FAA-STD-073A
December 9, 2019
SUPERSEDING
FAA-STD-073
January 29, 2014

U.S. Department
of Transportation

**Federal Aviation
Administration**

# U.S. Department of Transportation
# Federal Aviation Administration

## Standard Practice

# PREPARATION OF
# JAVA MESSAGING SERVICE DESCRIPTION DOCUMENTS

# FOREWORD

This standard is approved for use by all Departments of the Federal Aviation Administration (FAA).

This standard specifies the minimum acceptable content for documenting Java Message Services within the FAA.

This standard has been prepared in accordance with FAA-STD-068, Department of Transportation Federal Aviation Administration, *Preparation of Standards* [9].

Comments, suggestions, or questions on this document shall be addressed to:

Federal Aviation Administration
System Wide Information Management (SWIM) Program Office, AJM-316
800 Independence Avenue, SW
Washington, DC 20591

# Table of Contents

# List of Figures



# List of Tables

# 1 SCOPE

This standard provides a set of requirements for developing a Java Messaging Service (JMS) Description Document (JMSDD). The JMSDD provides the details needed to sufficiently describe a JMS-based service as a part of the FAA's implementation of a Service-Oriented Architecture (SOA).

Any configuration management (CM) or quality assurance (QA) policies, rules, or assertions to which the developed JMSDD may be subjected are outside the scope of this standard.

## 1.1 Introduction

The National Airspace System (NAS) is evolving the traditional mode of information exchange (including highly customized system-to-system interaction) to a paradigm of Service-Oriented Architecture (SOA). The objective of the System Wide Information Management (SWIM) program is to support the development of services in accordance with recognized practices and principles of SOA.

In a SOA environment, services are integrated through well-defined interfaces and communicate via messages. SOA, as an architectural style, does not specify any technological solution for implementing services. However, major industry institutions have established a number of open standards for implementing and describing services [24], [20].

In response to increased demand for event-driven, reliable asynchronous message handling and a higher degree of decoupling, many NAS SOA-driven implementations adopted the JMS Application Programming Interface (API) created by Sun Microsystems [30] as a technological solution for service-oriented development. The JMS API is an open standard for message exchange between loosely coupled distributed software components by means of Message Oriented Middleware (MOM). The concept of MOM has been realized by multiple vendors in industry as an Enterprise Service Bus (ESB). SWIM has established the NAS implementation of an ESB, termed NAS Enterprise Message Service (NEMS), which provides messaging capabilities and supports integration of all NAS collaborating SOA service nodes.

A service description is an integral and indispensable ingredient of a service deployment. In essence, the service description is an artifact that contains complete human-readable information about "how to interact with the service in order to achieve the required objectives, including the format and content of information exchanged between the service and the consumer and the sequences of information exchange that may be expected" [19]. All services require a service description. The FAA has already established procedures for developing Web Service Description Documents (WSDD) [8]. The aim of this standard is therefore to specify requirements for creating a service description document for a JMS-based client according to FAA system engineering standards and practices.

## 1.2 Motivation for Revising this Standard

The need to create Revision A of this standard was motivated by two main factors:

- Changes took place in the infrastructure responsible for supporting some provisions of the original standard.

  The original standard included procedures for service identification and classification that required using the FAA Data Registry (FDR), a tool which the FAA retired in 2017 in favor of more modern and cost-effective solutions. Revision A provides requirements that do not depend on the existence of the FDR.

- The opportunity arose to align the FAA SOA-based implementation of the service description with an architectural model developed together with international partners.

  The FAA worked extensively with the Single European Sky ATM Research (SESAR) organization to develop a shared vision of a service description. This effort resulted in the creation of a Service Description Conceptual Model (SDCM) [5] in 2016. Revision A makes the JMSDD compatible with the SDCM, while continuing to support FAA engineering practices.

In addition to addressing these factors, Revision A updates obsolete references and provides additional information about the NAS Service Registry/Repository (NSRR) and its relationship to the JMSDD.

## 1.3 Intended Audience

The intended audience for this standard includes architects and developers designing, identifying, or developing a system based on the JMS API, and decision makers seeking a better understanding of the application of SOA and enterprise messaging principles.

## 1.4 Basic Concepts

The goal of this section is to establish a clear and unambiguous understanding of several important concepts used in this standard. NOTE: the concepts discussed below are those that deal primarily with JMS-based services. For explanations of other concepts and artifacts that are common to all SOA services and their service descriptions, as well as a detailed description of the relationship between the parts of a service description and the Service Description Conceptual Model (SDCM) [5], refer to Standard Practice FAA-STD-065, Preparation of Web Service Description Documents (WSDD) [8].

### 1.4.1 Java Message Service Components

JMS is a Java-based application programming interface (API) that provides a common way for Java programs to create, send, receive, and read an enterprise messaging system's messages. [7] A JMS message is created by a message producer and consumed by a message consumer as shown in Figure I.

A JMS-based messaging system is composed of the following parts:

- *JMS provider* – Specialized software that accepts messages from sending processes and delivers them to receiving processes (typically across a network). NOTE: an enterprise messaging product like NEMS, besides being a JMS provider, also provides other capabilities such as fault tolerance, load balancing, mediation support, etc.
- *Message producer* – An application or process that creates and sends messages. A message producer is called a "queue sender" in the point-to-point messaging model and a "topic publisher" in the publish/subscribe messaging model.
- *Message consumer* – An application or process that receives messages. A message consumer is called a "queue receiver" in the point-to-point messaging model and a "topic subscriber" in the publish/subscribe messaging model.
- *JMS client* – A collective term used to refer to applications or processes that send and/or receive messages (*message producers* and *message consumers*).

Figure 1 illustrates some of the points described above.

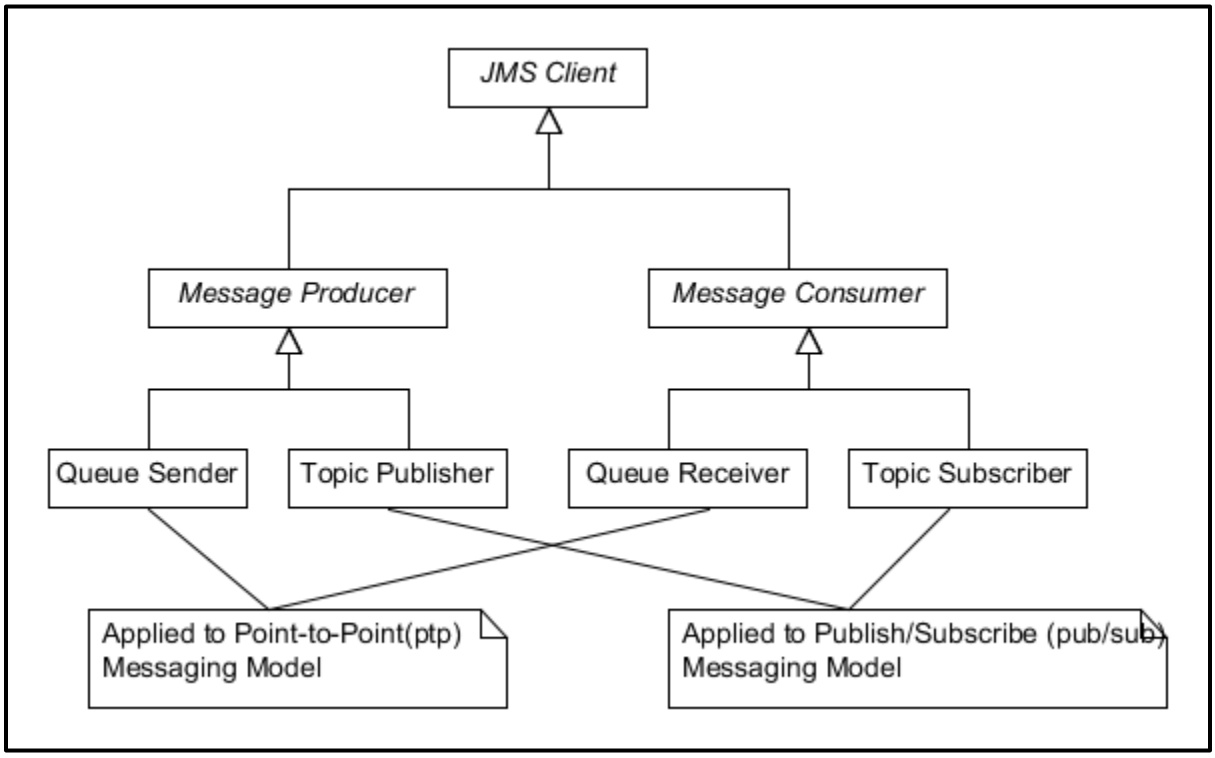


**FIGURE 1. JMS client terminology hierarchy**

### 1.4.2 Messaging Models

The JMS specification supports two messaging models: *point-to-point* or PTP and *publish/subscribe* or pub/sub. The following sections describe each of these models.

#### 1.4.2.1 Point-to-Point Messaging Model

The point-to-point (PTP) messaging model is built around the concept of message *queues*, where each queue is a staging area that contains messages that have been sent and are waiting to be read. Each message is addressed to one consumer, which is reminiscent of a one-to-one connection (thus "point-to-point"). The PTP messaging model supports asynchronous "fire and forget" messaging as well as synchronous messaging.

Figure 2 depicts the PTP messaging model.

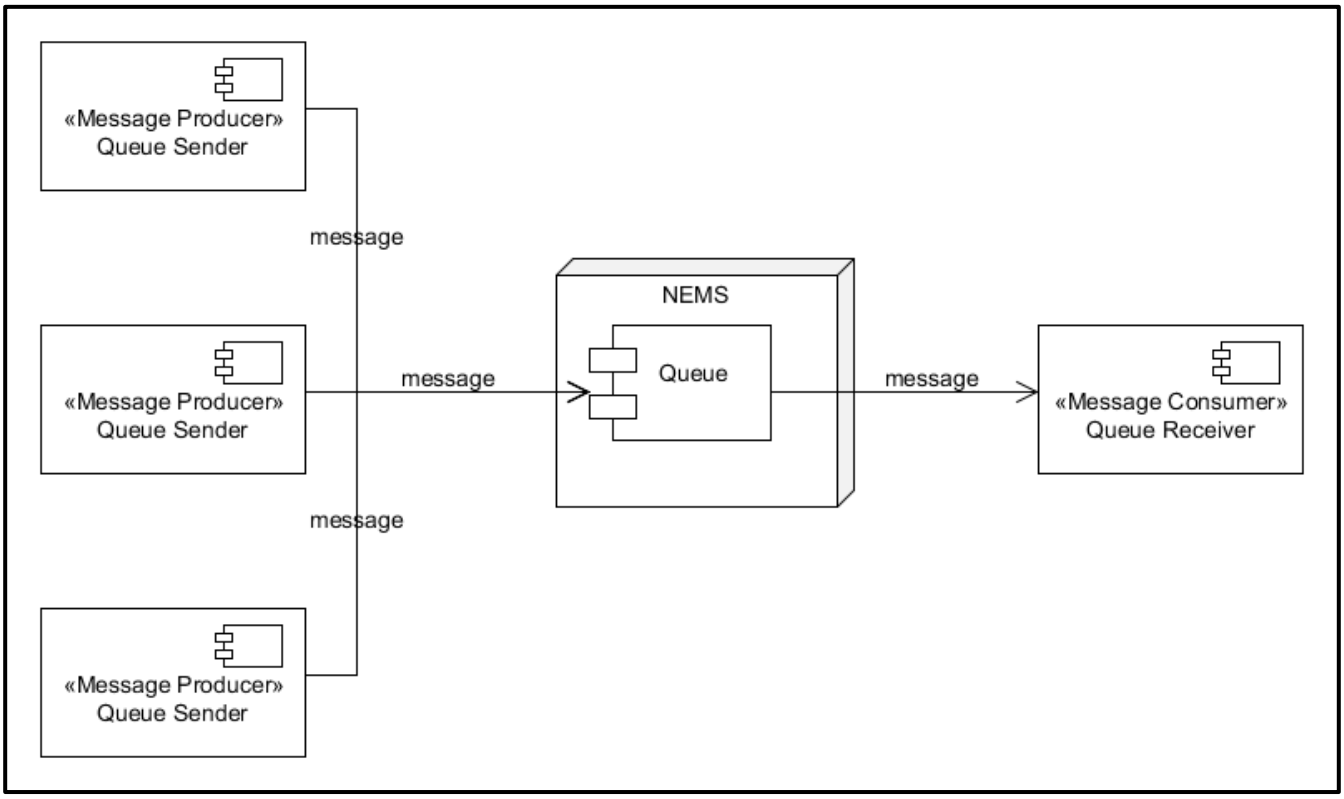


**FIGURE 2. Point-to-Point (PTP) messaging model**

#### 1.4.2.2 Publish/Subscribe Messaging Model

The publish/subscribe (pub/sub) messaging model is built around the concept of a *topic*, where each message may be delivered to multiple consumers that have subscribed, i.e., registered interest, to a specific topic. The pub/sub messaging model is similar to the notion of one-to-many relationships. The topic is published by a *topic publisher* to a *JMS provider* and then can be subscribed to and consumed by multiple *topic subscribers* (this is sometimes referred to as broadcasting the messages). Thus a *topic* is a distribution mechanism for publishing messages that are delivered to multiple subscribers.[28]

Figure 3 depicts the pub/sub messaging model.

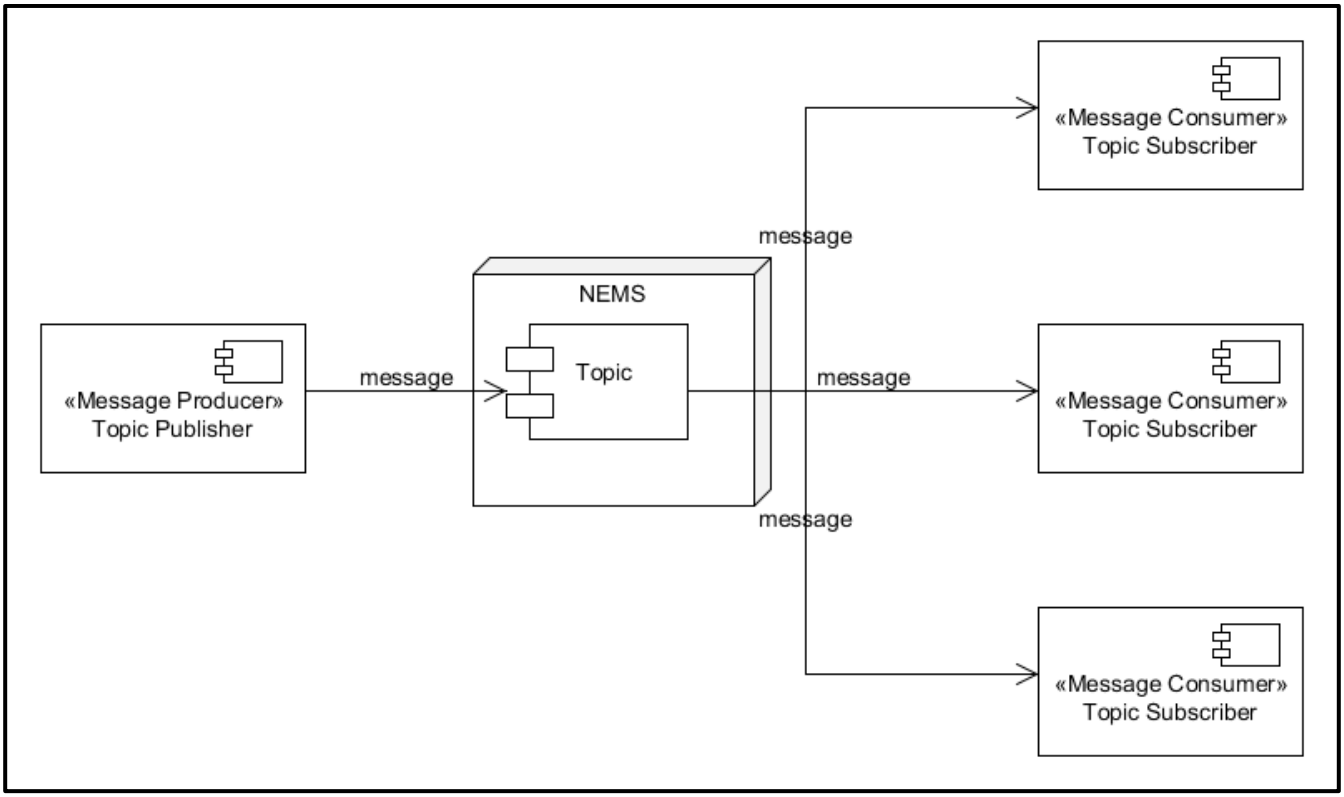


**FIGURE 3. Publish/subscribe messaging model**

### 1.4.3 JMS Message

In a JMS-based messaging system, a *message* is a self-contained package of business data and routing headers. JMS messages are entities that are used to exchange information between JMS clients.

The message's header and property information is used by a JMS provider for routing, filtering, and delivering the payload. The payload is the content of the message.

A JMS message consists of three parts:

- *Header* – all messages support a standard set of header fields. Header fields contain values used by both clients and providers to identify and route messages. [30]
- *Properties* – in addition to the standard header fields, messages provide a built-in facility for adding optional header fields to a message, e.g., properties are used to expose data for message filtering by a JMS provider. JMS defines three types of properties: JMS-defined, provider-specific, and application-specific.
  - *JMS-defined properties* are properties established by the JMS specification. The difference between JMS-defined properties and JMS headers is that vendors can choose not to support JMS-defined properties, or to support some or all of them.

  - *Provider-specific properties* are properties that can be established by JMS provider to support proprietary vendor features.
  - *Application-specific properties* are properties that are defined by the application developer.
- *Body (or Payload)* – the actual (business) data transferred by the message. JMS defines several types of message body which cover the majority of messaging styles currently in use. [30]

Figure 4 depicts the architecture of a JMS message.

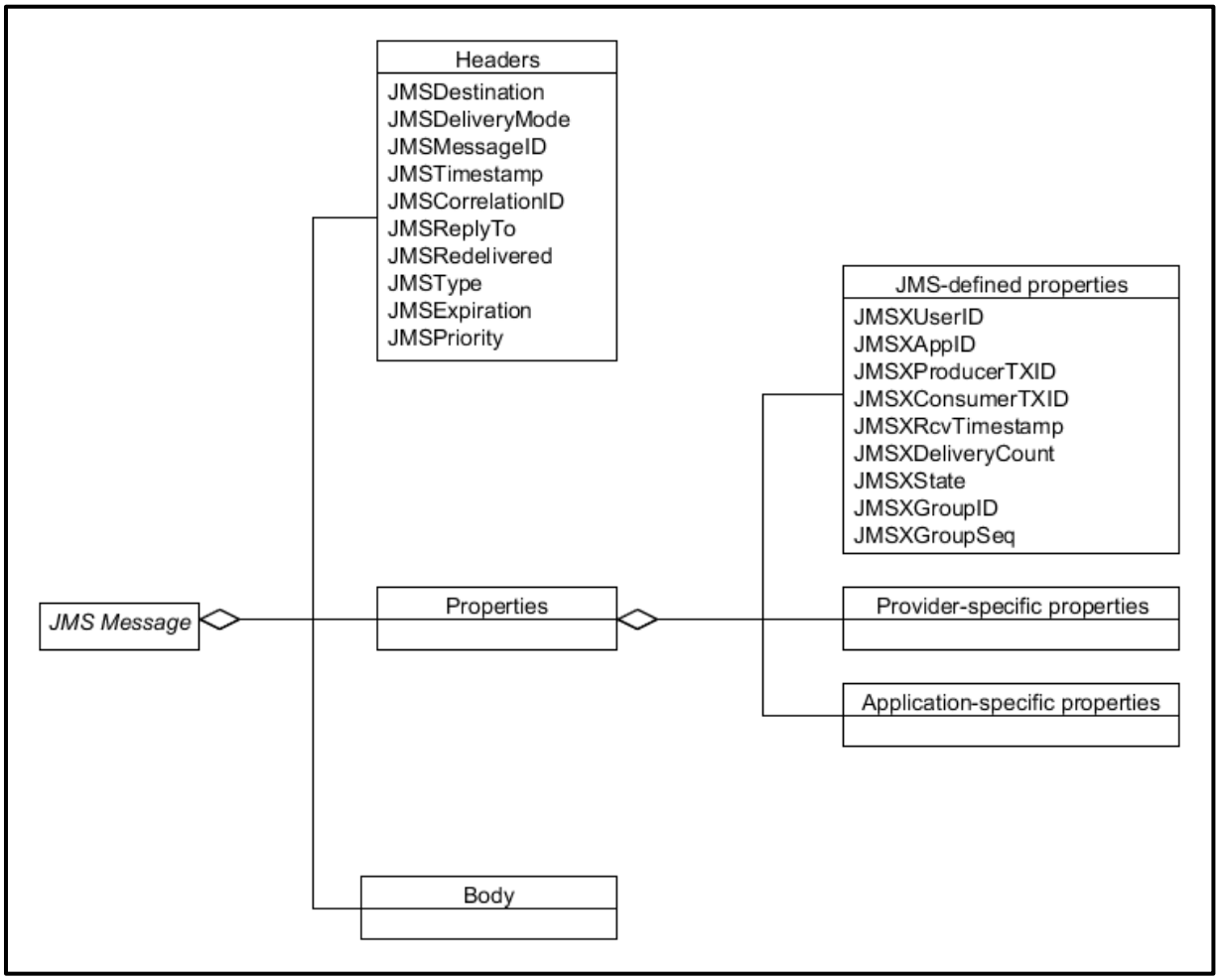


**FIGURE 4. JMS message architecture**

# 2 APPLICABLE DOCUMENTS

## 2.1 Government Documents

[1] FAA Order 1000.36, FAA Writing Standards, 31 March 2003.
http://www.faa.gov/documentlibrary/media/order/branding_writing/order1000_36.pdf

[2] FAA Order 1700.6C, FAA Branding Policy, Use of the FAA Logo, FAA Signature, and DOT Seal, 11 September 2006.
http://www.faa.gov/documentLibrary/media/order/branding_writing/Branding_Order_17006.pdf

[3] FAA Order 1800.66, FAA Configuration Management Policy with Change 3 Incorporated, 2 March 2012.
https://www.faa.gov/documentLibrary/media/Order/1800.66.pdf

[4] U.S. Government Printing Office Style Manual, 30th edition 2008.
http://www.gpoaccess.gov/stylemanual/

[5] Service Description Conceptual Model (SDCM) 2.0, SESAR CP 2.1, 3 June 2016.
http://swim.aero/sdcm/2.0.0/sdcm-2.0.0.html

[6] SWIM-005, Artifacts Versioning for SWIM-enabled Services 1.0.0, FAA, December 2015.
https://www.faa.gov/air_traffic/technology/swim/governance/standards/media/SWIM%20Service%20Versioning%20Spec.pdf

[7] SWIM Controlled Vocabulary v1.0, 25 March 2019.
https://semantics.aero/pages/swim-vocabulary.html

[8] FAA-STD-065B, Preparation of Web Service Description Documents, 15 July 2019.
https://www.faa.gov/air_traffic/technology/swim/governance/standards/media/FAA-STD-065B%207_15_2019.pdf

[9] FAA-STD-068, Preparation of Standards, 4 December 2009.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-068.pdf

[10] FAA-STD-070, Preparation of Web Service Requirements Documents, 12 July 2012.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-070.pdf

## 2.2 Non-Government Documents

[14] DCMI Glossary, Dublin Core Metadata Initiative, User Guide Committee, 23 April 2004.
http://dublincore.org/documents/usageguide/glossary.shtml

[15] Glossary of Security Terms, SANS Institute, August 2011.
http://www.sans.org/resources/glossary.php

[16] ISO/IEC 11404, Information technology — General-Purpose Datatypes (GPD), Second Edition, 15 December 2007.
http://standards.iso.org/ittf/PubliclyAvailableStandards/c039479_ISO_IEC_11404_2007(E).zip

[17] ISO/IEC 12207: 2008 Information Technology – Software Life Cycle Processes, 2008.
http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=43447

[18] ISO/IEC CD 20944-002, Information Technology – Metadata Interoperability and Bindings (MDIB) – Part 002, Common Vocabulary, 12 April 2004.
http://jtc1sc32.org/doc/N1101-1150/32N1105T-CD20944-002.pdf

[19] OASIS Reference Model for SOA 1.0, 12 October 2006.
http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[20] OGC Web Services Common Standard, Version 2.0.0, Open Geospatial Consortium Inc., 7 April 2007.
http://www.opengeospatial.org/standards/common

[21] RFC 2119, Key words for Use in RFCs to Indicate Requirement Levels, Network Working Group, March 1997.
http://www.rfc-editor.org/rfc/rfc2119.txt

[22] RFC 2828, Internet Security Glossary, Network Working Group, May 2000.
http://www.ietf.org/rfc/rfc2828.txt

[23] RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, Network Working Group, January 2005.
http://www.rfc-editor.org/rfc/rfc3986.txt

[24] Web Services Architecture, W3C Working Group Note, 11 February 2004.
http://www.w3.org/TR/ws-arch

[25] Web Services Description Requirements, W3C Working Draft, 28 October 2002.
http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/

[26] ISO/IEC 11179, Information Technology – Metadata Registries (MDR) – Parts 1 - 6.
http://metadata-standards.org/11179/

[27] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004.
http://www.w3.org/TR/xmlschema-2/

[28] Wikipedia, Java Message Service
http://en.wikipedia.org/wiki/Java_Message_Service

[29] Java 2 Platform, Enterprise Edition, v 1.3 API Specification
http://docs.oracle.com/javaee/1.3/api/

[30] Java Message Service Specification Version 1.1: Sun Microsystems, Inc.: April 12, 2002
http://download.oracle.com/otndocs/jcp/7195-jms-1.1-fr-spec-oth-JSpec/

[31] Wikipedia, Time to Live
https://en.wikipedia.org/wiki/Time_to_live

## 2.3 Order of Precedence

In the event of a conflict between the text of this document and the references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

# 3 DEFINITIONS

## 3.1 Key Words

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," “MAY," and "OPTIONAL," in this standard are to be interpreted as described in RFC 2119 [21]. These key words are capitalized when used to unambiguously specify requirements. When these words are not capitalized, they are meant in their natural-language sense.

All examples in the document are labeled as "non-normative", which means they are not to provide a canonical implementation for use in a registry or artifact, but merely to illustrate technical features of a particular approach.

## 3.2 Terms and Definitions

| | |
|---|---|
| ***Asynchronous*** | An interaction in which the associated messages are chronologically and procedurally decoupled. For example, in a request-response interaction, the client agent can process the response at some indeterminate point in the future when its existence is discovered. [24] |
| ***Audit*** | A process that records information needed to establish accountability for system events and for the actions of system entities that cause them. [22] |
| ***Authentication*** | The process of verifying an identity claimed by or for a system entity. [22] |
| ***Authorization*** | The granting of rights or permission to a system entity (mainly but not always a user or a group of users) to access a service. [10] |
| ***Binding*** | An association between an interface, a concrete protocol, and a data format. A binding specifies the protocol and data format to be used in transmitting messages defined by the associated interface. [25] |
| ***Business Function*** | A characteristic action or activity that needs to be performed to achieve a desired objective, or in the context of this standard, to achieve a real world effect. [8] |
| ***Confidentiality*** | Protective measures that assure that information is not made available or disclosed to unauthorized individuals, entities, or processes (i.e., to any unauthorized system entity). [22] |
| ***Data Element*** | A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes. [26] |
| ***Datatype*** | A set of distinct values, characterized by properties of those values, and by operations on those values. [16] |
| ***End Point*** | An association between a fully-specified binding and a physical point (i.e., a network address) at which a service may be accessed. [5] |

| | |
|---|---|
| ***Fault*** | A message that is returned as a result of an error that prevents a service from implementing a required function. A fault usually contains information about the cause of the error. [10] |
| ***Format*** | The arrangement of bits or characters within a group, such as a data element, message, or language. [10] |
| ***Idempotent*** | A term used to describe an operation in which a given message will have the same effect whether it is received once or multiple times; i.e., receiving duplicates of a given message will not cause any undesirable effect. [10] |
| ***Identifier (ID)*** | A sequence of characters, capable of uniquely identifying that with which it is associated, within a specified context. [26] |
| ***Input*** | Data entered into, or the process of entering data into, an information processing system or any of its parts for storage or processing. (Adapted from [18]) |
| ***Integrity*** | Protective measures that assure that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. [22] |
| ***Java Message Service (JMS)*** | A Java-based application programming interface (API) that provides a common way for Java programs to create, send, receive, and read an enterprise messaging system's messages. [7] |
| ***JMS Client*** | An application or process that produces and/or receives messages. [28] |
| ***JMS Provider*** | A messaging system that implements JMS in addition to the other administrative and control functionality required of a full featured messaging product. [29] |
| ***Message*** | A basic unit of communication from one software agent to another sent in a single logical transmission. [7] |
| ***Message Body*** | The actual (business) data transferred by a message. [7] |
| ***Message Consumer*** | A JMS client that receives messages. [28] |
| ***Message-Oriented Middleware (MOM)*** | Software or hardware infrastructure supporting sending and receiving messages between distributed systems. [7] |
| ***Message Producer*** | A JMS client that creates and sends messages. [28] |
| ***Namespace*** | A collection of names, identified by a URI reference, that are used in XML documents as element types and attribute names. The use of XML namespaces to uniquely identify metadata terms allows those terms to be unambiguously used across applications, promoting the possibility of shared semantics. [14] |

| | |
|---|---|
| ***NAS Enterprise Message Service (NEMS)*** | A NAS-based implementation of message-oriented middleware (MOM) that is responsible for distributing messages among information consumers and providers, as well as providing administrative functionality that includes (but is not limited to) fault tolerance, load balancing, mediation and orchestration support. [7] |
| ***NAS Service Registry/ Repository (NSRR)*** | A SWIM-supported capability for making services visible, accessible, and understandable across the NAS. The NSRR (https://nsrr.faa.gov/) provides a flexible mechanism for service discovery, an automated policies-based way to manage services throughout the service lifecycle, and a catalog for relevant artifacts. [7] |
| ***Non-Repudiation*** | Protective measures against false denial of involvement in a communication. [22] |
| ***Normative Document*** | A document that provides rules, guidelines, or characteristics for activities or their results. NOTE: The term "normative document" is a generic term that covers such documents as standards, technical specifications, codes of practice, and regulations. [18] In the context of this standard, a normative document is a set of rules that (1) determines the behavior of interacting entities and (2) has been developed by a recognized body in industry or academia and established by consensus in the FAA. [10] |
| ***Operation*** | A set of messages related to a single service action. (Adapted from [25]) |
| ***Organization*** | A unique framework of authority within which a person or persons act, or are designated to act, towards some purpose. Any department, service, or other entity within an organization which needs to be identified for information exchange. [7] |
| ***Output*** | Data transferred out of, or the process by which an information processing system or any of its parts transfers data out of, that system or part. (Adapted from [18]) |
| ***Payload*** | See message body. |
| ***Permissible Values*** | The set of allowable instances of a data element. [10] |
| ***Point-to-Point*** | A messaging model in which messages are routed to an individual consumer that maintains a queue of incoming messages. Each message is addressed to a specific queue, and the receiving clients extract messages from the queues established to hold their messages. While any number of producers can send messages to the queue, each message is guaranteed to be delivered to and consumed by one consumer. [7] |
| ***Protocol*** | A formal set of conventions governing the format and control of interaction among communicating functional units. [7] |
| ***Publish/Subscribe*** | A messaging model that supports publishing messages to a particular message topic. Subscribers may register interest in receiving messages on a particular message topic. [7] |
| ***Quality of Service (QoS)*** | A parameter that specifies and measures the value of a provided service. [7] |

| | |
|---|---|
| ***Queue*** | A staging area that contains messages that have been sent and are waiting to be read. [7] |
| ***Queue Receiver*** | A JMS Client that receives messages from a queue. [7] |
| ***Queue Sender*** | A JMS Client that sends a message to a queue. [7] |
| ***Real World Effect*** | An ultimate purpose associated with the interaction with a particular service. It may be the response to a request for information or the change in the state of some entities shared between the participants in the interaction. [7] |
| ***Security*** | The protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them. [17] |
| ***Security Mechanism*** | A process (or a device incorporating such a process) that can be used in a system to implement a security service that is provided by or within the system. [22] |
| ***Service*** | A mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. [7] |
| ***Service Consumer*** | An organization that seeks to satisfy a particular need through the use of capabilities offered by means of a service. [7] |
| ***Service Criticality*** | The level of significance given to a functional failure of a service. [10] |
| ***Service Description*** | The information needed in order to use, or consider using, a service. [7] |
| ***Service Interface*** | The means by which the underlying capabilities of a service are accessed. [7] |
| ***Service Provider*** | An organization that offers the use of capabilities by means of a service. [7] |
| ***Service-Oriented Architecture (SOA)*** | A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. A SOA provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations. [19] |
| ***Service Registry*** | An enabling infrastructure that uses a formal registration process to store, catalog, and manage metadata relevant to a service. A registry supports the search, identification, and understanding of resources, as well as query capabilities. [7] |
| ***Software Agent*** | A running program that drives services, both to implement them and to access them. [24] |
| ***Structured Data*** | Data that is organized in well-defined semantic “chunks” or units that are variously called fields, elements, objects, or entities. Individual units are often combined to form larger, more complex units. [10] |

| | |
|---|---|
| ***Synchronous*** | An interaction in which the participating agents must be available to receive and process the associated messages from the time the interaction is initiated until all messages are actually received or some failure condition is determined. [24] |
| ***Taxonomy*** | A system or controlled list of values by which to categorize or classify objects. [5] |
| ***Time to Live*** | A mechanism that limits the lifespan or lifetime of data in a JMS environment. (Adapted from [31]) |
| ***Topic*** | A distribution mechanism for publishing messages that are delivered to multiple subscribers. [7] |
| ***Topic Publisher*** | A JMS client that sends messages to a topic. [7] |
| ***Topic Subscriber*** | A JMS client that retrieves messages from a topic. [7] |
| ***Uniform Resource Identifier (URI)*** | A compact string of characters for identifying an abstract or physical resource. [23] |
| ***Uniform Resource Locator (URL)*** | A type of URI that identifies a resource via a representation of its primary access mechanism (e.g., its network "location"), rather than by some other attributes it may have. [23] |
| ***Unstructured Data*** | Data that does not follow any hierarchical sequence or any relational rules. Examples of unstructured data may include audio, video, and unstructured text such as the body of an e-mail or word processor document. [10] |
| ***User*** | A human, his/her agent, a surrogate, or an entity that interacts with information processing systems. [18] A person, organization entity, or automated process that accesses a system, whether authorized to do so or not. [15] |
| ***Web Service*** | A platform-independent, loosely-coupled software component designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format. Other systems interact with the Web service in a manner prescribed by its description by means of XML-based messages conveyed using Internet transport protocols in conjunction with other Web-related standards. [7] |

## 3.3 Acronyms and Abbreviations

| | |
|---|---|
| ***AIXM*** | Aeronautical Information Exchange Model |
| ***API*** | Application Programming Interface |
| ***DCMI*** | Dublin Core Metadata Initiative |
| ***DOT*** | Department of Transportation |

| | |
|---|---|
| ***DDD*** | Data Description Document |
| ***ESB*** | Enterprise Service Bus |
| ***FAA*** | Federal Aviation Administration |
| ***GML*** | Geography Markup Language |
| ***HTTP(S)*** | Hypertext Transfer Protocol (Secure) |
| ***ID*** | Identification |
| ***ISO/IEC*** | International Organization for Standardization/International Electrotechnical Commission |
| ***JMS*** | Java Message Service |
| ***JMSDD*** | Java Messaging Service Description Document |
| ***JNDI*** | Java Naming and Directory Interface |
| ***MOM*** | Message-Oriented Middleware |
| ***NAS*** | National Airspace System |
| ***NEMS*** | NAS Enterprise Message Service |
| ***OASIS*** | Organization for the Advancement of Structured Information Standards |
| ***OGC*** | Open Geospatial Consortium |
| ***PTP*** | Point-to-Point |
| ***QoS*** | Quality of Service |
| ***RFC*** | Request For Comment |
| ***SSL*** | Secure Sockets Layer |
| ***SOA*** | Service-Oriented Architecture |
| ***SWIM*** | System Wide Information Management |
| ***TCP/IP*** | Transmission Control Protocol/Internet Protocol |
| ***UML*** | Unified Modeling Language |
| ***URI*** | Uniform Resource Identifier |
| ***URL*** | Uniform Resource Locator |
| ***W3C*** | World Wide Web Consortium |
| ***WS*** | Web Service |
| ***XML*** | eXtensible Markup Language |

# 4 GENERAL REQUIREMENTS

This section describes requirements for the stylistic aspects of the JMSDD. Detailed requirements for the structure and content of the JMSDD are provided in section 5.

## 4.1 Text, Grammar and Style

**a.** The text SHALL be written in clear and simple language, free of vague terms, or those subject to misinterpretation.

**b.** All sentences SHALL be complete and grammatically correct. Refer to FAA Order 1000.36, FAA Writing Standards [1] for guidance.

**c.** The United States Government Printing Office Style Manual [4] SHALL be used as a guide for capitalization, spelling, punctuation, syllabification, compounding words, tabular work, and other elements of grammar and style.

## 4.2 Use of Diagrams

There are a number of sections in the JMSDD where using diagrams is suggested to enhance the understanding of a described topic.

**a.** Unified Modeling Language (UML) diagrams are RECOMMENDED since UML is able to concisely describe concepts without implying any specific technology. Information about UML diagrams is available at http://www.uml.org/.

# 5 DETAILED REQUIREMENTS

This section describes requirements for the structure and content of the JMSDD.

## 5.1 Cover Page

**a.** The JMSDD SHALL include a cover page as the first page.

**b.** The cover page SHALL include the FAA signature (the Department of Transportation triskelion figure with the words "U.S. Department of Transportation" and the words "Federal Aviation Administration" below it) in accordance with FAA Order 1700.6, FAA Branding Policy [2].

**c.** The line "Java Messaging Service Description Document" SHALL be centered above the title.

**d.** The title SHALL be the name by which the service will be known. NOTE: In most cases, the title will consist of the approved service's name issued by the activity authorized to assign the name. That name will be referred to throughout the JMSDD as the service name.

**e.** The cover page SHALL include the version of the service.

**f.** The cover page SHALL include the date on which the JMSDD was produced or generated.

While this standard does not prescribe configuration management policies, in most cases a JMSDD is assigned a document identifier by a governing or configuration management organization under whose authority the service is developed or functions. Refer to FAA Order 1800.66 Configuration Management policies [3] for information about when and how to obtain document identifiers for configuration-controlled documents

**g.** When a JMSDD has been assigned a document identifier for configuration management purposes, the document identifier SHALL be included on the cover page.

An example of a JMSDD cover page is shown in Appendix A.

## 5.2 Approval Page (Optional)

Signatures on this page ensure that the interested parties have approved the JMSDD content. The approval page may not be required based on the configuration management policies established within a given organization. The following statements apply when signed approval is required.

**a.** The approval page SHALL contain the centered line "Approval Signatures" above the list of cosigners.

**b.** The approval page SHALL include information for every cosigner.

**c.** The information SHALL include the cosigner's full name.

**d.** The information SHALL include the cosigner's organization code, i.e., the acronym by which the organization is commonly recognized within the FAA.

**e.** The information SHALL include the cosigner's signature.

**f.** The information SHALL include the date of the signature.

An example of a JMSDD Approval Page is shown in Appendix B.

## 5.3 Revision Record Page (Optional)

A revision record page may not be required based on the configuration management policies established within a given organization. The following statements apply when a record of revisions to the JMSDD is required.

**a.** The revision record page SHALL contain the centered line “Revision Record” above the list of revision records.

**b.** Only revisions SHALL be listed.

**c.** The revision record page SHALL include information for every revision listed.

**d.** The information SHALL include the revision letter or number.

**e.** The information SHALL include a brief description of the revision.

**f.** The information SHALL include the date of the revision.

**g.** The information SHALL include the full name of the person who entered this revision record (“Entered by”).

An example of a JMSDD Revision Record Page is shown in Appendix C.

## 5.4 JMSDD Structure

**a.** The JMSDD SHALL conform to the basic outline shown in Table I below. NOTE: the sections shown in italics are optional.

**TABLE I. JMSDD table of contents**

| |
|---|
| 5.2 Operations |
| 5.3 Messages |
| 5.4 Faults |
| 5.5 Data |
| 6 Service Implementation |
| 6.1 Bindings |
| 6.2 End Points |
| *Appendixes* |

## 5.5 Scope

**a.** Section 1 of the JMSDD SHALL provide a scope statement that is a clear, concise abstract of the coverage of the JMSDD.

**b.** Section 1 of the JMSDD SHOULD contain (but is not limited to) the following statement: "This Java Messaging Service Description Document (JMSDD) provides a description of the [service name]. The Information presented in the JMSDD is consistent with the requirements of Federal Aviation Administration (FAA) Standard Practice FAA-STD-073A, Preparation of Java Messaging Service Description Documents."

**c.** Section 1 of the JMSDD MAY include paragraphs on the service's purpose, applicability, background, etc. as needed to give readers of the JMSDD a context for understanding the body of the JMSDD.

## 5.6 Applicable Documents

The Applicable Documents section includes documents that are created at different stages of the service's lifecycle (e.g., service concept of operations, service requirements document, schemas, security approvals, impact analyses, etc.) as well as government or industry standards and protocols that are referenced in the JMSDD.

**a.** Section 2 of the JMSDD SHALL list all applicable documents.

**b.** Section 2 of the JMSDD SHALL present bibliographic information about each document listed.

**c.** The information SHALL include the full title of the document.

**d.** The information MAY include the alternate title or abbreviated name by which the document is known or recognized.

**e.** The information SHOULD include the publisher of the document.

**f.** The information SHOULD include the publication date of the document.

**g.** The information SHOULD include the appropriate version of the document (e.g., the latest version, the version needed for compatibility with other documents, the version of the document that is under contract by the project.)

**h.** The information SHOULD include a brief description of the document.

**i.** The information SHALL include the address or location (preferably a persistent Web location, i.e., URL) where a copy of the document can be obtained.

**j.** If a document is listed elsewhere in the JMSDD (e.g., a security protocol document in the JMSDD Security section; a data description document in the JMSDD Data section, etc.), listing the document again in section 2 is OPTIONAL.

## 5.7 Definitions

**a.** Section 3 of the JMSDD SHALL define all terms used in the JMSDD to provide for clarity, unless the terminology is generally accepted and not subject to misinterpretation.

**b.** Only terms that are specifically used in the JMSDD SHALL be listed in section 3.

**c.** Terms used in the JMSDD whose definitions are maintained in the *SWIM Controlled Vocabulary* [7] MAY be omitted from the Definitions section and referenced by providing hyperlinks to their definitions.

**d.** Section 3 of the JMSDD SHALL include a list of acronyms and abbreviations used in the JMSDD, together with their full spelling.

**e.** Only acronyms and abbreviations that are specifically used in the JMSDD SHALL be listed in section 3.

## 5.8 Service Profile

Section 4 of the JMSDD identifies and describes the service, its provider, its known consumers, its functional and non-functional characteristics, and constraints over its capabilities.

**a.** Section 4 of the JMSDD SHALL present information about the service profile.

**b.** The information SHALL include the name of the service.

**c.** The name SHALL be identical with the name of the service provided on the cover page of the JMSDD.

**d.** The information SHALL include a service identifier that uniquely identifies the service. For guidance on creating a service identifier, contact SWIM Governance. NOTE: The SWIM Governance team will support creation and maintenance of unique identifiers for services that are identified as part of the FAA SWIM implementation. Providers of other services should follow their appropriate organizational policy.

**e.** The information SHALL include a brief description of the service.

**f.** The information SHALL include a service version. For guidance on service versioning, see [6].

**g.** The information SHALL include a service interface type.

**h.** The single value representing the service interface type SHALL be selected from the Interface Type Taxonomy described in Appendix E of this standard.

**i.** The information SHALL include the type of messaging model deployed by the service. For more information on messaging models, see section 1.4.2 of this standard.

**j.** The single value representing the messaging model SHALL be selected from the Messaging Model Taxonomy described in Appendix E of this standard.

**k.** The information SHALL include the criticality level of the service.

**l.** The single value representing the service's criticality level SHALL be selected from the Service Criticality Taxonomy described in Appendix E of this standard.

**m.** The information MAY include the types of air traffic management (ATM) operations and services that the service supports.

**n.** One or more values representing an ATM service category MAY be selected from the ATM Service Category Taxonomy described in Appendix E of this standard.

**o.** The information MAY include the types of data products that the service delivers.

**p.** One or more values representing a service product category MAY be selected from the SWIM Service Product Category Taxonomy described in Appendix E of this standard.

It should be noted that the NSRR also maintains information about the service's current lifecycle stage as part of the service profile; however, this information is not replicated in the JMSDD itself due to its variable nature. The Lifecycle Stage Taxonomy is described in Appendix E of this standard.

### 5.8.1 Service Provider

This standard treats the service provider as an organization responsible for establishing and maintaining the service.

**a.** Section 4.1 of the JMSDD SHALL present information about the provider organization.

**b.** The information SHALL include the name of the organization.

**c.** The provided name SHALL consist of the full name spelled out followed by the acronym by which it is commonly recognized within the FAA.

**d.** The information MAY include a brief description of the organization.

**e.** The information MAY include an accessible reference (e.g., URL) for the Web page that supplies information about the service and/or organization.

**f.** The information SHALL include at least one point of contact, i.e., a person or group within the provider organization, suitable for making a human contact for any purpose.

**g.** The information SHALL include the full name of the contact.

**h.** The information SHALL include the contact's job title or a brief description of the contact's responsibilities.

**i.** The information SHALL include at least one telephone number for the contact.

**j.** The information SHALL include at least one e-mail address for the contact.

**k.** The information MAY include a postal address for the contact.

**l.** If the required point of contact information is maintained at a persistent Web location (URL), a hyperlink to this location MAY be provided in lieu of including the information itself.

**m.** Section 4.1 of the JMSDD MAY include additional points of contact, each documented as prescribed in requirements (g) through (l) above.

### 5.8.2 Service Consumers (Optional)

**a.** Section 4.2 of the JMSDD MAY present information about each known service consumer.

**b.** If service consumer information is presented, the information SHALL include the consumer organization's full name and acronym.

**c.** The information MAY include a brief description of the organization.

**d.** The information SHOULD include an accessible reference (e.g., URL) for the Web page that supplies information about the organization.

**e.** The information MAY include one or more points of contact, i.e., a person or group within the consumer organization, suitable for making a human contact for any purpose.

**f.** Each point of contact SHALL be documented as prescribed in section 5.8.1 requirements (g) through (l) of this standard.

### 5.8.3 Service Functionality

This standard asserts that every service represents a set of one or more identifiable business functions. The goal of section 4.3 of the JMSDD is to describe the business function(s) and the real world effects that result from invoking these business functions from a business point of view, that is, from the point of view of consumer organizations that will use the service to conduct their business. Section 4.3 should not address the mechanics of invocation (this aspect is addressed in the Service Interface section), but rather it should focus on answering the question of what the service does and what is the ultimate result of using the service. The ultimate result of using a service is referred to as the "real world effect". The real world effect may include:

"1. Information returned in response to a request for that information,
2. A change to the shared state of defined entities, or
3. Some combination of (1) and (2)." [19]

For example, a real world effect could be knowledge that "the flight has been rerouted" (change in the state) or a "weather forecast" (response to a request for information).

**a.** Section 4.3 of the JMSDD SHALL present information about the service functionality from the business perspective.

**b.** The information SHALL describe the service's business function(s) and real world effect(s).

**c.** The service's business function(s) SHALL be correlated with the real world effect(s).

### 5.8.4 Security

This standard defines security as collective measures that enable the service to provide protection against security threats. These threats may include (but are not limited to): unauthorized access to service information; unauthorized disclosure, modification and destruction of information; unknown status and repudiation in execution; and denial of service. To address the security threats, security mechanisms are utilized. Table II presents a list of the most typical security mechanisms commonly implemented by JMS applications, together with their intended purpose. NOTE: this list is neither exhaustive nor prescriptive.

**TABLE II. Java message service security mechanisms**

| *Mechanism* | *Purpose* |
|---|---|
| Authentication | To assure that system entities (individuals, agents, organizations, or processes) are who they claim to be. |
| Authorization | To assure that system entities have been granted the right or permission to access a service. |
| Integrity | To assure that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. |
| Confidentiality | To assure that information is not made available or disclosed to unauthorized system entities. |
| Non-Repudiation | To assure that the sender or recipient of a message cannot legitimately claim that they did or did not participate in the message exchange. |
| Audit | To record information needed to establish accountability for system events and for the actions of system entities that cause them. |

**a.** Section 4.4 of the JMSDD SHALL present information about the Java message service security.

**b.** Section 4.4 of the JMSDD SHALL describe all security mechanisms implemented by the JMS application.

**c.** Section 4.4 of the JMSDD SHALL specify all the security zones where the service is to be deployed, indicating whether it is provisioned in the "trusted" or "untrusted" regions. NOTE: provisioning in the untrusted region enables connectivity between NAS and Non-NAS for access to the service.

**d.** When a security mechanism is implemented by using a standard protocol or specification document, the information about this document SHALL be presented as prescribed in section 5.6 of this standard.

**e.** When the JMS client delegates one or more security measures to an external security service, the document that specifies the external security service SHALL be presented as prescribed in section 5.6 of this standard.

### 5.8.5 Qualities of Service

This standard defines Qualities of Service (QoS) as measurable characteristics that the service is expected to meet or possess. To be usable in practice, these QoS should be documented in a way that ensures clear and common understanding for the service stakeholders. Section 4.5 of the JMSDD lists all QoS parameters associated with the JMS client that produces messages.

**a.** Section 4.5 of the JMSDD SHALL list all Quality of Service (QoS) parameters associated with the provided service.

**b.** Section 4.5 of the JMSDD SHALL present information about each QoS parameter listed.

**c.** The information SHALL include the QoS parameter's name.

**d.** The information SHALL include the QoS parameter's value or range of values.

**e.** The information SHALL include the QoS parameter's definition. See Appendix G of this standard for guidance on writing good definitions to help ensure that the definition is as informative and understandable as possible.

**f.** The information SHALL include a description of how the values are measured or calculated.

**g.** The information SHALL include the unit of measure (e.g., seconds, percentage).

Appendix D of this standard contains examples of QoS parameters as they might be presented in the JMSDD. The parameters are commonly used in the NEMS environment.

### 5.8.6 Service Policies

This standard defines policies as constraints on the allowable actions of an agent or service consumer. The service policies can be described as a part of the JMSDD or, more frequently, in a separate document that contains common policies for a business or organizational domain.

**a.** Section 4.6 of the JMSDD SHALL provide information about policies that apply to the service.

**b.** When service policies are presented as a separate document, this policy document SHALL be presented as prescribed in section 5.6 of this standard.

**c.** If a referenced policy does not have a persistent Web location (URL), the policy SHALL be described in section 4.6 or included in an Appendix of the JMSDD.

### 5.8.7 Environmental Constraints

This standard understands environmental constraints as being characteristics of the "super-system", that is, the larger system within which the Java message service operates. Some examples of these constraints are: nature of the existing enterprise network (e.g., whether it is NAS or Non-NAS), firewalls, physical computing resources, etc.

**a.** Section 4.7 of the JMSDD SHALL describe all environmental constraints under which the service is operated and maintained.

## 5.9 Service Interface

Section 5 of the JMSDD provides detailed information about the types and content of messages that the service exchanges, message exchange models that the service deploys, and any conditions implied by these messages. NOTE: from the Java perspective, the interface is a programmatic construct that declares a set of methods that other Java classes may implement. However, this standard uses the term *service interface* in a broader sense as being "the means by which the underlying capabilities of a service are accessed". [19]

This standard also asserts that an interface of the service described in the context of a JMSDD is always defined in the JMS API specification [30].

**a.** Section 5.1 of the JMSDD SHALL specify the version of the JMS specification used by the service.

**b.** The specification used by the service SHALL be presented as prescribed in section 5.6 of this standard.

### 5.9.1 Interface

Section 5.1 of the JMSDD describes the interface deployed by the service.

**a.** Section 5.1 of the JMSDD SHALL present information about the interface being offered by the service.

**b.** The information SHALL include a description of the interface.

### 5.9.2 Operations

In the context of this standard, an *operation* is understood to be a single logical transmission between JMS clients. An operation identifies the sequence and cardinality of messages sent and/or received as well as the client they are logically sent to and/or received from.

In scenarios when a JMS client sends a message and expects to receive a message in return, synchronous request-response messaging can be used. This request-response messaging is often understood to be a subset of the PTP model. And, although JMS does not explicitly support request-response messaging, its API provides methods that allow for its implementation.

**a.** Section 5.2 of the JMSDD SHALL present information about each operation that the service implements.

**b.** The information SHALL include a name that uniquely identifies the operation throughout the JMSDD.

**c.** The information SHALL include a brief plain language summary of the pattern and goals of the actions that constitute the operation. For example, “allows client to retrieve current status of a specified flight”.

**d.** The information SHALL state if the operation is “synchronous” or “asynchronous”.

**e.** The information SHALL state if the operation is “idempotent” or “non-idempotent”.

**f.** The information SHALL list the input message(s) generated by a message consumer (either queue receiver or topic subscriber).

**g.** The information SHALL list the output message(s) produced by the message producer (either queue sender or topic publisher).

**h.** The information SHALL list the fault message(s) produced in response to conditions that result in operation failure. NOTE: errors generated by the message producer are described in section 5.9.4 of the JMSDD.

**i.** The information SHOULD include a diagram that shows how and in what order messages are exchanged within the context of the operation. Using Unified Modeling Language (UML) diagrams is a RECOMMENDED method for concisely describing concepts without implying a specific technology.

**j.** If the service does not deploy request-response messaging, the items called for in requirements (g) through (l) MAY be omitted and the phrase, “*This service does not deploy request-response messaging*” inserted in section 5.2 of the JMSDD.

For the purpose of this standard, *processing* is defined as a set of algorithms, calculations, or business rules that operate on input data in order to produce the required output or to produce a change of internal state. Processing might include actions such as the transformation of produced data by applying algorithms or business rules for compression or filtering.

**k.** The information SHALL include a description of the processing that takes place within the operation.

### 5.9.3 Messages

Section 5.3 of the JMSDD lists and describes all of the messages exchanged between the JMS clients.

**a.** Section 5.3 of the JMSDD SHALL list all messages produced or published by the service. (NOTE: fault messages are listed separately as described in section 5.9.4 below.)

**b.** Section 5.3.1 of the JMSDD SHALL present information about each message listed.

**c.** The information SHALL include a name that uniquely identifies the message throughout the JMSDD.

**d.** The information SHALL include a plain language description of the message. See Appendix G of this standard for guidance on writing good definitions to help ensure that the description is as informative and understandable as possible.

**e.** When deploying request-response messaging, the information SHALL indicate whether the message direction is “in” (provides input) or “out” (provides output).

**f.** When a message ID is assigned by the message producer, the information SHOULD include the message ID.

**g.** The information SHOULD include identification of the topic or queue to which the producer sends the message. (NOTE: identifying the message’s destination could be very valuable for those JMS clients that consume messages from more than one topic or queue.)

**h.** The information SHALL include the message delivery mode, indicating whether it is "persistent" or "nonpersistent". See Appendix E of this standard for details.

**i.** The information SHOULD include the message’s time to live, i.e., the length of time before the message expires.

**j.** The information SHALL include the message body (payload) type.

**k.** The single value representing the message body type SHALL be selected from the Message Body Type Taxonomy described in Appendix E of this standard.

**l.** When a message priority is assigned by the message producer, the information SHOULD include the message priority, indicated as a numeral ranging from 0 to 9. (NOTE: priorities 0–4 are usually categorized as “normal” priorities and 5–9 as “expedited” priorities.)

Each JMS message contains a built-in facility for supporting message producer-defined properties. Properties provide an efficient mechanism for supporting application-specific message filtering and routing.

**m.** Section 5.3 of the JMSDD SHALL list all properties defined by the message producer.

**n.** The information for each property SHALL include a name of the property.

**o.** The information for each property SHALL include a description of the property.

**p.** The information for each property SHALL include a list of permissible values and a description for each value.

**q.** The information for each property MAY include a description of its use for routing and filtering purposes.

Appendix F of this standard contains an example of a producer-defined application-specific property.

### 5.9.4 Faults

Section 5.4 of the JMSDD lists and describes all of the faults (errors)[1] that are generated in response to conditions that result in failure of an operation or set of operations.

**a.** Section 5.4 of the JMSDD SHALL list all custom errors generated by the message producer.

**b.** Section 5.4 of the JMSDD SHALL present information about each error generated by the message producer.

**c.** The information SHALL include each error code assigned to the error by the message producer.

**d.** The information SHALL include the text of the error message that is made visible to a message consumer.

**e.** The information SHALL include a plain language description of the reason for the error.

### 5.9.5 Data

Section 5.5 of the JMSDD provides detailed information (i.e., metadata) about the data that constitute a message's body or payload. For the purpose of this standard, two categories of data have been identified: *structured* (e.g., XML documents) and *unstructured* (e.g., images, binary-encoded documents). Among the SWIM-enabled systems, exchange of structured data, serialized as an XML document, is by far the most popular solution. For this reason, this standard focuses on structured data (requirements (**a**) through (**s**)), but it also addresses unstructured data (requirements (**t**) and (**u**)).

**a.** Section 5.5 of the JMSDD SHALL list all data elements that are part of the message payload.

**b.** Section 5.5 of the JMSDD SHALL present information about each element listed.

**c.** If the information requested in requirements (d) through (s) is already available in suitable Data Description Documents, such as XML Schemas, Information Exchange Models, (e.g. AIXM, FIXM), or other descriptive documents, section 5.5 of the JMSDD MAY reference these Data Description Documents in lieu of listing the elements and presenting the information about each element. See section 5.9.5.1 of this standard for Data Description Documents suitability requirements.

[1] The term "fault" is not ordinarily used in the JMS environment. We use it here to make the JMSDD's structure and nomenclature consistent with the Web Services Description Document and the Service Description Conceptual Model [5].

**d.** The information SHALL include a namespace for the element. NOTE: if all elements in the list are defined in the same namespace, the namespace can be indicated once for the whole list.

**e.** The information SHALL include a name for the element that uniquely identifies it within its namespace.

**f.** The information SHALL include a plain language definition of the element. See Appendix G of this standard for guidance on writing good definitions to help ensure that the definition is as informative and understandable as possible.

**g.** The information SHALL include a description of the element's permissible values in one of the following forms: a range of numbers, a list of individual values, a reference to a source that lists the values (e.g., "FAA Order 7350.8 Location Identifiers"), or a textual description (e.g., "Not Applicable").

**h.** For elements whose values represent codes, the information SHALL include the meanings of the codes (e.g., "BR = Mist, VA = Volcanic Ash, DU = Widespread Dust, etc.").

**i.** For elements whose values represent quantitative measures, the information SHALL include the unit of measure (e.g., feet, kilograms, degrees Fahrenheit, dollars).

**j.** The information SHALL include the element's datatype.

**k.** For primitive elements, i.e., elements that are not composed of other elements, datatype SHOULD be denoted using the typing system defined in section 3.2 of the W3C XML Schema Part 2: Datatypes specification [27].

**l.** If another typing system is used, the information SHALL include an explanation of the system or a reference to a source that describes the system. NOTE: if all datatypes are denoted using this system, the explanation or reference may be made once for the whole element list.

**m.** The information MAY include the element's maximum length together with units of length (e.g., characters, bytes, etc.), if applicable.

**n.** If an element is to be rendered in a special format, the information SHOULD include a format string.

**o.** Regular expressions as defined in appendix "F" of the W3C XML Schema Datatypes specification [27] are RECOMMENDED for use in describing format strings.

**p.** If another method is used to express format strings, the information SHALL include an explanation of the method or a reference to a source that describes the method. NOTE: if the method is used to express all format strings, the explanation or reference may be made once for the whole element list.

**q.** The information SHALL include the obligation ("Required" or "Optional") of the element, i.e., whether the element is required or optional in the context of its underlying information model.

**r.** The information SHALL include the multiplicity (occurrence) of the element in the context of its underlying information model (e.g., 0, 1,..., unbounded).

**s.** Section 5.5 of the JMSDD SHALL include a diagram, or a persistent Web location (URL) of a document or artifact containing such a diagram, that depicts a conceptual or logical model of the data elements listed.

**t.** If the data that appears in messages is unstructured, section 5.5 of the JMSDD SHALL provide the type, format, and plain language description of the data.

**u.** If the data that appears in messages is unstructured, section 5.5 of the JMSDD SHALL refer to the data protocol provided in section 6.1 of the JMSDD that describes how the data should be encoded (see also section 5.10.1.1 of this standard).

#### 5.9.5.1 Referencing Data Description Documents

Sometimes a document that describes the data exchanged by JMS clients is produced separately, often by an organization other than the organization responsible for developing the JMSDD. For the purpose of this standard, such a document will be referred to as a Data Description Document (DDD). Usually a DDD is developed for use by multiple services and not just for the service described in the JMSDD. If a DDD is to be used as a substitute for the content prescribed above for section 5.5 of the JMSDD, the DDD must comply with following requirements:

**a.** The DDD SHALL conform to the requirements (**d**) through (**s**) specified in section 5.9.5 of this standard.

**b.** The DDD SHALL maintain a versioning policy.

**c.** The DDD SHALL have a persistent Web location (URL).

**d.** The DDD specified in section 5.5 of the JMSDD SHALL be documented as prescribed in the Applicable Documents section 5.6 of this standard.

## 5.10 Service Implementation

Generally, implementation of JMS providers and clients is always constrained or enabled by a vendor-provided set of protocols. Section 6 of a JMSDD presents information about the software product that is used for developing and operating the service, as well as details about how the service can be accessed, i.e., information about data, transport, and message protocols as well as network address.

**a.** Section 6 of the JMSDD SHALL present information about the implementation of the service.

**b.** The information SHALL include the name of the product being deployed as a JMS provider.

**c.** The information SHALL include the version of the product being deployed as a JMS provider.

### 5.10.1 Bindings

For the purpose of this standard, a binding is understood to be a named collection of protocols which are used in the course of an execution of the service.

The binding concept is depicted in Figure 5.

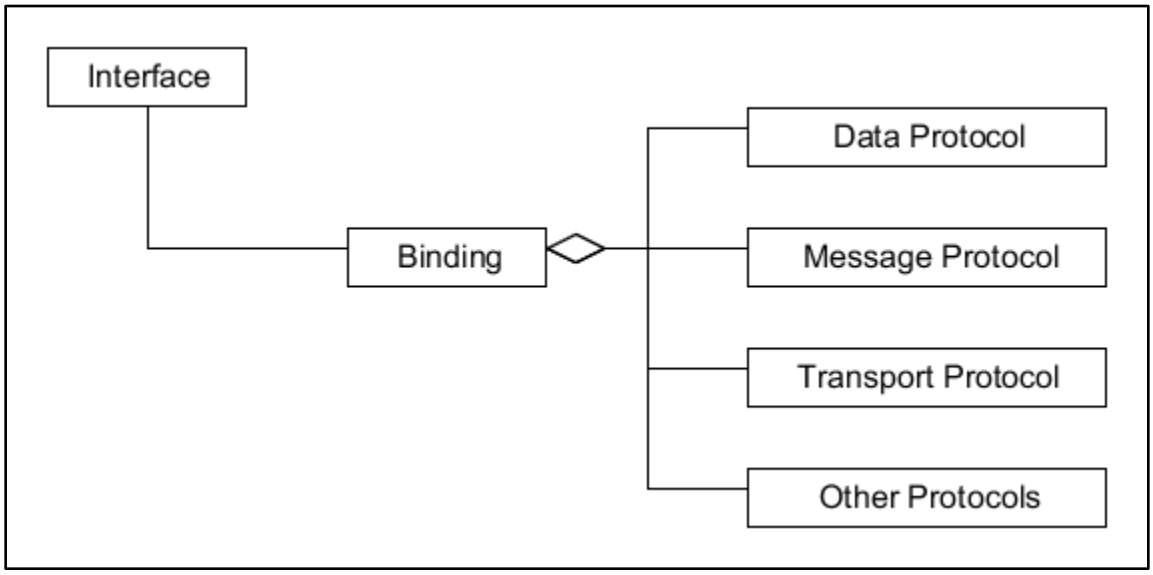


**FIGURE 5. Binding - conceptual diagram**

**a.** Section 6.1 of the JMSDD SHALL list all bindings implemented by the service.

**b.** Section 6.1 of the JMSDD SHALL present information about each binding listed.

**c.** The information SHALL include the name that is used throughout the JMSDD to refer to that binding.

**d.** The information SHALL include a description of the binding.

**e.** When the organization responsible for supporting and/or operating the NEMS provides a formal document that specifies information about the NEMS protocols, that document MAY be referenced in lieu of presenting binding information as prescribed in requirements (g) through (l).

**f.** The formal document specifying the NEMS protocols SHALL be presented as prescribed in section 5.6 of this standard.

Requirements (g) through (l) below pertain to bindings which are not supported by the NEMS.

**g.** The information SHALL include the name of the operation, and the name of an interface associated with this operation, that deploys this binding. NOTE: when all operations within the interface deploy the same binding, only the interface name is required.

**h.** The name of the operation and/or associated interface SHALL be consistent with the operation and/or interface name established in section 5.2 and/or 5.1 respectively of the JMSDD.

**i.** The information SHALL specify the data protocol(s) that the service uses for this binding, as described in section 5.10.1.1 of this standard.

**j.** The information SHALL specify the message protocol that the service uses for this binding, as described in section 5.10.1.2 of this standard.

**k.** The information SHALL specify the transport protocol that the service uses for this binding, as described in section 5.10.1.3 of this standard.

**l.** The information SHALL specify any other protocols that the service uses for this binding, as described in section 5.10.1.4 of this standard.

#### 5.10.1.1 Data Protocol

In order to exchange data between SOA components, an agreed-upon format must be used. A data protocol is a set of rules governing data serialization and coordination for data exchange among components.

The XML format is the protocol most often employed for exchanging textual data in JMS implementations. Besides transmitting textual data, an important use for XML is serializing data structures according to domain-specific conceptual models; e.g., the Geography Markup Language (GML) used to serialize information about geographical features or the Aeronautical Information Exchange Model (AIXM) used for transmitting aeronautical information.

**a.** Section 6.1 of the JMSDD SHALL present information about the data protocol for this binding.

**b.** The information SHALL include the normative document that regulates the data protocol.

**c.** The normative document SHALL be described as prescribed in section 5.6 of this standard.

**d.** If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

**e.** The data protocol SHALL be compatible with the data defined in section 5.5 (“Data”) of the JMSDD.

#### 5.10.1.2 Message Protocol

In an enterprise messaging environment, the communication and interaction between components is performed by exchanging messages of predefined content. A message protocol is a formal set of rules and conventions governing procedure calls and responses among communicating SOA components.

**a.** Section 6.1 of the JMSDD SHALL present information about the message protocol for this binding.

**b.** The information SHALL include the normative document that establishes the message protocol.

**c.** The normative document SHALL be described as prescribed in section 5.6 of this standard.

**d.** If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

#### 5.10.1.3 Transport Protocol

A transport protocol is a formal set of rules governing message transmission and port handling among communicating SOA components. JMS as an abstract specification does not specify any transport protocol. Vendors supporting JMS implementations, like WebLogic JMS or ActiveMQ, usually offer multiple transports. In ActiveMQ, for example, the transport protocol can be TCP, SSL or HTTP.

NOTE: this standard asserts that all JMS clients require a trusted network connection and use TCP/IP as an underlying transport protocol.

**a.** Section 6.1 of the JMSDD SHALL present information about the transport protocol for this binding.

**b.** The information SHALL include the normative document that establishes the transport protocol.

**c.** The normative document SHALL be described as prescribed in section 5.6 of this standard.

**d.** If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

#### 5.10.1.4 Other Protocols

Some modern protocols may combine data definitions with messaging conventions or messaging and transport governing conventions and cannot be unambiguously classified as strictly a data, message or transport protocol. This section of the standard describes requirements for such protocols.

**a.** Section 6.1 of the JMSDD SHALL present information about any other protocols that the service uses for this binding and that cannot be clearly identified as a data protocol, a transport protocol, or a message protocol.

**b.** The information SHALL include the normative document that establishes each described protocol.

**c.** The normative document SHALL be described as prescribed in section 5.6 of this standard.

**d.** If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

### 5.10.2 End Points

In the context of this standard, an end point is understood to be an association between a fully-specified binding, as described in section 5.10.1 of this standard, and "a physical point at which a service may be accessed" [10], i.e., a network address, which can be a fully qualified URI or a JNDI reference.

NOTE: in the context of JMS, end points are not directly accessible to each other: a message producer does not send messages to a message consumer end point, it delivers messages to an abstract end point (a queue or a topic); by the same token, a message consumer does not fetch messages from a message producer end point, it obtains the messages from one or more end points (queues or topics) internal to the JMS provider.

The end point concept is depicted in Figure 6.

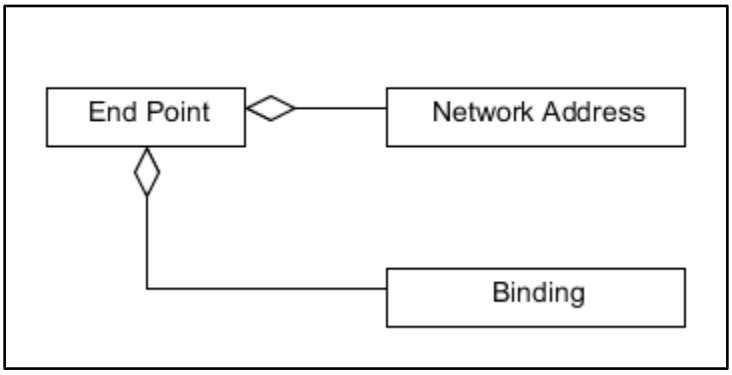


**FIGURE 6. End point - conceptual diagram**

**a.** Section 6.2 of the JMSDD SHALL list all end points implemented by the service.

**b.** Section 6.2 of the JMSDD SHALL present information about each end point listed.

**c.** The information SHALL include the name that is used throughout the JMSDD to refer to that end point.

**d.** The information SHALL include a description of the end point.

**e.** When the organization responsible for supporting and/or operating the NEMS provides the end point, an explanation of this fact SHALL be included in the description.

**f.** If the NEMS organization does not provide the end point, the information SHALL include the network address of the end point.

**g.** If the NEMS organization does not provide the end point, the information SHALL include the name of the associated binding established in section 6.1 of the JMSDD.

# APPENDIXES

## Appendix A. Example of a JMSDD Cover Page

NOTE: This Appendix is provided for guidance only; it is not normative.

**Generated by the NAS Service Registry and Repository (https://nsrr.faa.gov) on December 14, 2018**

U.S. Department of Transportation
**Federal Aviation Administration**

**FAA-X-XXX**
**Revision A**

**Java Messaging Service Description Document**

**Special Activity Airspace Management Service (SAAMS)**

**Service Version: 1.0.0**

## Appendix B. Example of a JMSDD Approval Signature Page

NOTE: This Appendix is provided for example only; it is not normative.

**Java Messaging Service Description Document**
**Special Activity Airspace Management Service (SAAMS)**
**Service Version: 1.0.0**

### Approval Signatures

| Name | Organi-zation | Signature | Date Signed |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

## Appendix C. Example of a JMSDD Revision Record Page

NOTE: This Appendix is provided for guidance only; it is not normative.

**Java Messaging Service Description Document**
**Special Activity Airspace Management Service (SAAMS)**
**Service Version: 1.0.0**

### Revision Record

| Revision Number | Description | Revision Date | Entered By |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## Appendix D. Examples of Quality of Service (QoS) Parameters

The table below provides examples of QoS parameters that are relevant to a message producer. JMSDD developers may reuse these parameters or provide their own, as well as their own values or range of values.

| QoS Parameter Name | Definition | Method | Unit of Measure |
|---|---|---|---|
| Availability | A measure of the lowest probability that a system or constituent piece will be operational during any randomly selected period of time, or, alternatively, the fraction of the total available operating time that the system or constituent piece is operational. | 100 * ((24 – Total Outage Time in Hours) / 24). Measurements are taken daily and apply to the preceding 24-hour period. | Percentage, accurate to 3 decimal places. |
| Mean Time Between Failure (MTBF) | Average time between hardware or software component failures that do not result in the loss of the service. | The sum of the individual times between noncritical failures divided by the number of noncritical failures. | Hours. |
| Mean Time To Restore (MTTR) | Average time required to localize a component failure, remove and replace the failed component, and to perform tests to confirm operational readiness of the component. | The sum of the individual times to repair divided by the number of repairs. | Hours. |

## Appendix E. Classification Schemes used in the NSRR

The following schemes are used to classify services and service artifacts in the NAS Service Registry/Repository, NSRR.

**Service Interface Type**[2] – A single value used to classify the service based on the kind of technological solution deployed by the service. Values are:

| | |
|---|---|
| **Method-oriented** | An interface that exposes service capabilities through a set of operations. Technologies that support this interface type are Web Service framework (WS*) and OGC Web Common Services. |
| **Message-oriented** | An interface that exposes service capabilities through creating, sending, receiving, and reading messages exchanged by distributed systems. The middleware technologies that support this interface type include Java Message Service (JMS) and .NET WCF. |
| **Resource-oriented** | An interface that supports the REST architectural style of interactions, that is, manipulation of XML representations of Web resources using a uniform set of stateless operations, usually a set of HTTP methods. |

**Messaging Model** – A single value used to classify a message-oriented service based on the type of messaging model it implements. Values are:

| | |
|---|---|
| **Publish/ Subscribe** | A messaging model that supports publishing messages to a particular message topic. Subscribers may register interest in receiving messages on a particular message topic. |
| **Point-to-Point** | A messaging model in which messages are routed to an individual consumer that maintains a queue of incoming messages. Each message is addressed to a specific queue, and the receiving clients extract messages from the queues established to hold their messages. While any number of producers can send messages to the queue, each message is guaranteed to be delivered to and consumed by one consumer. |

**Lifecycle Stage** – A single value used to classify the service based on its current Service Lifecycle Management Process (SLMP) stage. Values are:

| | |
|---|---|
| **Proposed** | The stage during which the business needs for the proposed service are identified and assessed as to whether needs can be met through the use of SOA. |
| **Definition** | The stage during which the service's business requirements are gathered and the service design is produced based on these requirements. |

[2] Service Interface Type is also one of a set of artifacts (see https://semantics.aero/) developed for supporting exploration and advancement of Semantic Web technologies in the international aviation community.

| | |
|---|---|
| **Development** | The stage during which the service specifications are developed and the service is built. |
| **Verification** | The stage during which the service is being inspected and/or tested to confirm that the service is of sufficient quality, complies with the prescribed set of standards and regulations, and is approved for use. |
| **Production** | The stage during which the service is available for use by its intended consumers. |
| **Deprecated** | The stage during which the service can no longer be used by new consumers. |
| **Retired** | The stage during which the service is disposed of and is no longer used. |

**Criticality Level** – A single value used to classify the service based on the level of significance given to a functional failure of the service. Values are:

| | |
|---|---|
| **Critical** | Loss of this service would significantly raise the risk associated with providing safe and efficient operations. |
| **Essential** | Loss of this service would raise the risk associated with providing safe and efficient operations to an unacceptable level. |
| **Routine** | Loss of this service would have a minor impact on the risk associated with providing safe and efficient operations. |

**ATM Service Category** – One or more values used to classify the service based on the strategic Air Traffic Management (ATM) operations and services it supports. Values are:

| | |
|---|---|
| **Flight Planning** | Operations and services that support the entry, update, and management of information that describes an intended flight or portion of an intended flight of an aircraft. |
| **Airport** | Operations and services that support the control of aircraft and vehicles on the airport surface. Such operations encompass movement from the gate or ramp to the runway at the departure airport and from the runway to the gate or ramp at the destination airport. |
| **Arrival and Departure** | Operations and services that support the control of aircraft in arrival and departure airspace, including departure operations to top of climb, arrival operations from top of descent to the airport surface, and transition flights. |
| **En Route Cruise/ Oceanic** | Operations and services that support the control of aircraft in that part of a flight in which aircraft are generally level at cruise altitudes, whether in domestic cruise airspace or oceanic airspace. |
| **NAS Management** | Operations and services that support the function of monitoring the National Airspace System (NAS) and taking appropriate action when abnormalities are detected. |

**SWIM Service Product Category –** One or more values used to classify the service based on the type of SWIM data product it delivers. Values are:

| | |
|---|---|
| **Aeronautical** | Data used to describe, manage and control aeronautical facts, concepts or instructions such as special use airspace restrictions, airport configuration, and Notices to Airmen (NOTAMS). |
| **Flight** | Data used to describe, manage, and control the safe movement of aircraft in the NAS, including information such as flight itinerary, flight identification, flight planning, flight events and status, and Air Traffic Management (ATM) control events that affect a single flight. |
| **Navigation** | Data used to locate the position and describe the course of aircraft. |
| **Surveillance** | Data produced by technologies (e.g., radar, beacon interrogator, automatic dependent surveillance-broadcast) for detecting and locating airborne and taxiing aircraft and ground support vehicles. |
| **Operation and Maintenance** | Data used to describe the status of communications and other equipment, systems, facilities, and maintenance schedules and requests. |
| **Weather** | Data used to describe current or predicted atmospheric conditions, including terminal and airborne weather observations, forecasts, and reports of weather phenomena. |

**Message Body Type –** A single value that indicates the nature of the actual (business) data transferred by the message. Values are:

| | |
|---|---|
| **Text** | The message body contains a text, e.g., XML. |
| **Stream** | The message body contains a stream of primitive values that are written and read sequentially. |
| **Map** | The message body contains a set of name-value pairs, where names are strings, and values are primitives. |
| **Object** | The message body contains a serialized object. |
| **Byte** | The message body contains an array of primitive bytes. |

**Message Delivery Mode** – A single value that describes the durability of a delivered message. Values are:

| | |
|---|---|
| **Persistent** | The “persistent” mode instructs the JMS provider to take extra care to insure the message is not lost in transit due to a JMS provider failure. A JMS provider must deliver a *persistent* message *once-and-only-once*. This means a JMS provider failure must not cause it to be lost, and it must not deliver it twice. [30] |

| | |
|---|---|
| **Non-persistent** | A JMS provider must deliver a "non-persistent" message *at-most-once*. This means that it may lose the message, but it must not deliver it twice. [30] |

## Appendix F. Example of Producer-Defined Message Property

The table below provides an example of a producer-defined message property as described in section 5.9.3 of this standard. NOTE: this example is non-normative.

| *Name* | *Description* | *Permissible Values* |
|---|---|---|
| FlightSensitivityType | The property, of the type String, that indicates the level of discretionary access to the flight information contained in the message. | **"military"** - a flight is considered to be a military flight if it is operated by any arm of the military of the United States or any other government.<br>**"sensitive"** - a flight is considered to be sensitive if the flight operator or owner requests the FAA to consider it to be sensitive. |

# Appendix G. Writing Good Definitions

NOTE: This Appendix is provided for guidance only; it is not normative.

The purpose of a definition is to define a concept with words or phrases that describe, explain, or make definite and clear its meaning. Precise and unambiguous definitions are one of the most critical aspects of ensuring interoperability. When two or more parties use a term, it is essential that all be in explicit agreement on the meaning of that term.

ISO/IEC 11179-4 [26] provides rules for writing good definitions. There are mandatory requirements with which all definitions must comply, and there are recommendations that should be followed when writing a definition. Note the difference between requirements and recommendations: compliance with the requirements can be objectively tested, whereas compliance with the recommendations can only be evaluated subjectively. The rules cited below are abstracted from this document.

***Requirements***

A definition *shall*:

1. Be stated in the singular.
2. State what the concept is, not only what it is not (i.e., never exclusively in the negative).
3. Be stated as a descriptive phrase or sentence(s).
4. Contain only commonly used abbreviations.
5. Be expressed without embedding definitions of other underlying concepts.

***Recommendations***

A definition *should*:

1. State the essential meaning of the concept.
2. Be precise and unambiguous.
3. Be concise.
4. Be able to stand alone.
5. Be expressed without embedding rationale, functional usage, domain information, or procedural information.
6. Avoid circular reasoning.
7. Use the same terminology and consistent logical structure for related definitions.

For further explanations of these rules as well as examples of definitions that pass and fail the tests, see the "Rules for Writing Good Definitions" section of the Guidelines for Using the SWIM Vocabulary located at https://semantics.aero/SWIM%20CV%20Guidelines.pdf.