MULTILATERAL INTEROPERABILITY PROGRAMME (MIP)

MIP4 Information Exchange Specification (MIP4-IES)

Exchange Mechanism Overview

09 October 2020

DOCUMENT CONTROL

Change Management

Change Control Authority	MIP Integrated Product Team 4 (IPT4) Change Manager, webmaster@mip-interop.org
Release Authority	MIP Program Management Group (PMG), pmgchair@mip-interop.org
Latest Public Release	www.mip-interop.org | Public Document Library | 07-MIP4-IES
Latest MIP Release	www.mip-interop.org | Members Only site | CM Portal | MIP4 (account required)
Current Working Draft (for comments)	MIP4-IES Collaborative Environment (account required)

Version History

Version	Author	Date	Reason for Change
1.5.2	IPT4 CM	12 Sep 2017	MIP4.0 official release [ref MSG-DL-26 at MSG20; July 2017]. Details of internal versions removed from Version History.
1.8.0	IPT4 EMT	19 Apr 2018	MIP4.1 official release. Details of internal versions removed from Version History.
1.9.1	IPT4 CM	27 Aug 2019	MIP4.2 official release. Details of internal versions removed from Version History.
1.12.1	IPT4 CM	09 Oct 2020	MIP4.3 official release. Details of internal versions removed from Version History.

Table of Contents

1 Overview **5**

2 Scope **6**

2.1 Note on non-normative examples 6

3 References **7**

3.1 Exchange Mechanism References 7

3.2 MIP4-IES References 8

3.3 MIP4-IES Exchange Mechanism Artifacts 8

4 Namespaces **9**

5 Information Exchange **13**

5.1 Overview 13

5.2 Use Cases 15

5.2.1 System Initialization 15

5.2.2 Situation recovery 15

5.2.3 Situation Low-frequency Refresh 16

5.2.4 System restart 16

5.2.5 Per object verification, inspection or elaboration 16

5.2.6 Offline Archive of the Situation 16

5.2.7 Consumption of the Situation 16

5.3 Roles 17

5.4 Initiating Information Exchange 17

5.5 MIP4-IES Request/Response Exchange Pattern 17

5.6 MIP4-IES Publish/Subscribe Exchange Pattern 18

5.7 MIP4-IES File Exchange Pattern 18

5.8 Error Handling 18

5.9 Specification Principles 19

6 Cross-pattern Requirements 20

6.1 Service Metadata 20

6.1.1 Endpoint References and Resource Locations. 20

6.2 Message Container 21

6.3 WS-I Profiles and Bindings 21

6.4 Filtering Data 22

6.4.1 General 22

6.4.2 Topics and Topic filtering 23

6.4.2.1 Usage of Topics 25

6.4.2.2 Usage of Topic Expressions 25

6.5 Unexpected content tolerance 26

6.6 Unnecessary Notifications and Large Payloads 26

6.7 Creation, Update and Removal 26

7 Glossary 27

1 Overview

The MIP4 approach to information exchange classifies and defines several exchange patterns based on the most appropriate open and standard technologies, according to exchange pattern-technology mapping rules documented as MIP4 design principles, rules and decisions. Furthermore, the information exchange patterns are analysed and detailed at two levels of abstraction: technology independent and technology specific. At the first level, the analysis aims at classifying and properly describing the patterns in order to support the decision making process required to specify which pattern is used for which information exchange. At the second level, each “technology independent pattern” is then mapped to the most applicable/ suitable standard technology, based on an assessment of the pros and cons for each option.

2 Scope

The scope of this document includes a holistic technical discussion of the MIP4 information exchange patterns, principles and rules that govern MIP4 information exchange across C2 systems. The discussion is presented at a relatively high level and is intended to give an overview prior to proceeding to the detailed implementation specifications.

This document also presents business rules and guidance that should be followed to achieve effective information exchange.

2.1 Note on non-normative examples

The examples in this document are non-normative and they are intended to illustrate the purpose of the concept or the functionality being described. The normative technical details in the document will take precedence over these non-normative examples. The actual implementation shall be based on the normative technical details and technical artifacts.

3 References

3.1 Exchange Mechanism References

Throughout the MIP4-IES suite of documents (particularly the portion of the specification that addresses the Exchange Mechanism patterns), references will be made to external artifacts. The following list provides the authoritative list of these sources, including the referenced version, and applies to all documents that comprise the MIP4-IES.

N°	Title	Version
REF-EM-01	Universal Resource Identifier (URI), http://www.ietf.org/rfc/rfc3986.txt	January 2005
REF-EM-02	OASIS WS-BaseNotification 1.3, OASIS Standard, 1 October 2006	1.3
REF-EM-03	TIDE Transformational Baseline[1]	V4.0
REF-EM-04	OASIS WS-BrokeredNotification, OASIS Standard, 1 October 2006	1.3
REF-EM-05	OMG Data Distribution Service (DDS), April 2015	1.4
REF-EM-06	OASIS WS-Topics, OASIS Standard, 1 October 2006	1.3
REF-EM-07	Eugster, P. Th., Felber, Pascal A., Guerraoui, R. 2003. The Many Faces of Publish/Subscribe. ACM Comput. Surv. 35, 2 (June 2003), 114-131.	June 2003
REF-EM-08	OASIS WS-ResourceProperties, OASIS Standard, 1 April 2006	1.2
REF-EM-09	TIDE Transformational Baseline, Request-Response SOAP Profile	4.0
REF-EM-10	TIDE Transformational Baseline, Publish-Subscribe SOAP Profile	4.0
REF-EM-11	NATO Web Service Messaging Profile (WSMP) Core Specification	STANAG 5644
REF-EM-12	Open GIS KML	2.2

[1] https://tide.act.nato.int/tidepedia/index.php/TIDE_Transformational_Baseline_v4.0

3.2 MIP4-IES References

For any references in this document to a MIP4-IES source (REF-MIP-##), refer to section 2.3 in the *MIP4-IES Overview* [REF-MIP-01].

3.3 MIP4-IES Exchange Mechanism Artifacts

Included with the *MIP4-IES Exchange Mechanism Overview* is a compressed archive containing *MIP4-IES XML Artifacts* [REF-MIP-06]. This file contains the following artifacts, which may be referenced individually in this document [REF-MIP-02] or any of the three Exchange Pattern specifications (REF-MIP-03, REF-MIP-04 and REF-MIP-05).

N°	**Title**	**Version**
ART-EM-01	Common.xsd	v1.6
ART-EM-03	Filter.xsd	v1.6
ART-EM-07	Base.xsd[2]	v1.4
ART-EM-08	porttypes.wsdl	STANAG 5644
ART-EM-09	WSMP-FaultMessages.xsd	STANAG 5644
ART-EM-10	WSMP-ResourceMessages.xsd	STANAG 5644
ART-EM-11	WSMP-Common.xsd	STANAG 5644

[2] Note: this artifact represents a special case, in that the schema is used by both the MIP4-IES Exchange Mechanism specification and the Information Definition schemas.

4 Namespaces

The following namespaces are used in this document:

Prefix	Namespace	Description
mip4-f	https://mip-interop.org/exchange/v4.3/Filter	MIP4-IES Filtering Profile
mip4-c	https://mip-interop.org/exchange/v4.3/Common	MIP4-IES Common
wsmp-s	urn:nato:stanag:5644:wsmp:1:3:soap:services	Web Services Messaging Profile Services
wsmp	urn:nato:stanag:5644:wsmp:1:3	Web Services Messaging Profile
soap	http://www.w3.org/2003/05/soap-envelope	Web Services Base Notification
wsa	http://www.w3.org/2005/08/addressing	Web Services Addressing
wsam	http://www.w3.org/2007/05/addressing/metadata	Web Services Addressing Metadata
wsaw	http://www.w3.org/2006/05/addressing/wsdl	Web Services Addressing WSDL 1.1 binding
wsn-b	http://docs.oasis-open.org/wsn/b-2	Web Services Base Notification
wsrf-r	http://docs.oasis-open.org/wsrf/r-2	Web Services Resource Framework

wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2	Web Services Resource Framework
wsrf-rpw	http://docs.oasis-open.org/wsrf/rpw-2	Web Services Resource Framework
wstop	http://docs.oasis-open.org/wsn/t-1	Web Services Topics
kml	http://www.opengis.net/kml/2.2	KML

The following table maps the namespaces to the location of the miscellaneous external artifacts that are used by the MIP4-IES Exchange Mechanism:

Prefix	Namespace	Type	Description	Location
soap12	http://www.w3.org/2003/05/soap-envelope	XSD	SOAP 1.2	http://www.w3.org/2003/05/soap-envelope
soapenc	http://www.w3.org/2003/05/soap-encoding	XSD	SOAP Version 1.2 Part 2	http://www.w3.org/2003/05/soap-encoding
wsa	http://www.w3.org/2005/08/addressing	XSD	Web Service Addressing	http://www.w3.org/2006/03/addressing/ws-addr.xsd
wsaw	http://www.w3.org/2006/02/addressing/wsdl	XSD	Web Service Addressing (WSDL binding)	http://www.w3.org/2006/02/addressing/wsdl/ws-addr-wsdl.xsd
wsdl	http://schemas.xmlsoap.org/wsdl/	XSD	Web Service Description Language 1.1	http://schemas.xmlsoap.org/wsdl/2004-08-24.xsd
wsmp	urn:nato:stanag:5644:wsmp:1:3	XSD	Web Service Messaging Protocol STANAG 5644	https://nmrr.nc3a.nato.int/rest/doc/NATO/DM/WSMP/1.2/WSMP-Common.xsd
wsmp-f	urn:nato:stanag:	XSD	Web Service	https://nmrr.nc3a.nato.int/re

	5644:wsmp:1:3:fault		Messaging Protocol STANAG 5644	st/doc/NATO/DM/WSMP/1.2/WSMP-FaultMessages.xsd
wsmp-r	urn:nato:stanag:5644:wsmp:1:3:resources	XSD	Web Service Messaging Protocol STANAG 5644	https://nmrr.nc3a.nato.int/rest/doc/NATO/DM/WSMP/1.2/WSMP-ResourceMessages.xsd
wsmp-m	urn:nato:stanag:5644:wsmp:1:3:notificationresourcemessages	XSD	Web Service Messaging Protocol STANAG 5644	https://nmrr.nc3a.nato.int/rest/doc/NATO/DM/WSMP/1.2/WS-NResourceMessages.xsd
wsmp-s	urn:nato:stanag:5644:wsmp:1:3:soap:services	WSDL	Web Service Messaging Protocol STANAG 5644	https://nmrr.nc3a.nato.int/rest/doc/NATO/DM/WSMP/1.2/porttypes.wsdl
wsn-b	http://docs.oasis-open.org/wsn/b-2	XSD	WS-BaseNotification 1.3	http://docs.oasis-open.org/wsn/b-2.xsd
wsn-bw	http://docs.oasis-open.org/wsn/bw-2	WSDL	WS-BaseNotification 1.3	http://docs.oasis-open.org/wsn/bw-2.xsd
wsn-t	http://docs.oasis-open.org/wsn/t-1	XSD	WS-Topics 1.3	http://docs.oasis-open.org/wsn/t-1.xsd
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2	XSD	WS-BaseFaults 1.2	http://docs.oasis-open.org/wsrf/bf-2.xsd
wsrf-r	http://docs.oasis-open.org/wsrf/r-2	XSD	WS-ResourceFramework 1.2	http://docs.oasis-open.org/wsrf/r-2.xsd
wsrf-rw	http://docs.oasis-open.org/wsrf/rw-2	WSDL	WS-ResourceFramework 1.2	http://docs.oasis-open.org/wsrf/rw-2.xsd
wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2	XSD	WS-ResourceProperties 1.2	http://docs.oasis-open.org/wsrf/rp-2.xsd

wsrf-rpw	http://docs.oasis-open.org/wsrf/rpw-2	WSDL	WS-ResourceProperties 1.2	http://docs.oasis-open.org/wsrf/rpw-2.xsd
wsoap12	http://schemas.xmlsoap.org/wsdl/soap12/	XSD	SOAP 1.2 (WSDL binding)	http://schemas.xmlsoap.org/wsdl/soap12/wsdl11soap12-20060302.xsd
xml	http://www.w3.org/XML/1998/namespace	XSD	Namespaces in XML	http://www.w3.org/2009/01/xml.xsd
xsd	http://www.w3.org/2001/XMLSchema	XSD	XML Schema 1.0	Embedded in specifications
				Part1: http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/
				Part2: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/
xsi	http://www.w3.org/2001/XMLSchema-instance	XSD	XML Schema Instance 1.0	http://www.w3.org/2001/XMLSchema-instance
kml	http://www.opengis.net/kml/2.2	XSD	KML 2.2	http://www.opengis.net/kml/2.2
atom	http://www.w3.org/2005/Atom	XSD	Atom	http://schemas.opengis.net/kml/2.2.0/atom-author-link.xsd
xal	urn:oasis:names:tc:ciq:xsdschema:xAL:2.0	XSD	xAL	https://docs.oasis-open.org/election/external/xAL.xsd

5 Information Exchange

5.1 Overview

The MIP4-IES focuses on information exchange and explores several exchange patterns through which the information constructs can be exchanged. This is partially to ensure that known implementation constraints are considered, but also to give a suggested exchange method that is well suited for several MIP4-IES integration scenarios.

Generally speaking, MIP4-IES aims to provide interoperability between distributed systems with several degrees of decoupling and includes features for effective, secure (information security at message and/or transport or network levels) and reliable information exchange. Messaging is one of the possible integration styles. Possible alternative styles are: File Transfer and Shared Database (RPI can be considered a variant of messaging where the exchanged messages represent the request and responses, if any).

One of the major messaging features is the message distribution or exchange. A Message can be described as a set of data containing a concrete kind of information (e.g. MIP4 Information Constructs) that is intended to be interchanged between two or more actors. The action of interchanging a set of data represents what is known as Messaging and is usually (but not only) applied within the distributed systems context. How different actors of a message passing system connect and communicate with each other is what in literature is called Message Exchange Patterns (MEP) or communication paradigm. A MEP consists of the description of the roles of distinct participants/actors, the actions expected to be performed according to such role and the sequence of messages exchanged in order to achieve an objective in a predictable manner. In order to ensure technical interoperability, all participants share the same definition of the MEP and they act as prescribed by the MEP.

General speaking, in a layered architecture, MEPs may exist at different abstraction levels and architecture layers. A MEP at a higher layer may be realized by a combination of other MEPs at lower layers (e.g. Request/Response at system/application layer may be implemented through asynchronous message queuing at a middleware/infrastructure layer).

In order to facilitate the understanding of the MEP and to support design decisions made when evaluating the most appropriate MEP for a given information exchange it is recognized that a well-structured characterization of each MEP is required. One interesting characterization available is the one provided in "*The Many Faces of Publish/Subscribe*" [REF-EM-07], where MEPs are characterized using the following attributes:

a) **Conversation direction**. A conversation is a series of related messages describing the sequencing and direction of the flow of these messages between the interacting participants.
b) **Interaction Cardinality**. It is the number of participants of the different roles participating in the exchange of messages.
c) **Decoupling degree**. It describes the degree of loose coupling between the interacting participants. Decoupling has three dimensions:
    i) **Synchronization dimension**: Synchronization decoupling means that the interacting parties are not blocked during the interaction and they can do other work;

ii) **Space dimension**: Space decoupling means that the interacting parties do not have to know each other to initiate and complete the interaction;

iii) **Time dimension**: Time decoupling means that the interacting parties do not have to be actively participating at the same time during the interaction.

The cardinality and decoupling attributes are useful in identifying the first two main MEPs: Request-Response and Publish-Subscribe.

Both MEPs can be further detailed. For instance for the Request-Response we may have the synchronous Request-Response that does not provide synchronization decoupling and asynchronous Request-Response that provides synchronization decoupling.

For the Publish-Subscribe exchange pattern, the space and time decoupling dimensions are used to further classify/evaluate different Publish-Subscribe standards and architectures. For instance, the OASIS standard for *WS-BaseNotification* [REF-EM-02] does not provide time and space decoupling between producer (publisher) and consumer (may also have a role of subscriber). This is because the broker-less architecture is adopted (no intermediary is foreseen). If those degrees of decoupling are needed, the OASIS standard *WS-BrokeredNotification* [REF-EM-04] has to be adopted. Those degrees of decoupling are achieved thanks to the adopted brokered architecture.

On the other hand, other Publish-Subscribe standards such as the *OMG Data Distribution Service* [REF-EM-05], even if based on a broker-less architecture, provide also space and time decoupling.

Currently, MIP4 information exchange is based on two basic MEPs: Synchronous Request-Response and Broker-less Publish-Subscribe. Both MEPs are implemented using WS-Services standards and profiles. For both of these MEPs, the WSMP Core Specification [REF-EM-11] is used. Thus, the following documents are essential pre-requisites to understanding the concepts in this document: (i) *NATO Web Service Messaging Profile Core Specification* [REF-EM-11], (ii) OASIS *WS-BaseNotification specification* [REF-EM-02], and (ii) the *TIDE Transformational Baseline* [REF-EM-03].

The two main exchange patterns are specified in *MIP4-IES Publish/Subscribe Exchange Pattern* [REF-MIP-03] and *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04]. Furthermore, in addition to the basic MEPs employed in message-based integration, an additional offline, file-based exchange pattern is described in *MIP4-IES File Exchange Pattern* [REF-MIP-05].

The MIP4-IES specifications is based on web services and file exchange. It is intended to be integrated in a full operational landscape, with technical infrastructure layers and operational procedures.

It is outside of the scope of the current specification to deal with the following subjects:

- Information Assurance: the Information Assurance over a coalition is a whole topic, which is already dealt with by others community. For example, STANAG 4778 and 4774 are expected to be the basis of a whole Information Assurance architecture.

5.2 Use Cases

Hereafter is described how the different exchange patterns are used in specific scenarios/Use cases.

5.2.1 System Initialization

As part of system initialization, it is required to obtain the Situation as known by another one or more systems playing the Provider role. Two options are foreseen:

a) The system being initialized obtains the Situation by using the MIP4-IES Request/Response exchange pattern (playing the Consumer role) and interacting with one or more systems (playing the Provider role) capable to provide the Situation as known by them. The MIP4-IES does not specify how the discovery (who is able to provide required information and its endpoint reference) of such providers is governed.
b) The system being initialized obtains the Situation by using the MIP4-IES File Based exchange pattern. The Situation is received in one or more files.

Implementation of this use case is detailed in *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04] and *MIP4-IES File Exchange Pattern* [REF-MIP-05].

5.2.2 Situation recovery

A system playing the Consumer or Base Consumer role has determined, or suspects, that it has lost the Situation as known by one or more Providers. The following is recommended:

a) The system recovers the actual Situation by using the MIP4-IES Request/Response exchange pattern (playing the Consumer role) and interacting with one/more systems (playing the Provider role) capable to provide the Situation as known by them. MIP4-IES does not specify how the discovery (who is able to provide required information and its endpoint reference) of such providers is governed.
b) The system recovers the actual Situation by using MIP4-IES File exchange pattern by receiving the Situation as one or more files created by the Producer. This is not the most efficient way, since a partial recovery is not possible.

Implementation of this use case is detailed in *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04] and *MIP4-IES File Exchange Pattern* [REF-MIP-05].

5.2.3 Situation Low-frequency Refresh

A system playing the Base Consumer role (e.g a Non-Governmental Organisation) needs to update the Situation at low-frequency (i.e it doesn't need to get every instantaneously changes on the situation).This use cases is addressed in the same way as the Situation Recovery. The following is recommended:

a) The system being refreshed obtains the Situation by using the MIP4-IES Request/Response exchange pattern (playing the Base Consumer role) and interacting with one/more systems (playing the Provider role) capable to provide the Situation as known by them. MIP4-IES does not specify how the discovery (who is able to provide required information and its endpoint reference) of such providers is governed.

b) The system being refreshed obtains the Situation by using the MIP4-IES File exchange pattern receiving the Situation as a files created by one/more systems.

Implementation of this use case is detailed in *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04] and *MIP4-IES File Exchange Pattern* [REF-MIP-05].

5.2.4 System restart

Depending on the restarting system's data persistency features, the system restart may be covered in the same way as foreseen for System Initialization or Situation recovery both.

5.2.5 Per object verification, inspection or elaboration

The purpose is to verify/inspect/elaborate a subset of information. The system interested in this subset obtains the required information by using the MIP4-IES Request/Response exchange pattern (playing the Consumer role) and interacting with one/more third systems (playing the Provider role) capable to provide the Situation as known by them. MIP4-IES does not specify how the discovery (who is able to provide required information and its endpoint reference) of such providers is governed.

Implementation of this use case is detailed in *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04].

5.2.6 Offline Archive of the Situation

The MIP4-IES File exchange pattern shall be used by both Consumer and Provider roles.

Implementation of this use case is detailed in *MIP4-IES File Exchange Pattern* [REF-MIP-05].

5.2.7 Consumption of the Situation

In the nominal case (not during initialization or recovery or low-frequency situation refreshing) one or more systems consume the situation by receiving updates as the situation evolves. It is recommended that the system interested in consuming the situation receives updates by using the MIP4-IES Publish/Subscribe exchange pattern (playing the Notification Consumer role) and receiving asynchronous notifications from one or more third systems (playing the Notification Producer role) capable to provide the Situation as known by them.

Implementation of this use case is detailed in *MIP4-IES Publish/Subscribe Exchange Pattern* [REF-MIP-03].

5.3 Roles

Three different roles are defined by MIP4-IES, according to the need to implement all or a subset of the preceding use cases:

a) A MIP4-IES conformant ***Producer*** shall implement all three exchange patterns (MIP4-IES Publish-Subscribe, Request-Response and File Exchange) according to the applicable MIP4-IES and to the role expected to play in the exchange (consumer specific aspects are not mandated).
b) A MIP4-IES conformant ***Base Consumer*** shall implement both MIP4-IES Request-Response and File based exchanges according to the applicable MIP4-IES and to the role expected to play in the exchange (provider specific aspects are not mandated).
c) A MIP4-IES conformant ***Consumer*** shall implement all three exchange patterns (Publish-Subscribe, Request-Response and File Exchange) according to the applicable MIP4-IES and to the role expected to be played in the exchange (provider and producer specific aspects are not mandatory).

The following sections provide an overview of the MIP4-IES exchange patterns. The technical details of each pattern can be found in their respective specifications.

5.4 Initiating Information Exchange

Prior to initiating exchange of MIP4 messages, each participant in the exchange will complete the *MIP4-IES Configuration Form* [REF-MIP-11] and share this questionnaire with their exchange partners. The exchange of completed questionnaires will:

a) Inform exchange partners of the capabilities of a partner system;
b) Allow the exchange of source identifiers; and
c) Aid in the early identification of potential issues.

The exchange partners are expected to resolve any identified issues prior to the actual exchange of MIP4 messages.

5.5 MIP4-IES Request/Response Exchange Pattern

Request/Response is a common pattern which computers use to communicate with each other. In this pattern the Consumer makes a request to the Provider and requires an immediate reply. This pattern is common in both system-to-system and application-to-system communication.

The *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04] defines a platform specific implementation based on a SOAP-based Web Service. The Web Service is suitable for the sharing of discrete messages between a Provider and a Consumer. This pattern provides the technical details required to instantiate a Request/Response Web Service and the usage details of how a Consumer would invoke service operations to implement one or more use cases applicable to this exchange pattern.

5.6 MIP4-IES Publish/Subscribe Exchange Pattern

In a Publish/Subscribe pattern the Consumer subscribes to content available from the Producer. The Producer then provides the content to the Consumer on an asynchronous basis as updates become available. The subscription generally holds until it is terminated or expires under the specific implementation rules.

The *MIP4-IES Publish/Subscribe Exchange Pattern* [REF-MIP-03] uses a platform-specific implementation of the *Web Service Messaging Profile* [REF-EM-11] based on a SOAP-based Web Service. The Web Services are suitable for the sharing of discrete messages between a Producer and a Consumer. This pattern provides the technical details required to instantiate a Publish/Subscribe Web Service and the usage details of how a Consumer would invoke service operations to fulfill one or more use cases applicable to this exchange pattern.

5.7 MIP4-IES File Exchange Pattern

The File Exchange pattern is defined as the ability to store bulk information in a single file for the purpose of exchanging information between systems or sharing between users. This exchange method supports the following use cases:

a) Offline Archive of the Situation;
b) System Initialization; and
c) System Recovery.

The *MIP4-IES File Exchange Pattern* [REF-MIP-05] defines an XML document element that can contain multiple discrete messages and be saved to a file for exchange. The method for sharing MIP4 files is context specific and thus not specified here.

5.8 Error Handling

The MIP4-IES exchange patterns handle protocol errors but not information payload errors. These errors will need to be handled through other means using the contact information provided in the *MIP4-IES Configuration Form* [REF-MIP-11].

5.9 Specification Principles

In order to achieve the highest amount of (re)use of existing implementations, either Commercial Off-the-Shelf (COTS) or proprietary, the specifications should be closely aligned with the base public standards. As a result MIP4-IES will aim to avoid restricting the usage of those standards. However if by doing so, the MIP4-IES would become unnecessarily complex, MIP retains the option to restrict the specifications in such a manner that existing implementations of those standards might not be usable without further investment. In the case that these further restrictions are enforced by the MIP4-IES, this will be explicitly stated in the related documentation. Furthermore that decision could be open for re-evaluation in any future version of the specification if deemed necessary.

6 Cross-pattern Requirements

This section captures the requirements that are common to multiple message exchange pattern, in order to avoid redefinition of the same requirements.

6.1 Service Metadata

In a web-service oriented architecture, a WSDL file is the contract between two or more interacting parties. It defines, at a technical level, how the exchange is expected to be formatted, and which interactions are to be performed in order to achieve correctly the data exchange.

A WSDL defines different types of information:

a) Services
b) Ports
c) Bindings
d) Port types
e) Message

Bindings (even defined in a standard way in a WSDL) often tend to be implementation specific. Thus, the MIP4-IES only considers the definition of the port types (and consequently messages and operations) to be contained in the “reference” WSDL. The bindings will be documented in the applicable specification, and provided as examples aside.

It is then expected that implementers will import these reference WSDLs into their own WSDLs and derive one or more services that, combined, implement the port types and all their operations. It means that this new WSDL has to be shared among the different partners, in order to achieve correctly the information exchange.

There are 3 possible means the Server can use to make available its WSDL:

a) The Consumer queries the Provider’s WS server (e.g. http://example.com/PublishSubscribeProducerService?wsdl). In production/operation it MAY not be recommended to allow this because of possible security risks on the Producer side.
b) The Consumer queries at design time (Repository) and/or at runtime (Registry, where each Provider has published its WSDL). There are well-known applicable and widely used standards such as UDDI and ebXML.
c) The Provider shares the WSDL file by appropriate means (emails, network folders, etc.) and the Consumer processes it at configuration and/or runtime.

The MIP4-IES does not mandate any means to provide such WSDL, but this consideration MUST be managed before connecting each partner.

6.1.1 Endpoint References and Resource Locations.

The WS-Addressing standard provides the ability to decouple the name of a WebResource from the actual resource location. Implementation must consider that received EndpointReferences can not be directly resolve to a URL. The URI in an EndpointReference may be a URN, in this case the MIP4IES system exposing these URNs is responsible in providing sufficient information to other systems that allows resolving the URN to a URL.

6.2 Message Container

MIP4-IES uses the wsmp:CRUDCommandType from WSMP that wraps each payload element according to the operation performed on data.

The actual data is inserted into the Data element of the wsmp:*CRUDCommandType*.

In MIP4-IES, the content of the data element of the wsmp:CRUDCommandType MUST be a single element of a type “base:ContextType“.

In order to detect that the payload corresponds to MIP4-IES, the Provider/Producer SHALL fill the Dialect attribute of the Data element with the following value: https://mip-interop.org/data/v4.3/Dialect.

The wsmp:*CRUDCommandType* can also contain a wsmp:Policy element. This element aims to allow the Consumer to provide the Provider with some characteristics about the response that will be sent back. Basic characteristics are the compression, payload encoding and binding.
MIP4-IES does not mandate to support any particular compression algorithm, encoding or any other binding than “inline”. That being said, it is highly recommended to any MIP4-IES consumer implementation to be “unsupported policy proof”: in this objective, a MIP4-IES consumer should be able to adapt its request after having receiving a wsmp-f:UnsupportedPolicyFault.

For more details about the wsmp:*CRUDCommandType*, refer to the *WSMP Core Specification* [REF-EM-11].

6.3 WS-I Profiles and Bindings

As stated in the *WSMP Core specification* [REF-EM-11], the MIP4-IES Web Services are WS-I Basic Profile 2.0 compliant. This profile is based upon, and is compliant with, the following specifications:

Specification	Namespace
XSI	http://www.w3.org/2001/XMLSchema-instance
XSD	http://www.w3.org/2001/XMLSchema
SOAP Encoding	http://schemas.xmlsoap.org/soap/encoding/
WSDL 1.1	http://schemas.xmlsoap.org/wsdl/
SOAP 1.2 Binding	http://schemas.xmlsoap.org/wsdl/soap12/

WS-Addressing	http://www.w3.org/2005/08/addressing

The Binding SHALL be according to Basic Profile 2.0 rules. The minimum set of rules ensuring service interoperability is:

a) The SOAP Binding style SHALL be set to document.
b) The SOAP Binding transport SHALL be set to HTTP (http://schemas.xmlsoap.org/soap/http).
c) Per SOAP Operation, the style SHALL be set to document.
d) Per SOAP Operation, the input, output and faults SOAP Body SHALL use literal.
e) The SOAP Binding SHALL enable the use of WS-Addressing.
f) Per SOAP Operation, the WS-Addressing [action] Message Addressing Property for input and output messages SHALL contain the URIs [REF-EM-01] provided in the applicable specification.
g) Each Port SHALL have a unique name.
h) The Ports SHALL link to the appropriate PortTypes via a Binding (following the MIP4-IES Binding rules.
i) The service SOAP Address location SHALL define each Port's endpoint.

WS-I organization delivered a set of tools that can be used to validate the compliance to the static part of the Basic Profile 2.0. These tools are available at the following URL: http://www.ws-i.org/deliverables/workinggroup.aspx. These tools allow to check if each implementation exposes a compliant WSDL, and if the content of the SOAP messages is correctly formatted.

It should be noted that the tools do not check any dynamic behaviour enforced by the Profile, thus do not exempt the implementer from checking the WS-I Basic Profile 2.0 compliance of the service.

6.4 Filtering Data

6.4.1 General

According to the given need of a given actor in a given phase of an operation, there are different needs for filtering. The filtering capabilities that are currently foreseen, are:

a) Topic filtering: the consumer expresses an interest in data incoming in a or multiple given topics;
b) Filtering by Identifier

These filtering capabilities are to be taken as examples, and there may be other filtering needs that arise during an operation. Based on WS-Notification standard for its filtering ability, the MIP4-IES does not exclude any other kind of filter as long as it is possible to express them using the standard. However, it specially defines technical solutions to address this previous list of filters.

In both Publish/Subscribe and Request/Response, the MIP4-IES defines exactly the same technical means to express the filters, and it follows what is defined in the WS-BaseNotification. The *WS-BaseNotification specification* [REF-EM-02] provides more detail about the behaviour of Consumer or Producer according to filtering. In particular, the MIP4-IES conforms strictly to the standard and the *TIDE Transformational Baseline* [REF-EM-03] on fault raising.

OASIS WS-N specifications define 3 boolean filters: *TopicExpression*, *ProducerProperties*, and *MessageContent* (The OASIS WS-BaseNotification specification [REF-EM-02] provides more details about this). Additional filters (e.g Domain Specific filters) are also available, but are not addressed in this specification. According to the identified need of filtering (see the previous list), two of them are identified to be used in the MIP4-IES:

a) TopicExpression
b) MessageContent

Attention: TopicExpression and MessageContent values are defined as *xsd:string* in the schemas. Still, these values should be processed in the way defined in the corresponding dialect. For example:

a) SimpleTopicExpression corresponds to a QName
b) ConcreteTopicExpression uses QName
c) FullTopicExpression uses QName

This means that the producer and consumer MUST consider these Strings as QName (or whatever is defined in the corresponding Dialect), and in the case of QName, MUST take care of the definition of the prefixes in the XML document.

As stated in the *WS-Topics specification* [REF-EM-06]: "...some XML processors might modify the namespace declarations. Designers should be aware that such transforms exist and might render the expression incoherent; as it is likely the change in namespace declaration will not update a QName embedded within a string."

6.4.2 Topics and Topic filtering

A Topic (also referred to as WS-Topic), is a technical mechanism that provides a way to categorize and organize the data space during an exchange. In the realm of MIP4, this categorization corresponds to an operational meaning that can be associated to an exchange (e.g.: NATO C3 Classification Taxonomy/Information Products). MIP4-IES does not expect this categorization to be known during the implementation phase, thus uses a generic technical mechanism.

However, the categorization SHALL be defined, agreed and shared among the partners, prior to the joining time of each of them. This definition is included in the *MIP4-IES Configuration Form* [REF-MIP-11]. It is technically possible to change the categorization during an operation, but it will require that each partner will be able to manage the change at runtime.

WS-Notification (as defined in the *WS-Topics specification* [REF-EM-06]) provides the wstop:*TopicNamespace* document to encode the semantics and message types associated with a Topic. A sample wstop:*TopicNamespace* document including three Topics follows:

```
<wstop:TopicNamespace name="MIP4-IES" targetNamespace="https://mip-interop.org/data/v4.3/topics"
xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wstop="http://docs.oasis-open.org/wsn/t-1" >
   <wstop:documentation>Sample Topics defined by the MIP4-IES Publish Subscribe web
services</wstop:documentation>
   <wstop:Topic name="ORBAT">
       <wstop:documentation></wstop:documentation>
   </wstop:Topic>

   <wstop:Topic name="LandTracks">
       <wstop:documentation></wstop:documentation>
   </wstop:Topic>
   <wstop:Topic name="FriendlyArmyCommandPost">
       <wstop:documentation>Associations between battlespace objects.</wstop:documentation>
   </wstop:Topic>

</wstop:TopicNamespace>
```

It is expected that prior to any exchange being performed using the MIP4-IES, an agreement on the TopicNamespace MUST be found between the different MIP4 actors.

This TopicNamespace is then shared by all nations. The means to share the TopicNamespace is not specified in this document, and relies on the capabilities of the underlying infrastructure (email, file exchange …)

In the *WS-BaseNotification specification* [REF-EM-02], there are multiple Dialects available to define a topic expression. In order to handle hierarchy of topics (and being compliant to *TIDE Transformational Baseline* [REF-EM-03]), MIP4-IES mandates the different server-side implementations (Provider, Producer) to support:

a) Simple Topic expression
b) Concrete Topic Expression
c) Full Topic Expression

A Consumer MAY choose to support one or more of these dialects, as long as it provides one filter expression during the request or the subscription (requirement PSWN-NP06 from the *TIDE Transformational Baseline Publish/Subscribe SOAP Profile* [REF-EM-10]).

In the case where the Consumer does not provide any filter on Topic during the request or subscribe, the server-side component (Producer, Provider) MUST reject the request replying with a Fault accordingly to the *WS-Notification specification* [REF-EM-02]).

Thus, in order to define a filter that matches any data in the data space, the Consumer SHALL use a FullTopicExpression such as "?/*". A non exhaustive list of possible filtering on topics using the FullTopicExpression dialect (*tns* indicates the current target namespace) includes:

a) "tns:t1/*" identifies all of the child Topics of the root Topic t1.
b) "tns:t1/*/t3" identifies all grandchildren of tns:t1 that have the name t3.
c) "tns:*" identifies all root Topics in the tns: Topic Namespace.
d) "tns:t1/t3//." identifies the sub-tree consisting of tns:t1/t3 and all its descendants.
e) "tns:t1/t3//*" identifies the sub-tree consisting of the descendants of tns:t1/t3 but, unlike the previous example, does not include tns:t1/t3 itself.

f) “tns://*” identifies all the Topics in the entire Topic Namespace.
g) “tns:t1//t3” identifies all descendants of tns:t1 that have the name t3.

6.4.2.1 Usage of Topics

The WSMP Exchange mechanism offers TopicTrees as the mechanism to allow producers to subscribe to published information, as well as the Request/Response.

In order to properly use the MIP4IES the following rules towards the usage of the Topics must be adhered.

- MIP4 systems shall only publish information, on a Topic specific to this System.
- When a StaffConcept is published on a Topic, all content shall be published on this Topic.
- When Synchronizing, any information previously received, but no longer Returned on a Topic shall be deemed unpublished or deleted by the Consumer.

6.4.2.2 Usage of Topic Expressions

The Message Exchanges of WSMP offer different places to provide TopicExpressions, it is difficult to understand these different locations as their uses are documented in several places. This paragraph provides clarity on the usage of the TopicExpressions

- **wsnt:NotificationMessage/Topic** [Pub-Sub-Only]
  This location is provided by WS-Notification as part of the Notification message. As the Publish Subscribe EM specifies this value must always be used when notifying, and always resolve to a single Topic.
- **wsmp:WSMPMsg/[Create|Update|Delete]/TopicExpression[0..*]**
  When providing WSMPMsg an implementation may place multiple TopicExpressions that resolve to multiple Topics, in order to express that the contained Data is present on all of those Topics.
  NOTE: In case the WSMPMsg is provided within a Notification by the Publish-Subscribe EM these TopicExpressions must resolve to a set of Topics which at least contain the Topic expressed in the NotificationMessage (see bullet above)
- **wsmp:WSMPMsg/[Create|Update|Delete]/Filter/TopicExpression**
  This TopicExpression should not be used within the context of a WSMPMsg, implementations that receive a TopicExpression in this element must ignore it.
- **wsmp:[Create|Read|Update|Delete]/TopicExpression[0..*]** [Req-Res-Only]
  This TopicExpression should not be used requesting a Create, Read, Update and Delete operation in the Request Response EM. Implementations that receive a TopicExpression in this element must ignore it.
- **wsmp:[Create|Read|Update|Delete]/Filter/TopicExpression** [Req-Res-Only]
  The Create, Read, Update and Delete elements are sent when requesting an operation of the Request-Response EM. As specified by WSMP in this case the TopicExpression is used to provide a selection of information, on which the operation is to be applied.

6.5 Unexpected content tolerance

Due to the generic definition of the underlying exchange mechanism (WSMP), there is a non insignificant risk that a Producer sends data using an unexpected or unsupported format. For example, this may happen when using extension points, or because of a badly configured Producer that sends other CoI data to a MIP4 Consumer.

In such a case, the Consumer SHALL be able to receive the content without blocking any new incoming data. There is no particular expectation about the processing to be performed on the unsupported data (it could be logged, stored in a dedicated or dropped, according to each national implementation constraints).

6.6 Unnecessary Notifications and Large Payloads

Each MIP4-IES Producer should implement measures to avoid sending unnecessary data, where feasible, in order to prevent overloading Consumers. This would include measures to minimize the re-sending of previously issued data, unless explicitly requested (using the Request Response Exchange Pattern). Similarly, MIP4-IES Consumers are encouraged to implement measures that protect themselves from high transaction volume (e.g. analyse received payload to detect if there is any update since the last notification).

While the MIP4-IES does not mandate a limit on the size of the notification payload, it should be recognized that the chance of successful transmission decreases as the payload size increases beyond normally expected limits. Each MIP4-IES Producer should implement measures to avoid generating payloads of excessive size.

6.7 Creation, Update and Removal

When an Identifiable is created on a producers system it will be published to all the topics it is deemed relevant. When that information changes the updated information will be sent to all those systems. Once the information is considered no longer relevant within a topic that it was previously published on, the Provider will locally remove the related Identifiable from that Topic, and then it will inform of the removed Identifiable to all of the served Subscribers of that Topic, refer to the specific EM pattern documents for details.

Removing an Identifiable from a topic is done with a WSMP-Delete operation [see below]. Publishing an Identifiable on a topic is done with a WSMP-Create or WSMP-Update operation *[see below]*.

The consumer will maintain the “state”/”value(s)” of an Identifiable is global with respect to its 3-part identifier (uri/source/discriminator), regardless of the operation used. This means that regardless of the topic a WSMP-Operation is received, the “state”/”value(s)” of an Identifiable will be applicable for all Topics this Identifiable is relevant. An Identifiable is considered relevant to a Topic as soon as a WSMP-Create or WSMP-Update has been received for that Topic. A received delete indicates that an Identifiable is no longer relevant to a Topic.

7 Glossary

Terms	**Definition**
Base Consumer	Information system acting as a client, and that is only able to request using synchronous pattern data from a Provider.
Broker	Message broker is an intermediary program module which translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver.
Consumer	Information System/Application consuming the MIP4-IES Request/Response service as per *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04]. Information System subscribed to published information (Notification) as per *MIP4-IES Publish/Subscribe Exchange Pattern* [REF-MIP-03].
COTS	Commercial Off-the-Shelf
Messaging	Have each application connect to a common messaging system, and exchange data and invoke behaviour using messages.
MEP	Message Exchange Pattern: describes the pattern of messages required by a communications protocol to establish or use a communication channel.
Notification	A one-way message sent to indicate that a change in the situation has occurred.
Port Type	For a Web Service, a Port Type defines a set of operations, and the message types these operations are performed on.
Provider	Service Provider: Information System exposing the MIP4-IES Request/Response service as per *MIP4-IES Request/Response Exchange Pattern* [REF-MIP-04].
Producer	Information System publishing information as per *MIP4-IES Publish/Subscribe Exchange Pattern* [REF-MIP-03]. (WS-N Note: it is also responsible for Subscription Handling)
SOAP	Simple Object Access Protocol: a protocol specification for exchanging structured information in the implementation of web services in computer networks
Subscription/Subscribe	Inscription from a Consumer to a Provider for receiving data
WSDL	Web Service Definition Language
WSMP	Web Service Messaging Profile [REF-EM-11]