Open Geospatial Consortium

Approval Date: 2013-01-18

Pending Date: 2013-01-03

Publication Date: 2013-02-06

Reference number of this document: OGC **12-151**

External Identifier for this document: http://www.opengis.net/def/doc-type/per/ows9-aviation-portrayal

Category: Public Engineering Report

Editor: Daniel Balog, Roger Brackin, Robin Houtmeyers

OGC® OWS-9 Aviation Portrayal Engineering Report

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 provides an overview of the portrayal work within the OWS-9 Aviation thread. Using open standards, a web services architecture was designed and prototyped to enable the retrieval of static airport maps in support of an ePIB. An ePIB, or Digitally Enhanced Pre-Flight Information Bulletin, provides the pilot with an easy-to-interpret representation of any relevant aeronautical and meteorological events that are likely to affect the flight, expressed as Digital NOTAMs. The static airport maps are an important part of an ePIB and should provide a graphical representation of the status of departure and arrival airports, showing only NOTAMs relevant to the particular context and represented geographically so that the effect of the NOTAM is clear. This approach avoids the pilot scanning through pages of textual description for potentially relevant NOTAMs, reducing the workload and the risk of missing a critical piece of information.

Keywords

ogcdoc, ows9, ows-9, aviation, portrayal, notam, epib, fps, wps, aixm, sld

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

Tables

Page

OGC® OWS-9 Aviation Portrayal Engineering Report

Executive Summary

Introduction

This document provides an overview of the portrayal work within the OWS-9 Aviation thread. Using open standards, a web services architecture was designed and prototyped to enable the retrieval of static airport maps in support of an ePIB. An ePIB, or Digitally Enhanced Pre-Flight Information Bulletin, provides the pilot with an easy-to-interpret representation of any relevant aeronautical and meteorological events that are likely to affect the flight, expressed as Digital NOTAMs. The static airport maps are an important part of an ePIB and should provide a graphical representation of the status of departure and arrival airports, showing only NOTAMs relevant to the particular context and represented geographically so that the effect of the NOTAM is clear. This approach avoids the pilot scanning through pages of textual description for potentially relevant NOTAMs, reducing the workload and the risk of missing a critical piece of information.

Background information

With an ever increasing amount of today's air traffic, providing pilots with accurate and up-to-date weather and aeronautical information is essential to ensure proper safety and efficiency. One of the planned improvements in this area includes the transition of the traditional text NOTAM message with its digital counterpart, i.e. the Digital NOTAM.

Within SESAR project 13.2.2, an investigation is being undertaken to compare the two NOTAM approaches of paper (PIB) and dynamic (ePIB) for a number of uses cases. One such use case is the provision of a pre-flight information bulletin (PIB) for the pilot, which includes the NOTAM updates relevant for his/her flight. A traditional PIB typically lists the text NOTAMs, resulting in large documents (10 - 50 pages for an internal European flight) that can be difficult to interpret. A digitally enhanced PIB (ePIB) on the other hand presents all relevant Digital NOTAMs by easy-to-interpret graphical representations / maps, providing the pilot with an immediate overview of the situation during a certain flight phase.

The ePIB generation workflow is complex and not part of OWS-9 (the selection of which NOTAMs to a given aircraft, pilot type with a range of other constraints is being developed as part of the SESAR project. But there is a specific requirement, once the set of NOTAMs selected for a specific plate/map have been determined, to render the NOTAMs against a relevant backdrop.

The focus of the portrayal work in OWS-9 is to design and prototype a web services architecture to generate the graphical maps for use within an ePIB. The goal is to have a web service that can render a list of Digital NOTAMs, submitted by the user, graphically on a map backdrop; the envisioned use cases are the generation of static airport maps for the departure and arrival airports as well as alternate airports defined for the flight plan.

The ePIB workflow is also part of an approach which allows multiple ways to access information from one infrastructure design, demonstrating that an integrated model using open standards is highly beneficial. The ePIB element covers the delivery of pre-flight information, which is produced in a more efficient way, more relevant to the specific user and more up-to-date. In order to demonstrate this, the ePIB concept is demonstrated as part of the OWS-9 scenario as the pre-flight assessment. The same information is then used in later phases of the scenario but in a dynamic update context.

The work under OWS-9 is focused around the static airport map requirement within the ePIB. In order to demonstrate the concept, a limited browser based ePIB generation interface was developed which allows the infrastructure to be demonstrated effectively within the scenario.

Concise analysis

The ePIB Graphical Map requirement driven out of the SESAR ePIB approach suggests a very specific interface model. Given that this was not easily rearranged to fit in with specific OGC interfaces the conclusion was to use standard OGC services but coordinate them with a custom service. The OGC Web Processing service has this role and so was used within the ePIB work together with the other underpinning OGC services. This follows the SOA paradigm: a set of independent, reusable service components identified to cover the ePIB static airport map generation functionality:

- An OGC Web Feature Service (WFS), capable of delivering any additional features (runways, taxiways, aprons …) needed to generate the airport map.
- An OGC Feature Portrayal Service (FPS), capable of rendering airport layout features and Digital NOTAM events.
- An OGC Web Registry (WRS), capable of delivering information about WFS data sources and styling information.
- An OGC Web Processing Service (WPS), acting as main component to interact with the ePIB user and the WFS, FPS and WPS components to be able to generate the static airport map.

In this architecture, the WFS, FPS and WRS components did not need any new functionality other than the capabilities defined by their latest standards. All custom business logic, such as consolidating all relevant feature data and styling information, is offloaded to a WPS.

Concise lessons learned / main conclusions

A proof-of-concept has been developed and tested during OWS-9 that puts the designed architecture into practice and successfully demonstrates its feasibility with a real-world scenario. The use of a WPS allows to isolate any custom business logic and to integrate the use of other standards-based components (WFS, FPS, WRS …) into a flexible architecture. This flexibility and the split-up of responsibilities between multiple service components also bring a challenge to get an optimal performance. For instance, the offloading of the rendering task to an FPS requires that all feature data is submitted to the FPS; as the AIXM format is verbose, this can have an impact on the required bandwidth, and data reducing techniques might be needed (e.g., filtering of the AIXM properties to the bare minimum, compression, …).

Recommendations

This document is a first step towards the design of a web services architecture to support the creation of static airport maps for use within an ePIB. It is recommended to further investigate and improve the architecture on a number of aspects:

- Generation of maps for other phases of the flight, next to the static airports maps for departure and arrival airports.
- Incorporation of weather data events.
- Performance / scalability improvements or best practices.
- Alternative map background approaches (imagery, raster maps etc)
- Review of alternative architecture configurations.
- Examination of the relevance of OWS Context

.

OGC® OWS-9 Aviation Portrayal Engineering Report

1 Introduction

1.1 Scope

This OGC® document describes the Portrayal elements of the Aviation Thread of OWS-9. The specific requirements of OWS-9 Aviation Portrayal are focused around delivering map graphics for use within a Digitally Enhanced Pre-Flight Information Bulletin or ePIB. The ePIB concept is being developed within SESAR and tends to present all Digital NOTAM data relevant for a flight by easy-to-interpret graphical representations / maps. This document includes:

- A description of the requirements for the ePIB map generation capability together with use cases.
- The design of a web services architecture to support the ePIB map generation capability.
- An overview of the implemented components and workflows followed by a short description of each component.
- Documentation of the issues, lessons learned as well as accomplishments.

1.2 Document contributor contact points

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

Name	Organization
Daniel Balog	Luciad
Roger Brackin	Envitia
Robin Houtmeyers	Luciad

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description

2012-07-24	draft	Roger Brackin		Initial draft
2012-11-28	pre-final	Robin Houtmeyers, Roger Brackin	Updates for Final Results, Annex A added.	Pre-final update
2012-01-03	Final	Roger Brackin	Addition of Performance issues paragraph	Final update

1.4 Future work

Improvements in this document are desirable to the following topics.

- **Test additional ePIB use cases.** The work during OWS-9 focuses on the generation of static airport maps for the destination and arrival airports. Other phases of the flight might include other types of NOTAMs and reveal new requirements to determine an appropriate map rendering.
- **Include weather events.** The current investigation focuses on aeronautical events expressed as Digital NOTAMs. Weather events are equally important though, and should preferably be included in future ePIB research.
- **Performance / scalability study.** The existing architecture has been designed to functionally cover the ePIB map generation use case. Additional investigation should be done with respect to performance and scalability.
- **Architecture.** Review alternative architecture configurations (balance of functionality in WPS versus other services) to maximize agility.
- **More flexible background map options.** Using imagery or raster mapping as alternatives to vector features.
- **OWS Context.** Potential use of OWS Context to describe the contents of a static map request.

1.5 Forward

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.

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.

2.1 OWS-9 Engineering Reports

1. [OGC 12-TBD] OWS-9 Aviation Architecture - Engineering Report

2.2 Other OGC Documents

1. [OGC 05-007r7] Web Processing Service
2. [OGC 06-042] OpenGIS Web Map Service (WMS) Implementation Specification
3. [OGC 05-078r4] OpenGIS Styled Layer Descriptor Profile of the Web Map Service Implementation Specification
4. [OGC-05-077r4] OpenGIS Symbology Encoding Implementation Specification

2.3 Aviation Documents

1. Digital NOTAM Event Specification, ed. 1.0 (Proposed Release), available at

   http://www.aixm.aero/public/standard_page/digital_notam_specifications.html

2. AIXM - Temporality Model v1.0, available at

   http://www.aixm.aero/public/standard_page/download.html

3. AIXM - AIXM Application Schema Generation, available at

   http://www.aixm.aero/public/standard_page/download.html

4. AIXM - UML to XML Schema Mapping v1.1, online at

   http://www.aixm.aero/public/standard_page/download.html

2.4 Other Documents

1. SWIM Service Compliance Requirements - http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/atc_comms_services/swim/documentation/media/compliancy/SWIM%20Service%20Compliance%20Requirements.pdf
2. Systems Rules Model (SRM) – see OWS-9 RFQ Annex B, Section 10 Appendix

3. The Unified Code for Units of Measure (UCUM), online at

   http://aurora.regenstrief.org/~ucum/ucum.html

4. SESAR Document: 13.02.02.D01_OSED_-_Appendix_-Example_of_Digitally_Enhanced_PIB.pdf

3 Abbreviated terms

AIXM	Aeronautical Information Exchange Model
AIXM-TM	AIXM Temporality Model
CRS	Coordinate Reference System
CSW	Catalog Service for the Web
DNES	Digital NOTAM Event Specification
DNOTAM	Digital NOTAM
ePIB	Digitally Enhanced Pre-Flight Information Bulletin
EFB	Electronic Flight Bag
WCS	Web Coverage Service
WFS	Web Feature Service
WFS-T	WFS-Transactional
WMS	Web Map Service
WPS	Web Processing Service
WXXM	Weather Information Exchange Model
XACML	Extensible Access Control Markup Language
XML	Extensible Markup Language
XPath	XML Path Language

4 Use Case and Requirements

With an ever increasing amount of today's air traffic, providing pilots with accurate and up-to-date weather and aeronautical information is essential to ensure proper safety and efficiency. One of the planned improvements in this area includes the transition of the traditional text NOTAM message with its digital counterpart, i.e. the Digital NOTAM.

Within SESAR project 13.2.2, an investigation is being done to compare the two NOTAM approaches for a number of uses cases. One such use case is the provision of a pre-flight information bulletin (PIB) for the pilot, which includes the NOTAM updates relevant for his/her flight.

A traditional PIB is mainly in the form of a "bulletin" given to the pilot, listing all NOTAM messages that might be relevant for the planned flight. The automatic NOTAM filtering capabilities are limited because the free text of the NOTAM, which contains the key information, is not suitable for querying and filtering. Additionally, the graphical representation of the NOTAM information is quite primitive, typically limited to a circle representing an "area of possible influence". Consequently, traditional PIBs are often large documents (10 – 50 pages for an internal European flight) that can be difficult to interpret with the attendant higher than necessary workload and risk of overlooking key NOTAMs (due to the volume of irrelevant information).

With Digital NOTAM, the briefing could be radically improved through more precise automated filtering and by graphically depicting the information. This leads to the concept of a Digitally Enhanced Pre-Flight Information Bulletin or ePIB: a representation of all Digital NOTAM events relevant for a flight by easy-to-interpret textual and graphical representations / maps. The ePIB approach avoids the pilot scanning through pages of textual description for potentially relevant NOTAMs, reducing the workload and the risk of missing a critical piece of information. The creation of an ePIB for a flight could take place directly on an Electronic Flight Bag (EFB) carried by pilots, using data provided by digital briefing applications on the ground and updated during the flight through data link communication.

The focus of the OWS-9 Aviation Portrayal work is delivering the use case for ePIB static airport map generation. The static airport map is an important part of an ePIB, presenting all relevant aeronautical and meteorological events for departure and arrival airports. OWS-9 investigates the degree to which an OGC web services architecture can deliver the capability to produce these static airport maps.

The ePIB static airport maps sub-system is built into the overall aviation scenario developed by the OWS-9 aviation thread as part of the pre-flight phase. The same infrastructure then delivers via the on-line EFB model with updates in-flight.

The architecture work under OWS-9 is focused around only the static airport map requirement within the ePIB, but in order to demonstrate the concept, a limited browser based ePIB generation interface was developed which allows the infrastructure to be demonstrated effectively within the scenario.

4.1 Operational Context

The operational goal is the generation of static airport maps, consisting of up-to-date airport information, via a web services infrastructure. The high-level architecture defined within SESAR is shown in Figure 1.

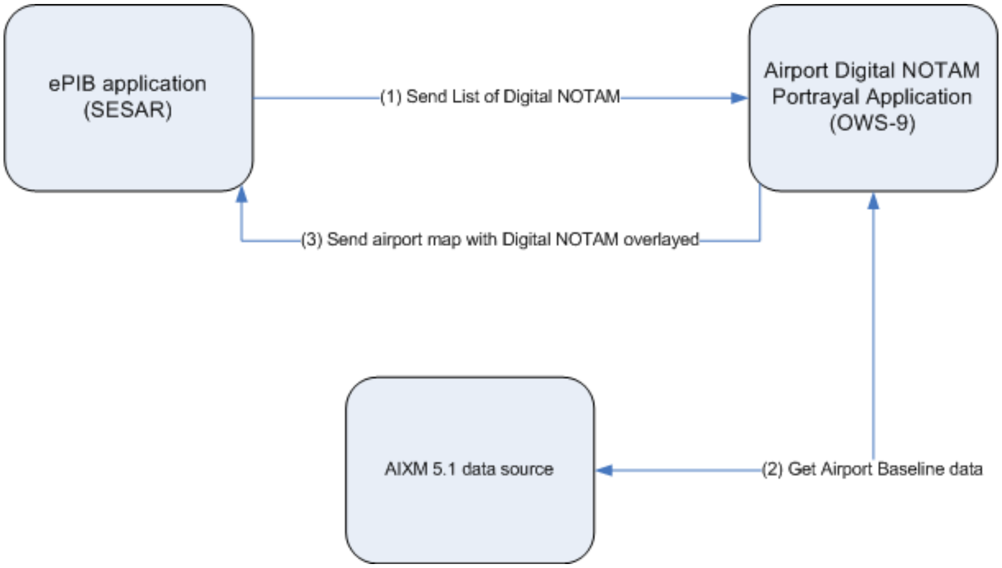

ePIB application
(SESAR)
(1) Send List of Digital NOTAM
Airport Digital NOTAM
Portrayal Application
(OWS-9)
(3) Send airport map with Digital NOTAM overlayed
AIXM 5.1 data source
(2) Get Airport Baseline data

Figure 1 – SESAR ePIB architecture

An “ePIB application” takes the responsibility of generating the ePIB document. To retrieve the static airport maps, it compiles a list of relevant Digital NOTAM events and sends them to a portrayal web service, called the “Airport Digital NOTAM Portrayal Application” in the figure. The list of Digital NOTAM events relate to a single airport.

The portrayal web service requests any additional information needed to generate the airport map from an AIXM 5.1 data source and produces an airport map as a raster image, ready to be embedded in the ePIB document. The ePIB application may make a number of requests for airport maps, as an ePIB will typically include an origin, destination and alternate airport for which information is required.

The research in OWS-9 focuses on the portrayal web service. The ePIB application itself is not considered; to illustrate the work, a client is developed to simulate the requests that an ePIB application might make.

A key benefit of an automated ePIB architecture based on OGC web services (and based on the SWIM platform) is that it is possible to generate an ePIB and produce updates in an automated way, ensuring up-to-date information including the highlighting of key changes to maximize the pilot’s situational awareness.

4.2 Operational Sequence

As input, an airport and a list of Digital NOTAMs are selected that will be displayed on an airport map. The necessary baseline data, as well as the styling should be fetched automatically. The overall description of the expected workflow as described by the sponsor is as follows:

1. The ePIB application (external to OWS-9) sends a request to the “portrayal” application , specifying an airport and a list of Digital NOTAMs that need to be displayed on an airport map. The Digital NOTAMs are represented according to the Digital NOTAM Event Specification, (based on the AIXM 5.1 format) and consist of event information and feature timeslices. It is the task of the ePIB application to identify which Digital NOTAMs need to be displayed on the airport map. For the purpose of this exercise, this is limited to scenarios from the Event Specification 1.0, with eventually some additional Events: - runway closures, taxiway closures, temporary obstacles, navaid unserviceable, displaced threshold, runway lights outage, etc.

2. The portrayal service analyses the request and retrieves any additional data necessary for creating the airport map from an AIXM 5.1 data source: runways, taxiways, aprons, buildings, etc. Satellite imagery as background is not used, as this is a bit too heavy for use within an ePIB and not considered needed to present the information.

3. The portrayal application overlays the information from the Digital NOTAM on the airport map and returns the image to the ePIB application. This could be a web service reply that contains a link to a WMS server where the image is available for retrieval.

5 OWS-9 Portrayal Architecture

5.1 Overview

The ePIB airport map generation use case requires a set of capabilities to query and consolidate all relevant data (i.e., Digital NOTAM events and baseline AIXM 5.1 feature data), to define the styling rules based on the type of data and to generate the resulting map. The OGC Feature Portrayal Service (FPS) was considered as a possible solution, but the SESAR requirement seems to lead to the need for a more specific service. The approach proposed is to use a multi-tier architecture laying a custom service on top of existing discovery (WRS), data provision (WFS) and portrayal (FPS) services. The OGC Web Processing Service (WPS) is proposed as the basis for this. This offers a good design pattern potentially relevant to other service requirements. The WPS still uses the core capabilities of existing services and the goal is to make the service as thin as possible, allowing the FPS, WFS and WRS to do the heavy lifting but provide the specific interface customized to the functional need.

5.2 Architecture Proposal

This section describes the proposed architecture to generate the static airport maps. The diagram below shows a high-level overview of the various components in the system and the workflow.

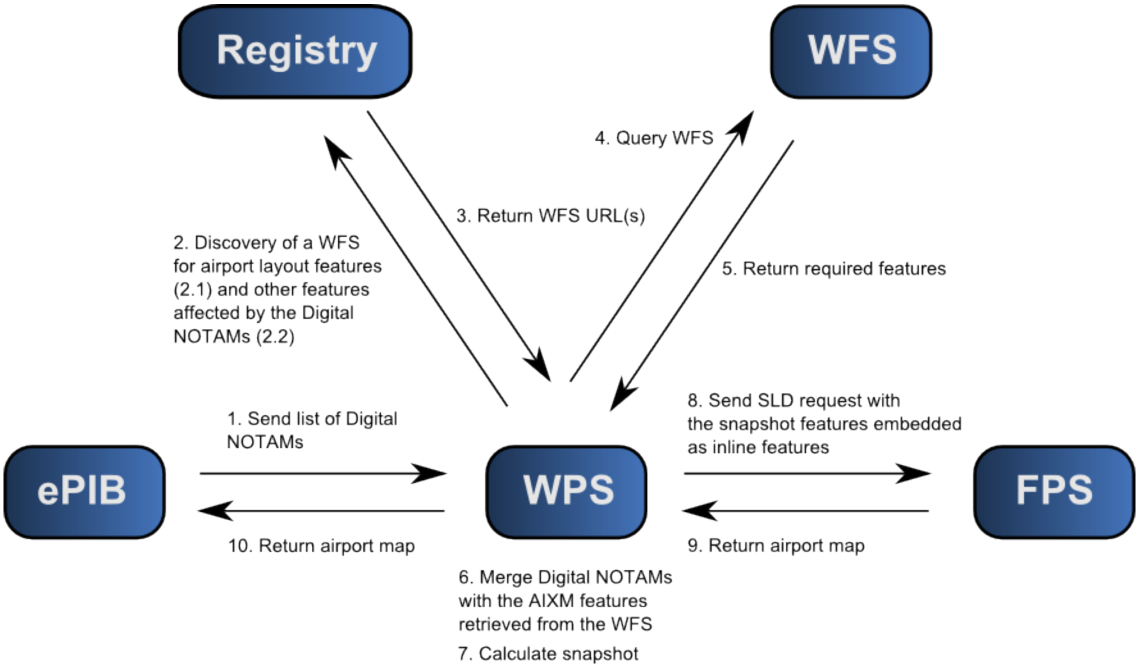

Registry
WFS
4. Query WFS
3. Return WFS URL(s)
2. Discovery of a WFS for airport layout features (2.1) and other features affected by the Digital NOTAMs (2.2)
5. Return required features
1. Send list of Digital NOTAMs
8. Send SLD request with the snapshot features embedded as inline features
ePIB
WPS
FPS
10. Return airport map
9. Return airport map
6. Merge Digital NOTAMs with the AIXM features retrieved from the WFS
7. Calculate snapshot

Figure 2 – Component and workflow overview

There are five components involved:

1. ePIB component (external to OWS-9): requests a static airport map, based on a an airport (GML) identifier and a set of relevant Digital NOTAM events
2. WPS component: main component to manage the static airport map request. It has two core responsibilities:
    a. Enrich the Digital NOTAM dataset with features from a WFS,
    b. Generate a map with the use of an FPS.
3. Registry Service component: enables discovery of a WFS that can deliver the necessary features

4. WFS component: provides AIXM 5.1 baseline feature data
5. FPS component: generates a map based on an SLD / SE request embedded with AIXM 5.1 data

The workflow is as follows:

1. The ePIB component sends the airport map request to the WPS, consisting of an airport (GML) identifier and a list of Digital NOTAMs. The Digital NOTAMs are encoded according to the Digital NOTAM Event Specification.
2. The WPS needs features to enrich the Digital NOTAM dataset, and relies on the Registry Service to discover a WFS (or multiple ones) that can deliver the necessary features. Discovery can be mainly based on the GML identifier of a feature and feature types. For instance, retrieve the WFS that can deliver the Navaid feature with this GML identifier, or retrieve the WFS that can deliver Runway features for the airport with a given GML identifier. The following features are considered necessary:
    1. Features relevant for the airport layout. All features related to the airport layout (i.e., runways, taxiways, aprons ...) are needed for the resulting airport map. For this, a WFS request is sent to query airport layout features (e.g., a hardcoded set of feature types) for the specified airport.
    2. Features impacted by the Digital NOTAMs. Each Digital NOTAM is analysed: if it concerns an airport layout feature (e.g., a TaxiwayElement, RunwayElement, VerticalStructure ...) that was already retrieved in 2.1, the corresponding feature representation on the map is updated / overlaid with the information coming from the Digital NOTAM; if it concerns a feature that is not yet part of the airport layout (e.g., the Digital NOTAM is about the unavailability of an ILS Navaid, which is not part of the standard airport layout), then this feature needs to be retrieved from a WFS and its graphical "temporary status" situation is overlaid on the map.
3. The Registry Service returns the URL(s) of the WFS components that can deliver the required features.
4. The WFS is contacted to retrieve the requested features.
5. The WFS returns the required features.
6. The WPS now has a set of AIXM 5.1 features retrieved from the WFS (BASELINE timeslices but potentially also TEMPDELTAs/PERMDELTAs) and a set of Digital NOTAM features (essentially TEMPDELTA timeslices). This data is merged to obtain one coherent AIXM 5.1 feature collection that serves as a basis for the airport map.

7. The AIXM 5.1 feature collection still contains multiple timeslices for each feature, valid for various time intervals. To avoid timeslice calculations in the FPS, the WPS calculates a snapshot for the AIXM 5.1 feature collection.

8. The WPS constructs an SLD request with a user layer embedded with the snapshot features and sends it to the FPS.

9. The FPS renders the snapshot features and returns the airport map as a bitmap.

10. The airport map (bitmap) is sent to the ePIB component.

In this design, the WPS is the central component to manage the airport map generation and contains all the business logic. An advantage is that the WRS, WFS and FPS components remain general and do not require any modification that is the specific for the ePIB setup.

5.3 Rationale for the WPS

A number of architectural approaches were considered but it was clear that a key requirement was to provide a static airport map interface to the ePIB component which minimized the complexity of the interaction. Eurocontrol had a well-defined requirement for this interface, involving passing a set of parameters including a list of NOTAMs, an airport, a geographic extent and the background features required.

While the original premise was that the Feature Portrayal Service (FPS) could provide the ePIB functionality, there were two significant issues.

- The nature of the interface requirements were not a good fit to the existing FPS model/call structure. The goal was an interface that operated at a highly parameterized level (with query details buried in the supporting service). For example if this was a WFS call it would typically be implemented as a stored query. However the FPS interface does not support this level of abstraction.

- The complexity of the queries to retrieve the data to be portrayed (given that there is the complexity of the time model to be dealt with) meant that even if the rules could be defined in the FPS interface (an interaction of the WFS query needed and the SLD Rule set) it is unlikely that any pre-existing FPS would deal with them.

The OGC WPS provides a blank canvas to implement custom services meeting specific needs and the ePIB seems to fit exactly this profile. Thus the WPS was chosen and a custom interface designed. But the goal was not to make the WPS component an amorphous mass, but to make it a broker interfacing to standard services (WFSs, WMSs, FPSs, Registries …).

Having decided on this model, it was also concluded that a good approach would be to make the WPS do all of the complex, time related data access and exploit a capability of the FPS service to accept data in-line rather than asking it to retrieve specified data. This is only one approach but was the one taken in OWS-9. This then exploits the capabilities of the FPS to render data, but takes away the issue of managing data extraction from the WFSs.

5.4 Component Providers

5.4.1 Snowflake Web Feature Service

A WFS component is provided by Snowflake Software. For more information, please refer to the OWS-9 Aviation Architecture Engineering Report.

5.4.2 Comsoft Web Feature Service

A WFS component is provided by Comsoft. For more information, please refer to the OWS-9 Aviation Architecture Engineering Report.

5.4.3 Luciad Feature Portrayal Service

An FPS component is provided by Luciad. For more information, please refer to the OWS-9 Aviation Architecture Engineering Report.

5.4.4 Luciad Web Processing Service

A WPS component is provided by Luciad. For more information, please refer to the OWS-9 Aviation Architecture Engineering Report.

5.4.5 Envitia ePIB Client

A web based client which emulates the functionality to be provided by the SESAR.This is primarily to allow the overall workflow to be demonstrated.

5.4.6 Galdos ebRIM Registry

A Registry Service is provided by Galdos. For more information, please refer to the OWS-9 Aviation Architecture Engineering Report.

6 Design Issues

The following design issues have been identified and resolved during OWS-9:

6.1 Comparison of FPS data exchange options

This section summarizes and compares all possible ways in which we can request an airport map from an FPS based on:

- a list of Digital NOTAMs
- the AIXM 5.1 baseline data/ feature types that need to be shown
- a style

The main approach is to define an OGC Styled Layer Descriptor (SLD) that refers to the data and style information, but multiple implementations are possible.

6.1.1 Inline features

The OGC SLD that is sent to the FPS can contain UserLayer elements with inline (embedded) features. This allows us to prepare all feature data in the WPS and embed them in the SLD request towards to the FPS. The preparation of the feature data includes the calculation of snapshot time slices calculated from the combined Digital NOTAM events and AIXM 5.1 baseline data. As an optimization, the baseline data to which no NOTAMs apply, can still be rendered from static WFS layers.

Advantages:

- The FPS only has to deal with baseline and snapshot time slices, so no complex temporal handling is required.
- Sending a snapshot time slice is more efficient compared to sending features with baseline and one or more tempdelta time slices.
- Sending only tempdelta time slices would require something to indicate that the server still has to fetch baseline time slices, which is more complex.

Disadvantages:

- SLD 1.1.0 only supports including GML 3.1 FeatureCollection, so a small change to the schema is needed to relax this constraint.
- Requests can grow quite large with all these snapshots (but you can argue that the FPS will have to get the NOTAM data from somewhere anyway).

- Perhaps fewer possibilities for optimizations on the FPS side, as the inline features can never be cached.
- Client or other service needs to generate snapshots

This approach has been chosen within OWS-9. To circumvent the GML 3.1 FeatureCollection restriction, the SLD XML Schema has been updated to relax this constraint. More specifically, the 'InlineFeature' element is modified as follows:

```
<xsd:element name="InlineFeature">
  <xsd:complexType>
   <xsd:sequence>
   <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
/>
   </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

6.1.2 Temporal WFS query with projection clause

An OGC SLD can also reference a WFS and use a filter to define the data that needs to be displayed. A projection clause can be used to filter specific tempdelta time slices.

Advantages:

- Client does not need to prepare data but prepares a filter instead.
- FPS might be able to cache baseline as well as tempdelta time slices.

Disadvantages:

- The current SLD version only supports encoding an ogc:Filter, while we really need a complete WFS query so we can also specify a filter on the time slices of the feature that are displayed. This seems to be a general shortcoming of the current standard that should be addressed.
- The FPS needs knowledge of the WFS AIXM 5 profile.
- This solution will not be usable until the WMS / SLD / SE schemas are updated to OWS 2.0 and Filter Encoding 2.0, because it requires the ability to embed a FES

2.0 abstract filter query in the SLD; it currently seems however that work on this has stalled.

6.1.3 WFS query with a temporal extent

Given that it is hard to filter out specific time slices with the limitations of WMS 1.3.0, we can also redefine the problem a bit: instead of specifying the desired NOTAMs, the user can also specify a relevant time instant. This time instant can be encoded next to the ogc:Filter for the features as an sld:Extent.

Advantages:

- The filter becomes less complex, as the time slice selection part is reduced to the time instant.
- The reduced complexity of the filter might mean that even lightweight clients can talk directly to the FPS.
- The same or better caching options on FPS side as with using WFS queries.

Disadvantages:

- This does not solve the problem if the user has a very specific list of NOTAMs to visualize.
- The FPS needs knowledge of the WFS AIXM5 profile.

6.2 Example requests / responses

This section describes a few example requests and responses to illustrate the communication between the various components in the architecture.

6.2.1 Example Digital NOTAM

As an example, we define a Digital NOTAM event that temporarily adds a crane at Chicago O' Hare airport:

```
<?xml version="1.0" encoding="UTF-8"?>
<message:AIXMBasicMessage
xmlns:message="http://www.aixm.aero/schema/5.1/message"
xmlns:gts="http://www.isotc211.org/2005/gts"
xmlns:gco="http://www.isotc211.org/2005/gco"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:gss="http://www.isotc211.org/2005/gss"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gsr="http://www.isotc211.org/2005/gsr"
xmlns:gmd="http://www.isotc211.org/2005/gmd"
xmlns:event="http://www.aixm.aero/schema/5.1/event"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/XMLSchema
http://www.w3.org/2001/XMLSchema.xsd http://www.opengis.net/gml/3.2
http://www.aixm.aero/gallery/content/public/schema/5.1/ISO_19136_Sch
emas/gml.xsd http://www.aixm.aero/schema/5.1/message
http://www.aixm.aero/gallery/content/public/schema/5.1/message/AIXM_
BasicMessage.xsd http://www.isotc211.org/2005/gmd
http://www.aixm.aero/gallery/content/public/schema/5.1/ISO_19139_Sch
emas/gmd/gmd.xsd http://www.aixm.aero/schema/5.1/event
http://www.aixm.aero/gallery/content/public/schema/5.1/event/Event_F
eatures.xsd http://www.w3.org/1999/xlink
http://www.aixm.aero/gallery/content/public/schema/5.1/xlink/xlinks.
xsd " gml:id="FNS-Message">
  <message:hasMember>
    <event:Event gml:id="uuid.75d06775-ce9b-4524-92e1-c06999fb7660">
      <gml:identifier codeSpace="urn:uuid:">75d06775-ce9b-4524-92e1-
c06999fb7660</gml:identifier>
      <event:timeSlice>
        <event:EventTimeSlice gml:id="ID_964">
          <gml:validTime>
            <gml:TimeInstant gml:id="ID_965">
              <gml:timePosition>2012-08-
01T06:00:00Z</gml:timePosition>
            </gml:TimeInstant>
          </gml:validTime>
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:featureLifetime>
            <gml:TimePeriod gml:id="ID_966">
              <gml:beginPosition>2012-08-
01T06:00:00Z</gml:beginPosition>
              <gml:endPosition>2012-08-
31T18:00:00Z</gml:endPosition>
            </gml:TimePeriod>
          </aixm:featureLifetime>
          <event:encoding>DIGITAL</event:encoding>
          <event:scenario>OBS.NEW</event:scenario>
          <event:revision>2012-07-30T14:00:00Z</event:revision>
        </event:EventTimeSlice>
      </event:timeSlice>
    </event:Event>
  </message:hasMember>
  <message:hasMember>
    <aixm:VerticalStructure gml:id="uuid.6a934ad9-79fe-47c2-bf59-
5c5c70efe9a6">
      <gml:identifier codeSpace="urn:uuid:">6a934ad9-79fe-47c2-bf59-
5c5c70efe9a6</gml:identifier>
      <aixm:timeSlice>
        <aixm:VerticalStructureTimeSlice gml:id="ID_972">
          <gml:validTime>
            <gml:TimeInstant gml:id="ID_973">
              <gml:timePosition>2012-08-
01T06:00:00Z</gml:timePosition>
            </gml:TimeInstant>
          </gml:validTime>
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:name>KORD_CRANE</aixm:name>
          <aixm:type>CRANE</aixm:type>
          <aixm:lighted>YES</aixm:lighted>
          <aixm:part>
            <aixm:VerticalStructurePart gml:id="ID_974">
              <aixm:verticalExtent
uom="FT">85.0</aixm:verticalExtent>
              <aixm:type>CRANE</aixm:type>
              <aixm:mobile>NO</aixm:mobile>
              <aixm:horizontalProjection_location>
                <aixm:ElevatedPoint
srsName="urn:ogc:def:crs:EPSG::4326" gml:id="ID_975">
                  <gml:pos>41.96548973 -087.89822257</gml:pos>
                  <aixm:elevation uom="FT">743</aixm:elevation>
                </aixm:ElevatedPoint>
              </aixm:horizontalProjection_location>
            </aixm:VerticalStructurePart>
          </aixm:part>
          <aixm:annotation>
            <aixm:Note gml:id="ID_976">
              <aixm:purpose>WARNING</aixm:purpose>
              <aixm:translatedNote>
                <aixm:LinguisticNote gml:id="ID_977">
                  <aixm:note>located at airport KORD</aixm:note>
                </aixm:LinguisticNote>
              </aixm:translatedNote>
            </aixm:Note>
          </aixm:annotation>
          <aixm:extension>
            <event:VerticalStructureExtension gml:id="ID_982">
```

```
              <event:theEvent xlink:href="#uuid.75d06775-ce9b-4524-
92e1-c06999fb7660" xsi:nil="true"/>
            </event:VerticalStructureExtension>
          </aixm:extension>
        </aixm:VerticalStructureTimeSlice>
      </aixm:timeSlice>
    </aixm:VerticalStructure>
  </message:hasMember>
</message:AIXMBasicMessage>
```

6.2.2 The request from the ePIB application to the WPS

In order to generate a static airport map for Chicago O' Hare with a graphical representation of the Digital NOTAM event from 6.3.1, the ePIB application might sent the following request to the WPS:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/XMLSchema
http://www.w3.org/2001/XMLSchema.xsd
http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd
http://www.opengis.net/ows/1.1
http://schemas.opengis.net/ows/1.1.0/owsAll.xsd
http://www.w3.org/1999/xlink http://www.w3.org/1999/xlink.xsd "
service="WPS" version="1.0.0">
  <ows:Identifier>GenerateMapGraphic</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>notam</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="text/xml"
schema="http://www.aixm.aero/gallery/content/public/schema/5.1/event
/Event_Features.xsd">
          <message:AIXMBasicMessage
xmlns:message="http://www.aixm.aero/schema/5.1/message"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2" gml:id="FNS-Message">
            <message:hasMember>
              <event:Event
xmlns:event="http://www.aixm.aero/schema/5.1/event"
gml:id="uuid.75d06775-ce9b-4524-92e1-c06999fb7660">
```

```
                <gml:identifier codeSpace="urn:uuid:">75d06775-ce9b-
4524-92e1-c06999fb7660</gml:identifier>
                <event:timeSlice>
                  <event:EventTimeSlice gml:id="ID_964">
                    <gml:validTime>
                      <gml:TimeInstant gml:id="ID_965">
                        <gml:timePosition>2012-08-
01T06:00:00Z</gml:timePosition>
                      </gml:TimeInstant>
                    </gml:validTime>

<aixm:interpretation>PERMDELTA</aixm:interpretation>
                    <aixm:sequenceNumber>1</aixm:sequenceNumber>
                    <aixm:featureLifetime>
                      <gml:TimePeriod gml:id="ID_966">
                        <gml:beginPosition>2012-08-
01T06:00:00Z</gml:beginPosition>
                        <gml:endPosition>2012-08-
31T18:00:00Z</gml:endPosition>
                      </gml:TimePeriod>
                    </aixm:featureLifetime>
                    <event:encoding>DIGITAL</event:encoding>
                    <event:scenario>OBS.NEW</event:scenario>
                    <event:revision>2012-07-
30T14:00:00Z</event:revision>
                  </event:EventTimeSlice>
                </event:timeSlice>
              </event:Event>
            </message:hasMember>
            <message:hasMember>
              <aixm:VerticalStructure gml:id="uuid.6a934ad9-79fe-
47c2-bf59-5c5c70efe9a6">
                <gml:identifier codeSpace="urn:uuid:">6a934ad9-79fe-
47c2-bf59-5c5c70efe9a6</gml:identifier>
                <aixm:timeSlice>
                  <aixm:VerticalStructureTimeSlice gml:id="ID_972">
                    <gml:validTime>
                      <gml:TimeInstant gml:id="ID_973">
                        <gml:timePosition>2012-08-
01T06:00:00Z</gml:timePosition>
                      </gml:TimeInstant>
                    </gml:validTime>

<aixm:interpretation>PERMDELTA</aixm:interpretation>
                    <aixm:sequenceNumber>1</aixm:sequenceNumber>
                    <aixm:name>KORD_CRANE</aixm:name>
                    <aixm:type>CRANE</aixm:type>
                    <aixm:lighted>YES</aixm:lighted>
                    <aixm:part>
                      <aixm:VerticalStructurePart gml:id="ID_974">
                        <aixm:verticalExtent
uom="FT">85.0</aixm:verticalExtent>
                        <aixm:type>CRANE</aixm:type>
                        <aixm:mobile>NO</aixm:mobile>
                        <aixm:horizontalProjection_location>
                          <aixm:ElevatedPoint
srsName="urn:ogc:def:crs:EPSG::4326" gml:id="ID_975">
                            <gml:pos>41.96548973 -
87.89822257</gml:pos>
                            <aixm:elevation
uom="FT">743</aixm:elevation>
                          </aixm:ElevatedPoint>
                        </aixm:horizontalProjection_location>
                      </aixm:VerticalStructurePart>
                    </aixm:part>
                    <aixm:annotation>
                      <aixm:Note gml:id="ID_976">
                        <aixm:purpose>WARNING</aixm:purpose>
                        <aixm:translatedNote>
                          <aixm:LinguisticNote gml:id="ID_977">
                            <aixm:note>located at airport
KORD</aixm:note>
                          </aixm:LinguisticNote>
                        </aixm:translatedNote>
                      </aixm:Note>
                    </aixm:annotation>
                    <aixm:extension>
                      <wstxns8:VerticalStructureExtension
xmlns:wstxns8="http://www.aixm.aero/schema/5.1/event"
gml:id="ID_982">
                        <wstxns8:theEvent
xlink:href="#uuid.75d06775-ce9b-4524-92e1-c06999fb7660"
xsi:nil="true" />
                      </wstxns8:VerticalStructureExtension>
                    </aixm:extension>
                  </aixm:VerticalStructureTimeSlice>
                </aixm:timeSlice>
              </aixm:VerticalStructure>
            </message:hasMember>
          </message:AIXMBasicMessage>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>timeRange</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="text/xml"
schema="http://schemas.opengis.net/gml/3.2.1/temporal.xsd">
          <gml:TimePeriod
xmlns:gml="http://www.opengis.net/gml/3.2">
            <gml:beginPosition>2010-08-
27T08:59:00.000Z</gml:beginPosition>
            <gml:endPosition>2012-08-
27T08:59:00.000Z</gml:endPosition>
          </gml:TimePeriod>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>width</ows:Identifier>
      <wps:Data>
        <wps:LiteralData dataType="http://www.w3.org/TR/xmlschema-
2/#int">640</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>height</ows:Identifier>
      <wps:Data>
        <wps:LiteralData dataType="http://www.w3.org/TR/xmlschema-
2/#int">480</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>areaOfInterest</ows:Identifier>
      <wps:Data>
        <wps:BoundingBoxData crs="CRS:84">
          <ows:LowerCorner>-87.93262984 -
87.87940131</ows:LowerCorner>
          <ows:UpperCorner>41.95313946 42.00232455</ows:UpperCorner>
        </wps:BoundingBoxData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>configuration</ows:Identifier>
      <wps:Data>
        <wps:LiteralData dataType="http://www.w3.org/TR/xmlschema-
2/#string">ApronElement,RunwayElement,TaxiwayElement,GuidanceLine,Ve
rticalStructure</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument>
```

```
      <wps:Output mimeType="image/png">
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        <ows:Abstract>result</ows:Abstract>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

6.2.3 The request from the WPS to the FPS

When the WPS has queried all necessary data and styling information, it is ready to generate an SLD and send it to the FPS. The following SLD is an example request illustrating the inline features approach chosen within OWS-9 - note that not all feature data is listed to limit the request size in this document:

```
<?xml version='1.0' encoding='UTF-8'?>
<sld:GetMap xmlns:sld="http://www.opengis.net/sld"
xmlns:ns2="http://www.opengis.net/wms"
            xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ns0="http://www.w3.org/2001/SMIL20/"
xmlns:se="http://www.opengis.net/se"
            xmlns:gml="http://www.opengis.net/gml/3.2"
            xmlns:ns1="http://www.w3.org/2001/SMIL20/Language"
            xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema"
            xsi:schemaLocation="http://www.opengis.net/ogc
http://schemas.opengis.net/filter/1.1.0/filter.xsd
http://www.w3.org/2001/XMLSchema
http://www.w3.org/2001/XMLSchema.xsd http://www.opengis.net/se
http://schemas.opengis.net/se/1.1.0/FeatureStyle.xsd
http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.1.0/StyledLayerDescriptor.xsd
http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gmlBase.xsd
http://www.opengis.net/ows
http://schemas.opengis.net/ows/1.0.0/owsAll.xsd
http://www.w3.org/1999/xlink
http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd"
            version="1.3.0">
  <sld:StyledLayerDescriptor version="1.1.0">
    <se:Name>KORD_layout</se:Name>
    <sld:UserLayer>
      <se:Name>VerticalStructures</se:Name>
      <sld:InlineFeature>
<message:AIXMBasicMessage
xmlns:message="http://www.aixm.aero/schema/5.1/message"
xmlns:gts="http://www.isotc211.org/2005/gts"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:smXML="http://www.isotc211.org/smXML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:gss="http://www.isotc211.org/2005/gss"
xmlns:gsr="http://www.isotc211.org/2005/gsr"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:scXML="http://www.isotc211.org/scXML"
xmlns:iso19115="http://www.isotc211.org/2005/gmd"
xsi:schemaLocation="http://www.isotc211.org/2005/gts
iso/19139/gts/gts.xsd http://www.isotc211.org/2005/gco
iso/19139/gco/gco.xsd http://www.w3.org/2001/XMLSchema
http://www.w3.org/2001/XMLSchema.xsd
http://www.isotc211.org/2005/gss iso/19139/gss/gss.xsd
http://www.isotc211.org/2005/gsr iso/19139/gsr/gsr.xsd
http://www.isotc211.org/2005/gmd iso/19139/gmd/gmd.xsd "
gml:id="M001">
  <gml:identifier
codeSpace="http://www.faa.gov/NASR">KORD</gml:identifier>
   <message:hasMember>
    <aixm:VerticalStructure gml:id="VPS51">
      <gml:identifier>VPS51</gml:identifier>
      <aixm:timeSlice>
        <aixm:VerticalStructureTimeSlice gml:id="TS51">
          <gml:validTime>
            <gml:TimePeriod gml:id="TSP51">
              <gml:beginPosition>2008-03-
23T14:00:00</gml:beginPosition>
              <gml:endPosition
indeterminatePosition="unknown"></gml:endPosition>
            </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>BASELINE</aixm:interpretation>
          <aixm:sequenceNumber>1</aixm:sequenceNumber>
          <aixm:type>Terminal Building</aixm:type>
          <aixm:part>
            <aixm:VerticalStructurePart>
              <aixm:verticalExtent
uom="FT">5.764645867621994</aixm:verticalExtent>
              <aixm:horizontalProjection_surfaceExtent>
```

```
                <aixm:ElevatedSurface
srsName="urn:ogc:def:crs:OGC:1.3:CRS84" gml:id="b9826ec1">
                  <gml:patches>
                    <gml:PolygonPatch>
                      <gml:exterior>
                        <gml:LinearRing>
                          <gml:posList>-87.8955933097789
41.990394826363406 -87.8955909232215 41.99027810108 -
87.89554754498229 41.9902785941685 -87.89554993153969
41.990395320791706 -87.8955933097789
41.990394826363406</gml:posList>
                        </gml:LinearRing>
                      </gml:exterior>
                    </gml:PolygonPatch>
                  </gml:patches>
                  <aixm:elevation
uom="FT">737.235354132378</aixm:elevation>
                </aixm:ElevatedSurface>
              </aixm:horizontalProjection_surfaceExtent>
            </aixm:VerticalStructurePart>
          </aixm:part>
        </aixm:VerticalStructureTimeSlice>
      </aixm:timeSlice>
    </aixm:VerticalStructure>
  </message:hasMember>
</message:AIXMBasicMessage>
  </sld:InlineFeature>
      <sld:LayerFeatureConstraints>
        <sld:FeatureTypeConstraint>
          <se:FeatureTypeName
xmlns:ns3="http://www.aixm.aero/schema/5.1">ns3:VerticalStructure
          </se:FeatureTypeName>
        </sld:FeatureTypeConstraint>
      </sld:LayerFeatureConstraints>
      <sld:UserStyle>
        <se:Name>VerticalStructureUserStyle</se:Name>
        <sld:IsDefault>true</sld:IsDefault>
        <se:FeatureTypeStyle>
          <se:Name>VerticalStructureFeatureTypeStyle</se:Name>

<se:FeatureTypeName>VerticalStructureType</se:FeatureTypeName>
          <se:Rule>
            <se:ElseFilter/>
            <se:MinScaleDenominator>0.0</se:MinScaleDenominator>
            <se:MaxScaleDenominator>INF</se:MaxScaleDenominator>
            <se:PolygonSymbolizer>
              <se:Geometry>
```

```
                <ogc:PropertyName
xmlns:ns3="http://www.aixm.aero/schema/5.1">
                  ns3:timeSlice/ns3:VerticalStructureTimeSlice
                </ogc:PropertyName>
              </se:Geometry>
              <se:Fill>
                <se:SvgParameter
name="fill">#4240F7</se:SvgParameter>
              </se:Fill>
              <se:Stroke>
                <se:SvgParameter name="stroke-opacity">
                  <ogc:Literal>1</ogc:Literal>
                </se:SvgParameter>
                <se:SvgParameter
name="stroke">#7D7D7D</se:SvgParameter>
                <se:SvgParameter name="stroke-width">
                  <ogc:Literal>1</ogc:Literal>
                </se:SvgParameter>
              </se:Stroke>
            </se:PolygonSymbolizer>
          </se:Rule>
        </se:FeatureTypeStyle>
      </sld:UserStyle>
    </sld:UserLayer>
  </sld:StyledLayerDescriptor>
  <sld:CRS>CRS:84</sld:CRS>
  <sld:BoundingBox>
    <ows:LowerCorner>-87.93262984 41.95313946</ows:LowerCorner>
    <ows:UpperCorner>-87.87940131 42.00232455</ows:UpperCorner>
  </sld:BoundingBox>
  <sld:Output>
    <sld:Size>
      <sld:Width>800</sld:Width>
      <sld:Height>600</sld:Height>
    </sld:Size>
    <ns2:Format>image/png</ns2:Format>
    <sld:Transparent>false</sld:Transparent>
    <sld:BGcolor>0xffffff</sld:BGcolor>
  </sld:Output>
</sld:GetMap
```

Annex A

ePIB map generation WPS profile

This annex describes the WPS profile that has been defined for the ePIB map generation process. Table 1 lists the required inputs for this process:

Parameter name	Description	Type	Multiplicity
notam	The Digital NOTAM events to be displayed on the map	String	0..*
format	The image format of the map	String	1..1
width	The map width	Integer	1..1
height	The map height	Integer	1..1
areaOfInterest	The geographica area (bounding box) for the graphic	Envelope	1..1
timeRange	The time range which the map will cover	Date Time Interval	1..1
configuration	The description of the configuration of the graphic	String	1..1

Table 3 – ePIB WPS process inputs

The execution of this process will return a graphic in the format requested in the “format” parameter. It is proposed that the execution is synchronous, assuming that the result can be returned before the typical http timeout (+- 20 seconds).

The following XML document describes the resulting WPS profile:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions
xmlns:wps="http://www.opengis.net/wps/1.0.0"
                         xmlns:ows="http://www.opengis.net/ows/1.1"
                         xmlns:xlink="http://www.w3.org/1999/xlink"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd
http://www.opengis.net/ows/1.1
http://schemas.opengis.net/ows/1.1.0/owsAll.xsd
http://www.w3.org/1999/xlink http://www.w3.org/1999/xlink.xsd"
                         service="WPS" version="1.0.0" lang="en-US">

  <wps:ProcessDescription processVersion="1.0.0"
storeSupported="false" statusSupported="false">
    <ows:Identifier>GenerateMapGraphic</ows:Identifier>
    <ows:Title>Generates a graphic depicting the Digital NOTAMS
requested against a relevant
      backdrop as defined by a configuration
    </ows:Title>
    <ows:Abstract>
      This process has been designed to support the OWS-9 Aviation
ePIB map generation use case: it
      takes a set of Digital NOTAMs and map configuration settings,
and generates a graphic
      depicting the NOTAMs against a relevant backdrop as defined by
the map configuration settings.
    </ows:Abstract>
    <ows:Metadata xlink:title="epib"/>
    <ows:Metadata xlink:title="Digital NOTAM"/>
    <wps:DataInputs>
      <wps:Input minOccurs="0" maxOccurs="100">
        <ows:Identifier>notam</ows:Identifier>
        <ows:Title>Input Digital NOTAMs</ows:Title>
        <ows:Abstract>The input Digital NOTAMs.</ows:Abstract>
        <wps:ComplexData>
          <wps:Default>
            <wps:Format>
              <wps:MimeType>text/xml</wps:MimeType>
<wps:Schema>http://www.aixm.aero/gallery/content/public/schema/5.1/e
vent/Event_Features.xsd</wps:Schema>
            </wps:Format>
          </wps:Default>
          <wps:Supported>
            <wps:Format>
              <wps:MimeType>text/xml</wps:MimeType>
```

```
<wps:Schema>http://www.aixm.aero/gallery/content/public/schema/5.1/e
vent/Event_Features.xsd</wps:Schema>
            </wps:Format>
          </wps:Supported>
        </wps:ComplexData>
      </wps:Input>
      <wps:Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>timeRange</ows:Identifier>
        <ows:Title>Time range to be reflected by the
EPIB</ows:Title>
        <ows:Abstract>The time range to be reflected by the
EPIB.</ows:Abstract>
        <wps:ComplexData>
          <wps:Default>
            <wps:Format>
              <wps:MimeType>text/xml</wps:MimeType>
<wps:Schema>http://schemas.opengis.net/gml/3.2.1/temporal.xsd</wps:S
chema>
            </wps:Format>
          </wps:Default>
          <wps:Supported>
            <wps:Format>
              <wps:MimeType>text/xml</wps:MimeType>
<wps:Schema>http://schemas.opengis.net/gml/3.2.1/temporal.xsd</wps:S
chema>
            </wps:Format>
          </wps:Supported>
        </wps:ComplexData>
      </wps:Input>
      <wps:Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>width</ows:Identifier>
        <ows:Title>Graphic width</ows:Title>
        <ows:Abstract>The width of the graphic that is
returned.</ows:Abstract>
        <wps:LiteralData>
          <ows:DataType
ows:reference="http://www.w3.org/TR/xmlschema-
2/#int">int</ows:DataType>
          <wps:DefaultValue>640</wps:DefaultValue>
        </wps:LiteralData>
      </wps:Input>
      <wps:Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>height</ows:Identifier>
        <ows:Title>Graphic height</ows:Title>
        <ows:Abstract>The height of the graphic that is
returned.</ows:Abstract>
        <wps:LiteralData>
```

```
          <ows:DataType
ows:reference="http://www.w3.org/TR/xmlschema-
2/#int">int</ows:DataType>
          <wps:DefaultValue>480</wps:DefaultValue>
        </wps:LiteralData>
      </wps:Input>
      <wps:Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>areaOfInterest</ows:Identifier>
        <ows:Title>Graphic area of interest</ows:Title>
        <ows:Abstract>The area of interest covered by the
graphic.</ows:Abstract>
        <wps:BoundingBoxData>
          <wps:Default>
            <wps:CRS>CRS:84</wps:CRS>
          </wps:Default>
          <wps:Supported>
            <wps:CRS>CRS:84</wps:CRS>
          </wps:Supported>
        </wps:BoundingBoxData>
      </wps:Input>
      <wps:Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>configuration</ows:Identifier>
        <ows:Title>Graphic configuration</ows:Title>
        <ows:Abstract>A comma-separated list of values that defines
the background of the graphic.
        </ows:Abstract>
        <wps:LiteralData>
          <ows:DataType
ows:reference="http://www.w3.org/TR/xmlschema-2/#string">string
          </ows:DataType>

<wps:DefaultValue>ApronElement,RunwayElement,TaxiwayElement,Guidance
Line,VerticalStructure</wps:DefaultValue>
        </wps:LiteralData>
      </wps:Input>
    </wps:DataInputs>
    <wps:ProcessOutputs>
      <wps:Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>Graphic result</ows:Title>
        <ows:Abstract>The graphic result, defined as a
bitmap.</ows:Abstract>
        <wps:ComplexOutput>
          <wps:Default>
            <wps:Format>
              <wps:MimeType>image/png</wps:MimeType>
            </wps:Format>
          </wps:Default>
          <wps:Supported>
            <wps:Format>
              <wps:MimeType>image/png</wps:MimeType>
            </wps:Format>
            <wps:Format>
              <wps:MimeType>image/jpeg</wps:MimeType>
            </wps:Format>
          </wps:Supported>
        </wps:ComplexOutput>
      </wps:Output>
    </wps:ProcessOutputs>
  </wps:ProcessDescription>
</wps:ProcessDescriptions>
```

Annex B

Alternative Approaches and Potential Future Work

This annex describes some of the other elements which were discussed in relation to ePIB and some potential future options which could be integrated.

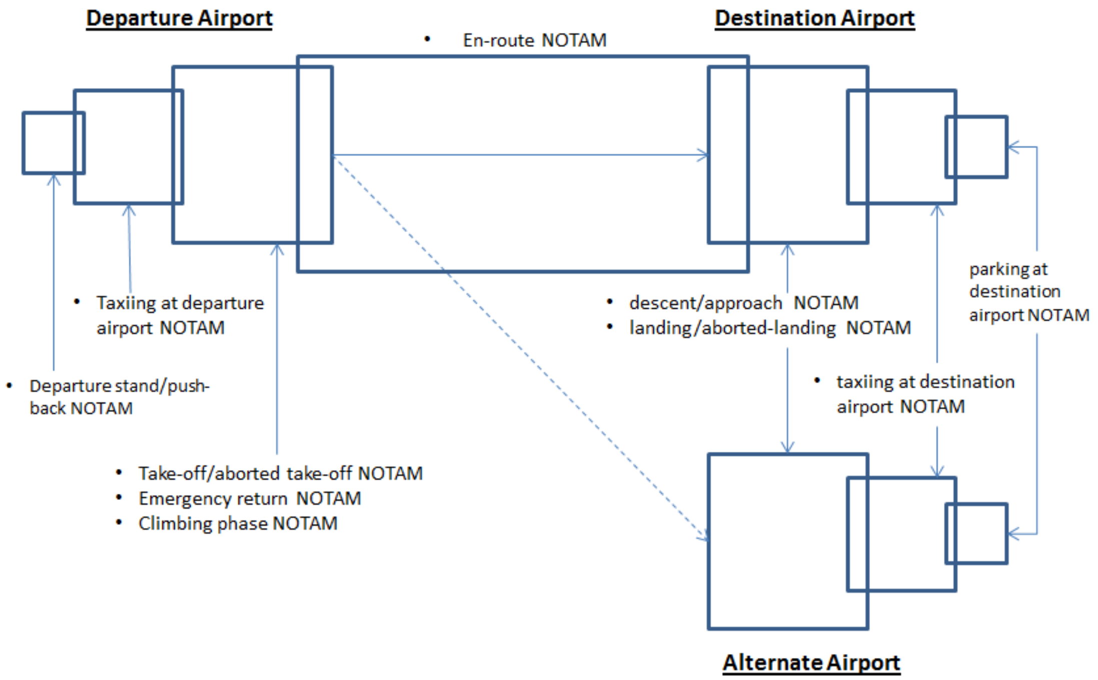
Background Map Alternatives

In the work under OWS-9 the background map was kept relatively simple with the main focus being on showing a sufficient backdrop. The features selected were defined by a GML query returning specific feature types (specified in the request). There was some discussion though around the benefit of different types of backdrop where detailed background information was not available in AIXM format or where, for interpretation or even cosmetic reasons, the information was more palatable when accessed in a different form, for example as say a satellite image backdrop.

In order to address this, it is firstly necessary to support alternative types of background requests in the WPS interface or separate out the background request and the foreground request (NOTAMS) and integrate the two images returned from each process separately.

Lastly if a scheme such as this were implemented, it is likely that there would be a need to define, for each airport, a set of potential backgrounds to suit the relevant level of static map display. The obvious place to store the individual services supported by a specific airport is probably in an ebRIM registry (characterized by their role, e.g. taxing, takeoff etc.).

The diagram below shows the potential static maps which might be required within an ePIB (this is an example only). The examples shown were taken from ref 2.4 (4).


Departure Airport
En-route NOTAM
Destination Airport
Taxiing at departure airport NOTAM
Departure stand/push-back NOTAM
Take-off/aborted take-off NOTAM
Emergency return NOTAM
Climbing phase NOTAM
descent/approach NOTAM
landing/aborted-landing NOTAM
parking at destination airport NOTAM
taxiing at destination airport NOTAM
Alternate Airport

Dynamic Content

While one output of an ePIB is PDF another output (and the one demonstrated in the prototype) is HTML There may be merit is being able to click on objects in the map and display the actual NOTAM details, or jump (hyperlink to) the textual description of the NOTAM. This would require more than a static airport map to be generated but also a set of link data. This is potentially available via a WFS but as with the airport maps would probably be most suitably provided via another WPS service which returns the HTML overlay.

Alternative Architectures/Capability of Current OGC Standards

The approach taken in OWS-9 is based on using the WPS for the bulk of the work, and only exploiting a relatively small capability in the Feature Portrayal Service. As stated the major issue is the ability of the FPS interface to support the complex temporality model present in AIXM. Although work is going on in SLD, the FPS specification (actually SLD-WMS) is relatively old and certainly doesn't take into account new capabilities in WFS 2.0. A more detailed investigation of what is needed here in terms of the FPS interface, SLD and the actual business rules to avoid any more customization than necessary. Having said that there is an argument for a thin integration tear, matching specific interface requirements to general capabilities (which is how the WPS is used here). If this is the right model for the evolution of specific service provision, is the mix of functionality implemented in the WPS and that exploited from the general service infrastructure appropriate? Interoperability and agility come through componentization (assembling generic components with a relatively light wrapper). Is this valid in this case?

Defining ePIB Content through Context Documents

In the OWS-9 experiment, the description of information in the ePIB was relatively simple, a list of feature types and a set of NOTAMS. For the ePIB and also potentially for other areas, it may be useful to consider how a set of information can be assembled by defining an OWS Context. This can define a set of information layers and potentially processes which when assembled for a given view (including time range and area of interest). This is very close to the need within the ePIB static maps. It would support some of the architectural flexibility useful in supporting other problems than simply ePIB. This may be an area worth further experimentation in order to establish if it is a useful, relevant and practical approach. This approach would, for example, allow a client supporting OWS Context to load each ePIB map definition and display it under other application data. This is of course speculating on the specific requirements but it does offer some flexibility in approach.

Issues of Performance

The current implementation of the ePIB Map workflow is very flexible, and guaranteed to be up-to-date because all data is retrieved and processed at the point of ePIB map creation. The disadvantage of this is that there is a significant delay in generating each map (10s of seconds). Although further investigation is needed, this is believed to be because of the number of service transactions and the amount of XML transported from service to service. There is a further potential issue that the service is likely to need to satisfy a significant community demand as all pilots or flight planners will be using it. Thus the significant processing will take place for each transaction resulting in significant scalability issues.

While it is difficult to significantly optimise the NOTAM section of the workflow, in reality this is not believed to cause a particularly heavy XML burden. It is the background maps that are believed to be pre-dominant. And as these are generated from AIXM directly, which is it'self likely to be generated on the AIRAC cycle, it would make sense to pre-generate the background map component on that cycle (doing it once per cycle rather than once per request which could run into 1000s of requests per day). Of course there is the intention to ultimately use the agility of a new aviation architecture to shorten the AIRAC cycle and ultimately move to an approach which doesn't impose a rigid cycle, but in either case a 'pre-generation' approach would still reduce the processing requirements as it would still only be done once per change rather than once per user transaction.

If this approach was taken one option would be to deploy the background maps using and integrated WMS (much more efficient than an FPS because of the removal of the XML transfers from WFS to WMS and the extra service interface latency). Another approach would be to build an optimized cache using the WFS/FPS architecture already created and then deploy this with a WMS or even WMTS if the portrayal can be defined a-priori.

These options do not preclude the ability to have different representations at different times within the AIRAC cycle; clearly within one cycle there are planned airspace changes which need to be dealt with. In terms of interface the WMS type interface does support the concept of dimensions which allows the specification of a time based request. This concept would also be possible with WMTS. This does of course require all combinations to be pre-built but given there are not that many changes within one cycle the extra generation work is not that high.

It is clear that work needs to be done on establishing the exact performance bottlenecks and load/potential quality of service requirements and measuring what can be achieved as has been undertaken for the main architecture in OWS-9. Future work should evaluate potential performance bottlenecks in more detail and identify quality of service requirements for the ePIB workflow.