



Federal Aviation
Administration

Service Oriented Architecture

Training Session

Presented to: STDDS Program

By: Leonid Felikson, Amresh Das
Veracity Engineering



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

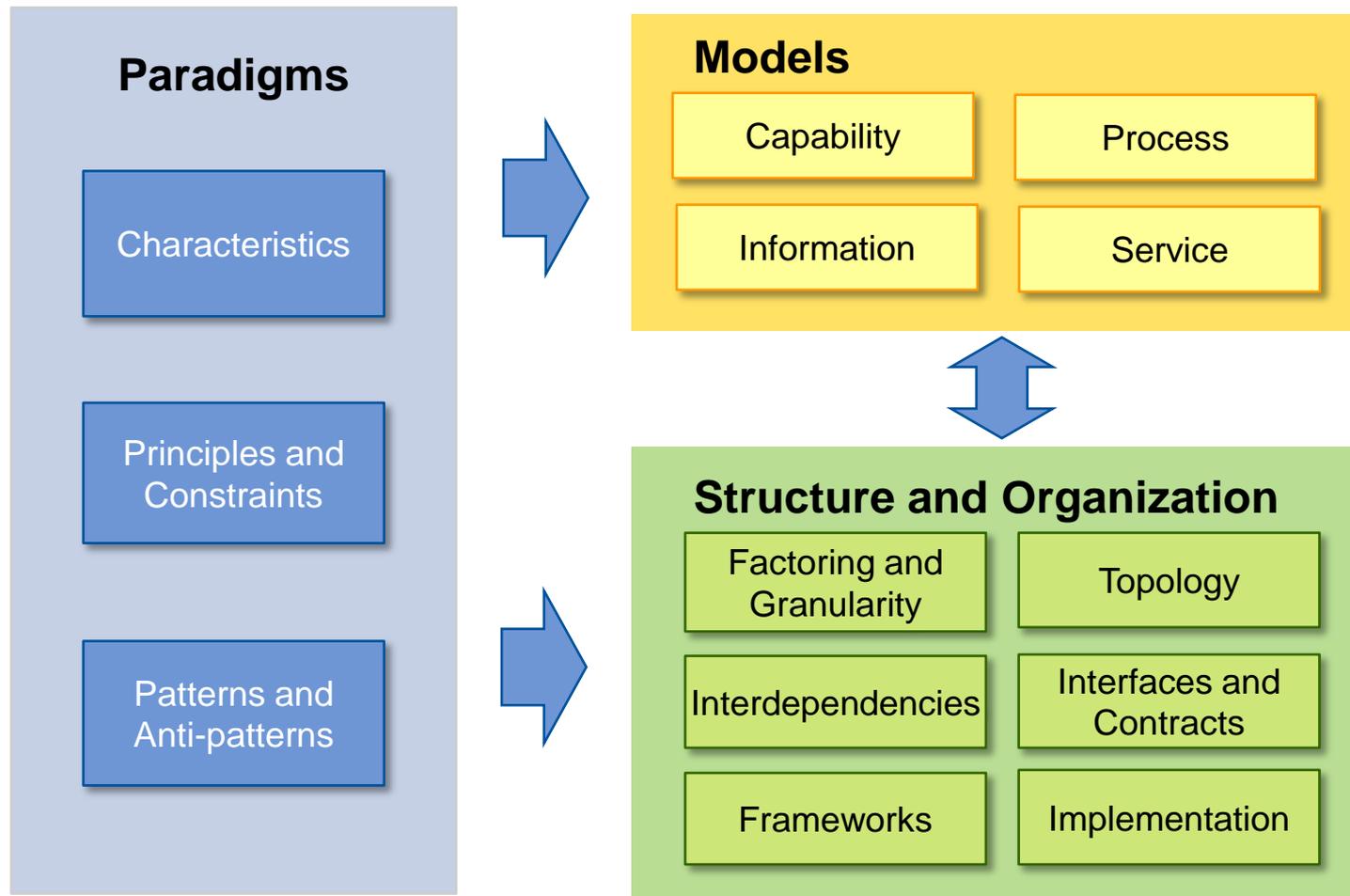


What is Service Oriented Architecture

- **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



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

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

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

Lifecycle Stage	Description
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.
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.

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

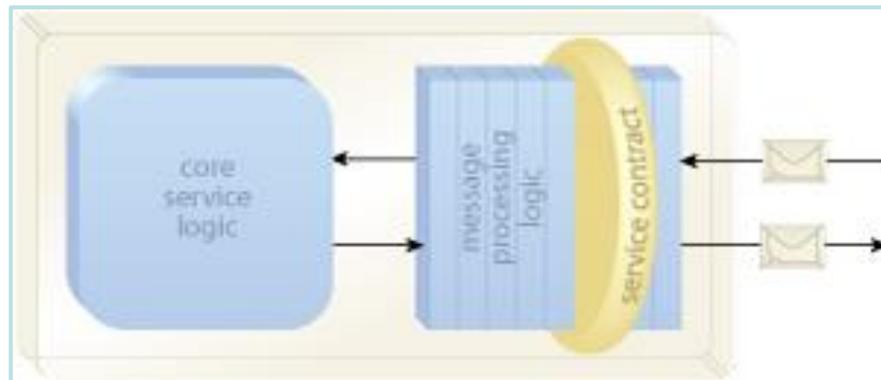


A *platform-independent, loosely-coupled* software component designed to support *interoperable* machine-to-machine interaction over a network. It has an *interface* described in a *machine-processable* format. Other systems interact with the Web service in a manner *prescribed by its description* by means of XML-based messages conveyed using Internet transport protocols in conjunction with other *Web-related standards*.

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.



SOA Services Overview, cont'd

- 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

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.



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



System Wide Information Management (SWIM)

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.

System Wide Information Management, cont'd

WHAT IS SWIM?

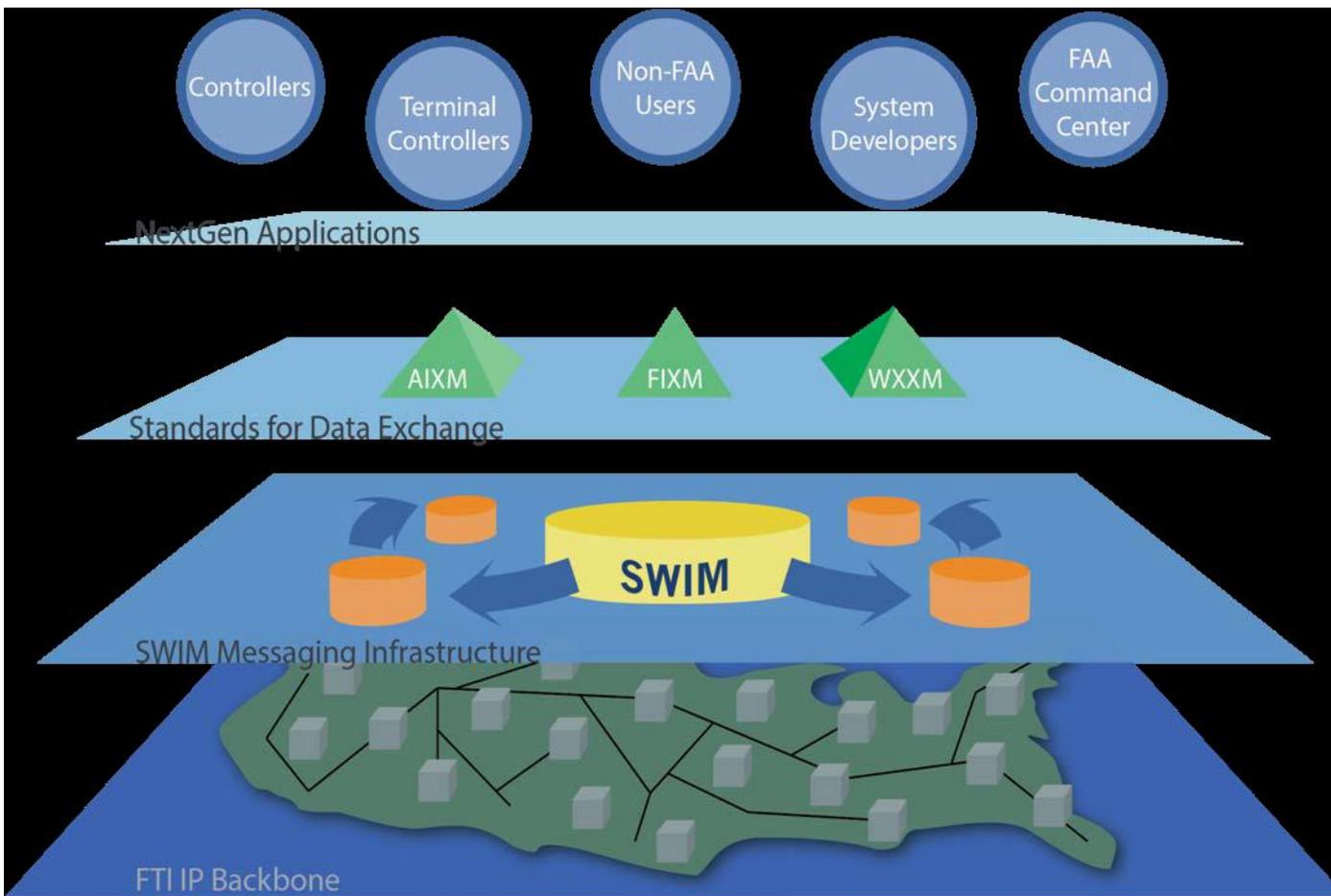
System Wide Information Management (SWIM) is the infrastructure that allows members of the Aviation Community to access the information needed to facilitate an innovative and efficiently run National Airspace System (NAS).

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.



System Wide Information Management, cont'd



Mission Services/ Application Layer:

FDPS, STDDS, ITWS, AIM, etc.

Common Support:

Data Standards & Harmonization

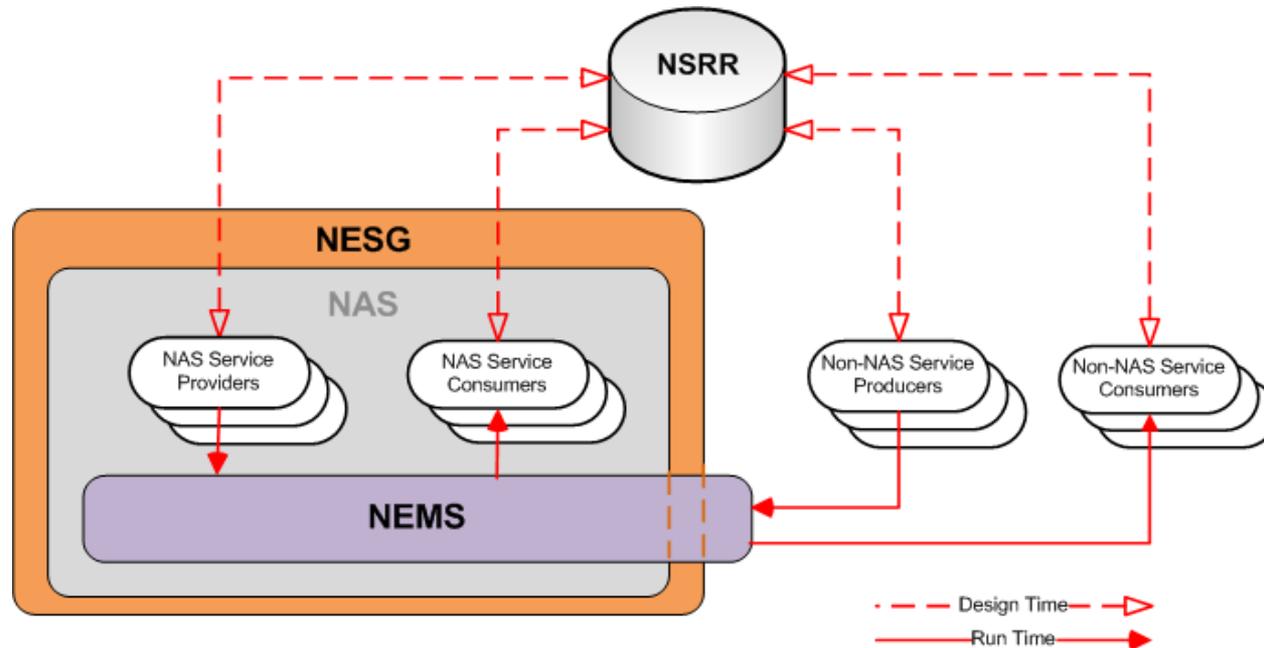
Enterprise Messaging:

SWIM Core Services

Physical Network:

FTI Operational IP network
provides secure transport

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.



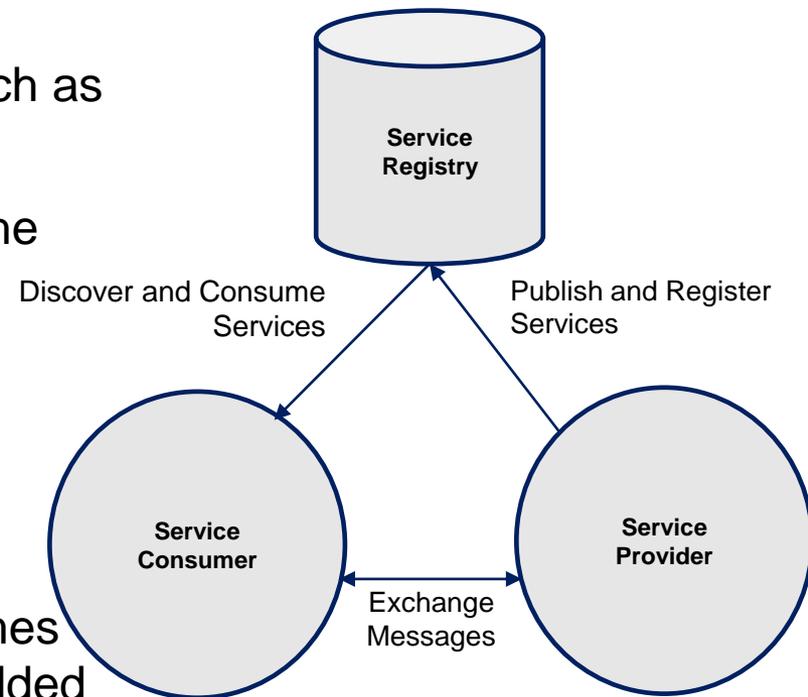
System Wide Information Management, cont'd

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.

System Wide Information Management, cont'd

- 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



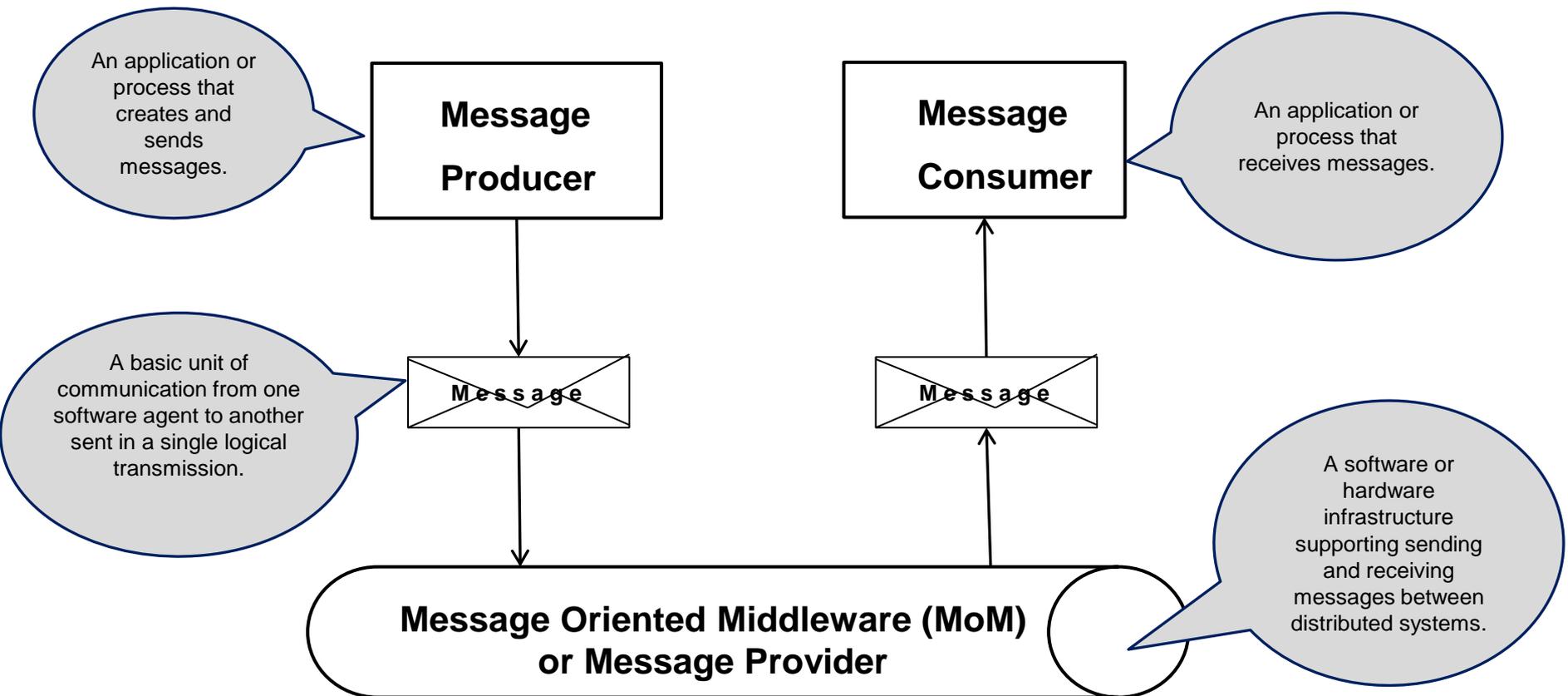
Agenda

- 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.



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.

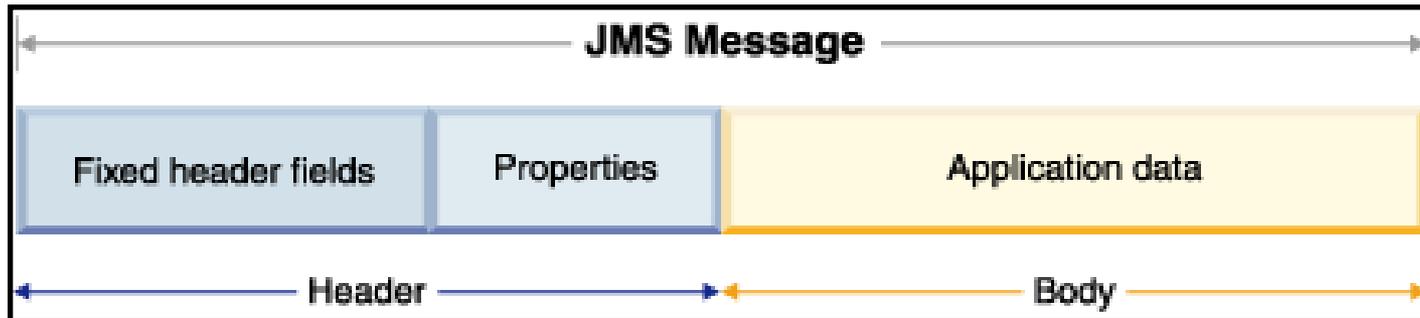
JMS Client is an application or process that produces and/or receives messages.

(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

NEMS, cont'd

The JMS specification supports two messaging models:

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

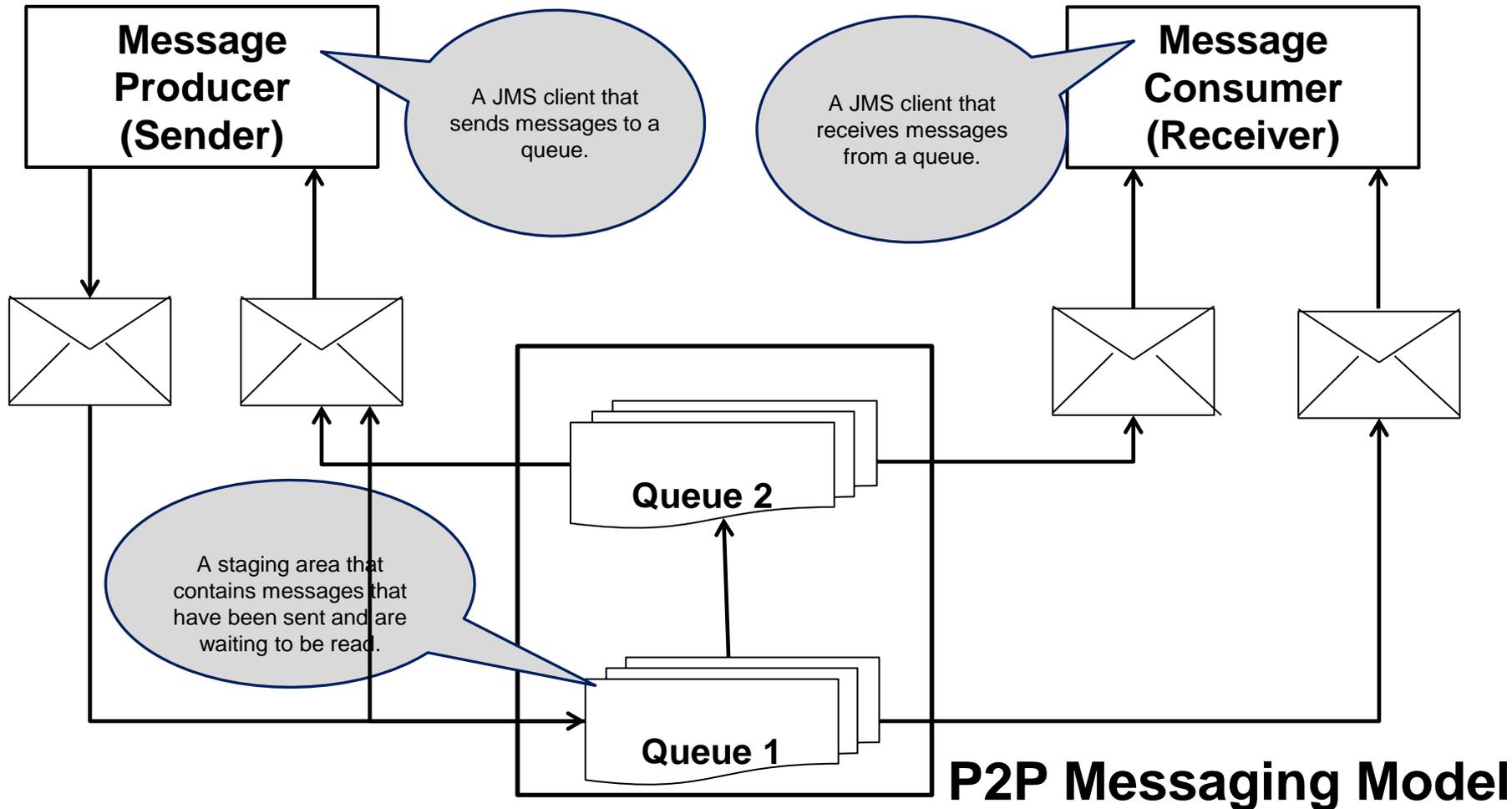
Each message is addressed to only one consumer. It allows users to send messages both asynchronously and synchronously using different channels (**queues**). Typically, in the PTP model, a user requests a message that a producer sends to the queue, rather than subscribing to a channel and receiving all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

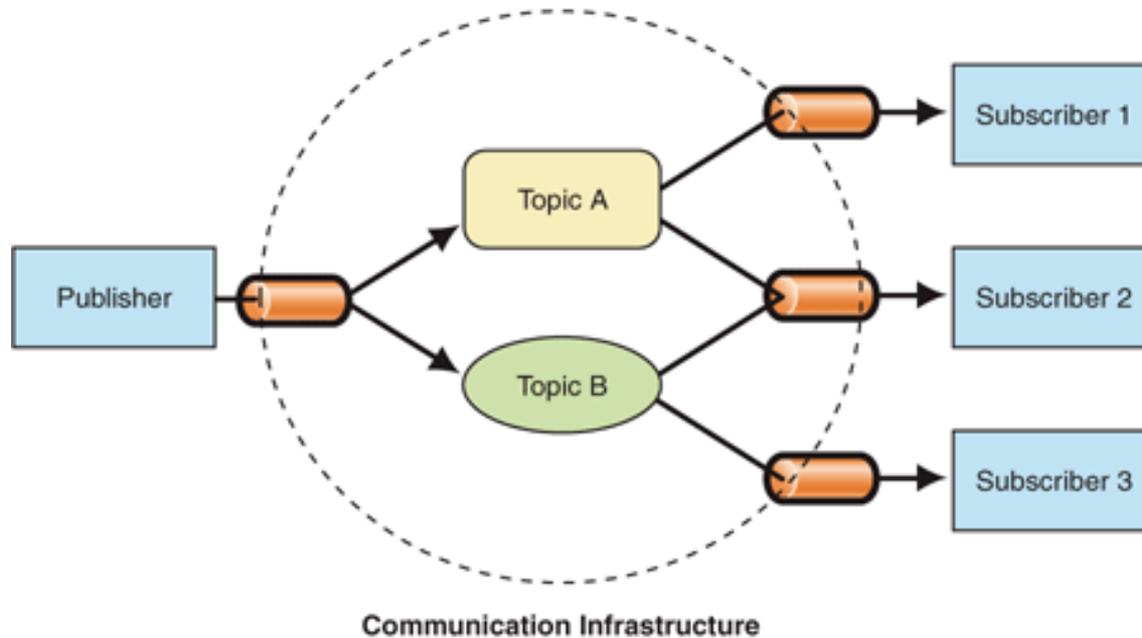
Note: Unlike a queue, a topic is a distribution mechanism for publishing messages that are delivered to multiple subscribers.

NEMS, cont'd



NEMS, cont'd

Pub/Sub Messaging Model



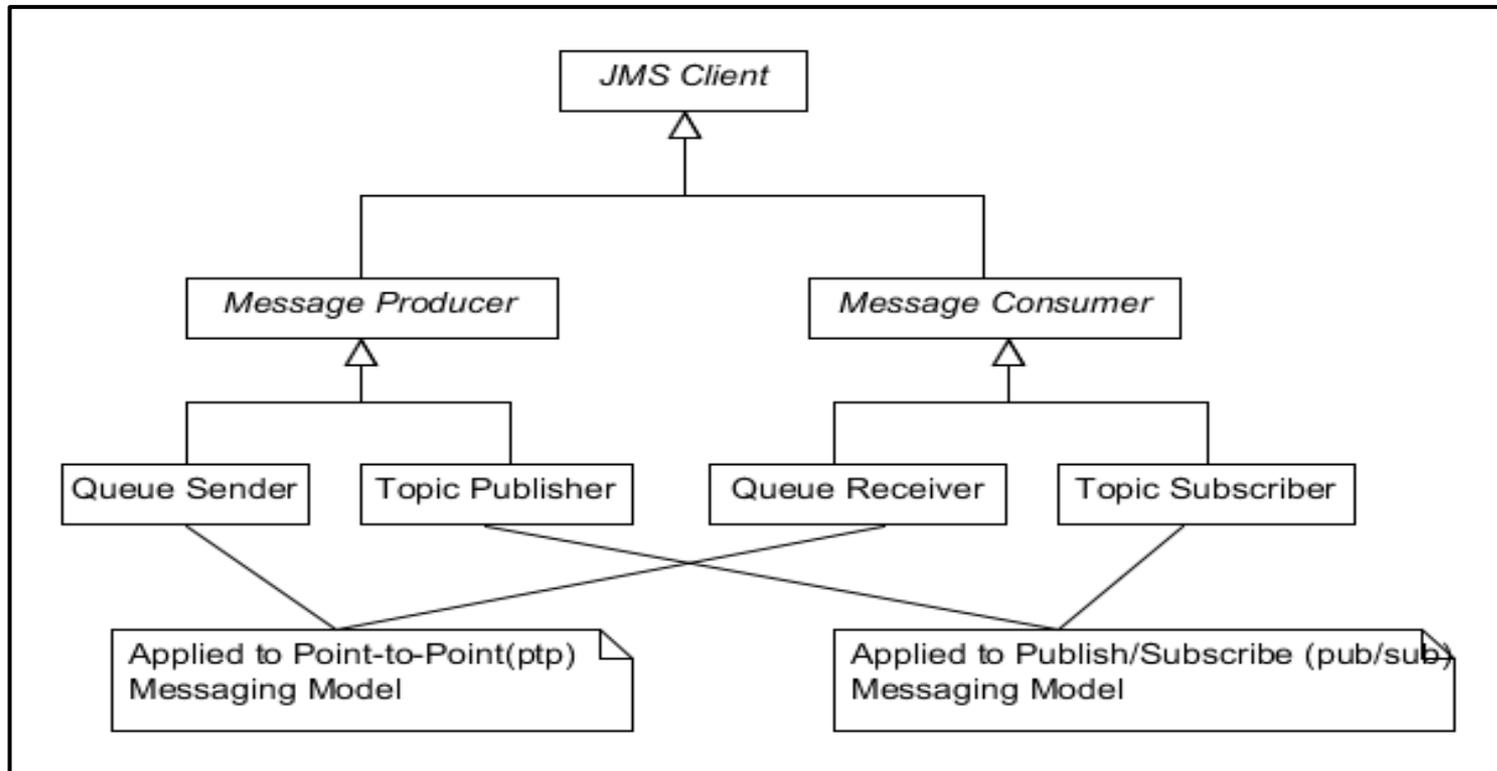
Message Provider

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

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

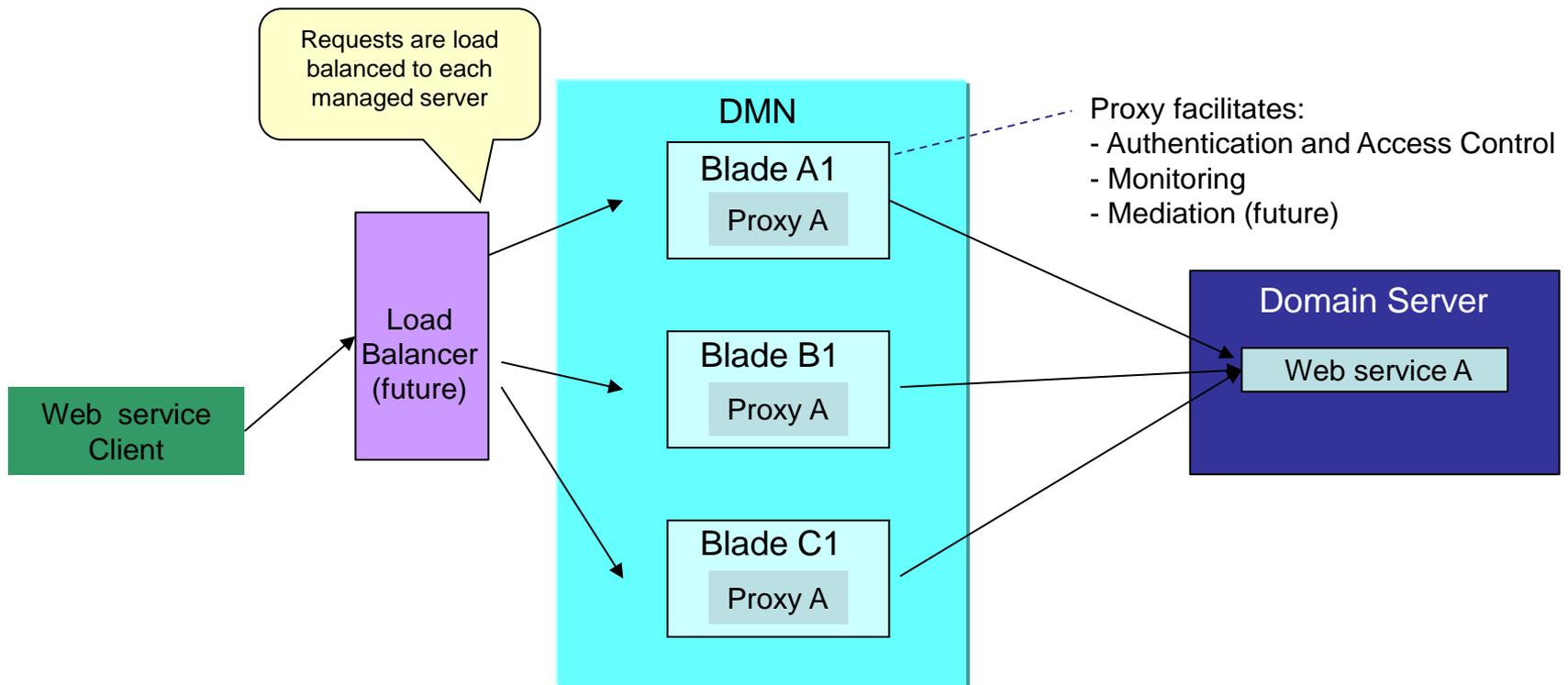
NEMS, cont'd

The following diagram depicts relationship between different roles in Messaging:



NEMS, cont'd

WS Proxy/Messaging Approach



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*

Infrastructure Services

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

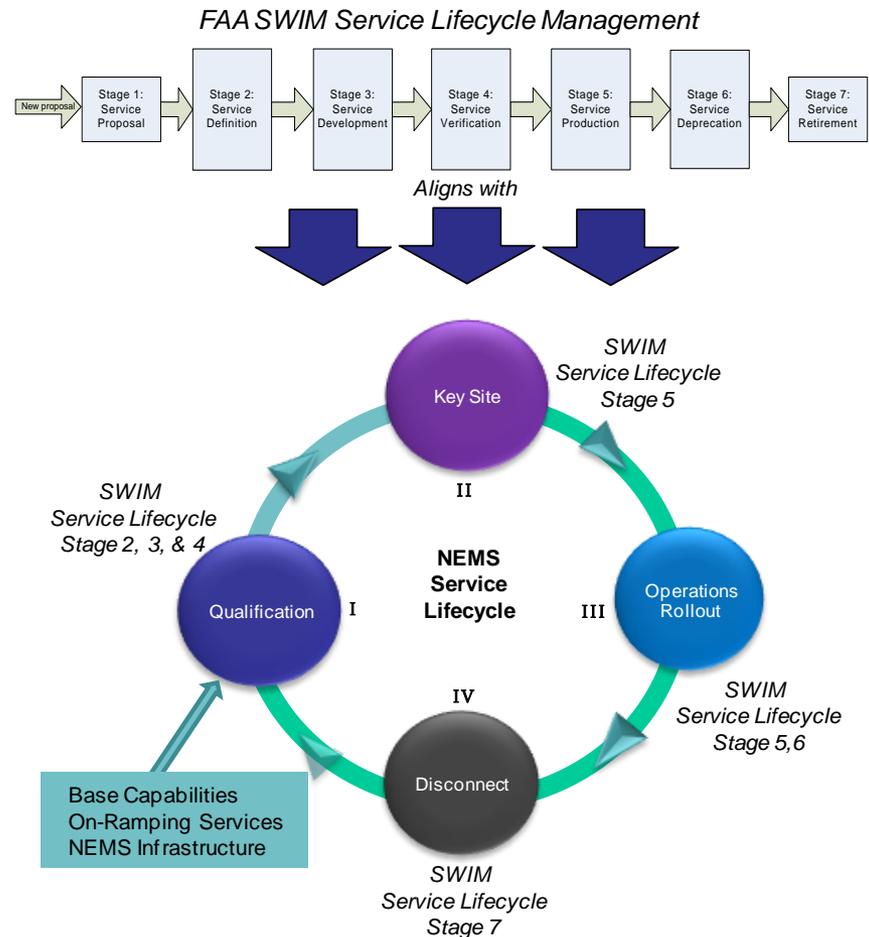
On-Ramping Services

Basic

- U1: WS-C: Web Service - Consumer
- U2: WS-P: Web Service - Producer
- U3: JMS-C: JMS Subscription - Consumer
- U4: JMS-P: JMS Publishing - Consumer

Enhanced

- E1: Mediation
- E2: Performance
- E3: Service Orchestration
- E4: SLAs
- E5: Security
- E6: Interoperability
- E7: Reports
- E8: Dynamic Subscriptions
- O1: Publications to R&D



Global Load Balancer Capabilities

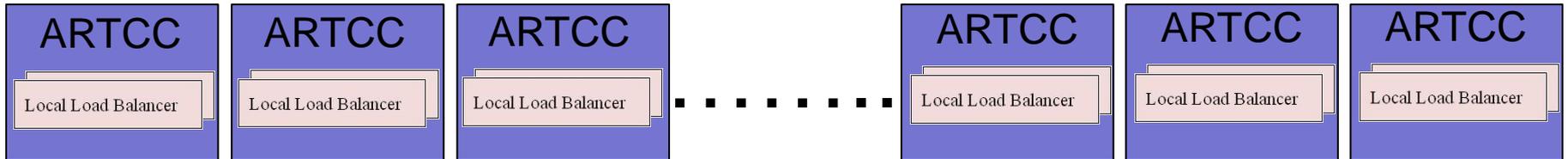
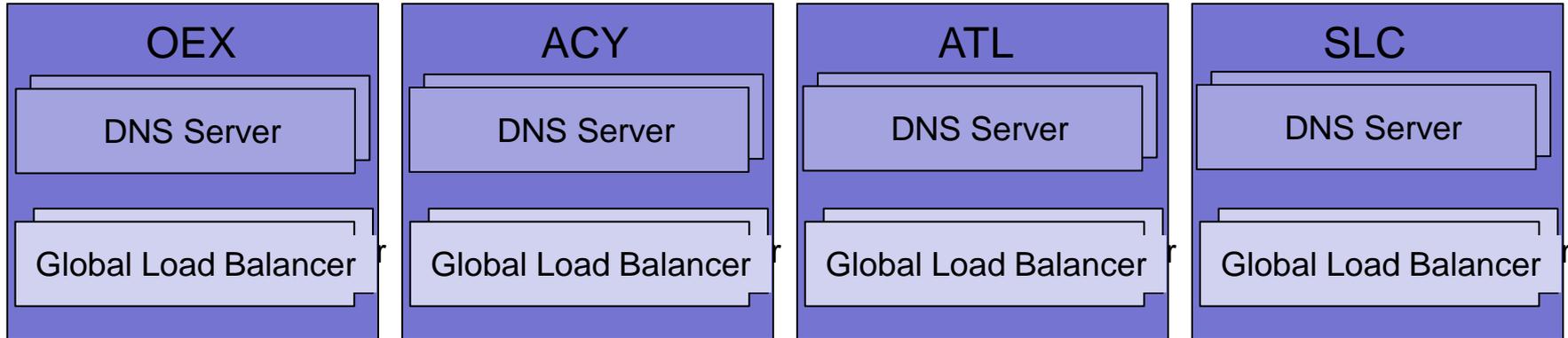
- Load balance traffic among multiple geographically distributed sites
- Select alternate NEMS sites in the event of site failures
- Direct specific clients to data centers based on configured policies
- Key Components
 - DNS—This front-end process focuses on the interaction with the DNS servers.
 - 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

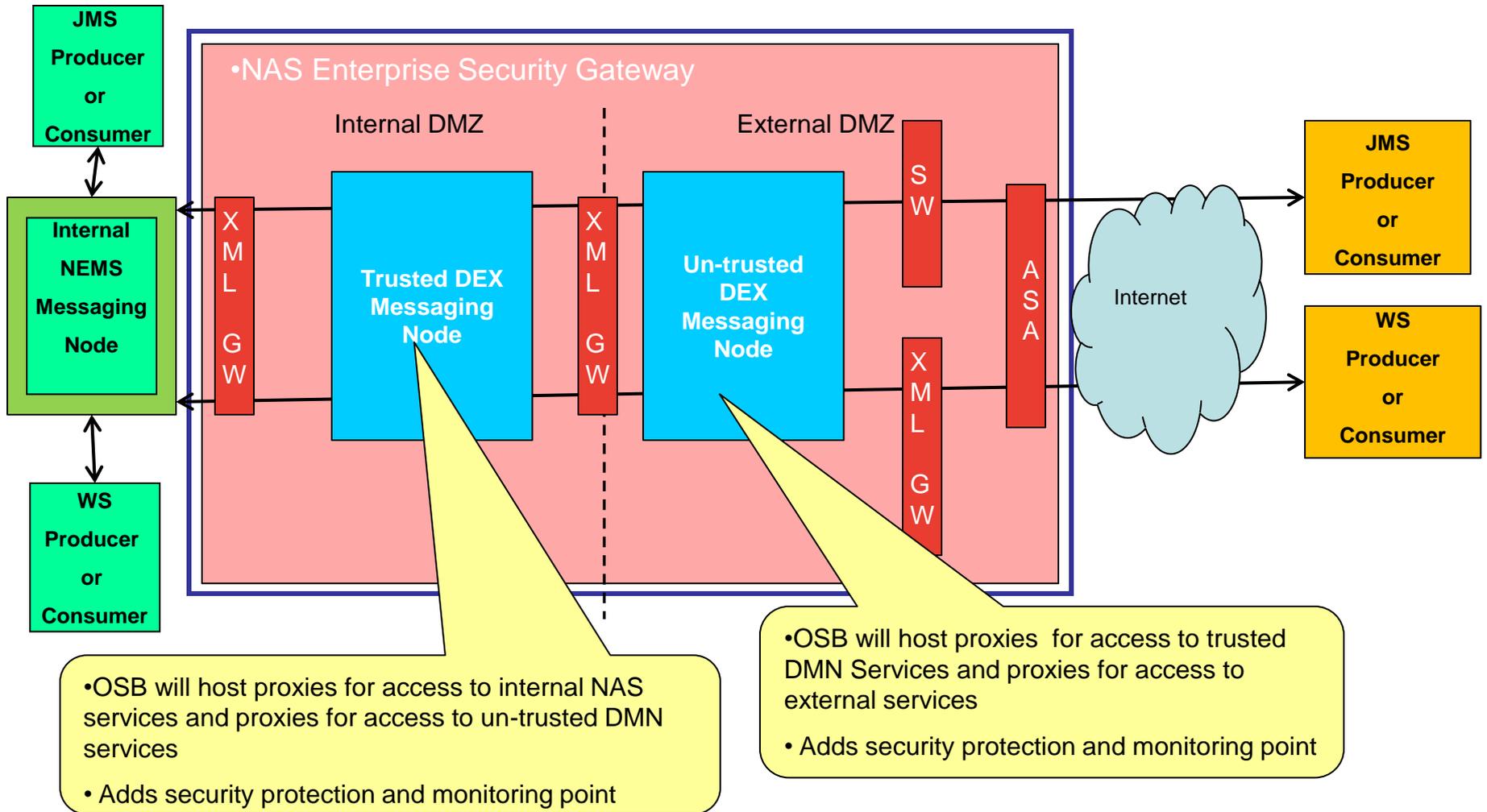
- Load balance traffic among local server pools
- Select alternate server in the event of server failures
- Direct specific clients to servers based on configured policies
- Key Components
 - VIPS— Provide entry points to the load balancer
 - Probes/Monitors—This back-end process gathers state and load 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



NESG Logical Architecture for 2-way SOA Services



NEMS Infrastructure Deployment



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

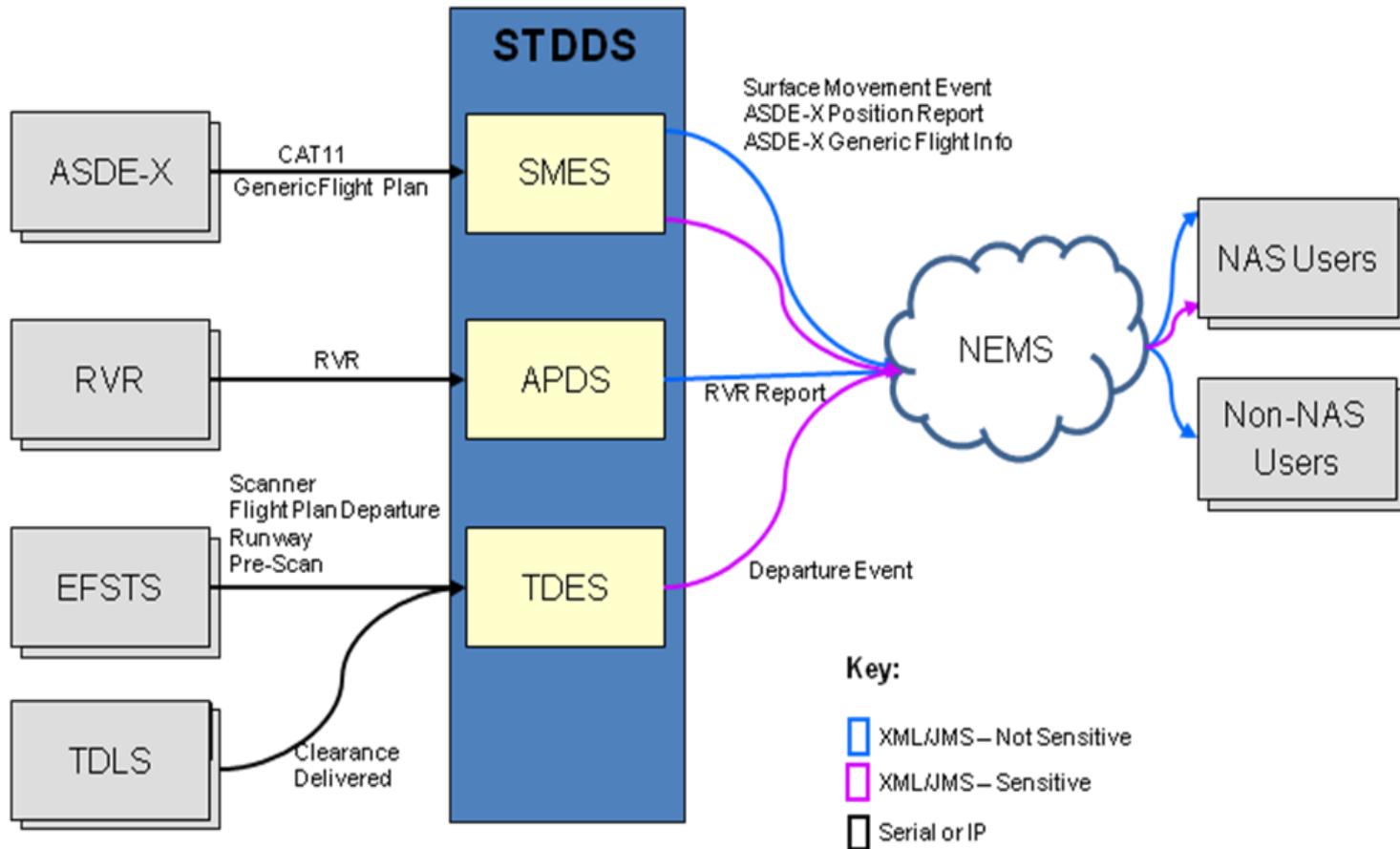


STDDS Program Using SWIM

SWIM Terminal Data Distribution Service (STDDS) Deployment



STDDS Current Architecture

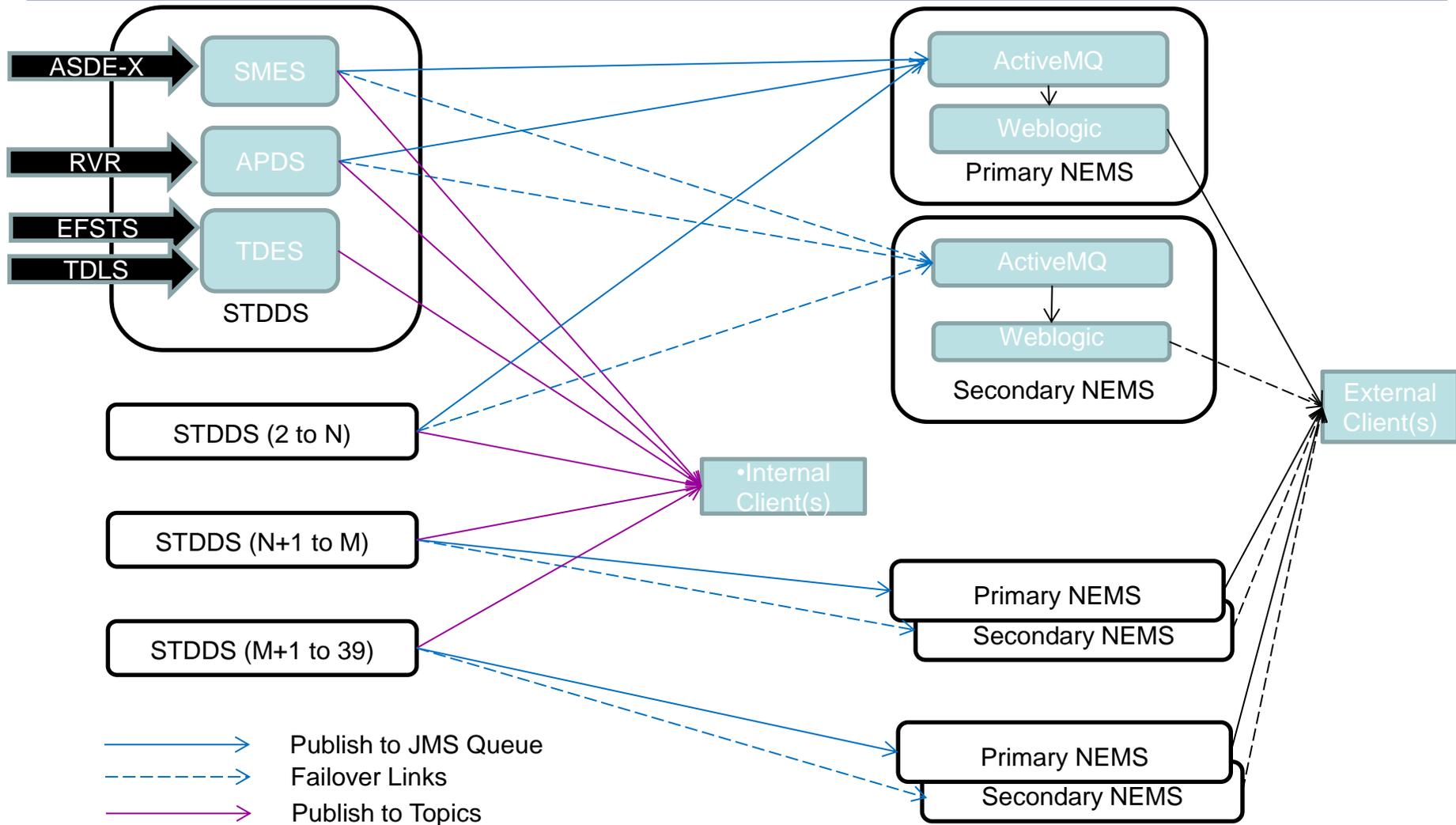


STDDS Current System

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



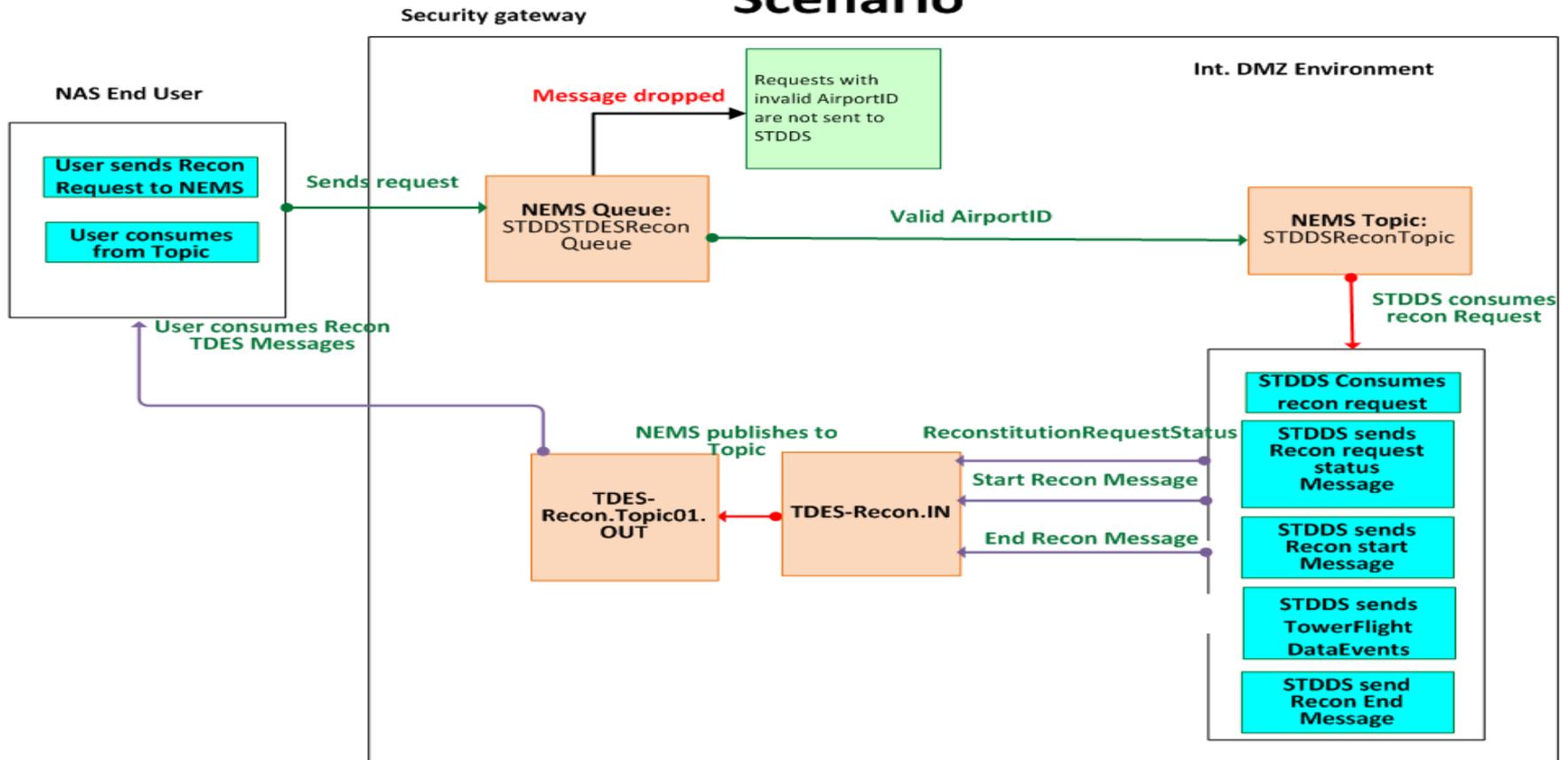
JMS vs Web services

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

- Most of the STDDS messages 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.

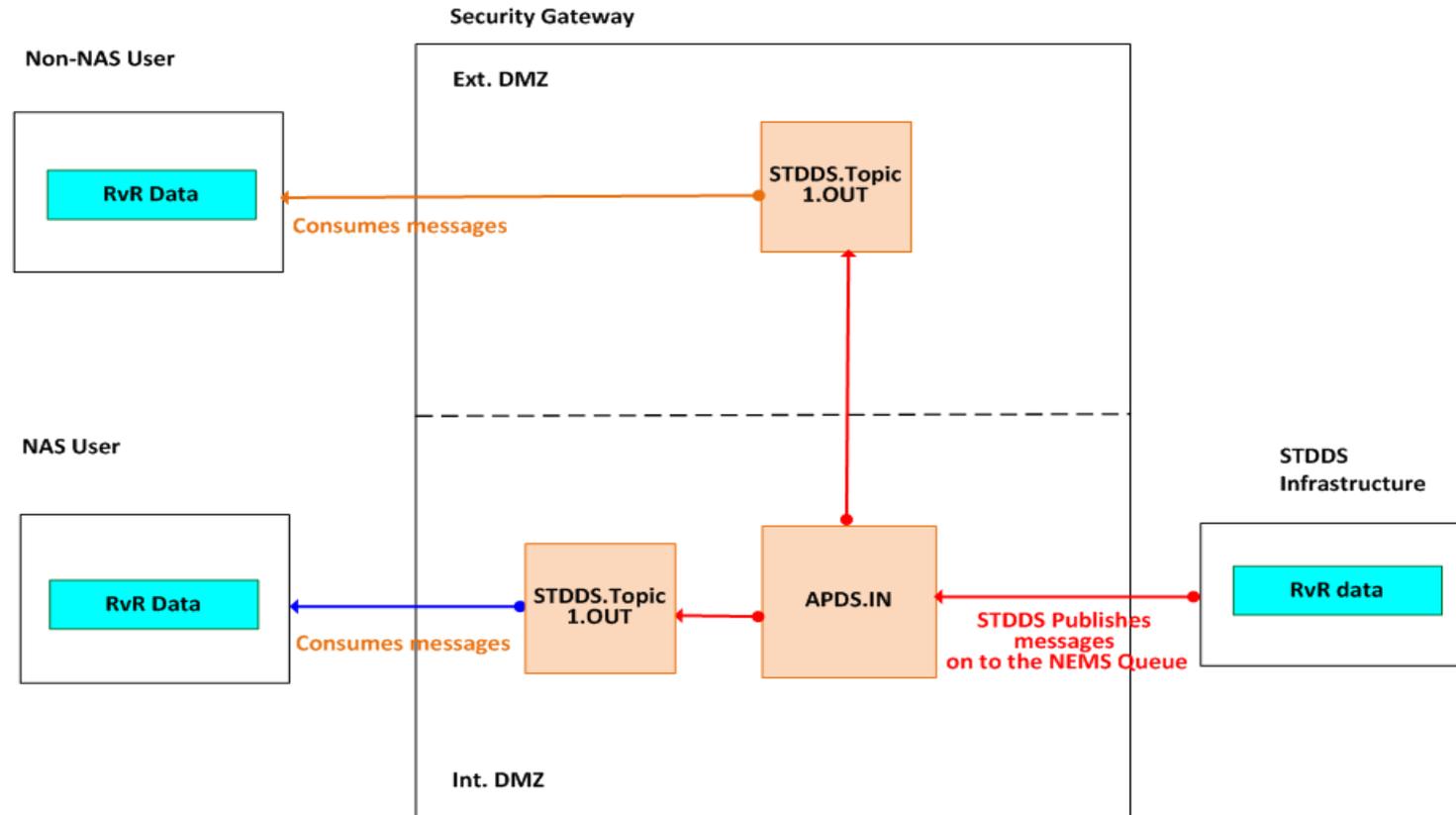
Reconstitution of TDES Message

NEMS/STDDS TDES Service Recon Scenario



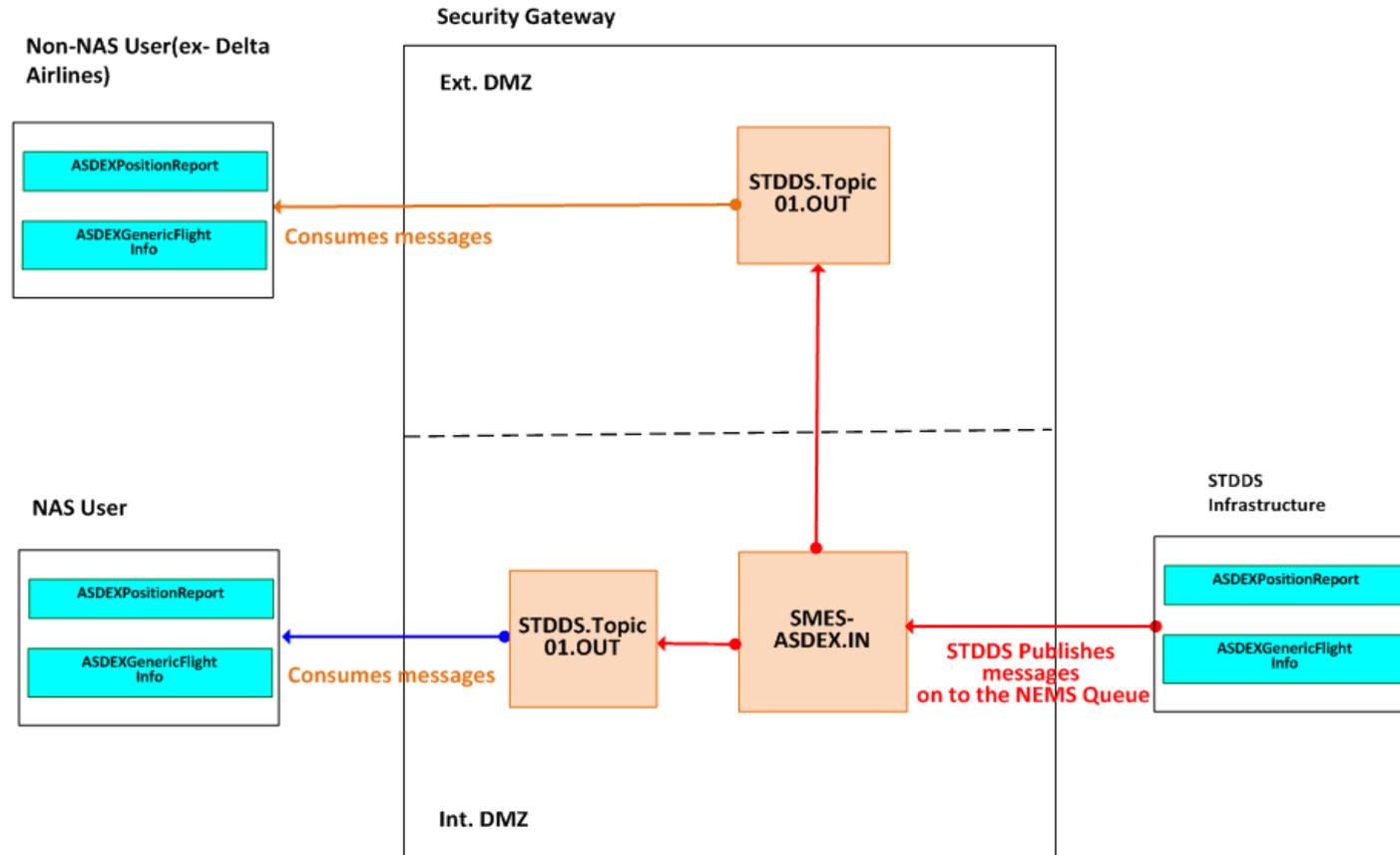
APDS Service

NEMS/STDDS APDS Service Scenario



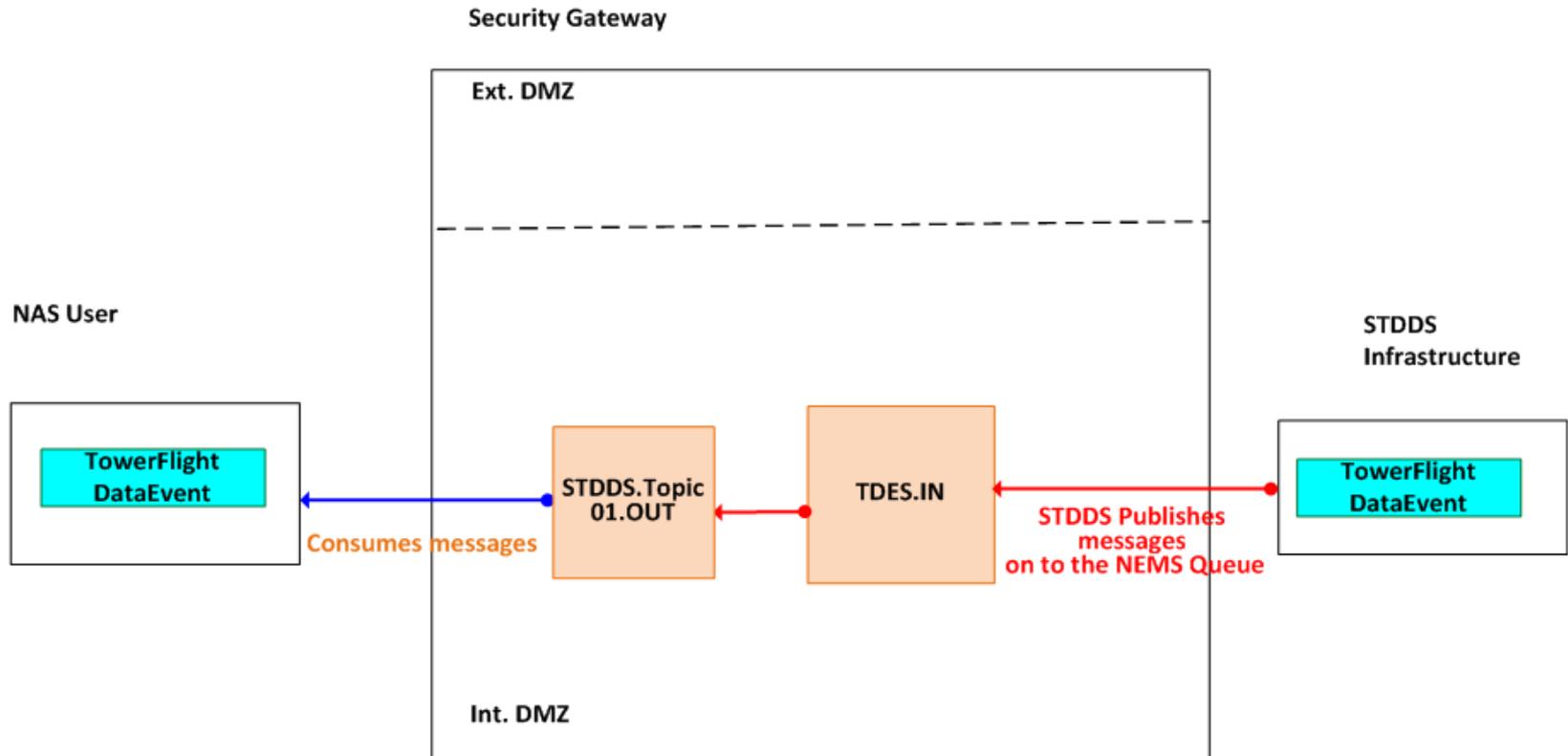
SMES ASDEX Service

NEMS/STDDS SMES Service ASDEX Scenario



TDES Event Service

NEMS/STDDS TDES Service Scenario



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





Federal Aviation
Administration

Thank You!

