Open Geospatial Consortium (OGC)

Annex B

OGC Testbed 12 Architecture

OGC Testbed-12 Advance Use of Catalog Service for Web (CSW) Advance use of Data Broker Advance use of Semantics for Aviation Advanced use of Semantic Business Vocabulary and Business Rules (SBVR) ArcticSDI Asynchronous Messaging for Geospatial Queries of Aviation Data Bandwidth constraints, 3D data delivery, compression (binaryXML, GeoJSON, GML) Compliance Interoperability Test Environment: TEAM Engine Conflation Tool Integration DGIWG Data Quality: Geospatial Data Image quality: multiple satellites integration Temporal accuracy metric: provide overview of mosaic based on age of all pixels Accuracy of pixels: spherical (3D) accuracy Flight Information Exchange Model (FIXM) GeoJSON, TopoJSON, GML GeoPackage Contents GeoPackage Evaluation GeoPackage Mobile App Tools GeoPackage Tiling Enhancements GeoPackage for an alternative delivery format for US topo vector data Geopackage routing, terrain, and symbology Geospatial analysis at large: alert detection, discrete grids, geospatial streaming JavaScript, JSON, JSON-LD Large tile store data base format definition NSG App. Schema with ShapeChange OGC General Feature Model > Object Model OGC Web Service Security OWS Context Ontologies for Portrayal and Semantic Mediation Optimize capabilities documents for large data offerings and explore alternatives Persistent Datasets, Symbols, Styles REST SOAP Support SWE for LiDAR and Streaming Tiling Raster & Vector User Guides Visualization of EO data WCS: merging with NetCDF/OPeNDAP WFS Synchronization WFS-T and WCS to support asynchronous request/response handling Web Integration Service necessity evaluation

OGC
TESTBED-12
innovative – creative – effective

version: November 6, 2015

Corrigenda

The following table identifies all corrections that have been applied to *Annex B* compared to the original release (version *October 21, 2015*).

Section	Description
8.11	Entire section has been revised. Ambiguous and partly missing requirements have been resolved.
8.12	Figure 8.10 has been updated to reflect changes in section 8.11
8.6	Figure 8.4 has been updated to reflect changes in section 8.11
8.6	Table 8.1 has been updated to reflect changes in section 8.11
8.20	Third paragraph, reference to *(A016)* removed.
8.23.1	Description of work items A050 and A051 updated to reflect changes in section 8.11
7.4.2	Description of work items A103 and A104 updated to reflect changes in section 8.11
All	The Consolidation thread abbreviation has changed from CON to CNS
All	The Compliance thread abbreviation has hanged from COM to CMP
8.23.2	References corrected for A066
8.17.1	First paragraph, last sentence removed.

TABLE 1: Overview of all corrections

Contents

Corrigenda i

1 Introduction 1
1.1 Aviation . . . 1
1.2 Geospatial Imagery Quality Framework . . . 2
1.3 Coverage Access and Visualization . . . 2
1.4 GFM, Catalogs, and Semantics . . . 3
1.5 OGC Baseline Enhancements . . . 3
1.6 Compression and Generalization . . . 3
1.7 ArcticSDI and GeoPackage . . . 3

2 Testbed Baseline 5
2.1 Types of Deliverables . . . 5
2.2 OGC Reference Model . . . 6
2.3 OGC Standards Baseline . . . 8
2.4 Data . . . 8
2.5 Services in the Cloud . . . 9

3 Testbed Threads 10
3.1 Background . . . 10
3.2 Field Operations (FO) Thread . . . 12
3.3 Large Scale Analytics (LSA) Thread . . . 12
3.4 Linked Data and Advanced Semantics for Data Discovery and Dynamic Integration (LDS) Thread . . . 13
3.5 Command Center (CMD) Thread . . . 13
3.6 Consolidation (CNS) Thread . . . 14
3.7 Aviation (AVI) Thread . . . 14
3.8 Compliance Testing (CMP) Thread . . . 15

4 Aviation 16
4.1 Background . . . 16
4.2 Requirements and Work Items . . . 18
4.3 Aviation Architecture . . . 18
4.4 Advance use of Data Broker . . . 20
4.5 Advance Use of Catalog Service for Web (CSW) . . . 22
4.6 Asynchronous Messaging for Geospatial Queries of Aviation Data 24
4.7 Advance use of Semantics for Aviation . . . 25

4.8 Advanced use of Semantic Business Vocabulary and Business Rules (SBVR) . . . 26
4.9 Aviation OGC Web Service Security . . . 27
4.10 Flight Information Exchange Model (FIXM) . . . 28
4.11 Summary . . . 29

5 Geospatial Imagery Quality Framework **31**
5.1 Background . . . 31
5.2 Requirements and Work Items . . . 32
5.3 Accuracy, Currency, Completeness, Consistency . . . 32
5.4 Summary . . . 35

6 Coverage Access and Visualization **37**
6.1 Background . . . 37
6.2 Requirements and Work Items . . . 41
6.3 Visualization of Earth Observation (EO) Data . . . 41
6.4 WCS: Merging with NetCDF/OPeNDAP . . . 43
6.5 Summary . . . 47

7 GFM, Catalogs, and Semantics **49**
7.1 Background . . . 49
7.2 Requirements and Work Items . . . 49
7.3 General Feature Model . . . 50
7.4 Prototype Implementations . . . 53
7.5 Clients . . . 58
7.6 Summary . . . 59

8 OGC Baseline Enhancements **60**
8.1 Background . . . 60
8.2 Requirements and Work Items . . . 61
8.3 Security and SOAP . . . 61
8.4 REST, JSON, and GeoJSON . . . 65
8.5 Semantic Enablement . . . 68
8.6 Asynchronous Service Interaction . . . 70
8.7 Tiling . . . 73
8.8 Compression . . . 74
8.9 SWE for LiDAR and Streaming . . . 76
8.10 WFS Synchronization . . . 77
8.11 Catalog . . . 78
8.12 Capabilities Document and Alternatives . . . 80
8.13 Big Data and Tile Stores . . . 81
8.14 Web Integration Service . . . 82
8.15 Conflation . . . 83
8.16 Data Quality . . . 85
8.17 GeoPackage . . . 86
8.18 OGC Web Services (OWS) Context . . . 91
8.19 UML Shape Change . . . 93

8.20 Data . . . 94
8.21 User Guides . . . 95
8.22 Compliance Testing . . . 96
8.23 Summary . . . 100

9 Compression and Generalization **112**
9.1 Background . . . 112
9.2 Requirements and Work Items . . . 113
9.3 Bandwidth Constraints . . . 113
9.4 Summary . . . 114

10 ArcticSDI and GeoPackage **116**
10.1 Background . . . 116
10.2 Requirements and Work Items . . . 118
10.3 ArcticSDI . . . 118
10.4 GeoPackage . . . 120
10.5 Summary . . . 122

List of Figures

2.1 RM-ODP 7

3.1 Assignment of Work Items to Threads 11

4.1 Requirements Aviation 19

5.1 Requirements Geospatial Data Quality 32

6.1 Requirements Aviation 38
6.2 Requirements Coverage Access and Visualization 41
6.3 WCS DAP Comparison 44

7.1 Requirements Data Access and Visualization 51
7.2 Requirements DCAT detail, part 1 55
7.3 Requirements DCAT detail, part 2 56

8.1 Requirements OGC Baseline Enhancements: Security and SOAP . 62
8.2 Requirements OGC Baseline Enhancements: REST and JSON . . . 66
8.3 Requirements OGC Baseline Enhancements: Semantic Enablement 69
8.4 Requirements OGC Baseline Enhancements: Asynchronous Service Interaction 71
8.5 Requirements OGC Baseline Enhancements: Tiling 74
8.6 Requirements OGC Baseline Enhancements: Compression 75
8.7 Requirements OGC Baseline Enhancements: SWE for LiDAR and Streaming 76
8.8 Requirements OGC Baseline Enhancements: WFS Synchronization 77
8.9 Requirements OGC Baseline Enhancements: Catalog 79
8.10 Requirements OGC Baseline Enhancements: Capabilities Document and Alternatives 80
8.11 Requirements OGC Baseline Enhancements: Big Data and Tile Stores 81
8.12 Requirements OGC Baseline Enhancements: Web Integration Service 82
8.13 Requirements OGC Baseline Enhancements: Conflation 84
8.14 Requirements OGC Baseline Enhancements: Data Quality 85
8.15 Requirements OGC Baseline Enhancements: GeoPackage 87
8.16 Requirements OGC Baseline Enhancements: OWS Context 92
8.17 Requirements OGC Baseline Enhancements: UML Shape Change 93
8.18 Requirements OGC Baseline Enhancements: Data and Portal . . . 94

8.19 Requirements OGC Baseline Enhancements: User Guides 96
8.20 Requirements OGC Baseline Enhancements: Compliance Testing 97

9.1 Requirements Compression and Generalization 113

10.1 Requirements ArcticSDI and GeoPackage 118

List of Tables

1 Overview of all corrections . . . i

3.1 Overview of the Testbed-12 work items . . . 10
3.2 Overview of the Field Operations Thread . . . 12
3.3 Overview of the Large Scale Analytics Thread . . . 12
3.4 Overview of the Linked Data and Advanced Semantics . . . 13
3.5 Overview of the Command Center (CMD) Thread . . . 14
3.6 Overview of the Consolidation (CNS) Thread . . . 14
3.7 Overview of the Aviation (AVI) Thread . . . 15
3.8 Overview of the Compliance Testing (CMP) Thread . . . 15

4.1 Aviation Architecture deliverables . . . 18
4.2 Advance use of Data Broker deliverables . . . 22
4.3 Advance Use of Catalog Service for Web deliverables . . . 23
4.4 Asynchronous Messaging for Geospatial Queries of Aviation Data deliverables . . . 25
4.5 Advance use of Semantics for Aviation deliverables . . . 26
4.6 Advanced use of Semantic Business Vocabulary and Business Rules deliverables . . . 27
4.7 Aviation OGC Web Service Security deliverables . . . 28
4.8 Flight Information Exchange Model deliverables . . . 29
4.9 Aviation work package deliverables summary . . . 30

5.1 Geospatial Imagery Quality Framework work package deliverables summary . . . 36

6.1 Data Access and Visualization work package deliverables summary . . . 48

7.1 Data Access and Visualization work package deliverables summary . . . 59

8.1 Asynchronous services and ER overview . . . 72
8.2 OGC Baseline Enhancements work package deliverables summary, part 1 . . . 109
8.3 OGC Baseline Enhancements work package deliverables summary, part 2 . . . 110
8.4 OGC Baseline Enhancements work package deliverables summary, part 3 . . . 111

9.1 Compression and Generalization work package deliverables summary . . . 115

10.1 ArcticSDI and GeoPackage work package deliverables summary 122

List of Abbreviations

ABI	Activity Based Intel- ligence
AMQP	Advanced Message Queuing Protocol
AtomPub	Atom Publishing Protocol
AVI	Aviation (thread name)
BBOX	Bounding Box
CDR	Content Discovery and Retrieval
CITE	Compliance Interoperability and Testing
CMD	Command Center
CMP	Compliance (thread name)
CNS	Consolidation (thread name)
CSMW	Community Sensor Model Working Group
CSW	Catalog Service Web
CTL	Compliance Testing Language
DAP	Data Access Protocol
DCAT	Data Catalog Vocabulary
DGIWG	Defence Geospatial Information Working Group
DISA	Defense Information System Agency
DWG	Domain Working Group
EO	Earth Observation
EOWCS	Earth Obseration Profile Web Coverage Service
EXI	Efficient XML Interchange format
FGDC	Federal Geographic Data Committee
FIXM	Flight Information Exchange Model
FO	Field Operations
GDAL	Geospatial Data Abstraction Library
GEOINT	Geospatial intelligence
GeoXACML	Geospatial XACML
GIBS	Global Imagery Browse Services
GML	Geography Markup Language
HDF	Hierarchical Data Format
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
JSON-LD	JSON Linked Data
KML	Keyhole Markup Language
LDS	Linked Data and Advanced Semantics (thread name)
LiDAR	Light detection and ranging
LSA	Large Scale Analytics (thread name)
MTOM	Message Transmission Optimization Mechanism

NASA	National Aeronautics and Space Administration
netCDF	network Common Data Form
NetCDF-CF	NETCDF Climate Forecasting
NSG	National System for Geospatial Intelligence
OBP	Object Based Production
OGC	Open Geospatial Consortium
OPeNDAP	Open-source Project for a Network Data Access Protocol
POI	Points-of-interest
PubSub	Publication Subscription
RDF	Resource Description Framework
SOS	Sensor Observation Service
SPARQL	SPARQL Protocol and RDF Query Language
SWE	Sensor Web Enablement
SWG	Standards Working Group
TEAM	Test, Evaluation, And Measurement Engine
TSPI	Time-Space-Position-Information Standard
TWMS	Tiled Web Mapping Service
U.S.	United States
UML	Unified Modeling Language
USGS	U.S. Geological Survey
W3C	World Wide Web Consortium
WCPS	Web Coverage Processing Service
WCS	Web Catalog Service
WFS	Web Feature Service
WIS	Web Integration Service
WKT	Well Known Text
WMS	Web Mapping Service
WMTS	Web Mapping Tile Service
WPS	Web Processing Service
WS	Web Service
WSDL	Web Services Description Language
XACML	eXtensible Access Control Markup Language
XOP	XML-binary Optimized Packaging
XXE	XML External Entity Injection

Chapter 1

Introduction

This Annex B provides background information on the OGC baseline, describes the Testbed-12 architecture and thread-based organization, and identifies all requirements and corresponding work items. For general information on Testbed-12, including deadlines, funding requirements and opportunities, please be referred to the Testbed-12 RFQ Main Body, which is available on the OGC Web site.

Each thread aggregates a number of requirements, work items and corresponding deliverables, which are funded by different sponsors. The work items are organized in bundles that correspond to one or more related requirements. For organizational reasons, the work items have been aggregated to work packages that are described in the following and reflected by chapters 4 to 10. The organization of work items in work packages serves organizational requirements only! The OGC Testbed-12 threads are defined in chapter 3 and briefly introduced in the following sub-chapters.

1.1 Aviation

The Aviation work package seeks to implement CCI components such as semantics, ontology, and linked data to improve interoperability of services provided by System Wide Information Management (SWIM) within FAA and Eurocontrol. Among these include the Data Broker, Catalog Service for Web (CSW), GeoSPARQL (Ontology), Semantics and Linked Data. Additionally, the Aviation thread is interested in integration of publish-subscribe messaging patterns for the exchange and request of geospatial information as well as the integration of FIXM with GML elements for compatibility with OGC web services.

1.2 Geospatial Imagery Quality Framework

The Geospatial Imagery Quality Framework work package addresses a quality frame work that includes quality parameters such as accuracy, currency, completeness, and consistency. The quality framework shall be used to compare imagery from multiple sources. Accuracy refers to spatial accuracy of a location derived from the pixel in X,Y dimensions and potentially in Z dimension. Currency refers to the temporal extent of the imagery products used to cover the associated area, since multiple overpasses are typically required to cover a large area. Completeness of imagery products refers to quality metrics including cloud cover, sensor specs on collection geometry, temporal range of the data, other spectral bands if any, radiometric depth of the pixels, etc. Finally the Consistency metric describes the consistency of colors, relative accuracy over time and over different sensors, spectral and spatial error propagation from collection to production, etc.

1.3 Coverage Access and Visualization

The Coverage Access and Visualization work package addresses several interoperability requirements in the context of coverage data access and enhanced visualization of earth observation data. The goal is to enhance the currently available WCS Earth observation profile to reflect new requirements coming from both the data provider and consumer communities. These involve new indexing mechanisms, support for data encodings such as HDF, NetCDF, and DAP, and coverage processing mechanisms. Further on, this package shall improve OGC service and encoding specifications to optimize its usage with GIBS. GIBS, the Global Imagery Browse Services system is a core EOSDIS component which provides a scalable, responsive, highly available, and community standards based set of imagery services. These services are designed with the goal of advancing user interactions with EOSDIS' inter-disciplinary data through enhanced visual representation and discovery.

1.4 GFM, Catalogs, and Semantics

The GFM, Catalogs, and Semantics work package addresses a number of topics that require further research, as non is currently supported by an endorsed OGC standard. The package reviews the baseline of the OGC view of the world, the General Feature Model (GFM), experiments with DCAT as an RDF vocabulary to facilitate interoperability between data catalogs, and addresses semantic aspects such as automated rendering of linked data and semantic mediation to support heterogeneous environments.

1.5 OGC Baseline Enhancements

The OGC Baseline Enhancement work package addresses a large number of services and information and exchange models that are already established and approved OGC Standards; or services that are ready for operational implementation based on previous research and evaluation. Work items in this package include topics such as REST and (Geo)JSON, catalogs, security and SOAP, data and data portals, data quality, GeoPackage, tiling, OWSContext, conflation, asynchronous service interaction, and ontologies for portrayal, semantic mediation, and compliance.

1.6 Compression and Generalization

The Compression and Generalization work package addresses aspects resulting from the unprecedented volumes of data that are generated continuously. The mechanisms for sharing and processing that data need to keep pace in order to continue to meet requirements. Therefore, this work package analyzes data compression and generalization techniques that shall help to effectively reduce the amount of processed and transferred data at minimal information loss.

1.7 ArcticSDI and GeoPackage

The Arctic SDI provides an infrastructure to improved access to geospatial data that can help us better to predict, understand and react to changes in the Arctic.

Its development is facilitated by the National Mapping Agencies of the eight Arctic countries. The OGC together with its collaborators Natural Resources Canada, the USGS, territories and states that are part of the Arctic, Arctic Council working groups, Arctic SDI member countries, and a number of OGC member organizations are currently engaged in a OGC Pilot project with the goal to articulate the value of interoperability and to demonstrate the usefulness of standards within the ArcticSDI domain. In its first phase, the ArcticSDI Pilot project develops an inventory of available geospatial Web services across the Arctic, which can be used to reflect a broad range of thematic data layers. In parallel, Phase 1 defines the core components of the ArcticSDI architecture that shall, at least partly, be developed in the context of this testbed.

The work package is concluded by an investigation of the OGC GeoPackage as a single alternative delivery format for the USGS Topo Combined Vector Product and the Topo TNM Style Template. The GeoPackage contents of the Topo Combined Vector Product shall be extended to include imagery and hillshade data. It shall be evaluated whether the point, multi-point, line, and polygon contents in the Geopackage can be tied directly to a predefined symbology set via the Symbology Encoding Implementation Specification.

Chapter 2

Testbed Baseline

2.1 Types of Deliverables

The OGC Testbed 12 threads require several types of deliverables. It is **emphasized that deliverable indications "funded" or "unfunded" in this Annex B are informative only**. Please be referred to Testbed-12 RFQ Main Body for binding definitions.

2.1.1 Documents

Engineering Reports (ER), Information Models (IM), Encodings (EN), and Change Requests (CR) will be prepared in accordance with OGC published templates. Engineering Reports will be delivered by posting on the OGC Portal Pending Documents list when complete and the document has achieved a satisfactory level of consensus among interested participants, contributors and editors. Engineering Reports are the formal mechanism used to deliver results of the Interoperability Program to sponsors and to the OGC Standards Program and OGC Standards Working Groups for consideration. It is emphasized that **participants delivering engineering reports must also deliver Change Requests that arise from the documented work**.

2.1.2 Implementations

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

2.2 OGC Reference Model

The OGC Reference Model (ORM) `OGC Reference Model version 2.1, document OGC 08-062r7` provides an architecture framework for the ongoing work of the OGC. Further, the ORM provides a framework for the OGC Standards Baseline. The OGC Standards Baseline consists of the member-approved Implementation/Abstract Specifications as well as for a number of candidate specifications that are currently in progress.

The structure of the ORM is based on the Reference Model for Open Distributed Processing (RM-ODP), also identified as ISO 10746. This is a multi-dimensional approach well suited to describing complex information systems.

The ORM is a living document that is revised on a regular basis to continually and accurately reflect the ongoing work of the Consortium. We encourage respondents to this RFQ to learn and understand the concepts that are presented in the ORM.

This Annex B refers to the RM-ODP approach and will provide information on some of the viewpoints, in particular the Enterprise Viewpoint, which is used here to provide the general characterization of work items in the context of the OGC Standards portfolio and standardization process, i.e. the enterprise perspective from an OGC insider.

The Information Viewpoint considers the information models and encodings that will make up the content of the services and exchanges to be extended or developed to support this testbed. Here, we mainly refer to the OGC Standards Baseline, see section 2.3.

The Computational Viewpoint is concerned with the functional decomposition of the system into a set of objects that interact at interfaces – enabling system

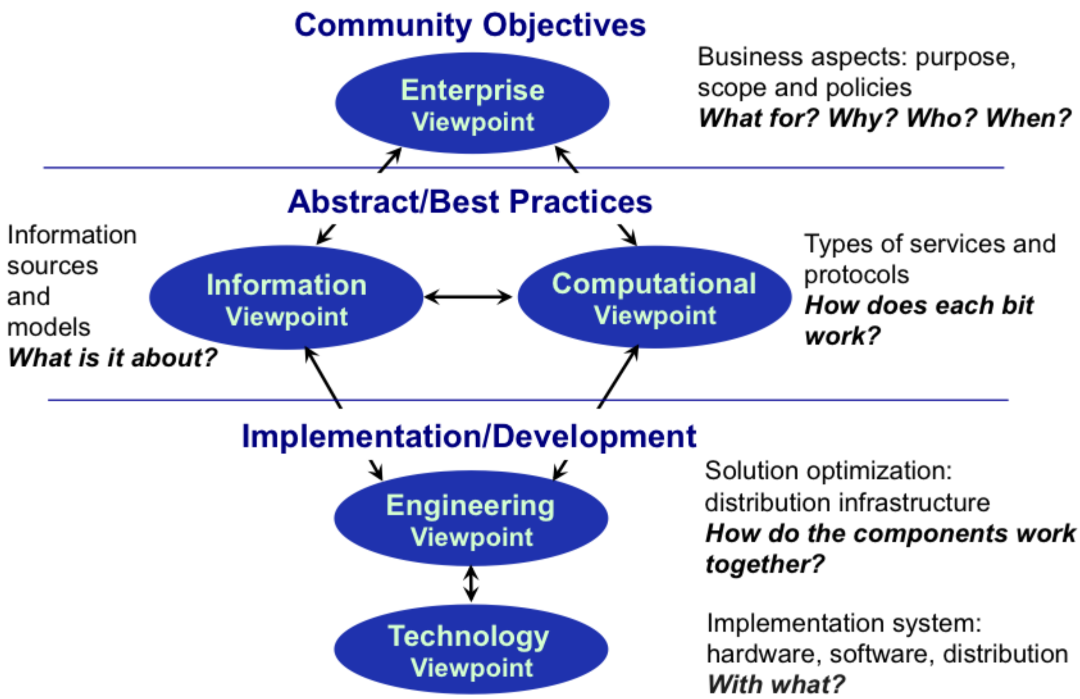

FIGURE 2.1: Reference Model for Open Distributed Computing

distribution. It captures component and interface details without regard to distribution and describes an interaction framework including application objects, service support objects and infrastructure objects. The development of the computational viewpoint models is one of the first tasks of the testbed, usually addressed at the kick-off meeting.

The Engineering Viewpoint is concerned with the infrastructure required to support system distribution. It focuses on the mechanisms and functions required to: a) support distributed interaction between objects in the system and b) hides the complexities of those interactions. It exposes the distributed nature of the system, describing the infrastructure, mechanisms and functions for object distribution, distribution transparency and constraints, bindings and interactions. The engineering viewpoint will be developed during the testbed, usually in the form of TIEs (Technical Interaction Experiments), where testbed participants define the communication infrastructure and assign elements from the computational viewpoint to physical machines used for demonstrating the testbed results.

2.3 OGC Standards Baseline

The OCG Standards Baseline is the currently approved set of OGC standards and other approved supporting documents, such as the OGC abstract specifications and Best Practice Documents. OGC also maintains other documents relevant to the Interoperability Program including Engineering Reports, Discussion Papers, and White Papers.

OGC standards are technical documents that detail interfaces or encodings. Software developers use these documents to build open interfaces and encodings into their products and services. These standards are the main "products" of the Open Geospatial Consortium and have been developed by the membership to address specific interoperability challenges. Ideally, when OGC standards are implemented in products or online services by two different software engineers working independently, the resulting components plug and play, that is, they work together without further debugging. OGC standards and supporting documents are available to the public at no cost. OGC Web Services (OWS) are OGC standards created for use in World Wide Web applications. For this testbed, it is emphasized that all OGC members have access to the latest versions of all standards. If not otherwise agreed with the testbed architects, these shall be used in conjunction with - in particular - engineering reports resulting from previous testbeds.

Any documents and Schemas (xsd, xslt, etc) that support an approved (that is, approved by the OGC membership) OGC standard can be found in the official OGC Schema Repository.

The OGC Testing Facility web page provides online executable tests for some OGC standards. The facility helps organizations better implement service interfaces, encodings and clients that adhere to OGC standards.

2.4 Data

All participants are encouraged to provide data that can used to implement the various scenarios that will be developed during the testbed. A number of testbed sponsors will provide data, but it might be necessary to complement these with additional data sets. Please provide detailed information if you plan to contribute data to this testbed.

2.5 Services in the Cloud

Participants are encouraged to provide data or services hosted in the cloud. There is an overarching work item to provide cloud-hosting capabilities to allow thread participants to move services and/or data to the cloud.

Chapter 3

Testbed Threads

3.1 Background

Testbed-12 is organized in a number of threads. Each thread combines a number of work packages that are further defined in chapters 4 to 10. The threads are built around a master scenario and integrate both an architectural and a thematic view. Those two pillars allow to keep related work items closely together, which helps understanding how the various aspects interact and partly even depend on each other. Table 3.1 lists all Testbed-12 work items:

Work-Item	**Work-Item**
Aviation	Geospatial Imagery Quality Framework
Coverage Data Access	Visualization of Earth Observation Data
General Feature Model	Web Feature Service and TopoJSON
Semantic Portrayal	Semantic Mediation
Clients	Security and SOAP
REST,JSON,and GeoJSON	Semantic Enablement
Async. Service Interaction	Tiling
Compression and Generalization	SWE for LiDAR and Streaming
WFS Synchronization	Catalog
Capabilities	Big Data and Tile Stores
Web Integration Service	Conflation
Data Quality	GeoPackage
OGC WebServices Context	UML Shape Change
Data and Portal	User Guides
Compliance Testing	ArcticSDI
Generalization	

TABLE 3.1: Overview of the Testbed-12 work items

Testbed-12 identifies seven threads in total, the field operations thread (**FO**), the large scale analytics thread (**LSA**), the linked data and advanced semantics thread (**LDS**), the command center thread (**CMD**), the OGC consolidation thread (**CNS**), the compliance thread (**CMP**), and the aviation thread (**AVI**). Each thread is defined in further detail below. Each thread will be guided by a thread architect. The close cooperation between all thread architects ensures an efficient and consistent development across all threads that leads – among all individual results – into a final joint demonstration event. The assignment of the various work items to threads is further outlined in the following sections and illustrated in figure 3.1.

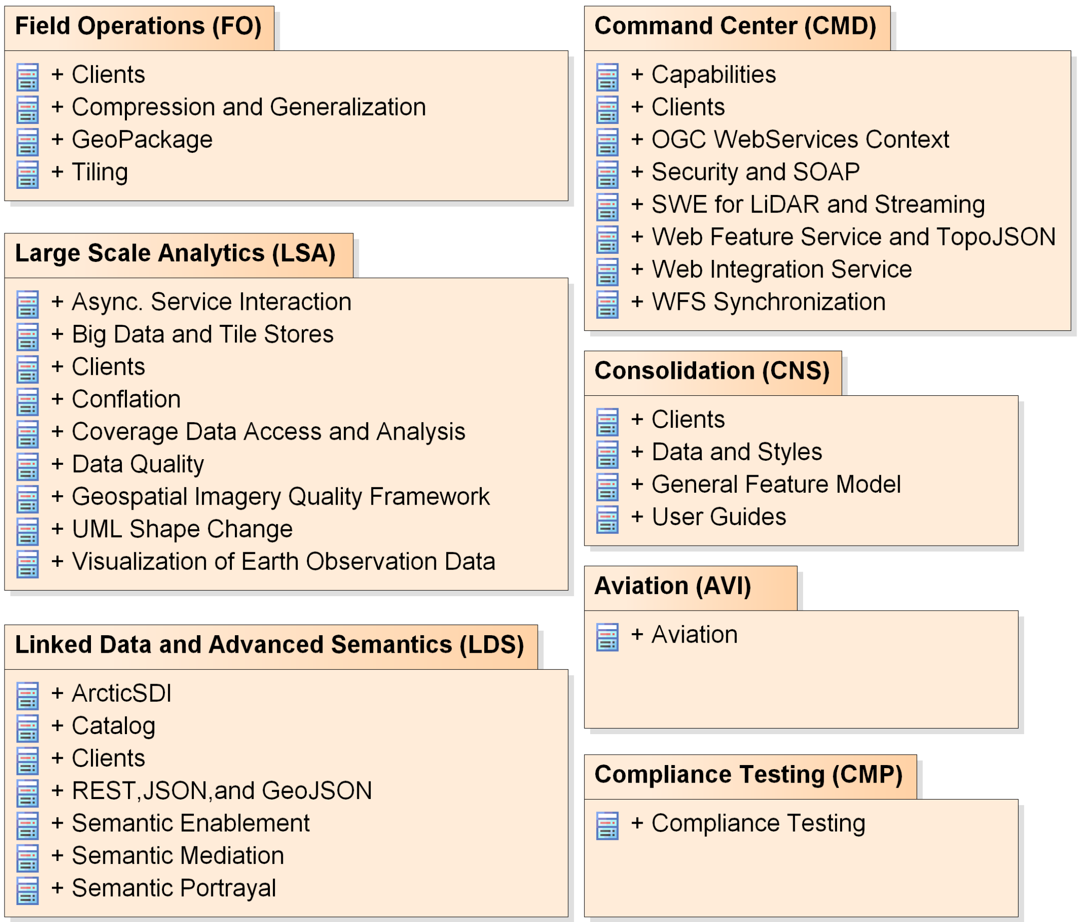

FIGURE 3.1: Assignment of Work Items to Threads

3.2 Field Operations (FO) Thread

In many large scale operations, such as wild fire fighting, evacuation and rescue, flooding, or earth quakes, response teams operate under sub-optimal conditions in terms of Internet connectivity and bandwidth. Therefore, response teams need to take data into the field for offline usage. Compression techniques and data generalization patterns become important when live data streams from servers are required.

Work-Item	**Section**
GeoPackage	8.17, 10.4
Compression and Generalization	8.8, 9.3
Tiling	8.7
Clients	7.5

TABLE 3.2: Overview of the Field Operations Thread

3.3 Large Scale Analytics (LSA) Thread

The large scale analytics thread addresses short and long term planning and analysis of geospatial topics, domains, and questions. It includes elements that help the investigator to discover, retrieve, process, and visualize data in an optimal way. It is complemented by data provision aspects, which include the conversion of conceptual models into application models.

Work-Item	**Section**
Visualization of Earth Observation Data	6.3
Big Data and Tile Stores	8.13
Coverage Data Access and Analysis	6.4
Geospatial Imagery Quality Framework	5
Data Quality	8.16
Async. Service Interaction	8.6
Conflation	8.15
UML Shape Change	8.19
Clients	7.5

TABLE 3.3: Overview of the Large Scale Analytics Thread

3.4 Linked Data and Advanced Semantics for Data Discovery and Dynamic Integration (LDS) Thread

The Linked Data and Advanced Semantics for Data Discovery and Dynamic Integration (LDS) Thread integrates all elements that go beyond traditional client server interaction with powerful query languages and complex Web service interfaces. It includes all aspects that allow semantic enrichment and mediation, and approaches such as JSON-LD, REST, and DCAT.

Work-Item	**Section**
Semantic Enablement	8.5
REST,JSON,and GeoJSON	8.4
Semantic Portrayal	7.4.3
Semantic Mediation	7.4.4
Catalog	8.11 , 7.4.2
ArcticSDI	10.3
Clients	7.5

TABLE 3.4: Overview of the Linked Data and Advanced Semantics

3.5 Command Center (CMD) Thread

The Command Center Thread aggregates all activities and work items related to incident management operations. During the course of an event, commanders require a common operational picture in order to take important decisions in an optimal way. The thread includes items that contribute to this common operational picture by using context documents to share common views, security settings to allow handling sensitive data, improvements to the capabilities documents to allow quick discovery of service and data offerings, and synchronization and streaming aspects.

Work-Item	Section
OGC WebServices Context	8.18
Security and SOAP	8.3
SWE for LiDAR and Streaming	8.9
Web Feature Service and TopoJSON	7.4.1
WFS Synchronization	8.10
Capabilities	8.12
Web Integration Service	8.14
Clients	7.5

TABLE 3.5: Overview of the Command Center (CMD) Thread

3.6 Consolidation (CNS) Thread

The Consolidation Thread includes activities that help improving the current technical baseline and its implementation by developing a set of user guides, analyses on how to handle and integrate traditionally non-geospatial data with the General Feature Model, and develops an environment of data that will be made available for future testbeds.

Work-Item	Section
General Feature Model	7.3
User Guides	8.21
Data and Portal	8.20
Clients	7.5

TABLE 3.6: Overview of the Consolidation (CNS) Thread

3.7 Aviation (AVI) Thread

The Aviation Thread includes all work items that contribute to the improvement of air traffic control. It addresses aspects such as advanced use of catalogs for aviation, brokering of information across services, asynchronous messaging, aviation specific security needs, aviation semantics and business rules, and aeronautical and flight information models.

Work-Item	Section
Aviation	4

TABLE 3.7: Overview of the Aviation (AVI) Thread

3.8 Compliance Testing (CMP) Thread

The compliance testing thread includes all improvements to the OGC compliance test and certification program, such as improvements to the test engine and tests, development of virtual machines to facilitate testing, and the development of reference implementations.

Work-Item	Section
Compliance Testing	8.22

TABLE 3.8: Overview of the Compliance Testing (CMP) Thread

Chapter 4

Aviation

4.1 Background

The Federal Aviation Administration (FAA) and EUROCONTROL have developed the Aeronautical Information Exchange Model (AIXM) as a global standard for the representation and exchange of aeronautical information. AIXM was developed using the OGC Geography Markup Language (GML) tailored to the specific requirements for the representation of aeronautical objects, including the temporality feature that allows for time dependent changes affecting AIXM features. The overall objectives of the FAA and EUROCONTROL are to use AIXM as a basis for modernizing their aeronautical information procedures and transitioning to a net-centric, global aeronautical management capability. The FAA and EUROCONTROL plan to use AIXM in the net-centric (System Wide) Information Management (SWIM) related components of the US NextGen and EU SESAR programs.

The Weather Information Exchange Model (WXXM) is the proposed standard for the exchange of aeronautical weather information in the context of a net-centric and global interoperable Air Transport System. It is currently under development by FAA and EUROCONTROL in support of the NextGen and SESAR programs. WXXM uses the OGC Geography Markup Language (GML) tailored to the specific requirements for aeronautical meteorology and is based on the OGC Observation & Measurement Model. Within the context of global harmonization, the WXXM is harmonized and coordinated with the World Meteorological Organization (WMO), the organization traditionally responsible for standards in meteorology. The OGC's Meteorology and Oceanography Domain Working Group has set up the appropriate mechanisms and interfaces between OGC and WMO to support this global harmonization and coordination effort.

The Flight Information Exchange Model (FIXM) is an exchange model capturing Flight and Flow information that is globally standardized. The need for FIXM was identified by the International Civil Aviation Organization (ICAO) Air Traffic Management Requirements and Performance Panel (ATMRPP) in order to support the exchange of flight information as prescribed in Flight and Flow Information for a Collaborative Environment (FF-ICE). FIXM is the equivalent, for the Flight domain, of AIXM (Aeronautical Information Exchange Model) and WXXM (Weather Information Exchange Model), both of which were developed in order to achieve global interoperability for, respectively, AIS and MET information exchange. FIXM is therefore part of a family of technology independent, harmonized and interoperable information exchange models designed to cover the information needs of Air Traffic Management.

By sponsoring the OGC Testbed-12, EUROCONTROL aims to increase industry adoption of these formats and to support the operational use and validation of these emerging standards. The Agency is particularly interested in the Testbed-12 deliverables which will help to enhance the OGC/ISO standards (WFS, Publication/Subscribe mechanisms, etc.) so that they meet SESAR SWIM requirements. EUROCONTROL also intends to use the Testbed as a SWIM verification platform that will help to further challenge the SWIM requirements. The results of Testbed-12 will be delivered to several SESAR WP13/14 projects, as contributions to the definition and implementation of SWIM-enabled services and systems.

The Aviation thread seeks to implement CCI components such as semantics, ontology, and linked data to improve interoperability of services provided by System Wide Information Management (SWIM) within FAA and Eurocontrol. Among these include the Data Broker, Catalog Service for Web (CSW), GeoSPARQL (Ontology), Semantics and Linked Data. Additionally, the Aviation thread is interested in integration of publish-subscribe messaging patterns for the exchange and request of geospatial information.

4.2 Requirements and Work Items

Figure 4.1 illustrates all work items that shall be addressed in this work package. They are described in further detail in the following sections. All funded work items are shaded in green, unfunded in blue. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements. Arrows indicate the associations between work items and requirements.

The aviation work package contains in total seven requirements. These requirements are further refined in the following sections. The aviation architecture definition complements this list as an overarching requirement that helps understanding the architectural classification and integration of the various work items.

4.3 Aviation Architecture

OGC and Testbed-12 participants will develop a technical architecture describing the arrangements of the service applications that satisfy the overall objectives for the project. The technical architecture will be reviewed and approved by the Testbed-12 sponsors.

Flight, aeronautical, and meteorological information necessary for the execution of OGC Testbed-12 scenarios should use as much as possible existing data sources. EUROCONTROL and the FAA may provide an AIXM, WXXM, FIXM, or AMXM set of files that contain static aeronautical data as needed for testing and for supporting the scenario.

Deliverables of the Aviation Architecture work item

ID	Funding	Name
E005	unfunded	Aviation Architecture ER

TABLE 4.1: Aviation Architecture deliverables

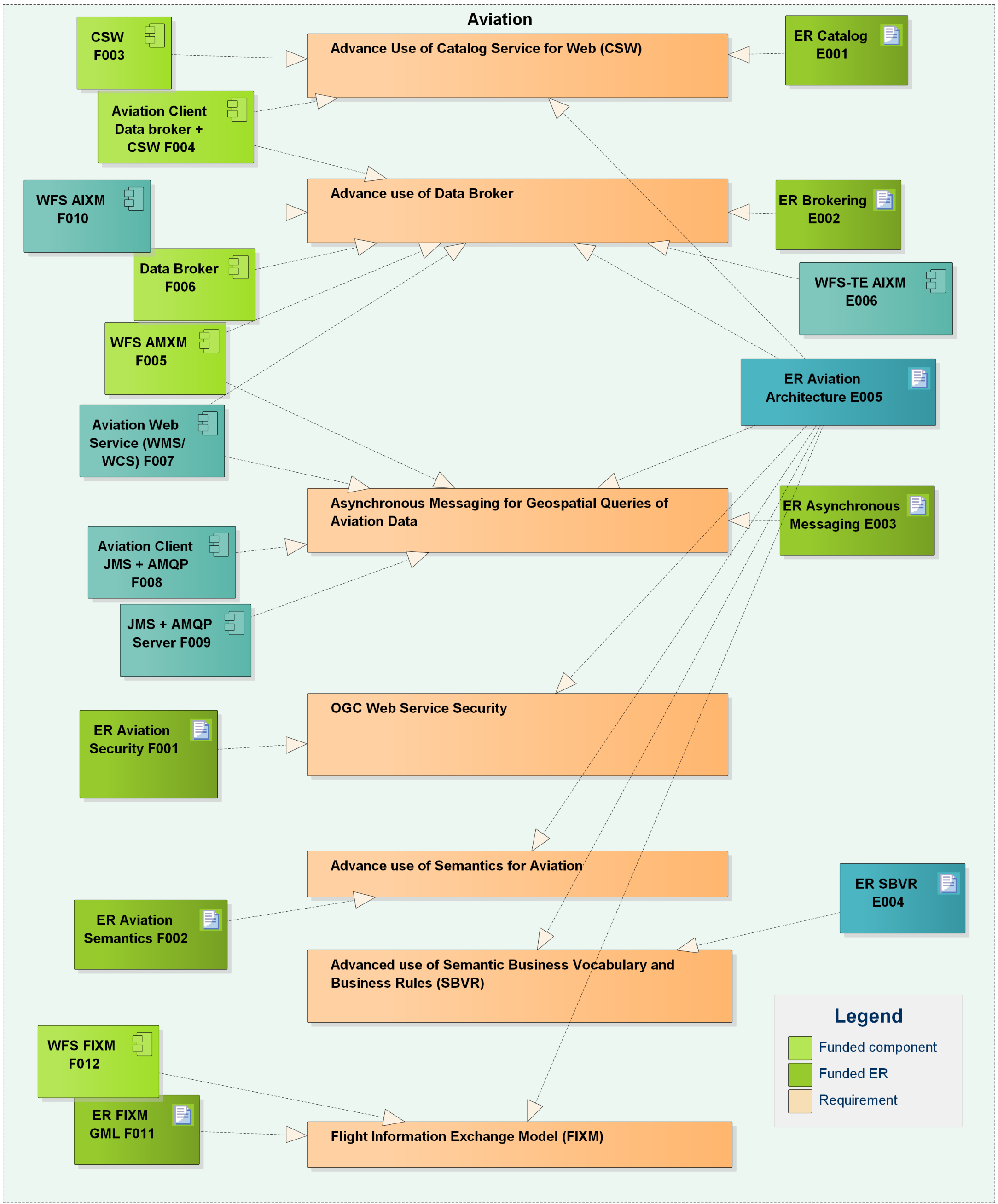

FIGURE 4.1: Aviation work package: Requirements and work items

4.4 Advance use of Data Broker

An important principle of SOA is the notion of composing capabilities provided by individual services into complex behavior. A requester should be able to compose a solution using functionality or data offered by multiple services without worrying about underlying differences in those services.

Each OGC service is designed to offer a specific type of data product (e.g. an XML document by WFS, a map by WMS, etc.) via a service-specific interface. This task seeks to establish a single service interface that would allow access to multiple data sources, possibly heterogeneous with respect to the types of data provided.

For example: A client may request data related to a specific air navigation route. The data may consist of a set of aeronautical and an associated set of maps served by a WFS or WMS component, respectively. Currently, the Data Broker, as developed in Testbed-11, is capable of combining multiple WFS responses into a single homogeneous response.

For this task, a Data Broker must combine two separate service request types from heterogeneous responses. The Data Broker task shall also utilize the CSW catalog service if possible. Additionally, the participant shall demonstrate the use of a data broker to discover services, discover related data using semantic linked data, and conflate the heterogeneous response. The participant shall demonstrate two approaches:

1. A request to separate WFS with heterogeneous data using AMXM and AIXM to demonstrate the conflation of two heterogeneous data sets from the same type of web service;
2. A request to a WFS and another type of service (e.g. WMS or WCS) with a heterogeneous response. In this second approach, the participant shall determine and propose a suitable secondary service type prior to implementation according to the available data provided.

Regarding the AIXM/AMXM transformation, the idea is that (i) the client uses an AMXM WFS to fetch AMXM data, and (ii) that the AMXM WFS issues a request to the AIXM WFS thus instantiating a data transformation chain from generic Airport Mapping Data (AIXM) into specific AMDB data (AMXM) as specified in the related EUROCAE/RTCA Standards. Eurocontrol will provide access to the existing EUROCAE/RTCA AMXM UML & XML Schema and samples of AMXM data. The AMXM UML and AMXM XML Schema are in a mature state of development and ready for publication. Depending on the use case envisaged it should be possible to share AIXM (source) data. It is to be noted that the testbed will not intend to achieve completeness in terms of data coverage.

The focus is on the demonstration/feasibility of a data chain by coupling WFS instances. Therefore the selection of a few relevant features should be sufficient ideally spread over some point/line/polygon types. EUROCONTROL will make available the earlier AMDB to AIXM5.1 mapping which can be used to get some insight in the transformation. Finally, the AIRM can also be used to bridge the semantics between the two WFS instances. Eurocontrol can provide the necessary insights if required. Although the EUROCAE/RTCA standards come with specific requirements regarding features (geometrical constraints & functional constraints) it is considered that these are not a key aspect of the testbed. Yet chaining feature geometry checking services would constitute an excellent bonus to demonstrate the power of the OGC approach, leveraging the end-goal of SWIM information services towards quality and trust. Upon request Eurocontrol will share any relevant AMDB insights the testbed members may require to perform their work.

Input documents for consideration include the following OGC documents but should not be limited to:

- OGC Testbed-11 – Data Broker Engineering Report
- Any OGC services standards for WFS, WMS, WCS, etc as applicable for the use case in heterogeneous OGC service request via Data Broker
- Current standards for Catalog Service for Web (CSW) for use of Catalog service with Data Broker for service and data discovery
- Engineering Reports from CCI thread regarding Catalog service and linked data for conflation.
- OWS-9 Data Transmission Management Engineering Report for the use of conflation of data

Deliverables Advance use of Data Broker

ID	Funding	Name
E002	funded	Brokering ER
E005	unfunded	Aviation Architecture ER
E006	unfunded	WFS-TE AIXM
F004	funded	Aviation Client Data Broker + CSW
F005	funded	WFS AMXM
F006	funded	Data Broker
F007	unfunded	OGC Web Service Aviation
F010	unfunded	WFS AIXM

TABLE 4.2: Advance use of Data Broker deliverables

4.5 Advance Use of Catalog Service for Web (CSW)

A service registry is an important component of SOA and a key building block in the context of the SWIM information infrastructure. Both FAA and EUROCONTROL implementations of SWIM have established their own registries (FAA's NAS Service Registry/Repository (NSRR) and EUROCONTROL's European SWIM Registry respectively). And although these registries were developed independently, they share the same conceptual model (SDCM) and are very similar in terms of basic functionalities. In much the same way, OGC Catalogue services support the ability to publish and search collections of descriptive information (metadata) for data, services, and related information objects.

However, because existing SWIM registries do not store spatial information (e.g. BoundingBox, CRS, etc.) and therefore lack abilities to support geospatial queries, the developers of SWIM OGC-compliant services are looking to establish OGC-compliant Catalogue services in parallel to the existing registries.

This task shall demonstrate a registry solution using Catalog Service for Web (CSW) with integration to the FAA NSRR as a single entry point for all SWIM service discovery support. Secondly, the task shall analyze and report on the current state of the FAA and Eurocontrol SWIM registries to determine which modifications may be required in order to harmonize the implementations.

The catalog/registry solution shall consider the functionality for content and service discovery while providing the following features:

- Simple user interface for searching for content via geospatial queries (e.g.

bounding box). Specifically, a RESTful service interface for semantic querying of geospatial data

- Provide discoverable content for both web services and publish/subscribe content (i.e. JMS data feeds)
- Provide capability to search across multiple SWIM domain registries such as FAA NSRR and Eurocontrol SWIM Registry
- Provide a harmonized service description conceptual model (SDCM)
- Cost effective approach to supporting the governance of service metadata including quality and integrity.

Input documents for consideration include the following OGC documents but should not be limited to:

- OGC Testbed-11 Implementing Linked Data and Semantically Enabling OGC Services Engineering Report
- Any FAA documentation on FAA SWIM registry and NSRR FAA SWIM registry and NSRR
- Any Eurocontrol documentation on European SWIM Registry
- Utilization of Faceted Classification in the Context of the SWIM Service Registry
- FAA/SESAR JU Service Description Conceptual Model (SDCM) 1.0
- Data.gov has a registry for their data that was originally not supporting geospatial data. Parts of the OGC Catalog Service for Web (CSW) were added to the Data.gov registry

Deliverables

ID	**Funding**	**Name**
E001	funded	Catalog ER
E005	unfunded	Aviation Architecture ER
F003	funded	CSW
F004	funded	Aviation Client Data Broker + CSW

TABLE 4.3: Advance Use of Catalog Service for Web deliverables

4.6 Asynchronous Messaging for Geospatial Queries of Aviation Data

JMS and AMQP are used in ATM for publish-subscribe messaging of aviation data. Currently, OGC Web Services only support synchronous web service request-response query capabilities. This task shall investigate the means to incorporate publish-subscribe messaging patterns for the retrieval of aviation data (i.e. AIXM, WXXM, FIXM, or AMXM) information using geospatial queries through a JMS interface and AMQP interface. This task shall demonstrate a capability using the recommended approach. Suggested solutions may consider, but are not limited to, the following:

- XML or JSON standard for the name-value pairs used in JMS/AMQP headers
- Standard data model using GML profiles (e.g. AIXM profile metadata header format for AIXM messages)
- SOAP over JMS

The results of this demonstration shall be reported in the Asynchronous response ER (A067, described in chapter 8.6) recommending a method for managing message JMS metadata in publish-subscribe models. The results shall contain the detailed architecture for the components developed for this activity.

References The following deliverables have been defined based on the scope of the requirement for this task:

- NEMS Interface Control Document (ICD) V6.0 (not available on-line) – To be provided by FAA upon request.
- SWIM NEMS User Guide
- OGC Event Service Specification

Deliverables

ID	Funding	Name
E003	funded	Asynchronous Messaging ER
E005	unfunded	Aviation Architecture ER
F005	funded	WFS AMXM
F007	unfunded	Aviation Web Service (WMS/WCS)
F008	unfunded	Aviation Client JMS + AMQP
F009	unfunded	JMS + AMQP Server

TABLE 4.4: Asynchronous Messaging for Geospatial Queries of Aviation Data deliverables

4.7 Advance use of Semantics for Aviation

The current service description standards, including WSDL, OWS and XML Schema, operate almost entirely at the syntactic level. They focus only on describing a service's exposed functionality (e.g. methods signatures, input/output types, etc.) while failing to capture enough semantic data (e.g. service capabilities, qualities of service, etc.). Service implementations often work around this limitation by augmenting the service descriptions with an external semantic layer. In FAA, this semantic layer has been implemented mostly through developing human-readable regulating documents like FAA-STD-065 and FAA-STD-073. Effort was also invested in developing an ontology (WSDOM) that describes all aspects of a service using formal languages like OWL and SKOS.

However, the increased demand for service interoperability has only been partially met by these activities. To take full advantage of the capabilities provided by OGC services, it is necessary to introduce semantic service descriptions that are suitable for use by both humans and machines. Semantic enrichment of service descriptions (and of related artifacts like data models) should be able to address service discovery, composition and interoperability to a significantly higher degree than occurs today.

This effort builds upon the work completed in OWS-9 Semantic Query Mediation, Testbed 10 Ontology, and Testbed-11 Advance use of a common symbology. For Testbed-12, this task shall evaluate the FAA Web Service Description Ontological Model (WSDOM) to determine how it can be improved for use with OGC Web Services and easily integrated to the SWIM registry with the Catalog service.

This task shall provide a demonstration of Semantic ontology based on the FAA WSDOM ontology, while demonstrating interoperability with OGC semantic capabilities. The task should include concepts from OGC Testbed-11 - Catalog Recommendation ER in which ontology URIs are embedded in metadata and service descriptions. The implementation should also consider RDF/RDFS, OWL, SKOS, and other open standards. The results and ontology implementation shall be recorded in a Semantic Ontology for Aviation Engineering Report.

References

- OGC Testbed 10 Cross Community Interoperability (CCI) Ontology Engineering Report FAA Web Service Description Ontological Model (WSDOM) OGC 14-049

Deliverables

ID	Funding	Name
E005	unfunded	Aviation Architecture ER
F002	funded	Aviation Semantics ER

TABLE 4.5: Advance use of Semantics for Aviation deliverables

4.8 Advanced use of Semantic Business Vocabulary and Business Rules (SBVR)

Semantics Of Business Vocabulary And Rules (SBVR) defines the vocabulary and rules for documenting business vocabularies, business facts, and business rules. It defines structural and operative rules. AIXM 5, following SBVR, has identified AIXM Structural rules and AIXM Operative rules. Some of the structural rules can be captured in the AIXM schemas. The ones that cannot be captured in a schema can be formalized in to Schematron or other similar rules-engine language.

- SBVR allows provision of rules in a friendly way. For example:
- Each *[...].lowerLevel* that has an uom equal to 'FL' should have 2 or 3 digits
- Each *AirportHeliport.ARP* must have *horizontalAccuracy* and *AirportHeliport.ARP.horizontalAccuracy* should be at most 1sec

This task will advance using SBVR at the data content level. An initial step was made in OGC Testbed-11, where an initial SBVR vocabulary for GML elements was developed, together with a tool prototype that converts SBVR into executable Schematron code. For OGC Testbed-12, the participant shall investigate the use of semantic vocabularies (SBVR) at the next level, trying to capture rules that identify operational constraints based on aircraft characteristics and the infrastructure characteristics, including the actual status. For example, express SBVR rules that relate:

- the aircraft type to the possibility to operate on a given runway, taking into consideration landing distances, risk of overshooting the last exit taxiway and limitations in turning radius of the aircraft, etc.
- the possibility to land on a given runway considering the actual weather data (RVR/visibility, availability of non-precision/precision approaches and of critical navaids, etc.)

Then, use such rules in order to identify airports along the trajectory that could be used in case of an emergency landing (less rules would apply) or as additional alternates (more rules would apply).

Deliverables

ID	Funding	Name
E004	unfunded	SBVR ER
E005	unfunded	Aviation Architecture ER

TABLE 4.6: Advanced use of Semantic Business Vocabulary and Business Rules deliverables

4.9 Aviation OGC Web Service Security

OGC Testbed-11 established a security framework for use with OGC web services. The need for cyber security measures within the Aviation community is critical for the safety of future operations with SWIM services. This task shall investigate the use of the security framework in OGC Testbed-11 and recommend best practices for the implementation with OGC services in OGC Testbed-12.

Deliverables

ID	Funding	Name
E005	unfunded	Aviation Architecture ER
F001	funded	Aviation Security ER

TABLE 4.7: Aviation OGC Web Service Security deliverables

4.10 Flight Information Exchange Model (FIXM)

The FAA and EUROCONTROL, in conjunction with multiple other international partners as well, are currently in the process of developing the Flight Information Exchange Model (FIXM). FIXM is an exchange model capturing Flight and Flow information that is globally standardized. The need for FIXM was identified by the International Civil Aviation Organization (ICAO) Air Traffic Management Requirements and Performance Panel (ATMRPP) in order to support the exchange of flight information as prescribed in Flight and Flow Information for a Collaborative Environment (FF-ICE).

FIXM is the equivalent, for the Flight domain, of AIXM (Aeronautical Information Exchange Model) and WXXM (Weather Information Exchange Model), both of which were developed in order to achieve global interoperability for, respectively, AIS and MET information exchange. FIXM is therefore part of a family of technology independent, harmonized and interoperable information exchange models designed to cover the information needs of Air Traffic Management. Previous OGC IP initiatives developed an architecture that supports the exchange of AIXM and WXXM data. This task shall demonstrate the integration of GML profile elements into FIXM, specifically, the Feature, Time, Point, UOM, CharacterString, Integer, and Decimal types, into FIXM version 3.0.1 or an alpha release of FIXM version 4.0. The purpose of this task is to ensure that the implementation of GML elements is usable by the FIXM development community.

The participant shall:

- Describe the integration of GML into the FIXM UML model provided by FAA, and generate a FIXM XML schema from the FIXM UML model
- Provide an updated UML and XML Schema for the modified FIXM
- Demonstrate a WFS FIXM with support for queries on FIXM feature data

- Verify and validate (through tests and demonstrations)
    1. the FIXM design and
    2. the capability of the OGC Aviation Architecture to support interoperable exchange of FIXM data, and document the results (especially any improvements for FIXM)
- Document potential enhancements and identified issues for relevant documents and references. For OGC documents, change requests (especially for Standard and Best Practice documents) will be created.

Deliverables

ID	Funding	Name
E005	unfunded	Aviation Architecture ER
F011	funded	FIXM GML ER
F012	funded	WFS FIXM

TABLE 4.8: Flight Information Exchange Model deliverables

4.11 Summary

The following tables summarizes all work items that shall be delivered as part of this work package.

ID	Funding	Name
E001	funded	Catalog ER
E002	funded	Brokering ER
E003	funded	Asynchronous Messaging ER
E004	unfunded	SBVR ER
E005	unfunded	Aviation Architecture ER
E006	unfunded	WFS-TE AIXM
F001	funded	Aviation Security ER
F002	funded	Aviation Semantics ER
F003	funded	CSW
F004	funded	Aviation Client Data Broker + CSW
F005	funded	WFS AMXM
F006	funded	Data Broker
F007	unfunded	Aviation Web Service (WMS/WCS)
F008	unfunded	Aviation Client JMS + AMQP
F009	unfunded	JMS + AMQP Server
F010	unfunded	WFS AIXM
F011	funded	FIXM GML ER
F012	funded	WFS FIXM

TABLE 4.9: Aviation work package deliverables summary

Chapter 5

Geospatial Imagery Quality Framework

5.1 Background

The last decade has seen proliferation of sensors on various platforms (satellites, aerial, UAV's) that collect imagery at multiple scales/resolutions. It is estimated that several hundred small satellites and UAV's will be added to this asset base in the next 5-10 years. Currently there is no consistent quality framework that will allow end users to compare imagery from multiple sources across the different quality attributes in order to have a holistic view of imagery value as it may apply to particular set of requirements.

Testbed-12 shall develop a quality frame work called "A3C" (Accuracy, Currency, Completeness, and Consistency) that can be used to compare imagery from multiple sources. Accuracy refers to spatial accuracy of a location derived from the pixel in X,Y dimensions and potentially in Z dimension. Currency refers to the temporal extent of the imagery products used to cover the associated area, since multiple dates of collects are typically required to cover a large area. Completeness of imagery products refers to quality metrics including cloud cover, sensor specs on collection geometry, temporal range of the data, other spectral bands if any, radiometric depth of the pixels, etc. Finally, the Consistency metric describes the consistency of colors, relative accuracy over time and over different sensors, spectral and spatial error propagation from collection to production, etc.

5.2 Requirements and Work Items

The following figure illustrates all work items that shall be addressed in this work package. They are described in further detail in figure 5.1. All funded work items are shaded in green, unfunded in blue. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements.

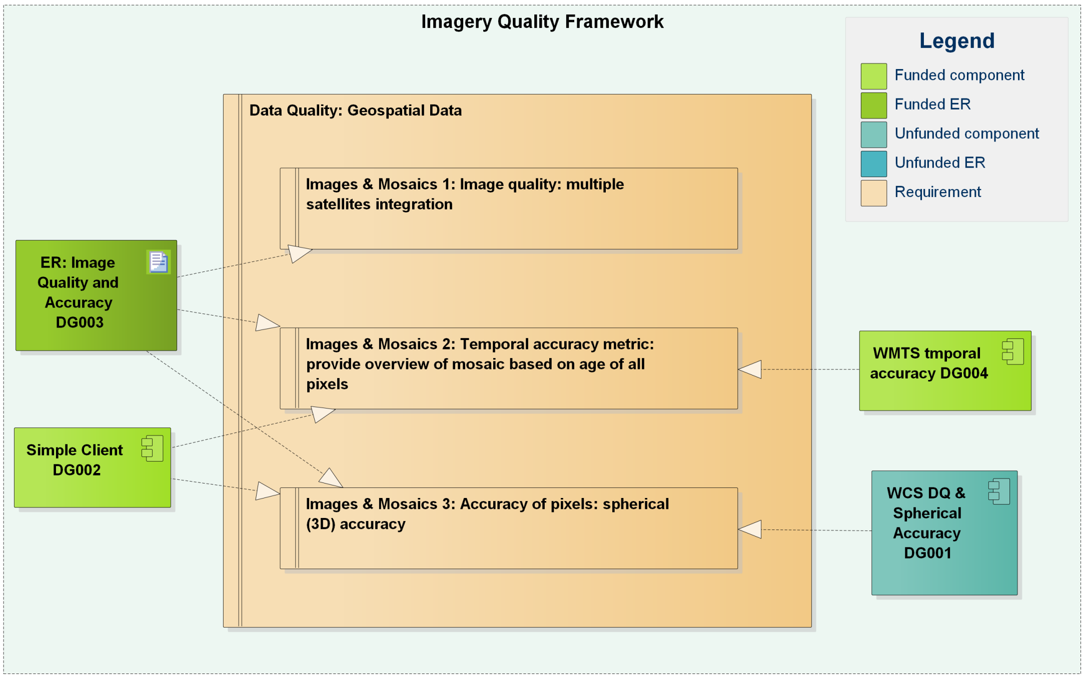

FIGURE 5.1: Geospatial Data Quality work package: Requirements and work items

5.3 Accuracy, Currency, Completeness, Consistency

The detailed image quality framework to be developed in Testbed-12 shall support comparing multiple sources of imagery. A framework of this nature, vetted by the OGC experts, improves interoperability of multiple imagery sources and provides users a much more consistent mechanism to assess imagery products.

5.3.1 Desired Output

The goal of this work package is to develop the image quality framework that addresses all aspects mentioned below. Its functionality shall be demonstrated using at least a WMTS implementation that supports temporal accuracy information. If funding allows, a WCS shall be implemented that supports all A3C data. Independent of the available services to demonstrate its functionality, the image quality framework shall be documented with all detail in the *Image Quality and Accuracy Engineering Report*. The report shall support at least the following aspects:

- Creation of use cases to show how different missions could use A3C metrics to assess the applicability of imagery to their goals
- Analysis of use cases to derive the set of metrics that can best support A3C goals of the use cases
- Definition of a data model which allows a source system to capture A3C metrics
- Specifications of data elements within the model to insure consistent, appropriate interpretation across a variety of sensors based on spatial resolution, spectral resolution, radiometric resolution and temporal resolution.
- Demonstration of use of the model to show how A3C metrics can distinguish value across several different use cases (can be same missions as defined above), for example:
  - Mapping Mission
  - Disaster Response
  - Monitoring Mission
- Proposed A3C Specification

5.3.2 Example Use Cases

As an example of the need for A3C metrics, consider two use cases: A Land Use analysis and a Mapping mission. In both cases the users may desire mosaic imagery, but there will be differences in the metrics for the intended purpose:

The Land Use analysis requires:

- Similar currency of the imagery, consistency across the mosaic
- Atmospherically corrected, consistent known color balancing
- Any color enhancements

The Mapping Mission is more concerned with:

- Horizontal accuracy
- Extent of collection dates
- DEMs and GCPs used
- Any seam line alteration

5.3.3 Proposed Example Metrics

- **Accuracy**
  - Accuracy of products:
    - Horizontal Accuracy CE90
    - Vertical Accuracy LE90
  - Accuracy of Raw Satellite Imagery (basically reflects satellite pointing accuracy)
    - Horizontal Accuracy CE90 based on observed locations on ground
- **Currency of products**
  - Raw imagery
    - Times of collection, seasonality, leaf-on or leaf-off
  - Finished products
    - Earliest collection date, Latest collection date, Average age, seasonality, leaf-on or leaf-off
- **Completeness**
  - Single scene/strip imagery
    - Metadata on sensor, orbit, spatial resolution, spectral resolution, radiometric resolution, collection geometry, resampling technique, data compression, DEM used, GCP's used, geometric alteration

(e.g. photoshop), color enhancements, image enhancements (e.g. sharpening)

- Finished Products
  * Metadata on sensors, orbits, spatial resolution(s), spectral resolution(s), radiometric resolution(s), collection geometries, resampling technique, data compression, DEM used, GCP's used, Seamlines, geometric alteration (e.g. photoshop), color enhancements, image enhancements

- **Consistency**
  - Atmospherically corrected or not
  - Atmospheric correction based on in scene data or ancillary data
  - Cross sensor calibration (constellation sensors)
  - Ground based spectral calibration
  - Color balancing technique (local color balancing or global color balancing)
  - NIIRS for imagery quality from sensor
  - Standard deviation of collection dates

5.4 Summary

The following tables summarizes all work items that shall be delivered as part of this work package. The client application is supposed to interact with the services and to display A3C information. The services shall support A3C information as described, the engineering report shall capture all results and may be complemented by change requests against existing services.

ID	Funding	Name
DG001	unfunded	WCS Server for Data Quality and Spherical Accuracy
DG002	funded	Generic Client to interact with the services
DG003	funded	Imagery Quality and Accuracy ER
DG004	funded	WMTS Server for Temporal Accuracy

TABLE 5.1: Geospatial Imagery Quality Framework work package deliverables summary

Chapter 6

Coverage Access and Visualization

6.1 Background

The National Aeronautics and Space Administration (NASA) has identified initial interoperability requirements in the context of coverage data access and enhanced visualization of earth observation data. The goal is to enhance the currently available WCS Earth Observation profile to reflect new requirements coming from both the data provider and consumer communities. These involve new indexing mechanisms, support for data encodings such as HDF, NetCDF, and DAP, and coverage processing mechanisms. Further on, NASA is interested in improving OGC service and encoding specifications to optimize its usage with GIBS. GIBS, the Global Imagery Browse Services system *is a core EOSDIS component which provides a scalable, responsive, highly available, and community standards based set of imagery services. These services are designed with the goal of advancing user interactions with EOSDIS' inter-disciplinary data through enhanced visual representation and discovery. [...] The GIBS imagery archive includes approximately 90 imagery products representing 35 visualized science parameters from the NASA Earth Observing System. Each imagery product is generated at the native resolution of the source data to provide "full resolution" visualizations of a science parameter* (NASA).

Figure 6.1 illustrates GIBS' architecture. GIBS is composed of two major functional areas:

- Ingest & Archival - GIBS servers regularly ingest and archive imagery from imagery providers into full-resolution, mosaicked layers; these layers are then chopped into imagery tiles stored at predefined zoom levels for access.

- Distribution - GIBS servers serve the archived imagery tiles created above through standard imagery access services.

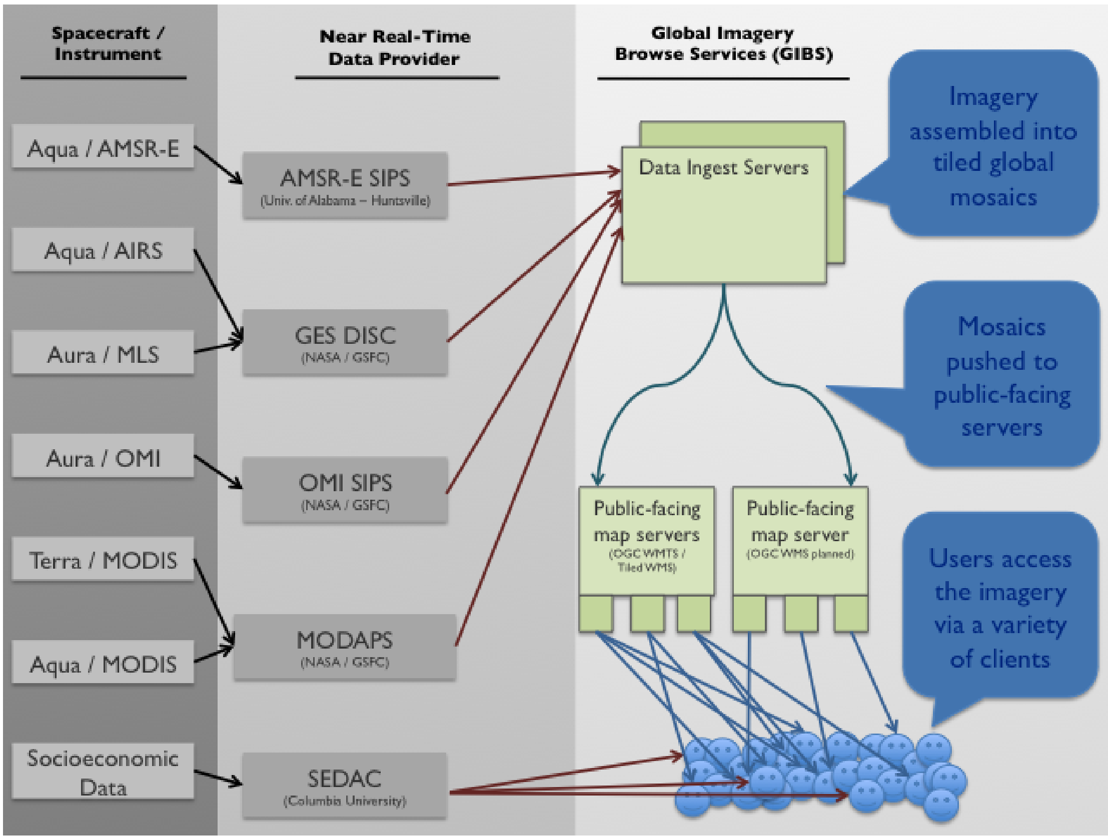

FIGURE 6.1: Global Imagery Browse Services (GIBS) system architecture (image by NASA)

The goal in Testbed-12 is to improve the second of these two aspects, the distribution of imagery data through standard interfaces. According to NASA, GIBS imagery is accessed through the following standards-based web services and formats:

- **Web Map Tile Service (WMTS)** - The WMTS implementation standard provides a standards-based solution for serving digital maps using predefined image tiles. Through the constructs of the specification, a WMTS service advertises imagery layers (e.g. imagery product) and defines the coordinate reference system, scale, and tiling grid available for access. The WMTS standard complements the existing Web Map Service (WMS) OGC standard by providing a less flexible but higher performing image request mechanism.

- **Tiled Web Mapping Service (TWMS)** - The TWMS specification is a custom extension to the OGS WMS standard developed by the NASA Jet Propulsion Laboratory. Similar to the OGC WMTS specification, TWMS introduces a "tiled" approach to imagery requests so that tiles may be pregenerated and cached for fast response. Unlike WMTS, the TWMS standard retains the usage of requests containing geographic coordinates for imagery. However, it only responds to a limited number of predefined geographic regions, creating a gridded access pattern.
- **Keyhole Markup Language (KML)** - The KML documentation standard provides a solution for imagery integration into mapping tools that utilize support the KML standard, specifically Google Earth. Using the constructs of the KML standard, GIBS infuses links to the TWMS web service endpoints to facilitate imagery viewing within supporting tools. A custom KML generation endpoint is provided by GIBS to dynamically generate KML documents.
- **Geospatial Data Abstraction Library (GDAL)** - GDAL is an open source translator library for raster geospatial data formats that presents a single abstract data model to the calling application for all supported formats. By providing integration into the GDAL command line utilities, GIBS imagery can be easily included in imagery processing work flows, including bulk access.

Ideally, work items in this work package consider additional computing optimization and corresponding interoperability aspects through the use of Discrete Global Grid Systems (DGGS). A DGGS is a spatial reference system that uses a hierarchical tessellation of cells to partition and address the globe. DGGS are characterized by the properties of their cell structure, geo-encoding, quantization strategy and associated mathematical functions.

The DGGS standards working group in OGC has recently submitted the **OGC Discrete Global Grid System (DGGS) Core Standard**, OGC 15-104. The OGC DGGS standard supports the specification of standardized DGGS infrastructures that enable the integrated analysis of very large, multi-source, multi-resolution, multi-dimensional, distributed geospatial data. Interoperability between OGC DGGS implementations is anticipated through extension interface encodings of OGC Web Services (OWS). Considering DGGS complements data fusion and conflation aspects as described in chapter 8.15, as described by the following excerpt from OGC 15-104:

As each cell in a DGGS is fixed in location, and the location provides an explicit area representation, basic geospatial enquiry – "Where is it?", "What is here?", and "How has it changed?" - are simplified into set theory operations. As any data values referenced to a particular DGGS are, by the nature of the grid, aligned, the high costs of integrating data in traditional systems are dramatically reduced.

A DGGS can even be designed for lossless encoding of vector geometry such that cells, and their integer addressing, converge monotonically to the Real number coordinate pairs of each observation with each successive refinement – an essential property of a conventional coordinate system.

DGGS are designed to eliminate requirements for complex data fusion processes. Reducing the reliance on an intermediary integrator or analyst is a key requirement for distributed participatory digital-Earth information system. "[Digital-Earth] can clearly benefit from developments in discrete global grid, which can provide the georeferencing, the indexing, and the discretization needed for geospatial data sets. They have properties, in particular hierarchical structure, uniqueness, explicit representation of spatial resolution, and consistency, that make them superior to any single alternative." Goodchild. A DGGS provides a uniform environment to integrate and visualize both vector geometry and raster-based geospatial data sources in much the same way that information within a computer graphics pipeline becomes the pixels on a computer screen. Efficiencies are gained through implementing the Dimensionally Extended nine-Intersection Model (DE 9IM) set of fundamental spatial operations [3-6] directly on the DGGS cell structure. This allows for higher order algebraic algorithms (via bindings to external analytic libraries) to be created on the DGGS structure itself, independent of the data sources. CyberGIS benefit from DGGS use. By conversion of traditional Earth observation data archives into standardized DGGS, massive amounts of data are made available to scientists as timely decision-support products in a transparent and repeatable fashion. Very large multi-resolution and multi-domain datasets are aligned and ready for distributed and/or high-performance parallel-processing computer environments.

6.2 Requirements and Work Items

The following figure 6.2 illustrates all work items that shall be addressed in this work package. They are described in further detail in the following sections. All funded work items are shaded in green, unfunded in blue. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements.

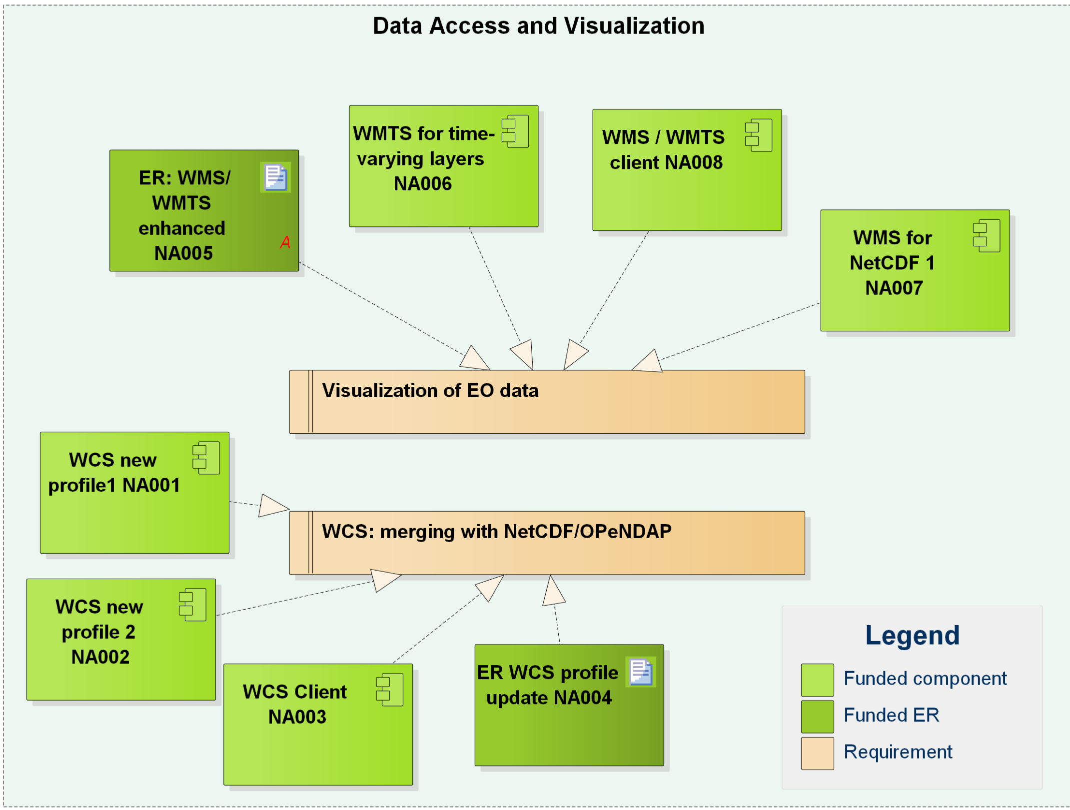

FIGURE 6.2: Coverage Access and Visualization work package: Requirements and work items

6.3 Visualization of Earth Observation (EO) Data

Many EO data sets have been stored in HDF4, HDF5 or NetCDF format, which traditionally have been difficult to analyze and visualize with geospatial tools. With the rising demand from the diverse end-user communities for geospatial tools to handle multidimensional products, many geospatial applications now have new functionalities that enable the end user to store, access, analyze, and visualize these EO data sets. One such visualization approach of NetCDF files uses the ncWMS. ncWMS extends WMS slightly to support features that are useful for scientific data, such as the selection of colour scales. Though ncWMS is fully backward-compatible with the WMS spec, extensions to the WMS specification shall be supported in Testbed-12 to better support NetCDF-CF data visualization and exploration. Testbed-12 shall create a WMS that requires minimal configuration to serve CF-compliant NetCDF files: the source data files should already contain most of the necessary metadata. The service shall implement the new extensions to WMS in order to support the non-standard WMS-behavior provided by ncWMS. In addition, the number of layers served by GIBS is foreseen to get into the hundreds if not thousands and will need a way for clients to intelligently filter them to find the most relevant ones. Testbed-12 shall develop a solution for this issue, e.g. by providing the ability to filter results of *GetCapabilities* requests, possibly using XPath.

Further on, visualization of EO data using WMTS needs to consider tile assembly with temporal considerations. This is an ongoing work item in OGC and there has been some discussion on this topic in the OGC Web Mapping Service 1.4 SWG. The WMTS/WMS SWG is considering facilitating dynamic tile assembly through a WMTS server and if so, whether the servers will likely see much value in request/result caching. Testbed-12 shall investigate if the same result could be achieved (vertical assembly, reduced bandwidth, etc.) through a WMS request that leverages pre-generated tiles. The work shall be based on results from the ESDIS Visualization Working Group (ESDSWG-Visualization), which evaluated leverage ncWMS services for publishing level2 (swath) and level 3 (gridded) satellite imagery for harvest by the Global Image Browse Service (GIBS). Participants interested in this work item are requested to consult with the OGC WMS SWG to discuss the latest details. The goal is to extend the WMTS specification to handle time-varying layers more robustly.

Additionally, if granule and swath imagery is served via WMTS, any given granule or swath often covers only a small fraction of the globe. This means that there can be many unnecessary requests for empty tiles in the areas where there is no imagery. Testbed-12 shall develop a solution that help clients making their requests, possibly by providing a bitmap of where there is valid imagery on the globe so a client knows which requests to make.

As those work items use current work performed in the OGC WMS SWG and within GIBS, Testbed-12 shall use all input available from the OGC WMS SWG and from NASA GIBS at the time of the Testbed-12 kick-off and shall work closely with these groups throughout the testbed.

References

- OGC 13-082r2: OGC Web Map Tile Service (WMTS) Simple Profile
- OGC 14-028r1: Performance of OGC Services in the Cloud

Deliverables

- The **WMS/WMTS Enhanced engineering report, NA005**, shall capture all discussions and results of the work items described in this section.
- The **WMTS for time-varying layers, NA006**, shall implement all requirements discussed in this section.
- The **WMS for NetCDF, NA007**, shall implement the additional features described above on top of a current WMS.
- The **WMS/WMTS client, NA008**, shall support the enhanced versions of WMS and WMTS as provided as NA005 and NA006. It can be implemented either desktop or browser-based.

6.4 WCS: Merging with NetCDF/OPeNDAP

The WCS Core standard allows for access to nearly unlimited varieties of coverages, i.e., WCS Core is not limited to quadrilateral grids. The current EO WCS profile as originally developed and tested in OWS-8 constrains WCS beyond what is needed to meet full Earth Observation needs. The activity in this section would result in either an expansion of existing EO WCS profile or a new profile in part by considering the DAP protocol (in particular DAP v4.0). WCS and DAP protocols provide access to Earth Observation data typically represented in OGC as Coverages. DAP4 is the current version as last revised on 04 July 2014. OPeNDAP is the most widely used implementation of the DAP Protocol. A useful comparison of WCS and DAP/OPeNDAP was provided in a 2012 paper by Baart et al, published in Transactions in GIS. Figure 6.3 from Baart provides a comparison summary.

Previously OPeNDAP was tested in OGC Testbed 9 with results presented to the OGC TC meeting in Austin, TX in March 2012. The DAP profile was posted

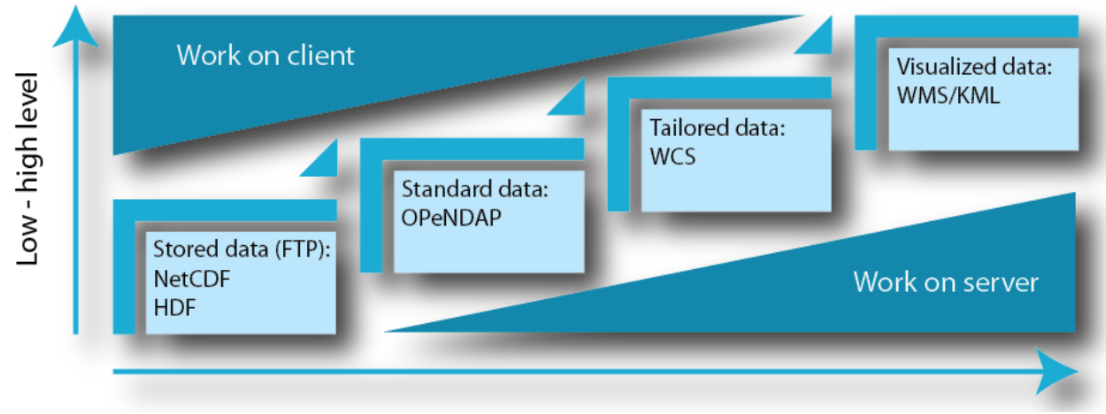

FIGURE 6.3: Overview of different levels for providing results from numerical models (*from Baart et al, 2012*)

as a pending document for the March meeting. It was discussed in the Architecture DWG and the Architecture Board meetings. The relevant documents are:

- OGC 12-095: OWS-9 Innovation - Coverages: Coverage Access (OPeNDAP) Study
- OGC 12-009: The Data Access Protocol — DAP 2.0

Recently NASA ESDIS commissioned a study that compared OPeNDAP and WCS. The study states that *the WCS and OPeNDAP parallels are remarkable and close, especially at the mathematical level. Most aspects of WCS have been considered in developing the latest OPeNDAP protocol (DAP4), though parts of Core WCS 2.0 (such as for polygonally-meshed domains) are handled only via DAP4 extensions. The same is not true with WCS application profiles. EOWCS, like OPeNDAP, offers netCDF encoding, but in other respects its capabilities are sometimes greater (as for CRS-based interpolation/re-projection) and sometimes less (as for higher dimensions and polygonally-meshed domains) than those presently available with OPeNDAP. In fact, WCS profiles and extensions embody key distinctions between WCS and OPeNDAP, thus representing areas where interoperability may be most challenging. In particular, some aspects of the EOWCS profile may enhance WCS-OPeNDAP consistency, whereas the profile's treatment (or exclusion) of higher dimensions, non-geographic coordinates and aggregation may impede interoperability.* The study developed recommendations on WCS, Interoperability and OPeNDAP that shall be addressed in Testbed-12:

Recommendation 1: OGC should consider making mandatory at least one form of response encoding for WCS. From the foregoing analysis, it appears that netCDF should be among the mandatory forms. Furthermore, the presence of netCDF libraries in multiple languages lowers the burden of this change on developers of WCS-compliant servers and clients, such as ESRI (author of ArcGIS packages that already exhibit netCDF compatibility). This recommendation is consistent with conclusions drawn in the aforementioned paper by Baart et al. In parallel, NASA [...] should help OGC adopt a new or improved WCS Application Profile. In particular, this profile should fully embrace the concept of Coverage as a function of time and other coordinates alongside the geospatial ones. Ideally, this profile should aim to for alignment with DAP4 specifications (or extensions).

Recommendation 2: OGC should consider placing greater emphasis on multilingual, open-source libraries as tools for promulgating use of its standards. Such tools seem especially important for standards with complex semantics, such as WCS, with its extensions and profiles. We suggest a specific step to this end, namely, formal OGC adoption of DAP4 – modified or extended, if necessary – as a standard (or standards). The form of such standardization remains to be defined, but it might include DAP4 as a Coverage encoding as well as a WCS interface protocol and implementation. Because such standardization will require significant effort, due in part to differences in the styles and formalities of OGC and OPeNDAP specifications, we recommend that NASA commit to the provision of the needed resources.

Based on the previously listed recommendations and the current baseline for WCS, Testbed 12 will conduct the following activities:

- WCS-EO profile update. Update or create a new profile that is better aligned with the specific needs of the EO data user community following recommendations of Baart et al. and the NASA study cited above.
- EO Data encoding for WCS. Specify and test access to EO Data using WCS in order to identify a recommended mandatory encoding for WCS access to EO data. The form of such standardization remains to be defined, but it might include DAP4 as a Coverage encoding as well as a WCS interface protocol and implementation.
- N-D Arrays. Building on the EO-WCS Application Profile, specify and test WCS access to EO data as a function of 3D space and 1D time along with operations for subsetting the range of the coverage, e.g., query data

using an index-based approach along arbitrary domain and range dimensions. Consider OGC Web Coverage Processing Service (WCPS) Language and DAP4 in this task.

- Aim for for alignment with DAP4 specifications.
- Developing of multi-lingual, open-source libraries as tools for promulgating use of its standards
- Consideration of formal adoption of DAPv4.0 with modifications/extensions as necessary
- Specify and test WCS access to EO data as a function of 3D space and 1D time along with operations for subsetting the range of the coverage, e.g., query data using an index-based approach along arbitrary domain and range dimensions. Consider OGC Web Coverage Processing Service (WCPS) Language and DAP4 in this task.

References

- OGC 10-140r1: OGC Web Coverage Service 2.0 Interface Standard - Earth Observation Application Profile
- OGC 08-059r4: OGC Web Coverage Service WCS - Interface Standard - Processing Extension
- OGC 12-039: OGC® Web Coverage Service Interface Standard - Scaling Extension
- OGC 12-040: OGC® Web Coverage Service Interface Standard - Range Subsetting Extension
- OGC 12-049: OGC® Web Coverage Service Interface Standard - Interpolation Extension
- OGC 13-057r1: OGC® Web Coverage Service Interface Standard - Transaction Extension
- OGC 14-110r1: OGC® Web Coverage Service Interface Standard - JP2KJPIP Extension
- OGC 14-121: OGC® Web Coverage Service Interface Standard – Xpath
- OGC 15-045r1: A MetOcean metadata profile for WCS2.0
- DAP v2.0: The Data Access Protocol, DAP 2.0

- DAP v4.0: DAP Specification
- DAP v4.0: Essential Features
- DAP v4.0: Design
- Baart et al (2015): A comparison between WCS and OPeNDAP for making model results available through the internet
- OGC 12-095: OWS-9 Innovation - Coverages: Coverage Access (OPeNDAP) Study
- OGC 12-009: The Data Access Protocol, DAP 2.0
- OGC 14-100r2: CF-netCDF 3.0 encoding using GML 3.2.1 Coverage Application Schema
- Fulker, D. and A. Schuster (2015): WCS, Interoperability and OPeNDAP. White paper. *Availability under negotiation*.

Deliverables

- The **WCS server with new profile 1, NA001**, shall capture all discussions and results of the work items described in this section.
- The **WCS server with new profile 2, NA002**, shall capture all discussions and results of the work items described in this section.
- The **WCS Client, NA003**, shall interact with the WCS implementations resulting from NA001 and NA002.
- The **WCS profile update ER, NA004**, shall support the enhanced versions of WMS and WMTS as provided as NA005 and NA006. It can be implemented either desktop or browser-based.

6.5 Summary

The following tables summarizes all work items that shall be delivered as part of this work package.

ID	Funding	Name
NA001	funded	WCS new profile 1
NA002	funded	WCS new profile 2
NA003	funded	WCS Client
NA004	funded	WCS profile update ER
NA005	funded	WMS/WMTS enhanced ER
NA006	funded	WMTS for time-varying layers
NA007	funded	WMS for NetCDF
NA008	funded	WMS/WMTS client

TABLE 6.1: Data Access and Visualization work package deliverables summary

Chapter 7

GFM, Catalogs, and Semantics

7.1 Background

The GFM, Catalogs, and Semantics work package addresses a number of aspects that are critical to meet new challenging requirements in the context of general data handling, semantics, catalogs, and clients.

The General Feature Model (GFM) was originally developed to support the Cartographic discipline. OGC being the Open GIS Consortium at the time, the documentation has a strong cartographic flavor. Today, the GFM is the foundation for OGC Web services and data models. It is widely implemented in SQL-based relational database systems. A similar capability is needed for Big-Data platforms. The goal is to review the General Feature Model and to research necessary modifications to broaden its scope. It shall allow to be re-used for non-geospatial centric applications and extended as necessary into a general model for all object types.

Other work items in this work package address new prototype OGC Draft Standards or recommendations. These implementations require additional research, development, testing and evaluation. Implementations shall inform and support the development of future OGC standards based solutions. These efforts will provide guidance and direction on future implementation to operational capabilities.

7.2 Requirements and Work Items

The following figure illustrates all work items that shall be addressed in this work package. They are described in further detail in figure 7.1. All funded work items are shaded in green. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements. Some of the work items address requirements that are further detailed in chapter 8.

7.3 General Feature Model

A continuously growing volume of data in being stored in Big-Data Clouds. These are very large collections of multi-disciplinary heterogeneous data. The Big-Data Clouds' goal is to perform complex analytics across these heterogeneous collections. The analytics require a means to extract equivalent information from dissimilar content. The more we can normalize that information, the easier it will be to correlate it. It has long been argued that almost all data has a spatial-temporal component. Therefore, a spatial-temporal model common to all data types would be a great benefit.

The General Feature Model is formally defined in ISO 19101 and ISO 19109. Today, the GFM is the foundation for OGC Web services and data models. It is widely implemented in SQL-based relational database systems. A similar capability is needed for Big-Data platforms.

OGC Simple Features: The OGC Simple Features standards were the first standards created by the OGC. They provide a common set of geometries and operators for dealing with spatial-temporal data. This implementation of the General Feature Model is the set of baseline requirements which must be supported.

BigTable: Google published their "BigTable" paper in 2004. The paper describes a storage architecture based on a sparse, distributed multi-dimensional sorted map. The Open Source community has implemented this architecture in Hadoop, Accumulo, and other Big Data platforms. Key characteristics of BigTable systems are:

1. It is a key-value store where the keys are built up from a row number, column name, and timestamp.

2. The keys are sorted, allowing rapid look-up.

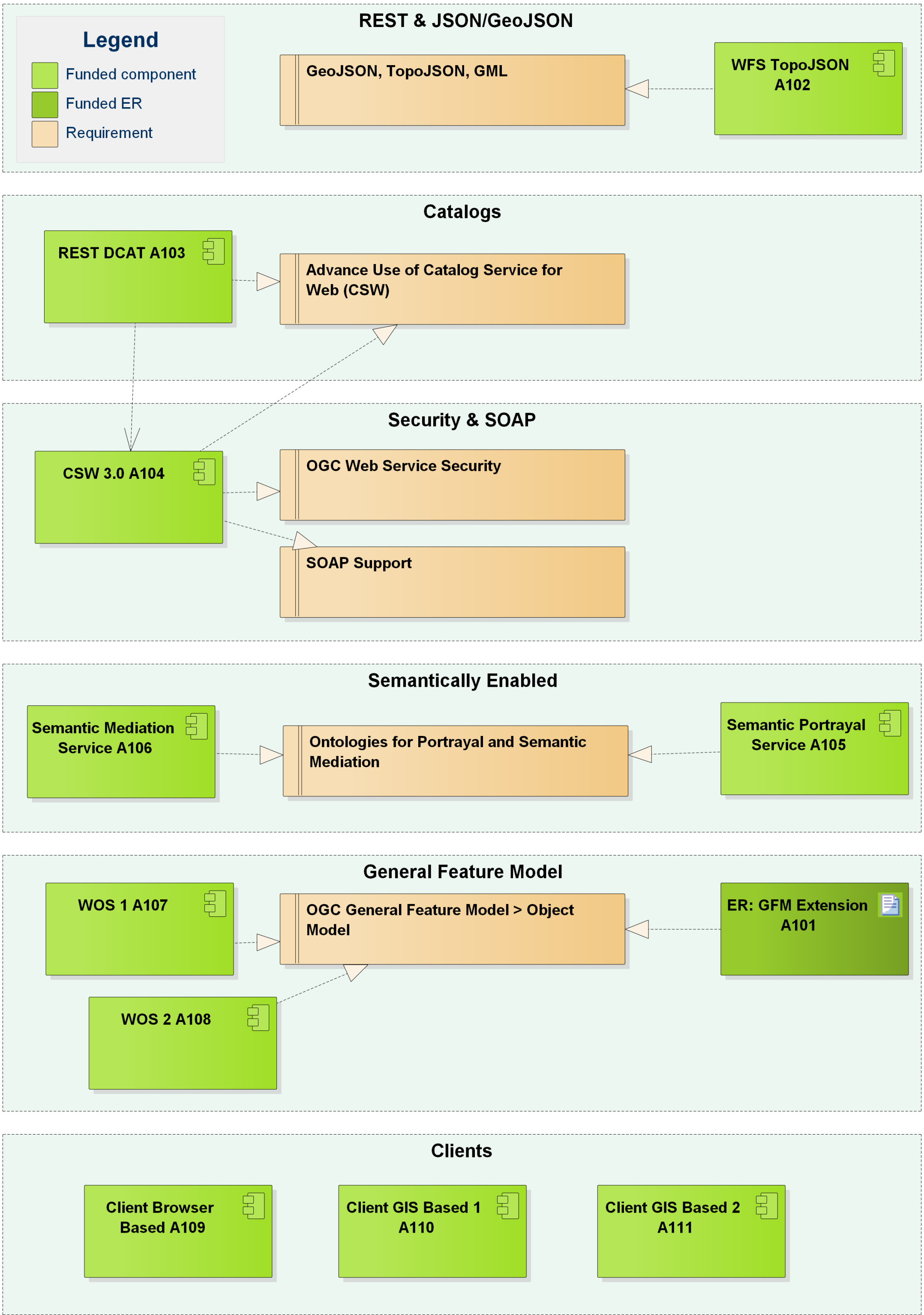

FIGURE 7.1: Data Access and Visualization work package: Requirements and work items

3. The key-value pairs are distributed over a large number of servers. This facilitates massively parallel processing. Since the data has been parallelized, the processing can come to the data.

4. It has no concept of spatial-temporal data or operations.

GeoWave: The GeoWave project is a spatial-temporal extension to Google's BigTable architecture. It is commonly implemented on Apache Hadoop and Accumulo although other BigTable platforms can be supported. GeoWave brings OGC Simple Features geometries (encoded as Well Known Text) and operations to these platforms. It also provides rapid and scaleable indexing through an n-dimensional Space Filling curve (see GeoWave and BigTable presentation).

TSPI: TSPI, the Time-Space-Position-Information Standard (current version 2.0), "*provides a single means of encoding spatiotemporal information for the storage, manipulation, interchange, and exploitation of spatiotemporal data. [...] The TSPI Version 2.0 standard is a robust mechanism for expressing "Where" and "When" spatiotemporal data in various core and extended XML-based information schemata throughout the US Federal Sector. Spatial information includes position, extent (shape), size, orientation, and rates of change in these characteristics, while temporal information includes position (instant), extent (duration), and periodic recurrence characteristics. The TSPI standard should be employed any time a standard eXtensible Markup Language (XML) encoding of spatiotemporal geospatial data is required. It is designed as a set of reusable data components upon which both Geography Markup Language (GML)-based application schemas and non-GML-based XML schemas may be developed. It specifies a registry-based extension mechanism enabling the development and reuse of additional spatiotemporal XML Schema components. It integrates the XML Schema for the United States Thoroughfare, Landmark, and Postal Address Data Standard, FGDC-STD-016-2011, including the necessary updates to work with the OGC GML 3.3, Extended schemas and encoding rules.*" (FGDC).

Tasking: Testbed-12 postulates the Spatial-Temporal Platform as a Service (ST-PaaS). The collection of data types, operations, and architecture patterns necessary to spatial-temporal enable any content. The task for Testbed-12 is to define the ST-PaaS:

1. Starting with the General Feature Model and WFS, what changes are necessary to make them applicable to any content type (i.e. replace Feature with Object)? Are they sufficient to support all spatial-temporal capabilities, which a content community may want to implement?

2. OGC Filter Encoding and OGC Simple Features – these define the fundamental geometries and operators, which characterize spatial-data content. How can these be applied to arbitrary data formats? How can this value be brought to other domains?

As a starting point, Testbed-12 can go back to OOA/OOD basics and consider objects, attributes, operations, associations (aggregation, composition, association), as well as the association Feature (object) from the GFM. These primitives should be sufficient to model most content types. If it is possible to express spatial-temporal as components of this general model, then it will be easier to see how they can be re-used for non-geospatial centric applications.

Testbed-12 shall use a collection of data and data formats that may benefit from a consistent representation of space and time. Existing work, such as TSPI, GeoWAVE, and others shall be taken into account to develop the OGC Core, which defines the fundamental spatial and temporal concepts and operations needed for heterogeneous spatial-temporal processing.

Deliverables

The following list identifies the following work items assignment to requirements. All work items are listed in the summary table 7.1 at the end of the chapter.

- WOS 1 (Web Object Service) as a potential modification of WFS to handle all types of objects, not only features (A107)
- WOS 2 (Web Object Service) as a potential modification of WFS to handle all types of objects, not only features (A108)
- Client to interact with the WOS service implementations (A109, A110)
- ER shall describe how the General Feature Model can be re-used for non-geospatial centric applications, how it can be extended into a general model for all object types. (A101)

7.4 Prototype Implementations

This workpackages identifies a number of prototype implementations that shall be implemented in support of requirements identified in chapter 8.

7.4.1 WFS 2.5 TopoJSON

The TopoJSON WFS shall support the evaluation of

1. TopoJSON as a model to create a JSON encoding that is different (not just an extension) but can be mapped and automatically converted into a GeoJSON file, see chapter 8 section 8.4.
2. work in previous testbeds about a WPS profiles for topological applications with the TopoJSON to study the applicability and interoperability of TopoJSON in OGC standards such as WPS and WFS.

Deliverables

The following list identifies the following work items assignment to requirements. All work items are listed in the summary table 7.1 at the end of the chapter.

- Prototype WFS 2.5 (A102)

7.4.2 DCAT and Catalog CSW v3.0

There are three general requirements on the DCAT as a service implementation.

First, the catalog services DCAT and CSW version 3.0 complement work items described in chapter 8 section 8.11 (Catalogs CSW 2.0.2 and CSW ebRIM) with further reference to sections 8.12 (enhanced capabilities) and 8.6 (asynchronous interaction patterns). For DCAT and CSW 3.0, no approved standard is in place. It shall be evaluated how DCAT can describe the same service and data sets in RDF as the other catalog services do using XML Schema instance documents compliant to ISO 19115. The CSW v3.0 server shall support OpenSearch, provide a SOAP binding, consider the recommendations on pycsw referenced and discussed in Testbed-11, and implement the NSG Metadata framework. For an overview see figure 7.2. In addition, both servers shall if possible support enhanced capabilities and asynchronous interactions as described in sections 8.12 and 8.6.

Second, the DCAT implementation shall serve as a Semantic Portrayal Catalog. The Semantic Portrayal Catalog uses an ontology model for managing styles and provides interfaces to access, create, read, update, and delete styles. For an overview see figure 7.3.

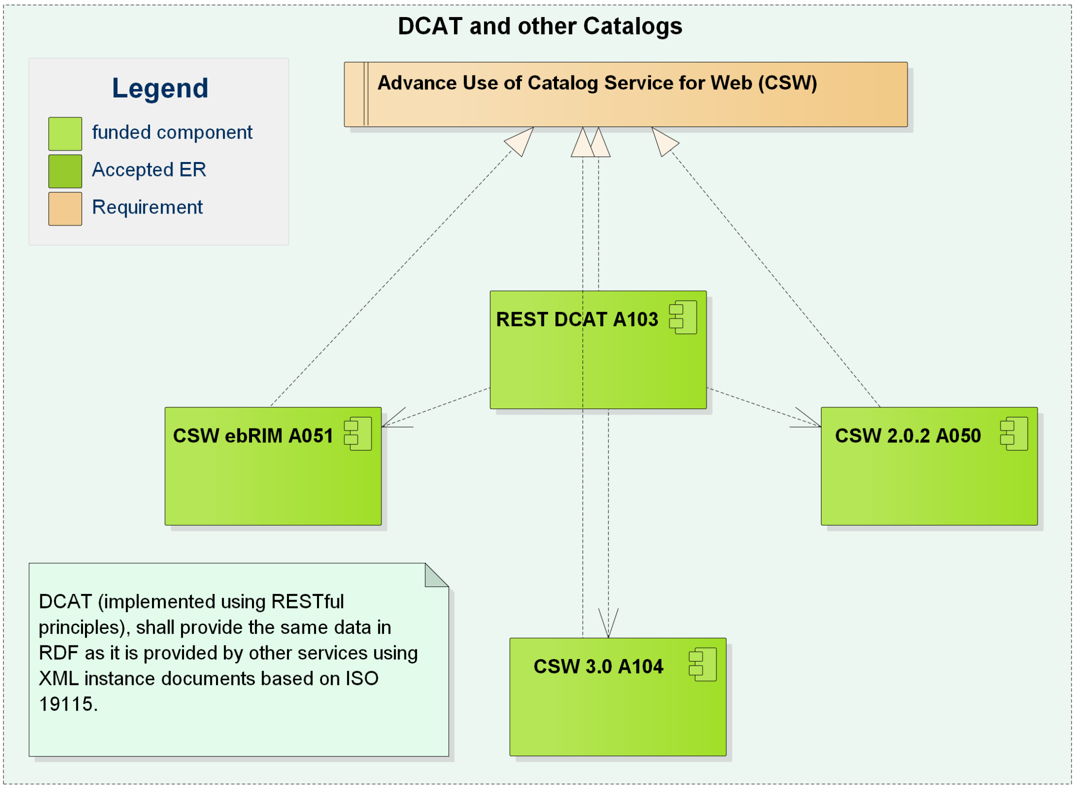

FIGURE 7.2: DCAT work item details: Requirements and work items, part 1

Third, the DCAT as a service shall interface with the Schema Registry described in chapter 8 section 8.5. The Schema Registry enables the discovery of XML Schemas, tranformation logic, and ontologies. These items shall be served by the DCAT as a service implementation.

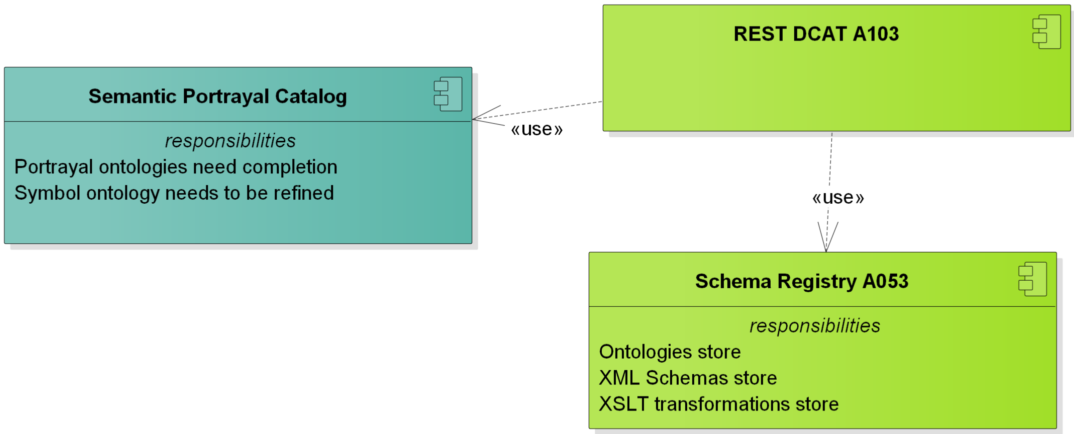

FIGURE 7.3: DCAT work item details: Requirements and work items, part 2

Deliverables

The following list identifies the following work items assignment to requirements. All work items are listed in the summary table 7.1 at the end of the chapter.

- DCAT REST service (A103)
- CSW v3.0 with OpenSearch, SOAP (104)

7.4.3 Semantic Portrayal Service

The Semantic Portrayal Service (implemented as a RESTful API) completes work from Testbed-11 by providing endpoints to access and create, update and delete styles, rules, graphics information. The portrayal ontologies need to be completed by formalizing further the graphics ontology by defining graphic objects and attributes for lines and areas. The Symbology ontology needs to be refined further to accommodate line and area-based symbols and well as composition of multiple symbols and their bindings with the geometric properties of features. Further on, the Semantic Portrayal Service shall be extended by providing a rendering endpoint to convert a Linked Data Model to a symbolic representation in well-known formats such as SVG or KML.

Deliverables

The following list identifies the following work items assignment to requirements. All work items are listed in the summary table 7.1 at the end of the chapter.

- Semantic Portrayal Service (A105)

7.4.4 Semantic Mediation Service

The Semantic Mediation Service supports query rewriting and supports JSON-LD and RDF serializations for interaction. The SPARQL Extension ontology needs to be further refined and documented and be exercised on a variety of use cases to reach a level of maturity and robustness needed to become a standard. More implementations leveraging this ontology should be pursued to validate the feasibility of using this standard to extend SPARQL endpoint capabilities.

The REST API for Semantic Mediation Service needs to be tested further and the serialization in JSON-LD needs to be improved to lower the bar of integration with web clients. Other use cases for the use the Semantic Mediation Service need to be investigated, such as query rewriting service (a SPARQL query for one source ontology to be converted to one or more SPARQL queries for the target ontology). Furthermore, the Semantic Mediation Service shall

1. leverage the Schema registry to translate messages and content from one schema to another.
2. support transformation chains: where the rules for a direct transformation are not defined but a sequence of transformations can fill the need.
3. leverage ontologies to not only perform syntactic transformation, but semantic as well
4. Validate the accuracy and correctness of the transformation

Deliverables

The following list identifies the following work items assignment to requirements. All work items are listed in the summary table 7.1 at the end of the chapter.

- Semantic Mediation Service (A106)

7.5 Clients

Client Applications: Testbed-11 identified a number of client application shortfalls especially in the area of support for standard mainstream IT security frameworks. Testbed-12 client implementations shall support SOAP, REST and Conventional OGC service communications. Client implementations may be either browser based, commercial based and/or open source. Not all clients are expected to support every aspect of the requirements below but all clients shall demonstrate interoperability between other clients in support of the Testbed-12 scenario:

- The client shall interface with OGC web services implementing a common method for integrating main stream IT security
- The client shall interface with SOAP, REST and Conventional OGC services
- The client shall interface with prototype OGC service implementations (WOS, WFS 2.5, etc.)
- The client shall interface with the services offering 3D/4D data
- The client shall interface with the different catalog types
- The client shall interface with Semantic Portrayal, Semantic Mediation services
- The client shall interface with the Conflation service (Hootenanny façade)
- The client shall support Pub/Sub for Asynchronous actions.
- The client shall interface with GeoPackage Web Processing Service
- The client shall support read/write OWS Context JSON
- The client shall interface with the Geopackage with symbology WPS
- The client shall interface with the Geopackage with terrain WPS
- The client shall interface with the Reprojection
- The client shall support the analysis of performance testing of the various compression techniques

- The client shall interface with the services offering LiDAR streams

Deliverables

The following list identifies the following work items assignment to requirements. All work items are listed in the summary table 7.1 at the end of the chapter.

- Client 1, browser based (A109)
- Client 2, GIS based (A110)
- Client 3, GIS based (A111)

7.6 Summary

The following tables summarizes all work items that shall be delivered as part of this work package.

ID	Funding	Name
A101	funded	General Feature Model ER
A102	funded	Prototype WFS 2.5
A103	funded	DCAT REST service
A104	funded	CSW v3.0 with OpenSearch, SOAP
A105	funded	Semantic Portrayal Service
A106	funded	Semantic Mediation Service
A107	funded	WOS 1
A108	funded	WOS 2
A109	funded	Client 1, browser based
A110	funded	Client 2, GIS based
A111	funded	Client 3, GIS based

TABLE 7.1: Data Access and Visualization work package deliverables summary

Chapter 8

OGC Baseline Enhancements

8.1 Background

This work package includes work items that enhance the existing OGC technology baseline. These are established in approved interface or encoding specifications that shall be revised in order to meet altering requirements and service demands. These maintenance cycle activities are complemented with specification enhancements to interface and encoding specifications that are ready for operational implementation based on previous research and evaluation. Most of the work items are based on results from previous testbeds, in particular Testbed-11. Those efforts pertained to OGC standards support for REST based services, services that output data using the (Geo)JSON encoding, services that implement a Semantic Level of Interoperability and services that could implement a Common Layer of Security. Support for JSON, REST, Linked Data and Semantic Web technologies will be an integral piece to the Testbed-12 scenario utilizing multiple data types from multiple sources. Support for a multiple sets of operational service implementations will be required. Some of these services shall implement the RESTful architecture and some shall implement a non-RESTful OGC Web service architecture. Some service operational implementations shall implement the recommendation guidelines for a Common Security Layer from Testbed-11 in order to demonstrate interoperability between secured and non-secured services. Testbed-12 participants are strongly advised to review all Testbed-11 results.

Testbed-11 evaluated 5 different but interrelated architectural designs, meaning their implementation crossed over several individual standards and services. These evaluations were documented in Testbed-11 Engineering Reports. Testbed-12 will consider and operationally implement those recommendations as defined below. The 5 activities were: a Common Security Layer across all OGC Web Services, a Common Implementation of REST and SOAP across all OGC Web Services, a Common use of JSON/GeoJSON across all OGC Web Services and a Common set of guidelines for Implementing Semantically Enabled OGC Web Services. Testbed-12 shall in its first phase evaluate and resolve those recommendations from Testbed-11 in order to operationally implement service and client instances as part of the second phase.

8.2 Requirements and Work Items

The OGC Baseline Enhancements work package identifies a huge amount of work items. The corresponding overview figures that illustrate all work items that shall be addressed in this work package have been removed due to readability issues. Instead, this chapter will use a number of figures that show parts of the overall work load. All figures use the same style. All funded work items are shaded in green, unfunded in blue. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements.

The work items overview has been split to improve readability of all elements. Each subsection contains an overview diagram that identifies requirements and corresponding work items.

Important: It is emphasized that organizations interested in proposing for one or more components need to read the entire chapter. A single component may implement many deliverables and is therefore named and extended throughout this chapter! There are correlation matrices given at the end of this chapter to better understand the relationship between requirements, and components or engineering reports.

8.3 Security and SOAP

Testbed-11 evaluated the current suite of OGC service standards to determine their level of support for interoperable fine-grained access control. A number of Change Requests are being drafted as a result. Testbed-12 would implement those recommendations.

Testbed-11 identified several disconnects between the current suite of OGC web services and the ability to support a common integrated layer of security at the Web service itself. Testbed-12 would implement the recommendations of Testbed-11.

The common security layer shall be implemented as an extension to the OGC OWS Common or OGC service standards as appropriate to allow for very fine-grained access control mechanisms. Those mechanisms could be performed by the service itself instead of in subsequent processing steps further down the processing chain. Testbed-12 shall evaluate the Security requirements including SOAP to build an OWS Common Security Extension. All work items are illustrated in figure 8.1.

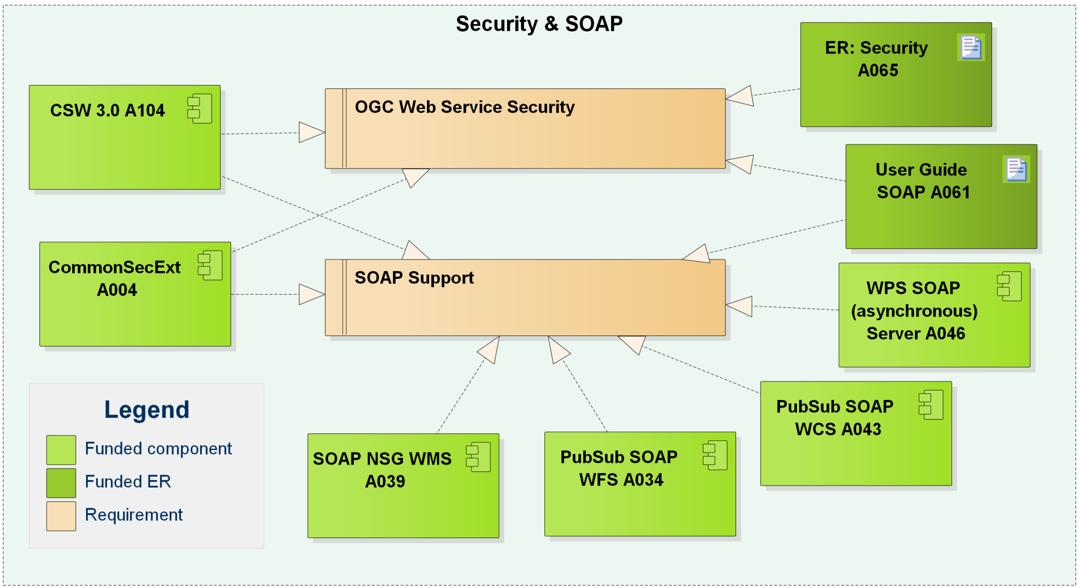

FIGURE 8.1: OGC Baseline Enhancements: Security and SOAP requirements and work items

Deliverables

The following list identifies the *Security and SOAP* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- All elements related to Common Security shall be addressed as part of a OWS Common Security Extension (A004, A065)
- Define a Common Security Architecture for OGC Web Services and Clients (A004, A065)
- Define a OGC Common security model that can be implemented by OGC Web Services (A004, A065)
- Define requirements for a common set of normative mainstream IT references for implementing the ISO frameworks on OGC Web Services (A065)
- Define a Common Security Capabilities extension, WSDL documents including guidance how to embed WS-* and WS-Policy when using SOAP as well as defining a common approach to the OGC Publish / Find / Bind paradigm for protected services. (A004, A065). *WS-* is a prefix used to indicate specifications associated with Web Services and there exist many WS* standards including WS-Addressing, WS-Discovery, WS-Federation, WS-Policy, WS-Security, and WS-Trust.*
- Define options for making the *GetCapabilities* operation publicly accessible or protected and accessible as a Capabilities instance document hosted on a web server that contains the <Operations> section including the security description. (A004, A065)
- Define and describe Common Security in Capabilities document; *ows:Constraint* shall be included for each operation. (A004, A065)
- Define the available names for ows:Constraint for: authentication, access control, Integrity, Confidentiality (A004, A065)
- Define a Codelist for supported Authentication options (A065)
- Define a GeoXACML and XACML policy based codelist for Authorization (A065)
- Relax the option to use protocol scheme HTTPS as a service operation description (A065)
- Define guidance on how to outline support for the HTTP GET and POST method in a capabilities document, referring to the same operation. As an example, GetCapabilities using HTTP GET and for GetFeature only HTTP POST, because the security framework is only available for POST but not for GET. Disallow a service to jump from one protocol verb to

another for the same operation once a session is established to relax the implementation of another security framework. (A065)

Testbed-12 identified a need for SOAP OGC Service bindings, particularly CSW. The SOAP subtask shall be worked in conjunction with the Security subtask. The main driver is the standardized, descriptive (policy-based) security that can be applied to SOAP bindings via WS-Security and related standards. WSDL examples are required to assist developers in building client applications.

Implementation is required to support the Distributed Common Ground System (DCGS) Integration Backbone, championing the adoption of CSW across the 7 DCGS Programs of Record.

The following list identifies the *SOAP* work items assignment to requirements. All work items are listed in section 8.23.1 and 8.23.2 as well as the summary tables 8.2, 8.3 and 8.4 at the end of the chapter.

- Define a common mechanism that any service can use to declare support for SOAP encoding and for each operation by means of the Operations-Metadata section of its ServiceMetadata document. (A004, A061, A065, A104)
- Define a consistent mechanism to transport binary or XML Schema data, e.g. SOAP with attachment, inline base64 encoding, or Message Transmission Optimization Mechanism (MTOM) in combination with XML-binary Optimized Packaging (XOP). (A004, A061, A065, A104)
- Develop mechanism for reporting all errors following the SOAP 1.2 recommendation of embedding a soap:Fault element in the soap:Body of an soap:Envelop. The value of the soap:Code/soap:Value shall be bound to "soap:Sender". Optionally, the type of error message may be indicated as part of the soap:SubCode element. (A034, A039, A043, A046, A061, A065, A104)
- Define how detailed error messages shall be part of the ows:ExceptionReport and follow the rules provided in OWSCommon, OGC06-121r9 and/or make recommendations for new OWSCommon extension. (A034, A039, A043, A046, A061, A065, A104)
- Define a consistent mechanism for WSDL files. (A034, A039, A043, A046, A061, A065, A104)

- Develop mechanism on how to organize support for SOAP in conformance classes. (A061, A065)
- Develop recommendations on the usage of the “action" attribute on the application/soap+xml media type typically provided as part of the HTTP header. (A061, A065)

8.4 REST, JSON, and GeoJSON

Testbed-11 evaluated and made recommendation for following a REST based best practice. Testbed-12 would require operational implementation of those recommendations for final analysis before including them in the OGC standards baseline.

REST has gained widespread acceptance across the Web as a simpler alternative to SOAP and WSDL-based Web services. The key word in that statement is simpler. While REST based services can be simpler for the developer to build and simpler for the end user it is still to be determined when simpler is the appropriate approach depending on the use case and the data being served. Current REST is almost always tied to JSON. Testbed-12 shall evaluate the requirements to implement REST and JSON for OGC Services.

Deliverables

The following list identifies the *REST, JSON, and GeoJSON* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Evaluate and recommend a Common Security Architecture for RESTful implementations (A005-1, A040, A044, A060)
- Define a common set of RESTful operations for; Get, Put, Post, Head, and Delete (A005-1, A040, A044, A060)

The JSON Subtask shall be worked in conjunction with the REST Subtask above. The following list identifies the *JSON/GeoJSON* work items assignment to requirements. All work items are listed in section 8.23.1 and 8.23.2 as well as the summary tables 8.2, 8.3 and 8.4 at the end of the chapter.

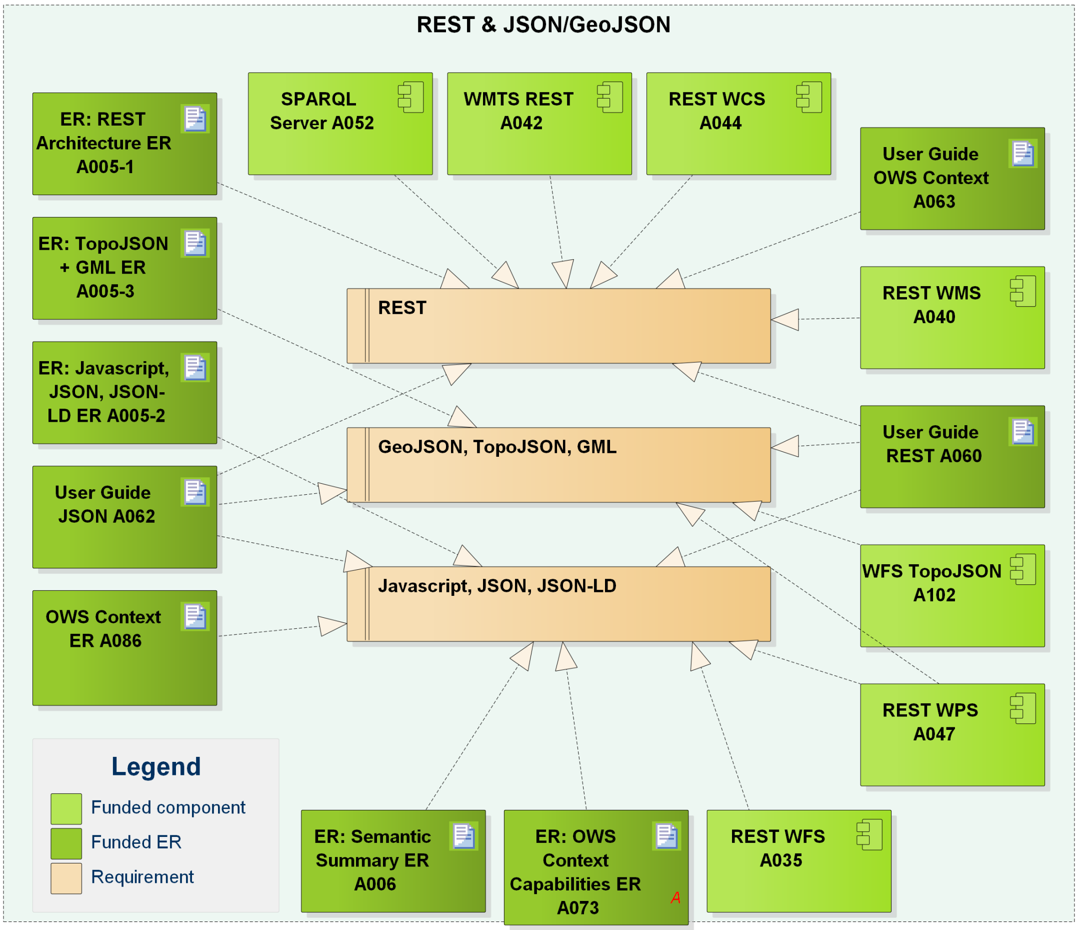

FIGURE 8.2: OGC Baseline Enhancements: REST and JSON/GeoJSON requirements and work items

- Identify and define any security related issues with the JavaScript interpreter to execute JSON text dynamically as embedded JavaScript. (A005-2, A060, A062)
- Extend JSON schema to fully describe the properties of a feature type, including units in alphanumeric properties and CRS in the geometric attributes instead of having to repeat them in each instance. (A005-2, A035, A060, A062)
- Evaluate the use of JSON schema and the @context section of a JSON-LD file (in combination with the ontologies linked to it) as a means for validating a JSON file. (A005-2, A035, A060, A062)
- Evaluate using the namespace URIs in *@context* section of a JSON-LD file as a means to connect to formal ontologies structured in OWL SKOS or other RDF encoding as a way to validate complex types in JSON files in the OGC. (A005-2, A035, A060, A062)
- Evaluate TopoJSON as a model to create a JSON encoding that is different (not just an extension) but can be mapped and automatically converted into a GeoJSON file. (A005-3, A047, A060, A062)
- Evaluate work in previous testbeds about a WPS profile for topological applications with the TopoJSON to study the applicability and interoperability of TopoJSON in OGC standards such as WPS and WFS. (A005-3, A047, A060, A062)
- Evaluate adding "@type" keys to JSON objects as a good practice to make the transition to JSON-LD and RDF easier. It is also good practice that type names are qualified with an abbreviated namespaces (e.g. *ows: ServiceIdentification*) that could be later dereferenced using JSON-LD @context. (A005-2, A035, A060, A062)
- Evaluate JSON-LD as an alternative for creating GML application schemas as a means of defining feature types and as a mean for validation. (A005-3, A035, A060, A062)
- Evaluate how to provide service metadata in JSON derived from the UML models. Include as part of the OWS Common Schemas @context documents for independent validation of the 4 main sections of the Service Metadata. A JSON schema can also be provided. (A005-2, A060, A062)

- Evaluate JSON-LD context as the Contents section of service metadata document. (A005-2, A035, A060, A062, A063)
- Define the *OWS Common* minimum set of parameters in a request as a *@context* fragment. A JSON schema can also be provided. (A005-2, A060, A062, A073)
- Define specific JSON schema documents as a means of defining feature types and as a means for feature instance validation (as the equivalent of GML application schema). (A005-3, A060, A062)
- For a WFS serving GeoJSON, force the features to have a property that contains the feature type. (A005-2, A060, A062)
- For a WFS *DescribeFeatureType* returns a *@context* section describing the Feature type in GeoJSON. (A005-2, A060, A062)
- Evaluate using either JSON Pointer or JSON Path in places where a XPath is required. (A005-2, A060, A062)
- Evaluate the unsolved issue where GeoJSON coordinates prevent a way to apply JSON-LD to GeoJSON and an automatic conversion to RDF. (A005-3, A060, A062)
- Evaluate as a solution to the bullet point above to respect the original format of GeoJSON and apply a piece of code to transform GeoJSON into WKT JSON for simple features on the fly to obtain RDF notation from GeoJSON. (A003, A060, A062)
- Evaluate the need to add a BBOX element in the WKT standard. (A005-3, A060, A062)
- Best practices using JSON and JSON-LD for OWS Context files. (A086)

8.5 Semantic Enablement

Testbed-11 defined the requirements for OGC support for Semantics including ontologies, taxonomies and Linked Data. Testbed-12 shall evaluate the requirements for implementing a suite of Semantic capabilities into the OGC architecture (A006 will define the common requirements on semantics using real world use cases). Testbed-12 will implement those recommendations specifically to demonstrate support for *Object Based Production* (OBP) and *Activity Based Intelligence* (ABI) concepts.

Operational Use Case: The requirement for capabilities supporting semantic understanding and reasoning in GEOINT is an all-encompassing paradigm shift from the past. Standards play a critical role in ensuring this is accomplished in a consistent and repeatable manner. Semantic standards and services supporting semantic capabilities are at a relatively early stage of development. Interoperability between semantic standards for encoding relationships and web based services for discovery, access, retrieval and visualization of those relationships requires more testing and evaluation.

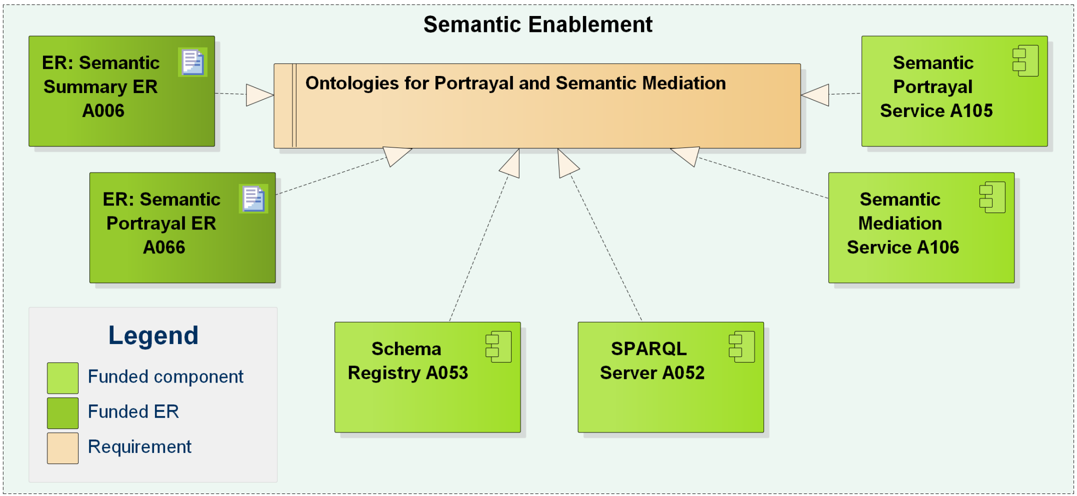

FIGURE 8.3: OGC Baseline Enhancements: Semantic Enablement requirements and work items

Deliverables

The following list identifies the *Semantic Enablement* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Evaluate providing endpoints to access and create, update and delete styles, rules, graphics information to the REST API of Semantic Portrayal Service. (A066, A105)

- Evaluate extending the Semantic Portrayal Service by providing a rendering endpoint to convert a Linked Data Model to a symbolic representation in well-known formats such as SVG or KML. (A066, A105)

- Formalize Portrayal Ontologies (Graphics) by defining graphic objects and attributes for lines and areas. (A066, A105)
- Refine the Symbology ontology to accommodate line and area-based symbols and well as composition of multiple symbols and their bindings with the geometric properties of features. (A066, A105)
- Refine the Portrayal Catalog ontology to get a model for managing registry of styles. (A053, A066)
- Refine the SPARQL Extension ontology as a result of efforts in a through e above with the intent to become a standard. (A052, A066)
- Define a RESTful API for semantically enabled services with support for Linked Data (A052, A066)
- Define the serialization in JSON-LD requirements for the REST API for Semantic Mediation Service (A066, A105)
- Evaluate the use of a Semantic Mediation Service for query rewriting (a SPARQL query for one source ontology to be converted to one or more SPARQL queries for the target ontology). (A066, A106)

8.6 Asynchronous Service Interaction

The *Asynchronous Service Interaction* subtask is part of the subtasks that concentrate on extending OGC architectural designs through efforts that cross over several individual standards and services and are applied in a much wider scope.

There are different approaches to handle asynchronous interaction with OGC Web services. The first approach (1) uses WPS facades; the second approach (2) extends each OGC Web service with asynchronous request/response capabilities. The third approach (3) refers to the OGC PubSub work, which serves as an overarching model to extend services with publish-subscribe capabilities and defines a publisher role that shall support a set of delivery methods such as e.g. ATOM, AMQP, or SOAP over HTTP. Testbed-12 shall compare, evaluate, and clarify the best approach to handle asynchronous response and data

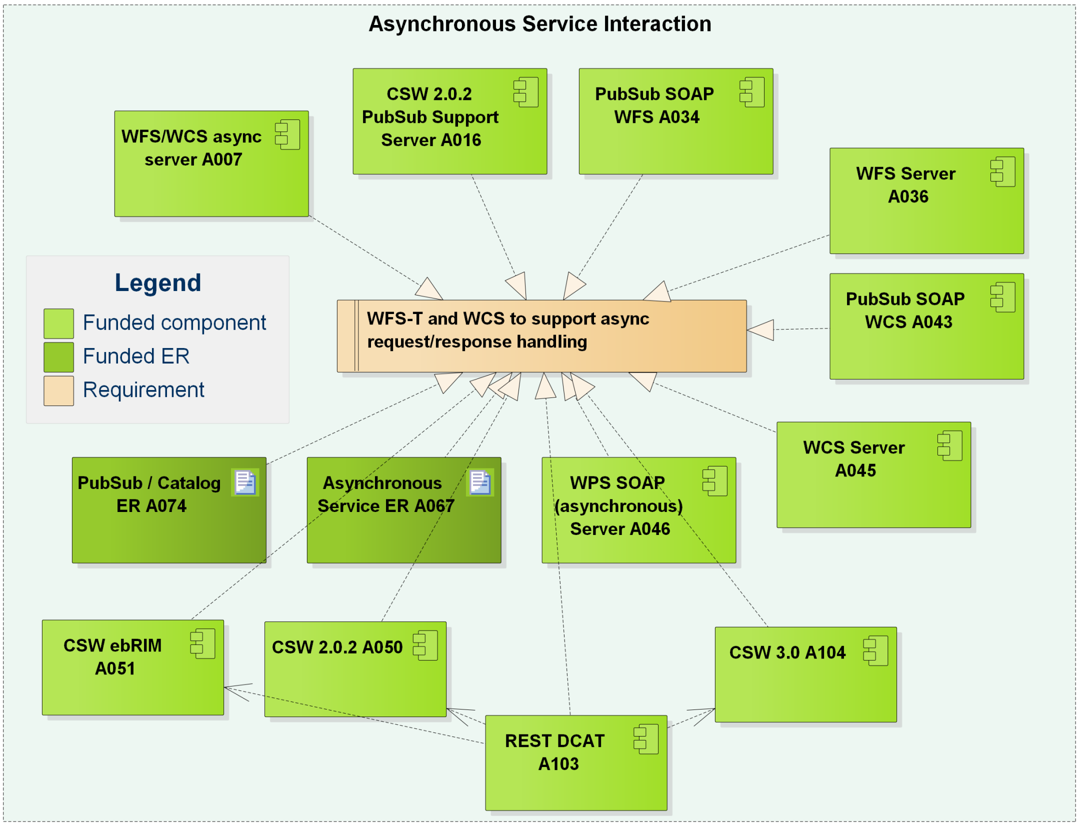

FIGURE 8.4: OGC Baseline Enhancements: Asynchronous Service Interaction requirements and work items

delivery for situations where big chunks of data. Testbed-12 shall develop asynchronous request/response for Web Feature Service Transactional (WFS-T) and Web Coverage Service (WCS).

The solution shall compare the different approaches and develop recommendations on asynchronous data delivery for OGC Web services. In order to compare the different approaches, a number of services need to be developed (WFS, WCS, CSW).

In terms of catalogs, it becomes evident that, given the volume of data, which will be available in the near future, it is important to provide methods that support notification (push) of new data as opposed to search (pull). Testbed-12 shall evaluate the use of PubSub with Catalog such that an analyst can register with PubSub to be notified when new data becomes available. PubSub shall be capable of implementing based on area of interest and/or keywords.

The following table provides an overview of the various tests that shall be implemented and evaluated:

Server	Approach	Components	ER
WFS	1	A036, A046	A067
WFS	2	A007	A067
WFS	3	A034	A067
WCS	1	A045, A046	A067, A074
WCS	2	A007	A067
WCS	3	A043	A067, A074
Cat	3	A016, A050, A051, A103, A104	A074

TABLE 8.1: Asynchronous services and ER overview

Deliverables

The following list identifies the *Asynchronous Service Interaction* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Extensions to existing WFS and WCS to deliver big chunks of data asynchronously according to the WPS-based approach described above as first approach (A036, A045, A046)
- WFS or WCS asynchronous server according to the second approach described above (A007)

- Extensions to existing WFS/WCS to deliver big chunks of data asynchronously according to the PubSub-based (third) approach described above (A034, A043)
- Asynchronous Service Response ER describing the comparison results of the various approaches to handle asynchronous response and data delivery for situations where big chunks of data which require asynchronous delivery (A067)
- PubSub / Catalog ER to describe how the OGC PubSub standard can be used as a mechanism to automatically notify analysts of data availability for WFS, WCS, and CSW (A074)
- CSW 2.0.2 with PubSub Core support to evaluate the use of PubSub with Catalog such that an analyst can register with PubSub to be notified when new data becomes available. PubSub shall be capable of implementing based on area of interest and/or keywords (A016)

8.7 Tiling

There are several major advantages to tiling. Each time the user pans, most of the tiles are still relevant, and can be kept displayed, while new tiles are fetched. This greatly improves the user experience, compared to fetching a single image for the whole viewport. It also allows individual tiles to be pre-computed. Testbed-12 shall evaluate a consistent method for tiling across where Raster tiles and Vector tiles can be overlaid utilizing the same tile pyramid structure.

Vector tiles are packets of geographic data, packaged into pre-defined roughly-square shaped "tiles" for transfer over the web. This is an emerging method for delivering styled web maps, combining certain benefits of pre-rendered raster map tiles with vector map data. As with the widely used raster tiled web maps, map data is requested by a client as a set of "tiles" corresponding to square areas of land of a pre-defined size and location. Unlike raster tiled web maps, however, the server returns vector map data, which has been clipped to the boundaries of each tile, instead of a pre-rendered map image. Testbed-12 shall develop a consistent tiling scheme to support both raster and vector tiling, including storage in the tile container database.

Deliverables

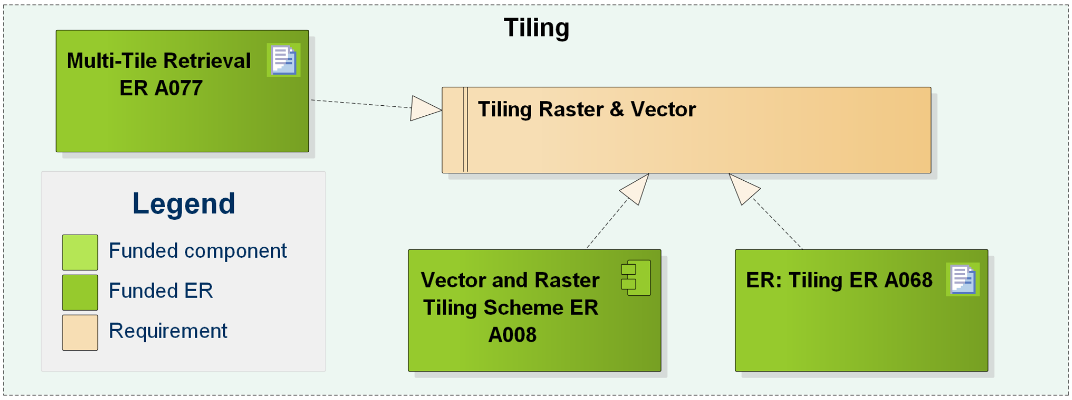

FIGURE 8.5: OGC Baseline Enhancements: Tiling requirements and work items

The following list identifies the *Tiling* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Testbed-12 shall develop a consistent tiling scheme to support both raster and vector tiling. Testbed-12 shall demonstrate this in support of GeoPackage tasks. (A008)
- Discusses all aspect relevant to vector tiling. Related to A008, the Engineering report on the vector and raster tiling scheme, but focuses on the implementation of vector tiles within GeoPackages. (A068)
- Multi-Tile Retrieval ER describe options considered and recommendations for delivery of large amounts of tiled data in support of GeoPackage. (A077)

8.8 Compression

The Army Geospatial Center has recommended U.S. Department of Defense adoption of EXI compression, however the OGC has defined the Well-Known Binary format (WKB) as part of the Simple Feature Specification as an option for XML compression.

EXI, the Efficient XML Interchange format, is a W3C recommendation, adopted February 2014: *"EXI is a very compact representation for the Extensible Markup Language (XML) Information Set that is intended to simultaneously optimize performance* *and the utilization of computational resources. The EXI format uses a hybrid approach drawn from the information and formal language theories, plus practical techniques verified by measurements, for entropy encoding XML information. Using a relatively simple algorithm, which is amenable to fast and compact implementation, and a small set of datatype representations, it reliably produces efficient encodings of XML event streams. The grammar production system and format definition of EXI are presented."* (W3C)

Testbed-12 shall evaluate the advantages and disadvantages of each method and make a recommendation for NSG adoption. Key to this will be the availability of commercial support for compression and decompression without user interaction.

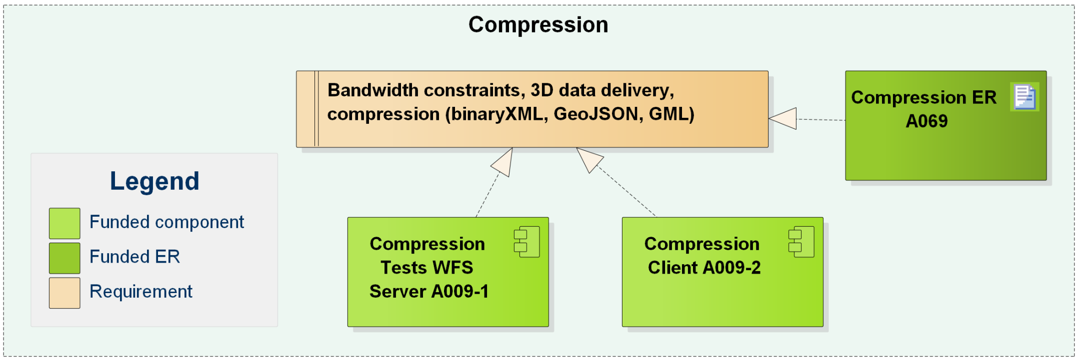

FIGURE 8.6: OGC Baseline Enhancements: Compression requirements and work items

Deliverables

The following list identifies the *Compression* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- The Compression Techniques ER shall document the various approaches, highlight pros and cons of the different approaches and compare performance aspects. (A069)
- Compression Tests WFS Server to send compressed data (A009-1)
- Compression Client to analyze compression performance (A009-2)

8.9 SWE for LiDAR and Streaming

LiDAR support is being developed in conjunction with the Community Sensor Model Working Group (CSMW). The Defense Information System Agency (DISA) has an established requirement for LiDAR to be compliant with OGC Sensor Web Enablement suite of standards. Testbed-12 shall compare the LiDAR recommendations of the CSMW and past OGC SWE recommendations on LiDAR. Testbed-12 shall define common ground and enable interoperability if possible between the CSMW recommendations and the OGC SWE suite of services. Testbed-12 shall determine the optimal method to deliver LiDAR data through a OGC Sensor Observation Service. The LiDAR SOS shall provide a streaming mechanism, similar to JPIP.

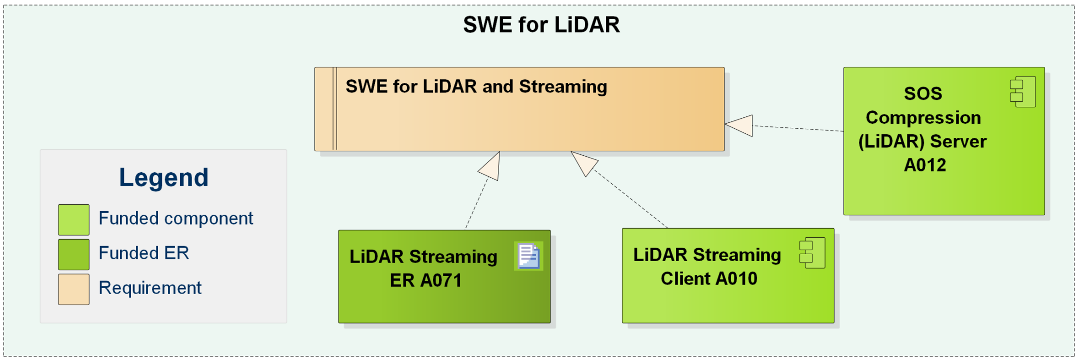

FIGURE 8.7: OGC Baseline Enhancements: SWE for LiDAR and Streaming requirements and work items

Deliverables

The following list identifies the *SWE for LiDAR and Streaming* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- The LiDAR Streaming ER shall describe how LiDAR streaming can be realized using the OGC Sensor Web Enablement suite of standards and how the Community Sensor Model and SWE LiDAR can be harmonized. (A071)
- SOS server supporting LiDAR streaming and EXI and WKB and potential other solutions for compressing of LiDAR data streams (A012)

- A010 LiDAR streaming client that interacts with the two SOS servers defined above (A010)

8.10 WFS Synchronization

Testbed-12 seeks an evaluation of modeling the *SyncResource* operation as an extension of the WFS *GetFeature* request as discussed in Testbed-11 report OGC 15-010, OGC Testbed-11 Summary Report of Findings for WFS-T Information Exchange Architecture. SyncResource only allows filtering on the Feature Type. Extending GetFeature would add support for:

1. **query** – support the full query expression, not just the Feature Types
2. **resultType** – do I really want all of the changes since the last sync? Let's get the hit count first.
3. **resolveWithPath** – Without this element the data set may be incomplete.

SyncResource then becomes an extension of *GetFeature* with additional elements for *checkpoint* and *serviceID*. These elements serve to identify the scope of the synchronization.

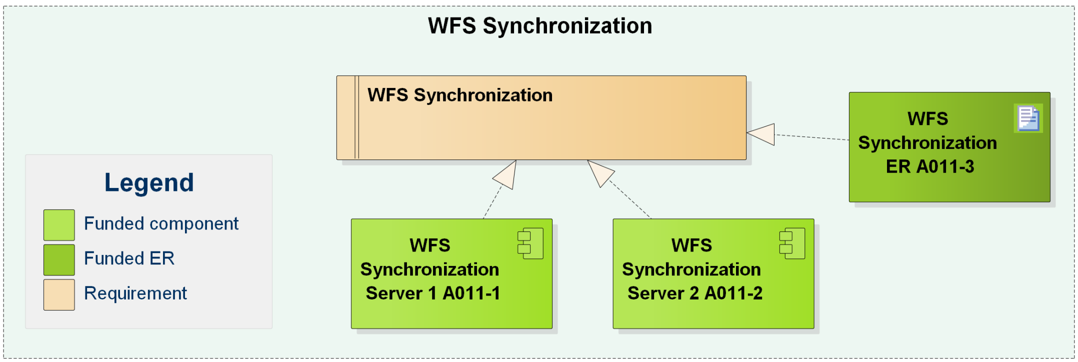

FIGURE 8.8: OGC Baseline Enhancements: WFS Synchronization requirements and work items

Deliverables

The following list identifies the *WFS Synchronization* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- WFS Synchronization Server 1 (A011-1)
- WFS Synchronization Server 2 (A011-2)
- WFS Synchronization ER (A011-3)

8.11 Catalog

Current Catalog solutions are highly dependent upon the metadata model employed for the service and data descriptions. Many of today's service instances and data holdings are based on an ISO metadata model.

Established U.S. Dept. of Defense catalogs are often accessed under a security constrained architecture implementing SOAP and WS-Security.

The W3C has released the DCAT recommendation in early 2014. *"DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. By using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume metadata from multiple catalogs. It further enables decentralized publishing of catalogs and facilitates federated dataset search across sites."* (W3C). Thus, DCAT defines a standard way to publish machine-readable metadata about a dataset. It does not make any assumptions about the format of the datasets described in a catalog.

Testbed-12 shall evaluate interoperability aspects in multi-catalog type environments, including CSW featuring ISO based metadata and OpenSearch, a second CSW offering a SOAP binding, and a third DCAT implementation, describing the same services and data sets using RDF.

Deliverables

The following list identifies the *Catalog* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Implement four different catalogs (CSW 3.0, CSW 2.0.2, CSW ebRIM, DCAT) and show how they can interact. Demonstrate what role DCAT can play as a heterogeneous catalog integration mechanism and as a possible simplification of the setup and use of catalogs. (A050, A051, A072, A103, A104).

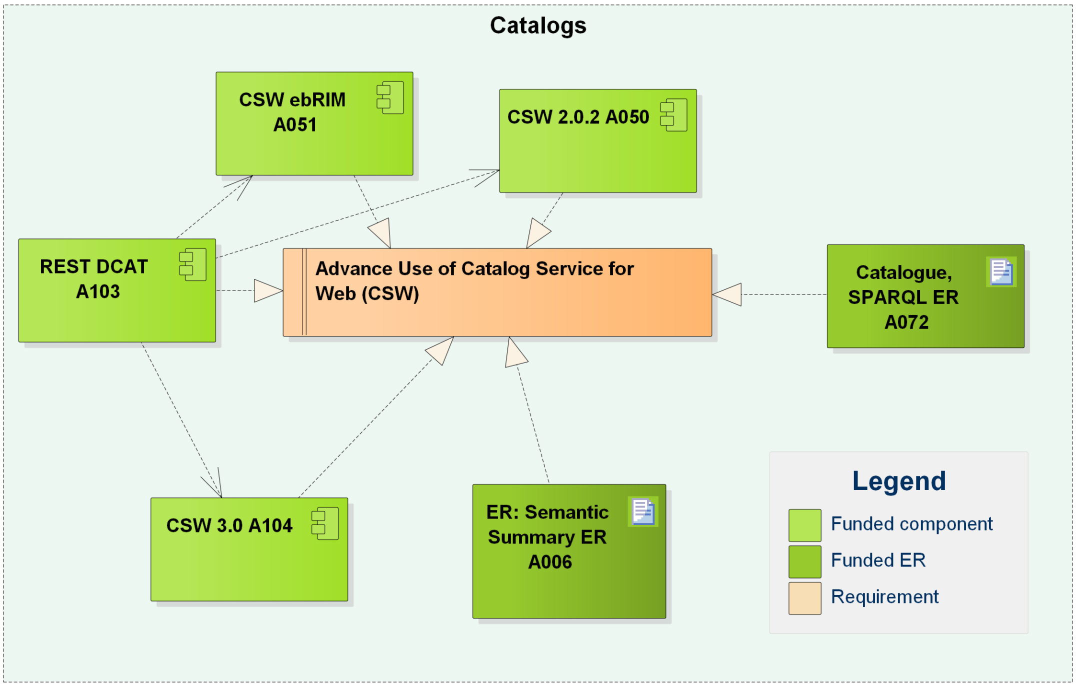

FIGURE 8.9: OGC Baseline Enhancements: Catalog requirements and work items

- All four catalogs (CSW 3.0, CSW 2.0.2, CSW ebRIM, DCAT) shall contribute to the discussion on optimized capabilities for catalogs and implement the discussion results. See section 8.12 for further details (A050, A051, A072, A103, A104).
- All four catalogs (CSW 3.0, CSW 2.0.2, CSW ebRIM, DCAT) shall contribute to the discussion on publish-subscribe interactions as described in section 8.6 and implement the discussion results if pub-sub functionality can be added (A050, A051, A072, A103, A104).
- Testbed-12 shall consider Testbed-11 recommendations related to pycsw CSW.
- Essential semantic aspects shall be summarized in the Semantic Summary ER (A006), see section 8.5 for further details.
- All aspects addressing SPARQL and catalogs shall be discussed as part of the Catalog SPARQL ER, A072.

8.12 Capabilities Document and Alternatives

Testbed-12 shall evaluate the current Capabilities document. With the introduction of Big Data extremely large amounts of data holding may be available on a service or in a Cloud environment which may be scattered across multiple servers. Servers may offer lots of data sets that are advertised in the Capabilities document traditionally.

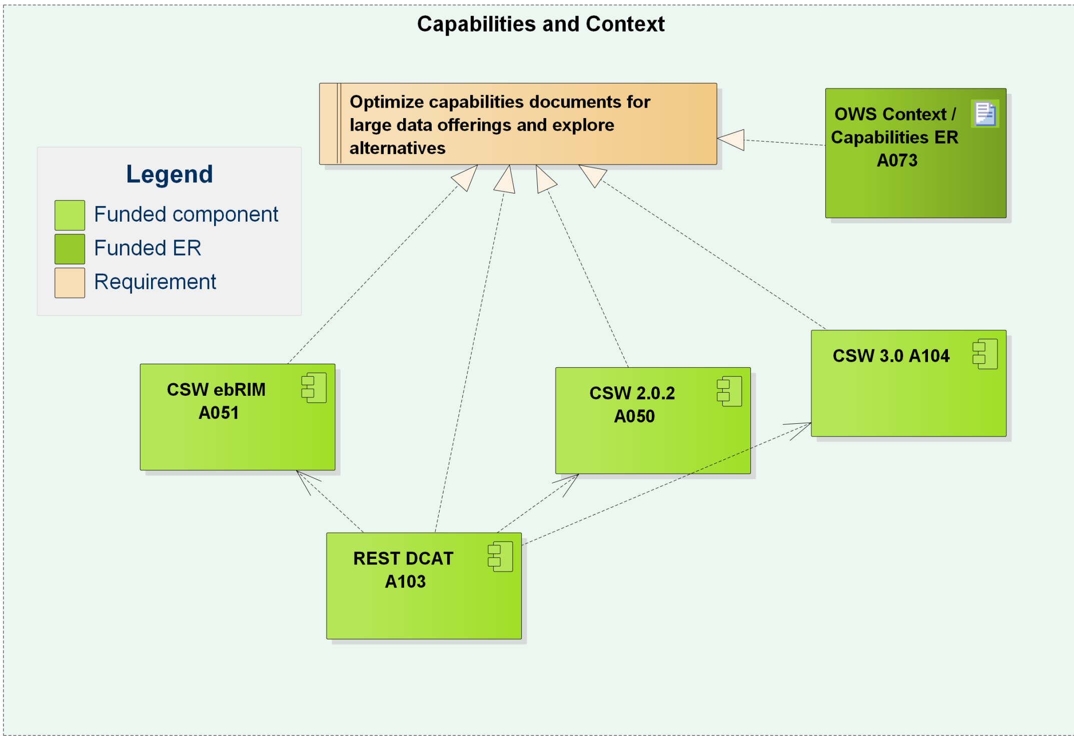

FIGURE 8.10: OGC Baseline Enhancements: Capabilities Document and Alternatives requirements and work items

In addition, Testbed-12 shall evaluate the potential of OWS Context or other standards working in conjunction with Catalog as support to the Cataloging of large data holdings. Evaluate using OWS Context as an index or container to describe the data contents of a web service. Consider the impact of a Cloud based architecture and describe how a OWS Context based approach relates to the Capabilities document approach and provide input for harmonization or change requests as appropriate.

Deliverables

The following list identifies the *Capabilities Document* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Engineering report on Capabilities documents and OWS Context as alternative solutions that explains how very large data offerings by services can be handled efficiently (A073)

8.13 Big Data and Tile Stores

Assessment of a file format (database format) for exchange of a large (global/Regional) tile store. Evaluate solutions such as simple reuse of what is defined for raster tiling in GeoPackage and applying it to a postGRESql database instead of a Sqlite DB.

Assessment of approach to an online Service to access multiple tiles at a time (a set of tiles by a bounding box) that possibly could be a tile access Web Processing Service (WPS) extension to WMTS.

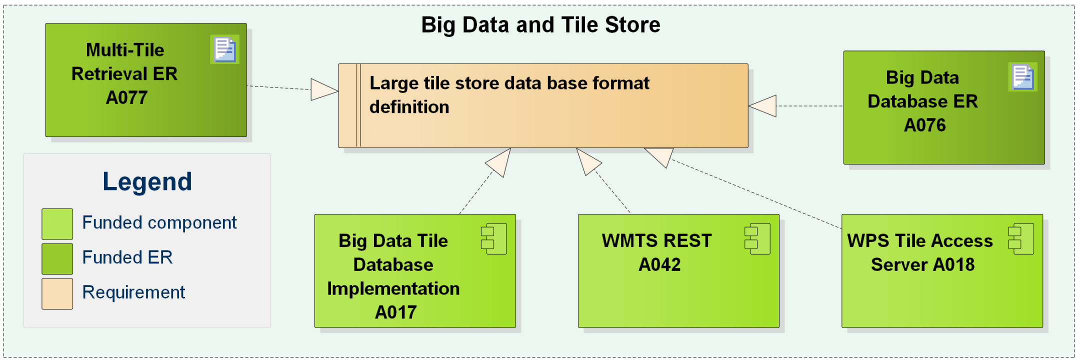

FIGURE 8.11: OGC Baseline Enhancements: Big Data and Tile Stores requirements and work items

Deliverables

The following list identifies the *Big Data and Tile Stores* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Big Data Tile DB implementation. (A017)
- Big Data Database ER describes options considered and recommendations for delivery of large amounts of data as a database delivery. (A076)
- Multi-Tile Retrieval ER describe options considered and recommendations for delivery of large amounts of tiled data in support of GeoPackage. (A077)
- WPS server that provides tiles access (A018)
- WMTS server to serve image tiles (A042)

8.14 Web Integration Service

Testbed-12 shall evaluate the Testbed-10 Service Integration Engineering Report OGC 14-031r1 and determine the need for a new service or whether these functions can be applied in the current architecture. Consider the work being done in Testbed-12 on the Capabilities document and where this service concepts may support better discovery of relevant data and services. This document specifies technical changes to the OGC web service architecture baseline to support better integration among the services. Although integration may be achieve in a number of ways and using a number of other technologies, the goal of this document is to achieve this integration within the current OGC service framework in order to leverage existing investments in OGC web services infrastructure.

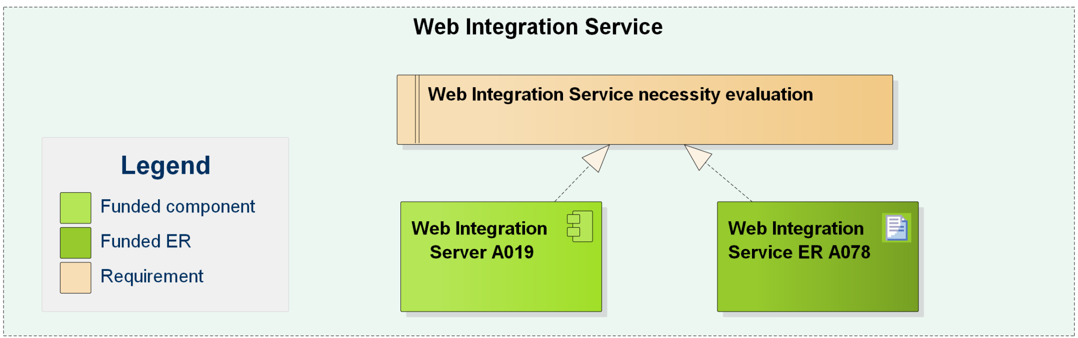

FIGURE 8.12: OGC Baseline Enhancements: Web Integration Service requirements and work items

Deliverables

The following list identifies the *Web Integration Service* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- Web Integration Service server (A019)
- Web Integration Service Recommendations ER (A078)

8.15 Conflation

Hootenanny was developed to provide an open source, standards-based approach to geospatial vector data conflation. Hootenanny is designed to facilitate automated and semi-automated conflation of critical foundation GEOINT features in the topographic domain, namely roads (polylines), buildings (polygons), and points-of-interest (POI's) (points). Conflation happens at the dataset level, where the user's workflow determines the best reference dataset and source content, geometry and attributes, to transfer to the output map.

Hootenanny is an open library of conflation algorithms applies various techniques to unify the geometry and metadata of topographic features. Conflicts can be visualized and resolved through an interactive application built on the iD Editor, an open source map-editing tool developed by Mapbox. Conflated datasets can be exported in a variety of GIS formats including ESRI Shapefile, File Geodatabase, Web Feature Service, and native OpenStreetMap™ formats. Hootenanny also enables Geospatial Extract Transform Load (ETL) capabilities supporting various schemas such as Topographic Data Store (TDS), and Multi-National Geospatial Co-Production Program (MGCP). Hootenanny leverages the open architecture of OpenStreetMap™ to facilitate integration of diverse geospatial datasets into a common key value data structure.

Testbed-12 shall evaluate and develop solutions to integrate the Hootenanny tool with enhancements if identified into standardized spatial data infrastructures. Hootenanny modifications shall be made available through the current GitHub location. The following aspects shall be addressed: How can data provided by OGC Web services such as WFS, WCS, or SOS together with other data sources such as crowd-sourced information or social media activity streams are conflated in an efficient manner using Hootenanny? How can Hootenanny be best operationalized, given that it integrates functionality to edit and conflate data and to resolve conflicts?

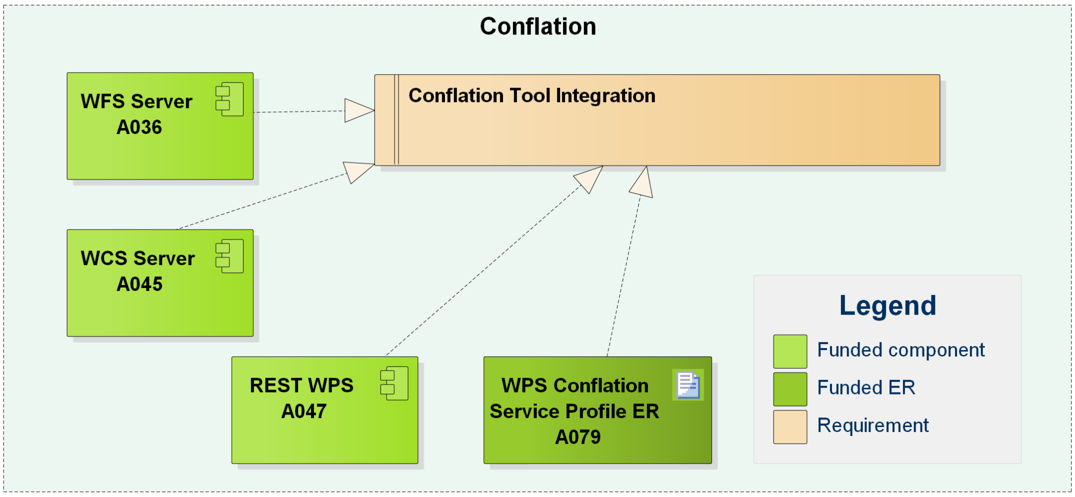

FIGURE 8.13: OGC Baseline Enhancements: Conflation requirements and work items

Deliverables

The following list identifies the *Conflation* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary tables 8.2, 8.3, and 8.4 at the end of the chapter.

- RESTful WPS server facading an Hootenanny implementation to demonstrate efficient conflation of data provided by OGC Web services such as WFS, WCS, or SOS together with other data sources such as crowdsourced information or social media activity streams (A047)
- WFS server serving data (A036)
- WCS server serving data (A045)
- The engineering report shall describe ho Hootenanny can be best operationalized, given that it integrates functionality to edit and conflate data and to resolve conflicts? (A079)

8.16 Data Quality

The Citizen Observatory WEB (COBWEB) project recently completed work on a workflow Orchestration for Quality Assurance of crowd-sourced Data. Testbed-12 requires a follow on to this work to include not only crowd-sourced data but all feature data. Testbed-12 shall evaluate and operationally implement the architecture as identified in the COBWEB project in support of a Data Quality analysis of all feature data sources used within the Testbed-12. The WPS services shall enable analysis of data and metadata content based on the ISO 19139 Data Quality elements. The results shall be automatically reported as conformant to the NSG Metadata Framework.

This work item is strongly linked to chapter 5. A solution shall be developed for both vector and coverage data.

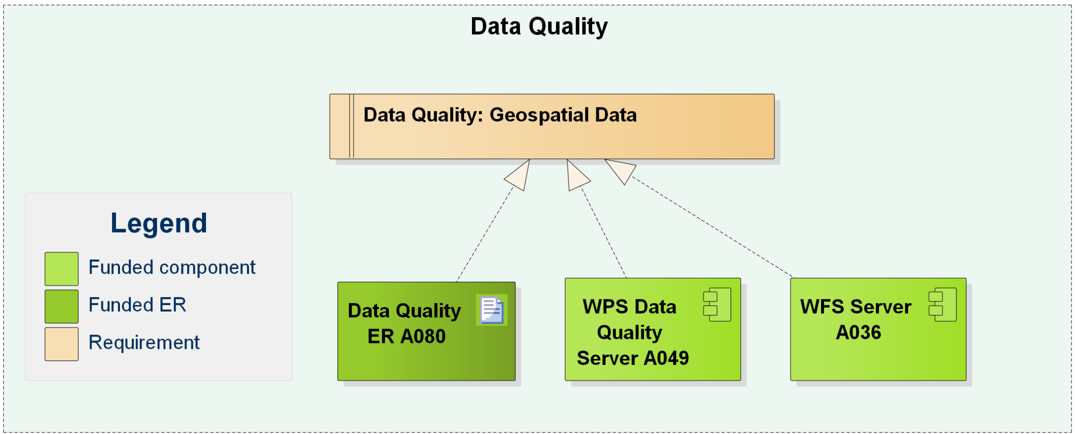

FIGURE 8.14: OGC Baseline Enhancements: Data Quality requirements and work items

Deliverables

The following list identifies the *Data Quality* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

- WPS service implementation that shall enable analysis of data and metadata content based on the ISO 19139 Data Quality elements. The results shall be automatically reported as conformant to the NSG Metadata Framework. (A049)
- WFS server to serve data (A036)

- Engineering report that captures all results from the the data quality WPS work (A080).

8.17 GeoPackage

The OGC GeoPackage standard supports the ability for the military (and others) to load tiled map/image data and vector data onto a handheld device and utilize that data in a disconnected or limited connectivity environment. The standard has been incrementally improved through the efforts of the Testbed activities each year. Current Testbed-11 efforts have enabled in-the-field data collections to update the database with the most current information. Testbed-11 also developed a peer-to-peer in-the-field update via Bluetooth connection. Testbed-12 shall evaluate and implement the *NSG GeoPackage Profile* with support for the requirements identified in the following subsections. NSG GeoPackage Profile defines and tailors the implementable provisions prescribed for the National System for Geospatial-Intelligence (NSG) for a GeoPackage based on the Open Geospatial Consortium GeoPackage Encoding Standard. It provides detailed direction on how to use the clauses, options, and parameters of the base standard GeoPackage standard. The guidance is designed to be specific enough for any two independent and compliant software implementations to 'plug and play' with each other.

The following subsections describe all *GeoPackage* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

8.17.1 GeoPackage: Vector Tiling

Testbed-12 shall utilize the GeoPackage as the implementation of Vector Tiling requirements stated above in section 8.7. Those include the development of a consistent tiling scheme to support both raster and vector tiling. Here, GeoPackages shall be created that implement that tiling scheme.

Deliverables:

- GeoPackage NSG Profile with Vector Tiles Implementation. (A022)

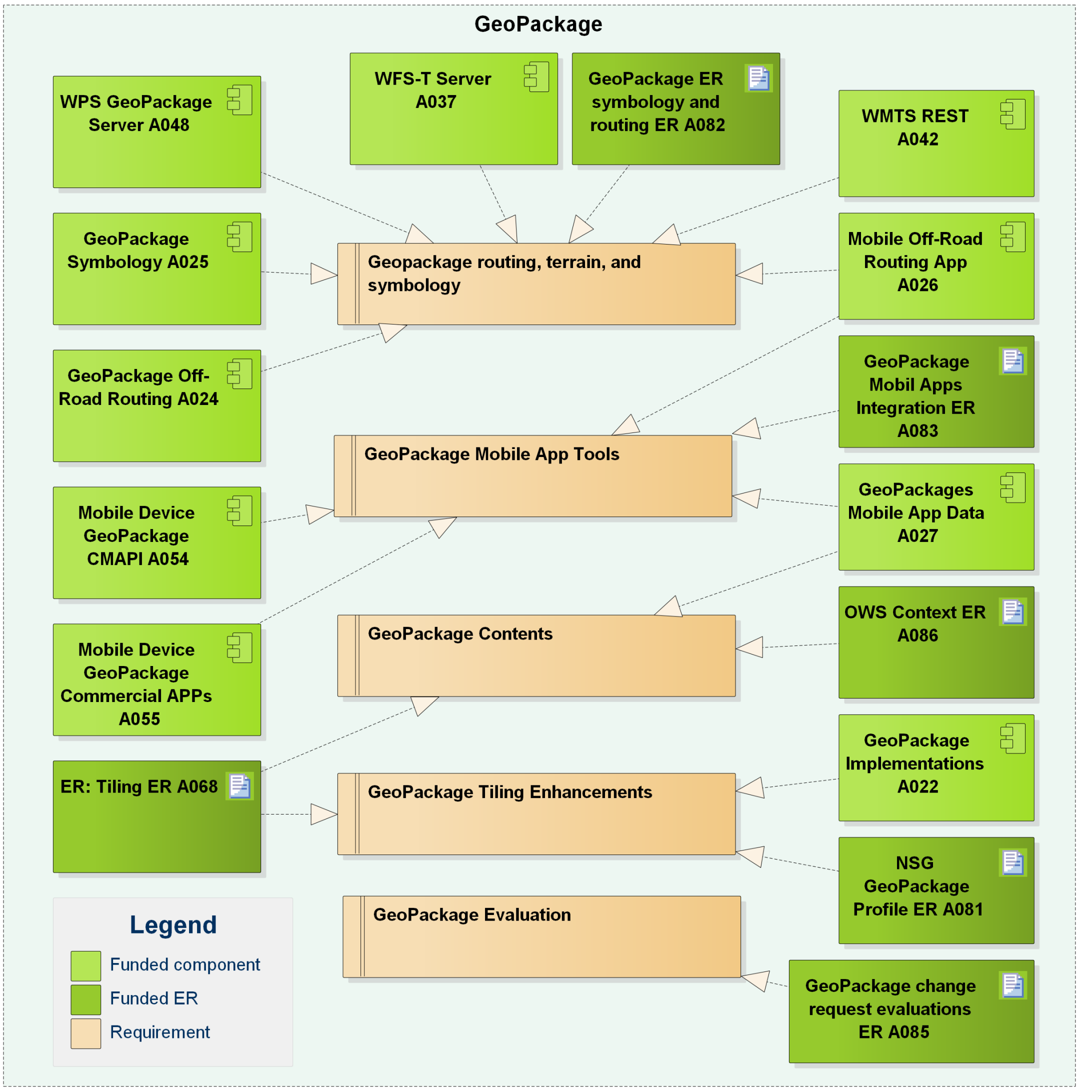

FIGURE 8.15: OGC Baseline Enhancements: GeoPackage requirements and work items

- Tiling ER describing both the Raster and Vector Tiling solutions and their implementation in GeoPackage. (A068)
- NSG GeoPackage Profile Assessment ER describes any concerns with alternative options considered and recommendations for implementation of the NSG GeoPackage Profile. (A081)

8.17.2 GeoPackage: Routing and Symbology

Testbed-12 shall expand on the recommendations and work accomplished in Testbed-11 for enhanced capabilities for symbology and routing. Testbed-12 shall implement the recommendations of the GeoPackage Interoperability Experiment for Terrain data. Testbed-11 provided a method for on road routing. Testbed-12 shall provide a method for the dismounted off road routing (cross country). This will require elevation data, cached/tiled map background and georeferenced imagery Testbed-12 shall define a mechanism for providing user defined map symbology of GeoPackage feature content. See Semantically Enabled section above for symbology requirements. Reference to figure 8.15: *GeoPackage routing, terrain, and symbology*.

Deliverables

- GeoPackage implementation with data for off road routing calculation (A024)
- GeoPackage implementation with symbology and styles in support of the off road routing GeoPackge (A025)
- Mobile App supporting off road routing calculation based on GeoPackage data (A026)
- WFS serving data for GeoPackage generation (A037)
- WMTS serving data for GeoPackage generation (A042)
- WPS server producing GeoPackages A024 (off road data) and A025 (symbology and styles) (A048)
- GeoPackage ER describing symbology and routing information solutions (A082)

8.17.3 GeoPackage: Mobile App Tools

Testbed-11 demonstrated the ability to use GeoPackage as the data container to report back updates from the field. It also demonstrated the ability to synchronize between mobile devices in the field (as developed during using Bluetooth ad hoc connections). Testbed-12 shall now evaluate the interoperability of the Common Map API tool with commercial vendor tools supporting GeoPackage. The goal is to understand how data can be exchanged between apps using GeoPackage as a shared memory space, potentially enriched with direct interprocess-communication or shared memory as supported by the host operating system.

As an example, consider an extension to the routing app described above (8.17.2 that stores all network data in a GeoPackage and calculates best routes based on a variety of aspects on the mobile device. A person follows a calculated route but decides at some stage to leave the vehicle behind to continue the journey by foot. Now, additional information is required, such as terrain, vegetation, hydrography, closed areas etc. This additional information is stored in other GeoPackage and maintained by other apps. How can these apps exchange information efficiently? How can GeoPackage be used efficiently? Testbed-12 shall evaluate the contents of the NSG Application Schema feature and attribute content and make recommendations on the suitable level of detail required to support GeoPackage on a handheld device. For example what are the feature/attribute data requirements needed to support a GeoPackage routing scenario, including aspects such as:

- Transportation (e.g., roads, railroads, and bridges)
- Hydrography (e.g., bodies of water, coastlines)
- Cultural (e.g., buildings, facilities, landmarks)
- Terrain (e.g., vegetation and soils)
- Administrative areas and boundaries

Reference to figure 8.15: *GeoPackage Mobile App Tools*.

Deliverables

- Mobile App supporting off road routing calculation based on GeoPackage data (A026)

- GeoPackages with the data types defined above (A027)
- Mobile Device GeoPackage Common Map API Implementation (A054)
- Mobile Device GeoPackage Commercial APPs with data as defined above (A055)
- GeoPackage Mobile Apps Integration ER (A083)

8.17.4 GeoPackage: Contents

As described in sections 8.7 and 8.18, Testbed-12 shall agree upon method to describe the contents of a GeoPackage. The GeoPackage and OWS Context SWGs have jointly worked to develop a potential solution using OWS Context. It has been agreed in principle that OWS Context could be used to describe the contents of the GeoPackage containing it or one or more external GeoPackages. Testbed-12 shall develop a consistent approach to using OWS Context (8.18) as an index descriptor for the content of GeoPackage files. The description shall include the indexing of multiple tiles of raster and vector content as required in the section *Tiling* (8.7) as well as be linked to the OWS Context discussion as part of 8.18.

Reference to figure 8.15: *GeoPackage Contents*.

Deliverables

- GeoPackage / OWS Context indexing implementation (A027)
- Tiling ER describing both the Raster and Vector Tiling solutions and their implementation in GeoPackage (A068)
- OWS Context ER describing how OWS Context can be used as an index descriptor for the content of GeoPackages (A086)

8.17.5 GeoPackage: Evaluations

Several Change Requests have been submitted against the current GeoPackage standard. These change requests require further evaluation to determine the path forward. Testbed-12 shall evaluate the following issues and provide recommended solutions to the GeoPackage standard.

Reference to figure 8.15: *GeoPackage Evaluation*.

1. Demoting metadata and schema sections in GeoPackage standard to an extension to the core standard
2. Evaluate Vertical CRS requirements as part of the Elevation Extension to GeoPackage alignment with SWE

Deliverables

- GeoPackage change request evaluations ER (A085)

8.18 OGC Web Services (OWS) Context

The OGC Web Services Context Document (OWS Context) 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 OGC Web Feature Service (WFS), Web Map Service (WMS), Web Map Tile Service (WMTS), Web Coverage Service (WCS) and others in a 'common operating picture'. Additionally OWS Context can deliver a set of configured processing services (Web Processing Service (WPS)) parameters to allow the processing to be reproduced on different nodes.

Testbed-11 identified a number of recommendations for the OWS Context standard. Testbed-12 shall evaluate and implement those recommendations and develop test OWS Context documents conformant with those efforts.

Deliverables

The following list identifies the *OGC Web Services (OWS) Context* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

- Develop an *OWS Context Document Production* client that can produce OWS Context documents for various services. (A030)
- Evaluate and implement recommendations from Testbed-11 and develop conformant OWS Context documents using the OWS Context client. (A030, documented in A086)

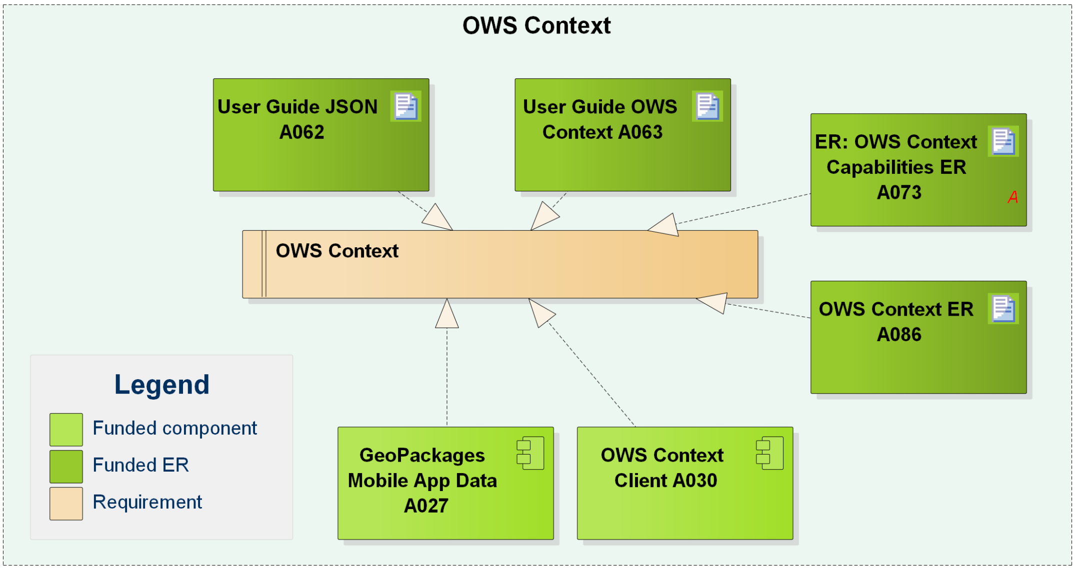

FIGURE 8.16: OGC Baseline Enhancements: OWS Context requirements and work items

- Develop an index descriptor for the content of GeoPackage files, taking recommendations from Testbed-11 into account (A027 (GeoPackage), A086 (documentation))
- Evaluate the need to create an extension of OWS Context JSON for illustrating how to reference GeoJSON data both embedded or linked. (A062, A063, A086)
- Consider the HTML Microdata approach as a new encoding for OWS Context as a way to improve auto-discovery of OGC services and geospatial resources. (A062, A063, A086)
- Consider the benefits and drawbacks of extending schema.org vocabularies with a new type for "geospatial resource" that completely matches with OWS Context conceptual model. (A062, A063, A086)
- Consider JSON-LD encodings for HTML structured data to create another encoding for OWS Context. (A030, A062, A063, A086)
- Develop an OWS Context User Guide that describes best practices on OWS Context taking all Testbed-12 developments into account. (A063)

8.19 UML Shape Change

Significant updates and enhancements to the NSG Application Schema have been made. Testbed-12 shall evaluate and extend the capabilities of the UGAS ShapeChange tool in order to address the following:

1. Extend the ShapeChange profiler transformation to support restrictions on property multiplicities and OCL constraints, and enable restrictions to be specified by an external configuration file/API rather than being carried exclusively as UML tag-values in the application schema itself.

2. Extend the ShapeChange output target for RDF/OWL based on TBD additional/alternative rules from those specified in ISO 19150-2 to improve interoperability with, and reuse of, ontologies from outside of the geospatial community. Add support for N-triples format.

3. Add ShapeChange output target for RDF/SKOS to support specification of Controlled Vocabularies and Taxonomies, including support for N-triples format.

4. Review and extend ShapeChange output target for JSON/GeoJSON in light of the evolving W3C and OGC standards and application software landscape. Enhance and extend the flattener transformation, as necessary, to support derivation of JSON/GeoJSON schemas from complex application schemas (e.g., the NAS).

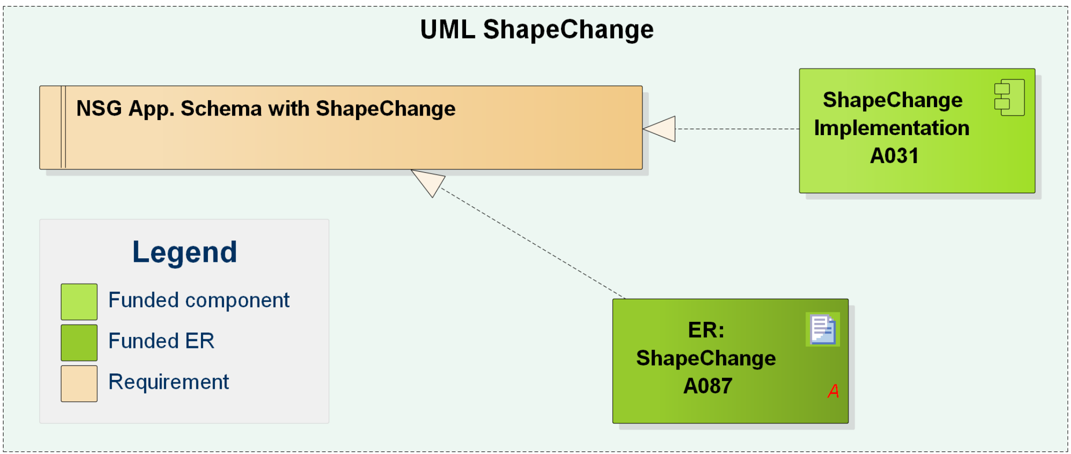

FIGURE 8.17: OGC Baseline Enhancements: UML Shape Change requirements and work items

Deliverables

The following list identifies the *UML Shape Change* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

- Implementation of the ShapeChange tool to support all elements identified above. (A031)
- Engineering report capturing all discussions and results from the ShapeChange implementation work. (A087)

8.20 Data

Testbed-12 shall continue to use the Testbed-11 "area of interest" (San Francisco Bay area) as defined by that scenario in order to make the most efficient use of data resources already available. Testbed-12 will supplement the Testbed-11 data with additional data content and service requirements.

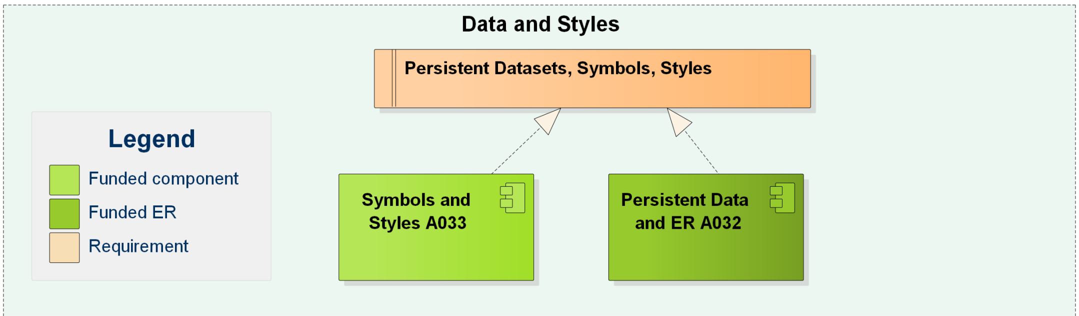

FIGURE 8.18: OGC Baseline Enhancements: Data and Portal requirements and work items

Testbed-12 requires the extraction of a 2D test dataset to support demo scenarios at the start of next testbed. Data shall be extracted over the Testbed-11 "area of interest" (San Francisco Bay area). Testbed-12 shall extract 2D data conformant with the feature, attribute and metadata requirements in the NSG Application Schema 7.0. This data shall be convertible to conform to the NSG Entity Ontology (NEO). Final delivery shall be 2 datasets, one encoded as GML (NAS 7.0) and a second encoded using RDF/OWL (NEO). These datasets shall be provided to OGC for future use on a royalty-free basis.

Testbed-12 shall evaluate and develop Symbol Encodings and Styled Layer Descriptors supporting the 2D data extraction requirements above. These symbol encodings and style layer descriptors shall support 3D/4D data conformant to the NAS descriptions as well.

Deliverables

The following list identifies the *Data* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

- Data sets and data sets descriptions (A032)
- Styles and symbols with descriptions (A033)

8.21 User Guides

Testbed-12 requires the development of a practical set of User Guides providing not only Web service implementer instructions but more specific guidance for a user of those services. The following User Guides are required to contain an Executive Summary for overall management level review and understanding, a User Guidance section complete with service discovery and access/retrieval of data guidance with a final section devoted to programmer/implementer instructions:

1. Implementing a REST based architecture for OGC service types User Guide
2. Implementing a OGC SOAP based WFS, WMS and WCS User Guide
3. OGC (Geo)JSON User Guide
   (a) The User Guide shall include examples and guidance on converting XML documents into (Geo)JSON.
   (b) Consider OGC 14-009r1 recommendations.
   (c) Include a subclause for linking to other objects in JSON, using the natural approaches that JSON-LD provides for both simple links and atom links.
   (d) Include a subclause for describing the process for if a fragment of a XML document contains a geospatial object then when converting to JSON, consider using the GeoJSON equivalent type.

4. OWS Context User Guide including descriptions of use with the GeoPackage standard

5. CITE Virtual Machine Installation and Execution User Guide (see section 8.22 for further details.

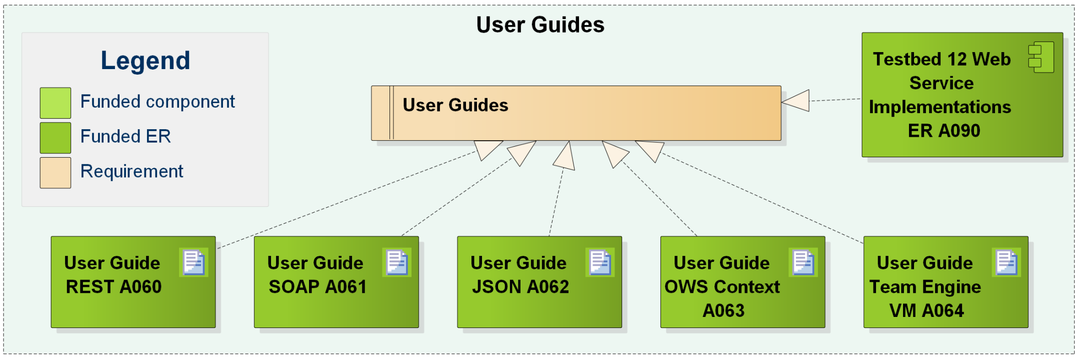

FIGURE 8.19: OGC Baseline Enhancements: User Guides requirements and work items

Deliverables

The following list identifies the *User Guides* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

- REST architecture user guide (A060)
- SOAP implementation user guide (A061)
- (Geo)JSON user guide (A062)
- OWS Context user guide (A063)
- CITE VM installation user guide (A064)
- CITE User Guide with Profile Testing (A088)

8.22 Compliance Testing

The goal of the OGC Compliance Program is to increase systems interoperability while reducing technology risks. It accomplishes this by providing a process whereby compliance for OGC standards can be tested. This program provides a mechanism by which users and buyers of software that implements OGC standards can be certain that the software follows the mandatory rules of implementation as specified in the standard. Vendors gain confidence that they are providing a product compliant with OGC standards, which will be easier to integrate and easier to market. Buyers gain confidence that a compliant product will work with another compliant product based on the same OGC standard, regardless of which company developed the product.

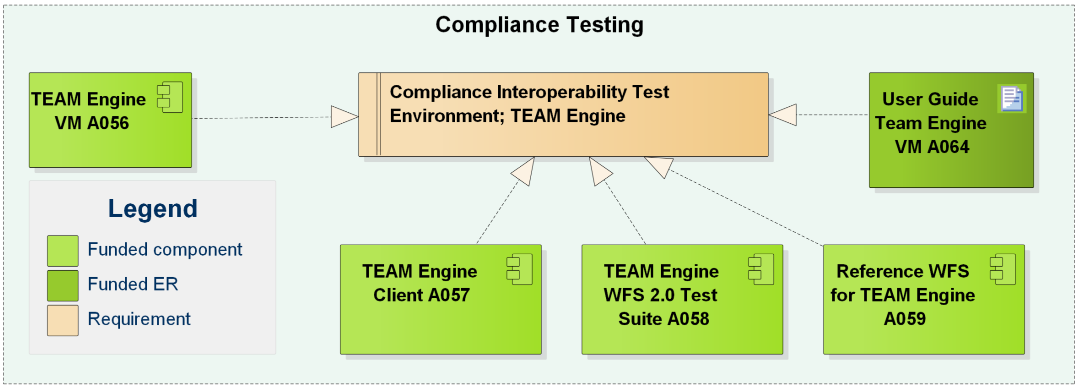

FIGURE 8.20: OGC Baseline Enhancements: Compliance Testing requirements and work items

As part of the Compliance program, OGC provides a free "self-service" web testing facility that can be used by any developer as often as they like to test their implementations of OGC standards. This facility is also used to capture evidence that implementations are properly implementing OGC standards. The OGC testing facility is based on Executable Test Scripts (ETS), which is software code that implements the Abstract Test Suites (ATS). The ATS usually appears as Annex of OGC Implementation Specifications.

The compliance work on Testbed-12 will advance the use of OGC compliance tools in secure environments and the use of community profiles. Some agencies have operational requirements to provide OGC service compliance testing on the secure networks. This will allow to test software that shall not be exposed to the general public. Communities also want to provide a simplified method for profile testing. Communities running validation tools desire to have a detailed report of the conformance classes that come form the core standards and the conformance classes that come from the community requirements.

Deliverables

The following list identifies the *Compliance Testing* work items assignment to requirements. All work items are further defined in section 8.23.1 and section 8.23.2 as well as listed in the summary table 8.2 and 8.3 at the end of the chapter.

- Provide a TEAM Engine Virtual Machine free of critical errors after it has been evaluated with the HP software evaluation tool, which requires changes to the current code base. The current implementation of TEAM Engine has been run with the HP Fortify Software Evaluation Tool. About 80 occurrences of "Dead Code" were flagged in the TEAM Engine code base and 50 "XML External Entity Injection" (XXE) vulnerabilities were depicted. The TEAM Engine code shall be fixed to address these issues. Then it shall be delivered as a virtual machine using either VMware vSphere or VirtualBox. The operating system needs to be defined during Testbed-12, it can be either Windows or Unix. Eventually, the TEAM Engine Virtual Machine's deployment in a virtual private cloud, such as Amazon Virtual Private cloud, shall be demonstrated. (A056)

- Advance the architecture of Team Engine and a user interface that allows for the selection of arbitrary combinations of conformance classes, executes the tests and provides test results in an effective way (A057). In addition, the following requirements shall be fulfilled (A057):

  - The application allows for storage (management) of test profiles for consistent repeatable tests.

  - The solution allows to clearly select and report what conformance classes where run, and which ones pass. The information also includes the standard or profile the conformance classes belong to.

  - The solution should work with CTL and TestNG based tests.

  - At least it should be demonstrated for

    * DGIWG WMS 1.3 Profile

    * NSG WMS 1.3 Profile

    * DGIWG WFS 1.1.0 profile

    * DGIWG WFS 2.0.0 profile

- Advance the WFS 2.0 Test Suite (A058)

  - Transactional WFS – Document the extend of this operation and the ability to create and update capabilities. The current test already supports transactions, which is restricted to features.
  - Locking WFS – Document the process for allowing multiple users to update simultaneously. The current test already supports locking.
  - Response Paging – Paging is supported in CSW and Content Discovery and Retrieval (CDR) applications. The test should be advance to be consistent with the WFS 2.0 standard and as much as possible compatible and consistent with CDR. It should also include transactional consistency checking.
  - Standard Joins – The test and reference implementation shall include the case of multi-source integration
  - Spatial Joins – The test and reference implementation shall include the case of multi-source integration.
  - Temporal Joins – The test and reference implementation shall include multi-source integration
  - Feature Versions – The test and reference implementation should test Object Based Production, which is producing feature objects and versioning them.
  - Manage Stored Queries.
- WFS 2.0 Reference Implementation (A059): The WFS 2.0 implementation shall comply with all of the conformance classes and functionality documented above. A reference implementation is an implementation that is fully functional, licensed copy of a tested, branded software that:
  - has passed the test for an associated conformance class in a version of an Implementation Standard and that
  - is free and publicly available for testing via a web service or download.
- WFS 2.0 CITE and Reference Implementation Installation ER (A089)
- Virtual Machine Installation User Guide (A064)
- CITE User Guide with Profile Testing (A088)

8.23 Summary

This section provides an overview over all components, engineering reports, and guides. Most of the components and their corresponding requirements are described in the sections 8.3 to 8.22 above.

8.23.1 Components

The following overview helps understanding all requirements for a given component. In some rare cases, additional requirements are added.

- **A004, Evaluate Security and SOAP, common security extension WFS server**: Common Security Extension, demonstrated by implementing it as part of a WFS server, see 8.3
- **A007, WFS/WCS server**: OGC Web service implementation of a WFS or WCS interface to support the asynchronous service interaction work item, see 8.6.
- **A009-1, Compression Tests WFS Server**: WFS server supporting compression techniques as described in 8.8
- **A009-2, Compression Client**: Client application to interact with the WFS server supporting compression techniques as described in 8.8
- **A010, LiDAR Streaming Client**: Client implementation to support LiDAR data streaming from SOS as described in 8.9
- **A011-1, WFS Synchronization Server 1**: WFS server supporting WFS to WFS synchronization operations as described in 8.10
- **A011-2, WFS Synchronization Server 2**: WFS server supporting WFS to WFS synchronization operations as described in 8.10
- **A012, SOS Compression (LiDAR) Server**: SOS server for LiDAR streaming; supports EXI and WKB and potential other solutions for compressing of LiDAR data streams, see 8.9
- **A016, CSW 2.0.2 with PubSub Core Support Server**, see 8.6
- **A017, Big Data Tile Database Implementation**, see 8.13
- **A018, WPS for Tile Access Server**, see 8.13

- **A019, Web Integration Server**, see 8.14
- **A022, GeoPackage NSG Profile with Vector Tiles Implementation**: GeoPackage with vector tiles, see 8.17
- **A024, GeoPackage Off-Road Routing Data Implementation**: GeoPackge with data for off-road routing apps, see 8.17
- **A025, GeoPackage Symbology Implementation**: GeoPackge with symbols and styles of off-road routing apps, see 8.17
- **A026, Mobile Off-Road Routing App based on GeoPackage Implementation**: Android or iOS based application that propvides off-road routing based on GeoPackge data, see 8.17
- **A027, GeoPackages Mobile App Data Implementation**: GeoPackage with handheld data, includes transportation, hydrography, cultural, terrain, administrative data. The GeoPackage(s) shall support indexed data suitable for OWSContext content descriptors, see 8.17 and 8.18
- **A030, OWS Context Document Production Client Implementation**: client application that produces OWS Context documents that describe among others the content in GeoPackages, see 8.18
- **A031, UGAS support for NAS RDF/OWL/SKOS Implementation**: ShapeChange implementation to support requirements defined in section 8.19
- **A032, 2D Test Dataset Implementation with Documentation**: data set and corresponding documentation to allow reuse of data from Testbed-12 in future testbed activities. Data shall be made available in most appropriate formats (formats to be discussed), see 8.20
- **A033, 2D Test Dataset Symbols and Styles Implementation with Documentation**: symbols and styles set supporting data defined in A032. Includes corresponding documentation to allow reuse in future testbed activities. Symbols and styles shall be made available in most appropriate formats (formats to be discussed), see 8.20
- **A034, WFS SOAP Server**: WFS server supporting SOAP service, implementing the NSG WFS 2.0 Profile, and supporting a pub-sub based solution using the latest draft of OGC Publish/Subscribe Interface Standard 1.0 - Core, see 8.4 and 8.6

- **A035, WFS REST Server**: RESTful WFS Server that leverages existing spatial RDBMS infrastructure to serve geospatial information as Linked Data without loss of performance using the RESTful API and GeoSPARQL query language in designing RESTful service APIs. The use of resolvable URIs for Linked Data and ontologies should be used as best practices in order to facilitate linking and integration of different resources. Further on, the WFS shall support the use of JSON-LD to bridge Linked Data with mainstream web development, see 8.4
- **A036, WFS Server**: supporting data quality, implementing the DGIWG WFS 2.0 Profile and the WFS server v2.5 NSG Application Schema feature specification encoded in GML and in JSON, see 8.6, 8.15, and 8.16
- **A037, WFS-T Server**: WFS-T 2.0, implementing the DGIWG WFS 2.0 Profile. The server shall support (see 8.17.2):
  - Integration in GeoPackage
  - Annotated Imagery
  - Expand the WFS-11 work on multi-media enabling the WFS. Mature the use of GetPropertyValue to retrieve multi-media properties. Investigate options for extracting metadata properties from the media itself. Exif and SMPTE 336M-2007 are standards of particular interest.
    * Multiple codestreams
    * Video encoding (Motion JPEG2000 based)
- **A039, WMS SOAP Server**: WMS supporting SOAP and implementing the NSG WMS 1.3 Profile: 8.3
- **A040, WMS REST Server**: WMS service 1.4, include an encoding for *GetFeatureInfo* responses based on GeoJSON but replacing the geometry part by the location of the position of the query and the position of the returned feature. If returned objects correspond to simple features, return an "id" that allows recovering the geometry using an additional WFS query. See recommendation 7 in OGC 15-053 Implementing JSON/GeoJSON in an OGC Standard ER for further details.
- **A041, WMS Server**: implementing the DGIWG WMS 1.3 Profile

- **A042, WMTS REST Server**: Implement TileJSON for the WMTS Simple profile; implement the WMTS Simple profile to support the two tile matrix sets discussed in section 8.7; and support the tiling Requirements for GeoPackage in section 8.17.2.
- **A043, WCS SOAP Server**: WCS 2.0.1 supporting SOAP and a pub-sub based solution using the latest draft of OGC Publish/Subscribe Interface Standard 1.0 - Core, see 8.3 and 8.6.
- **A044, WCS REST Server**: Elaborate a coverage JSON as a new standard encoding for GMLCov, and implements the requirements to implement REST and JSON for OGC Services as discussed in section 8.4.
- **A045, WCS Conventional Server**: Implementing the DGIWG WCS 2.0 Profile; supports the work on asynchronous service interaction (see 8.6) and conflation (see 8.15).
- **A046, WPS SOAP (asynchronous) Server**: Supporting Web Feature Service Transactional (WFS-T) asynchronous request/response and supporting Web Coverage Service (WCS) asynchronous request/response, see 8.3 and 8.6.
- **A047, WPS REST (conflation) Server**: supporting Social Media, TopoJSON, and interface to/façade Hootenanny to allow for conflation and conflation conflict resolution (see 8.4 and 8.15).
- **A048, WPS Conventional (GeoPackage) Server**: WPS to produce GeoPackage with symbology information and GeoPackage with terrain information, see 8.17.2.
- **A049, WPS Conventional (Data Quality) Server**: WPS supporting ISO Data Quality as discussed in 8.16.
- **A050, CSW 2.0.2 Server**: CSW 2.0.2 ISO Profile, implementing DGIWG Metadata Framework, and supporting enhanced Capabilities (see section 8.12) and asynchronous interaction (see section 8.6. For further details, see section 8.11.
- **A051, CSW ebRIM Server**: implementing semantic mapping between NSG Metadata Framework, DGIWG Metadata Framework, and the Department of Defense (DoD) Discovery Metadata Standard (DDMS) as necessary; and supporting enhanced Capabilities (see section 8.12) and asynchronous interaction (see section 8.6. For further details, see section 8.11.

- **A052, SPARQL / GeoSPARQL Server**: SPARQL Ontology Extension for Portrayal, serves geospatial information as Linked Data without loss of performance using the RESTful API and GeoSPARQL query language (RESTful service APIs), see 8.5.
- **A053, Schema Registry Server**: enables discovery of Schema, Transformation logic, and Ontologies. Initial capability should focus on XML Schema and XSLT transformation logic; see 8.5.
- **A054, Mobile Device GeoPackage Common Map API Implementation**: Mobile device application implementing the Common Map API, shall support GeoPackages from different providers, see 8.17.3
- **A055, Mobile Device GeoPackage Commercial APPs**: Mobile device application that supports the maintenance of GeoPackages, see 8.17.3
- **A056, TEAM Engine – Virtual Machine Implementation**: TEAM Engine Virtual Machine free of critical errors (as identified by HP Fortify software evaluation), see 8.22
- **A057, CITE User Interface Implementation**, see 8.22
- **A058, WFS Test update Implementation**, see 8.22
- **A059, WFS Reference Implementation**, see 8.22

8.23.2 Engineering Report and User Guides

The following overview helps understanding all requirements for a given engineering report or user guide. In some rare cases, additional requirements are added.

- **A005-1, REST Architecture ER**: captures all REST-related discussions, see 8.4
- **A005-2, Javascript, JSON, JSON-LD ER**: captures all discussions around the use of Javascript and JSON-LD, see 8.4
- **A005-3, TopoJSON and GML ER**: captures all REST discussions on data models and geospatial encodings, see 8.4
- **A006, Semantic Summary ER**: aggregates all semantics discussions from other Testbed-12 reports, see 8.4, 8.11, 8.5

- **A008, Vector and Raster Tiling Scheme**: Tiling scheme to support raster and vector tiles, see 8.7
- **A011-3, WFS Synchronization ER**: captures the discussion about service to service synchronization as described in section 8.10
- **A060, REST User Guide**: Implementing a REST based architecture for OGC service types User Guide, see 8.4 and address aspects form the architecture perspective, i.e. focus on REST in contrast to A062, which focuses on JSON
- **A061, SOAP User Guide**: Implementing a OGC SOAP based WFS, WMS and WCS User Guide 8.3, 8.6
- **A062, (Geo)JSON User Guide**: OGC (Geo)JSON User Guide: and address aspects form the architecture perspective, though there is overlap with A060, focus on JSON (in contrast to A060, which focuses on REST from the architectural perspective), see 8.4, 8.18:
  - The User Guide shall include examples and guidance on converting XML documents into (Geo)JSON.
  - Consider OGC 14-009r1 recommendations.
  - Include a subclause for linking to other objects in JSON, using the natural approaches that JSON-LD provides for both simple links and atom links.
  - Include a subclause for describing the process for if a fragment of a XML document contains a geospatial object then when converting to JSON, consider using the GeoJSON equivalent type.
- **A063, OWS Context User Guide**: OWS Context User Guide including descriptions of use with the GeoPackage standard, see 8.4, 8.17.4, 8.18
- **A064, Team Engine VM Installation Guide**: see 8.22
- **A065, OWS Common Security Extension ER**, see 8.3
- **A066, Semantic Portrayal, Registry, Mediation Services ER**: see 8.5 and results from A105, A106 (7.4.3 and 7.4.4), and A053
- **A067, Implementing Asynchronous Service Response ER**: Describing the comparison results of the various approaches to handle asynchronous

response and data delivery for situations where big chunks of data which require asynchronous delivery 8.6

- **A068, Vector Tiling ER**: Discusses all aspect relevant to vector tiling. Related to A008, the Engineering report on the vector and raster tiling scheme, but focuses on the implementation of vector tiles within GeoPackages, see 8.7 and 8.17.1
- **A069, Compression ER**: The Compression Techniques ER shall document the various approaches, highlight pros and cons of the different approaches and compare performance aspects, see 8.8
- **A071, LiDAR Streaming ER**: The ER shall describe the harmonization of the Community Sensor Model with SWE LiDAR. In addition, it shall describe how LiDAR streaming can be realized using the OGC Sensor Web Enablement suite of standards, see 8.9
- **A072, Catalog/SPARQL ER**: Evaluate Catalog Interoperability: catalog comparison, this work item shall load the same data set to a set of catalogs (A050, A051, A103, A104) and test using a multi-catalog client the interaction which each service to better understand interoperability aspects in multi-catalog environments. It shall evaluate the various DCAT contexts including RDF accessible via SPARQL endpoints, embedded in HTML pages as RDFa, or serialized as e.g. RDF/XML or Turtle and compare functionality, expressiveness and usability of CSW and DCAT, see 8.11. In addition, it shall capture all discussion related to the implementation of the SPARQL / GeoSPARQL Server (A052).
- **A073, OWS Context / Capabilities ER**: to describe how OWS Context can be used to supplement the OGC Capabilities document, see 8.4 and 8.12.
- **A074, PubSub / Catalog ER**: describe how the OGC PubSub standard can be used as a mechanism to automatically notify analysts of data availability, see 8.11 and 8.6
- **A075, General Feature Model ER**: describe how the GFM can be re-used for non-geospatial centric applications and/or how it should be extended into a general model for all object types, see 7.3
- **A076, Big Data Database ER**: describes options considered and recommendations for delivery of large amounts of data as a database delivery, see 8.13

- **A077, Multi-Tile Retrieval ER**: to describe options considered and recommendations for delivery of large amounts of tiled data in support of GeoPackage, see 8.7
- **A078, Web Integration Service Recommendations ER**: discusses the results from A107 and A108, see 7.3.
- **A079, WPS Conflation Service Profile ER**: describing the integration of tools such as Hootenanny into spatial data infrastructures. The ER shall describe the developed approach, issues that came up during the integration together with applied solutions and rationales, and suggest future work items to enhance and facilitate the integration of conflation tools into SDIs. This ER shall be written such that it can be nominated as a WPS Conflation Service Profile, see 8.15.
- **A080, Data Quality ER**, Data Quality ER describing data quality handling requirements, challenges and solutions. This ER shall focus on data quality in general that needs to be communicated from one service to another. In addition, it shall discuss WPS data quality solutions. A080 ER shall be written such that it can be nominated as a WPS ISO Data Quality Service Profile, see 8.16.
- **A081, NSG GeoPackage Profile Assessment ER**: describes any concerns with alternative options considered and recommendations for implementation of the NSG GeoPackage Profile, see 8.17.
- **A082, GeoPackage Routing and Symbology ER**: describing symbology and Routing information solutions, see 8.17.2.
- **A083, GeoPackage Mobil Apps Integration ER**: describing all aspects around GeoPackages on mobile clients, such as mobile app integration, GeoPackage shared access, data optimized and suitable for handhelds etc., see 8.17
- **A085, GeoPackage change request evaluations ER**: evaluates existing change requests and determines the way forward 8.17.5.
- **A086, OWS Context**: JSON, JSON-LD and HTML5 ER, describing JSON, JSON-LD and HTML5 solutions as well as how OWS Context can be used as an index descriptor for the content of GeoPackages, see 8.4, 8.17.4 and 8.18.

- **A087, UGAS Shapechange ER**: describes all results from the ShapeChange extensions and tests executed with respect to A031, see 8.19
- **A088, CITE User Guide with Profile Testing ER**, see 8.22
- **A089, WFS 2.0 CITE and Reference Implementation Installation ER**, see 8.22
- **A090, Testbed-12 Web Service Implementations ER**, this engineering report describes and issues recommendations based on the web service implementations in Testbed 12 to include SOAP, REST, Conventional implementations, Profile based implementations, requirement specific implementations. **This work item needs to integrate discussions, results, and best practices from multiple threads!**

The following tables summarizes all work items that shall be delivered as part of this work package.

ID	Funding	Name
A004	funded	Evaluate Security and SOAP, Common Security Extension WFS server
A005-1	funded	REST Architecture ER
A005-2	funded	Javascript, JSON, JSON-LD ER
A005-3	funded	TopoJSON, GML ER
A006	funded	Evaluate Semantic Enablement ER
A007	funded	WCS/WFS Async Server
A008	funded	Vector and Raster Tiling Scheme ER
A009-1	funded	Compression Tests WFS Server
A009-2	funded	Compression Client
A010	funded	LiDAR Streaming Client
A011-1	funded	WFS Synchronization Server 1
A011-2	funded	WFS Synchronization Server 2
A011-3	funded	WFS Synchronization ER
A012	funded	SOS Compression (LiDAR) Server
A016	funded	CSW 2.0.2 with PubSub Core Support Server
A017	funded	Big Data Tile Database Implementation
A018	funded	WPS for Tile Access Server
A019	funded	Web Integration Server
A022	funded	GeoPackage NSG Profile with Vector Tiles Implementation 1
A024	funded	GeoPackage Off-Road Routing Data Implementation
A025	funded	GeoPackage Symbology Implementation
A026	funded	Mobile Off-Road Routing App based on GeoPackage Implementation
A027	funded	GeoPackages Mobile App Data Implementation
A030	funded	OWS Context Document Production Client Implementation
A031	funded	UGAS support for NAS RDF/OWL/SKOS Implementation
A032	funded	2D Test Dataset Implementation with Documentation
A033	funded	2D Test Dataset Symbols and Styles Implementation with Documentation

TABLE 8.2: OGC Baseline Enhancements work package deliverables summary, part 1

ID	Funding	Name
A034	funded	WFS SOAP Server
A035	funded	WFS REST Server
A036	funded	WFS Conventional Server
A037	funded	WFS-T Server
A039	funded	WMS SOAP Server
A040	funded	WMS REST Server
A041	funded	WMS Server
A042	funded	WMTS REST Server
A043	funded	WCS SOAP Server
A044	funded	WCS REST Server
A045	funded	WCS Conventional Server
A046	funded	WPS SOAP (asynchronous) Server
A047	funded	WPS REST (conflation) Server
A048	funded	WPS Conventional (GeoPackage) Server
A049	funded	WPS Conventional (Data Quality) Server
A050	funded	CSW 2.0.2 Server
A051	funded	CSW ebRIM Server
A052	funded	SPARQL / GeoSPARQL Server
A053	funded	Schema Registry Server
A054	funded	Mobile Device GeoPackage Common Map API Implementation
A055	funded	Mobile Device GeoPackage Commercial APPs
A056	funded	TEAM Engine – Virtual Machine Implementation
A057	funded	CITE User Interface Implementation
A058	funded	WFS Test update Implementation
A059	funded	WFS Reference Implementation
A060	funded	REST User Guide
A061	funded	SOAP User Guide
A062	funded	(Geo)JSON User Guide
A063	funded	OWS Context User Guide
A064	funded	Team Engine VM Installation Guide
A065	funded	OWS Common Security Extension ER
A066	funded	Semantic Portrayal, Registry, Mediation Services ER
A067	funded	Implementing Asynchronous Service Response ER
A068	funded	Vector Tiling ER
A069	funded	Compression ER
A071	funded	LiDAR Streaming ER

TABLE 8.3: OGC Baseline Enhancements work package deliverables summary, part 2

ID	Funding	Name
A072	funded	Catalogue, SPARQL ER
A073	funded	OWS Context / Capabilities ER
A074	funded	PubSub / Catalog ER
A075	funded	General Feature Model ER
A076	funded	Big Data Database ER
A077	funded	Multi-Tile Retrieval ER
A078	funded	Web Integration Service ER
A079	funded	WPS Conflation Service Profile ER
A080	funded	WPS ISO Data Quality Service Profile ER
A081	funded	NSG GeoPackage Profile Assessment ER
A082	funded	GeoPackage Routing and Symbology ER
A083	funded	GeoPackage Mobil Apps Integration ER
A085	funded	GeoPackage change request evaluations ER
A086	funded	OWS Context:JSON, JSON-LD and HTML5 ER
A087	funded	UGAS Shape Change ER
A088	funded	CITE User Guide with Profile Testing ER
A089	funded	WFS 2.0 CITE and Reference Implementation Installation ER
A090	funded	Testbed-12 Web Service Implementations ER

TABLE 8.4: OGC Baseline Enhancements work package deliverables summary, part 3

Chapter 9

Compression and Generalization

9.1 Background

In today's developing digital world, data is created and processed at unprecedented volumes. Mechanisms for sharing data need to keep pace if they are to continue to meet requirements. This is particularly evident in the geospatial sector where the increase in big data from a multitude of sensors is combined with the need to share data quickly, potentially over low bandwidth connections. Rather static data sets such as street networks or buildings needs to be integrated with highly dynamic data stemming from real time observation systems such as e.g. traffic monitoring.

As a consequence the delivery of data could be delayed resulting in potentially poor or incorrect decisions being made. For example, in humanitarian disaster situations the delay in sharing data is potentially hindering the success of the operation. As such it is important to optimize the delivery of geospatial data as services.

Optimizing the dissemination of OGC Services across low/very low bandwidths to enable the delivery of faster, lighter and more robust geospatial services is an important part of efficient communication that shall be addressed. In this testbed, it is complemented by generalization techniques that prevent the potentially unaware client from requesting large amounts of data from a service. If the maximum data transfer capacities are known or can be guessed at the time of request reception, generalization algorithms help to balance the trade-off between transferred data volume and information content.

9.2 Requirements and Work Items

The following figure illustrates all work items that shall be addressed in this work package. They are described in further detail in figure 9.1. All funded work items are shaded in green, unfunded in blue. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements.

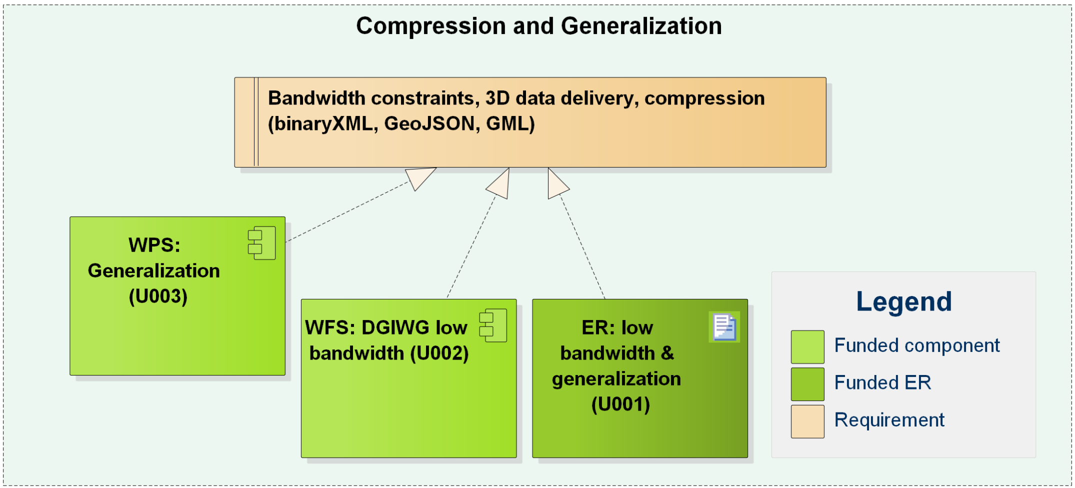

FIGURE 9.1: Compression and Generalization work package: Requirements and work items

9.3 Bandwidth Constraints

The Compression and Generalization work package identifies a single requirement, which includes the optimization of data transfer under bandwidth-constraint conditions using compression techniques, specific data serializations, and server-side applied generalization techniques. The goal is to test bandwidth constraint optimization following two approaches: First, set up a set of dummy services to test and experiment with data size reduction and compression techniques such as the use of JSON over GML, BinaryXML, zipped XML etc. The second approach shall develop techniques to use generalization mechanisms to reduce the amount of data that needs to be transferred from the server to the client; and the other way round, to collect data at one scale only on the client side but serve it at many on the server side. The work package has identified three work items to address these aspects:

- Optimizing the configuration of OGC services to determine the best practice for the efficient delivery of services over low (56kps)/very low (5 kbit/s) bandwidth. Also incorporating the efficient dissemination of complex 3D data.
- Optimizing OGC Services using novel binary compression techniques and data generalization
- Optimizing the dissemination of geospatial feature data by enabling the generalization of geospatial data e.g. collect data at one scale display it at many.

Deliverables

- **Low bandwidth and generalization engineering report** (U001) describing the sample services, outlined the best practice for configuring DGIWG profiles of key OGC services for delivery of both standard data and complex 3D data over very low bandwidths. The report shall further discuss Testbed-12 WFS/WPS implementations and comparing the advantages and disadvantages of disseminating geospatial data using compression techniques Binary XML and GeoJSON.
- **WFS DGIWG low bandwidth (U002)** service demonstrating the compression techniques and low bandwidth performance tests. A dummy service is sufficient to test performances. A client needs to be used to measure performance.
- **WPS service (U003)** demonstrating the generalization of geospatial data services

9.4 Summary

The following tables summarizes all work items that shall be delivered as part of this work package.

ID	Name	Funding
U001	funded	Low bandwidth and generaliization ER
U002	funded	WFS DGIWG low bandwidth implementation
U003	funded	WPS generalization implementation

TABLE 9.1: Compression and Generalization work package deliverables summary

Chapter 10

ArcticSDI and GeoPackage

10.1 Background

Testbed-12 has identified interoperability requirements in the context of the ArcticSDI and the conversion of US Topo Maps to GeoPackage.

National Spatial Data Infratructures (NSDI), such as the US-NSDI or the Canadian Geospatial Data Infrastructure (CGDI) help to gain new perspectives into social, economic, and environmental issues by providing an online network of resources that improves the sharing, use, and integration of information tied to geographic locations. The ArcticSDI complements these perspectives and allows pan-Arctic science; monitoring; and societal, economic, and environmental decision support. In a reciprocal process, developing the ArcticSDI helps to generate a better understanding of how NSDIs can be developed and applied to support Arctic priorities.

By implementing consistent means to share geographic data among all users, costs for collecting and using data can be significantly reduced while decision-making is enhanced. Based on the same design principles, it is the goal to share service interfaces and information models across the various SDI types to ensure maximum interoperability on all scales from local to national to Arctic region to global. To be successful, the ArcticSDI has to take particular requirements into account; such as the unique environment it is located in with zero/low bandwidth at some places, the specific situation of the frontier economy, or the requirement to be expanded to the entire arctic community.

The Arctic SDI provides such an infrastructure and its development is facilitated by the National Mapping Agencies of the eight Arctic countries. The

OGC together with its collaborators Natural Resources Canada, the USGS, territories and states that are part of the Arctic, Arctic Council working groups, Arctic SDI member countries, and a number of OGC member organizations are currently engaged in a OGC Pilot project with the goal to articulate the value of interoperability and to demonstrate the usefulness of standards within the ArcticSDI domain. In its first phase, the ArcticSDI Pilot project develops an inventory of available geospatial Web services across the Arctic, which can be used to reflect a broad range of thematic data layers. In parallel, Phase 1 defines the core components of the ArcticSDI architecture. Both activities serve as the basis for the definition of use cases that shall be implemented to articulate the value of interoperability and to demonstrate the usefulness of standards.

Testbed-12 will use this architecture definition, which in fact is implementing a system of systems approach, and test a number of components and corresponding information models. This activity will further support the architecture implementation tests that are planned for 2016 and beyond.

Further on, Testbed-12 shall investigate the OGC GeoPackage as a single alternative delivery format for the USGS Topo Combined Vector Product and the Topo TNM Style Template. The Topo Combined Vector Product includes point, multi-point, line, and polygon vector feature classes across The National Map (TNM) data themes of Hydrography, Governmental Units (Boundaries), Geographic Names, Elevation Contours, Land Cover polygons, Structures and Transportation. It is delivered in a 7.5-minute footprint in an Esri filegeodatabase (v10.x).

The Topo TNM Style Template is provided by the U.S. Geological Survey (USGS) National Geospatial Technical Operations Center (NGTOC). It has been developed according to the 24,000-scale, 7.5-minute layout and cartographic design of published US Topo Maps and is intended for use in any geographic location where data is available for download from The National Map (TNM). The template provides base topographic map symbology, links to Web Map Services, labeling rules, grids, standard map layout, and marginalia information that a GIS expert can use for advanced analysis and/or for developing tailored maps. The template is formatted as an Esri ArcGIS map document (MXD) and labeling rules are built using the Esri Maplex labeling engine.

10.2 Requirements and Work Items

The following figure illustrates all work items that shall be addressed in this work package. They are described in further detail in figure 10.1. All funded work items are shaded in green, unfunded in blue. Each work item may implement several requirements, i.e. components need to fulfill various requirements, or engineering reports need to address and summarize the results from various requirements.

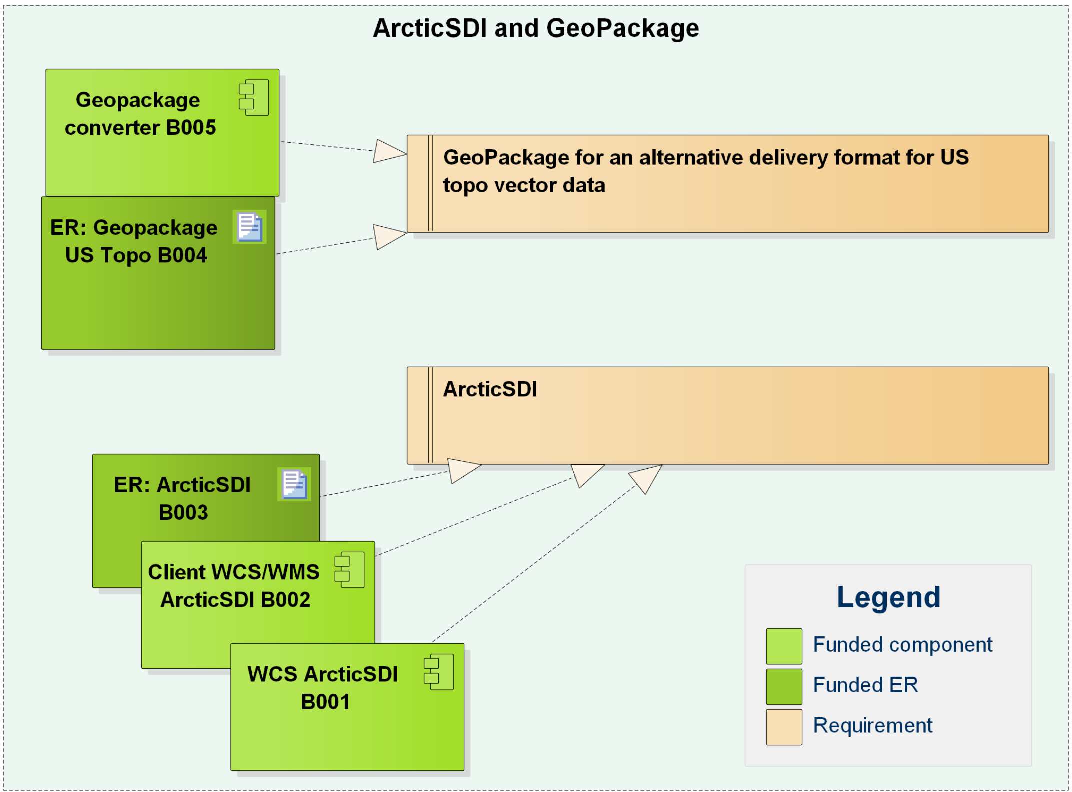

FIGURE 10.1: ArcticSDI and GeoPackage work package: Requirements and work items

10.3 ArcticSDI

Improved access to geospatial data can help us better to predict, understand and react to changes in the Arctic. Responses to the impact of climate change and human activities in the Arctic require accessible and reliable data to facilitate monitoring, management, emergency preparedness and decision-making.

Important data sets are produced and distributed by many stakeholders – public and private sector – and most of it can be geographically referenced. A spatial data infrastructure provides tools for data distributors to ensure that their geospatial data is easier for users to access, validate, and combine with other data.

All those aspects are currently addressed in the "Arctic Spatial Data Infrastructure Standards and Communication Pilot" (short: ArcticSDI Pilot). The goal of the ArcticSDI Pilot is to demonstrate the diversity, richness and value of Spatial Data Infrastructure (SDI) Web services to Arctic SDI stakeholders.

Phase 1 of the ArcticSDI defines the core components of the ArcticSDI architecture, complemented by an inventory of available geospatial Web services across the Arctic. Both activities serve as the basis for the definition of use cases that shall be implemented to articulate the value of interoperability and to demonstrate the usefulness of standards.

Testbed-12 will use this architecture definition, which in fact is implementing a system of systems approach, and test a number of components and corresponding information models. This activity will further support the architecture implementation tests that are planned for 2016 and beyond.

Testbed-12 shall address the following work items:

- Provision of Arctic data that is currently served at WMS interfaces as images only. The data shall be made available using WCS
- Implementation of a client application that can demonstrate the usage of ArcticSDI data, including data that is served at WMS and WCS interfaces
- Analyze if the current Earth Observation profile serves all needs to access ArcticSDI data
- Analyze how data served at OPeNDAP can be integrated
- Analyze how netCDF data can be served efficiently
- Discuss interoperability issues that are specific to the ArcticSDI

Deliverables

- **WCS ArcticSDI (B001)**: A WCS serving Arctic data
- **Client WCS/WMS ArcticSDI (B002)**: Client application supporting WCS and WMS with Arctic data

- **ArcticSDI ER**: Engineering report capturing discussions and results of this task

It is emphasized that all service and client components developed as part of Testbed-12 shall be made available for the entire lifetime of the ArcticSDI pilot, which ends April 2017.

10.4 GeoPackage

Testbed-12 shall investigate the OGC GeoPackage as a single alternative delivery format for the USGS Topo Combined Vector Product and the Topo TNM Style Template. The GeoPackage contents of the Topo Combined Vector Product shall be extended to include imagery and hillshade data. It shall be evaluate whether the point, multi-point, line, and polygon contents in the GeoPackage can be tied directly to a pre-defined symbology set via the Symbology Encoding Implementation Specification.

Symbolized layers within the Topo TNM Style Template can be linked to data downloaded from TNM (the Topo Combined Vector Product). Web Map Service layers in the template include orthoimagery and shaded relief services produced by the USGS, as well as a National Wetlands Inventory (NWI) Web Map Service produced by the US Fish and Wildlife Service. This standard USGS US Topo Map symbology as defined in the Topo TNM Style Template shall be preserved in the GeoPackage container.

The U.S. Geological Survey (USGS) will provide a number of data and products to conduct the Testbed-12 experiments. These include:

1. Topo TNM Style Template
    (a) Esri ArcMap v10.x
    (b) Layer files that are symbolized per US Topo Map specifications
2. One or more Topo Combined Vector Product samples
    (a) Esri File Geodb v10.x
    (b) Vector data layers clipped to 7.5-minute map cell
3. Links to imagery and hillshade services and/or clipped datasets

The following two use cases define the necessary functionality that shall be supported in Testbed-12:

1. State Geologist intends to produce a 24,000-scale map using base map data and symbology in a US Topo Map design and layout. The Geologist will use ArcGIS to create the map add will add her own geologic data layers to the base data. The base data will be downloaded in the OGC GeoPackage and the US Topo Map specified symbology will be encoded with the GeoPpackage.
2. County Emergency Manager would like to obtain symbolized base map data in a US Topo Map design to support an evacuation exercise. The Emergency Manager does not have any sophisticated GIS software. He has an iPad that he takes into the field with him and would like to be able to use a map-like display and query the attributes of the data. In addition, he would like to be able to display a hillshade and local imagery along with the map display.

References

- OGC 15-039 Envisioning a Tiled Elevation Extension for GeoPackage
- OGC 15-012r2 OGC GeoPackage Plugfest
- OGC 12-128r12 GeoPackage Encoding Standard – With Corrigendum #2
- OGC 15-067 Testbed-11 Multi-dimensional GeoPackage Supporting Terrain and Routes Engineering Report

Deliverables

- **GeoPackage US Topo Engineering Report (B004)**: The engineering report shall capture
  - all problems and obstacles with using the USGS USGS Topo Combined Vector Product to create the GeoPackage.
  - all problems and obstacles with encoding the GeoPackage with standard USGS US Topo Map symbology as defined in the Topo TNM Style Template
  - the process used to convert the USGS Topo Combined Vector Product data to GeoPackage format (so that USGS can reproduce the GeoPackage format)

  - the process used to encode GeoPackage data with US Topo Map symbology as defined in the Topo TNM Style Template (so that USGS can reproduce the encoded symbolized GeoPackage format)
  - the process for end users using the Use Cases described. Include any obstacles or limitations to the end users.
- **GeoPackage Converter (B005)**: The GeoPackage converter shall
  - convert the USGS Topo Combined Vector Product data to GeoPackage format (so that USGS can reproduce the GeoPackage format.
  - encode GeoPackage data with US Topo Map symbology as defined in the Topo TNM Style Template (so that USGS can reproduce the encoded symbolized GeoPackage format)

10.5 Summary

The following tables summarizes all work items that shall be delivered as part of this work package.

ID	Name	Funding
B001	funded	WCS ArcticSDI
B002	funded	Client WMS/WCS ArcticSDI
B003	funded	ArcticSDI ER
B004	funded	GeoPackage US Topo
B005	funded	GeoPackage converter

TABLE 10.1: ArcticSDI and GeoPackage work package deliverables summary