



Federal Aviation
Administration

Introduction to Message Exchange Models

Presented to: EES

By: Leonid Felikson, Dovel Technologies



Agenda

- Introduction
- Basic Concepts
 - Service Oriented Architecture (SOA)
 - SOA Services
 - Web Services and JMS services
 - Messaging
- Message Models
 - Point-to-Point Model
 - Publish-Subscribe Model
- FAA Programs Examples
- Q&A



Basic Concepts: SOA

- http://www.youtube.com/watch?v=A3_QIYJRVvk

Service Oriented Architecture (SOA) is 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

(from FAA SWIM Controlled Vocabulary)

- SOA is a way of organizing IT assets, policies, practices and frameworks that enable application functionality to be provided and consumed as services that can be invoked, published and discovered.
- The key points:
 - SOA is an architecture style. It applies to both business and IT domains.
 - SOA is independent of any vendor, product or technology.
- SOA goals are inter-related and can be further categorized into two groups:
 - strategic goals
 - resulting benefits

Basic Concepts: SOA (cont.)

What is in common between SOA and Legos?

Interoperability

Each Lego block has standard bumps and connectors allowing it to connect to other blocks.

In SOA, standards-based interfaces are the small bumps that make compatibility possible and connect software to each other.

Lego blocks are also virtually **unbreakable**, because they have to withstand the destructive power of a frenzied toddler.

Similarly, SOA applications designed correctly provides a strong and robust software solution that can endure the pressures and needs of business.



Composability

Both SOA and Lego blocks exhibit the ability to add and subtract pieces to build your own creation.

SOA allows businesses to compose services into flexible applications that satisfy a company's specific needs.

Reusability

Lego blocks are reusable, which means when one creation has served its purpose, it can be torn down to build something else entirely.

Basic Concepts: SOA Service

- SOA infrastructure is a set of software products that enable realization of services, provide necessary life-cycle support for hosting and managing services, enable communication between services and ensure required security capabilities.
- An example of FAA SOA infrastructure components:
 - NAS enterprise messaging services (NEMS)
 - NAS Service Registry/Repository (NSRR)
- Main building block for implementing SOA is a **service**.
- A service is implemented as a software.
- A service can take variety of different styles and shapes.
- In FAA SWIM environment, there are 2 types of services:
 - Web services
 - JMS services

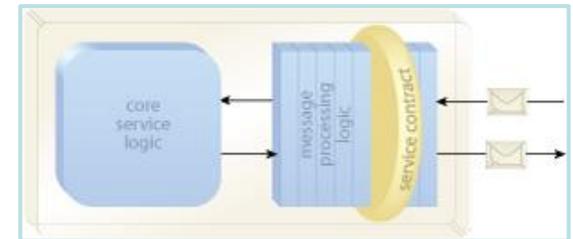
SOA is not = Web services or JMS services

Basic Concepts: Web Service

Web service is 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.

(from FAA SWIM Controlled Vocabulary)

- Web service is:
 - a software designed and developed in order to implement service oriented principles and architectural approach;
 - a software component implemented based on industry accepted standards.
- Web service interface is a self-contained description of service's and means of interacting and interoperating with it.



Basic Concepts: JMS Service

- JMS service ...:
 - ... is a software designed and developed in order to implement business capability using Java Messaging Service (JMS).
 - ... acts as a JMS client in message exchange model.
 - ... uses messaging transport protocol as oppose to HTTP protocol.
 - ... can exchange messages synchronously and asynchronously.
 - ... has a message structure that is defined by JMS API standard.



Web Service vs JMS Service

Web service

- uses SOAP standard message structure and HTTP transport protocol;
- its interface is defined by standard based (WSDL) service description document;
- its capabilities and qualities of service are described by FAA standard based Web Services Description Document (WSDD);
- allows primarily synchronous message exchange.

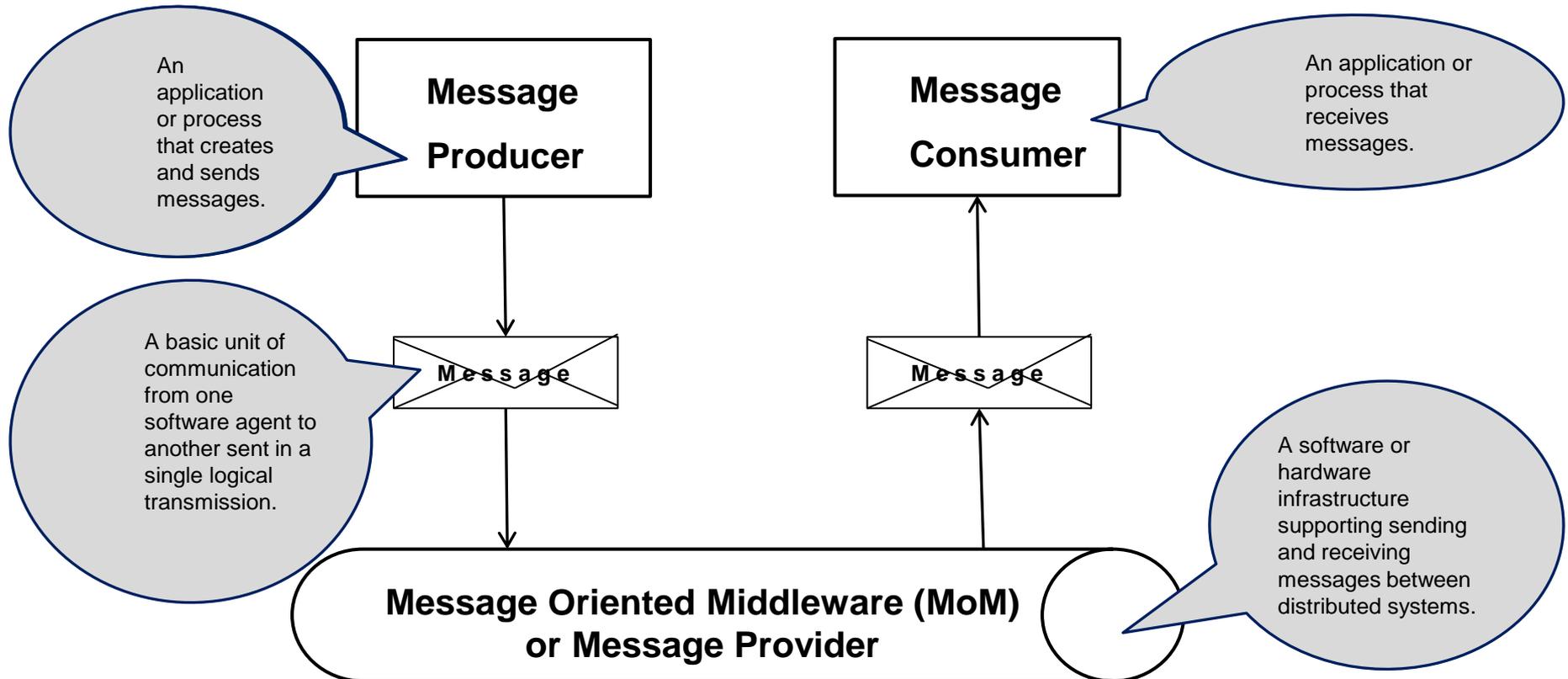
JMS service

- uses JMS based message structure, most often uses XML as message payload;
- always uses intermediary infrastructure, Message Provider, and acts as a JMS client in message exchange model;
- its capabilities and qualities of service are described by FAA standard based JMS Description Document (JMSDD);
- allows asynchronous message exchange.

Basic Concepts: Messaging

Messaging as a technology that enables high-speed, asynchronous, program-to-program communication with reliable delivery

(from FAA SWIM Controlled Vocabulary)



Basic Concepts: Messaging (cont.)

Java Message Service (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.

JMS Provider is a messaging system that implements the JMS application programming interface (API) in addition to the other administrative and control functionality required of a full-featured messaging product.

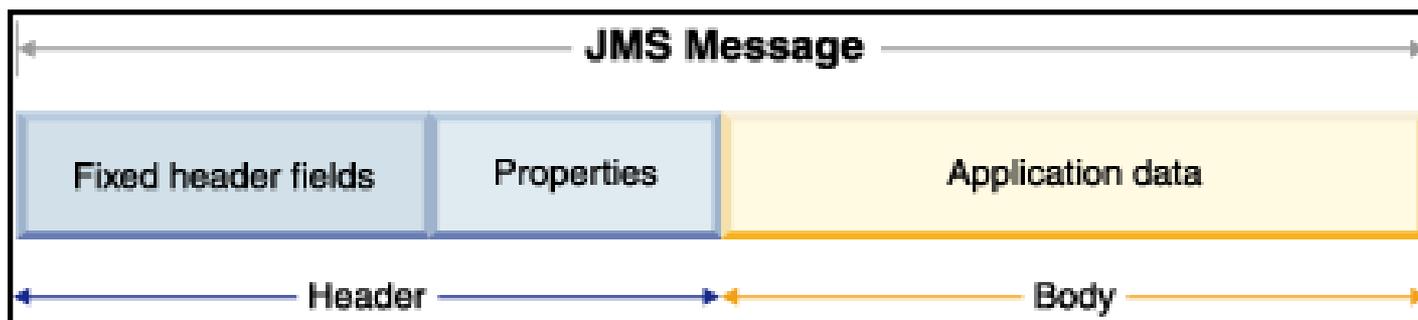
JMS Client is an application or process that produces and/or receives messages.

(from FAA SWIM Controlled Vocabulary)

Note. JMS Queue messages are processed only once but are not necessarily delivered in the order sent. Messages in the queue pool may be processed concurrently, resulting in a later message being received by a recipient before an earlier one.

Basic Concepts: Messaging (cont.)

JMS message structure includes message header (with message properties) and message body.



- **Message Header** contains values used by both clients and providers for identifying and routing messages
- **Message Properties** contain vendor-specific values and could also contain application-specific messages
- **Message Body (Payload)** contains the actual (business) data transferring the message

Messaging Models

The JMS specification supports two messaging models:

- **Point-to-Point (PTP) model**

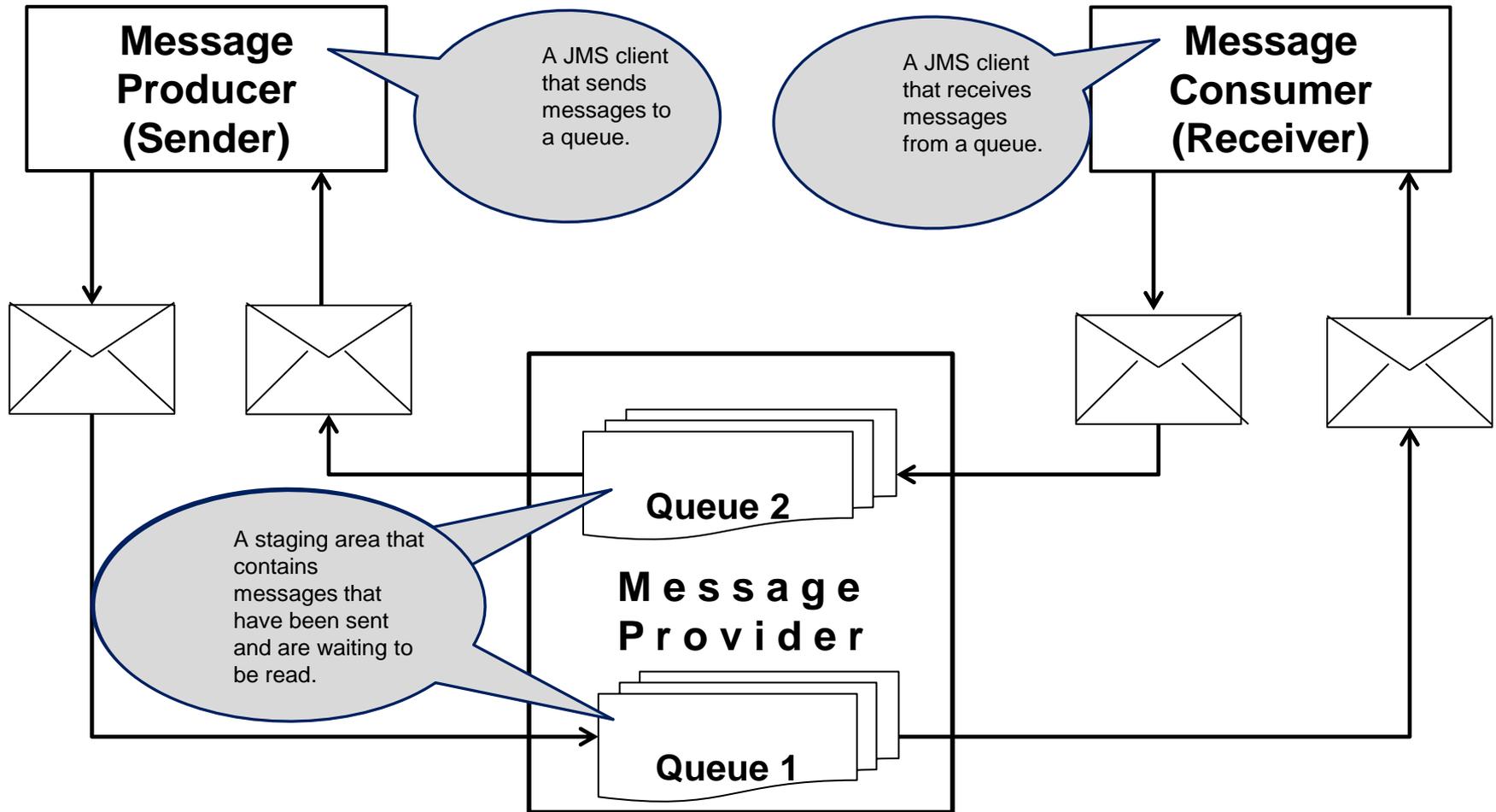
Each message is addressed to only one consumer. It allows users to send messages both asynchronously and synchronously using different channels (**queues**). Typically, in the PTP model, a user requests a message that a producer sends to the queue, rather than subscribing to a channel and receiving all messages sent on a particular topic.

- **Publish/Subscribe (Pub/Sub) model**

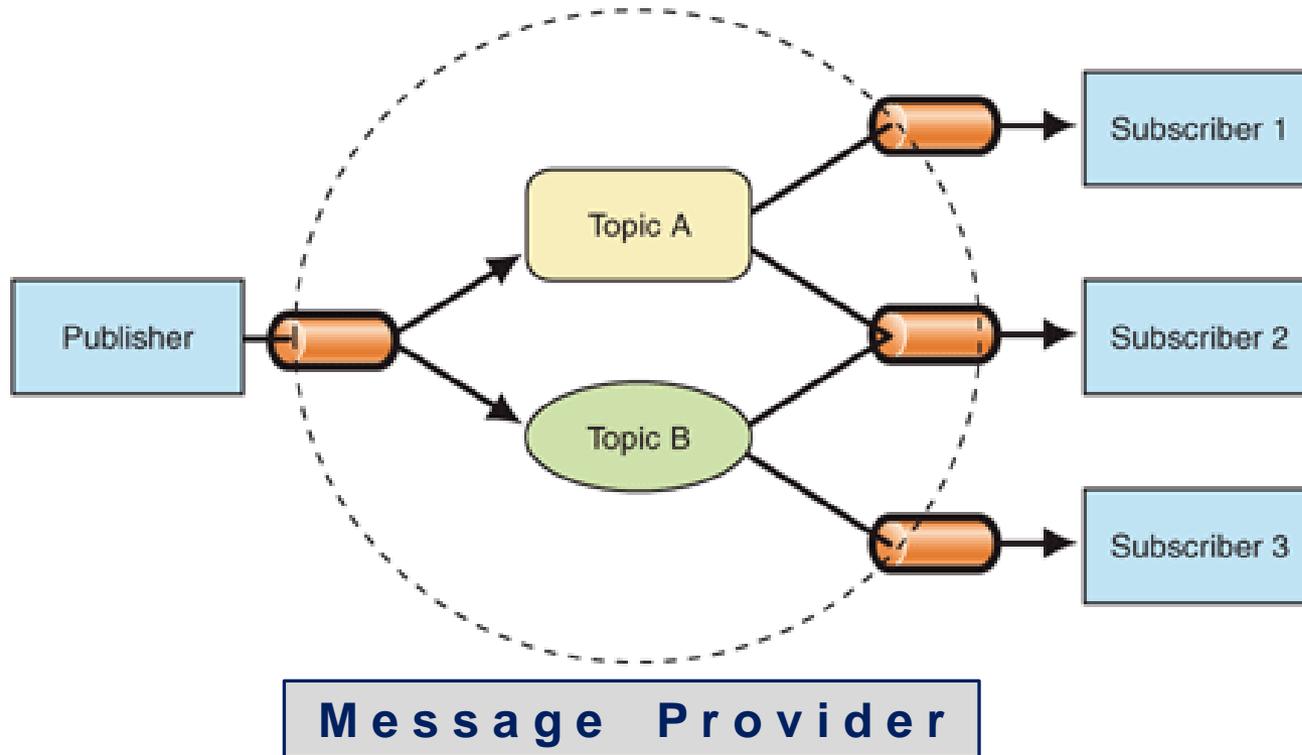
The pub/sub messaging model is similar to the notion of one-to-many relationships. It allows the producer to send messages to many users at one time. Consumers can subscribe to a particular **topic**, or channel, and receive all messages within the chosen topic. This model is always asynchronous.

Note: Unlike a queue, a topic is a distribution mechanism for publishing messages that are delivered to multiple subscribers.

Messaging Models, PTP



Messaging Models, Pub/Sub

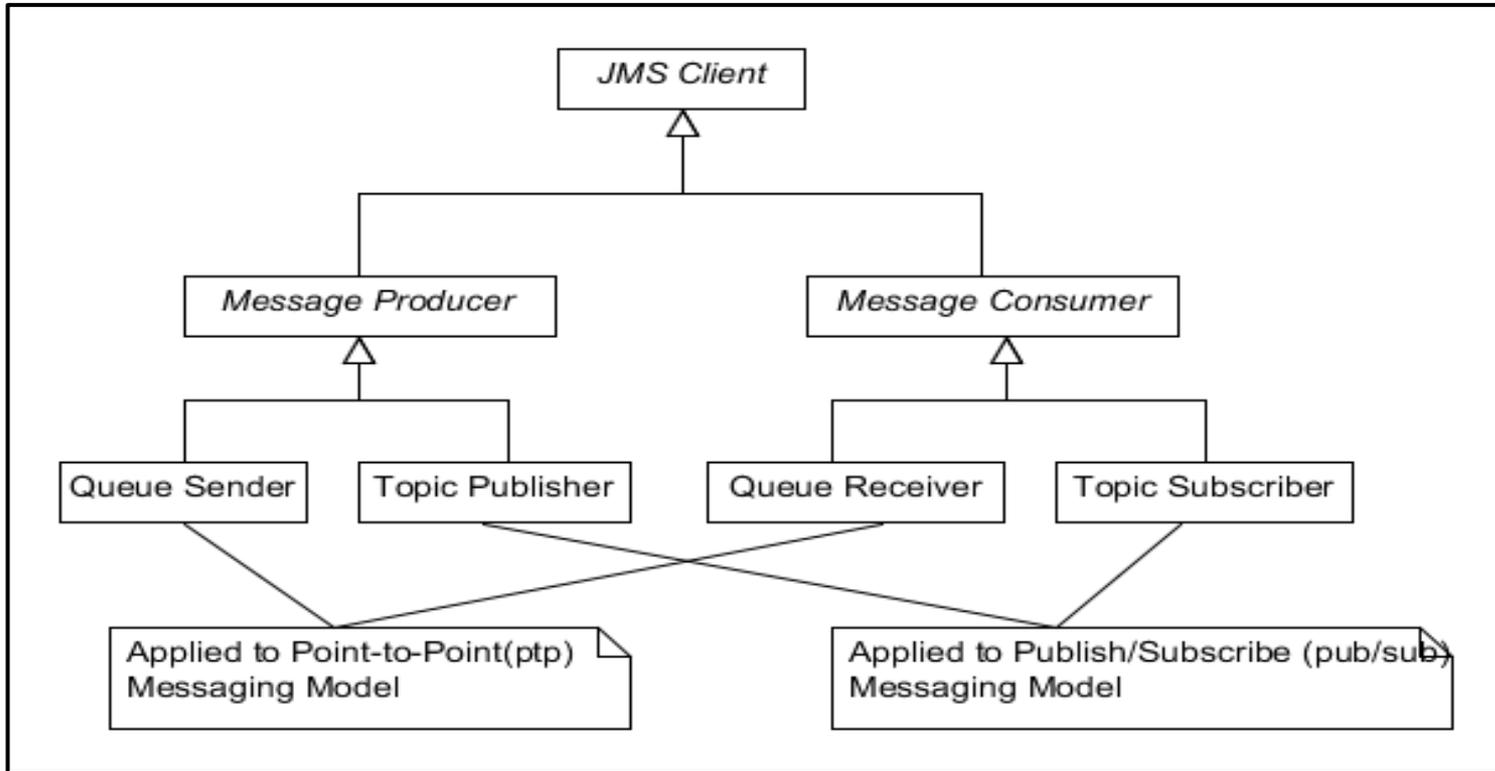


Note: A JMS client (Publisher) sends messages to a topic or topics.

Note: A JMS client (Subscriber) retrieves messages from a topic or topics.

Messaging Models (cont.)

The following diagram depicts relationship between different roles in Messaging:



AIM - Example 1

- **Aeronautical Information Management (AIM) Special Activity Airspace (SAA)**, SWIM Segment 1 Program, is using **Web services bases model** in majority of use cases
 - AIM SAA supports consumer-driven real time scheduling of SAA. This requirement necessitated the need to support synchronous communication between the scheduler's (clients) and the system of record (AIM SAA).
 - AIM required a well-defined service contract for the AIM SAA.
 - AIM required an interface that was compatible with widely used technologies and easy to interoperate with.
 - Because SOAP/HTTP communication protocol is based on a synchronous, contract-driven open standard for system communications, it was chosen as the transport for AIM SAA.
- There are use cases when AIM SAA needed a way to inform their consumers that a change had occurred. The publication of notices to these consumers necessitated an asynchronous messaging model to be used. That is why **JMS service model** was chosen because it offers a widely used and well understood mechanism for consumers to adopt and interoperate with.



STTDS – Example 2

- **SWIM Terminal Data Distribution System (STDDS)** provides Airport Data Service, Surface Movement Event Service and Tower Departure Event Service to consumers to receive runway visual range data, surface movement events, departure clearance event, taxi event and takeoff event.
- STDDS is designed with **JMS services based model**
 - Most of the STDDS message are small in size but the frequency is high so it makes sense to use JMS services, not Web services.
 - STDDS messages are feed based, so using a JMS based Pub/Sub messaging model with the STDDS service consumers subscribing to the feeds makes perfect sense.
 - STDDS also requires asynchronous messaging support, so that service consumers can consume data once they resume operation after any down time.



What Was Not Covered

- Web Service Proxy Model
- SWIM NEMS environment
- NEMS capabilities
- Enterprise Service Bus
 - Messaging Routing
 - Reliable Messaging
- SOAP messaging protocol
- WSDL standard for describing service capabilities



Q & A



References

- FAA STD-065, FAA Standard Practice, XML Namespaces
- FAA STD-073, FAA Standard Practice, Preparation of Java Messaging Services Description Documents
- SWIM Solution Guide for Segment 2a, MITRE, May 2013
- <http://www.enterpriseintegrationpatterns.com/>
- FAA SWIM Controlled Vocabulary,
http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techo/ato_comms_services/swim/vocabulary/#service-oriented-architecture

Contact Information

Mark Kaplun, SOA SME Lead, Mark.Kaplun@faa.gov

Leonid Felikson, SOA SME, Leonid.CTR.Felikson@faa.gov





Federal Aviation
Administration

Thank You!

