Agenda

• Service Oriented Architecture (SOA) Overview
• SOA Services: Web Services and JMS Services
• System Wide Information Management (SWIM)
• Break
• NAS Enterprise Management Services (NEMS)
• STDDS Program Using SWIM
• Q&A
Agenda

- **Service Oriented Architecture (SOA) Overview**
  - SOA services: Web services and JMS services
  - System Wide Information Management (SWIM)
  - NAS Enterprise Management Services (NEMS)
  - STDDS Program Using SWIM
  - Q&A
What is Service Oriented Architecture

- **Service-Oriented Architecture (SOA)** is a *software and architecture design pattern* based on a structured collection of discrete software modules known as *services*.
- SOA is a *set of principles and methodologies* for designing and developing software in the form of *interoperable services*.
- A *service* is the most *fundamental unit* of service-oriented logic, 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*.

SOA is something we *DO*, not something we buy! SOA is about *ARCHITECTURE*, not about technology!
SOA is Architectural Paradigm

Paradigms

- Characteristics
- Principles and Constraints
- Patterns and Anti-patterns

Models

- Capability
- Process
- Information
- Service

Structure and Organization

- Factoring and Granularity
- Topology
- Interdependencies
- Interfaces and Contracts
- Frameworks
- Implementation

Adapted from Gartner
The key differentiators of SOA

- SOA exhibits technology-neutral best practices.
- SOA is built on the standards leading to cost-effective implementations on a global basis with broad support by vendors.
- Services are “course-grained” and “loosely-coupled” allowing for much more flexibility than older technologies with respect to re-using and re_combining (composing) the services to create new business functions both within and across enterprises.
- SOA best practices create designs which embody business processes — and enhance the ability to outsource and extend processes to business partners.
- SOA encompasses legacy systems and processes so that the usefulness of existing investments can be preserved and even increased.
The key differentiators of SOA, cont’d

• SOA enables designing interoperable services for exchanging data.
• The *strategic nature* of service-oriented computing is one of its distinguishing characteristics. It contrasts the more tactical nature of traditional silo-based application development.
• Strategic goals of SOA:
  – Increased *Intrinsic Interoperability*
  – Increased *Federation*
  – Increased *Business and Technology Alignment*
  – Increased *Vendor Diversification Options*
  – Increased *ROI*
  – Increased *Organizational Agility*
  – Reduced *IT Burden*
# SOA Business Benefits and Outcomes

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Outcome</th>
<th>Achieved by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improve application quality</td>
<td>Increased revenue</td>
<td>Customer retention</td>
</tr>
<tr>
<td></td>
<td>Reduced cost</td>
<td>Employee productivity</td>
</tr>
<tr>
<td></td>
<td>Increased satisfaction</td>
<td></td>
</tr>
<tr>
<td>Reduce time-to-market</td>
<td>Increased revenue</td>
<td>Competitive advantage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reduced cycle time</td>
</tr>
<tr>
<td>Increase ease of doing business</td>
<td>Increased revenue</td>
<td>New customers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New products</td>
</tr>
<tr>
<td>Improve system consistency</td>
<td>Risk mitigation</td>
<td>Compliance</td>
</tr>
<tr>
<td></td>
<td>Cost avoidance</td>
<td>Reuse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Economies of scale</td>
</tr>
<tr>
<td>Reduced costs</td>
<td>Reduced costs</td>
<td>Operational efficiencies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reuse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Economies of scale</td>
</tr>
</tbody>
</table>

Adapted from Gartner
SOA Governance Overview

*Governance* means establishing and enforcing how people and solutions work together to achieve organizational objectives.

- **SOA Governance** is the application of *policies, rules, and standards* needed to ensure that all of the independent SOA-based efforts (whether in the design, development, deployment, or operations of a service) come together to meet *enterprise requirements*. (from SWIM Governance Policies Version 2.0)

- SOA governance is viewed as the application of Business governance, IT governance, and EA governance to SOA.
- SOA Governance ensures successful *business and IT alignment*. 
SOA Governance Overview, cont’d

- **SWIM Governance** is the realization of SOA Governance by the SWIM program, with the major goal of enabling a set of enforceable policies, procedures, processes, tools, and organizational activities that together ensure a consistent alignment between FAA/NAS business objectives and SOA best practices, methodologies, and technological solutions.

- SWIM SOA Governance processes are supported by use of the **NAS Service Registry/Repository (NSRR)**.

- NSRR is a SWIM-supported capability for making services visible, accessible, and understandable across the NAS.

- NSRR supports a flexible mechanism for service discovery, an automated policies-based way to manage services throughout the services lifecycle, and a catalog for relevant artifacts.
## Service Life Cycle Stages

<table>
<thead>
<tr>
<th>Lifecycle Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>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.</td>
</tr>
<tr>
<td>Definition</td>
<td>The stage during which the service's business requirements are gathered and the service design is produced based on these requirements.</td>
</tr>
<tr>
<td>Development</td>
<td>The stage during which the service specifications are developed and the service is built.</td>
</tr>
<tr>
<td>Verification</td>
<td>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.</td>
</tr>
<tr>
<td>Production</td>
<td>The stage during which the service is available for use by its intended consumers.</td>
</tr>
<tr>
<td>Deprecated</td>
<td>The stage during which the service can no longer be used by new consumers.</td>
</tr>
<tr>
<td>Retired</td>
<td>The stage during which the service is disposed of and is no longer used.</td>
</tr>
</tbody>
</table>
Service Oriented Architecture Overview, cont’d

• 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)

• Service design time policies are defined and enforced

• Industry and FAA standards are in-use and enforced. i.e.:
  – XML, SOAP, WSDL
  – FAA-STD-065A, Preparation of Web Service Description Documents
  – FAA-STD-066, Web Service Taxonomies
  – FAA-STD-073, Preparation of JMS Description Documents
Agenda

- Service Oriented Architecture (SOA) Overview
- **SOA Services: Web Services and JMS Services**
- System Wide Information Management (SWIM)
- NAS Enterprise Management Services (NEMS)
- STDDS Program Using SWIM
- Q&A
SOA Services Overview, cont’d

- SOA is architecture oriented toward **services**
- A **SOA service** is the most **fundamental unit** of service-oriented logic, 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**.

**SOA is not = Web services (or JMS services)**

Web services don’t require and are not necessary for SOA

**SOA or JABOWS?**
SOA Services Overview, cont’d

• The eight service-orientation design principles
  – Standardized Service Contract
  – Service Loose Coupling
  – Service Abstraction
  – Service Reusability
  – Service Autonomy
  – Service Statelessness
  – Service Discoverability
  – Service Composability
SOA Services Overview

SOA Services

Web Services

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.

JMS Services

A Java application or process that produces (Message Producer) and/or receives (Message Consumer) messages using Java Message Service (JMS) API.
SOA Services Overview, cont’d

- **Web service** is a software designed and developed in order to implement service oriented principles and architectural approach.
- Web service is a software component implemented based on industry accepted standards.
- Web service *interface* is a *self-contained description* of service and means of interacting and interoperating with it.
A **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.
# SOA Services Overview, cont’d

<table>
<thead>
<tr>
<th><strong>Web service</strong></th>
<th><strong>JMS service</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- uses SOAP standard message structure and HTTP transport protocol;</td>
<td>- uses JMS based message structure, most often uses XML as message payload;</td>
</tr>
<tr>
<td>- its interface is defined by standard based (WSDL) service description document;</td>
<td>- always uses intermediary infrastructure, Message Provider, and acts as a JMS client in message exchange model;</td>
</tr>
<tr>
<td>- its capabilities and qualities of service are described by FAA standard based Web Services Description Document (WSDD);</td>
<td>- its capabilities and qualities of service are described by FAA standard based JMS Description Document (JMSDD);</td>
</tr>
<tr>
<td>- allows primarily synchronous message exchange.</td>
<td>- allows asynchronous message exchange.</td>
</tr>
</tbody>
</table>
Web Services and JMS Service Description

• **Web Services Description Language (WSDL)** is a W3C recommendation
  – WSDL is an XML based protocol for information exchange in decentralized and distributed environments
  – WSDL definition describes how to access a web service and what operations it will perform

• **Web Service Description Document (WSDD)** is a Web service description that is rendered as a human-readable document in a way consistent with FAA acquisition process standards and practices.
  – The core of the WSDD consists of three parts: Service Profile, Service Interface, and Service Implementation. Each part represents important aspects of describing a Web service, and each can be characterized as answering a particular question
JMS Services Description

- JMS Service is described using **JMS Service Description Document (JMSDD)**.
  - JMSDD is JMS service description that is rendered as a human-readable document in a way consistent with FAA acquisition process standards and practices.
  - JMSDD consists of three parts: Service Profile, Service Interface, and Service Implementation. Each part represents important aspects of describing a JMS service, and each can be characterized as answering a particular question.
Agenda

- Service Oriented Architecture (SOA) Overview
- SOA Services: Web Services and JMS Services
- **System Wide Information Management (SWIM)**
  - NAS Enterprise Management Services (NEMS)
- STDDS Program Using SWIM
- Q&A
SWIM consists of *standards, infrastructure, and governance* that enable the management of Air Traffic Management related information and its exchange between qualified parties via interoperable services

- The SWIM Program implements SOA in the NAS.
- SWIM allows the FAA to create reusable system interfaces more quickly and cost effectively.
- SWIM facilitate interoperability and data-sharing needed for NextGen.
- SWIM replaces unique interfaces with modern standards-based data exchange.
- SWIM provides SWIM users access to information without directly connecting to another system.
- Provides enterprise security for incoming and outgoing data.
WHAT IS SWIM?

System Wide Information Management (SWIM) is the infrastructure that allows members of the Aviation Community to access the information needed to facilitate an innovative and efficiently run National Airspace System (NAS).

What is SWIM doing for the Aviation community?

SWIM makes it possible to have access to real-time, relevant information so users can respond faster, more accurately, creating collaboration opportunities with industry.

1. SWIM helps to reduce costs & increase agility for the Air Traffic Community.
2. DO MORE with LESS.
3. Enables the conversion from “data” to “information”

SWIM enables the efficient transition to global harmonization information standards.
System Wide Information Management, cont’d

SWIM operating environment: Data Exchange
(Published once, consumed by many)

Legacy operating environment: point-to-point connections
System Wide Information Management, cont’d

Mission Services/Application Layer:
FDPS, STDDS, ITWS, AIM, etc.

Common Support:
Data Standards & Harmonization

Enterprise Messaging:
SWIM Core Services

Physical Network:
FTI Operational IP network provides secure transport

Service Oriented Architecture Training Session for STDDS
November 2014
System Wide Information Management, cont’d

- SWIM data exchange capability is implemented via NAS Enterprise Messaging Service (NEMS)
- NAS Enterprise Security Gateway (NESG) supports SWIM data exchanges between NAS and non-NAS customers
- Service registration, discovery and lifecycle management is supported by the NAS Service Registry/Repository (NSRR)
SWIM Capabilities

Enterprise Messaging Services
- Provides a standards driven method of exchanging messages across a variety of NAS and non-NAS programs.
- Supports both JMS services & Web services.

Information Assurance
- NAS Enterprise Messaging Service (NEMS) deployment of the NESG meets NAS Boundary Protection Requirements.

SOA Suitability
- Provides assessment of FAA programs for SOA applicability and reports to NAS Technical Review Board (TRB).

SOA Governance
- Provides standards, policies, and procedural guidelines to support the functional requirements for implementing all aspects of service-oriented development.
- Making all services discoverable, searchable, and retrievable, through a formal cataloging process of service metadata, by establishing and operating the NSRR.
- Promoting interoperability among FAA systems by developing a common set of semantic and structural artifacts and promulgating them through communities of stakeholders.
Interoperability, discoverability and reuse are the key goals of **SWIM Governance**. To achieve these goals, SWIM Governance provides:

- **Standards**: FAA-STD-065A, FAA-STD-066, FAA-STD-073, etc.
- **Governance Policy** specifies applicable standards and procedures.
- **Service Lifecycle Management Process (SLMP)** describes the enforcement of SWIM Governance policies and standards from identification to retirement of a SOA Service. SLMP is synchronized with AMS milestones.
• Service registration, discovery and SLMP automation is supported by the NSRR
• NSRR is the authoritative source of all SWIM service metadata
  - Service metadata includes a set artifacts such as WSDL, XML schema, ConOps, SLA
  - Providers and Consumers must register in the NSRR
  - Registration of SWIM Services early in the lifecycle provides Consumers visibility into existing and future SWIM services
  - As the Service moves through the service lifecycle, the description in the NSRR becomes more refined, and supporting artifacts are added for consumer discovery
Agenda

• Service Oriented Architecture (SOA) Overview
• SOA Services: Web Services and JMS Services
• System Wide Information Management (SWIM)
• **NAS Enterprise Management Services (NEMS)**
• STDDS Program Using SWIM
• Q&A
NAS Enterprise Management Services (NEMS)

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

- **Message Producer**: An application or process that creates and sends messages.
- **Message Consumer**: An application or process that receives messages.
- **Message**: A basic unit of communication from one software agent to another sent in a single logical transmission.
- **Message Oriented Middleware (MoM) or Message Provider**: A software or hardware infrastructure supporting sending and receiving messages between distributed systems.
NEMS, cont’d

• The NAS Enterprise Messaging Service (NEMS) is an FTI Service, which provides for an Enterprise SOA messaging infrastructure for the NAS

• The NEMS supports two types of standards-based messaging exchange patterns: Publish/Subscribe and Request/Response

• Publish/Subscribe model is used when a Service Provider (Publisher) wants to continually publish data to multiple Service Consumers
  – The Publisher publishes a data stream to the NEMS, from which the Consumers consume data
  – The Publish/Subscribe model is typically used for services that require a constant, persistent feed of data
  – The Publish/Subscribe model is typically implemented using Java Message Service (JMS)
NEMS, cont’d

- The Request/Response model is better suited for services with data exchanges on an ad-hoc basis
  - Typically used when a Consumer sends a message to a Producer and expects to receive a message back from the Producer, such as an acknowledgement receipt or a reply message with a data payload
- The Request/Response model is typically implemented as SOAP-based Web services
- NEMS provides a Web service proxy capability and transports the data via HTTP protocol
### NEMS, cont’d

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.
NEMS, cont’d

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
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.
Message Producer (Sender)

A JMS client that sends messages to a queue.

Message Consumer (Receiver)

A JMS client that receives messages from a queue.

Queue 1

A staging area that contains messages that have been sent and are waiting to be read.

Queue 2

P2P Messaging Model
Pub/Sub Messaging Model

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

Note: A JMS client (Subscriber) retrieves messages from a topic or topics.
The following diagram depicts relationship between different roles in Messaging:
NEMS, cont’d

WS Proxy/Messaging Approach

Requests are load balanced to each managed server

Proxy facilitates:
- Authentication and Access Control
- Monitoring
- Mediation (future)

Web service Client

Load Balancer (future)

DMN

Blade A1
Proxy A

Blade B1
Proxy A

Blade C1
Proxy A

Domain Server

Web service A
NEMS Services & Service Lifecycle

System Level Capabilities

Basic
- B1: Run-time Subscription
- B2: Message Reliability QoS*
- B3: Mediation*
- B4: Availability & Performance*
- B5: Security Services*
- B6: Web Services*

Enhanced
- B7: Run-time Registry
- B8: Interoperability
- B9: Service Orchestration
- B10: Producer/Consumer SLAs*
- B11: Enterprise Repository
- B13: Global Load Balancer*
- B14: Local Load Balancers*

On-Ramping Services

Basic
- B1: Run-time Subscription
- B2: Message Reliability QoS*
- B3: Mediation*
- B4: Availability & Performance*
- B5: Security Services*
- B6: Web Services*

Enhanced
- E1: Mediation
- E2: Performance
- E3: Service Orchestration
- E4: SLAs
- E5: Security
- E6: Interoperability
- E7: Reports
- E8: Dynamic Subscriptions
- E9: Publications to R&D

Infrastructure Services
- DMN, DAN, DSN, vDEX, NESGs, GLB, LLB

Service Oriented Architecture Training Session for STDDS
November 2014
Global Load Balancer Capabilities

- Load balance traffic among multiple geographically distributed sites
- Select alternate NEMS sites in the event of site failures
- Direct specific clients to data centers based on configured policies
- Key Components
  - DNS—This front-end process focuses on the interaction with the DNS servers.
  - Probes—This back-end process gathers state and load information from devices within the data center such as local server load balancers.
  - Rules and associations—This process establishes a relationship between the front-end and back-end process.
  - Global load-balancing management policy and algorithms—These processes direct client traffic to the distributed data centers based on a defined policy.
Local Load Balancer Capabilities

• Load balance traffic among local server pools
• Select alternate server in the event of server failures
• Direct specific clients to servers based on configured policies
• Key Components
  – VIPS—Provide entry points to the load balancer
  – Probes/Monitors—This back-end process gathers state and load information from servers within the server pools.
  – Rules and associations—This process establishes a relationship between the VIPS and Health Checks.
  – Local load-balancing management policy and algorithms—These processes send client traffic to the servers within a server pool based on a defined policy.
• Status Update
  – Global and Local Load balancers are deployed and manageable in the NAS Ops today for NEMS
Global/Local Load Balancer Deployment

- OEX
  - DNS Server
  - Global Load Balancer

- ACY
  - DNS Server
  - Global Load Balancer

- ATL
  - DNS Server
  - Global Load Balancer

- SLC
  - DNS Server
  - Global Load Balancer

- ARTCC
  - Local Load Balancer

- ARTCC
  - Local Load Balancer

- ARTCC
  - Local Load Balancer

- ARTCC
  - Local Load Balancer

- ARTCC
  - Local Load Balancer
Nestor Logical Architecture for 2-way SOA Services

- **NAS Enterprise Security Gateway**

- **Internal DMZ**
  - Trusted DEX Messaging Node
  - Internal NEMS Messaging Node
  - WS Producer or Consumer

- **External DMZ**
  - Un-trusted DEX Messaging Node
  - JMS Producer or Consumer
  - WS Producer or Consumer

- **XML GW**

- **OSB**
  - Will host proxies for access to trusted DMN Services and proxies for access to un-trusted DMN services
  - Adds security protection and monitoring point

- **Internet**

- **JMS Producer or Consumer**

- **WS Producer or Consumer**

- **Internal DMZ**
  - Trusted DEX Messaging Node
  - JMS Producer or Consumer
  - WS Producer or Consumer

- **External DMZ**
  - Un-trusted DEX Messaging Node
  - JMS Producer or Consumer
  - WS Producer or Consumer

- **XML GW**

- **OSB**
  - Will host proxies for access to trusted DMN Services and proxies for access to internal NAS services
  - Adds security protection and monitoring point
NEMS Infrastructure Deployment

- Existing NEMS Nodes
- NEMS Gateway
- R&D and FNTB Nodes
- Planned NEMS Nodes 2014
- Mission Support Nodes (Admin)
- ARTCC Sites
- FTI Operations Center

Service Oriented Architecture Training Session for STDDS
November 2014
Agenda

- Service Oriented Architecture (SOA) Overview
- SOA Services: Web Services and JMS Services
- System Wide Information Management (SWIM)
- Break, 10:00 - 10:15
- NAS Enterprise Management Services (NEMS)
- **STDDS Program Using SWIM**
- Q&A
STDDDS Program Using SWIM

SWIM Terminal Data Distribution Service (STDDS) Deployment

Service Oriented Architecture Training Session for STDDS
November 2014
STDDS Current Architecture
STDDS Current System

STDDS interfaces with the following tower systems:

• Airport Surface Detection Equipment Model X (ASDE-X) - STDDS receives surface surveillance data, and sends the event data to NEMS via the Surface Movement Event Service (SMES). In addition, STDDS sends position report and generic flight plan data to NEMS via SMES.

• Runway Visual Range (RVR) - STDDS receives RVR observations and sends them directly to NEMS via the Airport Data Service (APDS).

• Electronic Flight Strip Transfer System (EFSTS) - STDDS receives controller entered departure events (e.g. taxi start, takeoff) and sends them directly to NEMS via the Tower Departure Event Service (TDES).

• Tower Data Link Services (TDLS) - STDDS receives clearance delivery information from TDLS and sends it as departure events directly to NEMS via the TDES.

• Airport Surface Surveillance Capability (ASSC) data delivered to NEMS from the ASSC system at SFO.

• STARS program to obtain TAIS data through the AIG processors.
Release 3 – Interfacing with NEMS

- ASDE-X
- RVR
- EFSTS
- TDLS

STDDS

SMES

APDS

TDES

STDDS (2 to N)

STDDS (N+1 to M)

STDDS (M+1 to 39)

Primary NEMS

Secondary NEMS

ActiveMQ

Weblogic

Primary NEMS

Secondary NEMS

External Client(s)

• Internal Client(s)

Publish to JMS Queue

Failover Links

Publish to Topics
JMS vs Web services

STDDS uses JMS and not Web services for the following reasons:

• Most of the STDDS message are small in size but the frequency is high so it makes sense to use JMS and not WS.
• STDDS messages are feed based so using a JMS pub/sub model makes perfect sense and have consumers subscribe to the feeds.
• STDDS system requirement was to provide asynchronous support so that users can consume data once they come back up in case of a down time.
Reconstitution of TDES Message

NEMS/STDDS TDES Service Recon Scenario

Security gateway

NAS End User

User sends Recon Request to NEMS
User consumes from Topic

Sends request

NEMS Queue: STDDSRecon Queue

Message dropped
Requests with invalid AirportID are not sent to STDDS

Valid AirportID

NEMS Topic: STDDSReconTopic

Int. DMZ Environment

STDDS consumes recon Request

STDDS Consumes recon request
STDDS sends Recon request status Message
STDDS sends Recon start Message
STDDS sends TowerFlight DataEvents
STDDS send Recon End Message

User consumes Recon TDES Messages

NEMS publishes to Topic

ReconstitutionRequestStatus

TDES-Recon.Topic01.OUT

Start Recon Message

TDES-Recon.IN

End Recon Message
NEMS/STDDS APDS Service Scenario

Non-NAS User

- RvR Data

Ext. DMZ

Security Gateway

STDDS.Topic 1.OUT

Consumes messages

APDS.IN

Consumes messages

Int. DMZ

NAS User

- RvR Data

STDDS Infrastructure

STDDS Publishes messages on to the NEMS Queue

RvR data
SMES ASDEX Service

NEMS/STDDS SMES Service
ASDEX Scenario

Non-NAS User (ex-Delta Airlines)

NAS User

Security Gateway

Ext. DMZ

Int. DMZ

STDDS.Topic 01.OUT

STDDS-ASDEX.IN

STDDS Publishes messages on to the NEMS Queue

Consumes messages

ASDEX Position Report
ASDEX Generic Flight Info

ASDEX Position Report
ASDEX Generic Flight Info

ASDEX Position Report
ASDEX Generic Flight Info
TDES Event Service

NEMS/STDDS TDES Service Scenario

Security Gateway

Ext. DMZ

Int. DMZ

NAS User

TowerFlight DataEvent

Consumes messages

STDDS.Topic 01.OUT

TDES.IN

STDDS Publishes messages on to the NEMS Queue

STDDS Infrastructure

TowerFlight DataEvent

Service Oriented Architecture Training Session for STDDS
November 2014
References

- http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/atc_comms_services/swim/
- http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/atc_comms_services/swim/vocabulary/
- http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/atc_comms_services/swim/governance/standards/

Contacts

Mark Kaplun, SWIM Governance Lead (mark.kaplun@faa.gov)
Leonid Felikson (leonid.CTR.felikson@faa.gov)
Amresh Das (amresh.CTR.das@faa.gov)
Q & A
Thank You!