Introduction to eLogbook Interoperability Services

Tim W. Anstey
Chief Architect, Crew Information Services
Associate Technical Fellow
Boeing

ATA eBusiness Forum
19 May 2010

Agenda

- **Brief overview of the Electronic Logbook Project Team (ELPT) and the current standard**
- **Overview of interoperability**
  - What is it? What are the benefits? How is it done?
- **Scope of the ATA eLogbook interoperability services standard**
- **Use of the eLogbook interoperability standard by OEMs, airlines, and software solution providers**
- **ELPT next steps**

ATA Electronic Logbook Project Team (ELPT) Charter

- **The Electronic Logbook Project Team (ELPT) is dedicated to the creation and maintenance of an industry standard data model to support the exchange of eLogbook data.**
  - Focus on line maintenance activities.
  - Define a set of transactions to promote interoperability among ground systems that generate or utilize eLogbook data.

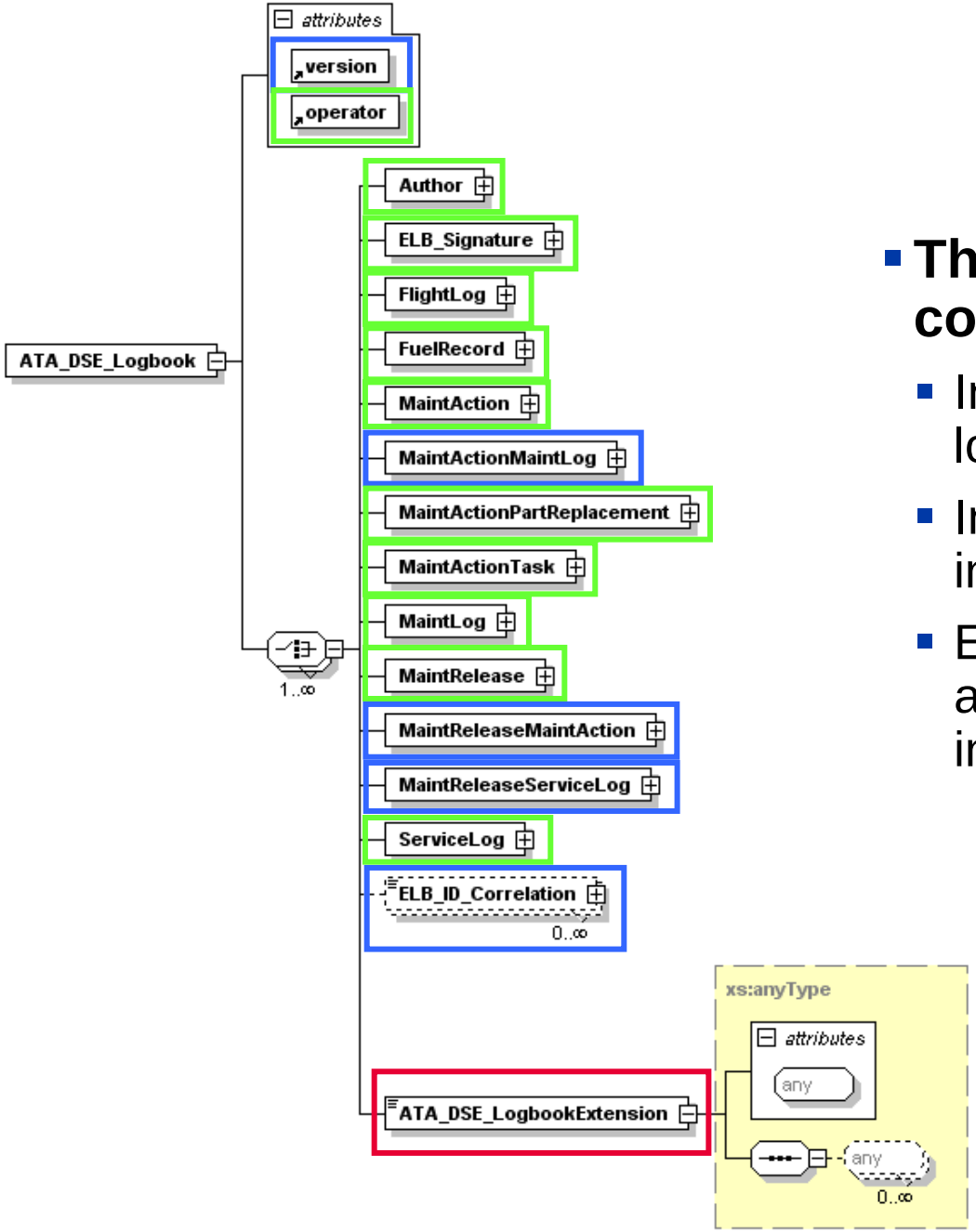
- **Coordinate efforts with other ATA e-Business project teams (including the RIG, XMLPT, MMELPT and DSWG) and other industry standards groups (e.g. ARINC 633 AOC Sub-committee).**

- **Specification used by: OEM airframers, third-party software companies, airline operators, and aviation maintenance information system providers.**
  - ELPT is well represented by the target audience.

ELPT History

- **2006: Project kickoff, object data model discussion, RIG coordination, 1st internal release of specification to ELPT (v1.0.00)**
- **2007: 1st draft object data model published to ESG (v1.0.01), XML signature approach discussed & added to specification**
- **2008: ARINC 633 AOC sub-committee coordination, begin interoperability effort, publish v2.0 of object data model (including fuel logbook data & servicing)**
- **2009: Interoperability draft spec work, object data model adjustments up to v2.03**
- **2010: Finalize interoperability standard and publish specification as v2.04.**

ATA DSE Logbook Object Data Model



- **The eLogbook data model consists of placeholders for:**
  - Information captured in paper logbooks
  - Information to support data interchange among systems
  - Extending the model to include airline or system unique information

ATA DSE Logbook Data Exchange

- **The purpose of the ATA DSE Logbook data model is to support data exchange. Why do we need to define transactions?**
  - The data model is in use today, but data exchange primarily lends itself to a point-to-point messaging model (message queues).
    - Other communication and service styles would have to be defined and implemented by the application provider.
  - Content of the message payload is defined by the application provider.
    - Example: Include **<MaintLog>** element with **<MaintAction>**?
    - Example: Include **<Author>** and **<ELB_Signature>** with **<MaintAction>**?
    - Example: How many **<MaintAction>** records per message?
  - Implementations are proprietary and customer dependent.
    - Interface specification required for each implementation.

Why Do We Need an eLogbook Interoperability Services Standard?

- **There is a need:**
  - Airlines are deploying Electronic Flight Bags with electronic logbooks.
  - Airlines want to integrate electronic logbooks with other back-office systems, especially systems that support line maintenance activities.
  - Airlines are creating their own applications using data from multiple systems (data mash-ups).

- **Reduce cost to airlines:**
  - Airlines do not want to absorb the cost of custom implementations.
    - Looking for vendor-neutral support.
  - Support multiple electronic logbook applications with a single line maintenance system (fleet-wide solution).

- **Reduce cost to software solution providers:**
  - Electronic Logbook solution providers want to deliver their products to multiple airlines with a variety of back-office line maintenance systems.
  - We believe that line maintenance system solution providers want to integrate in a cost-effective manner with multiple logbook vendors.

eLogbook Interoperability Standard Principles

The eLogbook Interoperability Standard should be:

- **Non-proprietary**
- **Platform neutral**
- **Use a common file format**
- **Leverage existing industry standards (ATA, W3C, IEEE, etc.)**
- **Implementation agnostic**
  - Do not constrain message exchange patterns
  - Bindings are defined
- **Maintain flexibility by limiting required data**
  - Trade-off: Too much flexibility may discourage adoption

What is Interoperability?

- **IEEE Glossary defines interoperability as:**
  - The ability of two or more systems or components to exchange information and to use the information that has been exchanged.
    - Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.

- **Interoperability is a broad term and may be further sub-divided:**
  - Technical/ syntactic, semantic, political/ human, inter-community, legal, and international.
  - Within the ATA Electronic Logbook Project Team, we actively discuss the first two types of interoperability: technical and semantic interoperability.

Types of Interoperability

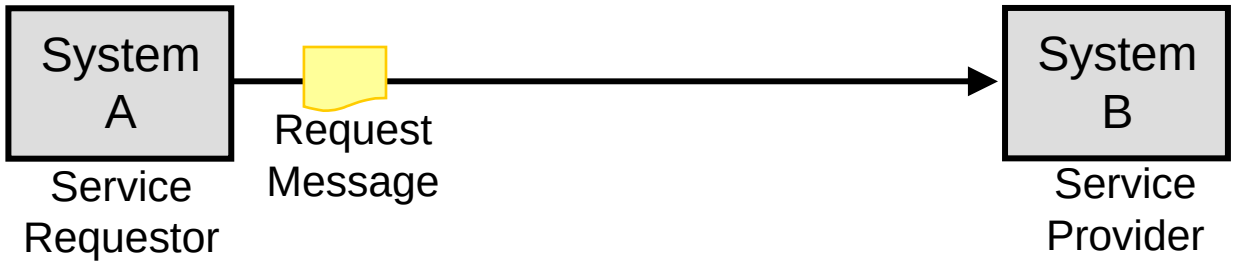
- **Semantic Interoperability**
  - The ability to automatically process data from another system without requiring additional interpretation. Both systems apply the same meaning to the data.
  - The user's perspective – context matters.
    - Example: Do we all agree that a Maintenance Release means the same thing?
  - Must avoid ambiguity to be successful.
  - **This is the greatest challenge to achieving interoperability!**

- **Technical/ Syntactic Interoperability**
  - Any two systems that can communicate and exchange data with each other.
  - This type of interoperability is based on defining: formats, communication protocols, communication styles and service styles.
  - Machine-oriented.

Interoperability Example: Logical Representation of Systems

- **Example: System A wants to use a service that System B provides.**
  - The system that provides a Web service to an external system is the Service Provider.
  - The system that sends a request to use a Web service is the Service Requestor.
  - The request from the Service Requestor to the Service Provider is sent as a request message.

- **The format of the request message, the Web service definition, and the method for discovering the Web service are dependent upon the communication and service style selected for use.**



Technical Interoperability: Formats and Protocols

- **To promote adoption of the eLogbook Interoperability Services Standard, preferred industry formats and protocols must be supported.**

- **Common formats:**
  - XML, JavaScript Object Notation (JSON), XHTML, and Atom (XML syndication format)

- **Common transport protocols:**
  - Hyper Text Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP)

Technical Interoperability: Communication Styles

- **The eLogbook Interoperability standard must consider that not all applications will want to interact with a Web service using the same message exchange pattern.**

- **Message exchange patterns include:**
  - One-way feed (fire and forget)
  - Event publishing (subscribe/ publish)
  - Synchronous request/ response
  - Asynchronous request/ response

- **Web service providers may need to support more than one pattern depending on their interaction requirements.**

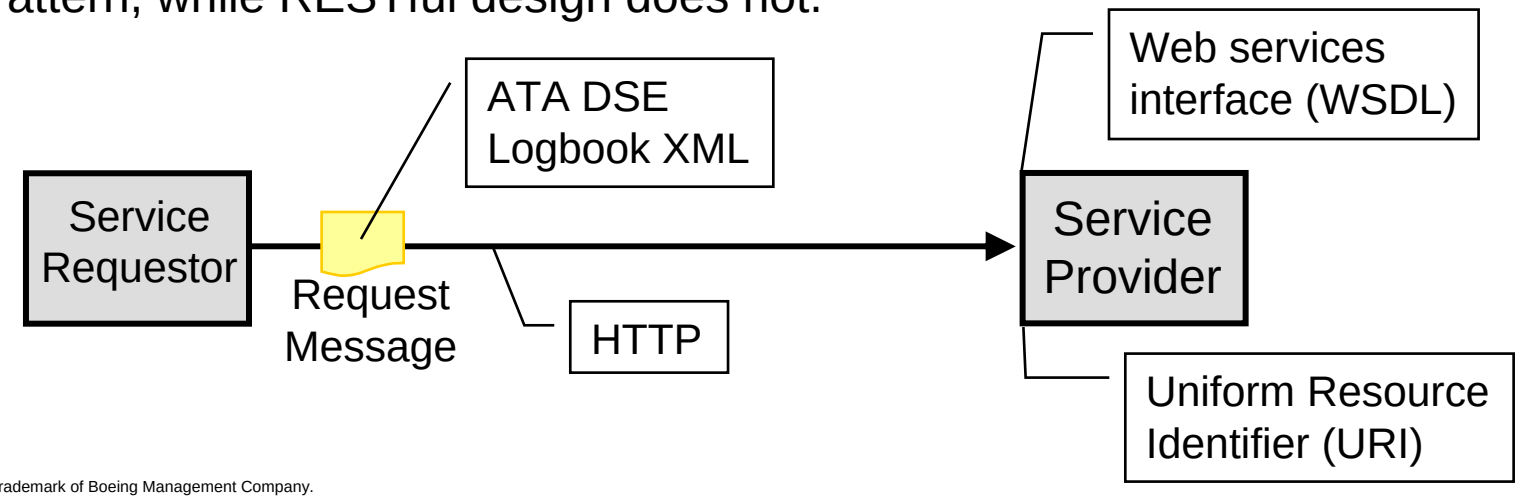
Technical Interoperability: Service Interface Styles

- **When exchanging data between systems, there are three main service interface styles to consider:**
  - Document-Oriented:
    - Exchange messages that contain documents.
    - Typically asynchronous.
    - Message queues typically used.
  - Method-Oriented:
    - Web service specific operation.
  - Resource-Oriented:
    - Services are exposed as Uniform Resource Identifiers (URIs).
    - Concept: Information is linked together.
    - REpresentational State Transfer (REST) architectural style

- **Some service interface styles may support more than one communication protocol or may require a specific protocol.**

eLogbook Interoperability Scope

- **File Format: ATA DSE Logbook XML**
- **Protocol: HTTP**
- **Service Interface Style: Web Services Framework (SOAP binding), REST**
- **Communication Styles: Implementation choice**
  - Does not preclude asynchronous, synchronous, or one-way message exchange
  - Web Services Framework Extensions support the Event Message Exchange Pattern, while RESTful design does not.



eLogbook Interoperability Services

- **Define Web services based on the Web Services Framework (SOAP binding) and the REST architectural style.**
  - Approach supported by OASIS Content Management Interoperability Services

- **eLogbook Interoperability Services:**
  - Query Service - Request any entity's data (e.g. Flight Log, Maintenance Log, etc.)
  - Add and update the following entities:
    - Flight Log
    - Maintenance Release
    - Author
    - Maintenance Action (including part replacement and tasks)
    - Electronic Signature is added to each entity, if applicable.

    Maintenance Log
    Fuel Record
    Service Log
  - Deletion operations are defined as an update request with a value of **`TRUE`** included in the element **`<IsVoid>.`**

- **eLogbook Event Notifications:**
  - ETOPS Status
  - Deferral Status
  - Flight Log Closed

  Maintenance Release Status
  Deferral Expiration Notification
  Maintenance Release Required

eLogbook Interoperability Services (continued)

- **In proposing a standard set of eLogbook transactions, the Project Team:**
  - Identified a set of use cases or scenarios
    - Large set to begin with and quickly narrowed
    - Scenarios refined over 18 months
  - Defined a generic set of services
  - Engaged ATA XMLPT to define ATA namespace
    - Necessary to avoid syntax collisions with other XML schemas
    - http://www.ataebiz.org/XMLSchema
  - Developed a Response XML schema
    - Included elements for capturing system errors
  - Defined the Web Services Framework Services
  - Defined the RESTful Web Services
  - Developed XML schemas for the messages

eLogbook Generic Service Description

addFlightLog	
Description	Creates a record of the flight to which observed aircraft faults may be associated.
Inputs	**Required:** •ATA DSE Logbook Flight Log object data **Optional:** •ATA DSE Logbook Author object data •ATA DSE Logbook ELB Signature object data
Outputs	**Required:** •ATA DSE Logbook Response success status **Optional:** •Correlating Author ID: The unique ID generated by the receiving system that is used to reference the ATA DSE Logbook Author object. •Correlating ELB Signature ID: The unique ID generated by the receiving system that is used to reference the ATA DSE Logbook ELB Signature object. •Correlating Flight Log ID: The unique ID generated by the receiving system that is used to reference the ATA DSE Logbook Flight Log object.
Exceptions	**Required:** •ATA DSE Logbook Response error status •ATA DSE Logbook Response error details **Optional:** •Return the ATA DSE Logbook Flight Log object data for a “Record Already Exists” exception. •Return the ATA DSE Logbook Author ID as a missing reference ID if the ELB Signature object is sent and not the Author object and the receiving system does not have a record of the author. **Applicable Exceptions:** •Common Service Exceptions •Record Already Exists •Missing Reference Data
Remarks	If the Author already exists in the receiving system, do not return a “record already exists” error since the transmission of this reference data to the flight log is optional.

Web Services Request: SOAP Definition - addMaintLog

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-
 ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding
 /">
  <SOAP-ENV:Header>...</SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ATA_ELB:addMaintLog
 xmlns:ATA_ELB="http://www.ataebiz.org/XMLSchema">
      <ATA_ELB:ATA_DSE_Logbook>
        <ATA_ELB:Author>...</ATA_ELB:Author>
        <ATA_ELB:ELB_Signature>...</ATA_ELB:ELB_Signature>
        <ATA_ELB:MaintLog>...</ATA_ELB:MaintLog>
      </ATA_ELB:ATA_DSE_Logbook>
    </ATA_ELB:addMaintLog>
    <SOAP-ENV:Fault>...</SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

addMaintLog Web Service Request: <MaintLog> Details

```
<MaintLog maintLogID="afd119580-2944">
  <TailNum>N1234</TailNum>
  <TypeOfMaintLog>FAULT</TypeOfMaintLog>
  <EntryDateTime>2008-07-24T23:39:00Z</EntryDateTime>
  <FlightLogID>khi997878909-98798</FlightLogID>
  <RecordName>ML2000000</RecordName>
  <FaultCode>52387200</FaultCode>
  <FaultDescription>Cargo door (bulk) - Difficult to
 latch or unlatch</FaultDescription>
  <Comments>Latched after 5 attempts</Comments>
  <EventCode>DLY</EventCode>
</MaintLog>
```

RESTful Services Overview

- **REST – REpresentational State Transfer**

- **The REST architecture or design pattern:**
  - It is not a standard, but uses other standards
    - HTTP (GET, PUT, POST, DELETE), URL, XML, HTML, GIF, etc.
  - It is not a tool
  - It is a Web service, but does not use the same standards that are more familiar with Web Services discussions.

- **With REST you create a resource for every service**
  - A URL is used to identify every resource
  - Everything has a globally unique identifier

RESTful Services Overview (continued)

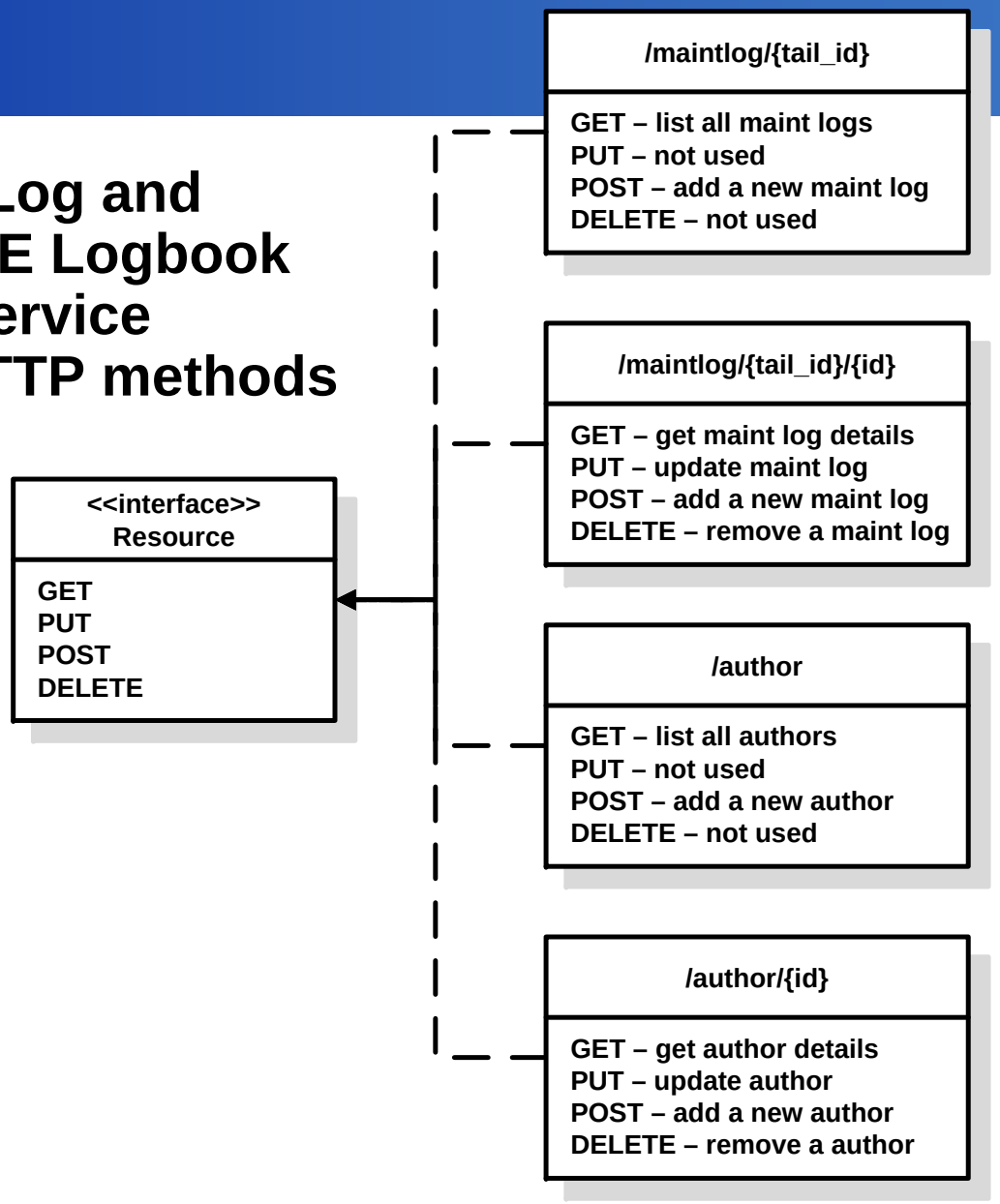
- **Data is represented as a network of information (links from higher level information to the lower level information or details)**
  - Contrast with other Web Service designs which encapsulate the data in an object-oriented manner
  - Example: http://www.ataebiz.org/maintlog/N1234/103458932

```
<MaintLog maintLogID='103458932'>
   <FlightLogID
   ref='http://www.ataebiz.org/flightlog/N1234/2526799172'/>
   <AuthorID ref='http://www.ataebiz.org/author/8831787'/>
</MaintLog>
```

- **Methods for requesting and transmitting information are standard**
  - HTTP – HyperText Transfer Protocol
  - Provides standard methods for: GET, PUT, POST, DELETE

RESTful Services Overview (continued)

- **Mapping MaintLog and Author ATA DSE Logbook RESTful Web service interfaces to HTTP methods**



RESTful Services Overview (continued)

- **Resources can present more than one representation**
  - A client (calling or requesting application) can ask for a representation of the data in a particular format

- **Example – XML representation of data:**

```
GET /flightlog/84090934 HTTP/1.1
Host: www.ataebiz.org
Accept: application/xml
```

- **Example – Text representation of data:**

```
GET /flightlog/84090934 HTTP/1.1
Host: www.ataebiz.org
Accept: application/text
```

RESTful Services Overview (continued)

- **REST is stateless**
  - Stateless constraint is designed to isolate the client against changes on the server so that it is not dependent on sending a request to the same server when processing consecutive requests
    - Constraint needed to support scalability
    - High Availability Systems
    - Load Balanced Systems
  - This does not mean that state does not exist
    - State is either maintained in the resource, or
    - State is maintained with the client

RESTful Services Definition: <FlightLog>

Resource	{URI}/flightlog/{tail_id}
HTTP Method	Description
GET	List all flight logs for a specific tail number
PUT	Not used
POST	Add a new flight log for a specific tail number. Action will create a flight log ID
DELETE	Not used

RESTful Services Example: <MaintLog> Details

```
<MaintLog maintLogID="afd119580-2944">
  <TailNum>N1234</TailNum>
  <TypeOfMaintLog>FAULT</TypeOfMaintLog>
  <EntryDateTime>2008-07-24T23:39:00Z</EntryDateTime>
  <FlightLogID
 href=http://www.MyCompany.com/flightLog/khi997878909-
 98798/details/>
  <RecordName>ML2000000</RecordName>
  <FaultCode>52387200</FaultCode>
  <FaultDescription>Cargo door (bulk) - Difficult to
 latch or unlatch</FaultDescription>
  <Comments>Latched after 5 attempts</Comments>
  <EventCode>DLY</EventCode>
</MaintLog>
```

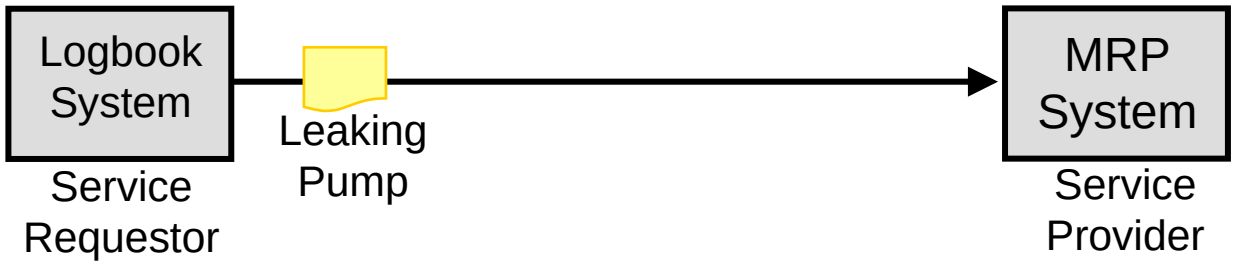
eLogbook Interoperability at an Airline

- **There are many potential system integrations that may be supported at an airline.**
  - The eLogbook interoperability standard may be adopted in whole or in part by a system to support the integration.
    - Dashboards: Implement Event Services
    - Health Management: Implement Maintenance Log Services
    - Billing: Implement Fuel Record Services
    - Maintenance Resource Planning: Implement Maintenance Log, Maintenance Action, Servicing, and Maintenance Release Services

- **Example eLogbook interoperability scenarios:**
  - eLogbook and MRP data exchange
  - Two systems providing the same Web service to each other
  - Mixed fleet scenario – two systems calling the same Web service
  - Event notification

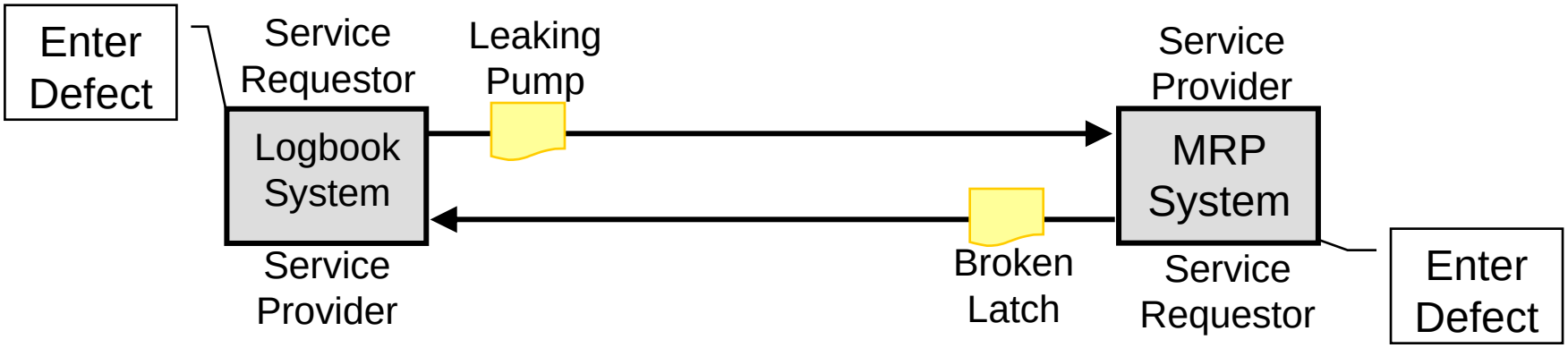
Interoperability Example: eLogbook & MRP Data Exchange

- **Example:**
  - System A is now represented by an eLogbook application.
  - System B is now a Maintenance Release Planning (MRP) application.
- **The eLogbook application wants to send an observed fault to the MRP system for line maintenance execution.**
- **The MRP system has a Web service, `addMaintLog`, that the eLogbook system will send a request to.**



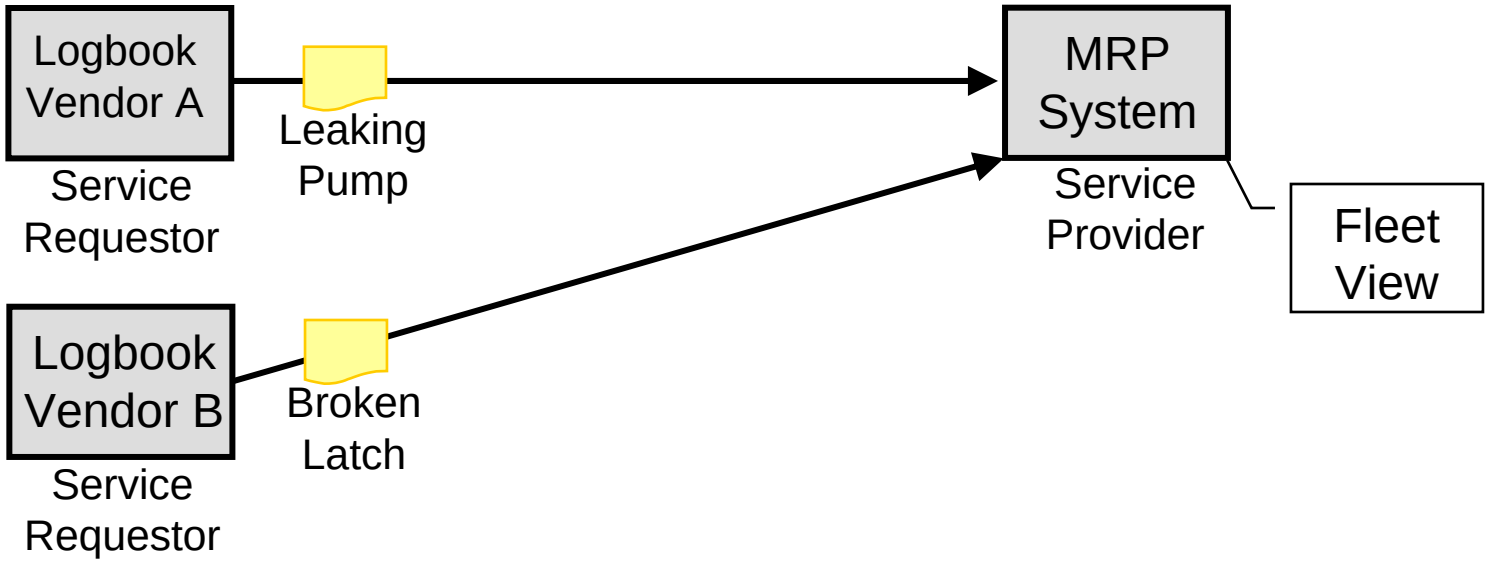
Interoperability Example: Two Systems Providing Same Service To Each Other

- **Two systems exchanging data may each provide the same Web service.**
- **Depending on the airline operations, both systems may request the Web service from each other.**
- **Example: Airline operations allow new aircraft defects to be reported in either the Logbook or MRP System, but require that both systems contain a record of the defect.**



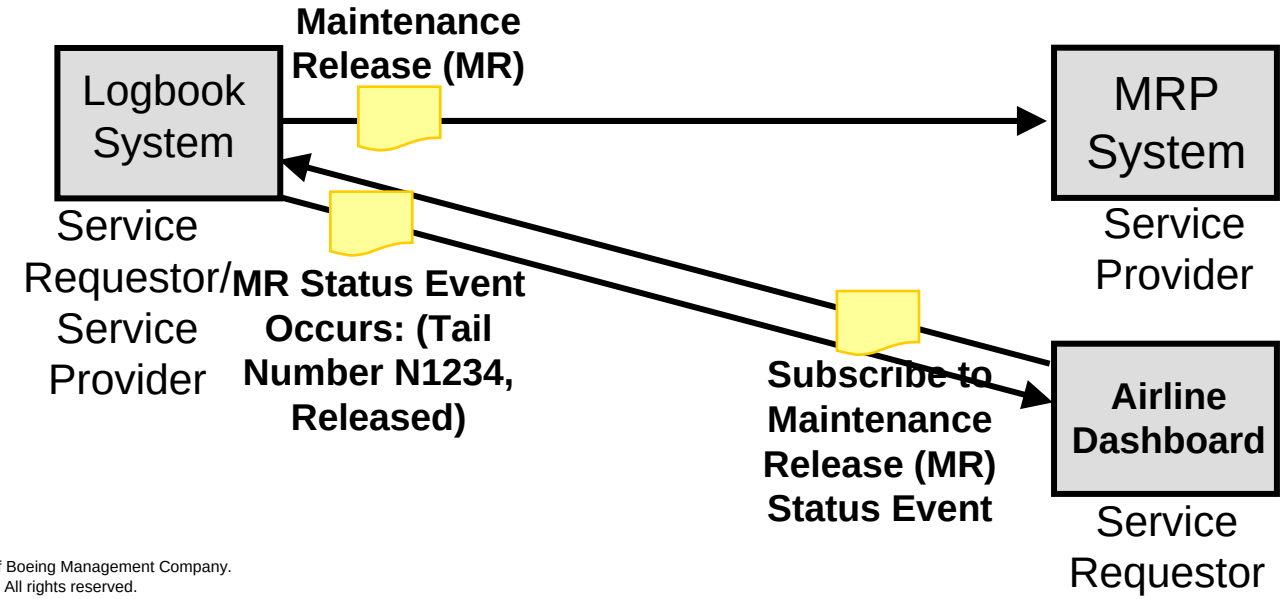
Interoperability Example: Mixed Fleet Scenario

- **Multiple systems may call the same Web service.**
- **Example: Airline with a mixed fleet uses an eLogbook from one vendor on one type of aircraft and an eLogbook from another vendor on other types of aircraft.**
- **Each Service Requestor calls the same Web service, addMaintLog, from the Service Provider.**



Interoperability Example: Event Notification

- **In addition to defining a set of Web services to support eLogbook transactions, the standard also defines a set of system events that may be subscribed to.**
- **In this case, the eLogbook application is both a Service Requestor (to the MRP System) and a Service Provider (to the Airline Dashboard).**



ELPT Next Steps

- **eLogbook Data Model**
  - Majority of work complete as represented in v2.04.
  - eLogbook data model is in production use by OEM airframers, airlines, and software solution providers.
    - Several years of experience
  - Considering new proposals in 2010 that will result in a major version number change (v3.00).
    - Parts Request, MaintAction Deferral, and Journey Log definitions.
    - XML Namespace
    - Semantics

- **eLogbook Interoperability Services**
  - Introduce standard in v2.04
  - Validate service scope and service definitions through production implementations
  - Consider new services and events based on data model proposals