Open Geospatial Consortium

Publication Date: 2013-06-18

Approved Date: 2013-01-18

Posted Date: 2012-07-20

Reference number of this document: OGC 12-158

Reference URL for this document: http://www.opengis.net/def/doc-type/per/ows9-aviation-performance

Category: Public Engineering Report

Editor: Matthes Rieke

OGC® OWS-9 Report on Aviation Performance Study

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

Abstract

This document is a deliverable of the OGC Web Services (OWS) Initiative - Phase 9 (OWS-9). The report summarizes the work carried out regarding performance and endurance testing of data provision services, namely Web Feature Service and Event Service. More specifically, the report deals with the performance and endurance testing of data provision services commonly used within OWS Aviation testbeds. Test runs have been evaluated on the basis of well-defined, service-specific test models and the results are documented in detail. Furthermore, a description of the service test environment is documented in alignment with the overall OWS-9 service architecture

Keywords

Ogcdoc, ogc document, ows9, ows 9, performance, aviation, wfs, filter, fe

What is OGC Web Services 9 (OWS-9)?

OWS-9 builds on the outcomes of prior OGC interoperability initiatives and is organized around the following threads:

- **Aviation**: Develop and demonstrate the use of the Aeronautical Information Exchange Model (AIXM) and the Weather Exchange Model (WXXM) in an OGC Web Services environment, focusing on support for several Single European Sky ATM Research (SESAR) project requirements as well as FAA (US Federal Aviation Administration) Aeronautical Information Management (AIM) and Aircraft Access to SWIM (System Wide Information Management) (AAtS) requirements.

- **Cross-Community Interoperability (CCI)**: Build on the CCI work accomplished in OWS–8 by increasing interoperability within communities sharing geospatial data, focusing on semantic mediation, query results delivery, data provenance and quality and Single Point of Entry Global Gazetteer.

- **Security and Services Interoperability (SSI)**: Investigate 5 main activities: Security Management, OGC Geography Markup Language (GML) Encoding Standard Application Schema UGAS (UML to GML Application Schema) Updates, Web Services Façade, Reference Architecture Profiling, and Bulk Data Transfer.

- **OWS Innovations**: Explore topics that represent either new areas of work for the Consortium (such as GPS and Mobile Applications), a desire for new approaches to existing technologies to solve new challenges (such as the OGC Web Coverage Service (WCS) work), or some combination of the two.

- **Compliance & Interoperability Testing & Evaluation (CITE)**: Develop a suite of compliance test scripts for testing and validation of products with interfaces implementing the following OGC standards: Web Map Service (WMS) 1.3 Interface Standard, Web Feature Service (WFS) 2.0 Interface Standard, Geography Markup

Language (GML) 3.2.1 Encoding Standard, OWS Context 1.0 (candidate encoding standard), Sensor Web Enablement (SWE) standards, Web Coverage Service for Earth Observation (WCS-EO) 1.0 Interface Standard, and TEAM (Test, Evaluation, And Measurement) Engine Capabilities.

The OWS-9 sponsors are: AGC (Army Geospatial Center, US Army Corps of Engineers), CREAF-GeoViQua-EC, EUROCONTROL, FAA (US Federal Aviation Administration), GeoConnections - Natural Resources Canada, Lockheed Martin Corporation, NASA (US National Aeronautics and Space Administration), NGA (US National Geospatial-Intelligence Agency), USGS (US Geological Survey), UK DSTL (UK MoD Defence Science and Technology Laboratory)

License Agreement

Contents

Page

Figures

Page

OGC® OWS-9 Report on Aviation Performance Study

1 Introduction

1.1 Scope

This OGC® document describes the result work performed in OWS-9 on the Aviation performance study. The report deals with the performance and endurance testing of data provision services commonly used within OWS Aviation testbeds. Test runs have been evaluated on the basis of well-defined, service-specific test models and the results are documented in detail. Furthermore, a description of the service test environment is documented in alignment with the overall OWS-9 service architecture.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Matthes Rieke (editor) (m.rieke<at>uni-muenster.de)	Institute for Geoinformatics (ifgi) – University of Muenster
Alexis James Brooker (alex.brooker<at>snowflakesoftware.com)	Snowflake Software

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2012-07-20	Draft	Matthes Rieke	all	First initial draft of the report.

2012-11-16	Pre-Final	Matthes Rieke	all	Pre-final revision for review
2012-12-21	Final	Matthes Rieke, Alexis Brooker	Test Results, Discussion	Incorporation of the final test results; Discussion on the interpretation of results

1.4 Future work

Improvements in this document are desirable to the following topics.

- **Extended Test models**
  - Determining the maximum number of simultaneous subscribers at an Event Service when still providing an acceptable performance: This test should be executed with an increasing number of subscribers for the same server
  - Extensive WFS testing: An extended model should decrease the interval between each test cycle to simulate a high request load. Additionally, multiple simultaneous requests should be performed to simulate real-world situations.
- **Adhoc Test model manipulation –** The tool currently used built-in WFS & Event Service requests. Future versions should allow the direct manipulation of requests. This could also make the tools workflow more transparent.
- **Taking server machine parameters into account –** evaluation should consider any server-related events (e.g. updates running in the background) and correlate these with performance peaks. Currently, only the network connection and the server setup are taken into consideration when assessing the results.
- **Test environment** – Future work should also focus on the test environment. Here, it would be of great benefit to host the tool and all services to be tested on the same machine in order to provide better comparable results.
- **Incorporation of additional services** – The study could be enhanced to also design models for additional services of the Aviation architecture (e.g. WFS, WCS).

1.5 Foreword

This document is a deliverable of the OGC Web Services (OWS) Initiative - Phase 9 (OWS-9). Its contents cover the summary of the work carried out regarding performance and endurance testing of data provision services, namely Web Feature Service and Event Service. It describes the definition of dedicated test models and the internal dataflow. Additionally, the underlying test environment is described in detail. The results of the performed tests are illustrated via graphs and diagrams for each individual test run.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OWS-9 Engineering Reports:

- OGC 12-147, OWS-9 Aviation Architecture Engineering Report

Aviation specific documents:

- EUROCONTROL & FAA (2010): AIXM 5 Temporality Model. Available online at http://www.aixm.aero/gallery/content/public/AIXM51/AIXM%20Temporality%201.0.pdf

3 Conventions

3.1 Abbreviated terms

AIXM	Aeronautical Information Exchange Model
DNOTAM	Digital NOTAM
ES	Event Service
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
NOTAM	Notice to Airmen
OGC	Open Geospatial Consortium
OWS-9	OWS Testbed Phase 9
UUID	Universally Unique Identifier
UML	Unified Modeling Language
WFS	Web Feature Service
WPS	Web Processing Service
WXXM	Weather Information Exchange Model
XML	Extensible Markup Language
XPath	XML Path Language

3.2 UML notation

Most diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3]. Besides static structure diagrams, sequence diagrams are used to illustrate information/data flow.

4 OWS-9 Report on Aviation Performance Study overview

Within previous OWS testbeds the focus was primarily on the realization of functional requirements. This study aims at the evaluation of the overall service architecture for an operational environment where performance and endurance is a fundamental requirement. To achieve this, several test models have been developed to provide insights on the workflow and the involved services, namely Web Feature Service (WFS) and Event Service (ES).

The performance and endurance testing is settled in two different service environments. Test results of a remote service setup (services deployed on physically distributed server machines) are compared to local service deployment (local server, local network). In order to provide meaningful results, the developed test models have been implemented and deployed into the performance study tool. This enabled the reusability of the developed tools. Hence, test results can easily be compared among different server instances and test environments.

5 Service Test Environment

To provide a most realistic scenario the main service environment tests will focus on a remote environment. As in a classis service-oriented architecture, clients and services will be physically distributed in most situations. Such a setup allows a reasonable evaluation of the involved components.

To get further insights on possible performance lacks an additional local environment was setup. The local setup highly depends on the service providers capabilities. In particular, the infrastructure must be able to host the developed tool in the same network (or even on the same machine) as the server to be tested. For OWS-9 all service providers provided the possibility to host the Performance Study tool in their local server environment.

5.1 Remote Environment

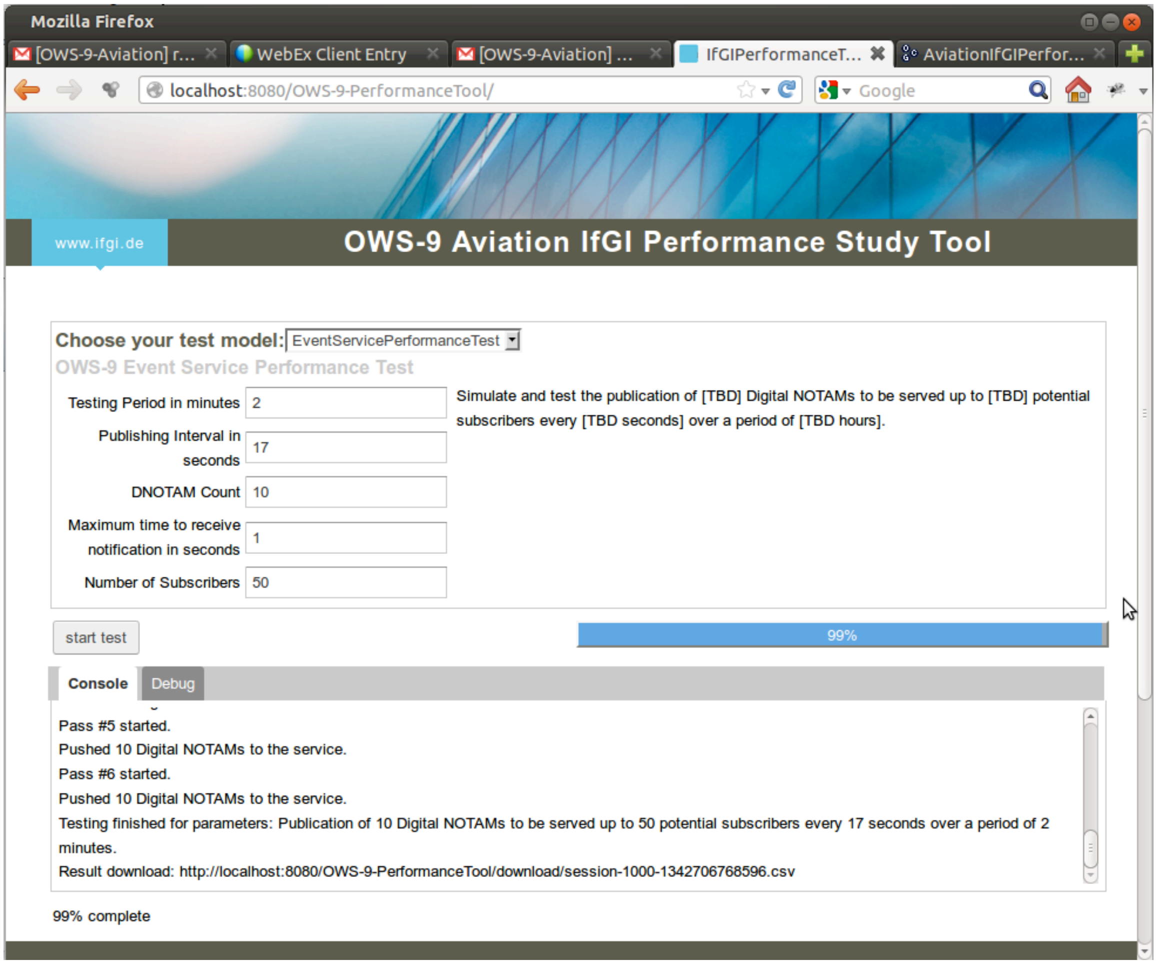
The remote test environment is a simulation of the general aviation service architecture. Here, the services were hosted physically within the network of the providing vendor.

5.2 Local Environment

In contrast to the remote environment, the tool was deployed on the same local network (or even the same server machine) as the tested service.

5.3 Test Model Execution

The Performance Study tool provides a reusable environment for service test models. The developed test models within the Performance Study ER have been implemented against the available (Java) interfaces of the tool. The user can select the test model he would like to perform within the user interface. The Tool is provided as a browser-based version (see Figure 1). The prototype provides an HTML form interface where the user is able to define the parameters of a specific test model. This has the benefit that most of the users are highly familiar with the usage of browsers and forms.



Choose your test model: EventServicePerformanceTest

OWS-9 Event Service Performance Test

Testing Period in minutes	2
Publishing Interval in seconds	17
DNOTAM Count	10
Maximum time to receive notification in seconds	1
Number of Subscribers	50

Simulate and test the publication of [TBD] Digital NOTAMs to be served up to [TBD] potential subscribers every [TBD seconds] over a period of [TBD hours].

```
Pass #5 started.
Pushed 10 Digital NOTAMs to the service.
Pass #6 started.
Pushed 10 Digital NOTAMs to the service.
Testing finished for parameters: Publication of 10 Digital NOTAMs to be served up to 50 potential subscribers every 17 seconds over a period of 2
minutes.
Result download: http://localhost:8080/OWS-9-PerformanceTool/download/session-1000-1342706768596.csv
```

99% complete

Figure 1 - Browser-based Performance Tool.

Different aspects of the workflow are measured (e.g. min/max/mean time it takes for a Notification to pass through the Event Service; see section 6 for details) and stored in dedicated attributes. Results are provided for download. Currently, a CSV file and a graph illustration are created for every test model run.

6 Test Models

The basis for this study was the design and implementation of abstract test models. The following sections provide insights on the setup and the workflow of the developed models.

6.1 Web Feature Service

The overall goal for WFS performance testing is to check if an answer is received by a client application in less than a defined timespan. Figure 2 provides a high-level representation of the developed WFS test models.

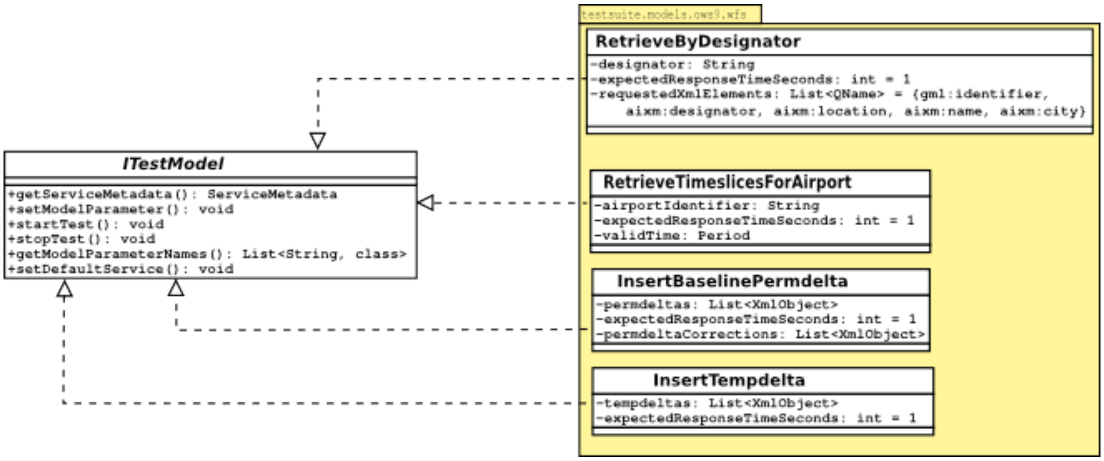

testsuite.models.ows9.wfs
RetrieveByDesignator
-designator: String
-expectedResponseTimeSeconds: int = 1
-requestedXmlElements: List<QName> = {gml:identifier, aixm:designator, aixm:location, aixm:name, aixm:city}
ITestModel
+getServiceMetadata(): ServiceMetadata
+setModelParameter(): void
+startTest(): void
+stopTest(): void
+getModelParameterNames(): List<String, class>
+setDefaultService(): void
RetrieveTimeslicesForAirport
-airportIdentifier: String
-expectedResponseTimeSeconds: int = 1
-validTime: Period
InsertBaselinePermdelta
-permdeltas: List<XmlObject>
-expectedResponseTimeSeconds: int = 1
-permdeltaCorrections: List<XmlObject>
InsertTempdelta
-tempdeltas: List<XmlObject>
-expectedResponseTimeSeconds: int = 1

Figure 2 - Web Feature Service Test Models.

6.1.1 GetFeature Models

GetFeature models are focusing on the function of WFS as a datastore. The performance of a GetFeature model in general is depending highly on the amount of data loaded to the service. At the time of model execution the WFS of Vendor A had 43649 and the WFS of Vendor B 22311 Airport features loaded. As both of the GetFeature models request Airport data the feature count of the remaining features have been considered.

6.1.1.1 Workflow

In contrast to the ES models the Request/Response pattern could be applied in the design of the workflow. Figure 3 illustrates the involved steps of a GetFeature model execution.

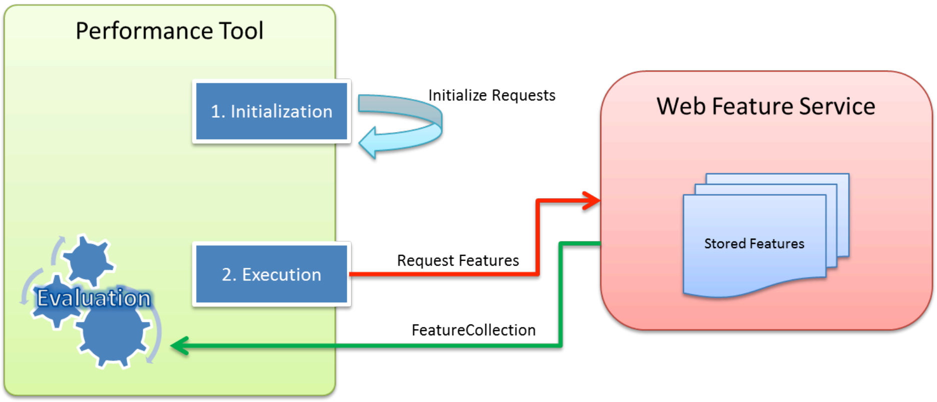


Figure 3 - GetFeature Model Workflow.

6.1.1.2 Parameters

The parameters differ for GetFeature and Transactional models. The following table provides an overview of the parameters for a GetFeature model.

Parameter	**Description**	**Default Value**
Testing Period	The overall time of model execution in minutes. The model performs a check after execution cycle if it has exceeded the testing periods.	120 (= 2 hours)
Request interval	Defines the time in seconds the tool will wait after successful execution of a test cycle before starting the next.	5
Maximum Response Time	Similar to the ES models this parameter defines a threshold. If it is exceeded the request is considered as overdue. Note: this parameter is currently not taken into consideration in the evaluation.	5
GML Identifier / AIXM Designator Portion	Used within the GetFeature query. This value should be set in a way it will produce a result (= existing in the WFS).	N/A
GML validTime	Used within the GetFeature query.	N/A

6.1.1.3 Evaluation

For the GetFeature model the following parameters have been used for the evaluation of a service instance.

Parameter	**Definition**
Response Time	This parameter measures the time it has taken for the request to execute. It indicated a probably high service trivially.
Matched Features	The amount of matched feature in for the used GetFeature query as defined in the wfs:FeatureCollection. Ideally this parameter is constant for each test cycle indicating a stable and reliable service (no transactions should have been made during the model execution).
Returned Features	The amount of returned feature in for the used GetFeature query as defined in the wfs:FeatureCollection. Similar to the previous parameter it should be constant amongst all test cycles.

6.1.1.4 Retrieve by Designator Portion

This model requests the gml:identifier, designator, name, position and city served of all airports which designators start with a specific portion provided as a wildcard pattern. As the OWS-9 WFS instances did not both provide the functionality to retrieve specific parameters of an AIXM feature the complete set of TimeSlices is requested instead. The goal is to assess the performance of the internal wildcard pattern matching. Listing 1 defines an example request which uses the FES operator “PropertyIsLike” which supports wildcard matching capabilities.

Listing 1 – Retrieve by Designator Portion Request.

```
<wfs:GetFeature xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:aixm="http://www.aixm.aero/schema/5.1" service="WFS" version="2.0.0" count="10">
    <wfs:Query typeNames="aixm:AirportHeliport">
        <fes:Filter>
            <fes:PropertyIsLike wildCard="%" singleChar="_" escapeChar="\">

    <fes:ValueReference>aixm:timeSlice/aixm:AirportHeliportTimeSlice/aixm:designator
</fes:ValueReference>
                <fes:Literal>LF%</fes:Literal>
            </fes:PropertyIsLike>
        </fes:Filter>
    </wfs:Query>
</wfs:GetFeature>
```

6.1.1.5 Retrieve by GML Identifier

In contrast to the previous model, this model retrieves all Airport TimeSlices which are valid for a specified time. The query limits the result to only match the Airport with a specific gml:identifier (represented via a UUID). Besides the UUID matching the temporal filtering methods of the WFS instance are tested within this model. An example request is provided in Listing 2.

Listing 2 – Retrieve by GML Identifier Request.

```
<wfs:GetFeature xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:aixm="http://www.aixm.aero/schema/5.1" service="WFS" version="2.0.0" count="100">
    <wfs:Query typeNames="aixm:AirportHeliport">
        <fes:Filter xmlns:fes="http://www.opengis.net/fes/2.0">
            <fes:And>
                <fes:PropertyIsEqualTo>
                    <fes:ValueReference>gml:identifier
          </fes:ValueReference>
                    <fes:Literal>C12-E7AB-5FC6-1EEEE040A8C0020260C5-1868
          </fes:Literal>
                </fes:PropertyIsEqualTo>
                <fes:AnyInteracts>
                    <fes:ValueReference>aixm:timeSlice//gml:validTime
          </fes:ValueReference>
                    <gml:TimePeriod gml:id="validTimeStart">
                        <gml:beginPosition>2010-11-09T10:28:58+0100
            </gml:beginPosition>
                        <gml:endPosition>2012-11-09T10:28:58+0100
            </gml:endPosition>
                    </gml:TimePeriod>
                </fes:AnyInteracts>
            </fes:And>
        </fes:Filter>
    </wfs:Query>
</wfs:GetFeature>
```

6.1.2 Transactional Models

As the WFS's within the OWS-9 Aviation thread have to deal with dynamic features the designed test models have to take the dynamic nature of features into account as well. On top of that, the AIXM 5.1 specifics also have to be considered. In particular, this includes the TimeSlice Commissioning/Cancellation patterns defined in the AIXM 5 Temporality Model [1]. The WFS-T models developed within this work follow these guidelines. Detailed workflows are described in the following sections.

6.1.2.1 Workflow

Similar to the GetFeature model the Request/Response pattern is applied. Though, four (Permdelta + Correction) respectively two (Tempdelta) requests have been posted to the WFS within one test cycle. Figure 4 illustrates the workflow of the transactional model.

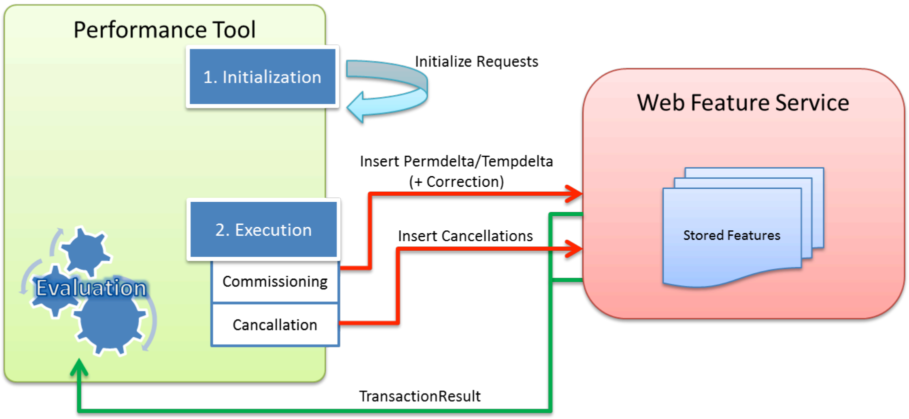


Figure 4 - WFS Transactional Model Workflow.

6.1.2.2 Parameters

The parameters of the inserted features are generated using an internally developed algorithm (based on randomization of textual patterns). Thus, the user needs to provide a limited set of parameters. These are described in the following table.

Parameter	**Description**	**Default Value**
Testing Period	The overall time of model execution in minutes. The model performs a check after execution cycle if it has exceeded the testing periods.	1440 (= 1 day)
Request interval	Defines the time in seconds the tool will wait after successful execution of a test cycle before starting the next.	40
Maximum Response Time	Similar to the ES models this parameter defines a threshold. If it is exceeded the request is considered as overdue. Note: this parameter is currently not taken into consideration in the evaluation.	5

6.1.2.3 Evaluation

The evaluation of the Transactional model is in similar to the GetFeature model evaluation. It differs as the number of requests differs for a single test cycle. For the insertion of Permdeltas/Correction four requests (1 commissioning, 1 correction, 2 cancellations) are needed, the insertion of a Tempdelta creates two requests (1 insertion, 1 cancellation). The following table illustrates the parameters which are used for the evaluation.

Parameter	**Definition**
Response Time	This parameter measures the time it has taken for the request to execute. It indicated a probably high service trivially. It is evaluated for all request of a test cycle (e.g. four) individually. Similar to the GetFeature models this parameter indicated a probably high service load.
Total Inserted Features	The amount of inserted features as defined in the wfs:TransactionResponse. Ideally this parameter has a value of "1" as each request inserts a single feature indicating a stable and reliable service.
Mean Response Time	This parameter represents the calculated mean of all requests posted in one test cycle.

6.1.2.4 Insert PERMDELTAs and a Corresponding Correction

The workflow of this model creates four separate requests:

1. Insertion of Permdelta (= Commissioning)
2. Insertion of a corresponding Permdelta Correction
3. Cancellation of the Correction
4. Cancellation of the Commissioning.

The used requests are listed in Annex A.2.1 of this document.

6.1.2.5 Insert TEMPDELTAs

The workflow of this model creates two separate requests:

1. Insertion of Tempdelta for an existing Airport
2. Cancellation of the Tempdelta.

The used requests are listed in Annex A.2.1 of this document.

6.2 Event Service

The following high-level UML diagram (Figure 5) summarizes the parameters of the ES test models.

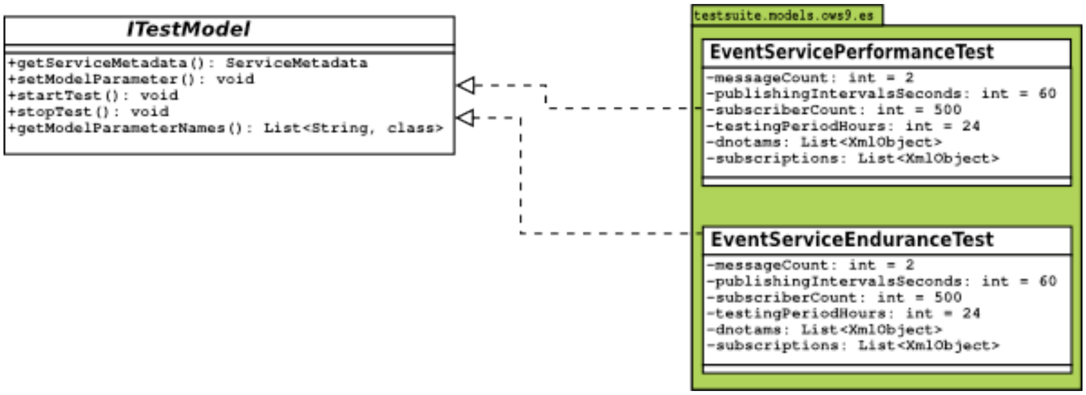

ITestModel
+getServiceMetadata(): ServiceMetadata
+setModelParameter(): void
+startTest(): void
+stopTest(): void
+getModelParameterNames(): List<String, class>
testsuite.models.ows9.es
EventServicePerformanceTest
-messageCount: int = 2
-publishingIntervalsSeconds: int = 60
-subscriberCount: int = 500
-testingPeriodHours: int = 24
-dnotams: List<XmlObject>
-subscriptions: List<XmlObject>
EventServiceEnduranceTest
-messageCount: int = 2
-publishingIntervalsSeconds: int = 60
-subscriberCount: int = 500
-testingPeriodHours: int = 24
-dnotams: List<XmlObject>
-subscriptions: List<XmlObject>

Figure 5 - Event Service Test Models.

Based on the UML Class diagram, the two ES models have been implemented.

6.2.1 Workflow

Figure 6 provides a high-level overview of the ES model workflow. The involved phases are described in the following sections in more detail.

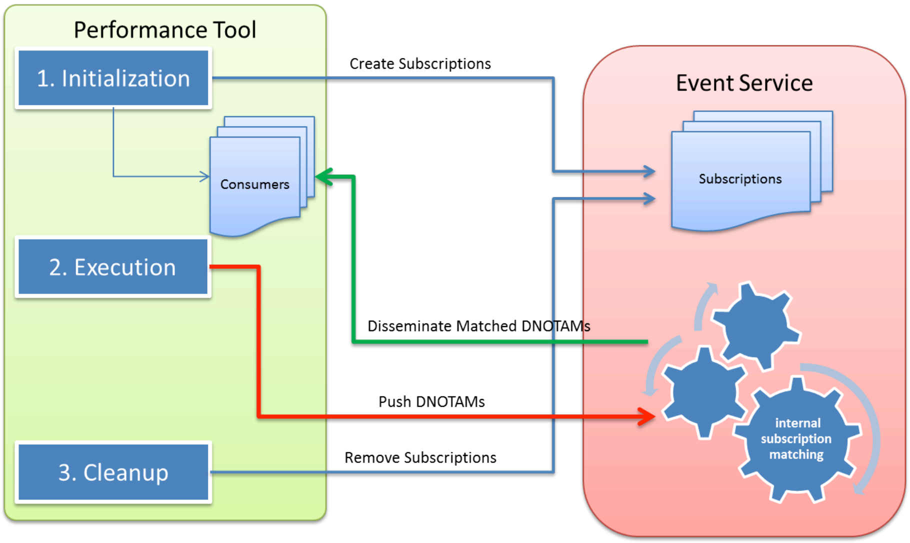


Figure 6 - Event Service Test Model High-Level Overview.

6.2.1.1 Initialization

The basis for performance testing model is designed to provide the essential logic for stress testing the event service. Here, the initial setup to put the event service into a test-ready state is applied. In particular, this includes the execution of all needed subscriptions as well as the initialization of dummy consumer for each subscription which receives notifications. Subsequent to the initial setup, the actual stress test is started with the provided parameters (see UML class properties) in a cyclic manner.

6.2.1.2 Performance and Endurance Execution

The developed test models provide capabilities to separate received notifications based on the subscriptions made. This gives additional insights on presumable observed performance drawbacks. Additionally, the individual time required for sending out a DNOTAM event and the reception of the corresponding notification at the dummy client is measured. The following section describes the measured parameters and their evaluation in more detail. After the execution of the test model has finished the tool cleans up the ES instance by removing all created subscriptions.

6.2.2 Parameters

The ES models can be configured with a set of parameters which all have influence on the results. The following table provides an overview of the available parameters and their effects. As both the endurance and the performance testing are based on the same workflow a common model has been developed. The table provides the parameter values used to set up endurance and performance testing.

Parameter	Description	Performance Value	Endurance Value
Testing Period	The overall time of model execution in minutes. The model performs a check after every DNOTAM publishing if it has exceeded the testing periods.	60 (= 1 hour)	1440 (= 1 day)
Number of Subscribers	The count of Subscribers which are considered to be simulations of Aviation clients.	500	500
DNOTAM Count	The number of Digital NOTAM pushed to the ES within one publishing interval.	10	2
Publishing Interval	The amount of seconds the tool waits after it has successfully published a	60	30

	set of DNOTAMs.		
Consumer URL	An option to provide an alternative Subscription consumer endpoint. Left blank to use the internal consumer evaluation.		
Maximum time to receive notification	This parameter is used within the evaluation. It determines whether a subscription is considered as an "overdue message". Here, the time in seconds is measured it takes from the DNOTAM publishing until the reception at the internal consumers.	2	1

6.2.3 Evaluation

The following table provides an overview of the parameters measured by the ES model. All of these parameters are measured separately for every subscription type (e.g. XPath or spatial filter) used in the model.

Parameter	**Definition**
Received Notifications	For each test cycle the model collects all messages which arrived within the specified publishing interval (see previous section). Ideally, this parameter is constant for every cycle meaning that every notification has been received successfully. A variation is an indicator for a performance issue or unstable service conditions.
Minimum Response Time	This parameter measures the shortest time it has taken for a Notification from being pushed to the service until the reception at the built-in consumer.
Maximum Response Time	Same as previous parameter but measuring the longest time.
Mean Response Time	As every single Notification is traced and its response time is measured, this parameter provides the mean response time similar to the previous parameters.
Overdue Notifications	Based on the measured response time and the user-defined value of "Maximum time to receive notification" (see previous section) this parameter indicates how many of the Notifications exceeded the user-defined time for every test

	cycle. In the resulting graphs (see section 7) it is illustrated by a red area indicating a probable high service load.

6.2.4 Used Digital NOTAM

To simulate a real-world information flow the DNOTAMs defined in the overall OWS-9 Aviation Scenario are used: DNOTAMs propagating a closure of a RunwayDirection feature (see Listing 3) followed by DNOTAMs with re-opening the same feature.

Listing 3 - Event Service Digital NOTAM Data.

```
<wsnt:Notify xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
    <wsnt:NotificationMessage>
        <wsnt:Message>
            <message:AIXMBasicMessage gml:id="gmlid1"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:event="http://www.aixm.aero/schema/5.1/event"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:gsr="http://www.isotc211.org/2005/gsr"
xmlns:gss="http://www.isotc211.org/2005/gss"
xmlns:gts="http://www.isotc211.org/2005/gts"
xmlns:message="http://www.aixm.aero/schema/5.1/message"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aixm.aero/schema/5.1/message
http://www.aixm.aero/gallery/content/public/schema/5.1/message/AIXM_BasicMessage.xsd
http://www.aixm.aero/schema/5.1/event
http://www.aixm.aero/gallery/content/public/schema/5.1/event/Event_Features.xsd">
                <message:hasMember>
                    <event:Event gml:id="uuid.ff3ab666-d8d0-428c-8304-
5922ee4636b5z2a46017">
                        <gml:identifier codeSpace="urn:uuid:">ff3ab666-d8d0-428c-8304-
5922ee4636b5</gml:identifier>
                        <event:timeSlice>
                            <event:EventTimeSlice gml:id="NOTAM_4_09_1">
                                <gml:validTime>
                                    <gml:TimePeriod>
                                        <gml:beginPosition>2012-11-
22T22:00:00.000Z</gml:beginPosition>
                                        <gml:endPosition>2012-11-
23T04:00:00.000Z</gml:endPosition>
                                    </gml:TimePeriod>
                                </gml:validTime>
                                <aixm:interpretation>BASELINE</aixm:interpretation>
                                <aixm:sequenceNumber>1</aixm:sequenceNumber>
                                <aixm:featureLifetime>
                                    <gml:TimePeriod>
                                        <gml:beginPosition>2012-11-
22T22:00:00.000Z</gml:beginPosition>
                                        <gml:endPosition>2012-11-
23T04:00:00.000Z</gml:endPosition>
                                    </gml:TimePeriod>
                                </aixm:featureLifetime>
                                <event:name>NOTAM_4_09_1</event:name>
                                <event:encoding>DIGITAL</event:encoding>
                                <event:scenario>RWY.CLS</event:scenario>
                                <event:textNOTAM>
                                    <event:NOTAM>
                                        <event:series>A</event:series>
                                        <event:number>5</event:number>
                                        <event:year>2012</event:year>
                                        <event:type>N</event:type>
                                        <event:issued>2012-11-
21T11:03:00.000Z</event:issued>
                                        <event:affectedFIR>EADD</event:affectedFIR>

    <event:selectionCode>QMRLC</event:selectionCode>
```

```
                                        <event:traffic>IV</event:traffic>
                                        <event:purpose>NBO</event:purpose>
                                        <event:scope>A</event:scope>

    <event:coordinates>5222N03157W</event:coordinates>
                                        <event:radius>999</event:radius>
                                        <event:location>EADD</event:location>

    <event:effectiveStart>1211222200</event:effectiveStart>

    <event:effectiveEnd>1211230400</event:effectiveEnd>
                                        <event:text>RWY 09L/27R CLOSED.</event:text>
                                        <event:lowerLimit>000</event:lowerLimit>
                                        <event:upperLimit>999</event:upperLimit>
                                        <event:publisherNOF>#urn.uuid.c225ae5c-540f-
4a48-8867-809b393b2407</event:publisherNOF>
                                    </event:NOTAM>
                                </event:textNOTAM>
                            </event:EventTimeSlice>
                        </event:timeSlice>
                    </event:Event>
                </message:hasMember>
                <message:hasMember>
                    <aixm:RunwayDirection gml:id="urn.uuid.c8455a6b-9319-4bb7-b797-
08e644342d64z2a46017">
                        <gml:identifier codeSpace="urn:uuid:">runway-1</gml:identifier>
                        <aixm:timeSlice>
                            <aixm:RunwayDirectionTimeSlice>
                                <gml:validTime>
                                    <gml:TimePeriod>
                                        <gml:beginPosition>2012-11-
22T22:00:00.000Z</gml:beginPosition>
                                        <gml:endPosition>2012-11-
23T04:00:00.000Z</gml:endPosition>
                                    </gml:TimePeriod>
                                </gml:validTime>
                                <aixm:interpretation>TEMPDELTA</aixm:interpretation>
                                <aixm:sequenceNumber>1</aixm:sequenceNumber>
                                <aixm:featureLifetime>
                                    <gml:TimePeriod>
                                        <gml:endPosition>unknown</gml:endPosition>
                                    </gml:TimePeriod>
                                </aixm:featureLifetime>
                                <aixm:availability>
                                    <aixm:ManoeuvringAreaAvailability>

    <aixm:operationalStatus>CLOSED</aixm:operationalStatus>
                                    </aixm:ManoeuvringAreaAvailability>
                                </aixm:availability>
                                <aixm:extension>
                                    <event:RunwayDirectionExtension gml:id="ex01">
                                        <event:theEvent xlink:href="#uuid.ff3ab666-
d8d0-428c-8304-5922ee4636b5z2a46017"/>
                                    </event:RunwayDirectionExtension>
                                </aixm:extension>
                            </aixm:RunwayDirectionTimeSlice>
                        </aixm:timeSlice>
                    </aixm:RunwayDirection>
                </message:hasMember>
            </message:AIXMBasicMessage>
        </wsnt:Message>
    </wsnt:NotificationMessage>
</wsnt:Notify>
```

6.2.5 Used Subscriptions

All listed subscriptions are used alongside each other within the Event Service testing. Such an approach is more realistic than separate testing of each subscription[1]. The overall sum of subscribers is equally parted into these subscriptions (e.g. 500 subscribers separated into 166 XPath subscriptions for gml:identifier="runway-1", 167 for "runway-2", 167 for spatial filter).

Listing 4 illustrates the used XPath subscription. An XPath subscription is often used to subscribe for updates on a certain feature (e.g. an Airport).

Listing 4 – Simple XPath Subscription.

```
<wsnt:MessageContent Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116"
xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
    //aixm:RunwayDirection//gml:identifier[text()="runway-1"]
</wsnt:MessageContent>
```

Listing 5 defines a subscription with a spatial filter. Such a subscription is used to determine updates for a specific area of interest (e.g. a buffer around the aircrafts flight route).

Listing 5 – Spatial Filter Subscription.

```
<wsnt:MessageContent xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:wsnt="http://docs.oasis-open.org/wsn/b-
2" Dialect="http://www.opengis.net/ses/filter/level2">
    <fes:Filter xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:swe="http://www.opengis.net/swe/1.0.1">
        <fes:Within>
            <fes:ValueReference>wfs-aixm:extentOf(.)</fes:ValueReference>
            <fes:Literal>
                <gml:Polygon gml:id="aoi_01">
                    <gml:exterior>
                        <gml:LinearRing>
                            <gml:posList>50.130536 7.083609 53.130536 7.083609
53.130536 7.983609 50.130536 7.983609</gml:posList>
                        </gml:LinearRing>
                    </gml:exterior>
                </gml:Polygon>
            </fes:Literal>
        </fes:Within>
    </fes:Filter>
</wsnt:MessageContent>
```

7 Test Results

The test results have been anonymized within the publicly available version of this ER. See section 7.4 for details.

[1] Nevertheless, the implemented tool provides the feature to only use one type of subscription in order to test scenario-specific requirements.

The vertical scale of the provided diagrams may vary across the different test models. Wherever possible it was attempted to use the same scale for one model. Nevertheless, this was not always possible as some details of the result would not have been displayed. Therefore, the reader should always take the vertical scale into consideration when comparing test results.

7.1 Vendor Server Setup

The server configuration hosting the web service to be tested can influence results strongly. Thus, all service vendors provided a brief overview of parameters which influence the performance of the web service which are outlined in the following sections.

These parameters are only a subset of external influences which can occur during a performance test. Consequently, additional observations should be taken into account in the future. Details on this aspect of this study are summarized in section 7.4.

In addition to the server configuration the spatial distance between the involved endpoints influence the measured response times of the remote testing environment. A simple route tracing algorithm was used to provide a vague assessment of the connection.

7.1.1 Vendor A

Parameter	**Value**
CPU	Intel Xeon 2.33 GHz L5410 (single core version, circa 2008)
RAM	4GB
Network Connection	8MB down speed and 512 Kb up speed (ADSL)
Operating System	Red Hat Enterprise Linux
Servlet Container	Apache Tomcat 6
Other Performance influencing parameters	15 other WFS instances running within the same Tomcat application server.
Route hops to remote testing endpoint	19

7.1.2 Vendor B

Parameter	**Value**

CPU	Virtual Machine: Intel Xeon E7420 @ 2.13Ghz (single core)
RAM	3GB
Network Connection	100Mbit LAN
Operating System	Fedora Linux FC11 i686
Servlet Container	Apache Tomcat 6.0.29
Other Performance influencing parameters	Every time slice about to be inserted is checked □ validity regarding XML Schema □ compliance to AIXM Temporality Model, including checks on feature associations and their temporal validity □ validity of the supplied geometries, including the resolution of referenced geometries, both according tithe GML and AIXM definitions □ in case of Airspace features: calculation of the decomposed extruded polygons for facilitating airspace altitude queries
Route hops to remote testing endpoint	17

7.1.3 Vendor C

Parameter	**Value**
CPU	Intel Core I5 (3.33GHz)
RAM	8GB
Network Connection	DSL 5 MBits/s (download/upload)
Operating System	Windows 7 64-bit embedded in a VMWare running CentOS 6.2 (64-bit)
Servlet Container	Apache Tomcat 6.0.35
Other Performance	The upload/download speeds depend on network usage. It

influencing parameters	comes close to that limit when it's not in business hours.
Route hops to remote testing endpoint	25

7.1.4 Vendor D

Parameter	**Value**
CPU	Intel Xeon E5430 (2.66 GHz)
RAM	2GB
Network Connection	Cable 20 Mbit/s (download), 12 Mbit/s (upload)
Operating System	Windows XP 32-bit
Servlet Container	Apache Tomcat 6.0.35
Other Performance influencing parameters	none
Route hops to remote testing endpoint	25

7.2 Remote Environment

The tool was deployed on a server machine at IfGI for all services to be tested except the IfGI ES. Here, IDS hosted the tool to enable remote testing of the IfGI ES.

7.2.1 Web Feature Service

7.2.1.1 Retrieve By Designator Portion

7.2.1.1.1 Vendor A

Figure 7 shows the results of this test model. The grey area indicates the number of matched features. As it is constant the service can be considered as stable. Some minor response peaks have been determined but the mean response time is below 500 milliseconds.

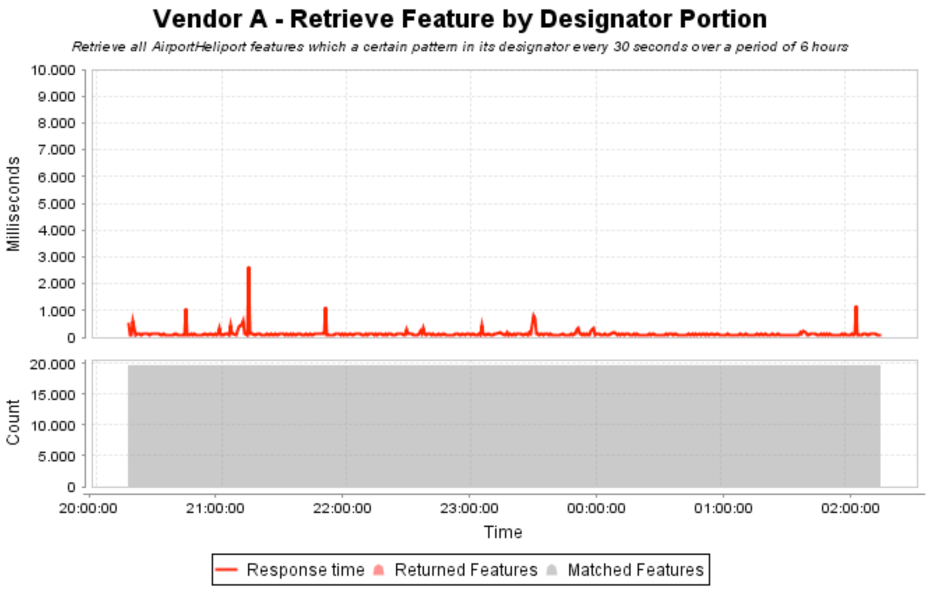


Figure 7 – Vendor A Designator Portion Results (Remote).

7.2.1.1.2 Vendor B

Figure 8 shows the results of this test model. The red area indicates the number of matched features. As it is constant the service can be considered as stable. One minor response peaks (response time greater than 6 seconds) have been determined but the mean response time is around 1 second.

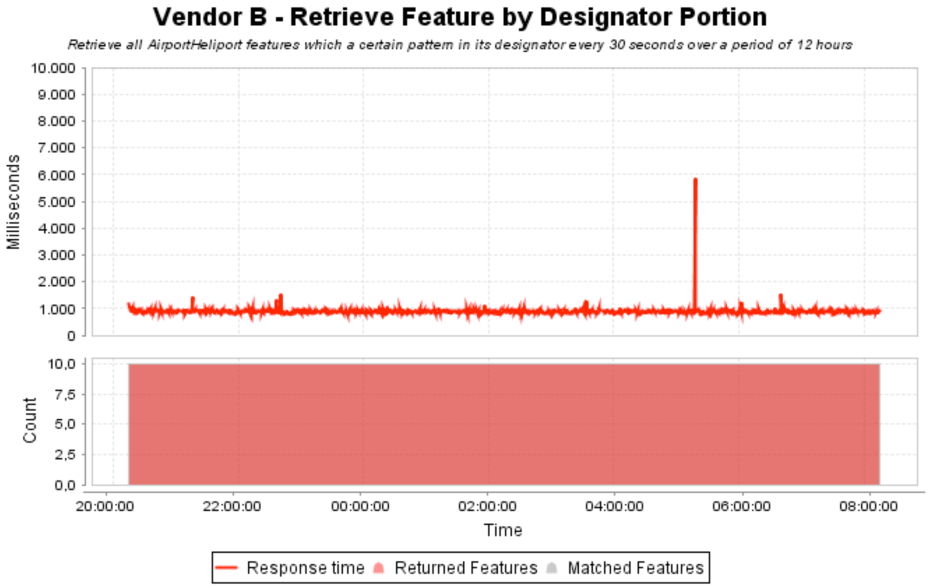


Figure 8 – Vendor B Designator Portion Results (Remote).

7.2.1.2 Retrieve by GML Identifier

7.2.1.2.1 Vendor A

Figure 9 shows the results of this test model. The red area indicates the number of returned features (one for all test cycles). As it is constant the service can be considered as stable. Some major response peaks have been determined.

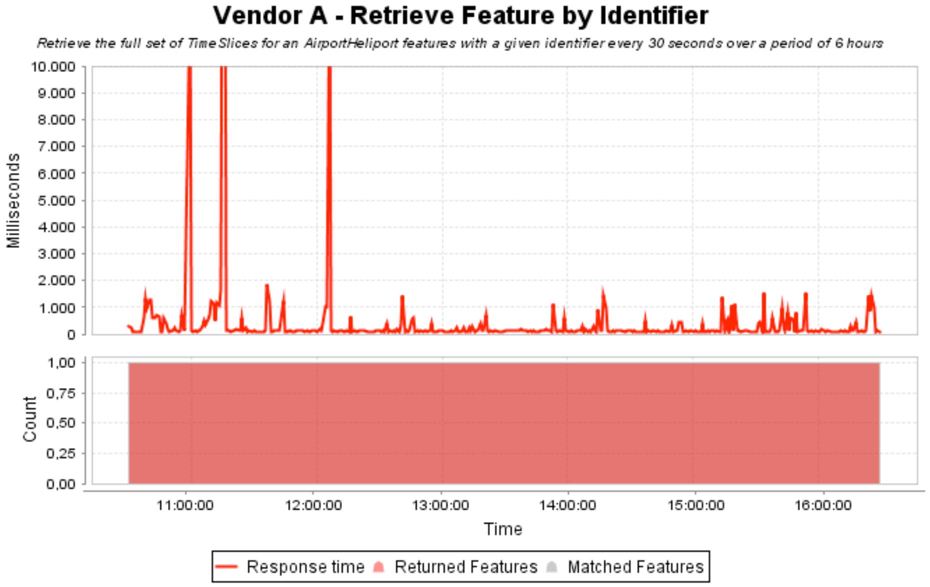


Figure 9 – Vendor A GML Identifier Results (Remote).

7.2.1.2.2 Vendor B

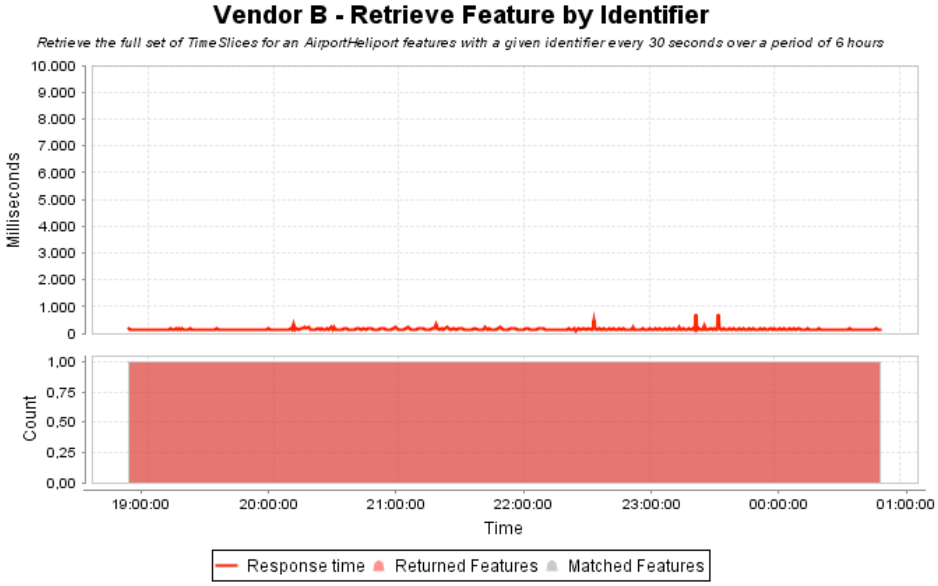


Figure 10 - Vendor B GML Identifier Results (Remote).

7.2.1.3 Insert PERMDELTAs and a Corresponding Correction

7.2.1.3.1 Vendor A

Figure 11 shows the results of this test model. The red area indicates the number of total inserted features (one for all test cycles). As it is constant the service can be considered as stable. One major (response time greater than 80 seconds) and a few minor response peaks have been determined.

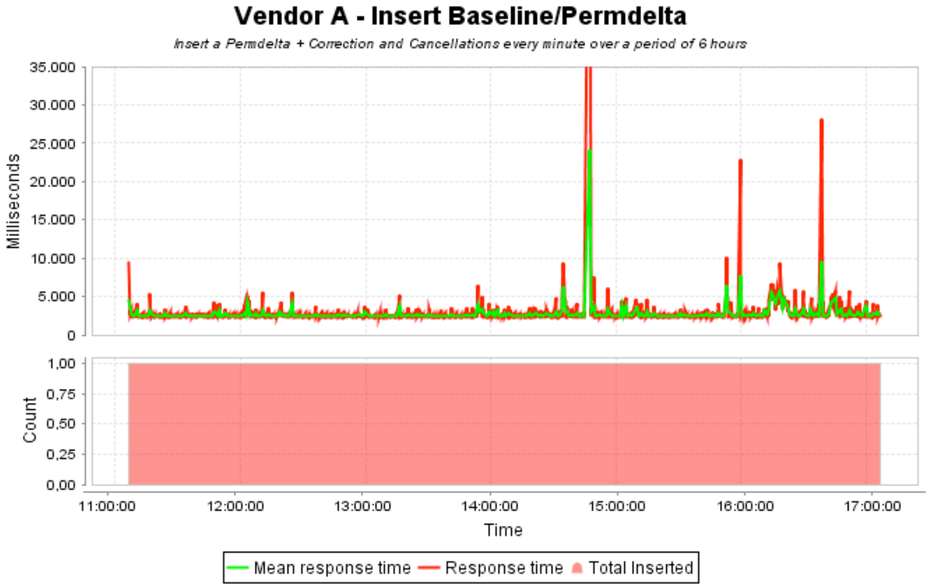


Figure 11 – Vendor A Permdelta Insertion Results (Remote).

7.2.1.3.2 Vendor B

Figure 12 shows the results of this test model. The red area indicates the number of total inserted features. As it is constant the service can be considered as stable. One minor response peak (response time greater than 35 seconds) have been determined and the mean response time is around 10 seconds.

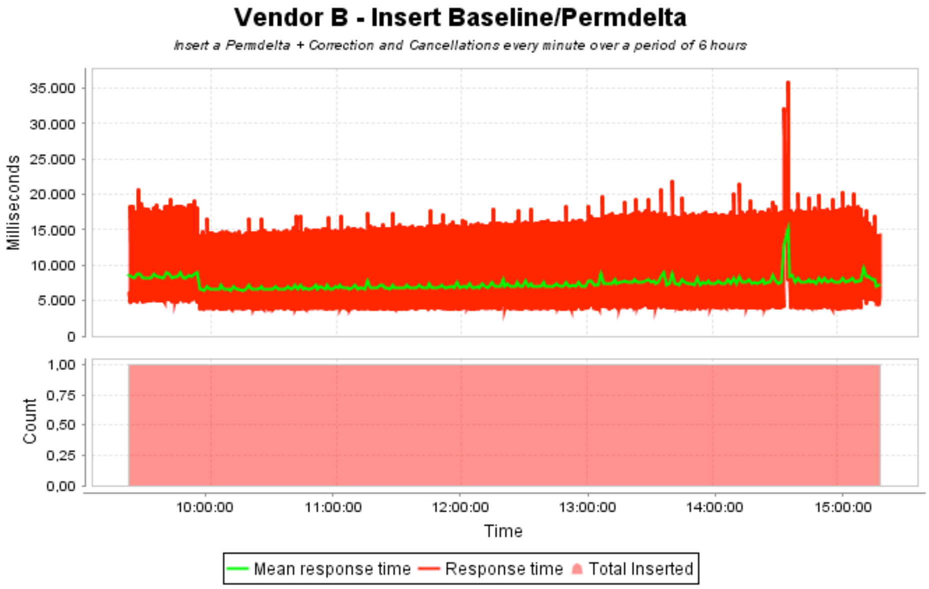


Figure 12 – Vendor B Permdelta Insertion Results (Remote).

7.2.1.4 Insert TEMPDELTAs

7.2.1.4.1 Vendor A

The remote testing results have not been available at the time of providing the final report. This was a result of networking issues which occurred at the vendor's infrastructure.

7.2.1.4.2 Vendor B

Figure 13shows the results of this test model. The red area indicates the number of total inserted features. One insertion has failed (indicated by zero inserted features). This failure correlates with the response time peak of more than 125 seconds.

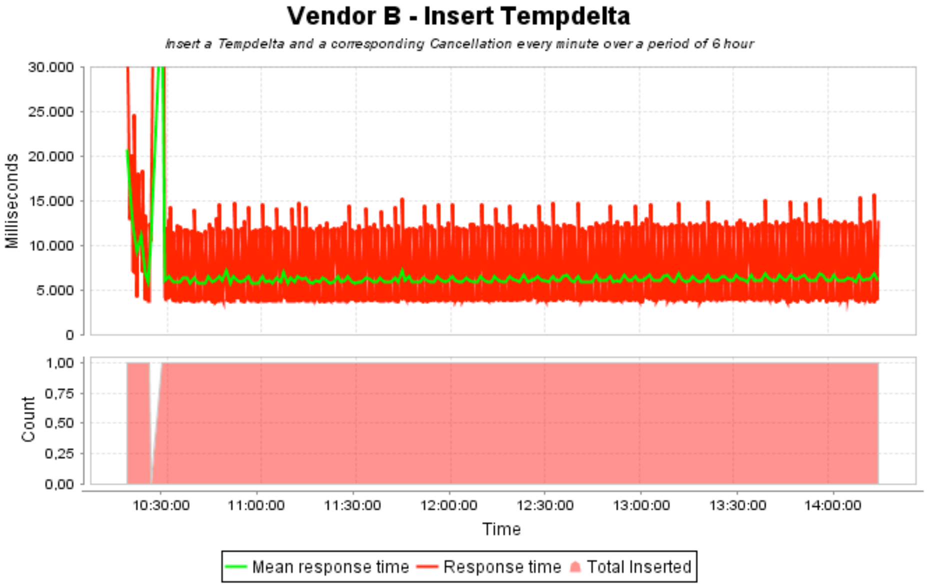


Figure 13 – Vendor B Tempdelta Insertion Results (Remote).

7.2.2 Event Service

A general observation was that the Event Service provided by Vendor C always returned a rather small amount of notifications back to the tool. This could have several causes:

- As the used subscription are very similar to each other (e.g. 500 subscriptions are created from 3 requests) the service may apply a similarity measurements and then ignore the newly created subscriptions

- A subscriptions consumer comparison is applied: When the hostname is equal to the one of a previous subscription, the subscription is ignored

- Other interal pre-processing.

It is for one of these reasons that the received notification count is always under 10 subscriptions per test run.

7.2.2.1 Endurance Model Results

Endurance models have been run for several hours in a cyclic manner. The following sections provide the detailed results.

7.2.2.1.1 Vendor C

The results presented in Figure 14 have a “hole” representing very fast response times. This is likely to be a measurement error. Future work could provide insights on such behavior.

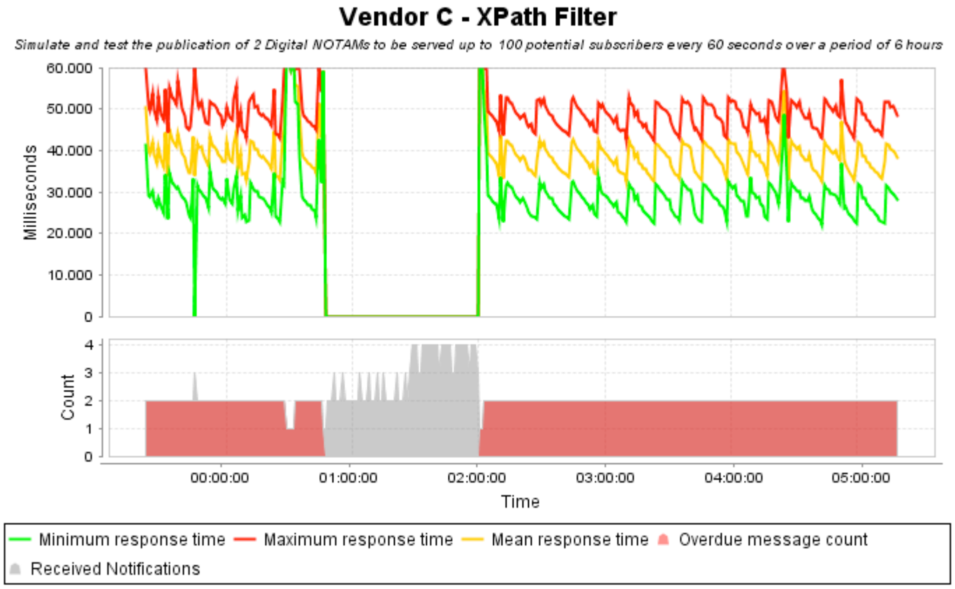


Figure 14 - Vendor C Endurance XPath Filter (Remote).

7.2.2.1.2 Vendor D

Figure 15 presents the results of the XPath subscription. The lower grey area illustrates the received notifications. As it is constant for the complete testing period, the service can be considered as reliable for XPath subscription matching.

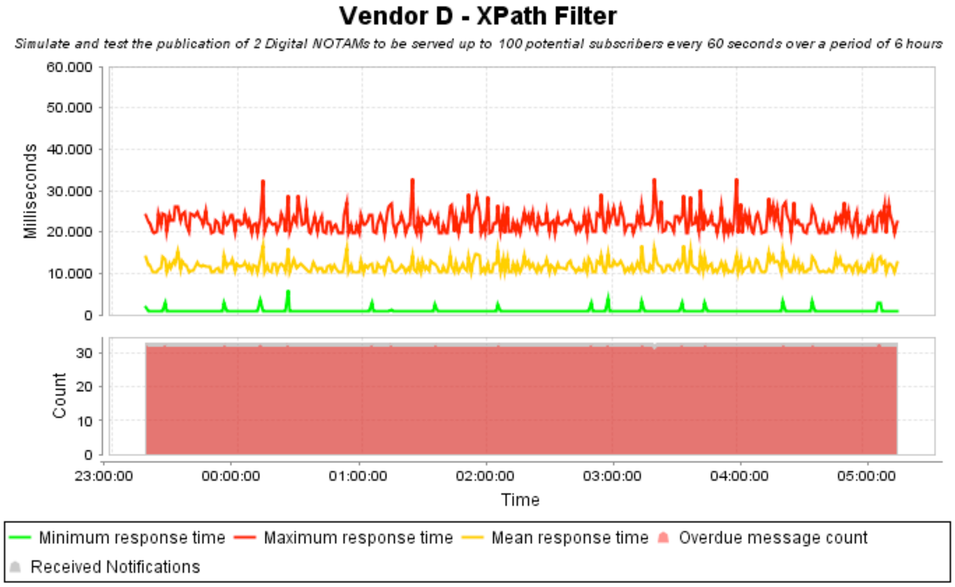


Figure 15 – Vendor D Endurance XPath Filter (Remote).

For the same test run the spatial subscription was used. Figure 16 presents the results of it. Here, the red area in the lower graph indicated the number of overdue messages.

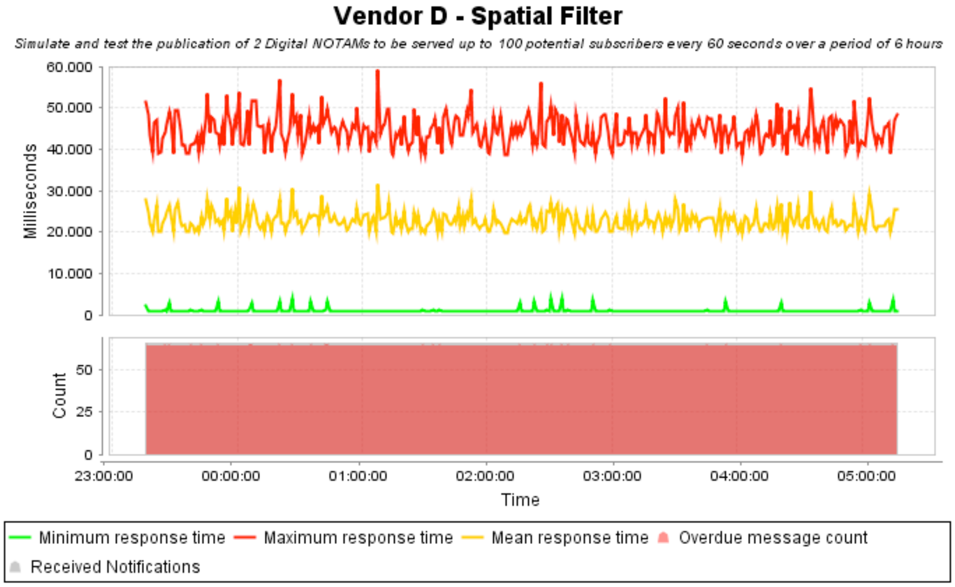


Figure 16 – Vendor D Endurance Spatial Filter (Remote).

7.2.2.2 Performance Model Results

Performance tests have been run for one hour with 500 subscribers at the Event Services. The following sections provide the results.

7.2.2.2.1 Vendor C

As the performance tests did not provide any valuable results, and alternative test run has been carried out. The publication of 4 DNOTAMs, 30 subscribers every minute over a period of 1 hour was tested. The results are provided in Figure 17.

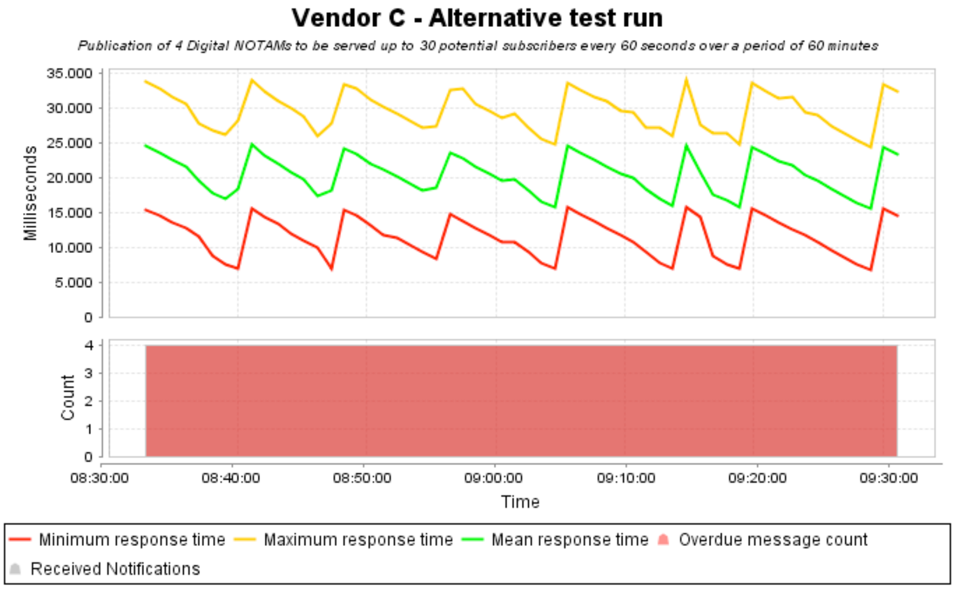


Figure 17 - Vendor C Alternative Test Run (Remote).

7.2.2.2.2 Vendor D

Figure 18 presents the results of the XPath subscription. The lower grey and red areas illustrate the received notifications. The highs and lows in the graph are probably the result of a service overload: 10 DNOTAMs have been sent out every 30 seconds which all have to be disseminated to 500 possible subscribers.

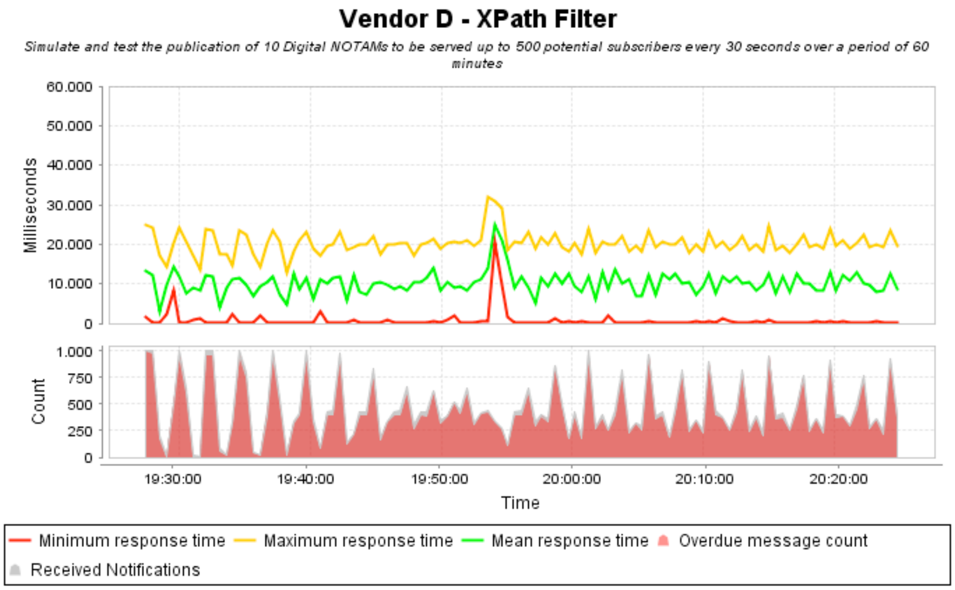


Figure 18 – Vendor D ES Performance XPath (Remote).

Similar to the XPath subscription the Spatial Subscription indicated a subscription malfunction (see Figure 19**Error! Reference source not found.**).

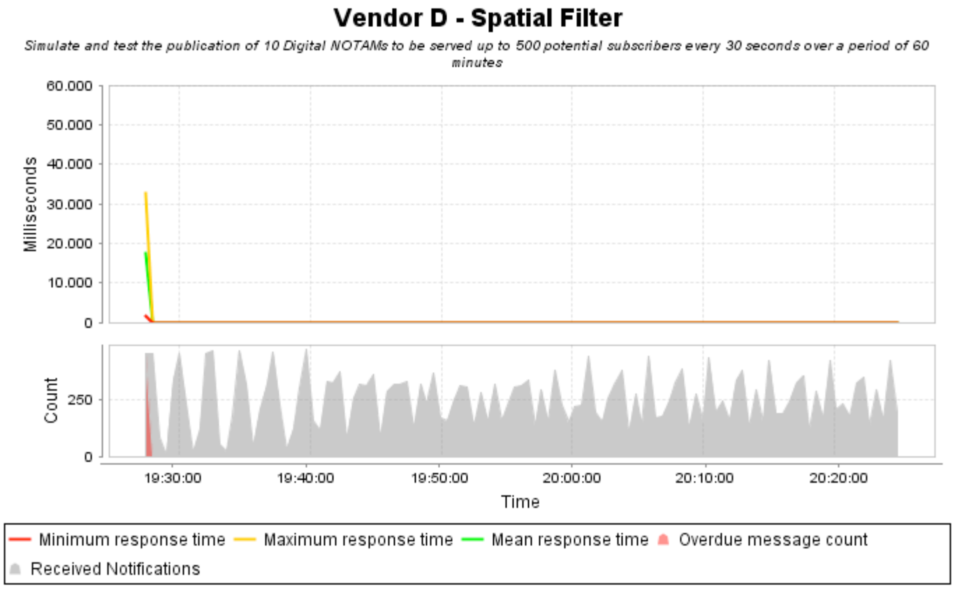


Figure 19 – Vendor D ES Performance Spatial Filter (Remote).

7.3 Local Environment

The tool was deployed on each vendor's local network to reduce the influence of network latencies.

7.3.1 Web Feature Service

7.3.1.1 Retrieve By Designator Portion

7.3.1.1.1 Vendor A

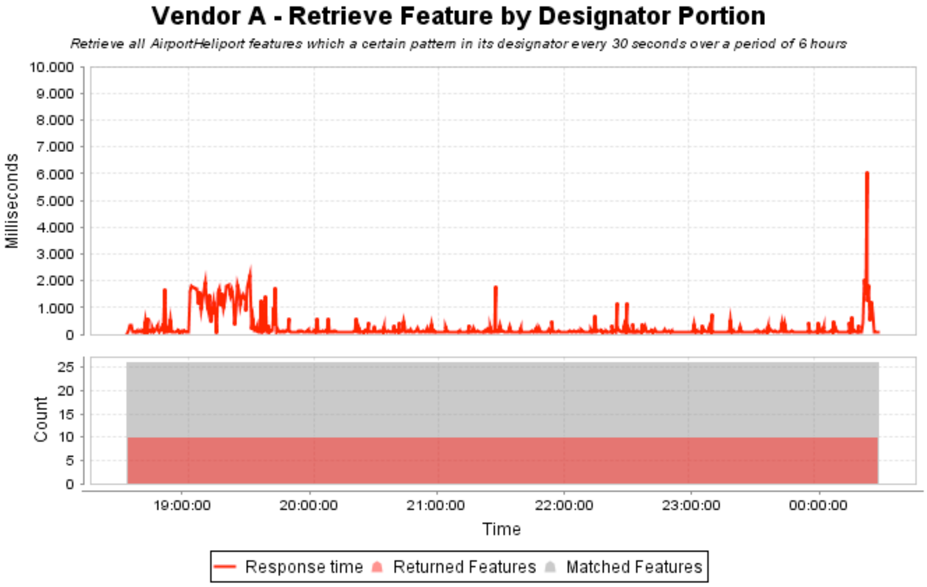


Figure 20 - Vendor A Designator Portion Results (Local).

7.3.1.1.2 Vendor B

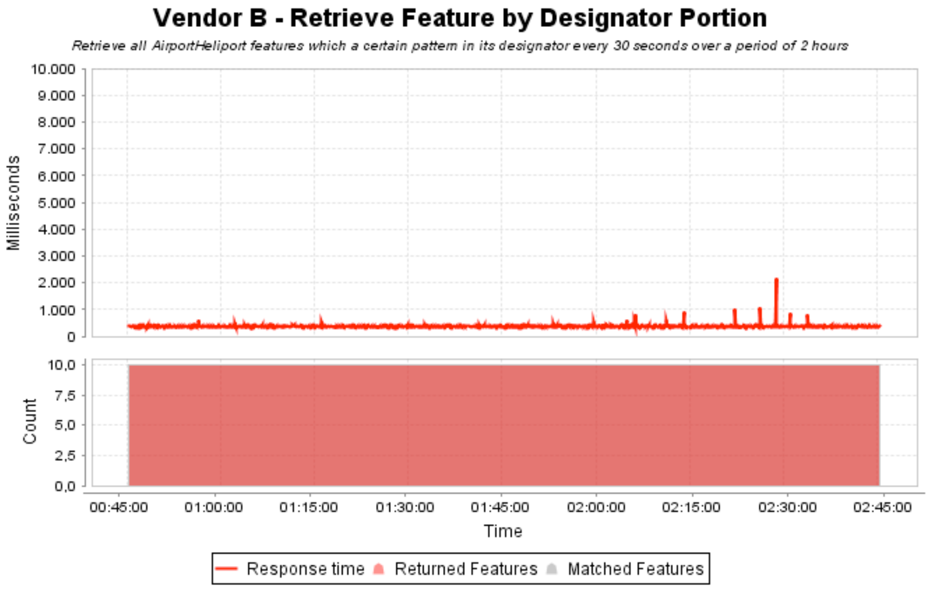


Figure 21 - Vendor B Designator Portion Results (Local).

7.3.1.2 Retrieve by GML Identifier

7.3.1.2.1 Vendor A

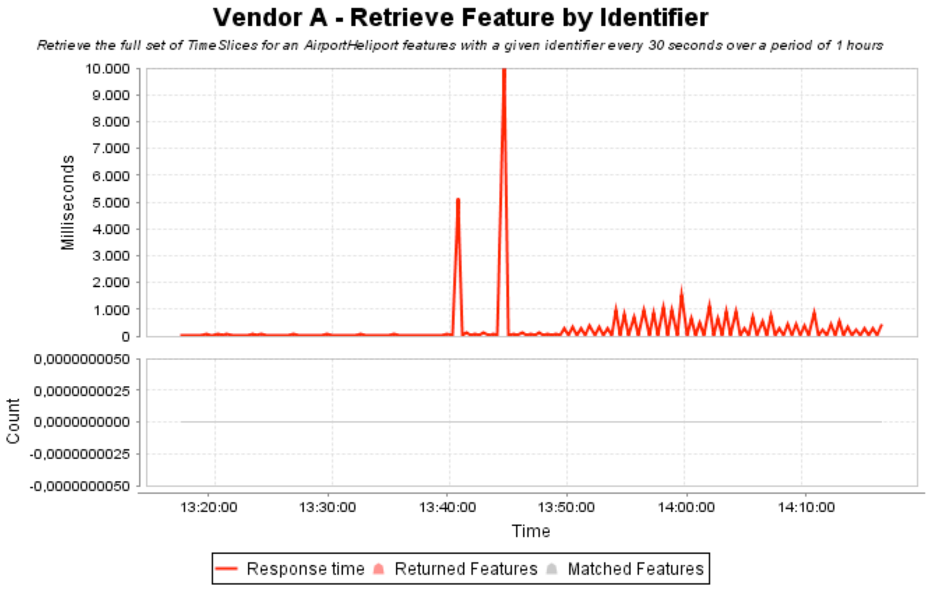


Figure 22 – Vendor A GML Identifier Results (Local).

7.3.1.2.2 Vendor B

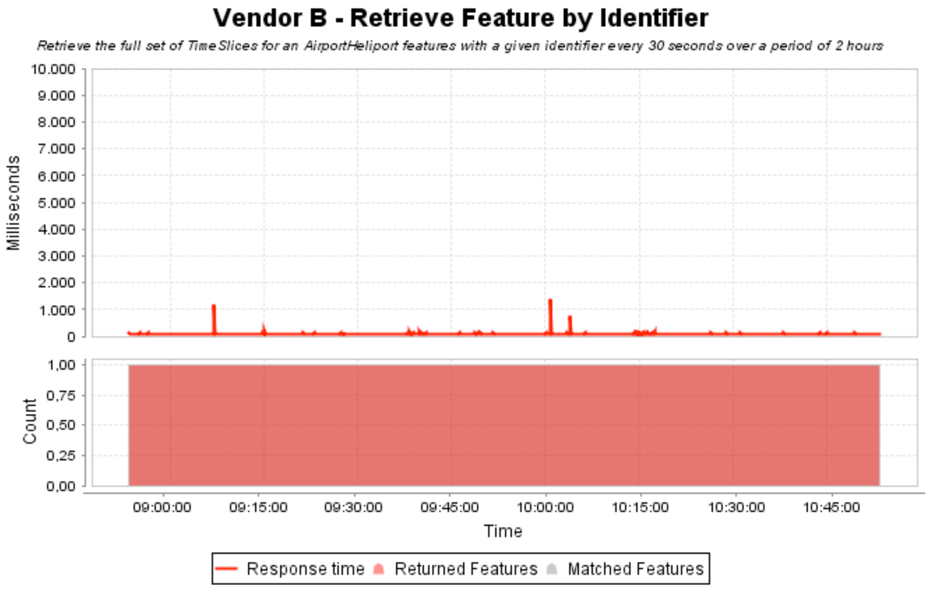


Figure 23 – Vendor B GML Identifier Results (Local).

7.3.1.3 Insert PERMDELTAs and a Corresponding Correction

7.3.1.3.1 Vendor A

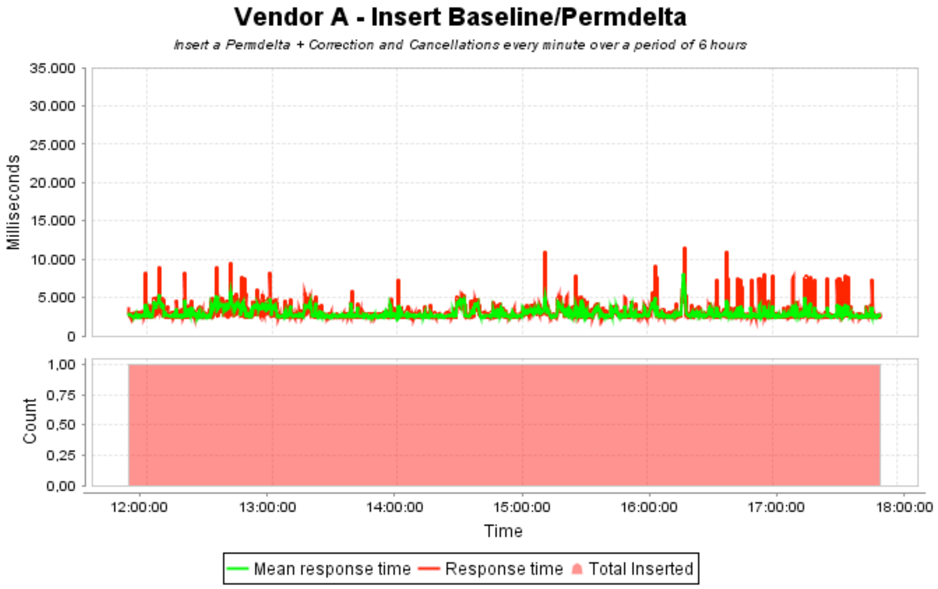


Figure 24 – Vendor A Permdelta Insertion Results (Local).

7.3.1.3.2 Vendor B

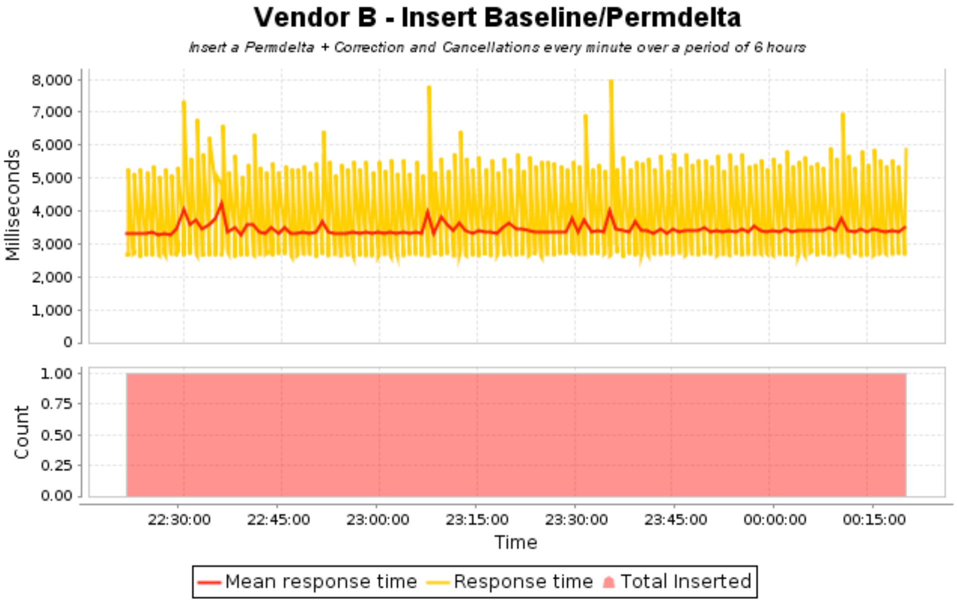


Figure 25 – Vendor B Permdelta Insertion Results (Local).

7.3.1.4 Insert TEMPDELTAs

7.3.1.4.1 Vendor A

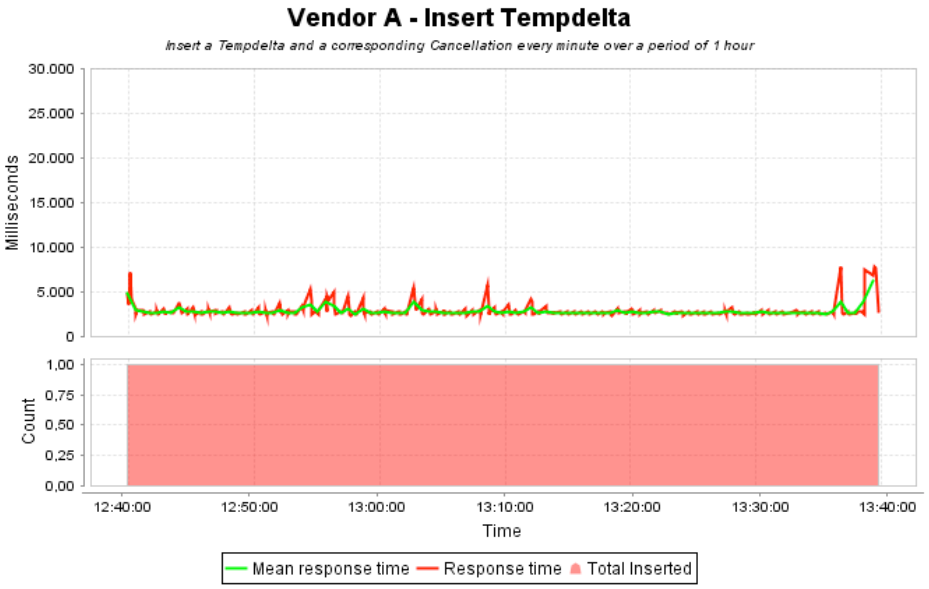


Figure 26 – Vendor A Tempdelta Insertion Results (Local).

7.3.1.4.2 Vendor B

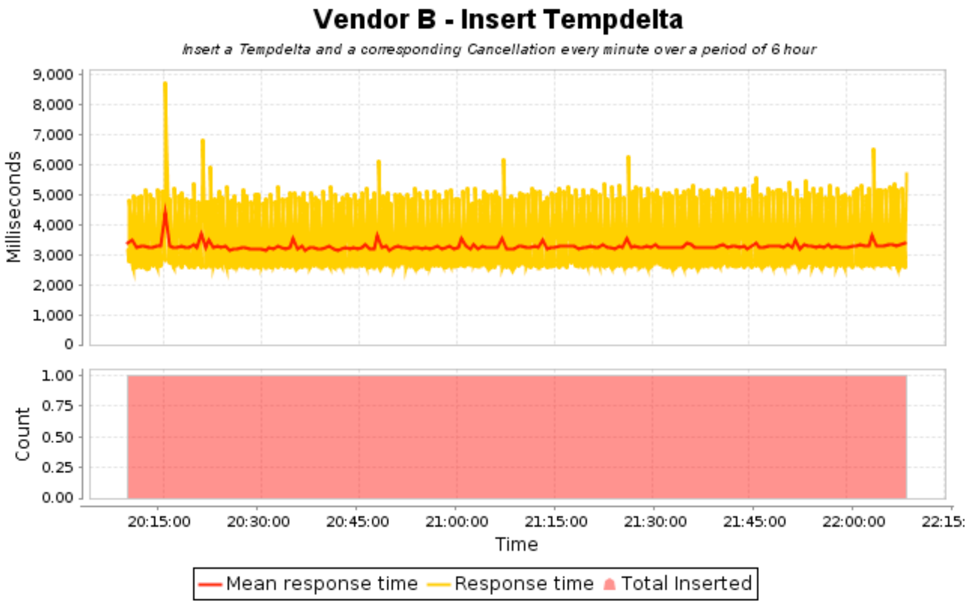


Figure 27 – Vendor B Tempdelta Insertion Results (Local).

7.3.2 Event Service

7.3.2.1 Endurance Model Results

Endurance models have been run for several hours in a cyclic manner. The following sections provide the detailed results.

7.3.2.1.1 Vendor C

The results for endurance testing are illustrated in Figure 28. As the graph is quite compact, an additional detailed view is provided in Figure 29.

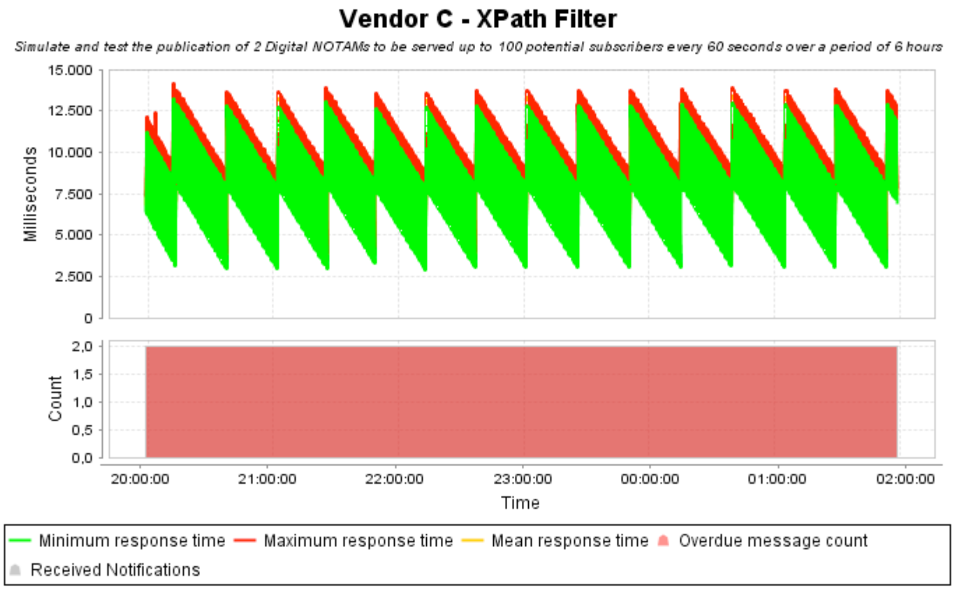


Figure 28 - Vendor C Endurance XPath Filter (Local).

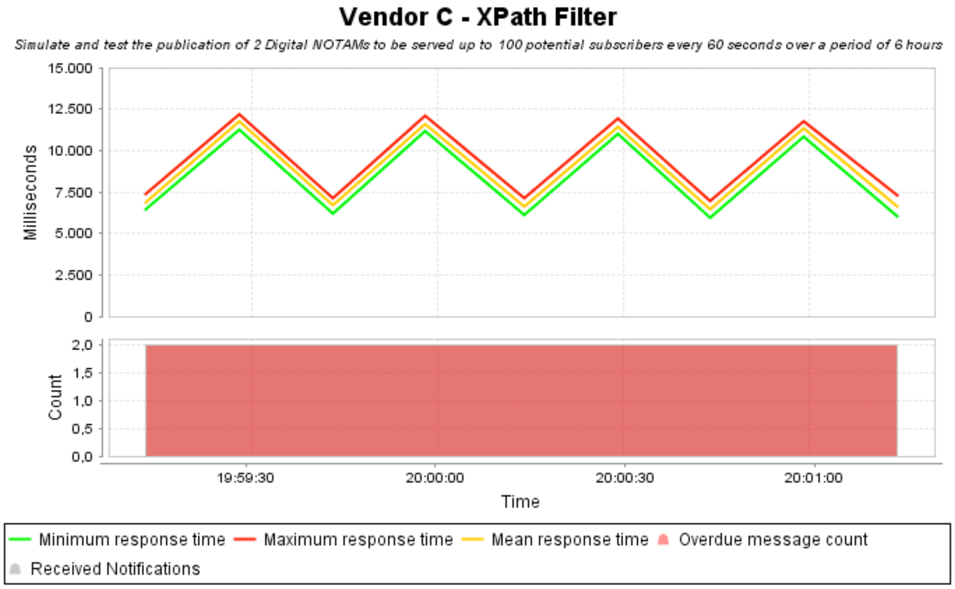


Figure 29 - Vendor C Endurance XPath Filter Details (Local).

For the same test run Figure 30 provides the results of a spatial filter.

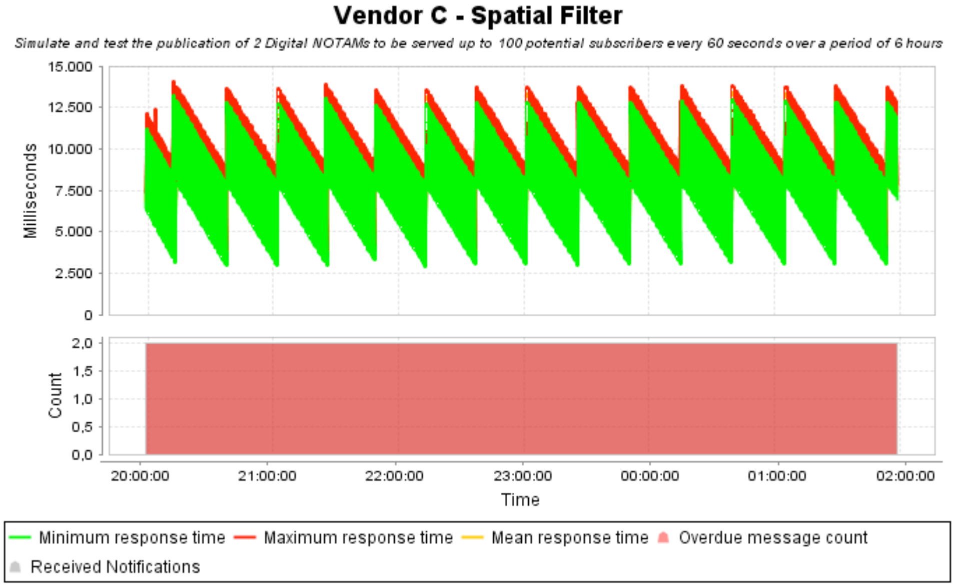


Figure 30 - Vendor C Endurance Spatial Filter (Local).

7.3.2.1.2 Vendor D

Figure 31 presents the results of the XPath subscription. The lower grey area illustrates the received notifications. As it is constant for the complete testing period, the service can be considered as reliable for XPath subscription matching. Only a few overdue messages have been received at the very beginning of the test model. This is not recognizable due to the period of 6 hours. The exact results are provided in a CSV file.

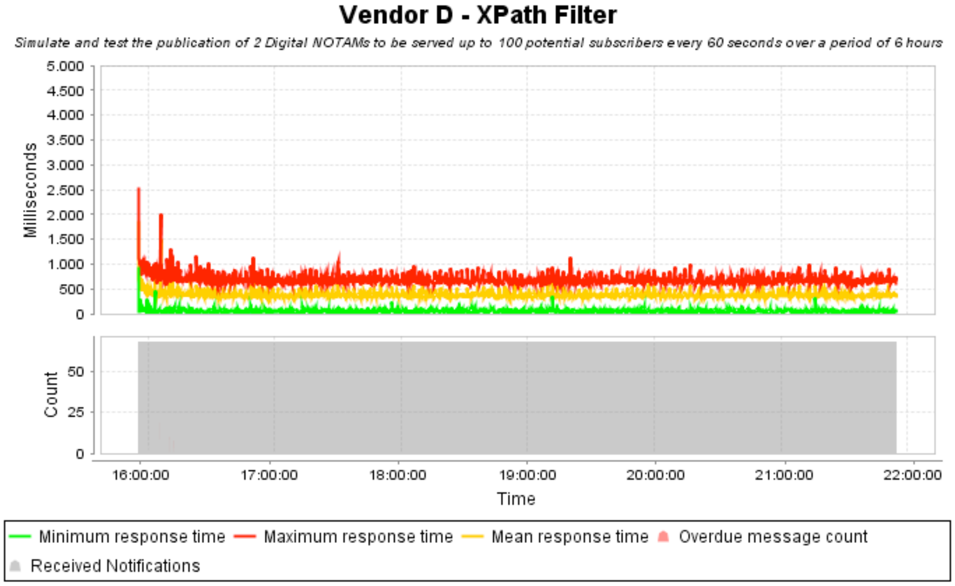


Figure 31 - Vendor D Endurance XPath Filter (Local).

For the same test run the spatial subscription was used. Figure 32 presents the results of it. Here, the red peaks in the lower graph indicated the number of overdue messages.

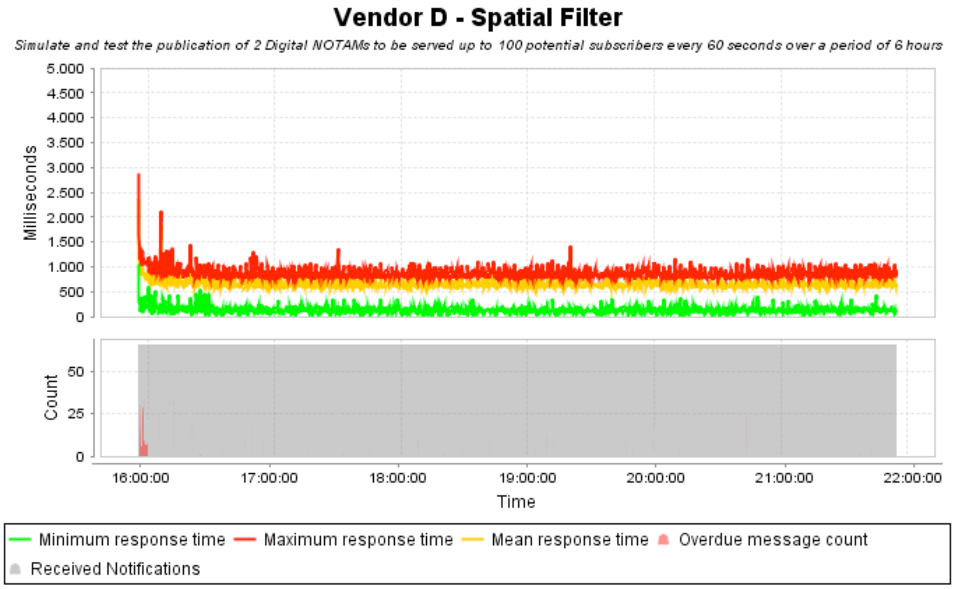


Figure 32 - Vendor D Endurance Spatial Filter (Local).

7.3.2.2 Performance Model Results

Performance tests have been run for one hour with 500 subscribers at the Event Services. The following sections provide the results.

7.3.2.2.1 Vendor C

As the performance tests did not provide any valuable results, and alternative test run has been carried out. The publication of 4 DNOTAMs, 3 subscribers every minute over a period of 1 hour was tested. The results are provided in Figure 33.

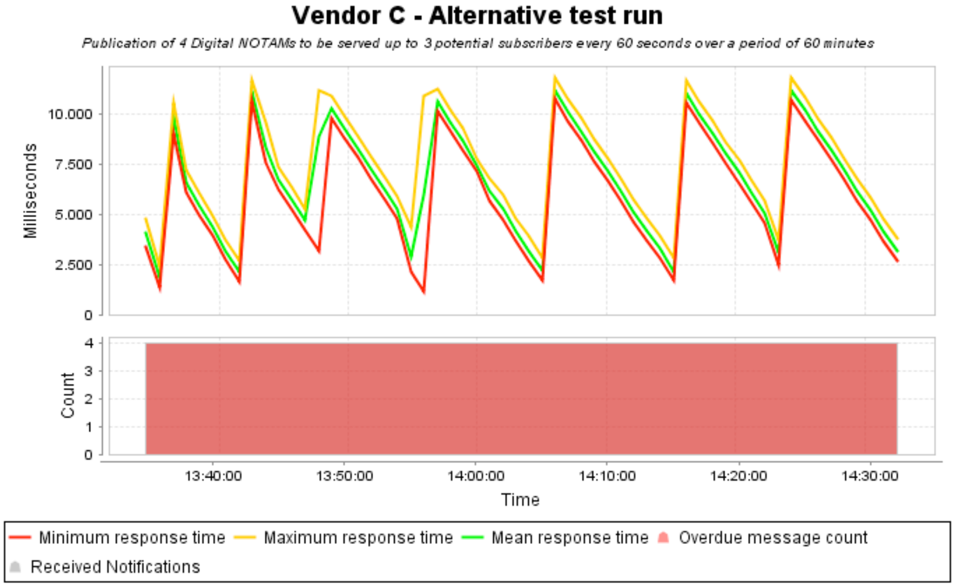


Figure 33 - Vendor C Alternative Test Run (Local).

7.3.2.2.2 Vendor D

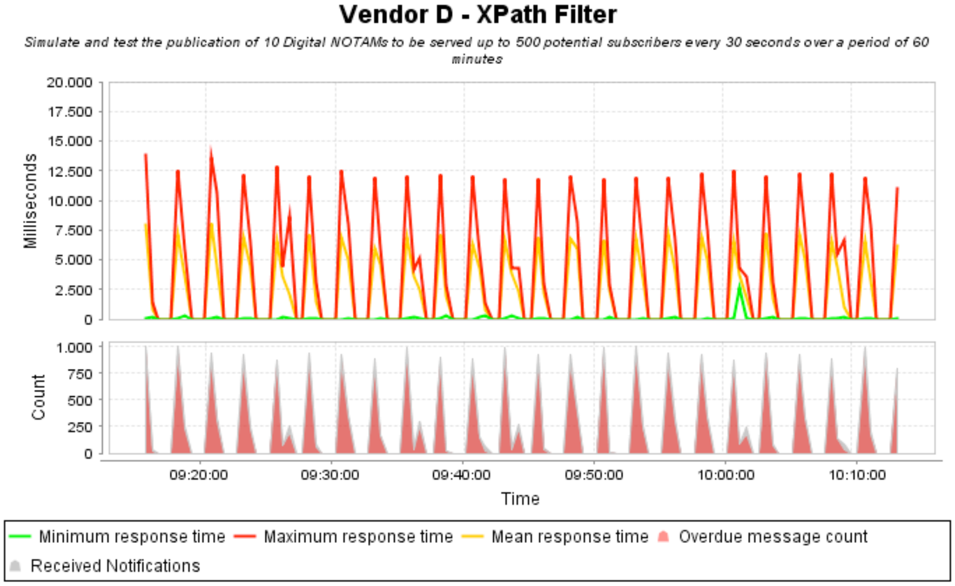


Figure 34 - Vendor D ES Performance XPath Filter (Local).

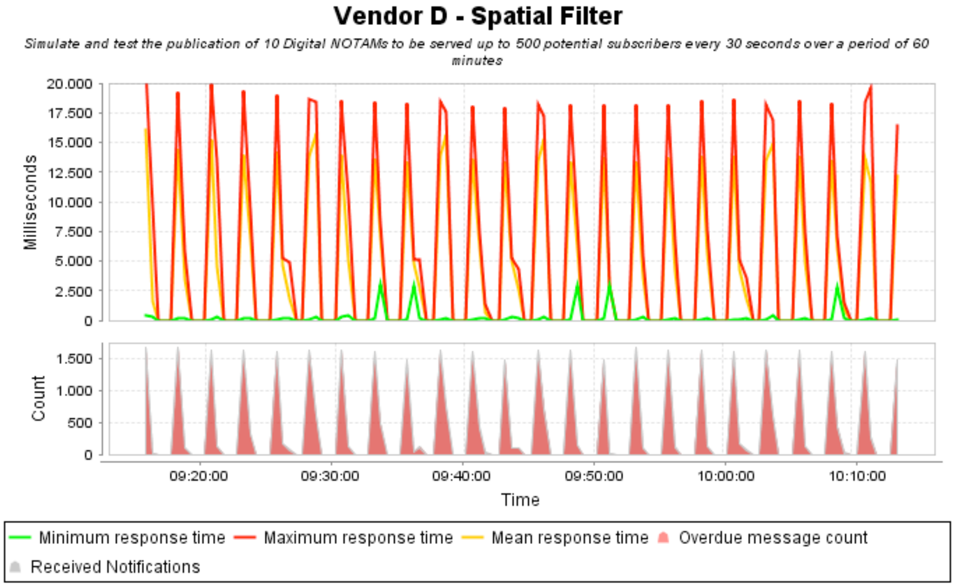


Figure 35 - Vendor D ES Performance Spatial Filter (Local).

7.4 Discussion

The measurements taken during this study, as round trip response times between the test client and the OWS Aviation Thread service endpoints, are composite timings comprising many elements. Some of these elements are outside the control of the OWS-9 participants, where aspects of the wider infrastructure, namely the internet, cause issues such as latency. The network latency between the client, incorporating multiple hops through the internet as well as between each vendor's Internet or Cloud Service provider and the actual service implementation, is highly variable and can add up to or even more than 1000 milliseconds to any single measurement. A highly utilized or congested network will also add a delay to the service request and response times with some services operating through extremely limited bandwidth links.

Other aspects are within the control of the OWS-9 participant or are at least definable, such as the bandwidth of the service implementation's connection to the internet, the supporting software configuration, any database optimization, hardware configuration and caching and some of this information is detailed within this report. Each configuration also varies between each OWS-9 participant.

A scientifically rigorous performance test for the provision of operational open standards based web services in the aviation domain would incorporate both the uncontrolled and controllable aspects detailed above, with defined network architecture, known latency, and the same supporting software and hardware configuration for every service implementation. However, with the current OWS-9 architecture, an "order of magnitude measurement" can be captured and compared with "indicative non-functional performance requirements" for the aviation domain, enabling this work to address the broad question over the suitability for OGC standards based web services and exchange models for a specific domain.

It is with these caveats that the numbers presented in this report must be viewed, and it is for this reason that the specific service implementer or software vendor information has not been correlated with the detailed results.

8 Accomplishments

Within this work a general approach on Aviation data provision services performance testing has been developed. Abstract test models form the basis for this approach. The service-specific requirements as well as real-world situations have been taken into consideration in the model design process. This is reflected by the internal workflow of these models (e.g. Commissioning and cancellations of a feature and its TimeSlices within WFS-T; push-based communication pattern for Event Services).

On top of this approach, a reusable tool has been created to provide test models which can be performed against the data provision web services of the Aviation Architecture. The developed test models are designed in a way to simulate real-world situations (e.g. using the event-based communication pattern of the Event Service). Thus, these models provide results which can provide details on the performance, reliability and stability of the tested service. In addition, the results of this study also provide evidence that a standards-based web service architecture is feasible to provide data and events in the Aviation domain.

8.1 Issues

The developed tool should be considered as a working prototype. A stable execution cannot be guaranteed always. During some test runs, a few malfunctions have been observed. These have been documented and future work on the tool should focus on fostering the stability of the model execution.

8.2 Lessons Learned

- If the tool is to be applied within a testbed similar to OWS-9, it should be used in an early phase of the project (e.g. when dataloading has been finished). As the tool is testing the commonly used functions of the services, it can early provide insights on possible issues.

9 Recommendations

The following subsections provide some recommendations in the field of data provision performance study work.

9.1 Complex Models

- Currently, the test models do not take the actual content of the requested data into account when providing results. Here, it should also be considered to develop complex models which have a-priori knowledge of the request data. These models could then assess the data based on the previously made predictions. This would provide an additional aspect besides performance results.

- If the tool is to be applied within a testbed similar to OWS-9, it should be used in an early phase of the project (e.g. when dataloading has been finished). As the tool is testing the commonly used functions of the services, it can early provide insights on possible issues.

9.2 Possible Architecture Improvements

In general, the architecture was observed as stable and reliable. A few minor tweaks or service-level adjustments could increase the performance of the overall Aviation architecture. For instance

- Development of general caching mechanisms (e.g. after a DNOTAM has been send out, the WFS should expect requests on the affected feature → cache the feature to allow performant retrieval)

- Using a local network for production environments – as the Event Services are closely coupled to the WFS instances when providing certain functionality (e.g. filtering of non-spatial features; see OGC 12-147 for details) a fast network connection would improve the performance of Event Services a lot.

Annex A – Used Request XML

A.1 WFS GetFeature Requests

All provided requests make use of placeholders which are filled with generated/user-specified contents. These placeholders are defined as e.g. “${placeholder}”.

A.1.1 GetFeature by Designator Portion

```
<wfs:GetFeature xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:aixm="http://www.aixm.aero/schema/5.1" service="WFS" version="2.0.0" count="10">
   <wfs:Query typeNames="aixm:AirportHeliport">
      <fes:Filter>
            <fes:PropertyIsLike wildCard="%" singleChar="_" escapeChar="\">

<fes:ValueReference>aixm:timeSlice/aixm:AirportHeliportTimeSlice/aixm:designator
</fes:ValueReference>
               <fes:Literal>${designatorPortion}</fes:Literal>
            </fes:PropertyIsLike>
      </fes:Filter>
   </wfs:Query>
</wfs:GetFeature>
```

A.1.2 GetFeature By GML Identifier and GML validTime

```
<wfs:GetFeature xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" service="WFS" version="2.0.0"
  count="100">
  <wfs:Query typeNames="aixm:AirportHeliport">
    <fes:Filter xmlns:fes="http://www.opengis.net/fes/2.0">
      <fes:And>
        <fes:PropertyIsEqualTo>
          <fes:ValueReference>gml:identifier</fes:ValueReference>
          <fes:Literal>${gmlIdentifier}</fes:Literal>
        </fes:PropertyIsEqualTo>
        <fes:AnyInteracts>
          <fes:ValueReference>aixm:timeSlice//gml:validTime</fes:ValueReference>
          <gml:TimePeriod gml:id="validTimeStart">
            <gml:beginPosition>${validTimeStartIso}</gml:beginPosition>
            <gml:endPosition>${validTimeEndIso}</gml:endPosition>
          </gml:TimePeriod>
        </fes:AnyInteracts>
      </fes:And>
    </fes:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

A.2 WFS Transactional Requests

A.2.1 Insert Baseline/Permdelta

Baseline/Permdelta Commissioning:

```
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WFS"
  version="2.0.0">
  <wfs:Insert>
    <aixm:AirportHeliport gml:id="uuid.${randomUUID}">
      <gml:identifier codeSpace="http://www.opengis.net/ows9/aviation/performanceStudy">
${randomUUID}</gml:identifier>
      <gml:boundedBy>
        <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
          <gml:lowerCorner>${coordinate}
          </gml:lowerCorner>
          <gml:upperCorner>${coordinate}
          </gml:upperCorner>
        </gml:Envelope>
      </gml:boundedBy>
      <aixm:timeSlice>
        <aixm:AirportHeliportTimeSlice gml:id="uuid.${randomUUID}-OPERATIONAL-1">
          <gml:validTime>
            <gml:TimePeriod gml:id="uuid.tp.nasr_arp.24889">
              <gml:beginPosition>2012-09-01T00:00:00.000</gml:beginPosition>
              <gml:endPosition />
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:featureLifetime>
            <gml:TimePeriod gml:id="TimePeriod1">
              <gml:beginPosition>2012-09-01T00:00:00.000</gml:beginPosition>
              <gml:endPosition />
            </gml:TimePeriod>
          </aixm:featureLifetime>
          <aixm:designator>${designator}</aixm:designator>
          <aixm:name>${name}</aixm:name>
          <aixm:type>AH</aixm:type>
          <aixm:servedCity>
            <aixm:City gml:id="urn-x.ows7.snowflake.city.nasr_arp.24889">
              <aixm:name>${city}</aixm:name>
            </aixm:City>
          </aixm:servedCity>
          <aixm:ARP>
            <aixm:ElevatedPoint gml:id="uuid.geom.nasr_arp.24889"
              srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:pos>${coordinate}</gml:pos>
            </aixm:ElevatedPoint>
          </aixm:ARP>
          <aixm:availability>
            <aixm:AirportHeliportAvailability
              gml:id="uuid.avail.nasr_arp.24889">
              <aixm:operationalStatus>OPERATIONAL</aixm:operationalStatus>
              <aixm:usage>
                <aixm:AirportHeliportUsage gml:id="uuid.usage.nasr_arp.24889">
                  <aixm:selection>
                    <aixm:ConditionCombination gml:id="uuid.concom.nasr_arp.24889">
                      <aixm:aircraft>
                        <aixm:AircraftCharacteristic
                          gml:id="uuid.concom.nasr_arp.24889_ac">
                          <aixm:type>LANDPLANE</aixm:type>
                        </aixm:AircraftCharacteristic>
                      </aixm:aircraft>
                    </aixm:ConditionCombination>
                  </aixm:selection>
                </aixm:AirportHeliportUsage>
              </aixm:usage>
            </aixm:AirportHeliportAvailability>
          </aixm:availability>
        </aixm:AirportHeliportTimeSlice>
      </aixm:timeSlice>
    </aixm:AirportHeliport>
  </wfs:Insert>
</wfs:Transaction>
```

Permdelta Correction:

```
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WFS"
  version="2.0.0">
  <wfs:Insert>
    <aixm:AirportHeliport gml:id="uuid.${randomUUID}">
      <gml:identifier codeSpace="http://www.opengis.net/ows9/aviation/performanceStudy">
${randomUUID}</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirportHeliportTimeSlice gml:id="uuid.${randomUUID}-OPERATIONAL-2">
          <gml:validTime>
            <gml:TimePeriod gml:id="uuid.tp.nasr_arp.24890">
              <gml:beginPosition>2020-01-01T00:00:00.000</gml:beginPosition>
              <gml:endPosition />
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          <aixm:sequenceNumber>2</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:availability>
            <aixm:AirportHeliportAvailability
              gml:id="uuid.avail.nasr_arp.24890">
              <aixm:operationalStatus>OPERATIONAL</aixm:operationalStatus>
            </aixm:AirportHeliportAvailability>
          </aixm:availability>
        </aixm:AirportHeliportTimeSlice>
      </aixm:timeSlice>
    </aixm:AirportHeliport>
  </wfs:Insert>
</wfs:Transaction>
```

Permdelta Correction Cancellation:

```
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WFS"
  version="2.0.0">
  <wfs:Insert>
    <aixm:AirportHeliport gml:id="uuid.${randomUUID}">
      <gml:identifier codeSpace="http://www.opengis.net/ows9/aviation/performanceStudy">
${randomUUID}</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirportHeliportTimeSlice gml:id="uuid.${randomUUID}-${STATUS}-
${SEQUENCE_NUMBER}">
          <gml:validTime nilReason="inapplicable" />
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          <aixm:sequenceNumber>2</aixm:sequenceNumber>
          <aixm:correctionNumber>1</aixm:correctionNumber>
        </aixm:AirportHeliportTimeSlice>
      </aixm:timeSlice>
    </aixm:AirportHeliport>
  </wfs:Insert>
</wfs:Transaction>
```

Basline/Permdelta Cancellation:

```
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WFS"
  version="2.0.0">
  <wfs:Insert>
    <aixm:AirportHeliport gml:id="uuid.${randomUUID}">
      <gml:identifier codeSpace="http://www.opengis.net/ows9/aviation/performanceStudy">
${randomUUID}</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirportHeliportTimeSlice gml:id="uuid.${randomUUID}-${STATUS}-
${SEQUENCE_NUMBER}">
          <gml:validTime nilReason="inapplicable" />
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          <aixm:sequenceNumber>2</aixm:sequenceNumber>
          <aixm:correctionNumber>1</aixm:correctionNumber>
        </aixm:AirportHeliportTimeSlice>
      </aixm:timeSlice>
    </aixm:AirportHeliport>
  </wfs:Insert>
</wfs:Transaction>
```

A.2.2 Insert Tempdelta

Tempdelta:

```
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WFS"
  version="2.0.0">
  <wfs:Insert>
    <aixm:AirportHeliport gml:id="uuid.${existingUUID}">
      <gml:identifier codeSpace="http://www.opengis.net/ows9/aviation/performanceStudy">
${existingUUID}</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirportHeliportTimeSlice gml:id="uuid.${existingUUID}-CLOSED-1">
          <gml:validTime>
            <gml:TimePeriod gml:id="uuid.tp.nasr_arp.24890">
              <gml:beginPosition>2020-01-01T00:00:00.000</gml:beginPosition>
              <gml:endPosition />
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>TEMPDELTA</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>0</aixm:correctionNumber>
          <aixm:availability>
            <aixm:AirportHeliportAvailability
              gml:id="uuid.avail.nasr_arp.24890">
              <aixm:operationalStatus>CLOSED</aixm:operationalStatus>
            </aixm:AirportHeliportAvailability>
          </aixm:availability>
        </aixm:AirportHeliportTimeSlice>
      </aixm:timeSlice>
    </aixm:AirportHeliport>
  </wfs:Insert>
</wfs:Transaction>
```

Tempdelta Cancellation:

```
<wfs:Transaction xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WFS"
  version="2.0.0">
  <wfs:Insert>
    <aixm:AirportHeliport gml:id="uuid.${existingUUID}">
      <gml:identifier codeSpace="http://www.opengis.net/ows9/aviation/performanceStudy">
${existingUUID}</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirportHeliportTimeSlice gml:id="uuid.${existingUUID}-CLOSED-1">
          <gml:validTime nilReason="inapplicable" />
          <aixm:interpretation>TEMPDELTA</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:correctionNumber>1</aixm:correctionNumber>
        </aixm:AirportHeliportTimeSlice>
      </aixm:timeSlice>
    </aixm:AirportHeliport>
  </wfs:Insert>
</wfs:Transaction>
```

A.3 ES Requests

A.3.1 Subscriptions

XPath Subscription:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsnt="http://docs.oasis-
open.org/wsn/b-2">
  <soap:Header>
    <wsa:To>${es_host}</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeRequest</wsa:Action>
    <wsa:MessageID>uuid:4e595160-185a-9b3c-3eb6-592c7c5b0c7a</wsa:MessageID>
    <wsa:From>
      <wsa:Address>http://www.w3.org/2005/08/addressing/role/anonymous</wsa:Address>
    </wsa:From>
  </soap:Header>
  <soap:Body>
    <wsnt:Subscribe>
      <wsnt:ConsumerReference>
        <wsa:Address>${consumer}</wsa:Address>
      </wsnt:ConsumerReference>
      <wsnt:Filter>
        <wsnt:MessageContent Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116"
xmlns:aixm="http://www.aixm.aero/schema/5.1" xmlns:gml="http://www.opengis.net/gml/3.2">
            //aixm:RunwayDirection//gml:identifier[text()="runway-1"]
        <wsnt:MessageContent>
      </wsnt:Filter>
    </wsnt:Subscribe>
  </soap:Body>
</soap:Envelope>
```

Spatial Filter subscription:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsnt="http://docs.oasis-
open.org/wsn/b-2">
```

```
  <soap:Header>
    <wsa:To>${es_host}</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/SubscribeRequest</wsa:Action>
    <wsa:MessageID>uuid:4e595160-185a-9b3c-3eb6-592c7c5b0c7a</wsa:MessageID>
    <wsa:From>
      <wsa:Address>http://www.w3.org/2005/08/addressing/role/anonymous</wsa:Address>
    </wsa:From>
  </soap:Header>
  <soap:Body>
    <wsnt:Subscribe>
      <wsnt:ConsumerReference>
        <wsa:Address>${consumer}</wsa:Address>
      </wsnt:ConsumerReference>
      <wsnt:Filter>
        <wsnt:MessageContent xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:wsnt="http://docs.oasis-open.org/wsn/b-
2" Dialect="http://www.opengis.net/ses/filter/level2">
          <fes:Filter xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:swe="http://www.opengis.net/swe/1.0.1">
            <fes:Within>
              <fes:ValueReference>wfs-aixm:extentOf(.)</fes:ValueReference>
              <fes:Literal>
                <gml:Polygon gml:id="aoi_01">
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList>50.130536 7.083609 53.130536 7.083609 53.130536
7.983609 50.130536 7.983609</gml:posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Polygon>
              </fes:Literal>
            </fes:Within>
          </fes:Filter>
        </wsnt:MessageContent>
      </wsnt:Filter>
    </wsnt:Subscribe>
  </soap:Body>
</soap:Envelope>
```

A.3.2 Notifications

Runway Closure DNOTAM Event:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsnt="http://docs.oasis-
open.org/wsn/b-2">
  <soap:Header>
    <wsa:To>${es_host}</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/wsn/bw-
2/NotificationConsumer/Notify</wsa:Action>
    <wsa:MessageID>uuid:1b4d3025-f80a-a5b6-aa37-864c47fa1a7e</wsa:MessageID>
    <wsa:From>
      <wsa:Address>http://www.w3.org/2005/08/addressing/role/anonymous</wsa:Address>
    </wsa:From>
  </soap:Header>
  <soap:Body>
    <wsnt:Notify>
      <wsnt:NotificationMessage>
        <wsnt:Message>
          <message:AIXMBasicMessage gml:id="gmlid1"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:event="http://www.aixm.aero/schema/5.1/event"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:message="http://www.aixm.aero/schema/5.1/message"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.aixm.aero/schema/5.1/message
http://www.aixm.aero/gallery/content/public/schema/5.1/message/AIXM_BasicMessage.xsd
http://www.aixm.aero/schema/5.1/event
http://www.aixm.aero/gallery/content/public/schema/5.1/event/Event_Features.xsd">
            <message:hasMember>
              <event:Event gml:id="uuid.ff3ab666-d8d0-428c-8304-5922ee4636b5z2a46017">
                <gml:identifier codeSpace="urn:uuid:">ff3ab666-d8d0-428c-8304-
5922ee4636b5</gml:identifier>
                <event:timeSlice>
                  <event:EventTimeSlice gml:id="NOTAM_4_09_1">
                    <gml:validTime>
                      <gml:TimePeriod>
                        <gml:beginPosition>${validTimeStartIso}</gml:beginPosition>
                        <gml:endPosition>${validTimeEndIso}</gml:endPosition>
                      </gml:TimePeriod>
                    </gml:validTime>
                    <aixm:interpretation>BASELINE</aixm:interpretation>
                    <aixm:sequenceNumber>1</aixm:sequenceNumber>
                    <aixm:featureLifetime>
                      <gml:TimePeriod>
                        <gml:beginPosition>${validTimeStartIso}</gml:beginPosition>
                        <gml:endPosition>${validTimeEndIso}</gml:endPosition>
                      </gml:TimePeriod>
                    </aixm:featureLifetime>
                    <event:name>NOTAM_4_09_1</event:name>
                    <event:encoding>DIGITAL</event:encoding>
                    <event:scenario>RWY.CLS</event:scenario>
                    <event:textNOTAM>
                      <event:NOTAM>
                        <event:series>A</event:series>
                        <event:number>5</event:number>
                        <event:year>2012</event:year>
                        <event:type>N</event:type>
                        <event:issued>2012-11-21T11:03:00.000Z</event:issued>
                        <event:affectedFIR>EADD</event:affectedFIR>
                        <event:selectionCode>QMRLC</event:selectionCode>
                        <event:traffic>IV</event:traffic>
                        <event:purpose>NBO</event:purpose>
                        <event:scope>A</event:scope>
                        <event:coordinates>5222N03157W</event:coordinates>
                        <event:radius>999</event:radius>
                        <event:location>EADD</event:location>
                        <event:effectiveStart>1211222200</event:effectiveStart>
                        <event:effectiveEnd>1211230400</event:effectiveEnd>
                        <event:text>RWY 09L/27R CLOSED.</event:text>
                        <event:lowerLimit>000</event:lowerLimit>
                        <event:upperLimit>999</event:upperLimit>
                        <event:publisherNOF>#urn.uuid.c225ae5c-540f-4a48-8867-
809b393b2407</event:publisherNOF>
                      </event:NOTAM>
                    </event:textNOTAM>
                  </event:EventTimeSlice>
                </event:timeSlice>
              </event:Event>
            </message:hasMember>
            <message:hasMember>
              <aixm:RunwayDirection gml:id="urn.uuid.c8455a6b-9319-4bb7-b797-
08e644342d64z2a46017">
                <gml:identifier codeSpace="urn:uuid:">runway-1</gml:identifier>
                <aixm:timeSlice>
                  <aixm:RunwayDirectionTimeSlice>
                    <gml:validTime>
                      <gml:TimePeriod>
                        <gml:beginPosition>${validTimeStartIso}</gml:beginPosition>
                        <gml:endPosition>${validTimeEndIso}</gml:endPosition>
                      </gml:TimePeriod>
                    </gml:validTime>
                    <aixm:interpretation>TEMPDELTA</aixm:interpretation>
                    <aixm:sequenceNumber>${sequenceNumber}</aixm:sequenceNumber>
                    <aixm:featureLifetime>
                      <gml:TimePeriod>
                        <gml:endPosition>unknown</gml:endPosition>
                      </gml:TimePeriod>
                    </aixm:featureLifetime>
                    <aixm:availability>
                      <aixm:ManoeuvringAreaAvailability>
                        <aixm:operationalStatus>${status}</aixm:operationalStatus>
                      </aixm:ManoeuvringAreaAvailability>
                    </aixm:availability>
                    <aixm:extension>
                      <event:RunwayDirectionExtension gml:id="ex01">
                        <event:theEvent xlink:href="#uuid.ff3ab666-d8d0-428c-8304-
5922ee4636b5z2a46017"/>
                      </event:RunwayDirectionExtension>
                    </aixm:extension>
                  </aixm:RunwayDirectionTimeSlice>
                </aixm:timeSlice>
              </aixm:RunwayDirection>
            </message:hasMember>
          </message:AIXMBasicMessage>
        </wsnt:Message>
      </wsnt:NotificationMessage>
    </wsnt:Notify>
  </soap:Body>
</soap:Envelope>
```

Annex B – Performance Tool User Tutorial

Prerequisites

The Performance Tool is a browser-based client which requires the use of JavaScript and AJAX (Asynchronous JavaScript and XML) technology. If not already done please enable these functions within your browser.

Installation

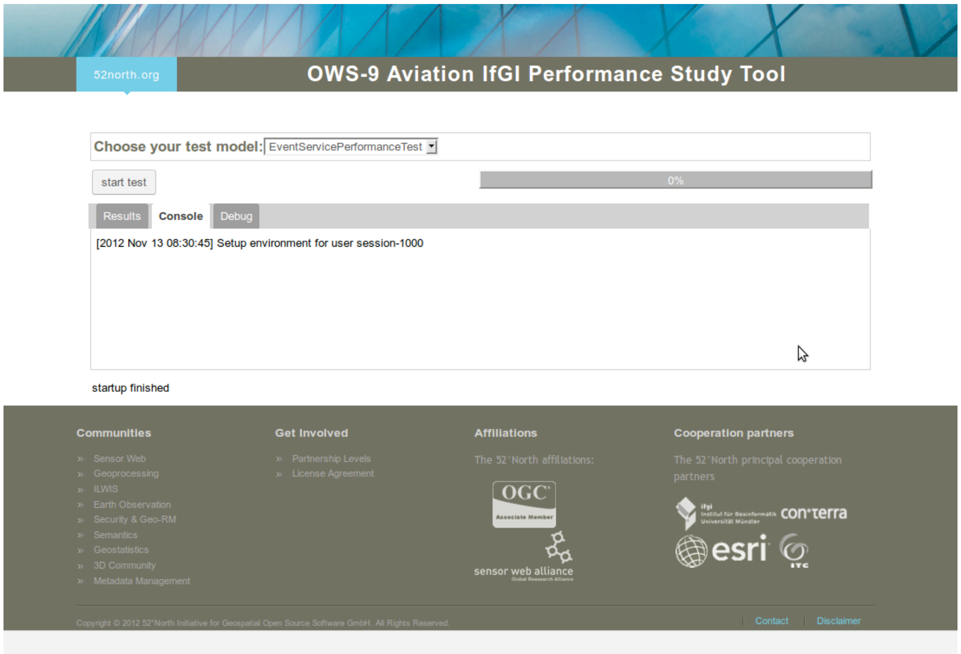
Installing the Tool is straightforward if you have a Servlet Container supporting the Servlet API 2.x running. If not, please install one by following its setup guidance (e.g. Apache Tomcat 6/7).

The provided .war file („OWS9-PerformanceTool.war“) must be copied to the „webapp“ folder of the Servlet Container or deployed according to the Container's specific installation process (e.g. through a web interface).

After successful deploying the .war file, the tool is accessible via a web browser at

Fehler! Hyperlink-Referenz ungültig..

It should look like illustrated in the following figure.



OWS-9 Aviation IfGI Performance Study Tool

Choose your test model: EventServicePerformanceTest

```
[2012 Nov 13 08:30:45] Setup environment for user session-1000
```

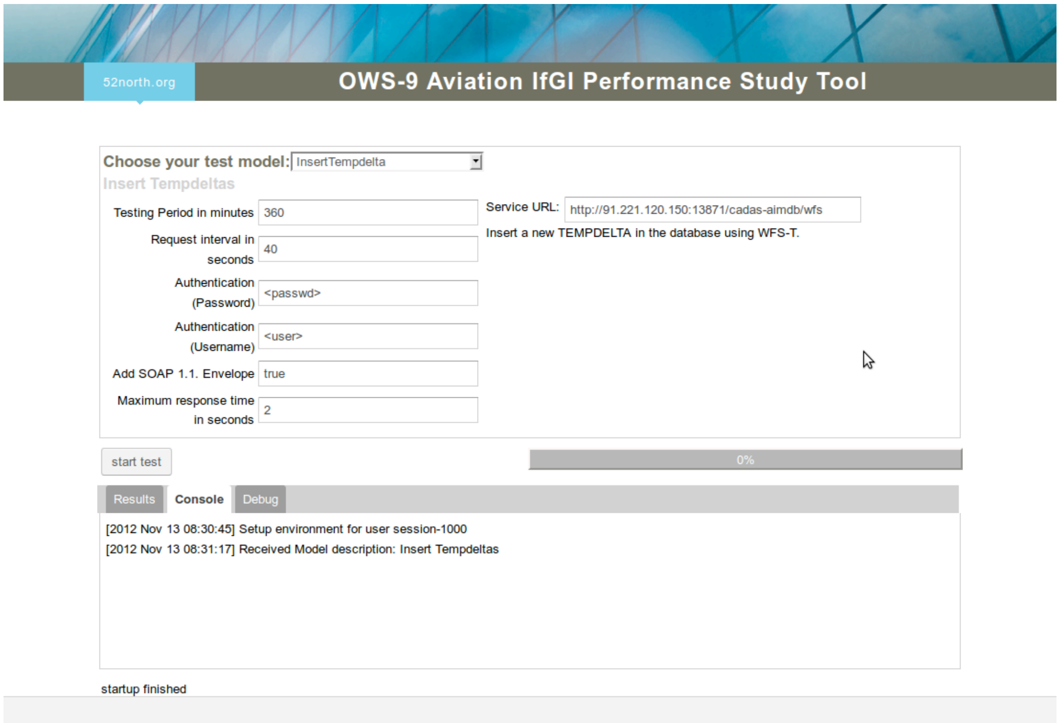
startup finished

Note the session id („session-1000“; this may vary depending on the previously created sessions). With this id the session can be accessed as a permalink after closing the actual *HTTP session* using the following URL:

Fehler! Hyperlink-Referenz ungültig..

Model Execution

As a first step a model must be selected from the dropdown menu on top of the UI. The tool is communicating the successful creation of the model via the Console area (see following figure).



Choose your test model: InsertTempdelta

Insert Tempdeltas

Testing Period in minutes	360
Request interval in seconds	40
Authentication (Password)	<passwd>
Authentication (Username)	<user>
Add SOAP 1.1. Envelope	true
Maximum response time in seconds	2

Service URL: http://91.221.120.150:13871/cadas-aimdb/wfs

Insert a new TEMPDELTA in the database using WFS-T.

```
[2012 Nov 13 08:30:45] Setup environment for user session-1000
[2012 Nov 13 08:31:17] Received Model description: Insert Tempdeltas
```

startup finished

Specifying Parameters

Several parameters must be defined for a model instance. This depends on what should be tested and varies across the available models. The following sections provide guidelines on parameters for Web Feature Services (WFS) and Event Services (ES) which have been found suitable.

General Parameters

The „**Test Period in minutes**“ can be adjusted to fit certain requirements. For OWS-9 the stability and reliability of a service is to be tested. Thus a long period in the range of 360 minutes to 1440 minutes (6 – 24 hours) should always be defined.

The „**Authentication (...)**“ parameters apply for a service secured with HTTP Basic Authentication (Comsoft's WFS for OWS-9) and must be provided for such a service.

WFS Models

By setting „**Add SOAP 1.1. Envelope**“ to `true`, the requests will be wrapped with SOAP Envelopes. Within OWS-9 this is needed for Comsoft's WFS. For Snowflake's WFS it is optional.

„**Request interval in seconds**“ defines the time the tool will wait after one execution (e.g. GetFeature request) has been finished.

InsertTempdelta / InsertBaselinePermdelta

As the transactional requests are generated using some randomization, no special parameters need to be defined. All AIXM Feature TimeSlice Commissionings are removed by TimeSlice cancellation after a test execution.

RetrieveTimeslicesForAirport

Parameter	***Descriptions***
gml:identifier	***The gml:identifier of an Airport available at the service.***
valid time start/end	***Valid time start and end positions. The requests are constructed using the fes:AnyInteracts operation with a gml:validTime/gml:TimePeriod operand.***

RetrieveByDesignator

Parameter	***Descriptions***
Designator portion (with wildcard %)	***The aixm:designator portion. An Airport with designator "LFGR" will match the wildcard "LF%". Requests are constructed using fes:PropertyIsLike operations.***

ES Models

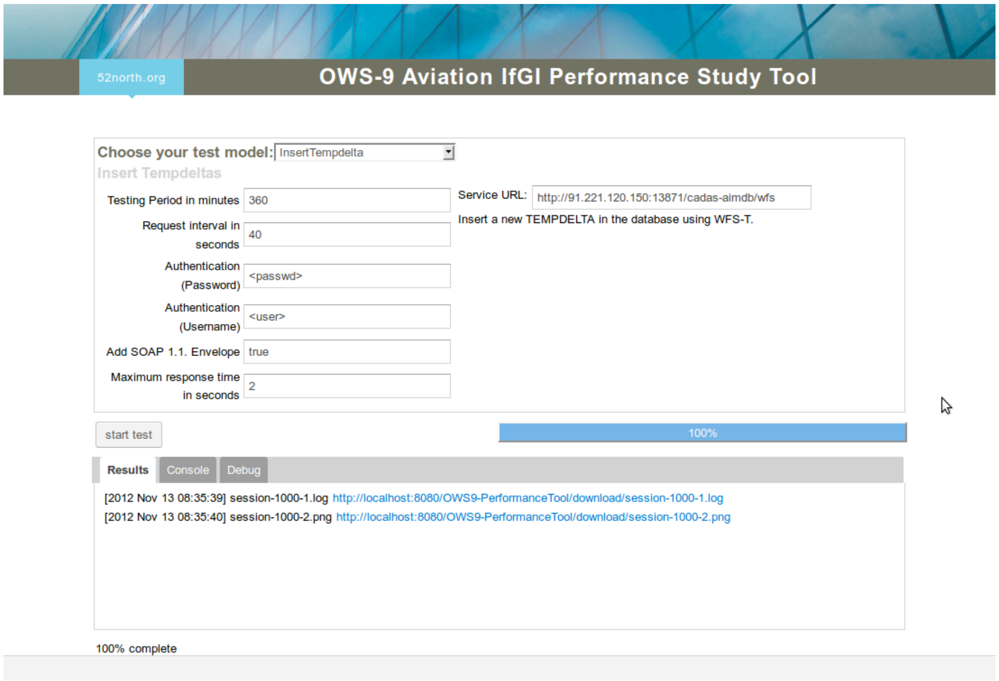
The two available ES models are using the same execution environment. The UI provides two selections to provide some default methods which are suitable to address the requirements of the different models following the OWS-9 RFQ.

EventServicePerformanceTest/EventServiceEnduranceTest

Parameter	***Descriptions***
Publishing Interval in seconds	*Defines the period the tool waits between two DNOTAM publishing cycles.*
DNOTAM Count	*Specifies the number of DNOTAMs send to the ES within one publishing cycle.*
Maximum time to receive Notifications in seconds	*As the tool is capable of receiving matched DNOTAMs this parameter defines the maximum response time. When a message exceeds this value it is treated as an "Overdue Notification".*
Number of Subscribers	*This defines the number of simultaneous subscriptions placed at the ES. A high value simulates a real-world situation for an ES instance.*
Service URL	*The service URL must specify the endpoint to where Subscriptions and Notifications are posted (the tool does not harvest WSDLs currently).*

Downloading Results

After successful execution of a test model the tool provides several result files for download (graphs, CSV, raw logs) within the „Results" area (see following figure). Please download these files and provide them back via Email to m.rieke@uni-muenster.de.



OWS-9 Aviation IfGI Performance Study Tool

Choose your test model: InsertTempdelta

Insert Tempdeltas

Testing Period in minutes: 360

Request interval in seconds: 40

Authentication (Password): <passwd>

Authentication (Username): <user>

Add SOAP 1.1. Envelope: true

Maximum response time in seconds: 2

Service URL: http://91.221.120.150:13871/cadas-aimdb/wfs

Insert a new TEMPDELTA in the database using WFS-T.

```
[2012 Nov 13 08:35:39] session-1000-1.log http://localhost:8080/OWS9-PerformanceTool/download/session-1000-1.log
[2012 Nov 13 08:35:40] session-1000-2.png http://localhost:8080/OWS9-PerformanceTool/download/session-1000-2.png
```

100% complete

Additional Functions

The tool provides access to application logs via a web browser:

Fehler! Hyperlink-Referenz ungültig. – log file of the current day (if available)

Fehler! Hyperlink-Referenz ungültig. – log file of the current minus 1 day (if available)

If some misbehavior is observed all model executions can be resetted and removed by accessing the following URL:

Fehler! Hyperlink-Referenz ungültig..

Bibliography

[1] EUROCONTROL & FAA (2010): AIXM 5 Temporality Model. Available online at http://www.aixm.aero/gallery/content/public/AIXM51/AIXM%20Temporality%201.0.pdf