# Open Geospatial Consortium

Publication Date: 2016-01-25

Approval Date: 2015-09-17

Posted Date: 2015-07-21

Reference number of this document: OGC 15-047r3

Reference URL for this document: http://www.opengis.net/doc/PER/tb-11-geo4niem-wfs-api

Category: Public Engineering Report

Editor(s): Jeff Harrison

# Testbed-11 NIEM-IC Feature Processing API using OGC Web Services



## Warning

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

| | |
|---|---|
| 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 thread in 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 on Open Geospatial Consortium (OGC) Web Services to be used by client applications. Access control is based on attributes of the end-user and the instance data.

The assessment included reviewing example IEPDs and performing test and demonstrations using OGC web services, such as Transactional Web Feature Services (WFS-T), Policy Enforcement Points (PEPs) and OGC Attribute Stores to process geographic feature with NIEM components and security tags. The Test and Demonstration included, but was not limited to feature retrieval and transactions. Recommendations to update these information exchanges were provided to reflect NIEM 3.0 architecture and security tags in a 'NIEM/IC Feature Processing API'. Results from this task helped provide a preliminary architecture for Geo4NIEM in Testbed 11, summarized in other OGC Testbed 11 Engineering Reports.

This task also identified potential change requests to OGC WFS or other OGC Services for handling security information in a federated role-based access control environment. These changes may help the NIEM/IC transform into more agile and customer-centric frameworks driven by collaborative partnerships. This transformation is vital to confronting the security challenges of the future.

## 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 rely upon much of 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 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 (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 web services issued by the OGC it will significantly improve inter-government information sharing.

This is especially important since geospatial interoperability efforts have matured to a point where broad acceptance is now dependent on the capacity to secure information exchanges. In fact, organizations that are considering participation in information exchanges must also consider how they can establish distributed security frameworks for role-based access control to geospatial and other resources. These requirements will continue to increase as data access transitions into data management with services like WFS-T - where loosely affiliated parties collaborate on maintenance of shared situational awareness resources.

## Keywords

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

# Testbed-11 NIEM-IC Feature Processing API using OGC Web Services

## 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.

This task also identified potential change requests to OGC WFS and other OGC Web Services for handling security information in a federated role-based access control environment. These changes may help the NIEM/IC transform into more agile and customer-centric frameworks driven by collaborative partnerships. This transformation is vital to confronting the security challenges of the future.

This is especially important since geospatial interoperability efforts have matured to a point where broad acceptance is now dependent on the capacity to secure information exchanges. In fact, organizations that are considering participation in information exchanges must also consider how they can establish distributed security frameworks for access control to resources. These requirements will continue to increase as data access transitions into data management with services like WFS-T where loosely affiliated parties collaborate on maintenance of shared situational awareness resources.

This effort builds on the previous work of the Geo4NIEM Pilot Project. 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) aligned to OGC Web Services, Phase 9 (OWS-9) Testbed related work.

- Testing and demonstrating use of 1) NIEM 3.0 architecture, and access control and security tagging metadata defined by the IC Data Encoding Specifications leveraging OWS-9; and 2) full 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. The IEPDs assessed were:

- Notice of Arrival IEPD
- 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 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

**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 process 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 (Preliminary)

**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 (Final)

**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 Policy Enforcement Points (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-IC Feature Processing API using OGC Web Services ER.

## 1.2 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

The following participants (listed in alphabetical order by surname) made substantial contributions to the content of this report. All questions regarding this document should be directed to the editor or any of the contributors.

| Name | Organization |
|---|---|
| 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 Standard (http://www.opengeospatial.org/standards/common )

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

In addition to this document, this report includes several XML Document files as specified in Annexes A and B.

## 3 Terms and definitions

For the purposes of this report, the definitions specified in the OGC Web Feature Service (WFS), the OGC) Filter Encoding Implementation Specification and the OWS Common Implementation Standard shall apply.

### 3.1 Abbreviated Terms

| | |
|---|---|
| ABAC | Access Based Access Control |
| AIXM | Aeronautical Information Exchange Model |
| API | Application Programming Interface |
| ARH | Access Rights and Handling |
| DES | Data Encoding Specification |
| EDH | Enterprise Data Header |
| FES | Filter 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 |
| LDAP | Lightweight Directory Access Protocol |
| 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 |
| SAML | Security Assertion Markup Language |
| 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 |

### 3.2 Used parts of other documents

This document uses significant parts of other OGC documents. This report refers to those documents by citing section designations, or copies some of those parts with small modifications.

## 4 API Development

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. The assessment was conducted by implementing prototype components that use a ‘NIEM-IC Feature Processing API’ 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. Details on the prototype test environment, test results

and demonstration are provided in a separate Engineering Report. A representation of the key API points for NIEM-IC Feature Processing is provided in Figure 1.

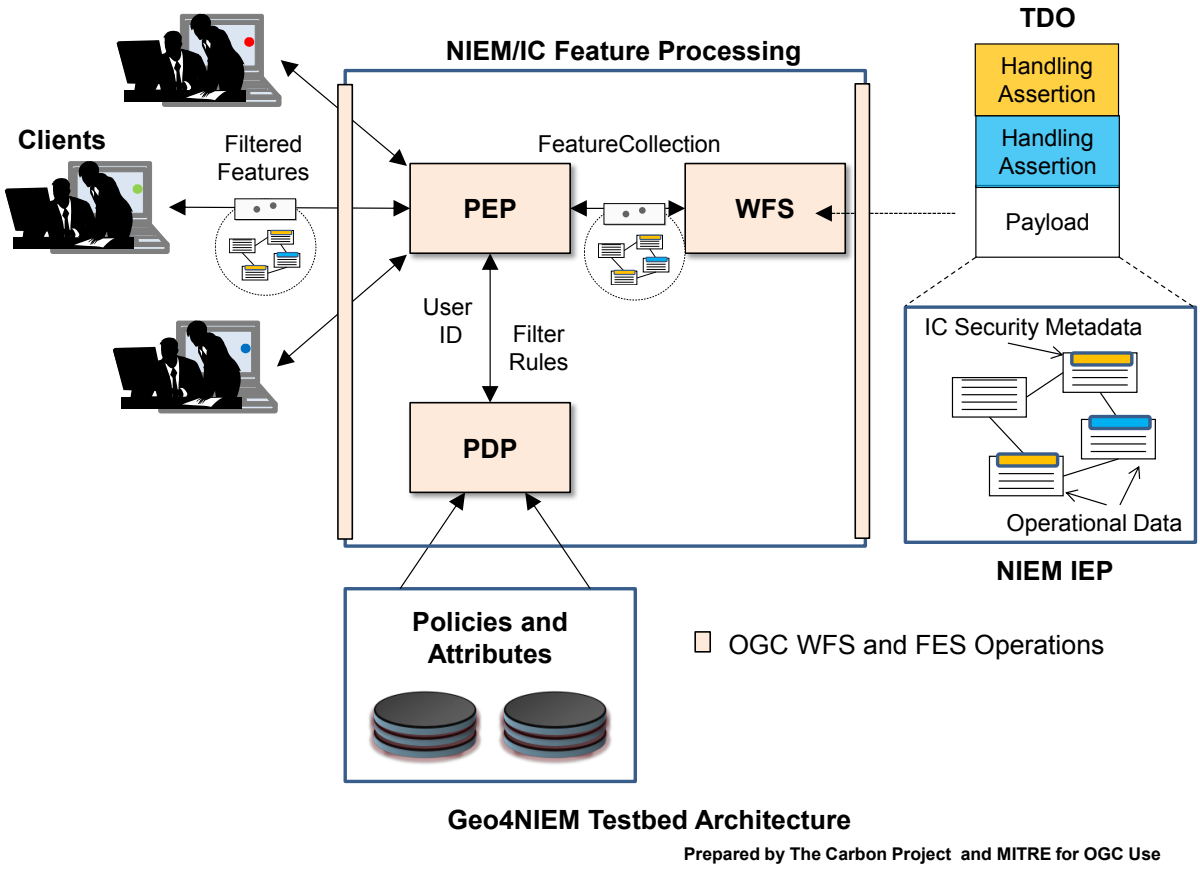



**Figure 1 – Geo4NIEM Testbed Architecture** [1]

For this testbed four service interfaces, encodings and information exchange frameworks were considered during API development:

- OGC Web Feature Services
- OGC Filter Encoding
- IC Data Encoding Specifications

[1] 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.

- NIEM 3.0

### 4.1 Web Feature Service (WFS)

The OGC Web Feature Service (WFS) Implementation Specification allows a client to retrieve geospatial data encoded in Geography Markup Language (GML) and other formats from multiple Web Feature Services. The standard defines operations for data access and manipulation operations on geographic features, using HTTP as the distributed computing platform. Via these interfaces, a Web user or service can combine, use and manage geodata -- the feature information behind a map image.

The WFS Standard specifies the behavior of a service that provides transactions on and access to geographic features in a manner independent of the underlying data store. It specifies discovery operations, query operations, locking operations, transaction operations and operations to manage stored parameterized query expressions:

- Discovery operations allow the service to be interrogated to determine its capabilities and to retrieve the application schema that defines the feature types that the service offers.
- Query operations allow features or values of feature properties to be retrieved from the underlying data store based upon constraints, defined by the client, on feature properties.
- Locking operations allow exclusive access to features for the purpose of modifying or deleting features.
- Transaction operations allow features to be created, changed, replaced and deleted from the underlying data store.
- Stored query operations allow clients to create, drop, list and described parameterized query expressions that are stored by the server and can be repeatedly invoked using different parameter values.

NOTE The WFS Standard does not address the access control issues. This is an important distinction for NIEM/IC interoperability testing, demonstration and operational implementation.

The WFS Standard defines eleven operations:

- GetCapabilities (discovery operation)

- DescribeFeatureType (discovery operation)
- GetPropertyValue (query operation)
- GetFeature (query operation)
- GetFeatureWithLock (query & locking operation)
- LockFeature (locking operation)
- Transaction (transaction operation)
- CreateStoredQuery (stored query operation)
- DropStoredQuery (stored query operation)
- ListStoredQueries (stored query operation)

Some WFS servers may implement the HTTP POST conformance class, and some may implement the HTTP GET conformance class. This is an important distinction for NIEM/IC interoperability testing, demonstration and operational implementation.

Some WFS servers may also support additional non-GML feature encodings and client applications may access them using the outputFormat parameter domains. However, the WFS Standard does not describe how a server would operate upon such encodings. This is an important distinction for NIEM/IC interoperability testing, demonstration and operational implementation.

### 4.2 Filter Encoding Specification

The OGC Filter Encoding Implementation Specification describes an XML and KVP encoding of a system neutral syntax for expressing projections, selection and sorting clauses collectively called a ‘query expression’. As background, a fundamental operation performed on a set of data or resources is that of querying in order to obtain a subset of the data which contains certain desired information that satisfies some query criteria and which is also, perhaps, sorted in some specified manner.

The Filter Encoding Standard defines the XML encoding for the following predicates.

- A standard set of logical predicates: and, or and not.
- A standard set of comparison predicates: equal to, not equal to, less than, less than or equal to, greater than, greater than or equal to, like, is null and between.

- A standard set of spatial predicates: equal, disjoint, touches, within, overlaps, crosses, intersects, contains, within a specified distance, beyond a specified distance and BBOX.

- A standard set of temporal predicates: after, before, begins, begun by, contains, during, ends, equals, meets, met by, overlaps and overlapped by.

- A predicate to test whether the identifier of an object matches the specified value.

#### 4.2.1 XPath and Filter Encoding

In cases where the data model of the service that implements Filter Encoding is represented as XML, as is the case with OGC 09-025r2 where GML (see ISO 19136) is used, value references can refer to parts of a complex property and shall be encoded using the XML Path Language (given in W3C XML Path Language).

The XML Path Language (as given in W3C XML Path Language) specification is a language for addressing parts of an XML document, or in the case of Filter Encoding, for referencing XML elements and attributes that represent the properties of an object encoded in XML.

The Filter Encoding Standard does not require that a filter expression processor support the full XPath language. In order to keep the implementation entry cost as low as possible, services that implements the Filter Encoding standard and require the use of XPath, shall support a subset of the XPath language.

### 4.3 IC Data Encoding & Service Specifications

The success of intelligence, defense, homeland security, and law enforcement missions is dependent on information producers and consumers being able to share, manage, discover, retrieve, and access information across national and international boundaries. IC Data Encoding Specifications (DES) are the result of IC collaboration and coordination in response to public law, executive orders, policy and guidance, and change requests submitted by IC elements. Data encoding specifications define agreed upon digital encodings or formats for information being shared or exchanged within the enterprise. These specifications should be viewed as component modules. Many of the specifications are tightly integrated and dependent on each other. They can be integrated into other data encoding specifications or profiled (i.e., configured or constrained) to achieve a particular mission or business objective - such as supporting security tagging within the NIEM.

While this flexibility exists, users of the IC Data Encoding Specifications are required to maintain conformance to the relevant specification. An instance document is considered conformant to an IC DES if it passes all of the normative validation steps. The IC DES

XML schemas (unless noted otherwise) CVE values from the XML CVE files, and the Schematron code version of the constraint rules are normative for the specifications.

#### 4.3.1 XML Data Encoding Specification for Information Security Marking (ISM) Metadata

This XML Data Encoding Specification (DES) for Information Security Markings (ISM.XML) defines detailed implementation guidance for using XML to encode Information Security Markings (ISM) metadata. This DES defines the XML attributes, associated structures and relationships, restrictions on cardinality, permissible values, and constraint rules for representing electronic information security markings.

#### 4.3.2 XML Data Encoding Specification for Need-To-Know (NTK) Metadata

This XML Data Encoding Specification (DES) for Need-to-Know Metadata (NTK.XML) defines detailed implementation guidance for using XML to encode metadata necessary to facilitate automated systems making access control decisions. This DES defines the XML elements and attributes, associated structures and relationships, restrictions on cardinality, and permissible values for representing access control data concepts using XML.

The metadata, are used to represent the system-specific properties assigned to an information resource that will be used, in conjunction with information about the user, and possibly other information, to determine the user's access to the data. A single information resource may include multiple occurrences of these metadata in order to specify access control information according to multiple, different access systems.

#### 4.3.3 XML Data Encoding Specification for Trusted Data Format (TDF)

This XML Data Encoding Specification (DES) for Trusted Data Format (IC-TDF.XML) defines detailed implementation guidance for using XML to encode IC-TDF data. This Data Encoding Specification (DES) defines the XML elements and attributes, associated structures and relationships, mandatory and cardinality requirements, and permissible values for representing trusted data format data concepts using XML.

The Intelligence Community (IC) has standardized the various classification and control markings established for information sharing within the Information Security Markings (ISM), Need-To-Know (NTK), Enterprise Data Header (EDH), and Access Rights and Handling (ARH) XML specifications of the Intelligence Community Enterprise Architecture (ICEA) Data Standards. The IC-TDF.XML specification further expands on this body of work, adapting and extending it as necessary for TDF to function as the IC submission format for binding assertion metadata with data resource(s). This TDF functionality supports the IC way-ahead strategy of implementing secure cloud-based information exchange and discovery on the IC Enterprise

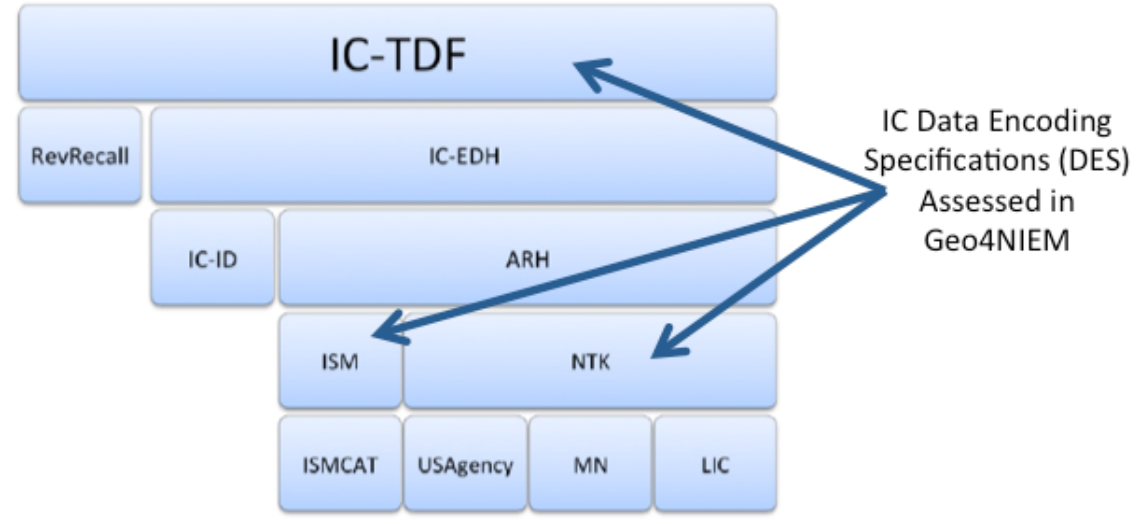



**Figure 2 - IC-TDF Dependencies[2]**

The IC-TDF.XML specification has a consistent and simple concept of Assertions and Payloads. There are two options for root elements: Trusted Data Object (TDO) and Trusted Data Collection (TDC). A TDO contains some data (the payload) and some statements about that data (the assertions). In the context of TDF, an 'assertion' is defined as a statement providing handling, discovery, or mission metadata describing a payload, TDO, or TDC, depending on the scope of the assertion. To facilitate handling and access control decisions, each TDO and TDC must contain at least one Handling Assertion. A Handling Assertion is a special type of structured assertion that contains the IC Enterprise Data Header (EDH) for the TDO or payload, providing the attributes needed for policy decisions regarding access control and how the data must be handled. ISM and NTK markings are contained in Handling Assertions, as part of the Access Rights and Handling (ARH) block. Additional discovery and mission assertions may also be provided. A TDC contains a list of TDOs (the payload) and some statements about those TDOs (the assertions). A TDC may also be a collection of collections, and contain other TDCs.

Each TDO consists of one or more assertions and a payload. Assertions may optionally be cryptographically bound to the payload to provide assurance over the integrity of the assertion, the payload, and the relationship between the assertion and payload. Each IC-TDF requires at least one handling assertion, optional discovery and mission assertions, and a payload. The handling assertion must consist of a structured IC-EDH block. Mission specific metadata may consist of a structured block (XML) or unstructured data

[2] Graphic provided by the Office of the Director of National Intelligence (ODNI) Office of the Chief Information Officer (OCIO) with annotations provided by Defense Information Systems Agency (DISA) and the NIEM Program Management Office (PMO).

(binary). The payload may be structured XML, unstructured data, or a reference. A TDC consists of a collection of TDOs or TDCs. It is expected but not required that the child TDOs and TDCs within a TDC are in some way related, with relationships encoded in the TDC assertions.

Information sharing within the national intelligence enterprise increasingly relies on information assurance metadata to allow interagency access control, automated exchanges, and appropriate protection of shared intelligence. This requires a structured, verifiable representation of security metadata bound to the intelligence data in order for the enterprise to become inherently "smarter" about the information flowing in and around it. This representation when implemented with other data formats, improved user interfaces, and data processing utilities, can provide part of a larger robust information assurance infrastructure capable of automating some of the management and exchange decisions now requiring human involvement. These specifications are in operational usage outside of the IC currently for other missions such as Defense and Law Enforcement. In Geo4NIEM they have been successfully applied to a disaster management scenario.

#### 4.3.4 IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set (UIAS)

The IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set (UIAS) codifies the minimum set of enterprise-level authorization attributes that IC elements are expected to provide if they participate in the Intelligence Community Unified Authorization and Attribute Service (UAAS) architecture. It provides a common, consistent way to identify IC enterprise authorization attributes of IC persons produced by, stored within, or shared throughout the IC's information domain. The name, definition, cardinality, and controlled vocabulary for each attribute are defined in order to promote interoperability between UAAS-compliant attribute services established by participating Agencies.

Defining the mandatory minimum set of IC enterprise authorization attributes and values for sharing through the IC UAAS federation supports consistent and assured information sharing across the enterprise. The IC UAAS supports Attribute-Based Access Control (ABAC) to promote on-demand access to information and other resources by IC users and services, and reduces authorization vulnerabilities by strengthening the access control decision process.

IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set (UIAS) specification is implemented by the OGC Attribute Store to define the user attributes used for the Testbed 11. While the UIAS specification codifies the minimum set of enterprise-level authorization attributes that IC elements are expected to provide if they participate in the Intelligence Community Unified Authorization and Attribute Service (UAAS) architecture, Testbed 11 applies the specification to state and local emergency responder participants. These attributes are explicitly used as parameters for access to the data assets tagged with NTK.XML.

### 4.4 NIEM 3.0

NIEM is a standards-based approach to the design of structured information exchange specifications. Figure 2 illustrates the process, which is described in reverse order (right to left) as follows: Producer and consumer software applications exchange structured information in the form of XML documents known as information exchange packages (IEPs). Developers of that software understand the expected content of those IEPs by understanding the exchange specification, which in NIEM is called an information exchange package documentation (IEPD). The designers of the IEPD follow the NIEM process, reusing data components from the NIEM data model and extending their exchange with new components as needed. The NIEM community [3] creates shared data components for those concepts on which they can agree and for which they believe a common definition will be useful.

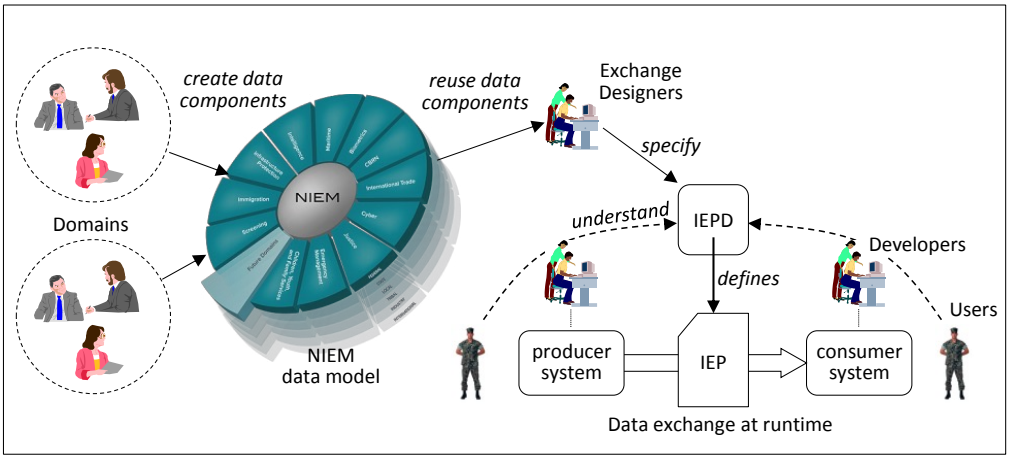



**Figure 3 - The NIEM Process**

An IEPD consists of a minimal but complete set of artifacts (XML schemas, documentation, sample XML instances, etc.) that defines and describes an implementable NIEM information exchange. A complete and conforming IEPD will contain all the schema definitions and instructional material necessary to:

- Understand information exchange content, semantics, and structure.
- Create and validate information exchanges defined by the IEPD.
- Identify the lineage of the IEPD and optionally its artifacts.

### 4.5 Access Control Frameworks and Scope

A key consideration at this phase in the project was describing the implementation of various ISM and NTK metadata in NIEM/IC Data Encodings and Service API. A key principal was that many different access control frameworks may be implemented on NIEM/IC Data Encodings and Services. Common in these approaches is the need to specify, maintain and manage roles, groups and policies in a NIEM-IC information exchange – for secure data exchange. By specifying Roles, ntk:AccessGroups, ism:classification and AccessPolicy PEPs, leveraging attributes defined in alignment of UIAS, can grant access to geospatial information exchange resources to some users, limited kinds of access to other users, and completely deny access to yet another set of users.

Each access control rule implemented by a different PEP grants (or denies) requests made by an individual or group of individuals, possibly depending on details associated with the request. Referring to one or more web services, rules can specify, for a given set of users, the conditions under which access is to be granted to them. A user can be associated with roles within an organization or with a group whose membership is known throughout the system.

The responsibility for implementing this access control is delegated to the PEP in this prototype NIEM/IC information exchange. NIEM/IC API responses and response pass through the PEPs, and each access control rule implemented by different PEPs grants (or denies) requests made by an individual or group of individuals, depending on the Roles, ntk:AccessGroups, ism:classification and AccessPolicy associated with the user making the request.

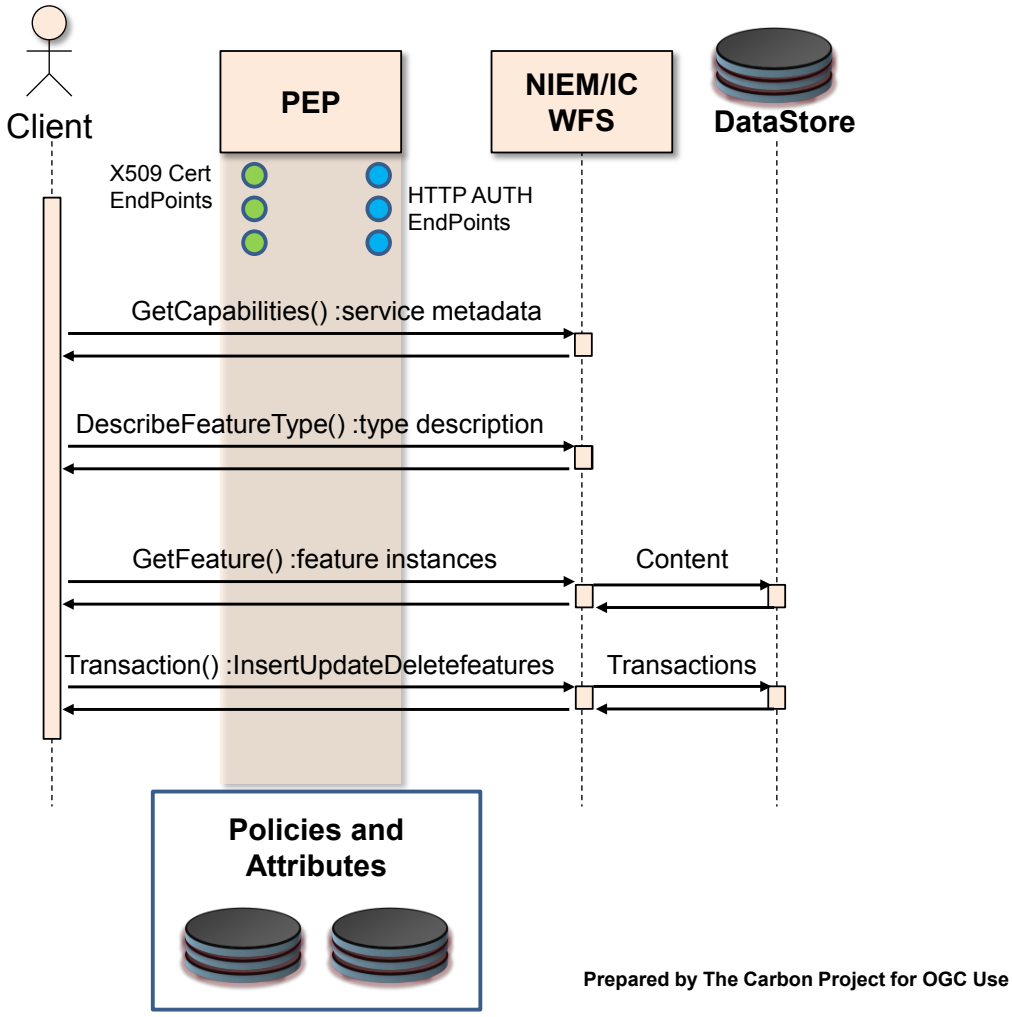



**Figure 4 –NIEM/IC Feature Processing API Operations and PEP Processing**

In addition, because rules will refer to user roles and names, security within NIEM/IC information exchange the test and demonstration implementation provides a way to name users and mechanisms to manage user identities, including the means by which users can be authenticated. A person is authenticated and assumes an identity by demonstrating knowledge of a secret (such as a password), or possession of some other information, that is associated with that identity.

NIEM/IC information exchange has a flexible authentication framework that supports multiple authentication methods. To authenticate a user known to an organization, and uses systems already used to authenticate users. This allows an organization to use existing authentication methods. For example, a user might be authenticated at an organization by providing a username/password (HTTP AUTH) that is recognized in the organization, or via X.509 certificates.

Key within this test and demonstration implementation is the OGC Attribute Store. The OGC Attribute Store implements the IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set (UIAS) specification. The specification documents a set of IC enterprise identity attributes and associated values that are required for participation in Intelligence Community Unified Authorization and Attribute Service (UAAS) architecture. Information about user and role assignment is stored in an LDAP. The data can be accessed via the OGC IdP Attribute Service interface.

With this access control framework in place the project also assessed how the principals of Attribute Based Access Control (ABAC) may be applied to NIEM/IC information exchange. ABAC is an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions. Attributes are characteristics of the subject, object, or environment conditions. Attributes contain information given by a name-value pair. A subject is a human user or NPE, such as a device that issues access requests to perform operations on objects. Subjects are assigned one or more attributes. An object is a system resource for which access is managed by the ABAC system, such as devices, files, records, tables, processes, programs, networks, or domains containing or receiving information. An operation is the execution of a function at the request of a subject upon an object. Operations include read, write, edit, delete, copy, execute, and modify. Policy is the representation of rules or relationships that makes it possible to determine if a requested access should be allowed, given the values of the attributes of the subject, object, and possibly environment conditions.[3]

As discussed above, the responsibility for implementing this access control is delegated to the PEP in this prototype NIEM/IC information exchange, with this document focusing on the API for NIEM/IC Feature Processing. NIEM/IC API responses and response pass through the PEPs, and each access control rule implemented by different PEPs grants (or denies) requests.

### 4.6 NIEM-IC Feature Processing API

To support testing of the NIEM/IC Feature Processing API a cloud-based test environment was established by The Carbon Project. PEPs from multiple Participants including Secure Dimensions, con terra and Jericho Systems were established and then accessed the 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.

---

[3] Guide to Attribute Based Access Control (ABAC) Definition and Considerations
http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

Hands-on collaborative engineering yielded the following set of parameters to guide the development of NIEM/IC Feature Processing API.

### 4.6.1 Operation Request Encoding

The encoding of operation requests in Geo4NIEM used HTTP GET with KVP encoding and HTTP POST with XML and/or KVP encoding. Table 1 summarizes the operations and their encoding methods and their status in the WFS specification.

**Table 1 — Operation Request Encoding**

| Operation | Request Encoding |
|---|---|
| GetCapabilities (required) | Mandatory KVP-GET |
| DescribeFeatureType (required) | Mandatory KVP–GET |
| GetFeature (required) | Mandatory XML-POST and KVP-GET |
| Transaction (optional) | Mandatory XML-POST |

### 4.6.2 GetCapabilities operation

This is a standard WFS requirement, with both the request and response requirements unchanged by the NIEM/IC Feature Processing API.

The GetCapabilities operation allows clients in Geo4NIEM to retrieve service metadata from a NIEM/IC Feature Processing server. The response to a GetCapabilities request was an XML document containing service metadata about the server, including specific information about the feature types it can service, and the supported operations on each feature type.

A sample Capabilities document from a NIEM/IC Feature Processing server is included as Annex A to this ER.

#### 4.6.2.1 Operation request

The Figure below describes the schema of a GetCapabilities request.

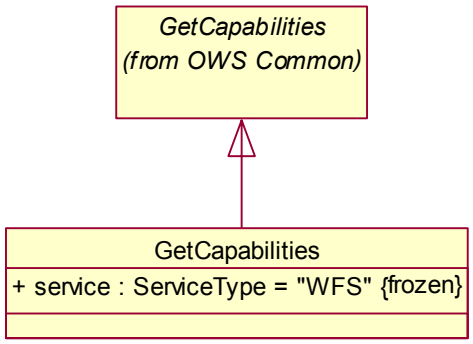



**Figure 5 - GetCapabilities request**

#### 4.6.2.2 XML encoding

The following XML Schema fragment defines the XML-encoding of a GetCapabilities request:

```
<xsd:element name="GetCapabilities"
  type="wfs:GetCapabilitiesType"/>
<xsd:complexType name="GetCapabilitiesType">
   <xsd:complexContent>
      <xsd:extension base="ows:GetCapabilitiesType">
         <xsd:attribute name="service" type="ows:ServiceType"
  use="required" fixed="WFS"/>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

The base type, ows:GetCapabilitiesType, is defined in the OWS Common Implementation Specification (see OGC 06-121r3:2010, 7.2.4).

#### 4.6.2.3 KVP encoding

The KVP encoding of the GetCapabilities request was as specified in OGC 06-121r3:2009, 7.2.2. A sample GetCapabilities request from Testbed 11 is shown below:

```
http://niems.someniemwebsites.net/wfs?SERVICE=WFS&REQUEST=GetCapa
bilities&VERSION=1.1.0
```

#### 4.6.2.4 Response

The NIEM/IC Feature Processing API creates no additional requirements of the WFS GetCapabilities operation beyond the ability to serve the required NIEM/IC featuretypes.

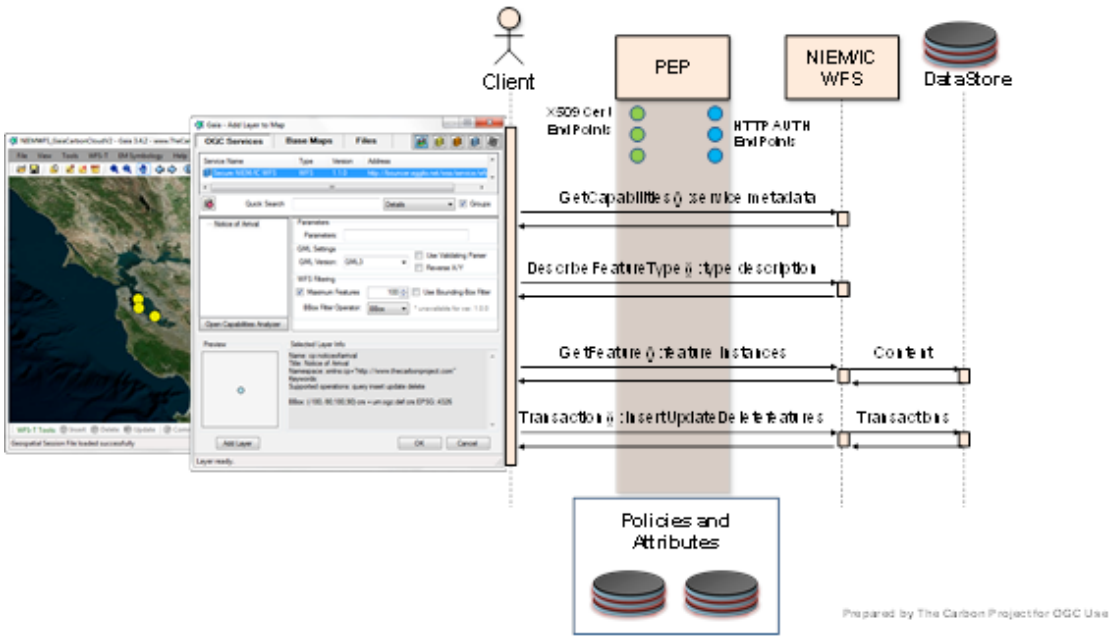



**Figure 6 - Gaia accessing NIEM/IC GetCapabilities on CarbonCloud WFS, through Secure Dimensions, con terra and Jericho Systems PEPs**

#### 4.6.2.5 Security

In the case where the served content is classified or security tagged, the full list of data offerings may only be returned if the user issues the GetCapabilities request as a recognized user. This implies the use of the publically assessable Capabilities instance document that does not contain the security tagged data offerings but outlines the GetCapabilities operation as protected.

### 4.6.3 DescribeFeatureType operation

This is a standard WFS requirement, with both the request and response requirements unchanged by the NIEM/IC Feature Processing API. However, the project is assessing

the potential need for NIEM and/or IC schemas to be present locally on a NIEM/IC Feature Processing server.

The DescribeFeatureType operation allows NIEM/IC Feature Processing clients to retrieve schema descriptions which define how the NIEM/IC Feature Processing server will generate feature instances on output (in response to GetFeature requests).

A sample, preliminary DescribeFeatureType document from a NIEM/IC Feature Processing server is included as Annex C to this ER. Due to ongoing sample IEPD data development efforts and the schedule of Testbed 11 a full DescribeFeatureType may be updated based on ongoing engineering activities.

#### 4.6.3.1 Operation request

The Figure below describes the schema of a DescribeFeatureType request.

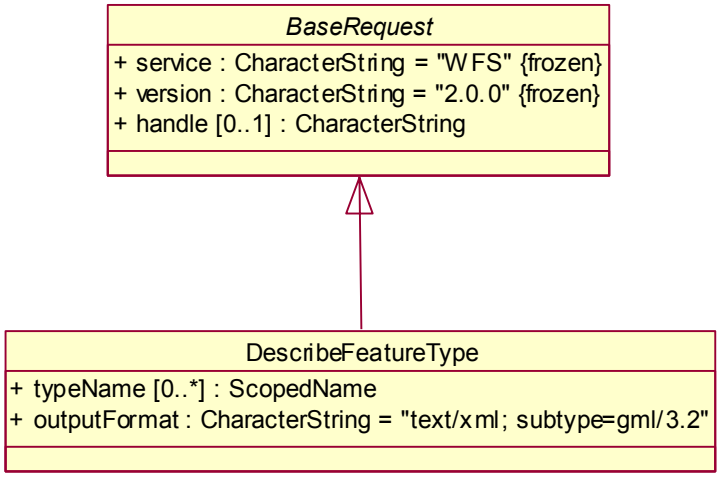



**Figure 7 - DescribeFeatureType request**

#### 4.6.3.2 XML encoding

The following XML Schema fragment defines the XML encoding of a DescribeFeatureType request:

```
<xsd:element name="DescribeFeatureType"
  type="wfs:DescribeFeatureTypeType"/>
<xsd:complexType name="DescribeFeatureTypeType">
   <xsd:complexContent>
      <xsd:extension base="wfs:BaseRequestType">
         <xsd:sequence>
            <xsd:element name="TypeName" type="xsd:QName"
                         minOccurs="0" maxOccurs="unbounded"/>
```

```
            </xsd:sequence>
            <xsd:attribute name="outputFormat" type="xsd:string"
                           default="application/gml+xml;
    version=3.2"/>
        </xsd:extension>
     </xsd:complexContent>
  </xsd:complexType>
```

Table 2 defines the KVP encoding for a DescribeFeatureType request.

**Table 2 — DescribeFeatureType KVP encoding**

| URL Component | O/M[a] | Description |
|---|---|---|
| *Common Keywords* (REQUEST=DescribeFeatureType) | | See Table 7. (Only keywords for all operations or the DescribeFeatureType operation.) |
| TYPENAMES | O | A comma separated list of feature types to describe. If no value is specified, the complete application schema offered by the server shall be described. |
| OUTPUTFORMAT | O | Shall support the value "application/gml+xml; version=3.2" indicating that a GML (see ISO19136:2007) application schema shall be generated. A server may support other values to which this International Standard does not assign any meaning. |

[a] O = Optional, M = Mandatory

#### 4.6.3.3 outFormat parameter

For KVP-encoded requests the outputFormat parameter may be encoded using the keyword OUTPUTFORMAT.

The outputFormat parameter was used for the NIEM/IC Feature Processing, server to advertise that multiple output formats, including versions with TDF encoding called ‘NIEMS’, are supported. Specifically, the project assessed two ways of delivering the data encoding:

- NIEM IEP with Information Security Marking metadata and XML for Need-To-Know metadata as wfs:FeatureCollection (called the ‘NIEM/IC WFS’)
- NIEM IEP with Information Security Marking metadata and XML for Need-To-Know metadata as wfs:FeatureCollection wrapped in TDF (made available via the outputFormat parameter called ‘NIEMS’)

This approach provided the NIEM/IC WFS as a default option since it was assessed this model may be more readily handled by server and client applications during initial testing.

#### 4.6.3.4 Response

In first prototypes, the NIEM/IC Feature Processing API creates no additional requirements of the WFS DescribeFeatureType operation beyond the ability to serve the required schema description. However, ongoing engineering efforts may identify the need to provide additional requirements on the WFS DescribeFeatureType operation for the NIEM/IC Feature Processing API with respect to TDO outputFormat and local schemas.

[This section may be adjusted based on engineering activities in TB11].

### 4.6.4 GetFeature operation

The GetFeature operation returns a selection of features from a NIEM/IC data store. A NIEM/IC Feature Processing server processes a GetFeature request and returns a response document to the client that contains a wfs:FeatureCollection that contains 0 or more gml:featureMember representing geographic features that satisfy the query expressions specified in the request. The wfs:FeatureCollection includes Information Security Marking metadata and XML for Need-To-Know metadata.

The use of the term gml:featureMember is based on the designation provided for in NIEM 3.0. This designation may be slightly different based on the Version of OGC WFS Specification implemented.

A sample GetFeature response from a NIEM/IC Feature Processing server is included as Annex B to this ER.

A sample response to OutputFormat type identified as ‘NIEMS’ (for TDF sample) t from a NIEM/IC Feature Processing server is included as Annex D to this ER.

#### 4.6.4.1 Operation request

The figure below describes the schema of a GetFeature request.

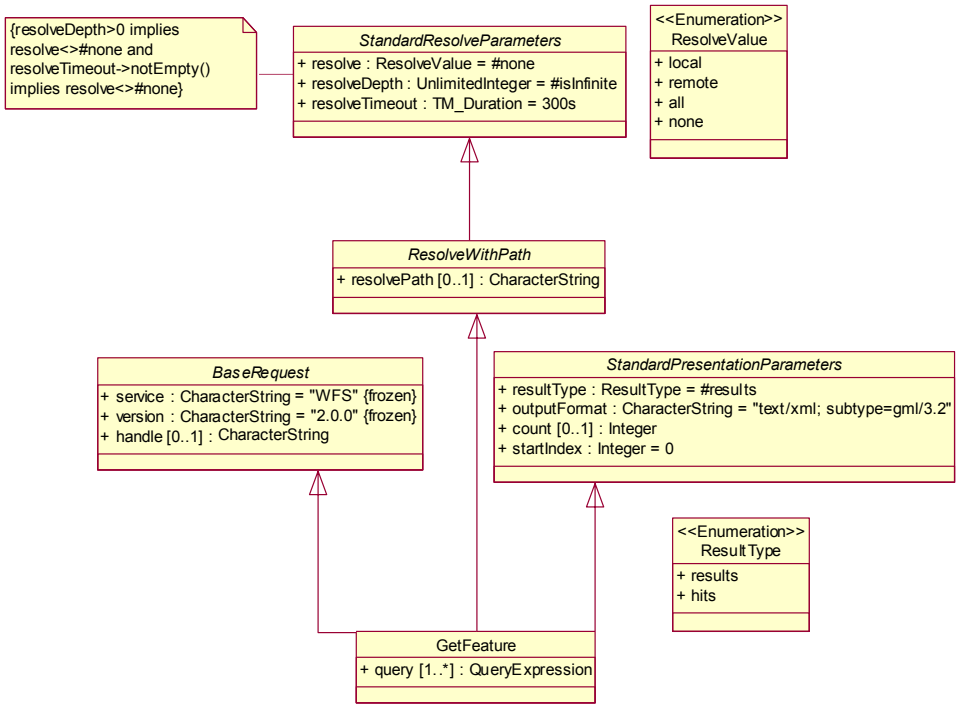



**Figure 8 - getFeature request**

#### 4.6.4.2 XML Encoding

The XML encoding of a GetFeature request is defined by the following XML Schema fragment:

```
<xsd:element name="GetFeature" type="wfs:GetFeatureType"/>
<xsd:complexType name="GetFeatureType">
   <xsd:complexContent>
      <xsd:extension base="wfs:BaseRequestType">
         <xsd:sequence>
            <xsd:element ref="fes:AbstractQueryExpression"
                         maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:attributeGroup
   ref="wfs:StandardPresentationParameters"/>
         <xsd:attributeGroup
   ref="wfs:StandardResolveParameters"/>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

A sample, very simple, request from Testbed 11 is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<GetFeature xmlns=" http://www.opengis.net/wfs " xmlns:mda="
     http://release.niem.gov/niem/domains/maritime/3.0/mda/ "
     xmlns:ogc=" http://www.opengis.net/ogc " xmlns:xsi="
     http://www.w3.org/2001/XMLSchema-instance " xmlns:gml="
     http://www.opengis.net/gml " service="WFS" version="1.0.0"
     outputFormat="GML3" maxFeatures="100" handle="" >
<Query typeName="mda:noticeofarrival" srsName="EPSG::4326" />
</GetFeature>
```

#### 4.6.4.3 KVP encoding

Table 3 defines the KVP-encoding for a GetFeature request.

**Table 3 — Keywords for GetFeature KVP-encoding**

| URL Component | Description |
|---|---|
| *Common Keywords* (REQUEST=GetFeature) | See Table 7 for additional parameters that may be used in a KVP-encoded GetFeature request. |
| *Standard Presentation Parameters* | See Table 5.(09-025r2) |
| *Standard Resolve Parameters* | See Table 6. (09-025r2) |
| *Adhoc Query Keywords* (Mutually exclusive with Stored Query Keywords) | See Table 8. (09-025r2) |
| *Stored Query Keywords* (Mutually exclusive with Adhoc Query Keywords) | See Table 10. (09-025r2) |

A sample, very simple, request from Testbed 11 is shown below:

```
https://ows11.secure-
dimensions.com/noa/basic?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFea
ture&TYPENAME=mda%3Anoticeofarrival&NAMESPACE=xmlns%28mda%3Dhttp%
3A%2F%2Frelease.niem.gov%2Fniem%2Fdomains%2Fmaritime%2F3.0%2Fmda%
2F%29&OUTPUTFORMAT=text%2Fxml%3B%20subtype%3Dgml%2F3.1.1
```

#### 4.6.4.4 Response

In first prototypes, the NIEM/IC Feature Processing API specifies the response to this request as an XML document with a root element, wfs:FeatureCollection for information exchange. The wfs:FeatureCollection contains 1 or more gml:featureMembers representing geographic feature and includes Information Security Marking metadata and

XML for Need-To-Know metadata. A sample GetFeature response from a NIEM/IC Feature Processing server is included as Annex B to this ER. A sample response to OutputFormat type identified as 'NIEMS' (for TDF sample) t from a NIEM/IC Feature Processing server is included as Annex D to this ER. Examples of this operation in action during the testbed are provided below.

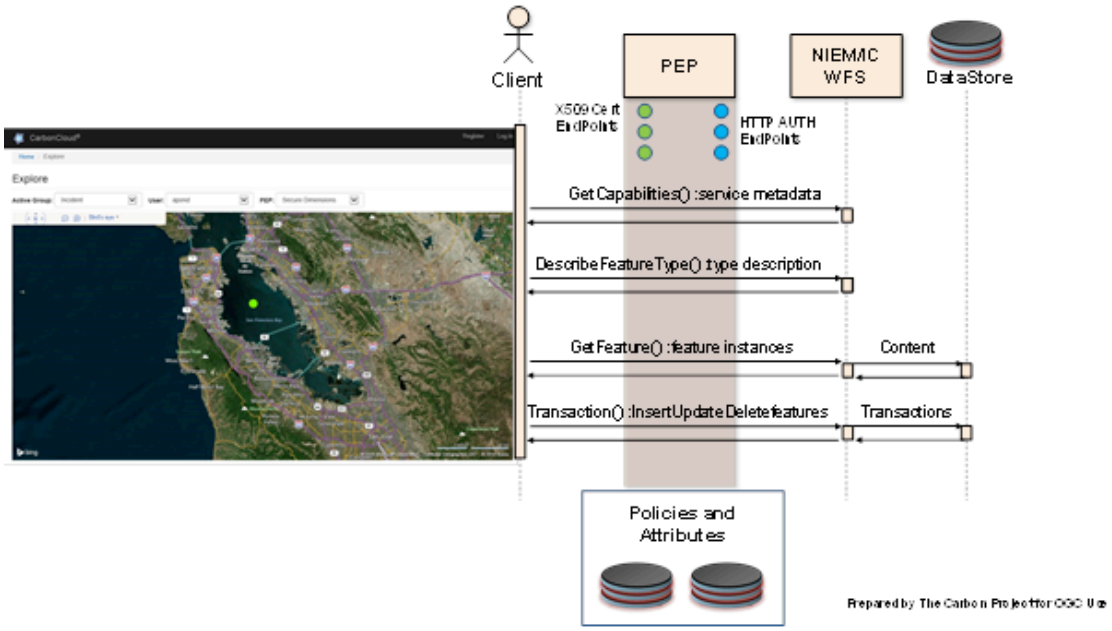



**Figure 9 - CarbonCloud Web Client getting Incident features from NIEM/IC Feature Processing server via Secure Dimensions PEP**

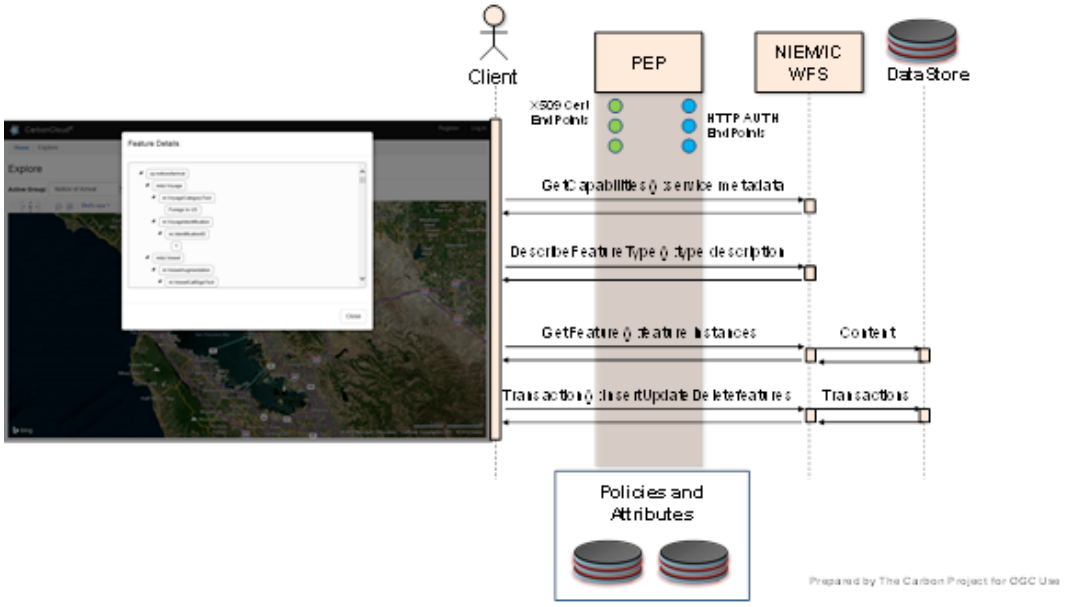



**Figure 10 - CarbonCloud Web Client getting NOA features from NIEM/IC Feature Processing server via Secure Dimensions PEP**

### 4.6.5 Transaction operation

The Transaction operation is used to describe data transformation operations to be applied to feature instances under the control of a NIEM/IC Feature Processing server. Using the Transaction operation clients can create, modify, replace and delete features in the NIEM/IC Feature Processing server's data store. GML is used as the canonical representation of features. When the transaction has been completed, a NIEM/IC Feature Processing server can generate an XML response document indicating the completion status of the operation. NIEM/IC Feature Processing services that support the optional Transaction operation can advertise this fact in their capabilities document.

#### 4.6.5.1 Operation request

The figure below describes the schema of a Transaction request.

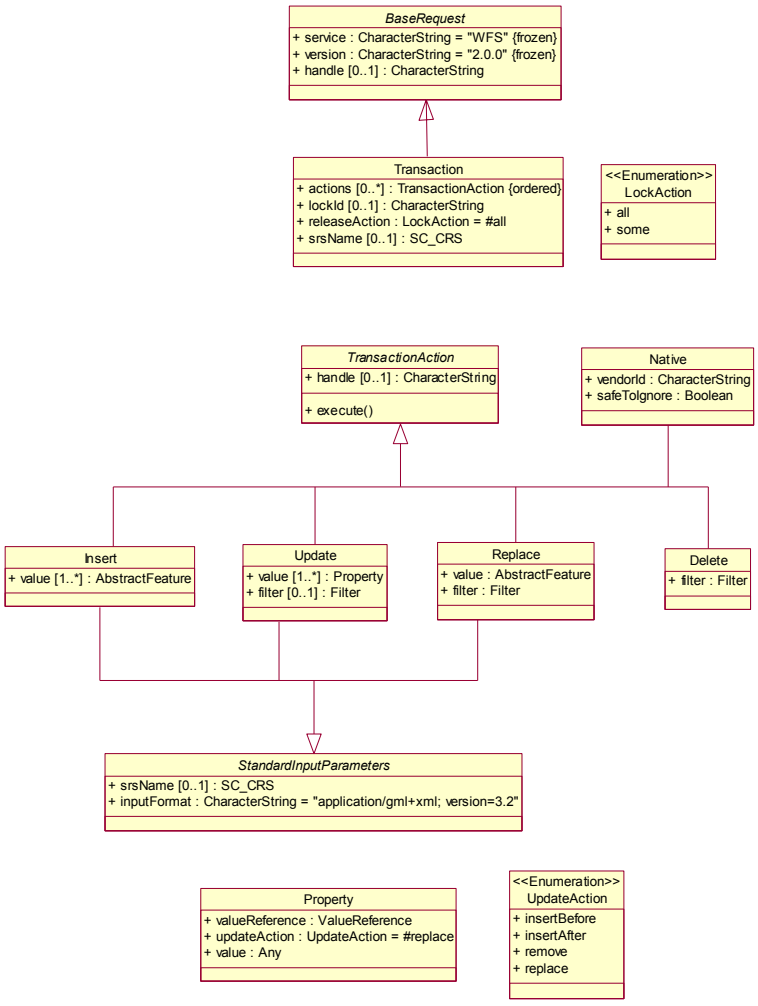



**Figure 11- Transaction request**

#### 4.6.5.2 XML encoding

The XML encoding of a Transaction request is defined by the following XML Schema fragment:

```
  <xsd:element name="Transaction" type="wfs:TransactionType"/>
   <xsd:complexType name="TransactionType">
      <xsd:complexContent>
         <xsd:extension base="wfs:BaseRequestType">
            <xsd:sequence>
               <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                  <xsd:element
     ref="wfs:AbstractTransactionAction"/>
               </xsd:sequence>
```

```
            </xsd:sequence>
            <xsd:attribute name="lockId" type="xsd:string"/>
            <xsd:attribute name="releaseAction"
      type="wfs:AllSomeType" default="ALL"/>
            <xsd:attribute name="srsName" type="xsd:string"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:element name="AbstractTransactionAction"
      type="wfs:AbstractTransactionActionType" abstract="true"/>
   <xsd:complexType name="AbstractTransactionActionType"
      abstract="true">
      <xsd:attribute name="handle" type="xsd:string"/>
   </xsd:complexType>
```

#### 4.6.5.3 Insert action

##### 4.6.5.3.1 XML encoding

The following XML Schema fragment shows a wfs:Insert element:

```
   <xsd:element name="Insert" type="wfs:InsertType"
      substitutionGroup="wfs:AbstractTransactionAction"/>
   <xsd:complexType name="InsertType">
      <xsd:complexContent>
         <xsd:extension base="wfs:AbstractTransactionActionType">
            <xsd:sequence>
               <xsd:any namespace="##other"
      maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup
      ref="wfs:StandardInputParameters"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
```

The wfs:Insert element is used to create new feature instances in a NIEM/IC Feature Processing service's data store. Multiple wfs:Insert elements may be enclosed in a single Transaction request and multiple feature instances may be created using a single wfs:Insert element.

A sample from Testbed 11 is shown below:

```
<?xml version=\"1.0\"?><wfs:Transaction version=\"1.1.0\"
service=\"WFS\"
xmlns:cp=\"http://www.thecarbonproject.com\"
xmlns:gml=\"http://www.opengis.net/gml\"
xmlns:ogc=\"http://www.opengis.net/ogc\"
xmlns:wfs=\"http://www.opengis.net/wfs\">
  <wfs:Insert>
    <cp:incident xmlns:nc=\"http://release.niem.gov/niem/niem-
core/3.0/\"
xmlns:emevent=\"http://mitre.org/emevent/0.9/\"
xmlns:mof=\"http://example.com/milops/1.1/\"
xmlns:cad=\"http://mitre.org/emevent/0.9/cad2cad/\">
      <mof:EventID>

<nc:IdentificationID>543301@richmondfd.richmond.ca.us</nc:Identif
icationID>
      </mof:EventID>
      <emevent:EventTypeDescriptor>

<emevent:EventTypeCode>BITS.REPORT.EM.DISPATCH.FIRE.HAZARD.WATER<
/emevent:Ev
entTypeCode>
      </emevent:EventTypeDescriptor>
      <mof:EventLocation>
        <mof:LocationCylinder>
          <mof:LocationPoint>
            <gml:Point>
              <gml:pos srsDimension=\"2\">-122.4031
37.7681</gml:pos>
            </gml:Point>
          </mof:LocationPoint>

<mof:LocationCylinderRadiusValue>1</mof:LocationCylinderRadiusVal
ue>

<mof:LocationCylinderHalfHeightValue>1</mof:LocationCylinderHalfH
eightValue>

<mof:LocationCreationCode>HUMAN.CREATED</mof:LocationCreationCode
>
        </mof:LocationCylinder>
      </mof:EventLocation>
      <mof:EventValidityDateTimeRange>
        <nc:StartDate>
          <nc:DateTime>2025-12-16T12:05:36</nc:DateTime>
        </nc:StartDate>
        <nc:EndDate>
          <nc:DateTime>2025-12-16T12:05:36</nc:DateTime>
        </nc:EndDate>
      </mof:EventValidityDateTimeRange>
```

```
      <mof:EventMessageDateTime>
        <nc:DateTime>2025-12-16T12:05:36</nc:DateTime>
      </mof:EventMessageDateTime>
      <emevent:USNGCoordinate>

<emevent:USNGCoordinateID>10SEH5931806506</emevent:USNGCoordinate
ID>

<emevent:USNGEastingValue>59318</emevent:USNGEastingValue>

<emevent:USNGNorthingValue>06506</emevent:USNGNorthingValue>

<nc:GeographicDatumText>http://metadata.ces.mil/mdr/ns/GSIP/crs/W
GS84E_3D</n
c:GeographicDatumText>
        <emevent:USNGGridZoneID>10S</emevent:USNGGridZoneID>

<emevent:USNGGridZoneSquareID>EH</emevent:USNGGridZoneSquareID>
      </emevent:USNGCoordinate>
      <emevent:EventComment>
        <nc:DateTime>2025-12-16T12:05:36</nc:DateTime>
        <nc:OrganizationIdentification>

<nc:IdentificationID>richmondfd.richmond.ca.us</nc:Identification
ID>
        </nc:OrganizationIdentification>
        <nc:CommentText>Water rescue from submerged
vehicle.</nc:CommentText>
      </emevent:EventComment>
      <cad:IncidentDetail>
        <cad:IncidentStatus>

<cad:IncidentPrimaryStatus>ACTIVE</cad:IncidentPrimaryStatus>
          <cad:IncidentPulsePointStatus>ON
SCENE</cad:IncidentPulsePointStatus>
        </cad:IncidentStatus>
        <cad:IncidentOwningOrganization>
          <nc:OrganizationIdentification>

<nc:IdentificationID>richmondfd.richmond.ca.us</nc:Identification
ID>
          </nc:OrganizationIdentification>
          <cad:IncidentIdentifier>
            <nc:IdentificationID>543301</nc:IdentificationID>
          </cad:IncidentIdentifier>
        </cad:IncidentOwningOrganization>
        <cad:IncidentLocationExtension>
          <nc:Address>
            <nc:LocationStreet>
              <nc:StreetNumberText>2068-
2071</nc:StreetNumberText>
```

```
              <nc:StreetName>Cypress</nc:StreetName>
              <nc:StreetCategoryText>Ave</nc:StreetCategoryText>
            </nc:LocationStreet>
            <nc:LocationCityName>San Pablo</nc:LocationCityName>
            <nc:LocationCountyCode>13</nc:LocationCountyCode>

<nc:LocationStateFIPS52AlphaCode>CA</nc:LocationStateFIPS52AlphaC
ode>

<nc:LocationCountryFIPS104Code>US</nc:LocationCountryFIPS104Code>
          </nc:Address>

<cad:AddressIntersectionIndicator>false</cad:AddressIntersectionI
ndicator>
        </cad:IncidentLocationExtension>
      </cad:IncidentDetail>
    </cp:incident>
  </wfs:Insert>
</wfs:Transaction>
```

#### 4.6.5.4 Update action

##### 4.6.5.4.1 XML encoding

The following XML Schema fragment shows a wfs:Update element:

```
   <xsd:element name="Update" type="wfs:UpdateType"
      substitutionGroup="wfs:AbstractTransactionAction"/>
   <xsd:complexType name="UpdateType">
      <xsd:complexContent>
         <xsd:extension base="wfs:AbstractTransactionActionType">
            <xsd:sequence>
               <xsd:element ref="wfs:Property"
     maxOccurs="unbounded"/>
               <xsd:element ref="fes:Filter" minOccurs="0"/>
            </xsd:sequence>
            <xsd:attribute name="typeName" type="xsd:QName"
     use="required"/>
            <xsd:attributeGroup
     ref="wfs:StandardInputParameters"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:element name="Property" type="wfs:PropertyType"/>
   <xsd:complexType name="PropertyType">
      <xsd:sequence>
```

```
         <xsd:element name="ValueReference">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:extension base="xsd:string">
                     <xsd:attribute name="action"
      type="wfs:UpdateActionType" default="replace"/>
                  </xsd:extension>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="Value" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="UpdateActionType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="insertBefore"/>
         <xsd:enumeration value="insertAfter"/>
         <xsd:enumeration value="remove"/>
         <xsd:enumeration value="replace"/>
      </xsd:restriction>
   </xsd:simpleType>
```

#### 4.6.5.5 Delete action

##### 4.6.5.5.1 XML encoding

The following XML Schema fragment declares the wfs:Delete element:

```
   <xsd:element name="Delete" type="wfs:DeleteType"
      substitutionGroup="wfs:AbstractTransactionAction"/>
   <xsd:complexType name="DeleteType">
      <xsd:complexContent>
         <xsd:extension base="wfs:AbstractTransactionActionType">
            <xsd:sequence>
               <xsd:element ref="fes:Filter"/>
            </xsd:sequence>
            <xsd:attribute name="typeName" type="xsd:QName"
      use="required"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
```

The wfs:Delete element is used to encode a delete request that removes one or more feature instances, of a specified feature type, from being queryable to a client application using the GetFeature.

### 4.6.6 Response

#### 4.6.6.1 Response Semantics

In response to a Transaction request, a NIEM/IC Feature Processing service can generate an XML document indicating the termination status of the transaction. In addition, if the Transaction request includes wfs:Insert elements, then the NIEM/IC Feature Processing shall report the feature identifiers of all newly created features.

The figure below describes the response to a Transaction operation.

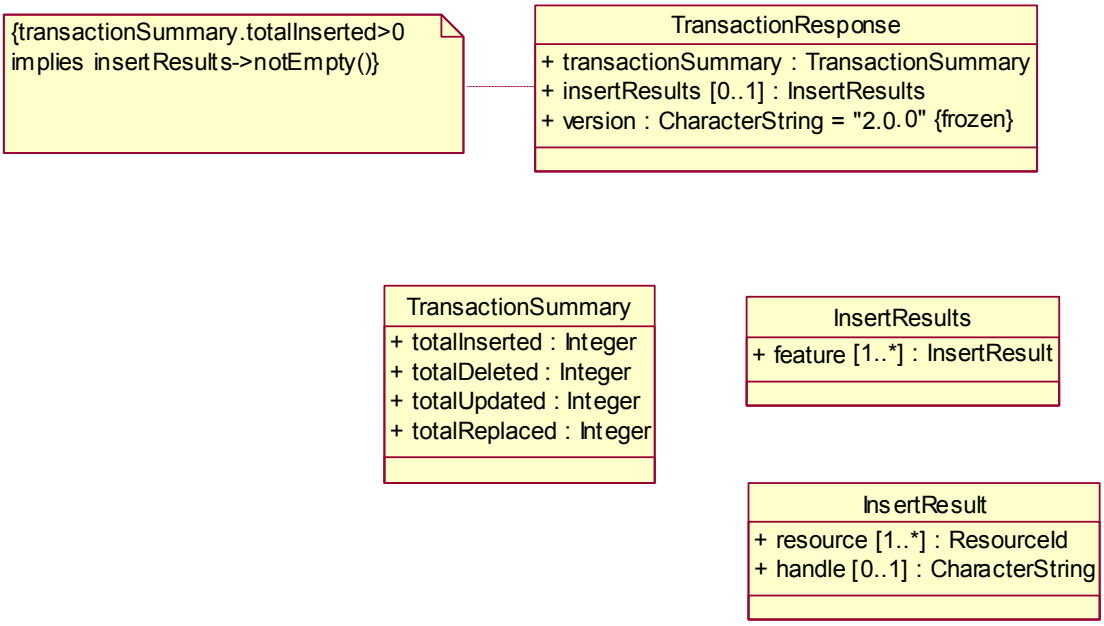



**Figure 12 - Response to a Transaction operation**

An example of a Transaction in use during Testbed 11 is shown below.

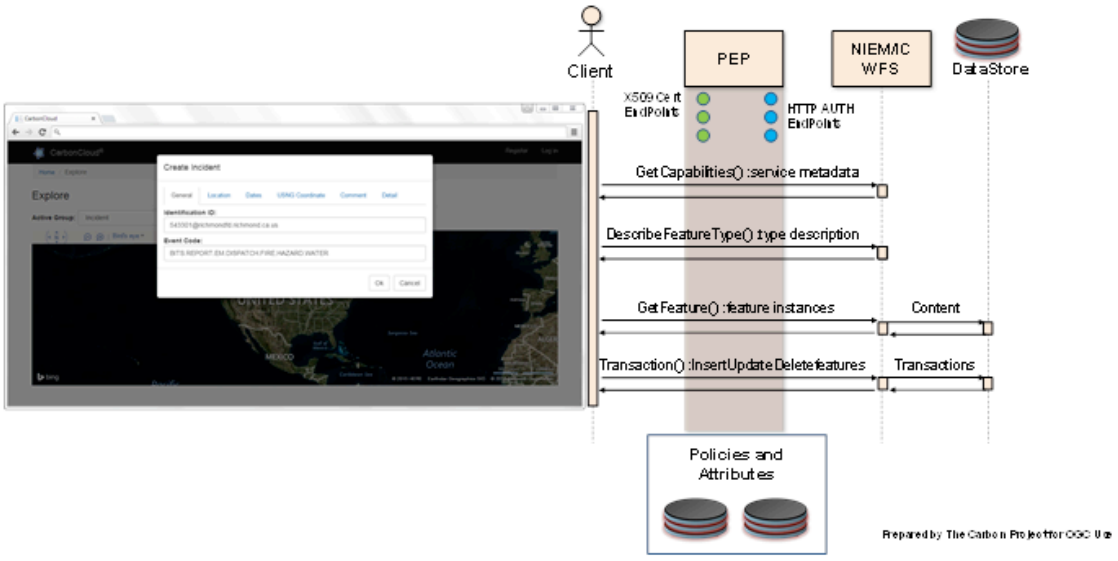



**Figure 13 - con terra PEP in The Carbon Project web client, executing WFS Transactions**

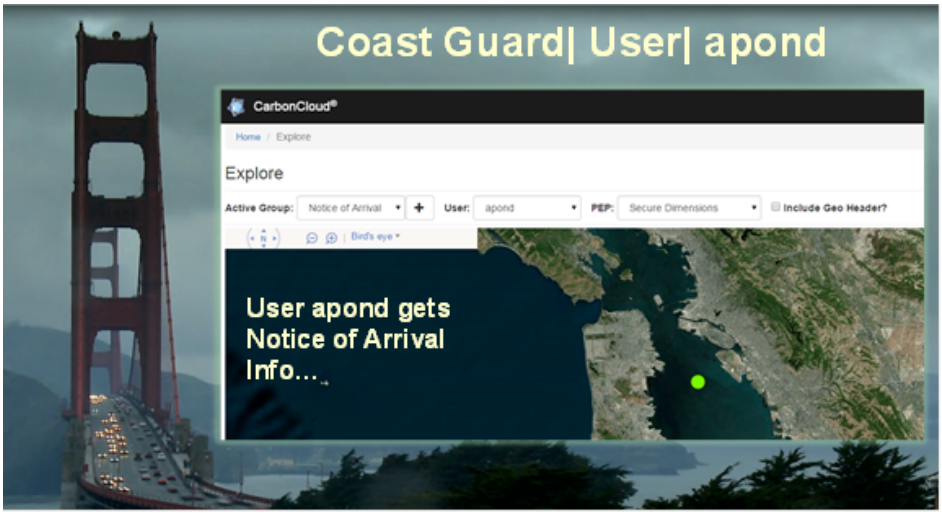
## 5 Other Examples of NIEM/IC Data Encoding in Use

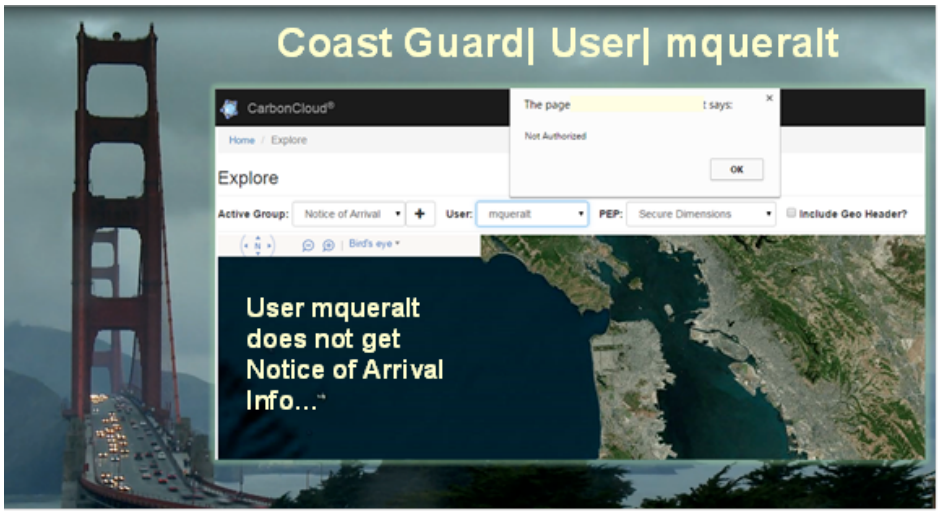
This section provides examples of the NIEM/IC Data Encoding in use by applications and services provided by Testbed 11 participants including a cloud-based test WFS from The Carbon Project and PEPs from multiple Participants including Secure Dimensions, con terra and Jericho Systems.

The following examples provide a very brief, sample overview of the demonstration scenario.

For a complete description please see the Test and Demonstration ER[4], and the actual Testbed 11 Geo4NIEM Demonstration videos presented at the June 2015 OGC Technical Committee meeting in Boulder, CO.

[4] Discussed in ER 15-050.

The basic flow of events is the scenario begins when the client authenticates by presenting a user certificate issued by the Testbed Certificate Authority. Then:

1. The PEP requests user attributes via a SAML attribute query to the OGC Identity Provider (IdP) Testbed Attribute Service.
2. The OGC IdP returns the user attributes to the PEP in the form of a SAML response; the PEP then associates the attributes with the client session. In this scenario instance, the user attributes are

| | |
|---|---|
| uid | tjacobs |
| CountryOfAffiliation | US |
| FineAccessControls | Restricted |
| AICP | FALSE |
| DigitalIdentifier | cn=Tim Jacobs,ou=SolanoOES,o=Solano County,c=US |
| Role | SEMS-CA-Msn-SolanoCounty-MAC |
| EntityType | GOV |
| DutyOrganization | SLT |
| Clearance | U |
| AdminOrganization | SLT |
| isICMember | FALSE |
| mail | tjacobs@geo4niem.example.com |

3. The client sends GetCapabilities, DescribeFeatureType, and GetFeature requests to the PEP (which is acting as a WFS proxy). Steps 4-9 describe the handling of the GetFeature request. (Handling of the other service invocations is similar and simpler.)
4. The PEP issues a XACML 2.0 compliant Authorization Decision Request to the PDP, including the user attributes from step 2 and the geolocation of the client.
5. The PDP retrieves the GeoXACML Policy from the Testbed Policy Store. In this scenario, the policy rules are expressed in terms of the user attributes for location, clearance, and role.

6. The PDP creates the Authorization Decision based on the policy and the user attributes. This may be Deny, Permit, or Permit with Obligations for rewriting rules that must be applied to the response from the WFS before the featureCollection is sent to the client. In this scenario, the rewriting rule removes elements classified C or above, and removes elements that have NTK portion marks which do not grant access for the role SEMS-CA-Msn-SolanoCounty-MAC.
7. If permitted, the PEP forwards the GetFeature request to the WFS server.
8. The WFS server returns a featureCollection to the PEP. Depending on the outputFormat parameter of the GetFeatureCollection request, the members of the featureCollection may be NIEM IEPs (the default), or TDOs with a NIEM IEP payload (with "niems" outputFormat).
9. The PEP executes any Obligations by applying any required rewriting rules to the featureCollection. These rules can have the effect of redacting elements that are classified above the user's clearance. In this scenario, the rewriting rule removes elements classified C or above, and removes elements that have NTK portion marks which do not grant access for the role SEMS-CA-Msn-SolanoCounty-MAC. The result is returned to the client as the output of the GetFeature request.

An architecture for this demonstration flow is provided below.

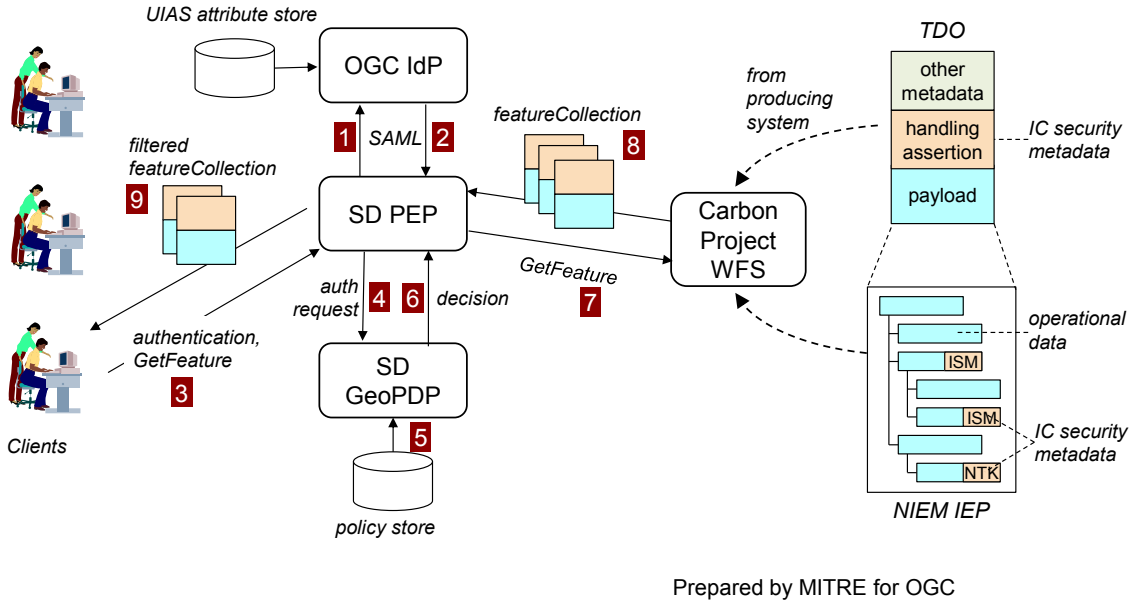



**Figure 14 - Sample Geo4NIEM Testbed 11 Demonstration Flow for one PEP**

### 5.1 The Carbon Project

The Carbon Project implemented the NIEM/IC Feature Processing API, 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 graphic below.

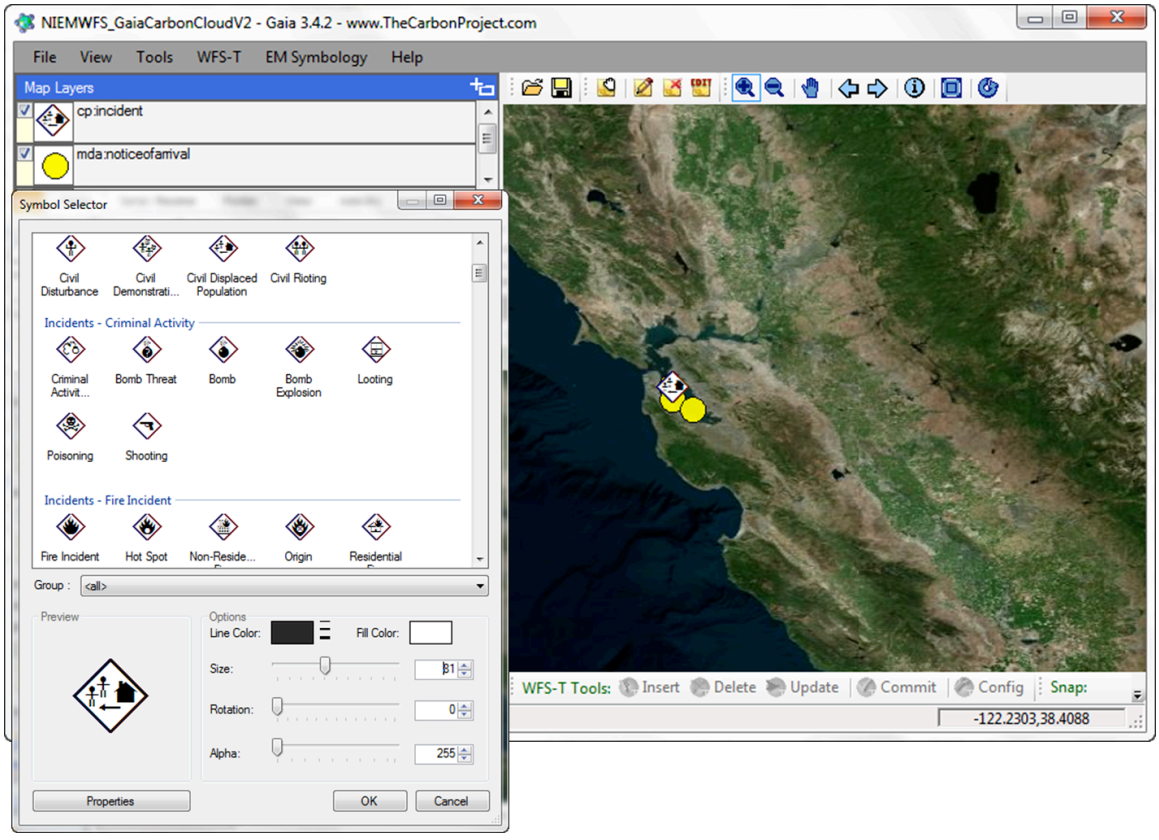



**Figure 15 - Incident and Notice of Arrival content from The Carbon Project NIEM/IC WFS in Gaia**

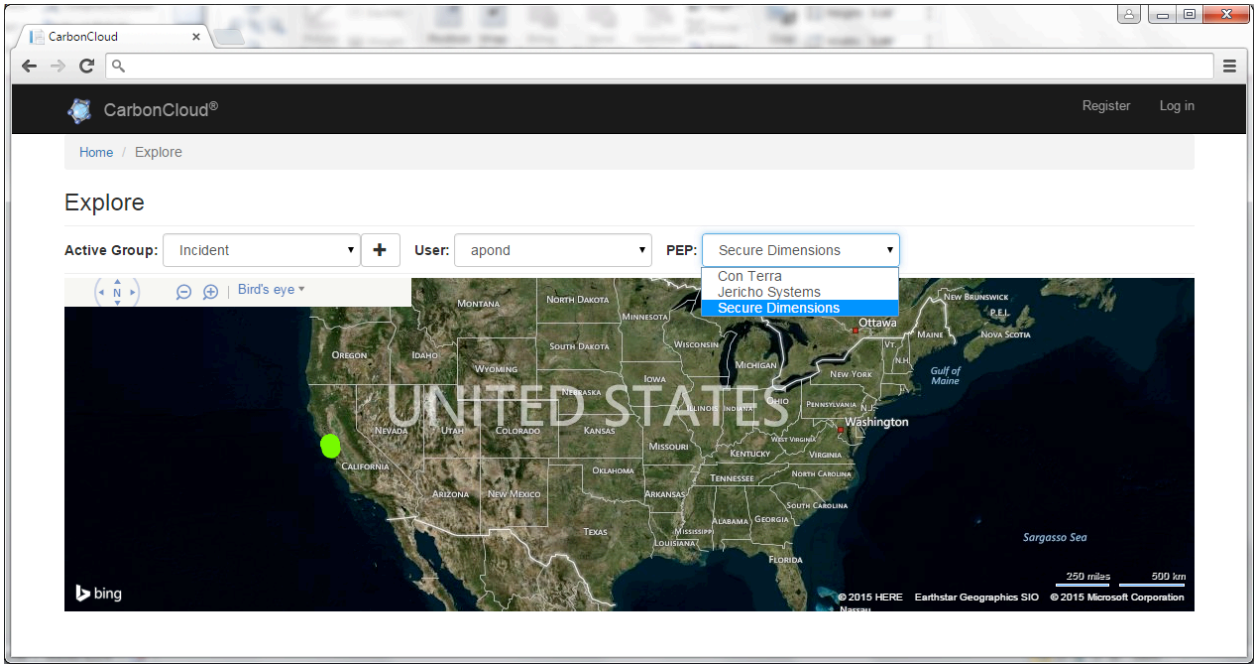



**Figure 16 - Web Client from The Carbon Project accessing NIEM/IC Data Encoding from Secure Dimensions, con terra and Jericho Systems PEP**

## 5.2 Secure Dimensions

Secure Dimensions implemented the NIEM/IC Feature Processing API. Examples with a simulated geographic location included in the NIEM/IC Feature Processing API are provided below. This 'GeoHeader' allowed the Testbed to assess NIEM/IC Feature Processing API requests with geographic access control rules implemented.

For example, a rule that allowed access only to users in San Francisco to add WFS Transactions, implemented in the Secure Dimensions PEP is shown below.

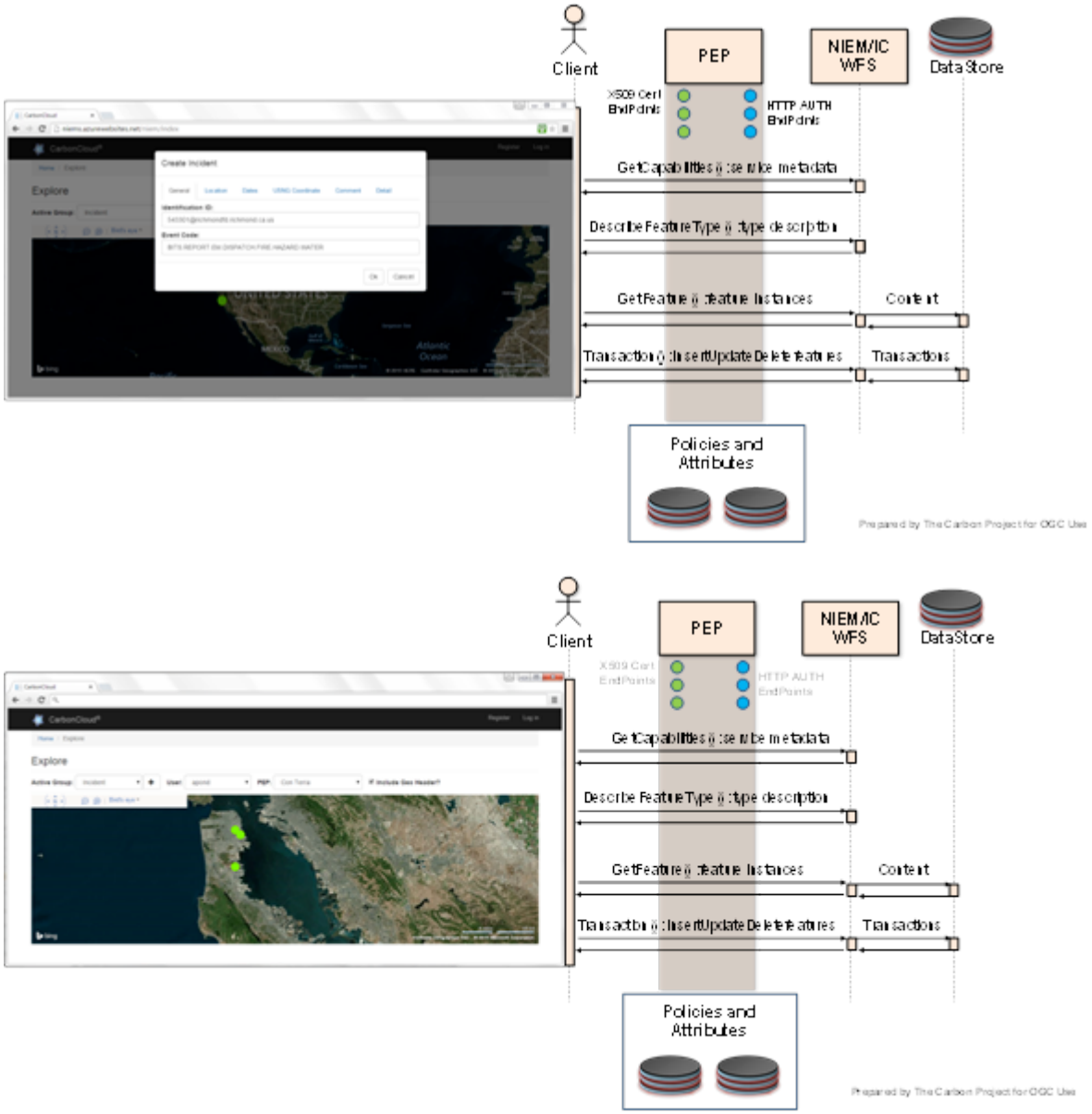



**Figure 17 - Web Client from The Carbon Project accessing Secure Dimensions PEP and executing WFS Transactions for NIEM/IC Incident encodings.**

There are several scenarios that were tested to provide or deny access based upon the location of the client. The client passes a location (lat/lon) to the PEP, and if not, the location will be determined by geolocating the IP address for the client (understanding that this is not accurate). By comparing the location with an allowed polygon, if the client is in the polygon(s), access should be allowed based upon the above filtering rules, if the client is outside the polygon(s) all access should be denied. The 'GeoHeader' followed this format:

```
Location: <gml:Point xmlns:gml="http://www.opengis.net/gml"
gml:id="TownHallSF" srsName="EPSG:4326"><gml:pos
srsDimension="2">37.77925 -122.419222</gml:pos></gml:Point>
```

or

```
Location: <gml:Point xmlns:gml="http://www.opengis.net/gml"
gml:id="WashingtonMonument" srsName="EPSG:4326"><gml:pos
srsDimension="2">38.889444 -77.035278</gml:pos></gml:Point>
```

### 5.3 Con terra

con terra implemented the NIEM/IC Feature Processing API. The security.manager PEP from con terra acts like a proxy component for the NIEM/IC Feature Processing API, exposing service endpoints which can be accessed by client applications. After authentication and authorization, these requests are passed on to the protected service. Responses are passed back the same way. An example is provided below.

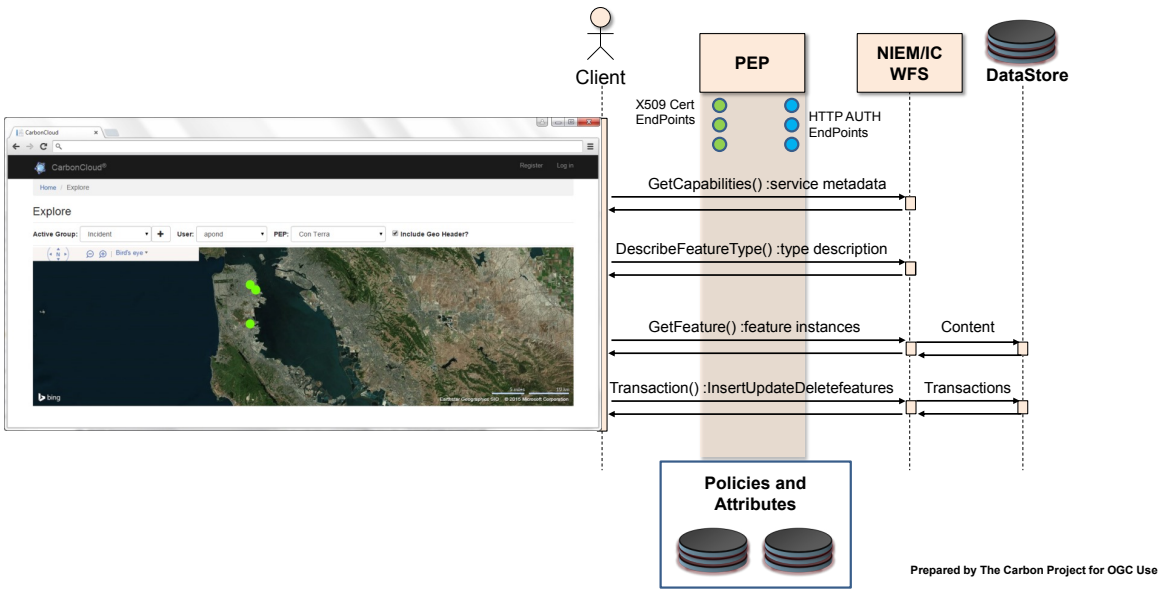



**Figure 18 - Web Client from The Carbon Project accessing con terra PEP with GeoHeader.**

The communication between the client and the service is intercepted by the con terra PEP. The PEP is responsible for authenticating the request, for gathering the required user information from the attribute service, for delegating the authorization decision to the Policy Decision Point (PDP), and for enforcing the authorization decision.

This approach was a deny-biased system, which means everything that shall be allowed must be expressed in a policy, otherwise the PEP will not grant access. This means, there must be a policy permitting the subject to perform the requested action on the requested resource. Policy decisions were of three different kinds:

- Not applicable, which means no policy is available that allows access, so access will be blocked.
- Permit, which means the request is forwarded to the protected service.
- Permit with obligations, which means the PEP's obligation handlers will enforce the obligations, and only if this is successful, the request will be passed to the protected service. Obligations might also need to be fulfilled on the service's response.

## 5.4 Jericho Systems

Jericho Systems implemented the NIEM/IC Data Encoding in PEP services. The PEP acts as an HTTP Reverse Proxy between the client and WFS with a SAML for XACML interface to the EnterSpace® component as shown in the figure below.

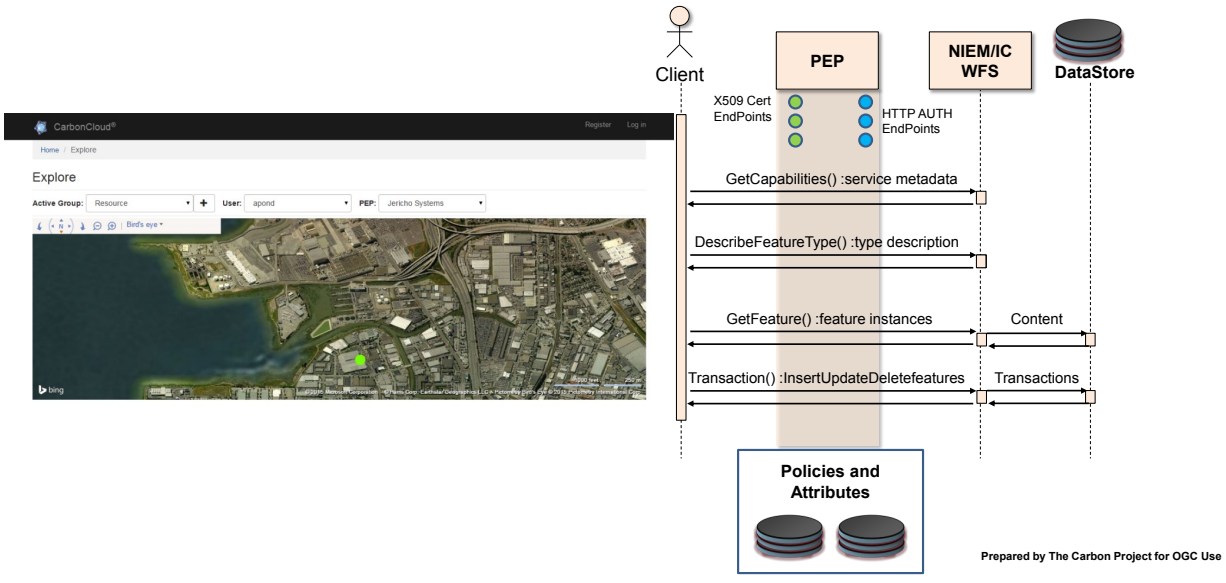



**Figure 19 – Jericho Systems PEP in The Carbon Project web client, accessing Resource encoding**

The PEP intercepts the client request and sends a SAML for XACML Authorization Decision Request to the EnterSpace component. The EnterSpace component makes a policy decision based on an attribute based access control (ABAC) policy previously created.

Attributes can be passed with the request context, such as a geolocation for the device making the request. If necessary, attributes about the request context required by the access control policy are retrieved to make an access control decision. Attributes about the entity making the request are retrieved from the OGC Testbed Attribute Service and Identity Provider (IdP). Additional attributes can also be collected, such as environmental attributes concerning the network or system date and time.

The EnterSpace component returns a SAML for XACML Authorization Response to the HTTP Reverse Proxy PEP. If a "permit," the request is forwarded to the WFS Service by the PEP. If not a "permit" the request is denied and a response can be provided to the browser from the PEP or the request can be dropped with no response.

# 6 Findings and Recommendations

The evidence obtained through the Testbed 11:Geo4NIEM thread supports three main 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 describe these findings and any associated recommendations.

### 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) took 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 stand-point within a particular community of interest.*

# Annex A

# Sample NIEM/IC Feature Processing API Capabilities Response

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:WFS_Capabilities xmlns:wfs="http://www.opengis.net/wfs"
xmlns:cp="http://www.thecarbonproject.com"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.opengis.net/wfs/1.0.0/wfs.xsd"
version="1.0.0">
   <wfs:Service>
      <wfs:Name>CarbonCloud WFS: Niem</wfs:Name>
      <wfs:Title>CarbonCloud WFS: Niem</wfs:Title>
      <wfs:Abstract>CarbonCloud Web Feature Service:
Niem</wfs:Abstract>

   <wfs:OnlineResource>http://niems.azurewebsites.net/wfs</wfs:On
lineResource>
   </wfs:Service>
   <wfs:Capability>
      <wfs:Request>
         <wfs:GetCapabilities>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Get
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Post
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
         </wfs:GetCapabilities>
         <wfs:DescribeFeatureType>
            <wfs:SchemaDescriptionLanguage>
               <wfs:XMLSCHEMA></wfs:XMLSCHEMA>
            </wfs:SchemaDescriptionLanguage>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Get
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
            <wfs:DCPType>
```

```
               <wfs:HTTP>
                  <wfs:Post
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
         </wfs:DescribeFeatureType>
         <wfs:GetFeature>
            <wfs:ResultFormat>
               <wfs:GML2></wfs:GML2>
            </wfs:ResultFormat>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Get
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Post
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
         </wfs:GetFeature>
         <wfs:Transaction>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Get
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
            <wfs:DCPType>
               <wfs:HTTP>
                  <wfs:Post
onlineResource="http://niems.azurewebsites.net/wfs" />
               </wfs:HTTP>
            </wfs:DCPType>
         </wfs:Transaction>
      </wfs:Request>
   </wfs:Capability>
   <wfs:FeatureTypeList xmlns:wfs="http://www.opengis.net/wfs">
      <wfs:Operations>
         <wfs:Operation>Query</wfs:Operation>
      </wfs:Operations>
      <wfs:FeatureType>
         <wfs:Name>cp:incident</wfs:Name>
         <wfs:Operations>
            <wfs:Operation>Insert</wfs:Operation>
            <wfs:Operation>Update</wfs:Operation>
            <wfs:Operation>Delete</wfs:Operation>
         </wfs:Operations>
         <wfs:SRS>EPSG:4326</wfs:SRS>
```

```
         <LatLongBoundingBox minx="-180" miny="-90" maxx="180"
maxy="90" />
      </wfs:FeatureType>
      <wfs:FeatureType>
         <wfs:Name>cp:noticeofarrival</wfs:Name>
         <wfs:Operations>
            <wfs:Operation>Insert</wfs:Operation>
            <wfs:Operation>Update</wfs:Operation>
            <wfs:Operation>Delete</wfs:Operation>
         </wfs:Operations>
         <wfs:SRS>EPSG:4326</wfs:SRS>
         <LatLongBoundingBox minx="-180" miny="-90" maxx="180"
maxy="90" />
      </wfs:FeatureType>
   </wfs:FeatureTypeList>
   <ogc:Filter_Capabilities
xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:Spatial_Capabilities>
         <ogc:Spatial_Operators>
            <ogc:BBOX></ogc:BBOX>
            <ogc:Equals></ogc:Equals>
            <ogc:Disjoint></ogc:Disjoint>
            <ogc:Intersects></ogc:Intersects>
            <ogc:Touches></ogc:Touches>
            <ogc:Crosses></ogc:Crosses>
            <ogc:Within></ogc:Within>
            <ogc:Contains></ogc:Contains>
            <ogc:Overlaps></ogc:Overlaps>
            <ogc:Beyond></ogc:Beyond>
         </ogc:Spatial_Operators>
      </ogc:Spatial_Capabilities>
      <ogc:Scalar_Capabilities>
         <ogc:Logical_Operators>
            <ogc:AND></ogc:AND>
            <ogc:OR></ogc:OR>
         </ogc:Logical_Operators>
         <ogc:Comparison_Operators>
            <ogc:LessThan></ogc:LessThan>
            <ogc:GreaterThan></ogc:GreaterThan>
            <ogc:LessThanEqualTo></ogc:LessThanEqualTo>
            <ogc:GreaterThanEqualTo></ogc:GreaterThanEqualTo>
            <ogc:EqualTo></ogc:EqualTo>
            <ogc:NotEqualTo></ogc:NotEqualTo>
            <ogc:Like></ogc:Like>
            <ogc:Between></ogc:Between>
         </ogc:Comparison_Operators>
      </ogc:Scalar_Capabilities>
   </ogc:Filter_Capabilities>
</wfs:WFS_Capabilities>
```

# Annex B

# NIEM/IC wfs:FeatureCollection Sample

This annex provides a sample NIEM/IC wfs:FeatureCollection.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:mda="http://release.niem.gov/niem/domains/maritime/3.0/mda/
" xmlns:m="http://release.niem.gov/niem/domains/maritime/3.0/"
xmlns:nc="http://release.niem.gov/niem/niem-core/3.0/">
  <gml:featureMember>
    <mda:noticeofarrival
mda="http://release.niem.gov/niem/domains/maritime/3.0/mda/"
ntk="urn:us:gov:ic:ntk" ism="urn:us:gov:ic:ism"
nc="http://release.niem.gov/niem/niem-core/3.0/" mda-
codes="http://release.niem.gov/niem/domains/maritime/3.0/mda/code
s/" m="http://release.niem.gov/niem/domains/maritime/3.0/"
geo="http://release.niem.gov/niem/adapters/geospatial/3.0/"
DESVersion="11" ownerProducer="USA" classification="C"
resourceElement="true" classifiedBy="USCG"
classificationReason="Classified due to sensitive maritime
security information." declassDate="2050-12-01"
id="noticeofarrival.1" p7="http://www.opengis.net/gml">
      <mda:Voyage ownerProducer="USA" classification="U">
        <m:VoyageCategoryText>Foreign to
US</m:VoyageCategoryText>
        <m:VoyageIdentification>
          <nc:IdentificationID>1</nc:IdentificationID>
        </m:VoyageIdentification>

<mda:VoyageClosedLoopIndicator>false</mda:VoyageClosedLoopIndicat
or>
      </mda:Voyage>
      <mda:Vessel ownerProducer="USA" classification="U">
        <m:VesselAugmentation ownerProducer="USA"
classification="U">
          <m:VesselCallSignText>H3LP</m:VesselCallSignText>
          <m:VesselCargoCategoryText>Harmful
Substances</m:VesselCargoCategoryText>
          <m:VesselCategoryText>Container
Ship</m:VesselCategoryText>

<mda:VesselCDCCargoOnBoardIndicator>true</mda:VesselCDCCargoOnBoa
rdIndicator>
```

```
          <mda:VesselCharterer ownerProducer="USA"
classification="C" access="#Roles|Group^MDA-USCG-Msn-District11-
ROC">
            <nc:EntityOrganization>
              <nc:OrganizationLocation>
                <nc:Address>

<nc:LocationCountryISO3166Alpha2Code>KR</nc:LocationCountryISO316
6Alpha2Code>
                </nc:Address>
              </nc:OrganizationLocation>
              <nc:OrganizationName>SK
Shipping</nc:OrganizationName>
            </nc:EntityOrganization>
          </mda:VesselCharterer>
          <m:VesselClassText>Bulk Carrier</m:VesselClassText>
          <m:VesselClassificationSocietyName>Germanischer
Lloyd</m:VesselClassificationSocietyName>
          <m:VesselContactInformation>
            <nc:ContactTelephoneNumber>
              <nc:InternationalTelephoneNumber>
                <nc:TelephoneNumberID>800-555-
1212</nc:TelephoneNumberID>
              </nc:InternationalTelephoneNumber>

<nc:TelephoneNumberCategoryCode>work</nc:TelephoneNumberCategoryC
ode>
            </nc:ContactTelephoneNumber>
            <nc:ContactEntity>
              <nc:EntityPerson>
                <nc:PersonName>
                  <nc:PersonFullName>James
Smith</nc:PersonFullName>
                </nc:PersonName>
              </nc:EntityPerson>
            </nc:ContactEntity>
          </m:VesselContactInformation>
          <m:VesselDOCCertificate>
            <nc:DocumentExpirationDate>
              <nc:Date>2028-04-24T00:00:00</nc:Date>
            </nc:DocumentExpirationDate>
            <nc:CertificateIssueDate>
              <nc:Date>2028-04-25T00:00:00</nc:Date>
            </nc:CertificateIssueDate>
            <m:CertificateIssuingAgency>
              <nc:EntityOrganization>
                <nc:OrganizationName>U.S. Coast
Guard</nc:OrganizationName>
              </nc:EntityOrganization>
            </m:CertificateIssuingAgency>
```

```
          </m:VesselDOCCertificate>
          <m:VesselISSC ownerProducer="USA" classification="C"
access="#Roles|Group^MDA-USCG-Msn-District11-ROC">
            <m:CertificateIssueDate>
              <nc:Date>2022-06-22T00:00:00</nc:Date>
            </m:CertificateIssueDate>
            <m:CertificateIssuingAgency>
              <nc:EntityOrganization>
                <nc:OrganizationName>Government of Bermuda,
Department of Maritime Administration</nc:OrganizationName>
              </nc:EntityOrganization>
            </m:CertificateIssuingAgency>
            <m:RecognizedISSCSecurityEntity>
              <nc:EntityOrganization>
                <nc:OrganizationName>Government of Bermuda,
Department of Maritime Administration</nc:OrganizationName>
              </nc:EntityOrganization>
            </m:RecognizedISSCSecurityEntity>
            <m:VesselSecurityOfficerContactInformation>
              <m:ContactTelephoneNumber>
                <nc:InternationalTelephoneNumber>
                  <nc:TelephoneNumberID>888-234-
5432</nc:TelephoneNumberID>
                </nc:InternationalTelephoneNumber>

<nc:TelephoneNumberCategoryCode>work</nc:TelephoneNumberCategoryC
ode>
              </m:ContactTelephoneNumber>

<nc:ContactEmailID>ftest@test.com</nc:ContactEmailID>
              <nc:ContactEntity>
                <nc:EntityPerson>
                  <nc:PersonName>
                    <nc:PersonFullName>Frank
Test</nc:PersonFullName>
                  </nc:PersonName>
                </nc:EntityPerson>
              </nc:ContactEntity>
            </m:VesselSecurityOfficerContactInformation>

<m:VesselSecurityPlanImplementedIndicator>true</m:VesselSecurityP
lanImplementedIndicator>
          </m:VesselISSC>
          <m:VesselMMSIText>352948000</m:VesselMMSIText>
          <m:VesselName>MSC NERISSA</m:VesselName>

<m:VesselNationalFlagISO3166Alpha2Code>PA</m:VesselNationalFlagIS
O3166Alpha2Code>

<m:VesselOfficialCoastGuardNumberText>US878N2</m:VesselOfficialCo
astGuardNumberText>
```

```
          <m:VesselOperator ownerProducer="USA"
classification="C" access="#Roles|Group^MDA-USCG-Msn-District11-
ROC">
            <nc:EntityPerson>
              <nc:PersonName>
                <nc:PersonFullName>Dan James</nc:PersonFullName>
              </nc:PersonName>
            </nc:EntityPerson>
          </m:VesselOperator>
          <m:VesselOwner>
            <nc:EntityOrganization>
              <nc:OrganizationName>MSC Mediterranean Shipping
Company</nc:OrganizationName>
            </nc:EntityOrganization>
          </m:VesselOwner>
          <m:VesselSafetyManagementCertificate
ownerProducer="USA" classification="C" access="#Roles|Group^MDA-
USCG-Msn-District11-ROC">
            <nc:DocumentExpirationDate>
              <nc:Date>2027-12-01T00:00:00</nc:Date>
            </nc:DocumentExpirationDate>
            <nc:CertificateIssueDate>
              <nc:Date>2017-03-12T00:00:00</nc:Date>
            </nc:CertificateIssueDate>
            <m:CertificateIssuingAgency>
              <nc:EntityOrganization>
                <nc:OrganizationName>U.S. Coast
Guard</nc:OrganizationName>
              </nc:EntityOrganization>
            </m:CertificateIssuingAgency>
          </m:VesselSafetyManagementCertificate>
        </m:VesselAugmentation>

<mda:VesselCargoOnBoardIndicator>true</mda:VesselCargoOnBoardIndi
cator>
        <mda:VesselCertificateOfFinancialResponsibilityOperator
ownerProducer="USA" classification="U" access="#Roles|Group^MDA-
USCG-Msn-District11-ROC">

<mda:VesselCertificateOfFinancialResponsibilityOperator>
            <nc:EntityOrganization>
              <nc:OrganizationName>MSC Mediterranean Shipping
Company</nc:OrganizationName>
            </nc:EntityOrganization>

</mda:VesselCertificateOfFinancialResponsibilityOperator>
        </mda:VesselCertificateOfFinancialResponsibilityOperator>
        <mda:VesselSubCategoryText>Anhydrous
Ammonia</mda:VesselSubCategoryText>
      </mda:Vessel>
```

```
      <mda:Arrival ownerProducer="USA" classification="U">
        <mda:VisitAnchorageText>Main
Anchorage</mda:VisitAnchorageText>
        <mda:VisitExpectedArrivalDateTime>
          <nc:DateTime>2025-12-10T14:30:00</nc:DateTime>
        </mda:VisitExpectedArrivalDateTime>
        <mda:VisitLocationInPort>
          <m:PortName>Oakland</m:PortName>
          <nc:LocationStateName>CA</nc:LocationStateName>
          <nc:LocationCityName>Oakland</nc:LocationCityName>
          <mda:PortAugmentation>
            <m:LocationPoint>
              <gml:Point gml="http://www.opengis.net/gml"
srsName="EPSG::4326">
                <gml:pos srsName="EPSG::4326" srsDimension="2">-
122.295 37.6965</gml:pos>
              </gml:Point>
            </m:LocationPoint>
          </mda:PortAugmentation>
        </mda:VisitLocationInPort>
        <mda:VisitReceivingFacilityName>Pier
57</mda:VisitReceivingFacilityName>
      </mda:Arrival>
      <mda:Departure ownerProducer="USA" classification="U">
        <mda:VisitExpectedDepartureDateTime>
          <mda:DateTime>2025-12-16T00:00:00</mda:DateTime>
        </mda:VisitExpectedDepartureDateTime>
      </mda:Departure>
      <mda:LastPortOfCall ownerProducer="USA" classification="U"
access="#Roles|Group^MDA-USCG-Msn-District11-ROC">
        <mda:VisitActualArrivalDateTime>
          <nc:DateTime>2025-11-25T00:00:00</nc:DateTime>
        </mda:VisitActualArrivalDateTime>
        <mda:VisitActualDepartureDateTime>
          <mda:DateTime>2025-11-30T00:00:00</mda:DateTime>
        </mda:VisitActualDepartureDateTime>
        <mda:VisitLocationInPort>
          <m:PortName>Port of Portland, Oregon</m:PortName>

<nc:LocationCountryISO3166Alpha2Code>US</nc:LocationCountryISO316
6Alpha2Code>
          <nc:LocationStateName>OR</nc:LocationStateName>
          <nc:LocationCityName>Portland</nc:LocationCityName>
        </mda:VisitLocationInPort>
      </mda:LastPortOfCall>
      <mda:NextPortOfCallList ownerProducer="USA"
classification="U" access="#Roles|Group^MDA-USCG-Msn-District11-
ROC">
        <mda:NextPortOfCall>
          <mda:VisitExpectedArrivalDateTime>
            <nc:DateTime>2026-01-02T00:00:00</nc:DateTime>
```

```
          </mda:VisitExpectedArrivalDateTime>
          <mda:VisitExpectedDepartureDateTime>
            <nc:DateTime>2026-01-07T00:00:00</nc:DateTime>
          </mda:VisitExpectedDepartureDateTime>
          <mda:VisitLocationInPort>
            <m:PortName>Port of Long Beach</m:PortName>

<nc:LocationCountryISO3166Alpha2Code>US</nc:LocationCountryISO316
6Alpha2Code>
            <nc:LocationStateName>CA</nc:LocationStateName>
            <nc:LocationCityName>Long Beach</nc:LocationCityName>
          </mda:VisitLocationInPort>
        </mda:NextPortOfCall>
      </mda:NextPortOfCallList>
      <mda:CDCCargoList ownerProducer="USA" classification="C"
access="#Roles|Group^MDA-USCG-Msn-District11-ROC
Roles|Group^NIMS-FEMA-Msn-RegionIX-IC">
        <mda:CDCCargo>
          <m:CargoDestinationLocation>
            <nc:Address>
              <nc:LocationStateName>CA</nc:LocationStateName>

<nc:LocationCountryISO3166Alpha2Code>US</nc:LocationCountryISO316
6Alpha2Code>
            </nc:Address>
            <m:LocationAugmentation>
              <m:LocationPort>
                <m:PortCodeText>USOAK</m:PortCodeText>
                <m:PortName>Port of Oakland</m:PortName>
              </m:LocationPort>
            </m:LocationAugmentation>
          </m:CargoDestinationLocation>
          <m:CargoHazmatDeclaration>

<m:HazmatDeclarationChemicalCommonName>Pesticide</m:HazmatDeclara
tionChemicalCommonName>
            <m:HazmatDeclarationDescriptionText>Division 2.3
Poisonous Gas</m:HazmatDeclarationDescriptionText>
            <m:HazmatDeclarationMaterialAmountMeasure>
              <nc:MeasureValueText>100</nc:MeasureValueText>
              <nc:MeasureUnitText>Barrel</nc:MeasureUnitText>
            </m:HazmatDeclarationMaterialAmountMeasure>

<m:HazmatDeclarationUNHazmatCode>UN3018</m:HazmatDeclarationUNHaz
matCode>
          </m:CargoHazmatDeclaration>

<m:CargoPackagedIndicator>true</m:CargoPackagedIndicator>

<m:CargoResidueIndicator>false</m:CargoResidueIndicator>
```

```
        </mda:CDCCargo>
      </mda:CDCCargoList>
    </mda:noticeofarrival>
  </gml:featureMember>
</wfs:FeatureCollection>
```

# Annex C

# NIEM/IC Schema Description Sample

This annex provides a sample of the schema description for a NIEM/IC wfs:FeatureCollection.

```
<?xml version="1.0" encoding="utf-16"
standalone="yes"?><xs:schema version="1.1"
elementFormDefault="qualified"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:cp="http://www.thecarbonproject.com"
xmlns:mda="http://release.niem.gov/niem/domains/maritime/3.0/mda/
" xmlns:ntk="urn:us:gov:ic:ntk" xmlns:ism="urn:us:gov:ic:ism"
xmlns:nc="http://release.niem.gov/niem/niem-core/3.0/" xmlns:mda-
codes="http://release.niem.gov/niem/domains/maritime/3.0/mda/code
s/" xmlns:m="http://release.niem.gov/niem/domains/maritime/3.0/"
xmlns:geo="http://release.niem.gov/niem/adapters/geospatial/3.0/"
><xs:import
schemaLocation="http://release.niem.gov/niem/domains/maritime/3.0
/maritime.xsd"
namespace="http://release.niem.gov/niem/domains/maritime/3.0/"
/><xs:import schemaLocation="http://release.niem.gov/niem/niem-
core/3.0/niem-core.xsd"
namespace="http://release.niem.gov/niem/niem-core/3.0/"
/><xs:import
schemaLocation="http://release.niem.gov/niem/domains/maritime/3.0
/mda/mda.xsd"
namespace="http://release.niem.gov/niem/domains/maritime/3.0/mda/
" /><xs:import
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd
" namespace="http://www.opengis.net/gml/" /><xs:element
substitutionGroup="gml:_Feature" type="cp:noticeofarrival_Type"
name="noticeofarrival" /><xs:complexType
name="noticeofarrival_Type"><xs:sequence><xs:element
name="Voyage" type="m:VoyageType" /><xs:element name="Vessel"
type="nc:VesselType" /><xs:element name="Arrival"
type="mda:PortVisitType" /><xs:element name="Departure"
type="mda:PortVisitType" /><xs:element name="LastPortOfCall"
type="mda:PortVisitType" /><xs:element name="NextPortOfCallList"
type="mda:NextPortOfCallListType" /><xs:element
name="CDCCargoList" type="mda:CDCCargoListType"
/></xs:sequence></xs:complexType></xs:schema>
```

# Annex D

# OutputFormat for Security Info Sample

```
<?xml version="1.0" encoding="UTF-8"?>
-<wfs:FeatureCollection
xsi:schemaLocation="http://schemas.opengis.net/wfs/2.0/wfs.xsd
http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wfs="http://www.opengis.net/wfs">
-<gml:member>
-<tdf:TrustedDataObject xmlns:tdf="urn:us:gov:ic:tdf">
-<tdf:HandlingAssertion tdf:scope="TDO">
-<tdf:HandlingStatement>
-<edh:Edh ntk:DESVersion="9" ism:DESVersion="10"
arh:DESVersion="2" icid:DESVersion="1" edh:DESVersion="3"
xmlns:ntk="urn:us:gov:ic:ntk" xmlns:ism="urn:us:gov:ic:ism"
xmlns:arh="urn:us:gov:ic:arh" xmlns:icid="urn:us:gov:ic:id"
xmlns:edh="urn:us:gov:ic:edh">
<icid:Identifier>guide://999123/NOA001EDH01</icid:Identifier>
<edh:DataItemCreateDateTime>2025-12-
10T00:00:35Z</edh:DataItemCreateDateTime>
-<edh:ResponsibleEntity>
<edh:Country>USA</edh:Country>
<edh:Organization>USG</edh:Organization>
</edh:ResponsibleEntity>
-<arh:Security ism:ownerProducer="USA" ism:classification="U">
-<ntk:Access ism:ownerProducer="USA" ism:classification="U">
-<ntk:RequiresAnyOf ism:ownerProducer="USA"
ism:classification="U">
-<ntk:AccessGroupList>
-<ntk:AccessGroup>
<ntk:AccessPolicy ism:ownerProducer="USA"
ism:classification="U">Roles</ntk:AccessPolicy>
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">NIMS-FEMA-Msn-RegionIX-
ICS</ntk:AccessGroupValue>
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">MDA-USCG-Msn-District11-
ROC</ntk:AccessGroupValue>
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">SEMS-CA-Ent-CoastalRegion-
MAC</ntk:AccessGroupValue>
```

```
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">SEMS-CA-Ent-StateOperationsCenter-
MAC</ntk:AccessGroupValue>
</ntk:AccessGroup>
</ntk:AccessGroupList>
-<ntk:AccessProfileList ism:ownerProducer="USA"
ism:classification="U">
-<ntk:AccessProfile ism:ownerProducer="USA"
ism:classification="U">
<ntk:AccessPolicy ism:ownerProducer="USA"
ism:classification="U">slt-ntk.aces</ntk:AccessPolicy>

<ntk:AccessProfileValue ism:ownerProducer="USA"
ism:classification="U"
ntk:vocabulary="urn:us:gov:ic:cvenum:usagency:agencyacronym">SLT<
/ntk:AccessProfileValue>
</ntk:AccessProfile>
</ntk:AccessProfileList>
</ntk:RequiresAnyOf>
</ntk:Access>
</arh:Security>
</edh:Edh>
</tdf:HandlingStatement>
</tdf:HandlingAssertion>
-<tdf:HandlingAssertion tdf:scope="PAYL">
-<tdf:HandlingStatement>
-<edh:Edh ntk:DESVersion="9" ism:DESVersion="10"
arh:DESVersion="2" icid:DESVersion="1" edh:DESVersion="3"
xmlns:ntk="urn:us:gov:ic:ntk" xmlns:ism="urn:us:gov:ic:ism"
xmlns:arh="urn:us:gov:ic:arh" xmlns:icid="urn:us:gov:ic:id"
xmlns:edh="urn:us:gov:ic:edh">
<icid:Identifier>guide://999123/NOA001EDH01</icid:Identifier>
<edh:DataItemCreateDateTime>2025-12-
10T00:00:35Z</edh:DataItemCreateDateTime>
-<edh:ResponsibleEntity>
<edh:Country>USA</edh:Country>
<edh:Organization>USG</edh:Organization>
</edh:ResponsibleEntity>
-<arh:Security ism:ownerProducer="USA" ism:classification="U">
-<ntk:Access ism:ownerProducer="USA" ism:classification="U">
-<ntk:RequiresAnyOf ism:ownerProducer="USA"
ism:classification="U">
-<ntk:AccessGroupList>
-<ntk:AccessGroup>
<ntk:AccessPolicy ism:ownerProducer="USA"
ism:classification="U">Roles</ntk:AccessPolicy>
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">NIMS-FEMA-Msn-RegionIX-
ICS</ntk:AccessGroupValue>
```

```
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">MDA-USCG-Msn-District11-
ROC</ntk:AccessGroupValue>
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">SEMS-CA-Ent-CoastalRegion-
MAC</ntk:AccessGroupValue>
<ntk:AccessGroupValue ism:ownerProducer="USA"
ism:classification="U">SEMS-CA-Ent-StateOperationsCenter-
MAC</ntk:AccessGroupValue>
</ntk:AccessGroup>
</ntk:AccessGroupList>
-<ntk:AccessProfileList ism:ownerProducer="USA"
ism:classification="U">
-<ntk:AccessProfile ism:ownerProducer="USA"
ism:classification="U">
<ntk:AccessPolicy ism:ownerProducer="USA"
ism:classification="U">slt-ntk.aces</ntk:AccessPolicy>

<ntk:AccessProfileValue ism:ownerProducer="USA"
ism:classification="U"
ntk:vocabulary="urn:us:gov:ic:cvenum:usagency:agencyacronym">SLT<
/ntk:AccessProfileValue>
</ntk:AccessProfile>
</ntk:AccessProfileList>
</ntk:RequiresAnyOf>
</ntk:Access>
</arh:Security>
</edh:Edh>
</tdf:HandlingStatement>
</tdf:HandlingAssertion>
-<tdf:StructuredPayload>
-<cp:noticeofarrival ism:DESVersion="11" ntk:DESVersion="9"
xmlns:ntk="urn:us:gov:ic:ntk" xmlns:ism="urn:us:gov:ic:ism"
ism:ownerProducer="USA" ism:classification="C"
fid="noticeofarrival.1" ism:declassDate="2050-12-01"
ism:classificationReason="Classified due to sensitive maritime
security information." ism:classifiedBy="USCG"
ism:resourceElement="true"
xmlns:geo="http://release.niem.gov/niem/adapters/geospatial/3.0/"
xmlns:m="http://release.niem.gov/niem/domains/maritime/3.0/"
xmlns:mda-
codes="http://release.niem.gov/niem/domains/maritime/3.0/mda/code
s/" xmlns:nc="http://release.niem.gov/niem/niem-core/3.0/"
xmlns:mda="http://release.niem.gov/niem/domains/maritime/3.0/mda/
" xmlns:cp="http://www.thecarbonproject.com">
-<mda:Voyage ism:ownerProducer="USA" ism:classification="U">
<m:VoyageCategoryText>Foreign to US</m:VoyageCategoryText>
-<m:VoyageIdentification>
<nc:IdentificationID>1</nc:IdentificationID>
</m:VoyageIdentification>
```

```
</mda:Voyage>
-<mda:Vessel ism:ownerProducer="USA" ism:classification="U">
-<m:VesselAugmentation ism:ownerProducer="USA"
ism:classification="U">
<m:VesselCallSignText>H3LP</m:VesselCallSignText>
<m:VesselCargoCategoryText>Harmful
Substances</m:VesselCargoCategoryText>
<m:VesselCategoryText>Container Ship</m:VesselCategoryText>
<mda:VesselCDCCargoOnBoardIndicator>true</mda:VesselCDCCargoOnBoa
rdIndicator>
-<mda:VesselCharterer ism:ownerProducer="USA"
ism:classification="C" ntk:access="#Roles|Group^MDA-USCG-Msn-
District11-ROC">
-<nc:EntityOrganization>
-<nc:OrganizationLocation>
-<nc:Address>
<nc:LocationCountryISO3166Alpha2Code>KR</nc:LocationCountryISO316
6Alpha2Code>
</nc:Address>
</nc:OrganizationLocation>
<nc:OrganizationName>SK Shipping</nc:OrganizationName>
</nc:EntityOrganization>
</mda:VesselCharterer>
<m:VesselClassText>Bulk Carrier</m:VesselClassText>
<m:VesselClassificationSocietyName>Germanischer
Lloyd</m:VesselClassificationSocietyName>
-<m:VesselContactInformation>
-<nc:ContactTelephoneNumber>
-<nc:InternationalTelephoneNumber>
<nc:TelephoneNumberID>800-555-1212</nc:TelephoneNumberID>
</nc:InternationalTelephoneNumber>
<nc:TelephoneNumberCategoryCode>work</nc:TelephoneNumberCategoryC
ode>
</nc:ContactTelephoneNumber>
-<nc:ContactEntity>
-<nc:EntityPerson>
-<nc:PersonName>
<nc:PersonFullName>James Smith</nc:PersonFullName>
</nc:PersonName>
</nc:EntityPerson>
</nc:ContactEntity>
</m:VesselContactInformation>
-<m:VesselDOCCertificate>
-<nc:DocumentExpirationDate>
<nc:Date>2028-04-24T00:00:00</nc:Date>
</nc:DocumentExpirationDate>
-<m:CertificateIssueDate>
<nc:Date>2028-04-25T00:00:00</nc:Date>
</m:CertificateIssueDate>
-<m:CertificateIssuingAgency>
-<nc:EntityOrganization>
```

```
<nc:OrganizationName>U.S. Coast Guard</nc:OrganizationName>
</nc:EntityOrganization>
</m:CertificateIssuingAgency>
</m:VesselDOCCertificate>
-<m:VesselISSC ism:ownerProducer="USA" ism:classification="C"
ntk:access="#Roles|Group^MDA-USCG-Msn-District11-ROC">
-<m:CertificateIssueDate>
<nc:Date>2022-06-22T00:00:00</nc:Date>
</m:CertificateIssueDate>
-<m:CertificateIssuingAgency>
-<nc:EntityOrganization>
<nc:OrganizationName>Government of Bermuda, Department of
Maritime Administration</nc:OrganizationName>
</nc:EntityOrganization>
</m:CertificateIssuingAgency>
-<m:RecognizedISSCSecurityEntity>
-<nc:EntityOrganization>
<nc:OrganizationName>Government of Bermuda, Department of
Maritime Administration</nc:OrganizationName>
</nc:EntityOrganization>
</m:RecognizedISSCSecurityEntity>
-<m:VesselSecurityOfficerContactInformation>
-<nc:ContactTelephoneNumber>
-<nc:InternationalTelephoneNumber>
<nc:TelephoneNumberID>888-234-5432</nc:TelephoneNumberID>
</nc:InternationalTelephoneNumber>
<nc:TelephoneNumberCategoryCode>work</nc:TelephoneNumberCategoryC
ode>
</nc:ContactTelephoneNumber>
<nc:ContactEmailID>ftest@test.com</nc:ContactEmailID>
-<nc:ContactEntity>
-<nc:EntityPerson>
-<nc:PersonName>
<nc:PersonFullName>Frank Test</nc:PersonFullName>
</nc:PersonName>
</nc:EntityPerson>
</nc:ContactEntity>
</m:VesselSecurityOfficerContactInformation>
<m:VesselSecurityPlanImplementedIndicator>true</m:VesselSecurityP
lanImplementedIndicator>
</m:VesselISSC>
<m:VesselMMSIText>352948000</m:VesselMMSIText>
<m:VesselName>MSC NERISSA</m:VesselName>
<m:VesselNationalFlagISO3166Alpha2Code>PA</m:VesselNationalFlagIS
O3166Alpha2Code>
<m:VesselOfficialCoastGuardNumberText>US878N2</m:VesselOfficialCo
astGuardNumberText>
-<m:VesselOperator ism:ownerProducer="USA" ism:classification="C"
ntk:access="#Roles|Group^MDA-USCG-Msn-District11-ROC">
-<nc:EntityPerson>
```

```
-<nc:PersonName>
<nc:PersonFullName>Dan James</nc:PersonFullName>
</nc:PersonName>
</nc:EntityPerson>
</m:VesselOperator>
-<m:VesselOwner>
-<nc:EntityOrganization>
<nc:OrganizationName>MSC Mediterranean Shipping
Company</nc:OrganizationName>
</nc:EntityOrganization>
</m:VesselOwner>
-<m:VesselSafetyManagementCertificate ism:ownerProducer="USA"
ism:classification="C" ntk:access="#Roles|Group^MDA-USCG-Msn-
District11-ROC">
-<nc:DocumentExpirationDate>
<nc:Date>2027-12-01T00:00:00</nc:Date>
</nc:DocumentExpirationDate>
-<m:CertificateIssueDate>
<nc:Date>2017-03-12T00:00:00</nc:Date>
</m:CertificateIssueDate>
-<m:CertificateIssuingAgency>
-<nc:EntityOrganization>
<nc:OrganizationName>U.S. Coast Guard</nc:OrganizationName>
</nc:EntityOrganization>
</m:CertificateIssuingAgency>
</m:VesselSafetyManagementCertificate>
</m:VesselAugmentation>
<mda:VesselCargoOnBoardIndicator>true</mda:VesselCargoOnBoardIndi
cator>
-<mda:VesselCertificateOfFinancialResponsibilityOperator
ism:ownerProducer="USA" ism:classification="U"
ntk:access="#Roles|Group^MDA-USCG-Msn-District11-ROC">
-<mda:VesselCertificateOfFinancialResponsibilityOperator>
-<nc:EntityOrganization>
<nc:OrganizationName>MSC Mediterranean Shipping
Company</nc:OrganizationName>
</nc:EntityOrganization>
</mda:VesselCertificateOfFinancialResponsibilityOperator>
</mda:VesselCertificateOfFinancialResponsibilityOperator>
<mda:VesselSubCategoryText>Anhydrous
Ammonia</mda:VesselSubCategoryText>
</mda:Vessel>
-<mda:Arrival ism:ownerProducer="USA" ism:classification="U">
<mda:VisitAnchorageText>Main Anchorage</mda:VisitAnchorageText>
-<mda:VisitExpectedArrivalDateTime>
<nc:DateTime>2025-12-10T14:30:00</nc:DateTime>
</mda:VisitExpectedArrivalDateTime>
-<mda:VisitLocationInPort>
<m:PortName>Oakland</m:PortName>
<nc:LocationStateName>CA</nc:LocationStateName>
<nc:LocationCityName>Oakland</nc:LocationCityName>
```

```
-<mda:PortAugmentation>
-<m:LocationPoint>
-<gml:Point xmlns:gml="http://www.opengis.net/gml">
<gml:pos srsDimension="2">-122.295 37.6965</gml:pos>
</gml:Point>
</m:LocationPoint>
</mda:PortAugmentation>
</mda:VisitLocationInPort>
<mda:VisitReceivingFacilityName>Pier
57</mda:VisitReceivingFacilityName>
</mda:Arrival>
-<mda:Departure ism:ownerProducer="USA" ism:classification="U">
-<mda:VisitExpectedDepartureDateTime>
<mda:DateTime>2025-12-16T00:00:00</mda:DateTime>
</mda:VisitExpectedDepartureDateTime>
</mda:Departure>
-<mda:LastPortOfCall ism:ownerProducer="USA"
ism:classification="U" ntk:access="#Roles|Group^MDA-USCG-Msn-
District11-ROC">
-<mda:VisitActualArrivalDateTime>
<nc:DateTime>2025-11-25T00:00:00</nc:DateTime>
</mda:VisitActualArrivalDateTime>
-<mda:VisitActualDepartureDateTime>
<mda:DateTime>2025-11-30T00:00:00</mda:DateTime>
</mda:VisitActualDepartureDateTime>
-<mda:VisitLocationInPort>
<m:PortName>Port of Portland, Oregon</m:PortName>
<nc:LocationCountryISO3166Alpha2Code>US</nc:LocationCountryISO316
6Alpha2Code>
<nc:LocationStateName>OR</nc:LocationStateName>
<nc:LocationCityName>Portland</nc:LocationCityName>
</mda:VisitLocationInPort>
</mda:LastPortOfCall>
-<mda:NextPortOfCallList ism:ownerProducer="USA"
ism:classification="U" ntk:access="#Roles|Group^MDA-USCG-Msn-
District11-ROC">
-<mda:NextPortOfCall>
-<mda:VisitExpectedArrivalDateTime>
<nc:DateTime>2026-01-02T00:00:00</nc:DateTime>
</mda:VisitExpectedArrivalDateTime>
-<mda:VisitExpectedDepartureDateTime>
<nc:DateTime>2026-01-07T00:00:00</nc:DateTime>
</mda:VisitExpectedDepartureDateTime>
-<mda:VisitLocationInPort>
<m:PortName>Port of Long Beach</m:PortName>
<nc:LocationCountryISO3166Alpha2Code>US</nc:LocationCountryISO316
6Alpha2Code>
<nc:LocationStateName>CA</nc:LocationStateName>
<nc:LocationCityName>Long Beach</nc:LocationCityName>
</mda:VisitLocationInPort>
```

```
</mda:NextPortOfCall>
</mda:NextPortOfCallList>
-<mda:CDCCargoList ism:ownerProducer="USA" ism:classification="C"
ntk:access="#Roles|Group^MDA-USCG-Msn-District11-ROC
Roles|Group^NIMS-FEMA-Msn-RegionIX-IC">
-<mda:CDCCargo>
-<m:CargoDestinationLocation>
-<nc:Address>
<nc:LocationStateName>CA</nc:LocationStateName>
<nc:LocationCountryISO3166Alpha2Code>US</nc:LocationCountryISO316
6Alpha2Code>
</nc:Address>
-<m:LocationAugmentation>
-<m:LocationPort>
<m:PortCodeText>USOAK</m:PortCodeText>
<m:PortName>Port of Oakland</m:PortName>
</m:LocationPort>
</m:LocationAugmentation>
</m:CargoDestinationLocation>
-<m:CargoHazmatDeclaration>
<m:HazmatDeclarationChemicalCommonName>Pesticide</m:HazmatDeclara
tionChemicalCommonName>
<m:HazmatDeclarationDescriptionText>Division 2.3 Poisonous
Gas</m:HazmatDeclarationDescriptionText>
-<m:HazmatDeclarationMaterialAmountMeasure>
<nc:MeasureValueText>100</nc:MeasureValueText>
<nc:MeasureUnitText>Barrel</nc:MeasureUnitText>
</m:HazmatDeclarationMaterialAmountMeasure>
<m:HazmatDeclarationUNHazmatCode>UN3018</m:HazmatDeclarationUNHaz
matCode>
</m:CargoHazmatDeclaration>
<m:CargoPackagedIndicator>true</m:CargoPackagedIndicator>
<m:CargoResidueIndicator>false</m:CargoResidueIndicator>
</mda:CDCCargo>
</mda:CDCCargoList>
</cp:noticeofarrival>
</tdf:StructuredPayload>
</tdf:TrustedDataObject>
</gml:member>
</wfs:FeatureCollection>
```

# Revision history

| Date | Release | Editor | Primary clauses modified | Description |
|---|---|---|---|---|
| 2015-05-14 | 03 | J Harrison | All | Initial project deliverable. |
| 2015-05-26 | 05 | J Harrison | All | Abstract, Introduction, Graphics, and Findings updated based on TB11 engineering activities |
| 2015-07-19 | 062 | J Harrison | All | Incorporated edits from Testbed 11 Participants and Sponsors |
| 2015-10-12 | | Carl Reed | Various | Prepare for publication |