Testbed-12 Aviation Security Engineering Report

Table of Contents

Publication Date: 2017-06-30

Approval Date: 2017-06-29

Posted Date: 2017-01-12

Reference number of this document: OGC 16-040r1

Reference URL for this document: http://www.opengis.net/doc/PER/tb12-F001

Category: Public Engineering Report

Editor: Aleksandar Balaban

Title: Testbed-12 Aviation Security Engineering Report

Testbed-12 Aviation Security Engineering Report

COPYRIGHT

WARNING

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by

destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Abstract

The information security is the state of being protected against the unauthorized use of information and services, or the measures taken to achieve that. This report has ben created as part of OGC Testbed 12 aviation thread and on behalf of sponsors from FAA. It gives the readers an overview into the topic of cyber security in the aviation domain, especially in conjunction with OGC compatible web services, which are today de facto standard for aeronautical traffic System Wide Information Management.

Business Value

Geospatial enabled access control and approaches for securing OGC OWS services, as well as identity management methods proposed in this report greatly improve the overall security of aviations services and enable interoperability among regional and global aeronautical traffic management stake holders.

What does this ER mean for the Working Group and OGC in general

This engineering report demonstrates the application of OGC geospatial standards (GeoXACML) in the domain of civil aviation.

Keywords

engineering report, testbed-12, aviation, cybersecurity, authentication, authorization

Proposed OGC Working Group for Review and Approval

Security Domain Working Group (DWG)

Chapter 1. Introduction

1.1. Scope

This Engineering Report (ER) gives guidelines for security architecture implementation with OGC OWS compatible web services especially for the domain of civil aviation and based on the Testbed-12 aviation architecture. The document is applicable to all readers interested in the aviation domain and the common information security topics related to the OGC web services.

1.2. Document contributor contact points

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

Table 1. Contacts

Name	Organization
Aleksandar Balaban	m-click.aero

1.3. Future Work

No future work is planned to this document.

1.4. Foreword

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.

Chapter 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.

- OGC 06-121r9, OGC® Web Services Common Standard

NOTE: This OWS Common Standard contains a list of normative references that are also applicable to this Implementation Standard.

- OGC 08-176r1: OGC® OWS-6 Secure Sensor Web Engineering Report NOTE This OWS-6 ER contains a comprehensive overview to security standards applicable to this ER.
- OGC 06-121r3: OGC® Web Services Common Standard
- OGC 04-095: OpenGIS® Filter Encoding Implementation Specification
- ISO, 35.100: Open systems interconnection (OSI)
- ISO/IEC 10181-1: Information technology — Open Systems Interconnection — Security frameworks for open systems: Overview, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24404
- ISO/IEC 10181-2: Information technology — Open Systems Interconnection — Security frameworks for open systems: Authentication framework, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=18198
- ISO/IEC 10181-3: Information technology — Open Systems Interconnection — Security frameworks for open systems: Access control framework, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=18199
- ISO/IEC 10181-4: Information technology — Open Systems Interconnection — Security frameworks for open systems: Non-repudiation framework, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=23615
- ISO/IEC 10181-5: Information technology — Open Systems Interconnection — Security frameworks for open systems: Confidentiality framework, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24329
- ISO/IEC 10181-6: Information technology — Open Systems Interconnection — Security frameworks for open systems: Integrity framework, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24330
- ISO/IEC 10181-7: Information technology — Open Systems Interconnection — Security frameworks for open systems: Security audit and alarms framework, ISO 1996: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=18200
- HTTP: RFC 2616 - Hypertext Transfer Protocol — HTTP/1.1 – IETF RFC 2616 (1999): http://tools.ietf.org/html/rfc2616
- HTTP Authentication: HTTP Authentication: Basic and Digest Access Authentication – IETF RFC 2617 (1999): https://tools.ietf.org/html/rfc2617
- TLS: Transport Layer Security – IETF RFC 2246 (1999): http://tools.ietf.org/html/rfc2246
- HTTPS: HTTP Over TLS – IETF RFC 2818 (2000): http://tools.ietf.org/html/rfc2818

- X.509/PKI: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks, ITU-T Standard, 08/2005: http://www.ietf.org/html.charters/pkix-charter.html
- URI: Uniform Resource Identifiers (URI): Generic Syntax – IETF RFC 2396 (1998): https://tools.ietf.org/html/rfc2396
- CRL: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile – IETF RFC 3280: http://tools.ietf.org/html/rfc3280
- Cookies: HTTP State Management Mechanism – IETF RFC 2109: https://tools.ietf.org/html/rfc2109
- Web Services Security: SOAP Message Security 1.1 (WS-Security 2004) – OASIS Standard Specification, 1 February 2006: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf
- XML Digital Signature: XML-Signature Syntax and Processing – W3C Recommendation 12 February 2002: http://www.w3.org/TR/xmldsig-core/
- XML Encryption: XML Encryption Syntax and Processing – W3C Recommendation 10 December 2002: http://www.w3.org/TR/xmlenc-core/
- XML Signature Best Practices: XML Signature Best Practices – W3C Working Group Note 11 April 2013: http://www.w3.org/TR/2013/NOTE-xmldsig-bestpractices-20130411/
- WSDL 1.1: Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001: http://www.w3.org/TR/wsdl
- WSDL 2.0: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Recommendation 26 June 2007: http://www.w3.org/TR/wsdl20/
- UDDI: UDDI Spec Technical Committee Draft, OASIS, Dated 20041019: http://www.uddi.org/pubs/uddi_v3.htm
- WS-Policy: Web Services Policy 1.5 – Framework, W3C Recommendation 04 September 2007: http://www.w3.org/TR/2007/REC-ws-policy-20070904/
- WS-Policy Attachment: Web Services Policy 1.5 – Attachment, W3C Recommendation, 04 September 2007: http://www.w3.org/TR/ws-policy-attach/WS-SecurityPolicy: WS-SecurityPolicy 1.2, OASIS Standard, 1 July 2007: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2- spec-os.pdf
- WS-Trust: WS-Trust 1.3, OASIS Standard, 19 March 2007: http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf
- WS-SecureConversation: WS-SecureConversation 1.3, OASIS Standard, 1 March 2007:http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.pdf
- WS-Reliable Messaging: Web Services Reliable Messaging (WS- ReliableMessaging) Version 1.2, Committee Draft, 28 February 2008: http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.2-spec-cd-01.pdf
- WS-RM Policy: Web Services Reliable Messaging Policy Assertion (WS-RM Policy) Version 1.2, Committee Draft, 28 February 2008: http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.2-spec-cd-01.pdf
- WS-MakeConnection: Web Services Make Connection (WS-MakeConnection) Version 1.1,

Committee Draft, 28 February 2008: http://docs.oasis-open.org/ws-rx/wsmc/200702/wsmc-1.1-spec-cd-01.pdf

- WS-Federation / WS-Authorization / WS-Privacy: Web Services Federation Language (WS-Federation) Version 1.2, Editors Draft – 06, May 21, 2008: http://www.oasis-open.org/committees/download.php/28360/ws-federation-1.2- spec-ed-06.doc
- WS-MetadataExchange: Web Services Metadata Exchange (WS- MetadataExchange), Version 1.1, August 2006, Microsoft, IBM, Sun and SAP: http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf
- WS-Transfer: Web Services Transfer (WS-Transfer), W3C Member Submission, 27 September 2006: http://www.w3.org/Submission/WS-Transfer/
- WS-RT: Web Services Resource Transfer (WS-RT), Version 1.0, August 2006: http://schemas.xmlsoap.org/ws/2006/08/resourceTransfer/WS- ResourceTransfer.pdf

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply. In addition, the following terms and definitions apply.

Table 2. Terms and definitions

Term	Definition
Asynchronous Operation	A type of operation whose message exchange pattern allows messages to be sent without precise sequencing, e.g., a flow of sensor event messages which need not be individually acknowledged.
Authentication	The process of verifying an identity claimed by or for a system entity.
Authorization	The granting of rights or permission to a system entity (mainly but not always a user or a group of users) to access a Web service.
Confidentiality	Protective measures that assure that information is not made available or disclosed to unauthorized individuals, entities, or processes (i.e., to any unauthorized system entity).
Datatype	A set of distinct values, characterized by properties of those values, and by operations on those values.
End Point	An association between a fully-specified binding and a physical point (i.e., a network address) at which a service may be accessed.
Format	The arrangement of bits or characters within a group, such as a data element, message, or language.
Idempotent	A term used to describe an operation in which a given message will have the same effect whether it is received once or multiple times; i.e., receiving duplicates of a given message will not cause any undesirable effect.
Input Data	entered into, or the process of entering data into, an information processing system or any of its parts for storage or processing.
Integrity	Protective measures that assure that data has not been changed, destroyed, or lost in an unauthorized or accidental manner.
Message	An identifiable collection of units of information (data elements), presented in a manner suitable for communication, interpretation, or processing within a context of interacting SOA components.
Message Exchange Pattern (MEP)	A template, devoid of application semantics, that describes a generic pattern for the exchange of messages between agents. It describes the relationships (e.g., temporal, causal, sequential, etc.) of multiple messages exchanged in conformance with the pattern, as well as the normal and abnormal termination of any message exchange conforming to the pattern.
Metadata	Data that defines or describes other data.

Namespace	A collection of names, identified by a URI reference, that are used in XML documents as element types and attribute names. The use of XML namespaces to uniquely identify metadata terms allows those terms to be unambiguously used across applications, promoting the possibility of shared semantics.
Non-Repudiation	Protective measures against false denial of involvement in a communication.
Operation	A set of messages related to a single Web service action.
Organization	A unique framework of authority within which a person or persons act, or are designated to act, towards some purpose. Any department, service, or other entity within an organization which needs to be identified for information exchange.
Output	Data transferred out of, or the process by which an information processing system or any of its parts transfers data out of, that system or part.
Precondition	A state or condition that is required to be true before an action can be successfully invoked.
Processing	A set of algorithms, calculations, or business rules that operate on input data in order to produce the required output or to produce a change of internal state.
Protocol	A formal set of conventions governing the format and control of interaction among communicating functional units.
Quality of Service (QoS)	A parameter that specifies and measures the value of a provided service.
Security	The protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.
Security Mechanism	A process (or a device incorporating such a process) that can be used in a system to implement a security service that is provided by or within the system.
Service Consumer	An organization that seeks to satisfy a particular need through the use of capabilities offered by means of a service.
Service Description	The information needed in order to use, or consider using, a service.
Service Provider	An organization that offers the use of capabilities by means of a service.
Service-Oriented Architecture (SOA)	A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. A SOA provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

Structured Data	Data that is organized in well-defined semantic “chunks” or units that are variously called fields, elements, objects, or entities. Individual units are often combined to form larger, more complex units.
Synchronous Operation	A type of operation whose message exchange pattern describes temporally coupled or "lock-step" interactions, e.g., remote procedure call (RPC)-style request-response interactions.
User	A human, his/her agent, a surrogate, or an entity that interacts with information processing systems. A person, an organization entity, or automated process that accesses a system, whether authorized to do so or not.
Web Service	A platform-independent, loosely-coupled software component designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format. Other systems interact with the Web service in a manner prescribed by its description by means of XML-based messages conveyed using Internet transport protocols in conjunction with other Web-related standards.
Web Service Interface	A logical grouping of operations, where each operation represents a single interaction between consumer agents and a Web service. Each operation specifies the types of messages that the service can send or receive as part of that operation without any commitment to transport or wire protocol.

Chapter 4. Conventions

4.1. Abbreviated terms

Table 3. Abbreviated terms

AIM	Aeronautical Information Management
API	Application Program Interface
ATM	Aeronautical Traffic Management
CA	Certification Authority (PKI)
COTS	Commercial Off The Shelf
DCMI	Dublin Core Metadata Initiative
FAA	Federal Aviation Administration
FDR	FAA Data Registry
GML	Geography Markup Language
HTTP(S)	Hypertext Transfer Protocol (Secure)
IdP	Identity Provider
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
MDR	Metadata Registry
NAS	National Airspace System
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
PKI	Public Key Infrastructure
QoS	Quality of Service
SOA	Service-Oriented Architecture
STS	Security Token Service
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WS	Web Service
WSDD	Web Service Description Document
WSDL	Web Service Description Language

XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

4.2. 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 06-121r9].

Chapter 5. Overview

The information security is the state of being protected against the unauthorized use of information and services, or the measures taken to achieve that. This report has ben created as part of OGC Testbed 12 aviation thread and on behalf of sponsors from FAA. The major goal is to give the readers a detailed overview into the topic of cyber security in the domain of aviation, especially in conjunction with (today de facto SWIM standard) OGC compatible web services.

The document first provides an introduction into the major elements of information security establishing a common vocabulary and classification useful for further explanations. The most significant security threads are also shortly explained in this section.

In the next section the aviation stakeholders and the most prominent business cases concerning the ATM/AIS (aeronautical traffic management) are identified in combination with the most relevant security threats, which affect them.

Security related work preformed during the aviation modernization programs in the USA, the EU and by ICAO has been explained and the comparison of current security standards and solutions deployed in the FAA and SESAR SWIM domains has been made.

Following the use cases and general introduction the methods of information security are presented in the form of security functions such as confidentiality, integrity and availability. Terminology and categorization used here has been taken from Testbed-11 recommendations and is based on the ISO standards.

This section introduces security concepts applicable to the OGC compatible aviation web services.

Security functions are usually applied utilizing different public standards and technical implementations, which operate at the different levels of the ISO communication reference model. This section explains the differences between transport and message based security, puts them in the relation with communication message exchange patterns and explains how those concepts are currently applied in the aviation domain.

Next section puts the focus on digital identities management and provision, as well as access right control with security policy enforcement based on geospatial attributes. This topics will become more important through progressing in the liberalization of access to aeronautical data and the data sources get globally connected and becomes accessible via SWIM.

Finally, the proposal for security enhanced Testbed 12 Aviation Architecture in accordance with Testbed 11 outcomes has been presented and explained.

Chapter 6. Status Quo Statement

This section explains the status quo, including the existing problems/issues that have been addressed by this ER.

Traditionally, the aviation IT services are based on unstructured data formats and legacy communication channels such as AFTN and AMHS. Some aviation services provide their functions via public internet, some are deployed inside of organizational's virtual private networks. The aeronautical navigation service providers (ANSP) are responsible to provide business services for aviation stake holders. Current operational solutions (in the most of the cases observed) don't consider the modern SOA concepts and therefore are incapable to enable required level of information availability and system interoperability.

As part of modernization programs on the both sides of Atlantic (FAA NextGen, SESAR JU), a new data formats, service definitions and the massaging/communication infrastructure (the system wide information management - SWIM) have been put in place. New data formats such as AIXM 5.1, iWXXM and FIXM on the OGC GML 3.2 will be exchanged via OGC OWS compatible service endpoints, which in combination with SWIM security requirements provides the main area of this engineering report.

Chapter 7. OGC Aviation Security Use Cases

This section depicts the most prominent business cases concerning the aviation services implemented as OGC OWS compatible services.

7.1. UC1 Testbed-12 Aviation Architecture

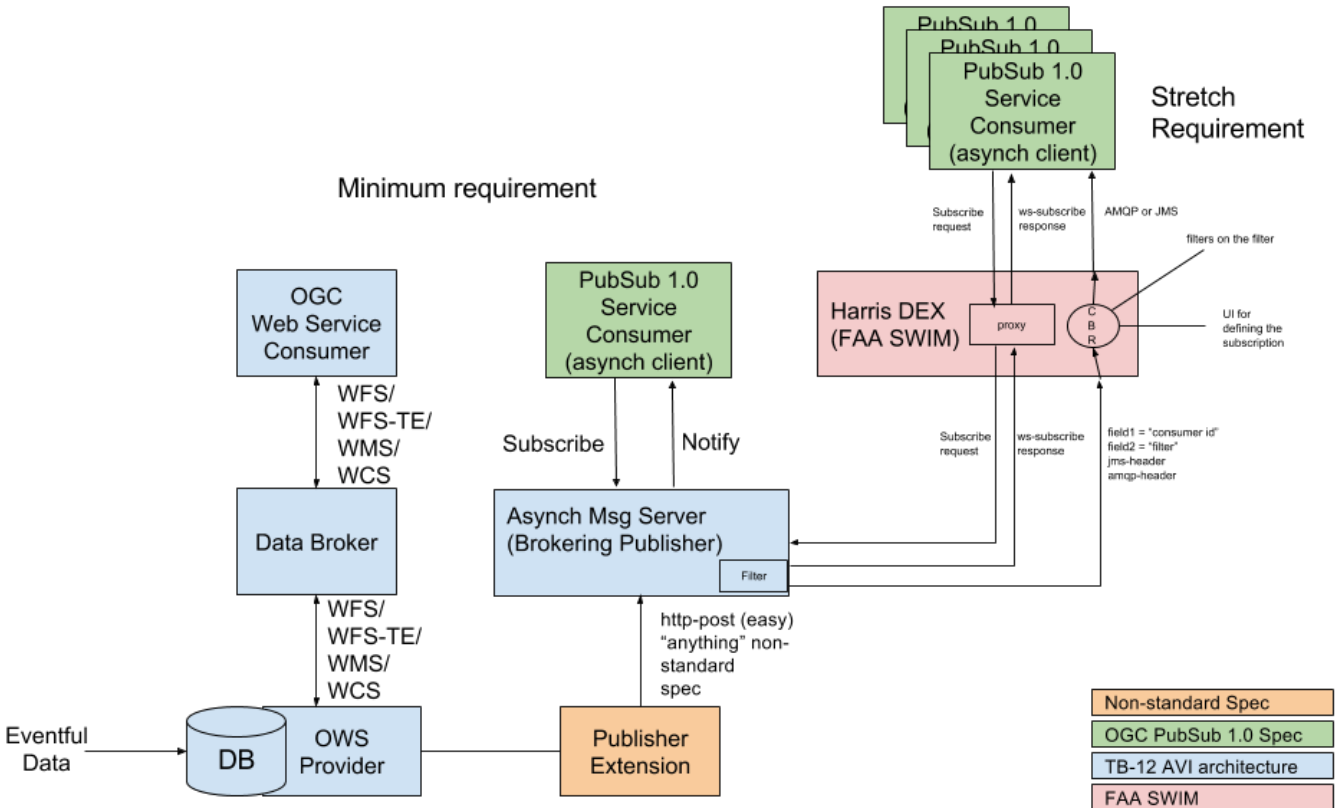

Figure 1. Testbed-12 Aviation Architecture

7.2. UC2 SESAR SWIM

In SESAR, System Wide Information Management (SWIM) has been specified at the technical (endpoints, solution stacks) and at the service (logical interfaces) level. At the technical level there are three so called profiles with different requirements which also affects security. Every profile contains many stack definitions. Following listing gives an overview into the one of them, the web technology (and thus relevant for OGC OWS) based Yellow Profile technical specification. Here we provide an incomplete list of generic interface binding titles. They are descriptive enough to explain what kind of communication stack and security is required:

- AMQP over SSL/TLS (amqps) over TCP.
- Over HTTPS GET/POST over TCP.
- Plain Old XML (POX) over HTTPS POST over TCP.
- SOAP 1.1 over HTTPS POST over TCP.
- SOAP 1.2 over HTTPS POST over TCP.
- SOAP 1.1 with WS-Security 1.1 and UsernameToken 1.1 over HTTPS POST over TCP.

- SOAP 1.1 with WS-Security 1.1 and UsernameToken 1.1 over HTTPS POST over TCP.
- SOAP 1.2 with WS-Security 1.1 and UsernameToken 1.1 over HTTPS POST over TCP.
- SOAP 1.1 with WS-Security 1.1 and WSSE X.509 Certificate Token Profile 1.0 or WSSE X.509 Certificate Token Profile 1.1 over HTTP POST over TCP.
- SOAP 1.2 with WSSE X.509 Certificate Token Profile 1.0 or WS-Security 1.1 and WSSE X.509 Certificate Token Profile 1.1 over HTTP POST over TCP.
- SOAP 1.1 with WS-Security 1.1, WS-Trust 1.4, WS-Federation 1.2, and WSSE X.509 Certificate Token Profile 1.1 and/or WSSE SAML Token Profile 1.1 over HTTP POST over TCP.
- SOAP 1.2 with WS-Security 1.1, WS-Trust 1.4, WS-Federation 1.2, and WSSE X.509 Certificate Token Profile 1.1 and/or WSSE SAML Token Profile 1.1 over HTTP POST over TCP.

Following given an detailed overview into a generic interface binding with support for WS-* standard stack:

SOAP 1.2 with WS-Security 1.1, WS-Trust 1.4, WS-Federation 1.2, and WSSE X.509 Certificate Token Profile 1.1 and/or WSSE SAML Token Profile 1.1 over HTTP POST over TCP.

Generic service instantiation shall be supported on the following interface binding:

- **Protocol stack:**
  - SOAP 1.2 with WS-Security 1.1 and WSSE SAML Token Profile 1.1 combined and/or WSSE SAML Token Profile 1.1 with any of WS-Trust 1.4, WS-Federation 1.2 over HTTP POST over TCP.
- **MEP:**
  - SRR-MEP, PSPUSH-MEP, PSPULL-MEP
- **Fault handling:**
  - The service shall be able to determine the content of the HTTP status code and HTTP reason phrase
- **Encoding**
  - Text encoding
  - Binary encoding: MTOM
- **Security**
  - Confidentiality: optionally network
  - Integrity: message
  - Authenticity: message mutual
  - Authorization: message
  - Non-repudiation: message
- **Contract**
  - Formalism of contract description: WSDL 1.1 and optionally WSDL 2.0 both including WS-

SecurityPolicy

- Minimum: OASIS WS-N and structure of Topics
- Reference: OASIS WS-N, ISRM

- **Interoperability**
  - WS-I Basic Profile 2.0, WSI Basic Security Profile 1.1

Chapter 8. Information Security Functions

This section explains basic security, the concept of digital identities, their provision, the management of information access rights and the concept of security policy enforcement. The subject will become more important through progressing in the liberalization of the aeronautical data access, when the aviation data sources get globally connected and become accessible via SWIM.

8.1. Authentication

Access management refers to the process of controlling and granting access to satisfy resource requests. This process is usually completed through a sequence of authentication, authorization, and auditing actions. Authentication is the process by which identity claims are proven. Authorization is the determination of whether an identity is allowed to perform an action or access a resource. Auditing is the accounting process for recording security events that have taken place. Together, authentication, authorization, and auditing are also commonly known as the “gold standards of security”.

In the Access Control mechanism for Request/Reply service consumption scenario expected in SWIM, a Service Consumer shall prove its identity in order to invoke a service provided by a Service Provider. At the same time, a Service Provider shall prove its identity in case mutual authentication is required. Such Access Control mechanism is realized through combination of Identity Management, Authentication and Authorization Functions. SWIM-TI Authentication Function aims at authenticating Digital Identity: this mainly consists of verifying that a claimed identity of an entity is legitimate by evaluating additional information (authentication credentials) that is bound to this identity and can only be provided by an entity with that identity.

8.2. Authorization

In the Access Control mechanism for service consumption scenario expected in SWIM, an (ATM SWIM Enabled) Service Consumer, whose identity has been proved previously by Authentication Function, shall be authorized in order to consume a service provided by an (ATM SWIM Enabled) Service Provider.

The authorization decision making process is based on two key inputs:

1. An authorization policy which describes the required security attributes of a user allowing it to access a resource;
2. Authenticated identity (user) and its list of security attribute. SWIM-TI allows to differentiate individual members of a group and to selectively allow or deny access based on a granular set of attributes provided into the security token, realizing the so-called Attribute-based Access Control (ABAC) model.

Authorization function relies on Authentication Function (identity authentication), Identity Management Function (for security attributes assigned to that identity), on Security Policy Enforcement (PE) and Security Policy Management.

The relationship between Authorization Function and Policy Enforcement Function consists of the fact that the authorization policy is enforced by the PE which relies on the authorization function as Policy Decision Point for Authorization Policies. For what concerns the Security Policy Management, the link consists of the fact that an authorization policy can be used across security domains only if there is a proper cross security domain policies lifecycle management. It is worth noting that there is a difference between authentication and authorization policies: a policy on the level of authentication required and how to achieve it, is called an authentication policy; deciding whether a given authentication level is sufficient for access is an authorization policy.

8.3. Confidentiality

Confidentiality function is the process by which sensitive information is only accessible by granted (“right”) users. It ensures non-disclosure of information to users that do not own a given secret. This function relies on the policy enforcement and management features and it is based on cryptographic mechanisms.

Confidentiality function covers both data encryption and data decryption) and data in transit (e.g. OGC aviation services invoked by a service consumer). Data encryption is performed on service consumer side at service-request sending and by service provider side at service-response sending. Data decryption is performed by service provider side at request reception and by service consumer side at response reception. In order to interoperate service consumer and provider shall agree on cryptographic mechanisms they will use. Cryptographic mechanisms include algorithms, encryption granularity and key validity. The agreement can be dynamically negotiated at connection time or statically configured in a dedicated policy. The fact that a single Confidentiality Policy is deployed on both service consumer and service provider ensures interoperability.

The link between the Confidentiality function and the PEP consists of the fact that "data" specific confidentiality security policies are enforced by the PEP which relies on the confidentiality service as PDP for that kind of policy. This kind of policy specifies, for a given data, assertions such as if confidentiality required or not, which key schema has to be applied (symmetric/asymmetric), which encryption algorithm has to be used, which parts of the messages have to be encrypted, etc. Thanks to the combination of these policies and the cryptographic enabler it is possible to support simple and very complex confidentiality requirements.

Cryptographic algorithm can be symmetric or asymmetric. Symmetric-key algorithm uses the same cryptographic key for both encryption and decryption. A simple transformation (that could even be identity) allows getting decryption key from encryption key. The key pair is the shared secret between the two parties. Asymmetric-key algorithm requires two separate keys. One of which is private, the other is public. The private key is kept secret by the owner and is never sent in a message. The public key is used for encryption and the private key is used for decryption. The public key shall be known by any confidentiality function requiring encrypting data. The management (creation, deployment and revocation of pairs of public/private keys) is handled by a Public Key Infrastructure (PKI).

Confidentiality function can be addressed none exclusively at network level, transport level or message level.

8.4. Data Origin Authentication

Data Origin Authentication function is the process ensuring data in transit is not altered (data integrity) and that they originate from the expected sender (authenticity). Data Origin Authentication also addresses Non-Repudiation because digital signature can provide evidence that an actor has performed some operations related to data, though the degree to which an entity can be held accountable shall be established in an agreement between parties.

Data origin authentication covers both data signing at the origin and data-signature verification at the destination. It does not cover data validity that is a mechanism ensuring the data correctness in the actual context of usage. This function is based on cryptographic mechanisms enabling digital signature.

Data Origin Authentication can be realized using a combination of either symmetric or asymmetric signature and hashing techniques. Symmetric signatures is performed by using a shared secret to sign and verify the message, producing what is called a Message Authentication Code (MAC) that consists of a checksum of the original message that is encrypted using the shared key.

Asymmetric Signature is performed with a scheme involving a pair of public and private keys. One of which is used to create the signature and the other is used to verify the signature. The private key is kept secret by the owner and is never sent in a message, while he public key is generally available and can be distributed with the message but its authenticity (i.e. association between the public key and the carrying entity) shall be guaranteed by a PKI using a digital certificate allowing a message recipient to verify the private key in a client's signature using the public key in the client's certificate. Since the private key is restricted to the owner of the key, the signature is a proof-of-ownership that can be used to support requirements for non-repudiation.

Data origin authentication can be addressed none exclusively at:

- Network level
- Transport level
- Message level

As for confidentiality, the link between the Data Origin Authentication and the PE consists of the fact that "data" specific integrity and authenticity security policy are enforced by the PE function which relies on the Data Origin Authentication service as PDP for that kind of policy. This kind of policy specifies, for a given data, assertions such as:

- the level of applicability: none, transport, message, both;
- the use of symmetric or asymmetric schemes;
- information about the hashing algorithm, the type of the key (dedicated or multipurpose).

Thanks to the combination of these policies and the cryptographic mechanisms it is possible to support simple and very complex integrity and authenticity requirements.

Policy driven confidentiality and integrity improves the flexibility. It is the policy itself, assigned to a given data, that requires or not the need for (for instance) encryption. Security architecture just processes these policies therefore it is not responsible to specify which data are sensitive and which

not.

8.5. Security Audit

Security audit is the process by which security-related events are recorded for real-time or differed analysis. The audit process typically involves the following phases:

1. Generation
2. Data collection and storage
3. Analysis and feedback

The Audit function is limited to the audit generation; it allows (when needed, e.g. when non-repudiation is needed) to create, submit, persistently store and report on audit events. All these aspects are combined to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data.

The data collection and storage involves other part of the communication participant and therefore cannot be dedicated to the security architecture itself. The analysis and feedback is similarly performed by correlating security events coming from various sources.

The Audit-Function policy defines which events need to be audited and for which activities or resources.

8.6. Digital Identity Management

Identity Management is a supporting infrastructure for maintaining and administering Digital Identities within organizations and/or communities, primarily for accessing resources such as services. It is essentially “user lifecycle management,” reflecting the creation, maintenance, and deletion of identities over time, where a “Digital Identity” is meant to consist of the following parts:

1. Identifier - A piece of information that uniquely identifies the subject of this identity within a given context.
2. Credentials - Private or public data that could be used to prove authenticity of an identity claim
3. Core and Context-specific Attribute - Data that help describe the identity and can be used across a number of business or application contexts or within specific context where the identity is used.

A Service Provider shall request a valid Digital Identity in order to be able to expose either an ATM specific service or an Enabling service. A Service Consumer needs to be identified in order to be authorized or not to consume such services.

From a functional viewpoint, identity management provides the following features:

1. Secure Digital Identity administration (creation, renewing, retiring) and storing, Efficient mapping of identity to resources using a management model (e.g., role-based) and identity assignment according to an appropriate set of attributes.
2. Return Digital Identity (e.g. SAML token) from validated credential (e.g. user/password)

3. Credential validation.

Identity Management also helps to manage trust relationships between service consumers and service provider without a direct trust relationship between them, eliminating the need for each participant to independently manage their own trust relationships, as well as to have prior knowledge of one another in order to communicate.

Within each security domain the identity information are stored in an Identity Store also called Identity Registry. The Identity Management represents an abstraction layer to access to identity registries which forms a functional point of view can be seen as an entity providing CRUD operations on digital identities data.

8.7. Concept of Policies and Policy Enforcement

The security policy represents a declarative way to describe which security functions need to be applied (enforced) in order to achieve desired level of security. Therefore the policy enforcement deals with the application of policies in security systems. The concept of policy enforcement is an architectural pattern for implementation of general cross cutting concerns; within the aviation architecture, policy enforcement might simplifies per OGC service endpoint definitions and maintenance of security relevant rules of communication.

The major functions within policy enforcement are the following:

- **Policy enforcement point (PEP)** which executes policy assertions (security relevant policy functions as part of policy assertions are authentication, authorization and cryptographic operations for ensuring of integrity and confidentiality).
- **Policy decision point (PDP)**, which provides function for policy definition evaluation. This evaluation occurs in combination with available information from PIP. Finally, it provides evaluation decisions to the PEP.
- **Policy information point (PIP)** provides additional mostly attribute based information about services, policies and identities. In case of authorization policies, the PIP provides (to PDP) the authorization attributes for Digital Identities helping the PDP to allow or deny the service access.Policy repository which is a repository containing and managing policy documents.
- **Policy administration point** provides end-point for policy management. It implements policy related CRUD operations, as well as the querying, retrieving and configuring of particular service end-points.

From the Testbed-9, the OGC 09-035 three different approaches to identify communication participants and to implement access control specific for geospatial services:

- Web services security using **XACML** policies with spatial obligations and related software implementations;
- Web services security using **GeoXACML** policies and related software implementations; and
- RESTful web services security using **OpenID or OAuth** and related software implementations.

8.8. Security functions at different OSI model layers

This section explains the differences between network, transport and message based security and helps to understand the benefits and trade-offs of each security implementation approach. There are several conceptual and technical ways to implement these security aspects. Overall, transfer security might be achieved through the use of network, transport or message security:

- Securing of underlying communication channel (Network Security)
- Securing of end to end connection (Transport Security)
- Securing of data payload (Message Security)

8.8.1. Network level security

Network based security provides an approach, which is positioned at the network layer of communication stack. Such security solution is based on the underlying network infrastructure and therefore transparent for the higher application layers. Special security layer, as part of IPV6, would be one prominent example for that kind of security implementation approach.

The network based security solution is of “point to point” type. Usual scenario would include two systems which (upon they have agreed on network based communication protection) introduce special network setup. The clients and services always interact via preconfigured gateways (communication routers), which ensure confidentiality and integrity, as well as good performances. They on the other side don’t provide fine grained authentication, authorization and access control. This kind of security is strictly used to provide secure networking between two organizations. In case of intermediary communication nodes, the data exchange cannot be considered as secure (see transport based security).

The network security is recommended in the following scenarios:

- Information will not be routed through intermediate systems (end to end communication).
- Service provider and service consumer both have sufficient technical capabilities to support that kind of networking.

Using network security offers the following advantages:

- It provides interoperability for upper communication layers, meaning that communicating parties do not need to deal with security constraints on their levels of abstraction.
- It frequently results in better performances, especially during long running interaction among communication participants.
- Hardware accelerators can be used to further improve the performance. This approach is transparent for upper communication layers, as well.

Using network security has the following disadvantages:

- Security is applied on a point-to-point basis, with no provision for multiple hops or routing through intermediate application nodes. If these hops are not trusted the overall solution is not considered as fully secure.

- It is transport-dependent upon the underlying platform, transport mechanism, and security service provider.

8.8.2. Transport level security

When using transport security, the communication security measurements are performed at the communication transport layer. User credentials are transport-dependent. Transport security is strictly used to provide transport session, point-to-point security between two communication participants (service consumers and providers). If there are intermediary systems between them, each intermediate must forward the message establishing new channel connection which imposes security risks, if intermediate is not considered as trusted communication participant.

Transport level security is mostly implemented using TLS standard. It provides cryptographic protection and authentication for whole TCP (end to end) communication session. When the transport session between communication participants is established, cryptographic material (symmetric keys) will be negotiated and then used securing of session scoped communication.

Basically, the transport security is recommended in the following scenarios (similar to network based security):

- Messages will not be routed through intermediate systems (end to end communication).
- Both service provider and consumer are located in as secure considered network.

Using transport security offers the following advantages:

- It provides interoperability for upper communication layers, meaning that those layers do not need to deal with security constraints.
- It provides better performances compared to message level security, especially for long running repeated interactions whit frequent information exchange.
- Hardware accelerators can be used in order to improve the performance of message based security. This approach is transparent for upper communication layers.

Using transport security has the following disadvantages:

- Security is applied on a point-to-point basis, with no provision for multiple hops or routing through intermediate application nodes. If these hops are not trusted the overall solution is not considered as fully secure.
- It supports a limited set of credentials and claims compared to message security.
- It is transport-dependent upon the underlying platform, transport mechanism, and security service provider.

8.8.3. Message level security

When using message security, the user credentials are encapsulated in every message. This option gives the most flexibility from the authentication perspective. User credentials are transport-independent, which allows more authentication options compared to network and transport level security implementations. Different types of security credentials can also be used largely independent of transport, as long as both the service consumer and service provider supports

them. The use of message level security in a communication stack is reasonable, when message expected to be forwarded may be routed through (non-trusted) intermediate systems, such as application layer routers or message brokers.

Using message security offers the following advantages:

- It provides end-to-end security. Intermediaries (communication nodes, routers) do not break the security because the cryptographic operations for ensuring of confidentiality and integrity are applied on the application (message payload) level.
- It allows selective application of cryptographic operations, thus improving overall application performance compared to maximal solution (which would include both measurements for confidentially and integrity).
- Message security is transport-independent and therefore can be used with any transport communication protocol.
- It supports a wide set of user credentials and claims, including the issued security token that enables federated security.

Message level security has following disadvantages:

- Reduced performances compared to network and transport security (TLS) because each individual message is encrypted and signed using computation resources intensive operation of asymmetric cryptography. This problem is mitigated (on cost of increased complexity) using symmetrical cryptography.
- It makes inter-system interoperability difficult, because, opposite to network and transport security which might be implemented in hardware, it requires both the consumer and provider to maintain full scale compatible communication stacks (up to the application layer).

8.9. Symmetric and asymmetric cryptography

The IT security functions of authentication, confidentiality and integrity can be implemented using either symmetric or asymmetric cryptography. The most obvious difference among them is that while the symmetric cryptography relies on the use of identical secret key material for complementary cryptographic operations, the asymmetric approach is based on the pairs of keys, the private and the public one. Both keys are always used for security operations but only the public key is distributed to the communication participants. Where symmetrical approach provides better performances, due to less expensive cryptographic operations (depends on introduced level of security, requested cryptographic operations might be executed several times for every single service invocation), it also means less flexibility. Secret keys must be distributed to the communication participants and this distribution needs to be secured very carefully.

Asymmetric approach has gained reasonable popularity (despite its lower performances), especially when the goal is to secure interactions among decoupled systems in a very dynamic IT environment.

Having said all that, we have two additional SWIM security architectural options for implementation of low level security functions (confidentiality, integrity check):

Asymmetric approach: It has indirectly already been introduced in this document. The PKI, which is the foundation of SWIM security, relies on the concept of asymmetric cryptography.

- Alternatively, the symmetric cryptography is another architectural option. In that case two systems (consumer and provider) would use same key material on both sides of communication channel for communication securing (that is the reason to call it “symmetric”).

Both options can be used to implement network, transport or message security as they were evaluated in the previous section.

Symmetric cryptography requests the secret key distribution via trusted communication channel. Secret keys (trusted secret) might be distributed using alternative electronic or non-electronic communication channels considered as sufficiently secured. Symmetric keys might have validity period of several hours, days or even be generated without any expiration period. The share secrets and their validity range define the boundaries of security session, the security context. Symmetric cryptography based on short validity security sessions introduces more security, because the keys are reissued (and redistributed) more often. On the other side, the frequent keys distribution via dedicated trusted channels is not an efficient option for dynamic IT environments. It generates increased maintenance and administration efforts due to the need for secret keys distribution.

Symmetric cryptography based security option is combination of previous two. It specifies the compromise between shorter security context time to live and increased effort for the key distribution, which shorter validity periods would imply. In such scenario, the security session negotiation among consumers and provides is protected with asymmetric PKI and the information exchange is then secured using symmetric approach. The well-known example for such symmetric key material distribution protected by an asymmetric approach is the frequently used securing of end to end connection with TLS (transport level security). In order to establish one TLS communication session, the participant’s first exchanges security context related key material encrypted using their PKI public and private keys.

The TLS and similar methods work only for end to end connections. Considering the architectural options already listed for types of communication protection (the network, end to end and message), we additionally introduce an option for symmetric approach and message security style.

Generalizing this approach, we further refine options regarding the symmetric cryptography:

- Symmetric cryptography based on shared secret exchange protected using PKI (such as TLS for transport level security). It provides protection for both transport and message levels.
- Message payload protection implemented using symmetric cryptography with dedicated security context service. Such service creates on demand new security contexts and per context security tokens with shared secrets. Further, it distributes tokens to service providers and consumers.
- Security context federation. It is implemented by a dedicated service, which extends the validity of previously mentioned security context to the realms of federation members. This function might be part of identity management system. As consequence, the service consumer gets the “single sign-in experience”.

Second and third options listed here are specified in the WS-* based WS-Trust and WS-SecureConversation specifications. Second and third options are also envisaged in the DDS security
-

standard being defined in Object management Group (OMG).

As an explanatory example, here is the depiction of the process of security context establishing using the dedicated security context broker. The example clearly depicts the assistance of the asymmetric PKI approach during the exchange of sensible security context material.

1. The service consumer requests the establishing of a new security context for consuming a service.
2. The security context service generates symmetric key, sets time constraint on it and includes meta-data. All that artifacts are returned back in two different versions: one of them is encrypted using service provider's PKI public key and another one using service consumer's public key.
3. Service consumer receives the response and successfully decrypts security context information which was encrypted with its PKI key. Service consumer starts service invocation using received symmetric cryptography material. As part of initial service request it also forwards the encrypted symmetric key created for service provider.
4. Service provider receives service request. It first decrypts received encrypted symmetric key and becomes aware of new security context. Then, it uses that symmetric key for cryptographic processing of service request and response.

Once successfully negotiated, security context has validity for some period of time, for example, for several hours. During that validity period, the communication can repeatedly be secured with same key material, which is a quite efficient. If the key issuer services are federated, the security context could be extended to enclose additional services, even those operated by different providers, allowing service consumer to invoke all of them from within the boundary of single security session.

The choice between symmetric and asymmetric cryptography also influence the function of security authentication. The authentication among participants in an information exchange can occur either per interaction or per session. The authentication per interaction means that the authentication is executed again and again for every atomic interaction among systems, where the session based authentication will be performed only once for a number of interactions enclosed into security session scope. Considering the fact that typical security implementation based on symmetric cryptography establishes timely constrained security context (exchanging per session symmetric keys) and the fact that the authentication is always part of that process, it basically means that such approach also implements per session based authentication. The possession of session cryptographic material ensures mutual system authentication.

Chapter 9. Concepts and Outcomes from Testbed 11

9.1. Introduction

Security functions such as authentication and authorization can be put in place utilizing different standards and technical implementations, which might be applied at different levels of the ISO OSI reference model. This section provides introduction into information security taxonomy introduced as part of the Testbed-11 Common OGC Service Security.

The main objective of the Testbed-11's security initiative was to describe Common Security for OGC Web Service standards. The Common Security has been found to require describing the constraints towards the following security requirements:

Authentication: ISO 10181-2 defines all basic concepts of authentication in Open Systems: It identifies different classes of authentication mechanisms, the services for their implementation and the requirements for supporting protocols. It further identifies requirements for the management of identity information.

Access Control: ISO 10181-3 defines all basic concepts for access control in Open Systems and the relation to other frameworks such as the Authentication and Audit Frameworks.

Non-repudiation: ISO 10181-4 refines and extends the concepts of non-repudiation, given in ISO 7598-2. It further defines general non-repudiation services and the mechanisms to provide these services.

Confidentiality: ISO 10181-5 defines the basic concepts of confidentiality, identifies classes of confidentiality mechanisms and their maintenance. It further addresses the interactions of the confidentiality mechanisms with other services.

Integrity: ISO 10181-6 defines the basic concepts of integrity, identical to the Confidentiality Framework.

Security Audits and Alarms: ISO 10181-7 defines the basic concepts for security audit and alarms and the relationship to other security services.

Following list provides outcomes from Testbed 11. It address the relevance for OGC compatible services and provides some implementation hints.

9.1.1. Authentication Framework

Authentication Framework can be implemented standalone but is typically required by other frameworks such as Access Control and Security Audits and Alarms Framework. Common use cases require its implementation for enabling attribute based access control and accountability such as auditing and logging.

9.1.2. Access Control Framework

This framework could be implemented and operate standalone. If implemented without the Authentication framework, this does mean in particular that deriving access decisions must be undertaken without user information. For Web Services, the typical information available is coming from the computing environment: HTTP protocol, server hostname, request URL, service operation and parameters, date & time, requested resources, and IP address of the client. Typically, the Authentication framework is always put in place in an architecture. It provides service consumer information. Then, authorization decisions can be derived based on user information.

9.1.3. Authentication framework

In the context of OGC Web Services, the Authentication framework is relevant and has indirect implication to the calling client. Indirect means that the calling client may be able to execute the service endpoint but then the attempt of executing the actual service with provided parameters may result in access denied.

As an example, a simple implementation of the Authentication Framework might imply access control to a Web Server path: the URI to access a service. Assuming the access control framework is implemented for /protected/ogc/service then the client is either able to bind to the service or not. In other words, the client is either able to execute any operation of the service (e.g. GetCapabilities, GetMap, GetFeatureInfo, etc.) or no operation at all. This is because all service operations are part of the query parameter that comes after the URI separator “?”.

Another example relevant for the Testbed-12 aviation architecture is an access control framework implementation based on the OGC GeoXACML, an XACML extension that introduces fine-grained access control on subject attributes, service operation, as well as geospatial resource characteristics and environment information.

9.1.4. Non-repudiation Framework

Non-repudiation Framework can be implemented standalone and independent from the other frameworks. This framework ensures for transactional services that an adversary cannot repeat the execution of an approved operation. In the context of OGC Web Services, the non-repudiation framework is not relevant, as the strict concept of transactions is not supported by OGC Web Services. Even though some OGC Web Services provide a write interface (Transactional WFS-T), this interface provides create/delete/modify and not transactional operations. However, if a particular WFS-T instance shall support transactions, then the implementation of this framework does make sense.

9.1.5. Confidentiality Framework

This framework can be implemented standalone and independent from the other frameworks. The existence of a Confidentiality framework has direct implication to the calling client, as the communication with the service must meet the established confidentiality requirements.

In the context of OGC Web Services, this framework is relevant but it puts constrains on OGC OWS endpoint types, as outlined in the Testbed 11.

9.1.6. Integrity Framework

This framework can be implemented standalone and independent from the other frameworks. The existence of an Integrity framework has direct impact to the service consumer, as the communication with the service must meet the established integrity requirements.

Again, this framework is OGC relevant and it puts constrains on OGC OWS endpoint types, as outlined in the Testbed 11.

9.1.7. Security Audits and Alarms Framework

This framework can be implemented standalone and should always be implemented, as it guarantees the proper functioning of the security system and trigger administrative actions in case alarms are issued.

Linking the implementation of this framework with OGC Web Services standards is difficult as such an implementation depends on the overall security policy. If the policy mandates a security watch-dog, then this framework must be implemented. The decision to implement is independent from the choice of service type.

Chapter 10. Testbed 12 Aviation Security Architecture

10.1. Introduction

This chapter presents the extension proposal for a security architectures (as a special view on the overall testbed's Aviation Architecture).

The Testbed 12 Aviation Architecture based on brokered service access and asynchronous communication with message brokers represents an example of an integration platform in a heterogenous communication SWIM environment. That implies complex requirements for security management. It includes several building blocks responsible (posses capabilities) for information exchange (messaging), service consumption, provisioning and discovery.

Further, there is a large messaging communication infrastructure operated by FAA, which shall be interoperable with the OGC web service ecosystem (become accessible using the OGC Pub/SUB 1.0 asynchronous massaging standard). The fact that the aviation architecture deal with heterogenous message exchange patterns and many distinctive service provider and consumers (an intermediary component called broker inclusive) has the big implications on the security architecture.

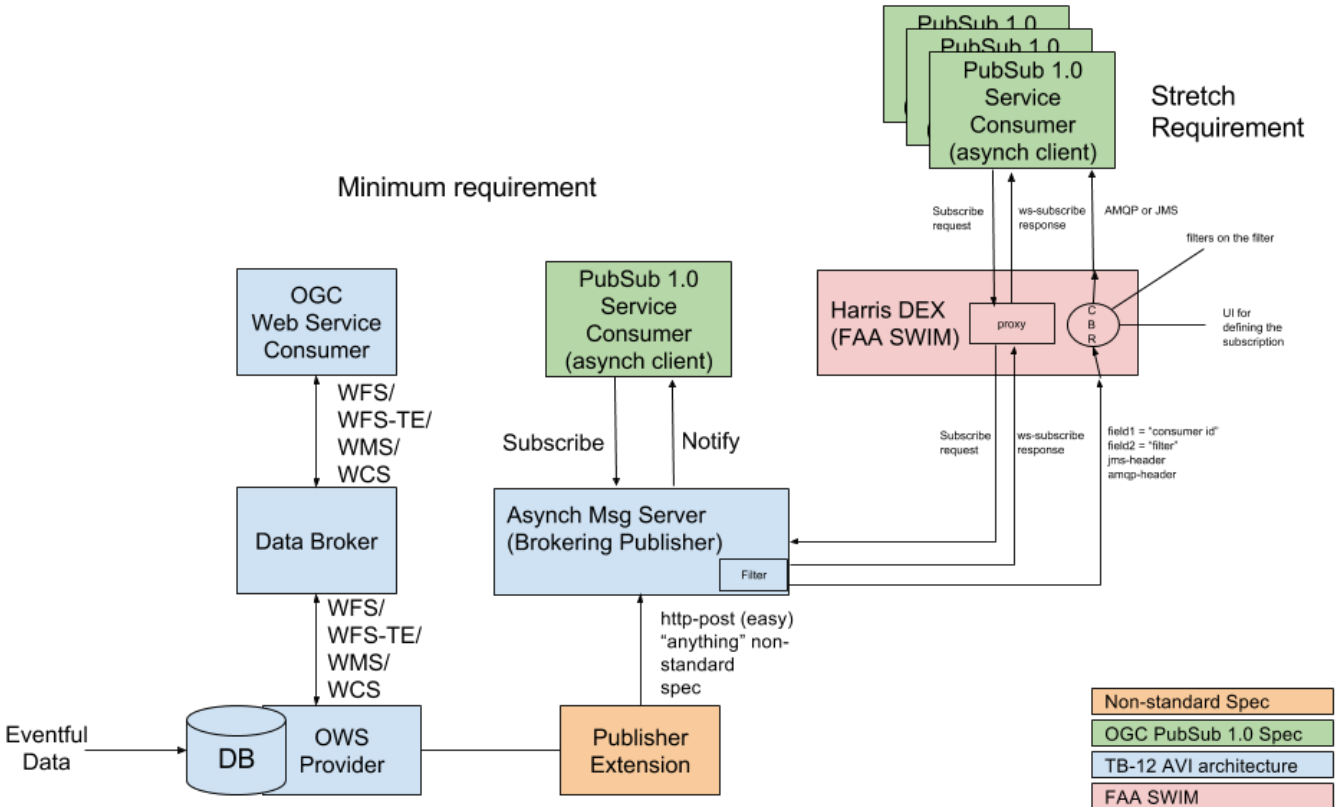

Figure 2. TB-12 Aviation Architecture

10.2. Aviation Business Background

In the global aviation business domain there is political, hierarchical structure, in which aviation stakeholders are organized and operates. In the most of cases the ANSP (aeronautical navigation service providers) maintain their own security domains and allow service consumers from own

domain to consume services. Frequently, the countries operate one single aeronautical service provider organization (ANSP). In the USA the FAA is the national agency, which operates local SWIM and a single security domain. In Europe there are a numerous national ANSP, which operate their own security domains. Additionally, there is one service pool operated by Eurocontrol, which provides aeronautical services for all member countries.

More complex communication and thus security architectures might emerge with adoption of brand new requirements for interoperability between the USA and the EU also including other countries and regions. The challenge here will be to find the ways to provide largest possible interoperability with maximum of security and minimum security threats and operational difficulties.

In order to better understand the security we will introduce the concept of communication infrastructure, which includes all logical and physical resources put in place to enable the communication. This ER will also propose an extension for the aviation architecture in order to configure it to work in a security regular fashion and also be capable to support use cases scenarios which would include communication between autonomous security domains. Such use case might be very important in the future, when many aviation security domains worldwide become required to work interoperable providing an uniform way to access data and services.

10.3. Security Architectural Options based on TB-11 Proposal

The proposed security architecture provides several options based on security guidelines from Testbed-11.

Testbed-11 security taxonomy is based on traditional information security definition which includes a set of functions such as: confidentiality, integrity, availability, non-repudiation and so on. This engineering report will propose how to implement those functions. The proposals shall primarily depends on at which communication level the security was supposed to be implemented, whether we had to deal with one single or multiple security domains and what level of trust we anticipated in the communication between aviation service providers, service consumers and the Testbed-12 Aviation Architecture communication infrastructure.

10.4. Other Security Architectural Options

10.4.1. Introduction

The objective of this chapter is to propose several advanced security related options for better extensibility and interoperability.

Security architecture proposal is based on advanced concept such as federated security (identity management). Federated security consists of rules, technology and the infrastructure put in place to allow efficient and flexible authentication of service providers and consumers in a highly dynamic, service oriented aviation architecture.

Following assumptions are considered and have influenced the advanced security architecture proposal:

- Topology of security domains (single aviation architecture security domain versus many domains)
- Application of security functions at network, channel or message level.
- Identity Management (federation, Single Sign-in)
- Advanced access management.
- Geospatial access management (GeoXCML)

In a single security domain there will be single identity management with single access control (role based access control or policy based access control with possible spatial attributes). Service providers from distinctive, multiple security domains are operated:

- Fully autonomously without any identity brokering.
- Associated into a sort of identity federation, which also may enable the single-sign-in function.

Options regarding security domain:

1. All stakeholders involved in the TB-12 aviation architecture belongs to a single security domain and every communication participant needs to be authenticated and authorized. Security functions are implemented using a common solutions implemented on the network level such as VPN.
2. Service providers operates from individual, distinctive security domains and protect communication implementing security functions on the transport level (TLS). They use PKI and certificates officially issued by authorized certificate authorities (CA). Every system maintains its own access right control and might also require additional authentication beside the option provided by PKI.
3. Service consumers and providers operate from their individual security domains and communicate implementing security functions on the message level. They protect the communication by utilizing the concepts and technology of public cryptography (PKI) with trusted certificates officially issued by authorized certificate authorities. Every service provider maintains its own access right control and might also require additional authentication beside the option provided by PKI but there is additional identity management and access control system put in place to assist whenever a service consumer wants to access some service. Service consumers doesn’t necessarily have to maintain multiple digital identities for every service they want to consume. Additionally, the additional security component dedicated to identity management allows the single sign-in.

Additional Options:

1. PKI used to secure connection between security domains but inside of every individual domain there might be a number of providers and consumers and every shall be identified individually (identity management).
2. The broker component mediates between providers and consumers securely interacts with service providers but the providers can not fully identify and capture the human actors or systems, on which behalf the broker has performed the service invocation (there is no "end to end" security). Alternatively, message based security might be used to authorize, authenticate and secure service consumption between consumers and providers and with assistance of

broker.

This section describes general architecture and design options for Identity Management and Access Control (IdM). It explains following functions:

- Identity provisioning (Identity Provider, IdP), for security provider's security realm
- Identity federation among distinctive service provider's security realms
- Authentication
- Authorization (access control)

SWIM security functions of authentication and authorization are logically assigned to Identity Management based on their dependency on digital identities during the service invocation decision making.

Following structural diagram depicts two major functions, the identity brokering, and the identity federation (authentication and authorization are not depicted, because the focus in this section was put on advnced security):

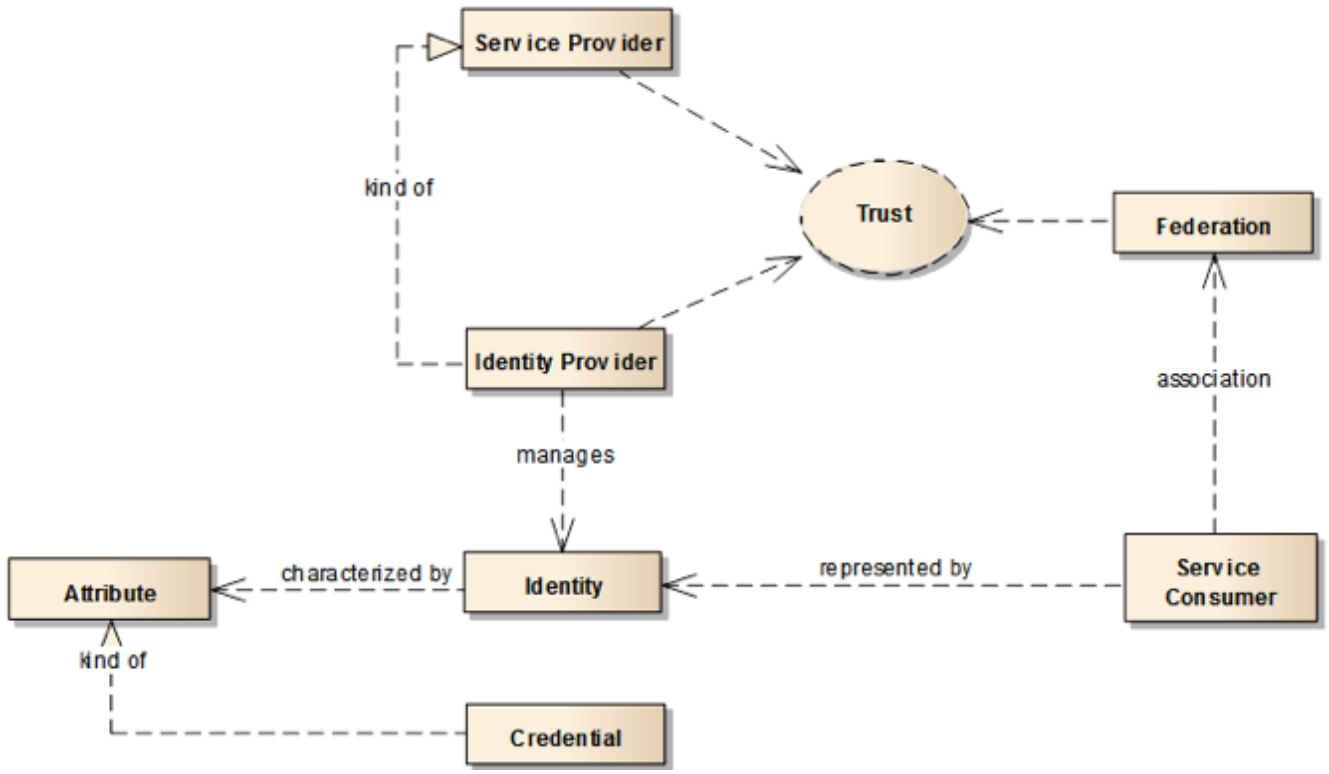

Figure 3. Identity Federation Conceptual Model

"Identity" represents an entity (service consumer) which is characterized though the set of attributes and their values. Digital identity acts as an identity proxy for real entities, represented in the diagram through the entity "Service Consumer". "Credentials" are identity attributes used for authentication and authorization. Identity instances are managed by an entity called "Identity Provider" (IdP). Identity providers, service consumers and service providers are associated through trusted relationship; which might be established on different ways, mostly using some cryptographic measurements which ensure integrity and confidentiality of issued security tokens.

Entity "federation" represents the logical concept of collaboration inside of an integration system with the goal to broker service consumer identities among distinctive service providers. This

collaboration avoids duplication of identities providing an alternative to the centralized identity management.

10.4.2. Identity Provisioning

An IdM primary provides identity brokering capability and decouples authentication and access control from service end-point implementation. Its major functions are issuing, renewing and validating of digital identities (security tokens) and digital identity brokering (identity federation). Digital identities in form of security tokens act as identity proxy for “real life” entities, for example for service consumers. Identity tokens contain data, which at some extent describes original identity (name, identity, key, group, privilege, capability) and might contain additional cryptographic content, such as public key and digital signature. Following list provides several examples for security token formats:

- Username or Identification number (plain text)
- X509 digital certificate (binary)
- SAML (XML document)
- oAuth 2.0

Identity provider (as part of identity management) provides service called Secure Token Service (STS), which is responsible to provide specific security token type instance (on service consumer’s request). STS end-point location and requested security token type might be defined in the service meta-data document stored in the service repository. If that information is not available, service consumer might retrieve service meta-data from repository and look up for the information about STS end-point and requested security token type.

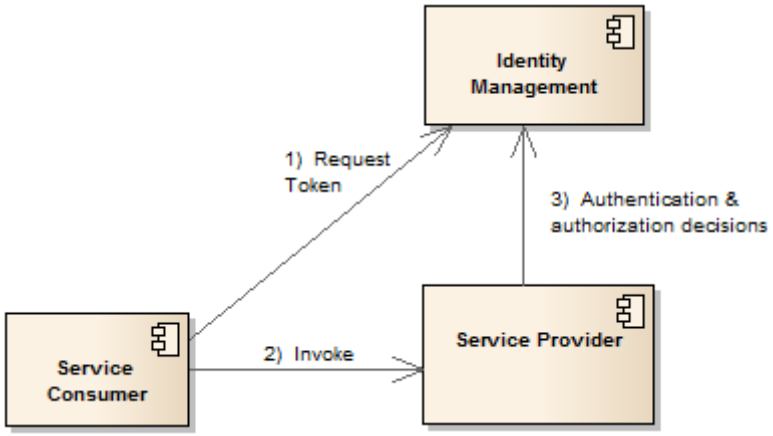

Figure 4. Identity Provider

As part of the token request process, service consumer authenticates himself against responsible security realm’s identity management and asks for particular token type issuing. Once issuing request has been initiated, the IdM returns requested security token. During the service consumption, service provider will provide issued security token as evidence of successful authentication.

10.4.3. Identity Federation

Identity Federation is as an extension of the concept of single IdP and STS. It uses dedicated services

and data types for user identity information exchange between different organizational security realms).

This type of collaboration is based on agreements established between system's identity providers. Security token issued by one security realm can be used for service consumer authentications in federated security realms. In anticipated identity federation, participating organizations (aviations stakeholders consuming and providing OGC web services) will make agreement to trust each other's consumer's authentications. That agreement will be implemented using available identity federation standards (for example, WS-Federation)

Following observation provides motivation for identity federation: In a SWIM SOA environment, as part of complex business activities, consumers invoke services offered by providers, which operate autonomous identity repositories, utilize different authorization rules and cryptographic measurements etc. For sake of interoperability the SWIM security architecture needs the concept of identity federation in order to be able to deal with following problems:

- An entity/user might be represented in distinct security realms through distinct digital identities (identity duplication).
- Consumers which consume services hosted by systems in distinct security realms, governed by distinct security services need to register themselves against every service provider's identity management system.
- Consumers have to authenticate themselves several times at distinct security services and have to locally manage issued security tokens (one for each distinct security realm). Without identity brokering there is no way to reuse already issued security token for several "cross-provider" service invocations.
- Redundant security tokens will be maintained for single physical identity representation - even if they are just slightly (syntactically) different.
- Existence of redundant identity information complicates maintenance.

The concept of identity federation for aviation architecture reduces amount of semantically equivalent digital identities maintained in the architecture and locally at the participants. It also increases system interoperability through the use of identity providers and their ability to issue and trust different types of security tokens. In the identity federation digital identities issued by one federation member can be used for authentication in any other federation member.

Security token is a sort of digital identity proof usually issued after successful user authentication. It has predefined validity period and can be (re)used several times for service consumptions as part of communication session. For example, when a service composition performed by Data Broker requires to consume several WFS and WMS based services.

Following diagrams depict two possible types of dealing with identity tokens in identity federation. In the first case, during the execution step 1, the service consumer receives security token from identity management in security realm 1. This token is not valid for service consumption in security realm 2 and because of that fact, the consumer requests realm 2 compatible security token in order to consume service provided in that realm (Step 2). As the consequence of identity federation - both identity managements trust each other's tokens, service consumer receives security token, which qualify him for service consumption in the realm 2. Now, service consumption can start (step 3)

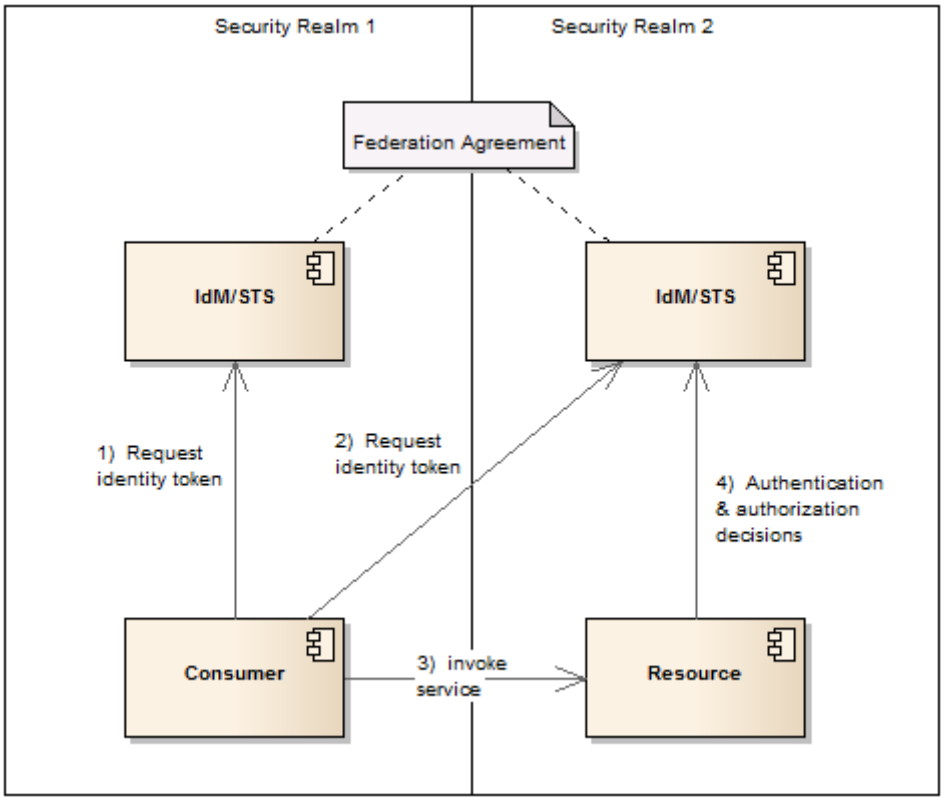

Figure 5. Federated Security Realms with Multiple Tokens

In this scenario digital identities are not duplicated across security realms. The consumer need to authenticate himself several times and to obtain a couple of security tokens, which will be used during the sequence of distinctive service invocations.

Next diagram depicts another, more complex variant, where the security token once issued in the realm 1 was sufficient for service consumptions in the security realm2. There is no need for explicit security token issuing because identity providers are instructed to trust each other and to mediate between distinctive digital identity representations.

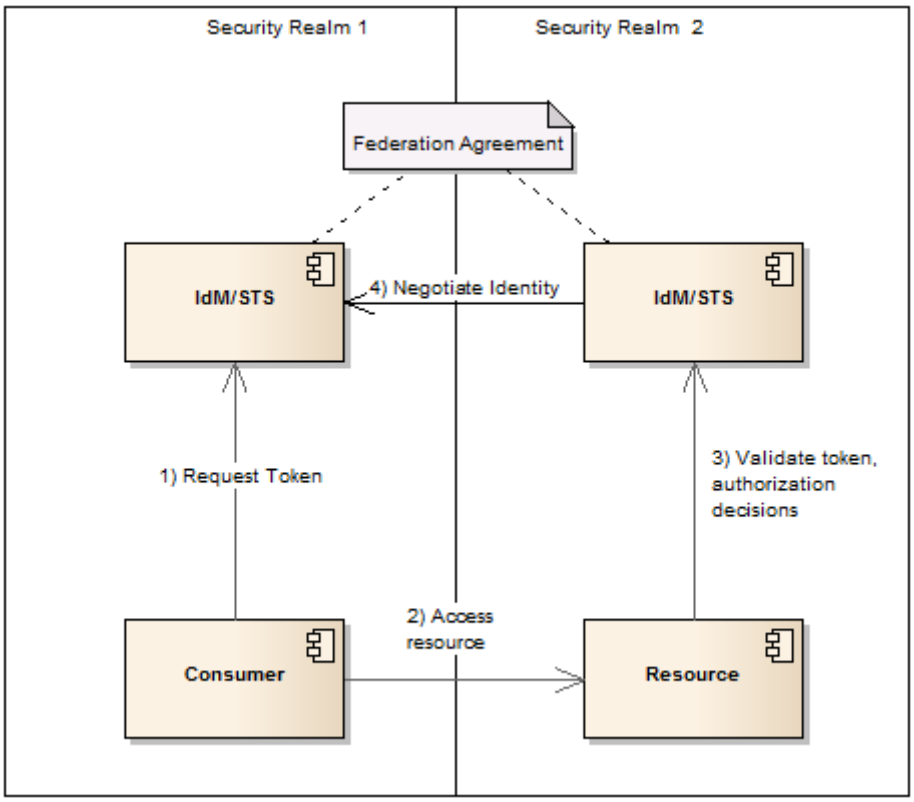

Figure 6. Federated Security Realms with Single Token

Security token issued in the step 1 is sufficient for service invocations in the security realm 2. During invocation, service end-point delegates authentication decision to IdM (Step 3). If IdM from realm 2 participate in federation, it will be able to authenticate security token issued in realm 1 using federation agreement rules or explicitly asking for authentication (as depicted in the Step4). Now, service consumers don't need to request security token for every particular security realm - they work with single token, as long as their identity managers trust each other.

10.4.4. Authorization and Authentication

Authentication and authorization are already described as security functions. In this section they are considered as part of identity management and access control because they both are dependent on digital identities (security tokens).

10.4.5. Identity Management Use Case Scenario: The Pre-flight Briefing Service

The pre-flight briefing generates on request a pre-flight briefing document. This document contains (among other information) flight data such as the route, relevant digital NOTAM messages and meteorological information. The briefing service is as an aggregated service composed of several distinct atomic services hosted inside of provider's security realms with autonomous identity management solutions. For this use case, we will make two assumptions about briefing service design:

- The pre-flight briefing service (BS) is data mash-up service executed inside of Testbed-12 Aviation Architecture's component Data Broker. It is an example for broker's capability to preform service composition

- The identity of an (human) actor, on which behalf the composite briefing service retrieves data is relevant for security. That means, the Broker, the NOTM repository and the Network Manager have to be fully aware of end user's identity. This implies that identification of involved/interacting systems would not be sufficient for this use case.
- For sake of simplicity, we omit other data retrieval steps such as the meteorological service invocation.

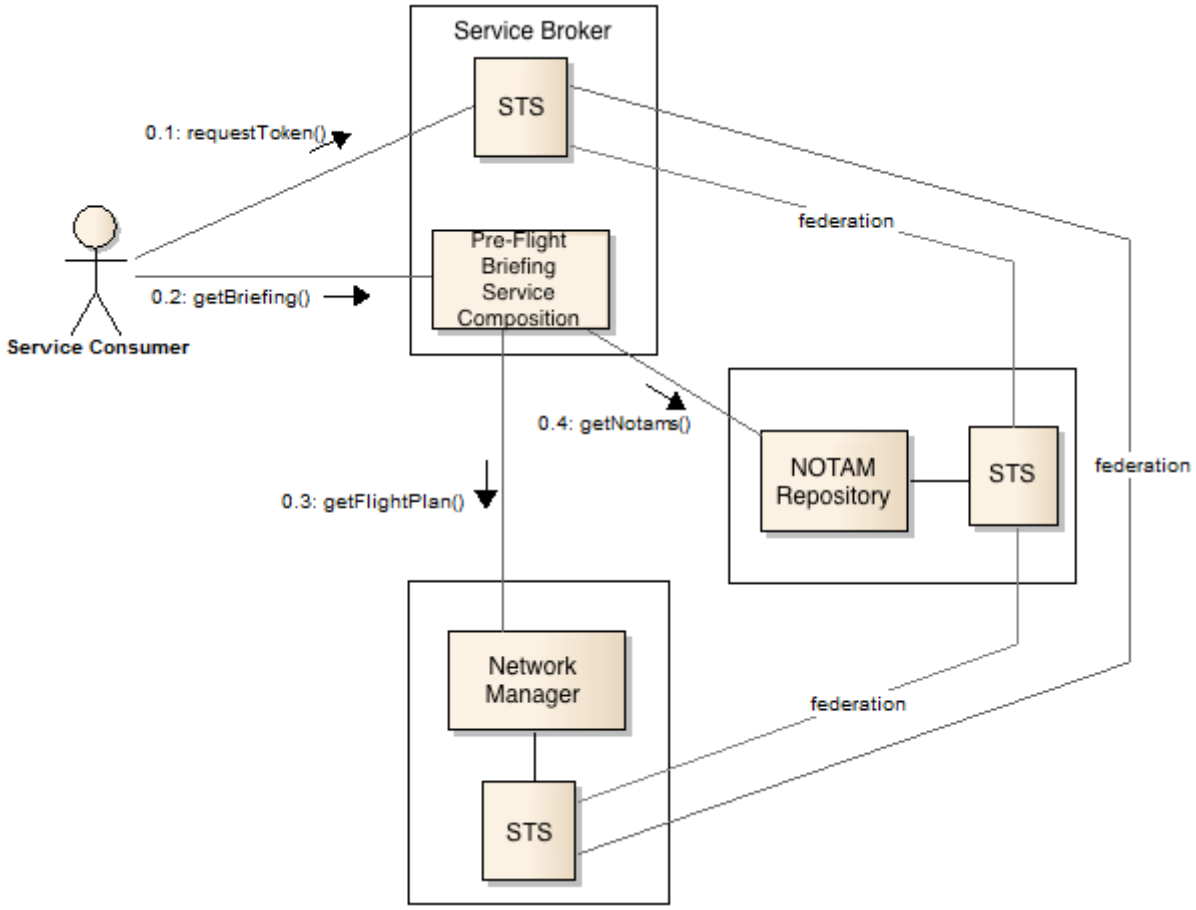

Figure 7. Pre Flight Briefing Service

Communication Diagram Execution flow:

1. Service requestor (for example, human actor participating in the process of flight planning) performs authentication against Service Broker's IdM invoking security token request. Authentication is executed successfully. IdM issues requested security token.
2. Service requestor invokes the briefing composition providing security credentials in form of security token. As part of composite service execution, two atomic services will sequentially be invoked during following two execution steps.
3. Briefing service composition execution starts. Because of federation agreement; the local IdM is able to identify identity credentials previously issued by original IdM.
4. NOTAM service execution begins. Because of federation agreement; the local IdM is able to identify identity credentials previously issued by Service Broker's IdM.

After successful data retrieval, partial results are merged into final briefing document and returned back to the service requestor.

10.4.6. Architectural Options for Identity Management

There are several options to design, configure and operate identity management. Following list provides the key aspects:

- Identity management configuration and responsibilities (who is managing and operation the repositories with digital identities and their attributes)
- Identity Resolution (which part of system will be identified when service consumption occurs)
- Identity Federation (how the sharing of identity information among distinctive security realms is organized.)
- Security Token Type (which concrete formats of digital identities)

Identity Management Configuration

From the IdM configuration point of view, we would have following architectural options:

- Identity management common service is responsible to provide digital identities for all systems and principals.
- Per system based IdM; every single organization consuming and providing services maintains its own IdM.
- Mixed solution. Some systems delegate Identity management to the architecture where other maintains their own solutions.

Identity Resolution

From the identity resolution point of view, the identity resolution (and therefore the authentication) might be:

- Based on identities of interacting systems: systems are not able to resolve identities of entities, on which behalf the collaborations were executed. The only resolvable identity is identity of direct service invoker. If system A consumes service provided by system B, which, during the service execution consumes service provided by system C, the system C is not aware of collaboration initiator's identity, the system A.
- Identification based on principals, on which behalf the inter-system collaboration was initiated.

Both types of authentication (system, entity) could be utilized for particular service types. Desired participant's level of authentication has to be described in the service registry.

Identity Management Federation

From the identity management federation point of view, we have following options:

- Local identity management solutions (at consumers and providers) are not federated. Identities are not cross-usable.
- Certain level of federation between systems. Several identity federations are established based on different technical and operational requirements.
- Aviation Architecture provides infrastructure service for federation support. Such service

would contain repository for federation meta-data and mediation between distinctive identity representations.

Regarding the implementation, the federation might be implemented using one of following options:

- Fully implemented using PKI infrastructure and certificate authorities. In that case, the certificates would be used as security tokens and the CAs would act as external identity federation service, which establishes trust among federated systems.
- PKI with service and security token types as defined in the WS-Trust and WS-Federation standards.
- Solution based on other available standards such as oAuth2.0.

Security Token Types

Collaboration among providers and consumers may require different types of digital identities (security tokens).Therefore, it is reasonable to assume that some end-points might specify usage of PKI digital certificates, where others might request XML based tokens like SAML. Type of digital identity shall be published as part of service end-point meta-data description.

- Aviation Architecture with all involved systems and services will use unique security token type. That could be for example a (PKI) digital certificates signed by some trusted CA. Such token contains attribute/value pairs, which contain entity name organization, country or origin etc. They are used for authentication and authorization.
- Text/XML based security tokens such as SAML (possibly combined with XACML). Such tokens usually contain more attributes and allow fine grained service access control. In order to ensure integrity and trust, they have to be combined with PKI (An IdP trusts token issued by another one, when it trusts its PKI signature).
- Without preferable token type.

10.4.7. Authentication using a federation of Certification Authorities

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption.

If security inside of Aviation Architecture isn't handled by single certification authority nor even by a single hierarchy of certification authorities, the objective is to build a system capable of federating existing CAs. This is important because the organizations are deploying Public Key Infrastructures (PKIs) to support internal business processes, implement virtual private networks, and secure their assets. In addition, most of them have partnerships with other organizations. If these alliances wish to collaborate, connection of their PKIs will be required. However, those PKIs may implement different architectures, security policies and cryptographic suites. A flexible mechanism is needed to link these PKIs. The USA and European aviation stakeholders in SWIM will be exactly in a same position when they start to design and deploy interoperable SWIM solutions.

Hierarchical structure

Isolated CAs can be combined to form larger PKIs in two basic ways: using superior-subordinate

relationships, or peer-to-peer relationships. A PKI constructed with superior-subordinate CA relationships is called a hierarchical PKI. In this configuration, all entities trust the same “root” CA. That is, all entities of a hierarchical PKI begin certification paths with the “root” CA’s public key. In general, the “root” CA does not issue certificates to entities but only issues certificates to subordinate CAs. Each subordinate CA may issue certificates to entities or another level of subordinate CAs. In a hierarchical PKI, the trust relationship is only specified in one direction. That is, subordinate CAs do not issue certificates to their superior CA. This is the typical structure of a PKI illustrated in Figure 7.

This configuration is quite easy to put in place and administrate provided it is deployed in a single organization. The hierarchical nature of the business organizations fits quite well with this particular configuration. The easy discovery and validation of the certification paths is another advantage of the hierarchical configuration.

But the hierarchical configuration is not very flexible. It is an excessively cumbersome process to include an external PKI in an existing hierarchy. The external CA has to be declared as a new subordinate CA in the existing hierarchy meaning that:

- Either all the existing certificates issued from the external CA have to be revoked and renewed,
- Or old certificates are kept valid and the certificate validation has to use the former root CA or the new one depending on the date of issuance.

The hierarchical configuration is also fragile. It relies on a single trust point. As a consequence the compromise of root CA results in a compromise of the whole PKI.

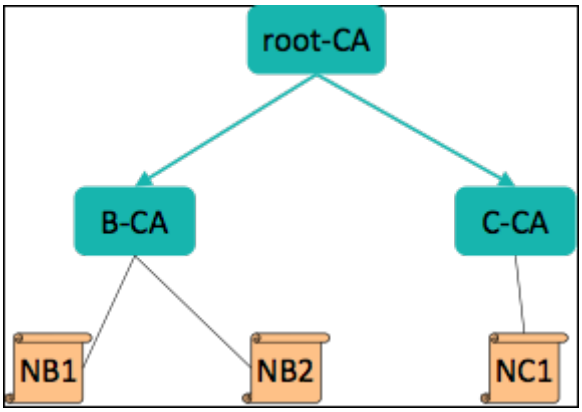

Figure 8. Hierarchy of Certification Authorities

“Cycle of trust” structure

The traditional alternative to a hierarchical PKI is to connect CAs with a peer-to-peer relationship. A PKI constructed of peer-to-peer CA relationships is called a “cycle of trust” as shown in Figure 8. CAs issue certificates to each other; the pair of certificates describes their bi-directional trust relationship.

Unlike hierarchical PKIs, “cycle of trust” PKI can easily be extended. Any one of the CAs in the cycle simply establishes a trust relationship with the external CA by exchanging a pair of cross-certificates. The whole set of CAs is more resilient. In case of compromising of one CA, the other CAs have simply to revoke certificates they have issued to the compromised CA. On the other hand, certification path building is more complex than in a hierarchy. Unlike a hierarchy, building a

certification path from an entity certificate to a trust point is nondeterministic. This makes path building more difficult and the path validation much longer since there are multiple choices.

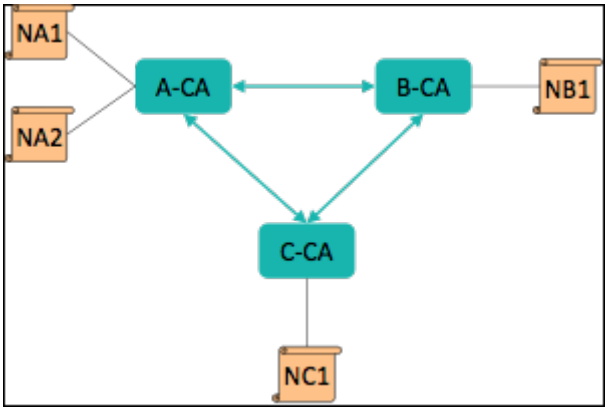

Figure 9. "Cycle of trust" of Certification Authorities

Bridged structure

Another alternative is to hierarchical PKI is to introduce a bridge Certification Authority. The Bridge CA (BCA) architecture was designed to address the shortcomings of the two basic PKI architectures, and to link PKIs that implement different architectures. Unlike a "Cycle of trust" CAs, the BCA does not issue certificates directly to entities. In addition, the BCA is not intended to be used as a trust point by the entities of the PKI, unlike the "root-CA" in a hierarchy. The BCA establishes peer-to-peer trust relationships with the different entity communities, which solves potential political issues between organizations and allows the entities to keep their natural trust points. These relationships are combined to form a "bridge of trust" enabling entities from the different communities to interact with each other through the BCA with a specified level of trust.

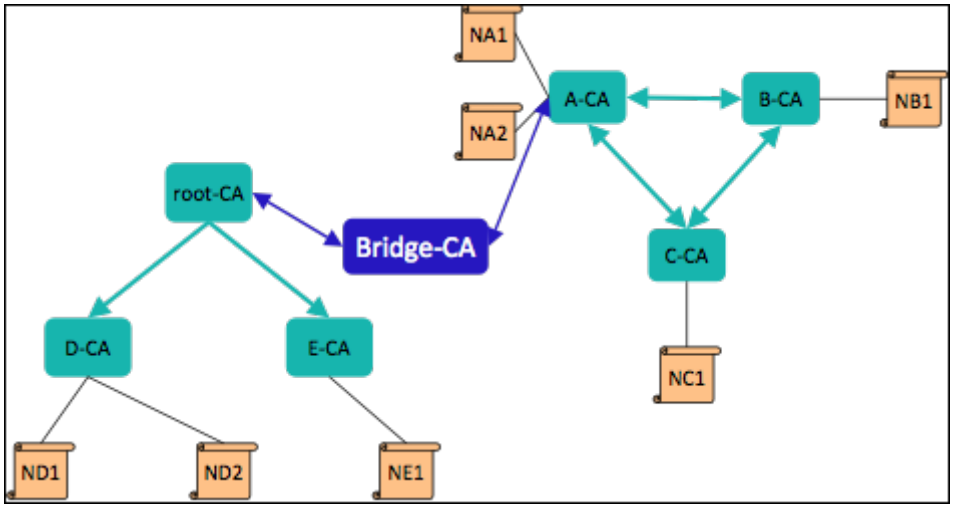

Figure 10. Bridge CA architecture

If a community implements a trust domain in the form of a hierarchical PKI, the BCA will establish a relationship with the PKI's "root" CA. However, if the community implements a trust domain by creating a "cycle of trust" PKI, the BCA need only establish a relationship with one of the PKI's CAs. In either case, the CA of the PKI that enters into a trust relationship with the BCA is termed a principal CA. In Figure 9, "root-CA" and "A-CA" are principal CA in the bridged configuration.

In comparison to a "cycle-of-trust" PKI, certification path discovery becomes easier in a bridge-configured PKI. Each entity typically knows its path to the BCA; it needs only to determine the path

from the BCA to the other entity certificate. In addition, a bridge-configured PKI will typically have shorter trust paths than a random “cycle-of-trust” PKI with the same number of CAs. Certification path discovery is still more difficult than in a hierarchy, and the typical path length is approximately doubled. However, the decentralized nature of a bridge-configured PKI more accurately represents the real world of organizational relationships.

10.4.8. Security policy enforcement in Aviation architecture

Introduction

This section provides options for design and implementation of security in aviation domain based on the declarative concept of policies and policy enforcement. The aim is to show how to implement “separation of (security) concerns” in SOA and ensure execution of security rules declaratively expressed using well structured policy documents. Rather than define the procedural sequence of implementation for particular functions, this concept includes declarative rule definitions (policy) and common purpose execution engine (which executes policy enforcement). The section also deals with options for policy implementation and management provided by infrastructure.

A policy is a set of rules that govern the behaviour of something. Major concepts of policy management are:

- Policy Definition
- Policy Enforcement
- Policy Monitoring

Policies are business rules constructed out of set of declarative statements made to specify actions executed upon occurrence of some kind of (business) event. For example, the communication among stakeholders might be constrained based on requestor’s identity, authorization rules and the types of data participants attempt to exchange.

At the highest level, some policy usually defines general enterprise concepts in a human readable textual form. Later on, they will be refined and contained in some sort of machine-readable policy form. In this particular case, security functions implemented via policy documents are executed independent from the service implementation, which makes them more robust, independently adaptable and configurable.

The infrastructure components, which deal with evaluation of policies, we call the policy enforcement points (PEP). Typical policy document consist of a set of atomic rules called policy assertions. They are usually stored in a dedicated policy repository. As already mentioned, the different level of granularity and abstraction might come into consideration for policy definition.

Following figure depicts a process of service consumption with security functions implemented using the concept of policy enforcement. By arrival of message, the policy enforcement machinery will start execution which is managed by policies defined for that service. The concept is called interception of communication on arrival of some incoming message. The specialized low level security functions (for example for cryptographic operations on the message payload) will be invoked on demand in order to support execution of policy assertions:

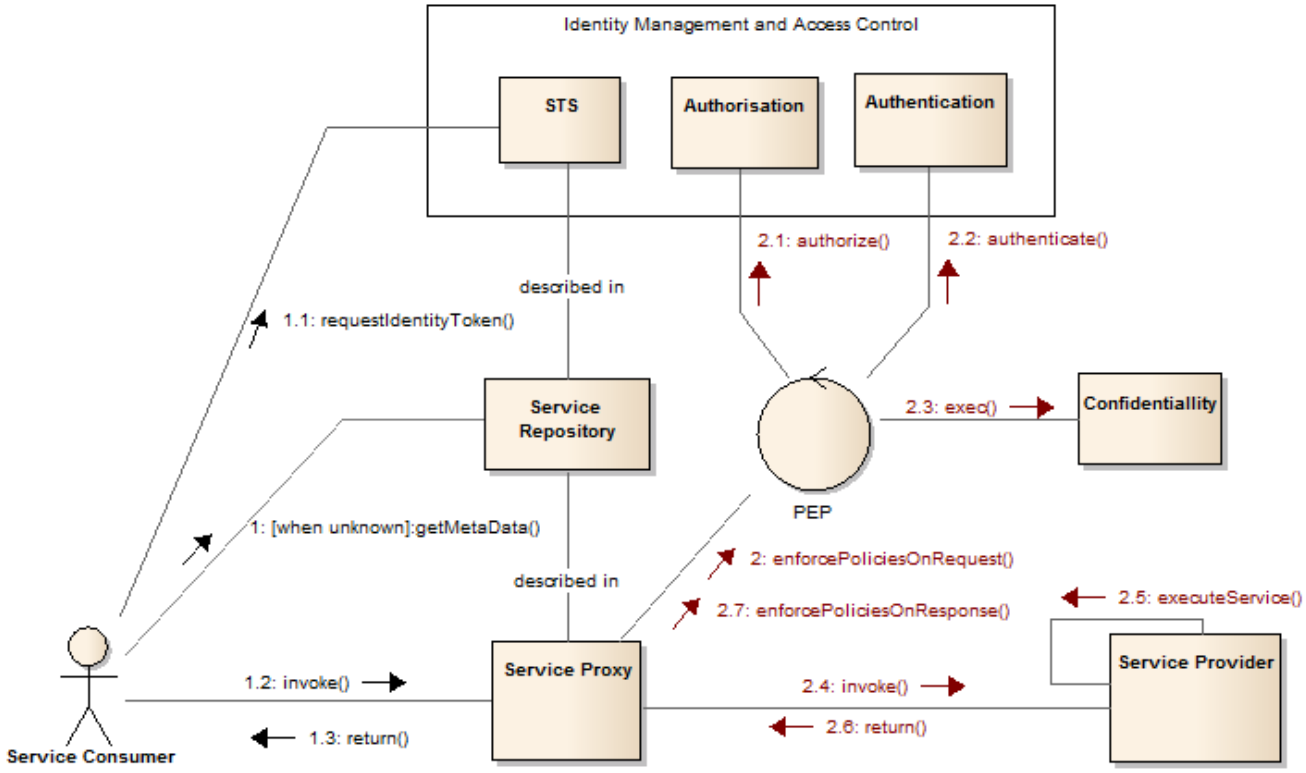

Figure 11. IdM and (Policy) Security Enforcement

Policy Document Definition Standard

Currently, the only one policy definition standard available for aviation architecture, the WS-Policy. This is the XML based taxonomy for general policy definitions. It is designed to be common purpose policy format to encode any kind of policy rules. WS-Policy compatible documents consist of a set of assertions which are combined using conditional expressions. Policy dialects for concrete functionality (for example for enforcement of some authentication rules) are then derived from general policy definition. Policy enforcement engines have to be designed and implemented to understand particular policy dialect or they must be extensible. That is because the WS-Policy defines only very high level common concepts and doesn't prescribe how security or logging assertions will to be interpreted by an execution engine.

Options for Security Policy Enforcement

Several options can be considered regarding security policy management. The major difference is in how much assistance the architecture (deploying dedicated components) might offer to service consumers and providers:

- Aviation Architecture does not provide any infrastructural components (services) for maintaining and execution of security policies. The policy definition, enforcement and monitoring are fully provided by enabled service consumers and providers. Policies are therefore beyond the boundary of TB-12 Aviation Architecture.
- The architecture provides technical means to enforce and monitor the security policies but the policy definition remains responsibility of the stakeholder providing the services. That means provision of such SWIM (common) services, which enforce security policies and finally forward service consumption call to the service end-point.
- The architecture provides technical means to enforce and monitor the security policy, as well as

common security rules that are defined at European level. It does not prevent the stakeholder to define its own rules that are locally enforced.

As far as architecture is concerned it means that:

- An common service shall be designed to support policy enforcement
- A policy taxonomy specification would be needed to express particular security needs and security functions, which need to be executed during the service invocation
- An optional component shall be considered for supervision/monitoring.

Architecture of Security Policy Enforcement

In order to depict all relevant parts of a common purpose policy enforcement system, we reuse the block diagram introduced in the XACML specification. It defines technology agnostic policy enforcement system. Later on, we will provide other options.

The concept of policy enforcement is an architectural pattern for implementation of general cross cutting concerns; in this section the security related functions are relevant. Having said that, we Within an SWIM Node, policy enforcement simplifies per service end-point definitions and maintenance of security relevant rules of communication.

The security policy enforcement executes:

- Authentication
- Authorization rules evaluation
- Confidentiality and integrity
- Security related auditing (logging, recording)

Following sequence diagram provides policy enforcement execution schema independent on whether it occurs in the aviation architecture or within the boundary of some service consumer/provider:

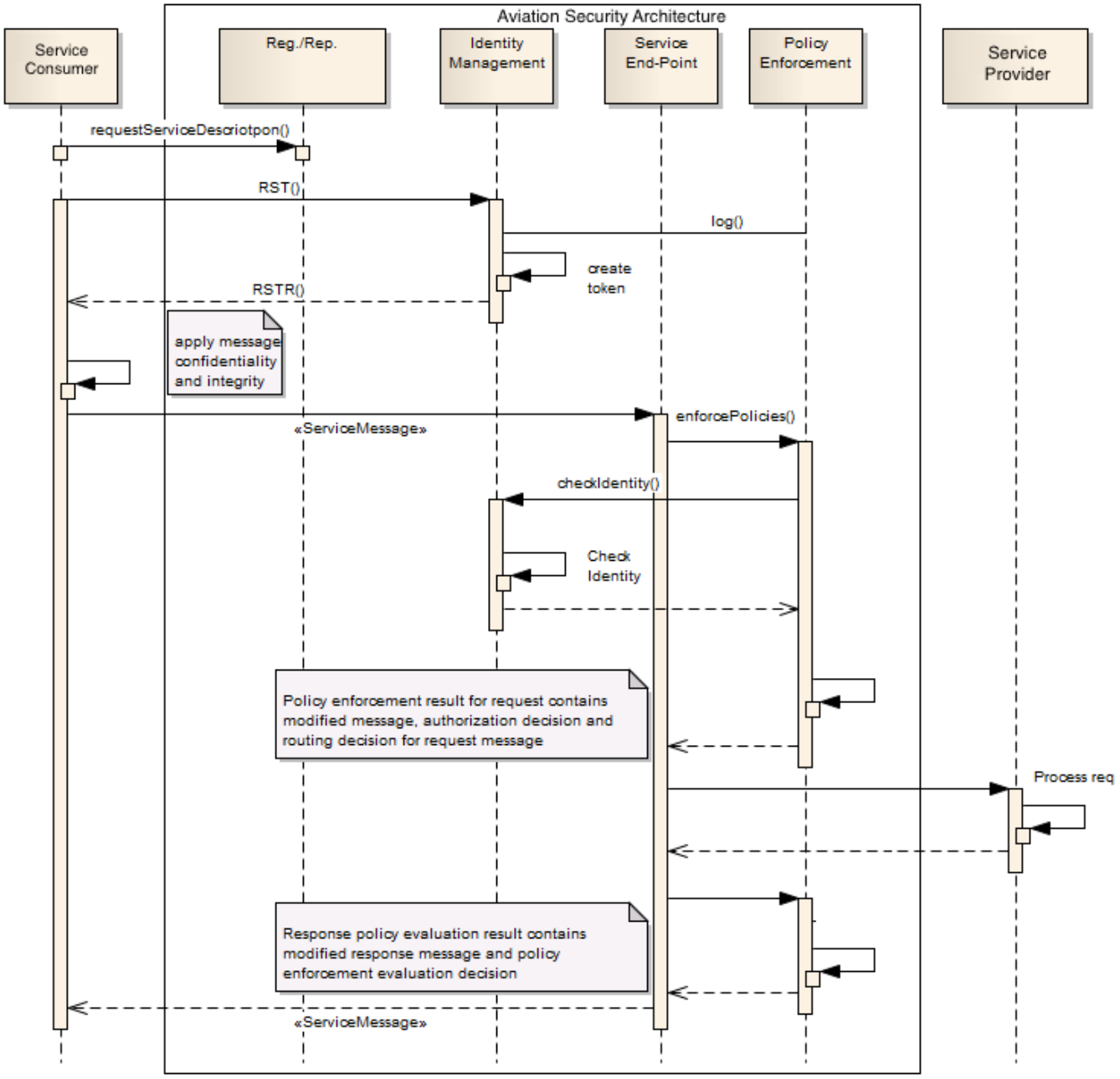

Figure 12. Policy Enforcement Sequence Diagram

The concept of policy enforcement according to XACML specification is given bellow. We identify major components, which actively participate in the policy enforcement:

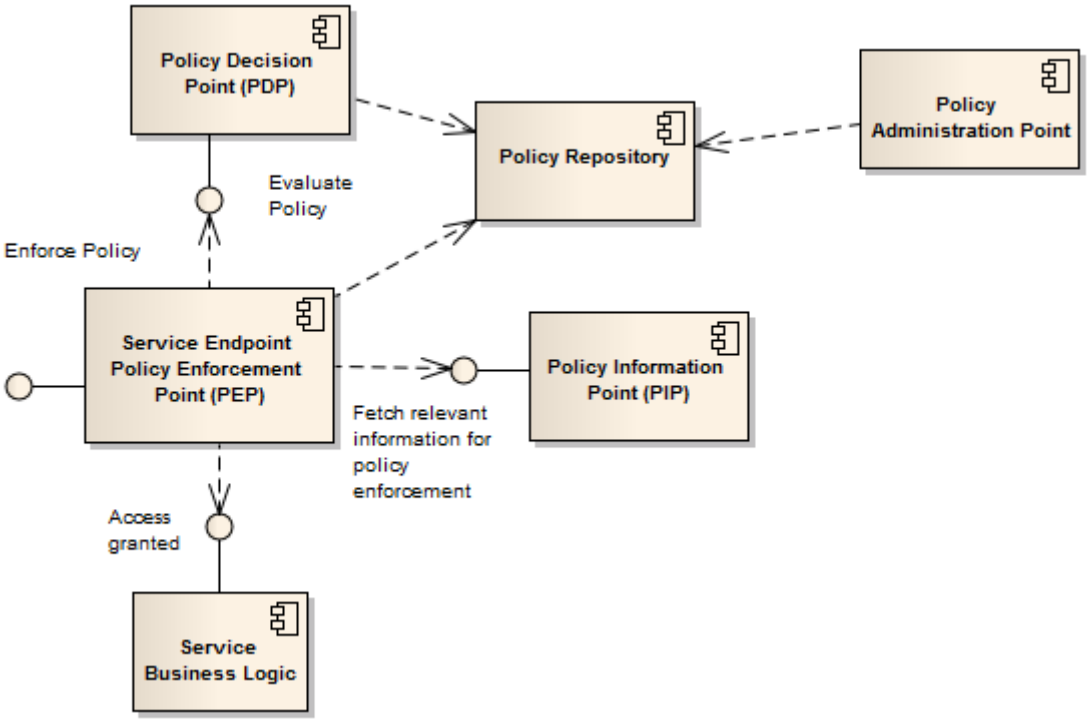

Figure 13. XACML Access Control Architecture

As depicted above, the major components are:

- Policy enforcement point (PEP) which executes policy assertions (security relevant policy functions as part of policy assertions are authentication, authorization and cryptographic operations for ensuring of integrity and confidentiality).
- Policy decision point (PDP), which provides function for policy definition evaluation. This evaluation occurs in combination with available information from PIP. Finally, it provides evaluation decisions to the PEP.
- Policy information point (PIP), which provides additional mostly attribute based information about services, policies and identities. In case of authorization policies, the PIP provides (to PDP) the authorization attributes for digital identities helping the PDP to allow or deny the service access.
- Policy repository which is a repository containing and managing policy documents.
- Policy administration point provides end-point for policy management. It implements policy related CRUD operations, as well as the querying, retrieving and configuring of particular service end-points.

Identity Provider and the Concept of Access Control

Access control belongs to basic IT security functions. While the term generally refers to the control of arbitrary resource consumption, in the SOA based systems and therefore in the SWIM, it always refers to service access control.

The authorization as part of access control is tightly coupled to the security function of authentication. Where authentication deals with the process of service consumer identification based on the concept of digital identity, credentials and the proof of possession, the authorization (or access rights management) provides the way to specify and evaluate whether potential service consumer, represented through its digital identity is allowed or not to execute certain service.

Generally, the access control implementations are based on declaratively defined authorization rules applied on the context of service consumption (service consumers and provider authorization attributes, as well as the message header and payload).

The access control is frequently implemented as based on role attributes and their values, as RBAC (Role Based Access Control). An alternative would be the more flexible approach based on authorization security policies and policy enforcement.

RBAC is traditional access control mechanism. It is implicitly declarative approach because it is always based on the same schema: the set of textual defined attributes/roles are assigned to digital identities or service end-points. The access control is performed based on evaluation whether an identity's role set matches specific service's role set. It does not include content based access control. The role/user/service associations are either stored in some dedicated storage (for every digital identity, there is a set of roles assigned to) or encoded directly into digital identity payload. For example, the X.509 PKI certificate might under certain conditions contain the roles encoded into one of certificate's predefined attributes. This option also would include general, mandatory role specification, as well as the specification of how to encode the roles inside of X.509 certificates.

Another way to organize the access management for aviation architecture is to introduce an extended security policy language with richer access control semantic. The access right policies might be considered as part of security policies. Such policy's access control assertions can also be evaluated based on the information payload content. Besides designing own authorization policy standard for the SWIM, the usage of WS-Policy, as well as the SAML and XACML from the WS-* standard stack gives the framework for identity and policy rule definitions, as well as the rule enforcement procedure. The figure 24 already depicts the policy enforcement scheme defined for XACML as an example for common purpose policy enforcement.

Third option introduces the concept of "federated access control". Basically, it is part of identity federation but might also be considered as dedicated service. Similar to the mapping of digital identity representations among distinctive security realms, we also need to map/broker authorization definitions and access control rules. Different digital identities used to represent same physical identity might have different access control attributes and use different semantic for organization's (consumers, providers) security domains. If the identity federation is the goal, authorization rules and identity definitions have to be considered for mapping, as well. The WS-Federation specification provides the specification for "attribute mapping service" as part of overall identity federation solution.

10.4.9. Extending the architecture to protect OGC OWS services

The section introduces access control specific components according to the XACML and GeoXACML. The components are the enforcement point (PEP) that intercepts the communication between the OGC WFS Client and the Web Feature Service. Login and authentication of user identities is provided by the Authentication component. For the proposed architecture, the WFS Client is not required to be modified. The major component of the access control is the decision point (PDP) that derives the authorization decisions based on GeoXACML policies. This architecture is shown in the following figure, where the standard stakeholders (in the aviation architecture those were the data broker and WFS services) are portrayed blue and the additional components required for establishing the access control are colored in yellow.

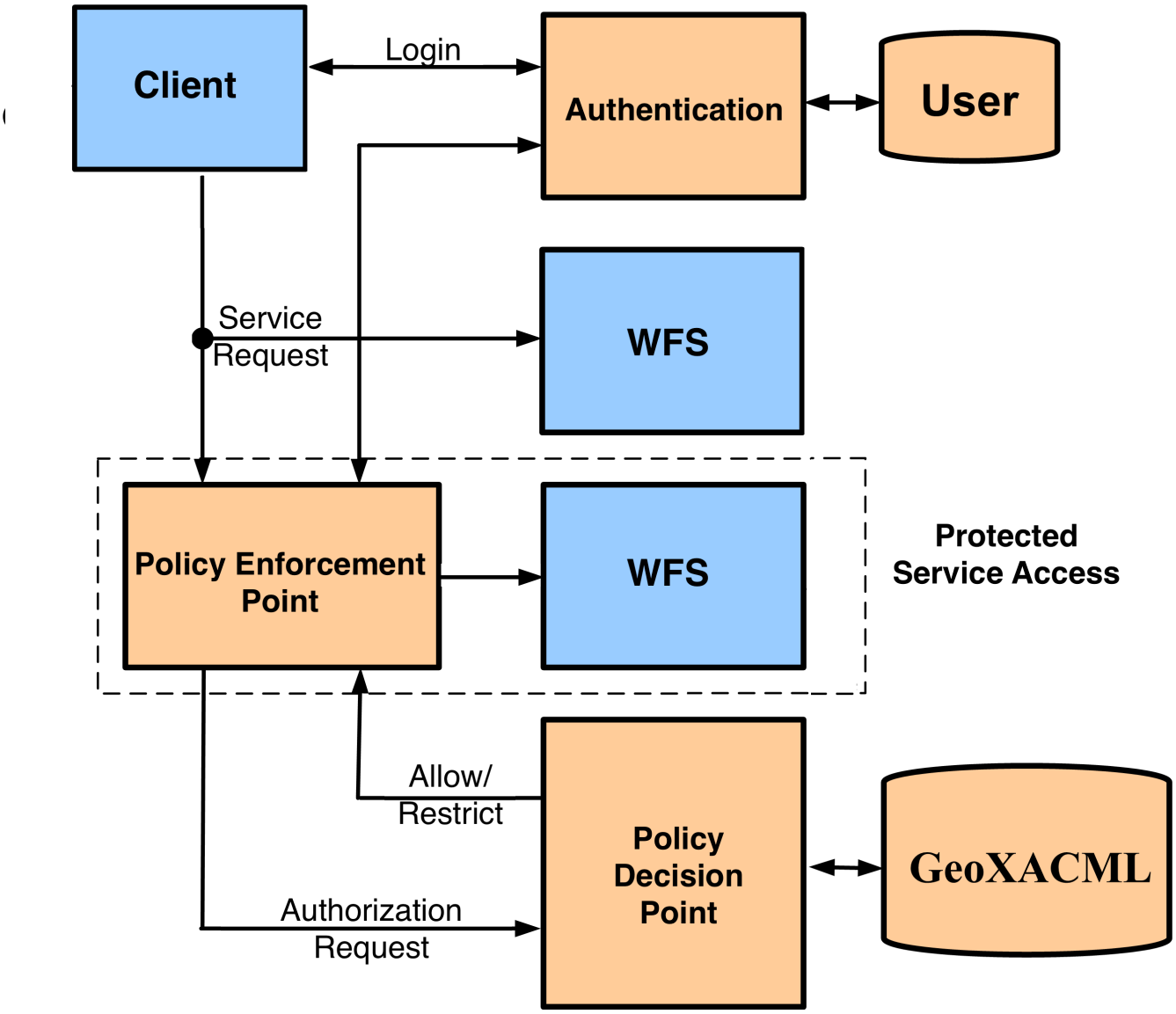

Figure 14. "GeoXACML as part of Architecture

One of goals for the proposed solution was not to influence the standard OGC interfaces at the client side. The solution for transporting the required identity from the WFS client to the PEP is based on authentication protocol SAML and its artifacts. In case that the Data Broker is required to implement security functions for service providers, it would implement and operate all these components.

Following listing provides an example for simple GeoXACML assertion. Such assertions are used to specify service or data access conditions and are evaluated by PDP component. Assertions and policies are specified either for users or data and services.

GeoXACML assertion example

```
 <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
  <Function FunctionId="www.opengis.org/geoxacml/1.0/function#within"/>
    <AttributeValue DataType="http://www.opengis.net/gml#polygon">
      <Polygon gid="Area-X" srsName="srs">
        <outerBoundaryIs>
          <LinearRing>
             <coordinates>3 0,6 1,6 5,1 5,0 2, 3 0</coordinates>
          </LinearRing>
        </outerBoundaryIs>
        </Polygon>
     </AttributeValue>
 <AttributeSelector RequestContextPath="//am:Intersection/am:location"
DataType="http://www.opengis.net/gml#point"/>
</Condition>
```

Chapter 11. Conclusions

For the concrete case of aviation architecture in the testbed 12 this engineering report has provided an overview of and proposal for the security architecture based on the previous engineering reports dedicated to this topic. The architecture originally doesn't specify any concrete security requirements. Based on some concrete applications (FAA SWIM and SWIM in Europe) a couple of preconditions, constraints and requirements have been made in order to distinguish between several architectural options.

OGC OWS services used in aviation don't differ from any other type of public web services based on GET/POST HTTP (+ SOAP). However, the HTTP POST or SOAP are recommended form the security perspective. As the spatial services they might have additional access right control based on geospatial attributes (for example a certain service consumer can request only those flight plans, which corresponds to the flights executed within his/hers domain of responsibility).

Considering the potential numerous aviation service providers, different message exchange patterns and possible multiple security domains, as well as the existence of intermediary communication components such as the broker we came to conclusion that the most efficient solution would be based on the VPN but it is not clear weather such architecture would be possible in the case of multiple organizational domains (for example in case of global SWIM integration). On the other side, the proposal which include brokered identities and security functions applied on the message level represents the most flexible solutions with a trade-off in complexity and possibly security.

Application of security functions on the transport level, such as the TLS seems to be the compromise between above two extremes. Using the certificates from the PKI authentication and confidentiality could easily be established but solution lacks in end-to-end security. Still, this option is the most frequently used in the publish internet and fits well with OGC OWS service endpoints based on HTTP POST approach (for end-to-end security, OGC service endpoints must be encoded using SOAP)

Appendix A: UML model

The figure gives an overview in WS-* security standards applicable for the Aviation Architecture with OGC OWS services implemented using SOAP compatible endpoints. The WS standard stack supports both basic and advanced security federation inclusive and is applicable for all kinds of OGC compatible web services. The diagram below depicts at which communication layer the security standards are applied and how they relate one to another.

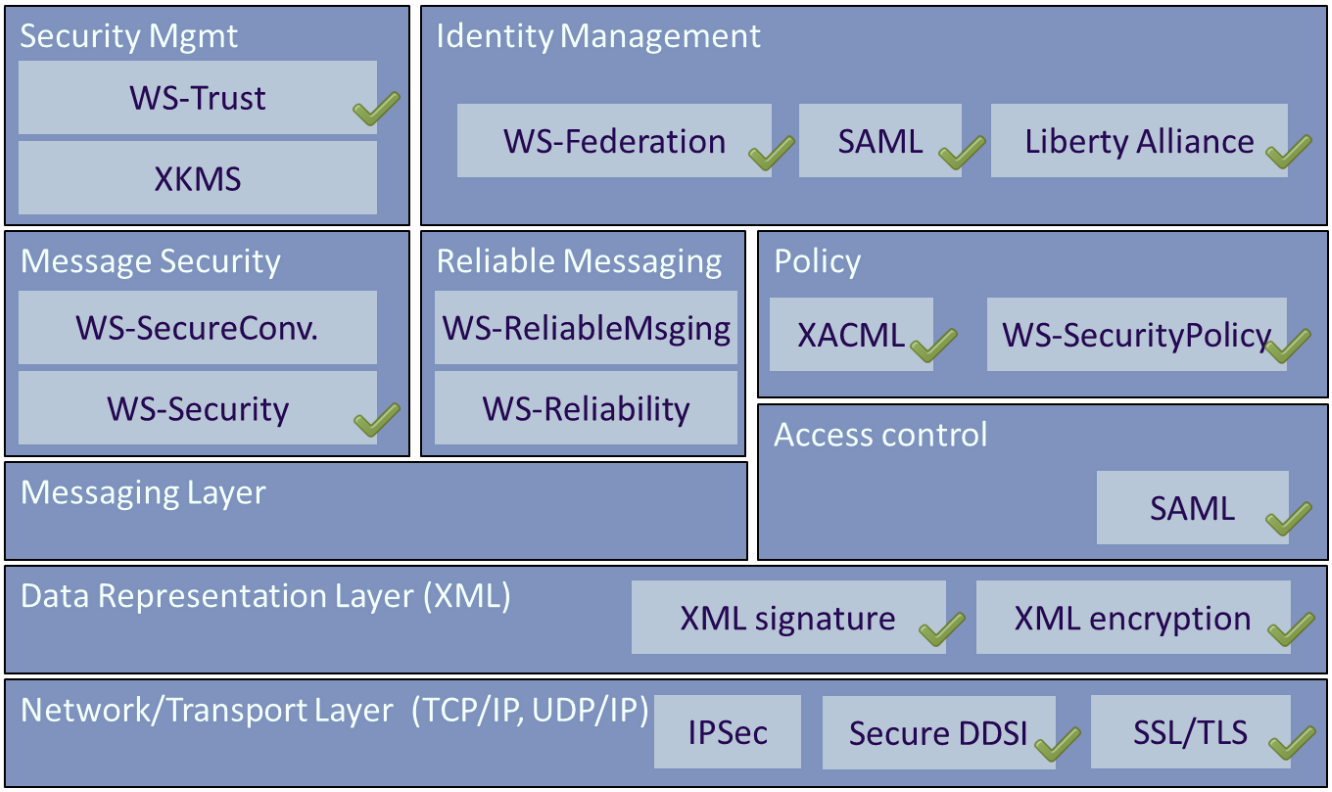

Figure 15. WS- standards involved in security*

Appendix B: Revision History

Table 4. Revision History

Date	Release	Editor	Primary clauses modified	Descriptions
February 29, 2016	.1	A. Balaban	all	initial version
April 27, 2016	.2	A. Balaban	all	first draft
June 29, 2017	.3	S. Serich	all	General Checking

Appendix C: Bibliography

[1] OGC: Testbed 11 Aviation - Guidance on Using Semantics of Business Vocabulary and Rules (SBVR) Engineering Report. (2015).

[2] Guidance on Writing AIRM Constraints

[3] OMG: Semantics of Business Vocabulary and Rules (SBVR) 1.1