ON-CALL
INTELLIGENT TRANSPORTATION SERVICES FOR

RADS TRAFFIC MANAGEMENT SYSTEM CENTER-TO-CENTER MESSAGE INTERFACE DESIGN

REVISION 1.2
OCTOBER 30, 2007

REVISION HISTORY

Revision	Date	Changes
1.0	19-Mar-2007	Initial release.
1.1	20-Sep-2007	Updated to reflect simplified messaging schema.
1.2	30-Oct-2007	Updated XSD by removing references to “Subscriber” paradigm. Improved status feedback provided to publisher. Added test client instructions.

Prepared by:

OZ Engineering | Systems Integrators Software Consultants

3230 East Broadway Road
Suite 216
Phoenix, AZ 85040
(602) 234-5930

GLOSSARY

Acronym	Definition
ADOT	Arizona Department of Transportation
C2C	Center-to-Center
CAD	Computer-Aided Dispatch
EMS	Emergency Management System
GUI	Graphical User Interface
HCRS	Highway Condition Reporting System
IEEE	Institute of Electrical and Electronics Engineers
IM	Incident Management
ITIS	International Traveler Information Interchange Standard provides a recognized list of codes and descriptions for road conditions.
MCDOT	Maricopa County Department of Transportation
NTCIP	National Transportation Communications for Intelligent Transportation Systems Protocol
RADS	Regional Archived ITS Data System
RCRS	Road Closure and Restriction System
RMI	Remote Method Invocation
WSDL	Web Service Definition Language
XML	Extensible Markup Language
XSD	XML Schema Definition

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Purpose

This document provides the design details for the Traffic Management System (TMS) Center-to-Center (C2C) interface module using the Regional Archived Data Server (RADS) as the central archive. Contained within are the system architecture, system requirements, interface control specifications and system test plans.

1.2. Introduction

Transportation planners, traffic engineers, operating agencies, and the general public all need comprehensive access to valid, current traffic data collected from real-time Intelligent Transportation System (ITS) sources. The AZTech™ Transportation and Public Safety Center-to-Center (C2C) Needs Assessment and Concept of Operations project has identified the desired functionality of center-to-center systems for transportation and public safety agencies.

The C2C System will support inter-jurisdictional exchange of traffic management system information using the Traffic Management System (TMS) C2C Protocol and the Regional Archived Data System (RADS) as the exchange mechanism. The first messages to be supported in this protocol are for the monitoring of traffic signal system status, inventory and timing plans.

1.3. Overview

The AZTech™ Center-to-Center Interface Specification has been developed through the consensus input of regional stakeholders. Following a systems engineering methodology, a user needs assessment and concept of operations (ConOps) were developed. Based on the ConOps system functional requirements and a regional stakeholder agreement were then developed. The resulting specification is available at http://www.consystec.com/c2cxml/rtm/rtm.htm.

In order to improve the possibility for successful deployment, the initial phase of this project will implement a simplified System Architecture. As shown in the following figure, various centers will be able to send TMS C2C information to the RADS. There will be no exchange of control commands, only inventory and status information will be shared. The RADS will archive messages received from each center, and enforce rules regarding the visibility of data available to

authorized clients. An administrative interface will permit center administrators to indicate which clients are authorized to see their information.

1.4. System Architecture

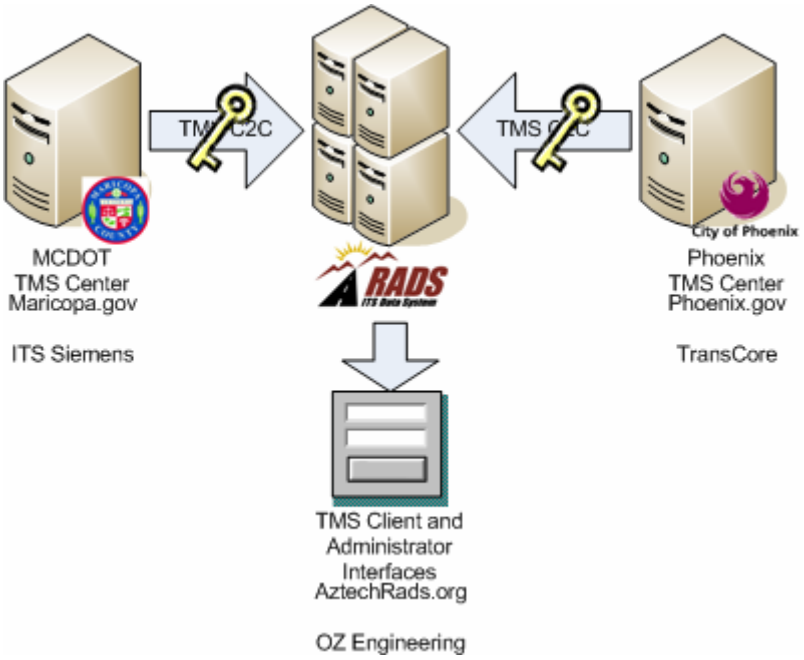

Figure 1 - System Overview

System	Description
MCDOT and Phoenix TMS Center(s)	Each Center will send TMS timing plan, inventory and status updates to the RADS server using a defined XML message and web service.
RADS	The MCDOT Regional Archived Data Server will receive information from the TMS Centers and archive it. This information will be available to other authorized systems that may request it through available web services.
TMS Client and Administrator Interfaces	Authorized Clients may access TMS information from the RADS. An administrative interface will permit center administrators to indicate which clients are authorized to see their information.

2. WEB SERVICE INTERFACE

2.1. Overview

This chapter describes the interface to be provided by the RADS TMS web service. Refer to the Use Case section for a detailed description of actions performed when each method is invoked.

Actual web service details are contained within: AZTech-Local-01-00-11.wsdl.

2.2. Web Service Dialogs

The following figure illustrates the supported web service request-response dialogs as defined in the WSDL.

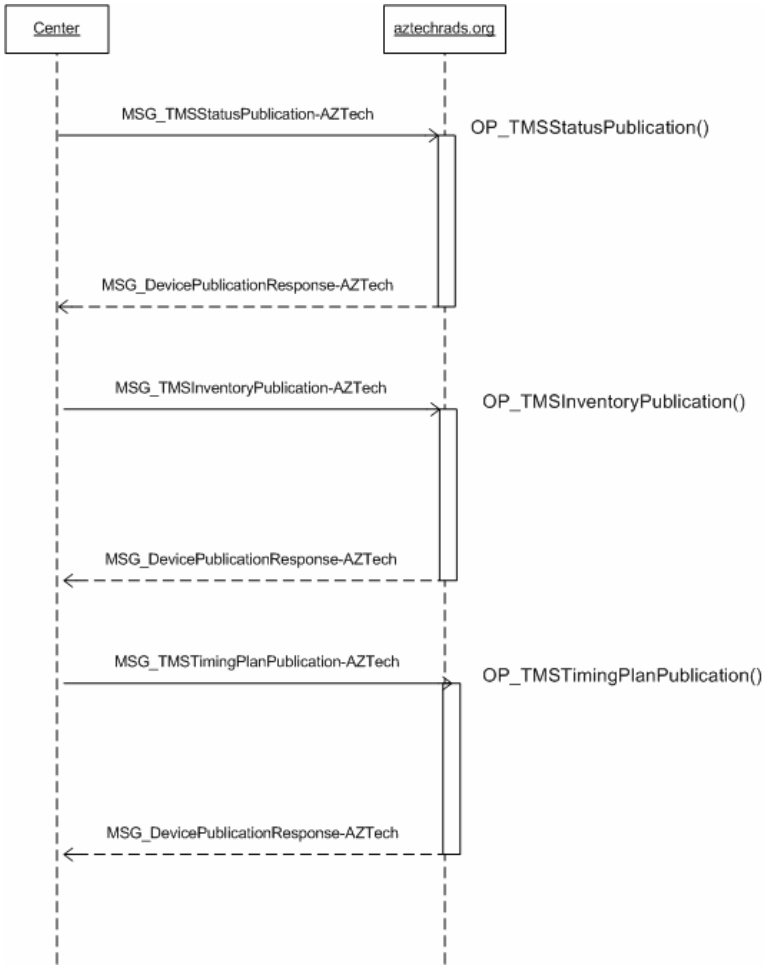

Figure 2 - Web Service Dialogs

2.3. Web Service Methods

This section describes the TMS C2C Web Service methods that are used by centers to send data into the RADS.

2.3.1. OP_TMSStatusPublication

Table 1 - OP_TMSStatusPublication Method

Web Service Method Name:	OP_TMSStatusPublication
Description:	A TMS Center will invoke this method to publish the status of its intersection detectors as necessary. Status information may be provided periodically or as available. All detectors for a single organization may be included within a single message or published across multiple messages. It is expected that this reporting rate will match the rate indicated within the DetectorDetailInventory.
Prototype:	MSG_DevicePublicationResponse-AZTech OP_TMSStatusPublication(MSG_TMSStatusPublication-AZTech)
Parameters:	**MSG_TMSStatusPublication-AZTech** Message containing the real-time status of one or more intersections.
Return Value:	**MSG_DevicePublicationResponse-AZTech** Response indicating how the request was processed.

2.3.2. OP_TMSInventoryPublication

Table 2 - OP_TMSInventoryPublication Method

Web Service Method Name:	OP_TMSInventoryPublication
Description:	A TMS Center will invoke this method to publish the intersection inventory details whenever they change. It is suggested that inventory information be provided for all devices at TMC Center startup, and subsequently for each device as its inventory information changes. Inventory details for an organization's devices may be provided within a single message or published across multiple messages.
Prototype:	MSG_DevicePublicationResponse-AZTech OP_TMSInventoryPublication(MSG_TMSInventoryPublication-AZTech)
Parameters:	**MSG_TMSInventoryPublication-AZTech** Message containing the inventory details for one or more of the intersections within a center.
Return Value:	**MSG_DevicePublicationResponse-AZTech** Response indicating how the request was processed.

2.3.3. OP_TMSTimingPlanPublication

Table 3 - OP_TMSTimingPlanPublication Method

Web Service Method Name:	OP_TMSTimingPlanPublication
Description:	A TMS Center will invoke this method to publish timing plan details for a single intersection as necessary. Timing plans for numerous intersections may be updated using the inventory update method: OP_TMSInventoryPublication.
Prototype:	MSG_DevicePublicationResponse-AZTech OP_TMSTimingPlanPublication(MSG_TMSTimingPlanPublication-AZTech)
Parameters:	**MSG_TMSTimingPlanPublication-AZTech** Message containing the updated timing plan details for one or more timing plans within a single intersection.
Return Value:	**MSG_DevicePublicationResponse-AZTech** Response indicating how the request was processed.

2.4. XML Schema Diagrams

This section contains the graphical renditions of the various TMS C2C message components. The first structures listed correspond to the messages exchanged within the web service and appear in the following order:

- TMSInventoryPublication-AZTech
- TMSStatusPublication-AZTech
- TMSTimingPlanPublication-AZTech
- DevicePublicationResponse-AZTech

Those structures that are referenced by the preceding structures follow in alphabetical order.

Actual schema details are contained in the XSD file: AZTech-Local-01-00-11.xsd.

2.4.1. TMSInventoryPublication-AZTech

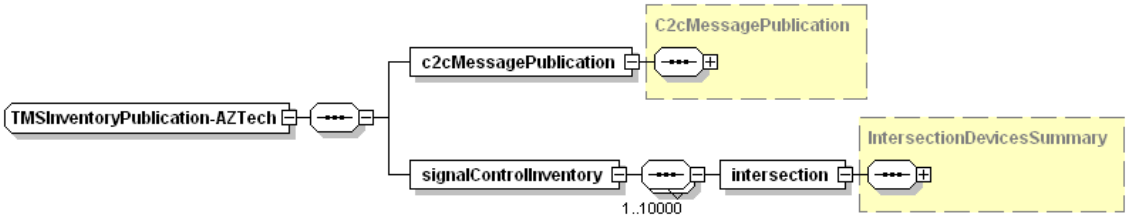

Figure 3 – TMSInventoryPublication-AZTech

2.4.2. TMSStatusPublication-AZTech

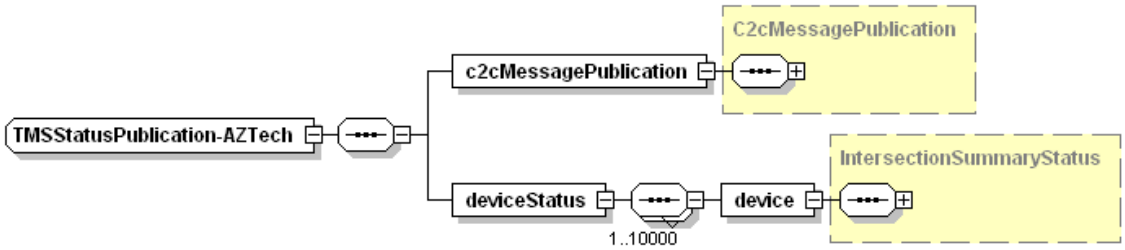

Figure 4 – TMSStatusPublication-AZTech

2.4.3. TMSTimingPlanPublication-AZTech

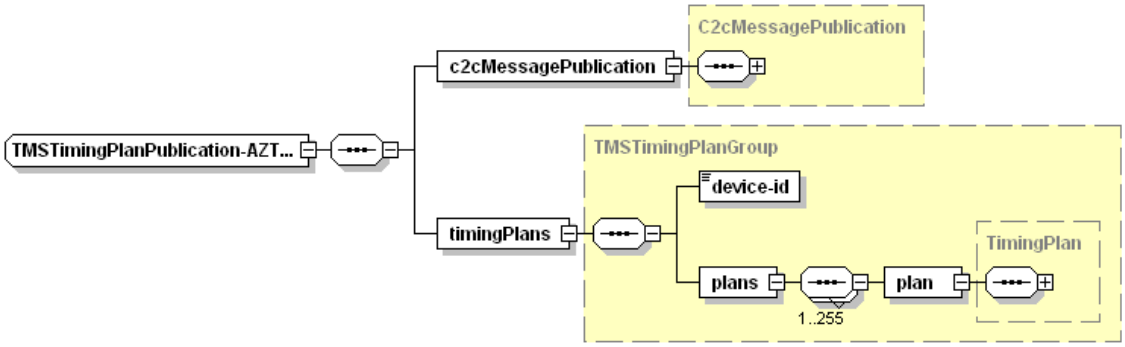

Figure 5 – TMSTimingPlanPublication-AZTech

2.4.4. DevicePublicationResponse-AZTech

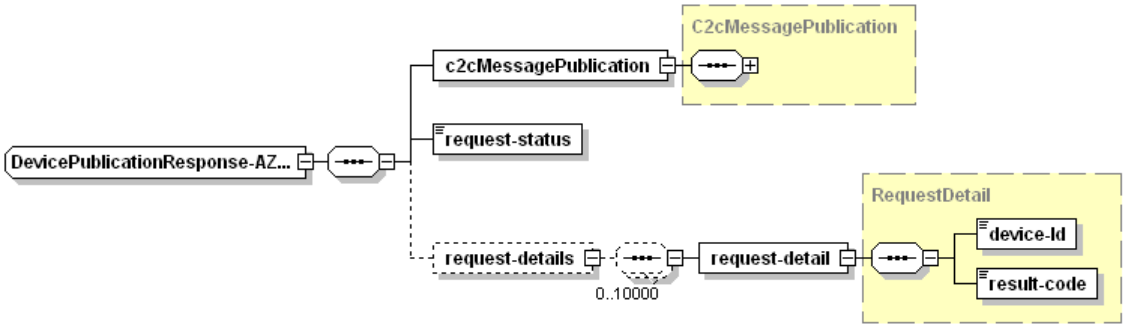

Figure 6 – DevicePublicationResponse-AZTech

2.4.5. C2cMessagePublication

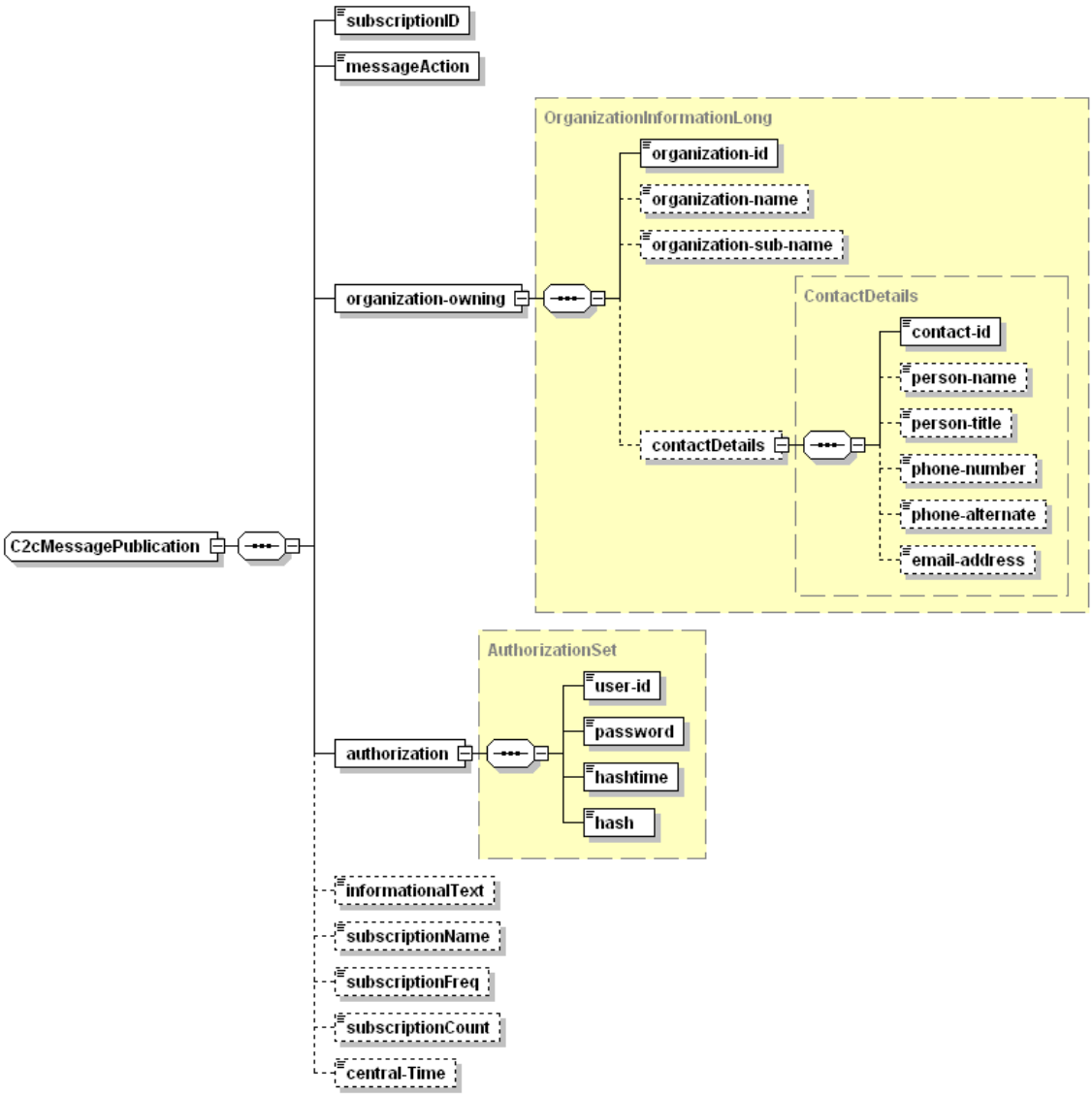

Figure 7 – C2cMessagePublication

2.4.6. DetectorDetailInventory

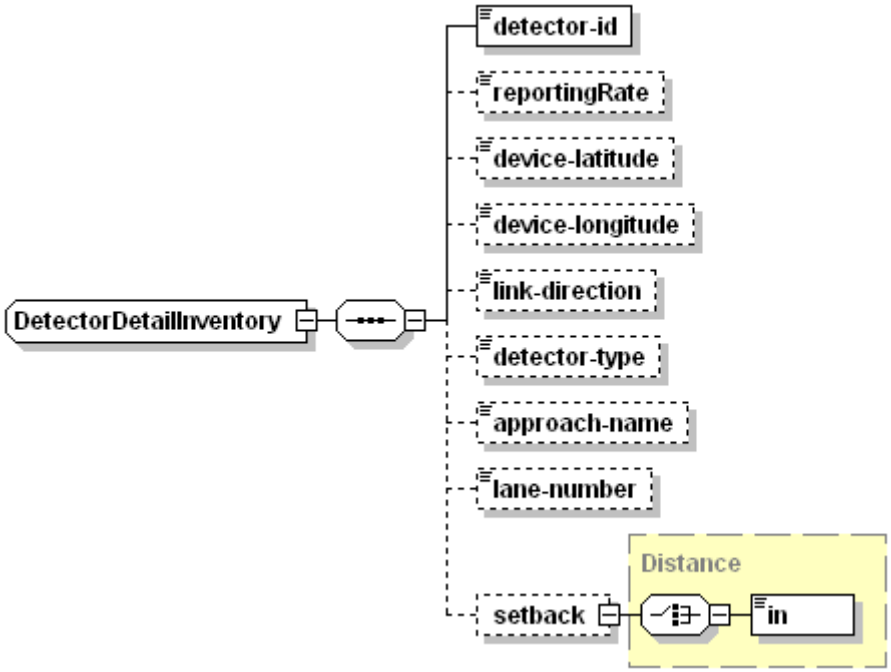

Figure 8 – DetectorDetailInventory

2.4.7. IntersectionDevicesSummary

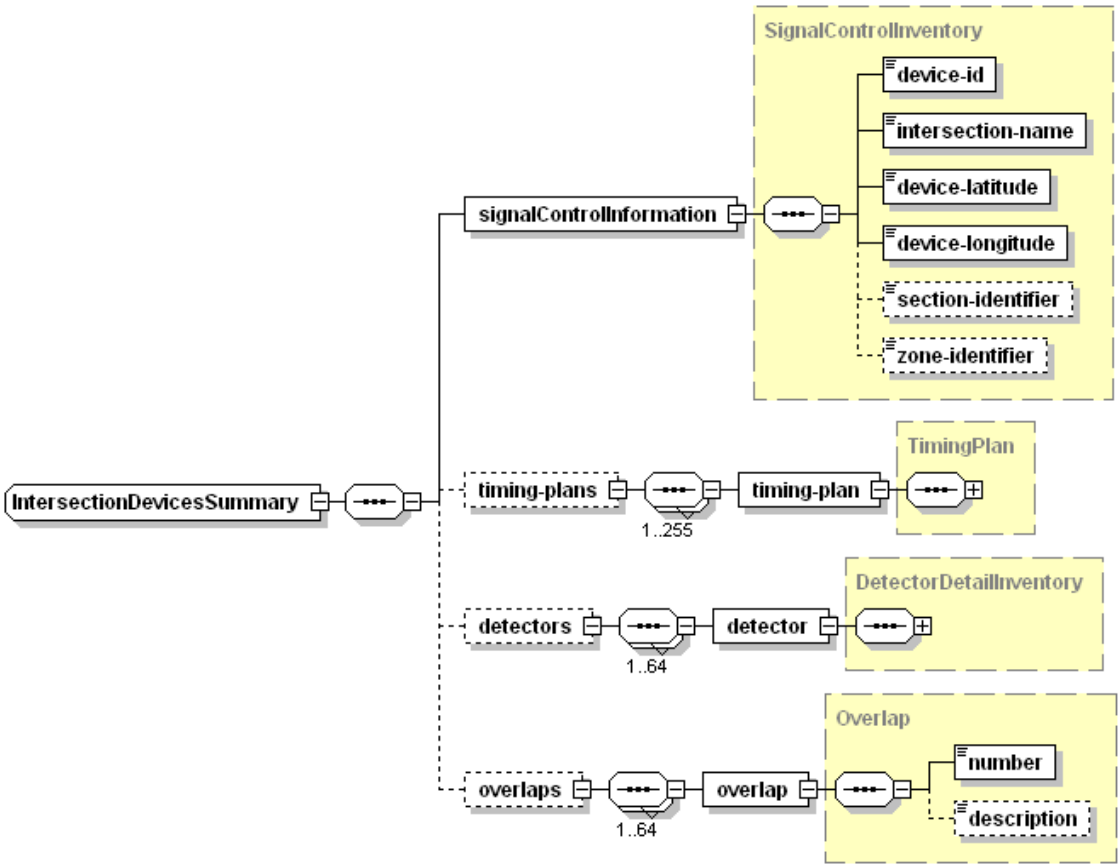

Figure 9 – IntersectionDevicesSummary

2.4.8. IntersectionSummaryStatus

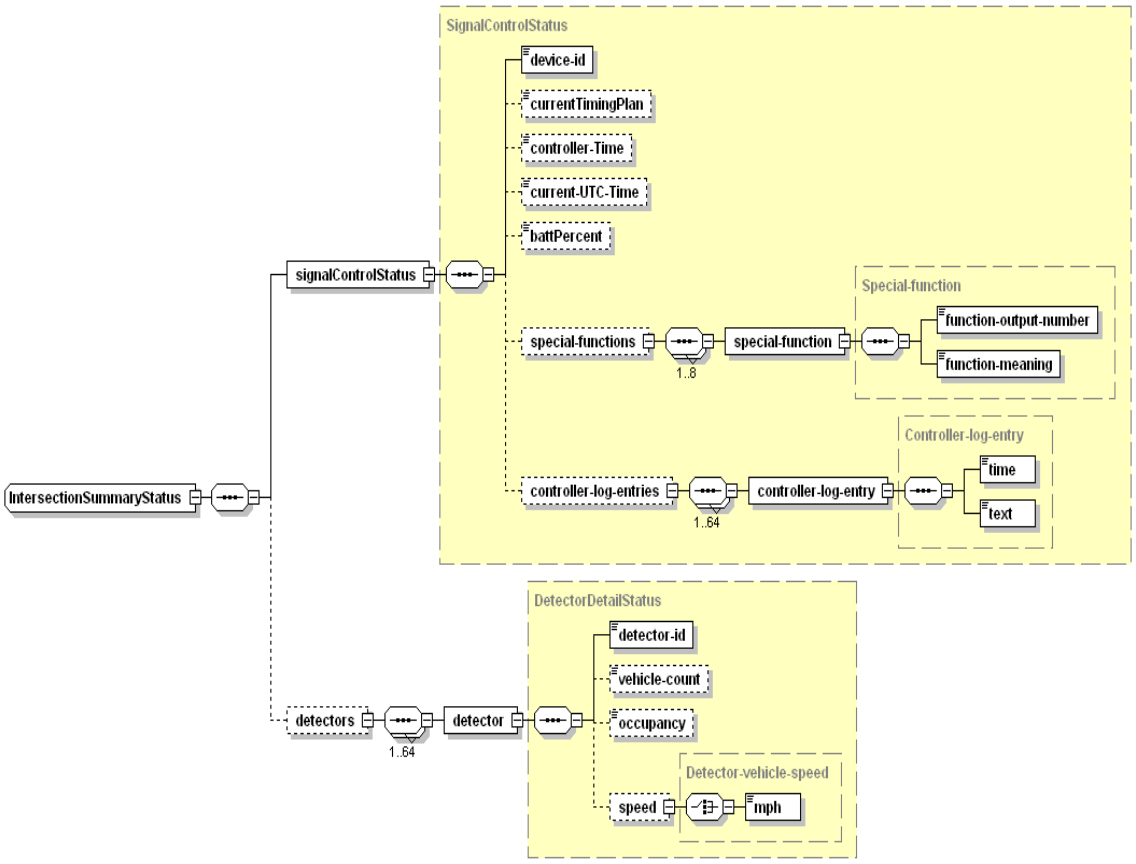

Figure 10 –IntersectionSummaryStatus

2.4.9. TimingPlan

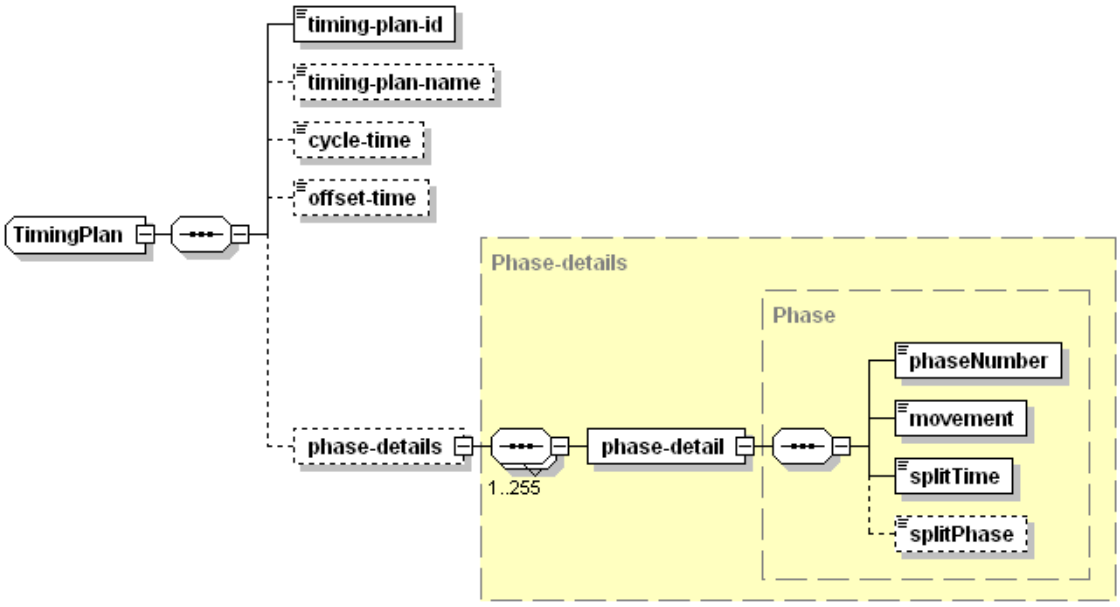

Figure 11 –TimingPlan

2.5. Data Element Definitions

This section defines the various data elements used; appearing in alphabetical order. Refer to the XSD definition for specific enumerations, value ranges and other limitations.

Table 4 - Data Element Definitions

Data Element	Definition
approach-name	String name or description of the intersection approach being monitored by the detector (e.g., NB Left Turn).
battPercent	Battery backup status.
central-Time	Current date and time at central.
contact-id	A unique identifier for a contact person as assigned by the organization.
controller-Time	Current time at controller.
currentTimingPlan	Id number of the traffic signal timing plan currently in use.
current-UTC-Time	Current Coordinated Universal Time (UTC).
cycle-time	Specifies the length of the pattern cycle in seconds.
detector-id	Unique identification number of an individual detector within a network.
detector-type	Code naming the type of a vehicular detector providing traffic data.
device-id	A unique alphanumeric device identifier.
device-latitude	The geographic latitude of a node, expressed in integer microdegrees, with reference to the horizontal datum specified by horizontalDatum. Our implementation will expect a projection of nad83.
device-longitude	The geographic longitude of a node, expressed in integer microdegrees, with reference to the horizontal datum specified by horizontalDatum. Our implementation will expect a projection of nad83.
email-address	The e-mail address of a person at an organization.
freeText	A simple free-text field used in the message when structured information will not serve.
function-meaning	A text description of the corresponding function-output-number.
function-output-number	Assigned value of any special function bits within a controller.
hash	The Security hash determined by the x.509 cert in base-64 form. Implementation details to be determined at a future time; set value to zero (0).
hashtime	A representation of date, time, and time zone in the standard XML formats. Should be set to current time when a message is sent. Implementation details to be

Data Element	Definition
	determined at a future time; set value to zero (0).
informationalText	Free text used to provide information.
intersection-name	The user-defined name of the intersection.
lane-number	The number which indicates the lane a particular detector is monitoring. Select one bit per lane. Lanes are numbered from the median out beginning with 1.
link-direction	The direction(s) of travel referenced on a link.
movement	Free text describing the given phase movement.
occupancy	The current average percent occupancy of the vehicles on a detector. This is percent of time within a given time period (reportingRate) that a point on the roadway is occupied by traffic.
offset-time	Defines by how many seconds the local time zero shall lag the system time zero (synchronization pulse) for this pattern.
organization-id	A unique identifier for an organization within a region. Typically, the domain name for that organization will be used: e.g., azdot.gov or maricopa.gov.
organization-name	The organization's name.
organization-sub-name	The organization's "sub organization" – Department or Bureau, for example – name.
password	The unique password of a user to login.
person-name	The name or identifier of a person at an organization.
person-title	The title of a person at an organization.
phaseNumber	The phase number in question.
phone-alternate	An alternate phone number (e.g., home phone number) of a person at an organization.
phone-number	The telephone number of a person at an organization.
reportingRate	A value for how often (in various units from years to seconds) an event should occur, such as updating a data file. Expressed in the native XML format when used. In our implementation the reportingRate shall be in units of seconds.
request-status	Acknowledgement of request command from one TMC to another for device action or message processing. • "requested changes completed" when message processed correctly. • "requested changes completed with errors" when only some of the devices requested could be processed correctly. The feedback message will indicate which device requests failed and the reason. • "request rejected invalid command parameters" when invalid XML format. • "request rejected insufficient privileges of the requesting operator" when authorization parameters

Data Element	Definition
	fail.
section-identifier	Unique identifier for the section using a common timing plan.
setback	Setback from stop bar.
speed	The average vehicle speed over a detector during accumulation period.
splitPhase	The phase number in question.
splitTime	The time in seconds the splitPhase is allowed to receive (i.e. before a Force Off is applied) when constant demands exist on all phases. In floating coordForceMode, this is always the maximum time a non-coordinated phase is allowed to receive. In fixed coordForceMode, the actual allowed time may be longer if a previous phase gapped out. The splitTime includes all phase clearance times for the associated phase. The split time shall be longer than the sum of the phase minimum service requirements for the phase. When the time is NOT adequate to service the minimum service requirements of the phase, Free Mode shall be the result. The minimum requirements of a phase with a not-actuated ped include Minimum Green, Walk, Pedestrian Clear, Yellow Clearance, and Red Clearance; the minimum requirements of a phase with an actuated pedestrian include Minimum Green, Yellow Clearance, and Red Clearance. If the cycleTime entry of the associated patternTable entry is zero (i.e. the device is in Free Mode), then the value of this object shall be applied, if non-zero, as a maximum time for the associated phase. If the critical path through the phase diagram is less than the cycleTime entry of the associated patternTable entry, all extra time is alloted to the coordination phase in each ring. If the critical path through the phase diagram is greater than the cycleTime entry of the associated patternTable entry (and the cycleTime is not zero) the device shall operate in the Free Mode. While the Free Mode condition exists, the Local Override bit of shortAlarm shall be set to one (1). Taken from NTCIP.
subscriptionCount	The nth time the publisher has sent content as part of a description to the subscriber.
subscriptionID	A unique name created by the subscriber for a subscription. This token will be returned in the message response and may be used to match requests with their corresponding responses.
subscriptionFreq	The subscription interval (period) in seconds.
subscriptionName	A textual name for the subscription for human readers.
text	Controller log entry text.

Data Element	Definition
time	Date and time when the controller log entry text was made.
timing-plan-id	Id number of the traffic signal timing plan used.
timing-plan-name	The user-defined name or short description of timing plan.
user-id	The unique User name used to login to the system. Typically, the user's email address will be used.
vehicle-count	The number of detected vehicles (typically from loop sensor hits) over a known location over a known period of time (reportingRate).
zone-identifier	Unique identifier for the zone using a common timing plan.

3. Interface Client Test

3.1. Overview

This chapter describes the steps to be followed in order to confirm the proper operation of the web service interface with a test client. The standard testing module (provided) will be used to confirm that the client computer communicates properly with the RADS web service using security certificates, even from behind a firewall.

3.2. Java-based Test Client

The following table lists the Java-based software components that comprise the test client application.

Table 5 - Java-based Test Client Components

File Name	Description
TestRMSC2Cws.bat *	Batch file that invokes the client application.
TestAztechTMSC2CServices.java *	Java source code for the test client.
tmsC2CTestClient.jar *	Java archive (zip) file containing necessary utilities and support routines.
radssrv.keystore *	A protected database that holds keys and certificates.
axis2-1.3 (folder) **	Open-source web service stack provided by apache.org implementing the Simple Object Access Protocol (SOAP).
* Contained within file TestTMSC2C.zip. ** Contained within file axis2-1.3-bin.zip which may be downloaded from http://ws.apache.org/axis2/.	

3.2.1. Installing the Test Client

Extract the contents of TestTMSC2C.zip and axis2-1.3-bin.zip into a directory, for example C:\c2cTest. The contents of the directory will be as follows:

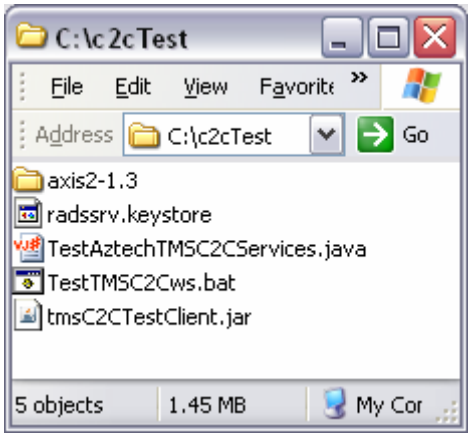

Figure 12 - Contents of Test Client Folder

3.2.2. Executing the Test Client

The batch file, TestTMSC2Cws.bat, will invoke the test program. This program will in turn perform the following operations:

- Use the computer's %username% environment variable to fill in the <person-name> element.
- Set the <hashtime> element with the computer's current date and time.
- Connect to the RADS web service and send a TMS Inventory Publication message.
- Receive the Device Publication Response message and display it in the command tool screen.

Open a command tool window and invoke the TestTMSC2Cws.bat file. Sample results are shown below. Confirm the <person-name> and <hashtime> elements.

Table 6 - Sample Test Client Output

```
Inventory response =
<xml-fragment xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<c2cMessagePublication><subscriptionID>Mon Oct 29 17:22:50 MST 2007</subscriptionID>
   <messageAction>delete all</messageAction>
   <organization-owning><organization-id>any org id</organization-id>
      <organization-name>any org name</organization-name>
      <organization-sub-name>any org sub name</organization-sub-name><contactDetails>
      <contact-id>contact id</contact-id>
      <person-name>tguerra</person-name>          %username%
      <person-title>any title</person-title>
      <phone-number>any phone</phone-number>
      <phone-alternate>any alt phone</phone-alternate>
      <email-address>any email address</email-address></contactDetails>
   </organization-owning>
   <authorization><user-id>Any User</user-id>
      <password>Any Password</password>
      <hashtime>2007-10-29T17:22:50.824-07:00</hashtime>   Current date/time
      <hash>any hash len 10</hash>
   </authorization>
   <informationalText>Some Information Text</informationalText>
</c2cMessagePublication>
<request-status>requested changes completed</request-status>
</xml-fragment>
```

3.2.3. Confirming Web Service Results

Immediately after invoking the test client batch file, access the RADS web service monitor page: http://aztechrads.org/monit.jsp. Press the View button.

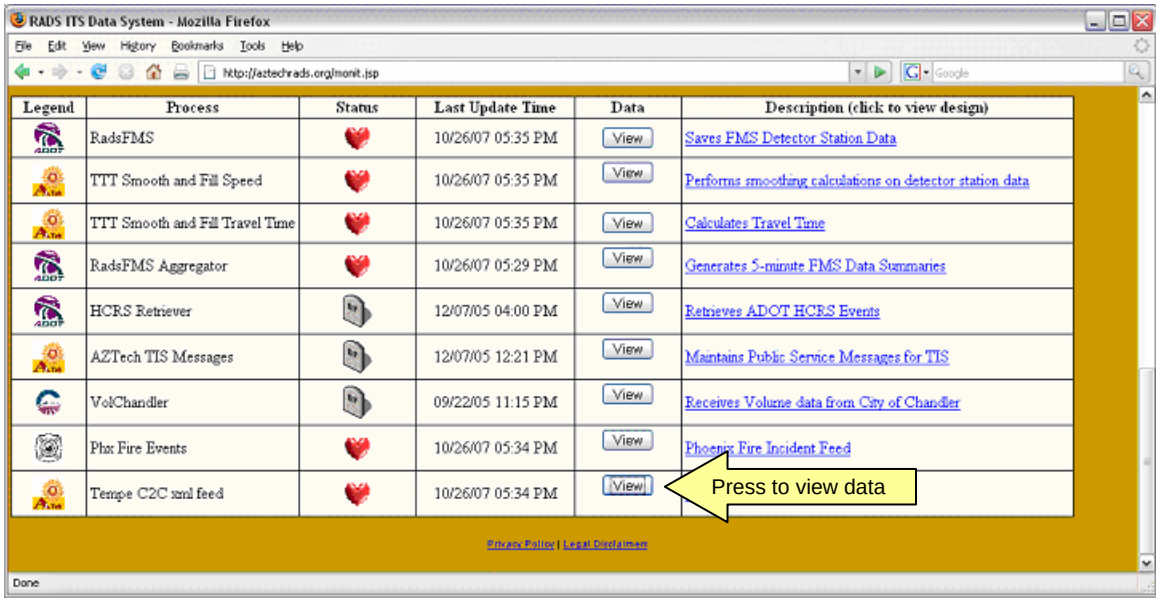

Figure 13 - RADS Web Service Monitor

Confirm that the <person-name> and <hashtime> elements match those sent by the test client.

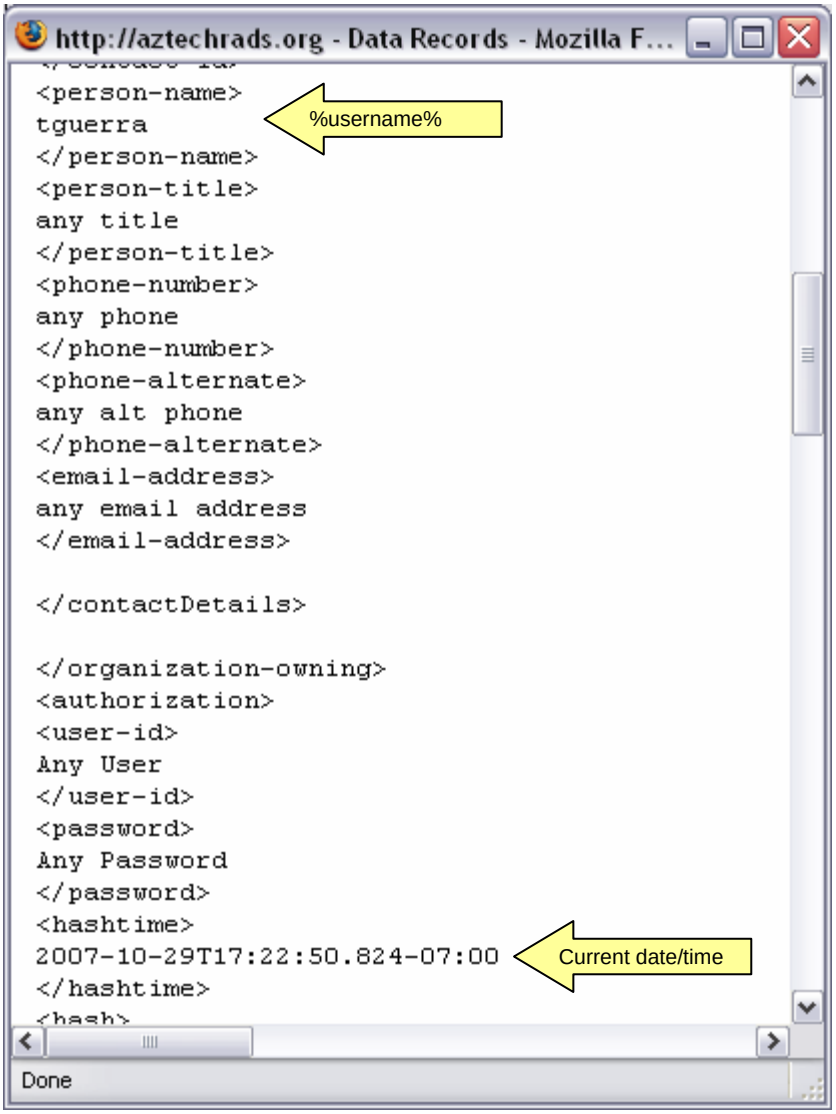

Figure 14 - Captured Test Client Data