Open Geospatial Consortium

Publication Date: 2016-01-25

Approval Date: 2015-09-17

Posted Date: 2015-10-16

Reference number of this document: OGC 15-030r3

Reference URL for this document: http://www.opengis.net/doc/PER/tb11-geo4niem-roundtrip

Category: Public Engineering Report

Editor: Scott Serich

Testbed 11 Geospatial Enhancement for the National Information Exchange Model (Geo4NIEM) Round Trip Engineering Report

Warning

This document is not an OGC Standard. This document presents a discussion of technology issues considered in an initiative of the OGC Interoperability Program. This document does not represent an official position of the OGC. It is subject to change without notice and may not be referred to as an OGC Standard. However, the discussions in this document could very well lead to the definition of an OGC Standard.

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

Contents

Page

Figures

Page

Abstract

The goal of the Geo4NIEM[1] thread in OGC Testbed 11 was to gain Intelligence Community (IC) concurrence of the National Information Exchange Model (NIEM) Version 3.0 architecture through the development, implementations, test, and robust demonstration making use of IC specifications, Geography Markup Language (GML), and NIEM in a simulated "real-world" scenario. The demonstration scenario begins with NIEM-conformant Information Exchange Packages (IEPs) containing operational data and IC security tags from the Information Security Marking (ISM) and Need-To-Know (NTK) access control metadata, and the Trusted Data Format (TDF) for binding assertion metadata with data resource(s). Those instance documents are deployed using Open Geospatial Consortium (OGC) enabled Web Services for access by client applications. Access control is based on attributes of the end-user and the instance data

Recommendations to update these information exchanges were provided to reflect NIEM 3.0 architecture and security tags in a 'NIEM/IC Data Encoding'. The assessment exercised this data encoding in OGC Web Feature Services (WFS) and Policy Enforcement Points (PEP) accessed by multiple client applications. The round-trip assessment also exercised the OGC Transactional Web Feature Services (WFS-T). Results from this task provided a preliminary architecture that was tested and demonstrated in Testbed 11, and summarized in other OGC Testbed 11 Engineering Reports.

Business Value

Geospatial information technologies are increasingly a foundation for supporting homeland security, law enforcement, emergency management, and public safety missions in the U.S. While these technologies often rely on the same data, they are typically developed in silos within a specific mission area. As a result, data duplication and data exchange delays occur.

In addition, many Information Sharing Environment (ISE), Homeland Security (HLS) and Law Enforcement (LE) mission partners have developed stand-alone geospatial information systems (GIS) or Common Operating Picture (COP)/Situational Awareness (SA) applications to support their stakeholder communities during incidents and for daily operational support. While different missions, these GIS or COP/SA capabilities rely upon much of the same data or generate specific data during an event. The data are often stove-piped and not exposed to a broader community that could benefit from these data, resulting in duplication and delayed or incorrect decisions. While mission partners do not need to use the same GIS or COP/SA tools, they could benefit from shared access to the

[1] http://www.ise.gov/mission-stories/geo4niem

common operating data and services used within these systems if they were exposed and exchanged using open standards.

To meet this challenge, the Program Manager for the Information Sharing Environment[2] (PM-ISE) is funding work to enhance NIEM. One focus of these efforts is to enhance NIEM's geospatial exchange capabilities to improve inter-government information sharing. Validating and testing the NIEM (Version 3.0) technical architecture related to the IC Data Encoding Specifications (i.e. security tags such as ISM, NTK, and TDF), aligned to OGC Web Services was identified as a need. Specifically, if the framework's geospatial exchange capability is enhanced with security and standards issued by the OGC inter-government information sharing will be significantly improved.

Keywords

ogcdocs, testbed-11, Geo4NIEM, NIEM, WFS, WFS-T, GML, PEP, security, access control, ISM, NTK and TDF

[2] http://www.ise.gov/

Testbed-11 Geospatial Enhancement for the National Information Exchange Model (Geo4NIEM) Round Trip Engineering Report

1 Introduction

1.1 Scope

The focus of the Geo4NIEM thread in OGC Testbed 11 was to assess the potential for security tagging and access control from IC Data Encoding Specifications to be combined with NIEM for information exchange. The purpose was to determine if the current NIEM architecture can be aligned with the IC Data Encoding Specifications, which include (but are not limited to) ISM, NTK and Trusted Data Format (TDF). This alignment would enable secure information exchange and enhance user/developer understanding. The assessment included review of real world data exchanges defined in the form of a NIEM Information Exchange Package Documentation (IEPD). A number of Extensible Markup Language (XML) instance documents from those real-world exchanges, populated with operational data and IC security tags, were deployed on OGC Web Services for testing. The test scenarios included use of OGC Transactional Web Feature Service (WFS-T) to demonstrate a round-trip scenario.

This effort builds on the previous work of the Geo4NIEM Pilot Project[3]. Much of the work was focused on the GML (ISO 19136) data exchange standard and the mechanisms by which GML and NIEM data could be intermingled. A key driver was to clarify how data conforming to one framework could be included or "embedded" in the other using various encapsulation strategies. A secondary goal was to conduct various software demonstrations in order to assess the feasibility of the various approaches and to explore the prospects for making use of fundamental OGC web services such as WFS.

Based on the results of the Geo4NIEM Pilot the sponsors of the Geo4NIEM thread in Testbed worked with OGC staff to articulate specific functional requirements in order to meet the following objectives:

- Validating the NIEM (Version 3.0) technical architecture related to the IC Data Encoding Specifications (i.e. ISM, NTK, and TDF).
- Testing and demonstrating use of 1) NIEM 3.0 architecture, access control, and security tagging metadata defined by IC Data Encoding Specifications; and 2) full

[3] https://www.niem.gov/technical/Pages/Geo4NIEM.aspx, http://www.opengeospatial.org/projects/initiatives/geo4niem

round tripping of NIEM-conformant information exchanges to GML feature(s) and back to a NIEM-conformant information exchange.

- Testing and demonstrating use of an application programming interface (API) for operating primarily on GML feature representations leveraging NIEM components; features may be searched, retrieved, inserted, updated, and deleted.

- Reviewing and documenting recommendations to enable full round tripping from NIEM-conformant information exchange to Geography Markup Language (GML) feature(s) and back to NIEM-conformant information exchange.

To accomplish these objectives, five primary tasks were identified:

Task 1: *NIEM & IC Data Encoding Specification Assessment and Recommendations*

This task assessed the potential for security tagging and access control from the IC Data Encoding Specifications to be leveraged with NIEM in support of information exchange. The purpose was to determine if the current architecture of NIEM can support IC specification alignment. The IC Data Encoding Specifications include but are not limited to ISM, NTK, and TDF metadata.

The assessment included review of real world IEPDs, where the Extensible Markup Language (XML) schema and NIEM instance documents were populated with relevant content and IC security tags. IEPDs assessed included:

- Notice of Arrival IEPD[4]

- Incidents IEPD

- Resources IEPD

Recommendations to update these information exchanges were provided to reflect NIEM 3.0 architecture and included sample security and dissemination control markings. The assessment exercised OGC standards enabled web services to test NIEM Version 3.0 conformant IEPDs containing the appropriate IC security markings. Results from this task provided a preliminary proposed architecture structure that was tested and demonstrated in Task 2.

This task produced one document:

- Testbed 11 NIEM IC Data Encoding Specification Assessment and Recommendations ER

[4] For an example: https://mise.mda.gov/drupal/node/24

Task 2: *NIEM & IC Data Encoding Specification Test and Demonstration*

This task used preliminary findings and recommended architectures for IC Data Encoding Specification support identified in Task 1, and performed a Test and Demonstration of the recommended architecture leveraging the results of Testbed 9 and previous Geo4NIEM initiatives where appropriate. Results of this task provided updates to the proposed architecture prepared in Task 1.

Results of this test and demonstration were documented in an Engineering Report containing the Findings and Recommendations with reference to refinements to the originally proposed architecture prepared in Task 1.

This task produced one document:

- Testbed 11 Results of Test and Demonstration of NIEM Using IC Data Encoding Specifications ER

Task 3: *NIEM-GML-NIEM Round-trip Assessment and Recommendations*

This task assessed the NIEM and GML support for geospatial data exchange round-trip workflow processes to include: creation, transfer, receipt, modification, return, and acceptance of XML content originating as NIEM IEPDs.

This task produced one document:

- Testbed 11 NIEM-GML-NIEM Round Trip Assessment and Recommendations ER

Task 4: *NIEM-GML-NIEM Round-trip Test and Demonstration*

This task used the findings and recommended architecture structure supporting NIEM-GML-NIEM round-trip assessment identified in Task 3 and performs a Test and Demonstration of the recommended architecture.

This task produced one document:

- Testbed 11 NIEM-GML-NIEM Round Trip Assessment and Recommendations ER

Task 5: *Test and Demonstration of an API for Processing GML Feature Representations*

This task performed Test and Demonstrations using OGC web services, such as Basic and Transactional Web Feature Service (WFS-T) and PEPs, to process GML feature representations leveraging NIEM components. The Test and Demonstration included, but are not limited to feature retrieval, insert, update and delete.

This task produced one document:

- Testbed 11 NIEM-GML Feature Processing API using OGC Web Services ER.

1.2 Sponsoring and Participating Organizations

1.2.1 Sponsoring Organizations

Geo4NIEM in Testbed 11 was sponsored by the following organizations:

- US Department of Homeland Security (DHS)

1.2.2 Participating Organizations

The following organizations played one or more roles in Geo4NIEM in Testbed 11 as participants (i.e., responded to the RFQ/CFP)

- The Carbon Project
- Secure Dimensions
- con terra
- Jericho Systems

This document also integrates comments and content from MITRE and Safe Software.

1.3 Document contributor contact points

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

Name	Organization
Scott Serich	Open Geospatial Consortium
Jan Drewnak	con terra
Rüdiger Gartmann	con terra
Jeff Harrison	The Carbon Project
Dean Hintz	Safe Software
Andreas Matheus	Secure Dimensions
Mark Mattson	The Carbon Project
Scott Renner	MITRE
Tim Schmoyer	Jericho Systems

Many thanks are extended to the reviewers who submitted comments over the course of the project.

1.4 Future work

Improvements in this document are desirable and will be included based on ongoing interoperability engineering activities.

1.5 Foreword

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

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

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- *Intelligence Reform and Terrorism Prevention Act of 2004* (IRTPA)
- *Guidelines and Requirements in Support of the Information Sharing Environment*, Presidential Memo, December 2005.
- Open Geospatial Consortium (OGC), Summary and Recommendations of the Geospatial Enhancement for the National Information Exchange Model (Geo4NIEM) Interoperability Program Pilot (http://www.opengeospatial.org/standards/per)
- Open Geospatial Consortium (OGC), Geography Markup Language (GML) Encoding Standard (http://www.opengeospatial.org/standards/gml)
- Open Geospatial Consortium (OGC), Web Feature Service (WFS) (http://www.opengeospatial.org/standards/wfs)
- Open Geospatial Consortium (OGC), Filter Encoding Implementation Specification (http://www.opengeospatial.org/standards/filter)
- Intelligence Community (IC) Data Encoding Specifications (http://www.dni.gov/index.php/about/organization/chief-information-officer/ic-cio-enterprise-integration-architecture)
- IC Enterprise Authorization Attribute Exchange between IC Attribute Services, Authorization Attribute Set (http://www.dni.gov/index.php/about/organization/chief-information-officer/idam-authorization-attribute-set)

- XML Data Encoding Specifications for Information Security Marking Metadata (http://www.dni.gov/index.php/about/organization/chief-information-officer/information-security-marking-metadata)
- XML Data Encoding Specification for Need-To-Know Metadata (http://www.dni.gov/index.php/about/organization/chief-information-officer/need-to-know-metadata)
- XML Data Encoding Specification for Trusted Data Format (http://www.dni.gov/index.php/about/organization/chief-information-officer/trusted-data-format)
- NIEM Version 3.0 (http://release.niem.gov/niem/3.0)
- NIEM.gov (http://www.niem.gov)
- Open Geospatial Consortium (OGC), Web Services Common Implementation Specification Version 2.0.0 (http://www.opengeospatial.org/standards/common) [OGC 06-121r9]

NOTE The OWS Common Standard contains a list of normative references that are also applicable to this Implementation Standard.

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply.

3.1 Abbreviated terms

DES	Data Encoding Specification
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL/TLS
IC	Intelligence Community
IEP	Information Exchange Package
IEPD	Information Exchange Package Documentation
ISM	Information Security Markings
MDA	Maritime Domain Awareness
NIEM	National Information Exchange Model
NTK	Need to Know
OGC	Open Geospatial Consortium

OWS	OGC Web Services
PDP	Policy Decision Point
PEP	Policy Enforcement Points
PM-ISE	Program Manager for the Information Sharing Environment
RFC	Request For Comments
SSL	Secure Sockets Layer
TDF	Trusted Data Format
TDO	Trusted Data Objects
TLS	Transport Layer Security
UAAS	Unified Attribute and Authorization Service
UIAS	Unified Identity Attribute Set
WFS	OGC Web Feature Service
WFS-T	OGC Web Feature Service – Transactional
XLink	XML Linking Language
XML	Extensible Markup Language

4 Geo4NIEM Round Trip Testing

In the Testbed 11 Geo4NIEM thread, participants assessed security and dissemination control markings leveraging the TDF, ISM and NTK IC Data Encoding Specifications, and how to provide appropriate access control to NIEM IEPs served through a WFS server, and a round-trip workflow process that included creation, transfer, receipt, modification, return, and acceptance of XML content originating as NIEM IEPs.

The assessment was conducted by implementing prototype components that use NIEM/IC Data Encodings in a functional test environment. Access control was conducted via one of several Policy Enforcement Points that filter based upon the user attributes stored in the OGC Attribute Store. The downstream activities of the round-trip workflow process were implemented through transactions against a WFS-T server.

4.1 Composition of the TDO and Conversion to a WFS Feature Collection

The Testbed 11 Round Trip scenario depended on the existence or creation of a Trusted Data Object (TDO) available for retrieval by components in the Geo4NIEM Testbed Architecture. A representation of the TDO makeup is provided in the following figure.

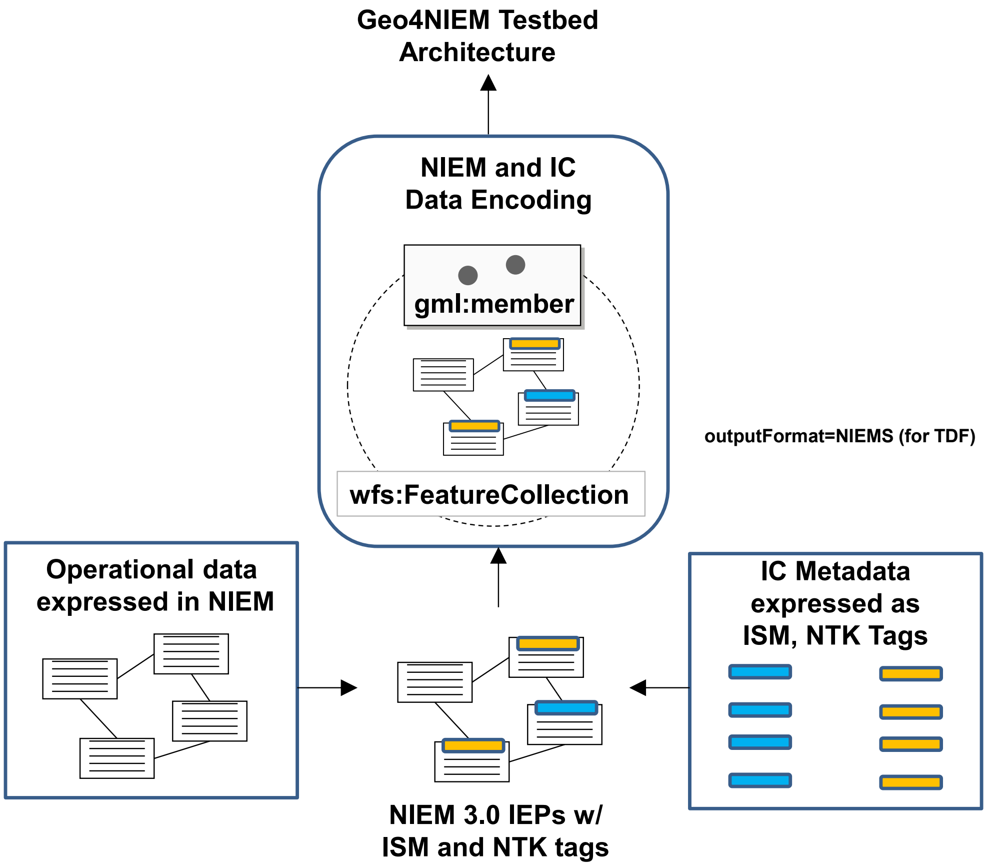

Figure 1 - Converting NIEM IEP with ISM/NTK tags into wfs:FeatureCollections

Operational data expressed as a NIEM 3.0 IEP was augmented with IC metadata expressed as ISM and NTK security tags, a capability that was enabled with the release of NIEM 3.0. The IEP contained operational data such as a notice of arrival (maritime domain) or an incident or resource (incident management domain). These data were wrapped as the TDO payload.

The handling assertions provided a place in the TDO for additional IC security metadata, including the ntk:Access element. In cases where the IEPs were to be released unwrapped and there were NTK restrictions, the IEPs had to include the ntk:Access element. In this case, the producing system took the affirmative step of including the Access element. In the Testbed 11 Round Trip test, this was readily demonstrated in a Notice of Arrival sample IEP.

4.2 High-Level Round Trip Testbed Architecture

A high-level overview of the Geo4NIEM prototype test environment is provided in the figure below. Details on the environment and test results are provided in separate Engineering Reports.

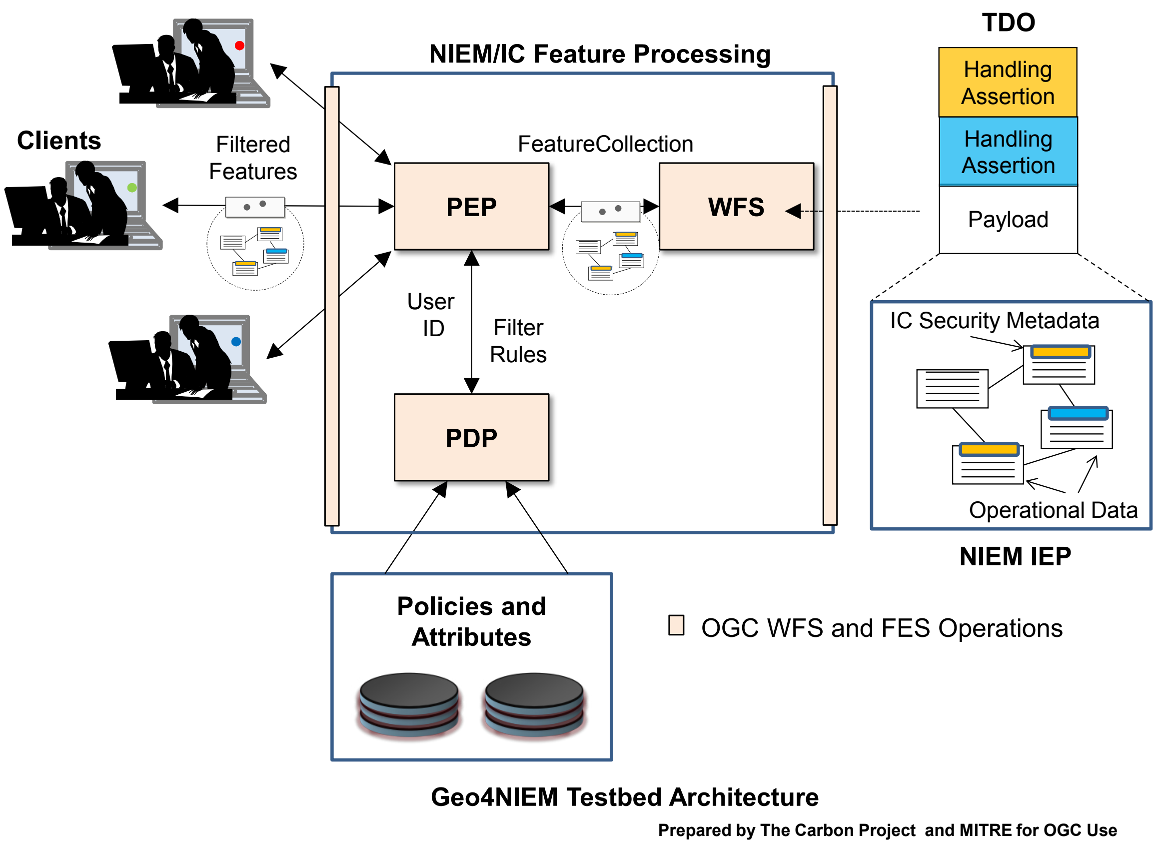

Figure 2 - Geo4NIEM Testbed Architecture[5]

This architecture supported the entire sequence of messages making up a Testbed 11 Round Trip. For example, featureCollection retrieval was carried out using the following steps.

- A Round Trip TDO was pre-loaded into a WFS server (represented on the right side of the figure).
- Clients (left side) would then authenticate themselves and then issue a getFeature request to the PEP, which was acting as a WFS proxy.
- The PEP passed the getFeature request to the WFS and received a featureCollection. The PEP then passed the user identity to the PDP, which performed a look-up on the user attributes. It also looked up the access control policies. From the user-attribute and policy findings, the PEP created a filtering rule expressed in terms of data attributes (i.e., the ISM and NTK security metadata attributes).

[5] User attributes created to support the Geo4NIEM Testbed 11 architecture were extended from the IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set (UIAS) to support fine-grained access control using NTK.

- The filter rule was then passed to the PEP, where it was applied to the featureCollection. In some cases the rule would result in the complete removal of a member from the collection. In other cases, it would redact information from a collection member.

- The filtered featureCollection was then returned to the client.

4.3 The Testbed 11 Round Trip Message Sequence

The full Round Trip scenario was comprised of four message pairs associated with a sequence of four invocations of an OGC Web Feature Service (WFS):

1. GetCapabilities() :service metadata

2. DescribeFeatureType :type description

3. GetFeature() :feature instances

4. Transaction() :InsertUpdateDeleteFeatures

A representation of the full message sequence is provided in Figure 3.

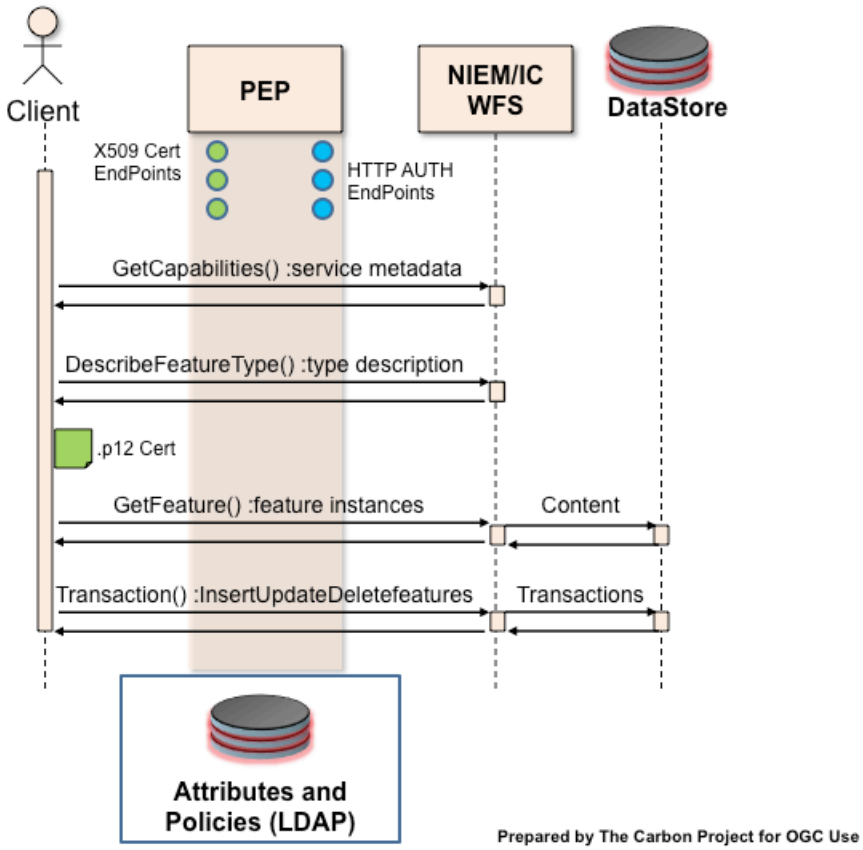

Figure 3 — The Testbed 11 Round Trip Message Sequence

A cloud-based test environment was established by The Carbon Project to broadly support the testing of this sequence. Multiple participants, including con terra, Jericho Systems, and Secure Dimensions, provided Policy Enforcement Points (PEPs) that accessed the WFS through a Feature Processing API on the cloud-based test environment. Multiple client applications were implemented to test connection to the PEP-NIEM/IC services including Gaia, QGIS, FME, and a new Geo4NIEM Web Client developed by The Carbon Project.

5 Examples Demonstrating the Round-Trip Workflow

This section provides examples demonstrating the Round Trip workflow. These examples provide a very brief, sample overview of the demonstration scenario. For a complete description please see the Test and Demonstration ER (*15-050 Testbed 11 Results of Test and Demonstration of NIEM Using IC Data Encoding Specifications ER*) and the actual Testbed 11 Geo4NIEM Demonstration videos.

As illustrated in the following figures, the basic workflow was triggered by a user request for feature data regarding a geolocated object of interest.

Figure 4 - Trigger for the Round Trip Workflow

As the next two figures show, a request will be served by actual data only for authorized users. A user without authorization will be denied access.

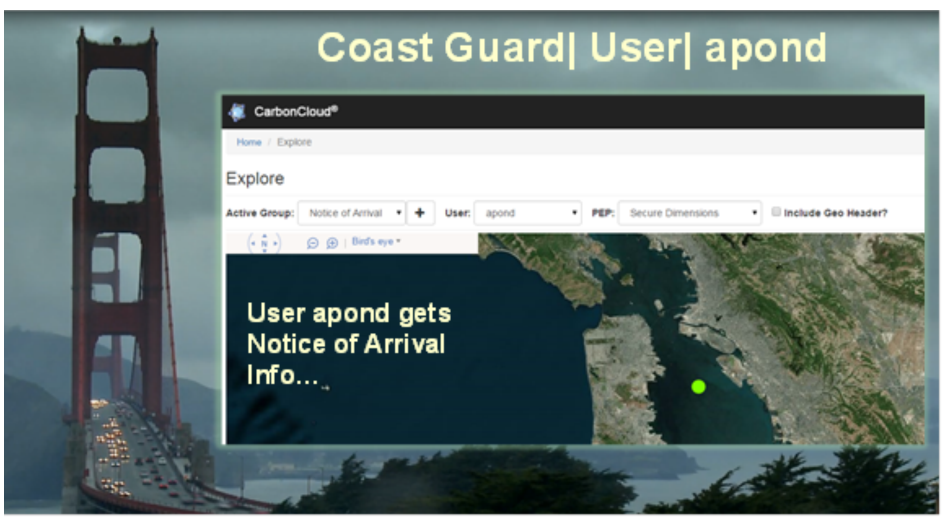

Figure 5 - Response to an Authorized User

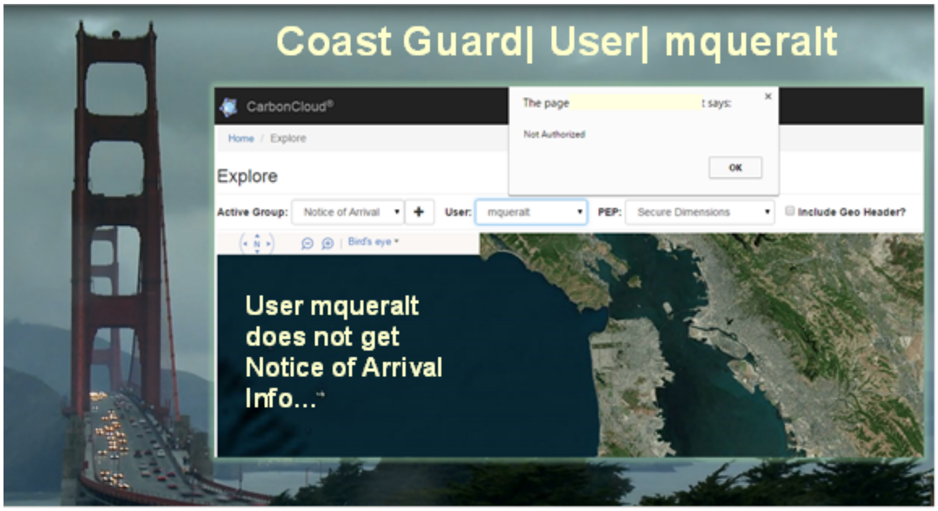

Figure 6 - Response to a Non-Authorized User

The basic flow of events in the demonstration was:

1. Clients authenticate.
2. Clients issue getCapabilities and getFeature requests to the PEP, which is acting as a WFS proxy.
3. The PEP passes a getFeature request to the WFS, receives a featureCollection.
4. The PEP passes the user identity to the PDP.
5. The PDP looks up user attributes and related access control policies.
6. The PDP creates a filter rule from the user attributes and access control policies. The rule is expressed in terms of data attributes; i.e. ISM and NTK security and access control metadata attributes.
7. The PDP forwards the filter rule to the PEP
8. The PEP applies the filter rule to the featureCollection. Sometimes the rule completely removes a member from the collection. Sometimes it redacts the information that is in a collection member.
9. The filtered featureCollection is returned to the client. Note – the effort demonstrated two different ways of handing NIEM IEPs that contain embedded GML data: with and without the TDO wrapper. In both cases the *feature* is the NIEM IEP. The TDO wrapper is treated as an *output format*.

10. Based on some user action, the client can send a WFS Transaction operation to the PEP, which is still acting as a WFS proxy for the WFS server.

11. The PEP executes authenticate and filtering steps again, this time to see if the policy permits the user to make the transaction request. The PEP lets the request pass through or it doesn't.

12. Based on the WFS-T request, the WFS server updates its internal state.

5.1 Round Trip Components from The Carbon Project

The Carbon Project implemented the NIEM/IC Data Encoding in OGC WFS and multiple client applications, including a new web client developed for Testbed 11. The Web Feature Service (WFS) provided NIEM/IC Data Encoding as wfs:FeatureCollections to multiple Policy Enforcement Point (PEP) services. In addition, the WFS provided NIEM/IC Data Encoding directly to client applications such as Gaia shown below with symbolized Incident wfs:FeatureCollections and Notice of Arrival content. It should be noted that Gaia represents an older geospatial application. The Carbon Project also developed new web clients able to access the NIEM/IC Data Encoding via PEP from Secure Dimensions, con terra and Jericho Systems, and NIEM/IC WFS from The Carbon Project. An example of this new web client for NIEM/IC is shown in the second figure below.

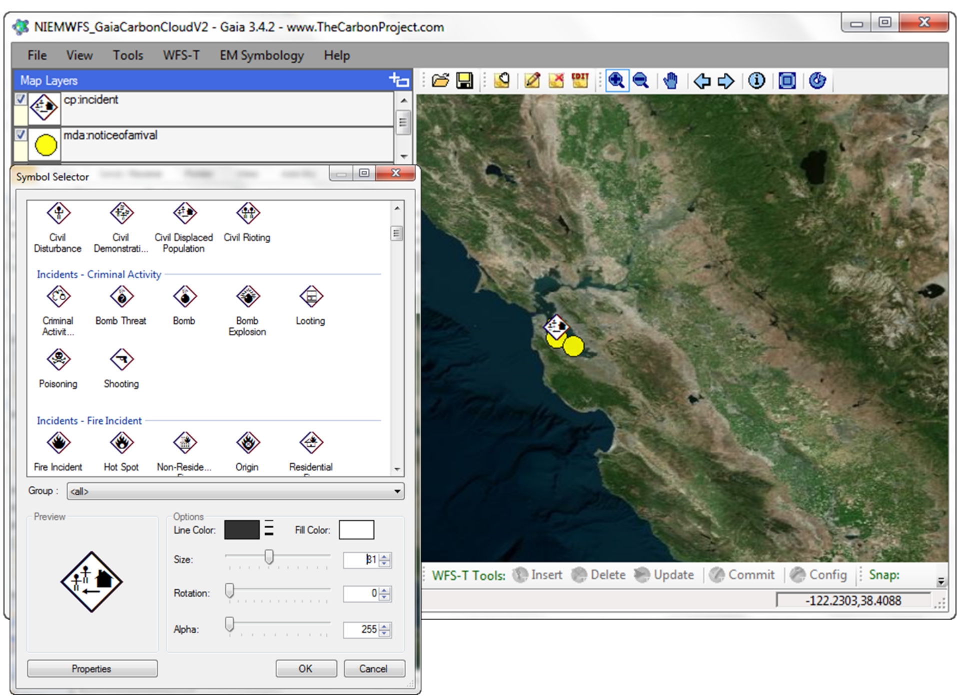

Figure 7 – Incident and Notice of Arrival content from The Carbon Project NIEM/IC WFS in Gaia

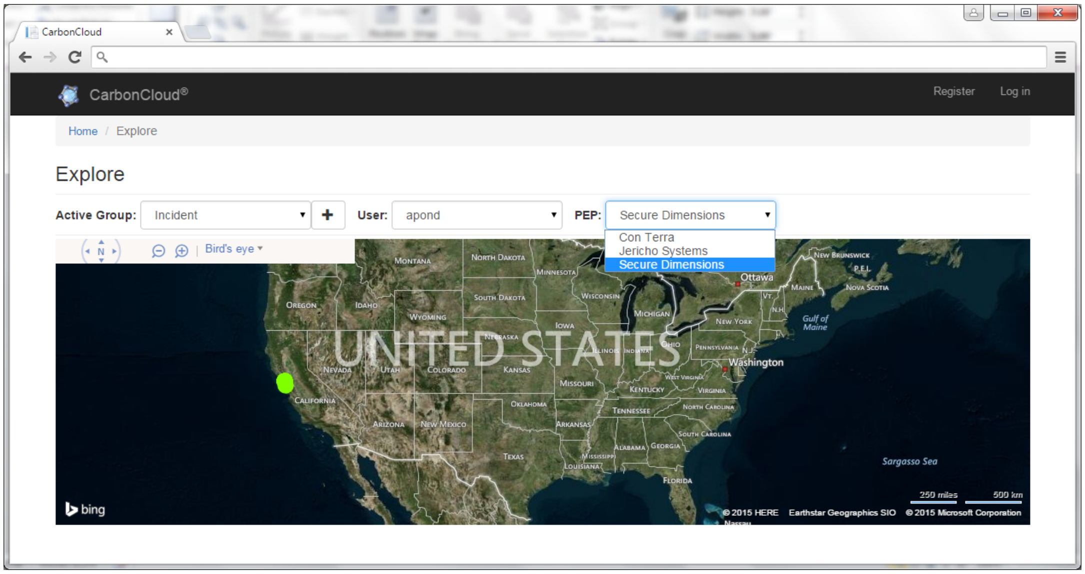

Figure 8 - Web Client from The Carbon Project accessing NIEM/IC Data Encoding from Secure Dimensions, con terra and Jericho Systems PEP

5.2 Round Trip Components from Secure Dimensions

Examples of the Secure Dimensions PEP and The Carbon Project NIEM/IC web client are provided below.

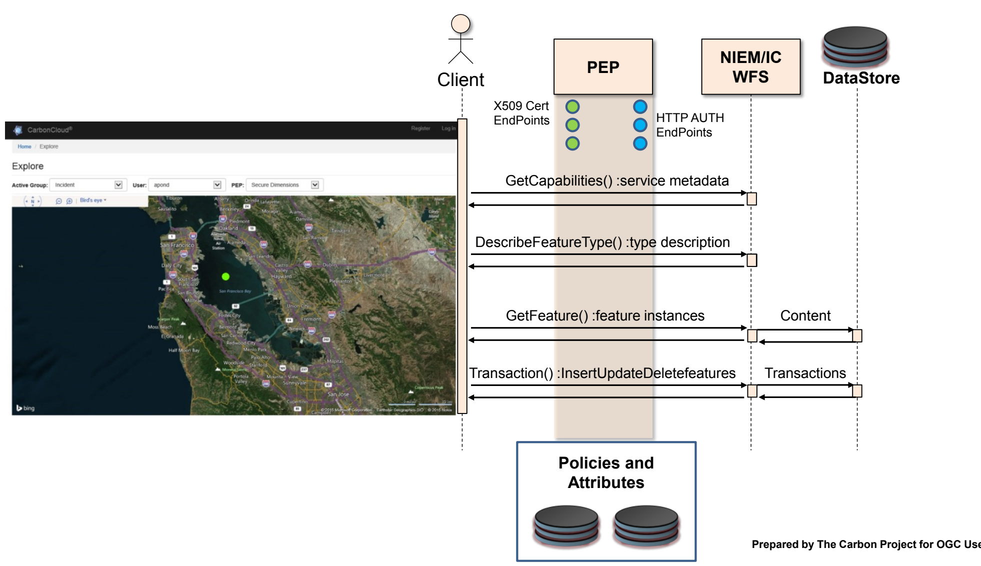

Figure 9 - Secure Dimensions PEP in The Carbon Project web client

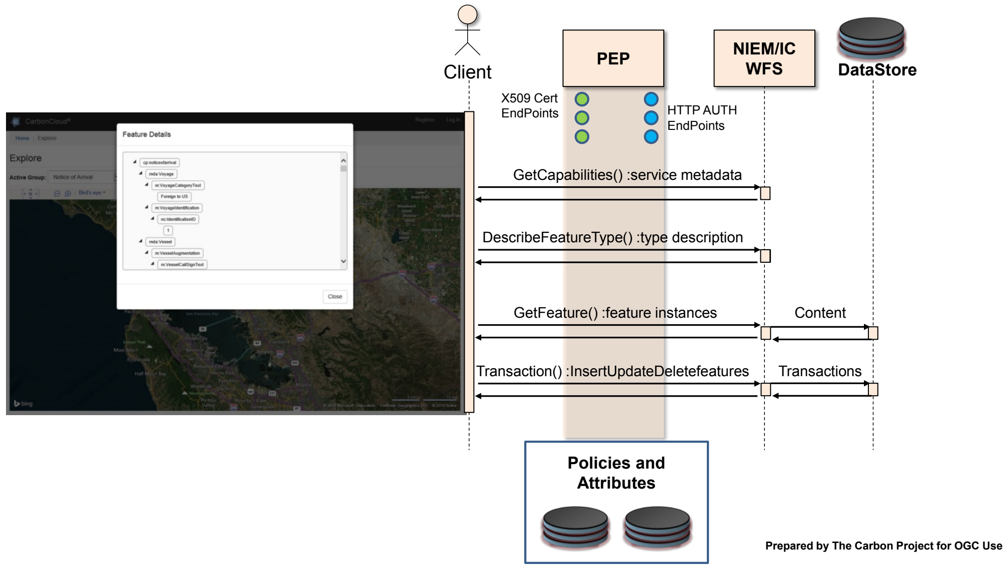

Figure 10 - Secure Dimensions PEP in The Carbon Project web client, feature detail displayed

5.3 Round Trip Components from con terra

con terra implemented the NIEM/IC Data Encoding in PEP services. Examples are provided below.

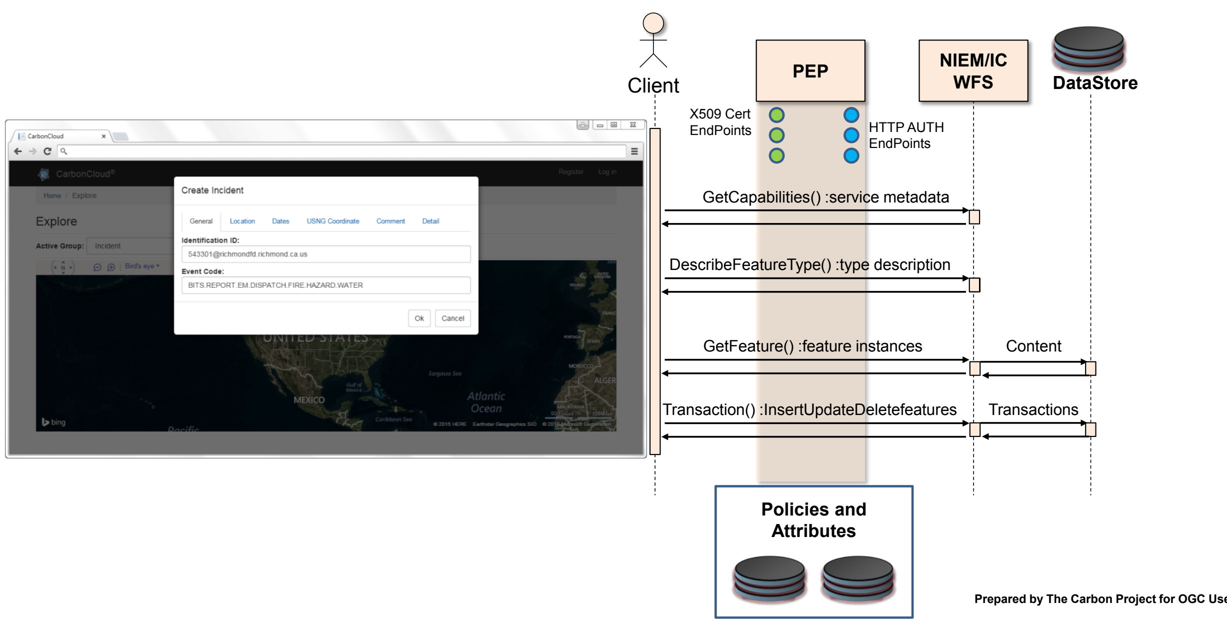

Figure 11 – con terra PEP in The Carbon Project web client, executing WFS Transactions

5.4 Round Trip Components from Jericho Systems

Jericho Systems implemented the NIEM/IC Data Encoding in PEP services. Examples are provided below.

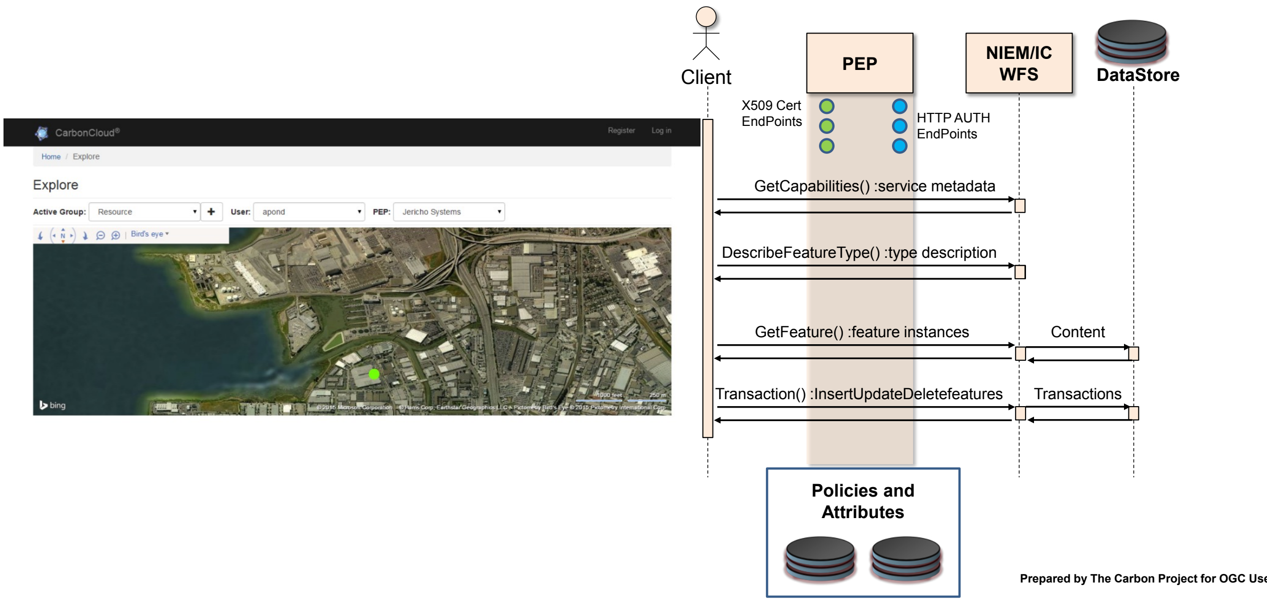

Figure 12 – Jericho Systems PEP in The Carbon Project web client, accessing Resource encoding

5.5 Round Trip Components from Safe Software

To support GML round trip testing for the Test Bed 11 Geo4NIEM thread, Safe Software provided a set of client and data transformation components based on its FME software. FME, often described as a spatial ETL tool, represents Safe Software's suite of data conversion, transformation and integration tools focused on managing the exchange of spatial and non-spatial data between systems with differing file formats and structures. FME also has capabilities to transform feature geometries, attribute tables, and coordinate systems to support data harmonization across disparate systems. Safe provides data integration and consumption components based on FME to many leading GIS and CAD vendors, such as the Data Interoperability Extension for ArcGIS. The 2 main components used for GML round trip testing were FME Data Inspector, for consuming WFS and reading GML, and FME Workbench for authoring data conversion and transformation workflows, performing XML validation tests, and modifying the No Schema custom GML reader.

FME Data Inspector is the primary FME tool for viewing and interrogating datasets and can read any FME supported format. This includes over 350 GIS, CAD, raster, point cloud, 3D, BIM, XML, JSON, web, database and tabular formats, along with comprehensive support for many OGC GML and web services formats (GML, CityGML, WMS, WFS, AIXM, WXXM, etc). Data Inspector also allows the user to overlay disparate data sources, and a background map feature is provided so that web mapping services can be used to provide context. Full inspection capabilities allow any complex attribute and geometry model to be fully explored. Data Inspector can also be configured to support a variety of security architectures. For the Geo4NIEM thread it was successfully tested with both HTTP BASIC authorization and X509 in the context of using the WFS client / reader.

5.5.1 Reading Geo4NIEM GML with a Schema-Based GML Reader

A number of tests using FME were performed in an attempt to read the sample dataset "LOATrackSample.xml" provided by the sponsor. The first set of tests involved reading this dataset using the GML application schema "LOA Tracks.xsd."

To begin with, the encoding was varied from one schema document to another. For example:

Niem-core.xsd has:

```
<?xml version="1.0" encoding="US-ASCII"?>
```

Maritime.xsd:

```
<?xml version="1.0" encoding="US-ASCII"?>
```

Mda.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
```

LOATracks.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
```

A mixture of different encodings in the same schema dependency tree could cause problems depending on the operating system and parser being used. For consistency, it would be best if they were all UTF-8.

As before, problems were encountered with missing schemas.

For example:

```
XML Parser error: 'Error at file:'' line-0 column:0 message:unable to open primary document entity
'Geo4NIEM\GMLRoundTrip\SSATF witn NIEM\SSATF
v3\imp\base\schemas\base\3\StyledLayerDescriptor.xsd'

XML Parser error: 'Error at file:'' line-0 column:0 message:unable to open primary document entity
'Geo4NIEM\GMLRoundTrip\SSATF witn NIEM\SSATF%20v3\ISM\IC-ISM.xsd'

XML Parser error: 'Error at file:'' line-0 column:0 message:unable to open primary document entity
'Geo4NIEM\GMLRoundTrip\SSATF witn NIEM\SSATF%20v3\NTK\IC-NTK.xsd
```


Also notice that some of these paths used ‘SSATF%20v3’ and some ‘SSATF v3’. For example, from LOA Tracks.xsd we see:

```
<xs:import namespace="http://metadata.dod.mil/mdr/ns/gcss/ssatf/3.0"

  schemaLocation="SSATF%20v3/ssatf-03/base/ssatf.xsd"/>
```

This problem seems to be limited to the sample dataset and schema provided (LOA Tracks.xsd). No spaces were encountered in the schema import paths within the NIEM schemas that were reviewed (note that not all were checked).

There are also multiple references to the same schemas at different locations. For example, IC-NTK.xsd is referenced by NIEM\LOA\xsd\extension\mda.xsd as:

```
<xs:import namespace="urn:us:gov:ic:ntk" schemaLocation="../external/NTK/IC-NTK.xsd"
appinfo:externalImportIndicator="true"/>
```

SSATF.xsd has:

```
<xs:import namespace="urn:us:gov:ic:ntk" schemaLocation="../../NTK/IC-NTK.xsd"/>
```

The highly nested and complex directory structure used to deploy NIEM makes it a challenge to configure tools just to read / interpret the data. For example, consider:

SSATF witn NIEM\NIEM\LOA\xsd\niem\domains\maritime\3.0 \maritime.xsd

```
 <xs:annotation><xs:documentation>Maritime</xs:documentation> </xs:annotation>

  <xs:import schemaLocation="../../../adapters/geospatial/3.0/geospatial.xsd"
namespace="http://release.niem.gov/niem/adapters/geospatial/3.0/"/>

  <xs:import schemaLocation="../../../codes/dot_hazmat/3.0/dot_hazmat.xsd"
namespace="http://release.niem.gov/niem/codes/dot_hazmat/3.0/"/>

  <xs:import schemaLocation="../../../codes/iso_3166-1/3.0/iso_3166-1.xsd"
namespace="http://release.niem.gov/niem/codes/iso_3166-1/3.0/"/>

  <xs:import schemaLocation="../../../codes/unece_rec20/3.0/unece_rec20-misc.xsd"
namespace="http://release.niem.gov/niem/codes/unece_rec20/3.0/"/>

  <xs:import schemaLocation="../../../niem-core/3.0/niem-core.xsd"
namespace="http://release.niem.gov/niem/niem-core/3.0/"/>

  <xs:import schemaLocation="../../../proxy/xsd/3.0/xs.xsd"
namespace="http://release.niem.gov/niem/proxy/xsd/3.0/"/>

  <xs:import schemaLocation="../../../structures/3.0/structures.xsd"
namespace="http://release.niem.gov/niem/structures/3.0/"/>
```

Note how the imports refer to schemas 3 levels above and 3 levels down a neighboring tree structure. Some type of nested structure is needed for schema organization. However, having a separate folder for each application schema file may be overcomplicating this somewhat. In the future, perhaps consideration could be given to consolidating this schema store where possible.

Given the above issues, FME was not initially able to read the NIEM GML provided in the test package. To diagnose the problem, a validation FME workspace based on the XMLValidator was used - see the following figure. This transformer invokes the open source Apache Xerces XML parsing library (http://xerces.apache.org/). It is a set of libraries for parsing, validating, serializing and manipulating XML.

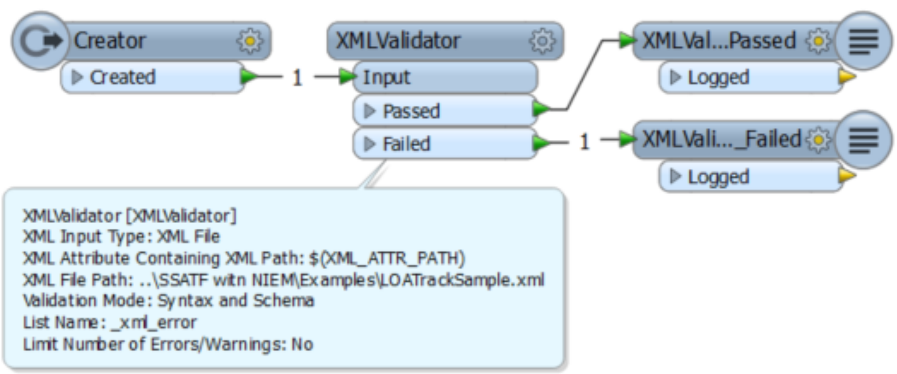

Figure 13 - FME Validation Workspace Using XML Validator

To address the above errors, our workflow for reading this data was as follows:

1. Run XMLValidator workspace to test LOATrackSample.xml against LOA Tracks.xsd
2. Review any errors related to missing application schemas
3. Locate required application schema in NIEM tree. In all cases we were able to locate the application schema somewhere in the NIEM / IEPD packages provided for the TestBed
4. Create a directory path to match what was required as implied by the error message
5. Copy the required application schemas to that new path
6. Repeat steps 1 to 5 until there were no more schema errors

For example, this process was used to locate and place the following schemas:

.\SSATF v3\ISM\IC-ISM.xsd

.\SSATF v3\NTK\IC-NTK.xsd

.\SSATF v3\TSPI*.xsd

Once all the schemas were place in their correct locations, FME was then able to read a total of 6 Assertion records from 'LOATrackSample.xml' with 'LOA Tracks.xsd using the standard schema based GML reader. However none of elements containing geometry were read. The elements that do contain geometry (<observation>) were skipped, so there may be some problem with the geometry parsing using these schemas. See Figure 4. ‘Reading LOATrackSample.xml using standard FME GML reader’.

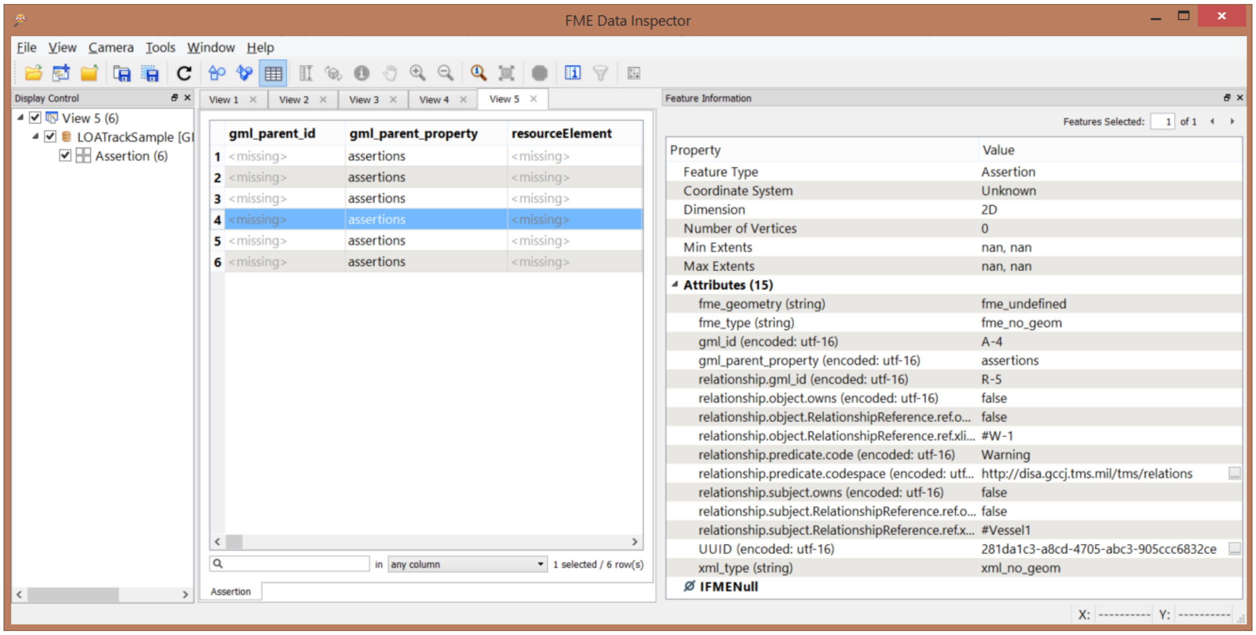

Figure 14 - Reading LOATrackSample.xml Using Standard FME GML Reader

5.5.2 Reading Geo4NIEM GML with a No-Schema GML Reader

During thread discussions of these GML parsing issues, it was mentioned that GML/WFS clients such as QGIS, and Gaia from Carbon Project, have a ‘fast parser’ read mode that ignores schema and allows them to be more tolerant when reading GML. Safe Software has been developing a similar ‘no schema’ reader for use in cases where no schema is available. Given the problems with the schemas as described above, it was decided that it would be useful to test reading the Geo4NIEM GML with this new ‘no schema’ GML reader. This reader functions much like QGIS’s GML reader and simply scans the XML dataset for GML geometries and XML elements given a set of match expressions.

In order to read LOATrackSample.xml, the reader was provided with a list of feature elements to search for, in this case: ‘observation relatedEntity Assertion Vessel’. This resulted in FME reading 17 records from 4 feature types including the observation elements that contain GML point geometry.

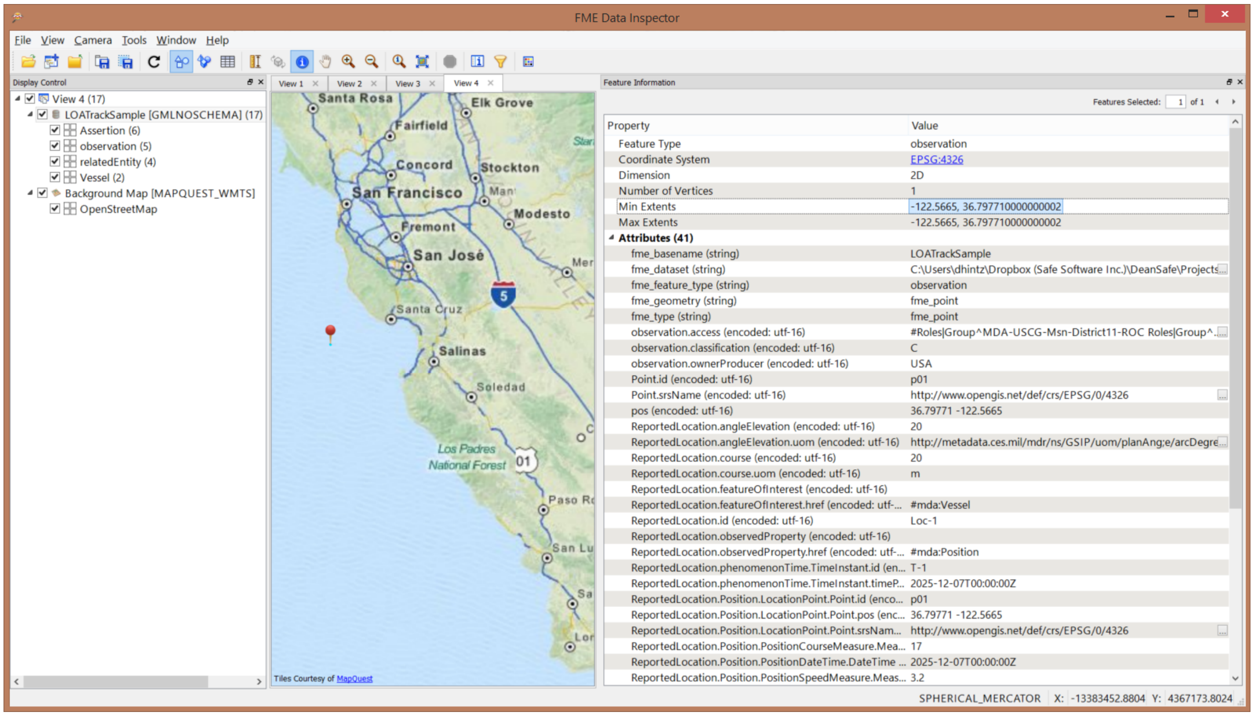

Figure 15 - Reading LOATrackSample.xml using FME GML No-Schema Reader

The following shows a summary of the feature types and number of records read from LOATrackSample.xml.

```
Features Read Summary
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Assertion 6
Vessel 2
observation 5
relatedEntity 4
==============================================================================
Total Features Read 17
```

Thus, when configured correctly, FME 2015.1's new 'no-schema' GML reader was able to successfully read all the GML elements and associated properties available from the sample Geo4NIEM GML source data provided. Note how GML geometries are automatically rendered when found. Also note how nested child elements are flattened into field names in the form of parent.child.

For example:

```
<ReportedLocation gml:id="Loc-2">
  <mda:Arrival>
    <mda:VisitAnchorageText>Main Anchorage</mda:VisitAnchorageText>
    <mda:VisitExpectedArrivalDateTime>
      <nc:DateTime>2025-12-10T14:30:00Z</nc:DateTime>
    </mda:VisitExpectedArrivalDateTime>
```


Becomes:

`ReportedLocation.id' has value `Loc-2'
`ReportedLocation.Arrival.VisitAnchorageText' has value `Main Anchorage'
`ReportedLocation.Arrival.VisitExpectedArrivalDateTime.DateTime' has value `2025-12-10T14:30:00Z'
etc.

See the Annex for examples of how Geo4NIEM sample data was rendered as relational records.

5.5.3 Discussion of Use of Components from Safe Software

A range of tests was performed on the Geo4NIEM GML provided. While ultimately, the data was readable using a no-schema approach, testing revealed a range of problems with the supplied schemas.

For typical WFS and GML clients to parse and consume Geo4NIEM, they would need access to a complex interdependent package of XSD files in the correct relative folder hierarchy. If any of these are missing, the reader can be expected to fail. It is also fairly demanding on the part of the client to parse and interpret the chain of schemas involved. The entire schema dependency has to be interpreted before a single feature can be rendered. Given that there may be 20+ schemas involved for something like notice of arrival, this parsing can take some time. Internet bandwidth can also affect performance given the amount of schema information that has to be read and processed. The exceptions to these schema challenges are clients which have a 'fast parser' or 'no-schema' mode for reading GML/WFS.

At the time of writing FME's WFS client was not yet able to read Geo4NIEM Notice of Arrival GML via PEP's post security WFS since FME's WFS client currently depends on a valid DescribeFeatureType response. While we had access to a package of Geo4NIEM schemas, some required schemas were missing. Given a complete set of valid local schemas, FME's WFS client should be able to read Geo4NIEM via WFS. The fact that it is difficult to assemble a complete package of required GML schemas for the implementation of NIEM GML used for this test bed likely contributed to the difficulties of building a complete and valid complete DescribeFeatureType response. By comparison, some tools such as Gaia or QGIS do not require a GML schema to interpret GML, but rather scan the XML for GML geometries and properties. Still, since DescribeFeatureType is part of the OGC WFS standard, many client tools do depend on this, and not having a valid response can make it impossible for these clients to consume features from associated WFS servers. Ignoring schema can also result in incorrect interpretations of the GML data.

Similar problems were encountered when trying to read the 'LOATrackSample.xml' sample Geo4NIEM GML dataset locally using FME's standard GML reader. After iterating through various schema location errors and adding schema paths and files as needed, the schema location problems were mitigated. Still, only a third of the available features were read, none of which included any geometry. Subsequently, FME's alternative ‘no schema’ GML reader was then tested. Using this no-schema approach all available features were read including the associated geometries.

However, the problems described with GML schemas remain. More work needs to be done to refine how the Geo4NIEM schemas are packaged and delivered in order to make them easier to read. Also, problems with consistency in geometry definitions need to be investigated. While the ‘no-schema’ reading approach may provide a workaround, there are risks and costs associated with ignoring the schema. The element names are interpreted at the time they are read based on the source GML and how the reader is configured. So the element names may not match what is expected or published in the schema or metadata. Any system that tries to automate the processing of the GML based on the schema is likely to run into problems if it reads the NIEM GML in a no-schema mode. In short, the ‘No Schema’ mode is useful for view only reading, or ad hoc exports, but is likely to cause problems if employed between systems or web services in an automated fashion.

It should be noted that the sample data tested was very limited. Future test beds would benefit from a greater variety of sample element types and geometries. As mentioned in the observations, only one dataset with 4 feature types and 17 records were tested. 2 records had point geometry and the others had none. It would be very beneficial to test with additional feature types with a variety of geometry types (lines and polygons), and a variety of coordinate systems. In general, while design discussions tend to center around the structure and nature of schemas, implementation of actual sample datasets tends to bring out issues that cannot be fully appreciated when examining schemas alone. We have observed that this has also been an ongoing challenge in the EU INSPIRE project.

Finally, we were not able to test the underlying goal of this task – that is to read Geo4NIEM GML, write it, and then compare the result with the source. FME’s GML writer is schema based and so cannot function without a complete set of schemas. So while the no-schema reader may serve as a workaround for reading NIEM GML in some cases, there is no equivalent option for FME’s GML writer. That said, FME’s XML writer can use a tool called XMLTemplater to define arbitrary XML templates and use those to write to any Geo4NIEM feature type. However this XMLTemplater based approach is manually intensive and appropriate only when a small number of feature types are required. Also, since it is not schema based it is more open to encoding errors.

In summary, Geo4NIEM holds a lot of potential as a framework for spatial data exchange among national security partners. FME was able to read both the georeferenced geometry and the associated element property data from the sample data provided. However, there remain significant challenges accessing and interpreting the GML schemas associated with the Geo4NIEM GML used in this project. More work needs to be done to develop best practices for applying geospatial NIEM to the serialization of GML and provision and consumption of geospatial OGC services. It is our recommendation that the GML application schemas used for Geo4NIEM be simplified and consolidated where possible, and consideration be given as to how these can be supported by WFS responses such as

DescribeFeatureType. Also, efforts should be made to better standardize the representation of location in Geo4NIEM. Where possible, this should be based on standard GML geometry such as GML Point rather than lat/long, intersections etc. Finally, from an external GIS industry perspective, it would be very helpful if documentation could be provided as part of the Geo4NIEM geospatial standard to specify the schema requirements for some example datasets. Providing a richer sample dataset that can be read and validated from a self-contained package, including a complete set of example WFS request and responses, would also greatly assist testing and development efforts.

6 Findings and Recommendations

Evidence obtained through the Testbed 11 Geo4NIEM thread supported three primary findings:

- First, with reasonable effort it is possible to combine NIEM, IC security specifications, OGC Web Service components, and GML-aware clients to support information exchange with authorized users.

- Second, implementing such an exchange requires extra work, compared to a typical exchange of features that conform to the GML Simple Features profile. However, this level of effort is not greater than encodings already in OGC, such as Aeronautical Information Exchange Model (AIXM), where a community of interest has defined a standard GML application schema for exchanging geographic data.

- Finally, it is possible to simplify the implementation of NIEM and IC security specifications and still meet information exchange needs. This simplification can reduce the technical overhead required to broadly implement secure information exchanges and emerging collaborative partnerships. Simplification options include NIEM IEPD development guidance or recommended practices that reduce the impact of generating excessive namespaces.

The following sections detail these findings and associated recommendations. Please also note several focused findings above in the section “Discussion of Use of Components from Safe Software.”

6.1 Combining NIEM, IC Security, and OWS is Feasible

The demonstration used real-world NIEM IEPs, containing embedded GML elements, properly tagged with IC access control and security metadata and optionally enclosed within the IC's dissemination format for binding assertion metadata with data resources (i.e. IC-TDF.XML/TDO).. The demonstration was constructed using a cloud-based WFS server, multiple Policy Enforcement Points that provide access controls and filters based upon the user attributes stored in the OGC Attribute Store and multiple GML-aware clients. Major OGC operations in a simulated distributed information exchange were assessed including:

- WFS server with GetCapabilities, DescribeFeatureType, GetFeature, and Transaction operations
- Access control engines enforcing access policy based on user attributes and IC metadata attributes in the WFS FeatureCollection payload
- Clients interpreting the WFS FeatureCollection elements and performing transaction operations

NIEM 3.0 was compatible with the IC security specifications access control and dissemination (ISM, NTK, and TDF) and supported the access control policies for the demonstration scenario. There is no evidence to suggest incompatibility with more complex policies, schemas and security markings. Access control engines can work with NIEM/IC Data Encoding, with or without the NIEM/IC Feature Processing API.

The participants spent most of their time learning about the NIEM exchange specifications and the IC security specifications. Implementation of the second and third information exchanges (based on Incident and Resource IEPs) required less development time since specialized tools were created to speed the ‘cloning’ of the first WFS instance (based on the Notice of Arrival IEP).

Recommendation 1: Develop, test and demonstrate tools that clone and adjust data elements of WFS instances of NIEM/IC Data Encodings to simplify and speed development and deployment of service-based information exchanges. Assess tools that promote export of NIEM/IC Data Encodings.

Recommendation 2: Assess how IC security specifications, access control and dissemination (ISM, NTK, and TDF) may further enable WFS and GML-based data exchange.

6.2 Extra Effort Relative to Typical Use of Simple Features Profile

The GML Simple Features profile defines fixed coding patterns for the use of a subset of XML Schema and GML constructs. It is intended to address the case where a client interacts with a previously unknown server offering. This is the typical case for many OWS components. Relative to that typical case, the demonstration implementation for the NIEM/IC Feature Processing API and NIEM/IC Data Encoding (Testbed 11 ER 15-048) required extra effort in three areas: complex non-spatial properties, multiple namespaces and DescribeFeatureType, and context-dependent value references in filter encodings.

6.2.1 Complex Non-Spatial Properties

Information exchanges implementing the draft NIEM/IC Feature Processing API required schemas in wfs:FeatureCollections roughly equivalent to those that comply with level SF-2 for GMLsf. This finding means that some current WFS and GML applications and services expecting GMLsf Level 0 or 1 tools may not able to fully operate with the NIEM/IC Feature Processing API ‘out of the box’. This finding also means that exporting NIEM/IC Data Encoding from a WFS implementing NIEM/IC Feature Processing API may not be possible in common GIS formats such as Shapefiles.

The SF-0 profile does not allow complex non-spatial properties, while these are permitted but unusual in the SF-1 profile. This simplicity can be exploited in server and client software, allowing off-the-shelf components to handle new application schemas with little or no special effort. However, this simplicity is not present in the NIEM/IC Feature

Processing API and NIEM/IC Data Encoding. For example, the Notice of Arrival IEPD defines a complex property with six levels of nested elements, resulting in data like this:

```
<mda:Vessel ...>
  <m:VesselAugmentation ...>
    <m:VesselCallSignText>H3LP</m:VesselCallSignText>
    <m:VesselCargoCategoryText>Harmful Substances ...
    <m:VesselCategoryText>Container Ship ...
    <m:VesselCDCCargoOnBoardIndicator>true ...
    <m:VesselCharterer ...>
      <nc:EntityOrganization>
        <nc:OrganizationLocation>
          <nc:Address>
            <nc:LocationCountryISO3166Alpha2Code>KR ...
          </nc:Address> ...
```

From the perspective of an Information Exchange designer or implementer, this level of complexity may require effort in the WFS server implementations when compared with less extensive SF-0 and SF-1 schemas, especially when implementing the WFS-T functions. It also requires extra effort in the client applications, where specialized Filter Encodings using XPath expressions are necessary to retrieve values from the complex properties. This extra effort can be reduced by careful NIEM-conformant IEPD design. Instead of using all available NIEM objects, designers can carefully construct IEPD schemas using just enough NIEM objects to meet the community's information exchange need.

Recommendation 3: Develop and test a Best Practice that defines more limited, but useful, subsets of NIEM schema components (including location as GML), with required IC DES components, to lower the ‘implementation bar’ of time and resources required for developing software that supports the NIEM/IC Feature Processing API. By lowering the level of effort, Information Exchange designers, geospatial developers and access control software implementers will be encouraged to take greater advantage of the rich functionality in NIEM/IC. The Best Practice should be designed around the business elements needed by Information Exchange Designers.

6.2.2 Multiple Namespaces, and DescribeFeatureType

The WFS DescribeFeatureType operation returns an XML Schema document containing a complex type definition for the specified feature type. In order to form a complete schema, the client must then either retrieve or already possess a separate schema document for each imported namespace. This is essential for WFS servers and GML clients implemented with validating parsers. On the other hand, implementations based on non-validating parsers do not need the schema and do not rely on DescribeFeatureType. Both approaches were tested in Testbed 11 Geo4NIEM Thread.

For application schemas conforming to the Simple Features profile, implementing the DescribeFeatureType operation is relatively simple. These schemas typically define features within a single namespace, and clients usually have schema documents for the imported GML namespaces.

Implementing the DescribeFeatureType operation for the NIEM/IC Feature Processing API is more complicated. The schema for such a feature type will have many namespaces, and clients may not always have the corresponding schema document. This can greatly complicate the implementation of the DescribeFeatureType operation.

Two aspects of NIEM IEPDs may be exploited in future work to reduce much of this complexity. A conforming IEPD contains the complete set of schema documents. It also contains a set of OASIS XML Catalog files providing a mapping between namespace URI and schema document file name. A WFS server could use the catalog to rewrite every <import> schema element so that the schemaLocation attribute resolves to a schema document on the server.

Recommendation 4: Develop, test and demonstrate the feasibility of making schemas available from WFS implementing the NIEM/IC Feature Processing API. This may or may not be part of the DescribeFeatureType operation so PEPs can create filter rules based upon them. This recommendation may also include assessing methods by which PEPs may process security tag information from the DescribeFeatureType.

Recommendation 5: Assess, develop, test and demonstrate governance methods to provide complete sets of public-accessible schema document. In particular, assess methods to assist IEPD developers in maintaining and accessing schemas.

6.2.3 Context-Dependent Value References in Filter Encodings

From the perspective of an OGC software developer or user the nested structure in the data encodings associated with the NIEM/IC Feature Processing API means implementing fully capable OGC Filter Encodings for WFS will require a subset of XPath. For example, the Notice of Arrival NIEM IEPD describes data like this:

```
<m:VesselDOCCertificate>
  <nc:DocumentExpirationDate>
    <nc:Date>2028-04-24T00:00:00</nc:Date>
  </nc:DocumentExpirationDate>
  <nc:CertificateIssueDate>
    <nc:Date>2026-03-11T00:00:00</nc:Date>
  </nc:CertificateIssueDate>
```


XPath is required to distinguish between the `nc:Date` of document expiration and certificate issue. There is a similar context dependency in NTK, where XPath is required to distinguish between the `ntk:AccessGroupList` element within `ntk:RequiresAnyOf`, and the same element within `ntk:RequiresAllOf` . Therefore, the use of either NIEM or IC security requires Filter processing with XPath enabled.

XPath is accounted for in the Filter Encoding specification, but it is a specialized case and not as broadly implemented as the standard spatial, logical and comparison operators of WFS.

Recommendation 6: Develop, test and demonstrate the feasibility of fully capable OGC Filter Encodings for WFS using a subset of XPath. This approach provides the potential for high fidelity queries on the NIEM/IC Feature Processing API in support of mission and community requirements.

6.3 Simplifying Use of NIEM and IC Security and Meeting Exchange Needs

The extra effort required to implement the NIEM/IC Feature Processing API is not unique to either of those standards. It is common in situations where a community of interest has defined a standard GML application schema for exchanging geographic data, and presumes understanding on the part of all community participants. For example, the Aeronautical Information Exchange Model (AIXM) provides a standard GML application schema for aeronautical information exchange. This application schema defines many complex non-spatial properties, uses multiple namespaces, and includes context-dependent element values. Implementing AIXM-based exchanges with off-the-shelf components requires the same sort of extra effort needed for the NIEM/IC encoding. For example, the Gaia client requires a special "AIXM extender" in order to process AIXM data.

This extra effort can be reduced by careful NIEM-conformant IEPD design. Instead of using all available NIEM objects, designers can carefully construct IEPD schemas using just enough NIEM objects to meet the community's information exchange need. It may be possible to satisfy a large set of information exchange needs with a simple "what, where, when" IEPD that approaches the Simple Feature profile, using reduced nesting and a subset of location designations and security tags.

Achieving broad implementation of these approaches will make it possible for the NIEM/IC Feature Processing API to support emerging agile information exchanges driven by collaborative partnerships. This transformation is vital to confronting the security challenges of the future.

Recommendation 7: Develop, test, and demonstrate the feasibility of a 'Generic' NIEM-conformant IEPD with location, time, what, who information as 'core' elements in simple GMLsf.

Recommendation 8: Develop, test and demonstrate the feasibility of a generic GML Application Schema leveraging NIEM-conformant components and IC specification components. This would extend the usefulness of NIEM components from an OGC implementation standpoint within a particular community of interest.

Annex A

Round Trip Components from Safe Software: Sample Records

Feature Type: ` Vessel'

`VesselAugmentation.VesselBeamMeasure.MeasureValueText' has value `120.0'
`VesselAugmentation.VesselCallSignText' has value `H3LP'
`VesselAugmentation.VesselCargoCategoryText' has value `Harmful Substance'
`VesselAugmentation.VesselCharterer.EntityOrganization.OrganizationLocation.Address.LocationCountryISO3166Alpha2Code' has value `KR'
`VesselAugmentation.VesselCharterer.EntityOrganization.OrganizationName' has value `SK Shipping'
`VesselAugmentation.VesselClassText' has value `Bulk Carrier'
`VesselAugmentation.VesselClassificationSocietyName' has value `Germanischer Lloyd'
`VesselAugmentation.VesselContactInformation.ContactEntity.EntityOrganization.OrganizationName' has value `Horizon Lines'
`VesselAugmentation.VesselContactInformation.ContactEntity.EntityPerson.PersonName.PersonFullName' has value `James Smith'
`VesselAugmentation.VesselContactInformation.ContactTelephoneNumber{0}.InternationalTelephoneNumber.TelephoneNumberID' has value `>800-555-121'
`VesselAugmentation.VesselContactInformation.ContactTelephoneNumber{0}.TelephoneNumberCategoryCode' has value `work'
`VesselAugmentation.VesselContactInformation.ContactTelephoneNumber{1}.InternationalTelephoneNumber.TelephoneNumberID' has value `800-555-1213'
`VesselAugmentation.VesselContactInformation.ContactTelephoneNumber{1}.TelephoneNumberCategoryCode' has value `fax'
`VesselAugmentation.VesselDOCCertificate.CertificateIssueDate.Date' has value `2018-04-25'
`VesselAugmentation.VesselDOCCertificate.CertificateIssuingAgency.EntityOrganization.OrganizationName' has value `U.S. Coast Guard'
`VesselAugmentation.VesselDOCCertificate.DocumentExpirationDate.Date' has value `2028-04-24'
`VesselAugmentation.VesselDraftMeasure.AngleUnitText' has value `m'
`VesselAugmentation.VesselDraftMeasure.MeasureValueText' has value `12.1'
`VesselAugmentation.VesselGrossTonnageValue' has value `54881'
`VesselAugmentation.VesselIMONumberText' has value `9278155'
`VesselAugmentation.VesselISSC.CertificateIssuingAgency.EntityOrganization.OrganizationName' has value `Government of Bermuda, Department of Maritime Administration'
`VesselAugmentation.VesselISSC.RecognizedISSCSecurityEntity.EntityOrganization.OrganizationName' has value `Government of Bermuda, Department of Maritime Administration'
`VesselAugmentation.VesselISSC.VesselSecurityOfficerContactInformation.ContactEmailID' has value `ftest@test.com'
`VesselAugmentation.VesselISSC.VesselSecurityOfficerContactInformation.ContactEntity.EntityPerson.PersonName.PersonFullName' has value `Frank Test'

`VesselAugmentation.VesselISSC.VesselSecurityOfficerContactInformation.ContactTelephoneNumber{0}.InternationalTelephoneNumber.TelephoneNumberID' has value `888-234-5431'
`VesselAugmentation.VesselISSC.VesselSecurityOfficerContactInformation.ContactTelephoneNumber{0}.TelephoneNumberCategoryCode' has value `work'
`VesselAugmentation.VesselISSC.VesselSecurityOfficerContactInformation.ContactTelephoneNumber{1}.InternationalTelephoneNumber.TelephoneNumberID' has value `888-234-5431'
`VesselAugmentation.VesselISSC.VesselSecurityOfficerContactInformation.ContactTelephoneNumber{1}.TelephoneNumberCategoryCode' has value `fax'
`VesselAugmentation.VesselISSC.VesselSecurityPlanImplementedIndicator' has value `true'
`VesselAugmentation.VesselMMSIText' has value `352948000'
`VesselAugmentation.VesselName' has value `MSC NERISSA'
`VesselAugmentation.VesselNationalFlagISO3166Alpha2Code' has value `PA'
`VesselAugmentation.VesselOfficialCoastGuardNumberText' has value `US878N2'
`VesselAugmentation.VesselOperationalConditionOfEquipmentDescriptionText' has value `Operational'
`VesselAugmentation.VesselOperator.EntityPerson.PersonName.PersonFullName' has value `Dan James'
`VesselAugmentation.VesselOverallLengthMeasure.MeasureValueText' has value `294.08'
`VesselAugmentation.VesselOverallLengthMeasure.SpeedUnitText' has value `m'
`VesselAugmentation.VesselOwner.EntityOrganization.OrganizationLocation.Address.LocationCountryISO3166Alpha2Code' has value `US'
`VesselAugmentation.VesselOwner.EntityOrganization.OrganizationName' has value `American Shipping Company'
`VesselAugmentation.VesselSafetyManagementCertificate.CertificateIssueDate.Date' has value `2017-03-12'
`VesselAugmentation.VesselSafetyManagementCertificate.CertificateIssuingAgency.EntityOrganization.OrganizationName' has value `U.S> Coast Guard'
`VesselAugmentation.VesselSafetyManagementCertificate.DocumentExpirationDate.Date' has value `2027-12-01'
`fme_basename' has value `LOATrackSample'
`fme_dataset' has value `.\Geo4NIEM\GMLRoundTrip\SSATF witn NIEM\Examples\LOATrackSample.xml'
`fme_feature_type' has value `Vessel'
`fme_type' has value `fme_no_geom'
`multi_reader_keyword' has value `GMLNOSCHEMA_1'
`sourceFile' has value `LOATrackSample'

Feature Type: ‘observation’

`Point.id' has value `LP1'
`ReportedLocation.Arrival.VisitAnchorageText' has value `Main Anchorage'
`ReportedLocation.Arrival.VisitExpectedArrivalDateTime.DateTime' has value `2025-12-10T14:30:00Z'
`ReportedLocation.Arrival.VisitLocationInPort.LocationCityName' has value `Oakland'
`ReportedLocation.Arrival.VisitLocationInPort.LocationStateName' has value `CA'
`ReportedLocation.Arrival.VisitLocationInPort.PortAugmentation.LocationPoint.Point.id' has value `LP1'
`ReportedLocation.Arrival.VisitLocationInPort.PortAugmentation.LocationPoint.Point.pos' has value `37.7955 122.2846'
`ReportedLocation.Arrival.VisitLocationInPort.PortName' has value `Oakland'
`ReportedLocation.Arrival.VisitReceivingFacilityName' has value `Pier 57'

`ReportedLocation.featureOfInterest' has value `'
`ReportedLocation.id' has value `Loc-2'
`ReportedLocation.observedProperty' has value `'
`ReportedLocation.phenomenonTime' has value `'
`ReportedLocation.procedure' has value `'
`ReportedLocation.reportFrequency' has value `'
`ReportedLocation.result' has value `'
`ReportedLocation.resultTime' has value `'
`fme_basename' has value `LOATrackSample'
`fme_dataset' has value `.\GMLRoundTrip\SSATF witn NIEM\Examples\LOATrackSample.xml'
`fme_feature_type' has value `observation'
`fme_geometry' has value `fme_point'
`fme_type' has value `fme_point'
`multi_reader_keyword' has value `GMLNOSCHEMA_1'
`observation.classification' has value `U'
`observation.ownerProducer' has value `USA'
`pos' has value `37.7955 122.2846'
`sourceFile' has value `LOATrackSample'
Coordinate System: `'
Geometry Type: IFMEPoint
Number of Geometry Traits: 1
GeometryTrait(encoded: utf-16): `gml_id' has value `LP1'
Coordinate Dimension: 2
(37.795499999999997,122.2846)

Revision history

Date	Release	Editor	Primary clauses modified	Description
2015-05-21	Draft	S. Serich	All	Initial project deliverable
2015-07-16	Final	S. Serich	All	Final project deliverable
2015-09-16	Final-final	S. Serich	All	Incorporated edits from Carl Reed and Justin Stekervetz