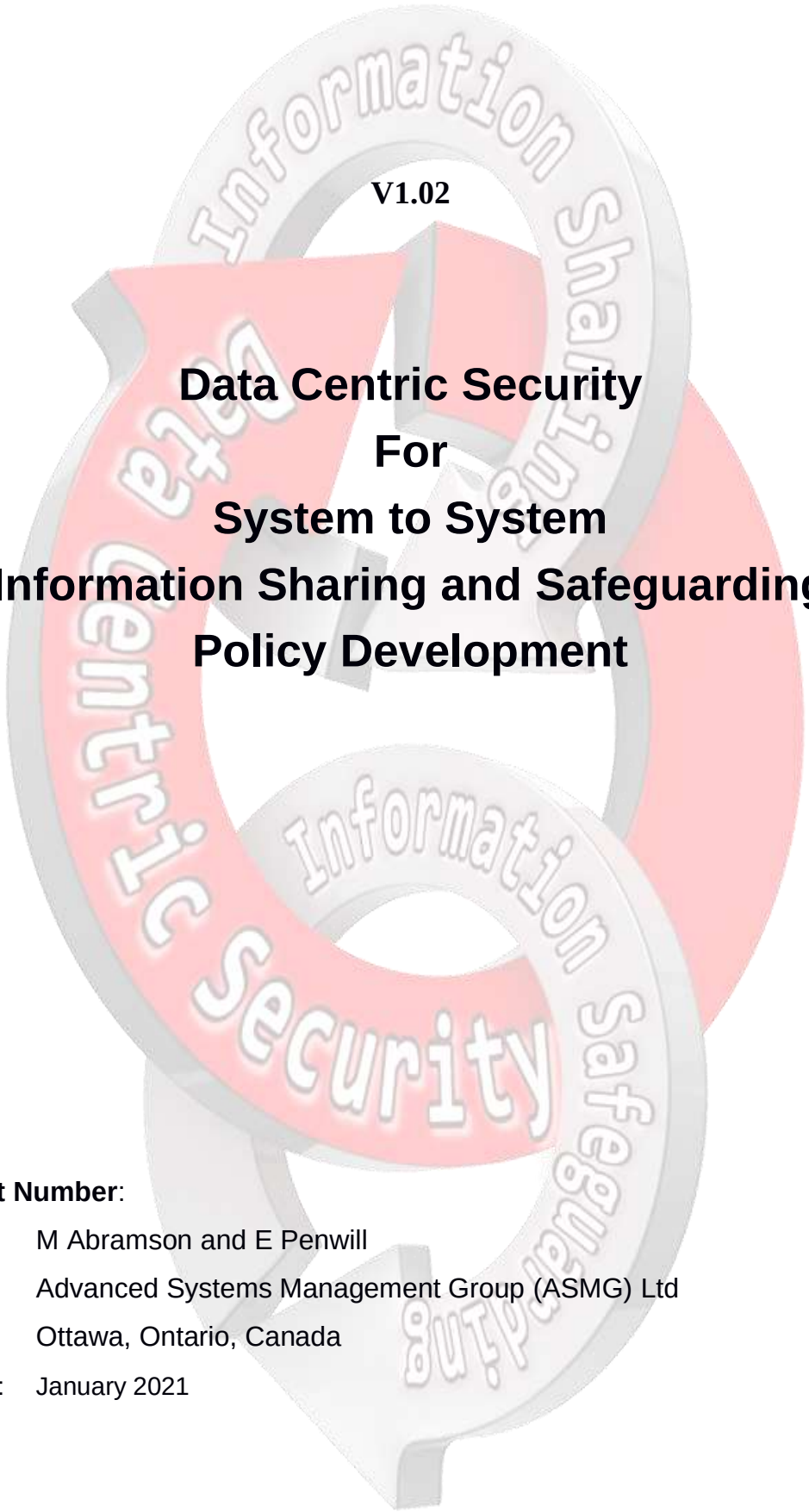

V1.02

Data Centric Security
For
System to System
Information Sharing and Safeguarding
Policy Development

Document Number:

Authors: M Abramson and E Penwill

Advanced Systems Management Group (ASMG) Ltd

Ottawa, Ontario, Canada

Issue Date: January 2021

Keywords

Interoperability, Standards, Information Sharing and Safeguarding, ISS, Data Centric Security, DCS, Secure Data Services, SDS, Information Exchange Framework, IEF™, IEF-RA™

Abstract

Concept document outlining the Information Sharing and Safeguarding (ISS) policy modelling paradigm and how it is employed by the Secure Data Service (SDS) to deliver system-to-system interoperability and Data Centric Security (DCS) capability.

Document Use

SUMMARY

Most organizations are seeking to develop a measure of information or decision advantage and are embarking on digitization efforts to harness the perceived advantages of such a process. Digitization efforts are focused on the harnessing of modern information technologies to enhance and expedite decision-making processes (e.g., **observe, orient, decide, act, and assess**), migrating more rapidly from sensor to effector in a mission or operational environment. However, information or decision advantage can only be delivered and sustained if the digital information and/or intelligence is not accessed, appropriated, obscured, corrupted, or manipulated by one's competitor or adversary. This means that digitization must risk manage the balance between making data and information available and the protection of that data. To do this, organizations need access to information describing the sensitivity of the data, information and intelligence holdings, potential risk to those holdings, and detailed understanding of how those holdings transition through networks, stores, system, applications, and processes, and where and with whom these holdings are shared. Unfortunately, there are few information and data management practices, standards, and/or tools that enable organizations to lifecycle manage the rules and constraints governing the use, sharing and safeguarding of data and information holdings at the content level. Most information and data management policy (rules and constraints) are encoded in software (e.g., SQL, JAVA, C++) APIs that are notoriously difficult to develop, manage and audit – leaving organizations blind to the fate of data and information holdings – and unable to manage Information and Data Management Risk.

Many modern information management capabilities (/information system) capture, process, analyse, present and share information and intelligence in ways that never involve humans. As organizations migrate to Software-as-a-Services (SaaS), or Commercial-off-the-Shelf (COTS) solutions, the little knowledge about their data and information, and how it is shared and safeguarded is being further eroded. Data breaches are becoming more, not less common. New more innovative solutions are needed for organizations to better use, share and safeguard

***Information Advantage**: The ability to maximize the use and quality of data and information throughout its lifecycle. And where necessary, deny that data and information to one's competitors or adversaries.*

***Decision Advantage**: The ability to provide decision makers with the highest quality (e.g., timely, accurate, relevant, complete, actionable, and trusted) information or intelligence to inform, enable and expedite decision making to achieve better outcomes. In addition to higher quality information, decision advantage requires at least one decision-maker that possesses the expertise, training and resources needed to assess and action the provisioned information or intelligence.*

***Data**: Facts and statistics collected.*

***Information**: Data in context or data that informs a decision.*

***Intelligence**: Information about a competitor, adversary, threat, or situation and/or the evaluated conclusions about such information.*

***Digitization**: The institutional process of developing and delivering Information and Decision Advantage.*

***Risk Management**: Forecasting an evaluation of risks and the identification of mechanisms to avoid or minimize their potential impact.*

***Information Sharing and Safeguarding (ISS):** Responsibly accessing, using, and sharing data and information assets in a manner that maximizes the availability to authorized users, while simultaneously protecting assets from unauthorized access, use, appropriation or manipulation.*

their information and enable decision-makers to trust the information and/or intelligence they are relying on to make strategic, operational and tactical decisions.

This document is written from the perspective of military usage or deployment. However, there is nothing inherently military in the Object Management Group's Information Exchange Framework Reference Architecture (Reference A), or any potential design or implementation. All of the concepts can be easily extended to any public or private sector solution. The modelling concepts underpinning the Object Management Group's Information Exchange Packaging Policy Vocabulary (IEPPV) and how they may be applied to sharing and safeguarding can be applied on many information domains using a variety of data and information management technologies.

The IEPPV is focussed on the sharing of information between systems, applications and devices that produce data and information elements in real-time and at machine speeds, that contain sensitive (e.g., private, confidential, legally-significant or classified), including but not limited to:

- Information of Things (IOT) devices;
- Environmental sensors;
- Situational, operational, or cyber awareness applications;
- Operational, business or national intelligence systems;
- Case management systems;
- Data lakes;
- Analytic systems;
- Personnel or Human Resources systems;
- Government program support or delivery systems; and
- Banking and financial systems.

The key concept for the IEPPV was the separation of ISS policy from the software (e.g., APIs) that adjudicates and enforces it. This single concept provides organizations with the ability to, for example:

1. Develop and retain institutional memory;
2. Review and audit the development and deployment of ISS policies;
3. Enable the continuous and rapid development, testing and deployment of ISS capability;
4. Define, implement and deliver secure Data-as-a-Service (sDaaS) where data is captured once and used for many purposes; and
5. Enhance the auditability of data and information environments.

TABLE OF CONTENTS

Page

TABLE OF FIGURES

1. INTRODUCTION

1.1 AIM

Information is a valuable institutional asset, whether that value is measured by the public or private sector organization. Most organizations are being directed to implement practices, processes and tools that will enable them to exploit all sources of data and information as part of their decision-making processes to achieve an information advantage over their competitors or adversaries.

However, as data and information are captured, stored, processed, curated, and analysed, the value of information steadily increases. Competitors or adversaries then seek to disrupt the flow of data and information, or seek to steal or manipulate the information in order to diminish any benefit it may provide, or gain an advantage of their own. The challenge then becomes the task of protecting data as it transitions through its lifecycle and assuring that the resulting information is responsibly shared with authorized decision makers.

This document describes how the Information Exchange Framework Reference Architecture (IEF-RA: Reference A) and the Information Exchange Packaging Policy Vocabulary (IEPPV: Reference B) can be employed to deliver a policy-driven data-centric information sharing and safeguarding capability that enables organizations:

a) To selectively share data and information elements tailored to each recipients' (e.g., individual, organization, partner, system, application or service) needs and authorizations;
b) To capture policies (rules and constraints) governing the sharing and safeguarding of data and information elements as Information Sharing and Safeguarding (ISS) views and viewpoints aligned to ones' own enterprise architecture;
c) To design, implement and execute ISS solutions' lifecycle;
d) To enhance institutional information and data management and deliver data centric security;
e) To improves runtime management and administration of ISS; and
f) To reduce ISS risk and cost.

Information Advantage: The ability to access or develop unique knowledge giving an organization or individual a strategic or tactical advantage in a particular business, operational or mission context.

Policy Driven: The adjudication and enforcement of rules and constraints derived from, and traceable to, user or community approved policy instruments (e.g., legislation, international agreements, regulations, directives, information sharing agreements, operating policy and operating procedures).

Data-Centric: The adjudication and enforcement of information sharing and guarding policies (rules and constraints) governing individual data and information elements.

Information Sharing and Safeguarding (ISS): A set of capabilities that provide users with the ability to responsibly share information based on user needs, user authorizations and data sensitivity.

Responsible Information Sharing: Maximizing the availability of information to authorized decision makers, while simultaneously assuring that data and information is protected from unauthorized access, use, appropriation and manipulation.

1.2 OVERVIEW

The Information Exchange Framework (Reference A) provides an approach for decomposing system-to-system information sharing agreements into their constituent parts. The approach outlined in the Information Exchange Packaging Policy Vocabulary (IEPPV: Reference B) describes a set of Information Sharing and Safeguarding (ISS) views and viewpoints that can be easily aligned to most architecture frameworks (e.g., DODAF, NAF, Zachman and TOGAF) in order to capture ISS policies (rules and constraints).

The ISS views define how data and information elements are processed, packaged and routed as they transition from data elements (e.g., facts and measurements) to information elements that inform decisions and deliver information advantage; Figure 1.

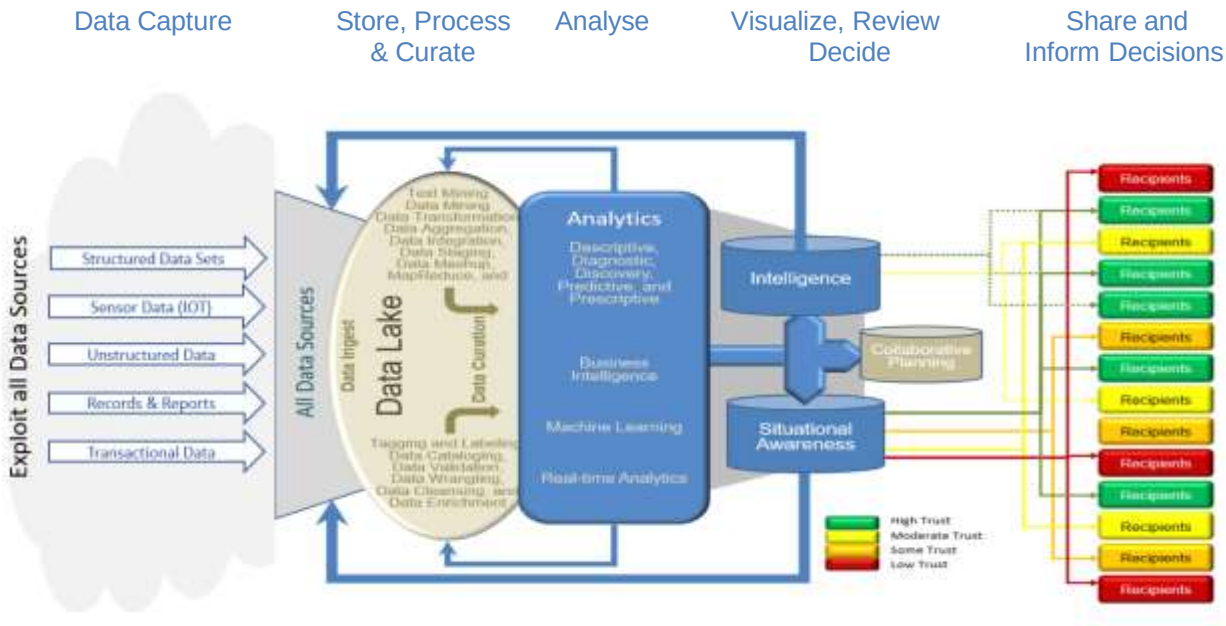

Figure 1 - Data Capture to Information Advantage

1.3 POLICY-DRIVEN DATA-CENTRIC INFORMATION SHARING AND SAFEGUARDING

A core objective for the Information Exchange Framework initiative at OMG was the separation of the ISS software (or service) development and the ISS policy lifecycles. The principle was that the separation would:

1. Increase user control of IEF service operations by extracting rules and constraints implementation from the software API development teams;
2. Increase development and operational flexibility, agility and adaptability by enabling policies to be replaced or updated during testing, exercises and operations;
3. Increase reuse of software and policy components with and across missions and systems; and
4. Reduce overall risk and cost.

Most of all, the IEF approach separates and addresses the concerns of:

1. The business (/operations) by increasing the flexibility, agility and adaptability of deployed information systems during operations, and improving practices, processes and tools to manage, administer and govern how data and information elements are accessed, used, processed, analysed and shared;
2. The Information and Data Management organizations by providing a better architectural and operational understanding of how data and information elements are collected, stored, processed, analysed and shared within the enterprise and with external partners and clients; and
3. The IT organizations by:
    a. Increasing the flexibility and agility of standard ISS infrastructure elements;
    b. Expanding the environments where sensitive data could be securely deployed (e.g., on-premises, deployed platforms, cloud and hybrid); and
    c. Improving opportunities to deliver a day-zero capability to operations.

The modelling practices outlined in this document focus on Concerns 1& 2 (above), enabling information and data management organizations, in partnership with operational Subject Matter Experts (SME), to model ISS policies as the need and requirements for S2S data and information exchange are discovered. Instead of initiating a software development project, these models can be transformed into executable policies (rules and constraints), tested/certified, and deployed to operations. The IEF practices will also provide business and operational units with IEF ISS solutions to better understand the operating capabilities, and provide the secure, flexible, agile and adaptable capability demanded from modern information systems.

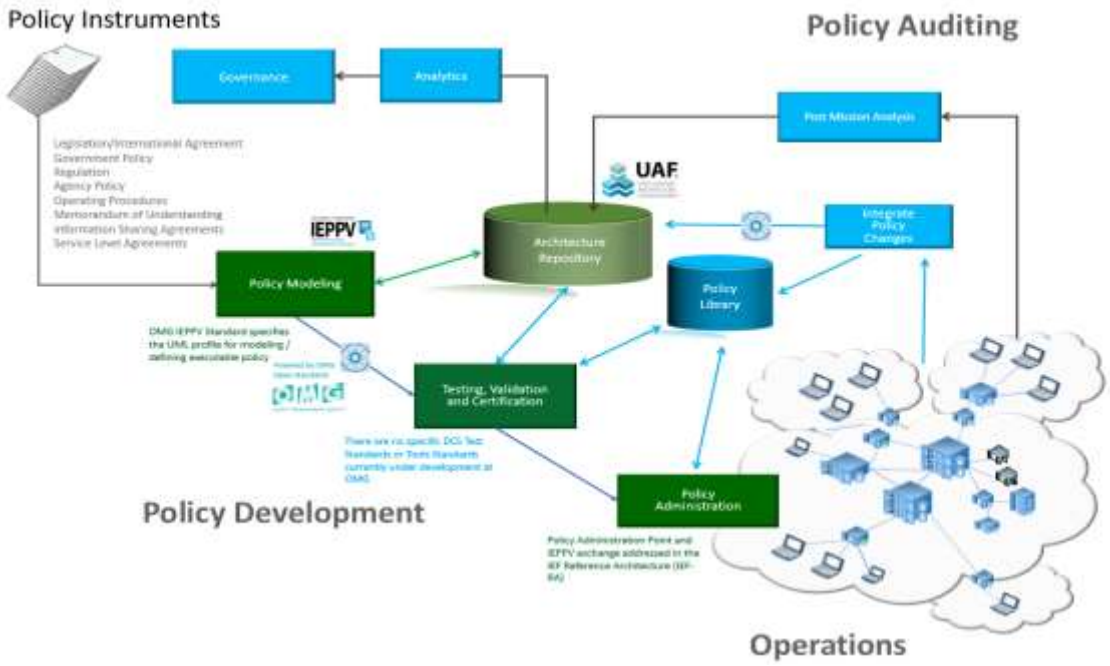

Figure 2: ISS Policy Life-cycle

This is accomplished through:

a) **Architecture Alignment (**Figure 2**)**: The IEF approach employs architecture practices, methods and tools for users to define ISS policies (rules and constraints). This delivers:

   i. Persisting Institutional Knowledge and Memory: Persistence through EA tools that capture information about the ISS environment that enable architecture and design audits, capability certification, and capability reviews, or the production of documentation often neglected during development and Operations and Maintenance (O&M) activities - All can reduce lifecycle cost and risk;

   ii. Traceable to policy instruments (e.g., Legislation/International Agreement, Institutional Policy, Regulation, Operating Procedures, Memorandum of Understanding, Information Sharing Agreements, and Service Level Agreements) through EA tools that improve understanding of how policy instruments are being addressed;

   iii. Aligned to the capabilities, systems, applications, services and interfaces that adjudicate and enforce ISS policy as well as the mission, platforms and nodes where they are scheduled for deployment. This information aids in the monitoring and auditing of ISS during and after missions; and

   iv. Translatable to machine readable and executable formats through the use of Model Driven Architecture (MBA) or Model Based Systems Engineering (MBSE) tools. This eliminates the traditional translation of requirements to API design and the API implementation, thereby minimizing O&M cost and risk.

b) **Policy Driven**: Enabling software applications, services and interfaces to adjudicate and enforce user defined policies (rules and constraints) derived from, and traceable to, user or community approved policy instruments (e.g., legislation, international agreements, regulations, directives, information sharing agreements, operating policy and operating procedures);

c) **Data-Centric**: Enabling software applications, services and interfaces to adjudicate and enforce ISS policies (rules and constraints) for content in the data and information elements and/or metadata labels describing the content; and

d) **Information Sharing and Safeguarding (ISS):** Enabling software applications, services and interfaces to responsibly share data and information content with each recipient based on user needs and authorizations in a manner conforming to user policy and risk tolerance for the information of a specific sensitivity (i.e., privacy, confidentiality, legal-significance or classification).

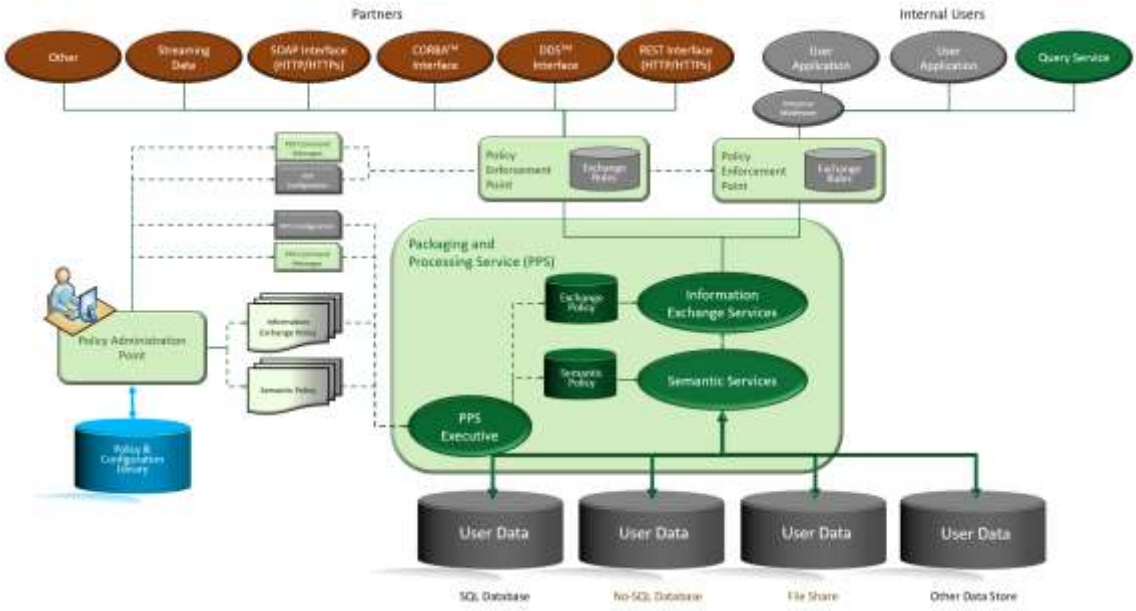

Figure 3: ISS Service Example

As illustrated in Figure 3, The ISS policy is managed and administered during operations using the IEF Policy Administration Point (PAP). For more information on the PAP, refer to the IEF Reference Architecture (reference A) and the Secure Data Service Operating Concept Document (Reference C).

1.4 DOCUMENT OUTLINE

The document is divided into six sections and two annexes, as follows:

Section 1: Introduction - Provides an overview of this document and Information Sharing and Safeguarding (ISS) Policy for System-to-System Interoperability.

Section 2: Modelling Information Sharing Agreements - Provides an overview of how to model an ISA using the IEPPV Profile.

Section 3: Semantic Patterns - Provides an overview of how to model a semantic pattern for the packaging of recipient authorized data sets using the IEPPV Profile.

Section 4: PPS Data Processing - Provides an overview of the processes performed upon the receipt of a message by the IEF Packaging and Processing Service (IEPPS).

Section 5: Data Packaging - Provides an overview of the processes performed to package a dataset authorised for release to a specific recipient.

Section 6: Conclusion - Provides a summary of the IEPPV structure and features of the constituent elements that comprise the IEF/IEPPV approach.

1.5 SCOPE

This document will be described in sufficient detail to allow stakeholders to understand the IEPPV core concepts and value provided by Policy-driven Data-centric ISS without delving into technical details. Though the IEF approach can be utilized in any ISS and data domain, many ASMG clients derive from the Military in Canada, the US and NATO. It is for this reason that the examples presented in this document are drawn from models ASMG developed for CWIX 2020, or are being developed for CWIX 2021. These example models are representative of the data, semantic and exchange models for:

a) The MIP Information Model (MIM) exchange schemas (Reference D);
b) The Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) storage schemas (Reference E); and
c) The NATO Vector Graphics (NVG) exchange schemas (Reference F).

1.6 TARGET AUDIENCE

This Document is provided to inform stakeholders (sponsors, architects, planners, developers, users, maintainers) about the ISS policy models and how they are developed.

1.7 BACKGROUND DOCUMENTS

The following documents inform the definition and development of the Secure Data Service.

A. Information Exchange Framework Reference Architecture IEF-RA), October 2019, Object Management Group, https://www.omg.org/spec/IEF-RA/
B. Information Exchange Packaging Policy Vocabulary (IEPPV), May 2015, Object Management Group, https://www.omg.org/spec/IEPPV/
C. 20201219 ASMG_DCS_SDS_Operating Concept.Docx, December 2020, Advanced Systems Management Group Ltd.
D. MIP Information Model (MIM: STANAG 5643), https://www.mimworld.org/portal/projects/welcome/wiki/Welcome.
E. Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM: STANAG 5525), https://mip.army.gr/.
F. NATO VECTOR GRAPHICS, APP-6D(1) BINDINGS, V1.1
A. Unified Architecture Framework, https://www.omg.org/spec/UAF/
B. Unified Profile for DODAF and MODAF (UPDM), https://www.omg.org/spec/UPDM/
C. Shared Operational Picture Exchange Services (SOPES), https://www.omg.org/spec/SOPES/
STANAG 4774, ADatP-4774 CONFIDENTIALITY METADATA LABEL SYNTAX, Edition A Version 1, DECEDMBER 2017
D. STANAG 4778, ADatP-4778 METADATA BINDING MECHANISM, Edition A Version 1, OCTOBER 2018

1.8 ISS POLICY ENFORCEMENT USE CASE

Figure 4, provides a conceptual use-case for the services used to adjudicate and enforce the policies described in this document:

1. Packaging and Processing Services (PPS): The service or set of services that enforce the ISS policies during the processing and packaging of information elements (messages). The PPS has separated the concerns between the receipt and release of messages, and the processing and packaging of messages. This is addressed by:
    a. The Information Exchange Controller Service(s): Provides the ability to receive data messages from Policy Enforcement Points (PEP), and publish user authorized content in formats and protocols agreed to by each user, and release messages to policy that define data to specified PEPs.
    b. The Semantic Processor Service(s): Provides the ability to process received message content and marshal it to user specified data stores. The semantic processor also provides the ability to package data and metadata in accordance with user specified policy;
2. PEP (Policy Enforcement Point): Provides the integration point and access and release controls for the receipt and release of information elements (messages) from user specified messaging or exchange infrastructure. PEPs may include:
    a. Inbound PEP: Provides receipt controls, assuring the PPS and data store(s) only receive message content they are authorized to process and store;
    b. Outbound PEP: Provides release controls, assuring the message content is only released to authorised users (e.g., individual, role, system, application, service, or device); and
    c. PAP PEP: Assures that administrative messages are only exchanged with an authorized Policy Administration Point (PAP); and
3. User Configurable Libraries: Provides the ability to implement mission specific environments through policy and configuration. Library areas include:
    a. ISS Policy Libraries: Loaded by an authorized PAP to govern how data and information elements are processed, packaged and published by the PPS (information exchange and semantic policies);
    b. Parser Library: Governs how messages received by the PPS are decomposed into distinct data elements;
    c. Schema Libraries: Define the structure and syntax of received and released messages;
    d. Mapping Files: Govern how received message data elements are assigned to data entities defined by the data store;
    e. Transformation Libraries: Define methods to perform data transformations on data elements as they transition between exchange and storage semantics; and
    f. Filter Rules: Define the rules for each filter in the included semantic pattern.

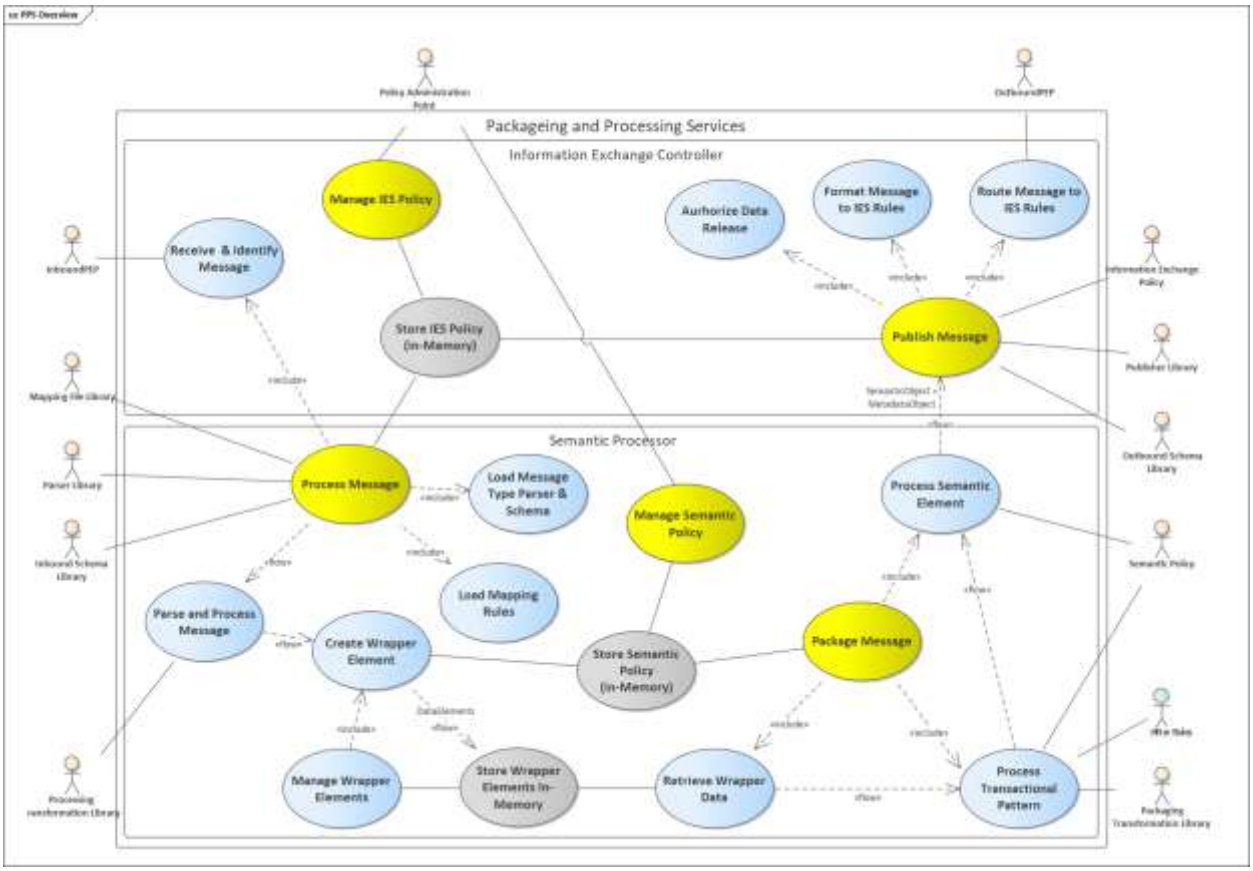

Figure 4: Policy Enforcement Services

1.9 EXAMPLES USED IN THE DOCUMENT

The example models used in this document were drawn from a set of models developed for and used during Coalition Warrior Interoperability Experimentation (CWIX) 2020. The models were developed using an IEPPV Profile for the SPARX Enterprise Architecture (EA) modelling tools.

The models were transformed, using MDA tools developed by ASMG, into a set of executable ISS policies (rules and constraints), deployed to the Secure Data Services (SDS) and used to successfully test and demonstrate Data Centric Security (DCS) with several NATO partners and technologies.

2. MODELING INFORMATION SHARING AGREEMENTS

2.1 ISA OVERVIEW

An Information Sharing Agreement (ISA) defines and documents an exchange between data-providers and data-recipients within a specified information domain (e.g., C2, ISR, Planning, Logistics, and Personnel). The Information Exchange Framework's Packaging Policy Vocabulary (IEPPV) uses the term Information Exchange Specification (IES) for expressing an ISA as part of an architectural construct. The IES encloses the following architectural elements:

a) An Information Specification (IS) that specifies the semantic patterns and associated filters that govern the packaging of data and information elements that are releasable to Data-Recipients under that agreement;
b) A Distribution Specification (DS) that documents the configuration of the exchange services used to route information elements to Data-Recipients; and
c) Configuration attributes that direct the operation of data processing, packaging and release services under the agreement.

An individual participant (e.g., individual, organization, node, system, application or service) may participate in any number of ISAs, each represented by a single IES. To participate, the participant must support the DS and the message types contained in the IS.

Information Sharing Agreement: An accord between data-providers and data-recipients regarding the terms, content and mechanisms to be used to exchange information. Also referred to as Information Exchange Specifications (IES), Information Exchange Requirements (IER), and Memorandum of Understanding (MOU).

Data User: Any individual, application, service, system, platform, node, appliance, or sensor that collects, stores, processes, analyses, visualizes, shares or employs data or information elements within the environment.

Data-Provider: Any data-producer, data-owner, data-custodian authorized to service (*provision data or information elements*) an information sharing agreement.

Data-Recipient: Any Data User authorized to access, use and/or manipulate data and information under an information sharing agreement.

Information Exchange Specification: Specifies the characteristics and configuration of a specific information sharing agreement.

Information Specifications: Identifies the information elements to be shared under the information sharing agreement between participants.

Distribution Specification: Defines the characteristics and configuration of the information exchange mechanism agreed by the participants.

Participant: Any participant (provider and/or recipient) to the information sharing agreement including: individuals, organizations, systems, applications, or services.

2.2 IES COMPONENTS

The IES is used to align information elements to be shared under the ISA to the configuration of mechanisms used to exchange those elements. The IES can define the agreement between the data producers and:

1. A single peer participant (e.g., individual, system, application, or service); or
2. A Community of Interest (COI); or
3. An organization or partner agency.

The IES is structured in a manner that enables the auto translation of models into executable ISS policies that can be managed and administered during operations.

2.3 IES EXECUTION

The Information Exchange Specification viewpoints, described below, are transformed into an executable form of policy (rules and constraints) and configuration files by the PPS Information Exchange Controller Services. The policy is provisioned to the PPS (Information Exchange Controller) at runtime – see Information Exchange Policy.

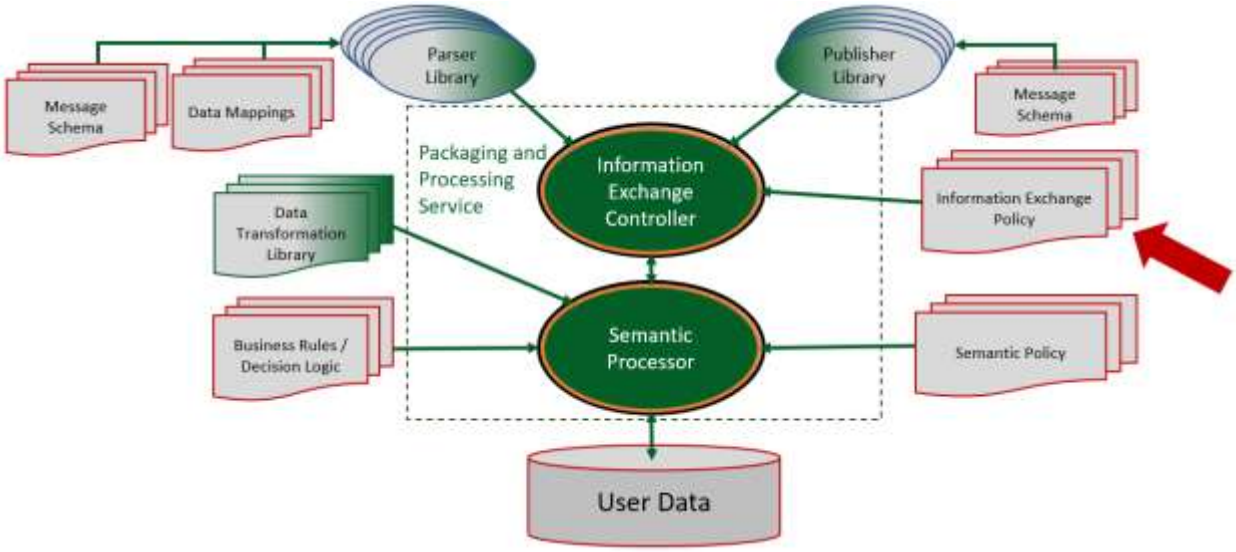

Figure 5: ISS Policy Enforcement Environment

The Information Exchange Controller (IEC) uses the policy to govern the receipt and release of information from the Policy Enforcement Points (References A and C). The IEC is responsible for:

1. Parsing received messages and transitioning data and metadata elements to marshalling services that use mapping files to create wrapper elements for the Semantic Processer; and
2. Formatting and routing data and metadata objects from the Semantic Processor.

The IEC also forms the interface between the PPS and the PEP or the Users' middleware solutions if the PEP is not employed.

2.4 IES STRUCTURE

As illustrated in Figure 6, the IES formalizes the definition and alignment of the information and distribution elements of an Information Sharing Agreement. The IES defined by the OMG IEPPV specification (Reference B) defines the following components:

1. Information Exchange Specification: Encloses the rules and constraints governing the information elements shared under an agreement and the mechanisms used to share those elements;
2. Distribution Specification: Defines the rules and configuration parameters governing the exchange of information elements between participants under the agreement;
3. Information Specification: Defines the messages and information objects exchanged under the agreement;

4. Message: Encloses the rules and constrains governing the packaging of complex, multi-element messages (e.g., digest, multiple payloads, linkages and attachments); and
5. Filtered Semantic Element: Defines rules and constraints governing the packaging of a single information object and determines its releaseability under the agreement.

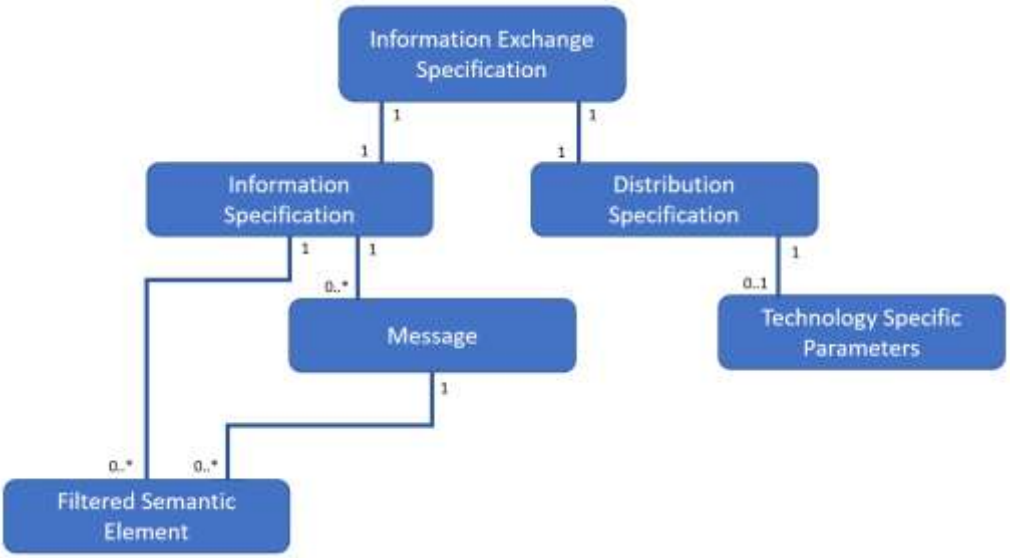

Figure 6: IES Structure

2.4.1 IES Example

The information exchange specification diagram, Figure 7, is a very simple construct. It contains a single entity, namely NCDF-BSO-Vessel to be exchanged between parties.

2.4.2 IES-Tags (Configuration Parameters)

As illustrated in Figure 7, the IES includes a set of "tags" or "tag-values" representing configuration parameters (attributes) that tailor the runtime adjudication and enforcement of policy and service operations to the needs, capabilities and authorizations of the IES recipients. The IEPPV does not specify specific sets of tags or tag-values to be implemented. The tag-values are an inherent part of UML and thus can be exploited by IEPPV profile implementers and IEF service providers to extend the profile to exploit capabilities within their service implementation.

The "tags" illustrated in Figure 7 were added to the ASMG implementation of the IEPPV UML profile and PPS Services to facilitate operations in a coalition environment (e.g., NATO or Whole of Government).

Tags include:

1. **AddInternalMarks** (Boolean): Identifies if metadata is created and bound at each aggregation point in the messages released by this IES;

2. **AddMarksManually** (Boolean): Identifies the metadata is to be manually generated during message construction (*for demonstration only*);
3. **AllowCanadianExtensions** (Boolean): Identify whether or not Canadian extensions to the data environment should be filtered out of the messages released by this IES;
4. **ApplyWSMP** (Boolean): Identifies whether or not WSMP is applied to the message under this IES;
5. **AuthorizedProtectionLevel** (Enumeration): Enables the user to set protection levels for information released under the IES;
6. **AuthorizedReleasableTo** (String): Enables the user to set a list of recipients authorized to receive information under the IES;
7. **AuthorizedSecurityLevel** (Enumeration): Enables the user to set a security level for information elements released under this IES;
8. **DefaultExchangeSchema** (String): Enables the user to specify a generic schema for the exchange of data using this IES. It is superseded if there is an AppliedExchangeSchema in the MessageSpecification or FilteredSemanticElements;
9. **DefaultMappingFile** (String): Enables the user to specify a generic mapping file for the marshalling of data received using this IES. It is superseded if there is an AppliedMappingFile in the MessageSpecification or FilteredSemanticElements;
10. **DefaultParser** (String): Enables the user to specify a generic schema for the exchange of data using this IES. It is superseded if there is an AppliedExchangeSchema in the MessageSpecification or FilteredSemanticElements;
11. **IES_StartStateActive** (Boolean): Sets the start state of the IES to Active when the IEPPS is started;
12. **IES_Active-StartDateTime** (String): Sets a time to enable or start an IES;
13. **IES_Active-EndDataTime** (String): Sets a time to disable or stop an IES;
14. **IncludeAttachments** (Boolean): Identifies whether or not attachment(s) should be included in the messages released by the IES;
15. **IncludeMetadata** (Boolean): Identifies whether or not metadata should be bound to the message released by this IES;
16. **isActive** (Boolean): Identifies the current state of the IES;

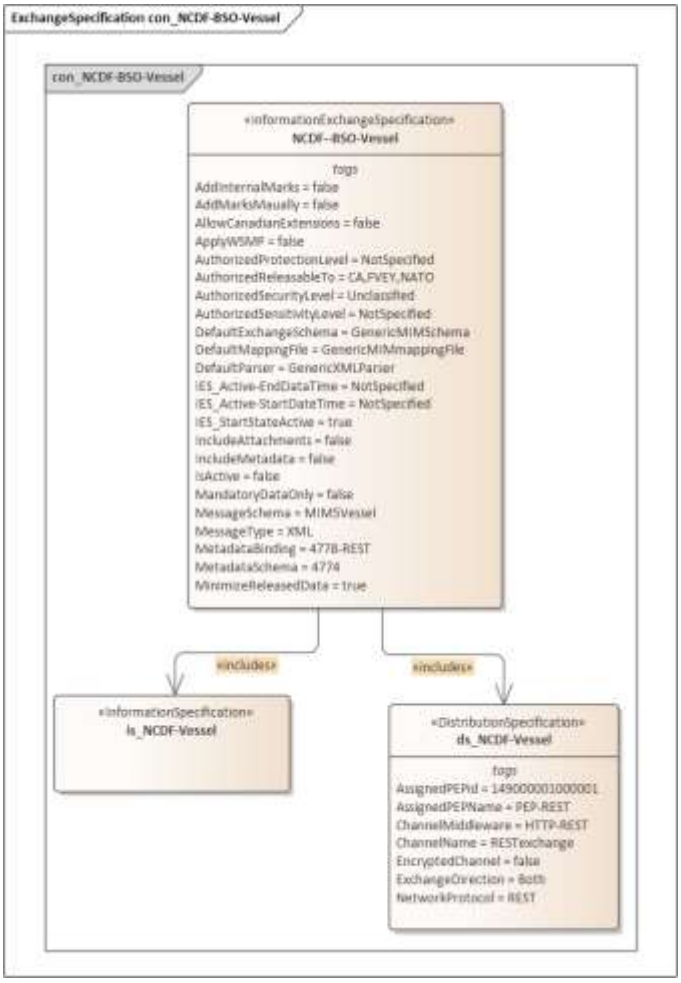

Figure 7: Information Exchange Specification Example

17. **MandatoryDataOnly** (Boolean): Identifies if the message should only include mandatory data and information elements (only mandatory data aggregates);
18. **MessageSchema** (String): identifies the message schema (e.g., XSD) used to format messages using this IES;
19. **MessageType** (Enumeration): Identifies the type of message (e.g., XML, JSON, other) exchanged using this IES;
20. **MetadataSchema** (String): Identifies the message schema (e.g., XSD) used to format message metadata using this IES;
21. **MetadataBinding** (Enumeration): Identifies the metadata binding to be used for messages using this IES; and
22. **MinimizeReleasedData** (Boolean): Identifies that only the minimum required data should be aggregated for messages released by this IES (minimization of data attributes).

The values for these attributes can be set during policy development, or set by an authorized operator during operations, providing continuous user control over the release of information. E.g.:

a) Activating or deactivating the entire agreement;
b) Activating or deactivating specific filters in the semantic (data redaction) and exchange (go-no-go) processing services, e.g.:
    a. Redact Canadian specific data; or
    b. Minimize the data set; or
c) Allowing all or only mandatory data to be released.

2.4.3 Information Specification Viewpoint

The "Information Specification", is an architecture construct that enables architects and analysts to develop reusable sets of messages and/or information objects. The "Information Specification" is primarily used to simplify the development and management of policy models. When it is reused by multiple IESs, the packaging can be varied by the IES-Tags, which direct variations in the aggregation, transformation, labelling and redaction of data and information element, or the generation of metadata.

As illustrated in Figure 8, the NCDF data information specification includes three filtered semantic elements or data patterns:

1. LocationWeather (weather for a specific location);
2. Vessel (information on vessels); and
3. Weather (general weather information).

2.4.4 Information Specification Example

Figure 8 illustrates how architects and analysts can develop patterns of Message Specifications and/or Filtered Semantic Elements to be used by multiple information exchange specifications. IES parameter (e.g., Tags) can then be used to activate semantic policy features (e.g., transformations, and filters) to govern the data and information objects for a specific user.

Notes:

1. The Tag-Values assigned to the "MessageSpecification" or FilteredSemanticElement" supersede the Tag-Values specified in the IES. The IES values are used if the subtended element values are not set or not specified; and
2. Actual values for the Tags can be set during operation by an authorized PAP and administrator, or as part of the configuration of the PPS instance.

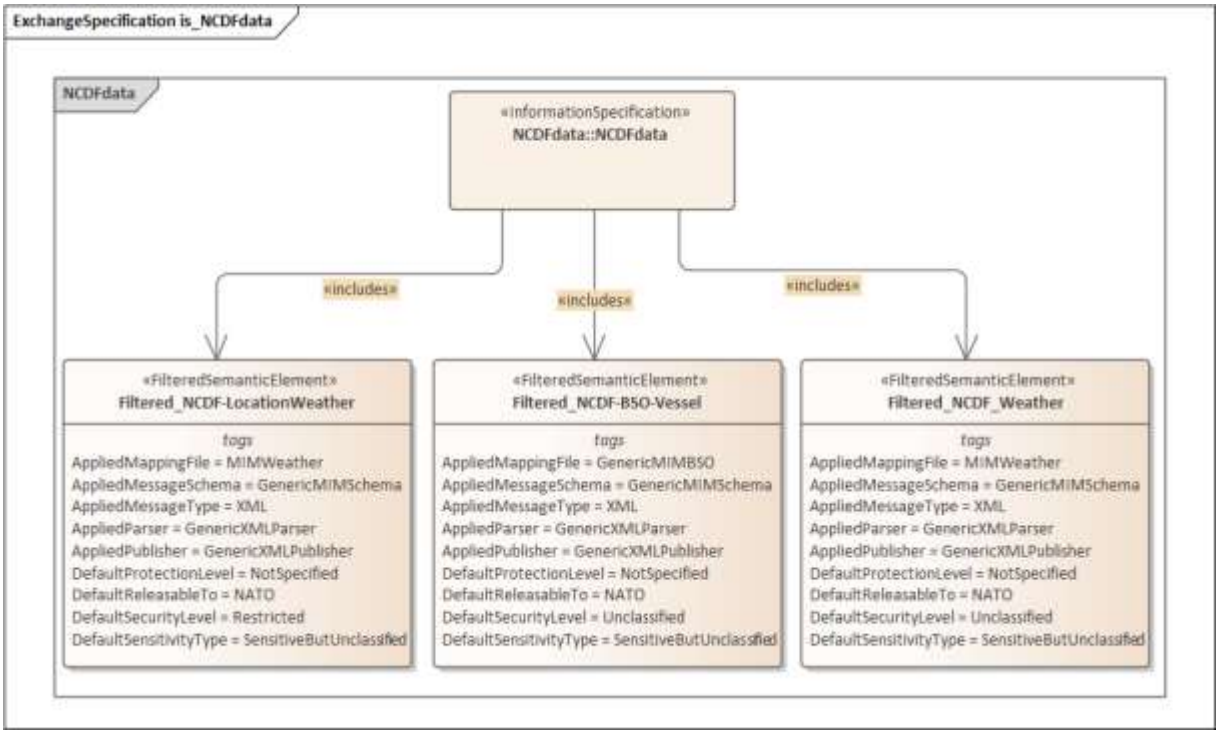

Figure 8: Information Specification Example

2.4.4.1 Message Specification

The Message Specification was not used during testing and not included in this document. For more information refer to the IEPPV Specification (Reference B).

2.4.4.2 Filtered Semantic Viewpoint

The "Filtered Semantic Element" provides the alignment between the semantic elements (data packaging patterns) used to produce a releasable data set for one or more participants to an IES, and the filters (go/no go) that govern the produced dataset as specified for, or releasable by, the IES. As the semantic elements are reusable by multiple IESs, these filters adjudicate the releaseability of the data.

Note:

1. Filters can be set for either data or metadata elements in the dataset; and
2. The releasable dataset contains both the data and metadata elements packaged by the semantic processing and policies.

2.4.4.2.1 Filtered Semantic Element Structure

As illustrated in Figure 9, the filtered semantic element encloses:

1. **Filtered Semantic Element**: Encloses and aligns a set of dynamic release filters to determine if a dataset is releasable to a specific Information Sharing Agreement. These filters get their dynamic nature because they can be managed and administered by authorized operators during operations;
2. **Filtered Transactional Element**: Identifies the attributes (data and/or metadata) that the operator can use to define a filter;
3. **Dynamic Filter**: Identifies the attributes, drawn from semantic patterns that the operator may use to define a release filter;
4. **Semantic Element**: Identifies the semantic pattern to which the filters apply. More on the semantic element in Section 3;
5. **Transactional Element**: Identifies the data pattern containing data and metadata elements used by the filter. *(included for completeness – not used in the filter patterns);*
6. **Wrapper Transactional Element**: Identifies the data element containing the data attributes used by the filters; and
7. **Wrapper Element**: Identifies the data source for the attributes used by the filter *(included for completeness but not used in the filter patterns).*

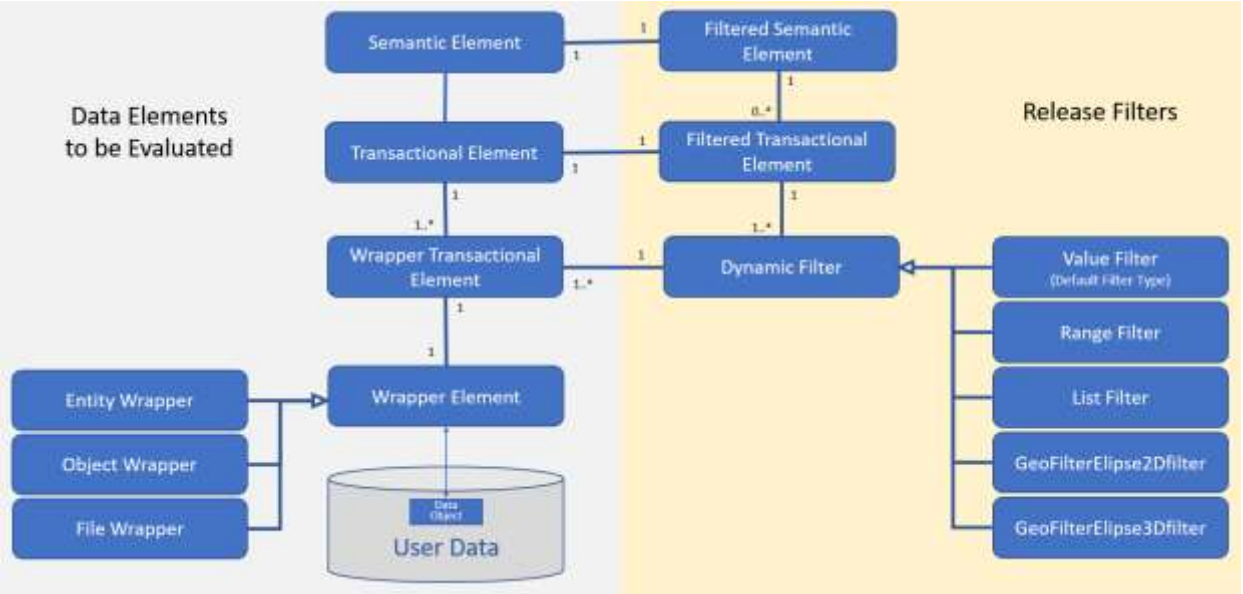

Figure 9: Filtered Semantic Element Structure

2.4.4.2.2 Filtered Semantic Element Example

Figure 10 provides an example of a filtered semantic element. As illustrated, the filtered semantic (Filtered_NCDF-BSO-Vessel):

- References its corresponding "Semantic Element" (NCDF-BSO-Vessel); and
- Includes two Transactional Filters:
  - The Vessels Hostility Status; and
  - The Vessels Position (e.g., Recipients Area of Interest).

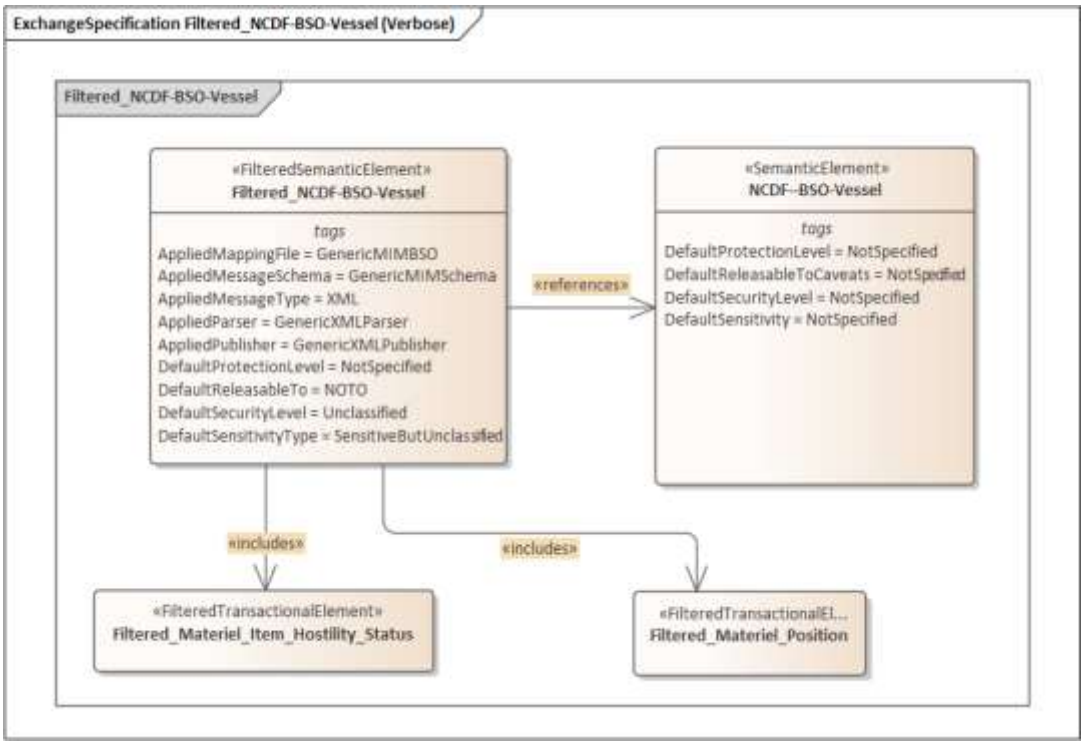

Figure 10: Filtered Semantic Element Example

Notes:

1. The naming of elements in IEPPV based policy models is at the discretion of the architect and/or analyst developing the model. The naming convention should assist users in the implementation of filters during operations. The naming convention used in the model examples were meaningful to the individual performing the testing during CWIX;
2. The addition of the release filters may reduce the sensitivity of the potential data released. In this case the default protection attributes (i.e., protection level, releasable to, security level and sensitivity type) may be relaxed over those of the "Semantic Element". Attributes of the "Filtered Semantic Element" supersede those of the "Semantic Element" when specified;
3. The addition of attributes (tags and specified values) to direct packaging and filtering of data and information elements can also be added to reduce sensitivity of the packaged dataset; and
4. Computed protection attributes always supersede defaults set at design or by the administrator.

2.4.4.2.3 Filtered Semantic Tags

The Tag-Values associated with the "FilteredSemanticElement", Figure 10, provide the analyst or architect with the ability to set specific values for an individual Filtered Semantic.

1. **AppliedMappingFile** (String): Identifies the mapping files defining the mapping between received data elements and the storage model;
2. **AppliedMessageSchema** (String): Identifies the message schema (e.g., XSD) used to format messages using this IES;
3. **AppliedMessageType** (Enumeration): Identifies the type of message (e.g., XML, JSON, other) exchanged using this IES;
4. **AppliedParser** (string): Identifies the parser to be used to extract data from received messages on this IES;
5. **AppliedPublisher** (String): Identifies the publisher to be used to prepare the message to be sent on this IES;
6. **DefaultProtectionLevel** (): Default protection level for the content of the dataset if a protection level is not computed during packaging;
7. **DefaultReleasableTo** (): Default releasable to list for the content of the dataset if a releasable to list is not computed during packaging;
8. **DefaultSecurityLevel** (): Default security level for the content of the dataset if a security level is not computed during packaging; and
9. **DefaultSensitivityType** (): Default sensitivity level for the content of the dataset if a sensitivity level is not computed during packaging.

2.4.4.3 Filtered Transactional Viewpoint

The filtered transactional element enables architects and analysts to define filter patterns for the semantic that are meaningful to the operational environment.

2.4.4.3.1 Filtered Transactional Element Structure

The filtered transactional structure is outlined in Figure 9, and not repeated here.

2.4.4.3.2 Filtered Transactional Element Tags

The ASMG implementation of the IEF services does not identify a need for parameterization of the filtered transactional element. Other implementers may find a use for adding tags (parameters) to these elements.

The ASMG implementation uses the parameters (tags) defined by the IES, and filtered semantic element to guide release filtering.

2.4.4.3.3 Filtered Transactional Element Example

Figure 11 provides an example of a filtered transactional element. As illustrated, the filtered transactional element semantic (Filtered_Materiel_Item_Hostility_Status) identifies the type of filter (DynamicFilter) and the source wrapper_TransactionalElement or wrapper that holds the data to be evaluated by the filter (i.e., Materiel and ObjectItemHostilityStatus).

Notes:

1. A filtered transaction may enclose multiple dynamic filters; and
2. The DynamicFilter is a ValueFilter by default and can be replaced by a ListFilter, RangeFilter, or Geospatial Filter as needed.

2.4.4.4 Message Specification

A "Message Specification" is used instead of a "Filtered Semantic Element" to enable the user to produce complex messages with many parts or payloads, e.g.:

a) Metadata (data describing the message and content);
b) Digest (summary of the message content);
c) Multiple Payloads (message content);
d) Links (references between the payloads and the attachments); and
e) Multiple attachments (documents or objects added to the message).

The Metadata, Digest, Payloads and Links each require the packaging of a specific semantic pattern. When a "Message Specification" is used, the IES processing waits for all elements to be packaged and authorized before formatting the message for release.

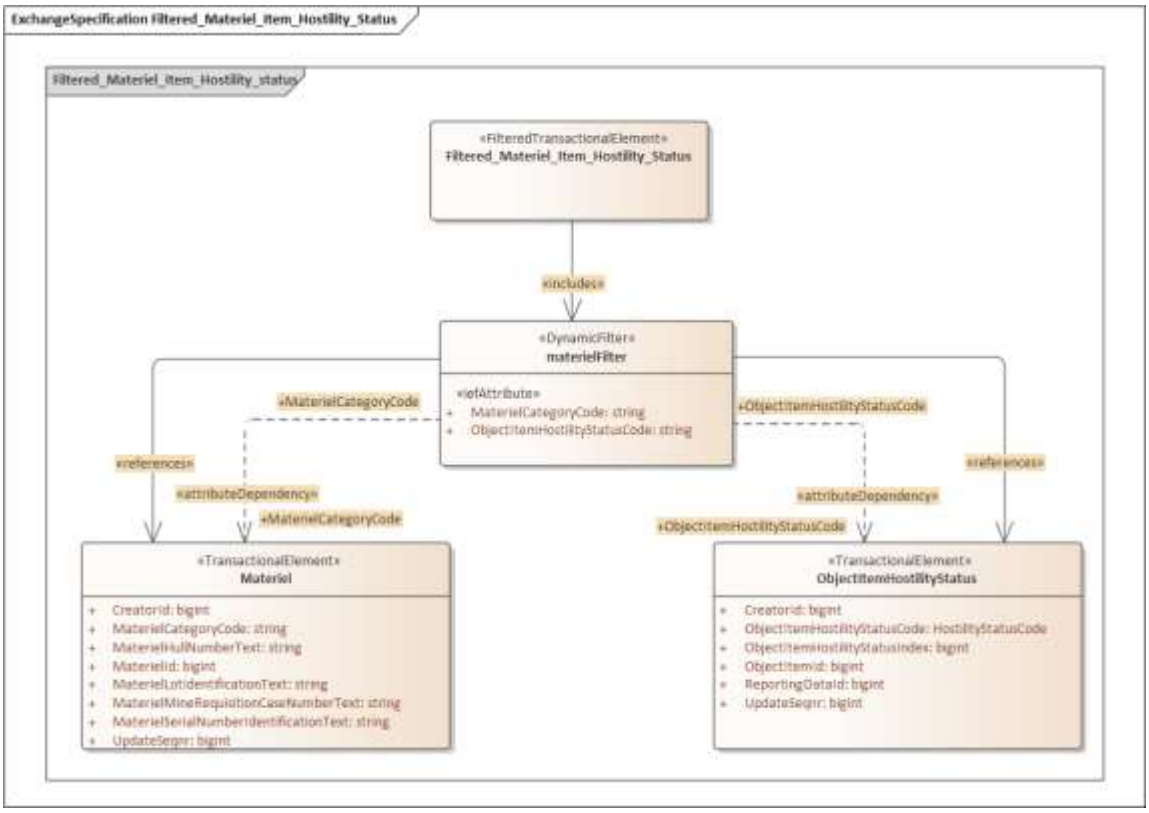

Figure 11: Filtered Transactional Element

2.4.5 Distribution Specification

As illustrated, in Figure 7, the distribution specification holds the information technology specification to be used for the IES. Within the ASMG implementation of the IEF-RA (Reference A), each exchange specification is assigned to a specific Policy Enforcement Point (PEP) that acts as:

1. The access control point to the IEF services (e.g., Secure Data Service) and the data they are protecting; and
2. The integration point for users and integrators to integrate the IEF services into the environment.

As much of the technical configuration is addressed by the PEP, the ASMG implementation only requires a small number of configuration parameters. Other implementers may require a rich set of distribution configuration parameters.

2.5 ISA ALIGNMENT TO ARCHITECTURE

The IEPPV policy models (views and viewpoints) may be aligned to architecture frameworks, and based on our focus on CWIX, include:

1. An IES assigned to an information resource exchange;
2. IESs focused alignment to participants; and
3. Participant focused alignment to IESs.

This document focuses on an architectural alignment between the NAF and DODAF.

2.5.1 Assign to Resource Exchange

In the case of NAF and DODAF, the IES is aligned to the resource exchange or Needline between two Nodes, Operational Performers or Participants (Figure 12). As we are discussing data or information exchange, the Information Exchange Specification enables the decomposition of the rules and constraints governing the exchange from the perspective of the data or information provider.

The IEPPV viewpoints also enable the provider to define the mapping and processing of data and information objects as they transition between exchange and storage semantics, and the storage semantics and the exchange semantics.

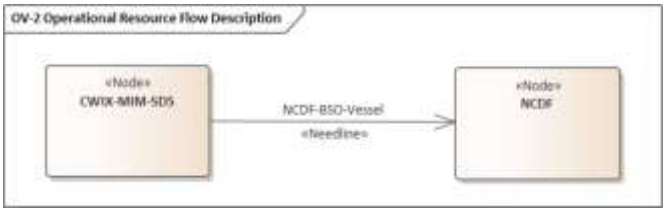

Figure 12: Resource Exchange

2.5.2 Participant Perspective

As illustrated in Figure 13, the IEPPV offers a viewpoint that enables architects and analysts to identify IES participation from the perspective of the Nodes, Operational Performers or Participants. The viewpoint is extremely useful when setting up node configuration files in preparation for testing, desktop exercises or operations.

2.5.3 IES perspective

Alternately, the models can reverse focus and identify all the participants to an agreement. This is useful when planning out information flows and determining if decision makers are receiving the information they need and are authorized to receive.

Notes:

1. Not all viewpoints need to be developed. In many cases all three viewpoints can be derived if one is entered into a modelling tool; and
2. The alignment of the IES to the EA views and viewpoints offers the opportunity to analyse the flow of information between nations, organizations, systems, applications and interfaces and identify gaps, leaks, risks, and threats starting during design and continuing during operations. Comparing operational logs with the original architecture will enable continual development and improvement of ISS capability.

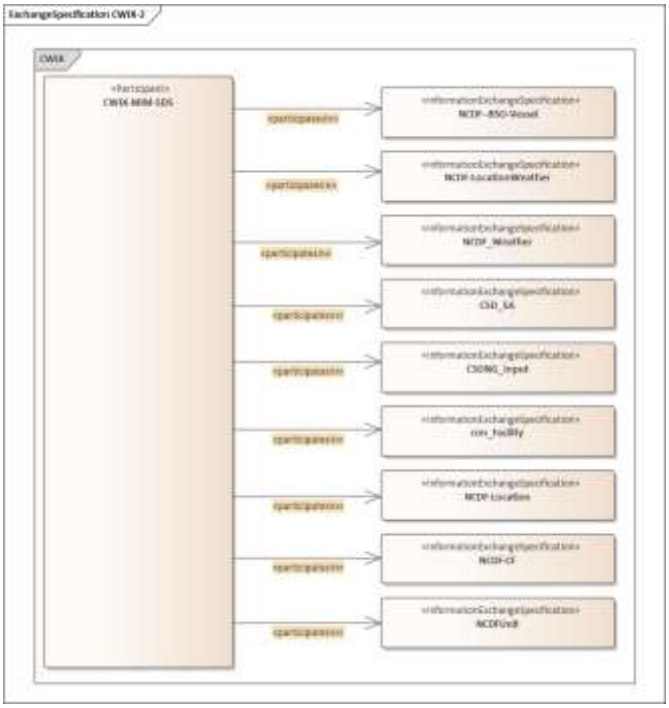

Figure 13: IES Perspective

2.6 IES EXTENSIBILITY AND CONFIGURABILITY

The function of the IES Controller, Figure 5, can be configured or extended in several ways:

1. During Implementation: Though continual development and DEVSECOPS processes:
    a. Deploy new PEP connections to new middleware and access controls that enable the PPS to securely exchange data with authorized users (individual, systems, applications, services, and devices);
    b. Deploy new parsers and publishers that enable the processing and packaging of additional message semantics and protocols;
    c. Deploy new or extended exchange policies; and
    d. Manually extend or adapt exchange policies using an authorized Policy Administration Point.

3. SEMANTIC PATTERNS

3.1 INTRODUCTION TO IEPPV SEMANTIC PATTERNS

Within the IEPPV™, the "Semantic Element" is a data pattern that encompasses the set of data packaging patterns that describe how data and information objects are aggregated, transformed, labelled and redacted (/filtered) to produce a releasable data set for a specified recipient or set of recipients associated with an IES. The semantic element tells a Packaging and Processing Service (PPS) how to transform data conforming to the semantics of the users' (e.g., data owners, data producers or data custodians) own data store to the agreed semantics of an information exchange.

The IEPPV™ defines a UML IEPPV™ profile for modelling IESs and semantic patterns. ASMG implemented this profile within SPARX Enterprise Architect, and used the architecture-based approach to model user defined ISS policies, transform them into an executable form, and have the SDS PPS execute these patterns and govern the release of information to authorized users. The process assures that the user, who owns the policy models, retains ownership of the operations in their information environment.

ASMG uses an architecture-based development environment as illustrated in Figure 14. ASMG uses a number of off-the-shelf features delivered by many

KEY CONCEPTS

Semantic Element: (IEPPV) Composite of rules governing the assembly of data elements in accordance with commitments defined by an information exchange (or sharing) agreement and policies pertaining to the safeguarding of sensitive information.

Transactional Element: (IEPPV) A reusable pattern comprising rules governing the assembly and processing of data and information elements.

Wrapper Element: (IEPPV) A logical construct that wraps or encapsulates the definition of a data set, table entity, triple, file, etc. A wrapper directly maps to a data instance (e.g., row of data in a database application, an object in a NoSQL datastore or a file on disk) in the logical data model and the physical data model.

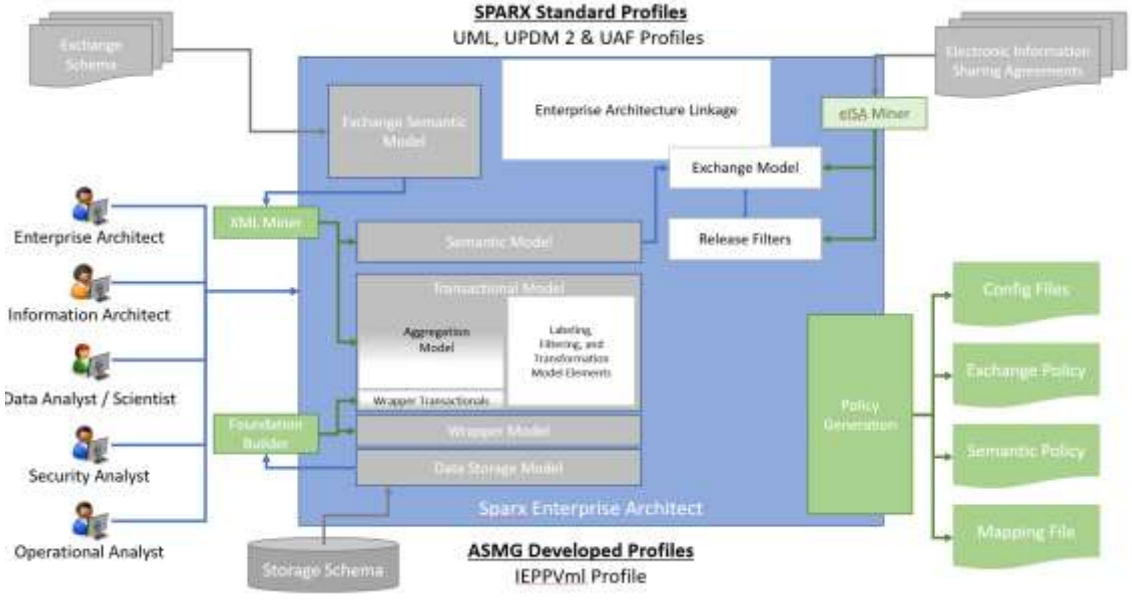

Figure 14: ISS Policy Development Environment

architecture/modelling tools. The ASMG selection of SPARX EA for the current implementation is based on its rich feature set and it's relatively low cost. Similar environments can be implemented using other architecture tools and modelling languages.

3.1.1 Development Environment Considerations

When ASMG undertook the development of an ISS policy development environment, we focused on:

1. **Standards Based**: As many components as possible are based on open international standards that can be implemented by multiple users, vendors or integrators. To this end, ASMG joined the Object Management Group (OMG) and championed or contributed to the development of:
    a. Shared Operational Picture Exchange Services (SOPES, Reference I);
    b. Information Exchange Framework Reference Architecture (IEF-RA, Reference A);
    c. Unified Architecture Framework (UAF, Reference G); and
    d. Unified Profile for DODAF and MODAF (UPDM, Reference H);

    These efforts[1] to develop standards continue at the OMG;

2. **Architecture Alignment**: Ensured the exchange and semantic models can be aligned to views and viewpoints within common architectural frameworks. This to enable users to align information sharing and safeguarding elements with the organizations, missions, systems, platforms, applications and interfaces that use or implement the capabilities. To this end, ASMG formally aligned the IEPPV (Reference B) to the UPDM and UAF and informally with TOGAF and Zachman. Evident in this document is the alignment through UPDM to DODAF and NAF;
3. **Evolutionary Process**: Provided users with the ability to evolve their ISS capabilities. The scale, scope and complexities of information sharing and safeguarding make the development of comprehensive specifications impractical, if not, impossible. Separating the system/software life-cycle and the ISS policy lifecycle helps to enable a more flexible, agile and adaptive ISS development environment;
4. **Enable Automations**: Provided users with a set of tools that enable users to reuse relevant environmental artefacts (e.g., exchange and storage schemas) to seed the ISS policy models. On completion of the models, provide tools that automate the generation of runtime artefacts (e.g., semantic policy, exchange policy and configuration files);
5. **Maximize Flexibility, Agility and Adaptability**: Ensured that practices, processes, technologies and tools enable users to rapidly and securely:
    a. Transition new or evolving ISS needs from concept to operations;
    b. Deploy a Day-0 ISS capability; and
    c. Adapt ISS policy and service configurations to address changing operational conditions (e.g., organization, roles, threats, and risks); and
6. **Enable ISS Governance**: Effectively and efficiently generate and capture the information needed to govern:
    a. The implementation, deployment and operation of ISS capability; and
    b. The receipt, storage, processing, analysis, use and release of data and information elements.

[1] https://www.omg.org

In this regard the ASMG approach is to maximize the capture of design and implementation data as part of standardized architecture metadata and artefacts. During operation, the IEF services are being designed:

1. To log transactional data for monitoring and auditing;
2. To log all user changes to the environment; and
3. To enable the capture and storage of policy and service configurations at any point during operations.

3.1.2 Development Environment Components

The core components of the ISS policy development environment include:

1. **Architecture Tool**: Provides the user interface needed:
    a. To complete a model not generated from existing artefacts and/or enhance existing models;
    b. To align the ISS policy models, other views and viewpoints in the architecture framework;
    c. To define transformations, labelling, and filters for the transactional model; and
    d. Evolve, enhance or correct exchange and semantic policy models;
2. **XML Mining Tool**: Mines an XML model imported into SPARX EA and generates:
    a. The semantic(s) specified in the exchange schema;
    b. The generation of the data aggregation models;
    c. The wrapper transactional; and
    d. Wrappers if no storage model(s) is defined;
3. **eISA Miners**: (Future) Mine a formal electronic Information Sharing Agreement (eISA) between information sharing partners;
4. **Foundation Builder Tool**: Mines the data storage model to identify the data elements needed by the semantic pattern and generates the foundation (wrappers and wrapper-transactionals) for the semantic model;
5. **Policy Generation Tool**: Mines the IEPPV view and viewpoint metadata and generates executable files for the PPS service, including:
    a. (Future) Configuration files for the PPS and PEP;
    b. Exchange policy governing the operation of the Information Exchange Controller;
    c. Semantic policy governing the operation of the Semantic Processor; and
    d. (Future) Mapping file governing the message parsing services; and
6. **Modelling Profiles**: UML profiles that provide the modelling semantics for the development of:
    a. Exchange schema;
    b. Storage schema;
    c. ISS policy viewpoints; and
    d. Related architecture views and viewpoints.

Note: If there are no artefacts from an existing environment, the entire environment can be developed from scratch using the architecture tools that enable the appropriate modelling profiles.

3.2 SEMANTIC PATTERN EXECUTION

The semantic viewpoints, described below, are mined and transformed into an executable set of policies (rules and constraints) that govern the operation of the Semantic Processor services. The executable policies are ingested at runtime by the PPS services. As illustrated in Figure 15, the Semantic Processor employs three extensible elements:

1. **Semantic Policy**: See Section 3.1.1 for information on policy development;
2. **Data Transformation Library**: Results of a standard software development activity to define and develop the operations to transform data elements. The library consists of standard transformations and user defined (specialized) transformations. These operations are reusable and can be shared between instances of the ISS environment; and
3. **(Future) Business Rules and Decision Logic**: Logic modules for things like complex filtering or labelling rules. ASMG is looking at Decision Modelling Notation (DMN) to model decision logic and align it with other ISS viewpoints.

The Information Exchange Controller (IEC) uses the policy to govern the receipt and release of information from the Policy Enforcement Points (Reference A and C). The IEC is responsible for:

1. Parsing received messages and transitioning data and metadata elements to marshalling services that use mapping files to create wrapper elements for the Semantic Processer; and
2. Formatting and routing data and metadata objects from the Semantic Processor.

The IEC also forms the interface between the PPS and the PEP or the Users' middleware solutions if the PEP is not employed.

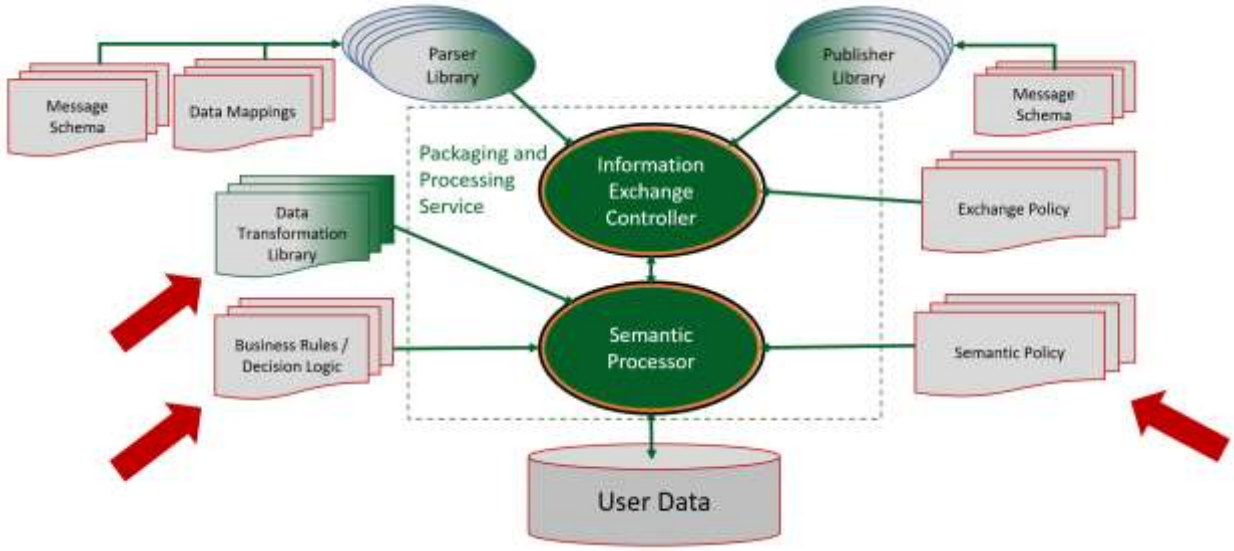

Figure 15: Semantic Processing

The Semantic Processor adjudicates policy that governs the packaging (aggregation, transformation, labelling and redaction) of data and metadata objects for release and the processing (parsing, transformation and marshalling) of data and information objects for storing in the User data repository.

3.3 ISS POLICY VIEWPOINTS

3.3.1 Semantic Element

The semantic element acts as the head of the data pattern for the packaging of data and information elements for a specific recipient or set of recipients (e.g., Community of Interest (CoI)) participating in the Information Exchange Specification.

3.3.1.1 Semantic Element Structure

As illustrated in Figure 16, the primary components of a semantic pattern include:

1. **Semantic Element**: The head of the model that encloses an entire semantic pattern that when executed produces a dataset or object, and metadata object releasable to a specified recipient or group of recipients;
2. **Transactional Element**: A reusable data pattern for a segment of a semantic pattern. The transactional takes two forms:
    a. Standard Transactional Element: Defines a specific data packaging pattern (aggregation, transformation, labelling and redaction) that produces a data object and metadata object; and
    b. Wrapper Transactional Element: Consumes data from a wrapper (memory instance of user data) and makes it usable within the packaging process; and
3. **Wrapper Element**: Takes three forms:
    a. Entity Wrapper: A memory instance of user data based on a relational table or entity;
    b. Object Wrapper: A memory reference to user data based on a data object (JSON, XML, Binary); and
    c. File Wrapper: A memory reference to a user file.

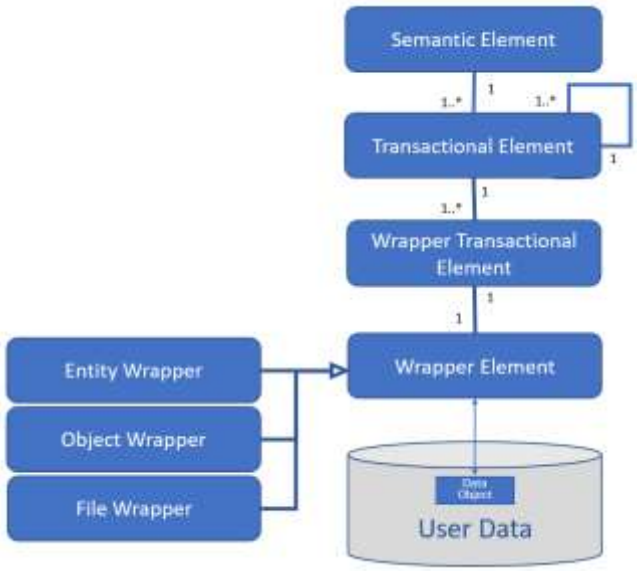

Figure 16: Semantic Element Structure

3.3.1.2 Semantic Element Example

As illustrated in Figure *17*, the semantic element (NCDF-BSO-Vessel) draws data and information elements from five (5) subtended data patterns:

1. Materiel_Item; Gathers the basic information about the vessel (e.g., identifier, name, and alias);
2. Materiel_Item_Type: Gathers descriptive information about the vessel (e.g., std type descriptor, hull number and category code);
3. Materiel_Position: Gathers the position of the vessel;
4. Materiel_Status: Gathers the vessels operational status; and
5. Materiel_Item_Hostility_Status: Gathers the vessels hostility status;

Not depicted in the model is the set of marks (or labels) that may be generated as part of the transactional processing within the data patterns (transactional element). For testing purposes, the labelling information was derived from the semantic element tags.

Alternately, labelling information can be derived using operations that use data or metadata gathered or generated as part of the data enclosed transactions. When processed, the semantic element seeks to derive an overarching set of labels for the content gathered. The semantic element labels are derived from:

1. An assessment based on user specified rules and the labels (or marks) generated during packaging;
2. If marks are not generated during packaging, use the labels contained in the metadata tagged values; or
3. If no tagged values exist, ask the user to enter the marks (impractical in most environments).

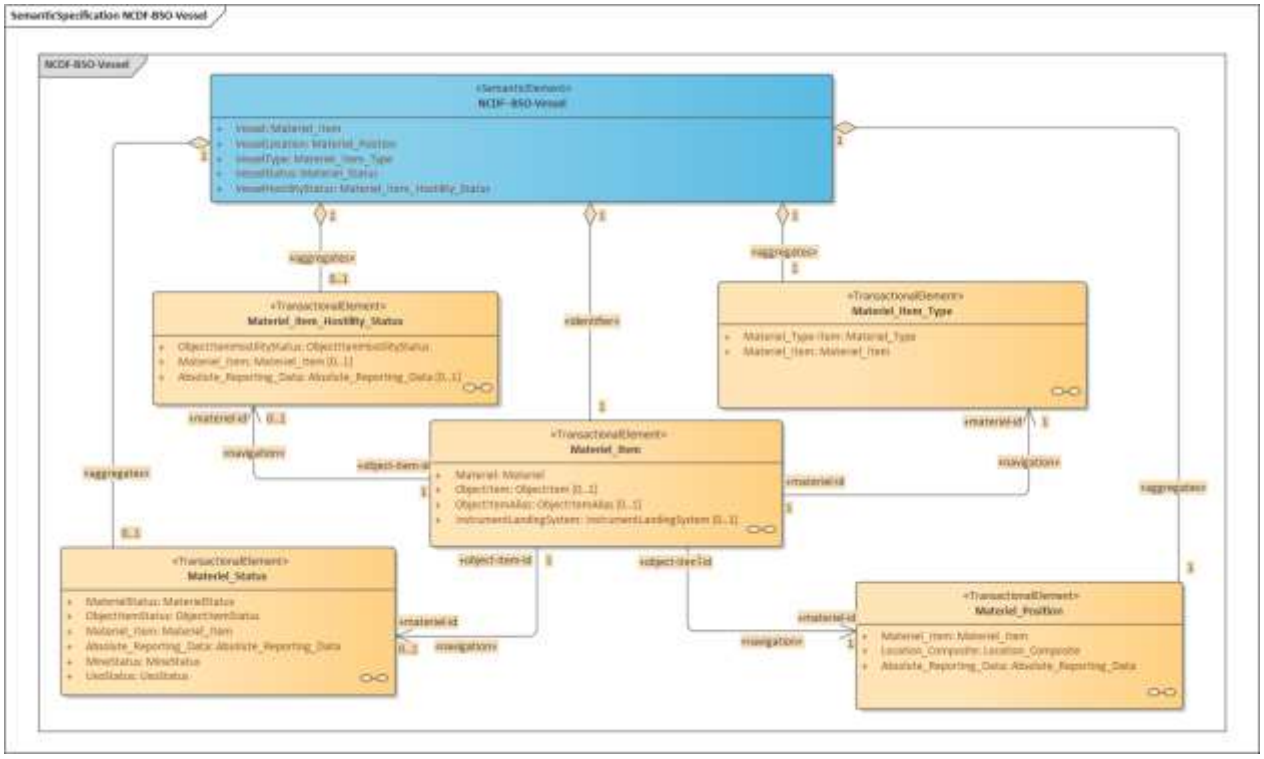

Figure 17: Semantic Element Example

3.3.1.3 Semantic Element Tags

As illustrated in Figure 18, the semantic element is configured with four (4) metadata tag values that can be set during design. These tags allow the user to establish a default set of security labels for the semantic element, including:

1. **Default Protection Level**: Identifies the assigned protection level for the content aggregated by the execution of the semantic element;
2. **Default Releasable to Caveats**: Identifies the partners the content aggregated by the execution of the semantic element can be released to;
3. **Default Security Level**: Identifies the security level of the content aggregated by the execution of the semantic element; and
4. **Default Sensitivity**: Identifies the sensitivity level assigned to the content aggregated by the execution of the semantic element.

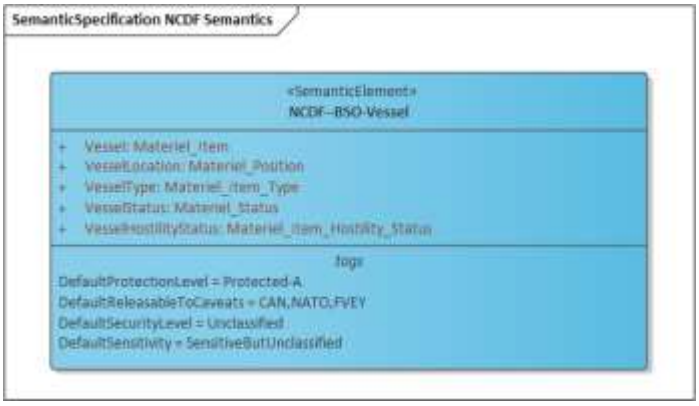

Figure 18: Semantic Element Tags

3.3.1.4 Semantic Element Execution

When triggered to build, the PPS services sequence through the subtended transactional elements starting with the "identifier". Using the navigation arc and the identifier relationship on the arcs, process each of the transactional elements, gathering the data object it produces. Once each of the transactional data and metadata objects are collected – the PPS passes the objects to the Information Exchange Controller (IEC) for formatting and release. This allocation of the formatting function to the IES allows for a common semantic to be used for multiple exchange agreements, while accommodating ISA preferred protocols.

3.3.2 Transactional Element

The transactional elements define the rules and constraints governing the aggregation, transformation, labelling and redaction of data and information elements. The transactional element packages data appropriate to the needs and authorizations of the recipients associated with that specific ISA or IES.

3.3.2.1 Transactional Element Structure

The transactional element structure is outlined as part of the semantic structure (Figure 16), standard transactional (Figure 19), and wrapper transactional (Figure 22).

ASMG added two variations to the IEPPV transactional to simplify processing. The selection of the transactional type is accomplished by setting a tagged value during design:

1. The standard transactional element which mirrors the definition in the IEPPV; and
2. The wrapper transactional that gets data from and puts data into the user specified data store.

3.3.2.2 Standard Transactional

As illustrated in Figure 19, the standard transactional element aggregates subtended objects as its internal attributes enabling the aggregation of a hierarchical set of data objects. It also has the capacity to collect a set of marks (e.g., metadata, labels or tags) that can be generated or assessed at each level of aggregation culminating in a set of metadata elements to be attached to the message being exchanged.

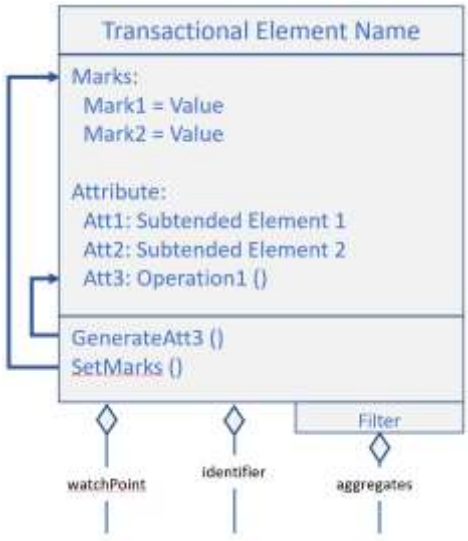

Figure 19: Standard Transactional Element

The transactional element also utilizes:

1. Operations to transform or create data or metadata (mark) elements;
2. An identifier arc to specify the identifier of keys for the data aggregation;
3. Aggregation arcs to identify objects to collect;
4. Watchpoint arcs to identify data changes to trigger the automatic release of data; and
5. Filters to redact data elements from the collection.

Both filters and transformations can be used to redact data and information elements based on the content of the subtended elements. Decisions can be made based on the values of the attributes, marks (labels) or both.

3.3.2.2.1 Standard Transactional Element Example

Figure 20 provides an example of a transactional element that gathers information about the hostility status and supporting information for a vessel. This uses the "Materiel_Item_Hostility_Status" derived from the structure "NCDF-BSO-Vessel" semantic element, Figure 17 and is one of the five transactions to be processed to complete the semantic.

As illustrated, for each transactional element, the identifier is gathered from a wrapper element. Wrapper elements are the data objects that retain their data between processing cycles. Transactional and semantic elements' memory locations are scrubbed after the semantic object is released to the IES. The semantic is rebuilt each time it is requested, assuring the most recent data is aggregated.

Only the transactional elements are aggregated into the semantic pattern. In the example, though used to provide the identifier for the build, the wrapper element ("OBJ_ITEM_HSTLY_STAT") is not aggregated into the semantic data object, only its wrapper transactional element "ObjectItemHostilityStatus". This pattern aggregates:

1. Materiel_Item (Standard Transactional Element);
2. ObjectItemHostilityStatus (Wrapper Transactional Element); and
3. Absolute_Reporting_Data (Standard Transactional Element).

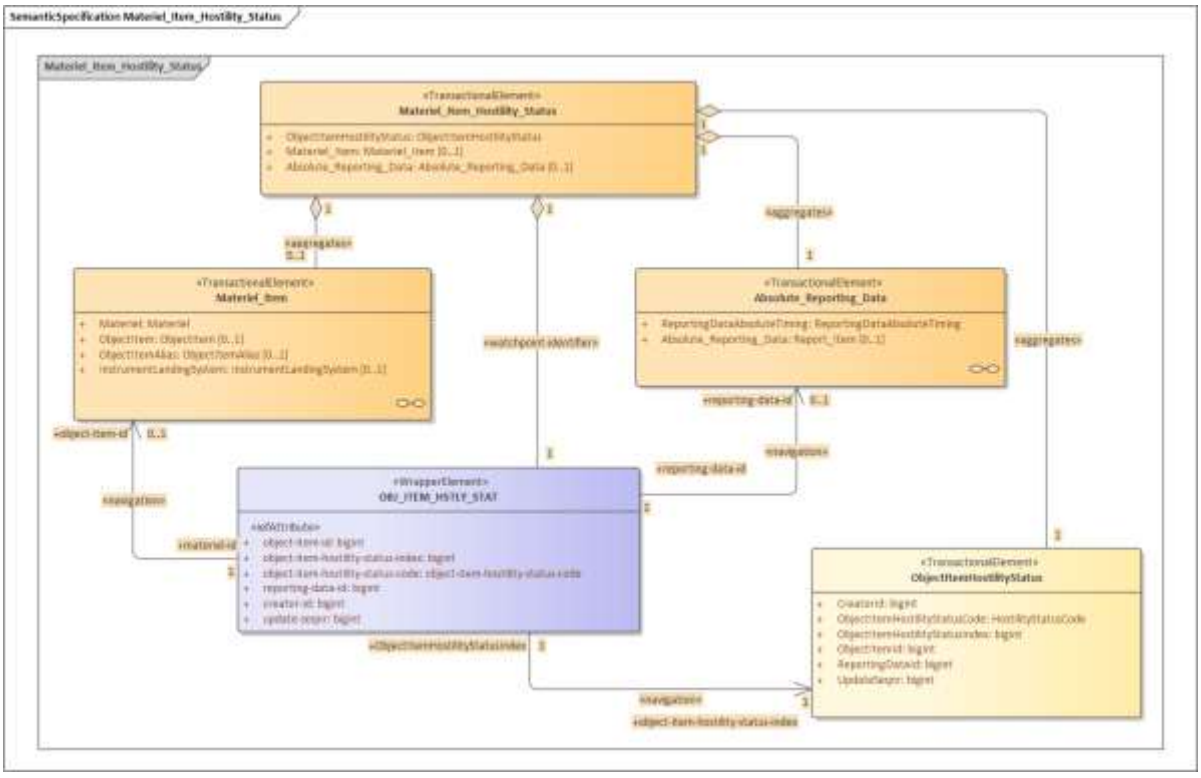

Figure 20: Standard Transactional Element Example

"ObjectItemHostilityStatus" is the wrapper transactional element associated with "OBJ_ITEM_HSTLY_STAT". The wrapper element is used to provide the unique identifier for construction of the pattern, however, it is the transactional element (or in this case the wrapper transactional element for the wrapper) that is aggregated into the pattern. This assures that any required transformations or filters are executed. See Section 3.3.2.2.3 for additional information on the role and function of the wrapper transactional element.

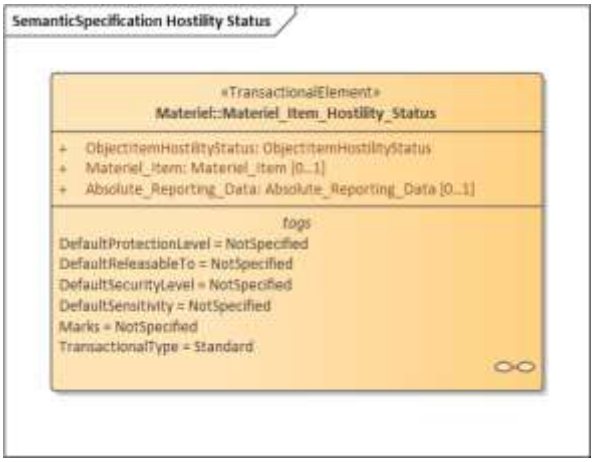

Figure 21: Transactional Element Tags

3.3.2.2.2 Standard Transactional Tags

As illustrated in Figure 18, the transactional element is configured with six (6) metadata tag values that can be set during design. These tags allow the user to establish a default set of security labels for the semantic element, including:

1. **Default Protection Level**: Identifies the assigned protection level for the content aggregated by the execution of the transactional element;
2. **Default Releasable to Caveats**: Identifies the partners the content aggregated by the execution of the transactional element can be released to;

3. **Default Security Level**: Identifies security level of the content aggregated by the execution of the transactional element;
4. **Default Sensitivity**: Identifies the sensitivity level assigned to the content aggregated by the execution of the transactional element;
5. **Marks**: Identify whether or not marks need to be generated for the transactional element; and
6. **Transactional Type**: Identifies if the transactional is a wrapper type or standard type.

3.3.2.2.3 Standard Transactional Execution

When the transactional is triggered to build, the PPS services sequence through the subtended transactional elements starting with the "identifier" wrapper element. Using the navigation arc of the identifier, each of the transactional elements is processed, gathering the data objects it produces. When complete, the PPS aggregates each of the collected data objects semantic data set and returns control to the semantic processing.

While processing the transactional element pattern, the PPS executes and enforces the data transformations (operations) and filters (context qualifiers) specified in the transactional elements.

3.3.2.2.4 Wrapper Transactional Element

The wrapper transactional element is an enhancement of the more generalized description in the IEPPV. By initiating a step of transferring the data elements into the transactional domain, the processing does not affect the integrity of the stored data. It also enables some basic processing before initiating packaging, e.g.:

1. Adopting the naming convention of the exchange semantics;
2. Redacting data precluded from release from entering the transactional domain;
3. Transforming data values to the semantics of the exchange domain; and
4. Generating metadata (marks) for tables, entities or objects that have no assigned metadata; and
5. Labelling rules are developed and executed (see SetMarks()) in Figure 22.

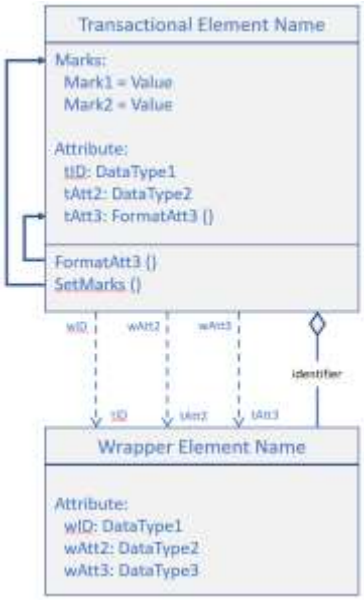

Figure 22: Wrapper Transactional Element

Where different configurations of attributes are needed by transactional builds, multiple wrapper transactional elements may be mapped to a single wrapper. This approach is often used when data attributes for large entities (or tables) need to be restructured to conform to exchange semantics.

The wrapper transactional is built each time a semantic pattern is triggered to build and is scrubbed after the semantic data and metadata objects are released to the IES Controller to be authorized, formatted and routed. This allows the Semantic Processor to maintain the integrity of wrapper data or encrypt the wrappers in memory. All operations (e.g., aggregating, transforming, restructuring, labelling and redacting) on data elements are executed in volatile memory in the transactional layer.

3.3.2.3 Wrapper Transactional Element Example

As illustrated in Figure 23, the naming convention of the wrapper transactional element differs from the wrapper element. The attributeDependency arcs are used to map between the naming conventions of the exchange and storage semantics.

In the example, only some of the attributes are mapped to the wrapper transactional element, meaning that several data elements are never aggregated into the semantic.

3.3.2.4 Wrapper Transactional Element Tags

See Section 3.3.2.2.2.

3.3.2.5 Wrapper Transactional Element Execution

When triggered, the wrapper transactional element collects the data from the wrapper element mapping the data to the attributes in its structure. The wrapper transactional element also executes any data transforms needed to convert content to the exchange semantics.

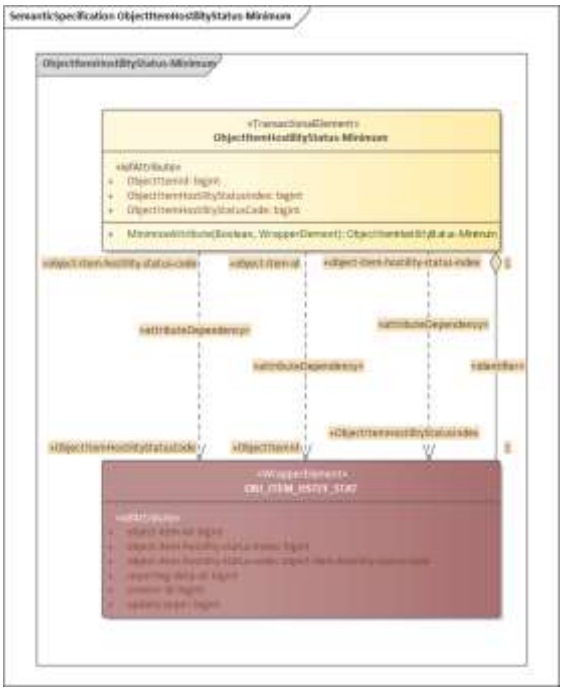

Figure 23: Wrapper Transactional Element

3.3.3 Wrapper Elements

The wrapper element is the holder of the data in the PPS environment. It is created when data is received, or by pulling data from a persistent data store. As illustrated in Figure 24 a wrapper is mapped directly to the structure in the data store it represents. In the example the wrapper represents a table or entity in the Relational Database Management System (RDBMS), though in addition to a table, it may represent:

1. An object in a NoSQL data base;
2. A file or object store in a record or document management system;
3. A file on disk; or
4. A temporary element only maintained in memory.

The wrapper represents a memory-based copy, or reference to the actual data object. Each wrapper type identifies the types of processing required to create, store (or persist) or retrieve the specific data element.

3.3.3.1 Wrapper Element Example

As illustrated in Figure 23, the wrapper element maps a set of attributes to the actual element in the RDBMS table. This allows for the mapping of received data to the RDBMS table attribute and the retrieval of the data during packaging. By convention, we map all the attributes at the wrapper element. As discussed earlier in the document, the alignment of naming conventions and/or the redaction of attributes is performed at the transactional layer.

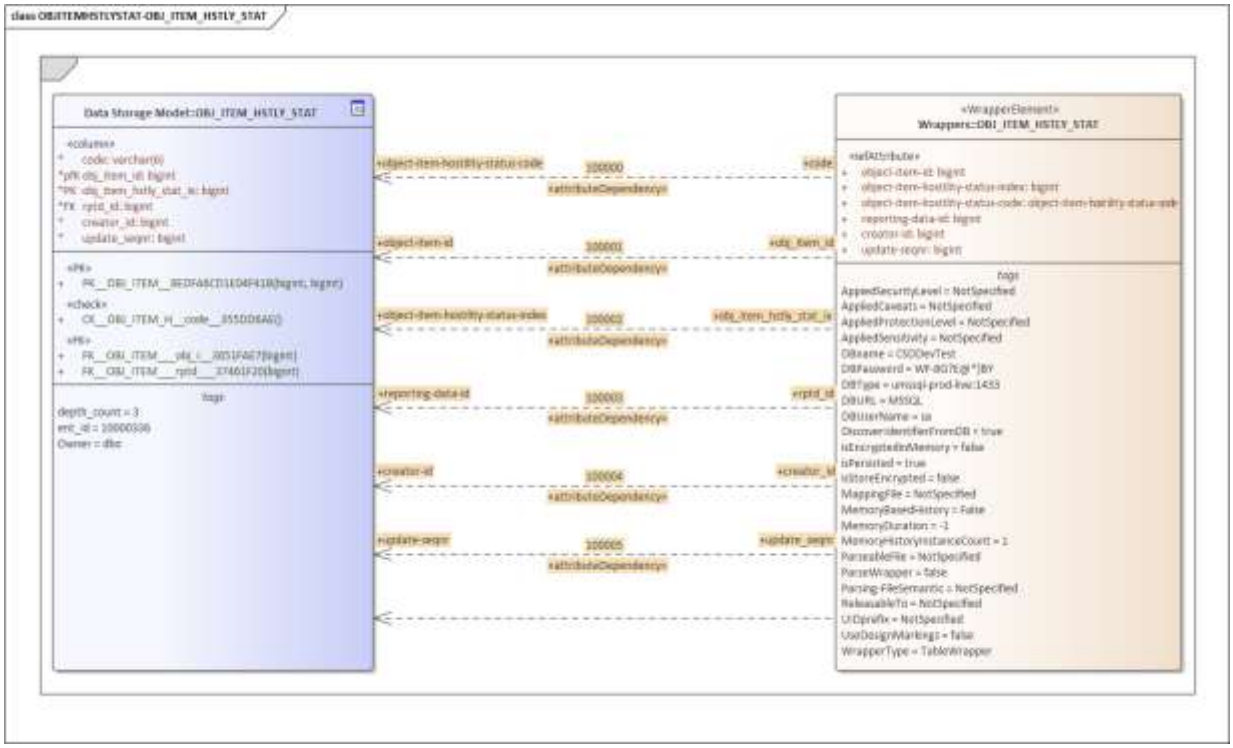

Figure 24: Wrapper Element Example with Tags

3.3.3.2 Wrapper Transactional Element Tags

See Section 3.3.2.2.2.

3.3.3.3 Wrapper Transactional Element Execution

When triggered, the wrapper element releases its data to the wrapper transactional or collects the data from the persistent store and releases it to the wrapper transactional in accordance with the mapping.

When the PPS is processing received data it parses out the data elements, creates an instance of a wrapper element to hold the data, and fills the wrapper with the data. When the wrapper element is populated, it uses its tagged values (see Section 3.3.4) to determine, e.g.:

1. If the data is to be persisted to the specified data store, or only be maintained in memory;
2. If the data in memory must be encrypted, and proceed appropriately; and/or
3. If the change in the data requires a release of data to subscribing participants.

3.3.4 Wrapper Element Tags

The ASMG implementation of the UML profile and PPS have added the following tags to the IEPPV:

1. **DBname**: Name of the database to be used to persist the wrapper;
2. **DBUserName**: User name for the specific instance of the PPS service;

3. **DBPassword**: Password for the specific instance of the PPS service;
4. **DBType**: The type of data store being used so the PPS can respect specific technology;
5. **DBURL**: Network location of the data store service;
6. **DefaultProtectionLevel**: Identifies the anticipated (analyst view) protection level of the information if this wrapper element is built. If set, this metadata value is added to the element if no metadata is added to the element during packaging. Metadata built during packaging override the values in the meta model;
7. **DefaultReleasableTo**: Comma separated list of warning orders (analyst view) for this wrapper element. If set, this metadata value is added to the element if no metadata is added to the element during packaging. Metadata built during packaging overrides the values in the meta model;
8. **DefaultSecurityLevel**: Identifies the anticipated (analyst view) security level of the information if this wrapper element is built. If set, this metadata value is added to the element if no metadata is added to the element during packaging. Metadata built during packaging overrides the values in the meta model;
9. **DefaultSensitivity**: Identifies the anticipated (analyst view) sensitivity level of the information if this wrapper element is built. If set, this metadata value is added to the element if no metadata is added to the element during packaging. Metadata built during packaging override the values in the meta model;
10. **DiscoverIdentifierFromDB**: Identifies that the PPS should retrieve the next available UID from the data store;
11. **isEncryptedinMemory**: Identifies that the wrapper data should be encrypted in memory;
12. **isPersisted**: Identifies whether or not the wrapper data should be persisted;
13. **isStoreEncrypted**: Identifies whether the wrapper data should be encrypted in the persistent store;
14. **MemoryBasedHistory**: Identifies whether the history of the wrapper data should be maintained in memory;
15. **MemoryDuration**: Identifies the amount of time the wrapper should be maintained in memory;
16. **MemoryHistoryInstanceCount**: Identifies the number of instances of a wrapper (e.g., plot points) should be maintained in memory;
17. **ParseableFile**: Identifies that the stored or persisted object is parseable;
18. **ParseWrapper**: Identifies that the persistent object must be filtered before release;
19. **ParsingFilterSemantic**: Identifies the semantic (e.g., parser, schema, filters, and mapping file) for an object referenced by the wrapper. This information will enable the PPS to parse and filter objects (e.g., attachments) in the persistent stores;
20. **UIDPrefix**: Identifies a unique UID prefix for this wrapper type;
21. **UseDesignMarkings**: Identifies that the design time marking should be integrated into instances of the wrapper object in memory; and
22. **WrapperType**: Identifies wrapper type.

3.3.5 Wrapper Element Types

The following sections describe the wrapper types supported by the ASMG implementation of the PPS and IEPPV.

3.3.5.1 Entity Wrapper Element

An entity or table wrapper element, as illustrated in Figure 24, links the ISS policy pattern to data stored in a relational database.

3.3.5.2 Object Wrapper Element

The object wrapper element, Figure 25, enables the PPS to persist and retrieve data objects and incorporate them into the set of objects being aggregated or include them as attachments to the outbound message(s). Based on policy, the PSS also has the option to process and repackage the object for reformatting or filtering for a specific recipient. The reformatting or redaction of the original object would be intended to make the object in whole or in part releasable and usable by the recipient.

3.3.5.3 File Wrapper Element

The file wrapper element, Figure 26 enables the PPS to persist and retrieve data objects and files to the local file system. The files are often exchanged as attachments to exchange messages.

3.3.5.4 Memory Only Wrapper Element

The memory-only wrapper element provides the option to only hold data elements in memory without any persistence.

Figure 25: Object Wrapper Example

Figure 26: File Wrapper Example

4. PPS DATA PROCESSING

4.1 INTRODUCTION

Upon the receipt of information (e.g., message), the PPS identifies the source of the information and the type of information (e.g., XML, JSON, other) that was received. As illustrated in Figure 27, after extracting the message content (metadata and payloads) from a received message, the Information Exchange Controller identifies the type of message received, and passes the data to the Semantic Processor:

1. Message Type: Including its language and semantic schema used to identify data elements and data groupings expected in the message;
2. Message Content: Including the payload(s) contained in the message; and
3. Message Metadata: Received in the message and that derived on receipt of the data, including:
    a. Source IES;
    b. Time received;
    c. Time released to processing; and
    d. Components released to processing.

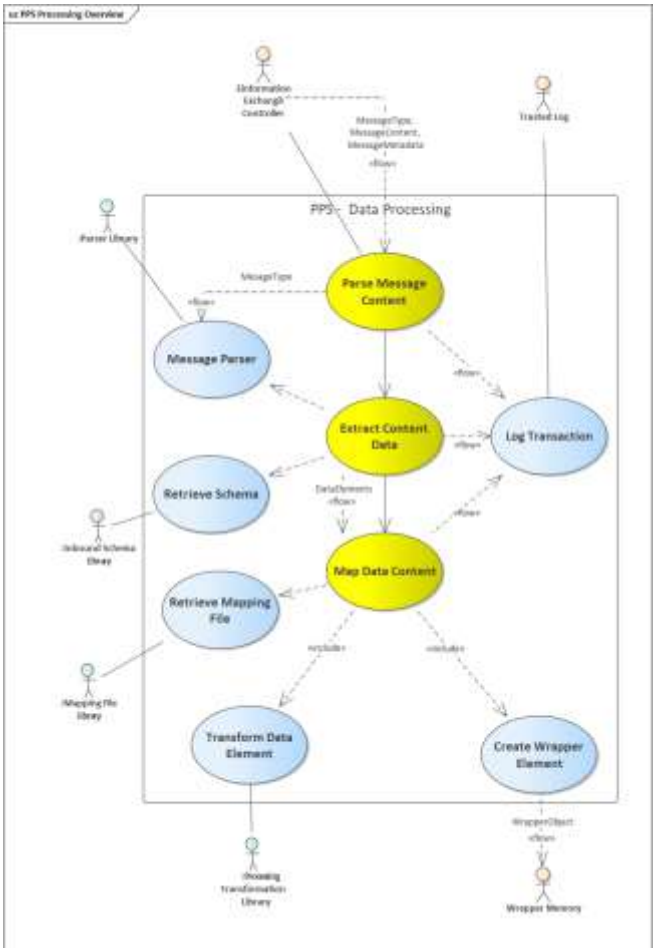

Figure 27: Processing Message Content Overview

Upon receipt of message content from the Information Exchange Controller, the Semantic Processor queues up the processing of the data. The first step is the retrieval of the appropriate parser from the parser library (e.g., GenericXMLParser) and the definition of the message's structure and syntax (e.g., XSD). At this point in the process the Semantic Processor divides the message content into its grammatical parts and the relationships between the parts. The resulting data map is mapped to the semantics of the users specified storage semantics. Each data element is mapped to the wrapper and wrapper attribute. These mappings instigate the creation of wrapper objects (memory-based collection of data elements) which are completed and placed in the PPS memory. As processing continues, the message data is transformed (as necessary) to translate (naming conventions and value types) message semantics to those used in the user domain. The process of mapping and transformation (as necessary) continues until all message data is transferred to wrapper elements.

As illustrated in Figure 27, transactions executed by the Information Exchange Controller and Semantic Processor are captured and sent to a Trusted Logging Service. This logging enables users to monitor and audit the activity of the PPS when processing inbound messages. Separating the parser library, schema libraries, and mapping files from the PPS enables the user to perform enhanced auditing of PPS activity and more rapidly address issues identified during monitoring and auditing. Updating libraries using DEVSECOPS practices further improves the flexibility, agility and adaptability of the PPS delivered ISS capability.

Notes and Considerations:

- **Data Loss Prevention**: As information is transferred between semantic domains, the semantic processing and user mapping files should address the potential for data loss. The PPS implementers should provide mechanisms in the mapping files and software services to address data loss; and
- **Unique Identification of Data**: The PPS and its policy environment are intended to receive and share data and information elements produced or owned by their internal organizations and mission partners. The PPS should be able to track data ownership through the processing and packaging of data elements, and assure that this data is only used or released in accordance with Information Sharing Agreements and users' data and information sharing policy. The PPS implementers should provide mechanisms in the mapping files and software services to address the tracking of data and information elements.

5. DATA PACKAGING

5.1 INTRODUCTION

Data Packaging can be initiated in two ways:

1. By request from an authorized[2] external user to the SDS or PPS for specific semantic elements in the ISS policy environment; or
2. Through defined watchpoint[3], events, in the policy models that trigger the release of data changes to all semantic elements impacted by the change(s) and all IESs using the semantics.

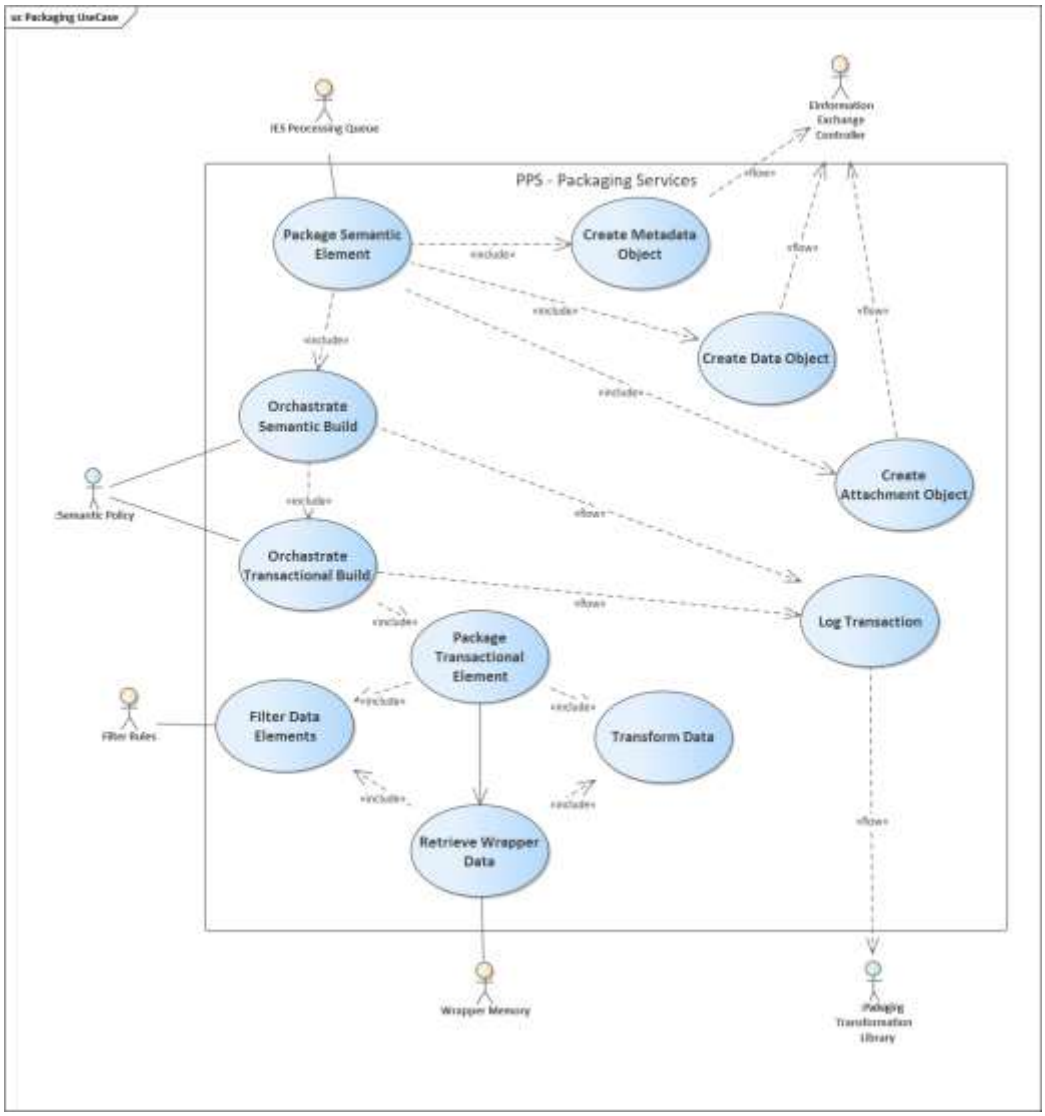

Figure 28: PPS Data Packaging

[2] Authorization is typically adjudicated by a PEP providing access control for the environment.

[3] A watchpoint is an element in a policy model that defines items of interest in a semantic pattern that require the sharing of data elements with specified partners each time new data is created or data is changed. Watchpoints enable the PPS to provide real-time event-driven global up date of available data.

5.2 AUTHORIZED USER REQUEST

An authorized user request can request information from the PPS or SDS through the governing Policy Enforcement Point (PEP) that controls access to the associated PPS or SDS depending on the solution architecture. One implementation for this type of access is illustrated in Figure 29.

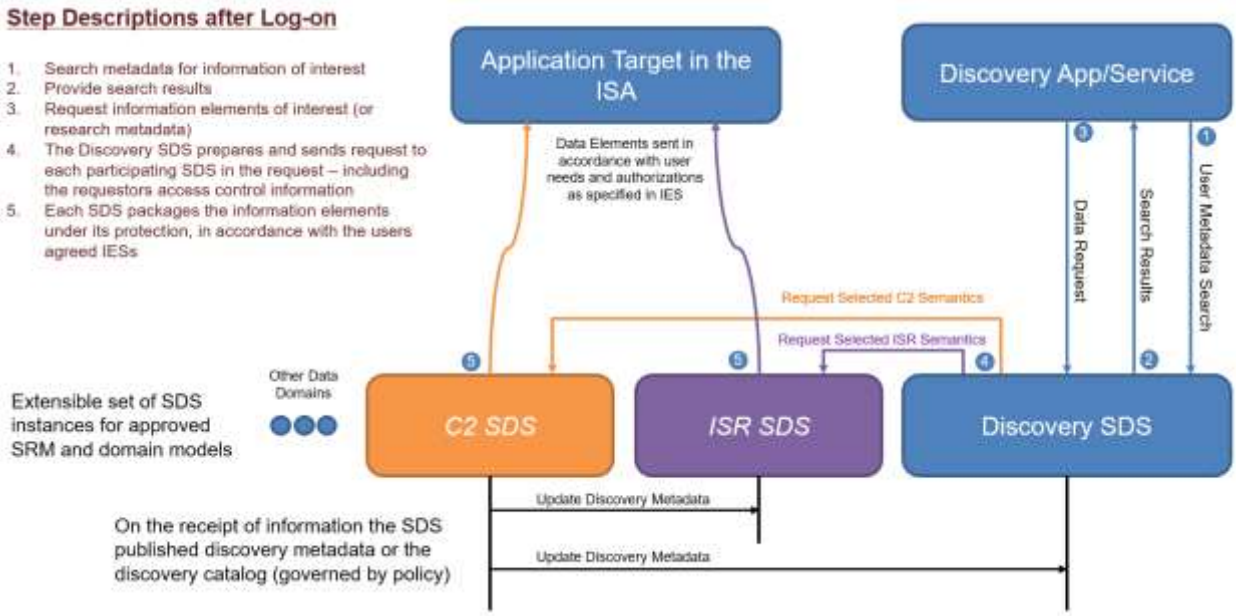

Figure 29: User Information Request

As illustrated, in this configuration we use a discovery service to discover, search for and request information in one or more of the data environments. Each request to an SDS instance is a data message containing parameters such as:

1. User information (e.g., identity);
2. IDs and semantic element reference(s) for the information being requested; and
3. IES reference(s) governing the exchange.

Upon receiving the request(s), the individual SDS verifies that the requests are allowed, based on the recipient's IES profile and authorizations. If authorized, the SDS queues the request and packages the authorized information for the user in accordance with the exchange and semantic policies. If the requestor does not have an active set of exchange policies for a given SDS, an error message will be sent to the user. The user may request changes from an authorized administrator. A new IES may be created, or an existing IES modified to accommodate the exchange.

Alternately, the requested data may be generated by any authorized user application and the target SDS will authenticate the application and user, and if authorized, release the information. Figure 29 illustrates how a federated search of a multi-SDS environment may be accommodated.

Note:

1. Each SDS is comprised of the same software services differing only in its policy environment, configuration and library disposition. The later elements are incorporated at runtime, so the SDS can be deployed as an infrastructural component, instantiated as many times as needed, and configured to meet mission parameters and data domains;
2. Many of the SDS features can be updated or modified by an authorized administrator to adapt the service to changing missions;
3. As policies, configurations and library elements can be treated as data, these elements may be recorded (stored) at any point, and reused in future missions, or as an input to an audit process; and
4. Metadata updates are shared in accordance with policy, treating the receiving discovery or user system as any other recipient and using metadata as the data element of exchange.

5.3 EVENT DRIVEN UPDATE

Setting up an automatic updating agreement involves two aspects in the policy environment:

1. Identifying watchpoints (data changes that trigger the release of data to subscribing recipients);
2. Including the watchpoint element in one or more semantic patterns (policy definitions); and
3. Including the semantic element (pattern) as part of one or more IESs.

Upon the receipt of data, each wrapper element identified as a watchpoint identifies the semantic patterns it participates in, and passes them to the packaging queue. The queue manager will eliminate duplications and identify which of the unique semantic elements are assigned to an active IES (discarding those that are not). The resulting queue is sent for packaging. Each semantic is packaged in accordance with its IES characteristics and released (routed) to authorized recipient(s).

These features enable the PPS and SDS to deliver event driven global updates to all users participating in the active IESs.

6. CONCLUSION

6.1 SUMMARY

The OMG Information Exchange Packaging Policy Vocabulary (IEPPV) provides a flexible, agile and adaptive approach to defining, deploying and sustaining information sharing and safeguarding operations at any scale, within and across data domains and organizations. Its architecture-based approach enables users to:

1. Define a common vocabulary for defining the policies (rules and constraints) governing the transition of data (including metadata) and information elements between exchange semantics and storage semantics:
    a. **Information Exchange Specifications (IES)**: Grouping of messages or filtered semantic elements for a given IES and includes:
        i. **Distribution Specification (DS)**: Defines the configuration of the distribution services to be used for the IES; and
        ii. **Information Specification (IS)**:
            a) Filtered Semantic Elements: Defines the release control filters on a semantic element for the specific IES; or
            b) Messages: Comprised of multiple filtered semantic elements;
    b. **Filtered Semantic Element (FSE)**: References the semantic element (SE) and binds that SE with the release filters to a specific IES and includes:
        i. **Dynamic Filters (DF)**: Links a release filter to the attributes that can be used in defining the filter; and
        ii. **Filtered Transactional Elements (FTE)**: References a transactional element and identifies the attributes within that transactional element used to define the filter;
    c. **Semantic Element (SE)**: Identifies the set of transactional elements (pattern) for the packaging of data and information elements that is releasable to a specified recipient (e.g., individual, community of interest, organization, system, application, node, or device) under an IES. The SE comprises a set of transactional elements (see below) with one set as identifier;
    d. **Transactional Element (TE)**: Defines a build pattern for a data object to be integrated into the semantic object being packaged and includes:
        i. **Identifier**: Identifies the wrapper holding the unique identifier for the elements in the transactional patterns;
        ii. **Watchpoints**: Identifies wrappers that trigger a semantic build if the data is changed or a new wrapper element is created;
        iii. **Navigations**: Identify how subtended elements are references by the identifier when packaging the data patterns;
        iv. **Data filters**: Used to remove or redact data elements (attributes) and/or data objects (branches) from the aggregated data object; and
        v. **Transformations**: May be used to filter data from the aggregations, generate data or metadata elements (e.g., marks /labels), or transform data structure or syntax;
    e. **Wrapper Element (WE)**: Links the semantic pattern (or SE) to the data structures (data and metadata) in the storage technology. The WE is the only element that persists data in memory during operations;

2. Define sharing and safeguarding policies in standards-based architecture views:
    a. Operational Resource Flow Descriptions / Node Interactions (e.g., DODAF OV2 or NATO NOV2) or any Architecture Framework (AF) construct used to define relationships between participants (e.g., individual, community of interest, organization, system, application, node, or device);
    b. IES Views, including:
        i. Participation: Links participants to IESs or IESs to participants as an alternative to an architecture framework (AF) provided view, if a user is not vested in a specific AF;
        ii. IES: Linking an exchange to participants' technologies:
            a) Information Specification: Links the IES to its information elements and release (filter) policies; and
            b) Distribution Specification: Links the IES to its exchange technology; and
    c. Semantic Views: Define the packaging and processing patterns that map the exchange to the storage semantics; and
3. Align IEPPV views to other AF views describing:
    a. Data and information (exchange and storage);
    b. Interfaces or APIs;
    c. Applications or services;
    d. Systems;
    e. Missions or Operations; and/or
    f. Strategic concepts or capabilities.

The IEPPV enables the user to align and integrate sharing and safeguarding policy (rules and constraints) in clear reusable architecture views, filling a gap in most traditional architecture frameworks and the complexity of ISS environments. The architecture elements provide users a rich vocabulary for processing and packaging data and metadata elements and information elements to selectively share data and information content with recipients in accordance with their needs and authorizations.

6.2 BENEFITS OF THE IEF/IEPPV APPROACH

As illustrated in many of the examples used in this document, the IEPPV provides a UML profile that enables users to model ISS policy. Modelling ISS policy provides many benefits, including:

1. **Retention of Institutional Memory**: When modelled in a modelling tool, the artefacts provide persistent documentation for the policies (rules and constraints) governing access, use and release of data for a specific data store or for multiple data stores across the enterprise or mission;
2. **Data Centric Security (DCS)**: Binds ISS policies (rules and constraints) to the data stores and exchanges charged with enforcing them;
3. **Traceability**: Modelling the ISS policy models in a UML tools enables users to develop and maintain traceability from requirements documents (e.g., legislations, regulation, memorandum of understanding, information sharing agreements and operating procedures) with architecture, design and implementation elements;

4. **Enterprise Alignment**: The IEPPV models can be aligned to strategic, capability, operational, system, service and other views and viewpoints in architecture. This will enable auditing ISS from architecture through operations. Also provides the opportunity to develop auditing tools to improve IM, DM and ISS governance;
5. **Design Automation**: The metadata underpinning the IEPPV models can be used by Model Driven Architecture (MBA) and Model Based Systems Engineering (MBSE) tools to automate the transformation of architecture, to design, to implementations. This would reduce transformation error, increasing reuse and reducing development cost and risk. It will also assist in reducing overall ISS lifecycle cost and risk;
6. **Flexibility, Agility and Adaptability**: If integrated with modelling tools, the IEPPV (specifically the UML profile) enables the separation of the ISS policy lifecycle from that of the services that enforce them (e.g., PPS). This places the responsibility for developing ISS policies on operational, information and security architects and analysts who have an operational verses technical focus on ISS. Automation, as mentioned in item 5, can dramatically streamline the implementation, test, certification and deployment options vis-à-vis a DEVSECOPS process for policy. Being able to deliver policy like data, further expedites the deployment of ISS policy to operations; adapting capability at an operational tempo;
7. **Management and Administration**:
    a. ISS policy represents an architectural artefact (regenerated as needed) that can be managed as models, or as artefacts catalogued by mission, phase, partners, or their attributes recalled and used to deliver a Day-0 or interim capability; and
    b. During operations, ISS policy can be shared as a data artefact using standard DCS protocols, or administered in real-time, by an authorized operator, who can activate, deactivate, or extend, and/or modify many aspects of a PPS services' policy environment. This provides much increased flexibility, agility and adaptability of ISS capability and their ability to address change in the operational environment;
8. **Monitoring and Auditing**: The IEPPV models, artefacts and metadata form a foundation for evaluating mission transactions against a known baseline with its logged operations. This data foundation can form input to analytic tools for assessing ISS designs, or ISS missions;
9. **ISS Risk Mitigation**: Application Program Interface (API) development and maintenance represents a high risk and high-cost element in the system and/or software lifecycles. APIs typically become rigid and brittle over time and cease to address evolving mission or operational requirements. Few organizations have the ability or capacity to understand how data and information flows through their organizations, systems or applications, placing them in a high risk of sensitive data loss.

    The IEF and IEPPV provide a systematic, yet agile way, for organizations to engineer ISS capabilities that are flexible, agile, adaptive, secure and auditable. The iterative approach to the development and deployment of policy means that not all ISS requirements need to be documented before starting. Organizations can selectively design, test and deploy capability as ISS rules and constraints are discovered; and
10. **Day-0 Capability**: The ability to capture and store mission configurations means that these artefacts can be reused as a starting point for an unplanned mission, using the real-time administration capability of the PPS to continually evolve mission policies and capability, and address changes in mission requirements. Alternately, captured mission policies and configurations can be replayed in

desktop exercises and policies modified, based on newly discovered ISS requirements. The resulting policies can be captured and stored for future training or operational requirements. Libraries of mission policy sets and configurations, developed or captured over time form an ever evolving and expanding ISS capability.

ANNEX A

The following definitions are used within the SDS OCD.

Agile Development	Practice approach discovering requirements and developing solutions through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s). It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change.
Application Program Interface	Definition of the rules, constraints and configuration governing interaction with the host application.
Data	Facts (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation.
Data as a Service	Information provision and distribution model in which data is made available to consumers over a network environment.
Data-Centric	The adjudication and enforcement of information sharing and guarding policies (rules and constraints) governing individual data and information elements.
Data Lake	System or repository of structured, semi-structured or unstructured data stored in its natural or raw format using a flat architecture (a data warehouse is a repository for structured, filtered data that has been processed for a specific purpose).
Day-0 Capability	A set of services and/or resources that can be employed to address or mitigate an incident, event or vulnerability on the day of discovery.
DevOps	Practice that combines software development (Dev) and IT operations (Ops). It aims to shorten the system development life cycle and provide continuous delivery with high software quality.
DevSecOps	Integration of security evaluation and testing at every phase of the software lifecycle, from initial design through integration, testing, deployment, delivery and maintenance.
Forensic Auditing	Ability to analyse the architectures, designs and/or operational logs to verify that components are operating properly, and effectively enforcing information sharing and safeguarding policies appropriately.
Identity, Credential and Access Management	Service to control access and release of information based on individual user authorisation and need to know.

Information	(1) Data in context; and (2) Data in a form that informs a decision.
Information Exchange Specification	Exchange specification between two or more parties specifying how information is to be shared between each party (equivalent to the Information Sharing Agreement used by the US).
Information Exchange Framework Reference Architecture	An OMG sponsored open reference architecture for information sharing and safeguarding, employing data centric security principles.
Information Exchange Packaging Policy Vocabulary	Vocabulary that will provide consistent concepts for the expression of rules governing information packaging and processing.
Information Sharing Agreement	Exchange agreement between two or more parties specifying how information is to be shared between each party.
Information Sharing and Safeguarding (ISS)	A set of capabilities that provide users with the ability to responsibly share information based on user needs, user authorizations and data sensitivity.
Intelligence	(1) Understanding / comprehension of the available information; (2) Insight into the current situation; and (3) Assessment of future events or situations.
Memorandum of Understanding	Statement defining the specific criteria that forms the basis of the understanding between parties.
Model Based Systems Engineering	Systems engineering methodology that focuses on creating and exploiting domain models as the primary means of information exchange between engineers, rather than on document-based information exchange.
Model Driven Architecture	Software design approach for the development of software systems providing a set of guidelines for the structuring of specifications which are expressed as models.

Operational Concept Document	Discussion paper describing the technical or operational need being addressed and the goals, objectives, features and functions of a proposed solution to address that need, along with an assessment of impact on user environment and operational use of the proposed solution.
Operational View-2	Applying the context of the operational capability to a community of anticipated users with the primary purpose of defining capability requirements within an operational context.
Packaging and Processing Service	Transition structured information elements between data stores and information exchange services in accordance with local information sharing and safeguarding policies.
Policy Administration Point	Provides an authorised user with an interface to access services needed to manage and administer the configuration and policy environments of IEF components.
Policy Decision Point	Adjudicates access to, or the release of resources to a specified user based on resource sensitivity, user privilege and operational context in which the decision is being made.
Policy Enforcement Point	An integration point between the User's infrastructure and the SDS service which enables the user to integrate access controls to the receipt and release of messages.
Policy Driven	The adjudication and enforcement of rules and constraints derived from, and traceable to, user or community approved policy instruments (e.g., legislation, international agreements, regulations, directives, information sharing agreements, operating policy and operating procedures).
Publish/Subscribe	Architectural design pattern that provides a framework for exchanging messages between publishers and subscribers. This pattern involves the publisher and subscriber relying on a message broker that relays messages from the publisher to the subscribers. The host (publisher) publishes messages to a channel that subscribers can then sign up to.
Request/Response	Message exchange pattern that generates a suitable response against a correctly prepared request.
Scaled Agile	A set of organization and workflow patterns intended to guide enterprises in scaling lean and agile practices to plan, prioritize and manage capability development. Scaled Agile enables an enterprise to expand Agile development practices beyond the application development process.

Security Services Gateway	Provides a secure access to the user specified security services.
Semantic Reference Model	A database model describing the structured entities found within the model and all the relationships that exist between them.
Software as a Service	Software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted.

ANNEX B

The following acronyms are used within the SDS OCD.

Acronym	Definition
AD-C4I	All Domain Consultation, Command, Control, Communications and Intelligence
AF	Architecture Framework
AI	Artificial Intelligence
API	Application Program Interface
ASMG	Advanced Systems Management Group Ltd.
C2I	Command, Control and Intelligence
C4I	Command, Control, Communications, Computers and Intelligence
CFI	Connected Forces Initiative
CIS	Communication and Information System
CoI	Community of Interest
CORBA	Common Object Request Broker Architecture
CTS	Cryptographic Transformation Service
CWIX	Coalition Warrior Interoperability Exercise
DaaS	Data as a Service
DataOps	Data Operations
Day-0	Day Zero
DCS	Data Centric Security
DDS	Data Distribution Service
Dev	Development
DevSecOps	Development, Security and Operations
DMN	Decision Modelling Notation
DODAF	Department of Defense Architecture Framework
DS	Distribution Specification
DTL	Data Transformation Library
EA	Enterprise Architecture
eISA	Electronic Information Sharing Agreement

ESB	Enterprise Service Bus
FMN	Federated Mission Networking
HQ	Headquarters
ICAM	Identity, Credential and Access Management
IDL	Interface Definition Language
IEC	Information Exchange Controller
IEF	Information Exchange Framework
IEF-RA	Information Exchange Framework Reference Architecture
IER	Information Exchange Requirement
IES	Information Exchange Specification
IEPPV	Information Exchange Packaging Policy Vocabulary
IM	Information Management
IS	Information Specification
ISA	Information Sharing Agreement
ISS	Information Sharing and Safeguarding
IT	Information Technology
JC3IEDM	Joint Consultation, Command and Control Information Exchange Data Model
JSON	JavaScript Object Notation
MBSE	Model Based System Engineering
MDA	Model Driven Architecture
MIM	MIP Information Model
MIP	Multilateral Interoperability Programme
MODAF	Ministry of Defence Architecture Framework
MOU	Memorandum of Understanding
MSDM	Message Schema and Data Mapping
NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organisation
NCDF	NATO Core Data Framework
NGO	Non-Government Organisation
NoSQL	Not Only Structured Query Language

NOV	NATO Operational View
NNEC	NATO Network Enabled Capability
NVG	NATO Vector Graphics
O&M	Operations and Maintenance
OCD	Operational Concept Document
OGD	Other Government Department
OMG	Object Management Group
OODB	Object Oriented DataBase
Ops	Operations
OS	Operating System
OV-2	Operational View 2 – Operational Resource Flow Description
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PPS	Packaging and Processing Service
QA	Quality Assurance
RDBMS	Relational DataBase Management System
REST	Representational State Transfer
S2S	System to System
SDS	Secure Data Service
Sec	Security
SDLC	Software Development Life Cycle
SME	Subject Matter Expert
SOAP	Simple Object Access Protocol
SOPES	Shared Operational Picture Exchange Services
SOS	Secure Operating System
SRM	Semantic Reference Model
SSG	Security Services Gateway
STANAG	Standard NATO Agreement
STF	Standards Transformation Framework

TLS	Trusted Logging Service
UAF	Unified Architecture Framework
UML	Unified Modelling Language
UPDM	Unified Profile for DODAF and MODAF
US	United States of America
VM	Virtual Machine
WSMP	Web Service Messaging Profile
XML	Extensible Markup Language