18

Approval Date: 2017-12-07

Posted Date: 2017-10-31

Reference number of this document: OGC 17-038

Reference URL for this document: <http://www.opengis.net/doc/PER/t13-DG001>

Category: Public Engineering Report

Editor: Jeff Harrison

Title: OGC Testbed-13: Fit-for-Purpose Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright © 2018 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Summary

The objective of the Fit for Purpose (FFP) effort in Testbed 13 was to develop and test filters and encodings in a platform that can ease the work of end-users, especially those who are not expert in dealing with geospatial data and satellite imagery. The platform was demonstrated in a scenario that showed how these filters can enable information exchange for humanitarian relief and analysis of mass movement of populations.

This section provides a summary of the interoperability tools and practices used by Testbed 13 participants in this platform. It includes descriptions and testing results of filters and encodings to help simplify access to satellite imagery. This technology was tested in a scenario that showed how OGC-based services, encodings, filters and applications can help coordinate humanitarian relief activities among nations and organizations.

1.1. Demonstration Scenario

Testbed 13 focused on addressing challenges related to coordinating multi-regional / multi-national operations and messaging related to the displacement and mass movement of populations in response to conflict [1]. The current exodus of people across the Middle East to multiple nations was used as a scenario for this discussion.

In OGC Testbed 13, participants assessed the ability of open geospatial web services, encodings and filters to support simulated users in a Mass Migration Scenario over Syria and Jordan. In this scenario, large numbers of people have been displaced from the Daraa region of Syria to the Zaatari refugee camp in Jordan due to conflict. As the conflict ends 'de-escalation zones' are established and plans are made to return displaced people from refugee camps. Understanding the situation on the ground and the infrastructure, as well as transporting these people from refugee camps into a former conflict zone is a major challenge for relief agencies. To accomplish this task users must understand the environment and infrastructure in the region between Zaatari and Daraa.

The following examples provide a brief sample of the scenario involved in testing the platform and approach.

Years of war have displaced hundreds of thousands from Daraa city and region. Thousands have sought shelter at the sprawling Zaatari refugee camp...



Figure 1. Testbed 13 assessed the ability of open geospatial technology to support simulated users in a Mass Migration scenario.

A ceasefire has been declared this year. Government and non-government organizations seek to assist displaced populations at Zaatari in returning to Daraa...



Figure 2. In the testing scenario, organizations seek to help displaced populations return home from refugee camps.

Need to access information for Daraa region to assist humanitarian relief operations...

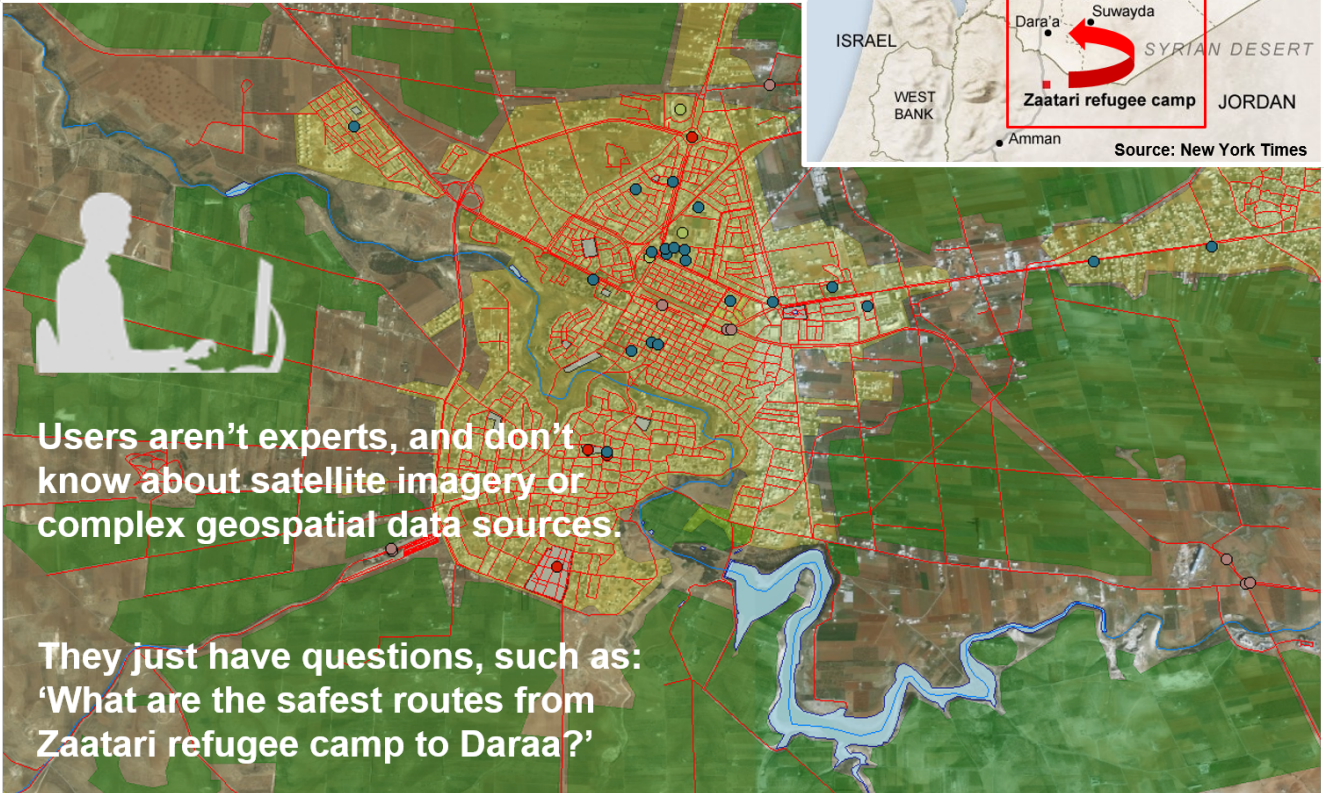


Figure 3. The test scenario included simulated users who are not familiar with satellite imagery or geospatial data sources.

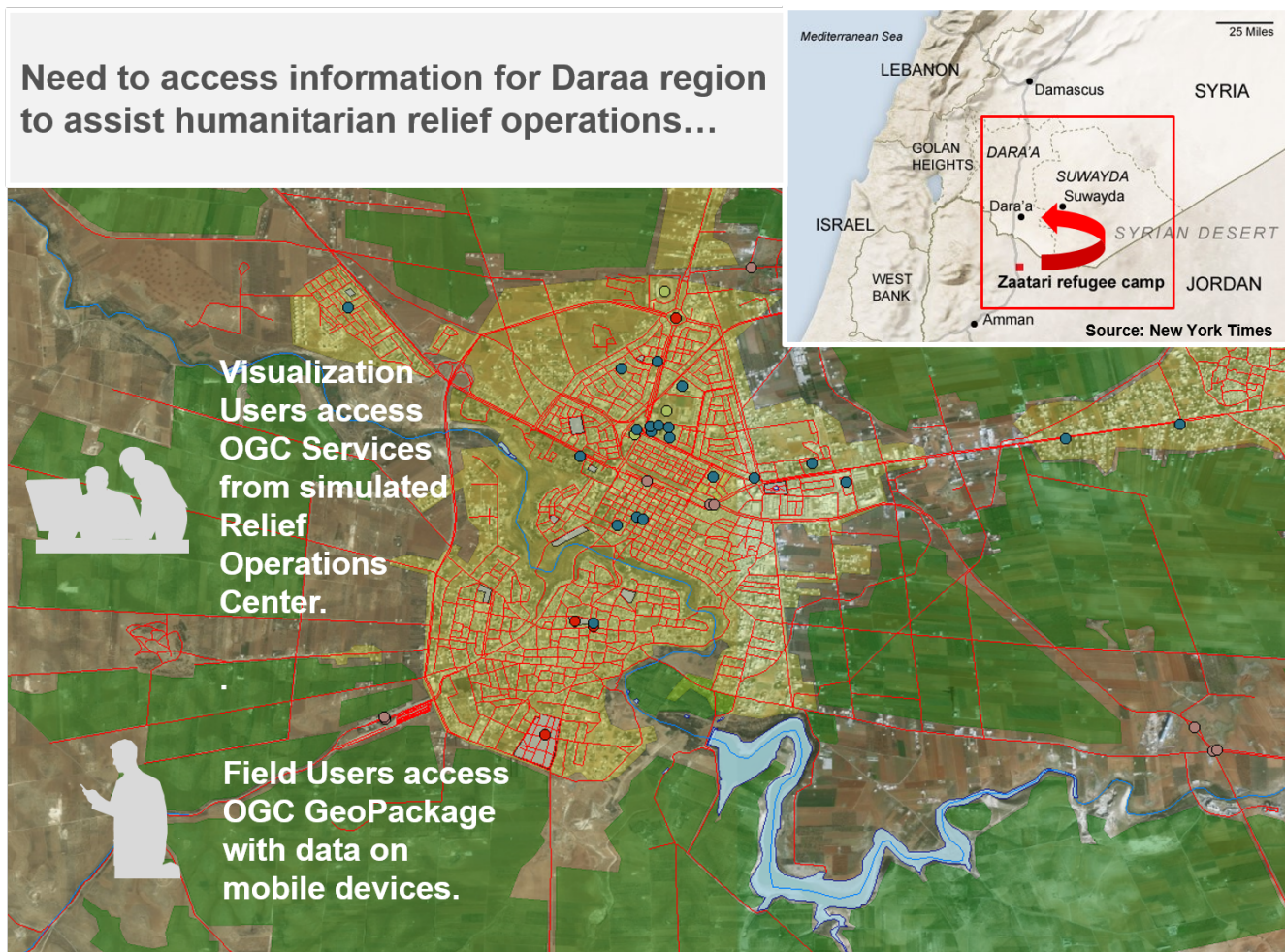


Figure 4. The test scenario also included users in a simulated operations center and users in the field.

Using this scenario as background, Testbed 13 participants demonstrated how new filters and encodings in the platform can ease the work of end-users, especially those who are not expert in dealing with geospatial data and satellite imagery.

1.2. Key Requirements

Currently, accessing geospatial data, and in particular satellite imagery, requires someone with expertise to know what content to look for - and how to discover and access the data. This challenge is increased given inconsistent use of metadata standards across providers. OGC services help with discovery and access of data, but most are still limited to the assumption that the client applications must explicitly provide filter criteria such as cloud cover, age or spatial resolution to the servers in the discovery process. This level of complexity needs to be reduced to open up geospatial data to broader markets.

A specific example was provided in the Testbed 13 Request for quotation (RFQ)/Call For Participation (CFP) about the ability for users to easily filter a catalog of satellite imagery exposed via Web Feature Service (WFS) according to their needs, and then exploit such imagery via other OGC services such as Web Map Service (WMS) and Web Coverage Service (WCS). This example includes the ability to perform sophisticated processing on the data using Web Processing Services (WPS).

It is clear in the Testbed 13 RFQ/CFP that the main interest in the FFP work comes from an imagery

provider perspective. In the RFQ/CFP the use of WFS is introduced as a way to allow users to query imagery metadata or library parameters (e.g. cloud cover, age, spatial resolution and so on) rather than actual vector data.

Apparently, the requestor had in mind a somewhat higher level filter built on top of a sequence of standard OGC services which:

- Starts by using WFS to query and discover image metadata and footprints.
- Then uses WMS to visualize on a map, browse or full resolution imagery.
- And then uses WCS and WPS to access and process imagery.

A graphical representation of this requirement is provided below. This representation is provided for the purposes of framing the discussion in the following section and not to provide a technical architecture. The technical architecture is provided in the following section.

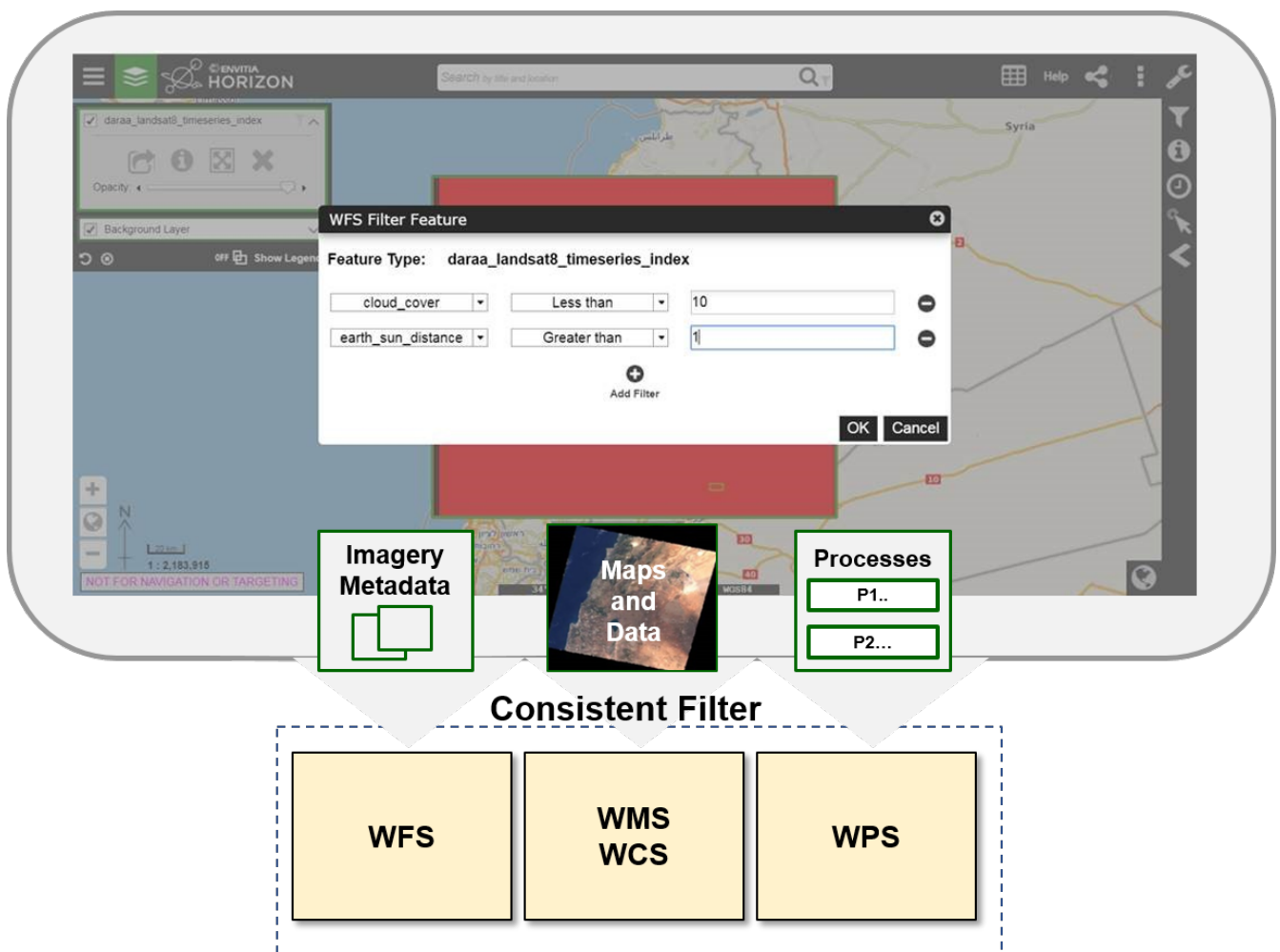


Figure 5. A key requirement was for users to easily filter a catalog of satellite imagery exposed as WFS and then exploit the imagery via other OGC services such as WMS, WCS and WPS, all using consistent filters across services.

Accordingly, the concept of 'profile' was proposed to hide the complexity of the queries needed to represent end user requirements in protocols like WFS, WMS, WCS for filtering and getting data.

The central concept in this approach is that data and/or service providers specify contexts that filter data automatically to satisfy certain needs without requiring end users to create such filters/queries themselves. Obviously, a certain level of trust between the data/service providers and the end users

must be in place - as the latter will have to trust the former as far as the content of the queries behind the profiles is concerned.

1.3. Component Implementation Design

Based on the requirements and demonstration scenario described above, an initial draft of the test platform architecture was developed during the project and is shown in the following diagram -

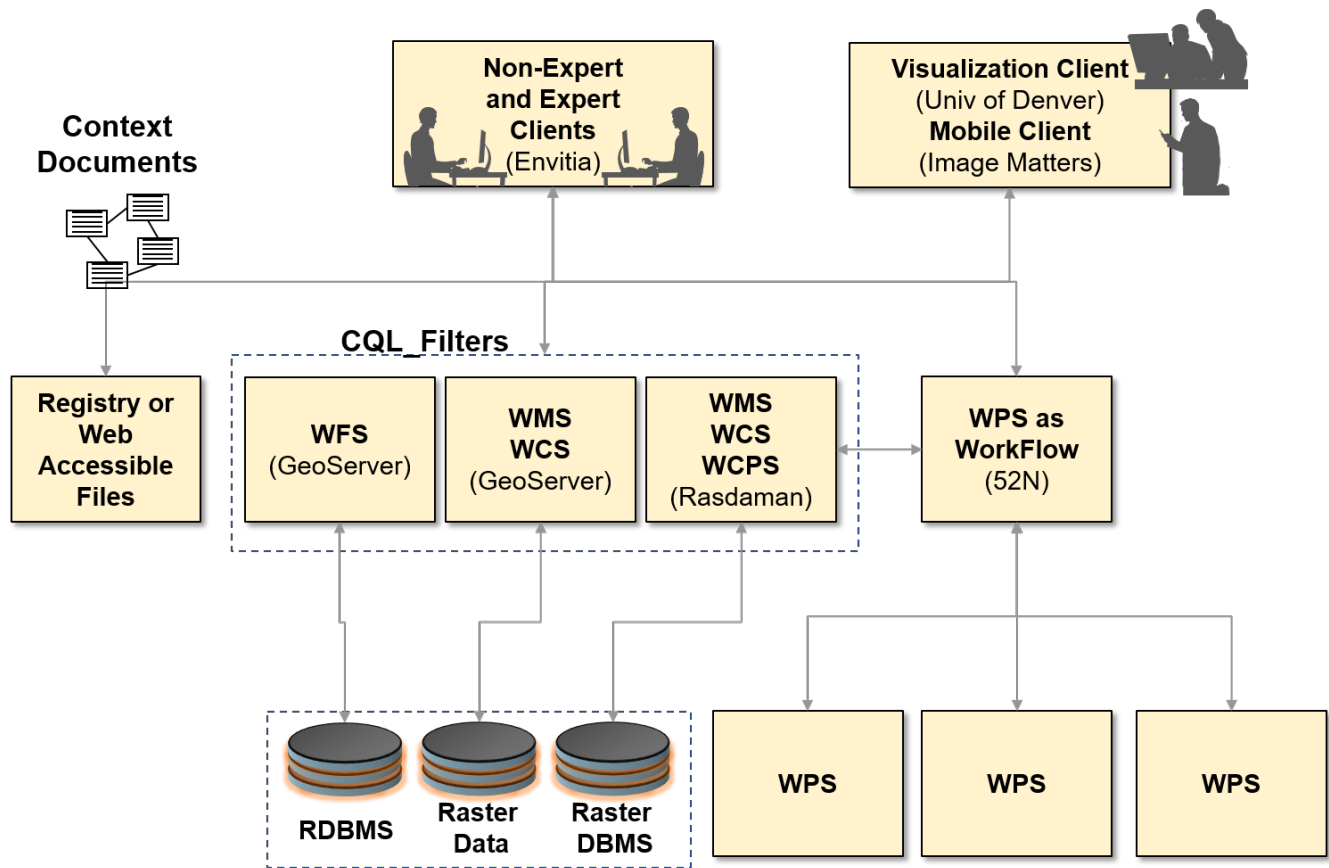


Figure 6. Testing architecture for the Fit for Purpose effort.

The platform architecture shows an initial sequence of interactions between different users, filters, encodings, services and data. The testing architecture for this part of OGC Testbed 13 was configured using a combination of the following components -

- **Filters and Encodings** - Multiple OGC encodings to support data exchange in the platform were developed and demonstrated during Testbed 13. These included filters on the WFS service that help access imagery metadata, but can also be applied to WMS and WCS for when extracting data, especially satellite imagery. JSON Context documents were used to capture collections of services for easy access to information on the platform and initiate processes on WPS such as conflation. Geography Mark Up Language (GML) was used for geospatial feature data as well GeoPackages that provided SQLite containers for geospatial feature and imagery data for use by mobile clients.
- **Client Applications** - Several types of client application components were exercised including two clients with new capability for Fit for Purpose filters and encodings to make it easy for 'non-geospatial experts to get quick answers to questions (NonExpert and Expert applications provided by Envitia). Other clients that focused on the ability to request and visualize data and from the platform were developed and demonstrated during Testbed 13. These included a

visualization client for use in a relief operations center (provided by the University of Denver), mobile clients for use in the field (provided by Image Matters).

- **Data and Services** - Sources of aerial or satellite imagery, available from commercial and government organizations, were used and represented a valuable source to support a wide variety of uses including visualization, infrastructure, population and land-use assessment. Government and private industry sources of feature data were identified and supported effective and efficient support in population centers and during mass migrations. Such geospatial data sources included transportation networks, locations and capacities of key facilities, food supplies and sources, and many others. This data was deployed on multiple OGC Web Services including WMS, WFS and WCS from GeoSolutions, WMS and WCS from Rasdaman and WPS from 52North.

The platform components were tested in a series of Technology Integration Experiments (TIEs) and demonstrated in the context of the scenario described in previous section. Interoperability experiments conducted during FFP are outlined in the following TIE table:




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 7. Technology Integration Experiments.

Three types of potential platform users were assessed, and sample client applications from Testbed 13 are shown for reference -

- **Non-Expert Users** - These users are not geospatial experts, and do not know about satellite imagery or complex geospatial data sources. They just have questions to enter into the platform,

such as: ‘What are the safest routes from Zaatari refugee camp to Daraa.

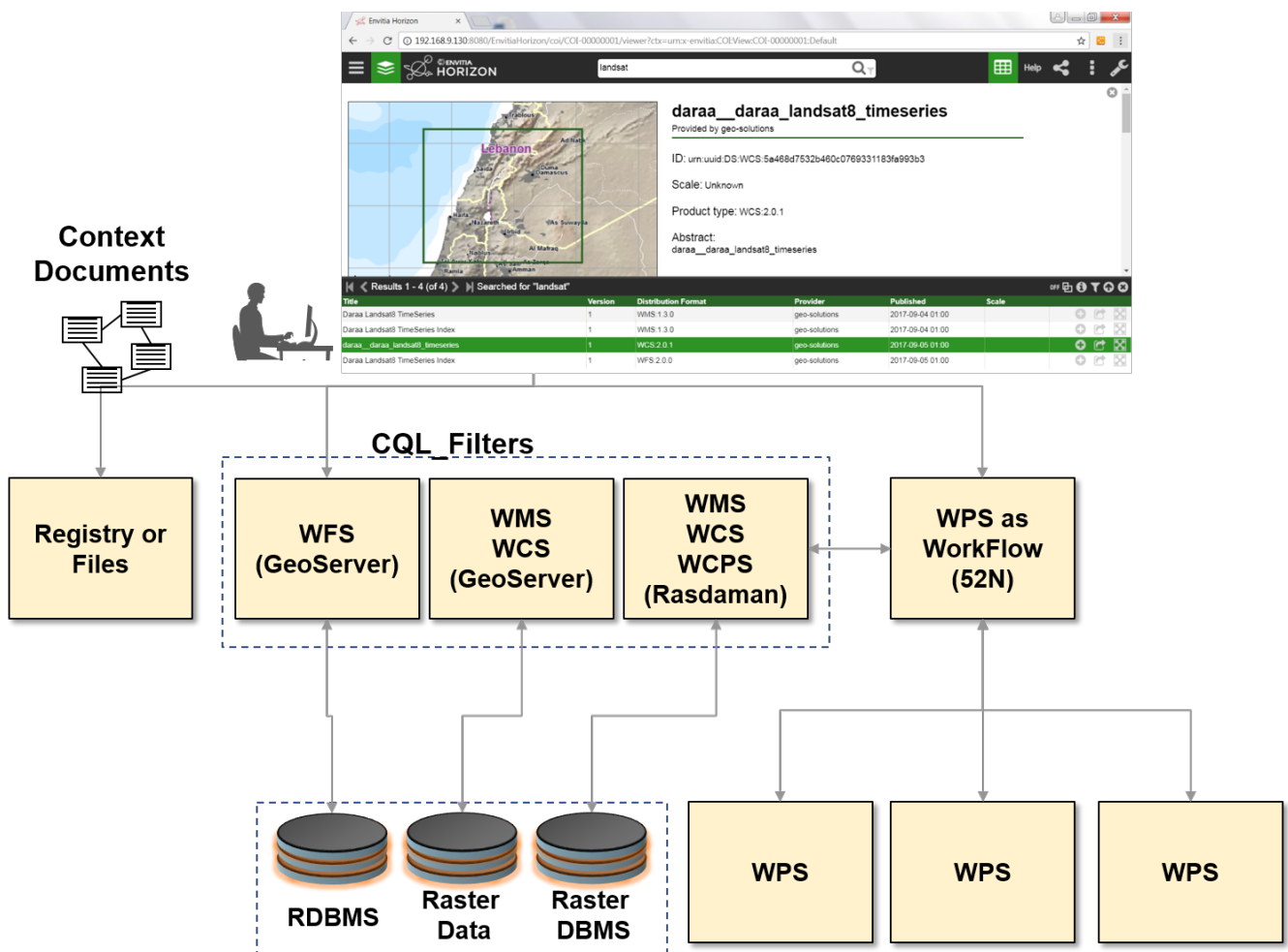


Figure 8. Client for non-expert users.

- **Expert Users** - These users are geospatial experts and understand the OGC services, satellite imagery and complex data sources in the platform. They set up the profiles and OGC Context Documents that provide help for Non-Expert Users.

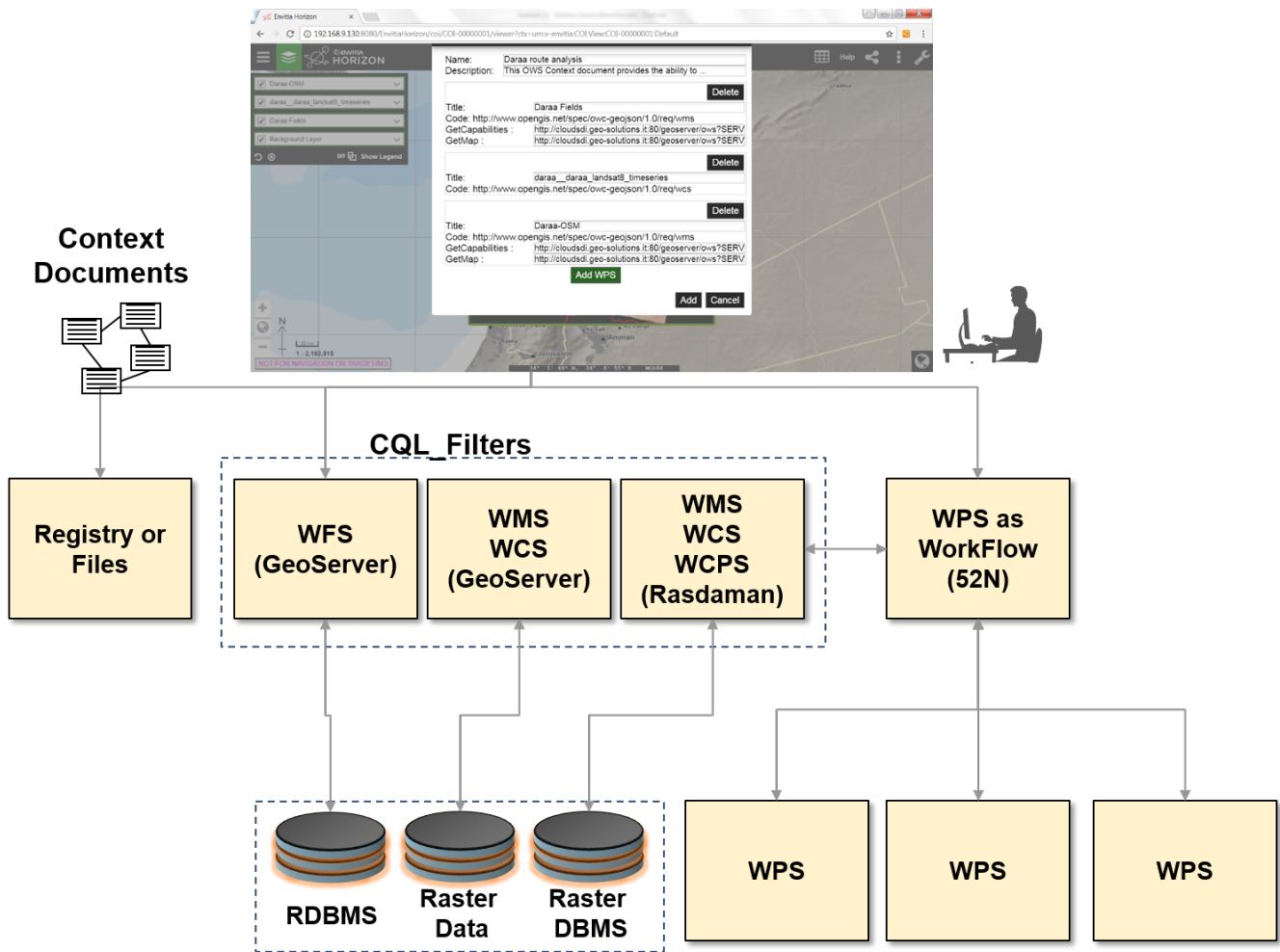


Figure 9. Client for expert users.

- **Visualization Users** – These users access the platform from simulated Relief Operations Centers, and have at least a laptop, client software and reliable internet connections.

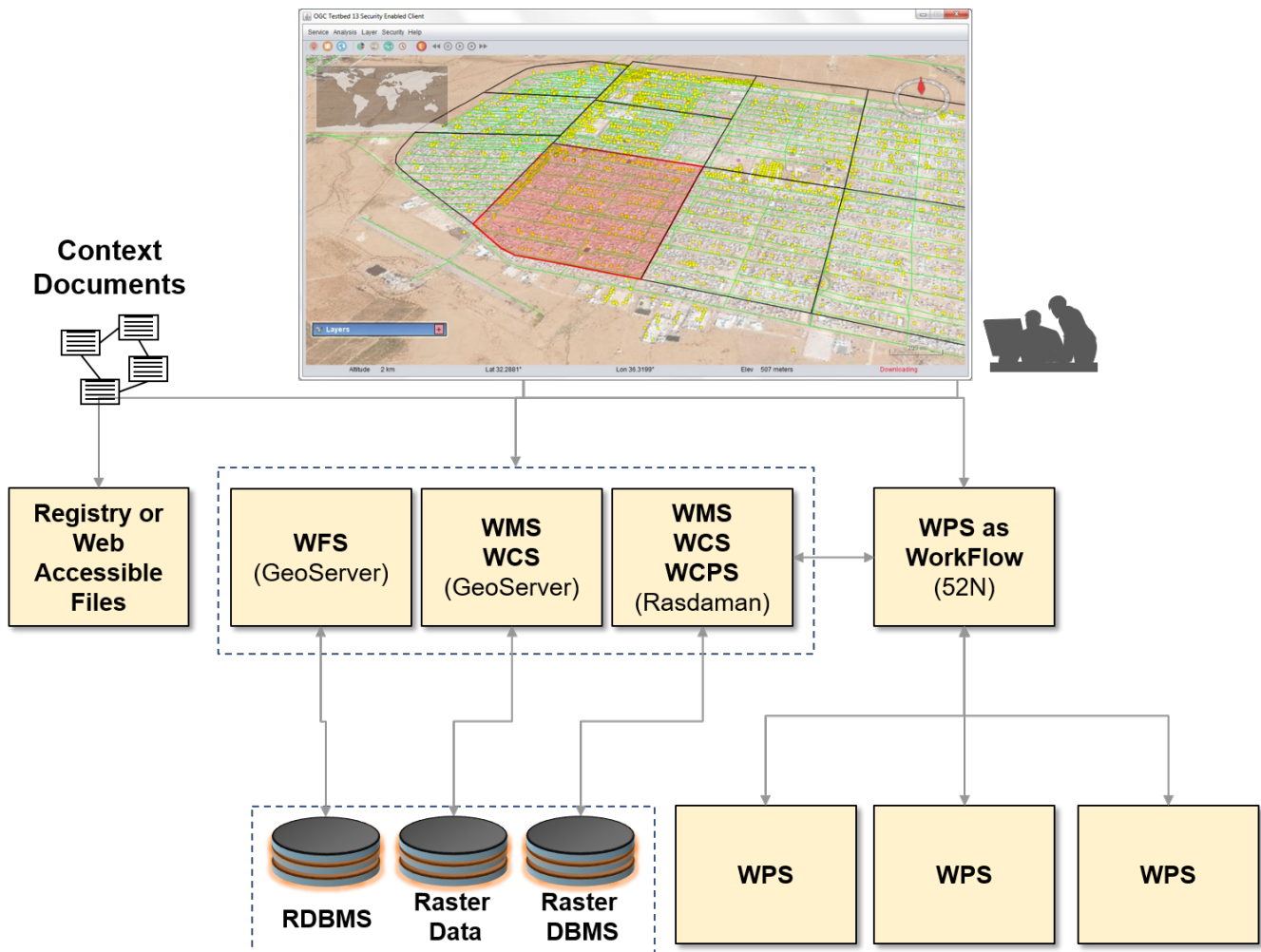


Figure 10. Visualization client.

The following sections describe the OGC standards that were assessed during platform filter and encoding design.

1.4. Prior-After Comparison

The following existing OGC standards were assessed to meet FFP requirements. In particular, the need to capture the concept of a profile in a way that allowed it to be applied between multiple OGC Services in the test platform including WFS, WMS, WCS and WPS was a key design consideration.

- WFS and Stored Query Support
- OGC Web Services (OWS) Context Documents
- Filtering data with WFS, WMS, WCS
- Role of a Client as a Mediator
- Role of the Workflow and WPS
- OGC Filtering in GeoServer
- Relationship with Stored Queries

Given the assessments described above, the Testbed 13 Component Implementation Design for FFP proposed to have an OWS Context document providing links with Common Query Language (CQL) expressions for filtering, along with ordering similarly using the Key Value Pair (KVP) syntax, be it

as native (WFS) or vendor extension (WMS/WCS).

A detailed description of these assessments is included in the [Experiments](#) section.

1.5. Technology Integration Experiments

This section describes the key implementations related to CQL_Filters and Context Documents as a mechanism for a 'profile' that can be applied between multiple OGC Services in the test platform including WMS, WCS and WPS.

The testing architecture for FFP in OGC Testbed 13 exercised the encodings and filters, services and data and client applications presented below in the following manner:

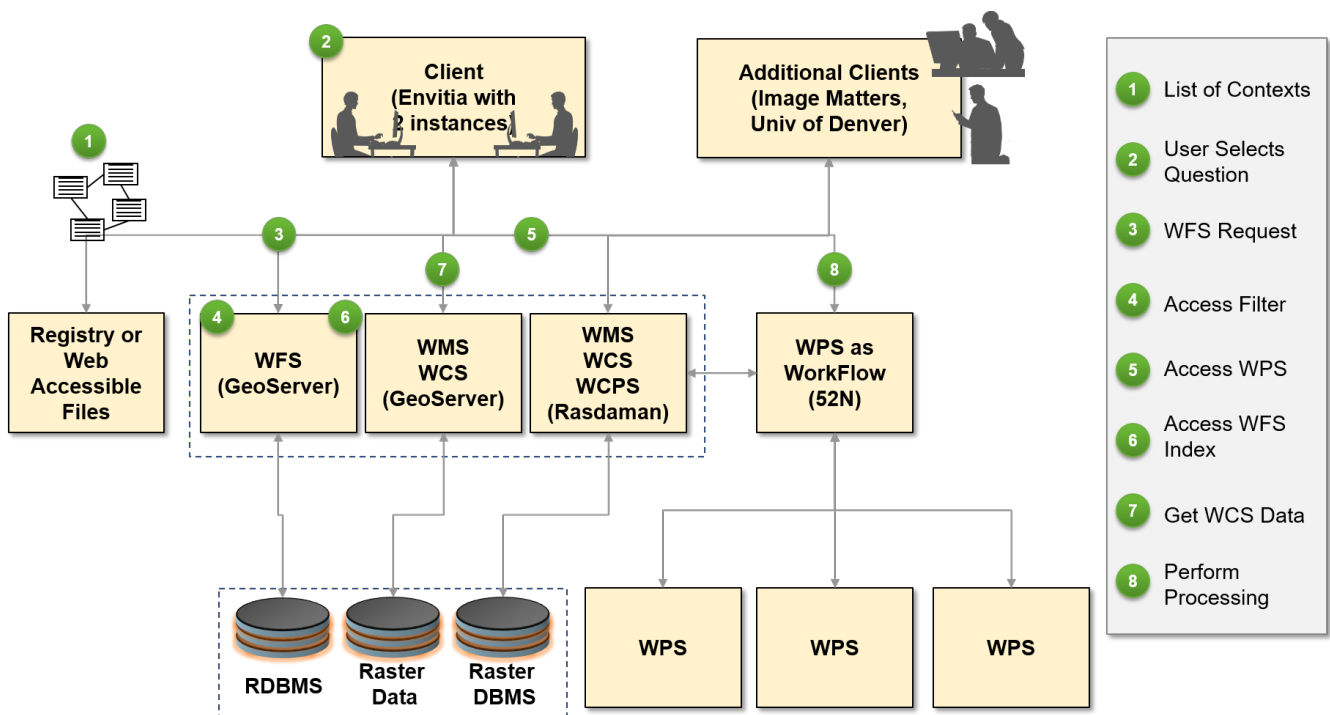


Figure 11. Technology Integration Experiments - sequence of interactions.

1.5.1. Filters and Encodings

As discussed, the Testbed 13 Component Implementation Design for FFP proposed to have an OWS Context document providing links with CQL expressions for filtering, along with ordering similarly using the KVP syntax, be it as native (WFS) or vendor extension (WMS/WCS).

The Zaatari sample context document in JSON can be found in Annex A under the Title: Zaatari Refugee Camp route analysis. This document includes references to a catalog of imagery as WFS and embeds CQL_Filters to help access imagery on WMS and WCS as needed.

The second sample Context Document in Annex A is under the Title: NASA Black Marble Global Timeseries for 2012-2016. National Aeronautics and Space Administration (NASA) Black Marble is a global view of the Earth's city lights acquired by Suomi National Polar Orbiting (Suomi NPP) satellite, and this context document references this data exposed as WCS and WMS.

The third OWS Context Document in Annex A is for Daraa and includes the ability to conflate existing road data sets, providing information for route analysis. In this document there is one WPS

request specifying two input data sets - United Nations (UN) roads data and OpenStreetMap (OSM) roads data.

The encodings and filters were assessed in a series of experiments as shown in the TIE table below.




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 12. Technology Integration Experiments - filters and encodings.

Graphical representations of the results when these context documents are executed are shown in the Client Applications section.

1.5.2. Services and Data

This section discusses the FFP approach implemented in services and data provided by Testbed 13 participants.

In the scenario for this part of Testbed 13, large numbers of people have been displaced from the Daraa region of Syria to the Zaatari refugee camp in Jordan due to ongoing conflict. As the conflict ends 'de-escalation zones' are established by major powers and plans are made to return displaced people from refugee camps to the region. Understanding the situation on the ground and the infrastructure, as well as transporting these people from refugee camps into a former conflict zone is a major challenge for relief agencies. To accomplish this task they must understand the environment and transportation infrastructure in the region between Zaatari refugee camp and the Daraa region.

In support of this challenge, satellite imagery from commercial and government organizations was deployed to support situational awareness and analysis for all types of users. Government and private industry sources of feature data were also identified and supported visualization and analysis of transportation networks, locations and capacities of key facilities, food supplies and sources, and many other key geographic features. This data was deployed on multiple OGC Web Services including WMS, WFS and WCS from GeoSolutions, WMS and WCS from Rasdaman and WPS from 52North.

The components were assessed in a series of experiments as shown in the TIE table below.




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 13. Technology Integration Experiments - services and data.

Data Deployed on Geoserver - Daraa and Zataari

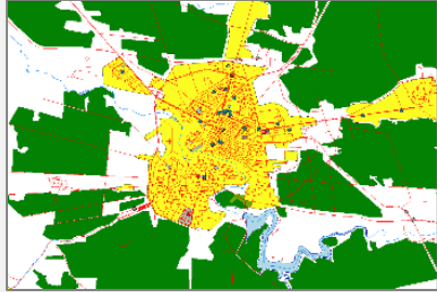
Geospatial data for the Daraa region was deployed using a combination of existing OGC standards, and new techniques to make access for users and client applications more dynamic and useful, and to support analysis of multi-regional / multi-national operations and messaging related to the displacement and mass movement of populations in response to conflict.

The following two graphics summarize the data deployed in this part of the information exchange platform.

Component: WFS with CQL_Filter capacity

Daraa Data Deployed: NGA TDS OSM data as WFS including InformationPnt, RecreationSrf, StructurePnt, StructureSrf, TransportationGroundCrv, TransportationGroundSrf, UtilityInfrastructureSrf, osm_points. Shapefiles for cultivated areas coming from DigitalGlobe.

Zaatari Data Deployed: WFS Service layer for the catalog of WorldView imagery over Zaatari with search parameters for time, cloud_cover, sun_azimuth, sun_elevation etc. Time values span from 2013 to 2017.



Daraa NSG TDS OSM



WFS Layers w/Imagery Metadata

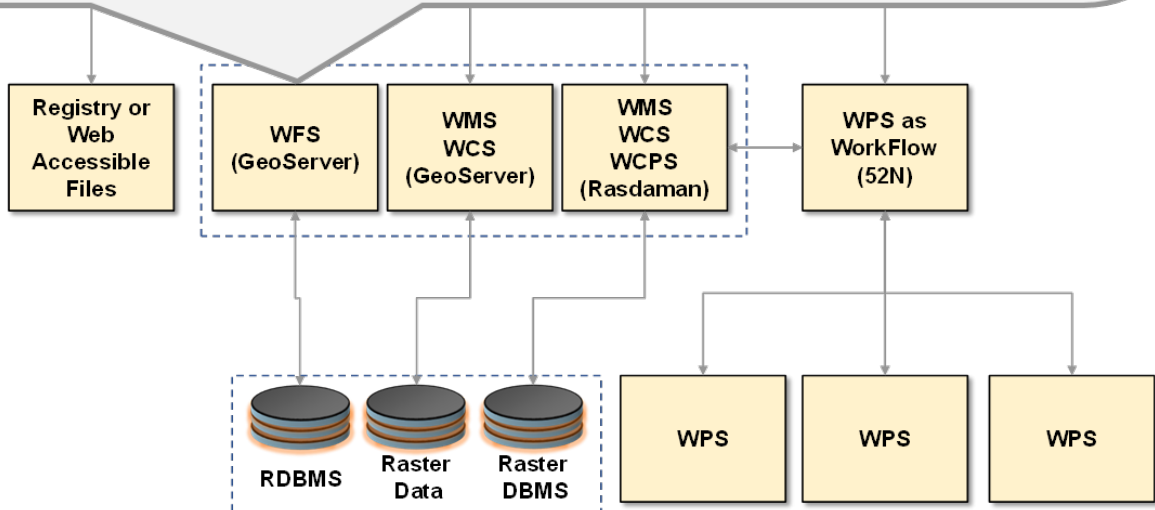


Figure 14. Data deployed on Geoserver WFS.

Component: WMS and WCS with CQL_Filter capacity

Zaatari and Daraa Data Deployed: WorldView (50cm resolution) scenes for Daraa, full 8 bands 16 bits which are exposed as WMS and WCS. Data for WMS and WCS may be accessed using same CQL_Filters as WFS. Landsat 8 imagery covering a large area (with 30m resolution), including Daraa and Zaatari refugee camps, spanning from March 2017 to June 2017. Black Marble data showing changes in lights at night (2012-20126).

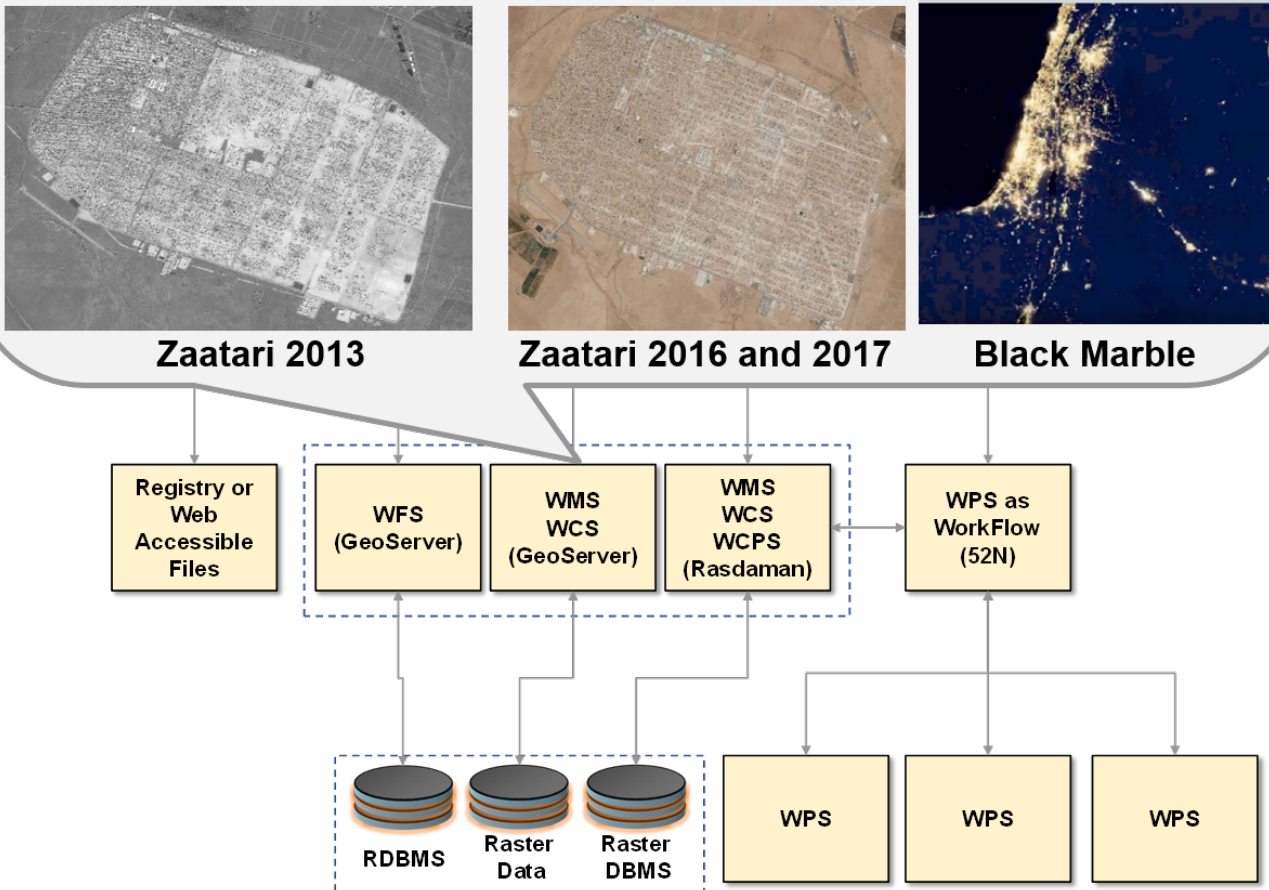


Figure 15. Data deployed on Geoserver WMS and WCS.

1.5.3. Client Applications

This section discusses the FFP approach as implemented in Client Applications provided by Testbed 13 participants.

The FFP Component Implementation Design for Client Applications recognizes two primary levels of users:

- **Non-Expert Users** - These users are not geospatial experts, and do not know about satellite imagery or complex geospatial data sources. They just have questions to enter into the platform, such as, "What are the safest routes from Zaatari refugee camp to Daraa?".
- **Expert Users** - These users are geospatial experts and understand the OGC services, satellite imagery and complex data sources in the platform. They set up the CQL-Filters and Context Documents that provide help for Non-Expert Users.

In addition, a third client was tested showing visualizations of the data deployed for Daraa and Zaatari:

- **Visualization Client** – These users access the platform from simulated Relief Operations Centers, and have at least a laptop, client software and reliable internet connections.

The client applications tested the filters and encodings in the platform as outlined in the TIE table.




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 16. Technology Integration Experiments - client applications.

Client for Non-Experts

In the demonstration scenario clients accessed the platform using client applications configured for users who are not geospatial experts, and do not know about satellite imagery or complex geospatial data sources. They just have questions to enter into the platform, such as, "What are the best routes from Zaatari refugee camp to Daraa?".

Examples of this client from Envitia are shown below:

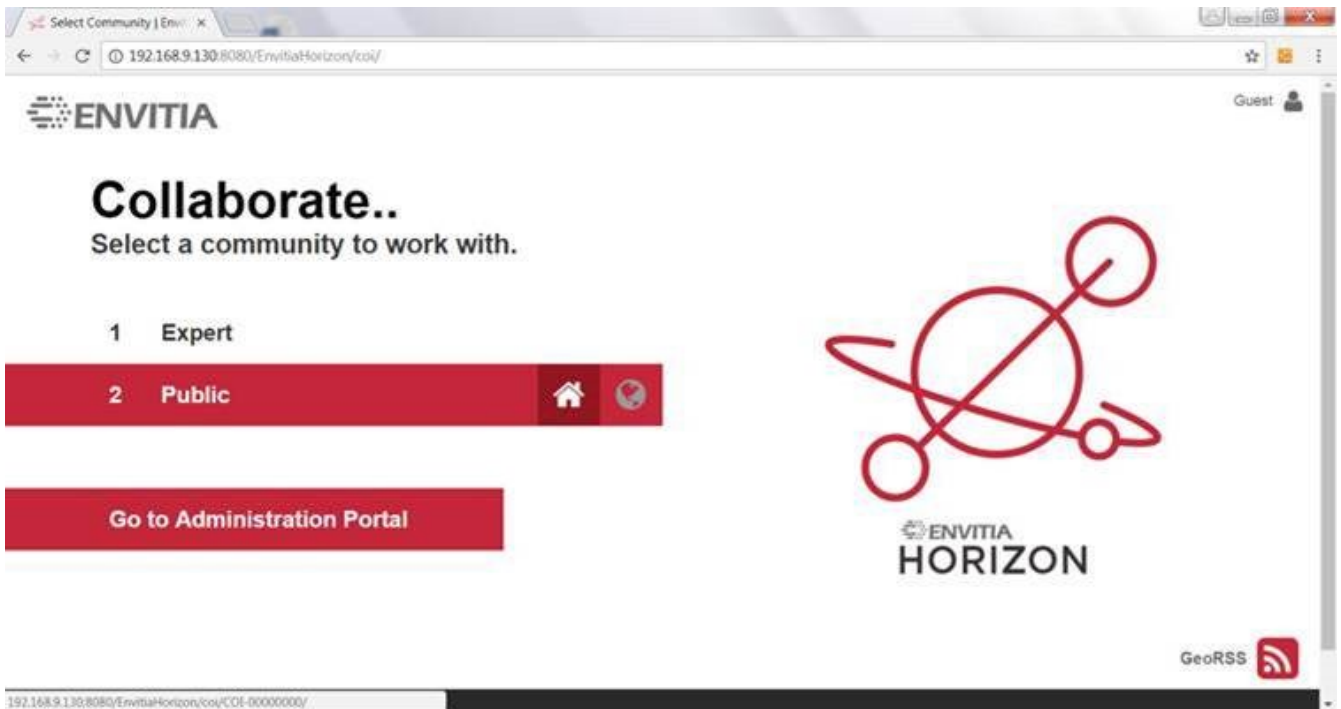


Figure 17. Client for non-experts - Public view.



Figure 18. Client for non-experts - User interface.

This application can use complex filters (CQL_Filter) and Context documents that capture collections of services for access to information on the platform - with the complexity hidden from non-expert users. All they have to do is search for OWS Context Documents providing a specific analytical service. For example, they may type, "What is the fastest route out of Zaatari?" They are then presented results, and assess the results via the abstracts. The application then automatically fires all the requests in the selected OWS Context document and visualizes them on screen.

Examples of this sequence are shown below:

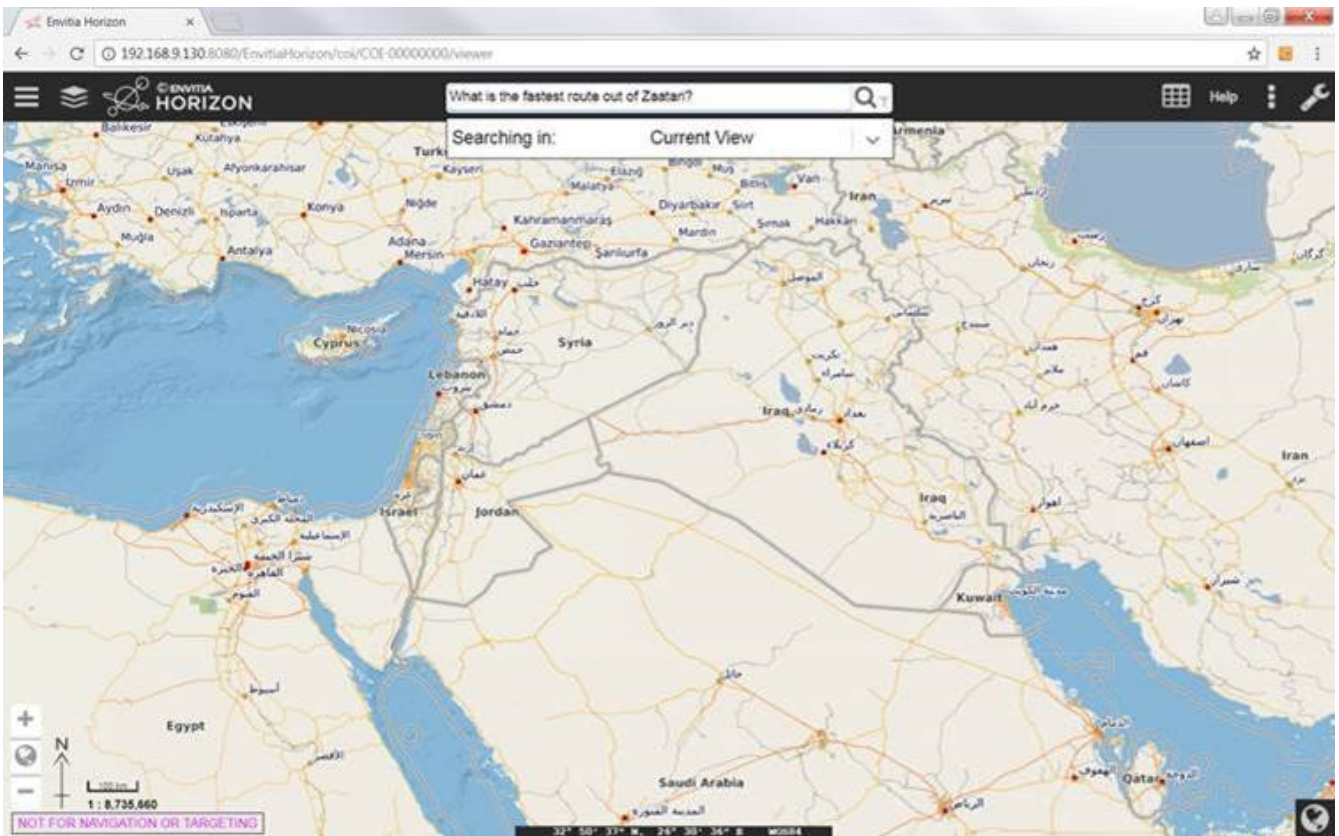


Figure 19. Entering a question to search for information about Zaatari.

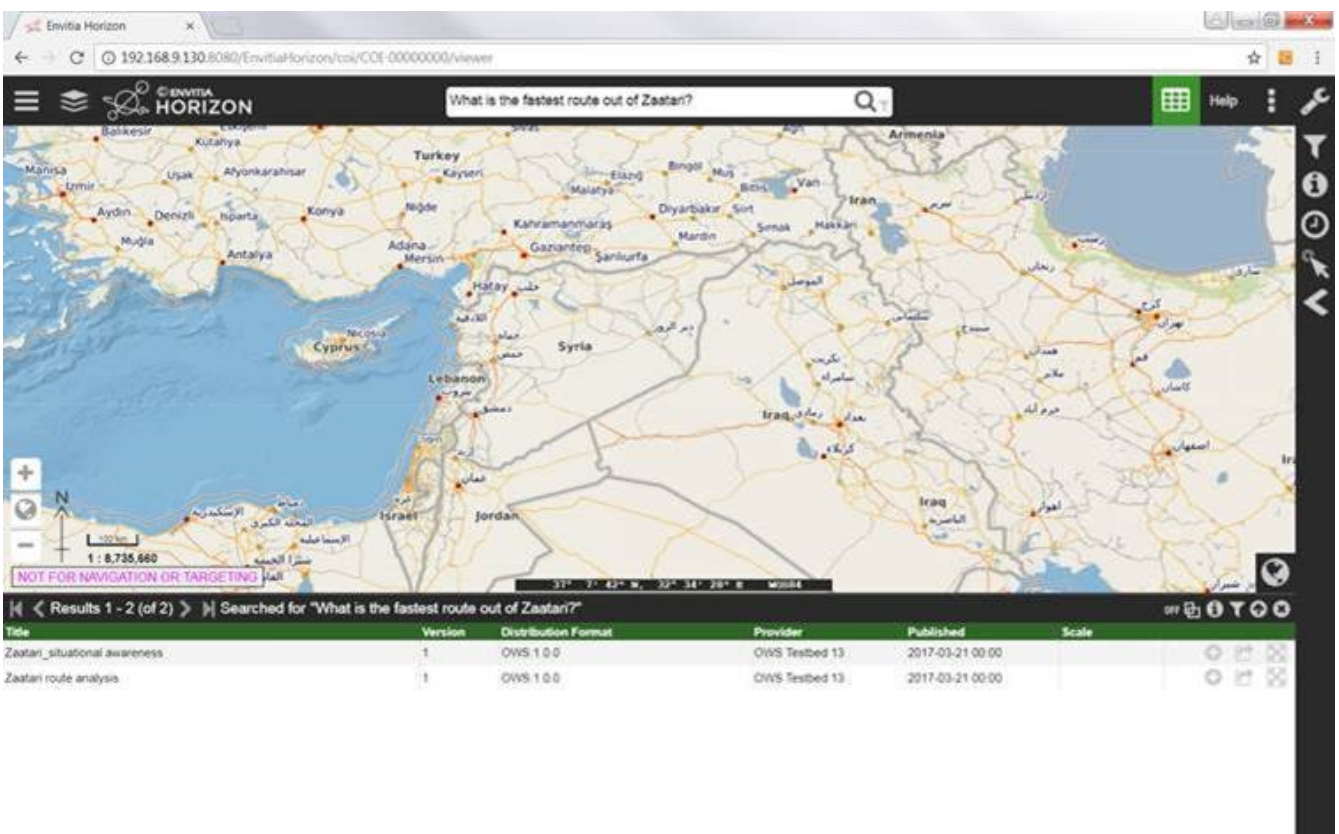


Figure 20. Results presented in response to a question.

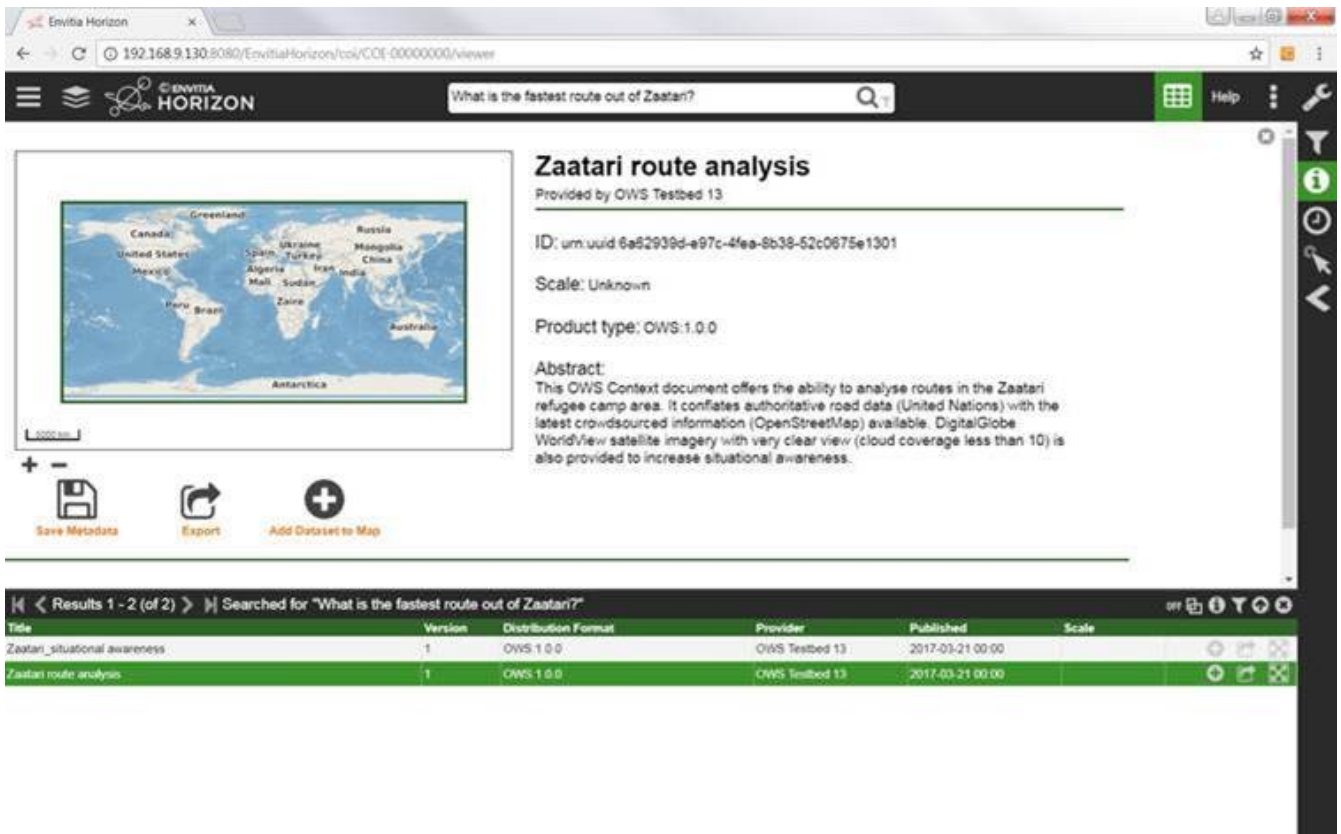


Figure 21. Results accessed via the abstracts.

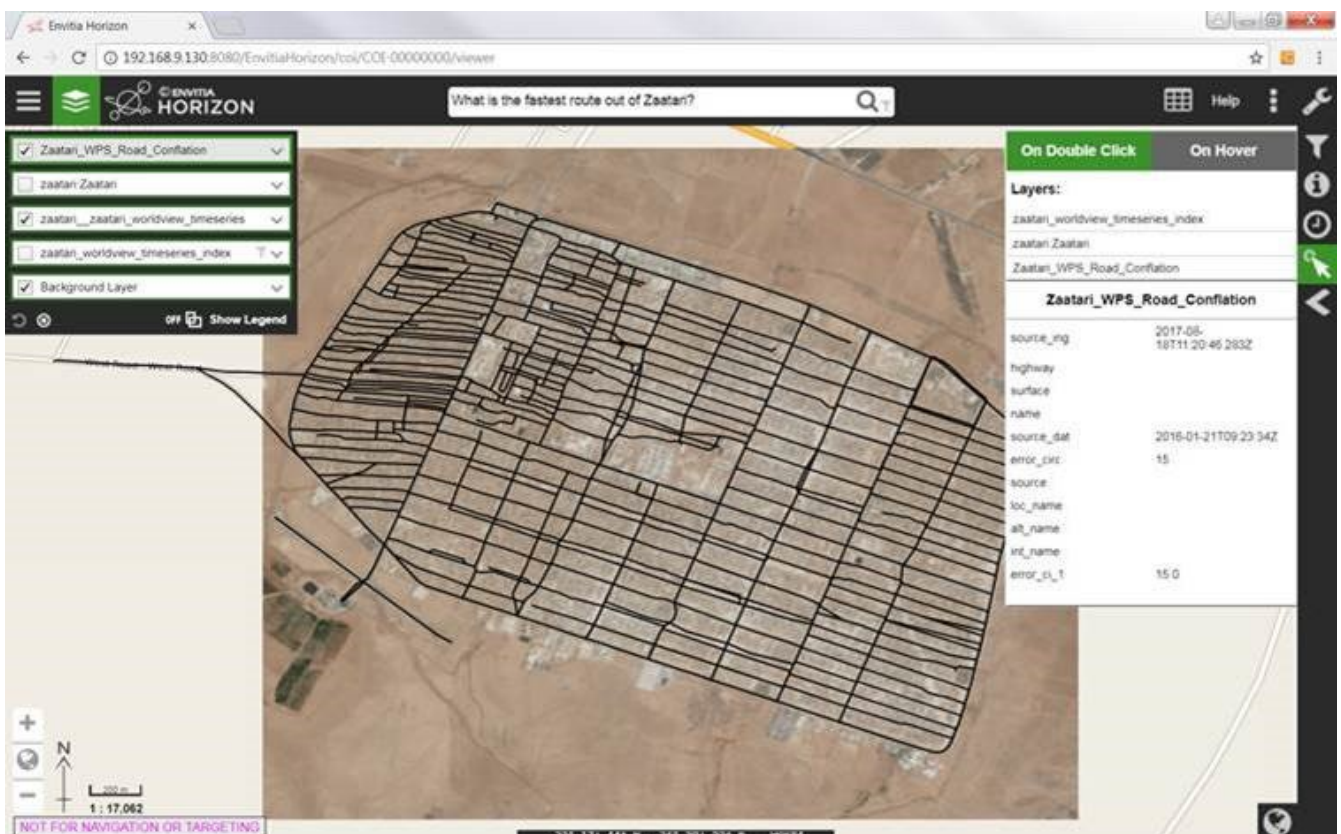


Figure 22. Application automatically requests in the selected OWS Context document, and visualizes results on screen.

Client for Geospatial Experts

This section explains how expert users set up the encodings and filters used in the application for

non-experts.

First, the user logs into the 'Expert' capabilities of the Envitia application:

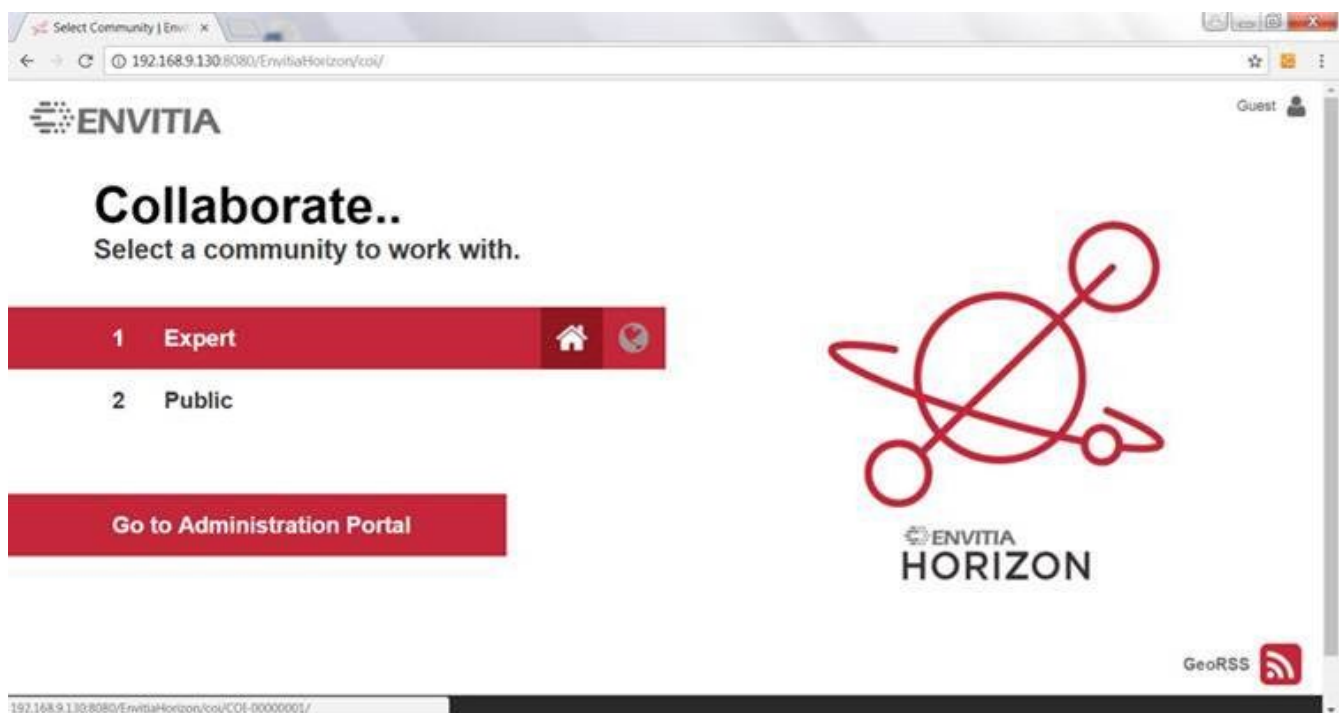


Figure 23. Client for experts - User login screen.

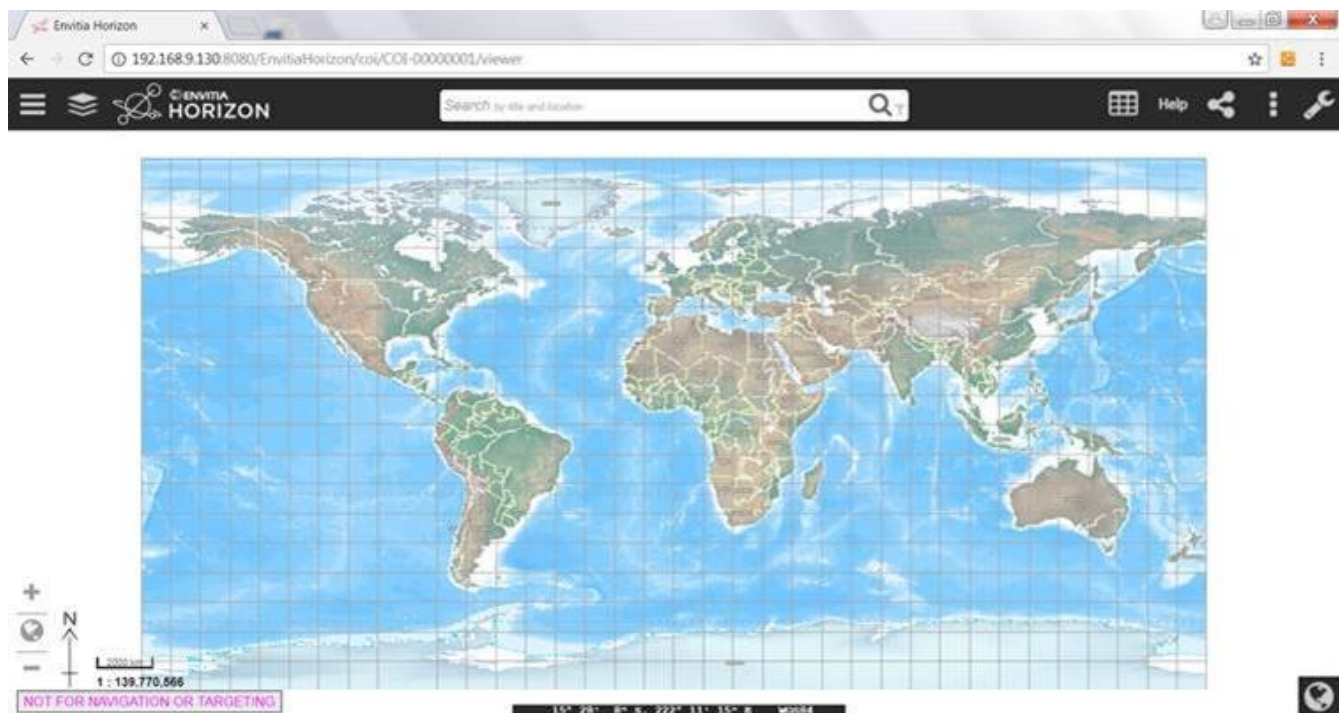


Figure 24. Client for experts - User interface.

They can then search for specific web services (e.g. WFS Daraa Landsat 8):

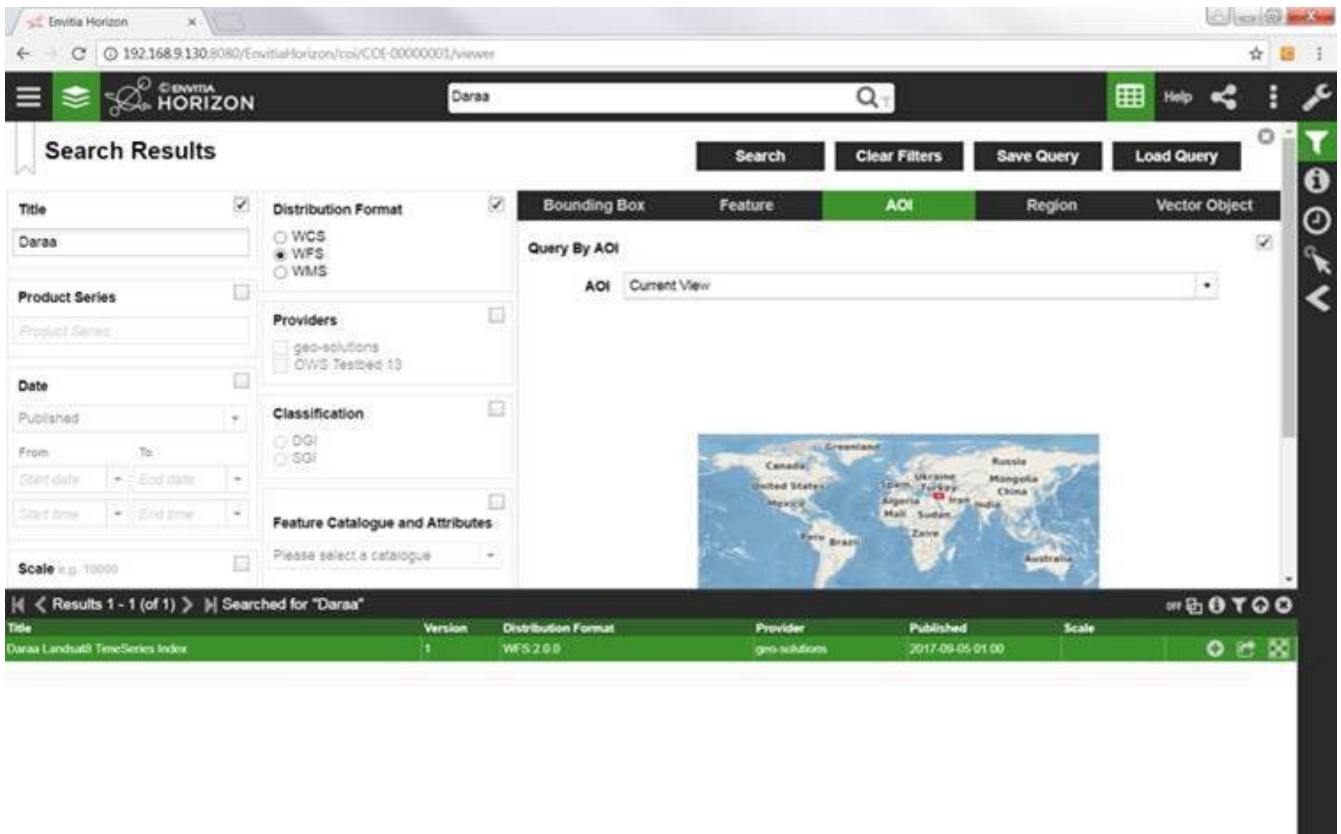


Figure 25. Searching for 'Daraa' and WFS.

They can then visualize the WFS Daraa Landsat 8 timeseries (bounding boxes and feature attributes):

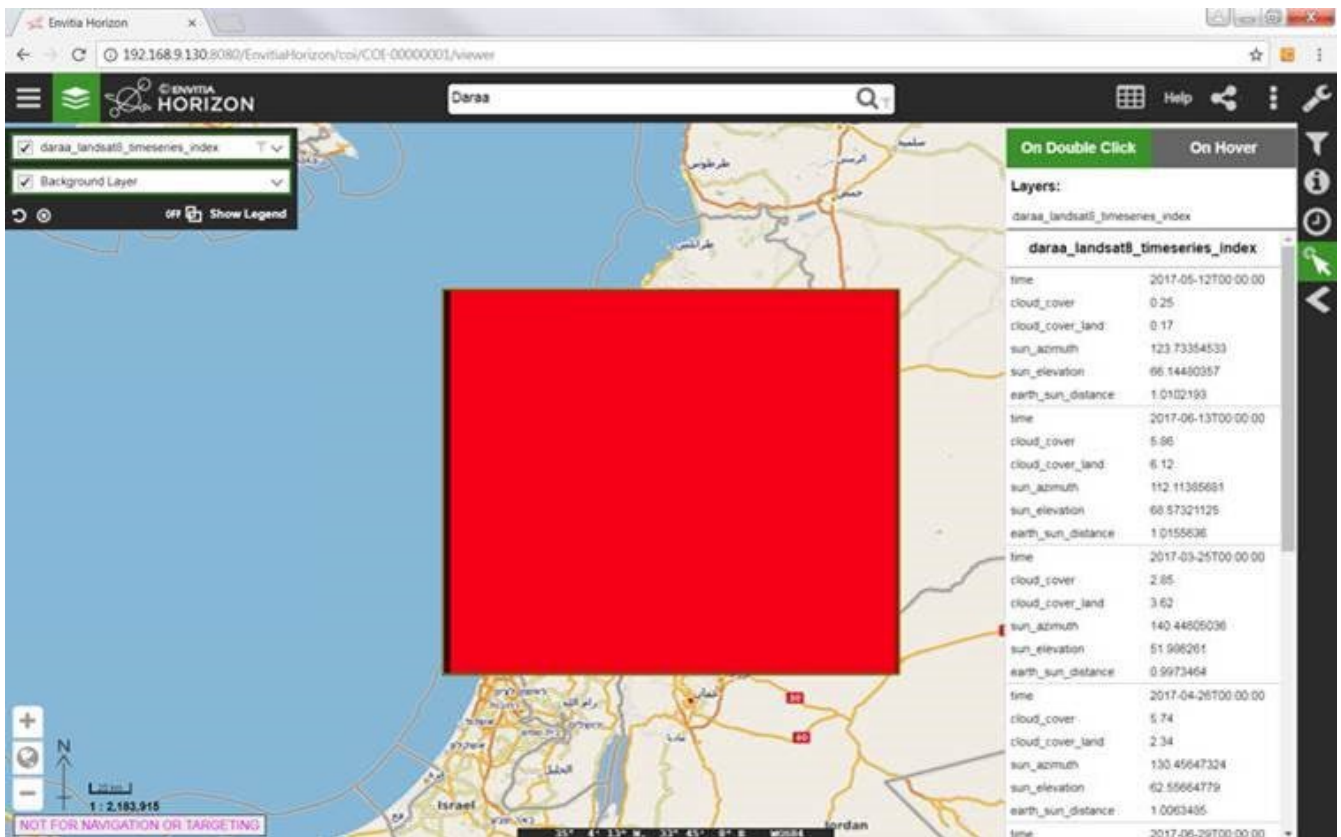


Figure 26. Visualizing metadata for Daraa Landsat timeseries from WFS.

Then they apply a CQL filter to access only features with `cloud_cover<10` and `earth_sun_distance<1`.

The first screenshot shows filter creation, the second the results of the filter, and then the CQL filter is automatically applied to GetFeature.

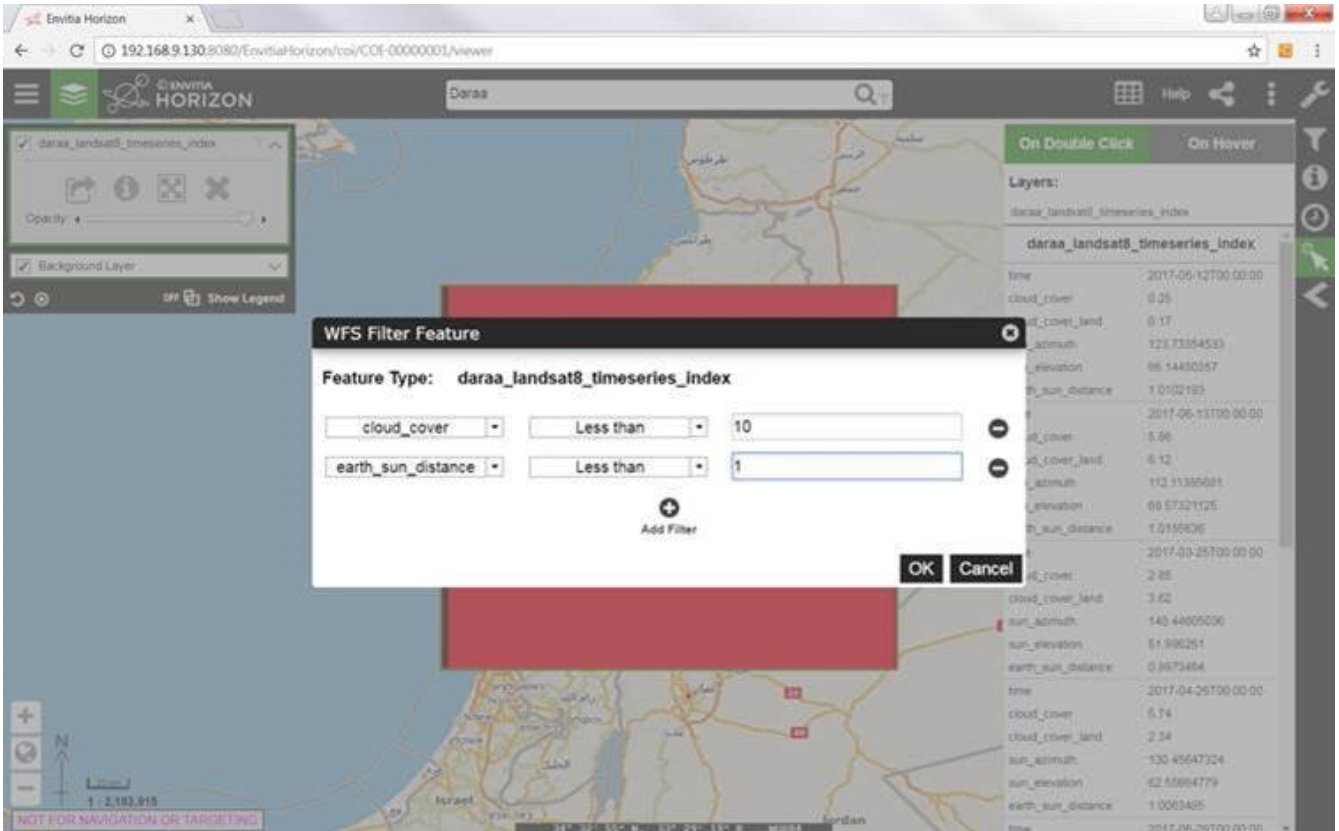


Figure 27. Applying a CQL filter to access only imagery metadata (features) with $cloud_cover < 10$ and $earth_sun_distance < 1$.

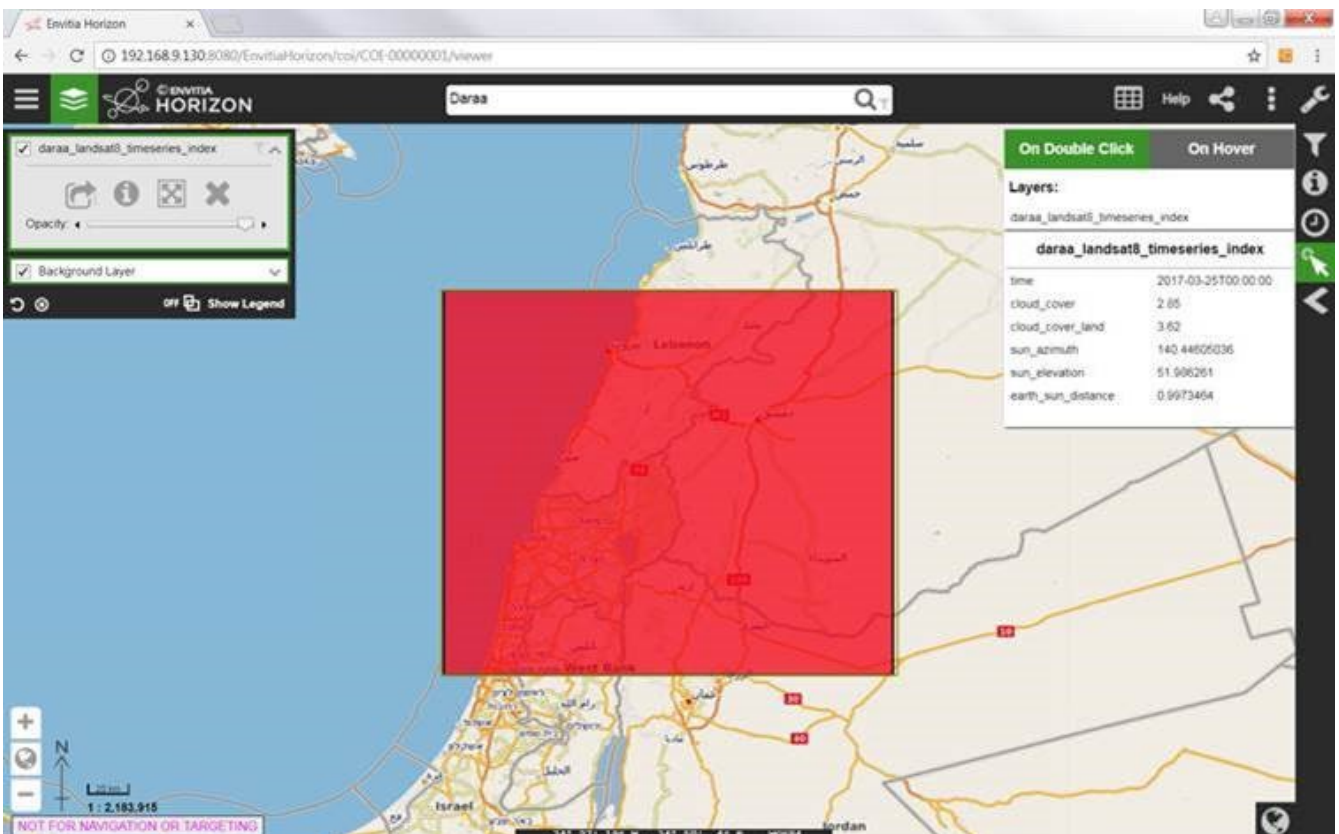


Figure 28. Reviewing results of the CQL filter.

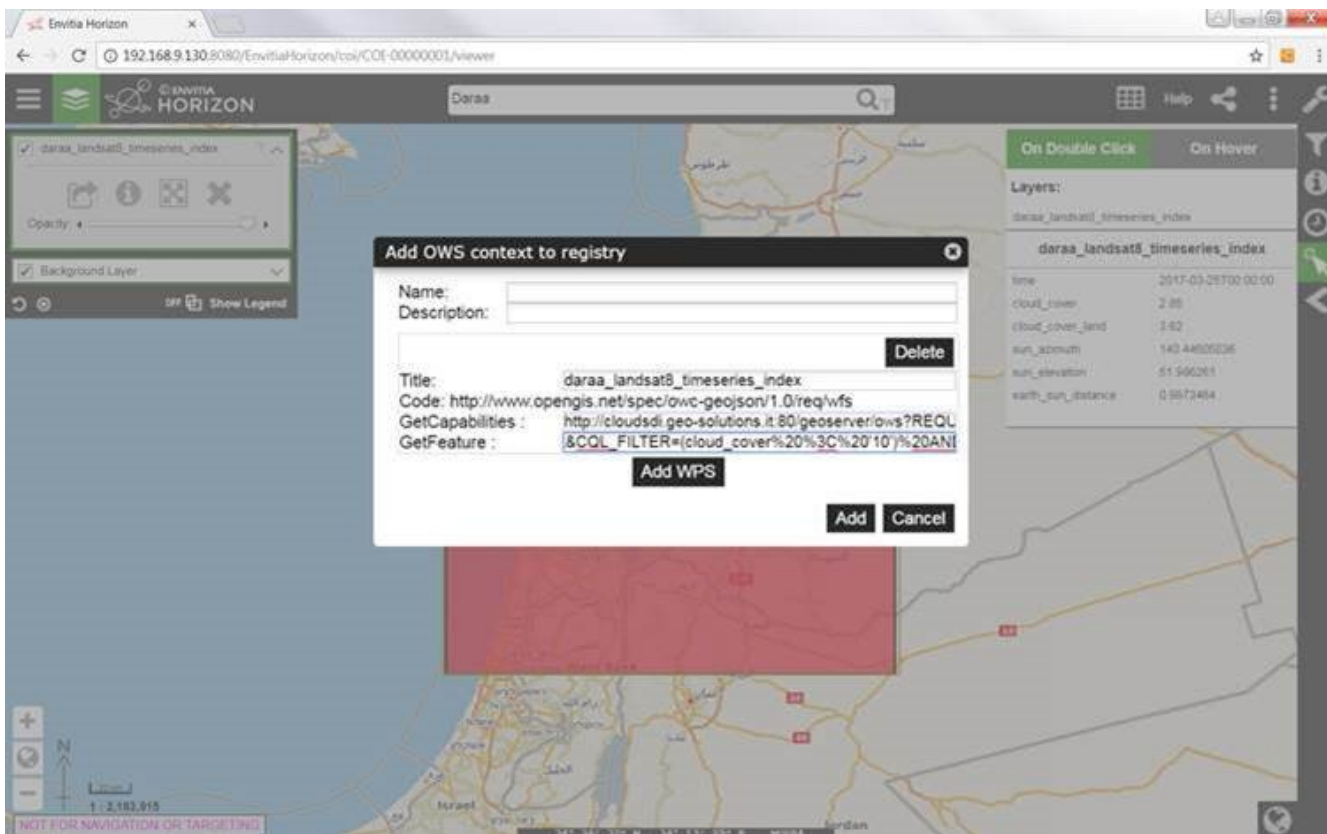


Figure 29. CQL filter automatically applied to GetFeature requests.

Below, the same CQL filter has been automatically applied to the related WMS (the only image satisfying that is 2017-03-25 00:00:00):

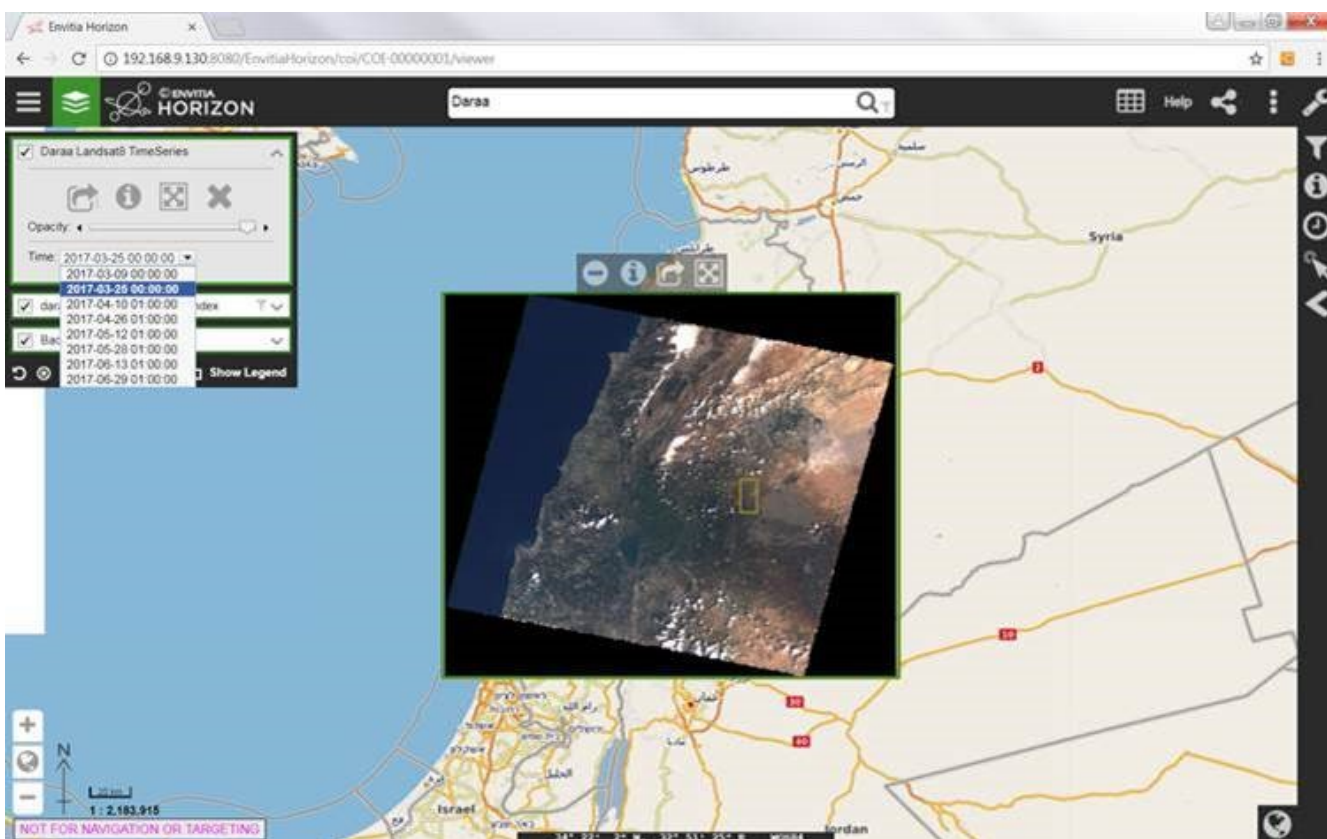


Figure 30. The same CQL filter used on imagery metadata WFS can be applied to the related WMS.

The client allows to access individual images at different timestamps of the time series via a slider

as shown below:

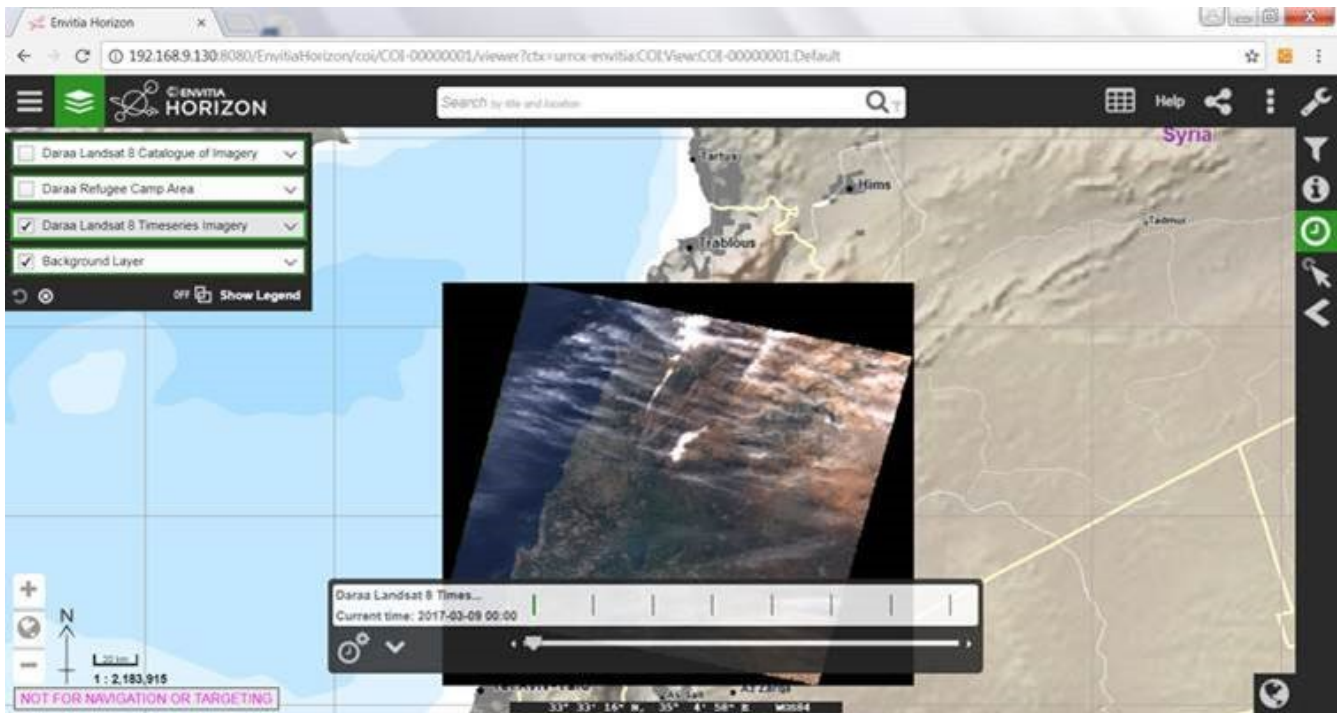


Figure 31. Individual images accessed at different timestamps of the time series via a slider.

They can search for specific web services (e.g. WCS Daraa Landsat 8 timeseries):

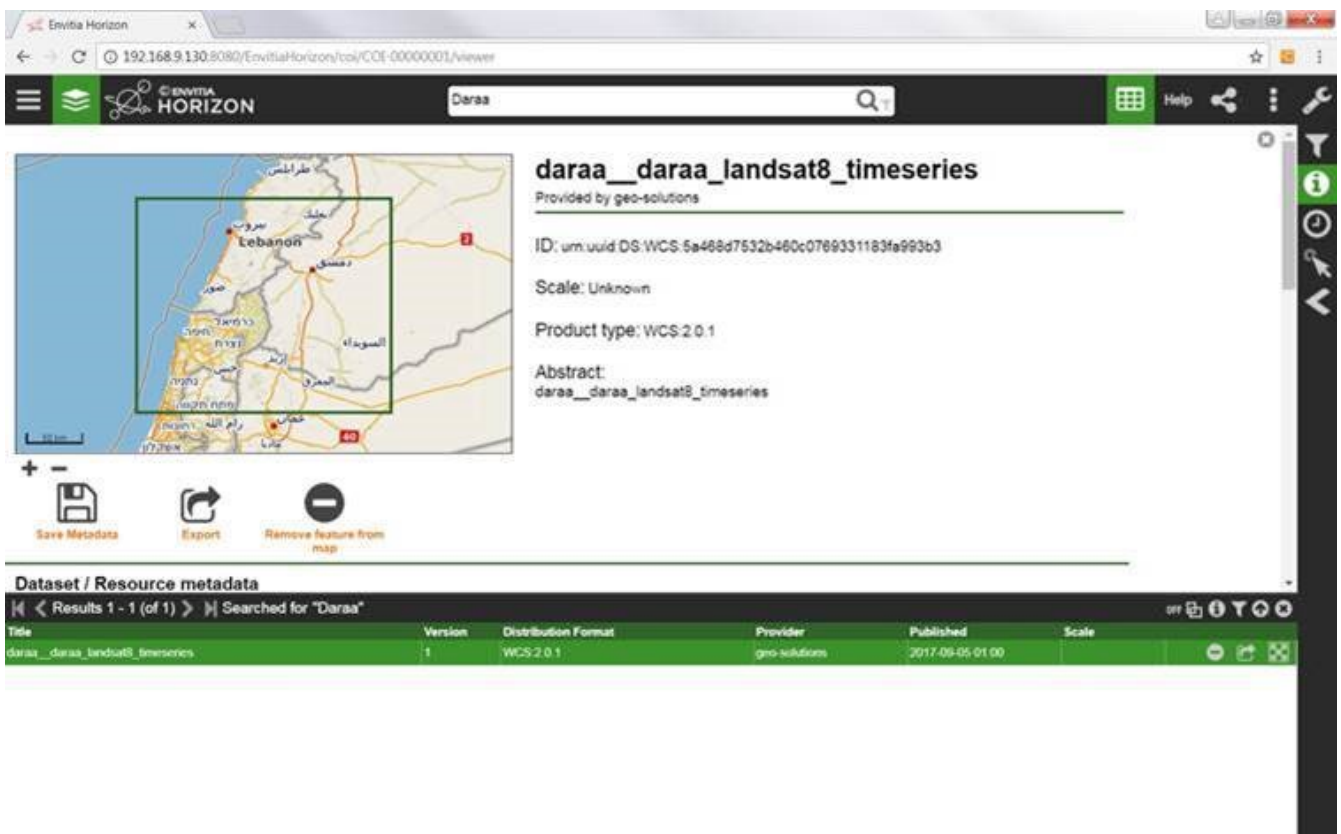


Figure 32. Searching for imagery data over Daara (WCS).

They can also subset data served by WCS using a lat/long filter. The screenshots below show the original, and then filtered using latitude and longitude:

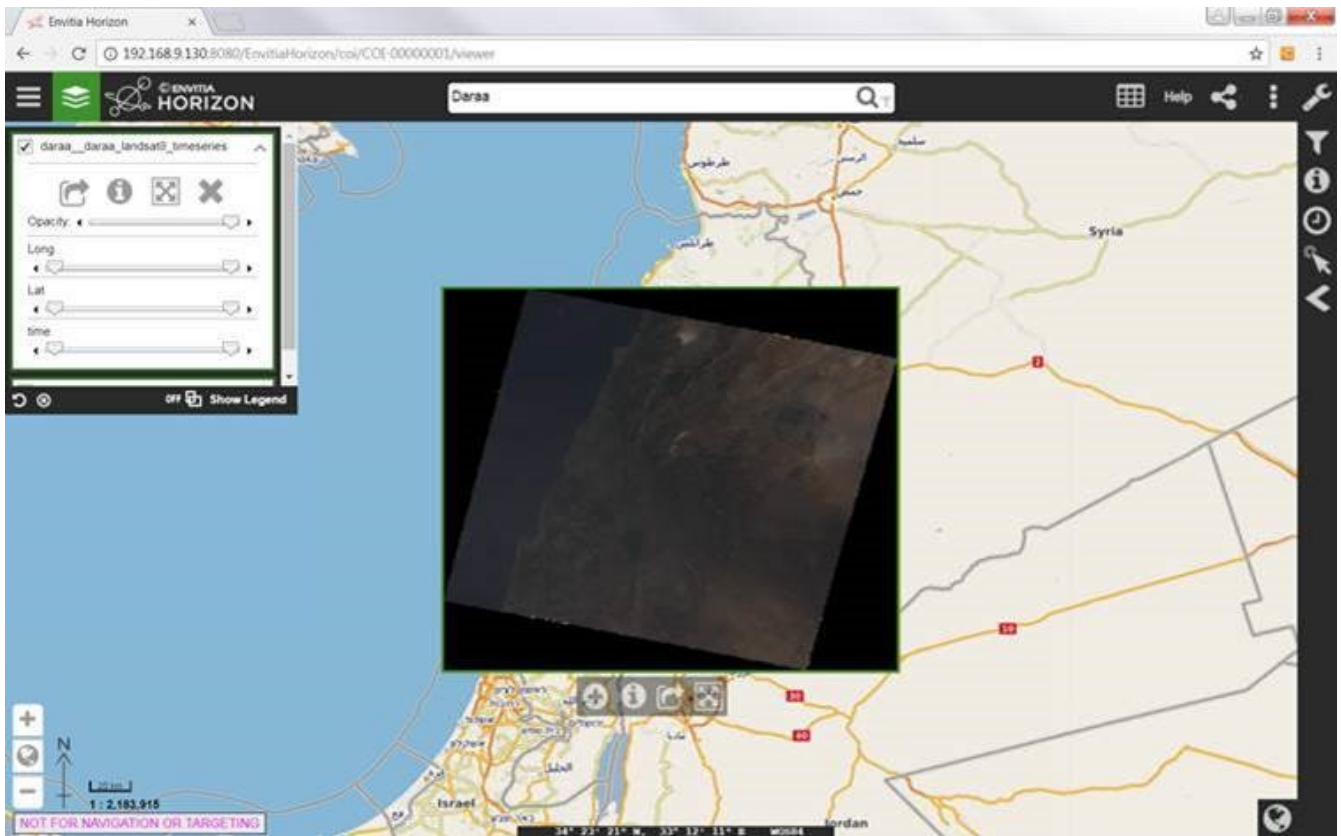


Figure 33. Unfiltered view of imagery data served by WCS.

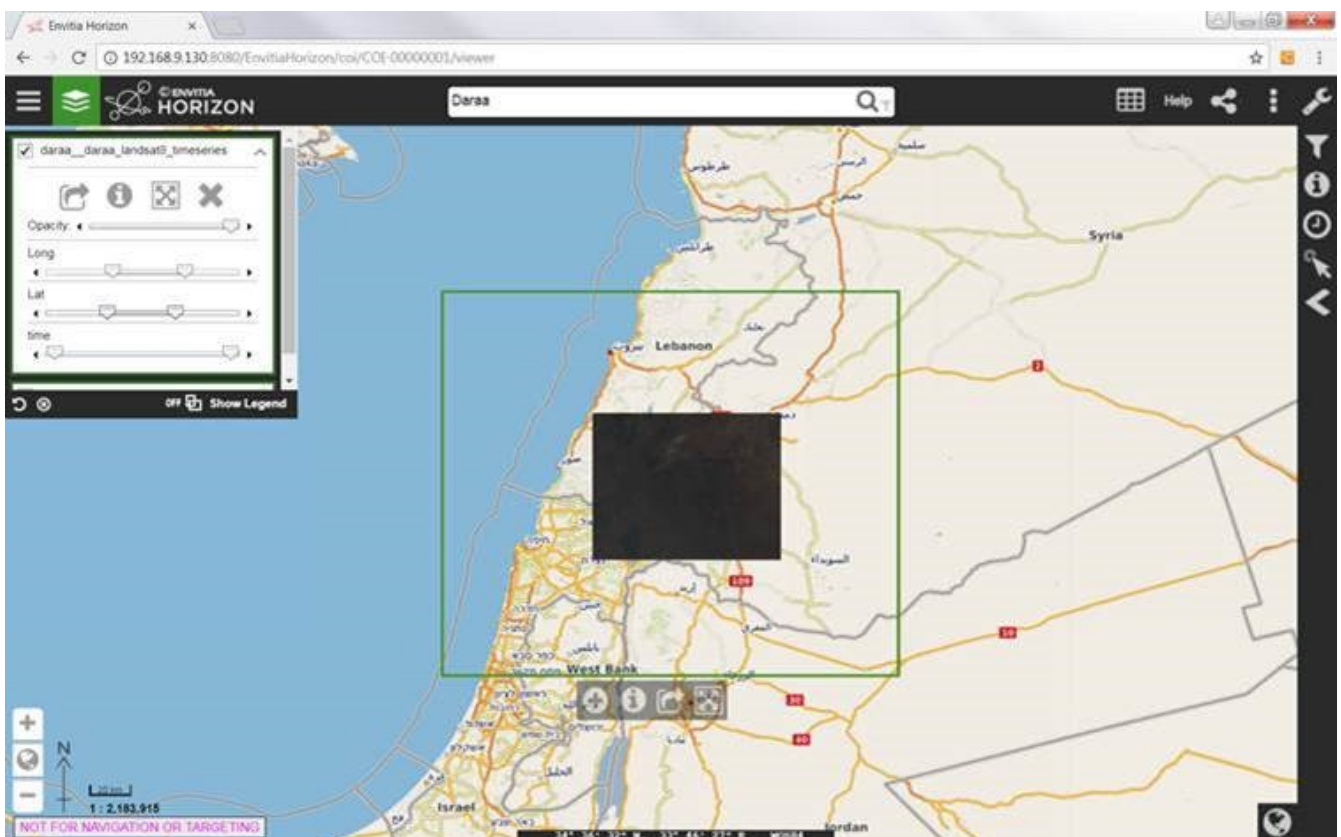


Figure 34. View of imagery data served by WCS after applying a lat/long filter.

The example below shows search for specific web services (e.g. WMS Deraa OSM).

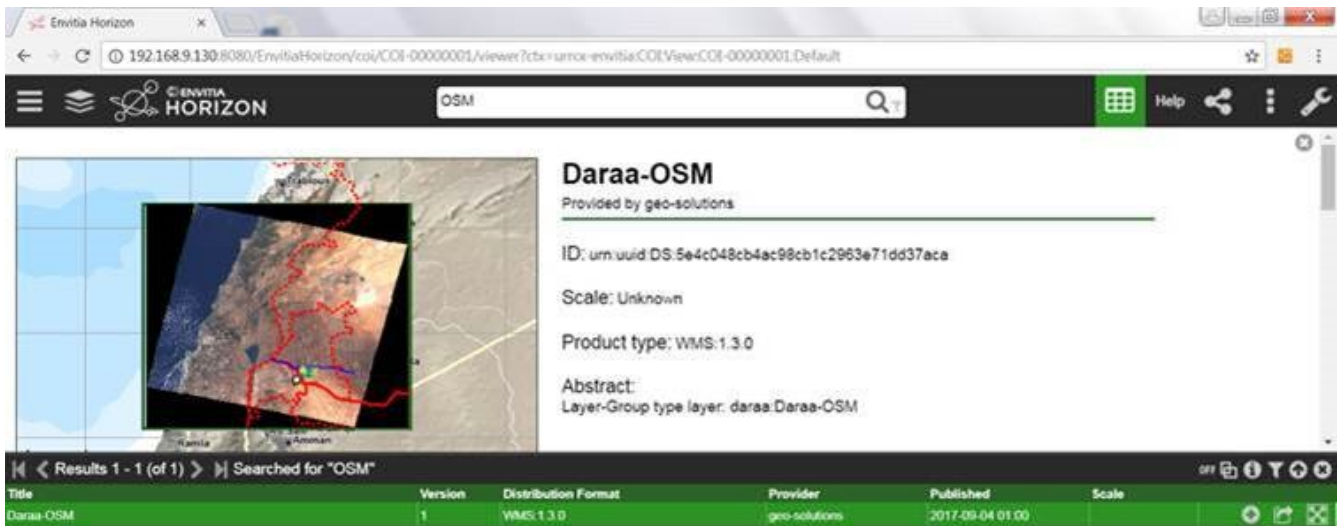


Figure 35. Searching for OpenStreetMap data.

In the following example the client is used to import a new web service and make it available to the client (e.g. WMS Daraa OSM).

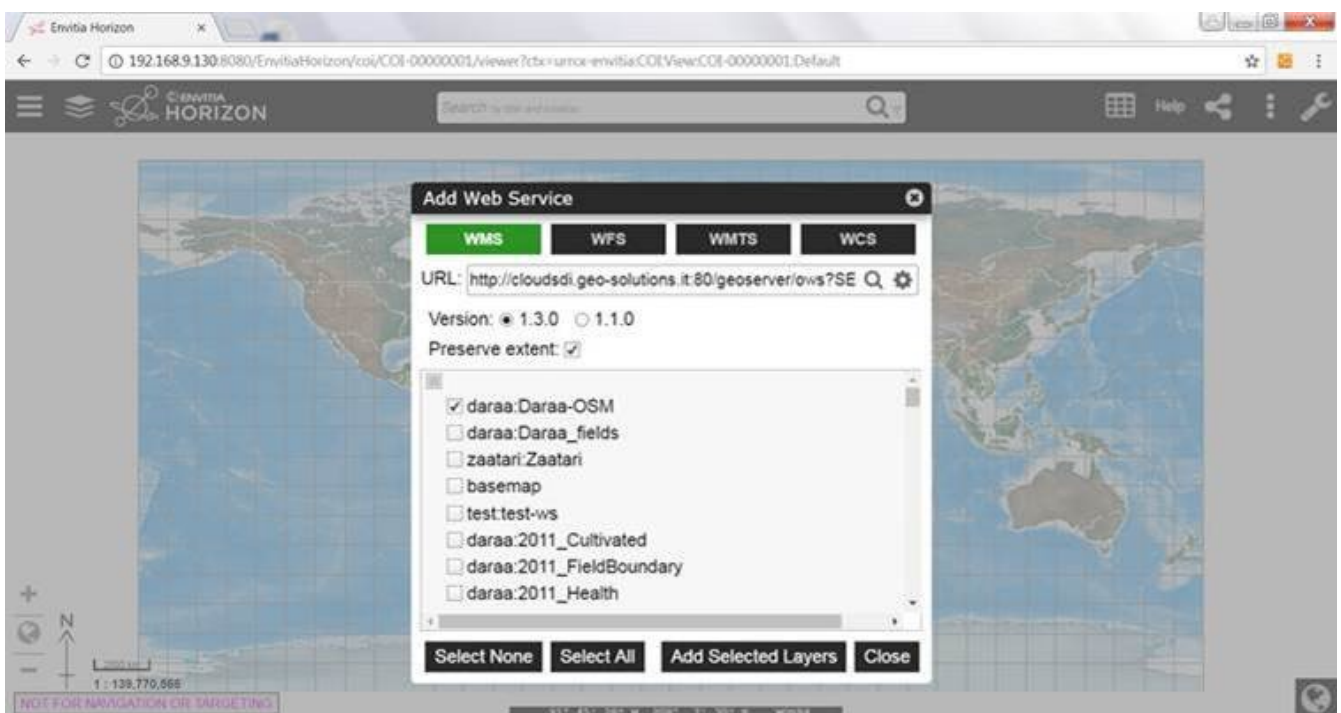


Figure 36. Importing a new web service.

In addition, the client supports creation of an OWS Context document with the discovered web services for Zaatari. Examples for WMS, WFS and WCS and then WMS, WFS, WCS and WPS are shown below.

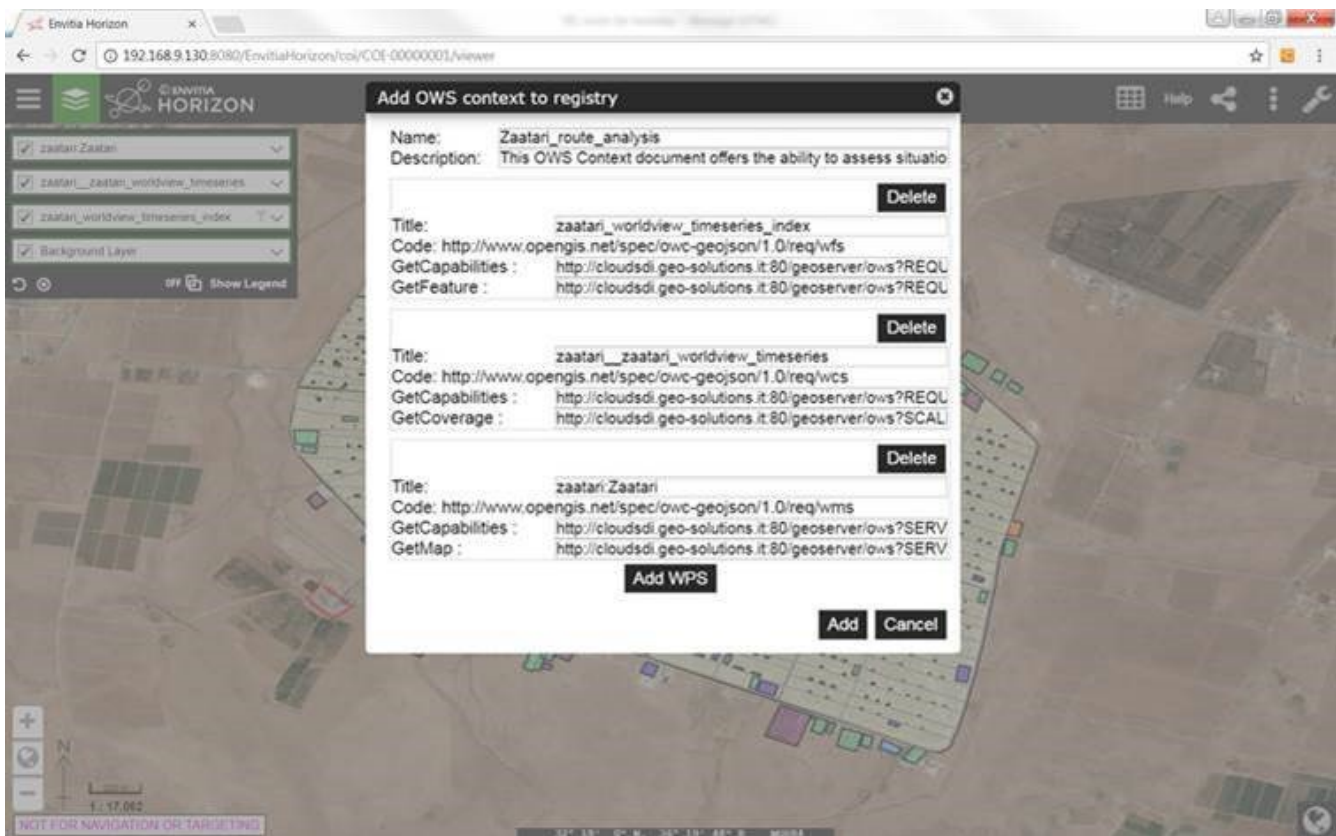


Figure 37. Creating an OWS Context document for Zaatari with WMS, WFS and WCS inputs.

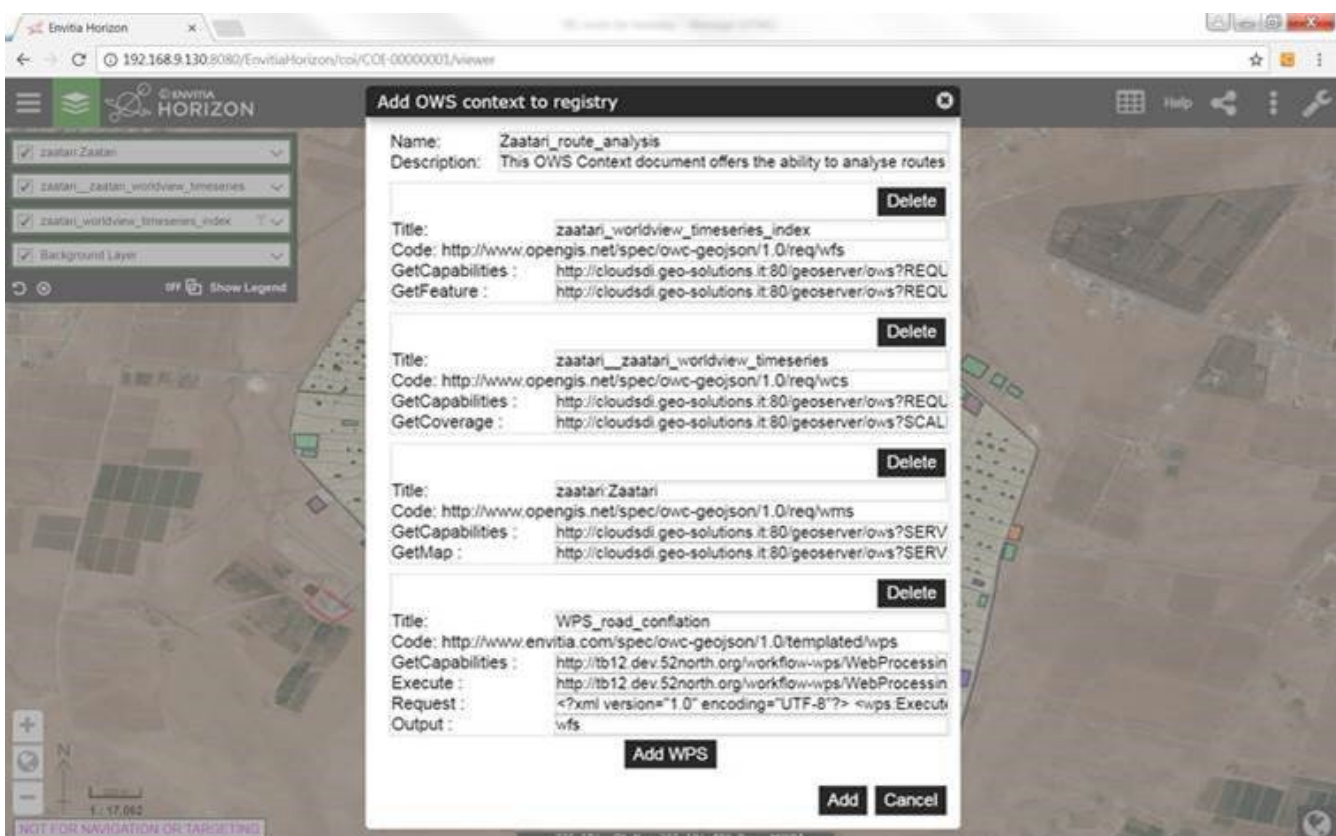


Figure 38. Creating an OWS Context document for Zaatari with WMS, WFS, WCS and WPS inputs.

Finally, WPS can be used to execute conflation. The graphic sequence below shows the output of the WPS Conflation first with OSM background, then with Landsat 8 background, then DigitalGlobe and finally the conflated data attributes:

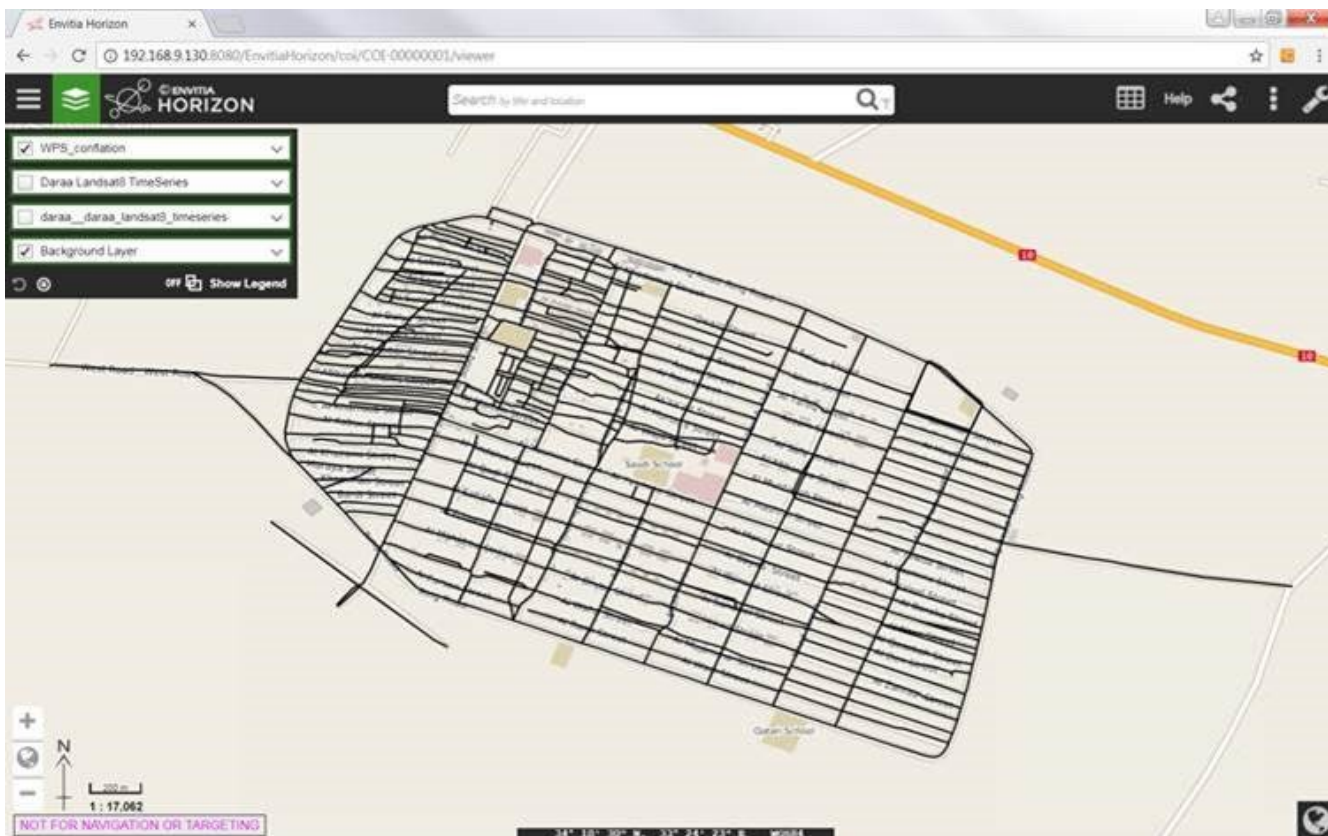


Figure 39. Output of WPS Conflation over Zaatari with OSM background.

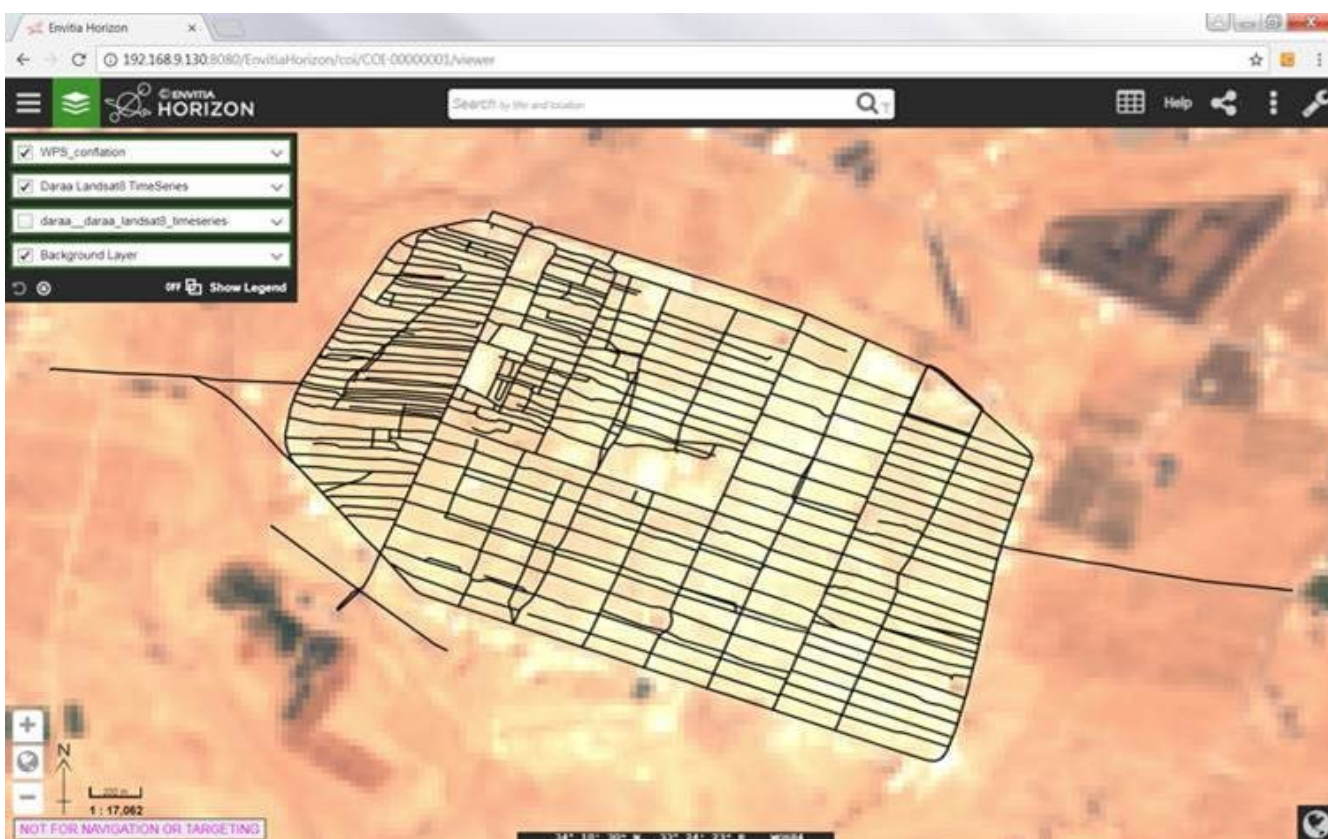


Figure 40. Output of WPS Conflation over Zaatari with Landsat 8 background.

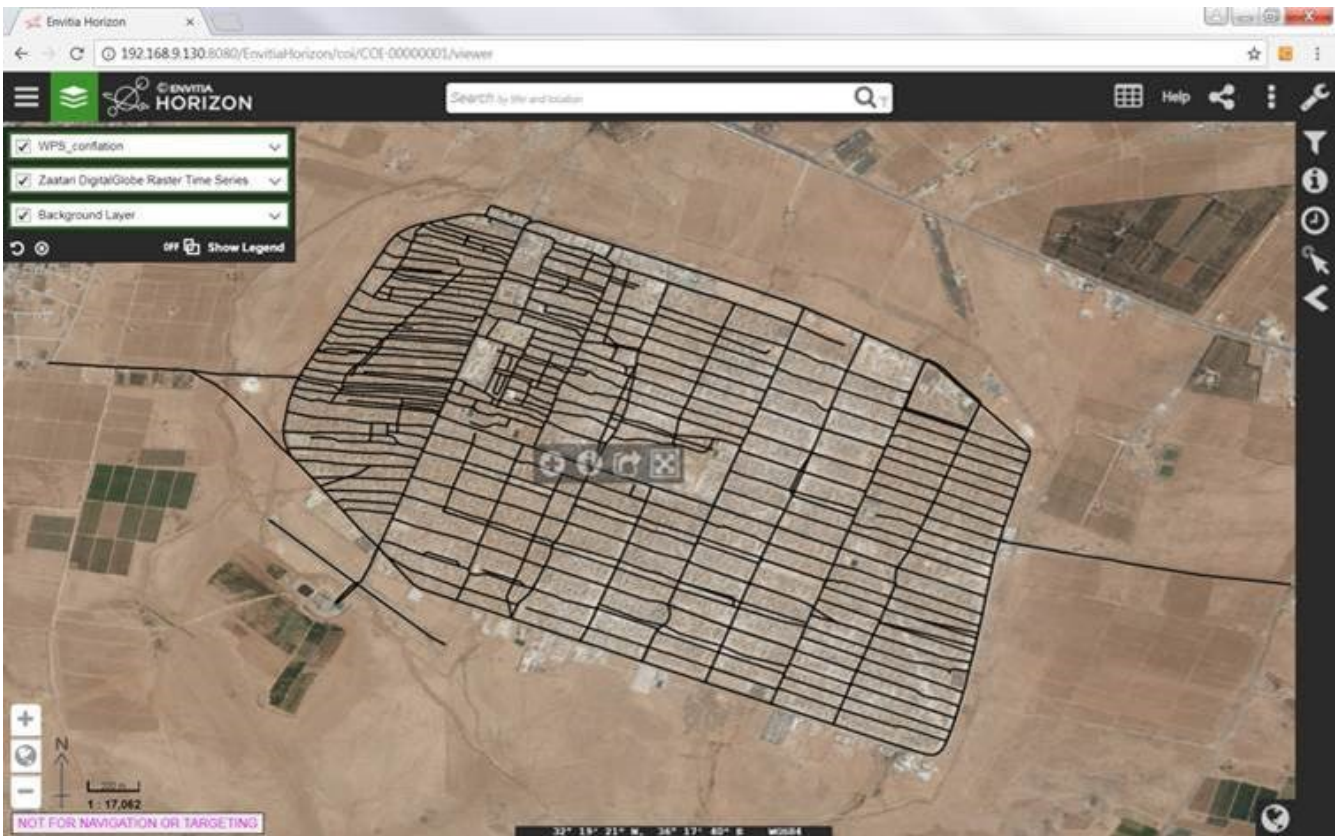


Figure 41. Output of WPS Conflation over Zaatari with DigitalGlobe background.

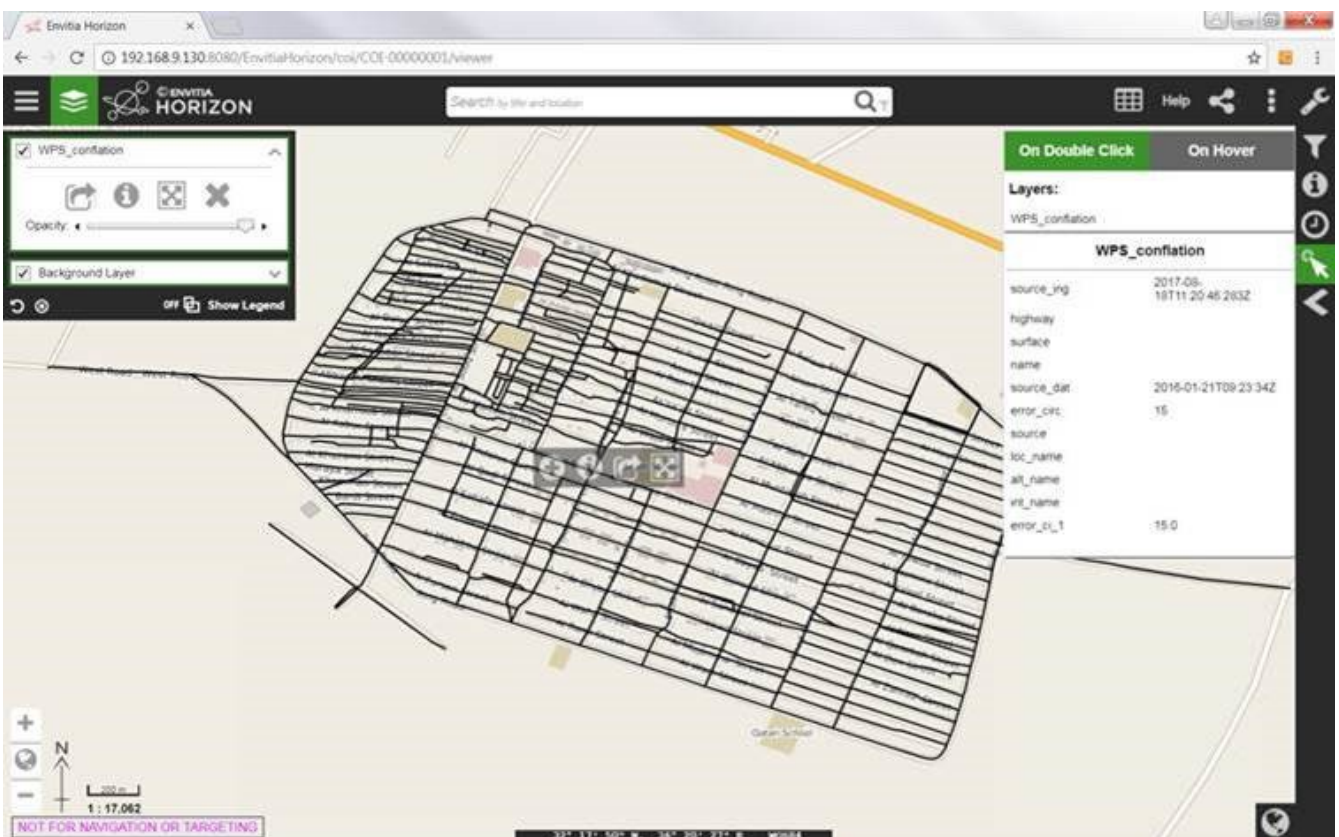


Figure 42. Output of WPS Conflation showing conflated data attributes.

Visualization Client

Visualization of data on the platform in the simulated Relief Operations Center was provided by University of Denver via integration with the World Wind environment (AB105 - Security enabled

Desktop client). This client can provide visualization capabilities for relevant geospatial web service standards including WMS, WFS, and WCS (e.g., via getting information and capabilities, rendering visualizations, providing data results). Users are provided with standard zoom, pan, adjusting, and layer management tools. This client also implemented OAuth 2.0 for secure connection to imagery and other data sources.

In the demonstration scenario one of the first actions personnel in the Relief Operations Center take is to try and understand and communicate the impact the conflict has on social conditions and infrastructure in the region. To help with this, they accessed NASA Black Marble from two different years. NASA Black Marble is a global view of the Earth's city lights acquired by Suomi National Polar Orbiting (Suomi NPP) satellite. Visualizations from 2012 and 2016 are shown below as examples.

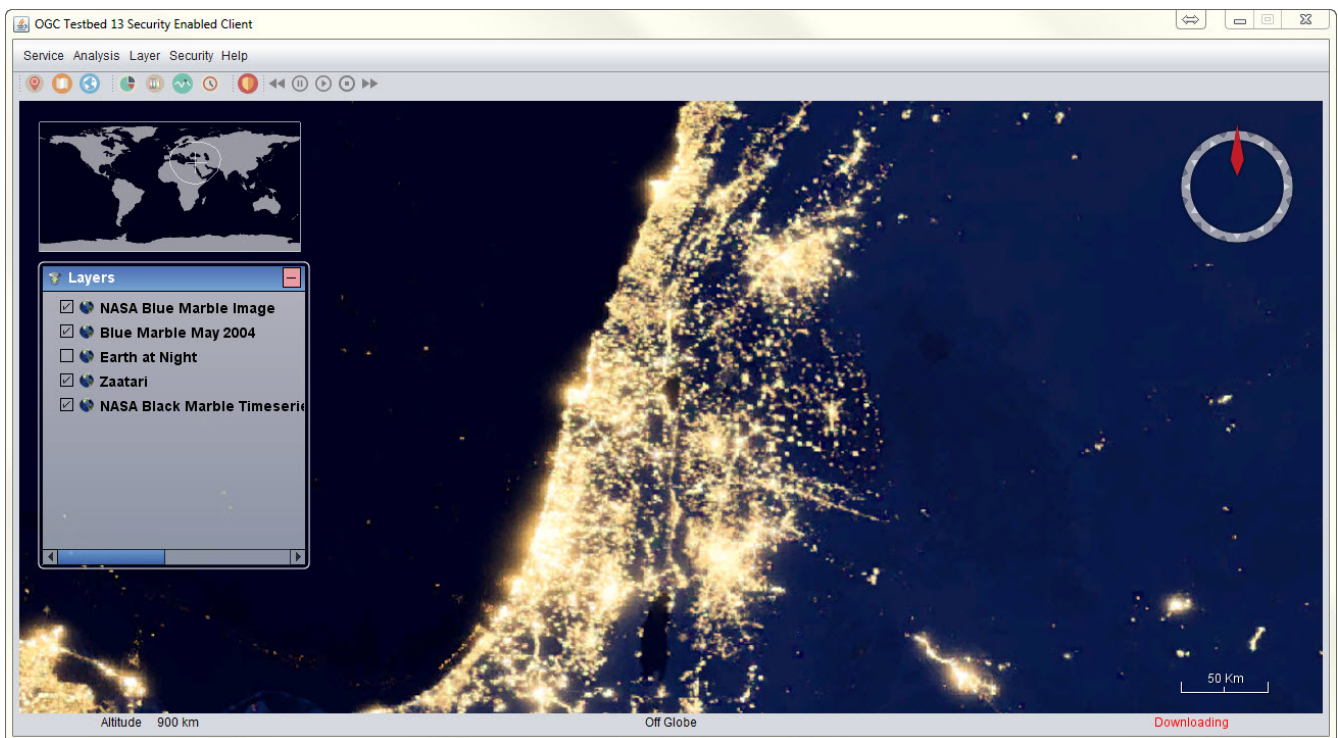


Figure 43. WMS visualization with satellite imagery from 2012, showing lights at night before the conflict.

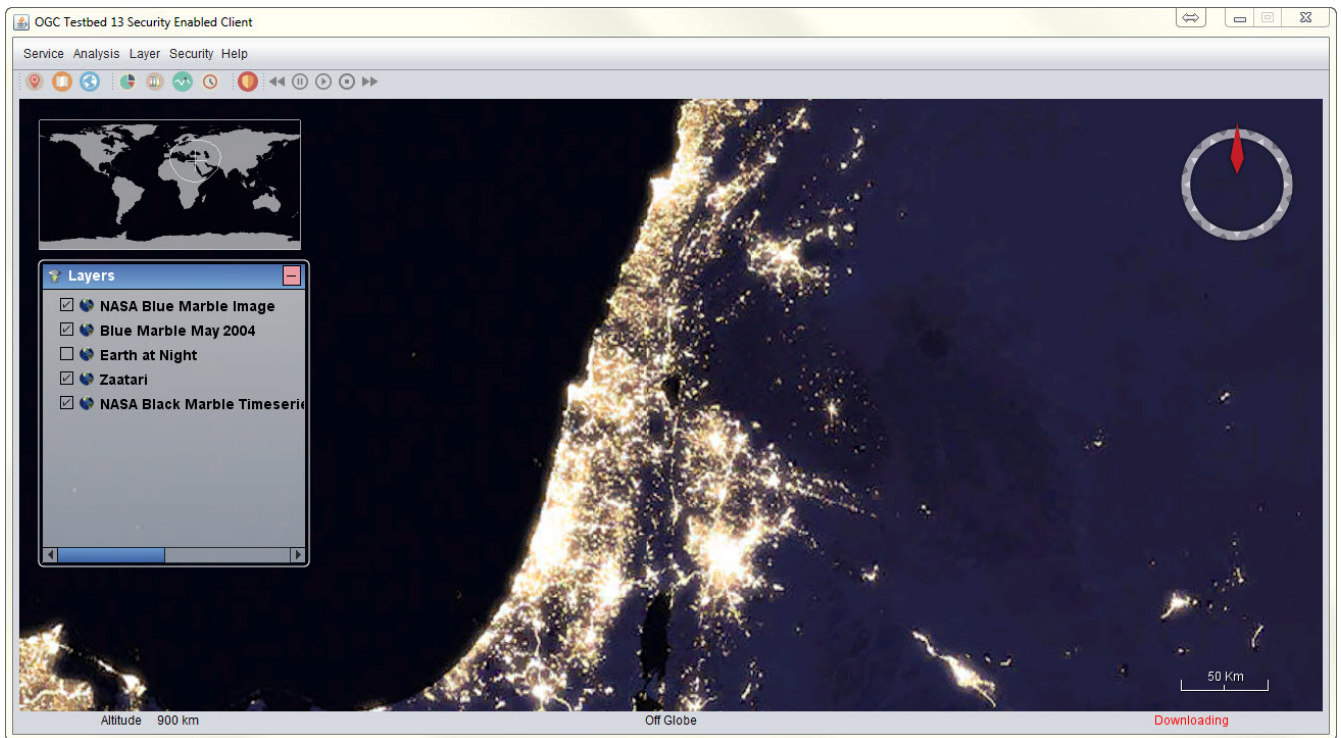


Figure 44. WMS visualization with satellite imagery from 2016, showing lights at night during the conflict.

As shown in the visualization, the change in lights at night is significant in the region. To make it even easier to distribute this visualization the Relief Operations Center also created a video of the changes and posted it on YouTube for anyone to access.



Figure 45. WMS visualization of lights at night on YouTube.

Examples for visualizing other layers from WFS and WCS in the Zaatari and Daraa region are shown in the following examples. The base imagery layer is from a secure WCS with data provided by DigitalGlobe (2017 imagery). Feature data is represented on the base imagery and includes transportation lines, structural points, settlement polygons and information points. Please note, additional information on data sources used is provided in the following section.

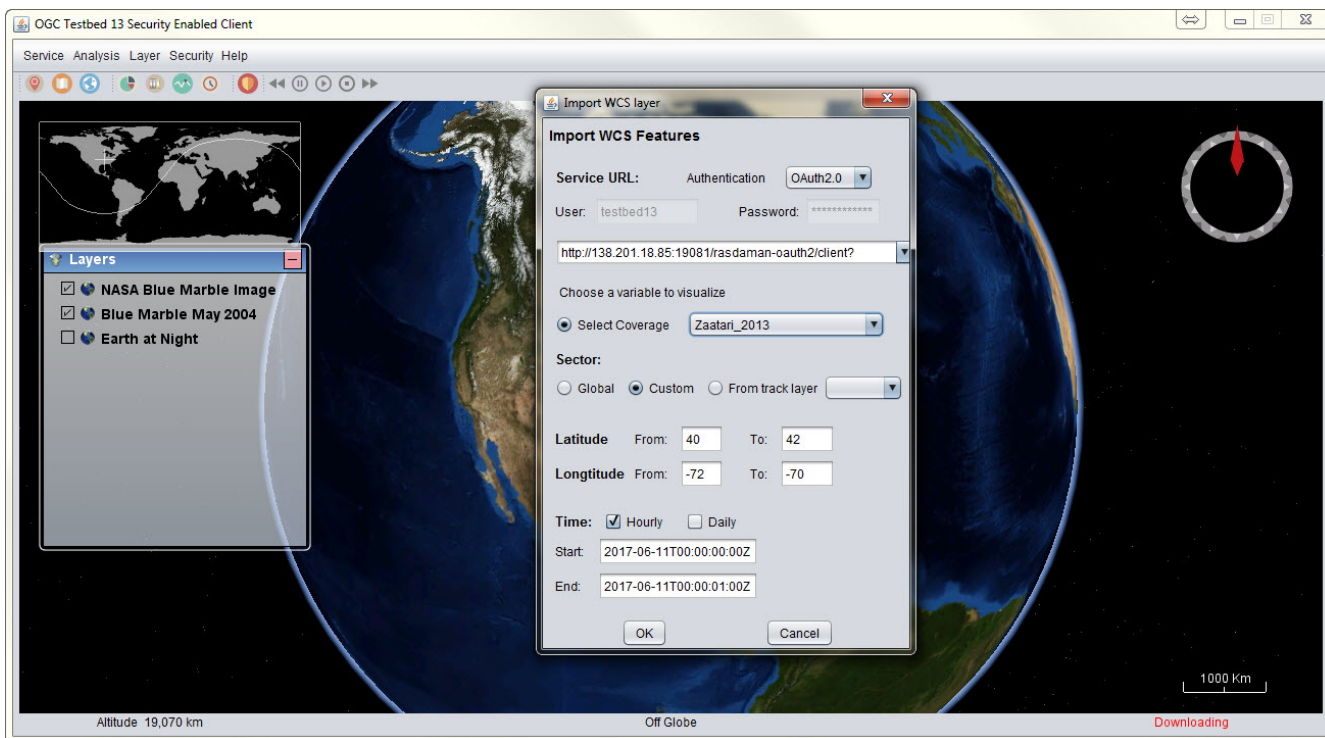


Figure 46. Visualization client accessing secure OGC services.

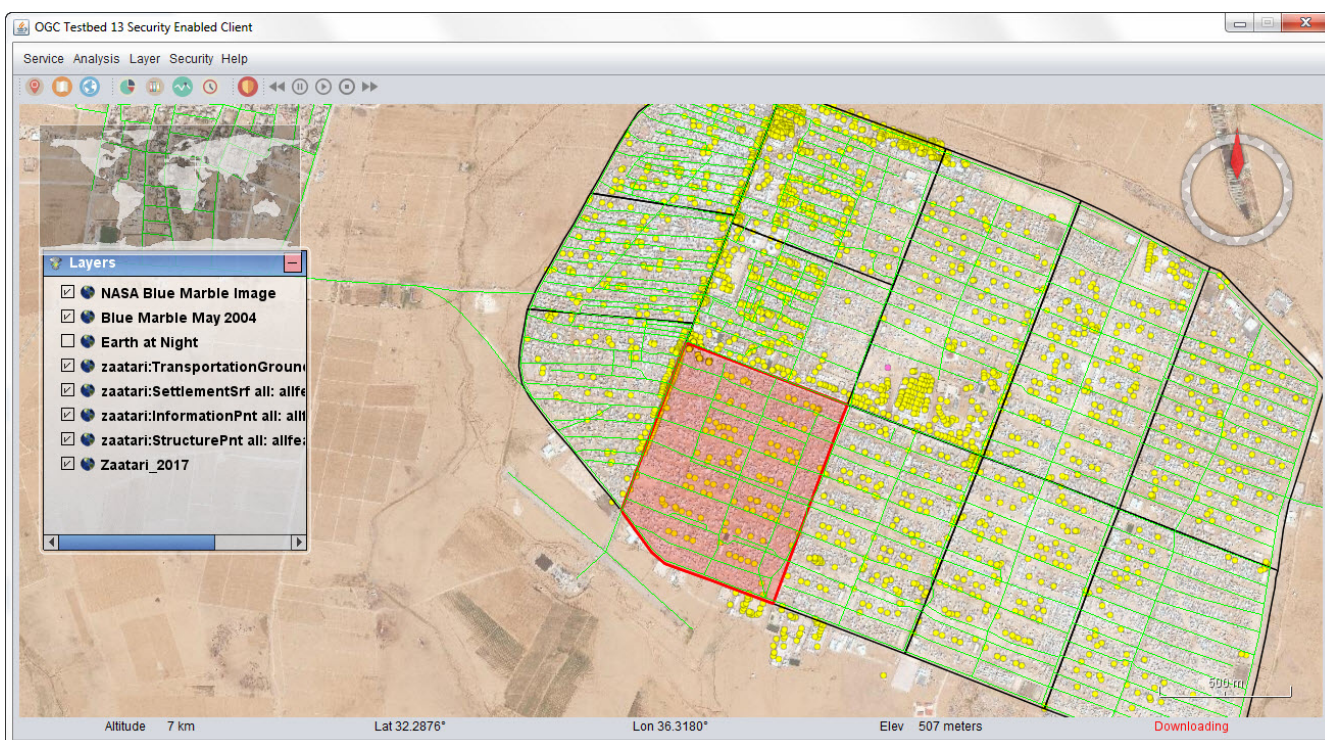


Figure 47. Security-enabled visualization client accessing data over Zaatari refugee camp.

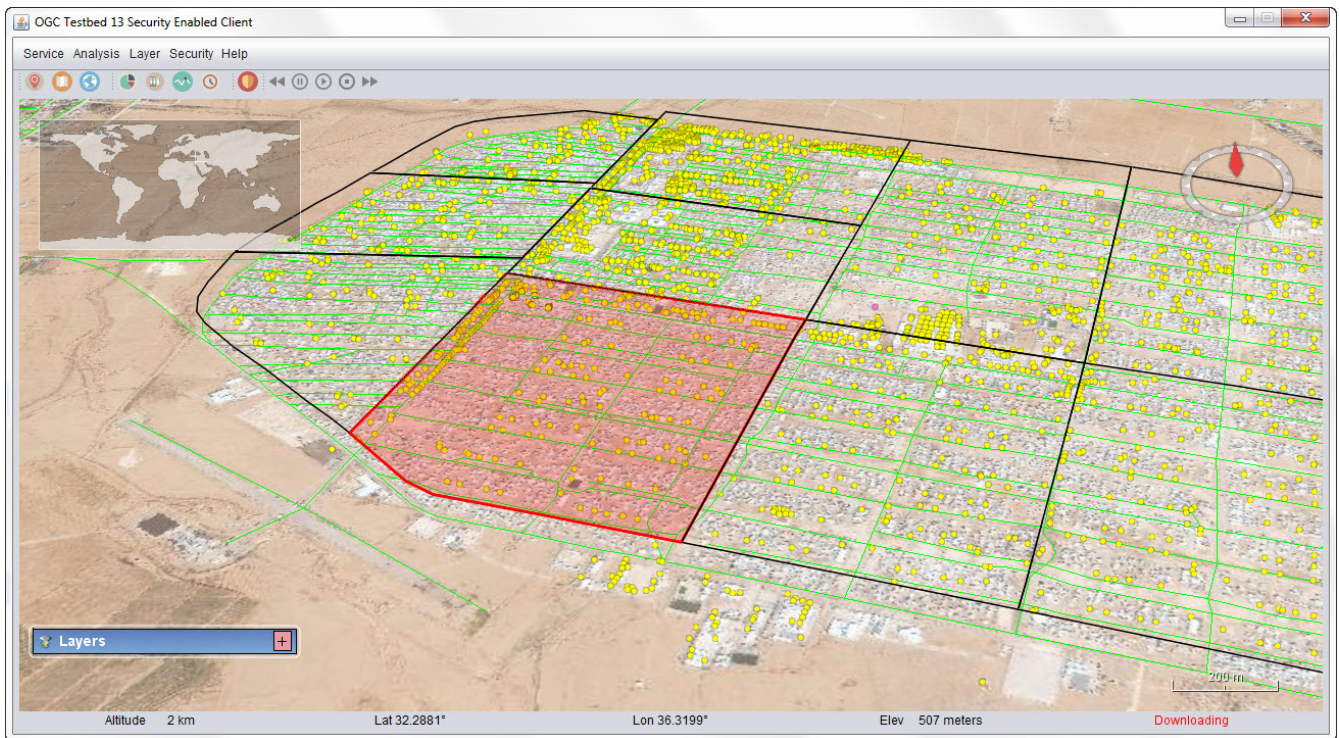


Figure 48. 3D visualization of data over Zaatari refugee camp.

1.6. Key Findings

The evidence obtained through Testbed 13 FFP Technology Integration Experiments supports the following findings:

1. It is possible to cost-effectively develop a shared information exchange platform based on open standards for situational awareness, and coordinate humanitarian relief activities among nations and organizations. In addition, such a platform can securely enable interoperable exchange of messages.
2. Users who do not know about satellite imagery or complex geospatial data sources can access the platform using client applications that allow them to simply ask questions such as, "What are the best routes from Zaatari refugee camp to Daraa?" and get answers in an easy to understand form.
3. Geospatial data experts can configure information sources to quickly respond to a variety of user questions - with encodings such as OGC Context.
4. Relief Operations Centers can access high quality visualizations of data on the platform, to easily understand the changes in the region and mass migration over time. Access to data in these visualizations can be secure, and delivered only to authorized users.
5. Applications can access satellite imagery data on the platform using imagery metadata via WFS, select a subset based on the scenes characteristics and then trigger WMS for viewing, WCS for download and also WPS for processing. This flow differs from 'normal' OGC approaches in that the WFS is used to store metadata as features. This approach is common in large scale operational deployments.
6. Mobile application clients can be used in the field to provide access to data and services of the information exchange platform. In this scenario, users access data from the platform in the field on mobile devices as OGC GeoPackages (SQLite containers for geospatial feature and imagery

data for use by mobile clients). These 'field users' often do not have a stable (or any) internet connection since network infrastructure is not reliable or available throughout much of the Daraa region.

7. The platform can support advanced analytics such as conflation of feature data sets.
8. It was noticed that in some cases it may be difficult for a generic client that starts a search on WFS to know which layer to use for WMS and WCS. Participants decided to use MetadataLinks from the GetCapabilities WFS response to link layer specific GetCapabilities of WMS and WCS. GeoServer has this capability to present a GetCapabilities call on each single layer hence it can provide this as a way to crosslink layers between different services
9. Participants enhanced GeoServer to allow the Envitia client to echo a CQL_FILTER appended to a getCapabilities call to the DCP in the response so that Desktop clients like QGIS will also be able to use basic filtering capabilities transparently.
10. For the use case above (desktop clients) participants also considered the possibility to accept a CQL_FILTER that is a URL that points to remotely stored CQL filter - so that it is easier to reuse and share them with less limitations.

1.7. What does this ER mean for the Working Group and OGC in general

This Testbed 13 effort demonstrated applications that can access satellite imagery data using imagery metadata via WFS, select a subset based on the scenes characteristics and then trigger WMS for viewing, WCS for download and also WPS for processing. This flow differs from usual OGC approaches in that the WFS is used to store metadata as features. However, this approach is common in large scale operational deployments and shows the potential for innovative new applications for WFS. Additional outcomes will be documented upon completion of further review by the Working Group.

1.8. Document contributor contact points

All questions regarding this Summary document should be directed to the editor or the contributors:

Table 1. Contacts

Name	Organization
Jeff Harrison	The Carbon Project
Mark Mattson	The Carbon Project
Simone Giannecchini	GeoSolutions
Benjamin Pross	52 North
Christoph Stasch	52 North
Stefano Cavazzi	Envitia
Jing Li	University of Denver
Gate Jantaraweragul	Image Matters

1.9. Future Work

It is expected that this document may result in changes in other documents, such as WFS, OWS Context and others based on Working Group review. Specifically, the ability for applications to access satellite imagery data on the platform using imagery metadata via WFS, select a subset based on the scenes characteristics and then trigger WMS for viewing, WCS for download and also WPS for processing seems especially useful. This flow differs from 'normal' OGC approaches in that the WFS is used to store metadata as features. This approach is common in large scale operational deployments.

Future efforts should be conducted to define and test the approach in additional scenarios, with the goal of developing an OGC Best Practices document to describe this approach as a profile of WFS.

1.10. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following normative documents are referenced in this document.

NOTE: Only normative standards are referenced here, e.g. OGC, ISO or other SDO standards. All other references are listed in the bibliography. Example:

- [OGC 06-121r9, OGC® Web Services Common Standard](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2]

NOTE

This OWS Common Standard contains a list of normative references that are also applicable to this Engineering Report.

- [OGC 07-036, OpenGIS Geography Markup Language \(GML\) Encoding Standard, August 2007](http://portal.opengeospatial.org/files/?artifact_id=20509) [http://portal.opengeospatial.org/files/?artifact_id=20509]
- [OGC 09-025r2, OGC® Web Feature Service 2.0 Interface Standard – With Corrigendum, July 2014](http://docs.opengeospatial.org/is/09-025r2/09-025r2.html) [http://docs.opengeospatial.org/is/09-025r2/09-025r2.html]
- [OGC 09-026r2, OGC Filter Encoding 2.0 Encoding Standard - With Corrigendum, August 2014](http://docs.opengeospatial.org/is/09-026r2/09-026r2.html) [http://docs.opengeospatial.org/is/09-026r2/09-026r2.html]
- [OGC 06-042, OpenGIS Web Map Service \(WMS\) Implementation Specification, March 2006](http://portal.opengeospatial.org/files/?artifact_id=14416) [http://portal.opengeospatial.org/files/?artifact_id=14416]
- [OGC 07-057r7, OpenGIS Web Map Tile Service Implementation Standard, April 2010](http://portal.opengeospatial.org/files/?artifact_id=35326) [http://portal.opengeospatial.org/files/?artifact_id=35326]
- [OGC 12-080r2, OGC OWS Context Conceptual Model, January 2014](https://portal.opengeospatial.org/files/?artifact_id=55182) [https://portal.opengeospatial.org/files/?artifact_id=55182]
- [OGC 14-065, OGC® WPS 2.0 Interface Standard, March 2015](http://docs.opengeospatial.org/is/14-065/14-065.html) [http://docs.opengeospatial.org/is/14-065/14-065.html]

Chapter 3. Terms and Definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- Client

The software component that can invoke an operation from a service.

- Context

A document that specifies a fully configured service set which can be exchanged (with a consistent interpretation) among clients supporting the standard.

- Coordinate

One of a sequence of n numbers designating the position of a point in n-dimensional space.

- Feature

An application object that represents a physical entity, e.g. a building, a river, or a person. A feature may or may not have geometric aspects.

- Filter

A filter expression predicate expression encoded using XML.

- GeoJSON

GeoJSON is an open standard format designed for representing simple geographical features, along with their non-spatial attributes, based on JavaScript Object Notation. The features include points (therefore addresses and locations), line strings (therefore streets, highways and boundaries), polygons (countries, provinces, tracts of land), and multi-part collections of these types.

- Geometry

The geometry data type is used to house information on recognized objects, like points, lines, and polygons.

- Interface

The named set of operations that characterize the behaviour of a service.

- Operation

The specification of a transformation or query that a service may be called to execute.

- Service

The distinct part of the functionality that is provided by an entity through interfaces.

3.1. Abbreviated Terms

Some of the more frequently used abbreviated terms in this document include:

- AOI Area of Interest
- API Application Program Interface
- COTS Commercial Off The Shelf
- DCE Distributed Computing Environment
- ER Engineering Report
- FFP Fit for Purpose
- GML Geography Markup Language
- HTML Hypertext Markup Language
- HTTP Hypertext Transfer Protocol
- JSON JavaScript Object Notation
- OGC Open Geospatial Consortium
- OSM OpenStreetMap
- TIE Technology Integration Experiment
- TDS Topographic Data Store
- URL Uniform Resource Locator
- W3C World Wide Web Consortium
- WWW World Wide Web
- WFS Web Feature Service
- WPS Web Processing Service

- WCS Web Coverage Service
- WMS Web Map Service
- XML Extensible Markup Language
- XSD XML Schema Definition

Chapter 4. Overview

The objective of the Fit for Purpose (FFP) activity is to ease the work of end-users, especially those who are not expert in OGC Protocols, when accessing OGC services to answer key business, social and security questions.

This document contains the following sections:

- **Preface** - This section presents information on administrative and legal aspects of this Engineering Report (ER).
- **Summary** - This section presents information on scope, what this ER means for the OGC in general and document contributor contact points.
- **References** - This section presents information on documents that are referenced in this Engineering Report.
- **Terms** - This section presents information on terms and abbreviations that are used in this Engineering Report.
- **Experiments** - This section presents information on the component implementations, architecture and the results of Technology Integration Experiments conducted.
- **Findings** - This section summarizes the findings. It also provides forward-looking recommendations.

Chapter 5. Experiments

The objective of the Fit for Purpose (FFP) effort in Testbed 13 was to develop and test filters and encodings in a platform that can ease the work of end-users, especially those who are not expert in dealing with geospatial data and satellite imagery. The platform was demonstrated in a scenario that showed how these filters can enable information exchange for humanitarian relief and analysis of mass movement of populations.

This section documents the interoperability tools and practices used by Testbed 13 participants in this platform. It includes descriptions and testing results of filters and encodings to help simplify access to satellite imagery. This technology was tested in a scenario that showed how OGC-based services, encodings, filters and applications can help coordinate humanitarian relief activities among nations and organizations.

5.1. Demonstration Scenario

Testbed 13 focused on addressing challenges related to coordinating multi-regional / multi-national operations and messaging related to the displacement and mass movement of populations in response to conflict. The current exodus of people across the Middle East to multiple nations was used as a scenario for this discussion.

In OGC Testbed 13, participants assessed the ability of open geospatial web services, encodings and filters to support simulated users in a Mass Migration Scenario over Syria and Jordan. In this scenario, large numbers of people have been displaced from the Daraa region of Syria to the Zaatari refugee camp in Jordan due to conflict. As the conflict ends 'de-escalation zones' are established and plans are made to return displaced people from refugee camps. Understanding the situation on the ground and the infrastructure, as well as transporting these people from refugee camps into a former conflict zone is a major challenge for relief agencies. To accomplish this task users must understand the environment and infrastructure in the region between Zaatari and Daraa.

The following examples provide a brief sample of the scenario involved in testing the platform and approach.

Years of war have displaced hundreds of thousands from Daraa city and region. Thousands have sought shelter at the sprawling Zaatari refugee camp...

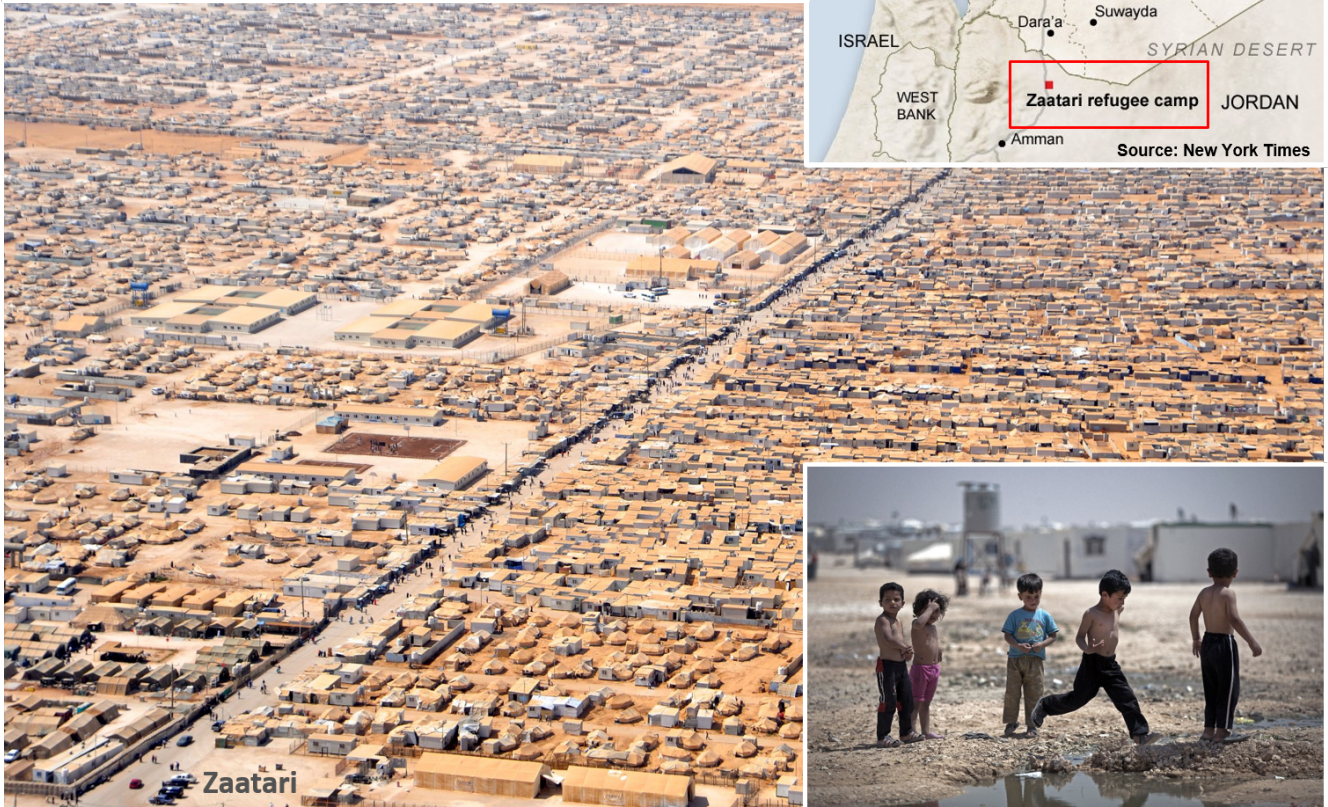


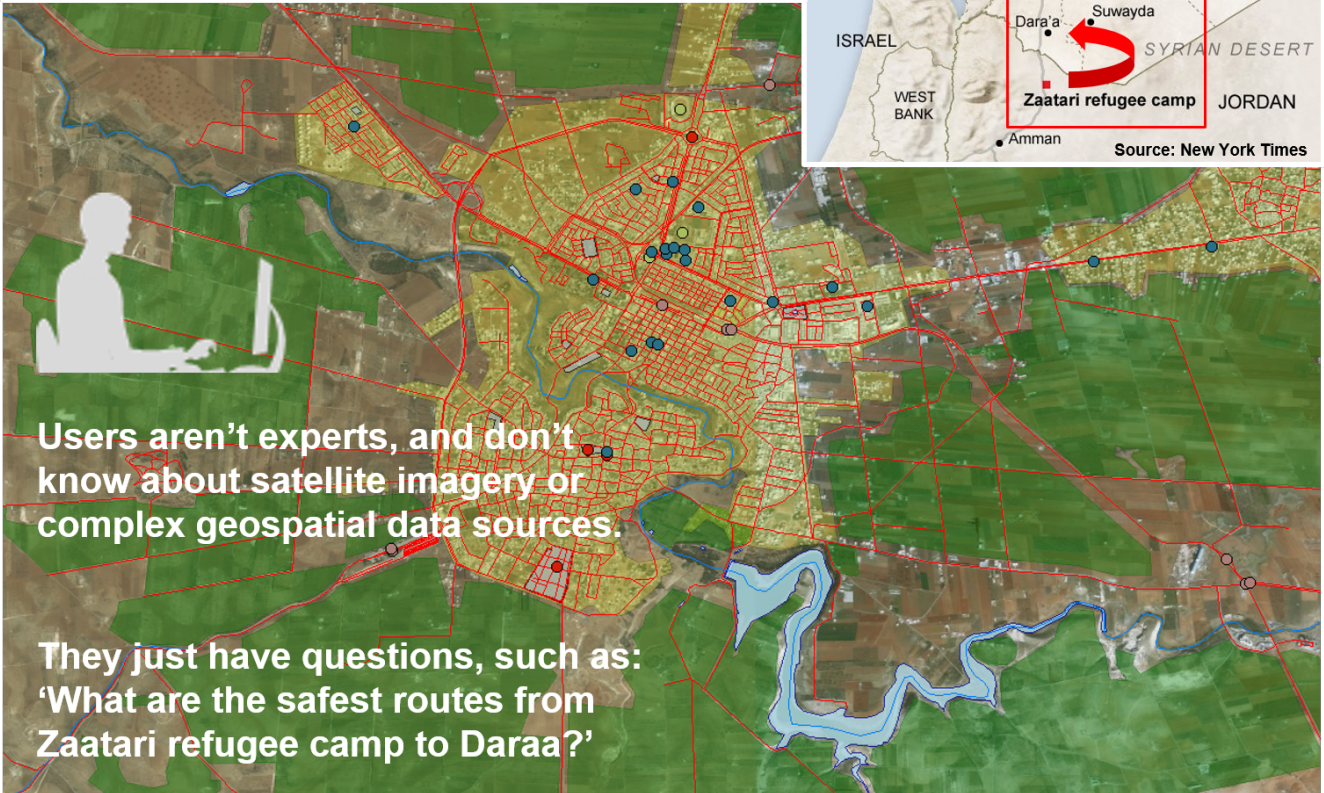
Figure 49. Testbed 13 assessed the ability of open geospatial technology to support simulated users in a Mass Migration scenario.

A ceasefire has been declared this year. Government and non-government organizations seek to assist displaced populations at Zaatari in returning to Daraa...



Figure 50. In the testing scenario, organizations seek to help displaced populations return home from refugee camps.

Need to access information for Daraa region to assist humanitarian relief operations...



Users aren't experts, and don't know about satellite imagery or complex geospatial data sources.

They just have questions, such as: 'What are the safest routes from Zaatari refugee camp to Daraa?'

Figure 51. The test scenario included simulated users who are not familiar with satellite imagery or geospatial data sources.

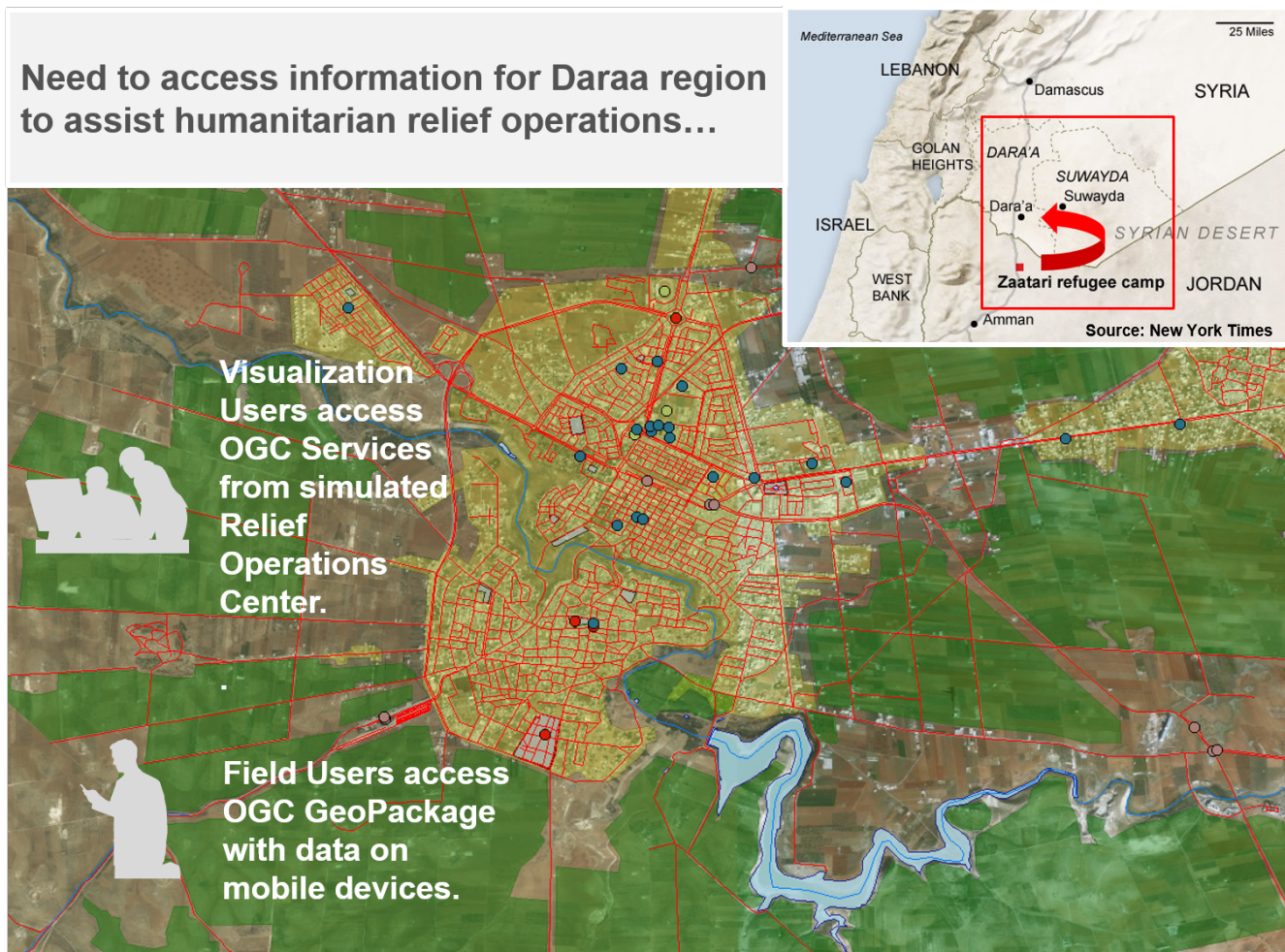


Figure 52. The test scenario also included users in a simulated operations center and users in the field.

Using this scenario as background, Testbed 13 participants demonstrated how new filters and encodings in the platform can ease the work of end-users, especially those who are not expert in dealing with geospatial data and satellite imagery.

5.2. Key Requirements

Currently, accessing geospatial data and in particular satellite imagery, requires someone with expertise to know what content to look for - and how to discover and access the data. This challenge is increased given inconsistent use of metadata standards across providers. OGC services help with discovery and access of data, but most are still limited to the assumption that the client applications must explicitly provide filter criteria such as cloud cover, age or spatial resolution to the servers in the discovery process. To open up geospatial data to broader markets we need to reduce this level of complexity.

A specific example was provided in the Testbed 13 RFQ/CFP about the ability for users to easily filter a catalog of satellite imagery exposed via WFS according to their needs, and then exploit such imagery via other OGC services (WPS, WCS and WMS). This example includes the ability to perform sophisticated processing on the data using WPS.

It is clear in the Testbed 13 RFQ/CFP that the main interest in the FFP work comes from an imagery provider perspective. In the RFQ/CFP the use of WFS is introduced as a way to allow users to query imagery metadata or library parameters (e.g. cloud cover, age, spatial resolution and so on) rather

than actual vector data.

Apparently, the requestor had in mind a somewhat higher level filter built on top of a sequence of standard OGC services which:

- Starts by using WFS to query and discover image metadata and footprints.
- Then uses WMS to visualize on a map browse or full resolution imagery.
- And then uses WCS and WPS to access and process imagery.

A graphical representation of this requirement is provided below. This representation is provided for the purposes of framing the discussion in the following section and not to provide a technical architecture. The technical architecture is provided in the following section.

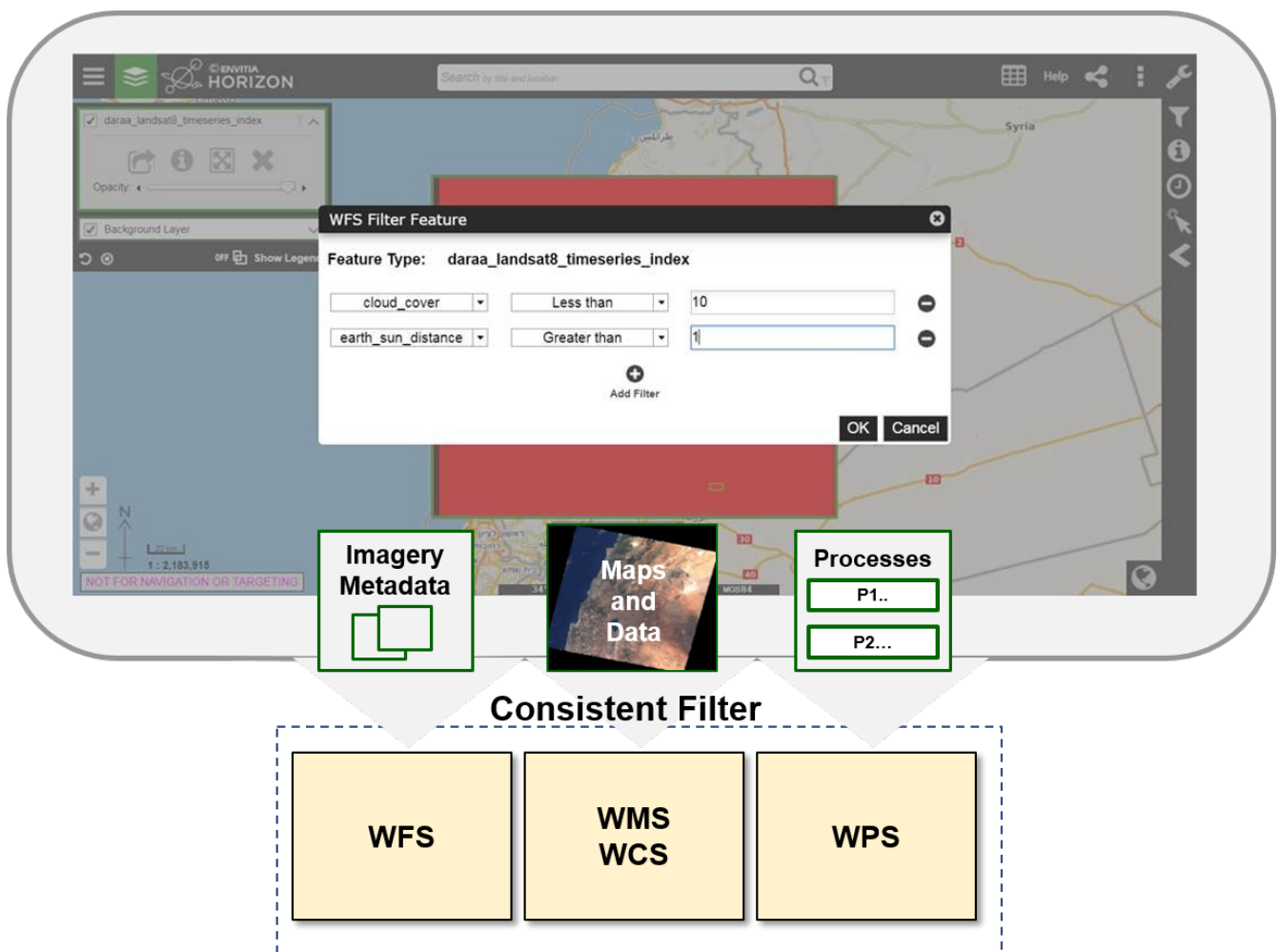


Figure 53. A key requirement was for users to easily filter a catalog of satellite imagery exposed as WFS and then exploit the imagery via other OGC services such as WMS, WCS and WPS, all using consistent filters across services.

Accordingly, the concept of 'profile' was proposed to hide the complexity of the queries needed to represent end user requirements in protocols like WFS, WMS, WCS for filtering and getting data.

The central concept in this approach is that data and/or service providers specify contexts that filter data automatically to satisfy certain needs without requiring end users to create such filters/queries themselves. Obviously, a certain level of trust between the data/service providers and the end users must be in place - as the latter will have to trust the formers as far as the content of the queries behind the profiles is concerned.

5.3. Component Implementation Design

Based on the requirements and demonstration scenario described above, an initial draft of the test platform architecture was developed during the project and is shown in the following diagram -

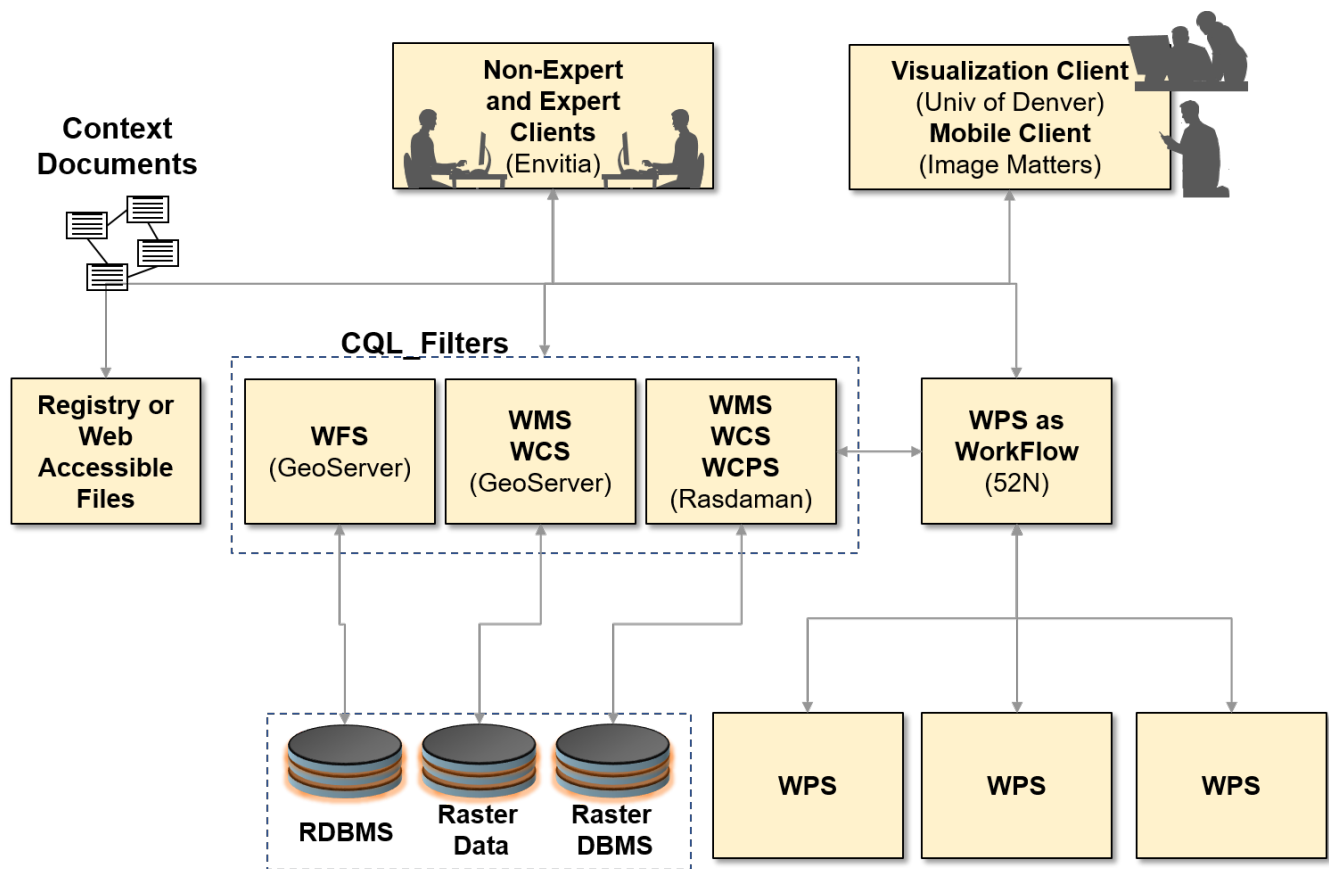


Figure 54. Testing architecture for the Fit for Purpose effort.

The platform architecture shows an initial sequence of interactions between different users, filters, encodings, services and data. The testing architecture for this part of OGC Testbed 13 was configured using a combination of the following components -

- **Filters and Encodings** - Multiple OGC encodings to support data exchange in the platform were developed and demonstrated during Testbed 13. These included filters on the WFS service that help access imagery metadata, but can also be applied to WMS and WCS for when extracting data, especially satellite imagery. JSON Context documents were used to capture collections of services for easy access to information on the platform and initiate processes on WPS such as conflation. Geography Mark Up Language (GML) was used for geospatial feature data as well GeoPackages that provided SQLite containers for geospatial feature and imagery data for use by mobile clients.
- **Client Applications** - Several types of client application components were exercised including two clients with new capability for Fit for Purpose filters and encodings to make it easy for 'non-geospatial experts to get quick answers to questions (NonExpert and Expert applications provided by Envitia). Other clients that focused on the ability to request and visualize data and from the platform were developed and demonstrated during Testbed 13. These included a visualization client for use in a relief operations center (provided by the University of Denver), mobile clients for use in the field (provided by Image Matters).
- **Data and Services**- Sources of aerial or satellite imagery, available from commercial and

government organizations, were used and represented a valuable source to support a wide variety of uses including visualization, infrastructure, population and land-use assessment. Government and private industry sources of feature data were identified and supported effective and efficient support in population centers and during mass migrations. Such geospatial data sources included transportation networks, locations and capacities of key facilities, food supplies and sources, and many others. This data was deployed on multiple OGC Web Services including WMS, WFS and WCS from GeoSolutions, WMS and WCS from Rasdaman and WPS from 52North.

The platform components were tested in a series of TIEs and demonstrated in the context of the scenario described in previous section. Interoperability experiments conducted during FFP are outlined in the following TIE table:




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 55. Technology Integration Experiments.

Three types of potential platform users were assessed, and sample client applications from Testbed 13 are shown for reference -

- **Non-Expert Users** - These users are not geospatial experts, and do not know about satellite imagery or complex geospatial data sources. They just have questions to enter into the platform, such as: ‘What are the safest routes from Zaatari refugee camp to Daraa.

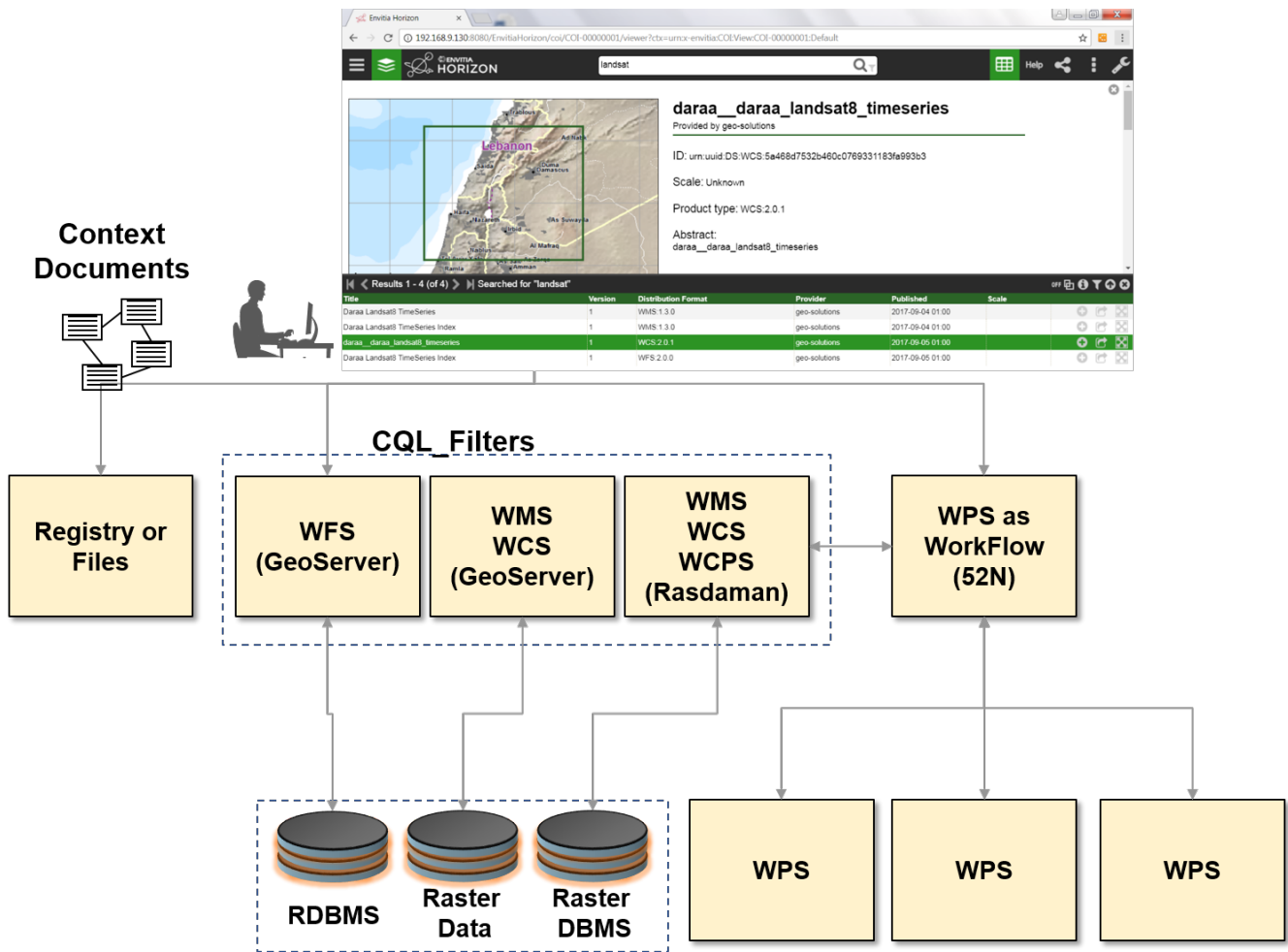


Figure 56. Client for non-expert users.

- **Expert Users** - These users are geospatial experts and understand the OGC services, satellite imagery and complex data sources in the platform. They set up the profiles and OGC Context Documents that provide help for Non-Expert Users.

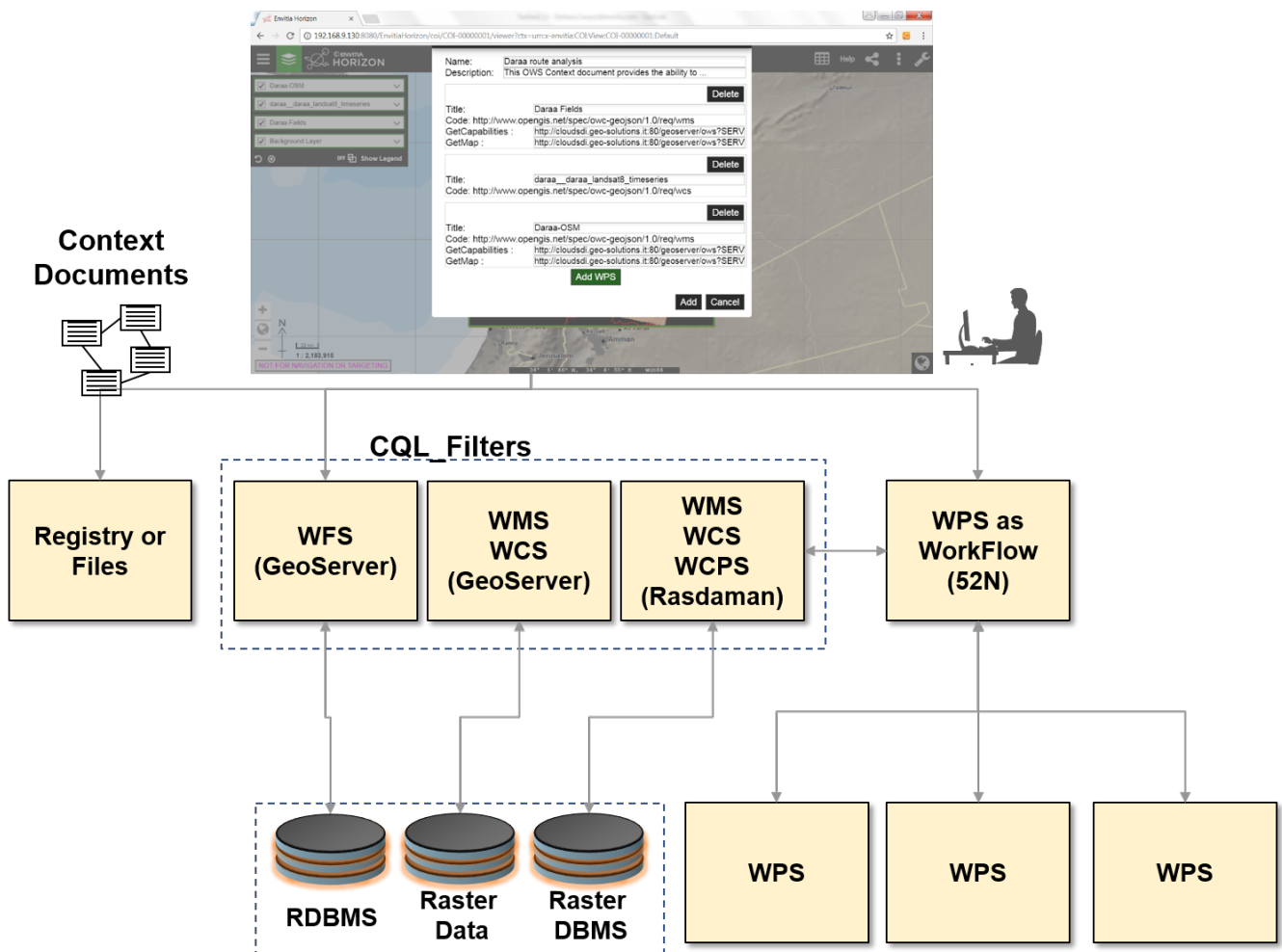


Figure 57. Client for expert users.

- **Visualization Users** - These users access the platform from simulated Relief Operations Centers, and have at least a laptop, client software and reliable internet connections.

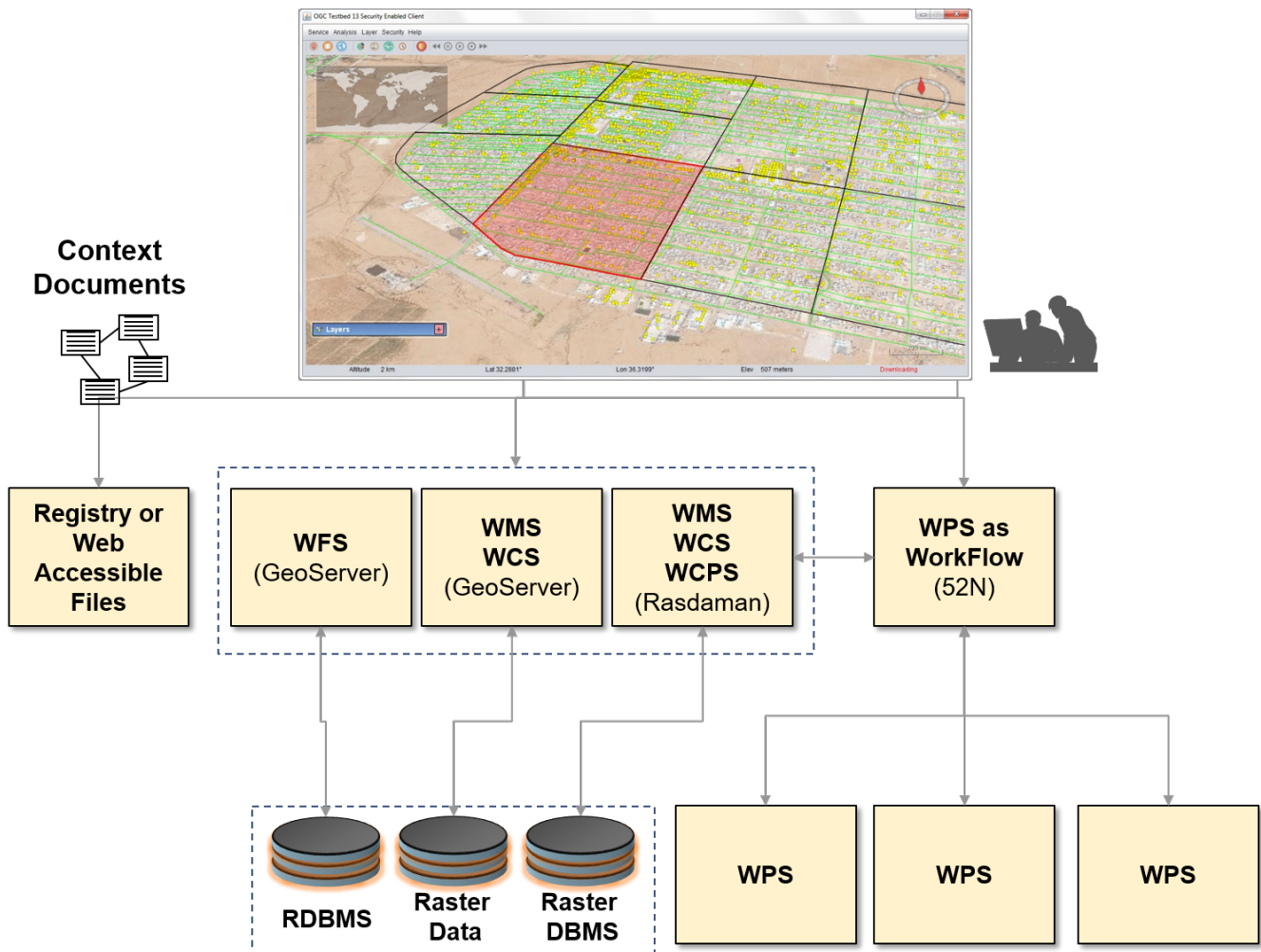


Figure 58. Visualization client.

The following sections describe the OGC standards that were assessed during platform filter and encoding design.

5.4. Prior-After Comparison

The following sections describe the potential use of existing OGC standards to meet FFP requirements. In particular, the need to capture the concept of a profile in a way that allowed it to be applied between multiple OGC Services in the test platform including WMS, WCS, WCPS and WPS was a key design consideration.

5.4.1. WFS and Stored Query Support Assessment

As explained above and in the Testbed 13 RFP, OGC WFS plays a central role as it can be used to:

- Serve relevant vector data, i.e. OpenStreetMap and allow sophisticated querying on that data.
- Serve metainformation about existing satellite imagery, which can be used to discover data for access or processing.

In Testbed 13 it was suggested that much of the concept of profile, as a quick and easy way to filter data according to predefined queries, is already provided by OGC WFS in the form of prior work on Stored Query.

The reader is referred to the WFS standard for a complete description of Stored Query functionality, but it is valuable to explain some of the functionalities and operations to understand why it was not selected for filtering data and applying filters between multiple OGC Services including WMS, WCS, WCPS and WPS.

In brief, a Stored Query is a predefined WFS query provided by the server itself that can be applied through a GetFeature operation using its unique identifier to filter data - without having to specify a query or know any details about parameters. For example, the request below makes use of the 'GetFeatureOverItaly' Stored Query:

<http://www.opengis.net/def/query/OGC-WFS/0/GetFeatureOverItaly>

The Stored Query extracts:

http://www.someserver.example.com/wfs.cgi?SERVICE=WFS&VERSION=2.0.0&REQUEST=GetFeature&STOREDQUERY_ID=http://www.opengis.net/def/query/OGC-WFS/0/GetFeatureOverItaly

Such queries can also be parametric, which means users can supply parameters together with the Stored Query ID to customize the query according to needs. For example, the query below has a parameter that must be provided by the user (feature ID) for the feature to extract:

http://www.someserver.example.com/wfs.cgi?SERVICE=WFS&VERSION=2.0.02.0.2&REQUEST=GetFeature&STOREDQUERY_ID=http://www.opengis.net/def/query/OGC-WFS/0/GetFeatureById&ID=INWATERA_1M.1013

Another example of a stored query is provided below using a bbox parameter:

http://geoserv.weichand.de:8080/geoserver/wfs?service=WFS&request=GetFeature&version=2.0.0&StoredQuery_ID=bboxQuery&x1=4466856&y1=5346319&x2=4466980&y2=5346396

Finally, WFS 2.0.2 provides operations to manage Stored Queries (i.e. create, delete) including CreateStoredQuery, DropStoredQuery, ListStoredQueries and DescribeStoredQueries.

The final consensus at the start of Testbed 13 was that WFS Stored Queries may meet a portion of the requirements, but that a method was needed to capture the concept of a profile in a way that allowed it to be applied between multiple OGC Services including WMS, WCS, WCPS and WPS.

5.4.2. OWS Context Documents Assessment

While WFS Stored Query was assessed to help capture the concept of profile for WFS services, it does not help with covering the concept of profile for other services like WMS, WCS, WCPS and WPS. It also does not provide a portable container for exchanging profiles.

Accordingly, during the Testbed 13 Kickoff it was proposed to capture the concept of profile in a way that allows it to be applied between multiple OGC services - by using prior work on OWS Context.

As stated on the OWS Context page, a 'context document' specifies a fully configured service set which can be exchanged among clients supporting the standard. The OWS Context standard was created to allow a set of configured information resources (service set) to be passed between

applications primarily as a collection of services. OWS Context is developed to support in-line content as well. The goal is to support use cases such as the distribution of search results, the exchange of a set of resources such as WFS, WMS, WCS, Web Map Tile Service (WMTS) and others. Additionally OWS Context can deliver a set of configured WPS parameters to allow the processing to be reproduced on different nodes.

An OWS Context Document is therefore an obvious candidate to hold information on how to access/filter, extract and process data from different data providers using different OGC protocols. A profile can be translated into an OWS Context Document that lists multiple data providers' services along with the stored queries they offer for the purpose, but it can also contain WCS and WCPS services for performing extractions as well as various WPS service endpoints listing available processes that are suitable for the 'purpose' itself.

An OWS Context Document can be represented in XML and JSON and can contain both service endpoints, data (i.e. FeatureCollection instances) and more. A relatively simple example is provided below.

```
**<?xml version="1.0">
  <link rel="profile"
        href="http://www.opengis.net/spec/owc-atom/1.0/req/core"
        title="This file is compliant with version 1.0 of OGC Context"/>
  Default
  <title>Default</title>
  <updated>2013-01-14T18:08:44Z</updated>
  <author>
    <name>Envitia</name>
    <email>support@envitia.com</email>
    <uri>www.envitia.com</uri>
  </author>
  <georss:where>
    <gml:Polygon>
      <gml:exterior>
        <gml:LinearRing>
          <gml:PosList srsDimension="2" srsName="EPSG:4326">29.323818 -85.906275
45.676182 -85.906275 45.676182 -52.407170 29.323818 -52.407170</gml:PosList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </georss:where>
  <entry>
    Web Map Service (Port Information)
    <title>Web Map Service (Port Information)</title>
    <updated>2013-01-14T18:08:44Z</updated>
    <content type="html">Web Map Service (Port Information)</content>
    <category scheme="http://www.opengis.net/owc/active" term="true"/>
    <owc:offering code="http://www.opengis.net/spec/owc/1.0/req/atom/wms">
      <owc:operation code="GetCapabilities"
href="http://dmargs.envitia.com/envitia.bgmaps/wms.aspx?REQUEST=GetCapabilities&SERVIC
E=WMS" method="GET"/>
      <owc:operation code="GetMap"
```

```

href="http://dmargs.envitia.com/envitia.bgmaps/wms.aspx?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image/png&BGCOLOR=0xFFFFFFFF&TRANSPARENT=TRUE&CRS=EPSG:4326&LAYERS=PIG
&
  BBOX=-85.906275,29.323818,-52.407170,45.676182&WIDTH=1280&HEIGHT=748"
method="GET"/>
  </owc:offering>
</entry>
<entry>
  Annotation Layer
  <title>Annotation Layer</title>
  <updated>2013-01-14T18:08:44Z</updated>
  <content type="html">Annotation Layer</content>
  <category scheme="http://www.opengis.net/owc/active" term="true"/>
  <owc:offering code="http://www.opengis.net/spec/owc/1.0/req/atom/gml">
    <owc:content type="application/gml+xml">
      <gml:FeatureCollection gml:id="1234567890"
xmlns:clk="http://www.envitia.com/clk" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.envitia.com/clk clk.xsd">
      <gml:boundedBy>
        <gml:Envelope srsName="urn:opengis:def:crs:EPSG::28992">
          <gml:lowerCorner>34.560849 -73.688164</gml:lowerCorner>
          <gml:upperCorner>40.278833 -63.215498</gml:upperCorner>
        </gml:Envelope>
      </gml:boundedBy>
      <gml:featureMembers>
        <clk:Annotation gml:id="1">
          <clk:geometry>
            <gml:Polygon srsName="urn:opengis:def:crs:EPSG::4326">
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList srsDimension="2">35.763229 -71.607057 34.560849
-68.216867 36.057144 -63.215498 40.278833 -70.163709 37.606878
-73.688164</gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </clk:geometry>
        </clk:Annotation>
      </gml:featureMembers>
    </gml:FeatureCollection>
  </owc:content>
</owc:offering>
</entry>
</feed>

```

It is worth highlighting that in the scenario described above OWS Context Documents could be offered as files or could also be registered into a registry service to allow users to discover them via advanced searches rather simple file listings.

5.4.3. CQL_Filter on WFS, WMS, WCS and WPS Assessment

An open point for discussion covered during the kick-off was the ability to carry along the filter and/or stored query which the user might have performed on the WFS service when extracting data via WCS and WMS, especially when dealing with satellite imagery.

A potential approach that was proposed and tested is the one implemented by the GeoServer Open Source product. Specifically, GeoServer has implemented a number of extensions to OGC standards in order to overcome existing limitations. In particular, support for filtering using the Common Query Language (CQL) borrowed by the OGC CSW specification has been implemented for both WFS and WMS to provide a simple way to filter data from web-based clients.

CQL (and its extended version called ECQL) enables sophisticated filters combining spatial and alphanumeric predicates which are also easy for users to read and reuse. Examples of CQL Filters are shown below:

- `DATE_COL AFTER 2016-01-01T00:00:00Z AND DATE_COL BEFORE 2016-12-31T23:59:59Z` to filter data by the year 2016
- `STATE_NAME IN ('New York', 'California', 'Montana', 'Texas')`
- `PERSONS BETWEEN 1000000 AND 3000000`
- `INTERSECTS(the_geom,%20POINT%20(-74.817265%2040.5296504))`

In GeoServer the same CQL filter can be used across WFS, WMS and WCS for raster time series layers in order to filter data extraction in WCS as well as data visualizations in WMS in a similar manner. This approach could also be applied to WCS when serving a catalogue of satellite imagery in order to filter the index of the imagery via WFS and the extract using the same exact filter via WCS.

This approach could be somehow alternative to the Stored Query approach as long as the concept of profile is captured by an OWS Context Document; the latter would hide the complexity of the filtering and could still allow a client to access and exploit multiple data provider services.

Alternatively, the concept of Stored Query would need to be implemented also for WCS and (optionally) WMS in order to have a querying mechanism that is portable between the various services, although this portability is not necessary as within an OWS Context Document we might end up coding the same query that implement a specific profile using different mechanism for different OGC services.

However, the issue participants faced is also that there is no straightforward way to filter a dataset in the same way using WFS and WMS (think about an underlying vector dataset) or WFS and WMS/WCS in case we are exposing a satellite imagery catalogue; this is one of the reasons why CQL filtering was implemented in GeoServer for WFS and WMS.

5.4.4. Role of a Client as a Mediator for Filters and Encodings

A fundamental role was played by the Testbed 13 client applications which were responsible for simplifying the sequence of actions and hiding complexity from the end user while presenting an interface that responds to user questions.

The client should be able to discover and exploit OWS Context Documents as well as mediate the interaction between the user and the various services involved in the FFP work.

5.4.5. Role of the Workflow and WPS

WPS will play a role in the scenario to support the decision process by implementing processing to conflate or fuse the discovered raster and vector data to produce higher-level information to drive the user's decision process. The specific product is described above in the context of the Mass Migration scenario.

The WPS interface might act as broker towards sophisticated workflows which, once triggered by the client, will access the data providers' service (i.e. WFS, WCS) to extract data to be processed.

5.4.6. OGC Filtering in GeoServer

GeoServer supports OGC filters in WFS, both in KVP and XML POST modes, as the WFS standard suggests. Usage of OGC filter is not a standard in WMS, however GeoServer still allows usage of it in WMS KVP requests, and again in WMS, it can be also used as part of dynamic SLD styles (however, this implies the client needs to send the entire style, not just the style).

At the start of Testbed-13, GeoServer did not support any sort of OGC filtering in WCS, although it was simple enough to align it with CQL support and add it to KVP requests as a vendor parameter.

The main drawback of using OGC filter in KVP requests is URL length, as XML is verbose and many of its characters need escaping. For example, a simple CQL like "LAND_KM between 100000 and 150000" (33 chars) gets encoded in OGC filter as (formatted here for legibility):

```
<Filter xmlns="http://www.opengis.net/ogc">
  <PropertyIsBetween>
    <PropertyName>topp:LAND_KM</PropertyName>
    <LowerBoundary>
      <Literal>100000</Literal>
    </LowerBoundary>
    <UpperBoundary>
      <Literal>150000</Literal>
    </UpperBoundary>
  </PropertyIsBetween>
</Filter>
```

Which in turn needs to be url-encoded and becomes 306 characters long, that is, 10 times longer than CQL (even keeping it as a non-formatted one liner):

```
%3CFilter%20xmlns=%22http://www.opengis.net/ogc%22%3E%3CPropertyIsBetween%3E%3CProperty
yName%3Etopp:LAND_KM%3C/PropertyName%3E%3CLowerBoundary%3E%3CLiteral%3E100000%3C/Liter
al%3E%3C/LowerBoundary%3E%3CUpperBoundary%3E%3CLiteral%3E150000%3C/Literal%3E%3C/Upper
Boundary%3E%3C/PropertyIsBetween%3E%3C/Filter%3E
```

5.4.7. Assessment on Best Method to Meet Filter Requirement

A key design consideration for FFP components in the OGC Testbed test suite is the ability to 'exchange' the filter and/or stored query which a client application may have used on the WFS when accessing data via WCS or visualizing data with WMS. This is especially important when dealing with satellite imagery.

Given this challenge, it is important to understand support for carrying filters and the parameters embedded in them across OGC standards. Accordingly, the following table summarizes the current level of support for the various filtering and sorting options across protocols and request bindings.

	KVP-GET	KVP-POST	XML-POST
CQL	GeoServer extension. Works on WFS, WMS, WCS.	GeoServer extension. Works on WFS, WMS, WCS.	Not supported anyway. If implemented would be a GeoServer extension.
OGC Filter	Works on WFS according to standard. GeoServer extension for WMS. GeoServer WCS currently does not support OGC filter for mosaics.	Works on WFS according to standard. GeoServer extension for WMS. GeoServer WCS currently does not support OGC filter for mosaics.	Works on WFS according to standard. GeoServer extension for WMS ¹⁰ , see this link . Requires passing the whole SLD. GeoServer WCS currently does not support OGC filter for mosaics
Sorting	Works on WFS, needs implementing in WMS/WCS	Works on WFS, needs implementing in WMS/WCS	Works on WFS, no clear place to add it in WMS, in WCS 2.0 it could be a profile extension.

Figure 59. Comparison of filters.

Given the assessments described above, the Testbed 13 Component Implementation Design for FFP proposed to have an OWS Context document providing links with CQL expressions for filtering, along with ordering similarly using the KVP syntax, be it as native (WFS) or vendor extension (WMS/WCS).

5.5. Technology Integration Experiments

This section describes the key implementations related to CQL_Filters and Context Documents as a mechanism for a 'profile' that can be applied between multiple OGC Services in the test platform

including WMS, WCS, WCPS and WPS.

The testing architecture for FFP in OGC Testbed 13 exercised the encodings, filters, services, data and client applications presented below and in the following manner:

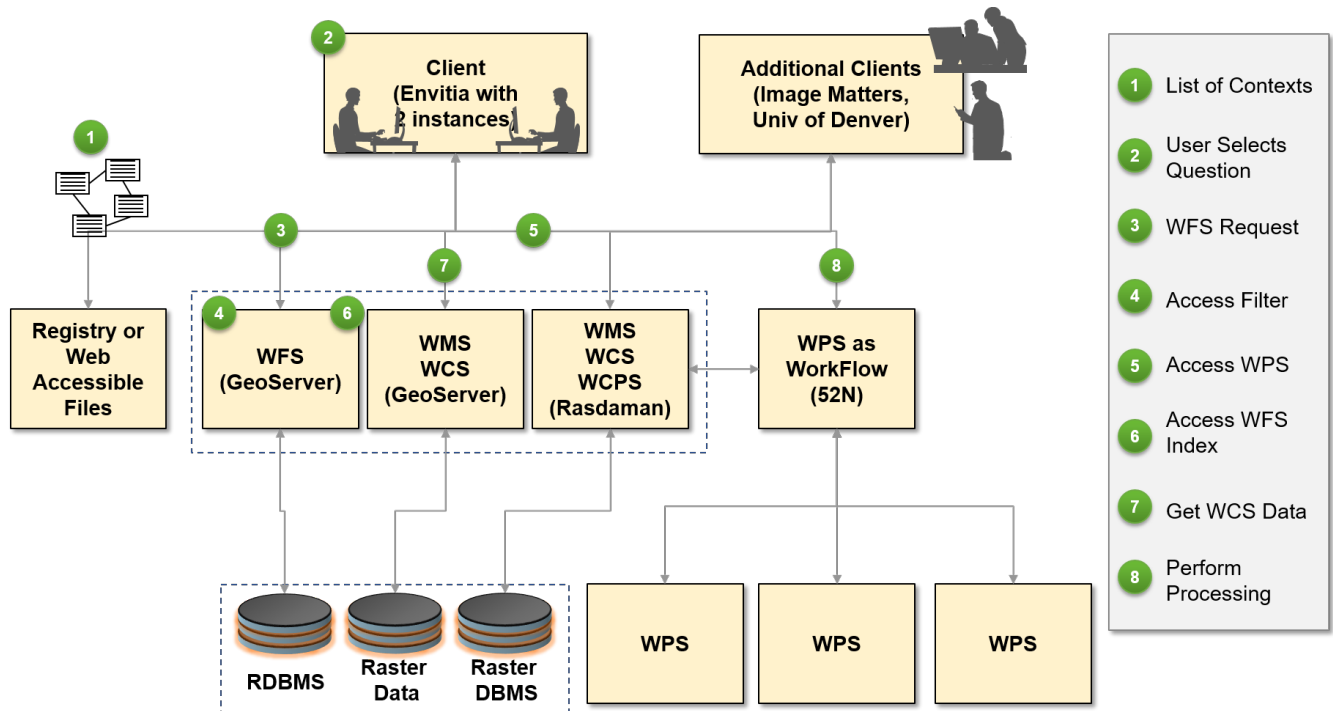


Figure 60. Technology Integration Experiments - sequence of interactions.

5.5.1. Filters and Encodings

As discussed, the Testbed 13 Component Implementation Design for FFP proposed to have an OWS Context document providing links with CQL expressions for filtering, along with ordering similarly using the KVP syntax, be it as native (WFS) or vendor extension (WMS/WCS).

Note that the layers in question might be time/elevation enabled at the WMS level, if that is the case time could be interfering with filtering, there are a few possible options.

The OWS context can deliberately use TIME, ELEVATION and custom dimensions to perform filtering in case it makes sense and the eventual complex filter to be encoded can be seen as an “and” between the dimensions and the other elements being filtered. For example,

```
time > startDate and cloudCover < 10
```

Could be encoded in WMS GetMap as:

```
time=startDate/current&cql_filter=cloudCover < 10
```

Alternatively, the OWS Context can decide to use CQL fully for the filter, either because the filter is complex enough that it cannot be represented as a simple logical conjunction, or out of a desire of consistency between protocols (using the exact same CQL syntax among them). In this case if layers are dimension-enabled, the dimension interference can be eliminated by specifying a very large

range, e.g. the above request could be encoded in WMS as

```
time=1900/current&cql_filter=time > startDate and cloudCover < 10
```

With these considerations in mind Context documents were prepared that:

- Use CQL filter on WFS, WMS and WCS to limit the data (the actual dataset spans into 2017). The CQL filter in WFS might be interesting in that it embeds also a bounding box filter, this happens because in WFS KVP the various filtering syntaxes (FEATURE_ID, FILTER, BBOX and, by extension, CQL_FILTER) are mutually exclusive, in accordance to the specification requirements
- Uses sortBy in WFS and WMS to sort data by most recent (currently working only in WFS this is work we have to perform on GeoServer to extend its behavior).
- Time dimension is enabled for the imagery layer, both WMS and WCS requests use the standard time filtering syntax with large ranges to effectively disable it and let the CQL filter perform the actual selection.

The Zaatari sample context document in JSON can be found in Annex A under the Title: Zaatari Refugee Camp route analysis. This document includes references to a catalog of WorldView-1 imagery as WFS and embeds CQL_Filters to help access imagery on WMS and WCS as needed.

The second sample Context Document in Annex A is under the Title: NASA Black Marble Global Timeseries for 2012-2016. NASA Black Marble is a global view of the Earth's city lights acquired by Suomi National Polar Orbiting (Suomi NPP) satellite, and this context document references this data exposed as WCS and WMS.

The third OWS Context Document in Annex A is for Daraa and includes the ability to conflate existing road data sets, providing information for route analysis. In this document, there is one WPS request specifying two input data sets - United Nations (UN) roads data and OpenStreetMap (OSM) roads data.

The encodings and filters were assessed in a series of experiments as shown in the TIE table below.




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 61. Technology Integration Experiments - filters and encodings.

Graphical representations of the results when these context documents are executed are shown in the Client Applications section.

5.5.2. Services and Data

This section discusses the FFP approach implemented in services and data provided by Testbed 13 participants.

In the scenario for this part of Testbed 13, large numbers of people have been displaced from the Daraa region of Syria to the Zaatari refugee camp in Jordan due to ongoing conflict. As the conflict ends 'de-escalation zones' are established by major powers and plans are made to return displaced people from refugee camps to the region. Understanding the situation on the ground and the infrastructure, as well as transporting these people from refugee camps into a former conflict zone is a major challenge for relief agencies. To accomplish this task, they must understand the environment and transportation infrastructure in the region between Zaatari refugee camp and the Daraa region.

In support of this challenge, satellite imagery from commercial and government organizations was deployed to support situational awareness and analysis for all types of users. Government and private industry sources of feature data were also identified and supported visualization and analysis of transportation networks, locations and capacities of key facilities, food supplies and sources, and many other key geographic features. This data was deployed on multiple OGC Web

Services including WMS, WFS and WCS from GeoSolutions, WMS and WCS from Rasdaman and WPS from 52North.

The components were assessed in a series of experiments as shown in the TIE table below.




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 62. Technology Integration Experiments - services and data.

GeoSolutions

Elements of the test architecture described above was implemented for GeoServer and pushed to the GeoServer master and deployed to the OWS playground instance available at

<http://cloudsdi.geo-solutions.it/geoserver>

This deployment makes it is possible to filter layers for WMS, WCS and WMS services with a CQL filter using the CQL_FILTER GeoServer extension. In addition, the deployment supports the ability to sort WMS raster time series on attributes using another GeoServer extension.

This section describes the data and services deployed in support of the demonstration scenario.

Data Deployed on Geoserver - Daraa and Zataari

Geospatial data for the Daraa region was deployed using a combination of existing OGC standards, and new techniques to make access for users and client applications more dynamic and useful, and to support analysis of multi-regional / multi-national operations and messaging related to the

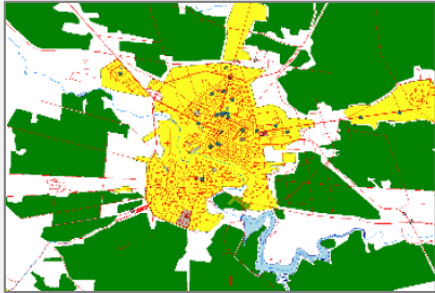
displacement and mass movement of populations in response to conflict.

The following two graphics and sections summarize the data deployed in this part of the information exchange platform.

Component: WFS with CQL_Filter capacity

Daraa Data Deployed: NGA TDS OSM data as WFS including InformationPnt, RecreationSrf, StructurePnt, StructureSrf, TransportationGroundCrv, TransportationGroundSrf, UtilityInfrastructureSrf, osm_points. Shapefiles for cultivated areas coming from DigitalGlobe.

Zaatari Data Deployed: WFS Service layer for the catalog of WorldView imagery over Zaatari with search parameters for time, cloud_cover, sun_azimuth, sun_elevation etc. Time values span from 2013 to 2017.



Daraa NSG TDS OSM



WFS Layers w/Imagery Metadata

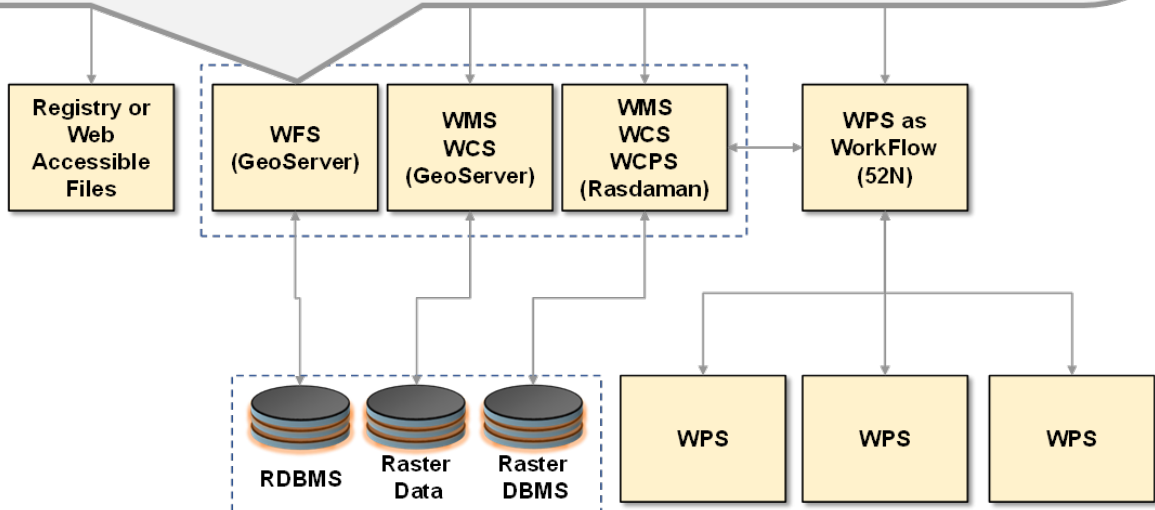


Figure 63. Data deployed on Geoserver WFS.

Component: WMS and WCS with CQL_Filter capacity

Zaatari and Daraa Data Deployed: WorldView (50cm resolution) scenes for Daraa, full 8 bands 16 bits which are exposed as WMS and WCS. Data for WMS and WCS may be accessed using same CQL_Filters as WFS. Landsat 8 imagery covering a large area (with 30m resolution), including Daraa and Zaatari refugee camps, spanning from March 2017 to June 2017. Black Marble data showing changes in lights at night (2012-20126).

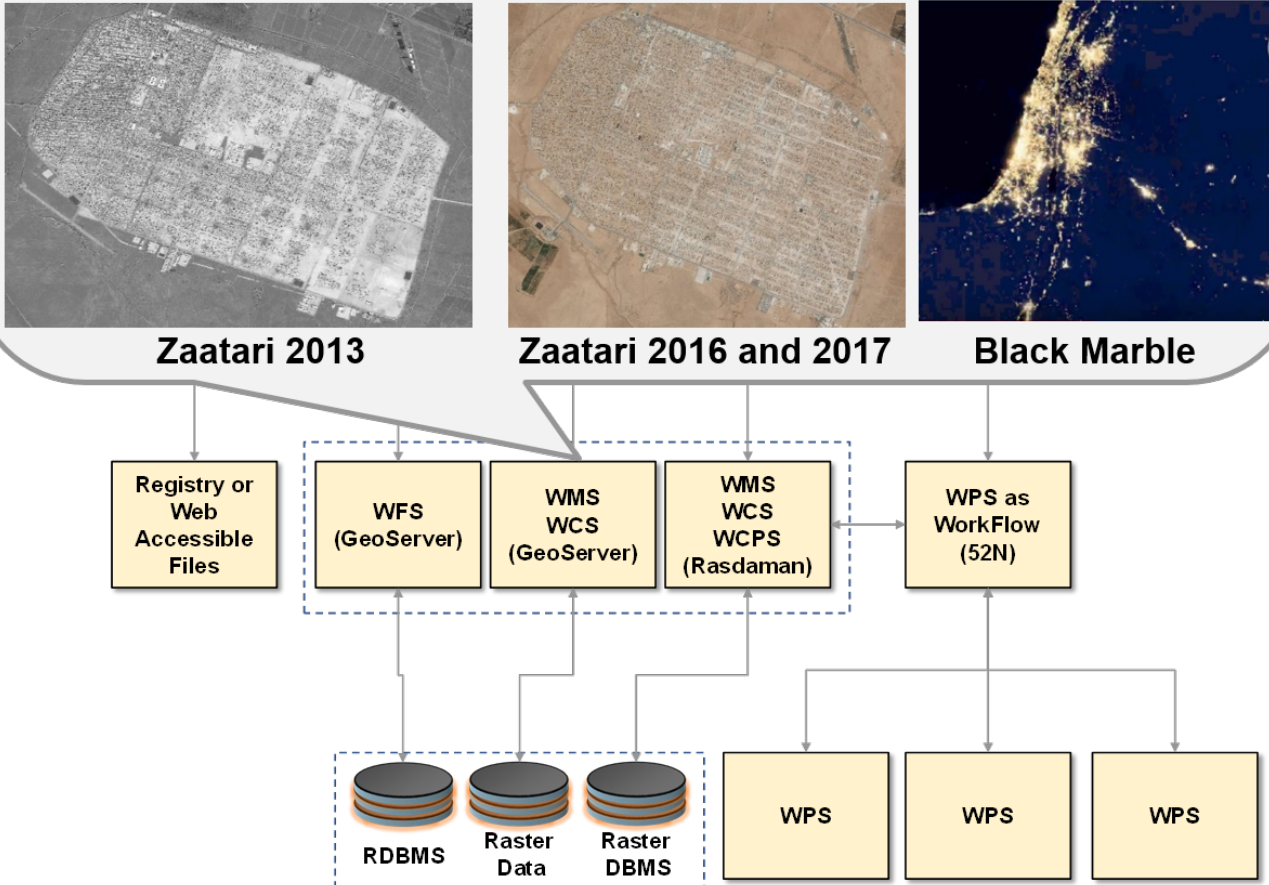


Figure 64. Data deployed on Geoserver WMS and WCS.

Data for Daraa Region Deployed on Geoserver

Initial sets of data were deployed on Geoserver at the following URL:

<http://cloudsdi.geo-solutions.it/geoserver/>

A complete description of the CQL_Filters deployed may be found in the Fit-for-Purpose Testbed 13 ER. Thanks to this work it is possible to use a WFS to store imagery metadata and for the metadata to simplify access to imagery on WMS and WCS and WMS services. This is accomplished by a CQL filter using the CQL_FILTER GeoServer extension. In addition, Geoserver supports the ability to sort WMS raster time series on attributes using another GeoServer extension.

This information was captured in the OWS Context document here:

<http://cloudsdi.geo-solutions.it/owscontext/daraa.json>

For Daraa OSM data from NGA and shapefiles for cultivated areas coming from DigitalGlobe were loaded. They are exposed both as WMS and WFS. Participants also loaded multiple WorldView 3 scenes (full 8 bands 16 bits) which were exposed as WMS and WCS as well as several True Color

Landsat 8 scenes (bands 4,3,2); some sample queries for WMS are shown below:

WorldView

```
http://cloudsdi.geo-solutions.it/geoserver/daraa/wms?service=WMS&version=1.1.0&
request=GetMap&layers=daraa:daraa_worldview_timeseries&styles=&
bbox=36.20134352401918,33.01777873311933,36.29241382075683,33.20029119387408&width=383&
height=768&srs=EPSG:4326&format=application/openlayers
```

Landsat 8

```
http://cloudsdi.geo-solutions.it/geoserver/daraa/wms?service=WMS&version=1.1.0&
request=GetMap&layers=daraa:daraa_landsat8_timeseries&styles=&
bbox=34.553315121788614,32.099447041750906,37.08220716431518,34.234622620081524&
width=768&height=648&srs=EPSG:4326&format=application/openlayers
```

NGA vector data over Landsat 8 imagery

```
http://cloudsdi.geo-solutions.it/geoserver/daraa/wms?service=WMS&version=1.1.0&
request=GetMap&layers=daraa:Daraa-OSM&styles=&
bbox=34.553315121788614,32.099447041750906,37.08220716431518,34.234622620081524&
width=768&height=648&srs=EPSG:4326&format=application/openlayers
```

NGA vector data is exposed as different layers, the list can be retrieved with a WFS

```
http://cloudsdi.geo-solutions.it/geoserver/daraa/ows?service=wfs&version=1.3.0&
request=GetCapabilities
```

Landsat 8 imagery covers a large area (with 30m resolution), including Daraa and Zaatari refugee camps, spanning from March 2017 to June 2017. The data is exposed as a GeoServer time series layer accessible via WMS and WCS: daraa_landsat8_timeseries; its WMS GetCapabilities link is shown below:

```
http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries/wms?service=WMS&
version=1.3.0&request=GetCapabilities
```

The layer exposes a time dimension as follows. Notice the default value for the TIME dimensions which is used in case no value is specified.

```
<Dimension name="time" default="2017-06-29T00:00:00Z" units="ISO8601">
2017-03-09T00:00:00.000Z,2017-03-25T00:00:00.000Z,2017-04-10T00:00:00.000Z,2017-04-
26T00:00:00.000Z,2017-05-12T00:00:00.000Z,2017-05-28T00:00:00.000Z,2017-06-
13T00:00:00.000Z,2017-06-29T00:00:00.000Z
</Dimension>
```

The properties of each element in the time series plus its bounding box are accessible as a vector layer via WMS and WFS; the layer is called daraa_landsat8_timeseries_index and its GetCapabilities for WMS and WFS can be reached here below:

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/wms?service=WMS&version=1.3.0&request=GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/wfs?service=WFS&version=2.0.0&request=GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/wms?service=WMS&version=1.3.0&request=GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/wfs?service=WFS&version=2.0.0&request=GetCapabilities

In particular, the DescribeFeatureType call can be found here:

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/wfs?service=WFS&version=2.0.0&request=DescribeFeatureType&typeName=daraa:daraa_landsat8_timeseries_index

Attributes indicated above are available for querying via WFS. CQL_FILTER can be used to jump between different services to access the same data. An example is shown below:

For example the daraa_landsat8_timeseries_index layer can be queried using WFS and a CQL filter with JSON output format to access data where sun_elevation > 51 and cloud_cover < 1 AND cloud_cover_land < 0.21

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/wfs?service=WFS&version=2.0.0&request=GetFeature&typeName=daraa:daraa_landsat8_timeseries_index&CQL_FILTER=sun_elevation%3E51%20AND%20cloud_cover%20%3C%201%20AND%20cloud_cover_land%20%3C%200.21&outputFormat=application/JSON

Using the same filter we can jump to WMS and bring up the imagery associated to this same filter:

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=daraa%3Adaraa_landsat8_timeseries&SRS=EPSG%3A4326&WIDTH=768&HEIGHT=648&BBOX=36.06820106506348%2C32.591285705566406%2C36.13411903381348%2C32.64690399169922&CQL_FILTER=sun_elevation%3E51%20AND%20cloud_cover%20%3C%201%20AND%20cloud_cover_land%20%3C%200.21&time=P1Y/PRESENT

As well as to WCS:

[http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=daraa_landsat8_timeseries&subset=Long\(36.06,36.14\)&subset=Lat\(32.59,32.65\)&subset=time\(%222017-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22\)&CQL_FILTER=sun_elevation%3E51%20AND%20cloud_cover%20%3C%201%20AND%20cloud_cover_land%20%3C%200.21](http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=daraa_landsat8_timeseries&subset=Long(36.06,36.14)&subset=Lat(32.59,32.65)&subset=time(%222017-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22)&CQL_FILTER=sun_elevation%3E51%20AND%20cloud_cover%20%3C%201%20AND%20cloud_cover_land%20%3C%200.21)

Please notice that the time parameter is set to P1Y/PRESENT to encompass all the available imagery.

The result is shown below.



Figure 65. Result of CQL Filter.

It is important to point out that the TIME dimensions applies a restriction to the provided CQL_FILTER (it is ANDed with the latter); the CQL_FILTER is a GeoServer vendor extension while the TIME dimension is standard WMS hence participants decided to adopt this behavior to not confuse users. Since the time dimension is assigned a default value if we don't specify anything such value will be assigned by default to the provided CQL_FILTER, hence not specifying the time might lead to confusing results. As an instance if we remove the TIME from the WMS query above, the default time value will trigger but since there is no combination that corresponding to the intersection of the provided CQL_FILTER with the default time an empty image will be returned. See link below:

http://cloudsdi.geo-solutions.it/geoserver/daraa/daraa_landsat8_timeseries/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=daraa%3Adaraa_landsat8_timeseries&SRS=EPSG%3A4326&WIDTH=768&HEIGHT=648&BBOX=36.06820106506348%2C32.591285705566406%2C36.13411903381348%2C32.64690399169922&CQL_FILTER=sun_elevation%3E51%20AND%20cloud_cover%20%3C%201%20AND%20cloud_cover_land%20%3C%200.21

More information on how GeoServer supports TIME via WMS can be found at this [link](http://docs.geoserver.org/latest/en/user/services/wms/time.html) [http://docs.geoserver.org/latest/en/user/services/wms/time.html]. Additional information on CQL_FILTERING can be found below:

Data for Zaatari Deployed on Geoserver

For Zaatari OSM data from NGA was loaded and exposed as WMS and WFS. The layers are the following:

- InformationPnt
- RecreationSrf
- StructurePnt
- StructureSrf
- TransportationGroundCrv
- TransportationGroundSrf
- UtilityInfrastructureSrf
- o2s_A
- o2s_L
- osm_points

Examples of WFS and WMS GetCapabilities and requests are shown below:

- WFS GetCapabilities

<http://cloudsdi.geo-solutions.it/geoserver/zaatari/ows?service=wfs&version=2.0.0&request=GetCapabilities>

- WMS GetCapabilities

<http://cloudsdi.geo-solutions.it/geoserver/zaatari/ows?service=wms&version=1.3.0&request=GetCapabilities>

- A layer group showing worldview data as well as OSM data on top

<http://cloudsdi.geo-solutions.it/geoserver/zaatari/wms?service=WMS&version=1.1.0&request=GetMap&layers=zaatari:Zaatari&styles=&bbox=36.28552436828613,32.27472407452737,36.36792397545432,32.31376157457074&width=768&height=363&srs=EPSG:4326&format=application/openlayers>

WorldView Data

Raster data coming from DigitalGlobe, which was a time series of 5 elements, was also loaded. It is imagery from WorldView-1 (50cm resolution) but acquisition steps indicate only the years, 2013, 2016 and 2017. Some useful links are included below:

- WMS GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WMS&

[version=1.3.0&request=GetCapabilities](http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=1.3.0&request=GetCapabilities)

- WCS GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.1&request=GetCapabilities

The layer is configured with a TIME dimension as follows:

```
<Dimension name="time" default="2017" units="ISO8601">
2013-01-01T00:00:00.000Z,2016-01-01T00:00:00.000Z,2017-01-01T00:00:00.000Z
</Dimension>
```

Notice that the default time for the layer is configured to be the whole 2017 year since participants assumed the image would cover that time span. Similar considerations to those introduced before about time management for WMS apply in this case.

GeoSolutions also exposed its catalog together with the properties which this time series exposes as both WMS and WFS. Some useful links below:

- WMS GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/ows?service=WMS&version=1.3.0&request=GetCapabilities

- WFS GetCapabilities

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/ows?service=WFS&version=2.0.0&request=GetCapabilities

The properties accessible via the WFS layer are the following:

- time
- cloud_cover
- sun_azimuth
- sun_elevation

According to the following FeatureType Schema, obtained with this request:

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/wfs?service=WFS&version=2.0.0&request=DescribeFeatureType&typeName=zaatari:zaatari_worldview_timeseries_index&outputFormat=application/JSON

Note that JSON output is desired this time, but if the requestor removes the outputFormat the response goes back to XML.

Filtering on this data is the same as the previous example, but since the time values span from 2013

to 2017 an inclusive filter can be `time=2013/2017`. Below are some sample requests to go from the imagery catalogue to WMS and WCS for the imagery using a CQL_FILTER.

STEP 1: Filter Imagery Catalogue via WFS

Doing a GetFeature request to check which imagery respond to this characteristics: `sun_azimuth < 153 AND sun_elevation < 38`

```
http://cloudsdi.geo-solutions.it/geoserver/zaatari/ows?service=WFS&version=1.0.0&
request=GetFeature&typeName=zaatari:zaatari_worldview_timeseries_index&
outputFormat=application%2Fjson&
CQL_FILTER=sun_azimuth%20%3C%20153%20AND%20sun_elevation%3C38
```

We have two scenes.

STEP 2: Get a Map via WMS

```
http://cloudsdi.geo-solutions.it/geoserver/zaatari/wms?service=WMS&version=1.1.0&
request=GetMap&layers=zaatari:zaatari_worldview_timeseries&styles=&
bbox=36.28552436828613,32.27472407452737,36.36792397545432,32.31376157457074&width=768&
height=363&srs=EPSG:4326&format=image/jpeg&
CQL_FILTER=sun_azimuth%20%3C%20153%20AND%20sun_elevation%3C38
```

GeoServer filters on the fly the time series to show only those two images. Notice that the two images were acquired in 2017, hence, although a TIME value was not specified in the GetMap request a valid image is still returned.

STEP 3: Get data via WCS

The following request extracts the same data at lower resolution.

```
http://cloudsdi.geo-solutions.it/geoserver/zaatari/ows?service=WCS&version=2.0.0&
request=GetCoverage&coverageId=zaatari__zaatari_worldview_timeseries&
subset=Long(36.28552436828613,36.36792397545432)&
subset=Lat(32.27472407452737,32.31376157457074)&subset=time(%222017-01-
01T00:00:00Z%22,%222017-12-31T00:00:00Z%22)&
CQL_FILTER=sun_azimuth%20%3C%20153%20AND%20sun_elevation%3C38&scaleaxes=i(0.1),j(0.1)
```

SORT BY Support

This WV catalogue of imagery over Zaatari can be used to show the behavior of the sortby parameter in GeoServer for WMS and WCS since the imagery at different years does not cover exactly the same area. In the following two requests the requestor is asking GeoServer to filter results on the time property, first ascending and then descending.

```
http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/wms?
SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&
TRANSPARENT=true&STYLES&LAYERS=zaatari%3Azaatari_worldview_timeseries&
SRS=EPSG%3A4326&
bbox=36.28552436828613,32.27472407452737,36.36792397545432,32.31376157457074&width=768&
height=363&time=2010/2017&cql_filter=sun_elevation%3C54&sortby=time%20A
```



Figure 66. Filter results from Geoserver using the time property, ascending sort.

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=zaatari%3Azaatari_worldview_timeseries&SRS=EPSG%3A4326&bbox=36.28552436828613,32.27472407452737,36.36792397545432,32.31376157457074&width=768&height=363&time=2010/2017&cql_filter=sun_elevation%3C54&sortby=time%20D



Figure 67. Filter results from Geoserver using the time property, descending sort.

The same parameter can be used for WCS:

[http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long\(36.28,36.37\)&subset=Lat\(32.27,32.31\)&subset=time\(%222010-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22\)&cql_filter=sun_elevation%3C54&scaleaxes=i\(0.05\),j\(0.05\)&sortby=time%20A](http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long(36.28,36.37)&subset=Lat(32.27,32.31)&subset=time(%222010-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22)&cql_filter=sun_elevation%3C54&scaleaxes=i(0.05),j(0.05)&sortby=time%20A)

[http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long\(36.28,36.37\)&subset=Lat\(32.27,32.31\)&subset=time\(%222010-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22\)&cql_filter=sun_elevation%3C54&scaleaxes=i\(0.05\),j\(0.05\)&sortBy=time%20D](http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long(36.28,36.37)&subset=Lat(32.27,32.31)&subset=time(%222010-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22)&cql_filter=sun_elevation%3C54&scaleaxes=i(0.05),j(0.05)&sortBy=time%20D)

More information on the support for sortBy in GeoServer can be found in the WFS reference [here](http://docs.geoserver.org/stable/en/user/services/wfs/reference.html) [http://docs.geoserver.org/stable/en/user/services/wfs/reference.html]. All the properties that are exposed to the time series catalogue (so also cloud cover as an instance) can be used to sort the results.

Landsat 8 Data

The data described above covers also Zaatari, hence the same description applies in this context.

WFS Filtering for Satellite Imagery Data - Expert Users

A common activity for an Expert geospatial data user is to connect to a catalog of imagery metadata via WFS, select a subset based on the scenes characteristics and then trigger WMS for viewing, WCS for download and also WPS for processing. This flow differs from 'normal' OGC approaches in that the WFS is used to store metadata as features. This approach is actually common in large scale operational deployments.

An example based on the Zaatari data from WorldView is shown below:

STEP 1: Load the proper OWS Context

STEP 2: Select and load on map the WFS Service layer for the catalogue of WorldView imagery over Zaatari

http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/wfs?request=getcapabilities&service=wfs&version=2.0.0

http://cloudsdi.geo-solutions.it/geoserver/zaatari/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=zaatari:zaatari_worldview_timeseries_index&maxFeatures=50&outputFormat=application%2Fjson

The catalog is shown here below by using the GeoServer internal preview. This is the overall catalog from DigitalGlobe. It is a few scenes at different points in time, more or less covering the same area. See the available properties in the graphics below showing the footprints from the catalog of Worldview over Zaatari with the available filtering properties:

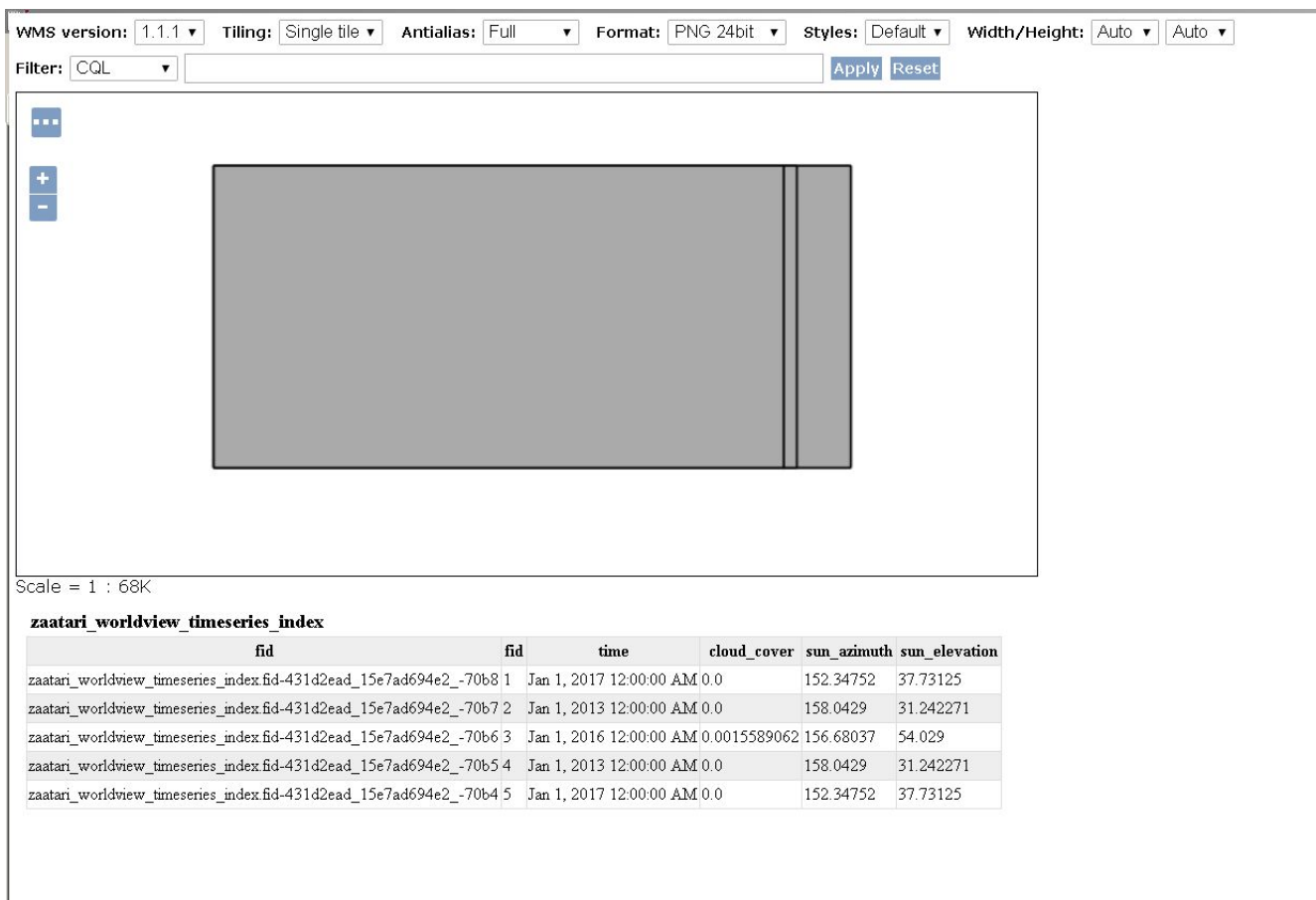


Figure 68. Metadata for satellite imagery over Zaatari on WFS and WMS.

The requestor can now filter on cloud cover, sun azimuth and time to extract 2 scenes with no cloud cover: `cloud_cover <= 0` and `sun_azimuth > 158` and time after `2012-01-01T00:00:00.000` as in the following example:

The graphic and examples below shows the footprints after having applied a CQL Filter:

WMS version: 1.1.1 ▼ Tiling: Single tile ▼ Antialias: Full ▼ Format: PNG 24bit ▼ Styles: Default ▼ Width/Height: Auto ▼ Auto ▼

Filter: CQL ▼ cloud_cover <=0 and sun_azimuth >158 and time after 2012-01-01T00:00:00.000 Apply Reset

Scale = 1 : 68K 36.33848, 32.30102

zaatari_worldview_timeseries_index

fid	fid	time	cloud_cover	sun_azimuth	sun_elevation
zaatari_worldview_timeseries_index.fid-431d2ead_15e7ad694e2_-70aa4		Jan 1, 2013 12:00:00 AM	0.0	158.0429	31.242271

Figure 69. Footprint of available satellite imagery over Zaatari after applying a CQL Filter.

WFS GetFeature Request (in GeoJSON) http://cloudsdi.geo-solutions.it/geoserver/zaatari/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=zaatari:zaatari_worldview_timeseries_index&maxFeatures=50&outputFormat=application%2Fjson&&CQL_FILTER=cloud_cover%20%3C%3D0%20and%20sun_azimuth%20%3E158%20and%20time%20after%202012-01-01T00%3A00%3A00.000

Now the requestor can go to the WMS service and filter using the same CQL_FILTER:

WMS GetMap Request

http://cloudsdi.geo-solutions.it/geoserver/zaatari/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=zaatari%3Azaatari_worldview_timeseries&CQL_FILTER=cloud_cover%20%3C%3D0%20and%20sun_azimuth%20%3E158%20and%20time%20after%202012-01-01T00%3A00%3A00.000&SRS=EPSG%3A4326&WIDTH=768&HEIGHT=363&BBOX=36.32453441619873%2C32.28879690170288%2C36.35749340057373%2C32.30437517166138

The graphic below shows the standard WMS view on the imagery associated to the catalog, unfiltered:

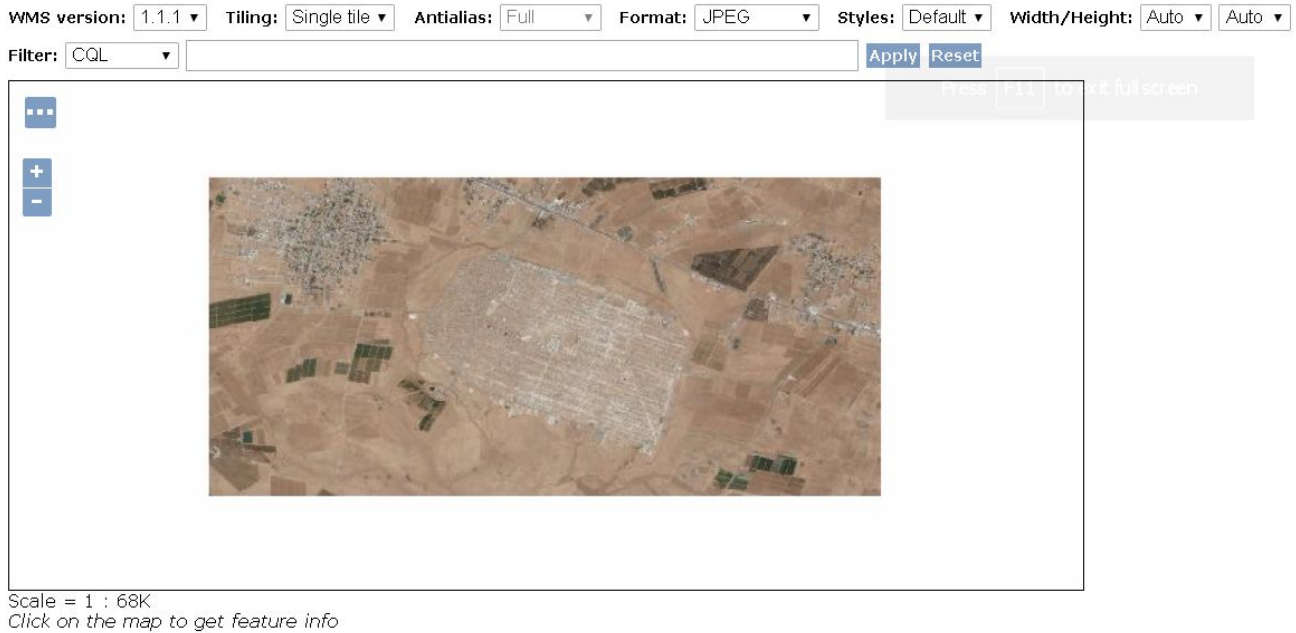


Figure 70. Unfiltered view of satellite imagery over Zaatari.

The graphic below shows the WMS view of the imagery when applying the same CQL_FILTER as above:



Figure 71. WMS view of satellite imagery over Zaatari using same CQL_FILTER as WFS.

Now the requestor goes to the WCS to get the raw data using the same CQL_FILTER:

[http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long\(36.28,36.37\)&subset=Lat\(32.27,32.31\)&](http://cloudsdi.geo-solutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long(36.28,36.37)&subset=Lat(32.27,32.31)&)

CQL_FILTER=cloud_cover%20%3C%3D0%20and%20sun_azimuth%20%3E158%20and%20time%20after%202012-01-01T00%3A00%3A00.000&scaleaxes=i(0.05),j(0.05)

Web Processing Service for Conflation

52North developed and deployed a WPS with Fit for Purpose support. The Fit for Purpose WPS was intended to support the FFP Demo Vignette, with the ability for road datasets in the area of the Zaatari refugee camp to be conflated to create a more complete dataset. Prior to the conflation, data quality checks were performed on the input datasets. The different processing steps were executed as a workflow and described in the Business Process Model and Notation (BPMN) workflow description notation. The execution was performed using the Camunda BPMN Engine and the workflow encapsulated by a WPS 2.0 process.

The process will forward the inputs to the BPMN Engine, start the workflow and return the results back to the client. Also, for dynamic creation of workflow processes by uploading BPMN to the WPS, a new operation InsertProcess will be specified. The workflow WPS process will be referenced in a OWS Context document and can be started by the FFP client. The conflated features will be created on-the-fly by the WPS. Inputs are (at least) two datasets for conflation and a reference to a vector or raster dataset for ground control.

As background, conflation, also known as map matching or merging, is the process of combining two datasets based on a set of rules to produce a more complete dataset. One example is the combination of authoritative data, e.g. TIGER road data [1: <https://www.census.gov/geo/maps-data/data/tiger-line.html>] with crowd-sourced data, e.g. OpenStreetMap [2: <http://www.openstreetmap.org/>]. The official data is collected with a high effort and follows strict requirements e.g. regarding data quality, which leads to a smaller frequency of updates. Crowd-sourced data on the other hand can be collected by virtually everyone e.g. by digitalizing streets from satellite imagery. The quality of the data is monitored by the community itself. Crowd-sourced data has therefore a high frequency of updates, with a trade off of having the danger of poor data quality for newly collected data. Several software systems exist to perform conflation tasks, e.g. the Roadmatcher software that was investigated in the OWS-9 testbed [2]. In Testbed-12 the Hootenanny conflation software was investigated regarding the feasibility for web-based conflation [3]. The software is described in the following section.

Hootenanny

Hootenanny [3: <https://github.com/ngageoint/hootenanny>] is an open source conflation tool. It features the automated and semi-automated conflation of polylines, polygons and points. It internally uses the OpenStreetMap data structure. Hootenanny software offers a command line interface. Also, a webapp is provided including a Web interface to visualize the data. In the past, a WPS 1.0.0 interface was provided as part of Hootenanny to enable web-based conflation however, the WPS 1.0.0 interface has now been deprecated [4: <https://github.com/ngageoint/hootenanny/issues/820>]. Testbed-13 used the 52 North WPS, which provides a WPS 2.0.0 interface for Hootenanny.

The following data formats are supported by Hootenanny:

Import: Hootenanny can ingest from:

- Shapefile (.shp)

- OpenStreetMap (.osm)
- ESRI File Geodatabase (.gdb)
- .zip files containing shapefiles and/or .gdb files
- geonames.org (.geonames)
- OSM API database sources

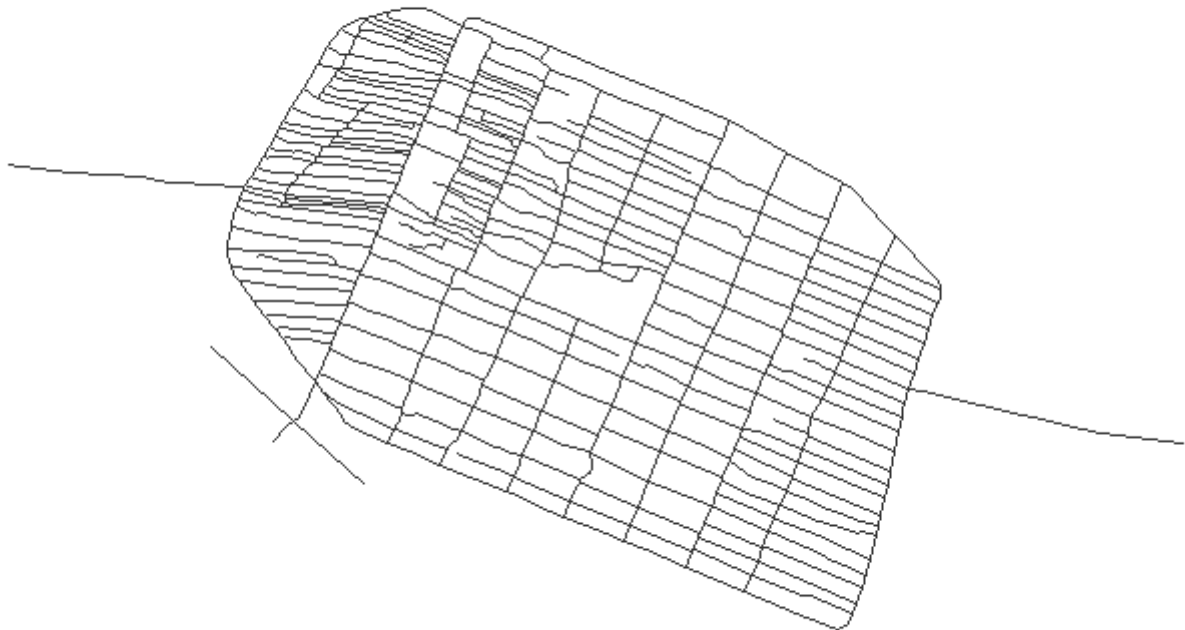
Export: Hootenanny can export to:

- Shapefile (.shp)
- OpenStreetMap (.osm)
- ESRI File Geodatabase (.gdb)
- Web Feature Service (WFS)
- OSM API database

Scenario

For the Mass Migration and Fit-for-Purpose scenario road datasets from the area of the Zaatari Refugee Camp should be conflated. For the authoritative dataset, road data from the United Nations was used footnote: [https://github.com/impact-initiatives/reach-jor-zaatari-data/raw/master/data/roads/Roads_2016_12.shp.zip?raw=true]

UN Road Data

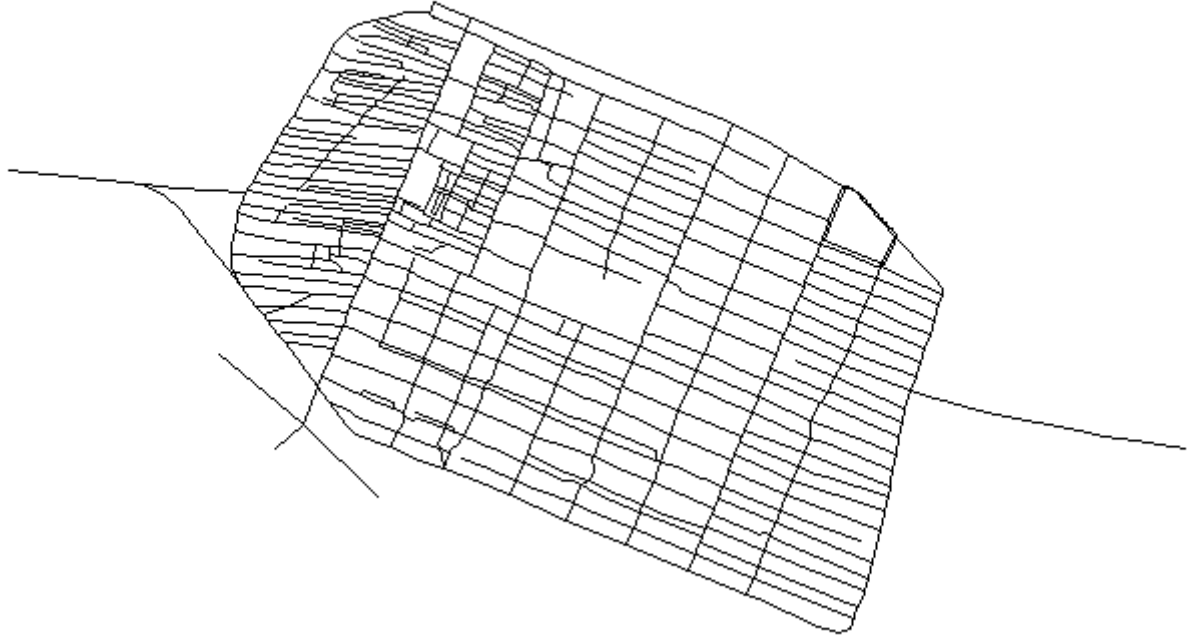


Source: https://github.com/impact-initiatives/reach-jor-zaatari-data/raw/master/data/roads/Roads_2016_12.shp.zip?raw=true

Figure 72. UN roads data for Zaatari refugee camp.

The second dataset used for the conflation was exported from OpenStreetMap (OSM) [5: <http://www.openstreetmap.org/export#map=15/32.2956/36.3382>]

OSM Road Data

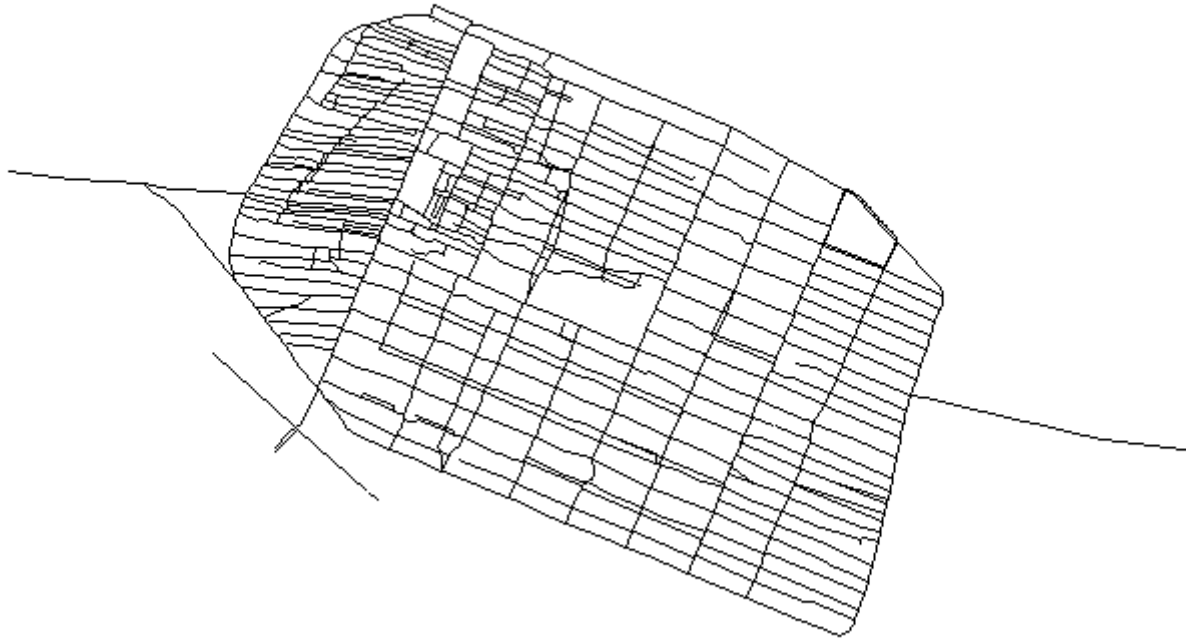


Source: <http://overpass-api.de/api/map?bbox=36.3045,32.2797,36.3719,32.3115>

Figure 73. OSM roads data for Zaatari refugee camp.

There are visible differences, the UN data is from December 2016, whereas the OSM data is constantly updated by the users. The conflated result looks like the following:

Conflated Road Data



Source: <http://tb12.dev.52north.org/SecurityProxy/service/wps>

Figure 74. Conflated roads data from WPS for Zaatari refugee camp.

The Conflation process was accessed in the Mass Migration and Fit-for-Purpose scenario by the Expert user (Envitia client) discussed below.

5.5.3. Client Applications

This section discusses the FFP approach as implemented in Client Applications provided by Testbed 13 participants.

The FFP Component Implementation Design for Client Applications recognizes two primary levels of users:

- **Non-Expert Users** - These users are not geospatial experts, and do not know about satellite imagery or complex geospatial data sources. They just have questions to enter into the platform, such as, "What are the safest routes from Zaatari refugee camp to Daraa?".
- **Expert Users** - These users are geospatial experts and understand the OGC services, satellite imagery and complex data sources in the platform. They set up the CQL-Filters and Context Documents that provide help for Non-Expert Users.

In addition, a third client was tested showing visualizations of the data deployed for Daraa and Zaatari:

- **Visualization Client** - These users access the platform from simulated Relief Operations Centers, and have at least a laptop, client software and reliable internet connections.

The client applications tested the filters and encodings in the platform as outlined in the TIE table.




Clients	Non-Expert (Envitia) 	Expert (Envitia) 	Visualization (Univ of Denver) 
Filters and Encodings			
Context Encoding	X	X	
CQL_Filters	X	X	
Services and Data			
Geoserver WMS	X	X	X
Geoserver WFS	X	X	X
Geoserver WCS	X	X	X
52N WPS	X	X	

Figure 75. Technology Integration Experiments - client applications.

Client for Non-Experts

In the demonstration scenario clients accessed the platform using client applications configured for users who are not geospatial experts, and do not know about satellite imagery or complex geospatial data sources. They just have questions to enter into the platform, such as, "What are the best routes from Zaatari refugee camp to Daraa?".

Examples of this client from Envitia are shown below:

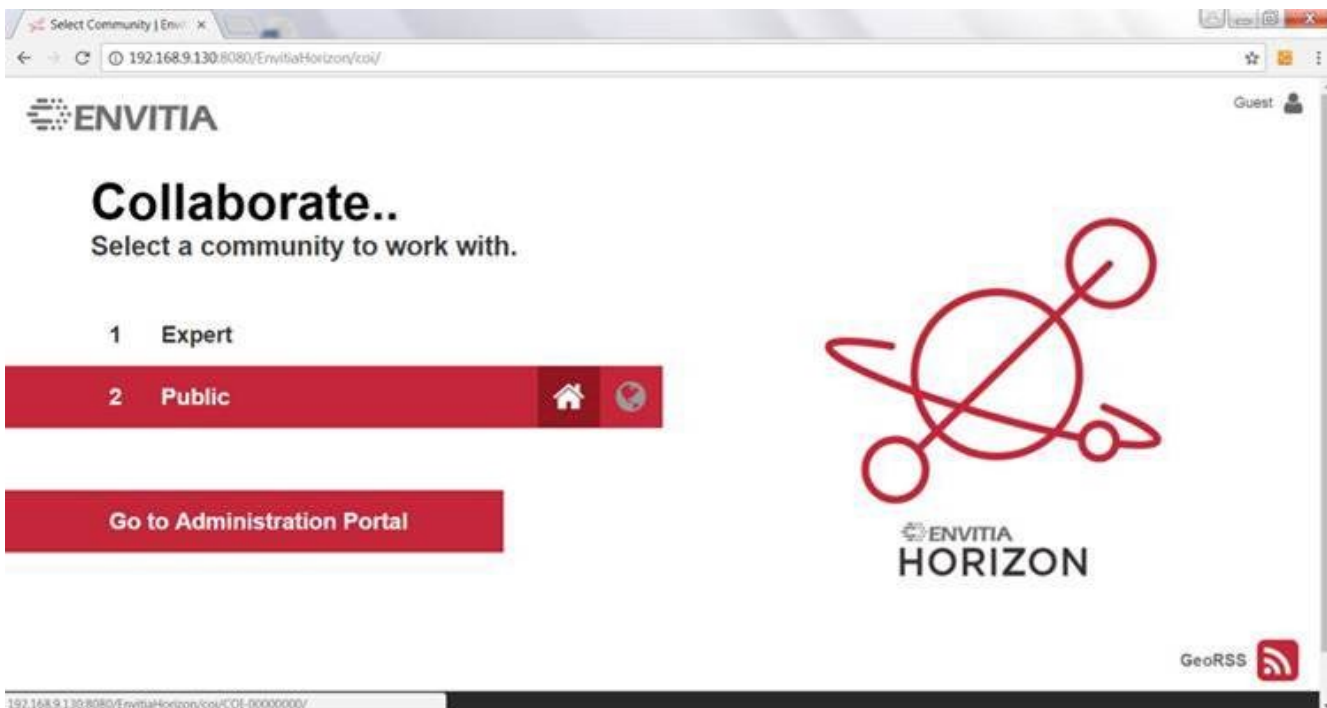


Figure 76. Client for non-experts - Public view.



Figure 77. Client for non-experts - User interface.

This application can use complex filters (CQL_Filter) and Context documents that capture collections of services for access to information on the platform - with the complexity hidden from non-expert users. All they have to do is search for OWS context documents providing a specific analytical service. For example, they may type, "What is the fastest route out of Zaatari?" They are then presented results, and assess the results via the abstracts. The application then automatically fires all the requests in the selected OWS Context document and visualizes them on screen.

Examples of this sequence are shown below:

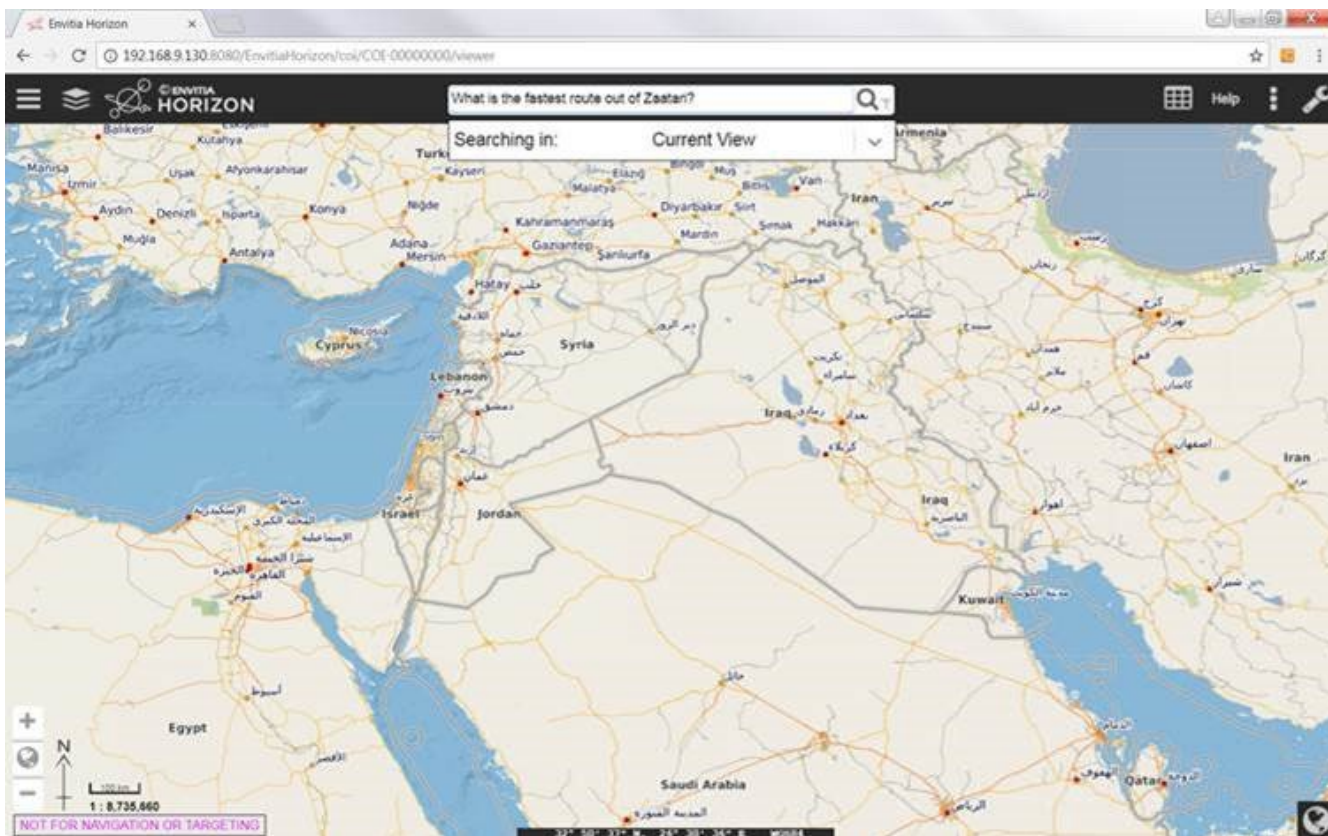


Figure 78. Entering a question to search for information about Zaatari.

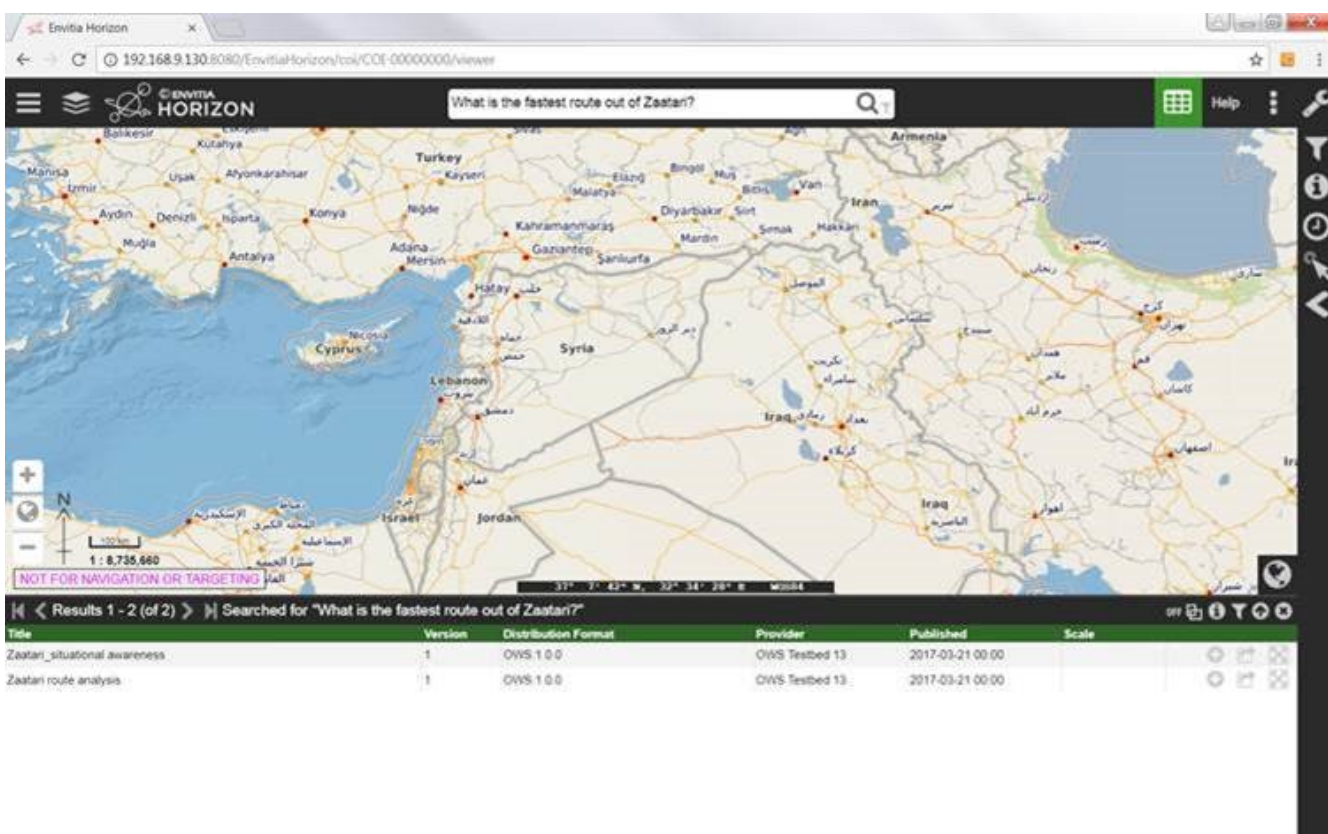


Figure 79. Results presented in response to a question.

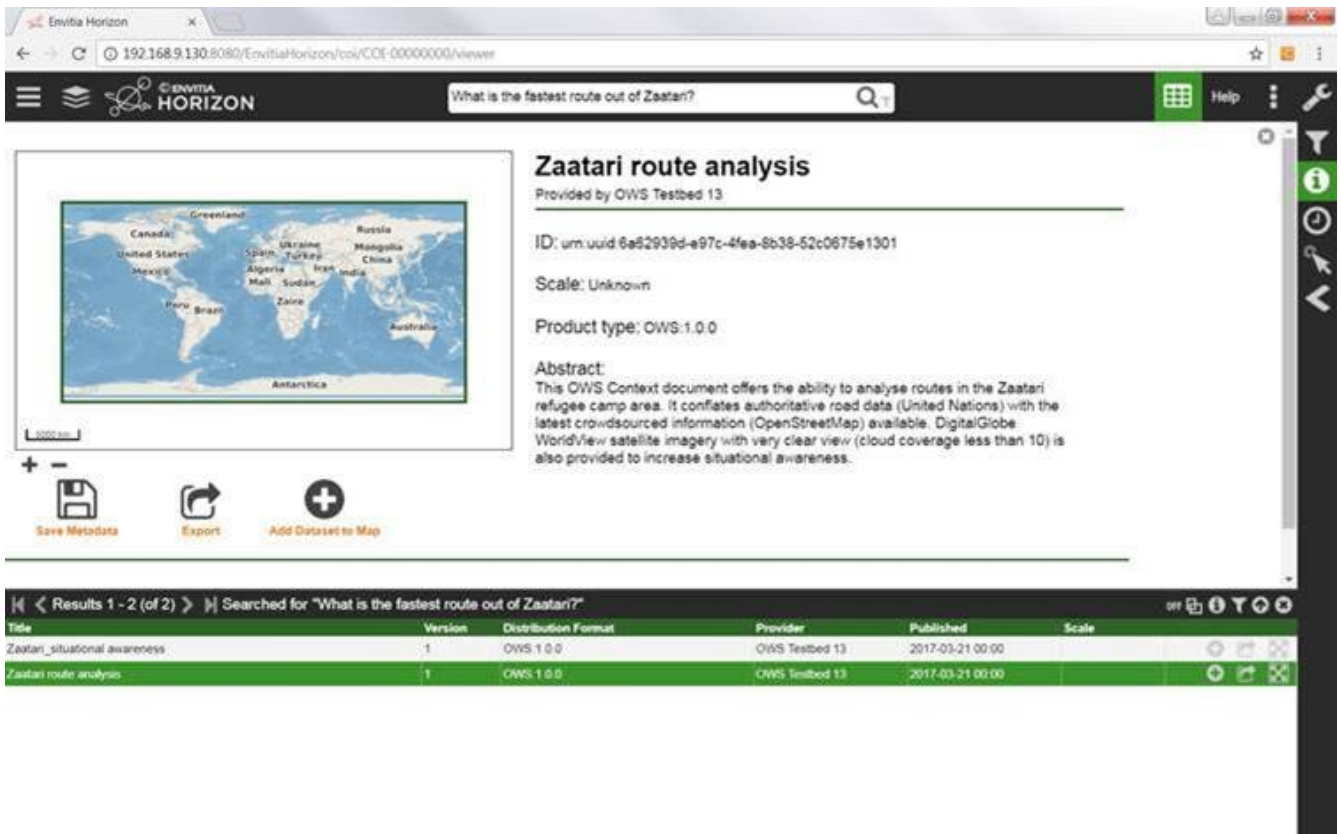


Figure 80. Results accessed via the abstracts.

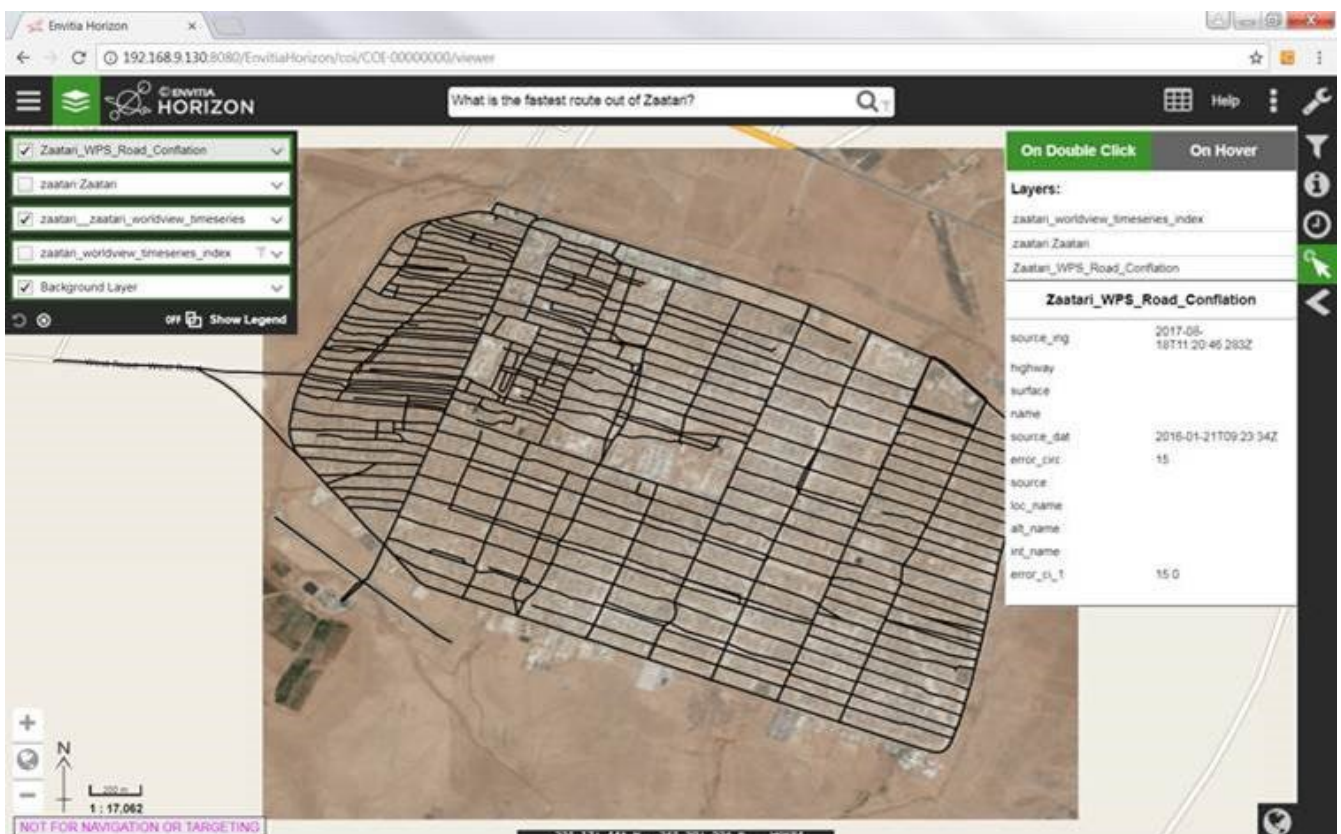


Figure 81. Application automatically requests in the selected OWS Context document, and visualizes results on screen.

Client for Geospatial Experts

This section explains how expert users set up the encodings and filters used in the application for

non-experts.

First, the user logs in to the 'Expert' capabilities of the Envitia application:

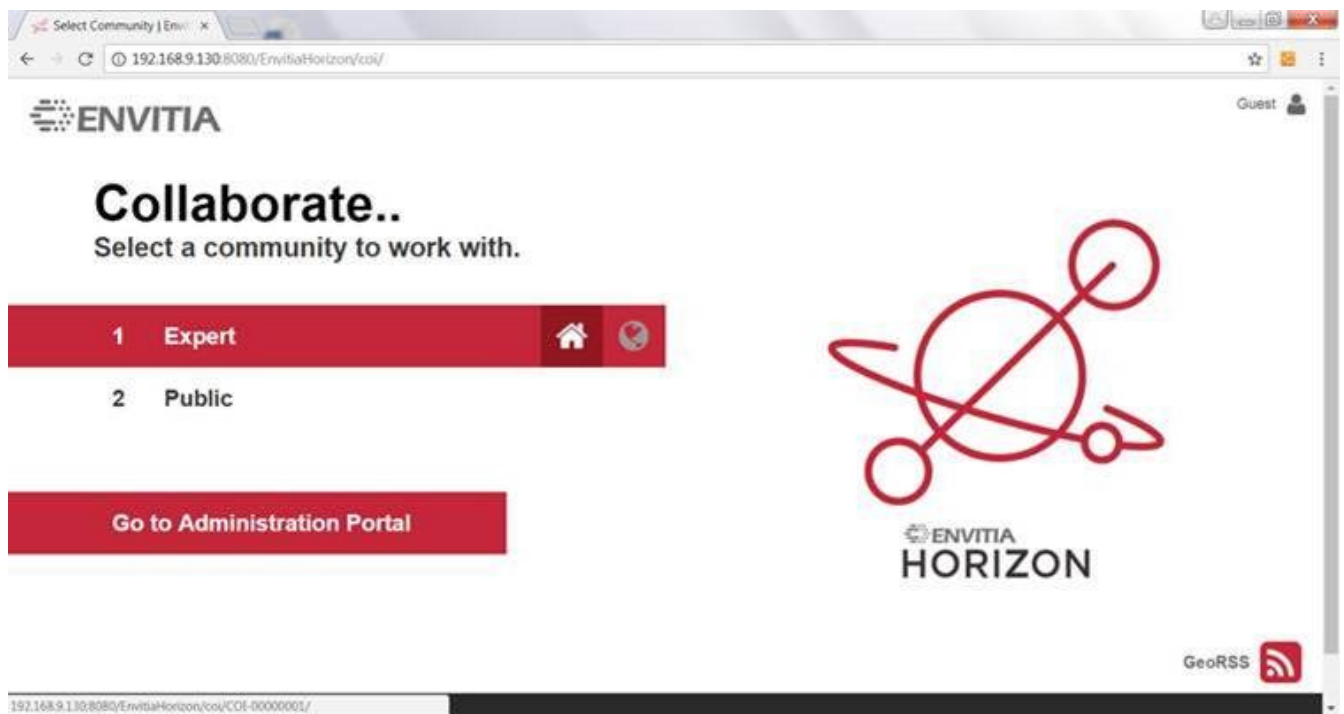


Figure 82. Client for experts - User login screen.



Figure 83. Client for experts - User interface.

They can then search for specific web services (e.g. WFS Darea Landsat 8):

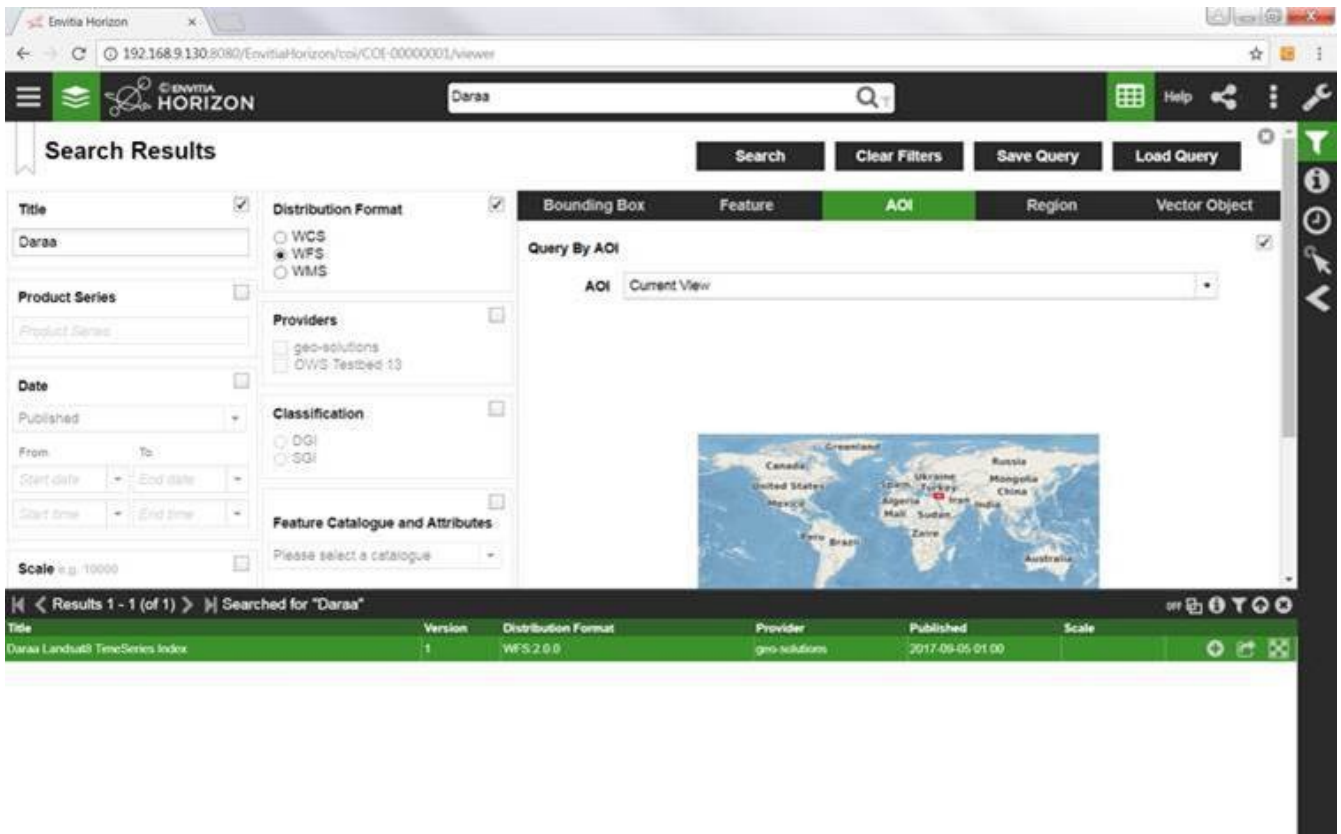


Figure 84. Searching for 'Daraa' and WFS.

They can then visualize the WFS Daraa Landsat 8 timeseries (bounding boxes and feature attributes).

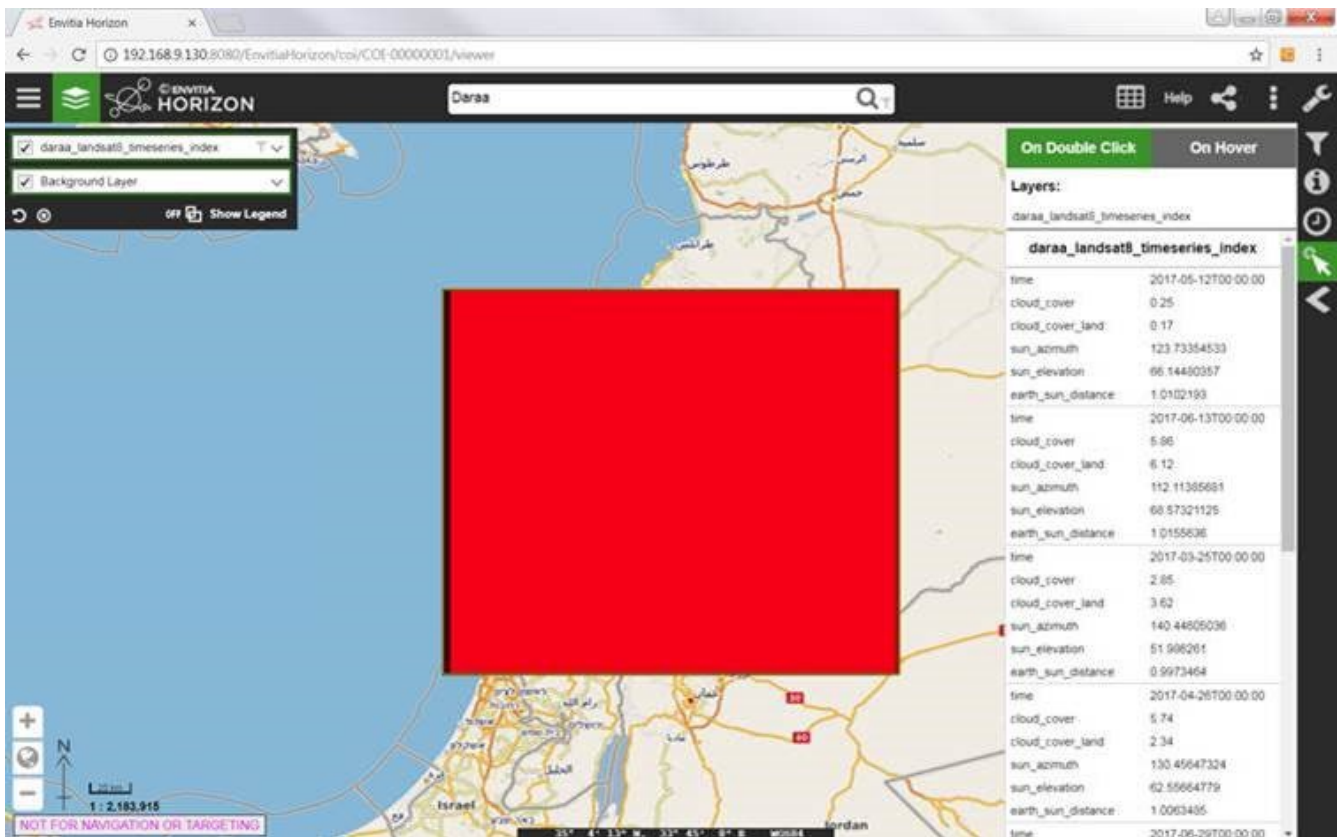


Figure 85. Visualizing metadata for Daraa Landsat timeseries from WFS.

Then they apply a CQL filter to access only features with cloud_cover<10 and earth_sun_distance<1.

The first screenshot shows filter creation, the second the results of the filter, and then the CQL filter is automatically applied to GetFeature.

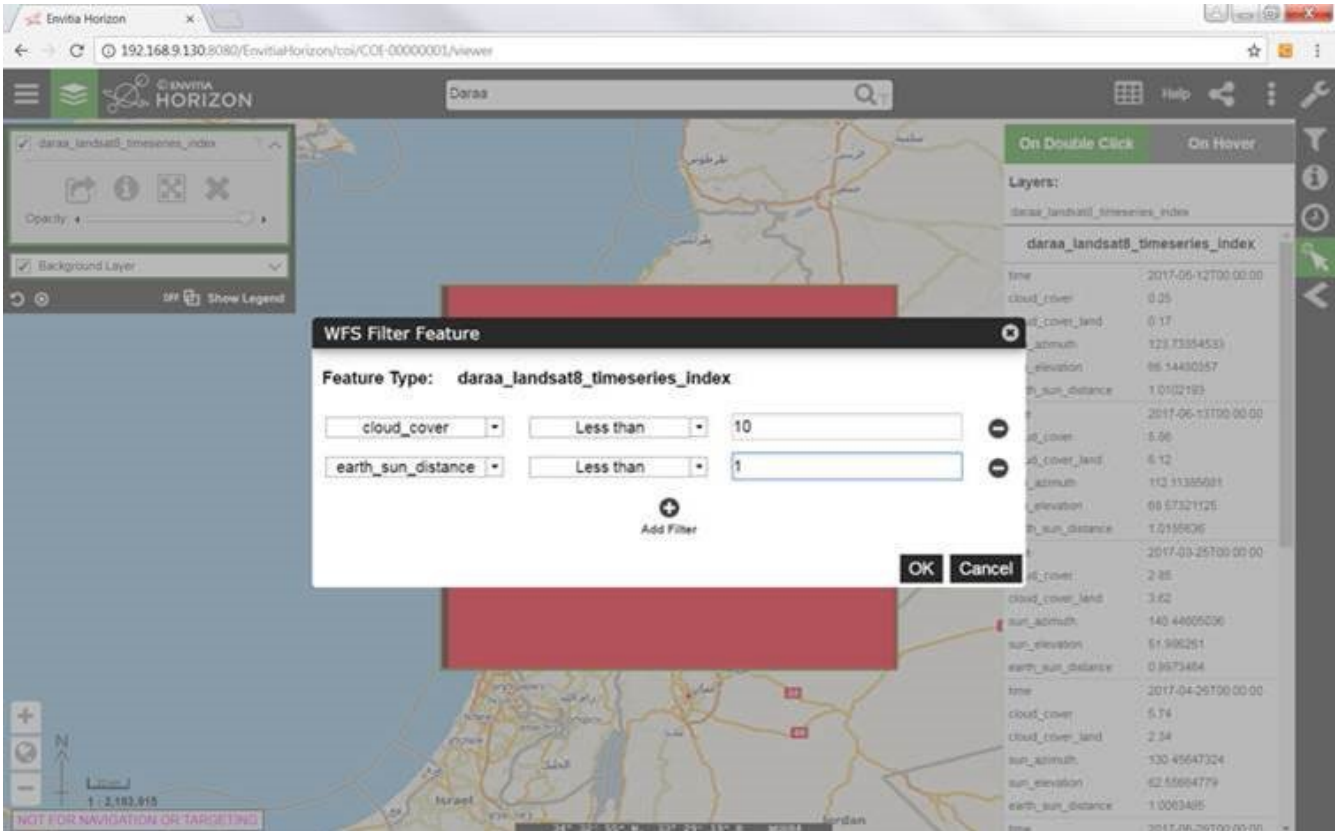


Figure 86. Applying a CQL filter to access only imagery metadata (features) with $cloud_cover < 10$ and $earth_sun_distance < 1$.

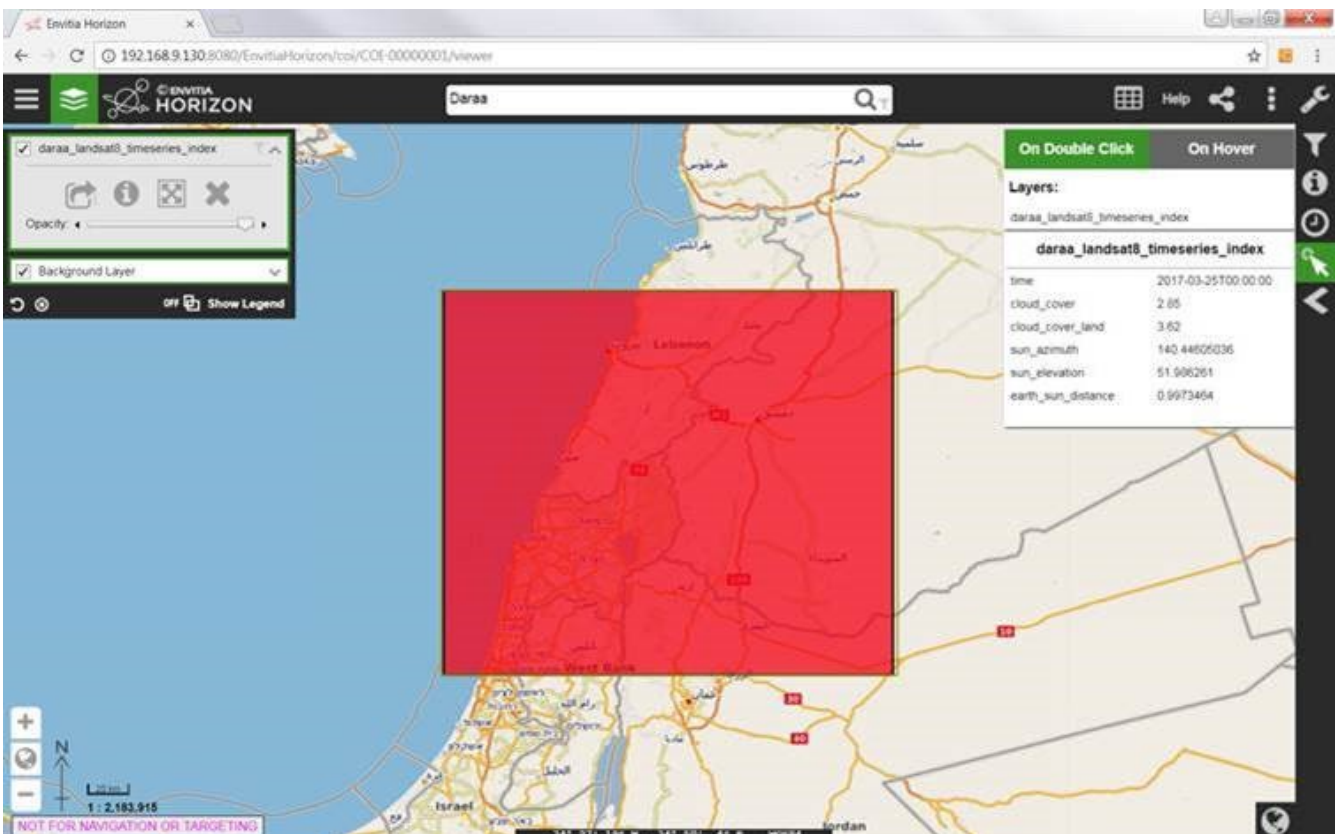


Figure 87. Reviewing results of the CQL filter.

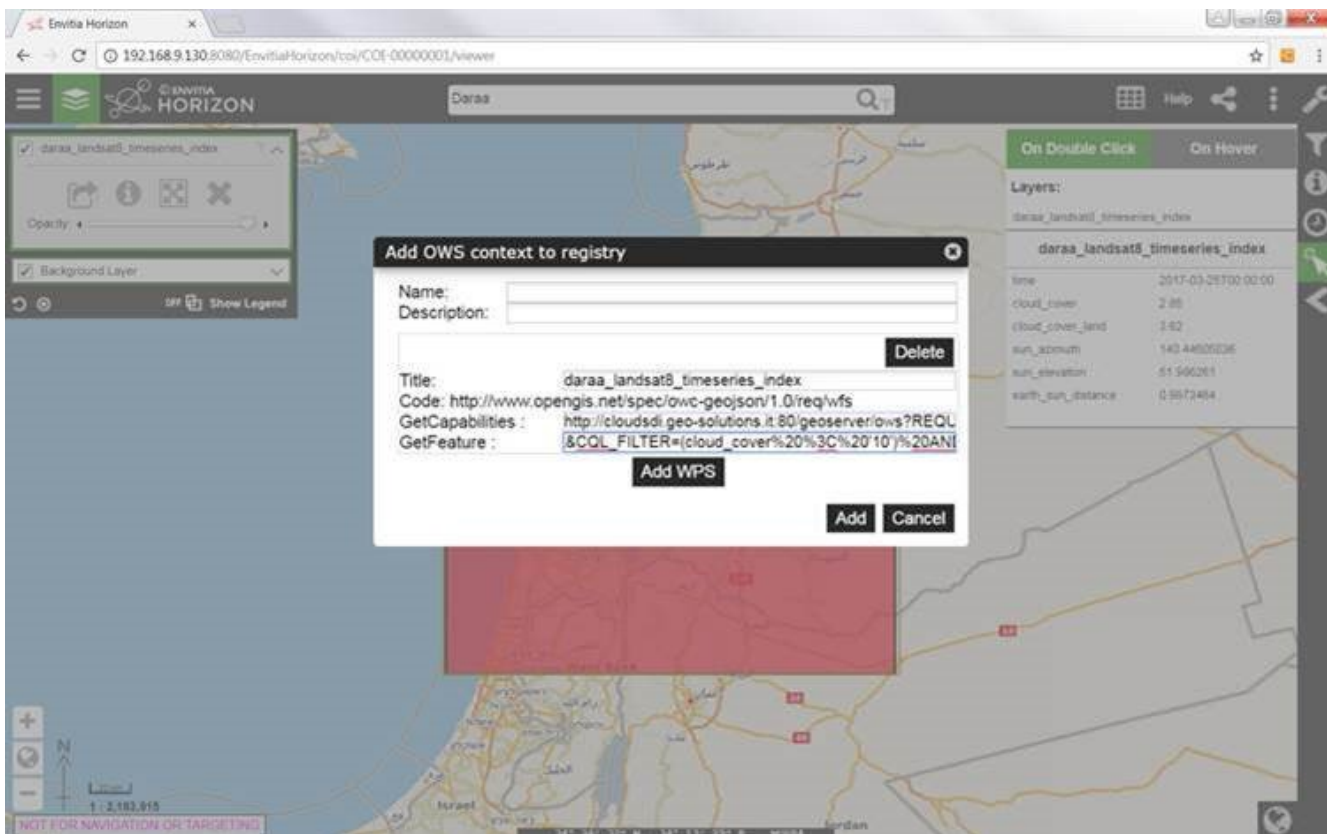


Figure 88. CQL filter automatically applied to GetFeature requests.

Below the same CQL filter has been automatically applied to the related WMS (the only image satisfying that is 2017-03-25 00:00:00).

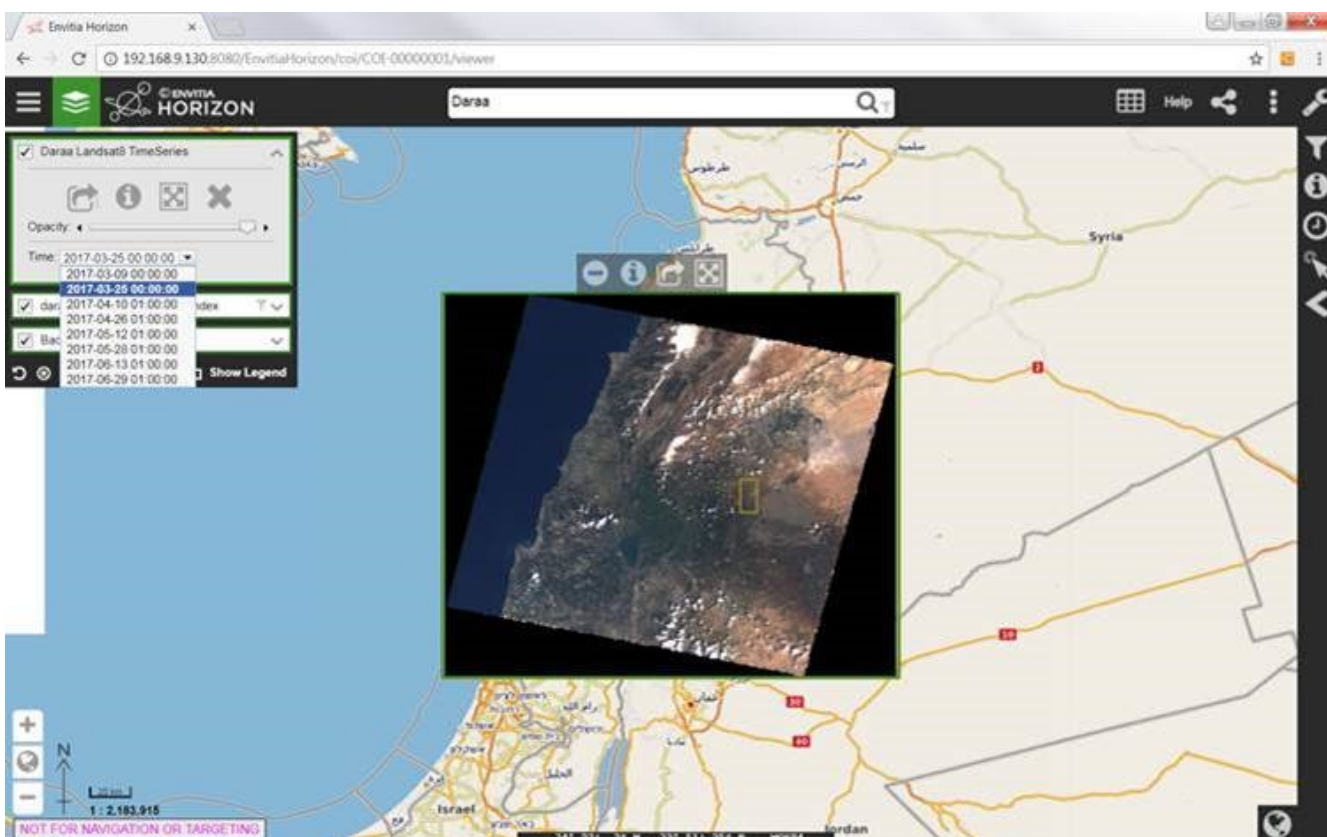


Figure 89. The same CQL filter used on imagery metadata WFS can be applied to the related WMS.

The client allows access to individual images at different timestamps of the time series via a slider.

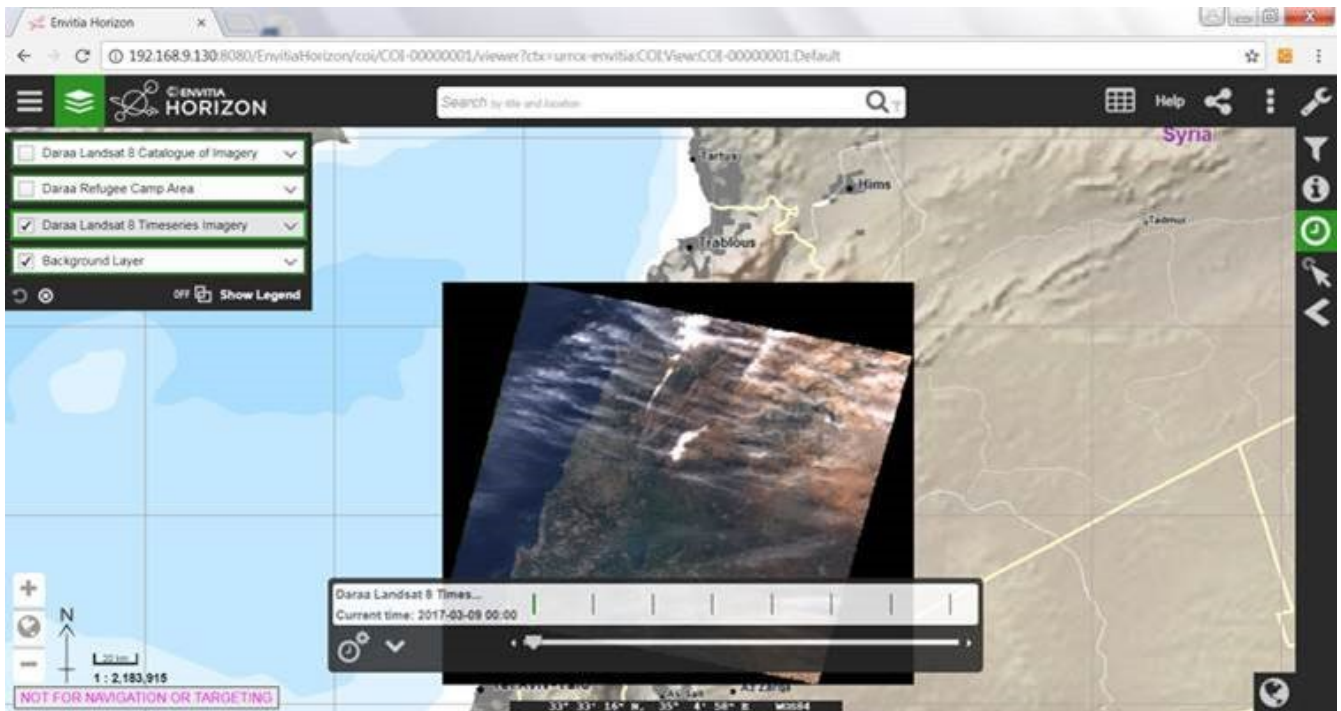


Figure 90. Individual images accessed at different timestamps of the time series via a slider.

They can search for specific web services (e.g. WCS Daraa Landsat 8 timeseries).

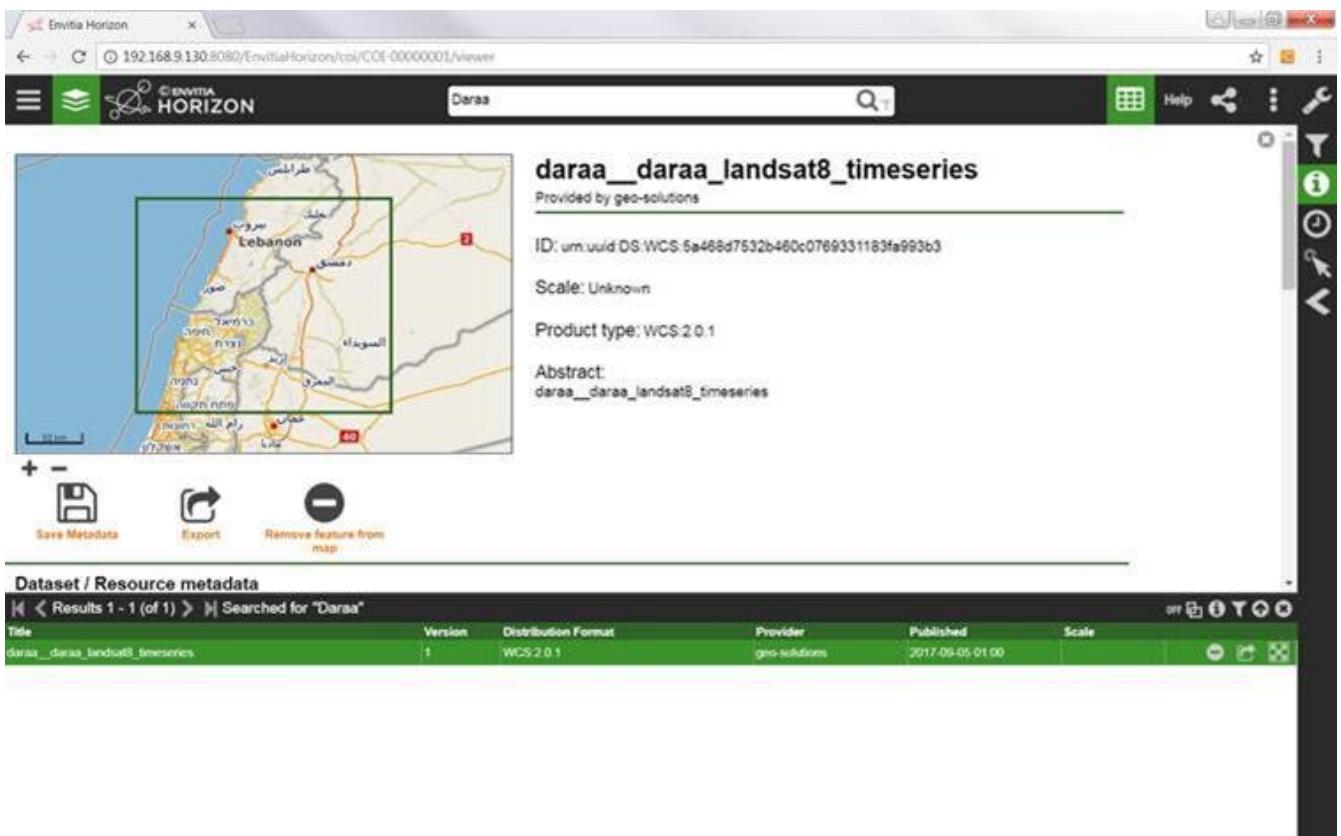


Figure 91. Searching for imagery data over Daara (WCS).

They can also subset data served by WCS using a lat/long filter. The screenshots below show the original, and then filtered using latitude and longitude:

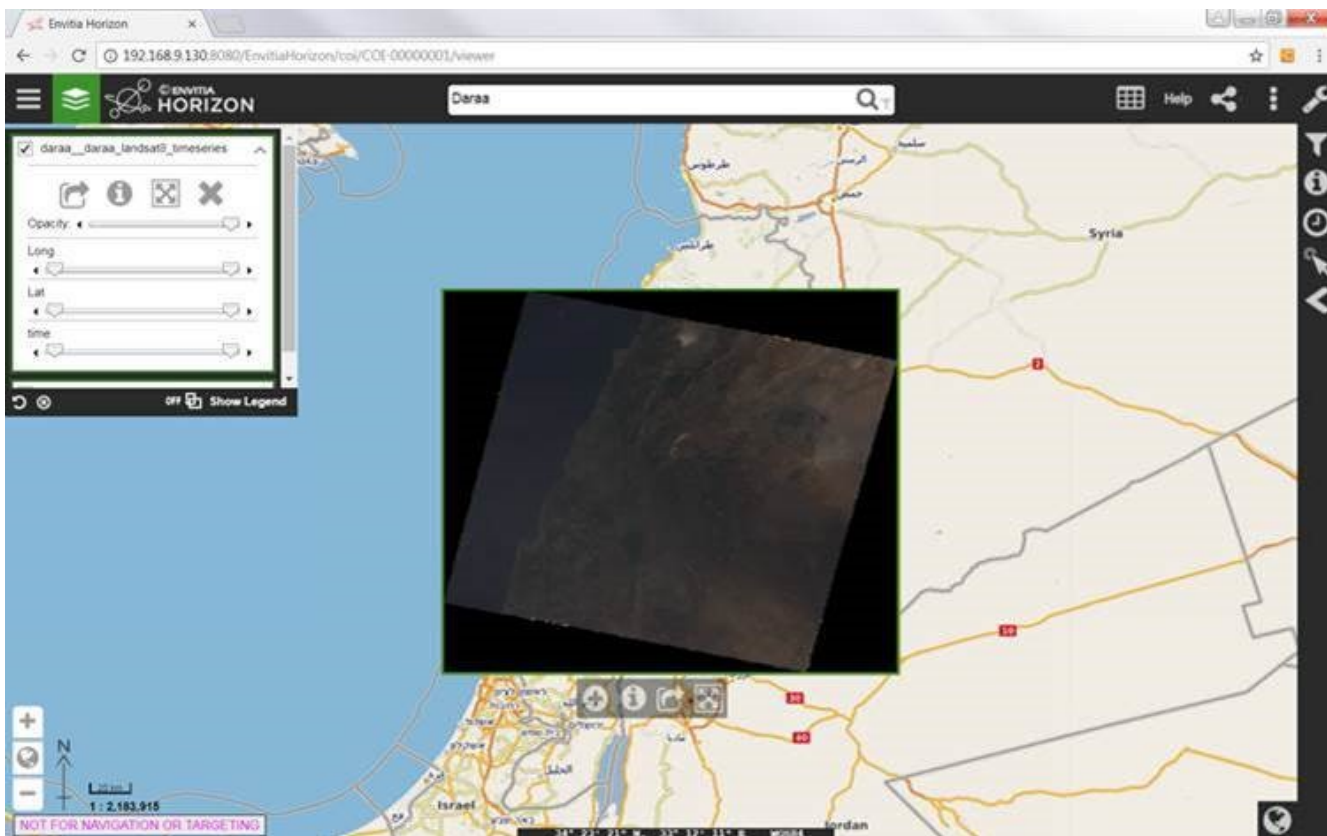


Figure 92. Unfiltered view of imagery data served by WCS.

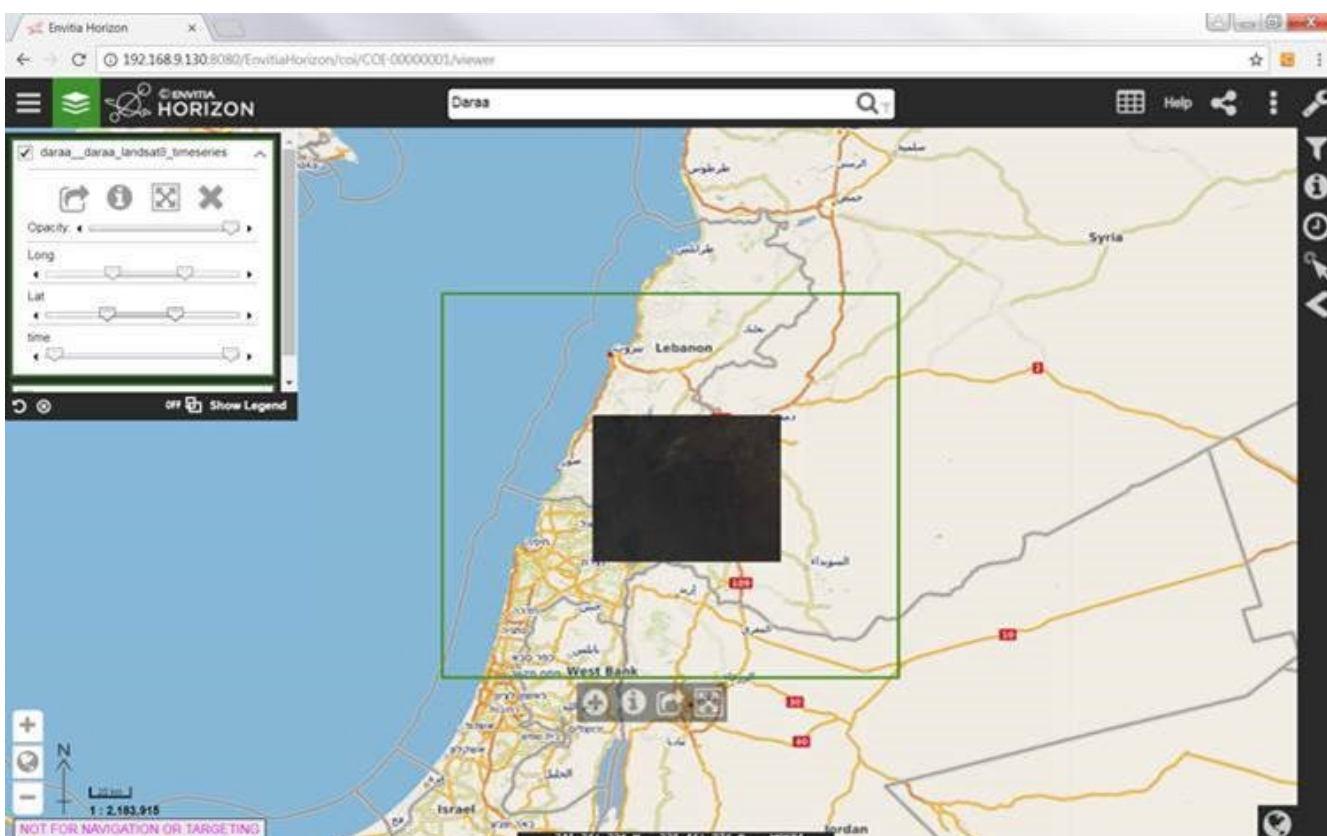


Figure 93. View of imagery data served by WCS after applying a lat/long filter.

The example below shows search for specific web services (e.g. WMS Deraa OSM).

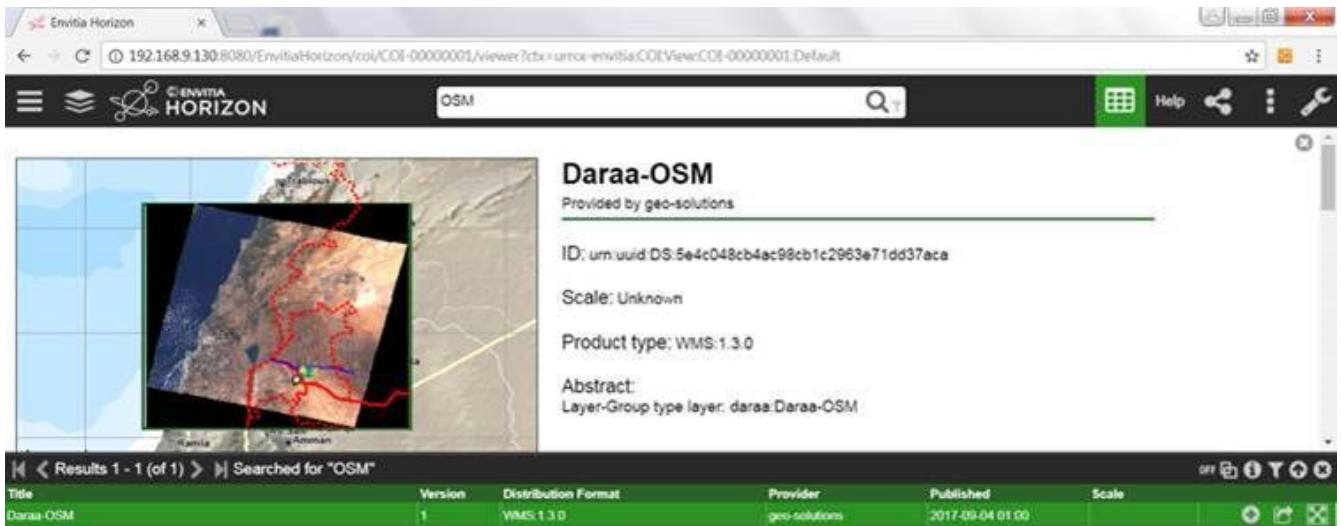


Figure 94. Searching for OpenStreetMap data.

In the following example the client is used to import a new web service and make it available to the client (e.g. WMS Daraa OSM).

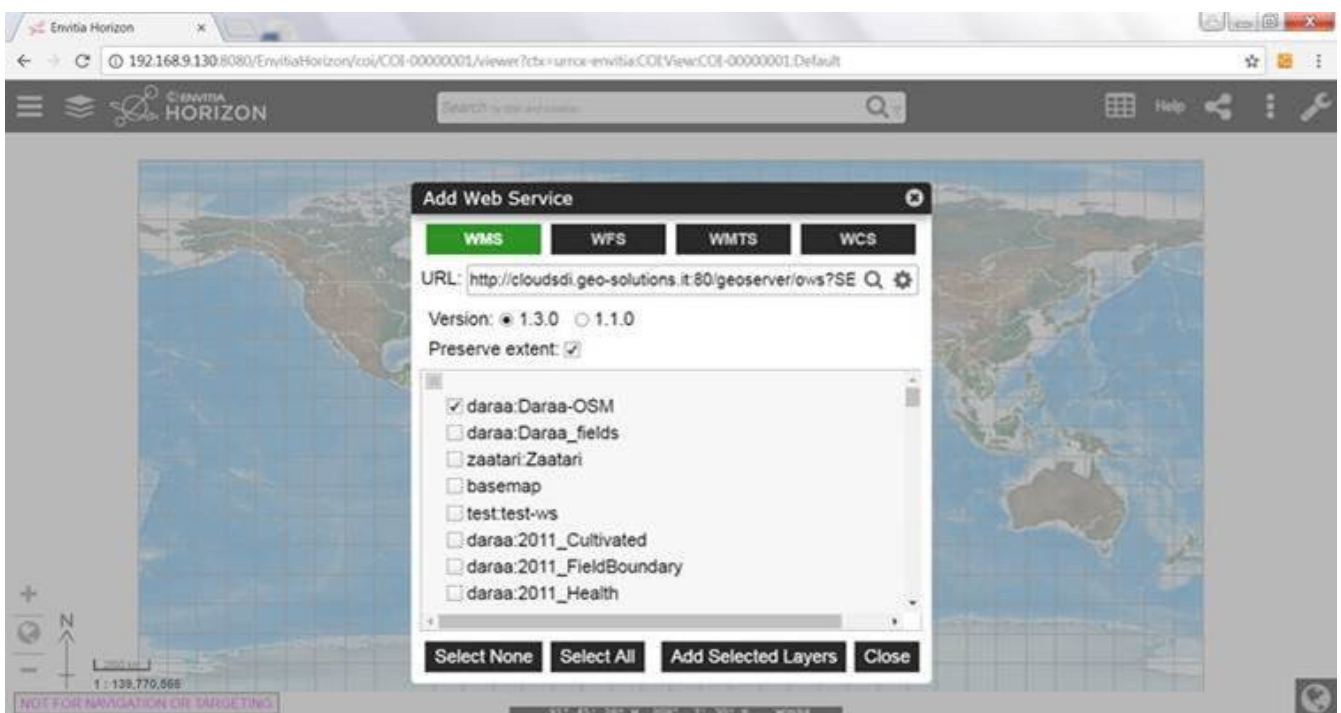


Figure 95. Importing a new web service.

In addition, the client supports creation of an OWS Context document with the discovered web services for Zaatari. Examples for WMS, WFS and WCS and then WMS, WFS, WCS and WPS are shown below.

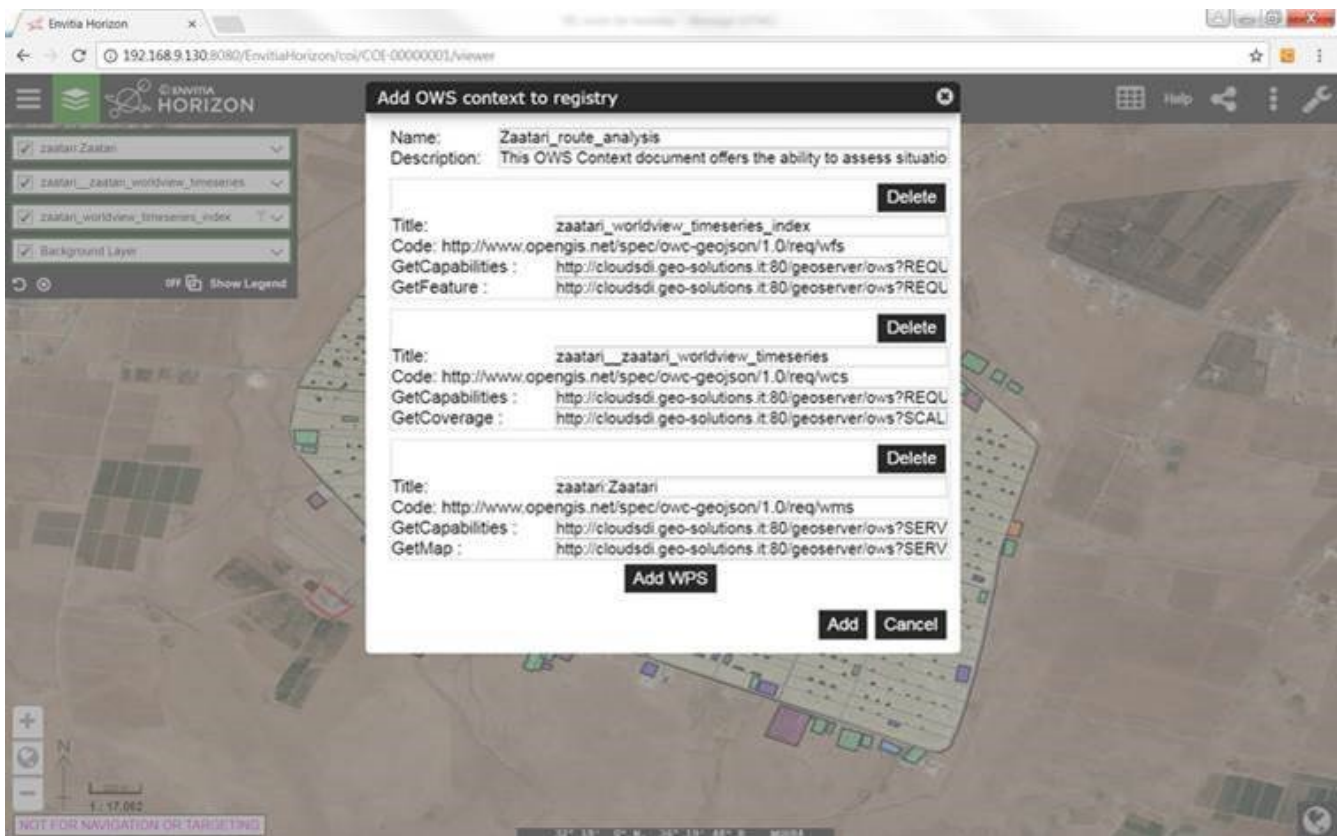


Figure 96. Creating an OWS Context document for Zaatari with WMS, WFS and WCS inputs.

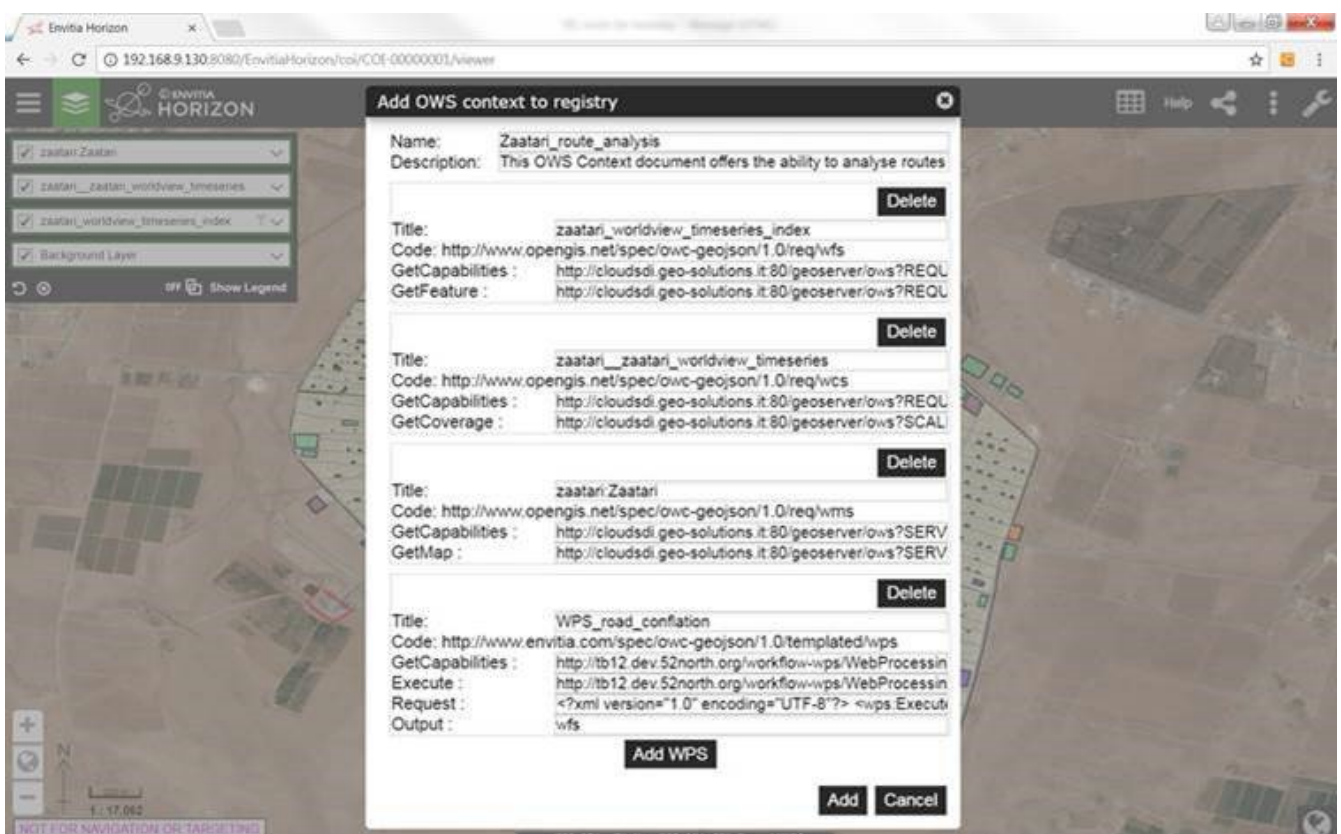


Figure 97. Creating an OWS Context document for Zaatari with WMS, WFS, WCS and WPS inputs.

Finally, WPS can be used to execute conflation.


```
<?xml version="1.0" encoding="UTF-8" ?>
<wps:Execute xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/2.0 http://schemas.opengis.net/wps/2.0/wps.xsd" service="WPS"
version="2.0.0" response="document" mode="sync">
  <ows:Identifier>testbed13.dsi.ConflationWorkflow</ows:Identifier>
  <wps:Input id="datasets">
    <wps:Reference xlink:href="
http://geoprocessing.demo.52north.org:8080/geoserver/tb13/ows?service=WFS&version=1.0.0&request=GetFeature&type=tb13:un-zataari-roads&
outputFormat=SHAPE-ZIP" mimeType="application/x-zipped-shp" />
  </wps:Input>
  <wps:Input id="datasets">
    <wps:Reference xlink:href="http://geoprocessing.demo.52north.org:8080/data/osm-zataari-roads.osm" mimeType="application/x-openstreetmap+xml" />
  </wps:Input>
  <wps:Output id="conflated-dataset" mimeType="text/xml" schema="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd" transmission="value" />
</wps:Execute>
```

Figure 98. WPS Conflation 'Execute' command.

Finally, WPS can be used to execute conflation. The graphic sequence below shows the output of the WPS Conflation first with OSM background, then with landsat8 background, then DigitalGlobe and finally the conflated data attributes:

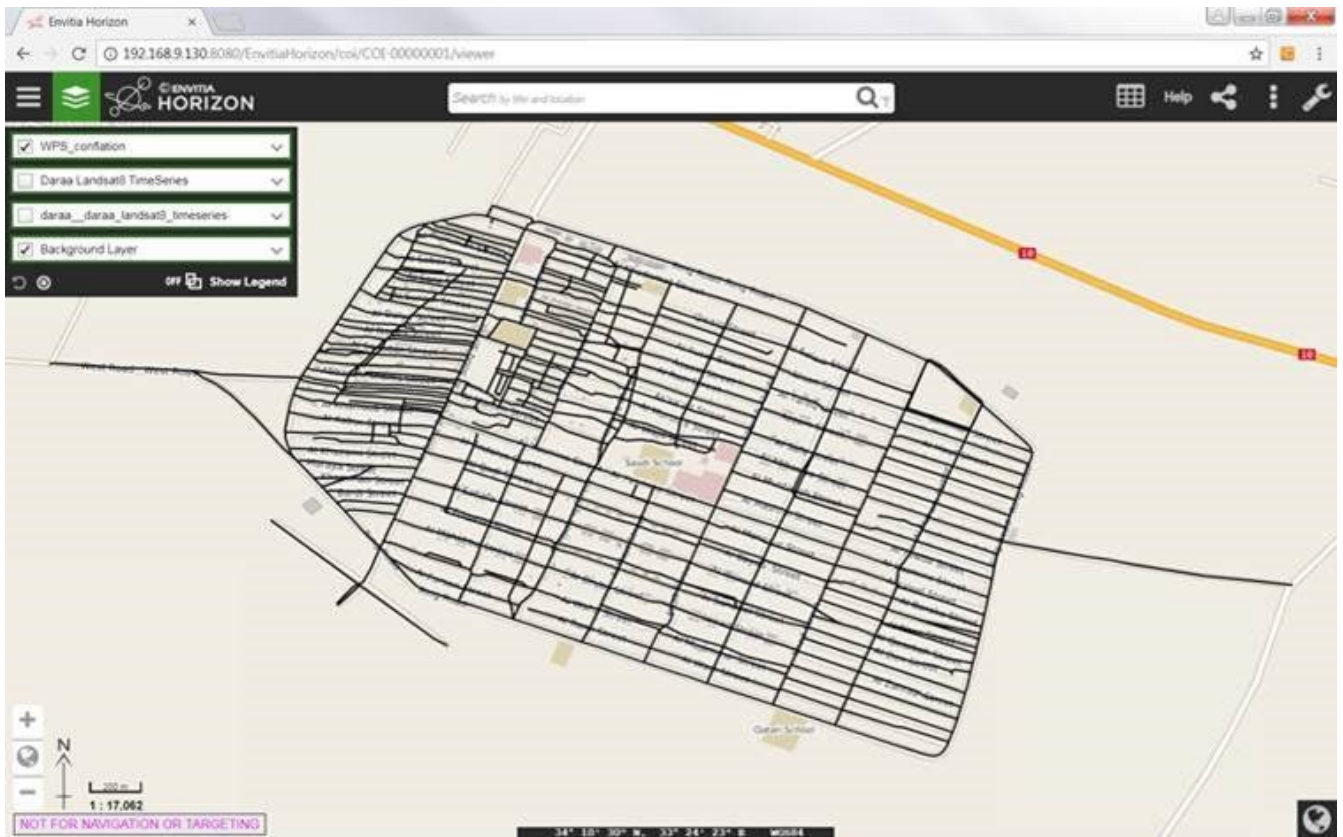


Figure 99. Output of WPS Conflation over Zaatari with OSM background.

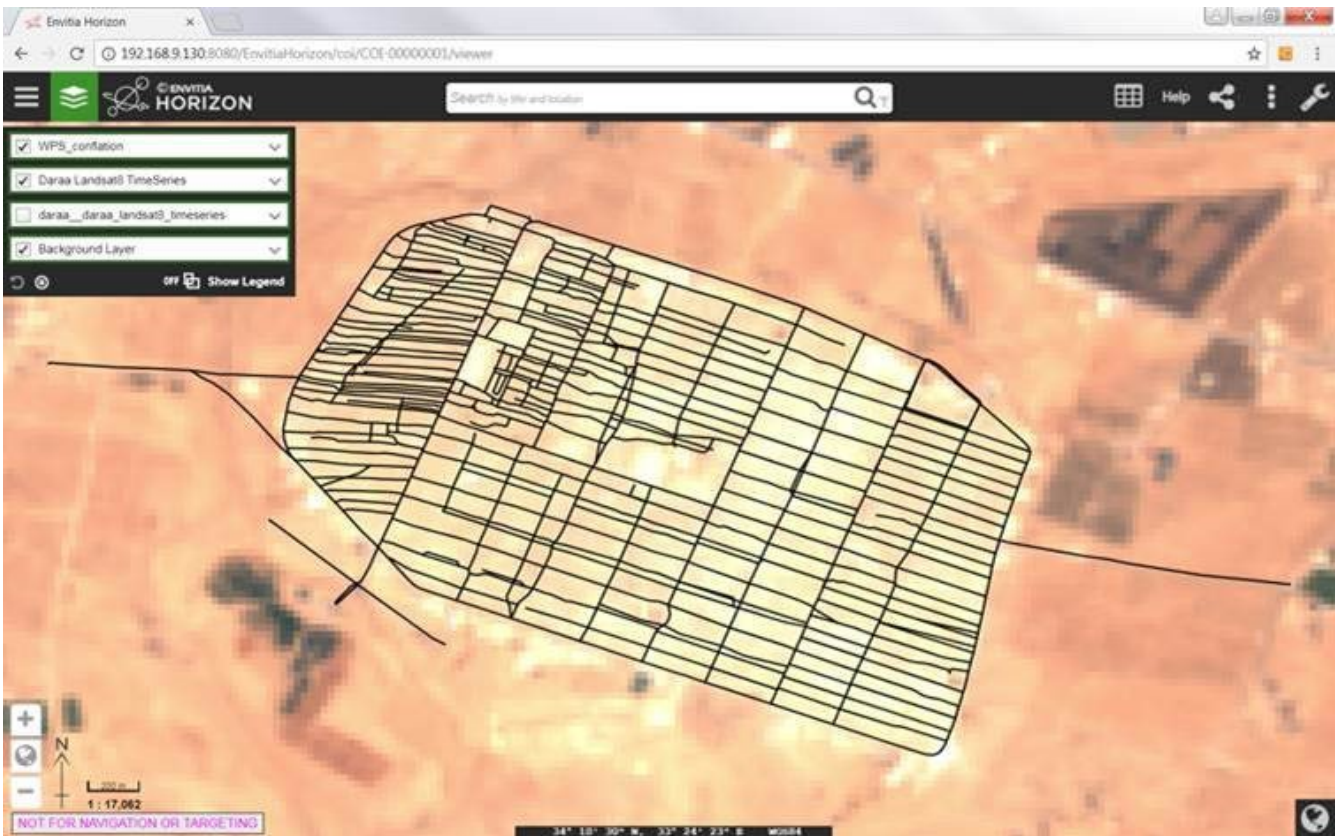


Figure 100. Output of WPS Conflation over Zaatari with Landsat 8 background.

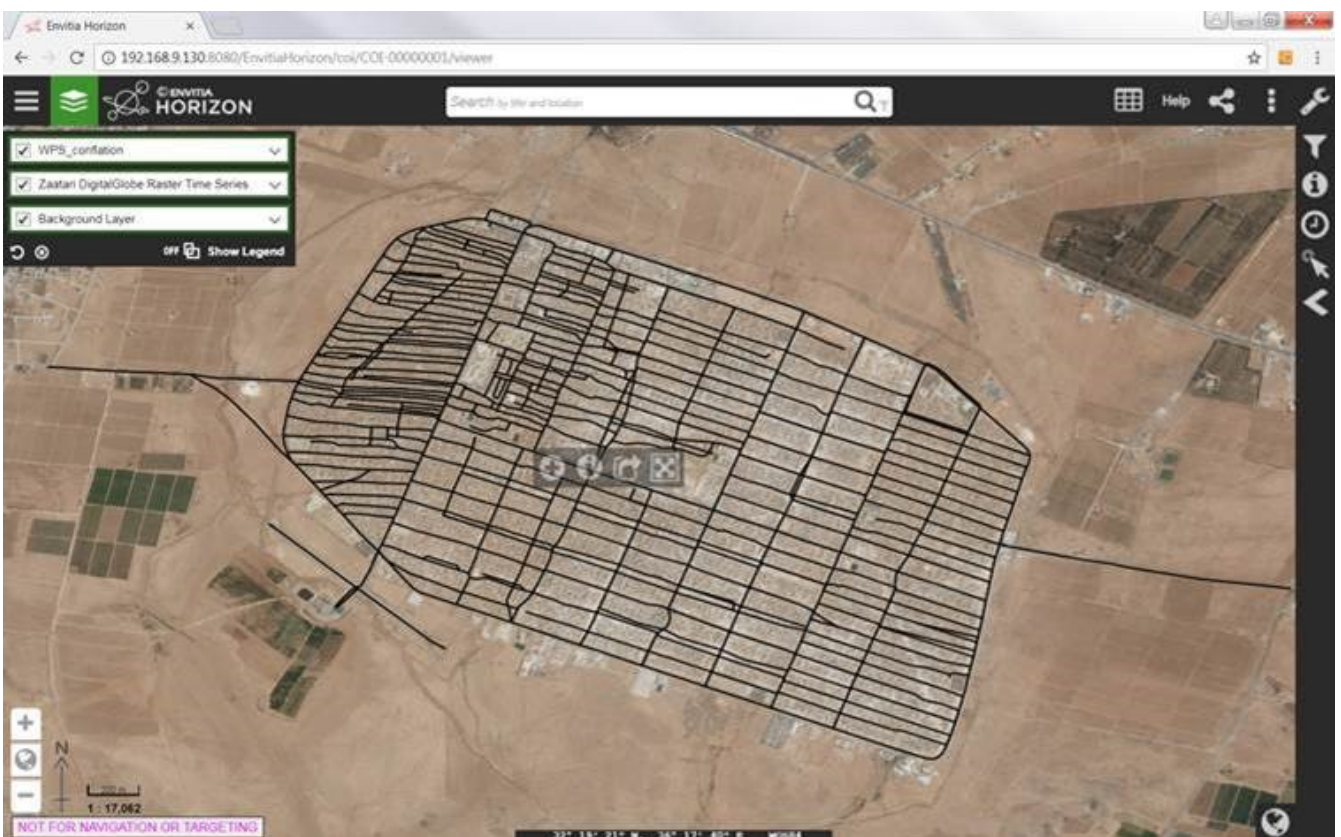


Figure 101. Output of WPS Conflation over Zaatari with DigitalGlobe background.

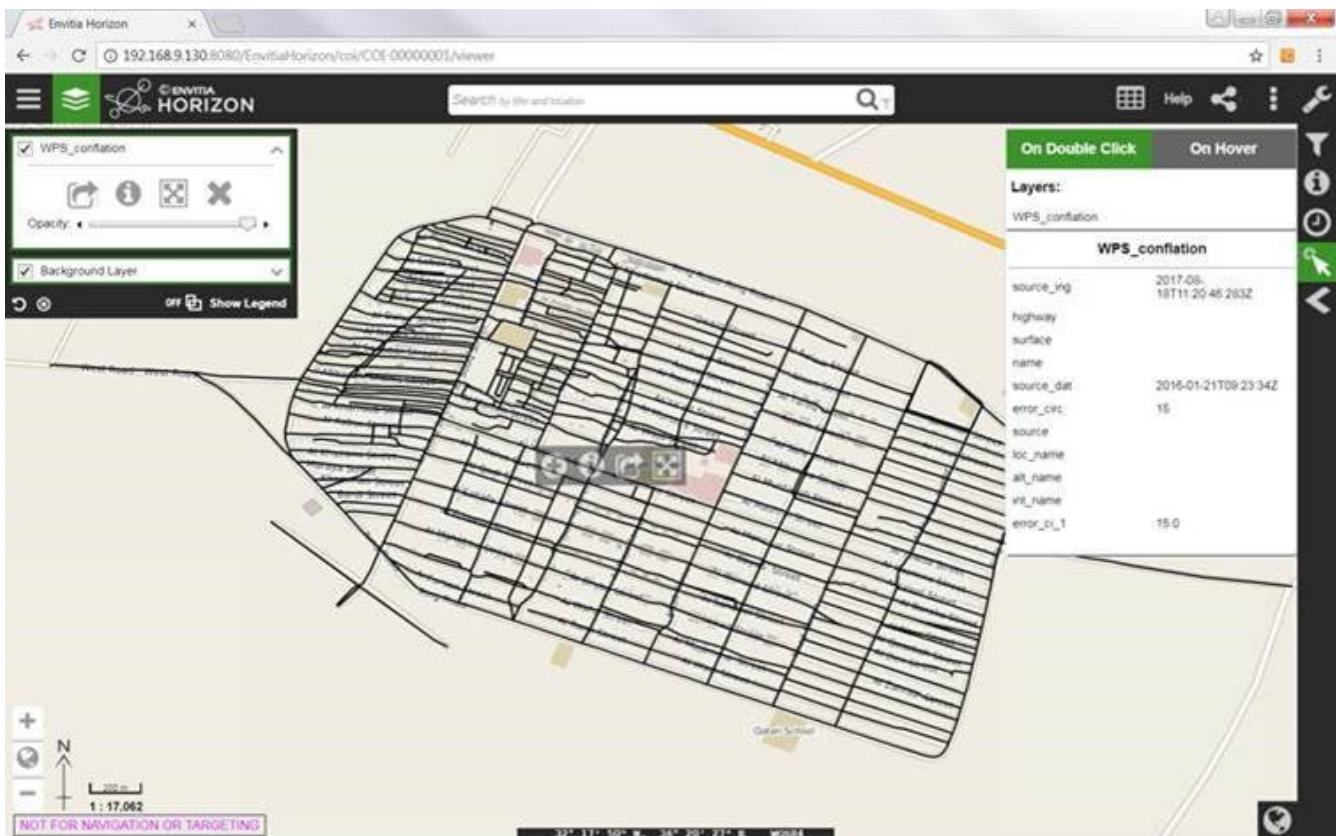


Figure 102. Output of WPS Conflation showing conflated data attributes.

Visualization Client

Visualization of data on the platform in the simulated Relief Operations Center was provided by University of Denver via integration with the World Wind environment (AB105 - Security enabled Desktop client). This client can provide visualization capabilities for relevant geospatial web service standards including WMS, WFS, and WCS (e.g., via getting information and capabilities, rendering visualizations, providing data results). Users are provided with standard zoom, pan, adjusting, and layer management tools. This client also implemented OAuth 2.0 for secure connection to imagery and other data sources.

In the demonstration scenario one of the first actions personnel in the Relief Operations Center take is to try and understand and communicate the impact the conflict has on social conditions and infrastructure in the region. To help with this, they accessed NASA Black Marble from two different years. NASA Black Marble is a global view of the Earth's city lights acquired by Suomi National Polar Orbiting (Suomi NPP) satellite. Visualizations from 2012 and 2016 are shown below as examples.

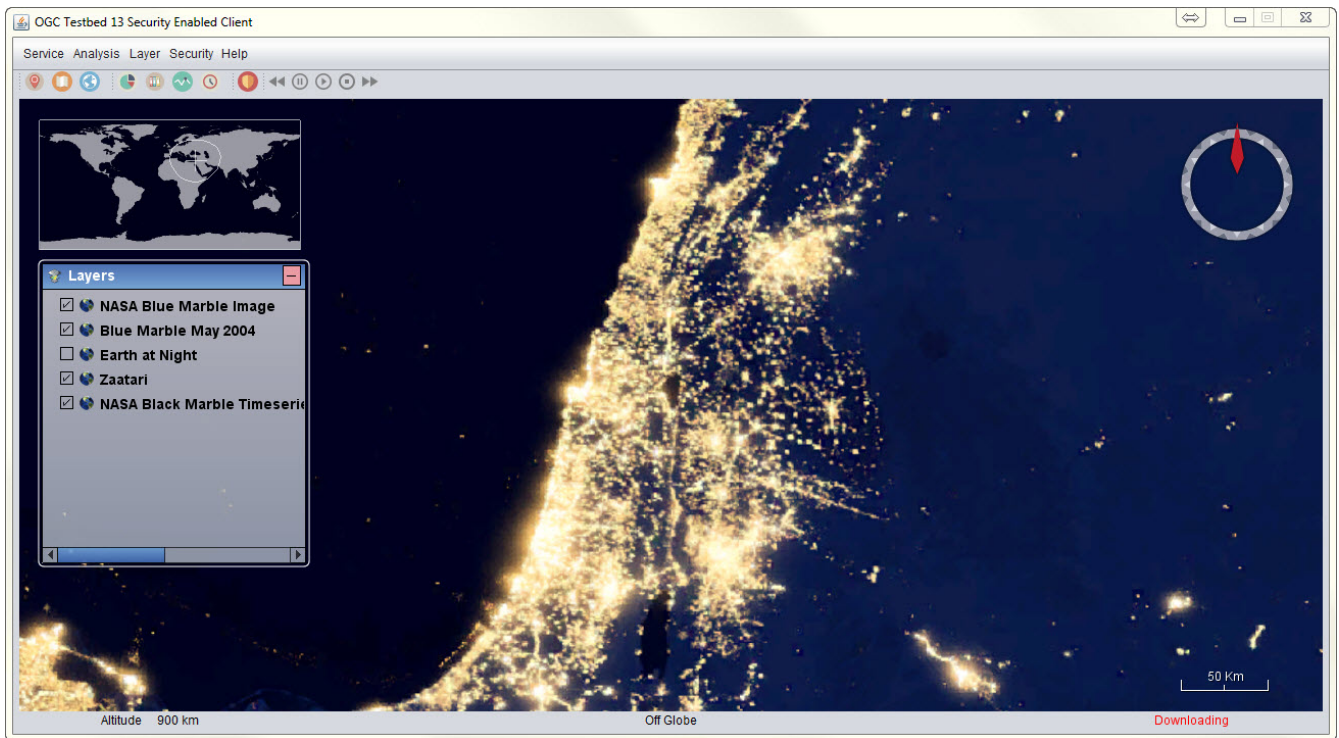


Figure 103. WMS visualization with satellite imagery from 2012, showing lights at night before the conflict.

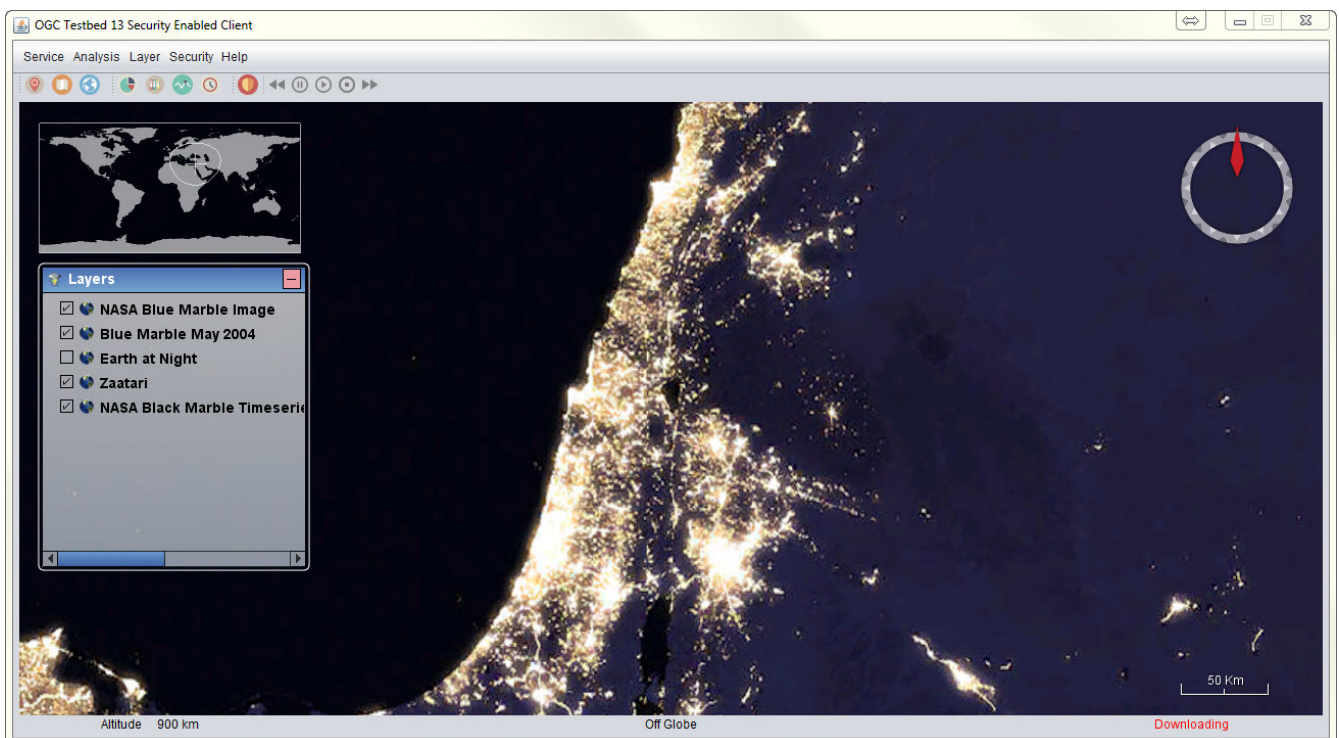


Figure 104. WMS visualization with satellite imagery from 2016, showing lights at night during the conflict.

As shown in the visualization, the change in lights at night is significant in the region. To make it even easier to distribute this visualization the Relief Operations Center also created a video of the changes and posted it on YouTube for anyone to access.



Figure 105. WMS visualization of lights at night on YouTube.

The visualization is available at the following link:

<https://www.youtube.com/watch?v=Sx4bjGVUrq8>

Examples for visualizing other layers from WFS and WCS in the Zaatari and Daraa region are shown in the following examples. The base imagery layer is from a secure WCS with data provided by DigitalGlobe (2017 imagery). Feature data is represented on the base imagery and includes transportation lines, structural points, settlement polygons and information points. Please note, additional information on data sources used is provided in the following section.

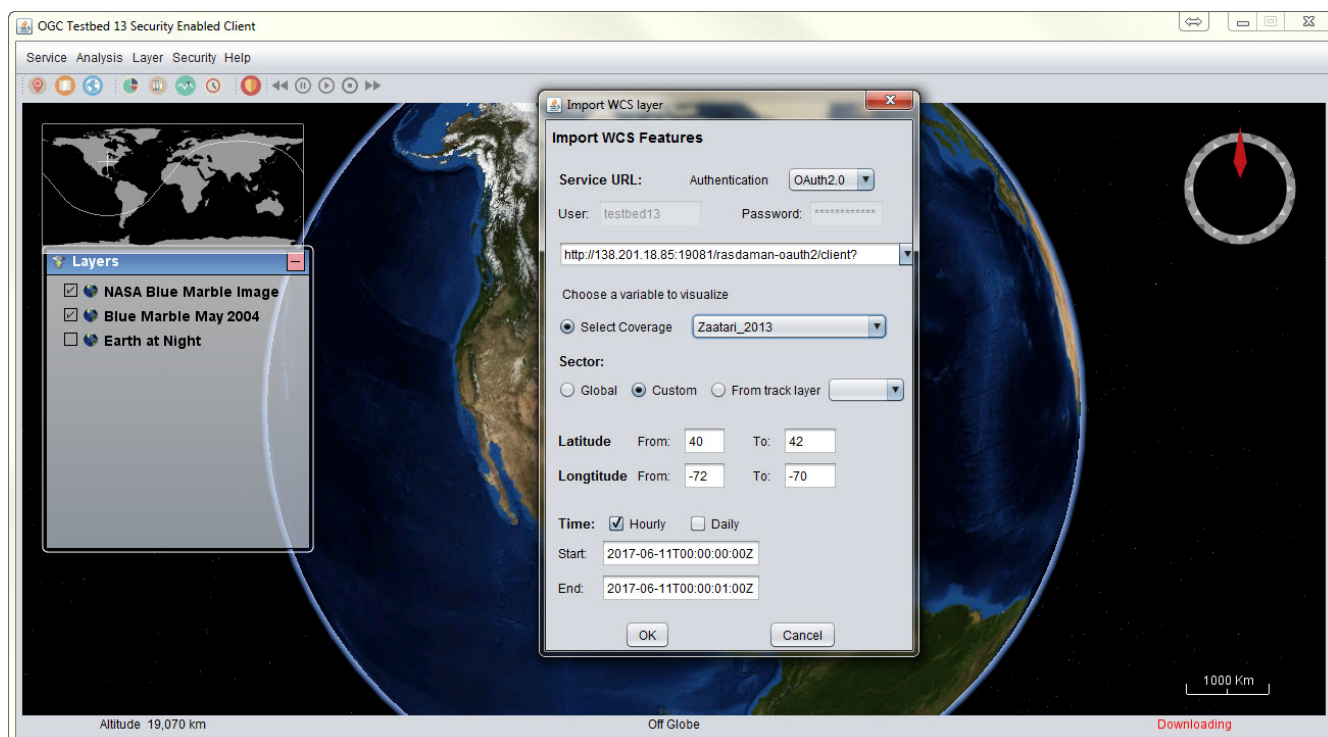


Figure 106. Visualization client accessing secure OGC services.

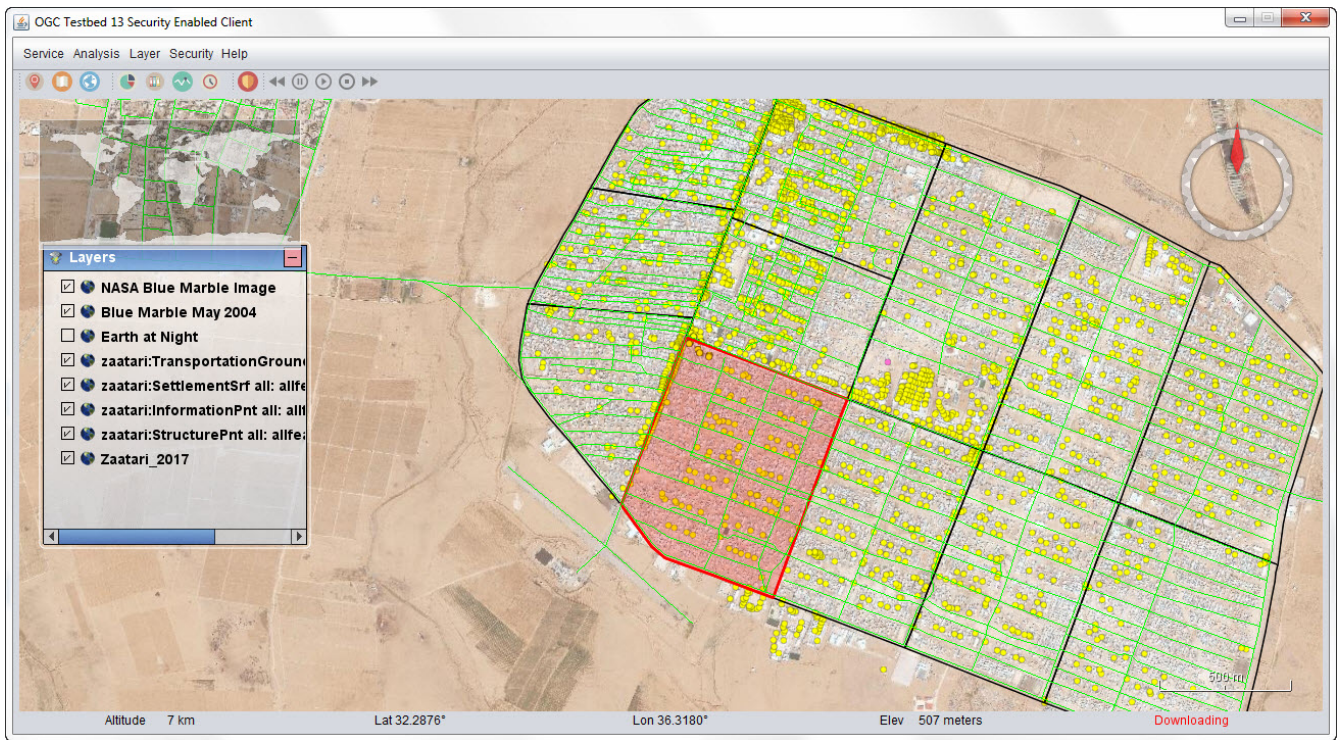


Figure 107. Security-enabled visualization client accessing data over Zaatari refugee camp.

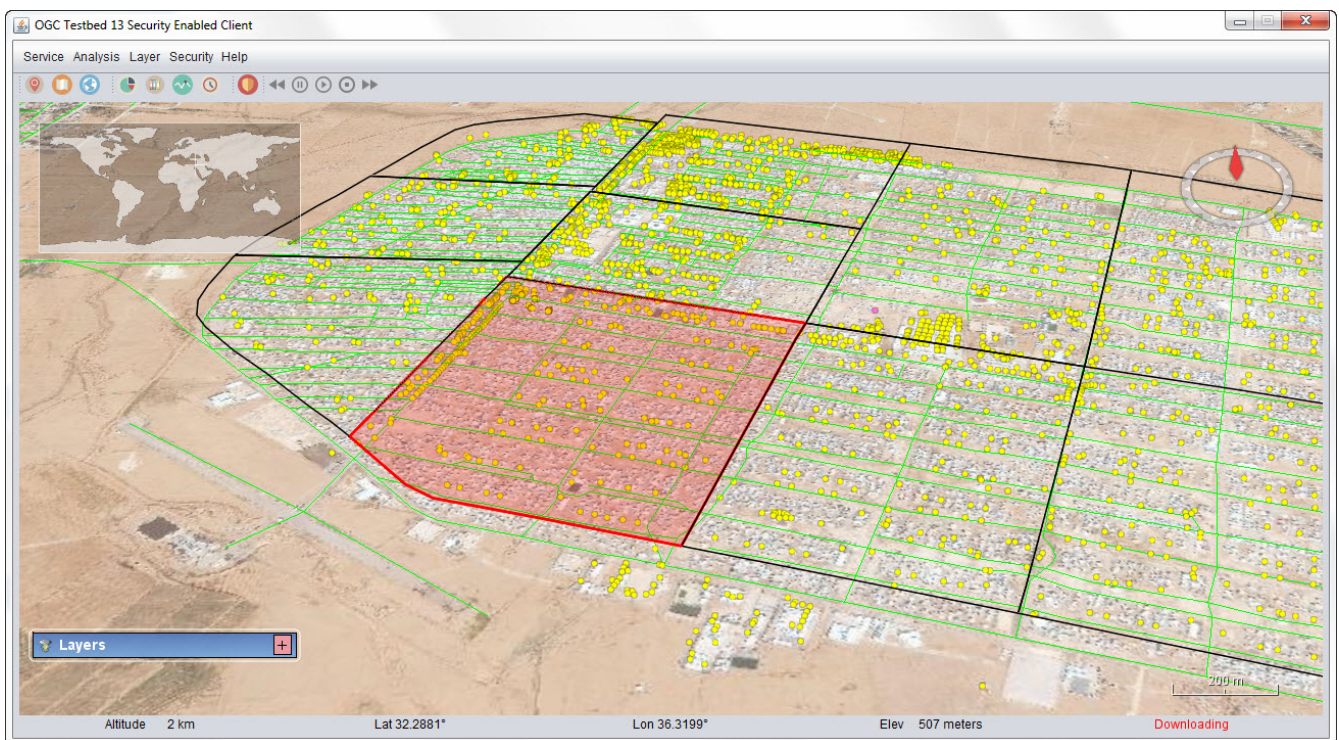


Figure 108. 3D visualization of data over Zaatari refugee camp.

Chapter 6. Findings

This section describes findings and any associated recommendations.

The evidence obtained through Testbed 13 FFP Technology Integration Experiments supports the following findings:

1. It is possible to cost-effectively develop a shared information exchange platform based on open standards for situational awareness, and coordinate humanitarian relief activities among nations and organizations. In addition, such a platform can securely enable interoperable exchange of messages.
2. Users who do not know about satellite imagery or complex geospatial data sources can access the platform using client applications that allow them to simply ask questions such as, "What are the best routes from Zaatari refugee camp to Daraa?" and get answers in an easy to understand form.
3. Geospatial data experts can configure information sources to quickly respond to a variety of user questions - with encodings such as OGC Context.
4. Relief Operations Centers can access high quality visualizations of data on the platform, to easily understand the changes in the region and mass migration over time. Access to data in these visualizations can be secure, and delivered only to authorized users.
5. Applications can access satellite imagery data on the platform using imagery metadata via WFS and CQL_Filter, select a subset based on the scenes characteristics and then trigger WMS for viewing, WCS for download and also WPS for processing. This flow differs from 'normal' OGC approaches in that the WFS is used to store metadata as features. This approach is actually common in large scale operational deployments.
6. Mobile application clients can be used in the field to provide access to data and services of the information exchange platform. In this scenario, users access data from the platform in the field on mobile devices as OGC GeoPackages (SQLite containers for geospatial feature and imagery data for use by mobile clients). These 'field users' often do not have a stable (or any) internet connection since network infrastructure is not reliable or available throughout much of the Daraa region.
7. The platform can support advanced analytics such as conflation of feature data sets.
8. It was noticed that in some cases it may be difficult for a generic client that starts a search on WFS to know which layer to use for WMS and WCS. Participants decided to use MetadataLinks from the GetCapabilities WFS response to link layer specific GetCapabilities of WMS and WCS. GeoServer has this capability to present a GetCapabilities call on each single layer hence it can provide this as a way to crosslink layers between different services
9. Participants enhanced GeoServer to allow the Envia client to echo a CQL_FILTER appended to a GetCapabilities call to the DCP in the response so that Desktop clients like QGIS will also be able to use basic filtering capabilities transparently.
10. For the use case above (desktop clients) participants are also considering the possibility to accept a CQL_FILTER that is a URL that points to remotely stored CQL filter - so that it is easier to reuse and share them with less limitations.

Appendix A: Context Encodings

Zaatari Context Document - Note the use of CQL_Filters

```
{
  "type": "FeatureCollection",
  "properties": {
    "lang": "en",
    "title": "Zaatari Refugee Camp route analysis",
    "keywords": [
      "fitForPurpose",
      "geoServer",
      "ows13",
      "Zaatari",
      "OSM",
      "demo"
    ],
    "taxonomy": "",
    "description": "Sample OWS context for Fit for Purpose Exercise for Zaatari",
    "timeFrame": "",
    "location": "Zaatari",
    "updated": "2017-08-01T08:00:00Z",
    "links": [
      {
        "rel": "profile",
        "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core",
        "title": "This file is compliant with version 1.0 of OWS Context"
      }
    ],
    "authors": [
      {
        "name": "GeoSolutions",
        "email": "info@geo-solutions.it",
        "uri": "http://www.geo-solutions.com"
      }
    ]
  },
  "id": "1496301789866",
  "bbox": [
    34.5,
    32,
    37.05,
    35.01
  ],
  "features": [
    {
      "type": "Feature",
      "id": "zaatari_worldview_timeseries_index",
      "geometry": {
```



```

    "type": "Polygon",
    "coordinates": [
      [
        [
          36.28,
          32.27
        ],
        [
          36.28,
          32.31
        ],
        [
          36.37,
          32.31
        ],
        [
          36.37,
          32.27
        ],
        [
          36.28,
          32.27
        ]
      ]
    ],
    "properties": {
      "title": "Zaatari WV01 Timeseries Catalogue",
      "updated": "2017-07-29T12:00:00Z",
      "content": "Catalogue of WorldView-1 imagery over Zaatari exposed as WFS and
WMS",
      "categories": [
        {
          "term": true,
          "scheme": "http://www.opengis.net/owc/active"
        },
        {
          "term": 1,
          "scheme": "http://www.envitia.com/horizon/layer/opacity"
        }
      ],
      "offerings": [
        {
          "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wfs",
          "operations": [
            {
              "code": "GetCapabilities",
              "method": "GET",
              "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/ows?REQUEST=GetCapabilities&SERVICE=WFS"
            }
          ]
        }
      ]
    }
  }
}

```

```

    },
    {
      "code": "GetFeature",
      "method": "GET",
      "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/ows?request=GetFeature&SERVICE=WFS&VERSION=1.1.0&TYPENAME=zaatari:zaatari_worldview_timeseries_index&CQL_FILTER=sun_elevation%3E54"
    }
  ]
},
{
  "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
  "operations": [
    {
      "code": "GetCapabilities",
      "method": "GET",
      "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/ows?REQUEST=GetCapabilities&SERVICE=WMS"
    },
    {
      "code": "GetMap",
      "method": "GET",
      "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries_index/ows?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fpng&LAYERS=zaatari%3Azaatari_worldview_timeseries_index&CQL_FILTER=sun_elevation%3E54&SRS=EPSG%3A404000&WIDTH=768&HEIGHT=363&BBOX=36.261749267578125%2C32.262468338012695%2C36.393585205078125%2C32.32478141784668"
    }
  ]
}
]
}
},
{
  "type": "Feature",
  "id": "zaatari_worldview_timeseries",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          36.28,
          32.27
        ],
        [
          36.28,
          32.31
        ],
        [

```

```

        36.37,
        32.31
    ],
    [
        36.37,
        32.27
    ],
    [
        36.28,
        32.27
    ]
]
],
},
"properties": {
  "title": "Zaatari WPS conflation",
  "updated": "2017-04-10T14:00:00Z",
  "content": "Conflation operation for the zaatari data set",
  "categories": [
    {
      "term": true,
      "scheme": "http://www.opengis.net/owc/active"
    },
    {
      "term": 1,
      "scheme": "http://www.envitia.com/horizon/layer/opacity"
    }
  ],
  "offerings": [
    {
      "code": "http://www.envitia.com/spec/owc-geojson/1.0/templated/wps",
      "operations": [
        {
          "code": "GetCapabilities",
          "method": "GET",
          "href": "http://tb12.dev.52north.org/workflow-wps/WebProcessingService?service=wps&version=2.0.0&request=GetCapabilities"
        },
        {
          "code": "Execute",
          "method": "POST",
          "href": "http://tb12.dev.52north.org/workflow-wps/WebProcessingService",
          "request": {
            "type": "application/xml",
            "content": "<?xml version='1.0' encoding='UTF-8'?> <wps:Execute
xmlns:wps='http://www.opengis.net/wps/2.0'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:ows='http://www.opengis.net/ows/2.0'
xmlns:xlink='http://www.w3.org/1999/xlink'
xsi:schemaLocation='http://www.opengis.net/wps/2.0

```

```

http://schemas.opengis.net/wps/2.0/wps.xsd" service="WPS" version="2.0.0"
response="document" mode="sync">
<ows:Identifier>testbed13.dsi.ConflationWorkflow</ows:Identifier> <wps:Input
id="datasets"> <wps:Reference
xlin:href="http://geoprocessing.demo.52north.org:8080/geoserver/tb13/ows?service=WFS&
amp;version=1.0.0&request=GetFeature&typeName=tb13:un-zataari-
roads&outputFormat=SHAPE-ZIP" mimeType="application/x-zipped-shp" \/>
</wps:Input> <wps:Input id="datasets"> <wps:Reference
xlin:href="http://geoprocessing.demo.52north.org:8080/data/osm-zataari-roads.osm"
mimeType="application/x-openstreetmap+xml" \/> </wps:Input> <wps:Output
id="conflated-dataset" mimeType="text/xml"
schema="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"
transmission="value" \/></wps:Execute>
    },
    "replacements":[
    ],
    "output":"wfs"
  }
]
}
]
}
},
{
  "type": "Feature",
  "id": "SettlementSrf",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          34.8844371486709,
          29.1834009533069
        ],
        [
          42.3745686975872,
          29.1834009533069
        ],
        [
          42.3745686975872,
          37.3184588992791
        ],
        [
          34.8844371486709,
          37.3184588992791
        ],
        [
          34.8844371486709,
          29.1834009533069
        ]
      ]
    ]
  }
}
]

```

```

]
},
"properties": {
  "title": "Zaatari Refugee Camp Area",
  "updated": "2017-07-29T12:00:00Z",
  "content": "Vector Layer",
  "categories": [
    {
      "term": true,
      "scheme": "http://www.opengis.net/owc/active"
    },
    {
      "term": 1,
      "scheme": "http://www.envitia.com/horizon/layer/opacity"
    }
  ],
  "offerings": [
    {
      "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
      "operations": [
        {
          "code": "GetCapabilities",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/SettlementSrf/ows?REQUEST=GetCapabilities&SERVICE=WMS"
        },
        {
          "code": "GetMap",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/wms?service=WMS&version=1.1.0&request=GetMap&layers=SettlementSrf&styles=&&WIDTH=768&HEIGHT=557&BBBOX=36.31161689758301%2C32.280471324920654%2C36.34457588195801%2C32.30437517166138&srs=EPSG:4326&format=image%2Fpng&TRANSPARENT=true"
        }
      ]
    },
    {
      "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wfs",
      "operations": [
        {
          "code": "GetCapabilities",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/SettlementSrf/ows?REQUEST=GetCapabilities&SERVICE=WFS"
        },
        {
          "code": "GetFeature",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/SettlementSrf/ows?request=GetFeature&SERVICE=WFS&VERSIO

```

```
N=1.1.0&TYPENAME=SettlementSrf"
```

```
    }
  ]
}
],
{
  "type": "Feature",
  "id": "zaatari_worldview_timeseries",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          36.28,
          32.27
        ],
        [
          36.28,
          32.31
        ],
        [
          36.37,
          32.31
        ],
        [
          36.37,
          32.27
        ],
        [
          36.28,
          32.27
        ]
      ]
    ]
  },
  "properties": {
    "title": "Zaatari WV01 Imagery",
    "updated": "2017-07-29T12:00:00Z",
    "content": "WorldView-1 imagery over Zaatari exposed as WCS and WMS",
    "categories": [
      {
        "term": true,
        "scheme": "http://www.opengis.net/owc/active"
      },
      {
        "term": 1,
        "scheme": "http://www.envitia.com/horizon/layer/opacity"
      }
    ]
  },
},
```

```

"offerings": [
  {
    "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
    "operations": [
      {
        "code": "GetCapabilities",
        "method": "GET",
        "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?REQUEST=GetCapabilities&SERVICE=WMS"
      },
      {
        "code": "GetMap",
        "method": "GET",
        "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=zaatari_worldview_timeseries&SRS=EPSG%3A4326&bbox=36.28552436828613,32.27472407452737,36.36792397545432,32.31376157457074&width=768&height=363&time=2010/2017&cql_filter=sun_elevation%3C54&sortby=time%20A"
      }
    ]
  },
  {
    "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wcs",
    "operations": [
      {
        "code": "GetCapabilities",
        "method": "GET",
        "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?REQUEST=GetCapabilities&SERVICE=WCS"
      },
      {
        "code": "GetCoverage",
        "method": "GET",
        "href": "http://cloudsdi.geosolutions.it/geoserver/zaatari/zaatari_worldview_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=zaatari_worldview_timeseries&subset=Long(36.28,36.37)&subset=Lat(32.27,32.31)&subset=time(%222010-01-01T00:00:00Z%22,%222018-01-01T00:00:00Z%22)&cql_filter=sun_elevation%3C54&scaleaxes=i(0.05),j(0.05)&sortby=time%20A"
      }
    ]
  }
]
}

```

```

{
  "type": "FeatureCollection",
  "properties": {
    "lang": "en",
    "title": "NASA Black Marble Global Timeseries for 2012-2016",
    "keywords": [
      "fitForPurpose",
      "geoServer",
      "ows13",
      "Daraa",
      "NASA",
      "Zataari",
      "World",
      "Black Marble"
    ],
    "taxonomy": "",
    "description": "Sample OWS context for 'Fit for Purpose'",
    "timeFrame": "",
    "location": "Middle East",
    "updated": "2017-08-01T08:00:00Z",
    "links": [
      {
        "rel": "profile",
        "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core",
        "title": "This file is compliant with version 1.0 of OWS Context"
      }
    ],
    "authors": [
      {
        "name": "GeoSolutions",
        "email": "info@geo-solutions.it",
        "uri": "http://www.geo-solutions.com"
      }
    ]
  },
  "id": 1496301789899,
  "bbox": [
    -180, -90,
    180, 90
  ],
  "features": [
    {
      "type": "Feature",
      "id": "black_marble_timeseries",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [

```



```

        [-180, -90],
        [-180, 90],
        [180, 90],
        [180, -90],
        [-180, -90]
    ]
]
},
"properties": {
    "title": "NASA Black Marble Timeseries for 2012 and 2016",
    "updated": "2017-07-T12:00:00Z",
    "content": "NASA Black Marble Timeseries for 2012 and 2016, it is exposed as
WCS and WMS",
    "categories": [
        {
            "term": true,
            "scheme": "http://www.opengis.net/owc/active"
        },
        {
            "term": 1,
            "scheme": "http://www.envitia.com/horizon/layer/opacity"
        }
    ],
    "offerings": [
        {
            "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
            "operations": [
                {
                    "code": "GetCapabilities",
                    "method": "GET",
                    "href": "http://cloudsdi.geosolutions.it/geoserver/geosolutions/black_marble_timeseries/ows?REQUEST=GetCapabilities&SERVICE=WMS"
                },
                {
                    "code": "GetMap",
                    "method": "GET",
                    "href": "http://cloudsdi.geosolutions.it/geoserver/geosolutions/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=black_marble_timeseries&SRS=EPSG%3A4326&WIDTH=1900&HEIGHT=1000&BBOX=32.2998046875%2C24.466552734375%2C53.173828125%2C35.452880859375&TIME=2016"
                }
            ]
        },
        {
            "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wcs",
            "operations": [
                {
                    "code": "GetCapabilities",
                    "method": "GET",

```

```
      "href": "http://cloudsdi.geo-
solutions.it/geoserver/geosolutions/black_marble_timeseries/ows?REQUEST=GetCapabilities&SERVICE=WCS"
    },
    {
      "code": "GetCoverage",
      "method": "GET",
      "href": "http://cloudsdi.geo-
solutions.it/geoserver/geosolutions/black_marble_timeseries/ows?service=WCS&version=2.0.0&request=GetCoverage&coverageId=geosolutions__black_marble_timeseries&subset=Long(33,44)&subset=Lat(31,38)&subset=time(%222016-01-01T00:00:00Z%22,%222017-01-01T00:00:00Z%22)"
    }
  ]
}
]
}
```

Daraa Context Document

```
{
  "type": "FeatureCollection",
  "properties": {
    "lang": "en",
    "title": "Daraa",
    "keywords": [
      "fitForPurpose",
      "geoServer",
      "ows13",
      "Daraa",
      "OSM",
      "demo"
    ],
    "taxonomy": "",
    "description": "This OWS context document provides the ability to conflate existing road datasets (UN and OSM) providing exhaustive road information for route analysis.",
    "timeFrame": "",
    "location": "Daraa",
    "updated": "2017-08-01T08:00:00Z",
    "links": [
      {
        "rel": "profile",
        "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core",
        "title": "This file is compliant with version 1.0 of OWS Context"
      }
    ],
  }
}
```

```

    "authors": [
      {
        "name": "GeoSolutions",
        "email": "info@geo-solutions.it",
        "uri": "http://www.geo-solutions.com"
      }
    ],
    "id": 1496301789865,
    "bbox": [
      34.5, 32,
      37.05, 35.01
    ],
    "features": [
      {
        "type": "Feature",
        "id": "daraa_landsat8_timeseries_index",
        "geometry": {
          "type": "Polygon",
          "coordinates": [
            [
              [34.5, 32],
              [34.5, 35.01],
              [37.05, 35.01],
              [37.05, 32],
              [34.5, 32]
            ]
          ]
        },
        "properties": {
          "title": "Daraa Landsat 8 Catalogue of Imagery",
          "updated": "2017-07-29T12:00:00Z",
          "content": "Catalogue of Landsat 8 imagery over Daraa, it is exposed as WFS
and WMS",
          "categories": [
            {
              "term": true,
              "scheme": "http://www.opengis.net/owc/active"
            },
            {
              "term": 1,
              "scheme": "http://www.envitia.com/horizon/layer/opacity"
            }
          ],
          "offerings": [
            {
              "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wfs",
              "operations": [
                {
                  "code": "GetCapabilities",
                  "method": "GET",

```

```

        "href": "http://cloudsdi.geo-
solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/ows?REQUEST=GetCapabiliti
es&SERVICE=WFS"
    },
    {
        "code": "GetFeature",
        "method": "GET",
        "href": "http://cloudsdi.geo-
solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/ows?request=GetFeature&SE
RVICE=WFS&VERSION=1.1.0&TYPENAME=daraa:daraa_landsat8_timeseries_index&CQL_FILTER=sun_
elevation%3C51"
    }
]
},
{
    "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
    "operations": [
        {
            "code": "GetCapabilities",
            "method": "GET",
            "href": "http://cloudsdi.geo-
solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/ows?REQUEST=GetCapabiliti
es&SERVICE=WMS"
        },
        {
            "code": "GetMap",
            "method": "GET",
            "href": "http://cloudsdi.geo-
solutions.it/geoserver/daraa/daraa_landsat8_timeseries_index/ows?SERVICE=WMS&VERSION=1
.1.1&REQUEST=GetMap&FORMAT=image%2Fpng&TRANSPARENT=true&STYLES&LAYERS=daraa%3Adaraa_l
andsat8_timeseries_index&SRS=EPSG%3A4326&WIDTH=950&HEIGHT=700&BBOX=34.48883056640625%2
C32.354736328125%2C37.09808349609375%2C34.27734375&CQL_FILTER=sun_elevation%3C51"
        }
    ]
}
]
}
},
{
    "type": "Feature",
    "id": "SettlementSrf",
    "geometry": {
        "type": "Polygon",
        "coordinates": [
            [
                [35.9759318108738,32.5086069097217],
                [36.1709080890199,32.5086069097217],
                [36.1709080890199,32.6754186248039],
                [35.9759318108738,32.6754186248039],
                [35.9759318108738,32.5086069097217]
            ]
        ]
    }
}
]
}
}

```

```

]
},
"properties": {
  "title": "Daraa Refugee Camp Area",
  "updated": "2017-06-29T12:00:00Z",
  "content": "Vector Layer for settlements from OSM over Daraa",
  "categories": [
    {
      "term": true,
      "scheme": "http://www.opengis.net/owc/active"
    },
    {
      "term": 1,
      "scheme": "http://www.envitia.com/horizon/layer/opacity"
    }
  ],
  "offerings": [
    {
      "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
      "operations": [
        {
          "code": "GetCapabilities",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/daraa/SettlementSrf/ows?REQUEST=GetCapabilities&SERVICE=WMS"
        },
        {
          "code": "GetMap",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/daraa/wms?service=WMS&version=1.1.0&request=GetMap&layers=SettlementSrf&styles=&bbox=35.97593181087375,32.50860690972173,36.17090808901993,32.675418624803896&width=768&height=657&srs=EPSG:4326&format=image%2Fpng&transparent=true"
        }
      ]
    },
    {
      "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wfs",
      "operations": [
        {
          "code": "GetCapabilities",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/daraa/SettlementSrf/ows?REQUEST=GetCapabilities&SERVICE=WFS"
        },
        {
          "code": "GetFeature",
          "method": "GET",
          "href": "http://cloudsdi.geosolutions.it/geoserver/daraa/SettlementSrf/ows?request=GetFeature&SERVICE=WFS&VERSION=1.1.0&TYPENAME=daraa:SettlementSrf"
        }
      ]
    }
  ]
}

```

```

    }
  ]
}
},
{
  "type": "Feature",
  "id": "daraa_landsat8_timeseries",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [34.5, 32],
        [34.5, 35.01],
        [37.05, 35.01],
        [37.05, 32],
        [34.5, 32]
      ]
    ]
  },
  "properties": {
    "title": "Daraa Landsat 8 Timeseries Imagery",
    "updated": "2017-07-29T12:00:00Z",
    "content": "Landsat 8 Timeseries over Daraa, it is exposed as WCS and WMS",
    "categories": [
      {
        "term": true,
        "scheme": "http://www.opengis.net/owc/active"
      },
      {
        "term": 1,
        "scheme": "http://www.envitia.com/horizon/layer/opacity"
      }
    ],
    "offerings": [
      {
        "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
        "operations": [
          {
            "code": "GetCapabilities",
            "method": "GET",
            "href": "http://cloudsdi.geosolutions.it/geoserver/daraa/daraa_landsat8_timeseries/ows?REQUEST=GetCapabilities&SERVICE=WMS"
          },
          {
            "code": "GetMap",
            "method": "GET",
            "href": "http://cloudsdi.geo-

```

```

solutions.it/geoserver/daraa/daraa_landsat8_timeseries/wms?SERVICE=WMS&VERSION=1.1.1&R
EQUEST=GetMap&FORMAT=image%2Fjpeg&TRANSPARENT=true&STYLES&LAYERS=daraa_landsat8_timese
ries&SRS=EPSG%3A4326&WIDTH=768&HEIGHT=648&BBOX=36.06820106506348%2C32.591285705566406%
2C36.13411903381348%2C32.64690399169922&time=2017&cql_filter=sun_elevation%3E51"
    }
  ]
},
{
  "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wcs",
  "operations": [
    {
      "code": "GetCapabilities",
      "method": "GET",
      "href": "http://cloudsdi.geo-
solutions.it/geoserver/daraa/daraa_landsat8_timeseries/ows?REQUEST=GetCapabilities&SER
VICE=WCS"
    },
    {
      "code": "GetCoverage",
      "method": "GET",
      "href": "http://cloudsdi.geo-
solutions.it/geoserver/daraa/daraa_landsat8_timeseries/ows?service=WCS&version=2.0.0&r
equest=GetCoverage&coverageId=daraa_landsat8_timeseries&subset=Long(36.06,36.14)&subse
t=Lat(32.59,32.65)&subset=time(%222017-01-01T00:00:00Z%22,%222018-01-
01T00:00:00Z%22)&cql_filter=sun_elevation%3E51"
    }
  ]
}
]
}
]
}
}
}
}

```

Appendix B: Revision History

Table 2. Revision History

Date	Release	Editor	Primary clauses modified	Descriptions
May 15, 2017	Initial Engineering Reports (IER)	J. Harrison	all	initial version
July 31, 2017	Updated Content	J. Harrison	all	architecture and content integrated
September 30, 2017	Preliminary Draft Engineering Reports (DERs)	J. Harrison	all	added TIE content and preparation for publication

Appendix C: Bibliography

[1] Harrison, J.: OGC Testbed-13: Concepts of Data and Standards for Mass Migration Engineering Report, OGC 17-078, Open Geospatial Consortium (2018)

[2] Rieke, M., Pross, B.: OGC® OWS-9 Cross Community Interoperability (CCI) Conflation with Provenance Engineering Report, OGC 12-159, Open Geospatial Consortium (2012)

[3] Pross, B.: OGC Testbed-12 WPS Conflation Service Profile Engineering Report, OGC 16-022, Open Geospatial Consortium (2017)