Misc. Contracts & Agreements
No. 26630

INTERAGENCY AGREEMENT
Oregon Interoperability Service

THIS AGREEMENT is made and entered into by and between THE STATE OF OREGON, acting by and through its Department of Transportation, hereinafter referred to as "ODOT" and THE STATE OF OREGON, acting by and through its STATE POLICE DEPARTMENT, hereinafter referred to as "OSP", both herein referred to individually or collectively as "Party" or "Parties".

RECITALS

1. By the authority granted in ORS 190.110 and 283.110, state agencies may enter into agreements with units of local government or other state agencies for the performance of any or all functions and activities that a party to the agreement, its officers, or agents have the authority to perform.

2. An Acronym list is attached for easy reference of Acronyms listed in this Agreement.

3. The purpose of this Agreement is to formalize and document the understanding between ODOT and OSP as to the business operations, cost sharing and resource commitment necessary to implement and maintain an Interoperable Data System (System) between ODOT's TOCS and OSP's CAD systems.

4. The purpose of the System is to achieve cooperative and efficient disposition of calls for service requests and highway incident responses for OSP and ODOT resources. It will allow for the exchange of incident and calls for service data between OSP CAD and ODOT's TOCS software system (currently under development).

NOW THEREFORE, the premises being in general as stated in the foregoing Recitals, it is agreed by and between the Parties hereto as follows:

TERMS OF AGREEMENT

1. Under such authority, ODOT wishes to retain the services of the OSP Dispatch Support Unit to participate in the work of developing the OSP-CAD/TOCS System as shown on Exhibit "A", attached hereto and by this reference made a part hereof.

2. The work for the System shall be on going.

3. OSP will provide access to real-time CAD data through a web services interface that will be capable of connecting through the hardware and software combined to emulate the System Connection Architecture model specified in document "SCA

Specification PDCC-CAD Integration", Prepared for: PSSI, by: On-line Business Systems; or by another transport mechanism which supports the same web service interface functionality as described in the PDCC-SCA document, as shown on Exhibit B, attached hereto and by this reference made a part hereof.

4. ODOT agrees that all connections shall have final authorization from OSP and that OSP Dispatch Support Unit shall be responsible for managing the interfacing hardware and software connected to the OSP CAD Server and upstream of the ODOT TOCS System.

5. ODOT agrees that all interfacing software developed to send messages from OSP-CAD and receive messages from the TOCS system will be handled by OSP's CAD vendor or other service provider, subject to the specifications of Exhibit A and that OSP Dispatch Support will facilitate communications between the ODOT TOCS Project and OSP CAD. ODOT is responsible for all interfacing software involving all other entities (i.e. Downstream of the interface hardware and software between OSP CAD and TOCS).

OSP OBLIGATIONS

1. OSP shall operate the interface hardware and software system of behalf of ODOT. The system shall meet ODOT's requirements as listed in Exhibit A.

2. ODOT and OSP will have data that can be shared with members of the public and data that must be considered confidential - subject to freedom of information act constraints - both systems will need to support the filtering and security of confidential information. Neither ODOT nor OSP shall release CAD data of the other and requests for such data shall be forwarded to the appropriate agency representative.

3. OSP will be responsible for the maintenance of their CAD software and databases and, on behalf of ODOT, for operation of the System.

4. OSP Dispatch Support shall retain the right to cease System operations in the event the System is causing or suspected of causing interference with on-going CAD System Operations. In the event of such cessation of operation, OSP Dispatch Support shall promptly notify each individual ODOT Transportation Operations Center.

5. OSP's point of contact is Jerold Martin, Oregon State Police, 3225 State St SE, Salem, OR 97301 or mailing address PO Box 14360, Salem, OR 97309. Phone 503-378-8750. Email: jerold.martin@state.or.us, or assigned designee upon individual's absence. ODOT's Project Manager shall be notified in writing of any contact

information changes during the term of this Agreement.

ODOT OBLIGATIONS

1. ODOT will be responsible for the cost of the System and for procurement of required hardware and software comprising the Interface.

2. ODOT agrees all users with security access to OSP CAD Data, including but not limited to Calls for Service and Law Enforcement Data Systems (LEDS) or DMV information, will be fingerprinted for Criminal Justice Information System (CJIS) background purposes. Also anyone that has unescorted access to the immediate location in which the OSP CAD information accessible TOCS console(s) resides will be finger printed. OSP calls-for-service records are not to be viewed by any persons who have not been OSP operator certified until the record has been transferred as redirected call for service to ODOT. In addition, any vendor, working on ODOT's behalf, with access to OSP CAD information will be fingerprinted for CJIS, undergo a background check, and sign a non-disclosure of information form.

3. Message infrastructure and connectivity hardware will be the responsibility of ODOT to purchase, or lease and maintain or to provide access to.

4. ODOT will be responsible for maintaining a role based security model that limits access to OSP data based on levels of security permission granted.

5. ODOT will not directly connect to the OSP CAD application server.

6. ODOT will be responsible for the maintenance of the ODOT software and databases.

7. ODOT's Project Manager for this Project is Galen McGill, 800 Airport Rd SE Rm 81, Salem, OR 97301-4798, Phone: 503-986-4486 or assigned designee upon individual's absence. ODOT's Project Manager shall be notified in writing of any contact information changes during the term of this Agreement.

JOINT OBLIGATIONS

1. A project development team (Team), consisting of at least one representative from each Party, shall participate in the following:

   a. The Team shall provide technical assistance on an as needed basis to answer technical questions related to the integration.

   b. An interface contract will be required to define the translation of data elements between systems as well as data format and protocol of data exchange.

GENERAL PROVISIONS

1. This Agreement may be terminated by mutual written consent of both Parties.

2. Upon conclusion or termination of the development phase of this project, OSP shall retain authority as the system operator of the CAD interface developed herein. OSP shall retain access to developed interface hardware and software, from whatever source.

3. By participation in this project, it is the intent of OSP that access to this technology shall be extended to other public safety dispatch facilities as soon as practical, after the system is operational.

4. The Parties agree that any tort liability claim, suit, or loss resulting from or arising out of the Parties' performance of and activities under this Agreement shall be allocated, as between the state agencies, in accordance with law by the Risk Management Division of the Department of Administrative Services (DAS) for purposes of their respective loss experiences and subsequent allocation of self-insurance assessments under ORS 278.435. Each Party to this Agreement agrees to notify the Risk Management Division and the other agency in the event it receives notice or knowledge of any claims arising out of the performance of, or the agencies' activities under this Agreement.

5. The Parties understand that each is insured with respect to tort liability by the State of Oregon Insurance Fund, a statutory system of self-insurance established by ORS 278, and subject to the Oregon Tort Claims Act (ORS 30.260-30.300). Each Party agrees to accept that coverage as adequate insurance of the other Party with respect to personal injury and property damage.

6. This Agreement may be executed in several counterparts (facsimile or otherwise) all of which when taken together shall constitute one agreement binding on all parties, notwithstanding that all parties are not signatories to the same counterpart. Each copy of this Agreement so executed shall constitute an original.

7. This Agreement and attached exhibits constitute the entire agreement between the Parties on the subject matter hereof. There are no understandings, agreements, or representations, oral or written, not specified herein regarding this Agreement. No waiver, consent, modification or change of terms of this Agreement shall bind either Party unless in writing and signed by both Parties and all necessary approvals have been obtained. Such waiver, consent, modification or change, if made, shall be effective only in the specific instance and for the specific purpose given. The failure of ODOT to enforce any provision of this Agreement shall not constitute a waiver by ODOT of that or any other provision.

THE PARTIES, by execution of this Agreement, hereby acknowledges that its signing representatives have read this Agreement, understand it, and agree to be bound by its terms and conditions.

The Oregon Transportation Commission on December 29, 2008, approved Delegation Order No. 2, which authorizes the Director to approve and execute agreements for day-to-day operations. Day-to-day operations include those activities required to implement the biennial budget approved by the Legislature, including activities to execute a project in the Statewide Transportation Improvement Program.

On September 15, 2006, the Director of the Oregon Department of Transportation approved Subdelegation Order No. 2, in which the Director delegates to the Deputy Director, Highways the authority to approve and sign agreements over $75,000 when the work is related to a project included in a line item in the biennial budget approved by the Director.

OSP/ODOT
Agreement Number 26630

STATE OF OREGON, by and through its Oregon State Police Department

By _____

Date 8/10/10

By _____

Date _____

APPROVED AS TO LEGAL SUFFICIENCY

By _____
OSP Legal Counsel (Optional)

Date _____

OSP Contact: Jerold Martin
3225 State Street SE
Salem, Oregon 97301

STATE OF OREGON, by and through its Department of Transportation

By _____
Deputy Director, Highway Division

Date 8/17/10

APPROVAL RECOMMENDED

By _____
Statewide Maintenance & Operations Engineer

Date 8/13/10

By _____
ITS Manager

Date 8/13/10

ODOT Contact:
Galen McGill
ODOT
800 Airport Rd
SE Rm 81
Salem, OR 97301-4798
Phone: 503-986-4486

Exhibit A
System Requirements

A. ODOT requires that the Interoperable data system (System) meets the following requirements:

1. The System shall:
   a. Support one-to-many and many-to-one message transfer.
   b. Support TCP/IP protocol.
   c. Support requests for data.
   d. Support a Web Services Architecture.
   e. Utilize Standardized SOAP Protocol.
   f. Utilize Standardized Web Services Definition Language (WSDL).
   g. Utilize Standardized Web Services XML.
   h. Utilize Standardized Web Services XML Schema Definition (XSD).
   i. Be highly available.
   j. Support user authentication.
   k. Utilize Standard Web Services Security.
   l. Log all system errors and alerts.
   m. Provide a mechanism for monitoring system performance.
   n. Provide a mechanism for monitoring system utilization.
   o. Support a scalable architecture that can accommodate growth in the number of clients.
   p. Provide data to subscribers based on message types, location, and priority.
   q. Support two-way message transmissions that include return receipts/responses.
   r. Support XML data schema for IEEE 1512-2000 (Common Incident Management Message Sets (IMMS) for use by EMCs.
   s. All message interfaces will be W3C complaint web service connections.
   t. Neither system will have access to functions in the others CAD system; all data access is read-only, messages will be sent with data for the receiving system to filter and write to its own persistent data storage as configured by the receiving system.

u. No call for service will be considered as being handed off until the receiving system operator has acknowledged receipt and ownership of the call.

v. Security- authentication between system by digital certificates or SSL connections using Basic Authentication with encryption. Optional security functionality such as registered IP addresses to be based on mutual agreement.

w. Standard message sets to be implemented with the IEEE-1512 group of standards for common incident management messages.

B. In addition, ODOT requires that the System provide the following CAD data to ODOT for incidents on state highways when the information is available:

1. Incident ID
2. Incident Location – including Highway and Milepost when applicable
3. Incident type
4. Incident severity
5. Incident impact to travel
6. Highway lane blockages or closures
7. Incident responders
8. Estimated duration of incident
9. Request for service from ODOT

C. TOCS Requirements for integration with OSP CAD

The requirements listed below are defined at a high level to capture all of the functionality currently used between ODOT and OSP.

1. The TOCS interface to OSP will enable the transfer of incident records between OSP CAD and ODOT TOCS.
   a. The TOCS interface to OSP will enable authorized TOCS users read data access to all published OSP incidents (calls) – both current and archival.
   b. The TOCS interface to OSP will enable all OSP CAD users read data access to all published TOCS incidents.
   c. The TOCS interface to OSP CAD will enable authorized TOCS users read access to all OSP radio traffic attached to a published call or unit.
   d. The TOCS interface to OSP CAD will enable all OSP CAD users read data access of all published TOCS radio traffic that is attached to an incident or unit.

e. The TOCS system shall enable redirection of incidents between TOC centers and OSP. (Example: ODOT crew calls in a vehicle, TOC runs plate check, if the vehicle is stolen: the crew is notified, the incident is redirected to OSP, and the ODOT crew leaves the scene). These requirements apply to both open and closed incidents.
    i. Authorized TOCS users shall be able to select and copy (or reopen and redirect) published incidents from OSP CAD
    ii. Authorized TOCS users shall be able to send copies (or reopen and redirect) of incidents to OSP CAD.
    iii. TOCS incidents that are redirected to OSP CAD will include all radio traffic attached to the incident at the time the incident is sent.
    iv. OSP CAD users shall be able to select and copy (or reopen and redirect) published incidents from ODOT TOCS
    v. OSP CAD users shall be able to send (or reopen and redirect) incidents to ODOT TOCS
    vi. When an incident is redirected to OSP CAD, TOCS shall document the redirection in the remarks of the TOCS incident.
f. For published incidents that are copied (or reopened and redirected) from OSP CAD, TOCS will make all OSP radio traffic attached to that incident "read only" within the TOCS system.
g. TOCS Interface to OSP will support geo-coding (geo-basing of location) of incidents.

2. Incidents within both systems will have the capability to be related to other incidents within both TOCS and OSP CAD. The TOCS interface to OSP will enable authorized TOCS users read data access to all published OSP incidents (calls) – both current and archival.
    a. The TOCS system shall support an incident having multiple related incident numbers.
    b. The TOCS system shall maintain all OSP CAD incident numbers related to a TOCS incident.
    c. The TOCS system shall include the TOCS incident number as a related incident number on all TOCS incidents copied to OSP CAD.
    d. The TOCS system shall provide a method of identifying the type of relationship of the related incident numbers. (Examples of relationship types: from OSP CAD, sent to OSP and Parent/child within TOCS.)
    e. The TOCS system shall allow TOCS users to view a report of all related incidents.
3. The TOCS connection to OSP shall include automatic updates for shared incidents. (Example: any call that has been copied (or reopened and redirected) to OSP from ODOT will automatically be updated to notify OSP when ODOT closes the call)
4. The TOCS shall provide Towing Dispatch functionality using the OSP Tow List.

5. The TOCS system shall automatically send an incident to OSP CAD when ODOT staff is in distress (as currently indicated by 12-98 and 12-99 codes).
6. The TOCS Interface to OSP CAD will support "Screen to Screen Messaging" functionality between TOCS users and OSP CAD users.
7. The TOCS system will **NOT** provide direct access to LEDS. Administrative Messaging and other LEDS communication is currently available and will continue to be available through ForseCom.
8. The endpoint web services for TOCS and OSP CAD will be capable of direct connection through a network or VPN in the event of Sonic ESB failure or lack of availability.

D. No license fees shall be charged by either Party for the data access.

ACRONYMS

CAD	Computer Aided Dispatch
CJIS	Criminal Justice Information System
DMV	Driver and Motor Vehicle Services
EMCs	Emergency Management Centers
ESB	Enterprise Services Bus
ID	Identification
IEEE	Institute for Electrical and Electronics Engineers
IMMS	Incident Management Message Sets
IP	Internet Protocol
LEDS	Law Enforcement Data Systems
PDCC	Portland Dispatch Center Consortium
PSSI	Public Safety Systems Inc.
SCA	System Connection Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TOC	Transportation Operations Center
TOCS	Transportation Operations Center System
W3C	World Wide Web Consortium
WSDL	Web Services Definition Language
XML	eXtensible Markup Language
XSD	eXtensible Schema Definition

OSP/ODOT 26630
Agreement Number ~~22897~~

Exhibit B
SCA Specification PDCC-CAD Integration

UASI CAD to CAD EIS Project

CAD - Enterprise Service Bus (ESB) Connection Overview

Prepared for:
City of Portland

Prepared by:

Online Business Systems

One World Trade Center
121 Salmon Street S.W. 11th Floor
Portland, Oregon 97204

Table of Contents

1.0 INTRODUCTION

The Purpose of this document is to describe the requirements for the connection between a CAD System and the Enterprise Service Bus (ESB). The objective of this connection architecture is to give the CAD Vendors the most flexibility in their selection of toolsets to implement the connection architecture while providing a generic and abstracted interface between all CAD systems and the ESB.

The ESB will expose several Web Service access points to all CAD locations. These will be specified in a document literal WSDL defining multiple operations. The operations will correspond to the business scenarios outlined in the Business Scenarios document (CFS, Update, Acknowledgement, Information Only, Heartbeat, Schema Update, Error).

1.1 Problem Statement

The solution architecture must solve the following:

1. Place well formed XML content on the ESB
2. Wait for a reply when a reply is expected as part of the process
3. Asynchronously accept incoming documents containing well formed XML content
4. Be able to read and write all XML content as described in this document and the Message Exchange documentation
5. Guarantee delivery between the systems by handling exception conditions
6. Provide a secure environment for the message exchanges

1.2 Reference Documentation

Date	Ver	Document	Description
	1.0	CFI XML Schema	The schema reference that ensures an XML document is well formed
June 1999	1.1	http://www.ietf.org/rfc/rfc2616.txt	HTTP Specification
June 1999	1.1	http://www.faqs.org/rfcs/rfc2617.html	HTTP Authentication and Digest Access Authentication
April 2000	1.0	http://schemas.xmlsoap.org/specs/ws-security/ws-security.htm	Web Service Security Language specification. Used to express and share security information in a standard format
Dec 2002	1.0	http://www.w3.org/TR/xmlenc-core/	XML Encryption Syntax and Processing standards. Used by WS-Security standards to protect XML documents

2.0 BUSINESS SCENARIOS

The following Scenarios define the business scope for this project and are required to be supported by both the ESB and the CAD systems. Any operational details will need to be ratified by the CAD Vendors with their respective agency representatives. It is not the intent of this document to dictate operational details of the CAD systems beyond identifying the integration business scenarios that are required to be supported by the PDCC.

These scenarios include call-outs for business scenarios that are specific to the CAD Systems and are out of scope for the ESB architecture.

2.1.1 CFS Transfer

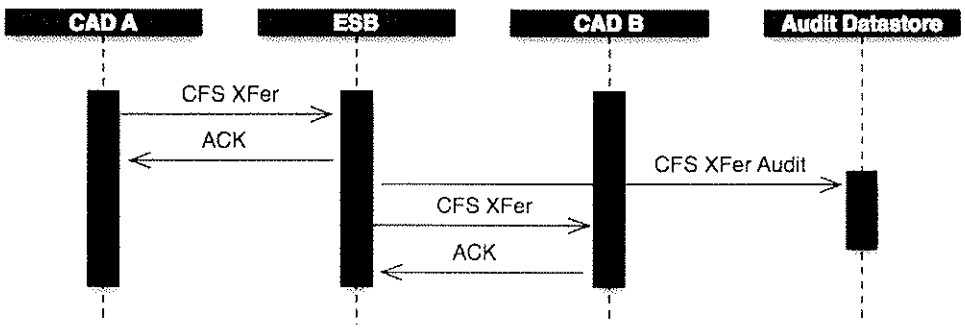

Figure 1

Narrative

The CFS Transfer Message represents the transfer of incident information from one PSAP to another. In this scenario:

1. CAD A, representing a CAD system at a PSAP, is triggered to transfer incident information to a destination, in this case CAD B which represents a CAD system at another PSAP.
2. CAD A creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD A will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.

4. The ESB will apply a message itinerary to the CFS Transfer which will determine how to route the message
5. The first step in the itinerary will be to create an audit record of the message. In this step the ESB sends an audit message, including all of the CFS data, to the Audit Datastore
6. The message is then routed to its destination (In the example in figure 2, this is CAD B) and the reverse process occurs. The web service layer on the ESB transfers the message to the Web Service implementation at the PSAP and the transfer is complete.
7. Once completed the result of the transfer is itself audited and becomes a part of the audit history for this CFS Transfer.

Alternate Paths

CAD A cannot connect to ESB

- In this scenario, CAD A is unable to connect the ESB.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A does receive acknowledgement from the Enterprise Service Bus

- In this scenario, CAD A does not receive an acknowledgement back from the ESB.
- This should lead the CAD system to determine that the message was NOT successfully transferred to the bus.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A does not receive acknowledgement or Rejection from CAD B

- In this scenario, CAD A successfully sends the message but does not receive an acknowledgement or rejection from CAD B as describe in Section 2.1.3.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A receives an Error from ESB

- In this scenario, CAD A successfully sends the message but receives an error message back from the bus. This could be an invalid route error, and error to indicate the ESB could not successfully deliver the message to the destination, or possibly an error on the bus itself which would indicate the message was not deliverable. Should any of these messages be received, the CAD system to determine that the message was NOT successfully received at the destination.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A provides a message with an invalid destination

- In this scenario, on of the destinations provided in the message is not valid. Please refer to Section 2.1.6 for a complete description of this scenario.

ESB is unable deliver the message

- In this scenario, the ESB cannot deliver the message in a pre-determined number of retries/time. Please refer to Section 2.1.6 for a complete description of this scenario.

2.1.2 CFS Update

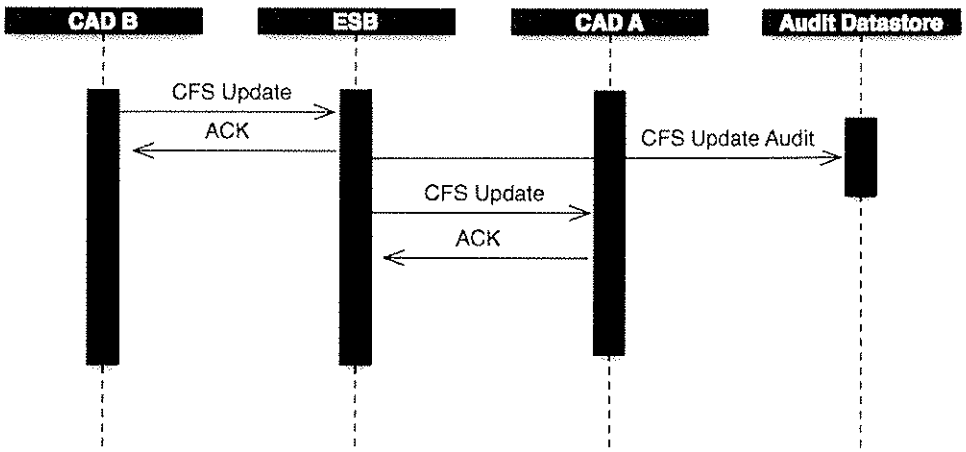

Figure 2

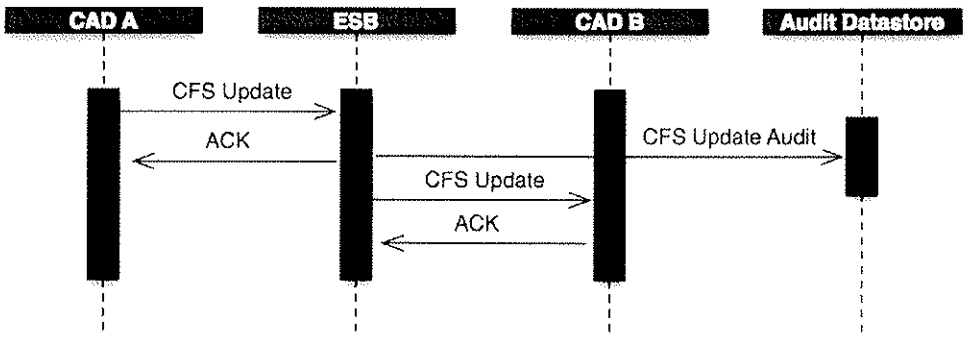

Figure 3

Narrative

The CFS Updates has two valid scenarios. In Figure 2 we see CAD B, which received the initial CFS transfer, sends an update to the incident, back to the originating CAD,

CAD A. In Figure 3 we see CAD A, the CAD system that sent the initial CFS to CAD B, sends an update to CAD B. From a workflow perspective, both scenarios follow the same path, however, we will use Figure 2 for the narrative:

1. CAD B, representing a CAD system at a PSAP, is triggered to send an update to an previously sent by CAD A.
2. CAD B creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD B will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. The ESB will apply a message itinerary to the CFS Transfer which will determine how to route the message
5. The first step in the itinerary will be to create an audit record of the message. In this step the ESB sends an audit message, including all of the CFS data, to the Audit Datastore
6. The message is then routed to its destination (In the example, CAD A) and the reverse process occurs. The web service layer on the ESB transfers the message to the Web Service implementation at the PSAP and the transfer of the Incident Update is complete.

Alternate Paths

A CAD system cannot connect to ESB

- In this scenario, CAD A is unable to connect the ESB.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system does receive acknowledgement from the Enterprise Service Bus

- If a CAD system does not receive an acknowledgement back from the ESB, this should lead the CAD system to determine that the message was NOT successfully transferred to the bus. Should this occur, the PSAP and CAD vendor will need to agree on what action should be taken by the CAD system. This path is out of scope of the ESB.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system does not receive acknowledgement or Rejection from another CAD system

- In this scenario, a CAD system successfully sends the message but does not receive an acknowledgement or rejection from the CAD system is was sending to.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system receives an Error from ESB

- In this scenario, a CAD system successfully sends the message but receives an error message back from the bus. This could be an invalid route error, and error to indicate the ESB could not successfully deliver the message to the destination, or possibly an error on the bus itself which would indicate the message was not deliverable. Should any of these messages be received, the CAD system to determine that the message was NOT successfully received at the destination.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system provides a message with an invalid destination

- In this scenario, on of the destinations provided in the message is not valid. Please refer to Section 2.1.6 for a complete description of this scenario.

ESB is unable deliver the message

- In this scenario, the ESB cannot deliver the message in a pre-determined number of retries/time. Please refer to Section 2.1.6 for a complete description of this scenario.

2.1.3 Message Acknowledgement/Rejection

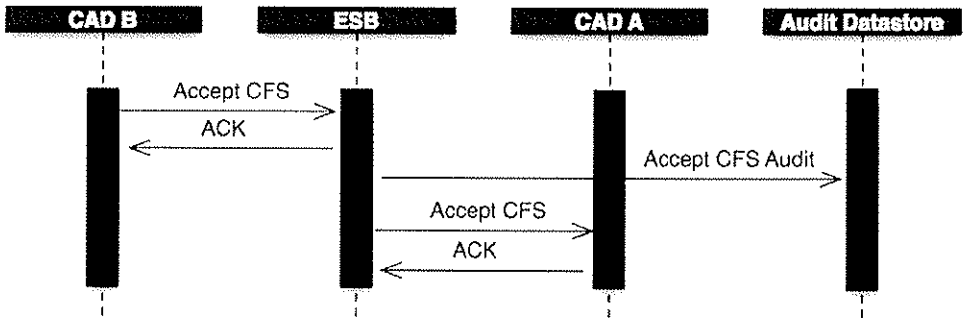

Figure 4

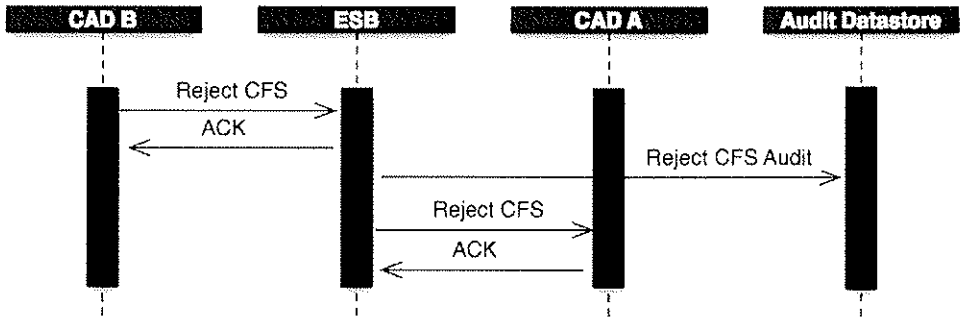

Figure 5

Narrative

Figure 4 and Figure 5 show the two possible scenarios of a CAD user responding to a message he/she received. In the first scenarios, a user at CAD B is accepting the message he/she received from CAD A. In the second, a user at CAD B is rejecting the CFS transfer they received from CAD A. under the current design, CFS Transfer and CFS Updates both require an acknowledgment from the receiving CAD user, whereas Information Only message do not require this acknowledgement. The requirement for an acknowledgement is determined by the value of a data element in the message structure.

It is up to the sending and receiving CAD systems to ensure that on the sending CAD, it keeps track of acknowledgements it is receiving, and on the receiving side, it prompts the user to acknowledge the message. This is particularly important if there is a communication error and the ESB cannot transmit the message or acknowledgement as expected.

1. CAD B, representing a CAD system at a PSAP, triggers a user to acknowledge (accept or reject) a message originally sent from CAD A.
2. CAD B creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD B will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. The ESB will apply a message itinerary to the CFS Transfer which will determine how to route the message
5. The first step in the itinerary will be to create an audit record of the message. In this step the ESB sends an audit message, including all of the CFS data, to the Audit Datastore
6. The message is then routed to its destination (In the example, CAD A) and the reverse process occurs. The web service layer on the ESB transfers the message to the Web Service implementation at the PSAP and the transfer of the Incident Update is complete.

Alternate Paths

CAD A does not receive an acknowledgement from CAD B in a timely manner

- This scenario occurs is CAD A send a message that it expects a user acknowledgement to, and CAD A does not receive that acknowledgement in the timeframe it is expecting it.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system cannot connect to ESB

- If a CAD System is unable to connect the ESB the PSAP and CAD vendor will need to agree on what action should be taken by the CAD system.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system does receive acknowledgement from the Enterprise Service Bus

- If a CAD system does not receive an acknowledgement back from the ESB, this should lead the CAD system to determine that the message was NOT successfully transferred to the bus.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system does not receive acknowledgement or Rejection from another CAD system

- In this scenario, a CAD system successfully sends the message but does not receive an acknowledgement or rejection from the CAD system is was sending to.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system receives an Error from ESB

- In this scenario, a CAD system successfully sends the message but receives an error message back from the bus. This could be an invalid route error, and error to indicate the ESB could not successfully deliver the message to the destination, or possibly an error on the bus itself which would indicate the message was not deliverable. Should any of these messages be received, the CAD system to determine that the message was NOT successfully received at the destination.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

A CAD system provides a message with an invalid destination

- In this scenario, on of the destinations provided in the message is not valid. Please refer to Section 2.1.6 for a complete description of this scenario.

ESB is unable deliver the message

- In this scenario, the ESB cannot deliver the message in a pre-determined number of retries/time. Please refer to Section 2.1.6 for a complete description of this scenario.

2.1.4 Multiple Destination Routing

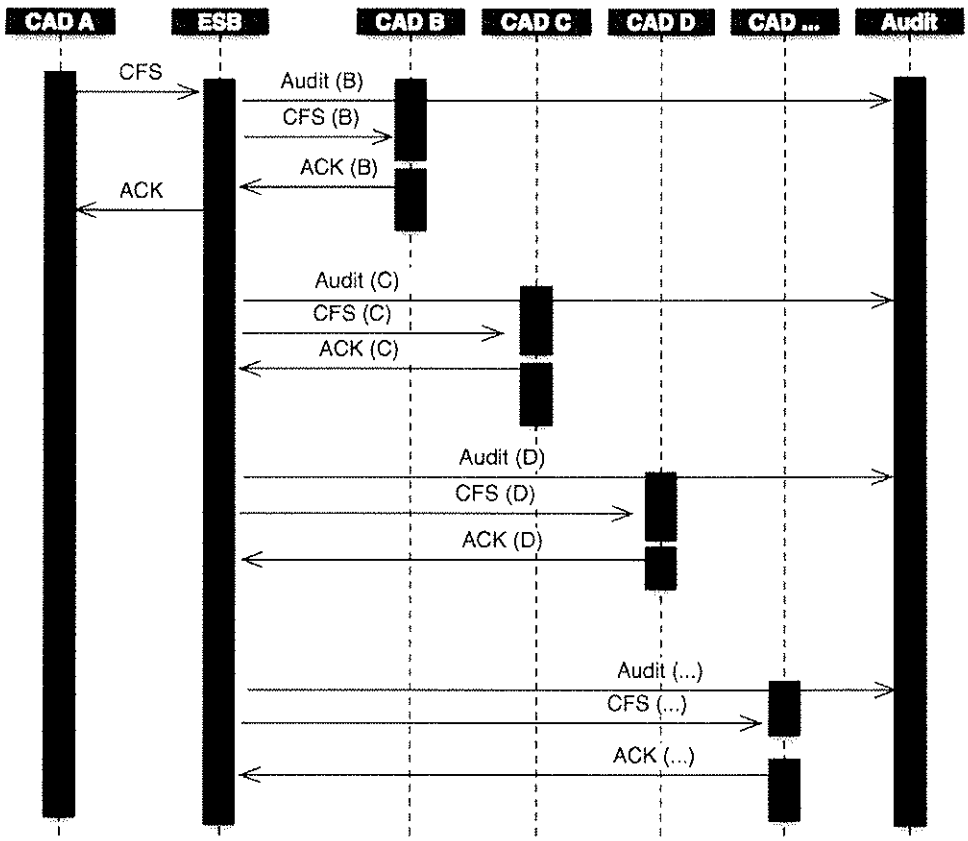

Figure 6

Narrative

This scenario involves the sending a single message (in this case an initial CFS Transfer) to multiple destinations. This scenario provides a one to many, or broadcast capability for PSAPs to send information efficiently to multiple destinations.

In this scenario:

1. CAD A, representing a CAD system at a PSAP, is triggered to transfer incident information to multiple destinations, in this case CAD B,C, & D which represents CAD systems at other PSAPs.
2. CAD A creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD A will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. The ESB will apply a message itinerary to the CFS Transfer which will determine how to route the message to all of the valid destinations
5. The first step in the itinerary will be to create an audit record of the message. In this step the ESB sends an audit message, including all of the CFS data, to the Audit Datastore
6. The message is then routed to its destinations and the reverse process occurs. The web service layer on the ESB transfers the message to the Web Service implementation at the PSAP and the transfer is complete.
7. Once completed the result of the transfer is itself audited and becomes a part of the audit history for this CFS Transfer.

Alternate Paths

CAD A provides a message with an invalid destination

- In this scenario, on of the destinations provided in the message is not valid. Please refer to Section 2.1.6 for a complete description of this scenario.

CAD A cannot connect to ESB

- In this scenario, CAD A is unable to connect the ESB.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A does receive acknowledgement from the Enterprise Service Bus

- In this scenario, CAD A does not receive an acknowledgement back from the ESB. This should lead the CAD system to determine that the message was NOT successfully transferred to the bus.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A does not receive acknowledgement or Rejection from CAD B

- In this scenario, CAD A successfully sends the message but does not receive an acknowledgement or rejection from CAD B as describe in Section 2.1.3.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A receives an Error from ESB

- In this scenario, CAD A successfully sends the message but receives an error message back from the bus. This could be an invalid route error, and error to indicate the ESB could not successfully deliver the message to the destination, or possibly an error on the bus itself which would indicate the message was not deliverable. Should any of these messages be received, the CAD system to determine that the message was NOT successfully received at the destination.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

ESB is unable deliver the message

- In this scenario, the ESB cannot deliver the message in a pre-determined number of retries/time. Please refer to Section 2.1.6 for a complete description of this scenario.

2.1.5 Information Only Message

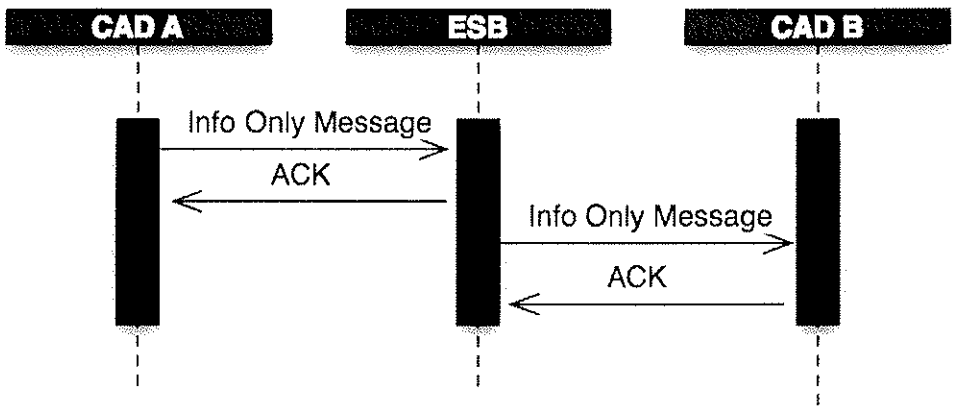

Figure 7

Narrative

Information Only messages are designed to provide a feature similar to an "instant message". Although not as interactive as a traditional IM type application in that the

message cannot be sent to an individual, only a PSAP, the information only message nevertheless provides the ability to send messages to PSAPs that do not require acknowledgement and do not necessarily require a response by the receiving PSAP.

The sequence of events for this scenario are:

1. CAD A, representing a CAD system at a PSAP, is triggered to send an informational message to another PSAP represented by CAD B.
2. CAD A creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD A will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. The ESB will apply a message itinerary to the Informational Message which will determine how to route the message to all of the valid destinations The message is then routed to its destinations and the reverse process occurs. The web service layer on the ESB transfers the message to the Web Service implementation at the PSAP and the transfer is complete.

Alternate Paths

CAD A receives an Error from ESB

- In this scenario, CAD A successfully sends the message but receives an error message back from the bus. This could be an invalid route error, and error to indicate the ESB could not successfully deliver the message to the destination, or possibly an error on the bus itself which would indicate the message was not deliverable. Should any of these messages be received, the CAD system to determine that the message was NOT successfully received at the destination.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

CAD A provides a message with an invalid destination

- In this scenario, on of the destinations provided in the message is not valid. Please refer to Section 2.1.6 for a complete description of this scenario.

ESB is unable deliver the message

- In this scenario, the ESB cannot deliver the message in a pre-determined number of retries/time. Please refer to Section 2.1.6 for a complete description of this scenario.

CAD A cannot connect to ESB

- In this scenario, CAD A is unable to connect the ESB.

This scenario is out of scope of the ESB implementation. In this case, the Call Center representatives and the CAD vendor will need to agree on what action should be taken by the CAD system.

2.1.6 Error Condition

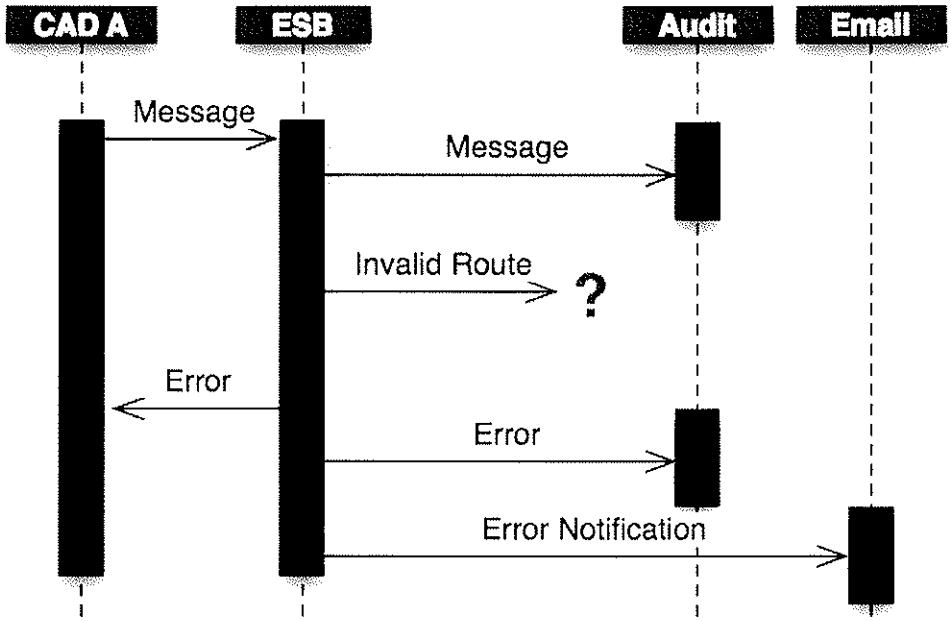

Figure 8

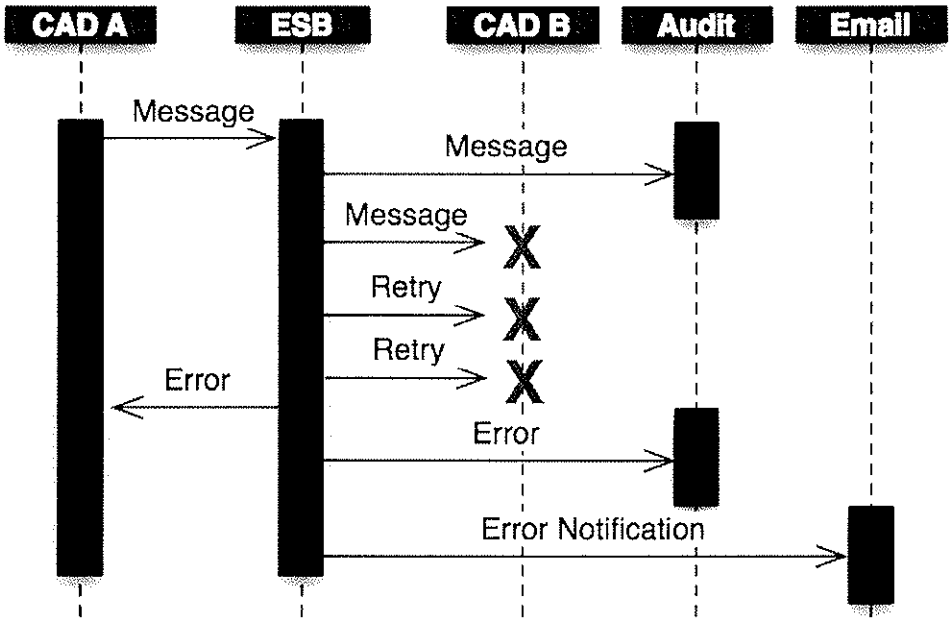

Figure 9

Narrative

Error scenarios are alternate paths from the other business scenarios described in this document. Error conditions can fall into one of three types: Invalid route Errors, Unable to Deliver Errors and Unexpected ESB Errors.

The sequence of events for these three types are:

Invalid Route

1. CAD A, representing a CAD system at a PSAP, is triggered to send a message to another PSAP represented by CAD B.
2. CAD A creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD A will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. The ESB will apply a message itinerary to the Informational Message which will determine how to route the message. When the routing rules cannot find a matching destination on the bus for the one specified in the message, the ESB will put this message into an error itinerary.
5. The error itinerary will complete several tasks
    a. Generate an error message that identifies an invalid route as the error, and send that along with the original message to the sending PSAP (CAD A)
    b. Audit the error message
    c. Send an email notification to a predefined list of recipients to alert them to error.

Unable to deliver Message

1. CAD A, representing a CAD system at a PSAP, is triggered to send a message to another PSAP represented by CAD B.
2. CAD A creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD A will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. The ESB will apply a message itinerary to the Informational Message which will determine how to route the message.
5. The first step in the itinerary will be to create an audit record of the message. In this step the ESB sends an audit message, including all of the CFS data, to the Audit Datastore
6. The message is then routed to its destinations and delivery attempts are made. If the ESB is unable to deliver the message within a pre-defined number of attempts over a pre-defined period of time, the ESB will put this message into an error itinerary.
7. The error itinerary will complete several tasks

a. Generate an error message that identifies that it was unable to deliver the message, and send that along with the original message to the sending PSAP (CAD A)
b. Audit the error message
c. Send an email notification to a predefined list of recipients to alert them to error.

Unexpected Error

1. CAD A, representing a CAD system at a PSAP, is triggered to send a message to another PSAP represented by CAD B.
2. CAD A creates an XML formatted message that conforms to the message architecture in use for this message exchange (see Section **Error! Reference source not found.** of this document for a complete description of the message architecture)
3. The web service for CAD A will connect to the ESB via a web service layer in front of the ESB and transfer the message. The Sonic ESB will receive the data as a JMS Message.
4. If at any point on the ESB an unknown or unexpected error occurs, an error itinerary will be applied to the message.
5. The error itinerary will complete several tasks
    a. Generate an error message that identifies the error with what information it has, and send that along with the original message to the sending PSAP (CAD A)
    b. Audit the error message
    c. Send an email notification to a predefined list of recipients to alert them to error.

2.1.7 System Heartbeat

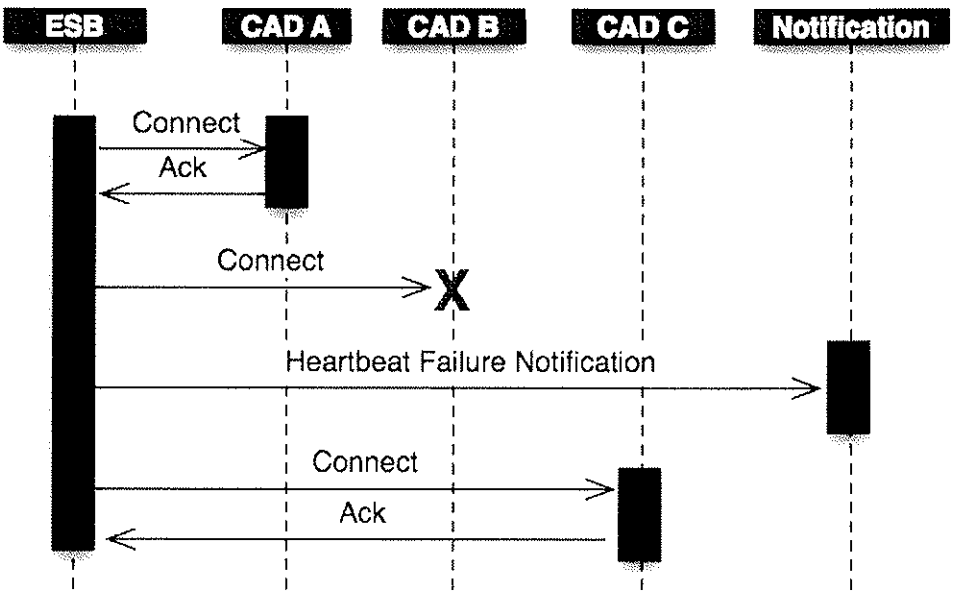

Figure 10

Narrative

System Heartbeat is a scenario designed pro-actively look for connectivity issues on the ESB. It will ideally provide advance warning of connection issue, so that PSAP staff can implement business continuity plans in the event that destinations cannot be reached by the ESB.

The sequence of events for these this scenario are:

1. Based on a pre-defined schedule, the ESB will attempt to connect to all of the valid destinations on the ESB.
2. If any connection does not acknowledge the connection attempt, the ESB will send an email notification to a predefined list of recipients to alert them to error.

2.1.8 Distribute Updated Message Schema

Narrative

The message schemas define, not only the valid data elements, but in some cases, the actual list of valid values such as valid destinations. In order to more efficiently distribute changes to the schemas, this scenario provides for distribution via the ESB.

The sequence of events for these three types is:

1. An administrator for the ESB will place the updated schemas on the bus.
2. The ESB will then route these messages to all valid destinations on the bus.

Alternate Paths

ESB is unable deliver the message

- In this scenario, the ESB cannot deliver the message in a pre-determined number of retries/time. At that time, the ESB will distribute email notifications of the error.

2.2 Message Routing and Filtering Rules

There currently no additional Routing or Filtering Rules that have been identified as in scope.

3.0 SOLUTION SET

3.1 Overview

This solution-set represents the technical approach for the incoming and outgoing communication between the PDCC Enterprise Service Bus (ESB) and the CAD Systems. The ESB will expose several Web Service access points to all Call Centers. These will be specified in a document literal WSDL defining multiple operations. The operations will correspond to the business scenarios outlined in the Business Scenarios section of this document.

This architecture specifies that all communication between the ESB and CAD Systems is based on a web-services model. This means that the ESB will act as a Web Service Host for incoming messages from CAD Systems, as well as a web service client for outgoing messages begin sent to a CAD System.

As part of the design of this system, 2 WSDLs will be provided to the participating Call Centers. The first WSDL will describe the methods available on the ESB, and the other will describe the interface that the ESB is expecting at the Call Center in order for the ESB to communicate with the CAD system.

3.2 Security

The Web Service acceptors will be available over https encrypted with:

RSA with 128-bit AES CBC SHA

The ESB will authenticate each user using Basic Authentication (username and password). It is assumed that the CAD Systems will provide the same security architecture for the ESB to post to the CAD Systems.

It should be noted for the encryption cipher, that the minimum requirement is 128-bit FIPS certified ciphers. Within this requirement, we look forward to working with the CAD vendors to produce a solution that meets all Parties' needs.

3.3 ESB to CAD Messaging

To facilitate ESB to CAD messaging, a CAD System must expose a series of web service operations to the ESB. These operations will resemble the Business Case scenarios plus a delivery failure operation (used to notify the CAD when errors delivering a message occur).

These business scenario operations will match those exposed by the ESB along with additional operations for system heartbeat and schema updates.

Call Centers will need to identify whither or not they will support multiple URLs for the ESB to access the exposed operations. This will be based on each Call Centers ability to support failover features

3.3.1 Web Service Call

The ESB will call the appropriate message on the CAD location with the message to be delivered. The Delivery Failure Operation will be invoked f the ESB experienced exceptions will trying to deliver a previous message sent from this CAD location.

3.3.2 WSDL

Exposed operations will be exposed via a document literal WSDL that will be initially distributed by Online Business Systems and will be unique to each CAD system. This WSDL will be jointly ratified by both Online and the individual CAD Vendors.

3.3.3 Delivery Failure

As part of the standard operation list, each CAD must expose a Delivery Failure Notification operation. This is called by the ESB when exceptions are encountered while attempting to delivery a message originating at this CAD location.

3.3.4 Message Types

The ESB will consume a Web Service provided by the CAD system. The WS request will contain the XML message of one of the following types:

Call for Service

These are all CFS messages that the ESB has previously received from a CAD system (see Incoming Communication Section) and must be sent to the destination CAD System(s)

Call for Service Accept/Reject

These are all Acknowledgement or Rejection messages that the ESB has previously received from a CAD system (see Incoming Communication Section) and must be sent to the destination CAD System.

Call for Service Update

These are all CFS Update messages that the ESB has previously received from a CAD system (see Incoming Communication Section) and must be sent to the destination CAD System.

Information Only

These are all Information Only messages that the ESB has previously received from a CAD system (see Incoming Communication Section) and must be sent to the destination CAD System.

Heartbeat

The heartbeat method is only intended to provide an "OK" response to the post, which will tell the ESB that the Call Center it was posting to is in fact available to the ESB for message traffic.

Error

These are all messages created by the ESB and sent to a CAD Systems, should an error occur.

Message Schema Update

This method is for distribution of updated message schemas if required. This would be done in the case of a new destination being added to the ESB for example.

3.4 CAD to ESB Messaging

The ESB will have multiple access points for the Call Centers to connect to, each accessed as an https URL. These URLs will provide access both to backup locations within a physical site as well as secondary site URLs for full failover. Currently we are planning 4 such URLs. These URL will be provided to the CAD in a priority-ordered list. Upon the event of a delivery failure a

CAD must try sending the message to the next URL on the list. The list of access points may be customized for each CAD to load balance ESB access across physical locations and servers.

3.4.1 Web Service Call

CAD Systems will invoke the appropriate operation on the ESB using the primary URL. This message will contain an XML message that adheres to the XSD for the message type (CFS, Update, etc). As soon as the ESB receives the message, authenticates the user and validates the XML message a standard Web Service response is sent back to Web Service consumer (the CAD). Once the CAD receives the Web Service response it can consider the message delivered to the ESB. It is important to note that this Web Service acknowledgement does not mean that the message has been delivered to the destination CAD locations.

3.4.2 WSDL

Exposed operations will be exposed via a document literal WSDL that will be initially distributed by Online Business Systems and will be unique to each CAD system. This WSDL will be jointly ratified by both Online and the individual CAD Vendors.

3.4.3 Delivery Failure

Each message delivered to the ESB will specify one or more destination CAD locations. The ESB will attempt to delivery the message to all of these locations. In the event that one or more of these locations are unreachable or do not acknowledge receipt of the message then the ESB will invoke a Delivery Failure Notification Web Service operation on the source CAD location. This operation will be invoked once per delivery failure. This means that if a CAD has specified multiple delivery locations in a message sent to the ESB it can receive multiple delivery failure notifications back from this single message. The delivery failure message will contain the original message as well as basic details as to the nature of the failure.

3.4.4 Message Types

Call for Service

These are all messages being sent to the ESB from a CAD system representing a call for service that must be routed to a destination by the ESB

Call for Service Accept/Reject

These are all messages being sent to the ESB from a CAD system representing the acceptance or rejection of a previously sent CFS

Call for Service Update

These are all messages being sent to the ESB from a CAD system representing an update to a previously sent CFS

Information Only

These are all messages being sent to the ESB from a CAD system representing an informational message that must be routed to a destination by the ESB

4.0 HTTP / SOAP

The following HTTP Headers are used in all web services:

Header Value	Required	Description
Authorization	Required	A Base64 encoded string containing the username/password pair. The full Authorization specification is as follows. To encode and decode Base64, please read the following: http://www.securitystats.com/tools/base64.php
Content-Length	No	The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient or, in the case of the HEAD method, the size of the entity-body that would have been sent had the request been a GET. Content-Length = "Content-Length" ":" 1*DIGIT An example is Content-Length: 3495

		Applications SHOULD use this field to indicate the transfer-length of the message-body.
Content-Type	Required	The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient or, in the case of the HEAD method, the media type that would have been sent had the request been a GET. Content-Type = "Content-Type" ":" media-type The presently supported value is: Content-Type: text/xml; charset=utf-8
SOAP-Action	Required	
X-JMS-Priority	No	Defaults to 4, but can be used to increase the priority of the messaging. Increasing the priority means that the post will be handled before other messages with lower priority.

4.1.1 Content

The HTTP content consists of an XML SOAP envelope with the CFS types in the SOAP body. It is assumed that the XML will be encoded using the UTF-8 character-set.

4.1.2 Error Codes

As a result of issuing a HTTP request, one of the following return codes will be returned to the client.

For example:

```
HTTP/1.1 500 Internal Server Error
Date: Thu, 06 Jan 2005 17:32:36 GMT
Server: Jetty/3.0 (Mac OS X 10.3.7 ppc)
Servlet-Engine: Jetty/3.0 (JSP 1.1; Servlet 2.2; java 1.4.2_05)
Content-Type: text/xml; charset=utf-8
Content-Length: 561
```

If the result code is not 200, the CAD system is responsible for handling the result code and either correcting the problem if it is a client side process, retrying or entering a manual business contingency process.

Status Description	Code	Description
OK	200	Successfully posted or request was successfully processed
Bad request	400	Malformed header or properties
Unauthorized	401	Invalid authorization credentials; unknown destination; access authorization denied
Forbidden	403	License Restrictions
Message too large	413	A message cannot be enqueued because the queue has insufficient free space.
Internal server error	500	Internal processing error. Invalid SOAP envelope or error creating message
Not supported or implemented	501	Unregistered or improperly configured URL
Service unavailable	503	Flow controlled destinations

Beyond the HTTP return codes, additional data will be present as part of the SOAP envelope.

4.2 XML

A well-defined message structure will be used to define all content with the use of an XML schema. XML will not be accepted on ESB if it does not pass validation. The XML schema will be provided to all participants as will the WSDL document used to describe the web services.

4.2.1 SOAP

XML Web Service extension standards can be used for future additions, but are not used for this iteration. SOAP will be the visible XML to these services but will only be used as an envelope only and its data will not be required to be read. The CFS content and all broadcast information will also have a SOAP envelope so it is expected that the outgoing web services will write the SOAP envelope but again, there will not be any meaningful information in it.

4.2.2 WSDL

These documents will be provided, to the call centers in order to define the interface between the CAD systems and the ESB. They will be unique to each call center.

4.3 Exceptions

As a Web Service consumer the CAD vendor should anticipate handling the following exception cases:

- ESB unavailable
  - This occurs if the CAD System is unable to connect to the primary URL or any of the secondary connection URLs provided by the ESB in the WDLs.
  - Note, this should be done on a message by message basis for the purposes of load balancing. i.e. the primary URL should always be attempted first for each message being sent to the ESB.
- ESB bad URL
  - This is an error resolving the provided URL to the ESB
- Web Service response timeout
  - This is the Web Service response to the post. It is a separate case from the business case where the destination Call Center does not accept or reject a message within an acceptable time limit.
- Invalid message type
  - This would occur if the message is improperly formatted.
- Invalid destination
  - This message would be sent by the ESB to a CAD System if it placed a message on the ESB that contained a Destination Call Center that the ESB does not have a record of.
- Web Service Error operation
  - Any Errors that occur in the processing of a message will be reported back to the CAD System that originally sent the message.

Exceptions are grouped into HTTP errors and into SOAP exceptions. In general, HTTP errors, as described as part of the Web Services client and server sections are used to describe any HTTP transport problems. SOAP exceptions or faults contain any information related to processing problems. A SOAP fault can be combined with a successful HTTP POST. A failed POST may not be bound to a SOAP fault message.

SOAP faults will be used for failures and error artifacts. When a web client receives a SOAP fault, the systems must assume that an error has occurred and enter into fault processing. Fault processing is a general description of what may or may not be recoverable actions.

All SOAP faults must have a `/SOAP-ENV:Envelope/SOAP-ENV:Body/SOAP-ENV:Fault/faultcode` and `/SOAP-ENV:Envelope/SOAP-ENV:Body/SOAP-ENV:Fault/faultstring` description fields. The `faultcode` must conform to the SOAP specification for faultcodes.

4.4 SOAP Fault Codes

The following documents the understood SOAP faults:

Code	Schema	Description
Server	http://schemas.xmlsoap.org/soap/envelope	General Fault placeholder. Whenever there is a fault of this type code, assume that the coarse of action is that a change is required to the information being provided.
Server.Exception	http://schemas.xmlsoap.org/soap/envelope	A Class of server exception. But this is a general server exception. The faultstring tag should add detail.
Client	http://schemas.xmlsoap.org/soap/envelope	General Fault placeholder for client side errors. Whenever there is a fault of this type code, assume that the coarse of action is that a change is required to the information being provided.
VersionMismatch	http://schemas.xmlsoap.org/soap/envelope	

4.5 Invalid XML Post

```
HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 17:32:36 GMT
Server: Jetty/3.0 (Mac OS X 10.3.7 ppc)
Servlet-Engine: Jetty/3.0 (JSP 1.1; Servlet 2.2; java 1.4.2_05)
Content-Type: text/xml; charset=utf-8
Content-Length:

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server.Exception:</faultcode>
  <faultstring>parsing error: org.xml.sax.SAXParseException: XML document
structures must start and end within the same entity.</faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.6 Not Well-Formed XML

```
HTTP/1.1 500 Internal Server Error
Date: Thu, 06 Jan 2005 17:32:36 GMT
Server: Jetty/3.0 (Mac OS X 10.3.7 ppc)
Servlet-Engine: Jetty/3.0 (JSP 1.1; Servlet 2.2; java 1.4.2_05)
Content-Type: text/xml; charset=utf-8
Content-Length: 561

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>SOAP-ENV:Server.Exception:</faultcode>
<faultstring>parsing error: org.xml.sax.SAXParseException: The element type
&quot;SOAP-ENV:Envelope&quot; must be terminated by the matching end-tag
&quot;&lt;/SOAP-ENV:Envelope&gt;&quot;.</faultstring>
</SOAP-ENV:Fault>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.7 Timeout Server Request

```
HTTP/1.1 500 Internal Server Error
Date: Thu, 06 Jan 2005 17:39:11 GMT
Server: Jetty/3.0 (Mac OS X 10.3.7 ppc)
Servlet-Engine: Jetty/3.0 (JSP 1.1; Servlet 2.2; java 1.4.2_05)
Content-Type: text/xml; charset=utf-8
Content-Length: 419

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>SOAP-ENV:Server</faultcode>
<faultstring>Request timed out while waiting for response</faultstring>
</SOAP-ENV:Fault>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.0 MESSAGE SCHEMAS

Please refer to the following documents for a complete description of the message architecture, as well as the current message schemas:

PDCC Message Architecture.doc
PDCC_Common.xsd
PDCC_ATIS.xsd
PDCC_TMDD.xsd
PDCC_LOCAL.xsd
PDCC_CFS.xsd
PDCC_CONFIG.xsd
PDCC_SYSTEM.xsd
PDCC_IM.xsd

Please note that although the data elements have been finalized by the PDCC, the message schemas may be updated based on the results of the design phase. The schemas will be finalized on August 23rd, 2005.

6.0 TEST SCENARIOS

The testing documentation enumerates the test cases that the processes must pass before being considered ready for production.

6.1 Business Scenarios

6.1.1 Sunny Day Test Case Scenarios

1. CFS Transfer

Description
This scenario will test the functionality to allow one PSAP to send a CFS Message to another PSAP.

Test Sequence
1. Using a web service test client to connect to the ESB web service layer, a valid CFS message is placed on the bus where it is routed and successfully consumed at the destination either a web service test client or a CAD system.

or

1. A CAD system will connect to the ESB and place a valid CFS message on the ESB where it is routed and successfully consumed at the destination by either a web service test client or a CAD system.
2. Perform Step 1 for each valid from/to destination pair (i.e. the initial 7 PSAPs form 49 valid from/to destination pairs)

Inputs
- CFS Message

Expected outcomes
- The destination the message has been sent to, successfully consumes the message off of the ESB.
- The audit trail for this business process is correctly persisted in the audit database

2. CFS Acknowledgement (Accept)

Description
This scenario will test the functionality for a PSAP to acknowledge a message that it has received.

Prerequisites
- CFS Message has been sent to the PSAP that is going to Acknowledge the message

Test Sequence
1. Using a web service test client to connect to the ESB web service layer, a valid Acknowledgement (Accept) message is placed on the bus where it is routed and successfully consumed at the destination by either a web service test client or a CAD system representing the system that originally sent the CFS.

or

1. A CAD system will connect to the ESB and place a valid Acknowledgement (Accept) message on the ESB where it is routed and successfully consumed at the destination by either a web

service test client or a CAD system representing the system that originally sent the CFS

2. Perform Step 1 for each valid from/to destination pair (i.e. the initial 7 PSAPs form 49 valid from/to destination pairs)

Inputs

- Acknowledgement (Accept Message) with valid Incident Cross Reference Data.

Expected outcomes

- The client that originally sent the CFS Transfer, successfully consumes the acknowledgement message off of the ESB
- The audit trail for this business process is correctly persisted in the audit database

3. CFS Acknowledgement (Reject)

Description

This scenario will test the functionality for a PSAP to acknowledge a message that it has received.

Prerequisites

- CFS Message has been sent to the PSAP that is going to Acknowledge the message

Test Sequence

1. Using a web service test client to connect to the ESB web service layer, a valid Acknowledgement (Reject) message is placed on the bus where it is routed and successfully consumed at the destination by either a web service test client or a CAD system representing the system that originally sent the CFS.

or

1. A CAD system will connect to the ESB and place a valid Acknowledgement (Reject) message on the ESB where it is routed and successfully consumed at the destination by either a web service test client or a CAD system representing the system that originally sent the CFS
2. Perform Step 1 for each valid from/to destination pair (i.e. the initial 7 PSAPs form 49 valid from/to destination pairs)

Inputs

- Acknowledgement (Reject Message) with valid Incident Cross Reference Data.

Expected outcomes

- The client that originally sent the CFS Transfer, successfully consumes the acknowledgement message off of the ESB

- The audit trail for this business process is correctly persisted in the audit database

4. CFS Update

Description

This scenario will test the functionality for a PSAP to send an update to a previously sent CFS.

Prerequisites

- CFS Message has been sent to the PSAP that is going to send an update back to the originating PSAP.

Test Sequence

1. Using a web service test client to connect to the ESB web service layer, a valid CFS Update message is placed on the bus where it is routed and successfully consumed at the destination by either a web service test client or a CAD system representing the system that originally sent the CFS.

or

1. A CAD system will connect to the ESB and place a valid CFS Update message on the ESB where it is routed and successfully consumed at the destination by either a web service test client or a CAD system representing the system that originally sent the CFS
2. Perform Step 1 for each valid from/to destination pair (i.e. the initial 7 PSAPs form 49 valid from/to destination pairs)

Inputs

- CFS Update Message with valid Incident Cross Reference Data that references the original Incident ID as well as the ID generated by the PSAP which received the CFS Transfer.

Expected outcomes

- The client that originally sent the CFS Transfer, successfully consumes the update off of the ESB
- The audit trail for this business process is correctly persisted in the audit database

5. Information Only Message

Description

This scenario will test the functionality for a PSAP to send an information only message to another PSAP. This message differ from others in that they do not require an acknowledgement

Prerequisites

- None.

Test Sequence

1. Using a web service test client to connect to the ESB web service layer, a valid Information Only message is placed on the bus where it is routed and successfully consumed at the destination by either a web service test client or a CAD system.

or

1. A CAD system will connect to the ESB and place a valid Information Only message on the ESB where it is routed and successfully consumed at the destination by either a web service test client or a CAD system.
2. Perform Step 1 for each valid from/to destination pair (i.e. the initial 7 PSAPs form 49 valid from/to destination pairs)

Inputs

- Information Only Message.

Expected outcomes

- The client representing to intended destination, successfully consumes the update off of the ESB

6. Broadcast Message

Description

In this scenario the broadcast functionality will be tested using the Information Only Message Type. It will test the functionality that allows a PSAP to send a single message to multiple recipients.

Prerequisites

- None.

Test Sequence

1. Using a web service test client to connect to the ESB web service layer, a valid Information Only message is placed on the ESB with multiple destinations specified. It will then be routed and successfully consumed at the destinations by either a web service test client or a CAD system.

or

1. A CAD system will connect to the ESB and place a valid Information Only message on the ESB with multiple destinations specified. It will then be routed and successfully consumed at the destinations by either a web service test client or a CAD system.
2. Perform Step 1 for each valid "from" destination to all other valid destinations.

Inputs

- Information Only Message.

Expected outcomes

- The clients representing to intended destinations, successfully consume the message off of the ESB

7. Distribute Updated Schemas

Description

In this scenario the broadcast the ability for the ESB to distribute updated XML schemas is tested.

Prerequisites

- None.

Test Sequence

1. Using a web service test client to connect to the ESB web service layer, a valid Schema Update message is placed on the ESB with all valid destinations specified as recipients. It will then be routed and successfully consumed at the destinations by either a web service test client or a CAD system.

Inputs

- Schema Update Message.

Expected outcomes

- The clients representing to intended destinations, successfully consume the message off of the ESB

6.1.2 Error Test Case Scenarios

1. CAD is unable to connect to the ESB

Description

If the CAD systems attempts to post a message to the ESB and does not receive a satisfactory response from the ESB service, the CAD system should identify this as an error condition.

This test case scenario is specific to the CAD system and is not in scope for the ESB implementation.

2. Message Contains an invalid destination

Description

This test scenario will measure the ESB's response to an invalid routing instruction.

Prerequisites
- None.

Test Sequence
1. Using either a test client or a CAD System to connect to the ESB web service layer, a valid message with at least one invalid destination specified as recipients is placed on the bus. It will then be attempted to be routed by the ESB.
2. Repeat this test until all valid CAD Systems on the ESB have received an error message

Inputs
- Any valid Message.

Expected outcomes
The ESB should successfully identify the invalid route and generate an error message that indicates this condition to be sent back to the originating CAD System.

3. Message format is invalid

Description
This test scenario will measure the ESB's response to an invalid message format.

Prerequisites
- None.

Test Sequence
1. Using either a test client or a CAD System to connect to the ESB web service layer, an invalid message is placed on the bus.
2. Repeat this test until all valid CAD Systems on the ESB have sent an invalid message.

Inputs
- Any Message Type that has at least one error in it.

Expected outcomes
The ESB should successfully identify the invalid format and send the response back to the originating CAD system.

4. ESB is unable to deliver message

Description

This scenario will test the ESB's ability to detect and respond to a message delivery failure.

Prerequisites
- None.

Test Sequence
1. Using either a test client or a CAD System to connect to the ESB web service layer, a valid message is placed on the bus and is routed to a destination that has been disabled.
2. Perform this test for all CAD systems.

Inputs
- Any valid Message Type.

Expected outcomes
The ESB should exceed its maximum delivery attempts at which point it should generate an error message that identifies the problem, and forwards it to the originating CAD system.

5. ESB Heartbeat fails

Description
This scenario will test the availability of all of the connected CAD Systems.

Prerequisites
- None.

Test Sequence
1. Using either test clients or CAD systems, simulate the availability of some but not all of the valid destinations on the ESB.
2. Start the heartbeat process.

Inputs
- None.

Expected outcomes
The ESB should identify the destinations that it was unable to connect to and send out email notifications describing this issue.

6. CAD does not receive message acknowledgement

Description

This tests the condition that occurs when a pre-determined period of time is exceeded while the CAD system is expecting an accept/reject return message for a previously sent message.

This test case scenario is specific to the CAD system and is not in scope for the ESB implementation.

7. CAD receives error message from ESB

Description
This tests the condition that occurs when the ESB encounters an error and reports it back to the originating CAD system.

This test case scenario is specific to the CAD system and is not in scope for the ESB implementation.

8. CAD is unable to connect to the primary Host

Description
This tests the condition that occurs if 1 or more of the URL's provided by the PDCC to connect to the ESB are unavailable.

Expected outcomes
The CAD System should be capable of re-trying the message send using the alternate URLs provided by the PDCC.

This test case scenario is specific to the CAD system and is not in scope for the ESB implementation.

9. CAD is unable to connect to any ESB Service

Description
This describes the condition where a CAD system has attempted to connect to all provided URLs and is unable to connect to the ESB through any of them.

This test case scenario is specific to the CAD system and is not in scope for the ESB implementation.

6.2 Technical Scenarios

The technical scenarios will be designed to measure the performance of all of the technical components of the ESB. The results of the test will be used by the team to tune the architecture in order to meet all expected performance levels.

6.2.1 Performance Testing

Performance testing will put stress on the architecture to measure the performance metrics as the load on the server increases. Any bottlenecks to performance will be identified and resolved as part of these scenarios.

1. 1x Expected Load
2. 2x Expected Load
3. 5x Expected Load
4. 10x Expected Load

6.2.2 Stability Testing

These test scenarios are designed to measure the stability of the system over long periods of time. As volumes of messages are sent through the ESB, it will be monitored for memory and performance issues as well as any other possible impact to the overall stability of the architecture

1. Peak Load Short Term
2. Peak Load Long Term
3. 1x Expected Load Long Term

6.2.3 High Availability Testing

High availability testing will ensure that the redundancy designed into the ESB functions as expected and provides the expected levels of service.

1. Single Broker Failure
2. Multiple Broker Failure
3. Cluster Failure
4. Broker Server Failure
5. O-Server Host Failure
6. XML Server Host Failure

7. **Server Re-start**

8. **Service Re-start**

9. **Communication Line Failure**

10. **Network Failure**

6.2.4 Security Testing

These tests should be carried out by a third party due to conflict of interest concerns. The testing should span the entire ESB architecture from in-flight messages to ESB host operating systems.