



# SWIM (GG AG ) Architectural Definition - Final

## Document information

Project Title	SWIM Design
Project Number	14.01.03
Project Manager	Indra
Deliverable Name	SWIM (GG AG ) Architectural Definition - Final
Deliverable ID	D30
Edition	00.01.01
Template Version	03.00.00

## Task contributors

ENAV - EUROCONTROL - FREQUENTIS - INDRA - NORACON - THALES

*Please complete the advanced properties of the document*

## Abstract

This document presents the Architectural Definition of the SWIM Technical Infrastructure.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

1 of 284

## Authoring & Approval

Prepared By - <i>Authors of the document.</i>		
Name & Company	Position & Title	Date
██████████ INDRA	██████████	25/04/2016
██████████ FREQUENTIS	██████████	08/06//2016
██████████ THALES	██████████	06/06/2016

Reviewed By - <i>Reviewers internal to the project.</i>		
Name & Company	Position & Title	Date
██████████ INDRA	██████████	20/05/2016
██████████ INDRA	██████████	07/06/2016
██████████ FREQUENTIS	██████████	13/05/2016
██████████ THALES	██████████	07/06/2016
██████████ THALES	██████████	06/06/2016
██████████ EUROCONTROL	██████████	07/06/2016

Reviewed By - <i>Other SESAR projects, Airspace Users, staff association, military, Industrial Support, other organisations.</i>		
Name & Company	Position & Title	Date
██████████ SELEX	██████████	20/05/2016
██████████ EUROCONTROL	██████████	07/06/2016
██████████ DSNA	██████████	07/06/2016
██████████ DSNA	██████████	07/06/2016

Approved for submission to the SJU By - <i>Representatives of the company involved in the project.</i>		
Name & Company	Position & Title	Date
██████████ INDRA	██████████	10/06/2016
██████████ ENAV	██████████	10/06/2016
██████████ EUROCONTROL	██████████	10/06/2016
██████████ NORACON	██████████	10/06/2016
██████████ FREQUENTIS	██████████	10/06/2016
██████████ THALES	██████████	10/06/2016

Rejected By - <i>Representatives of the company involved in the project.</i>		
Name & Company	Position & Title	Date

Rational for rejection
None.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

## Document History

Edition	Date	Status	Author	Justification
00.00.01	15/04/2016	Draft		First evolution from D037 to D030.
00.00.02	13/05/2016	Draft		Document re-structured: <ul style="list-style-type: none"> <li>• moved ontology chapter to 2.1 (was 2.3)</li> <li>• moved deployment chapter to 2.4 (was 2.2.6)</li> <li>• re-arranged security related sections</li> <li>• removed PKI and BCA Functional Blocks</li> </ul>
00.00.03	20/05/2016	Draft		Updated FB dependency figures; Incorporated most of the review comments from Indra and Frequentis.
00.00.04	25/05/2016	Draft		Updates according webex review
00.00.05	31/05/2016	Final draft for partners review.		System Ports replaced by Interface Bindings; Glossary of terms and acronyms updated.
00.01.00	08/06/2016	Final version for approval		Updates according to review comments from WP14.01.03 and from WPB.04.03.
00.01.01	15/07/2016	Final		Updates after SJU assessment.

## Intellectual Property Rights (foreground)

This deliverable consists of SJU foreground.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

3 of 284

# Table of Contents

<b>AUTHORING &amp; APPROVAL</b> .....	<b>2</b>
<b>TABLE OF CONTENTS</b> .....	<b>4</b>
<b>LIST OF TABLES</b> .....	<b>10</b>
<b>LIST OF FIGURES</b> .....	<b>10</b>
<b>EXECUTIVE SUMMARY</b> .....	<b>13</b>
<b>1 INTRODUCTION</b> .....	<b>14</b>
1.1 PURPOSE OF THE DOCUMENT .....	14
1.2 INTENDED READERSHIP .....	14
1.3 INPUTS FROM OTHER PROJECTS .....	15
1.4 GLOSSARY OF TERMS.....	16
1.4.1 <i>Common terminology</i> .....	16
1.5 ACRONYMS AND TERMINOLOGY.....	23
1.5.1 <i>Specific terminology</i> .....	27
<b>2 ARCHITECTURE OF THE SYSTEM</b> .....	<b>33</b>
2.1 ONTOLOGY, TERMINOLOGY, RELATIONSHIPS AND SEMANTICS.....	34
2.1.1 <i>Introduction</i> .....	34
2.1.2 <i>Distributed system (of systems)</i> .....	34
2.1.3 <i>Middleware in a Distributed system (of systems)</i> .....	34
2.1.4 <i>Architectural Views</i> .....	34
2.1.5 <i>Organisation models</i> .....	41
2.1.6 <i>Broker</i> .....	43
2.1.7 <i>Conformance and interoperable implementations</i> .....	43
2.2 FUNCTIONAL VIEW.....	52
2.2.1 <i>Functional breakdown</i> .....	54
2.2.2 <i>Functional Analysis</i> .....	113
2.3 TECHNICAL VIEW .....	114
2.3.1 <i>SWIM-TI Technical Entities</i> .....	114
2.3.2 <i>SWIM-TI Interface Bindings</i> .....	116
2.3.3 <i>Technical Architecture View</i> .....	116
2.3.4 <i>Architecture and Network Domains</i> .....	119
2.3.5 <i>Architectural Options</i> .....	121
2.4 DEPLOYMENT OPTIONS.....	181
2.4.1 <i>SWIM-TI Messaging</i> .....	182
2.4.2 <i>SWIM-TI Security</i> .....	184
<b>3 ENABLERS ALLOCATION</b> .....	<b>191</b>
3.1 ALLOCATION OF ENS TO FUNCTIONAL BLOCKS .....	191
3.2 ALLOCATION OF ENS TO SYS PRIMARY PROJECTS .....	193
<b>4 SWIM TECHNICAL INFRASTRUCTURE DESIGN PRINCIPLES</b> .....	<b>195</b>
4.1 DESIGN PRINCIPLES.....	196
4.2 DESIGN DECISIONS: ARCHITECTURAL CHOICES VS. DESIGN PRINCIPLES .....	198
<b>5 REFERENCES</b> .....	<b>203</b>
5.1 APPLICABLE DOCUMENTS .....	203
5.2 REFERENCE DOCUMENTS .....	203
<b>APPENDIX A P14.01.03 TAD INTEGRATION WITH B4.3 ADD</b> .....	<b>206</b>

A.1	ADD MAIN CONCEPTS/STEREOTYPES.....	206
A.2	P14.01.03 MAIN ARCHITECTURAL ELEMENTS.....	209
A.3	P14.01.03 PROPOSALS.....	210
A.3.1	SWIM-TI PROFILES AND ADD INTEGRATION.....	210
A.3.2	SWIM-TI INFRASTRUCTURE SYSTEMS AND ADD INTEGRATION.....	212
A.3.3	SWIM-TI SUPPORT INFRASTRUCTURE PROPOSALS .....	213
APPENDIX B	PROCESS TO INTEGRATE A15 AND RESULTS .....	217
B.1	SUPPORT OF THE DEPLOYMENT CHOICE.....	217
APPENDIX C	PROCESS TO INTEGRATE MILITARY SYSTEMS INTO SWIM-TI.....	225
C.1	MILITARY INPUTS.....	225
C.1.1	ARCHITECTURAL OPTIONS .....	226
C.1.2	MILITARY SPECIFIC CONSTRAINTS .....	227
APPENDIX D	SWIM-TI SUPERVISION .....	229
D.1	SWIM-TI SUPERVISION HIERARCHY .....	229
D.1.1	SWIM-TI L1 SUPERVISION .....	229
D.1.1.1.	ARCHITECTURAL OPTIONS .....	229
D.1.2	SWIM-TI L2 SUPERVISION .....	231
D.1.2.1.	ARCHITECTURAL OPTIONS .....	231
D.2	SWIM-TI SUPERVISION SERVICES.....	232
APPENDIX E	COMMUNICATION RELATED ONTOLOGY, TERMINOLOGY, RELATIONSHIPS AND SEMANTICS.....	235
E.1	CLASSIFICATION.....	235
E.1.1	INTRODUCTION.....	235
E.1.2	KEY CHARACTERISTICS.....	235
E.1.2.1	DECOUPLING.....	235
E.1.2.2	PERSISTENCE .....	237
E.1.2.3	DISCRETE/STREAMING .....	237
E.1.2.4	CLASSIFICATIONS.....	237
E.1.2.5	CARDINALITY .....	238

E.1.2.6	COORDINATION .....	238
E.1.3	HIGH LEVEL STRUCTURE.....	238
E.1.3.1	RPC.....	238
E.1.3.2	MESSAGE ORIENTED.....	238
E.1.3.3	STREAMING .....	239
E.2	SPECIFIC NOTIONS.....	239
E.2.1	MULTICAST .....	239
E.2.2	MESSAGE BROKER.....	240
E.2.3	MESSAGE BROKER VERSUS ENTERPRISE SERVICE BUS .....	241
E.2.4	MESSAGING BUS .....	241
E.2.5	TERMINOLOGY TO USE.....	241
APPENDIX F	AMQP V1.0.....	243
F.1	THE PROBLEM/NEED .....	243
F.1.1	PROPRIETARY TRANSPORT PROTOCOLS FOR “ASYNCHRONOUS MESSAGING” ..	243
F.1.2	STANDARDS-BASED TRANSPORT PROTOCOLS ARE NOT APPROPRIATE FOR “ASYNCHRONOUS MESSAGING” .....	243
F.1.2.1	INTRODUCTION.....	243
F.1.2.2	X.400.....	243
F.1.2.3	DDS-I OVER RTPS (DDS INTEROPERABILITY WIRE PROTOCOL (DDSI-RTPS)) .....	244
F.1.2.4	SOAP OVER HTTP .....	244
F.1.2.5	SOAP OVER EMAIL .....	245
F.1.2.6	SOAP OVER JMS .....	245
F.2	CANDIDATE STANDARDS.....	245
F.2.1	AMQP V1.0.....	245
F.2.2	XMPP .....	245
F.2.3	MQTT 3.1.1.....	246
F.3	PREVIOUS WORK ON AMQP IN SESAR .....	246
F.3.1	WP14.1.2.....	246
F.3.2	OTHER WP.....	246

F.4	WHAT IS NEW .....	246
F.5	OUTLOOK .....	246
F.5.1	EXTENSIONS.....	246
F.5.2	BINDINGS.....	247
F.5.3	INTENTIONS.....	247
F.6	COMPARISON BETWEEN AMQP 0-9-1 AND AMQP 1.0.....	247
F.6.1	COMPARING SCOPE AND FEATURES .....	247
F.6.1.1	GENERAL SCOPE.....	247
F.6.1.2	INDIVIDUAL FEATURES.....	248
F.6.2	COMPARING THE ADOPTION OF STANDARDS.....	248
F.6.2.1	BROKER ADOPTION.....	248
F.6.2.2	CLIENT ADOPTION.....	249
F.6.2.3	SUPPORTING TOOLS .....	250
F.6.3	COMPARING THE STANDARDIZATION .....	250
F.6.4	ADDITIONAL INFORMATION.....	251
APPENDIX G	REST.....	255
G.1	THE PROBLEM/NEED .....	255
G.2	CONSIDERATIONS .....	255
G.2.1	STANDARD.....	255
G.2.2	REST-STYLE ACCESSIBILITY .....	255
G.2.3	REST STYLE VERSUS SOAP.....	255
G.2.3.1.	THE MORE DIFFICULT THINGS .....	255
G.2.3.2.	SERVICE DESCRIPTION .....	256
G.2.3.3.	SERVICE DISCOVERY .....	257
G.2.3.4.	PERFORMANCE .....	257
G.2.4	SECURITY .....	257
G.2.4.1.	SOAP BASED WEB SERVICES .....	257
G.2.4.2.	REST-STYLE .....	257
APPENDIX H	BASELINE AND STEP1 ENABLERS ALLOCATION.....	259

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

H.1	ALLOCATION OF BASELINE AND STEP 1 ENS TO FUNCTIONAL BLOCKS, PROJECTS AND ASSESSMENT .....	259
APPENDIX I	FO OVERLAY NETWORK CANDIDATE ARCHITECTURES .....	268
I.1	PROPOSAL 1: THE DILLON MODEL.....	268
I.1.1	CONTROL PLANE.....	268
I.1.2	DATA PLANE .....	269
I.1.3	ADVANTAGES .....	269
I.1.4	DISADVANTAGES.....	269
I.1.5	FOLLOW-UP .....	269
I.1.6	DDS QOS.....	269
I.1.6.1.	DURABILITY .....	269
I.1.6.2.	PARTITION.....	270
I.1.6.3.	TRANSPORT_PRIORITY .....	270
I.1.6.4.	SUPPORT FOR WORK SETS .....	270
I.1.6.5.	LOCAL RECOVERY .....	270
I.2	PROPOSAL 2 .....	270
I.3	PROPOSAL 3 .....	271
APPENDIX J	INTERFACE EVOLUTION .....	272
J.1	SCOPE .....	272
J.2	OBJECTIVE.....	273
J.3	CONCEPTS.....	274
J.3.1	SERVICE TECHNICAL INTERFACE .....	274
J.3.2	COMPATIBILITY .....	274
J.3.3	VERSIONING.....	275
J.3.4	STRATEGIES .....	275
J.3.5	CHANGE TYPES.....	276
J.3.5.1	<i>Data Model Changes</i> .....	276
J.3.5.2	<i>Message Changes</i> .....	277
J.3.5.3	<i>Interface Operations Changes</i> .....	277
J.3.5.4	<i>Binding Changes</i> .....	277
J.4	INTERFACE EVOLUTION TECHNIQUES .....	278
J.4.1	MINOR VERSIONING COMPATIBILITY TECHNIQUES.....	278
J.4.1.1	<i>Optional Data Elements</i> .....	278
J.4.1.2	<i>Selectable Data Elements</i> .....	278
J.4.1.3	<i>Flexible Data Types</i> .....	278
J.4.1.4	<i>Wildcards</i> .....	279
J.4.1.5	<i>Minor Version Techniques to Change Types Relationship</i> .....	279

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



J.4.2 MAJOR VERSIONING COMPATIBILITY TECHNIQUES .....	280
J.4.2.1 <i>Static Binding</i> .....	280
J.4.2.2 <i>Adapter</i> .....	281
J.4.2.3 <i>Default Values</i> .....	282
<b>J.5 SWIM-TI INTERFACE EVOLUTION STRATEGY DEFINITION NEEDS .....</b>	<b>283</b>

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

## List of tables

Table 1 – Classes of Availability .....	28
Table 2 – SWIM ConOps Functional Block requirements .....	55
Table 3 – SWIM ConOps Registry requirements.....	56
Table 4 – SWIM ConOps requirements on Policy Related Function .....	58
Table 5 – Registry Functional Block Dependencies .....	59
Table 6 – MEP characterisation.....	68
Table 7 – Messaging Functional Block Dependencies .....	85
Table 8 – SWIM ConOps requirements on Security.....	90
Table 9 – Security Functional Block Dependencies .....	99
Table 10 – Supervision Functional Block Dependencies.....	105
Table 11 – SWIM ConOps requirements on SWIM-TI Recording Functional Block.....	108
Table 12 – Recording Functional Block Dependencies .....	109
Table 13 – Shared Object Functional Block Dependencies .....	111
Table 14 – Functional block dependencies.....	113
Table 15 – SWIM ConOps requirements on SWIM-TI Node.....	115
Table 16 – SWIM-TI interfaces .....	119
Table 17 – Self-contained root Registry roles.....	121
Table 18 – Root registry with external reference roles .....	122
Table 19 – Root registry with provider affiliate roles.....	123
Table 20 – Root registry with consumer affiliate roles .....	124
Table 21 – Registry architectural options Pros and Cons.....	125
Table 22 – SWIM-TI Registry Architectural choice .....	125
Table 23 – OSI standard model.....	127
Table 24 – Routing Layer 3 Pros and Cons.....	131
Table 25 .....	146
Table 26 – Security FB Architectural Options Pros and Cons .....	164
Table 27 – SWIM-TI Security FB Architectural choice.....	164
Table 28 – Shared Object FB Architectural Options Pros and Cons.....	175
Table 29 – Summary of Architectural Proposals with impact on DDS.....	176
Table 30 – Routing Deployment Options Pros and Cons .....	184
Table 31 – Allocation of ENs to functional blocks.....	193
Table 32 – Allocation of ENs to SYS primary projects.....	194
Table 33 – SWIM ConOps SWIM 10 Key Principles .....	197
Table 34 – Architectural Choices vs. Design Principles .....	202
Table 35 – ADD Main Concepts/Stereotypes .....	208
Table 36 – A15 functions vs. SWIM-TI FB.....	217
Table 37 – Military constraints .....	228
Table 38 – SWIM-TI Supervision responsibilities .....	232
Table 39 – SWIM-TI Supervision L1 responsibilities .....	233
Table 40 – SWIM-TI Supervision L1 + L2 responsibilities .....	233
Table 41 – Allocation of Baseline and Step 1 ENs to functional blocks, SYS Primary Projects and Assessment.....	267
Table 42 – Techniques to assure backwards compatibility between minor versions .....	279

## List of figures

Figure 1 – Connecting Technical Systems via System Ports .....	38
Figure 2 – SWIM-TI Interface Bindings.....	39
Figure 3 – SWIM profile, Part, Role and Self-standing Set.....	48
Figure 4 – Relevant Self-standing Sets: Consumer role.....	49
Figure 5 – Relevant Self-standing Sets: Consumer role.....	50
Figure 6 – Relevant Self-standing Sets: Provider role.....	51
Figure 7 – SWIM-TI model.....	53
Figure 8 – SWIM-TI Functional Breakdown .....	54

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

10 of 284

Figure 9 – Registry FB .....	58
Figure 10 – Registry Functional Block Dependencies .....	59
Figure 11 – Registry FB use of SEC FB Dependencies .....	59
Figure 12 – Messaging Functional Block breakdown .....	61
Figure 13 – Advanced Routing Mechanisms .....	64
Figure 14 – Topic-based Publish-Subscribe .....	72
Figure 15 – Synchronous Request/Reply MEP sequence diagram .....	72
Figure 16 – Asynchronous Request/Reply MEP sequence diagram.....	73
Figure 17 – Observer Push MEP sequence diagram .....	74
Figure 18 – Observer Pull MEP sequence diagram .....	75
Figure 19 – Asynchronous Fire & Forget MEP sequence diagram .....	76
Figure 20 – Fully decoupled Request/Reply MEP sequence diagram .....	76
Figure 21 – Layering in SWIM data representation .....	83
Figure 22 – Messaging Functional Block Dependencies.....	86
Figure 23 – MSG FB use of SEC FB Dependencies .....	87
Figure 24 – MSG FB use of REG FB Dependencies.....	88
Figure 25 – MSG FB use of REC FB Dependencies.....	88
Figure 26 - SWIM-TI Security FB functional decomposition overview.....	89
Figure 27 – Security Functional Block breakdown.....	90
Figure 28 – Security Functional Block Dependencies .....	100
Figure 29 – SEC FB use of REG FB Dependencies .....	100
Figure 30 – Supervision Functional Block breakdown.....	102
Figure 31 – SWIM-TI Supervised Entity Lifecycle .....	104
Figure 32 – Supervision Functional Block Dependencies .....	106
Figure 33 – SPV FB use of REC FB Dependencies.....	106
Figure 34 – SPV FB use of REG FB Dependencies.....	106
Figure 35 – SPV FB use of MSG FB Dependencies .....	107
Figure 36 – Recording Functional Block Dependencies.....	109
Figure 37 – Shared Object Functional Block Dependencies .....	112
Figure 38 – SO FB use of MSG FB Dependencies .....	112
Figure 39 – SWIM-TI Technical Architecture View .....	117
Figure 40 – Network Domains.....	120
Figure 41 – Typical Architecture .....	120
Figure 42 – Self-contained root Registry .....	121
Figure 43 – Root registry with external reference.....	122
Figure 44 – Root registry with provider affiliate.....	123
Figure 45 – Root registry with consumer affiliate.....	124
Figure 46 – Registry Architectural options .....	126
Figure 47 – Federated Routing: example .....	132
Figure 48 – WS-N stack of protocols .....	141
Figure 49 – WS-BrokeredNotification Notifications and Routing channels.....	143
Figure 50 .....	145
Figure 51 .....	147
Figure 52 – Topic-based Filtering .....	153
Figure 53 – Content-based Filtering .....	153
Figure 54 – Protocol Bridge with Data Encapsulation .....	155
Figure 55 –Protocol Bridge with Data Format Transformation .....	155
Figure 56 – Identity at functional and technical views .....	156
Figure 57 – Brokered Authentication (logical) design pattern and related views.....	157
Figure 58 – Brokered Authentication Based on X.509 Certificates.....	159
Figure 59 – Brokered Authentication Based On Security Token .....	160
Figure 60 – Use of PKI in Security Token based Brokered Authentication .....	162
Figure 61 – Use of X.509 attribute certificates and XACML request for Authorization.....	163
Figure 62 – Overview of Flight Object data model.....	166
Figure 63 – Efficient use of Network .....	168
Figure 64 – General Architecture of the SWIM FO Overlay Network.....	170
Figure 65 – SWIM FO Overlay network – FO Layer.....	171
Figure 66 – FO CLUSTERS distribution.....	172

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Figure 67 – FO Overlay Architecture .....	173
Figure 68 – General Architecture (one HA alternative from BU-09) .....	177
Figure 69 – IGMP basic architecture (source Wikipedia) .....	180
Figure 70 – SWIM-TI and different SWIM-TI Functional Block deployment options .....	181
Figure 71 – Delegation by Area .....	182
Figure 72 – Delegate by Stakeholder and Area .....	183
Figure 73 – Certification Authority delegation .....	185
Figure 74 – Bridge CA architecture .....	186
Figure 75 – Physical view of CA .....	187
Figure 76 – Identity Federation Option 1 .....	188
Figure 77 – Identity Federation Option 2 .....	188
Figure 78 – Identity Federation Option 3 .....	189
Figure 79 – Federation of SWIM-TI trust domains .....	189
Figure 80 – SWIM-TI and different SWIM-TI Functional Block deployment options .....	211
Figure 81 – Example: Two SWIM-TI Profiles supporting one CC .....	212
Figure 82 – Example: SWIM-TI Functional Block in the ADD .....	213
Figure 83 – Supervision Infrastructure System .....	214
Figure 84 – Access Point Infrastructure System .....	215
Figure 85 – Military Access Point Infrastructure System .....	216
Figure 86 – A15 SWIM-TI A/G Deployment Options .....	218
Figure 87 – Ground System implements the SWIM-TI A/G Profile .....	220
Figure 88 – Several Ground System implements the SWIM-TI A/G Profile .....	221
Figure 89 – Ground System doesn't implement the SWIM-TI A/G Profile .....	222
Figure 90 – Several Ground System doesn't implement the SWIM-TI A/G Profile .....	223
Figure 91 – Ground System doesn't implement the SWIM-TI A/G Profile (legacy integration) .....	224
Figure 92 – Military SWIM-TI gateway .....	227
Figure 93 – One SWIM Node for SWIM-TI L1 Supervision .....	229
Figure 94 – One SWIM Node for one or more connected ATM Systems and one SWIM-TI L1 Supervision .....	230
Figure 95 – One or more SWIM Node for SWIM-TI L1 Supervision .....	230
Figure 96 – Different SWIM Node deployments without SWIM-TI L1 Supervision .....	231
Figure 97 – SWIM-TI L2 Supervision .....	232
Figure 98 – FO SUMMARY distribution .....	269
Figure 99 – Key components for Interface Evolution .....	272
Figure 100 – Static Binding .....	280
Figure 101 – Adapter .....	281

## Executive summary

The purpose of this deliverable is to provide Iteration 3.1 version of the SWIM-TI architecture. SWIM-TI Iteration 3.0 architecture (ref. [56]) has been improved performing a gap analysis against it and the SWIM-TI Iteration 3.1 definition. The gap analysis has been driven by a set of bottom-up and top-down activities carried out by P14.1.4 and P14.1.3.

The approach adopted when working on a specific bottom-up or top-down activity is the following:

- P14.1.4-P14.1.3 defined a detailed description of each activity detailing also the expected results,
- P14.1.4-P14.1.3 defined a set of technical SWIM-TI use cases aiming at driving the design phases that is architecture definition and requirements specification.
- According to the use case model, P14.1.3 and P14.1.4 established a shared SWIM-TI functional view which mainly consists of the SWIM-TI functional blocks (FBs) description.
- P14.1.3 derived the architectural options suitable for SWIM-TI.
- P14.1.4 implemented the set of requirements associated to the aforementioned technical activities.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

13 of 284

# 1 Introduction

## 1.1 Purpose of the document

This document refines the functional decomposition defined in the ADD produced by B4.3 for the SWIM Technical Infrastructure (SWIM-TI). However SWIM-TI is a special case as the technical infrastructure is not understood as a system as such. SWIM-TI is a set of software components distributed over a network infrastructure providing functions enabling collaboration among ATM systems.

Section 2 provides the Architecture of the SWIM-TI, describing it in different views (Functional View, Technical View and Deployment View).

Section 3 provides the Analysis of the Enablers that are allocated to SWIM in Dataset 00.00.16.

Section 4 recalls the design decision and assumptions that guide SWIM-TI design development.

Appendix A presents the process to integrate P14.01.03 TAD into B4.3 ADD, following B4.1 models.

Appendix B describes the process followed to integrate A15 (A/G) and describes its results

Appendix C describes the process followed to integrate Military Systems into SWIM-TI.

Appendix D provides additional considerations regarding SWIM-TI Supervision<sup>1</sup>.

Appendix E provides considerations on ontology, terminology and semantics that have been adopted in the document.

Appendix F gives rationale on the introduction of AMQP 1.0 as a possible technology.

Appendix G provides considerations about Representational State Transfer (REST).

Appendix H includes the analysis for the baseline and Step 1 Enablers.

Appendix I describes the candidate network architectures for supporting the FO.

Appendix J provides considerations and techniques to face Interface Evolution properly.

## 1.2 Intended readership

The intended audience of this document is:

- SJU/IS in order to manage the SWIM Technical Infrastructure TAD,
- P14.01.04
- SWP14.2 projects in order to review and prototype this TAD;
- WP08 in order to coordinate with the different ConOps and AIRM/ISRM production
- B4.1 for model consistency.
- B4.3 for consistency checking and consolidation in ADD.

<sup>1</sup> At the time being, the need for SWIM-TI Supervision at other level different than Local (e.g. regional, sub-regional) is being challenged.

## 1.3 Inputs from other projects

The following input documentation has been used to perform the architecture description of the SWIM Technical Infrastructure:

- P14.01.02-D04 [17] and P14.01.02-D07 [18] for description and evaluation of security options related to considered technologies,
- P14.02.09-D03 [9] for Step1 SWIM architecture,
- P14.01.03-D31 [27] for Step2 Iteration 2.0 SWIM architecture,
- 08.01.01-D42 [11] for requirements related to SWIM Concept of Operation,
- B.04.03-D98 [6] for high level description of the ATM EA architecture,
- 08.03.02-D03 [12] for requirements related to SWIM Registry Concept of Operation,
- 08.03.01-D14 [15] for requirements related to SWIM Supervision Concept of Operation,
- 08.03.10-D64 [19] for ISRM V1.4,
- B4.1 - D136 [63] for European ATM Architecture (EATMA) Guidance Material v6.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

15 of 284

## 1.4 Glossary of terms

### 1.4.1 Common terminology

The table below is the terminology shared with the Technical Specification [13]. Note that many terms are defined in accordance with the Recommendation ITU-T X.1252, “Baseline identity management terms and definitions”, from the International Telecommunication Union (ref. [62]).

Term	Definition
<b>Access Control</b>	ITU-T IdM X.1252 defines this term as a procedure used to determine if an entity should be granted access to resources, facilities, services, or information based on pre-established rules and specific rights or authority associated with the requesting party.
<b>Address</b>	ITU-T IdM X.1252 defines this term as an identifier for a specific termination point that is used for routing.
<b>Agent</b>	ITU-T IdM X.1252 defines this term as an entity that acts on behalf of another entity.
<b>Alarm</b>	An indication of an error or an abnormal and/or undesirable condition for a resource. An example of an alarm would be for a “connection down” in a data communications channel, or a non-booting processor in a hardware platform. Alarms originate with the hardware, software, and data communications infrastructure, and the infrastructure provides an indication to the Supervision when an alarm is raised or cleared. The Supervision notifies the local owner or authorized requester when an alarm is raised or cleared for a monitored resource.
<b>Alliance</b>	ITU-T IdM X.1252 define this term as an agreement between two or more independent entities that defines how they relate to each other and how they jointly conduct activities.
<b>Archive</b>	Information storage that is used for by the automation for long-term retention of information produced and/or used at the local SWIM Node. An archive may be offline with respect to the SWIM Node, meaning that it is not directly accessible to processes and services running on the SWIM Node; or it may be online with respect to the SWIM Node, meaning that the archive is directly accessible to processes and services running on the SWIM Node. Information that is logged by the SWIM Supervision is retained online for a configurable time period, after which it is archived and is then no longer guaranteed to be available in the same manner as information that has not reached its retention time limit. Each SWIM Node will have local processes and procedures for storing, maintaining, and accessing archived information. Archived information will be available to the reporting capability; however, the response time for accessing archived information will vary according to the storage approach used by the node.
<b>Assertion</b>	ITU-T IdM X.1252 defines this term as a statement made by an entity without accompanying evidence of its validity.
<b>ATM Service or SWIM ATM Service</b>	A service representing the exchange of well-defined ATM information. These services are defined by WP8 and are part of the ISRM.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

16 of 284



Term	Definition
<b>Attribute</b>	ITU-T IdM X.1252 defines this term as information bound to an entity that specifies a characteristic of the entity.
<b>Attribute Based Access Control (ABAC)</b>	In attribute-based access control (ABAC), access is based on attributes of the user. The user has to prove these attributes to the access control engine. An attribute-based access control policy specifies which attributes need to be satisfied in order to grant access to an object.
<b>Attribute Value</b>	ITU-T IdM X.1252 defines this term as a particular instance of the class of information indicated by an attribute type.
<b>(Entity) Authentication</b>	ITU-T IdM X.1252 defines this term as a process used to achieve sufficient confidence in the binding between the entity and the presented identity.
<b>Authorization</b>	ITU-T IdM X.1252 defines this term as the granting of rights and, based on these rights, the granting of access.
<b>Authorized requester</b>	A human user or automated process, at the local SWIM Node or at a remote SWIM Node, that has been authenticated and is authorized per security requirements to make a service request.
<b>Binding</b>	ITU-T IdM X.1252 defines this term as an explicit established association, bonding, or tie.
<b>Bridge Certificate Authority (BCA)</b>	The Bridge Certification Authority (BCA) architecture addresses the shortcomings of the two basic PKI architectures, and to link PKIs that implement different architectures. The BCA does not issue certificates directly to users. The BCA is not intended to be used as a trust point by the users of the PKI, unlike the "root" CA in a hierarchy. The BCA establishes peer-to-peer trust relationships with the different user communities, which allows the users to keep their natural trust points. These relationships are combined to form a "bridge of trust" enabling users from the different user communities to interact with each other through the BCA with a specified level of trust.
<b>Certificate</b>	ITU-T IdM X.1252 defines this term as a set of security-relevant data issued by a security authority or a trusted third party, that, together with security information, is used to provide the integrity and data origin authentication services for the data.
<b>Claim</b>	ITU-T IdM X.1252 defines this term as to state as being the case, without being able to give proof.
<b>Confidentiality Ensuring</b>	Confidentiality Ensuring aims at providing the ability to ensure "non-disclosure" of information. This service relies on the policy enforcement features and to the cryptographic mechanisms provided by the Cryptography security enabler to ensure information confidentiality at message level.  Note: These services breakdown is described in §2.
<b>Credential</b>	ITU-T IdM X.1252 defines this term as a set of data presented as evidence of a claimed identity and/or entitlements.
<b>Data Origin Authentication</b>	Equivalent expression for Information Origin Authentication.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

17 of 284

Term	Definition
<b>Data Validation</b>	Data validation allows checking for conformance to message/data type descriptions as defined by SWP8.1, SWP8.3 and P14.01.04. The conformance conditions are expressed in form of well-defined policy assertions assigned to the SWIM service definition.
<b>Delegation</b>	ITU-T IdM X.1252 defines this term as an action that assigns authority, responsibility, or a function to another entity.
<b>Digital Identity</b>	ITU-T IdM X.1252 defines this term as a digital representation of the information known about a specific individual, group or organization.
<b>Digital Signature (algorithm)</b>	Digital Signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that a known sender created the message, and that it was not altered in transit. Unlike a Message Authentication Code, a Digital Signature also provides support for non-repudiation.
<b>Enabling Service</b>	A service provided by the SWIM-TI.
<b>Entity</b>	ITU-T IdM X.1252 defines this term as something that has separate and distinct existence and that can be identified in context. An entity can be a physical person, an animal, a juridical person, an organization, an active or passive thing, a device, a software application, a service, etc., or a group of these entities. In the context of telecommunications, examples of entities include access points, subscribers, users, network elements, networks, software applications, services and devices, interfaces, etc.
<b>European Network of Excellence in Cryptology (ECRYPT)</b>	ECRYPT (European Network of Excellence for Cryptology) is a 4-year European research initiative launched on 1 February 2004. The stated objective is to, "intensify the collaboration of European researchers in information security and more in particular in cryptology and digital watermarking".
<b>Federation</b>	ITU-T IdM X.1252 defines this term as an association of users, service providers, and identity service providers.
<b>Identification</b>	ITU-T IdM X.1252 defines this term as the process of recognizing an entity by contextual characteristics.
<b>Identifier</b>	ITU-T IdM X.1252 defines this term as one or more attributes used to identify an entity within a context.
<b>Identity</b>	ITU-T IdM X.1252 define this term as a representation of an entity in the form of one or more attributes that allow the entity or entities to be sufficiently distinguished within context. For identity management (IdM) purposes, the term identity is understood as contextual identity (subset of attributes), i.e., the variety of attributes is limited by a framework with defined boundary conditions (the context) in which the entity exists and interacts. Each entity is represented by one holistic identity that comprises all possible information elements characterizing such entity (the attributes). However, this holistic identity is a theoretical issue and eludes any description and practical usage because the number of all possible attributes is indefinite.

Term	Definition
<b>Identity Management (IdM)</b>	ITU-T IdM X.1252 define this term as a set of functions and capabilities (e.g., administration, management and maintenance, discovery, communication exchanges, correlation and binding, policy enforcement, authentication and assertions) used for assurance of identity information (e.g., identifiers, credentials, attributes); assurance of the identity of an entity and supporting business and security applications.
<b>Identity Provider (IdP)</b>	ITU-T IdM X.1252 define this term as an entity that verifies, maintains, manages, and may create and assign identity information of other entities. Depending on the type of digital identity, an Identity Provider may be Public Key Infrastructure (PKI) or Security Tokens Infrastructure (STI).  IdP is also named Identity Service Provider (IdSP)
<b>Information Origin Authentication</b>	SWIM-TI service to authenticate the originator entity of a message by several techniques at message level and transport level.
<b>Interface Control Document (ICD)</b>	An interface control document (ICD) in systems engineering and software engineering, describes the interface or interfaces between subsystems or to a system or subsystem.
<b>IOP Status</b>	Indicates the ability of the SWIM Node to provide shared object services.
<b>Messaging FB or SWIM-TI Messaging FB</b>	Messaging Functional Block provides a decoupled, interoperable and effective communications between information producer and the information consumers. This supports different message exchange patterns (e.g. publish-subscribe, request-response, push, etc...), different subscription styles (e.g. durable, non-durable) and different set of QoS (e.g. best-effort and reliable delivery).
<b>Mutual Authentication</b>	ITU-T IdM X.1252 defines this term as a process by which two entities (e.g., a client and a server) authenticate each other such that each is assured of the other's identity.
<b>Non-Repudiation</b>	ITU-T IdM X.1252 define this term as the ability to protect against denial by one of the entities involved in an action of having participated in all or part of the action.
<b>Pan-European Network Service (PENS)</b>	A joint EUROCONTROL-ANSPs led initiative to provide a common IP based network service across the European region covering voice and data communication and providing efficient support to existing services and new requirements that are emerging from future Air Traffic Management (ATM) concepts.
<b>Persistent</b>	ITU-T IdM X.1252 defines this term as existing and able to be used in services outside the direct control of the issuing assigner, without a stated time-limit.

Term	Definition
<b>Public Key Cryptography</b>	Public Key Cryptography refers to a cryptographic technique in which one key is secret private and a corresponding key one is public. Information is encrypted using the public key and can only be decrypted by the corresponding secret/private key or vice-versa, information is encrypted using the private key and can only be decrypted by the corresponding public key. Public Key Cryptography can also be used for Digital Signatures; in this case the private key is used for signing, and the corresponding public key for verifying.
<b>Public Key Infrastructure</b>	A Public Key Infrastructure (PKI) is a system, which may include hardware, software, human in the loop, policies and procedures, needed to create, manage, distribute, use, store and revoke digital identities in X.509 certificates based IdM.  PKIs represent the instantiation of the ITU-T X.1252 IdP when the X.509 certificates based security is adopted.
<b>Recording Functional Block or SWIM-TI Recording FB</b>	Recording FB includes the ability to collect, store and to retrieve on demand of information related to communication being performed via the SWIM Interfaces and supervision actions and events.
<b>Registry Functional Block or SWIM-TI Registry FB</b>	Registry FB includes two main groups of functions:  - Information Management enabling the management several kinds of ATM-specific service meta-data allowing to discover, to subscribe and to publish/update these information.  - Policy Management enabling the definition, validation and distribution of several kinds of policies including security. It covers policy administration (including creation, maintenance, change and deletion) and policy distribution and transformation and policy auditing.
<b>Revocation</b>	ITU-T IdM X.1252 defines this term as the annulment by someone having the authority, of something previously done.
<b>SAML Token</b>	Security Assertion Markup Language (Token)
<b>Security Attribute</b>	An abstraction representing the basic properties or characteristics of an entity with respect to safeguarding information; typically associated with internal data structures (e.g., records, buffers, files) within the information system and used to enable the implementation of access control and flow control policies, reflect special dissemination, handling or distribution instructions, or support other aspects of the information security policy.
<b>Security Domain</b>	ITU-T IdM X.1252 define this term as a set of elements, a security policy, a security authority, and a set of security-relevant activities in which the elements are managed in accordance with the security policy.
<b>Security Token</b>	Security tokens are used to prove one's identity electronically. The token acts like an electronic key to access something. Besides the information needed to authenticate an identity, a token can provide additional information (identity attributes) that are used for (e.g.) authorization purposes. Security tokens imply trust of a third party that issues the security tokens.

Term	Definition
<b>Security Token Infrastructure (STI)</b>	A Security Tokens Infrastructure (STI) is a system, which may include hardware, software, human in the loop, policies and procedures, needed to create, manage, distribute, use, store and revoke digital identities in security token based IdM.  STIs represent the instantiation of the ITU-T X.1252 IdP when the security tokens based security is adopted.
<b>Security Token Service (STS)</b>	A Security Token Service (STS) is a software based identity provider responsible for issuing and verifying security tokens as part of a claims-based identity management.
<b>Shared Object Functional Block or SWIM-TI Shared Object FB</b>	Shared Object FB is a special category that holds a pattern used to share data across multiple SWIM Nodes according to specific roles and rules.
<b>Security Functional Block or SWIM-TI Security FB</b>	Security Functional Block provides confidentiality, integrity, access control, accountability and non-repudiation functionalities, allowing data exchanged through the SWIM-TI to be protected.
<b>(Security) Policy</b>	An agreement upon which entities (e.g. Systems) can collaborate. A typical example of this is Authorization Policy and Audit Policy.
<b>(Security) Policy Life Cycle Management</b>	The Policies lifecycle management is a key aspects enabling the appropriate confidentiality policy enforcement.
<b>Service</b>	When used without further qualification, Service indicates either a SWIM Service or a SWIM Enabling Service that is to be managed by SWIM Supervision at the local SWIM Node.
<b>Service Agent SOA Design Pattern</b>	Service agents can be designed to automatically respond to predefined conditions without invocation via a published contract. Refer to SOA Patterns <a href="http://www.soapatterns.org/service_agent.php">http://www.soapatterns.org/service_agent.php</a>
<b>Service Virtualisation (Through Service Agent SOA design pattern)</b>	Service Virtualization helps insulate service infrastructure details such as service endpoint location, service inter-connectivity, policy enforcement, service versioning and dynamic service management information from service consumers.  Refer to: <a href="http://www.soapatterns.org/service_virtualization.php">http://www.soapatterns.org/service_virtualization.php</a>
<b>Supervision Functional Block or SWIM-TI Supervision FB</b>	Monitoring and Control FB includes control, fault management and performance monitoring at SWIM Node level (local supervision).
<b>SWIM Enabled System/Application</b>	A SWIM Enabled System/Application is a system/application exchanging information with other ATM actors according to the SWIM ATM Services defined by WP8 and the appropriate SWIM-TI defined by WP14.

Term	Definition
<b>SWIM Message Exchange Pattern (MEP)</b>	SWIM Exchange Pattern is a definition to provide data exchanges of a SWIM profile. The message exchange patterns can be defined in terms of a set of technical attributes including interaction pattern, security, quality of service, network infrastructure, middleware functional needs and mandated standards.
<b>SWIM Node or SWIM-TI Node</b>	SWIM-TI Node provides a collection of SWIM-TI Functional Blocks, compliant with one or more SWIM profiles, allowing a given ATM application to use the SWIM-TI.  A SWIM-TI Node is an autonomous point of presence in the Distributed System (of Systems) that interacts with other SWIM-TI Nodes in the Distributed System (of Systems).
<b>SWIM Profile Assertion (SPA)</b>	Declaration of the existence of a SWIM Profile combined with precisions on scope and motivation and with design considerations.
<b>SWIM Service</b>	A service that is managed by the SWIM Supervision capability at a local SWIM Node. SWIM Supervision is responsible for the data, process control, event-reporting, and statistics for these services.
<b>SWIM Technical Infrastructure (SWIM-TI)</b>	The SWIM Technical Infrastructure (SWIM-TI) contributes to the services' solution, aspects providing means supporting effective and secure ATM-specific service provision and consumption among SWIM-enabled ATM systems.
<b>Symmetric Key Cryptography (algorithms)</b>	A Symmetric Key algorithm uses the same cryptographic key (shared secret key) for both encryption of plaintext and decryption of cipher text.
<b>System of systems (SoS)</b>	System of systems (SoS) is the viewing of multiple, dispersed, independent systems in context as part of a larger, more complex system. A system is a group of interacting, interrelated and interdependent components that form a complex and unified whole.
<b>XML Encryption</b>	XML Encryption is a specification (by W3C recommendation) that defines how to encrypt the contents of an XML element.  <i>Note: W3C (World Wide Web Consortium) is the main standards organization for the world wide web.</i>
<b>XML Signature</b>	XML Signature is the XML syntax for digital signatures.
<b>X.509 certificates</b>	In cryptography, X.509 is an ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

## 1.5 Acronyms and Terminology

Term	Definition
<b>A/G</b>	Air/Ground
<b>ABAC</b>	Attribute Based Access Control
<b>ACC</b>	Air Traffic Control Centre
<b>ACCS</b>	Air Command and Control System (NATO terminology)
<b>ADD</b>	Architecture Description Document
<b>AFF-MEP</b>	Asynchronous Fire & Forget Message Exchange Pattern
<b>AIRM</b>	Aeronautical Information Reference Model
<b>AIS</b>	Aeronautical Information Services
<b>AIXM</b>	Aeronautical Information eXchange Model
<b>AMHS</b>	Aeronautical Message Handling System
<b>AMQP</b>	Advanced Message Queuing Protocol
<b>ARR-MEP</b>	Asynchronous Request/Reply Message Exchange Pattern
<b>ASM</b>	Any-Source Multicast
<b>ATC</b>	Air Traffic Control
<b>ATFCM</b>	Air Traffic Flow and Capacity Management
<b>ATM</b>	Air Traffic Management
<b>BCA</b>	Bridge Certification Authority
<b>BP</b>	Blue Profile
<b>BPMN</b>	Business Process Model and Notation
<b>CA</b>	Certification Authority (in the context of PKI)
<b>CONOPS</b>	Concept of Operations
<b>COTS</b>	Commercial Off The Shelf
<b>CRL</b>	Certificate Revocation List
<b>CRUD</b>	Create Read Update Delete
<b>CSR</b>	Certificate Signing Request

Term	Definition
DCPS	Data-Centric Publish-Subscribe
DDS	Data Distribution Service
DoS	Denial of Service
EAD	European AIS Database
EATMA	European ATM Architecture
ECRYPT	European Network of Excellence for Cryptology
EN	Enabler
ESB	Enterprise Service Bus
FAA	Federal Aviation Administration
FDMP	Flight Data Manager/Publisher
FDRR-MEP	Fully Decoupled Request/Reply Message Exchange Pattern
FO	Flight Object
G/G	Ground/Ground
GAT	General Air Traffic
HTTPS	HyperText Transfer Protocol Secure
ICD	Interface Control Document
IdM	Identity Management
IdP	Identity Provider
IdSP	Identity Service Provider
IER	Information Exchange Requirement
IGMP	Internet Group Management Protocol
INTEROP	Interoperability Requirements
IP	Internet Protocol
IS	Industrial Support
ISRM	Information Service Reference Model
IT	Information Technology
LAN	Local Area Network

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



Term	Definition
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>MET</b>	Meteo or Meteorological
<b>MEP</b>	Message Exchange Pattern
<b>MLD</b>	Multicast Listener Discovery
<b>MTU</b>	Maximum Transmission Unit
<b>NAF</b>	NATO Architecture Framework
<b>NATO</b>	North Atlantic Treaty Organization
<b>NFR</b>	Non-Functional Requirement
<b>NM</b>	Network Management (CFMU)
<b>NOP</b>	Network OPERations or Network Operations Portal
<b>NOTAM</b>	NOTice To AirMen
<b>OASIS</b>	Organization for the Advancement of Structured Information Standards
<b>OAT</b>	Operational Air Traffic
<b>OCSP</b>	Online Certificate Status Protocol
<b>OMG</b>	Object Management Group
<b>OPULL-MEP</b>	Observer Pull Message Exchange Pattern
<b>OPUSH-MEP</b>	Observer Push Message Exchange Pattern
<b>OS</b>	Operating System
<b>OSI</b>	Open Systems Interconnection
<b>OTS</b>	Off The Shelf
<b>PDP</b>	Policy Decision Point
<b>PE</b>	Policy Enforcement
<b>PENS</b>	Pan-European Network Service
<b>PEP</b>	Policy Enforcement Point
<b>PIM</b>	Protocol Independent Multicast
<b>PIM-SM</b>	PIM Sparse Mode
<b>PIM-SSM</b>	PIM Source-Specific Multicast

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

25 of 284

Term	Definition
<b>PKI</b>	Public Key Infrastructure
<b>PP</b>	Purple Profile
<b>PSPULL-MEP</b>	Publish/Subscribe Pull Message Exchange Pattern
<b>PSPUSH-MEP</b>	Publish/Subscribe Push Message Exchange Pattern
<b>QoS</b>	Quality of Service
<b>RA</b>	Registration Authority (in the context of PKI)
<b>REST</b>	REpresentation State Transfer
<b>RFC</b>	Request For Comments (Internet Engineering Task Force terminology)
<b>SAML</b>	Security Assertion Markup Language
<b>SESAR</b>	Single European Sky ATM Research Programme
<b>SESAR Programme</b>	The programme which defines the Research and Development activities and Projects for the SJU.
<b>SJU</b>	SESAR Joint Undertaking
<b>SJU Work Programme</b>	The programme which addresses all activities of the SESAR Joint Undertaking Agency
<b>SLA</b>	Service Level Agreement
<b>SM</b>	Sparse Mode
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>SO</b>	Shared Object
<b>SOA</b>	Service Oriented Architecture
<b>SOAP</b>	Simple Object Access Protocol
<b>SoS</b>	System of Systems
<b>SPA</b>	SWIM Profile Assertion
<b>SPV</b>	SuPerVision
<b>SRR-MEP</b>	Synchronous Request/Reply Message Exchange Pattern
<b>SSL</b>	Secure Socket Layer
<b>SSM</b>	Source-Specific Multicast
<b>SSO</b>	Single Sign-On

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

26 of 284

Term	Definition
STI	Security Token Infrastructure
STS	Secure Token Service
SWIM	System Wide Information Management
SWIM-TI	SWIM Technical Infrastructure
TAD	Technical Architecture Description
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TS	Technical Specification
UDDI	Universal Description Discovery and Integration
UDP	User Datagram Protocol
UML	Unified Modeling Language™
VA	Validation Authority (in the context of PKI)
VPN	Virtual Private Network
WAN	Wide Area Network
WP	Work Package
WS	Web Services
WSDL	Web Services Description Language
XACML	eXtensible Access Control Markup Language
YP	Yellow Profile

## 1.5.1 Specific terminology

### Availability

Measures of availability have traditionally been used by hardware manufacturers within technical documentation accompanying their servers. It is computed the following way:

$$\text{Availability} = \text{MTBF} / (\text{MTTR} + \text{MTBF})$$

(= Mean Time Between Failures / (Mean Time To Recover + Mean Time Between Failure)).

For software-based systems, availability is the percentage of time that a system is up and running for a particular duration of operation.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

27 of 284

Availability is most often expressed as a percentage also known as classes of availability including two nines, three nines, four nines, five nines, six nines (four nines means 99.99 percent uptime, which translates to 53 minutes of downtime (excl. planned maintenance) per year).

The following table shows the minutes of downtime allowed per year for a given availability level:

Availability	Downtime/Year
90 %	876 hours (more than 36 days)
95 %	438 hours (more than 18 days)
99 %	87 hours, 36 minutes (more than 3 days and a half)
99,9 %	8 hours, 45 minutes, 36 seconds
99,99 %	52 minutes, 33,6 seconds
99,999 %	5 minutes, 15,36 seconds
99,9999 %	31,68 seconds

Table 1 – Classes of Availability

### Blue Profile

Certain types of information sharing in ATM take place under a high safety critical context and hence, the requested infrastructure needs to be ready to support demanding requirements. These types of information share demand to be reliable and to deliver the required performance. This set of supported demands is usually identified as “Rock Solid” QoS, meaning that is trustable and not easily breakable.

SWIM-TI’s Blue Profile (BP)<sup>2</sup> is explicitly targeted at:

- Real-time or near real-time uses
- Extremely high availability
- Secured interactions
- Severe constraints with respect to the available resources

### Certification path

A certification path is a chain of certificates that uses trust relationships between issuers to determine if a certificate signed by another issuer is trustworthy. The chain starts from the public key of a Certification Authority (CA) trusted by the verifier<sup>3</sup>, and ends with the certificate that contains the public key. Each certificate in the certification path is signed by its predecessor’s key.

### DDoS (Distributed Denial of Service) attack

Distributed Denial of Service is a type of data infrastructure network service attack where multiple (compromised) systems perform DDoS attacks against a target system(s).

### DoS (Denial of Service) attack

A Denial of Service attack is a type of attack on a network service by flooding a node with useless traffic in order to prevent it from handling genuine traffic. DoS attacks usually exploit limitations of network protocols.

### Downtime

The time during which a system stops working or stops behaving as expected is referred to as downtime. A downtime may be scheduled for some kind of upgrade or for maintenance reasons; or unscheduled when it occurs because of failure of one or more system components.

<sup>2</sup> The interested reader can consult related ADD terminology in section 2.2.1 or Blue Profile SPA in section 2.4.1 in BP TS.

<sup>3</sup> ITU-T IdM X.1252 define this term as an entity that verifies and validates identity information

## Failover

Failover is an automatic action triggered by the failure of the primary component in a redundant system. The redundant system may be a processor, a server or a network device. The failure may be caused by hardware or software. The failover action makes the secondary components replacing the primary ones to operate in its stead.

## Failures

A *failure* occurs when the system generates a result that does not satisfy the system specification or when the system does not generate a result that is required by the system specification.

A *fault* is the external manifestation of a failure of a system component.

Flaviu Cristian defines the following failure classification<sup>4</sup>:

- An *omission* failure occurs when a server omits to respond to an input.
- A *timing* failure occurs when the server's response is functionally correct but untimely. Timing failures thus can be either *early* timing failures or *late* timing failures (*performance* failures).
- A *response* failure occurs when the server responds incorrectly. It is either a *value* failure if its output is incorrect or a *state transition* failure in case of incorrect state transition.
- A *crash* failure is when after a first omission to produce output, a server omits to produce output to all subsequent inputs until its restart. Crash failures can be further classified depending on the server state at restart:
  - An *amnesia-crash* occurs when the server restarts in a predefined initial state that does not depend on the inputs seen before the crash.
  - A *partial-amnesia-crash* occurs when, at restart, some part of the state is the same as before the crash while the rest of the state is reset to a predefined initial state.
  - A *pause-crash* occurs when a server restarts in the state it had before the crash.
  - A *halting-crash* occurs when a crashed server never restarts.

## Fault Tolerance

Fault tolerance is the ability of a system to respond gracefully to an unexpected hardware or software failure. It is based on redundancy and is generally implemented by error detection and subsequent system recovery<sup>5</sup>.

Fault tolerance has traditionally been hardware fault tolerance. As systems grow in complexity and rely heavily on software, they become more and more prone to design or transient faults. Faults may be induced by repair activities, hardware upgrading, or periodic maintenance. Mechanisms used for software fault tolerance include checkpoint/restart, recovery blocks and multiple-version programs.

Fault tolerance solutions generally rely on the following:

- Entity redundancy: the node and/or process are commonly used as unit of redundancy. Some infrastructures support finer grained entities such application components or objects.
- Fault detection: detecting the presence of a fault in the system and generating a fault report.
- Logging and Recovery: message logging may be used during recovery.

## Firewall

The role of a firewall is to keep a network secure by controlling incoming and outgoing traffic and analysing data packets for compliance with a predetermined set of security rules. It is either software or hardware based (sometimes both) and is usually used to protect intranets from unauthorized users from internet.

<sup>4</sup> "Understanding Fault-Tolerant Distributed Systems", Flaviu Cristian, Communications of the ACM, February 1991, Vol. 34, No. 2.

<sup>5</sup> In "Fundamental Concepts of Dependability", Avizienis, Laprie, and Randell

## Load Balancing

Load balancing is about distributing workload across multiple computing resources, network links and storage devices to optimise resource usage and maximise throughput.

Network-attached storage provides storage services that may be used to build load-balanced and fault-tolerant systems.

## Network Address Translation (NAT)

Network Address Translation (NAT) refers to the process of modifying IP address information in IP packet headers when transiting through some routing device<sup>6</sup>.

There are many types of translations:

- One-to-one NAT (basic NAT)
- Network address and port translation (NAPT)
- Port address translation (PAT): allows many internal hosts to share a single external IP address
- IP masquerading,
- NAT Overload, and
- Many-to-one NAT.

For more information on Network address translation refer to the exhaustive summary on <http://en.wikipedia.org/wiki/NAT>.

## P2P (Peer-to-Peer)

A peer-to-peer computer network designates a computer network where each computer, aka node, in the network can act as a client or server for the other nodes in the network for sharing resources (such as processing power, disk storage, or network bandwidth) without using a central server.

Unlike the client/server model where a node acts as a resource consumer (*client*) or a resource provider (*server*), a *peer* is both consumer and supplier of resources at the same time. As this does not rely on a central node for coordination, peer-to-peer networks offer high resilience to node failure and efficient usage of available resources.

## PEP (Policy Enforcement Point)

A software component in charge of intercepting access requests to a resource and enforcing decision related to the current policy. As a result the request is either processed normally or rejected for policy reason. For example a security policy can restrict access to a particular resource to authorize users. The PEP is in charge to check that the user issuing the request is in the list of authorized users.

## Purple Profile

Some types of information sharing in ATM are performed in challenging conditions and/or constraints and need specific infrastructure.

SWIM-TI's Purple Profile (PP)<sup>7</sup> is explicitly targeted at:

- High latency and/or low bandwidth conditions
- Need to minimize the communication overhead and transport connection number due to high cost of use of the communication

<sup>6</sup> NAT is solely of relevance for IPv4, but is not used in IPv6

<sup>7</sup> The interested reader can consult related ADD terminology in section 2.2.1 or Purple Profile SPA in section 2.4.1 in PP TS.

- Intermittent and unpredictable availability of end-to-end connectivity over the communication infrastructure
- Facilitate Security controls through constraints in the interaction patterns
- A permanent need for time, space and synchronization decoupling between participants (asynchronous operations)
- Simple and easily certifiable information producers and consumers

### **Scalability**

Scalability is the ability of a system, network, or process, to handle growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth<sup>8</sup>.

Scalability is often achieved using redundancy (by adding hardware resources and/or multiple application copies) and may include load balancers.

Load balancers can work in pairs (active–standby or active–active) and be used for High Availability solutions<sup>9</sup>.

### **Service Level Agreement (SLA)**

The service-level agreement (SLA) is a part of a service contract that formally the level of service. It is a negotiated agreement between the service consumer and the service provider.

The SLA will typically contain non-functional requirements such as mean time between failures (MTBF), mean time to repair or mean time to recovery (MTTR), and other measurable metrics such as response times.

### **Single Point of Failure**

Any hardware, electrical or software component which failure causes a malfunction in the entire system is referred to as a Single Point of Failure (SPOF). Redundancy is used within systems to continue operation under failure of such component.

### **SWIM-TI Profile**

A SWIM profile is a coherent, appropriately-sized grouping of middleware functions/services for a given set of technical constraints/requirements that permit a set of stakeholders to realize Information sharing. It will also define the mandated open standards and technologies required to realize this coherent grouping of middleware functions/services.

### **Trust Domain**

A Trust Domain is an administered security space. Within that space it is possible to specify the set of credentials a particular actor shall provide to be trusted by another actor of the security space. This specification is part of the security policy in place in the domain. Usually the credentials are combined in a digital certificate that is signed by the certification authority of the security space.

### **Virtualisation**

Virtualisation is a technique to improve hardware resource utilisation and scalability; by creating virtual copies of resources such as a hardware platforms, operating systems, storage devices, or network

<sup>8</sup> André B. Bondi, 'Characteristics of scalability and their impact on performance', Proceedings of the 2nd international workshop on Software and performance, Pages 195-203

<sup>9</sup> Load Balancing Servers, Firewalls, and Caches, Chandra Kopparapu, Wiley Computer Publishing, John Wiley & Sons, Inc.

resources. Administrative tasks are eased and accessible via a central 'location' and overall availability is less reliant on physical hardware.

Hardware platform virtualisation creates virtual machines that act like real computers with an operating system each. A virtual machine can easily be relocated to a different physical location in case of 'physical' hardware failure and restarted or reconfigured transparently (with no interruption of service) to service consumers (clients). When more resources are required for servicing an increasing number of service consumers, an administrator allocates more hardware resources the virtual machine without significant impact of the service consumers.

Virtualisation is currently used for both scalability and fault tolerance.

### **Yellow Profile**

Many types of information sharing in ATM do not have an immediate high safety critical context and can be satisfied by infrastructure that is less demanding and less sophisticated. Many services can be satisfied by a middleware providing generic functionality with a "Best Effort" QoS.

SWIM-TI's Yellow Profile (YP)<sup>10</sup> is explicitly targeted at:

- Support for a wide variety of interactions in a flexible manner and that is affordable for the service consumer.
- The interaction must be able to run over Internet and must be sufficiently secured
- Use of technology based on the Web Services stack of standards
- The technology must be supported out-of-the-box by the mainstream development frameworks as well as mainstream execution frameworks.
- Keeping as many options open as possible re. deployment.

---

<sup>10</sup> The interested reader can consult related ADD terminology in section 2.2.1 or Yellow Profile SPA in section 2.4.1 in YP TS.



## 2 Architecture of the System

Describing Systems is a complex task. The Architecture of a System intends to facilitate such description by providing its decomposition into smaller entities, further describing such smaller entities and by providing the relationships among them (whose addition represents the whole System) via different views.

It has to be noted that SWIM-TI is already a part of a bigger System (referred to as System of Systems, SoS or European ATM). The description of the overall System of Systems is provided by the ADD (ref. [6]) and accessible in the European ATM Architecture (EATMA) portal.

The following definitions are provided in the ADD:

<b>Capability</b>	Capability is the ability of one or more of the enterprise's resources to deliver a specified type of effect or a specified course of action to the enterprise stakeholders ( <i>Source: NAF [64]</i> ).
<b>Capability Configuration</b>	A <b>Capability Configuration (CC)</b> is a combination of Human Resources and Systems (and their Functional Blocks) configured to provide a Capability derived from operational and/or business need(s) of a stakeholder type.

The ADD (ref. [6]) provides an overall view of the European ATM as a Federation of Capability Configurations; according to the ADD and according to the above definitions, SWIM-TI can be understood as an Infrastructure Capability Configuration.

The EATMA is an architecture framework and repository of content and programme-related information that provides the structure to the work of the 300 projects that have made up the first SESAR R&I activities (SESAR 1). In doing so, the EATMA ensures a coherent architecture framework for developing interoperable solutions, as well as for identifying gaps or duplication of technical system work between projects. As a framework, the EATMA federates the performance framework as well as operational, systems and service architectures. The EATMA is accessible through the European ATM Portal that captures, maintains, validates and reports on architecture-related content (<https://www.eatmportal.eu/working/signin/>).

The integration of SWIM-TI into the ADD is further analysed in Appendix A.

The section 2.2 describes the functional decomposition of the SWIM-TI and section 2.3 identifies the possible technical components that can realize the functions with the main objective to make interoperate the ATM systems. More considerations on the functional and technical views are given in section 2.1.

Please, note that SWIM Node concept used in 2.2 is defined in section 2.3.

## 2.1 Ontology, terminology, relationships and semantics

### 2.1.1 Introduction

### 2.1.2 Distributed system (of systems)

There is no authoritative definition of the term “Distributed System” and hence multiple definitions can be found.

One such definition is given by Andrew S. Tanenbaum & Maarten Van Steen in Distributed Systems, Principles and Paradigms, second Edition:

*A distributed system is a collection of independent computers that appear to its users as a single coherent system.*

Another definition can be found on Wiki:

*A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal.*

The SWIM-TI is to be seen as a Distributed System of Systems.

The use of the term Distributed System of Systems stresses the autonomy of the systems that participate in the overall Distributed System. Despite the absence of the level of “control” by a single authority and not being under the responsibility of a single organisation, the SWIM-TI needs the establishment and enforcement of governing rules and procedures to ensure service interoperability.

Following characteristics can be observed for the SWIM-TI:

- behaves as a single coherent system for its users
- the common goal of this system is targeted at provision of interoperability to its users
- the components of this system interact through a network, whether they are distributed geographically or geographically/physically co-located
- is not operated as a single coherent system from deployment and technical supervision point of view

### 2.1.3 Middleware in a Distributed system (of systems)

The main objective of the middleware layer in a distributed system consists of providing interoperability to the applications.

The layering at this point creates a boundary with the applications that rely on the middleware layer and targets a decoupling of the applications from the technology.

### 2.1.4 Architectural Views

#### 2.1.4.1 Introduction

##### 2.1.4.1.1 Needs

Following extract from <http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap31.html> reflects well the reason of existence of architectural views on the SWIM-TI:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

34 of 284

*In summary, then, architecture views are representations of the overall architecture in terms meaningful to stakeholders. They enable the architecture to be communicated to and understood by the stakeholders, so they can verify that the system will address their concerns.*

*Concerns are the key interests that are crucially important to the stakeholders in the system, and determine the acceptability of the system. Concerns may pertain to any aspect of the system's functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability.*

*A view is a representation of a whole system from the perspective of a related set of concerns.*

### 2.1.4.1.2 Reference material

A wide variety of documentation and insights are available on how to structure the description of an architecture (<http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html> documents examples of architecture frameworks collected by WG42 in the context of ISO/IEC/IEEE 42010:2011).

The complexity of the matter is highly increased through the use of concepts that are not clearly/unambiguously defined and through the use by distinct classifications of identical terminology with different semantics.

An example that discusses and illustrates such complexity can be found at <http://www.slideshare.net/wweinmeyer79/an-introduction-to-fundamental-architecture-concepts-25828722>.

The matter of architecture views is dealt with in:

- ISO/IEC/IEEE 42010:2011 Systems and software engineering -- Architecture description, [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=50508](http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508)
- in a variety of Architectural Frameworks such as TOGAF, DODAF and NAF.
- by Kruchten in The "4+1" View Model of Software Architecture ([http://en.wikipedia.org/wiki/4%2B1\\_architectural\\_view\\_model](http://en.wikipedia.org/wiki/4%2B1_architectural_view_model))
- specific frameworks such as IBM View and Viewpoints Framework ([http://www.ibm.com/developerworks/rational/library/08/0108\\_cooks-cripps-spaas/index.html](http://www.ibm.com/developerworks/rational/library/08/0108_cooks-cripps-spaas/index.html)).

### 2.1.4.1.3 Approach

In order to be able to describe the architecture of the SWIM-TI in a sufficiently detailed and structured manner, it is proposed that the description of the architecture of the SWIM-TI is approached as a system as used in ISO/IEC/IEEE 42010:2011: a system is a placeholder whereby it could refer to a subsystem as well as to a distributed system of systems.

Also the different views are provided based on a definition that is specific for the SWIM-TI due to

- the lack of precision in the existing classifications
- the difficulty to map the SWIM-TI as a distributed system or system itself
- the tendency to limit the scope of the "technical" part or technology related part

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

35 of 284

- the need to remain focussed on what is relevant

At the highest level of abstraction of the SWIM-TI itself, it is mapped to a single complex system in the System View of the EATMA framework. The interfaces of the SWIM-TI with the external entities such as ATM specific services and the Network are described in the Service View, which references the Technical view.

### 2.1.4.2 Functional view

From the highest level of abstraction, the SWIM-TI is considered as a single entity that provides and possibly consumes services. It implements technical functions with the purpose of ensuring interoperability of systems using the SWIM-TI.

The Functional decomposition recursively breaks this single system into smaller and more granular functions with particular focus on the provision and consumption of services and the interfaces through which these service interactions take place.

- At the highest level, the SWIM-TI can be seen as a single system that provides an interoperability service. At this level the functional decomposition view shows the interoperability with external actors.
- A Functional decomposition starting at the highest level reveals a function such as messaging.
- The messaging function itself can be broken down into a series of smaller functions such as routing, distribution and bridging.

The Functional decomposition of the SWIM-TI will yield:

- a detailed insight in what functionality the SWIM-TI must provide,
- the interaction and dependency between the functions at various levels of detail.

The Functional decomposition deals with the “what” aspect of the SWIM-TI.

- The Functional decomposition does not address the “where” or “how” aspects of the SWIM-TI.
- In particular the Functional decomposition remains entirely independent of and totally silent about specific technology. This allows the Functional view of the SWIM-TI to persist and remain valid across technology changes.

The Functional decomposition of the SWIM-TI is normative for SWIM.

### 2.1.4.3 Technical view

#### 2.1.4.3.1 The foundation

The technical view of the SWIM-TI will:

- identify the possible architectural components that can provide/realize/deploy the functions of the Functional decomposition view of the SWIM-TI,
- will describe the technology that can be used to realize the functions of the Functional decomposition view of the SWIM-TI,
- will describe the possible organizations of such components in a network.

The technical view will demonstrate that the organization of such components in a network will support the basic forms following a distributed, a federated and/or a centralized model, or any hybrid combination of such basic forms for each function for which this is deemed relevant and whereby this is compatible with the documented design principles and decisions.

The technical view deals with the “how” aspect of the SWIM-TI.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

The technical view of the SWIM-TI is normative for SWIM.

## 2.1.4.3.2 The SWIM-TI Node

### 2.1.4.3.2.1 Reference

The SWIM-TI is a Distributed System (of Systems). The Node is a key component of a Distributed System.

The ArchiMate specification from The Open Group states  
([http://pubs.opengroup.org/architecture/archimate-doc/ts\\_archimate/chap6.html](http://pubs.opengroup.org/architecture/archimate-doc/ts_archimate/chap6.html))

*The main structural concept for the technology layer is the node. This concept is used to model structural entities in this layer. It is identical to the node concept of UML 2.0. It strictly models the structural aspect of a system: its behaviour is modelled by an explicit relationship to the behavioural concepts*

The concept of Node is widely used. There are many variants in the semantics and the context determines how to interpret/understand. The node concept appears in all major Enterprise Architecture Frameworks such as DODAF, NAF, TOGAF.

Despite the many variations, generally speaking, a Node is an autonomous point of presence in a Distributed System that interacts with other Nodes in the Distributed System.

The point of presence makes a set of functionality on one Node available to any Node or allows one or more Nodes to use the functionality that is made available by a Node.

The autonomy of a Node reflects the freedom re. physical implementation/deployment choices (such as hardware/virtualisation, OS, high availability techniques and platform) and governance (such as development process, running of operations and definition of policies).

Interaction implies the availability of means for bi-directional access to the Distributed System.

### 2.1.4.3.2.2 Node in the context of the SWIM-TI

The key component that can provide/realize/deploy the functions of the Functional decomposition view of the SWIM-TI is the SWIM-TI Node.

A SWIM-TI Node is an autonomous point of presence in the Distributed System (of Systems) that interacts with other SWIM-TI Nodes in the Distributed System (of Systems).

The point of presence makes a set of functionality via one SWIM-TI Node available to any SWIM-TI Node or allows use of the functionality that is made available by a SWIM-TI Node via one or more SWIM-TI Nodes.

The autonomy of a SWIM-TI Node reflects the freedom re. implementation choices (such as but not limited to hardware/virtualisation, OS, high availability techniques and platform) and governance (such as but not limited to development process, running of operations and definition of policies).

The SWIM-TI Node is a generic element that could be specialised in categories. At the time of writing no need for such specialisation has emerged.

At the time of writing, there are two categories of specifications. The notion of SWIM-TI Node is suitable for the realization of functions grouped and organised in whatever form.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

- The next chapter defines the first category of specifications that are captured and grouped under the notions of SWIM Profile, Profile Part, Role and Self-standing set.
- The second category of specifications consists of those captured and grouped under the notions of shareable functions.

### 2.1.4.3.2.3 The definition of SWIM-TI Node

SWIM-TI Node provides one or more collections of SWIM-TI functions, grouped in accordance with deployment conformance specifications. A SWIM-TI Node allows a given ATM application to use the SWIM-TI and/or a SWIM-TI Node supports the SWIM-TI.

### 2.1.4.3.3 The organisation

The possible organisation of components where functions are realized/deployed are using the key concepts distributed, federated, centralized, or any hybrid form or composition of these concepts.

These key concepts are further detailed in 2.1.5.

### 2.1.4.3.4 System Ports vs. Interface Bindings

In the terminology of NAF and B.04.03, the interconnection of “Technical Systems” is based on “System Ports” and “System Port Connectors”, as depicted in Figure 1, where “a *System Port* is an interface provided by a Technical System.” and “a *System Port Connector* asserts that a connection exists between two System Ports”.



**Figure 1 – Connecting Technical Systems via System Ports**

System Ports and System Port Connectors are used to describe in detail the physical communication means used to transport the data from one system to another. To do so, the following elements are involved:

- System Port specification, defining the ports on each system and the protocols/standards relevant for the port (HTTP, SOAP, TCP, IP, etc.)
- System Port connectivity defined as a System Port Connector, i.e., the communication link between two system ports, which is enabled by a specific communication system (e.g. the SWIM-TI, Communications infrastructure).

The B.04.03 ADD (ref. [6]) defines the system port as follows:

<b>System Port</b>	<p>A <b>System Port</b> is an interface provided by a System.</p> <p>A <b>System Port Connector</b> asserts that a connection exists between two System Ports.</p>
--------------------	--

On the other hand, the WP14 defines several kinds of Interface Bindings<sup>11</sup>, as depicted in Figure 2. A variety of interface bindings is defined in the various SWIM profiles [13].

Note that the SWIM-TI is not seen as a “Technical System” at the same level as the Technical Systems in Figure 1; the SWIM-TI is an Infrastructure System (see “Communication system” above) providing the means for realizing the connectivity between systems.

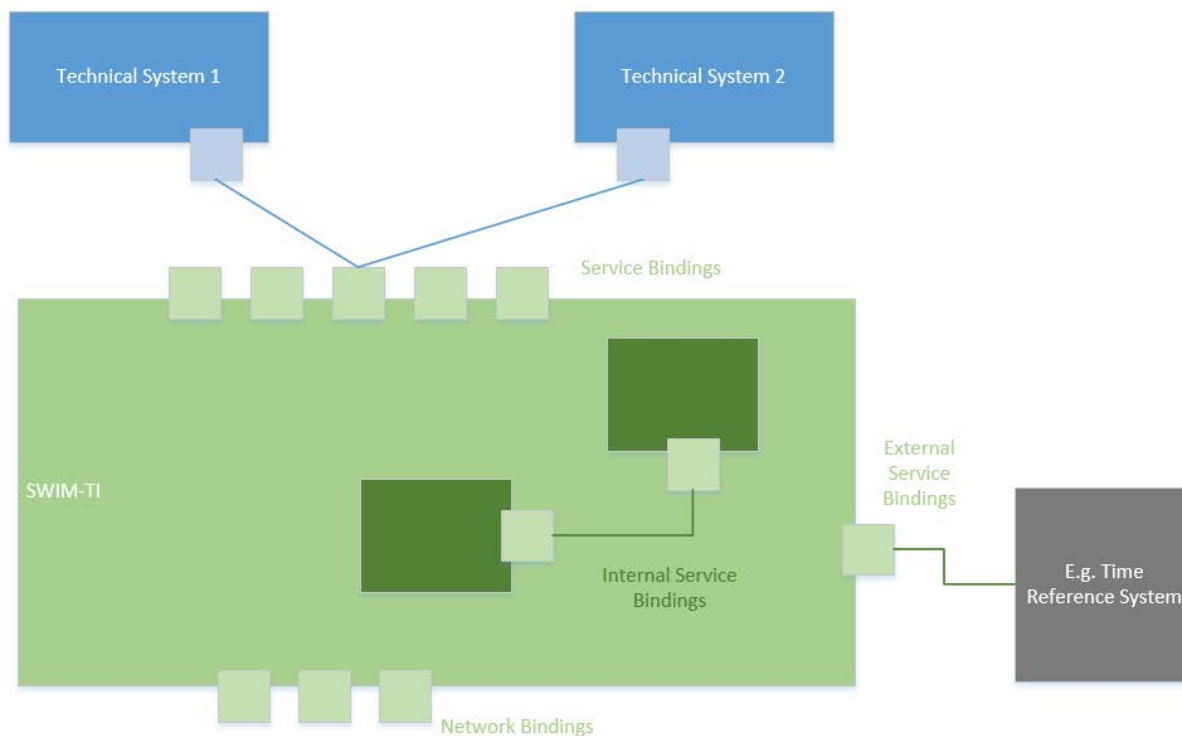


Figure 2 – SWIM-TI Interface Bindings

This figure shows the different kinds of interface bindings defined in WP14:

- “Service Bindings” define the protocol stacks for ATM Systems. These are the primary access specifications for systems using the SWIM-TI for interacting.
- “Internal Service Bindings” define the protocol stacks to be used between sub-components of the SWIM-TI.
- “Network Bindings” define the protocol used by the SWIM-TI to interact with the underlying network infrastructure
- “External Service Bindings” define the protocol for communication with other (external) infrastructure systems

As visualised in Figure 2, there is a close relationship between the System Ports of Technical (ATM-) Systems and the Service Bindings of the SWIM-TI:

<sup>11</sup> Note that the term “binding” is not defined in NAF. Nevertheless, WP14 uses the term “Binding” to define the protocols/standards for system interoperability. According to NAF, the System Port is an architecture element that is used to define the protocols/standards for data exchange between systems.

In the example above, the two Technical Systems interact with each other via System Port. The necessary System Port details (protocols/standards relevant for the port (HTTP, SOAP, TCP...)) are described in the SWIM-TI Service Bindings. This means, SWIM-TI Service Bindings may be used to describe Interface Ports for technical systems.

As a summary: although System Ports and Service Bindings are very similar, they are not the same. While System Ports are the interaction end-points provided by individual systems, the Service Bindings are definitions of protocol stacks for these interaction end-points. Strictly speaking: a system does not interact with the SWIM-TI via a service binding; a system interacts with another system via a service binding offered by the SWIM-TI.

In WP14, the actual service endpoint is composed by SWIM Profile Interface Binding + (service) contract part. The contract part is service interface specific (e.g. WSDL).

## 2.1.4.4 Deployment view

### 2.1.4.4.1 The foundation

The effective choice of Deployment Architecture is not made by WP14 but by the interested Stakeholders. Deployment views are nevertheless provided for two reasons:

- as a demonstration that the distinct Architecture views as well as the overall Architecture are consistent and effectively allow distinct deployment options,
- to illustrate concrete distinct deployment options in distinct environments and for different Stakeholder needs as a clarification for the reader.

Deployment options are constrained:

- Deployment options should not break interoperability,
- Deployment options should not be in contradiction with documented design decisions and principles,
- If any additional requirement is needed to support such deployment option it shall be identified and specified.

The Deployment view of the SWIM-TI deals with the “where” aspect of the SWIM-TI.

The Deployment view of the SWIM-TI is not normative for SWIM. The Deployment view of the SWIM-TI is illustrative for SWIM.

### 2.1.4.4.2 Distributed, Federated and Centralized

For each function the deployment view will identify the model that is chosen to organize the components that realize the function. The ontology to be used is described in the Organisation Models

### 2.1.4.4.3 Physical elements

A deployment view may describe the elements at lowest level of abstraction, i.e. the physical details such as physical addresses, geographic locations, networks, servers, virtualization, clustering, operating system, application server, execution framework and resilience.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



However in the context of the SWIM-TI, the physical details have no relevance and it is more useful to keep the physical elements out of the deployment view.

## 2.1.5 Organisation models

### 2.1.5.1 Overarching ontology for organisation

This ontology impacts the technical view and deployment view.

The SWIM-TI consists of a set of interconnected systems offering various functions. The functions are related to ATM specific services as well as to the SWIM-TI itself. The SWIM-TI Architecture and Design targets agnosticity regarding the deployment of the functions in the SWIM-TI. This means agnosticity regarding the technical organisation and/or the organisation of responsibilities.

Some forms of organisation introduce interoperability, functional and/or system behaviour related requirements that are not necessary in other forms of organisation. The SWIM-TI Architecture and Design needs to make such requirements visible and explicit to ensure that all forms of organisation can be dealt with. Conversely, it must be clear to which form of organisation, the requirements are applicable. Hence, a non-ambiguous classification of distinct forms of organisation is needed to describe common (i.e. mainstream, frequently occurring) forms of organisation and to enable a shared understanding amongst the impacted Stakeholders.

Such classification is not a trivial given because:

- Typically there are a number of variants of a high level classification but they overlap in terminology, sometimes with different semantics or use of different terminology with overlapping semantics. Such classifications typically use terms such as centralized, decentralized, distributed, federated and hybrid.
- Depending on the level of abstraction the organisation as seen at one level can be different from the organisation as seen at another level. A typical element of this mechanism, is the recursivity of the notion system of systems: each system in a system of systems could be seen as a system of systems itself and conversely any number of system of systems could be abstracted into a single system of systems. Such recursion stops when a system is under a single instance of control and governance.

To avoid misinterpretations and false expectations, a clear and un-ambiguous reference definition of high-level terminology and relations is required.

An interesting overview is provided by Stijn Peeters at <http://networkcultures.org/unlikeus/resources/articles/what-is-a-federated-network/>.

Sources of ambiguity:

- different views on decentralized versus distributed,
  - decentralized organisation and distributed organisation are considered synonyms,
  - decentralized is hierarchy of distributed organisation itself consisting of centralized organisations,
- different views on federation versus decentralized and distributed,
  - federated organisation and decentralized organisation are considered synonyms
  - federated organisation and distributed organisation are 2 distinct forms of decentralized organisation.

Based on what is considered “the most sensible approach” by Stijn Peeters:

- distributed organisation: each element can collaborate directly with each other element,

founding members



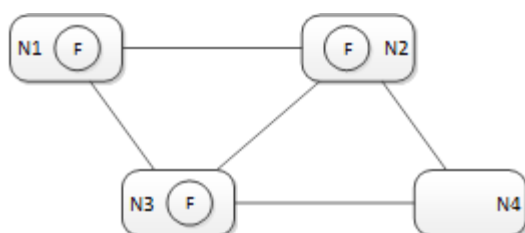
Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

- centralized organisation: all collaboration between elements goes through a single intermediary,
- decentralized organisation is an umbrella concept that includes:
  - distributed organisation,
  - federated organisation,
- federated organisation is a distributed organisation of multiple centralized organisations. A federation is an association of users, service providers, and identity service providers.<sup>12</sup>

Any use of distributed, federated, centralized and decentralized should conform to above semantics. In case a deviation of semantics for any of these terms is needed, then this deviation should be explicitly indicated and the mapping with above ontology and semantics be provided.

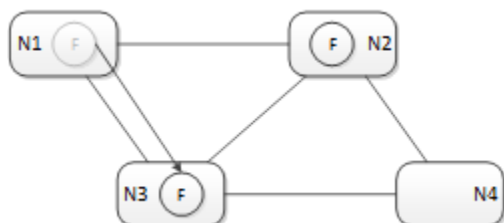
### 2.1.5.2 Illustrations

In a fully Distributed model, a function is realized at each point of presence where the function is needed in the Distributed System.



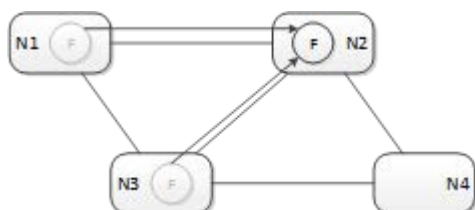
Function F is realized at each of the nodes N1 to N3, assuming that F is not needed at N4

In a Federated model, a function is realized at some points of presence in the Distributed System and shared with a limited group of other points of presence in the Distributed System. Distinct groups that share a function, share this function with other groups to allow collaboration across distinct groups.



Function F is realized by nodes N2 and N3 only, node N1 accesses F from N3 (N3 shares F with N1), N4 does not access F (not needed at N4)

In a Centralized model, a function is realized at a single point of presence in the Distributed System and shared with all other points of presence in the Distributed System.



F is implemented in N2 only, all other nodes needing F have to access N2

The Technical Architecture will identify on which SWIM-TI architectural element, a SWIM-TI function can be allocated. The Deployment Architecture will document where a SWIM-TI function will be

<sup>12</sup> According to ITU-T IdM X.1252

allocated. Allocation is to be understood as the location where the realization of the function takes place.

## 2.1.6 Broker

Impact: functional view, technical view, deployment view

The term Broker is often used in the context of distributed middleware. Some examples:

- Authentication Broker,
- Web Authentication Broker,
- Identity Federation Broker,
- Identity Trust Broker,
- Service Broker,
- Policy Broker,
- Routing Broker,
- Message Broker,
- “lightweight broker-based publish/subscribe messaging protocol”.

In most cases there is no clear/unambiguous definition of the term Broker and even when qualified, such as Message Broker, the semantics of such qualified homonyms can vary significantly and remain often implicit.

The lack of precision in the use of the term Broker, is an important potential source of misalignment, of difficulty or inability to understand and ultimately of unsuccessful interoperability.

In order to avoid diverging interpretations of the term Broker each use needs to use the SWIM-TI dictionary that is provided for the SWIM-TI. If an interpretation is required that diverges from the SWIM-TI dictionary, then such interpretation must be explained for each occurrence of such divergence.

## 2.1.7 Conformance and interoperable implementations

### 2.1.7.1 Analysis

#### 2.1.7.1.1 Need

The main objective of the SWIM-TI is to provide interoperability. In theory, interoperability is best when there are numerous identical, complete, and correct implementations (citation from Variability in Specifications, W3C Working Group Note 31 August 2005).

If that were technically possible, such a complete implementation of all the technology, functional and non-functional specifications needed to satisfy all the interoperability needs on the SWIM-TI, would be too large and/or too expensive for a significant number of implementers.

Additionally, for reasons such as constraints, competing requirements and risks, a single complete implementation is not possible and/or acceptable (see Chapter 2 of P14.1.3-D34 SWIM Profiles for Step 2 for more detailed information).

Hence the conclusion "one size does not fit all" from an implementation perspective and the need to modularise the specifications from an implementation perspective.

Modularisation introduces complexity in interoperability: only specific combinations will interoperate. Modularisation may impede interoperability but, when carefully chosen, modularisation may also enhance interoperability.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

43 of 284

To avoid diverging interpretations of such modularisation by implementers and/or users, and to promote interoperability, at least one and possibly more conformance clauses are needed for each identified entity in such modularisation. The conformance clause is a part or collection of parts of a specification that defines the requirements, criteria, or conditions that shall be satisfied by an implementation in order to claim conformance (citation from OASIS Conformance Requirements for Specifications v1.0, Committee Specification 15 March 2002)

### 2.1.7.1.2 Reference material

A series of prominent standardization bodies (e.g. IEEE, ISO/IEC, OGC, OASIS and W3C) all provide standards and/or guidelines that describe in general how to write specifications:

IEEE: <http://www.ietf.org/rfc/rfc2119.txt>

ISO/IEC: 10000-1: Information technology — Framework and taxonomy of International Standardized Profiles — Part 1: General principles and documentation framework

ISO/IEC: ISO/IEC Guide 2:2004 Standardization and related activities – General vocabulary

ISO/IEC: ISO/IEC Directives Part 2: Rules for the structure and drafting of International Standards, 2004

OGC, The Specification Model — A Standard for Modular specifications, OGC 08-131r3

OASIS, Conformance Requirements for Specifications v1.0, Committee Specification 15 March 2002  
[http://www.oasis-open.org/committees/download.php/305/conformance\\_requirements-v1.pdf](http://www.oasis-open.org/committees/download.php/305/conformance_requirements-v1.pdf)

OASIS, Guidelines to Writing Conformance Clauses revision 25 April 2014

W3C, QA Framework: Specification Guidelines, W3C Recommendation 17 August 2005,  
<http://www.w3.org/TR/qaframe-spec/>

Similarly such standardization bodies provide publications (in the form of draft document, specifications and/or guidelines) that specifically and explicitly address the need for modularisation:

OASIS, Conformance Requirements for Specifications v1.0, Committee Specification 15 March 2002  
[http://www.oasis-open.org/committees/download.php/305/conformance\\_requirements-v1.pdf](http://www.oasis-open.org/committees/download.php/305/conformance_requirements-v1.pdf)

W3C, Variability in Specifications, W3C Working Group Note 31 August 2005  
<http://www.w3.org/TR/spec-variability/>

W3C, QA Framework: Specification Guidelines, W3C Recommendation 17 August 2005,  
<http://www.w3.org/TR/qaframe-spec/>

### 2.1.7.1.3 Approach

It is proposed to use a model to capture the variability in the specifications in a similar but not identical manner to the principles set out in "Variability in Specifications" (W3C Working Group Note 31 August 2005).

At the time of writing, not all seven "dimensions of variability" are considered relevant for the SWIM-TI: four dimensions are considered relevant. The dimensions "Profile", "Functional levels", "Class of Product" and "Modules" have been reused as is or with a slight adaptation to fit the purpose of the SWIM-TI.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

44 of 284

The table below provides an overview of the four dimensions of variability considered relevant for the SWIM-TI. Each of the elements is further elaborated in the following chapters.

Dimension	Scope	Sub-grouping of	Ensures
Profile	SPA	All possible solutions	A solution type for information sharing that is aligned with the high-level criteria of the Stakeholders of a Domain.
Profile Part	Large amount of shared specifications with variations on completeness	Profile	Interoperability when all involved participants stick to the specifications.
Role	Behaviours/functions to be assumed	Profile Part	Focused and coherent specifications for behaviours/functions avoiding overkill and ensuring separation of concerns.
Self-standing Set	Deployment of specifications	Profile Part	Flexibility at deployment while maintaining coherence.

Each entity of the modularisation will have one or more accompanying conformance clauses.

## 2.1.7.2 Structure

### 2.1.7.2.1 Concepts

#### 2.1.7.2.1.1 SWIM Profile

"Dimensions of variability" defines the notion of "Profile":

*to define a subset of a set of technologies defining how they are required to operate together.*

A SWIM Profile extends this definition by also including functional and non-functional requirements:

*a SWIM profile is a coherent, appropriately-sized grouping of middleware functions/services for a given set of technical constraints/requirements that permit a set of stakeholders to realize Information sharing. It will also define the mandated open standards and technologies required to realize this coherent grouping of middleware functions/services.*

#### Example:

The set of messaging technologies used includes AMQP, DDS and Web-Services. A single complete and correct implementation of all these messaging technologies would represent a huge footprint: for instance DDS has many distinct Quality of Service, leading for a large number of possible combinations, Web-Services over SOAP allows for composition with a series of other WS-\* standards leading to a large amount of possible combinations.

A SWIM Profile will select subsets of such technologies that provide a solution for information sharing for a given set of objectives.

A particular type of application of the notion of "Profile" is illustrated through of the OASIS WS-I Profiles.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

The OASIS WS-I Profiles are entirely separated specifications that define subsets of SOAP, WDSL, WS-“ and bindings and how they are required to interoperate.

The OASIS WS-I Profiles target to promote interoperability by removing, limiting or better specifying the extension points in the existing standards.

Where applicable to the SWIM-TI such “Profile” specifications are included in the specifications.

### 2.1.7.2.1.2 Profile Part

"Dimensions of variability" defines the notion of "Functional levels":

*Functional levels — or in common usage simply levels — are used to group functionality into nested subsets, ranging from minimal or core functionality to full or complete functionality. Level 1 is the minimum or core of the technology. Level 2 includes all of level 1 plus additional functionality. This nesting continues until level n, which consists of the entire technology.*

A Profile Part is used to group functionality into subsets ranging from minimal or core functionality to full or complete functionality. The difference with the notion of Level is that Profile Parts are not necessarily nested (called stacked pattern) but can also exist side-by-side.

The “Profile Part” is the encompassing and coordinating entity that ensures the coherent information sharing by all participants in a Plug & Play manner.

It suffices for a participant to stick to the specifications of a Profile Part in a conforming manner to be ensured of coherent information sharing with other participants in the context of the same Profile Part.

#### Example:

A SWIM Profile consists of Core Part that represents minimal functionality. Three additional side-by-side Profile Parts have been defined: Messaging+, Security+ and Advanced that can exist in any combination with each other on top of the Core Part. The full or complete functionality is represented by the presence of all Profile Parts: Core, Messaging+, Security+ and Advanced.

### 2.1.7.2.1.3 Role

"Dimensions of variability" defines the notion of "Class of Product":

*The class of product separates the different kinds of implementations a specification may have.*

A Role corresponds fully to the notion of "Class of Product".

Despite identical semantics, the term "Role" is used instead of "Class of Product". This to avoid possible misinterpretation: a Product could be understood as the packaging boundary of the solution provided by an implementer. The packaging boundary of the solution provided by an implementer could be limited to a single Role but could as well include multiple Roles.

#### Example:

Within the set of WS-\* technology, each of following kinds of implementations can be distinguished and can exist separately: service provider, service consumer, WS-N broker, notification consumer, notification subscriber, notification provider, Secure Token Service, identity provider, relying party<sup>13</sup>

<sup>13</sup> ITU-T IdM X.1252 define this term as an entity that relies on an identity representation or claim by a requesting/asserting entity within some request context.

#### 2.1.7.2.1.4 Self-standing Set

"Dimensions of variability" defines the notion of "Modules":

*Modules are discrete divisions or functional groupings of the technology and do not necessarily fit in a simple hierarchical structure.*

*Modules generally can be implemented independently of one another — e.g., audio vs. video module. That notwithstanding, it is possible for one module's definition (and therefore implementation) to have explicit dependency upon another module. It is not only possible, but also common to implement multiple modules.*

A Self-standing Set corresponds fully to the notion of "Module".

Despite identical semantics, the term "Self-standing Set" is used instead of "Module". This to avoid possible confusion: the term Module is more generic and can be misinterpreted.

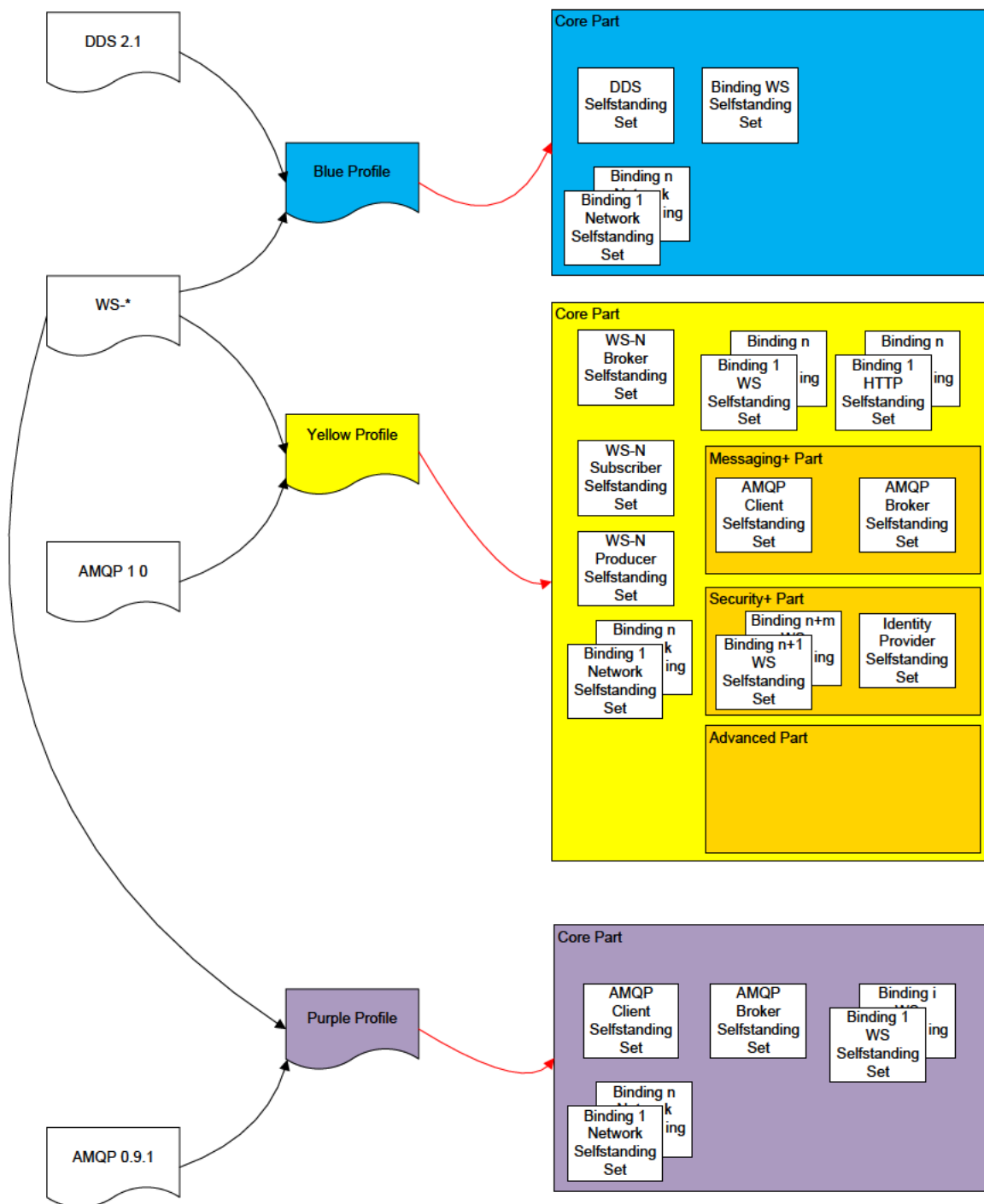
#### Example:

Within the set of WS-\* technology a large number protocol stacks are possible. To facilitate implementation and to improve interoperability by focus on particular needs, a set of discrete protocol stacks are selected. These protocol stacks are not part of a hierarchy.

#### 2.1.7.2.2 Integration and organisation of SWIM Profile, Part, Role and Self-standing Set

SWIM Profile, Part, Role and Self-standing Set represent distinct dimensions that can be combined.

- The figure below depicts 3 of these elements and their relation.



**Figure 3 – SWIM profile, Part, Role and Self-standing Set**

The left side of the figure, contains complete specifications of technologies such as AMQP 0.9.1, AMQP 1.0, DDS v2.1 and WS-\*. This is not an exhaustive list but serves as an illustration only.

The middle of the figure, contains SWIM Profiles such as Blue Profile, Purple Profile and Yellow Profile. The black lines with arrow illustrate the technologies whereof the SWIM Profiles create subsets.

The right side of the figure, contains Profile Parts and Self-standing Sets. The red lines with arrow illustrate the constituent elements of a SWIM Profile. Each SWIM Profile has a Core Part. The Yellow Profile has 3 additional Profile Parts.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



Each Profile Part contains 0 to n Self-standing Sets. The Advanced Part of the Yellow Profile does not contain a Self-standing Set: this can occur when a Profile Part only has specifications related to non-functional requirements.

Whether Self-standing Sets are applicable or not, depends on the Role. The Roles are not reflected in this figure. The figures below illustrate the mechanism of Roles using the Yellow Profile.

The figure below depicts the relevant Self-standing Sets for a Consumer role of the Yellow Profile Core Part.

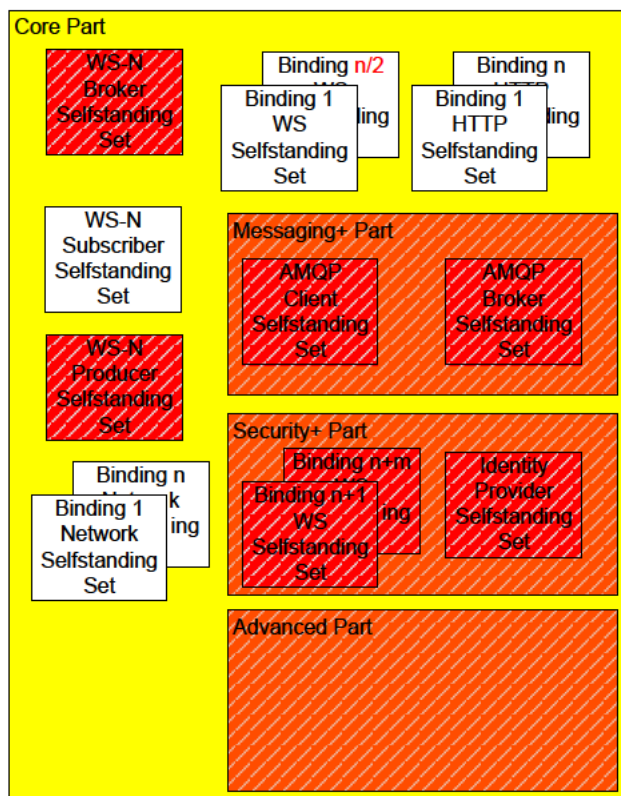
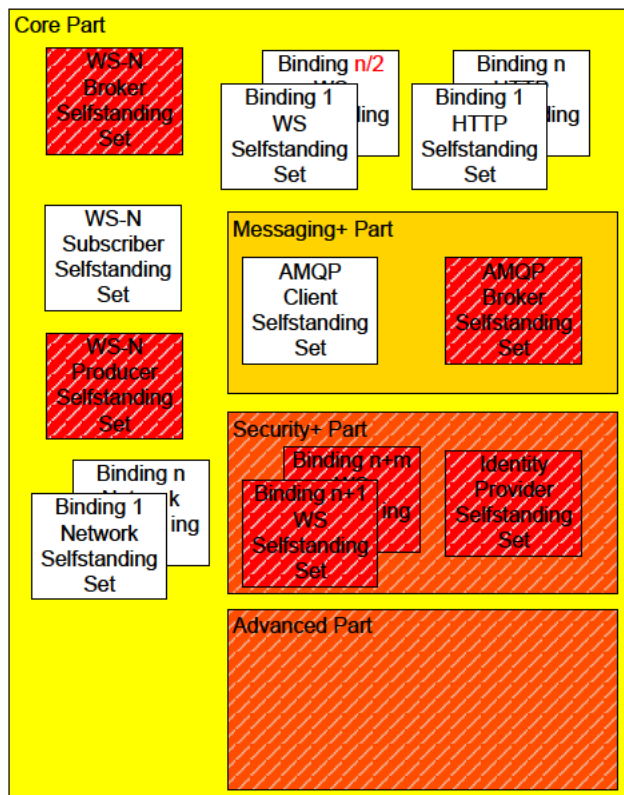


Figure 4 – Relevant Self-standing Sets: Consumer role

Any implementation that conforms to these Self-standing Sets, will be able to interoperate as consumer with any other role in the Yellow Profile Core Part whatever the choices made in that other role. Note that n has become n/2 in the WS Self-standing Sets: this because a consumer can chose to use SOAP1.1 or SOAP1.2 and does not have to support both.

The figure below depicts the relevant Self-standing Sets for a Consumer role of the Yellow Profile Core Part combined with Messaging+ Part.



**Figure 5 – Relevant Self-standing Sets: Consumer role**

Any implementation that conforms to these Self-standing Sets, will be able to interoperate as consumer with any other role in the Yellow Profile Core Part combined with Messaging+ Part whatever the choices made in that other role.

The figure below depicts the relevant Self-standing Sets for a Provider role of the Yellow Profile Core Part.

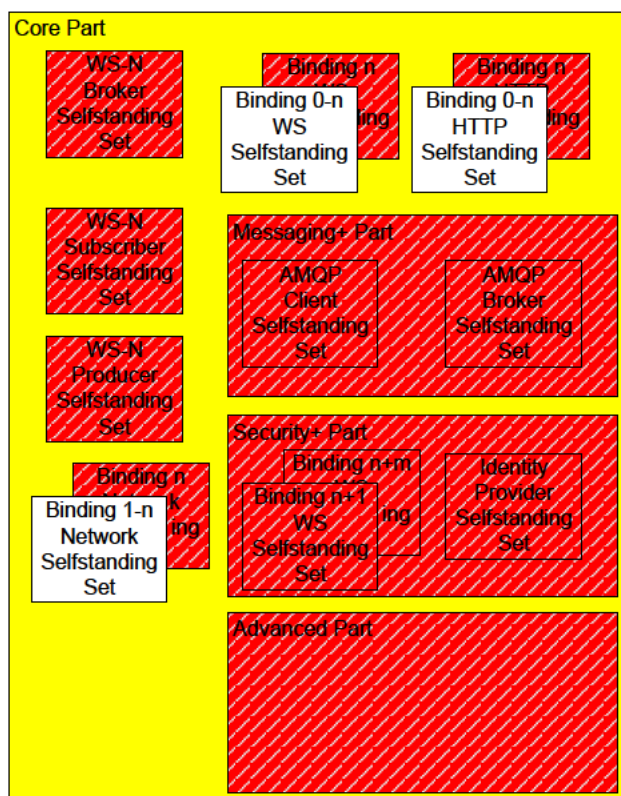


Figure 6 – Relevant Self-standing Sets: Provider role

Any implementation that conforms to these Self-standing Sets, will be able to provide services that can be consumed by any consumer role in the Yellow Profile Core Part. Note that the applicable number of Self-standing Sets linked to binding varies between 0 and n or 1 and n. This reflects the choice that is available to the provider role to only implement the technology required in the context of the provider.

### 2.1.7.2.3 Conformance

Each of the concepts (SWIM Profile, Profile Part, Role and Self-standing Set) shall have one or more specifications categorised as a conformance specification, that set the conditions, the criteria and the requirements for claiming conformance.

## 2.2 Functional View

The purpose of the Functional View is to provide a common structure for the specification of technical requirements by P14.01.04 which will then facilitate the assurance of completeness and consistency of specifications and the eventual consolidation of requirements.

From the highest level of abstraction, the SWIM-TI is considered as a single entity that provides and possibly consumes services. It implements technical functions with the purpose of ensuring interoperability of systems using the SWIM-TI. The Functional decomposition recursively breaks this single entity into smaller and more granular functions with particular focus on the provision and consumption of services and the interfaces through which these service interactions take place.

At the highest level, the SWIM-TI can be seen as a single entity that provides an interoperability service. At this level the functional decomposition view shows the interoperability with external actors.

The Functional decomposition of the SWIM-TI will yield:

- a detailed insight in what functionality the SWIM-TI must provide,
- the interaction and dependency between the functions at various levels of detail.

The Functional decomposition deals with the “what” aspect of the SWIM-TI. In particular the Functional decomposition remains entirely independent of and totally silent about specific technology. This allows the Functional view of the SWIM-TI to persist and remain valid across technology changes.

In order to understand the Architectural Definition of SWIM-TI via its Functional View, a set of definitions need to be established. The following definitions are provided in the EATMA Guidance document (ref. [63]):

<b>Function</b>	An activity which is specified in context of the resource (human or machine) that performs it. Note: Contrast with Operational Activity, where the actor performing the activity is not known (i.e. it is just a logical node). A Function is implementation-specific.
-----------------	--

<b>Functional Block</b>	A Functional Block (FB) represents a grouping of functions within a System that are assembled to assist in the conducting of one or more Operational Activities.
-------------------------	--

According to this and particularized to the SWIM Technical Infrastructure, the following terminology stands:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

<b>SWIM-TI Function</b>	A technical activity which is specified in context of the SWIM-TI. It represents the <b>lowest level of decomposition</b> of the SWIM-TI and aims for the completeness (its definition is complete and self-contained).
<b>SWIM-TI Functional Block</b>	A SWIM-TI Functional Block represents a <b>logical aggregation</b> of functions within an instance of the SWIM-TI that are assembled to assist in the conducting of one or more SWIM-TI Activities.
<b>SWIM-TI shareable function</b>	<p>A SWIM-TI shareable function is a SWIM-TI function which realization could be performed once for the benefit of several function users. Depending on the effective architecture option such a function is actually shared or not.</p> <p>A SWIM-TI shareable function may be used by SWIM-TI functions from various SWIM profiles. In this regards it does not belong to a SWIM profile in particular.</p> <p>When a SWIM-TI shareable function is used by a SWIM-TI function it interfaces it using an open standard.</p>

According to these definitions and aligned with the European ADD structure (ref. [6]), the model for the Functional View of the SWIM-TI can be depicted as:

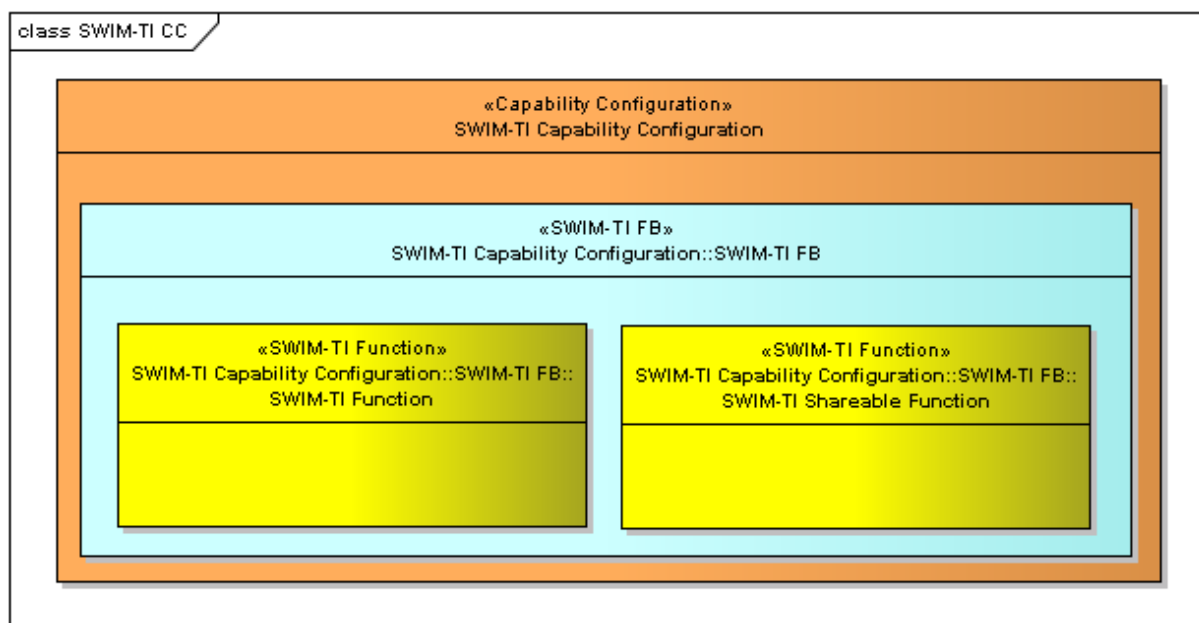


Figure 7 – SWIM-TI model

## 2.2.1 Functional breakdown

The Technical Architecture Description Document provides a functional decomposition (decomposition in SWIM-TI Functions) to be incorporated into the SESAR ADD produced by B4.3 (ref. [6]).

As such, the Functional Breakdown can be considered a “generic logical decomposition” and no inference shall be made as to the actual physical implementation.

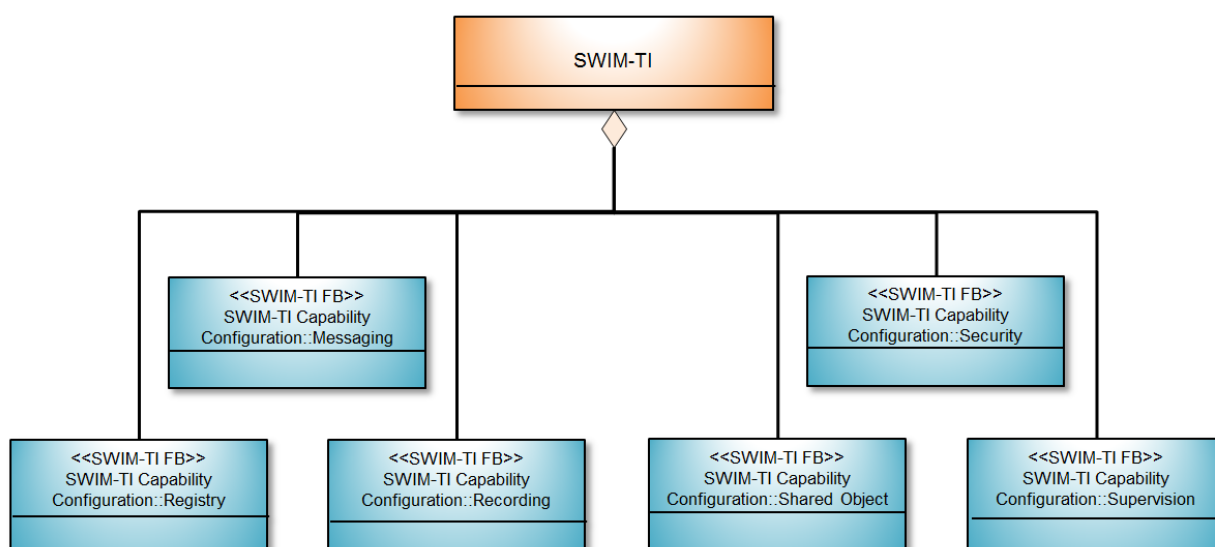


Figure 8 – SWIM-TI Functional Breakdown

Hence, SWIM-TI can be broken down to the following functional blocks:

- Registry Functional Block
- Messaging Functional Block
- Security Functional Block
- Supervision Functional Block
- Recording Functional Block
- Shared object Functional Block

As defined, a Functional Block is a **logical grouping of functions** used by other SWIM-TI Functional Blocks (shared or not) in order to ensure the correct behaviour of the SWIM-TI and the interoperability of ATM systems.

Some functions or functional blocks are tagged as shareable to report the fact that their implementation can possibly be shared through several profiles.

SWIM-TI Functional Blocks are designed to meet the SWIM ConOps requirements (see [11]) listed in the table below.

Identifier	Statement
REQ-08.01.01-CONOPS-REMP.0030	information management functions (including governance), such as operational and organisational functions for the management of user identities, discoverability of resources, security aspects such as authentication, encryption and authorization, notification services and registration shall be defined.

**Table 2 – SWIM ConOps Functional Block requirements**

As defined, a Functional Block (FB) represents a grouping of functions that together support or perform one or more Operational Activities. Likewise, A SWIM-TI Functional Block represents a logical aggregation of functions within an instance of the SWIM-TI that are assembled to assist in the conducting of one or more SWIM-TI Activities.

### 2.2.1.1 Registry Functional Block

SWIM-TI Functional Block Registry is designed to meet the SWIM ConOps requirements (see [11]) listed in the table below.

Identifier	Statement
REQ-08.01.01-CONOPS-ASDE-0010	It is assumed that the SWIM registry common component of the SWIM infrastructure is provided at the start of SWIM deployment.
REQ-08.01.01-CONOPS-ASRE.0010	It is assumed that the SWIM registry stores information about the service in all the steps of the lifecycle management.
REQ-08.01.01-CONOPS-ASRE.0020	It is assumed that it is not mandatory for the services to access the SWIM registry during runtime.
REQ-08.01.01-CONOPS-ASRE.0030	It is assumed that the SWIM registry has a public and private part. When the provider agrees and if no security conditions apply, the public part provides, in addition to generic information regarding SWIM, a high-level description of what is available in the private part.
REQ-08.01.01-CONOPS-ASRE.0040	It is assumed that the SWIM registry public part can be accessed by anyone.
REQ-08.01.01-CONOPS-ASRE.0050	It is assumed that only registered users can access the private part, provided they have the proper authorization. The scope of the information accessible to a registered user in the private part of the SWIM registry will depend on the authorization rights of the user.
REQ-08.01.01-CONOPS-ASRE.0060	It is assumed that the SWIM registry can also contain the 'non-compliant' services from SWIM Stakeholders to have a complete overview of all services available in ATM.
REQ-08.01.01-CONOPS-ASRE.0070	It is assumed that the "SWIM Collaboration Authority" manages the authorization rights for accessing the SWIM registry.
REQ-08.01.01-CONOPS-ASRE.0080	It is assumed that the "SWIM Collaboration Authority" will define the attributes to be provided during registration.
REQ-08.01.01-CONOPS-ASRE.0090	It is assumed that both the general information (web pages) on SWIM and the registration form will be publicly available for the stakeholders to submit their registration request.
REQ-08.01.01-CONOPS-ASRE.0100	It is assumed that the SWIM registry contains information (e.g. service descriptions, standards, policies, certifications, regulations ...) that is only available after registration (i.e. information in the private part of the registry).
REQ-08.01.01-CONOPS-ASRE.0110	It is assumed that the SWIM registry maintains the information on compliance.

**Table 3 – SWIM ConOps Registry requirements**

The Registry functional block provides functions to retrieve Meta-Information about the Services and the ATM Information provided by them. According to [12] registry function covers:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

56 of 284



- Discovery Functionality enables to identify registered resources, obtain their descriptions, identify related resources and follow up their evolution. It includes:
  - Search functionality.
  - Subscription functionality.
  - Highlight reporting functionality.
- Registration Functionality enables the controlled and structured registration of resources in the registry. It includes:
  - Registration workflows.
  - Information forms.
  - Categorization.
- Security Functionality enables to ensure that only authorized users are able to view or edit certain information in the registry. It includes:
  - Authentication.
  - Registry public are.
  - Registry restricted area.
  - Dual zone access.
  - Audit trail.
- System Interface Functionality enables the registry to exchange information with other systems. It includes:
  - Inter registry synchronization.
  - System queries.
  - Runtime information farming.
  - Runtime policy information exchange.

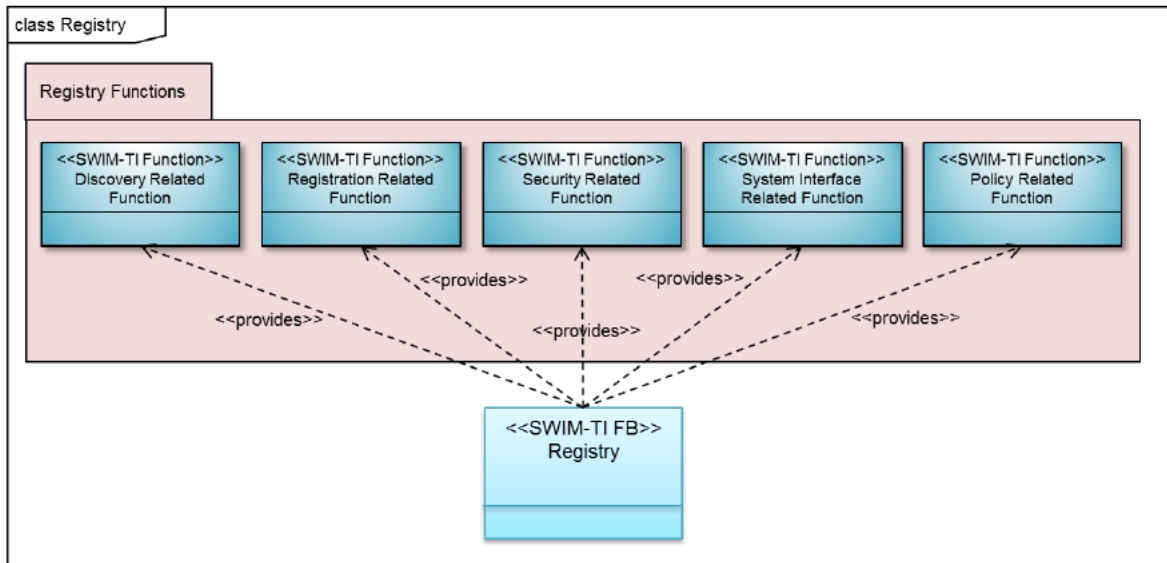


Figure 9 – Registry FB

### 2.2.1.1.1 Policy Related Function

As part of the policy management, a specific case is applicable to security policy for which an audit function is required.

Policy Related Function is a particular case of the policy management. It is the common function in charge of defining and distributing the common security policies applicable to all the SWIM stakeholders as part of the general policy management. It covers **policy publication and management** (including creation, maintenance, change and deletion), **policy deployment and policy auditing**. The security rules described in the policy have to be distributed to all the SWIM policy enforcement points. Depending on the nature of the Policy Enforcement Point (PEP) (appliance of a dedicated hardware device, portal or application server) the policy description may require a transformation to properly configure the PEP. Finally Changes to security policies must be tightly controlled, access to them must be traced, and audit trails must be supplied so that the security procedures can be adequately monitored.

This common function only makes sense if it is possible and desirable to manage a common set of security policies at regional level. It does not mean that they are the only applicable policies. Each SWIM stakeholder can manage its own security rules that are combined in the policy enforcement point with the common ones.

This function is linked with policy enforcement point described in 2.2.1.2.7 and 2.2.1.3.7.

Policy Related Function is designed to meet the SWIM ConOps requirements (see [11]) listed in the table below.

Identifier	Statement
REQ-08.01.01-CONOPS-ASDE-0020	It is assumed that the policies to be applied in the service provision and consumption (e.g. policies on security, on authentication, etc) will be defined by the “SWIM Collaboration Authority”. Once endorsed, the implementation of such policies will be done on a voluntary and collaborative basis by the impacted stakeholders.

Table 4 – SWIM ConOps requirements on Policy Related Function

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

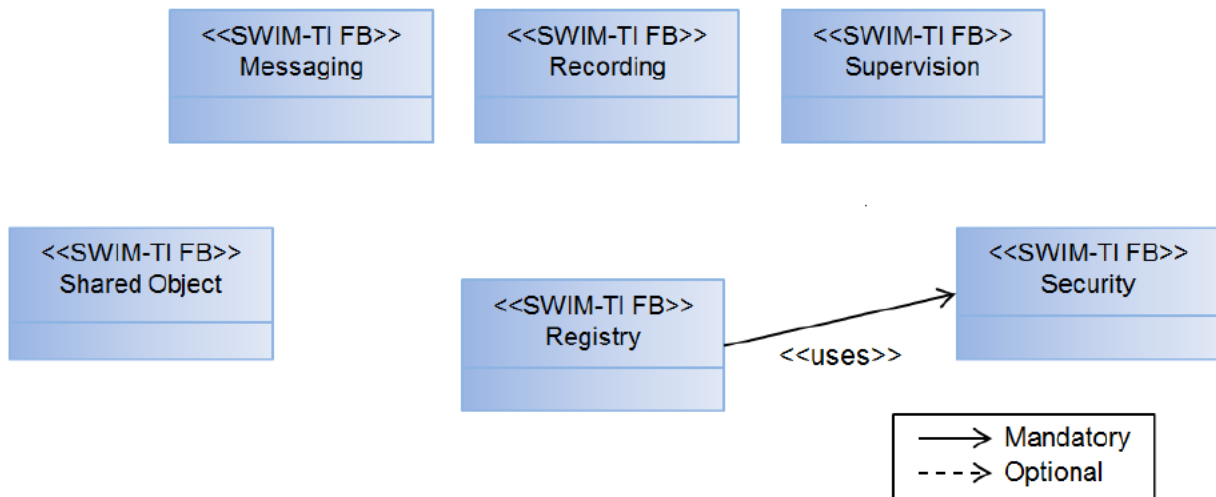
### 2.2.1.1.2 Registry Functional Block Dependencies

The table below summarizes the dependencies Used by Registry FB:

FB	Dependency	Optional / Mandatory	Dependency Description
Registry (REG)	Security (SEC)	Mandatory	Use of Security functions in order to authorize and authenticate users of the Registry.

**Table 5 – Registry Functional Block Dependencies**

The figure below summarizes the dependencies Used by Registry FB:

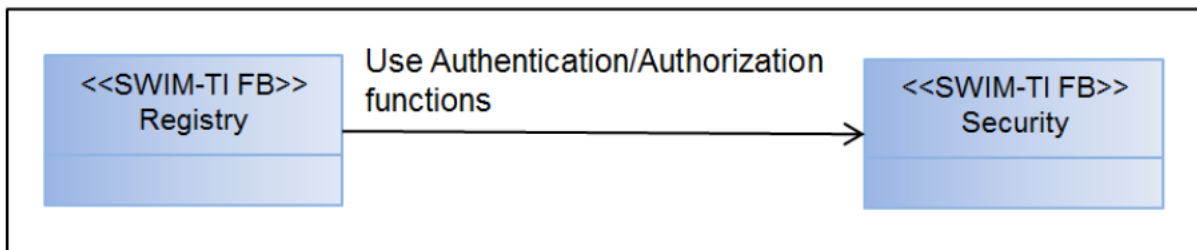


**Figure 10 – Registry Functional Block Dependencies**

#### 2.2.1.1.2.1 Use Dependencies

##### 1. Security (SEC FB) Dependency

For the Registry FB, the optional Security Functional Block (SEC FB) dependency is a ‘use’ one to authorize and authenticate the users of the Registry.



**Figure 11 – Registry FB use of SEC FB Dependencies**

## 2.2.1.2 Messaging Functional Block

A Message can be described as a set of data containing a concrete kind of information that is intended to be interchanged between two or more actors. The action of interchanging that set of data represents what is known as Messaging and is usually (but not only) applied in the ambit of distributed systems<sup>14</sup>.

SWIM-TI Messaging aims at providing interoperability between distributed systems with varying degrees of decoupling and including features for effective and reliable communication.

SWIM-TI Messaging aims at providing the following features:

- Support for a variety of Messaging Technologies & Protocols.<sup>15</sup>
- Support for a variety of routing mechanisms<sup>16</sup>. The routing will determine where a message will be delivered as well as define through which communication paths a message will reach its intended destination or destinations.
- Support for a variety of distribution mechanisms.
- Support for filtering mechanisms. The filtering allows the elimination of messages based on filtering criteria.
- Support of a variety of Quality of Services (QoS), including reliable delivery, best effort delivery, durable subscriptions, transaction management and message handling specification according to priority and response time requirements.
- Support for Protocol Bridge. The Protocol Bridge performs the transformation from source messaging protocol and underlying stack into an output messaging protocol and underlying stack
- Support for Data Management. The SWIM-TI Data Management is in charge of operations on the data that is transported by the SWIM-TI Messaging.
- Support for Data Validation. Data Validation function is able to check data payload against the expected format prior to the service execution and allow or deny a service access or to check response data payload prior to further usage of the data in the same way described above for the provider but does not cover semantic checking that requires domain knowledge

The SWIM-TI Messaging Functional Block is broken down following the diagram below.

---

<sup>14</sup> A distributed system consists of one or more sub-systems, no monolithic, that interact and communicate through a network to achieve the business of the system

<sup>15</sup> Such as SOAP, RTPS, XML, HTTP(s), AMQP, etc...

<sup>16</sup> According to the OSI Model and such as TCP, UDP, etc...

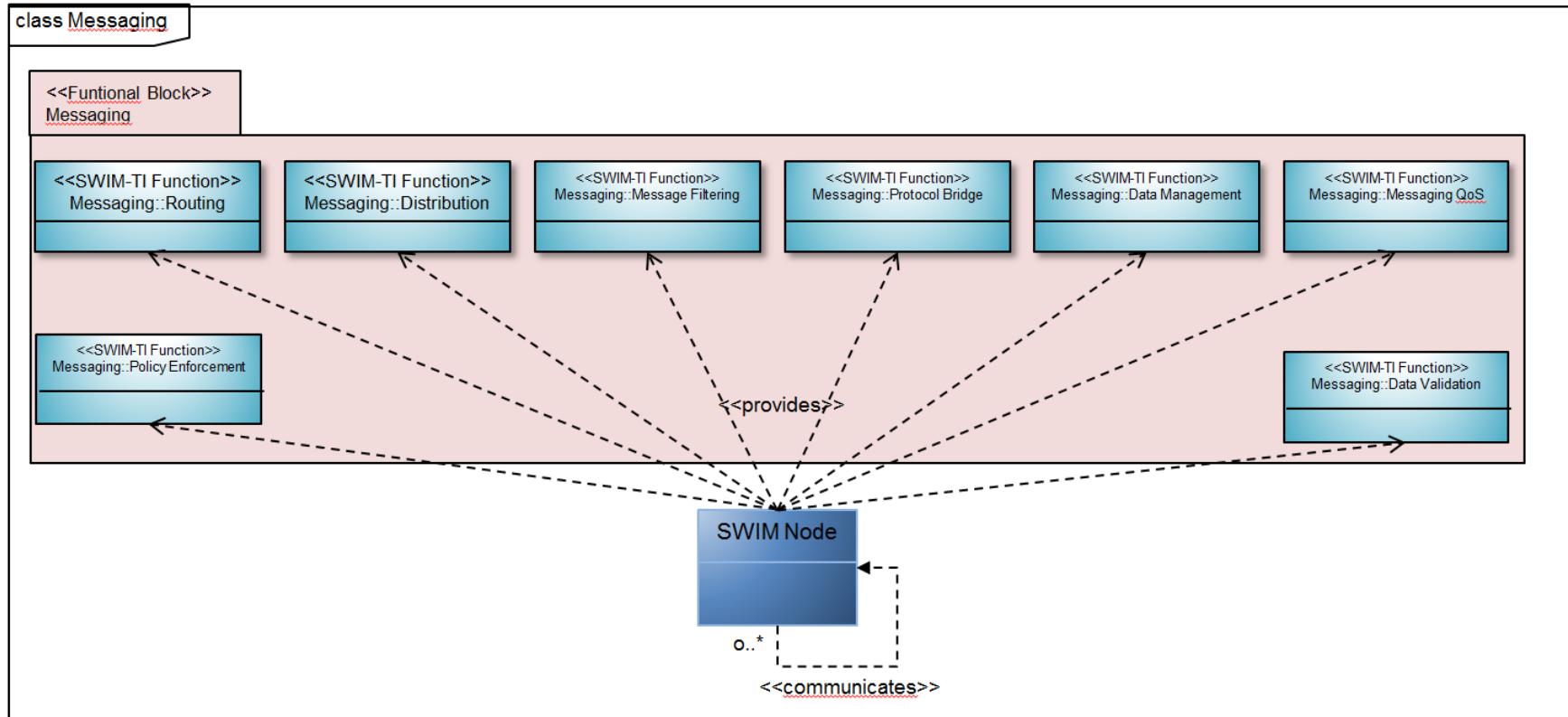


Figure 12 – Messaging Functional Block breakdown

### 2.2.1.2.1 Routing

The SWIM-TI Routing is in charge of the determination of where a message will be delivered as well as defining through which communication paths a message will reach its intended destination or destinations.

#### 2.2.1.2.1.1 Ontology<sup>17</sup>

##### 2.2.1.2.1.1.1 Direct or routing

From an ontology perspective on routing, in a communication between participants at any level in the layered model, there is choice between either direct interaction or interaction through routing.

In many use cases, routing provided through IP may be sufficient: then there is no more need for routing in any of the layers above IP.

In other use cases, the use of routing in one or more layers above IP may be necessary. Hence, multiple distinct routing mechanisms can be active in the same stack in distinct layers: this can introduce significant additional complexity and/or constraints.

Description of routing (including the necessity, the trade-offs) in these higher layers in the technical view is almost absent. There are no explicit statements either on the case of absence of routing. Hence the reader is not provided with arguments for choosing one or the other approach.

##### 2.2.1.2.1.2 Policy

In order to allow for interoperability between participants, collaboration between multiple instantiations of the routing function in a network may be needed.

The configuration of the routing function instantiation at one SWIM Node, can be optimal from the point of view and interest of that SWIM Node. Each of the other SWIM Nodes that are also involved in the routing may also have a specific configuration for its routing function instantiation that is optimal from its specific point of view and interest.

However from a perspective of the interest of the entire distribution system such per SWIM Node choices may be highly inefficient and/or ineffective.

Hence, the need for various levels of coordination of configuration of the instantiations of the routing functions for the interest of the entire distributed system.

A Messaging/Routing policy expresses such coordination of configuration. Such policy is not only a matter of governance, but may also imply mandated technology, such as a Registry.

##### 2.2.1.2.1.3 Criteria

The decision on where a message will be delivered by the SWIM-TI Routing and through which communication paths is based on criteria.

<sup>17</sup> There exists no standardized ontology for the entire domain of messaging. In the context of some open standard, an ontology is defined but with a local scope only. Besides that collection of patterns have been defined such as <http://www.eaipatterns.com/MessageRoutingIntro.html> and <https://msdn.microsoft.com/en-us/library/ee236697%28v=bts.10%29.aspx>. However, such patterns represent only a discrete subset and are not targeted at the definition of an ontology.

The literature and products do not provide or use a standardized classification for the criteria. In some cases, the semantics of identical terms are different too. Some products offer functionality classified under Routing, that is not genuinely available.

There are 3 main types of criteria that can be used to determine where a message will be delivered and through which communication path:

- Content-based (also called payload based. Note the contrast with an interpretation of Content-based that includes payload plus what is covered by subject-based below.)
- Subject-based (also called header-value or meta-data)
- Context-based

Complex criteria can be composed by combining all types.

There is resemblance between the SWIM-TI Routing and the SWIM-TI Message Filtering but they are not identical:

- The SWIM-TI Message Filtering decides whether a message is dropped or not.
- The SWIM-TI Message Filtering does not decide where a message is sent nor which path is used.

#### 2.2.1.2.1.4 Exception conditions

The SWIM-TI Routing handles exception conditions. Exception conditions are diverse such as:

- Depending on the nature of the communication path, some types of exceptions such as timeout can occur or not.
- Depending on the nature of the destinations, the Routing function may not be able to resolve the address of the destinations
- Expiration of the validity of the message

The handling of the exception conditions by the SWIM-TI Routing can consist of actions such as:

- Retry
- Use of a failover mechanism
- Reporting of non-delivery to a dead-letter mechanism

#### 2.2.1.2.1.5 Content-based criteria: challenges

Content-based criteria imply examination of the payload (content) of the message. Several levels of depth of encoding and understanding of the payload can be distinguished.

The payload of a message is encoded in a format that is specific to messaging protocol. This type of encoding falls in the area of responsibility of the SWIM-TI itself. It is performed by the SWIM-TI and it is invisible for the ATM specific service layer.

The payload itself may be further encoded in one – i.e. at least the physical data exchange format -, or more formats and possibly in multiple layers.

It can be expected that all of the payload encodings use a standardized syntax.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

A standardized syntax allows the SWIM-TI to analyse the content on a syntactical base and to take decisions on the base of more or less complex criteria.

The capacity by the SWIM-TI to handle encoding in the area of responsibility of the SWIM-TI itself is assumed to be trivial.

That is however not enough for content-based filtering as the SWIM-TI also needs to understand the relevant standardized syntax of each encoding used in the payload itself.

This creates a strong dependency between SWIM-TI and ATM specific service. The SWIM-TI will need to be kept synchronized with the evolution of the encoding(s) inside the payload itself (see Appendix J instructions to carry out Interface Evolution properly).

Ideally, none of the SWIM-TI functions should need to understand the payload of the message.

Nevertheless, whenever understanding of the payload of the message is needed in the SWIM-TI, it is proposed that this capability is delegated to and assumed by the SWIM-TI Data Management in the SWIM-TI exclusively. Hence to allow the SWIM-TI Routing to take decisions based on the content without the SWIM-TI Routing having to understand the payload format, following approach could be adopted.

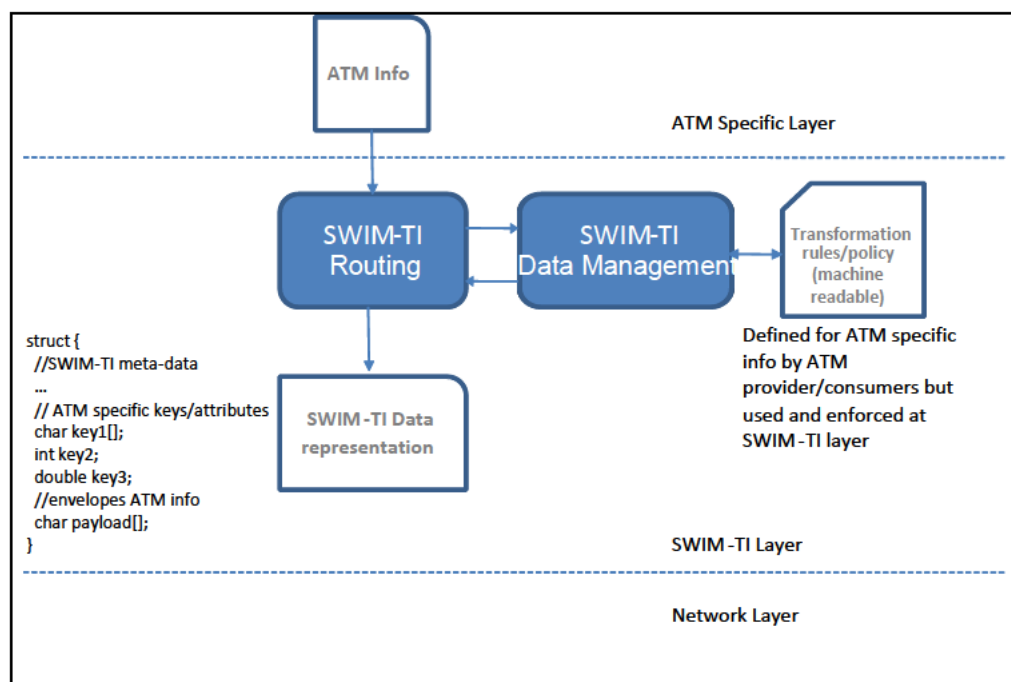


Figure 13 – Advanced Routing Mechanisms

As depicted in the figure, it is possible to enrich data at SWIM-TI layer based on the payload and without modifying the payload. We could consider that, for a given ATM information exchange, the message structure at SWIM-TI layer is the one depicted above:

- It includes a meta-data section aiming at including SWIM-TI specific data used only at SWIM-TI layer.
- It includes a payload which represents the whole ATM information encoded in a given format without altering the ATM content.



- It includes ATM specific keys/attributes that are provided by the SWIM-TI Data Management enforcing machine readable transformation rules/policy that automatically extract such information from the payload. Those keys/attributes are used as routing criteria.
- In this scheme the SWIM-TI Routing technically uses Subject-based criteria on subject-like data that has been created from the payload.

SWIM-TI only enforces the transformation rules/policy. The responsibility for the definition of the transformation rules/policy remains with ATM information provider/consumer.

#### 2.2.1.2.1.6 Subject-based criteria

Besides the generic capacity to examine the subject information of any message to feed the decision taking on the destination and which through which paths, subject-based routing can use number of criteria in a specific way that is ubiquitous and therefore recognized as a pattern.

Two of these patterns are supported in the SWIM-TI Routing:

- Routing Slip

In the Routing Slip pattern, the message is delivered to destinations in a predefined order. The order as well as the destinations can be established statically or dynamically. The Routing Slip pattern avoids duplicated processing of a message.

- Recipient List

In the Recipient List pattern, the message is cloned and delivered to each of the destinations.

#### 2.2.1.2.1.7 Context-based criteria

A number of criteria are neither content-based nor subject-based but related to the context wherein a message is sent.

The context can be represented through criteria such as following. Note that is not an exhaustive list:

- Size of the message
- Conditions on the communication paths (e.g. load, availability)
- The number of attempts to send a message
- Failure and timeout
- The time of day

#### 2.2.1.2.1.8 Delivery

To realize delivery, the routing function relies on communication paths and addressing. Different types of delivery method can be distinguished based on communication path and addressing patterns.

- **Unicast**

Single communication from a single sender and a single receiver. The receiver is identified by a unique destination address.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- **Anycast**

Single communication from a single sender to a single receiver, which is selected from a group of receivers. The selected single receiver is the topologically nearest to the sender. The receivers in the group are identified by a shared destination address.

- **Multicast**

Single communication from a single sender from possibly many senders to all receivers in a group of receivers. The receivers in the group are identified by a shared destination address.

- **Broadcast**

Single communication from a single sender to all receivers in a network.

### 2.2.1.2.2 Distribution

Distribution is the core function of the SWIM-TI Messaging. The Distribution function is realized via the support to specific Message Exchange Patterns (MEP).

MEPs are characterised through 4 groups of attributes<sup>18</sup>:

- **Conversation direction**

A conversation is a series of related messages, such as those depicted in a BPMN Conversation model (It is supported by IERs for information exchanges). The Conversation direction describes the sequencing and direction of the flow of these messages between the interacting parties<sup>19</sup>.

- **Cardinality**

Cardinality describes the number of participants in the exchange of messages.

- **Decoupling**

Decoupling describes the degree of loose coupling between the participants. Decoupling is subdivided into 3 dimensions:

- **Time:** Time decoupling means that the interacting parties do not have to be actively participating at the same time.
- **Space:** Space decoupling means that the interacting parties do not have to know each other.
- **Synchronization:** Synchronization decoupling means that the interacting parties are not blocked and can do other work

- **Push/Pull<sup>20</sup>**

Push/Pull indicates whether a subscriber will receive the data at the initiative of the publisher (Push) or whether the subscriber needs to fetch the data (Pull).

---

<sup>18</sup> The structuring is fully aligned with the content of "The Many Faces of Publish/Subscribe

<sup>19</sup> Refer to the BPMN and transaction scenario modelisations available in the SWIM-TI specification to obtain information related to interacting participants

<sup>20</sup> This structure is fully aligned with [www.eaipatterns.com](http://www.eaipatterns.com)

Push and Pull create a significant difference within the conversation, as Pull introduces a Synchronous Request/Reply in the conversation.

The table below documents for each of the patterns some aspects of these attributes.



MEP	Direction conversation Consumer=C Provider=P Subscriber=S Publisher=Pu	Cardinality	Time Decoupling		Synchronization Decoupling		Space Decoupling	
			Consumer / Subscriber	Provider / Publisher	Consumer / Subscriber	Provider / Publisher	Consumer / Subscriber	Provider / Publisher
Synchronous Request/Reply (SRR-MEP)	2 way (C-> P > C)	1-1	No	No	No	Yes	No	No
Asynchronous Request/Reply (ARR-MEP)	2 way (C-> P > C)	1-1	No	No	Yes	Yes	No	No
Observer Push (OPUSH-MEP)	1 way (Pu -> S)	1-many	No	No	Yes	Yes	No	No
Observer Pull (OPULL-MEP)	1 way (Pu -> S) Synchronous R/R	1-many	No	No	Yes	Yes	No	No
Publish/Subscribe Push (PSPUSH-MEP)	1 way (Pu -> S)	many-many	Yes	Yes	Yes	Yes	Yes	Yes
Publish/Subscribe Pull (PSPULL-MEP)	1 way (Pu -> S) Synchronous R/R	many-many	Yes	Yes	Yes	Yes	Yes	Yes
Asynchronous Fire & Forget (AFF-MEP)	1 way (C -> P), 2 way (C-> P > C)	1-1	Yes	Yes	Yes	Yes	No	No
Fully Decoupled Request/Reply (FDRR-MEP)	2 way (C-> P > C)	1-1	Yes	Yes	Yes	Yes	Yes	Yes

Table 6 – MEP characterisation

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

## 2.2.1.2.2.1 Message Exchange Patterns (MEPs)

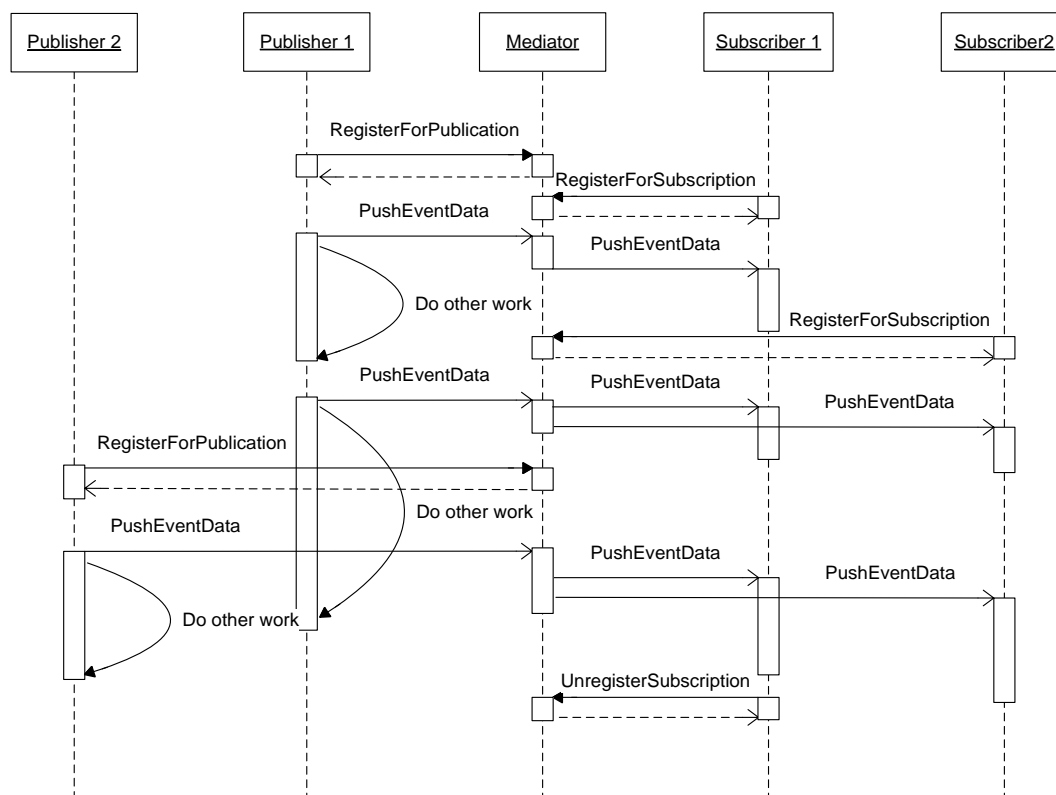
### 2.2.1.2.2.1.1 Publish/Subscribe style MEPs

#### 2.2.1.2.2.1.1.1 Characterization of Publish/Subscribe style

The Publish/Subscribe style of interaction is characterised by:

- Subscribers that have the ability to express their interest in an event, or a pattern of events, and are subsequently notified of any event, generated by a publisher, which matches their registered interest.
- Full decoupling in time, space and synchronization between publishers and subscribers.
- In case of Publish/Subscribe Push MEP:

A publisher sends event data in the messaging service. The messaging service sends the event data to all subscribers that have manifested their interest through a subscription. The publisher and subscribers do not have to know of each other. The messaging service maintains the subscriptions. Publisher and subscribers do not need to be simultaneously present.



Publish/Subscribe Push

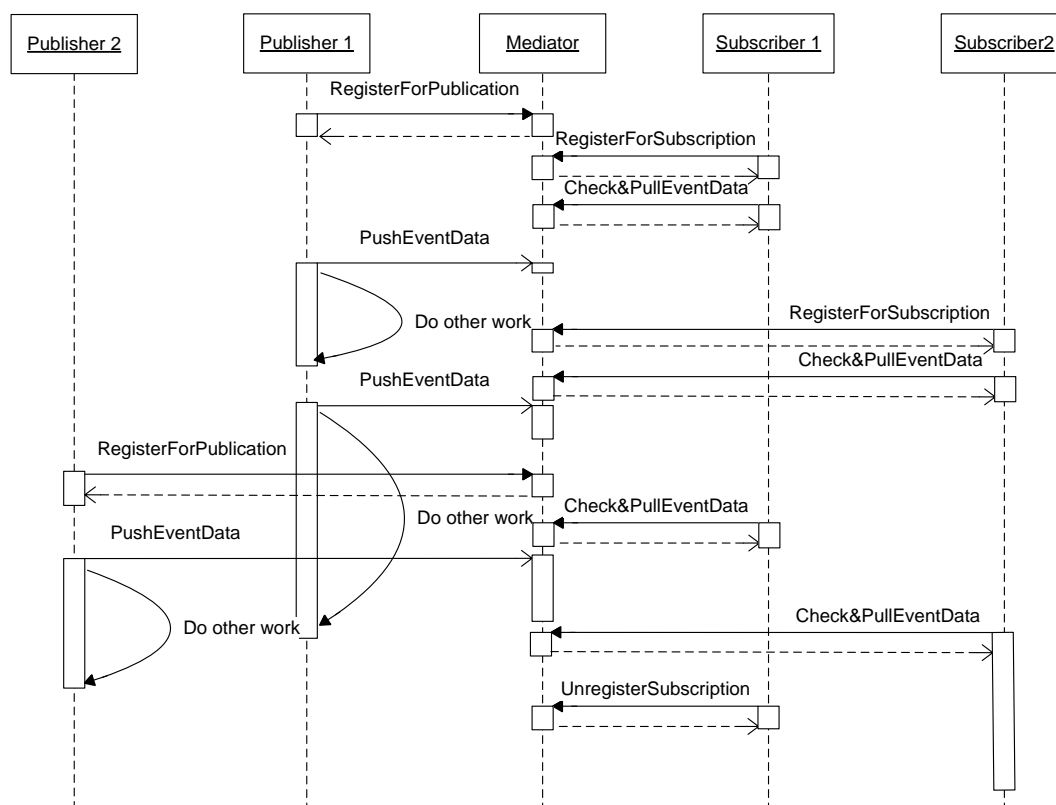
- In case of Publish/Subscribe Pull MEP:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

A publisher sends event data in the messaging service. The messaging service possibly sends a notification of the presence of the event (but not the event data itself) to all subscribers that have manifested their interest through a subscription. Notwithstanding the sending or not of such notification, a subscriber can periodically check for any new data/update. In any case the subscriber has to fetch the event data through an interaction that is equivalent to a Synchronous Request/Reply. The publisher and subscribers do not have to know of each other. The messaging service maintains the subscriptions. Publisher and subscribers do not need to be simultaneously present.



Publish/Subscribe Pull

### 2.2.1.2.2.1.1.2 Classification of interaction patterns

There exist several other interaction patterns that resemble the Publish/Subscribe style interaction pattern but that are clearly differentiated from the Publish/Subscribe style interaction pattern, through the 3 dimensions of decoupling: time, space and synchronization.

As an example the Observer style MEPs has been identified: it does neither have time decoupling nor space decoupling.

Note that [eaipatterns.com](http://eaipatterns.com) and related publications have a concept called publish-subscribe Channel. This concept is not identical to the Publish/Subscribe style MEPs described here. This concept could however be used as a building block of a model for an implementation of the Publish/Subscribe style MEPs described here.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

#### 2.2.1.2.2.1.1.3 Subscription schemes, overview

There exist different ways for a subscriber to specify the events of interest. These distinct ways are called subscription schemes.

The classification of the subscription schemes is typically performed according to 2 high-level types:

- subject-based.

Messages in subject-based Publish/Subscribe systems are sent to subscribers, based on the subject that describes the contents of the message.

Within the subject-based category several variants are included such as topic-based, group-based and logical channel based.

- content-based (also known as property-based).

Messages in content-based Publish/Subscribe systems are sent to subscribers based on the contents of the message itself.

Other subscription schemes:

- type-based,
- combinations of subject-based and content-based, sometimes called hybrid subscription scheme,
- channel-based.

#### 2.2.1.2.2.1.1.4 Topic-based Publish/Subscribe

In a topic-based Publish/Subscribe system, messages are published to named "topics" and the subscribers will receive all messages published to the topics to which they are subscribed, and all subscribers to a topic will receive the same messages. In this paradigm, a publisher is responsible for defining the classes of messages to which subscribers can subscribe.

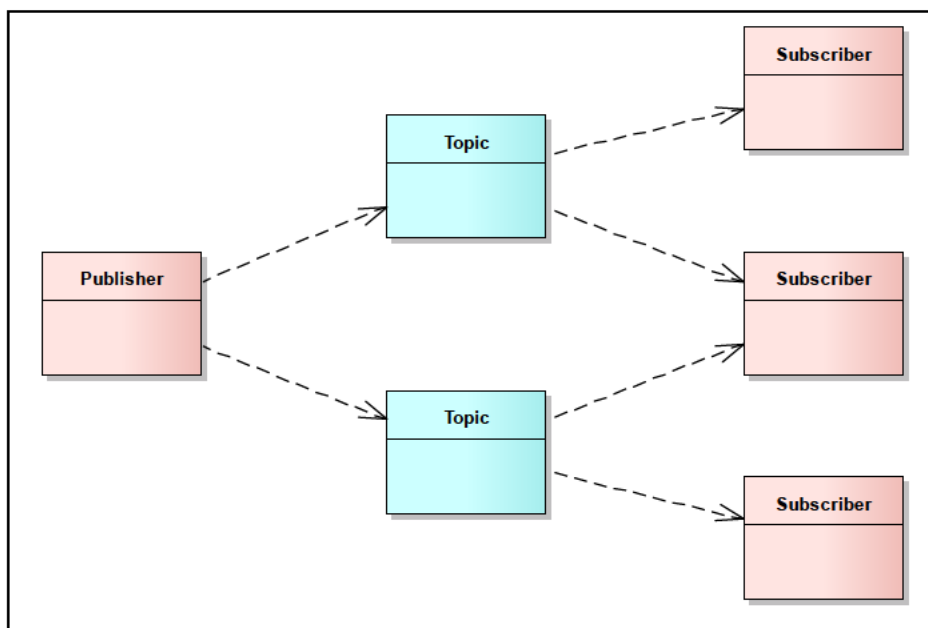


Figure 14 – Topic-based Publish-Subscribe

### 2.2.1.2.2.1.2 Request/Reply style MEPs

- **Synchronous Request/Reply MEP**

Message Exchange Pattern in which a requestor sends a message to the replier, who in turn processes the request and returns a response. The requestor waits for the response or time out before doing other work.

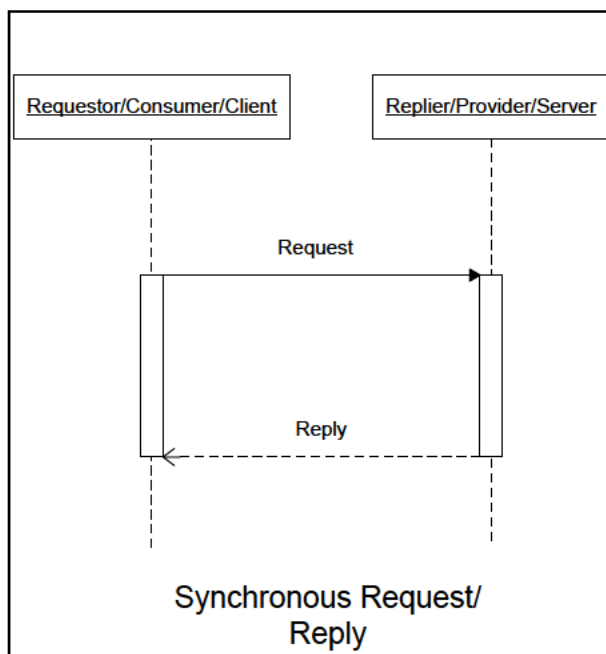


Figure 15 – Synchronous Request/Reply MEP sequence diagram

- **Asynchronous Request/Reply MEP**

A requestor sends a request message to a replier system which receives and processes the request, ultimately returning a message in response. It allows two applications to have a two-way conversation

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
 www.sesarju.eu



with one another. The requestor does not wait for the response and the response might be returned at some unknown later time.

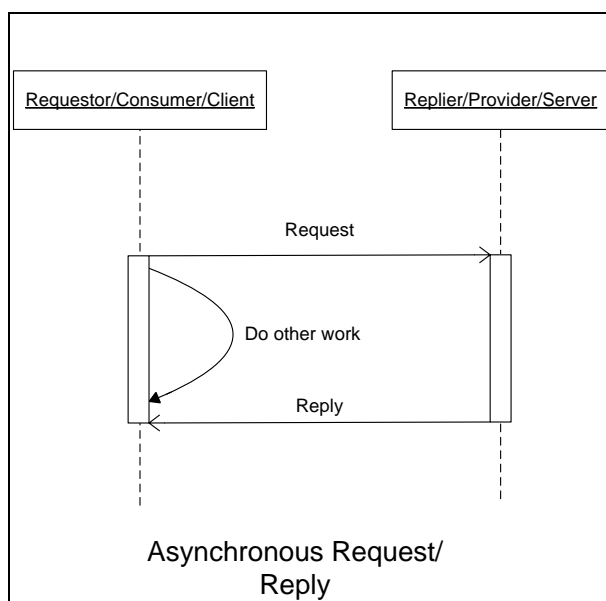


Figure 16 – Asynchronous Request/Reply MEP sequence diagram

#### 2.2.1.2.2.1.3 Observer style MEPs

The Observer style of interaction is characterised by:

- Subscribers that have the ability to express their interest in an event, or a pattern of events, and are subsequently notified of any event, generated by a publisher, which matches their registered interest.
- No decoupling in time and space between publishers and subscribers
- Decoupling in synchronization between publishers and subscribers
- In case of Observer Push MEP

A publisher sends event data to all subscribers that have manifested their interest through a subscription. The publisher knows and maintains the subscriptions. Publisher and subscriber need to be simultaneously present.

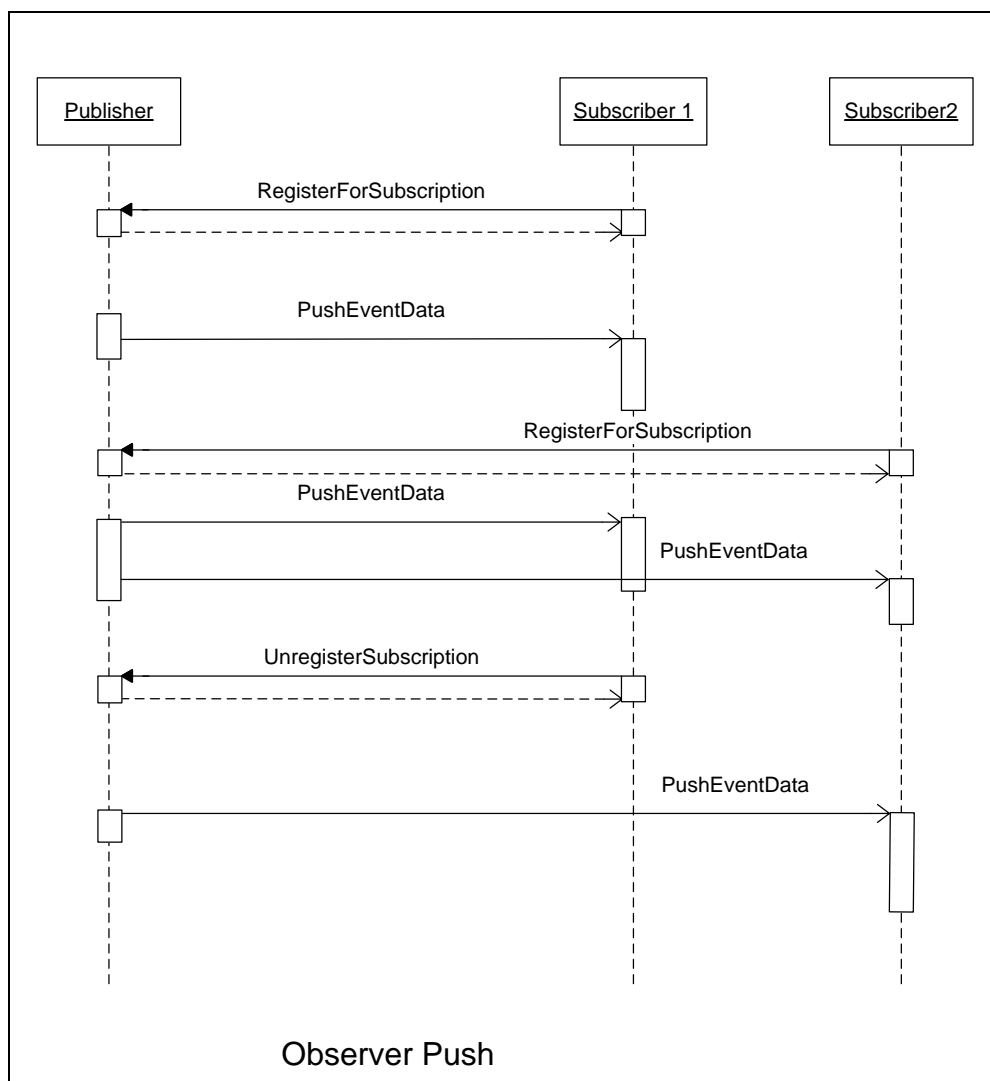


Figure 17 – Observer Push MEP sequence diagram

- In case of Observer Pull MEP

A publisher possibly sends a notification of the presence of an event (but not the event data itself) to all subscribers that have manifested their interest through a subscription. Notwithstanding the sending or not of such notification, a subscriber can periodically check for any new data/update. In any case the subscriber has to fetch the event data through the equivalent of a Synchronous Request/ReplyMEP. The publisher knows and maintains the subscriptions. Publisher and subscribers need to be simultaneously present.

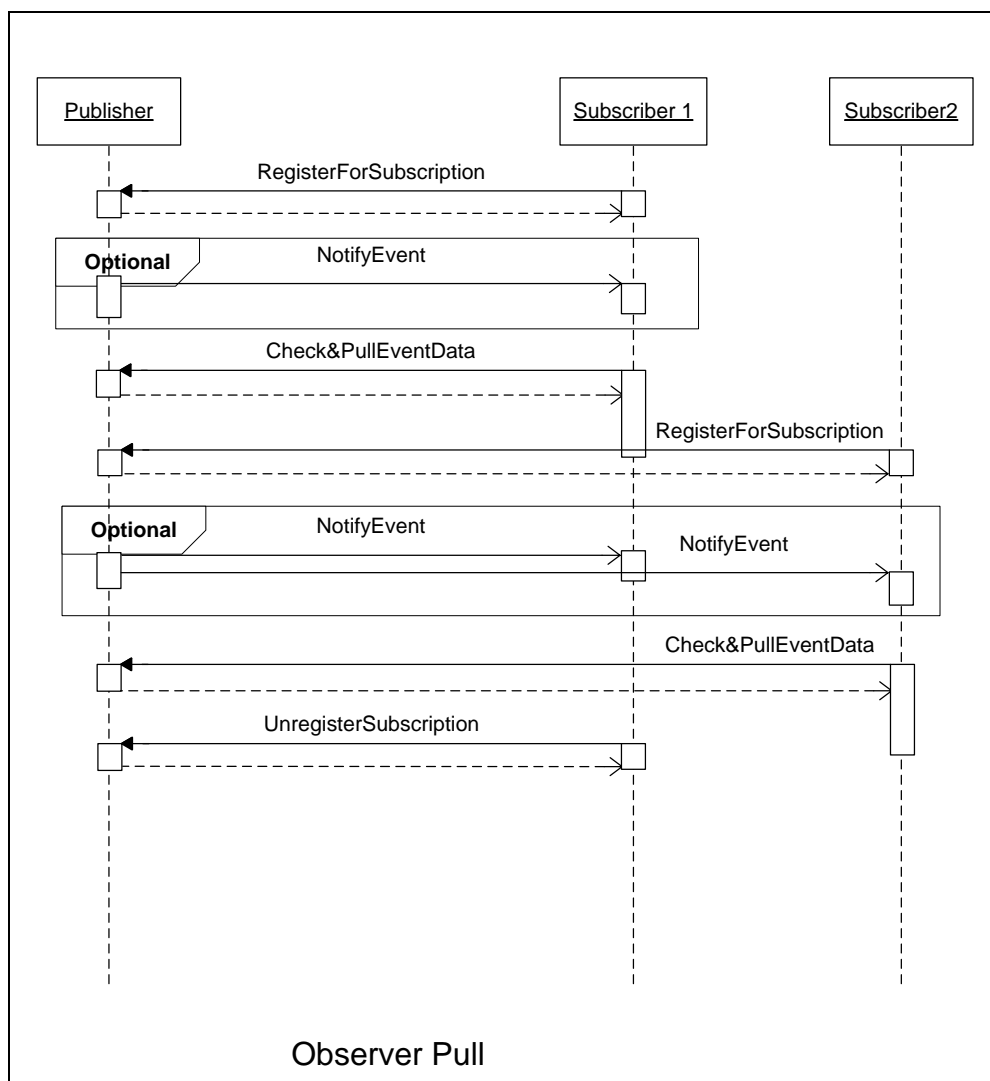


Figure 18 – Observer Pull MEP sequence diagram

#### 2.2.1.2.2.1.4 Asynchronous Fire & Forget MEP

A requestor sends a request message in the messaging service targeted at a provider system which at some undetermined time receives and processes the request. The requestor is not informed on the outcome of the request. Requestor and provider system do not need to be simultaneously present.

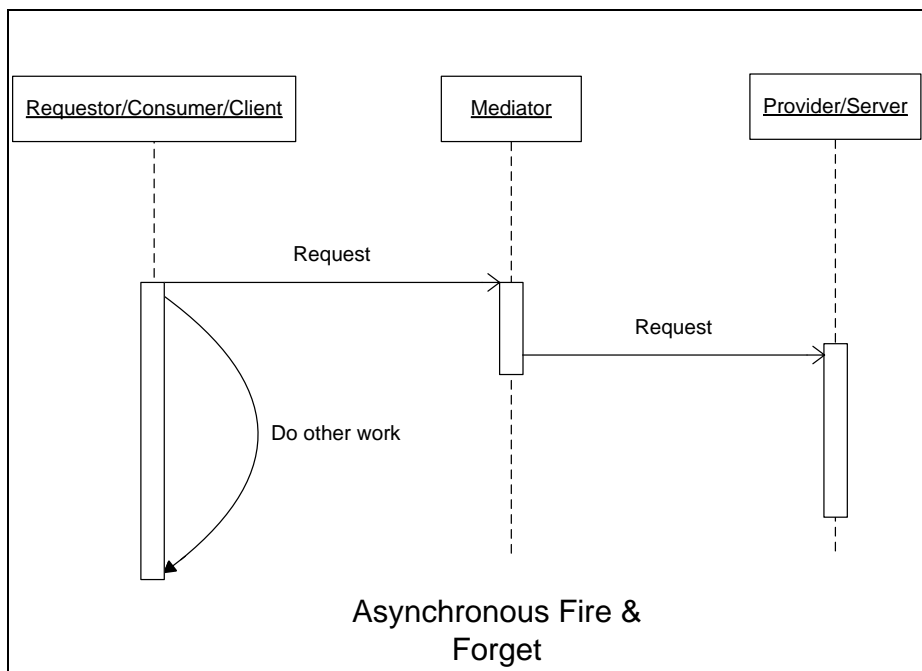


Figure 19 – Asynchronous Fire & Forget MEP sequence diagram

2.2.1.2.2.1.5 Fully decoupled Request/Reply MEP

A requestor sends a request message in the messaging service. The identity of the provider of the service is unknown by the requestor. The messaging service attempts to send the request message to a provider. When the request reaches the provider, the provider receives the message at some undetermined time and processes the request. The provider sends a reply to the messaging system which forwards it to the requestor. Both requestor and provider do not know each other nor do they know how many publishers and subscribers there are. The requestor and provider do not have to be present at the same time. The requestor and provider are not blocked waiting on each other.

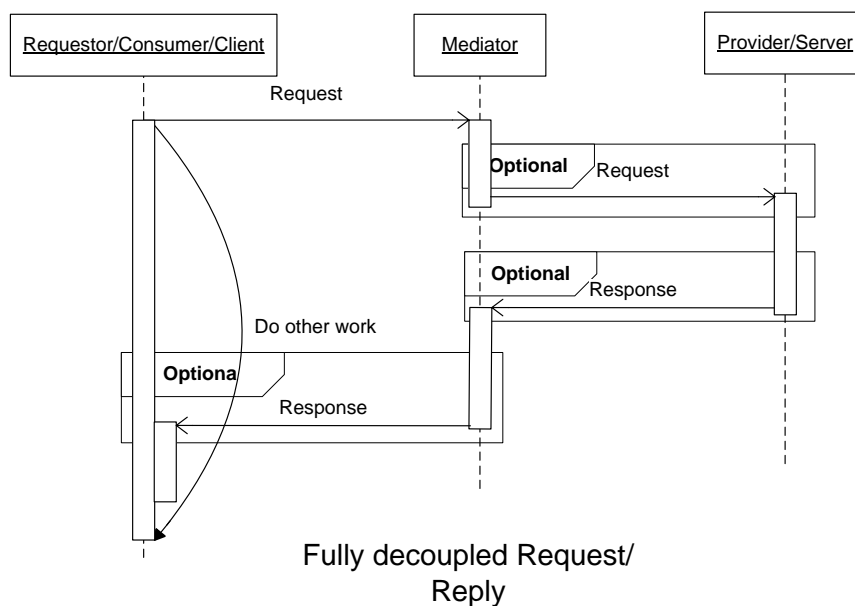


Figure 20 – Fully decoupled Request/Reply MEP sequence diagram

## 2.2.1.2.2.2 Roles

The MEPs identify distinct participants in the communication with distinct roles.

A classification of the roles is relevant because a role provides a unit of responsibility and activity that can have its own:

- security
- deployment
- operations
- governance

The classification of roles currently consists of:

- Service provider  
A generic role for providing an ATM specific service that can be consumed.
- Service consumer  
A generic role for consuming an ATM specific service that is provided.
- Publisher  
A publisher produces information that is potentially of interest for a Publication consumer.
- Subscriber  
A Subscriber subscribes interest for receiving information by a Publication consumer, negotiates the modalities for delivery of this information and manages the lifecycle of the subscription.
- Publication consumer  
A Publication consumer receives information for which the Subscriber has subscribed.
- Publication mediator  
A Publication mediator receives the information produced by a Publisher and forwards that information to all Publication consumers for which the subscriptions match the Publication characteristics.
- Subscription handler  
A Subscription handler interacts with the Subscriber and maintains the subscriptions.
- Registration handler  
A Registration handler interacts with the Publisher and maintains the registrations.
- Fire & Forget Mediator  
A Mediator provides time decoupling between client and server in a Fire & Forget MEP.
- Request Reply Mediator  
A Mediator provides time and synchronization decoupling between consumer and provider in a Fully decoupled Request/Reply MEP.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

77 of 284

### 2.2.1.2.3 Message Filtering

The SWIM-TI Message Filtering is in charge of the elimination of undesired messages based on criteria.

#### 2.2.1.2.3.1 Criteria

The 3 main types of criteria are identical to the criteria used in the SWIM-TI Routing at §2.2.1.2.1.3.

There is resemblance between the SWIM-TI Routing and the SWIM-TI Message Filtering but they are not identical:

- The SWIM-TI Message Filtering decides whether a message is dropped or not.
- The SWIM-TI Message Filtering does not decide where a message is sent nor which path is used.

#### 2.2.1.2.3.2 Content-based criteria: challenges

The challenges for SWIM-TI Message Filtering are identical to the ones for SWIM-TI Routing at §2.2.1.2.1.5.

#### 2.2.1.2.3.3 Subject-based criteria

Subject-based criteria reflect the generic capacity to examine the subject information of any message to feed the decision taking on whether the message is dropped or not.

#### 2.2.1.2.3.4 Context-based criteria

A number of criteria are neither content-based nor subject-based but related to the context wherein a message is sent.

The context can be represented through criteria such as following. Note that is not an exhaustive list:

- Size of the message,
- Conditions on the communication paths (e.g. load, availability),
- The time of day.

#### 2.2.1.2.4 Protocol Bridge

The SWIM-TI Protocol Bridge performs the transformation from source messaging protocol and underlying stack into an output messaging protocol and underlying stack.

##### 2.2.1.2.4.1 Problem statement

There are various messaging protocols in use each with their own set of features. Some features are overlapping with features equally available in other messaging protocols and some features are specific for a messaging protocol.

The messaging protocols themselves rely on underlying protocols (such as security protocols and transport protocols) to be able to provide their set of features. The combination of the messaging protocol and the underlying protocols makes such a combination particularly suitable for a set of Use Cases or not. Hence the existence of distinct messaging protocols.

Any further mention of messaging protocol in this chapter assumes the inclusion of these underlying protocols.

An interested consumer using one messaging protocol cannot send/receive a message to/from a service that uses another messaging protocol.

- The consumer and/or provider could include support for both messaging protocols: the consumer would then be able to directly communicate using the other messaging protocol or the service provider could make the service available via multiple messaging protocols.

There are reasons why such approach is not suitable such as for instance:

- the cost and complexity when many participants have to implement support for multiple messaging protocols,
- the unavailability of a technically suitable underlying infrastructure for end to end communication between the participants using the same messaging protocol.
- An alternative for above consists of a function that transforms between message protocols.

The function that performs the transformation between protocols is called Protocol Bridge.

The Protocol Bridge can abstract and shield the consumer and/or provider from the specificities of one or more other message protocols. Depending on the specificity of the feature set of each messaging protocol and the degree of overlap, the transformation will be more or less difficult and more or less complete.

In many cases the protocol transformation will introduce constraints.

##### 2.2.1.2.4.2 Protocol Bridge functionality

Below a more detailed view of the functionality that a Protocol Bridge offers and the areas of difficulty:

- Mapping of Message Exchange Patterns (MEPs).
  - The mapping of the MEPs between messaging protocols that are not natively supporting each other's MEPs
  - This includes keeping track of and coordinating the more complex MEPs that consist of more than one message and/or more than 2 participants.
- Mapping of meta-data (subject).

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- Some messaging protocols allow for meta-data that is not natively available in other messaging protocols.
- Mapping of security controls.
  - Mapping of credentials, identities and security controls between heterogeneous security infrastructures. This difficulty can be mitigated and/or avoided through federated identity.
  - Minimising the impact of loss of end-to-end security.
- Mapping of transactions.
  - Dealing with the presence and/or absence of support for transactions.
  - Maintaining and extending the notion of transaction into another messaging protocol.
- Mapping of NFR/QoS.
  - Maintaining the assumed QoS across messaging protocols.

The transformation of a messaging protocol does not always require a transformation of the payload data: in some cases, the transformation may impact only the underlying protocols. When a transformation of data is required then the Protocol Bridge function will use another function of the Messaging: the Data Management.



### 2.2.1.2.5 Data Management

The SWIM-TI Data Management is in charge of operations on the data that is transported by the SWIM-TI Messaging.

#### 2.2.1.2.5.1 Overview

The physical representation of data is performed through the encoding of the data in a particular format. Data can be encoded in different formats. In order to be able to access the data, it is necessary to understand the format wherein the data is represented.

An ATM specific application encodes its data in a format that is not necessarily suitable for the messaging protocol in the SWIM-TI:

- The format can be technically incompatible with the capabilities of the messaging protocol in the SWIM-TI. For example an ATM specific application format uses characters in the format that fall outside the set of characters that are allowed by the messaging protocol. In such case an ATM specific application will adapt the format of its data into a format that complies with the capabilities of the messaging protocol in the SWIM-TI for an outgoing message and apply the reverse process to access the data from an incoming message.
- In general, the SWIM-TI has no knowledge about the format used by an ATM specific application (or any other application that uses the SWIM-TI Messaging) to represent its data nor the number and the types of adaptations possibly performed by an ATM specific application to be able to provide the payload for the messaging protocol in the SWIM-TI in a format that is compliant with the capabilities of the messaging protocol in the SWIM-TI. In the interaction between the SWIM-TI and an ATM specific application, the SWIM-TI considers what comes from and goes to an ATM specific application as raw data only.
- In particular cases, the messaging protocol in the SWIM-TI explicitly constrains an ATM specific application to use a particular format the represent its data, such as XML, in which case the SWIM-TI has knowledge about the format of data. Only in such cases can the SWIM-TI perform transformation on the data.

#### 2.2.1.2.5.2 Data Format Transformation and Data Encapsulation functionality

Inside the SWIM-TI, there can be situations whereby the format used to represent the data as provided by an originating ATM specific application is no longer suitable. For instance:

- in case of a switch of messaging protocol, there can be technical incompatibility between the capabilities of the messaging protocols involved.
- in case the format used to represent the data, is not suitable for the context wherein a messaging protocol will be used

In such case the SWIM-TI will adapt the format of the payload of the messaging protocol in the SWIM-TI. Such adaptation can take two forms:

- Data Encapsulation

If the SWIM-TI does not understand the structure of the payload in the messaging protocol in the SWIM-TI, then it can only keep the original payload and put one or more layers around it in a manner that is compatible with the targeted messaging protocol in the SWIM-TI.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

When the message is delivered to an ATM specific application, the SWIM-TI will remove the layer(s) it has put around the original payload and provide the ATM specific application with the payload in its original format.

- Data Format Transformation

If the SWIM-TI does understand the structure of the payload of the messaging protocol in the SWIM-TI, then it can access the data and encode it into another format that is suitable without having to put one or more layers around the payload. Contrary to the encapsulation above whereby the original payload is maintained, transformation does not maintain the original payload but replaces it while maintaining the ATM semantics.

When the message needs to be delivered to an ATM specific application and if the receiving ATM specific application expects and understands the new format, the payload of the messaging protocol in the SWIM-TI is delivered as is to the ATM specific application. Otherwise the SWIM-TI will transform the data in the payload of the messaging protocol in the SWIM-TI into a format that is expected by the ATM specific application.

However both these processes can have an impact in the performance that needs to be taken into account.

### 2.2.1.2.5.3 Enrichment

When the SWIM-TI does understand the structure of the payload of the messaging protocol in the SWIM-TI, then it can access to the data and use this data to enrich the message.

In the figure below, the payload of the messaging protocol in the SWIM-TI is ATM data. As depicted in the figure, each message of the messaging protocol in the SWIM-TI, can be seen as composed by three main parts:

- optional SWIM-TI Metadata, which includes SWIM-TI specific information used to properly distribute and route the ATM data at SWIM-TI layer,
- optional payload specific keys/attributes, for instance ATM specific keys/attributes, which is information extracted from the payload and that can be used for different purposes such as filtering (when it is demanded to the SWIM-TI),
- Payload, which is, for instance, the concrete ATM message being distributed through the SWIM-TI Messaging.

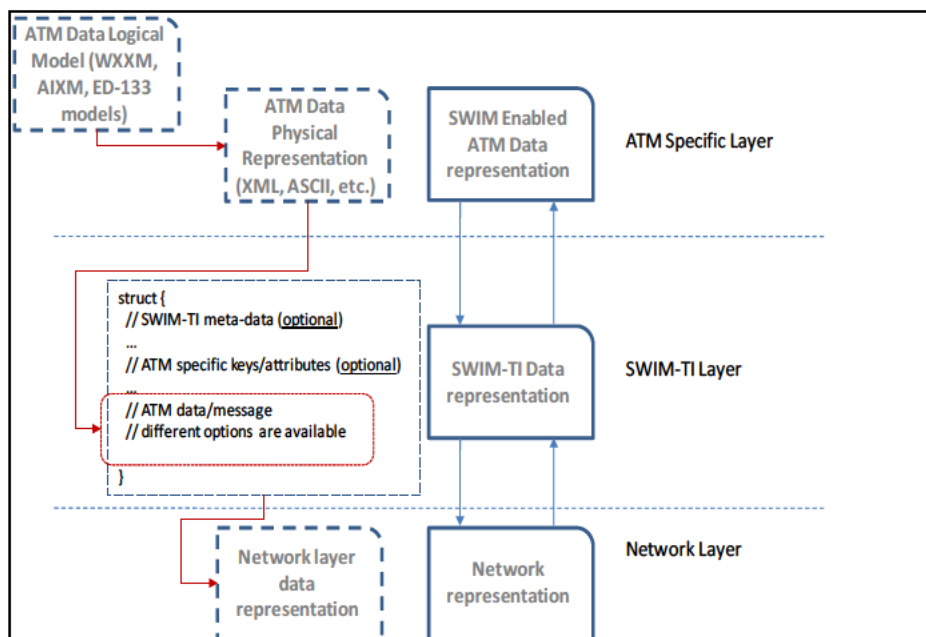


Figure 21 – Layering in SWIM data representation

It is possible to enrich data at SWIM-TI layer without modifying the payload: the payload specific keys/attributes can be filled by enforcing machine-readable transformation rules/policies that automatically extract information from the payload. Those keys/attributes can be used in criteria for SWIM-TI Routing and SWIM-TI Filtering.

#### 2.2.1.2.5.4 Data Management requirements

Related to the subject of data format, four distinct types of requirements can be provided:

- A requirement for the messaging protocol in the SWIM-TI to allow for a particular data format in its payload as provided directly by the ATM specific application.
- A requirement for the Data Format Transformation to provide a transformation between 2 formats.
- A requirement for the Data Encapsulation to be able to encapsulate and de-capsulate a payload of the messaging protocol in the SWIM-TI when bridging between messaging protocols in the SWIM-TI while keeping this encapsulation transparent for the ATM specific application.
- A requirement for the Enrichment to enrich the message with extracts from the payload.

### 2.2.1.2.6 Messaging Quality of Service (QoS)

Quality of Service aims at describing how different messaging aspects behave or are configured.

- The QoS characteristics can include:
  - Reliable delivery. The ability to offer guaranteed delivery of message semantics (e.g. at-most-once, at-least-once, exactly-once). This QoS characteristic is able to be defined in a SWIM profile.
  - Best effort delivery. The ability to offer delivery which the network does not provide a guarantee that data is delivered or that a user is given a guaranteed quality of service level or a certain priority. This service provides an unspecified variable bit rate and delivery time, depending on the current traffic load. This QoS characteristic is able to be defined in a SWIM profile.
  - Durable Subscriptions. The extent to which the middleware will retain messages for later joining systems. Typically, the size of the maintained history is an important aspect of this QoS.
  - Ordering. This relates to the ordering constraints on the recipient side. This constraint can be limited to messages originated from the same sender. In case absolute ordering is required (multiple senders), time-stamping and time synchronization issues are raised.
  - Periodicity. An agreement between a service provider and consumer of the rate of which an exchange of information is needed.
  - Lifespan. The validity life span of a given message. The middleware should prevent the routing and delivery of expired messages.
  - Priority. The priority permits certain messages to be prioritised above others. The priority can be identified in the message and is managed not by SWIM-TI but by the end-user.

### 2.2.1.2.7 Policy Enforcement

The messaging policies of interest up to now can be summarised:

- **Compression:** be able to describe and implement a compression algorithm of message payload. This should be valid for WS, WS/N and DDS.
- **WS/N QoS:** WS/N comes with a set of qualities of services. To ensure interoperability it may be necessary to specify those which are mandatory and those which are optional. In the latter case a policy description could say whether or not the QoS is applied.
- **Request retries:** be able to describe as a policy the number of retries the Messaging FB is allowed to perform after a request failure.
- **Data validation:** be able to prevent malformed messages to be propagated up to the application layer.

There are possibly several QoS Policies that can be applied. However to meet the needs for interoperability it's necessary to define the minimum set of policies that need to be supported.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

84 of 284

### 2.2.1.2.8 Data Validation (and Transformation)

The Data Validation<sup>21</sup> and transformation supports checking for conformance to message/data type descriptions as defined by SWP8.1, SWP8.3 and P14.01.04. The conformance conditions are expressed in form of well-defined policy assertions assigned to the SWIM service definition.

The Data Validation is able to inspect data payload prior to the service execution and allow or deny a service access. The decision is made through the application of policy rules on the available data payload. With other words, the conformance check might also be related to the structure of messages exchanged among the ATM systems, for its syntax and semantic.

The example of data validation usage we consider here is a service, with an interface, which requests use of pub/sub message exchange pattern and additionally defines the rules like that service clients can publish their notification on certain topics only if the notification payload fulfils predefined data type criteria. The data types might be defined using some of data modelling standards, such as XML Schema or IDL. Several concrete message types which are likely to be used by SWIM enabled ATM services are OGC WFS query, AIXM5.1 document, or digital NOTAM.

### 2.2.1.2.9 Messaging Functional Block Dependencies

The table below summarizes the dependencies Used by MSG FB:

FB	Dependency	Optional / Mandatory	Dependency Description
Messaging (MSG)	Security (SEC)	Mandatory	Use of Security functions in order to: - authorize and authenticate message producer and consumer ensure data confidentiality, integrity and authenticity
	Registry (REG)	Optional	Use of Registry functions in order to: Messaging related policies enforcement relies on the existence of a policy management.
	Recording (REC)	Optional	Use of Recording to keep track of the selected messages.

**Table 7 – Messaging Functional Block Dependencies**

The figure below summarizes the dependencies Used/Provided by MSG FB:

<sup>21</sup> Data Validation as part of SWIM-TI since it performs syntactical validations/transformations rather than semantically, which are expected to take place at ATM Application Level.

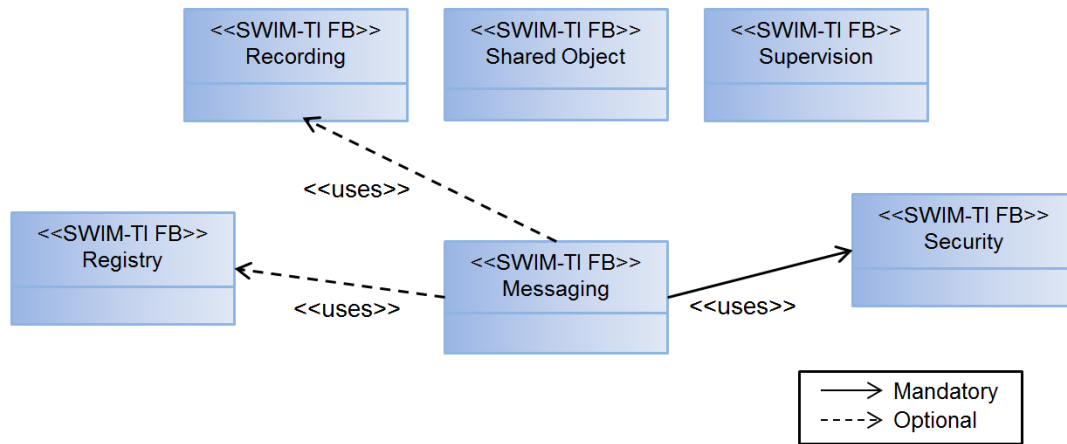


Figure 22 – Messaging Functional Block Dependencies

### 2.2.1.2.9.1 Use Dependencies

#### 1. Security (SEC FB) Dependency

For Message FB, the mandatory Security Functional Block (SEC FB) dependency is a 'use' one to authorize and authenticate the message producers and consumers, ensure data confidentiality, integrity and authenticity.

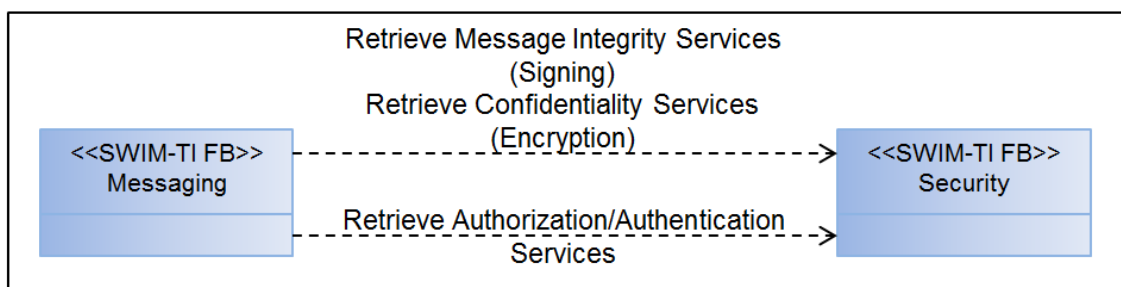


Figure 23 – MSG FB use of SEC FB Dependencies

#### Encryption:

A message to be sent is encrypted according to the specific Policy. This action is performed by the interaction between SWIM-TI MSG and SEC Functional Blocks.

WS Security specification for Encryption (such as that published by OASIS) is proposed to be applied to encrypt SOAP messages for confidentiality of messages exchanges in the R/R MEP: WS-Security (X.509) XML Signature & Encryption.

Note that, according to the policy basis approach, not all the information exchange will be encrypted, the specific policy shall state, apart from other info, if the encryption is required or not.

WS Security specification for Encryption (such as that published by OASIS) when applied to encrypt SOAP messages for confidentiality of messages exchanges in the R/R MEP: WS-Security (X.509) XML Signature & Encryption. This is done at transport level if the encryption of the entire message is needed or alternatively at message level using XML encryption in case of particular encryption policy needs (e.g. partial payload encryption).

#### Signing:

In general, a message payload being sent is signed using the data producer Digital Identity and according to the specific Policy. This action is performed by the interaction between SWIM-TI MSG and SEC Functional Blocks.

WS Security defines how to sign each SOAP message for integrity of the message. XML signature specifies an XML syntax and processing rules for creating and representing digital signatures (see W3C recommendations for XML signature).

Note that, according to the policy basis approach, not all the information exchange will be signed. The specific policy shall state, apart from other info, if it is required or not.

## 2. Registry FB (REG FB) Dependency

The Registry FB provides a set of functionalities aiming at managing policies lifecycle. To provide its service quality, respecting service level agreements etc, the Messaging FB relies on policy management and policy enforcement and therefore Registry shall be able to manage several kinds of policies for messaging, also storing and distributing messaging policies for enforcement purposes.

The three modes for policy deployment apply also for messaging:

- Polling mode: SWIM node periodically requests the Registry to see if a new policy is available and applicable to it.
- Push mode: Registry deploy some time in advance on all appropriate SWIM nodes the new policy with an effective date. When the date happens each SWIM node switches to the new policy.
- Subscription mode: SWIM node subscribes to a policy (e.g. by profile) by providing the SWIM profile(s) it is supporting. Each time a policy creation or change is deployed the SWIM node is notified of this event



Figure 24 – MSG FB use of REG FB Dependencies

## 3. Recording FB (REC FB) Dependency

Recording FB provides a set of functionalities aiming at allowing the recording of selected data and messages.

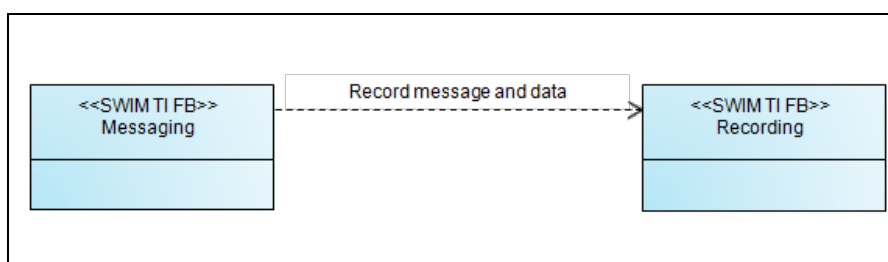


Figure 25 – MSG FB use of REC FB Dependencies

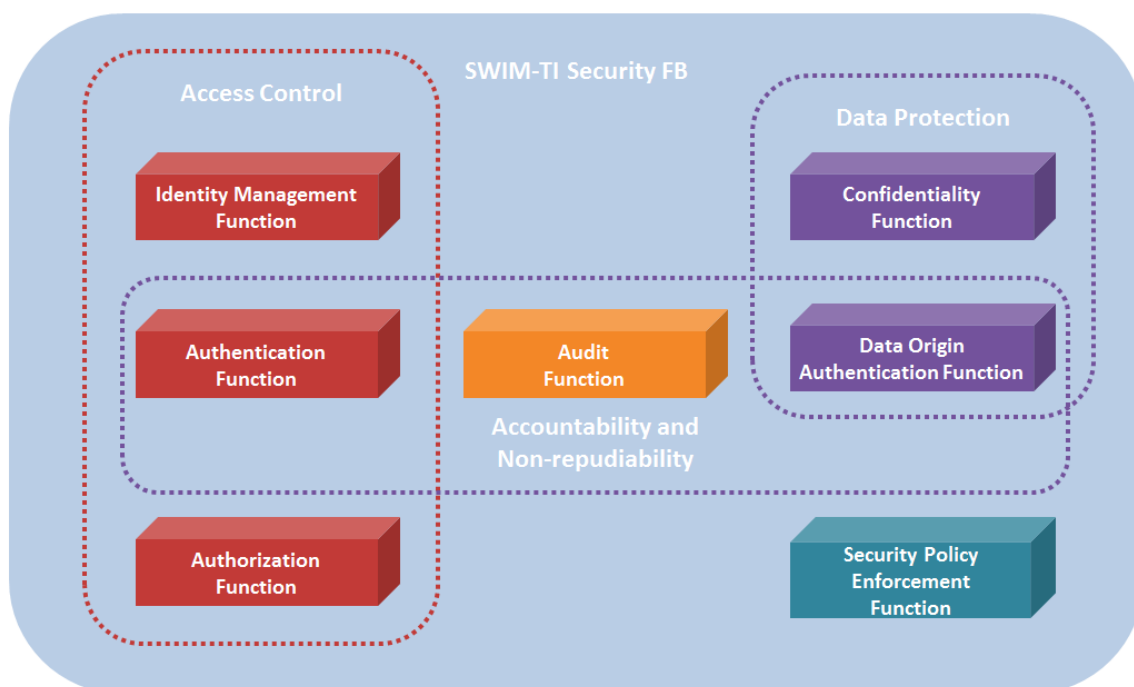


### 2.2.1.3 Security Functional Block

SWIM-TI Security FB provides technical functions enabling the **Access Control** (Authentication and Authorization), **Audit** and **Data Protection** in a federation of security domains.

This is used to control and to protect services and resources in the whole security chain (ATM specific and SWIM-TI).

**Access Control** relies on **Authentication** (authentication of a Digital Identity - refer to a process used to achieve sufficient confidence in the binding between the entity and the presented identity<sup>22</sup>), on **Authorization** (authorization of an authenticated Digital Identity to use a given resource - refer to the granting of rights and, based on these rights, the granting of access.<sup>23</sup>) and on **Identity Management** (provisioning, mapping and federation - refer to a set of functions and capabilities (e.g., administration, management and maintenance, discovery, communication exchanges, correlation and binding, policy enforcement, authentication and assertions) used for assurance of identity information (e.g., identifiers, credentials, attributes); assurance of the identity of an entity and supporting business and security applications.<sup>24</sup>). The figure below shows how these functions relate to each other.



**Figure 26 - SWIM-TI Security FB functional decomposition overview**

The SWIM-TI Security Functional Block can also be broken down as the diagram below.

<sup>22</sup> According to ITU-T IdM X.1252

<sup>23</sup> According to ITU-T IdM X.1252

<sup>24</sup> According to ITU-T IdM X.1252

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

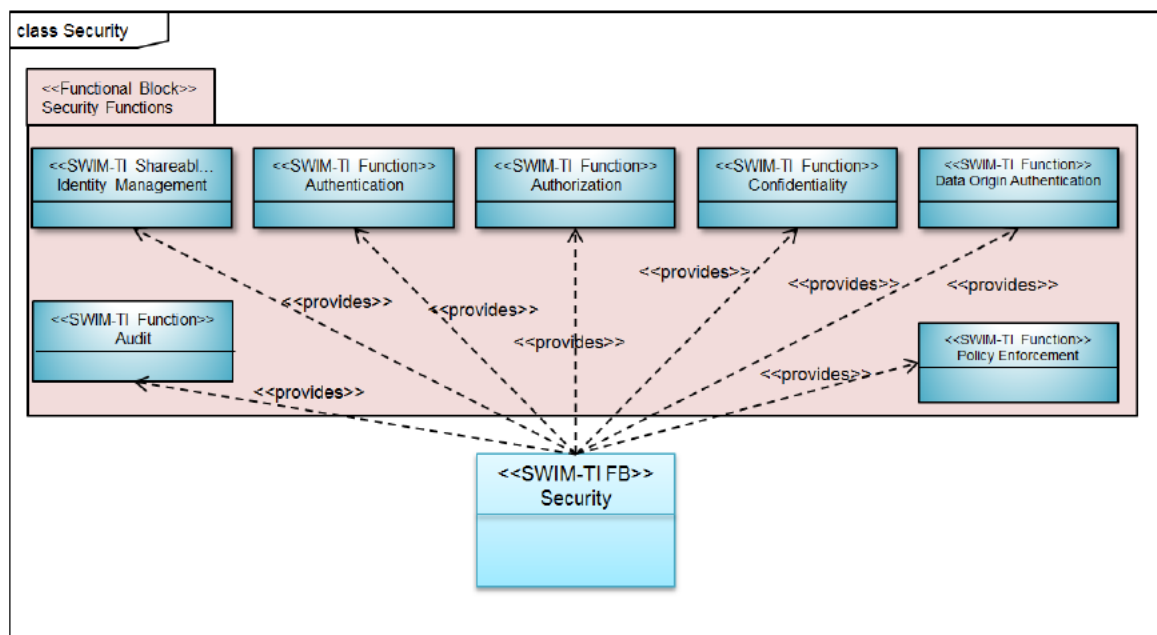


Figure 27 – Security Functional Block breakdown

The Security Functional Block meets the SWIM ConOps requirements (see [11]) listed in the table below.

Identifier	Statement
REQ-08.01.01-CONOPS-ASDE-0020	It is assumed that the policies to be applied in the service provision and consumption (e.g. policies on security, on authentication, etc) will be defined by the “SWIM Collaboration Authority”. Once endorsed, the implementation of such policies will be done on a voluntary and collaborative basis by the impacted stakeholders.
REQ-08.01.01-CONOPS-ASDE.0070	It is assumed for the time being that there will be no common component dedicated to the Security and Authentication functions, although this matter is still subject to further analysis in the context of the SESAR Programme. Although there will be no comment components, the Security and Authentication standards and policies to be applied will be defined for SWIM.
REQ-08.01.01-CONOPS-ASDE.0080	It is assumed that SWIM will not provide common components to manage user profiles or access rights; this management will have to be done by each individual service provider.

Table 8 – SWIM ConOps requirements on Security

### 2.2.1.3.1 Identity Management

#### 2.2.1.3.1.1 Definition

SWIM-TI Identity Management Function is a collection of IT capabilities, business processes and a supporting infrastructure for maintaining and administering Digital Identities within organizations and/or communities, primarily for accessing business operations and resources. It is essentially “user life-cycle management,” reflecting the creation, maintenance, and deletion of identities over time, where a “Digital Identity” is meant to consist of the following parts:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

1. Identifier - One or more attributes used to identify an entity within a context.
2. Credentials - A set of data presented as evidence of a claimed identity and/or entitlements<sup>25</sup>.
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.

SWIM-TI Identity Management Function provides primary activities concerning Digital Identities used in authenticated ATM-information exchanges among SWIM participants. 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:

- 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,
- Return Digital Identity (e.g. SAML token) from validated credential (e.g. user/password),
- Credential validation.

SWIM-TI Identity Management Function contributes to the realization of Brokered Authentication Pattern (see 2.3.5.3.1), which allows to manage trust relationships between Service Consumer 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.

Furthermore, associations between cryptographic keys and corresponding entities, on which Data Origin Authentication and Confidentiality functions are based, are part of Digital Identities information and therefore they are provided by Identity Management Function

---

<sup>25</sup> According to ITU-T IdM X.1252

### 2.2.1.3.1.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Identity Management	Authentication	Identity need to be presented during authentication process to achieve sufficient confidence in the binding with an entity.
	Authorization	Identity Management, Authentication and Authorization allow using Access Control procedure. Authorization leverages on metadata associated with Digital Identities to provide the granting of rights and, based on these rights, the granting of access.
	Confidentiality	Confidentiality mechanisms are based on secrets related to specific identities.
	Data Origin Authentication	Data Origin Authentication mechanisms are based on secrets related to specific identities.,
	Audit	Identity Management relies on Audit to report relevant security events.

### 2.2.1.3.2 Authentication

#### 2.2.1.3.2.1 Definition

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<sup>26</sup> is the process used to achieve sufficient confidence in the binding between the entity and the presented identity. Authorization is granting of rights and, based on these rights, the granting of access. 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”<sup>27</sup>.

In the Access Control procedure 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.

#### 2.2.1.3.2.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Authentication	Identity Management	Identity need to be presented during authentication process to achieve sufficient confidence in the binding with an entity

<sup>26</sup> According to ITU-T X.1252

<sup>27</sup> The periodic symbol for Gold, 'Au' is the prefix for all three processes.

Security Function	Relationship	Relationship Description
	Authorization	Identity Management, Authentication and Authorization allow to use Access Control procedure. The Authorization Function allows the granting of rights and, based on these rights, the granting of access
	Data Origin Authentication	Data Origin Authentication Function, Authentication function and Audit Function together provide accountability and non-reputability to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data.
	Audit	Data Origin Authentication Function, Authentication function and Audit Function together provide accountability and non-reputability to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data. Authentication Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour, e.g. multiple attempts to authenticate with wrong credentials
	Policy Enforcement	Authentication is driven by policies. Authentication policies specify the level of authentication and how to achieve it.

### 2.2.1.3.3 Authorization

#### 2.2.1.3.3.1 Definition

In the Access Control procedure 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: (a) an authorization policy which describes the required security attributes of a user allowing it to access a resource; (b) authenticated identity (user) and its list of security attributes. 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.

In SWIM-TI authorization related features are assigned to the Authorization Function. This function consists on the granting of rights and, based on these rights, the granting of access<sup>28</sup>. This 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.

<sup>28</sup> According to ITU-T X.1252

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.

### 2.2.1.3.3.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Authorization	Identity Management	Identity Management, Authentication and Authorization allow to use Access Control procedure. Authorization leverages on metadata associated with Digital Identities to provide fine grained Attribute Based Access Control
	Authentication	Identity Management, Authentication and Authorization allow to realize Access Control strategies. The Authorization Function allows the granting of rights and, based on these rights, the granting of access.
	Audit	Authorization Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour, e.g. multiple attempts to access a resource by an unauthorized entity
	Policy Enforcement	Authorization is driven by policies. Authorization policies specify the access privileges of authenticated users to given resources.

### 2.2.1.3.4 Confidentiality

#### 2.2.1.3.4.1 Definition

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.

In the SWIM-TI, Confidentiality function covers both data encryption and data decryption for data in transit (e.g. ATM services invoked by a service consumer). Data encryption is performed by the SWIM-TI on service consumer side at service-request sending and by the SWIM-TI on service provider side at service-response sending. Data decryption is performed by the SWIM-TI at service provider side at request reception and by the SWIM-TI at 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 PE 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

founding members



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 non-exclusively at network level, transport level or message level. Only transport and message level are addressed by the SWIM-TI.

Confidentiality Policy determines

- the level of applicability: none, transport, message, both;
- the use of symmetric or asymmetric schemes;
- information about the encryption/decryption algorithm.

#### 2.2.1.3.4.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Confidentiality	Identity Management	Confidentiality mechanisms are based on secrets related to specific identities.
	Audit	Confidentiality Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour.
	Policy Enforcement	Confidentiality is driven by policies. Policies specify the required confidentiality for given data.

#### 2.2.1.3.5 Data Origin Authentication

##### 2.2.1.3.5.1 Definition

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

In SWIM-TI 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 relies on the policy management and enforcement features and it 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 the 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.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Data Origin Authentication function can be addressed non-exclusively at network level, transport level or message level. Only transport and message level are addressed by the SWIM-TI.

As for confidentiality function, 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 of SWIM-TI SEC FB: it is the policy itself, assigned to a given data, that requires or not the need for encryption (for instance). SWIM-TI SEC just processes these policies; therefore SWIM-TI SEC FB is not responsible to specify which data are sensitive and which not<sup>29</sup>.

### 2.2.1.3.5.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Data Origin Authentication	Identity Management	Data Origin Authentication mechanisms are based on secrets related to specific identities.
	Authentication	Data Origin Authentication Function, Authentication function and Audit Function together provide accountability and non-reputability to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data.
	Audit	Data Origin Authentication Function, Authentication function and Audit Function together provide accountability and non-reputability to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data. Furthermore, Data Origin Authentication Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour.
	Policy Enforcement	Data integrity is driven by policies. Policies specify the required integrity levels for given data.

### 2.2.1.3.6 Audit

#### 2.2.1.3.6.1 Definition

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

- Audit generation,
- Data collection and storage,
- Analysis and feedback.

<sup>29</sup> If the data is an ATM specific data, the system projects and WP8 are responsible to define the corresponding policy. There could be SWIM-TI specific data, exchanged across SWIM Nodes, which could be classified as sensitive; in this case WP14 is responsible to define the corresponding policy.



In SWIM-TI the Audit function is limited to the audit generation; it allows (when needed, e.g. when non-repudiation<sup>30</sup> is needed) to create, submit, persistently store and report on audit events<sup>31</sup>. 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 system and therefore cannot be dedicated to the SWIM-TI. The analysis and feedback is similarly performed by correlating security events coming from various sources outside the SWIM-TI.

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

---

<sup>30</sup> The ability to protect against denial by one of the entities involved in an action of having participated in all or part of the action.

<sup>31</sup> Examples of events/activities such as authentication failures or successes, authorized or unauthorized access, encryption/decryption successes and failures, digital signature verification successes and failures.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

### 2.2.1.3.6.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Audit	Identity Management	Identity Management relies on Audit to report relevant security events.
	Authentication	Authentication Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour, e.g. multiple attempts to authenticate with wrong credentials. Data Origin Authentication Function, Authentication function and Audit Function together provide accountability and non-reputability to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data.
	Authorization	Authorization Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour, e.g. multiple attempts to access a resource by an unauthorized entity.
	Confidentiality	Confidentiality Function leverages on Audit Function to report any relevant suspicious or incorrect behaviour.
	Data Origin Authentication	Data Origin Authentication Function, Authentication function and Audit Function together provide accountability and non-reputability to proof of origin of data, proof of submission of data, proof of transport of data and proof of delivery of data.
	Policy Enforcement	The Audit-policy defines which events need to be audited and for which activities or resources.

### 2.2.1.3.7 Policy Enforcement

#### 2.2.1.3.7.1 Definition

The Policy Enforcement<sup>32</sup> Function deals with the application of policies at SWIM-TI Node level.

Policy Enforcement enforces policies for admission control and policy decisions in response to a request from other SWIM-TI FBs wanting to use a resource.

The concept of policy enforcement is an architectural pattern for implementation of general cross cutting concerns; within an SWIM Node, policy enforcement simplifies per service end-point definitions and maintenance of security relevant rules of communication.

- The major capabilities (entities<sup>33</sup>) of the Policy Enforcement Function 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).

<sup>32</sup> Taking into account that in general a policy driven approach can be adopted not only for security purpose (e.g. Data Validation, Specific Message policies, etc.), dedicated Policy Enforcement and Policy Management have been identified. For what concerns the Policy Management, all the identified functions have been allocated to the Registry FB whereas a new FB has been defined for Policy Enforcement.

<sup>33</sup> Usually associated with SW instances.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- **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.

### 2.2.1.3.7.2 Relationship with other Security functions

Security Function	Relationship	Relationship Description
Policy Enforcement	Authentication	Authentication is driven by policies. Authentication policies specify the level of authentication and how to achieve it.
	Authorization	Authorization is driven by policies. Authorization policies specify the access privileges of authenticated users to given resources.
	Confidentiality	Confidentiality is driven by policies. Policies specify the required confidentiality for given data.
	Data Origin Authentication	Data integrity is driven by policies. Policies specify the required integrity levels for given data.
	Audit	The Audit-policy defines which events need to be audited and for which activities or resources.

### 2.2.1.3.8 Security Functional Block Dependencies

The table below summarizes the dependencies Used by SEC FB:

FB	Dependency	Optional / Mandatory	Dependency Description
Security (SEC)	Registry (REG)	Mandatory	Use of the common security policy management in order to retrieve/be notified security policies. This dependency is not mandatory for all the security policies but it is mandatory only for those applicable at regional level.

**Table 9 – Security Functional Block Dependencies**

The figure below summarizes the dependencies Used/Provided by SEC FB:

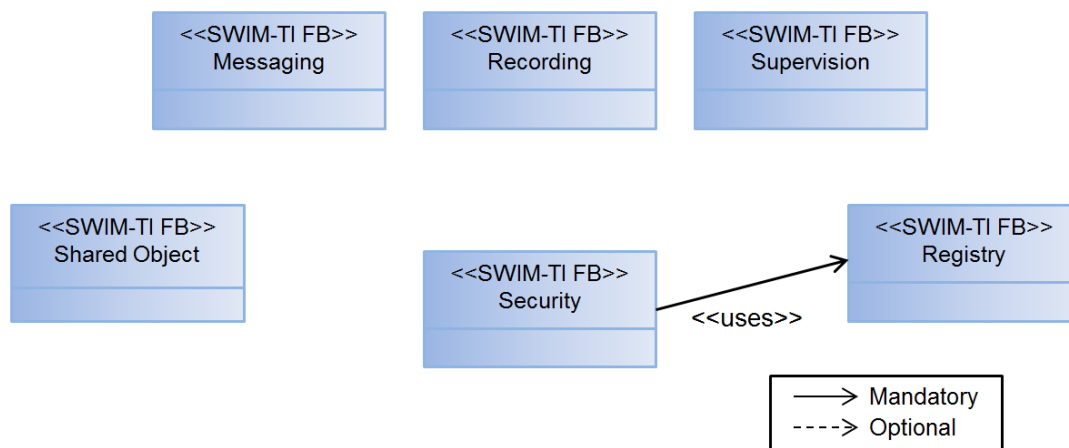


Figure 28 – Security Functional Block Dependencies

### 2.2.1.3.8.1 Use Dependencies

#### 1. Registry FB (REG FB) Dependency

The Registry FB provides a set of functionalities aiming at managing policies lifecycle. To provide security policies, the Security FB relies on policy management and policy enforcement and therefore Registry shall be able to manage several kinds of policies for security purposes.

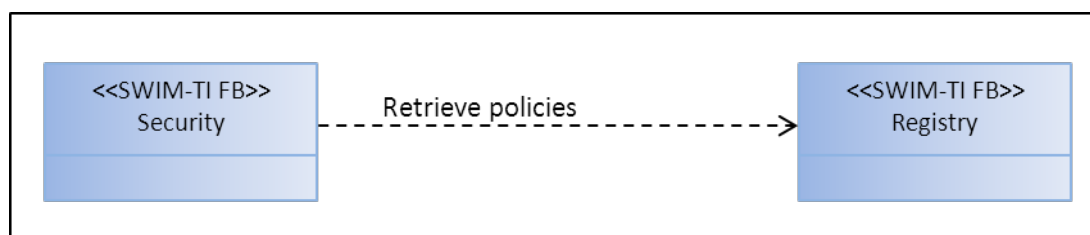


Figure 29 – SEC FB use of REG FB Dependencies

## 2.2.1.4 Supervision Functional Block

The SWIM-TI Functional Block Supervision supports all SWIM related supervision functions collocated with the system. It is broken down following the diagram below.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

101 of 284

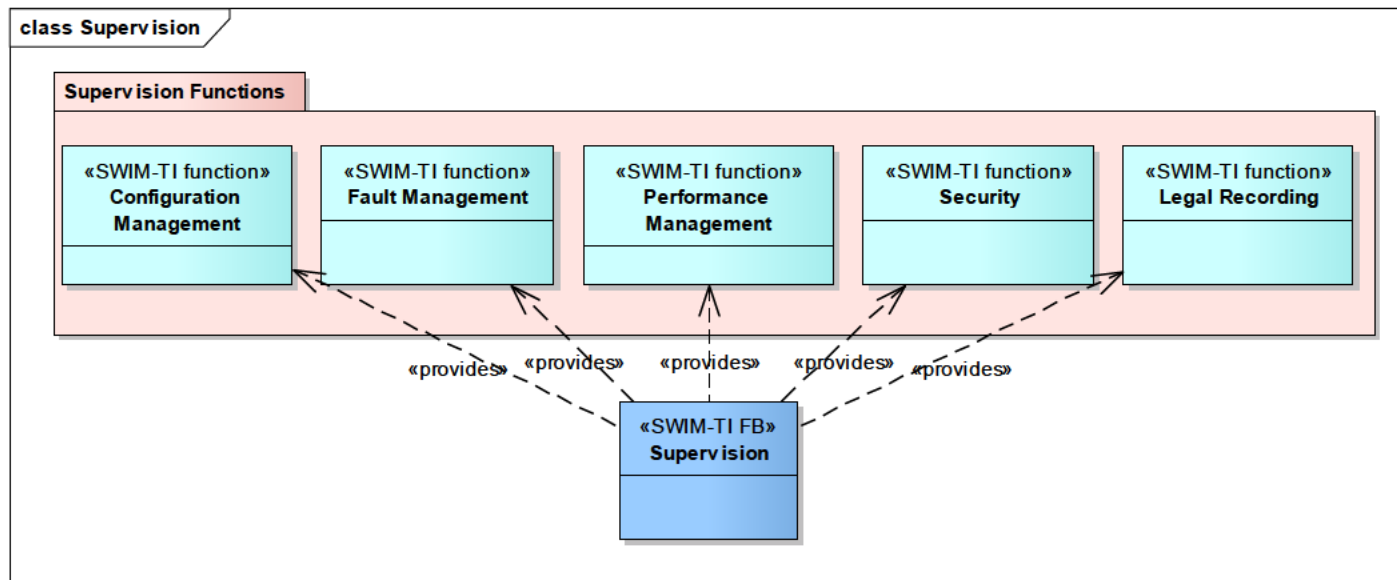


Figure 30 – Supervision Functional Block breakdown

### 2.2.1.4.1 Supervised Entities

SWIM-TI Supervision Functional Block manages the following entities<sup>34</sup>:

It's needed to highlight again the notion of SWIM-TI Node, being a logical aggregation of SWIM-TI Functional Blocks instantiated on a SWIM-TI Profile Basis. However, concrete deployments of this concept could be associated to physical assets (e.g. a SWIM Node deployed in a dedicated hardware element).

- SWIM-TI Node Hardware

The SWIM-TI Node Hardware is composed of those physical elements that somehow are involved in enabling the functionality of the SWIM-Node (e.g. a PC, a router, etc.).

Not in all the deployments this hardware needs to be monitored by SWIM-TI Supervision (e.g. those nodes that are embedded in a System that already has a function that deals with the management of all the hardware elements).

- SWIM-TI Node Processes

The SWIM-TI Node Processes are those software elements that somehow are involved in enabling the functionality of the SWIM-Node (e.g. the Security COTS, the messaging COTS, etc.).

- SWIM-TI Enabling Services Status

The SWIM-TI Enabling Services are those infrastructure services provided by SWIM-TI (not to be mistake with the SWIM Functional Blocks) acting as a System to another System (e.g. the provision of the information regarding the status of the supervised entities from a SWIM-TI Supervision towards another SWIM-TI Supervision)

- SWIM-TI ATM Services Status

The SWIM-TI ATM Services are those services offered by an ATM System via the SWIM-TI (via the SWIM-TI) (e.g. the provision of the Weather info from a MET System towards an aircraft).

- SWIM-TI Node

As described, a SWIM Node is a logical aggregation of certain capabilities. The supervision of the SWIM Node means the report of all those elements (sw/hw/services/etc.) that are described as a part of such SWIM-TI Node.

### 2.2.1.4.2 Configuration Management

Configuration Management focuses on establishing and maintaining consistency of a system's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life [4].

SWIM-TI Configuration Management function is limited to the SWIM-TI, in terms of SWIM-TI Node hardware/software and applications for the provision of SWIM Services.

---

<sup>34</sup> To be noted that this entities are not mandatory to be supervised by SWIM-TI Supervision; what is being described is the capability of apply certain SWIM-TI Supervision functionalities to certain items of the SWIM Node. This is also aligned with the notion of SWIM Node as logical entity acting as an aggregator of capabilities/functionalities.

### 2.2.1.4.3 Fault Management

Fault management is the set of functions that detect, isolate, and correct malfunctions in a SWIM-TI Node.

Fault management is the functional area that provides **control** towards the different entities that are part of the SWIM-TI Node. **Control** function allows the supervisor to perform command and actions on the supervised object which is part of the SWIM node. All these actions are logged and stored for future access using the Recording function (see 2.1.1.7).

In order to detect the malfunctions, a **lifecycle** is needed to be associated to the different entities that are part of the SWIM-TI Node. A concrete status will identify whether the entities are working appropriately or, on the contrary, they are in a degraded/error **status**.

In case of error, and also as a part of the Fault Management, an **Alarming** sub function is foreseen.

Also in case of error, **Recovery procedures** and **safety measures** should be analysed and provided by SWIM Supervision.

According to SWIM-TI Supervision Step1 ([9]) the following generic will be applied for the Fault Management of the supervised entities:

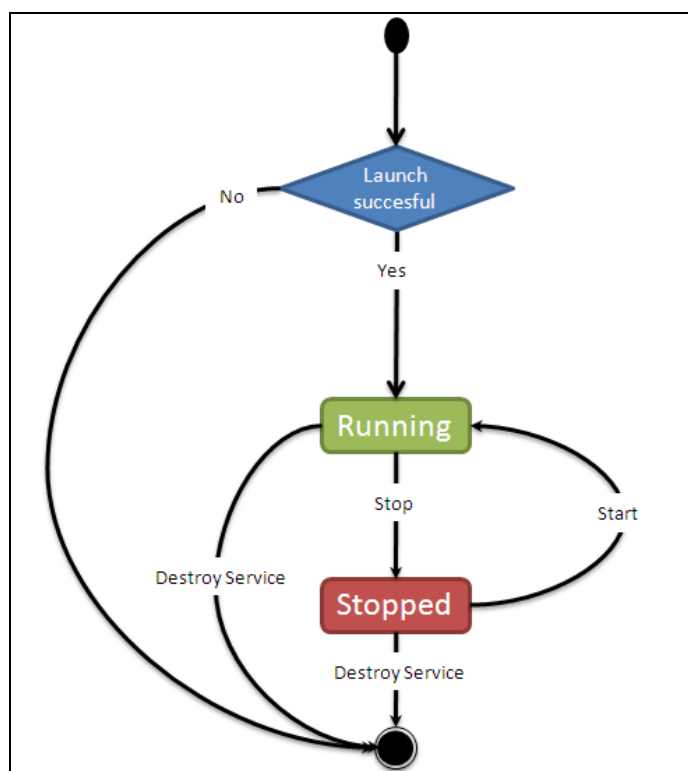


Figure 31 – SWIM-TI Supervised Entity Lifecycle

Where the Supervised Entity will report its status<sup>35</sup> as:

- **Running.** The Supervised Entity it is available and ready to be used.

<sup>35</sup> Note that a more complex lifecycle can exist within each node, but not necessarily to be standardized



- **Stopped.** The Supervised Entity stopped its execution

### 2.2.1.4.4 Performance Management

Performance management focuses on monitoring and managing the performance and service availability of the supervised entities.

It can be defined as the process to detect, and report application’s performance regarding the end-users’ expectations (often formalized as Service Level Agreements, SLA).

It usually deals with the monitoring of the performance and the provision of statistics associated to it.

### 2.2.1.4.5 Security

Security Management at SWIM-TI Supervision Level relates to the ability of enabling (or denying) the access to the SWIM-TI Node elements to an external actor.

### 2.2.1.4.6 Legal Recording

Legal recording refers to the recording of all data exchanged in the context of the supervision function (e.g. events), commands issued and their results, etc.

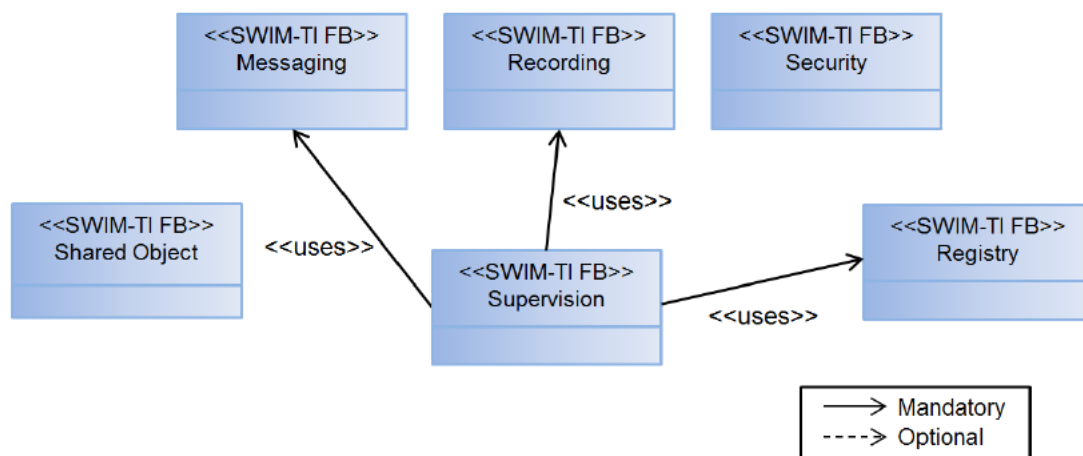
### 2.2.1.4.7 Supervision Functional Block Dependencies

The table below summarizes the dependencies Used by SPV FB:

FB	Dependency	Optional / Mandatory	Dependency Description
Supervision (SPV)	Recording (REC)	Mandatory	To record supervision information.
	Registry (REG)	Mandatory	Supervision policy enforcement relies on the existence of a policy lifecycle management.
	Messaging (MSG)	Mandatory	Used to communicate service status to consumer. Possible use of supervision is the provisioning of metrics (in Step 1 there are already some defined for this) concerning the two main MEPs.

**Table 10 – Supervision Functional Block Dependencies**

The figure below summarizes the dependencies Used by SPV FB:



founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Figure 32 – Supervision Functional Block Dependencies

### 2.2.1.4.7.1 Use Dependencies

#### 1. Recording (REC FB) Dependency

Recording FB provides a set of functionalities aiming at allowing the recording of selected Supervision data and messages.

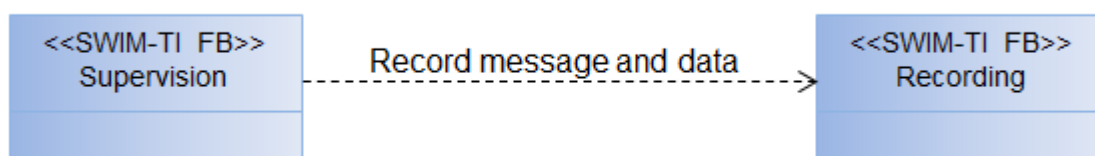


Figure 33 – SPV FB use of REC FB Dependencies

#### 2. Registry (REG FB) Dependency

The Registry FB provides a set of functionalities aiming at managing policies lifecycle. Also Supervision Policies may be included among them. To provide the appropriate SPV measures the Messaging FB relies on policy management and policy enforcement and therefore Registry shall be able to manage several kinds of policies for Registry, also storing and distributing messaging policies for enforcement purposes.

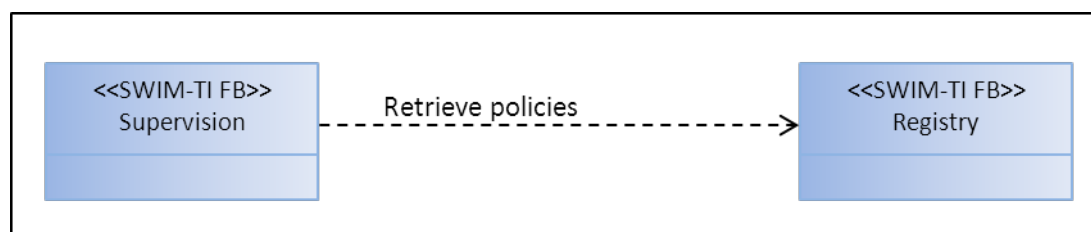
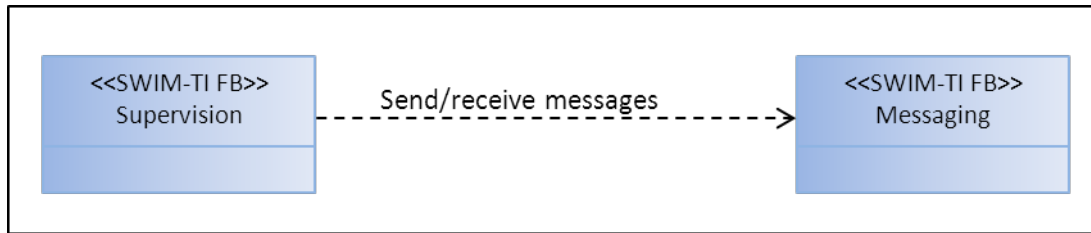


Figure 34 – SPV FB use of REG FB Dependencies

#### 3. Messaging (MSG FB) Dependency

The Supervision FB may want to communicate service status to consumer. Possible use of supervision is the provisioning of metrics (in Step 1 there are already some defined for this). For doing that, on top of a pure SWIM-TI MSG FB, it incorporates certain logic that enables sending SPV information among different stakeholders. As such, it makes use of all the dependencies used by MSG, but indirectly through that FB.



**Figure 35 – SPV FB use of MSG FB Dependencies**

The following dependencies from MSG FB will be used:

- Security (SEC) (Mandatory)  
Use of security functions in order to:
  - authorize and authenticate message producer and consumer
  - ensure data confidentiality, integrity and authenticity
- Registry (REG) (Optional)  
Messaging related policies enforcement relies on the existence of a policy management
- Recording (REC) (Optional)  
Use of Recording to keep track of the selected messages.

## 2.2.1.5 Recording Functional Block

The SWIM-TI Functional Block Recording includes the ability to collect, store and on demand retrieval of information related to:

- communication being performed via the SWIM Interfaces,
- supervision actions and events as defined in 2.1.1.6.

The Recording function does not include the audit logging required by security function. For security reason these particular data are kept apart.

This function collects the communication session data based on predefined configuration. For example, recorded information might be formatted in a record entity per interaction and could include following attributes:

- Service execution time stamp
- Service name
- Service requestor and provider identity information (user name, security token)
- Data payload (possible encrypted)
- Data payload signature

The number and type of session attributes which will be recorded is dependent on the MEP (see P14.1.3-D38 [10] for MEP characterisation) associated with the SWIM service, its technical implementation and the SWIM node role (i.e. service provider or service consumer).

The major goal of recording capability is to provide evidence about processed communication sessions, exchanged data payloads and involved communication participants. In case of communication sessions with message encryption and signatures based on PKI approach with trusted certificates, the recordings can be used to support the concept of legal recording. Legal recording has to be tamper-proof and requires a minimum retention time. A legal recording policy will define which events are recorded and how long is their minimum retention time.

The SWIM ConOps (see [11]) lists the following assumption related to recording.

Identifier	Statement
REQ-08.01.01-CONOPS-ASDE-0060	It is assumed that there will be no common component dedicated to the legal recording function. This implies that each service provider and consumer will have to check the necessity to implement legal recording for the services it provides or consumes.

**Table 11 – SWIM ConOps requirements on SWIM-TI Recording Functional Block**

It is not realistic to design the SWIM-TI to provide a common component to centralize the legal recording of all the European ATM systems connected to the SWIM. Such a design would require storage capacities and network capacities that are far beyond what current technologies can provide at an affordable cost.

The SWIM ConOps requirement is kept as an assumption in the TAD so it can be traced in case the hypothesis is no more valid.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

108 of 284

ASM-0006	SWIM-TI does not provide any centralized component for legal recording.
----------	---

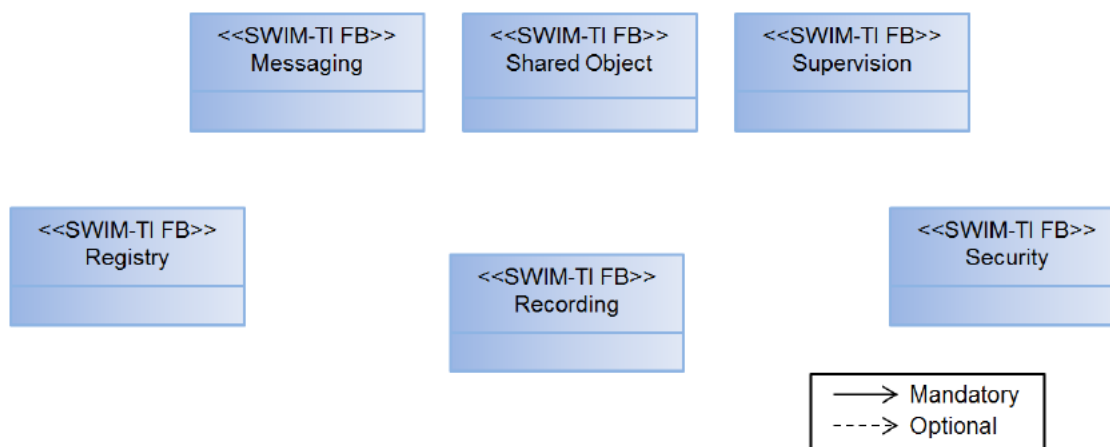
### 2.2.1.5.1 Recording Functional Block Dependencies

The table below summarizes the dependencies Used by Recording FB (none identified):

FB	Dependency	Optional / Mandatory	Dependency Description
Recording (REC)	None	N/A	N/A

**Table 12 – Recording Functional Block Dependencies**

The figure below summarizes the dependencies Used by Recording FB (none identified):



**Figure 36 – Recording Functional Block Dependencies**

#### 2.2.1.5.1.1 Use Dependencies

Recording FB doesn't rely on any other SWIM-TI FB to be able to provide its functionality.

### 2.2.1.6 Shared object Functional Block

The SWIM-TI Functional Block Shared Object function allows the sharing of data across multiple SWIM Nodes. This capability uses publish/subscribe and request/reply messaging pattern and allows multiple operations on an agreed data model:

- Creation, update, delete, search;
- Request for service;
- Restore data;
- Recovery

Every shared object has a manager. If a participant has the manager role, it is responsible of the shared object and it is the only one allowed to update and delete that shared object.

Every shared object has a distribution list. The distribution list is the list of participants interested in a shared object. The Shared Object function uses data validation to perform compliance checking of message payload.

Shared Object function is specified in P14.01.04-D44 ([13]).

#### 2.2.1.6.1 Recovery

As ED-133 lacks a mature recovery process definition, an alternative recovery process has been designed and validated

The following assumptions have been considered when specifying the Blue Profile Recovery process:

- The Recovery process can be triggered on different ways:
  - On demand by the IOP Application,
  - Automatically (if certain criteria are met, e.g. at start-up or upon reconnection after temporarily isolation from the IOP network). In that case, a local 'automatic recovery policy' will define the rules to drive the recovery process, with no input (configuration file) or limited input from the IOP Application.
- The Recovery process aims at recuperating:
  - The most up-to-date version of the Flight Objects for which the SWIM Node is part of its Distribution List,
  - Summaries of all the Flight Objects in the SWIM Network.
- Efficiency issues are considered:
  - Minimize the possibility of a "storm of updates",
  - Updates to ED-133 Flight Object Data Model and Services will be considered if necessary but kept to a minimum.
- The Recovery process for Blue Profile does not constitute an ATM Information Exchange as it doesn't support the provision of any new information that was not already available. Instead it should be seen as a technical feature of the SWIM-TI that allows its autonomous recovery in case a sub-set of the SWIM-TI Network falls down

An overview of the designed recovery mechanism is provided hereafter.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

110 of 284

The approach is based on “Recovery Tiers” (i.e. Recovery Tier 1, Recovery Tier 2 up to Recovery Tier n). Each SWIM Node in the Distribution List of a Flight Object is associated with a Tier.

This tiered approach allows to:

- perform the recovery process in sequential steps, in order to prevent a storm of updates arriving at the recovering SWIM Node,
- ensure that the most interesting Flight Objects are recovered first.

Each Flight Object has an enriched Distribution List in which every stakeholder is assigned a Tier according to its priority in the recovery process. It is important to note that a Tier is associated to every SWIM Node in the Distribution List for each Flight Object. Hence a SWIM Node can have different Tiers associated for different Flight Objects (since he might be further downstream for some Flight Objects than others).

The number of Tiers can be configured to the optimal value that ensures, the most critical Flight Objects are received soon enough while mitigating a “storm of updates” on the receiving SWIM Node.

An example of assignment logic for the Tiers is provided below for a given Flight Object:

- Tier 0 is associated to the SWIM Node whose ATSU holds responsibility of the Flight.
- Tier 1 is associated to the SWIM Nodes whose ATSU are crossed next downstream.
- Tier 2 is associated to all the other SWIM Nodes in the Distribution List.

It is important to notice that the Tier approach is quite generic in concept and does not depend on the particular definition of Tiers. It consists on a sequential recovery process together with a particular criterion to determine the sequence of recovery. The specific definition of Tiers is out of scope of the SWIM-TI, for which a Tier is only a priority of recovery associated to a FO. This specification doesn't intend to define the business logic to map stakeholders and Flight Objects to Tiers; the definition provided above should serve simply as an example providing guidance.

Assumption: the definition of the mapping that associates a Tier for each stakeholder in the Distribution List is to be provided by the IOP Application (P10.02.05).

The Blue Profile recovery mechanism takes the strengths of PS-MEP Recovery (FlightObjectDistribution interface) with the added flow control of SRR-MEP Recovery (FlightObjectManagement interface) as a back-up mechanism. This allows for an efficient approach while ensuring that the Recovering SWIM Node can rely on a Request/Reply mechanism in case any of the expected Flight Objects are not recovered during the process for unexpected reasons.

### 2.2.1.6.2 Shared Object Functional Block Dependencies

The table below summarizes the dependencies Used by Shared Object FB:

FB	Dependency	Optional / Mandatory	Dependency Description
Shared Object (SO)	Messaging (MSG)	Mandatory	Used to exchanged data and services according to publish-subscribe and request-response MEPs.

**Table 13 – Shared Object Functional Block Dependencies**

The figure below summarizes the dependencies Used by SO FB:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

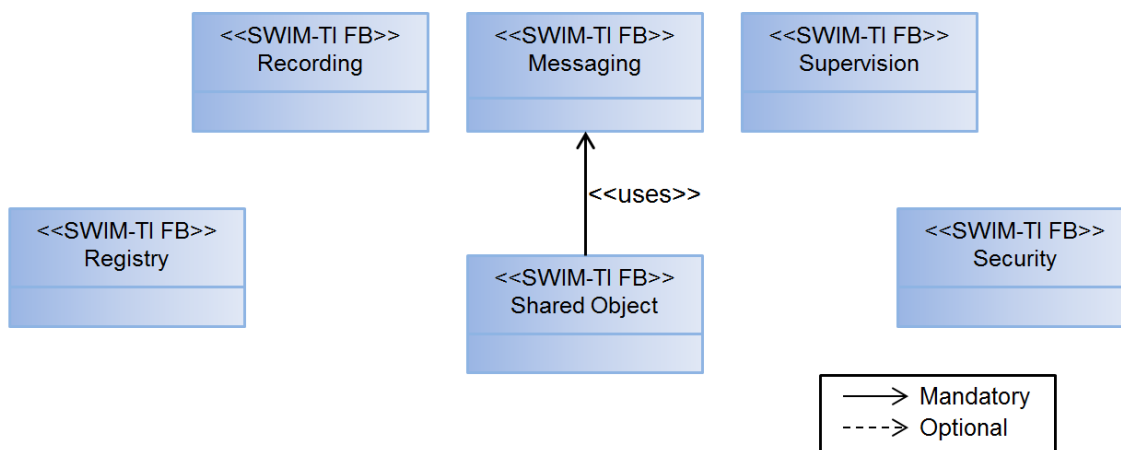


Figure 37 – Shared Object Functional Block Dependencies

### 2.2.1.6.2.1 Use Dependencies

#### 1. Messaging (MSG FB) Dependency

The Shared Object FB provides a set of functionalities aiming at interchanging Shared Objects between different Systems. On top of a pure SWIM-TI MSG FB, it incorporates certain logic that enables the interoperability among these Systems. As such, it makes use of all the dependencies used by MSG, but indirectly through that FB.



Figure 38 – SO FB use of MSG FB Dependencies

The following dependencies from MSG FB will be used:

- Security (SEC) (Mandatory)  
Use of security functions in order to:
  - authorize and authenticate message producer and consumer,
  - ensure data confidentiality, integrity and authenticity.
- Registry (REG) (Optional)  
Messaging related policies enforcement relies on the existence of a policy management.
- Recording (REC) (Optional)  
Use of Recording to keep track of the selected messages.



## 2.2.2 Functional Analysis

The table below summarizes the dependencies between Functional Blocks:

FB	Dependency	Optional / Mandatory	Dependency Description
Registry (REG)	Security (SEC)	Optional	Use of Security functions in order to authorize and authenticate users of the Registry.
Messaging (MSG)	Security (SEC)	Mandatory	Use of security functions in order to: - authorize and authenticate message producer and consumer ensure data confidentiality, integrity and authenticity.
	Registry (REG)	Optional	Messaging related policies enforcement relies on the existence of a policy management.
	Recording (REC)	Optional	Use of Recording to keep track of the selected messages.
Security (SEC)	Registry (REG)	Mandatory	Use of the common security policy management in order to retrieve/be notified security policies.
Supervision (SPV)	Registry (REG)	Mandatory	Supervision policy enforcement relies on the existence of a policy lifecycle management.
	Messaging (MSG)	Mandatory	Used to communicate service status to consumer. Possible use of supervision is the provisioning of metrics (in Step 1 there are already some defined for this) concerning the two main MEPs.
	Recording (REC)	Mandatory	To record supervision information.
Recording (REC)	<i>None</i>	N/A	N/A
Shared Object (SO)	Messaging (MSG)	Mandatory	Used to exchanged data and services according to publish-subscribe and request-response MEPs.

**Table 14 – Functional block dependencies**

## 2.3 Technical View

### 2.3.1 SWIM-TI Technical Entities

The Technical View of a System identifies the Technical Entities that compose the System and identifies and describes the interfaces between these technical entities. SWIM-TI is a set of software components distributed over a network infrastructure providing functions enabling collaboration among ATM systems.

In order to understand the Technical view of SWIM-TI, a set of definitions need to be established. The following definitions are provided in the ADD document (ADD, ref. [6]):

<b>Domain System</b>	A <b>Domain System</b> is a System that provides a set of functionalities which are closely linked to the domain's business needs.
<b>Infrastructure system</b>	An <b>Infrastructure System</b> is a System providing a collection of functionalities which are agnostic to the ATM business processes.

According to this and particularized to the SWIM Technical Infrastructure, the following terminology stands:

<b>SWIM-TI Infrastructure System</b>	A SWIM-TI Infrastructure System is a System providing a collection of SWIM-TI shareable functions.
<b>SWIM-TI Profile</b> <sup>3637</sup>	<p>A <b>SWIM-TI Profile</b> is a <b>SWIM-TI Infrastructure Systems</b> that groups a coherent, appropriately-sized <b>set of SWIM-TI Functional Blocks</b> for a given set of technical constraints/requirements that permit a set of stakeholders to realize Information sharing.</p> <p>It also defines the mandated open standards and technologies required to realize this coherent grouping of middleware functions/services.</p> <p>A SWIM-TI Profile is a concrete group of SWIM-TI Functional Blocks. For each SWIM-TI Functional Block, a SWIM-TI Profile Instantiation derived from the <b>SWIM-TI Profile Descriptor</b> will define a concrete set of requirements<sup>38</sup></p> <p>Each SWIM-TI Profile Instantiation can be understood as a <b>specific instance</b> of the SWIM-TI FB decomposition.</p>

The main technical entities provided by SWIM-TI are SWIM-TI Infrastructure Systems that are realization of SWIM-TI Functional Blocks.

A SWIM-TI Infrastructure System is either:

<sup>36</sup> Also known as SWIM Profile.

<sup>37</sup> As defined in P14.01.03 D038 (ref. [11]).

<sup>38</sup> Two different SWIM-TI Profiles don't necessarily have to share the same requirements even if they are implementing both the same SWIM-TI Functional Blocks.

- **An Infrastructure Systems** (realization of a consistent set of **SWIM-TI Shareable Functions**)
- **A SWIM-TI Profile (SPIs, realization of a consistent set of SWIM-TI Functional Blocks)**

The last entity to be used in the technical view of the SWIM-TI is the SWIM-TI Node.

<b>SWIM-TI Node</b> <sup>3940</sup>	<p>SWIM-TI Node provides one or more collections of <b>SWIM-TI functions</b>, grouped in accordance with deployment conformance specifications. A SWIM-TI Node allows a given ATM application to use the SWIM-TI and/or a SWIM-TI Node supports the SWIM-TI.</p> <p>Being a logical entity, its deployment choices are free for the implementation to be chosen.</p>
-------------------------------------	--

As described above, A SWIM-TI Node is a **logical entity** that **groups one or more SWIM profiles** (and hence, that provides a collection of SWIM-TI Functional Blocks).

SWIM-TI Node is designed to meet the SWIM ConOps requirements (see [11]) listed in the table below. They are the only requirements from the SWIM ConOps applicable to the SWIM node. Nonetheless and being SWIM Node a logical entity, these requirements should be traced to SWIM-TI Infrastructure Systems (e.g. SWIM-TI Profiles).

Identifier	Statement
REQ-08.01.01-CONOPS-ASDE-0090	It is assumed that the SWIM Node is defined as a logical entity. This implies that they can be implemented and deployed a) as their own dedicated environment, independent of the software components of the domain system or b) as a software component integrated together with other software components of the domain system.
REQ-08.01.01-CONOPS-ASDE-0100	It is assumed that the interfaces between the SWIM node and the software components of the domain systems can be standardized (open) or proprietary defined. The analysis of the cases where such standardization could be needed is on-going.
REQ-08.01.01-CONOPS-ASDE-0060	It is assumed that there will be no common component dedicated to the legal recording function. This implies that each service provider and consumer will have to check the necessity to implement legal recording for the services it provides or consumes.
REQ-08.01.01-CONOPS-ASDE.0040	It is assumed that technical SWIM supervision encompassing one or more SWIM nodes and ATM-Specific services (e.g. the part under the responsibility of an ANSP, or under the responsibility of a FAB) can be needed.

**Table 15 – SWIM ConOps requirements on SWIM-TI Node**

<sup>39</sup> Also known as SWIM Node.

<sup>40</sup> The concept of SWIM-TI Node is kept to enable the retro-traceability to the material and for being used in several documents in the SWIM-TI, but is recognized as a way to present the SWIM-TI rather than a Technical Entity.

## 2.3.2 SWIM-TI Interface Bindings

According to the ADD (ref. [6]), STG (ref. [4]), and EATMA Guidance (ref. [63]) System Ports are defined as:

<b>System Port</b>	<p>A <b>System Port</b> is an interface provided by a System.</p> <p>A <b>System Port Connector</b> asserts that a connection exists between two System Ports.</p>
--------------------	--

The SWIM Technical Infrastructure supports the definition of system ports for application systems using the SWIM-TI by defining Interface Bindings.

<b>SWIM-TI Interface Binding</b>	<p>A SWIM-TI Interface Binding is the definition of a set of protocols/standards describing a dedicated interface supported by the SWIM-TI.</p> <p>A SWIM-TI Interface Binding can be used to describe the details of System Ports for systems communicating via SWIM.</p>
----------------------------------	--

Each SWIM-TI Profile represents a different combination of SWIM-TI functionalities and will use different underlying communication technologies.

The list of SWIM-TI Interface Bindings contains a broad variety of bindings, differentiated by MEP, protocol versions, security features, etc. Each SWIM-TI profile defines a set bindings; one binding may be part of one or several profiles.

The entire SWIM-TI includes the following groups of interface bindings:

- SWIM-TI Blue Profile Service Bindings
- SWIM-TI Purple Profile Service Bindings
- SWIM-TI Yellow Profile Service Bindings
- SWIM-TI Runtime Registry Internal Service Bindings
- SWIM-TI Identity Management Internal Service Bindings

Each SWIM-TI Interface Binding is described by a set of standards whose concrete configuration and usage are what represent the SWIM-TI functionality. The set of standards/protocols used by each SWIM-TI Interface Binding are defined in the SWIM-TI TS (ref. [13]).

## 2.3.3 Technical Architecture View

SWIM-TI is composed by distributed SWIM-TI Infrastructure Systems (realization of SWIM-TI Functional Blocks) and SWIM-TI Profiles (SPIs, realization of sets of SWIM-TI Functional Blocks).

The following figure represents all the current identified Technical Entities that can be found in the SWIM-TI:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

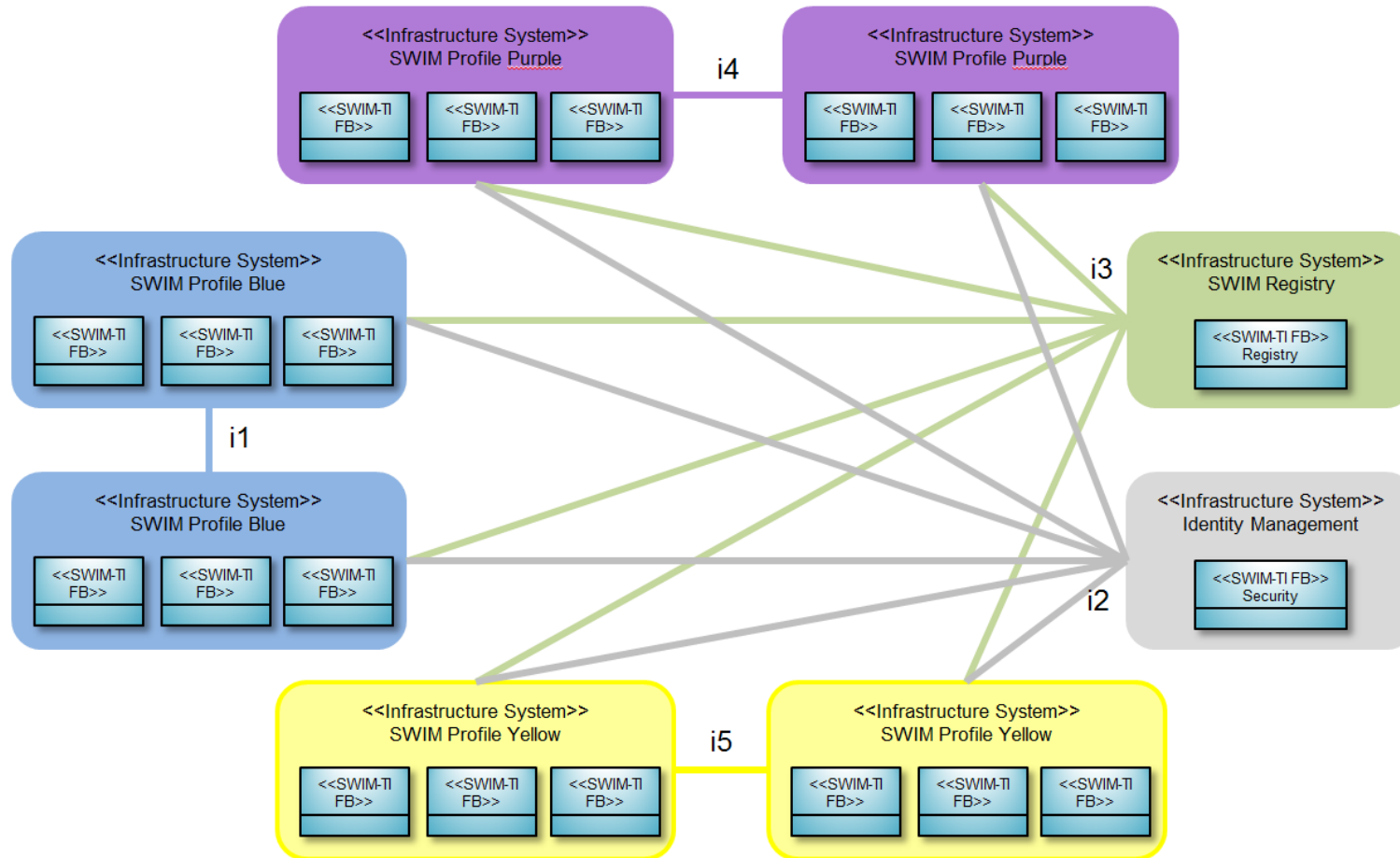


Figure 39 – SWIM-TI Technical Architecture View

In the figure above, the following interfaces have been identified:

Interface	Name	Description
i1	BP-BP Interface	<p>Blue Profile – Blue Profile SWIM-TI Infrastructure Systems Interface</p> <p>Defines the interface offered by the Blue Profile.</p> <p>Uses the following SWIM-TI interface bindings:</p> <ul style="list-style-type: none"> <li>• SWIM-TI Blue Profile Service Bindings</li> </ul>
i2	Identity Management Interface	<p>SWIM-TI Identity Management Infrastructure Systems Interface</p> <p>Defines the interface to access the Identity Management from any SWIM-TI Infrastructure Systems.</p> <p>Uses the following SWIM-TI interface bindings:</p> <ul style="list-style-type: none"> <li>• SWIM-TI Identity Management Internal Service Bindings</li> </ul>
i3	Registry Interface	<p>Registry Infrastructure Systems Interface</p> <p>Defines the interface to access the Registry from any SWIM-TI Infrastructure Systems.</p> <p>Uses the following SWIM-TI interface bindings:</p> <ul style="list-style-type: none"> <li>• SWIM-TI Runtime Registry Internal Service Bindings</li> </ul>

Interface	Name	Description
i4	PP-PP Interface	<p>Purple Profile – Purple Profile SWIM-TI Infrastructure Systems Interface</p> <p>Defines the interface offered by the Purple Profile.</p> <p>Uses the following SWIM-TI interface bindings:</p> <ul style="list-style-type: none"> <li>• SWIM-TI Purple Profile Service Bindings</li> </ul>
i5	YP-YP Interface	<p>Yellow Profile – Yellow Profile SWIM-TI Infrastructure Systems Interface</p> <p>Defines the interface provided by the Yellow Profile.</p> <p>Uses the following SWIM-TI interface bindings:</p> <ul style="list-style-type: none"> <li>• SWIM-TI Yellow Profile Service Bindings</li> </ul>

Table 16 – SWIM-TI interfaces

### 2.3.4 Architecture and Network Domains

The target network architecture includes multiple network domains under the responsibility of multiple stakeholders. Therefore, an end-to-end architecture has to take into account the many ownership and/or security domains in effect and highlight the requirements and responsibilities allocated to stakeholders.

A quick classification of such domains as in Figure 40 involves two domain categories:

- **Intra-Domain:** applies to the network infrastructure within the ANSP's Local Area Network (or Airspace User). This may also include firewalls and specific security requirements.
- **Inter-Domain:** refers to the interconnection network such as PENS. This is typically a Wide Area Network under the responsibility of one or more network providers.

For a clear separation of responsibilities and a successful integration of SWIM-TI implementations to SWIM, the SWIM ICD should tackle both such domains.

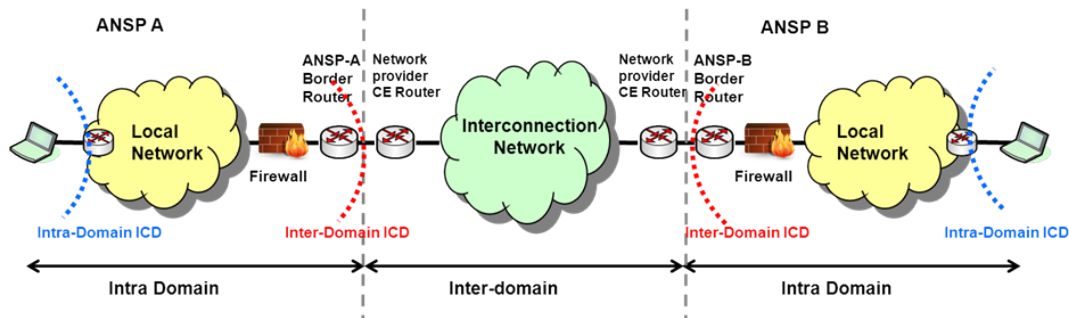


Figure 40 – Network Domains

The Intra-Domain ICD will contain the protocols that must be supported by the SWIM node, connected to the local ANSP’s local network.

The Inter-Domain ICD will contain the protocols that must be supported by the Inter-Domain routers on the border between the WAN backbone and the ANSP.

### 2.3.4.1 Network architecture

SWIM-TI only assumes that end-to-end IP connectivity is available for what network architecture is concerned. This end-to-end IP connectivity can be achieved by IPV6 “Mobile IP”<sup>41</sup> for the particular case of the A/G connection.

Figure 41 shows the typical architecture connecting two stakeholder systems through the PENS (or any IP network used for SWIM exchanges) via the SWIM-TI. The depicted architecture is limited to the G/G interconnecting networks.

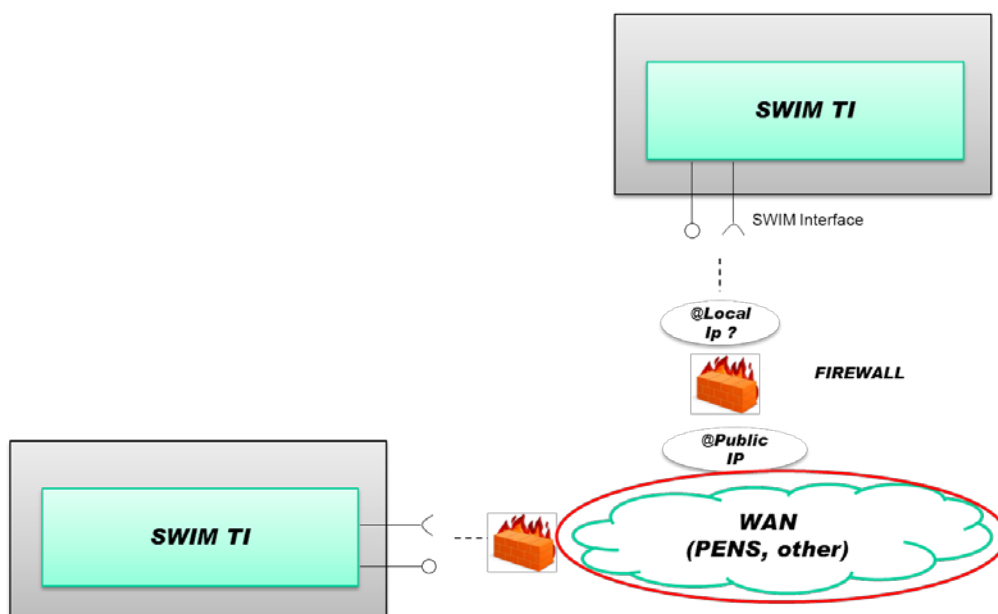


Figure 41 – Typical Architecture

SWIM interface is exposed through a “local” IP address that is an address part of the stakeholder IP-addressing-plan. The at least one firewall needs to be traversed for security reason. Finally the exchanged traffic is routed through a “public” IP address that is part of the PENS IP-addressing-plan.

<sup>41</sup> As defined by IETF RFC 6275



## 2.3.5 Architectural Options

The Architectural Options for the SWIM-TI aim at describing the various possibilities of realizing the different Functional Blocks described in the Functional View.

### 2.3.5.1 Registry FB Architectural Options

SWIM-TI Registry Functional Block can be realized into a SWIM-TI Infrastructure System in various manners.

Different architectures can be described, depending on the centralization or distribution of the information storage and the kind of relationship between root repository and affiliate repository.

The options are described hereafter.

#### 2.3.5.1.1 Architectural options

##### 1. Self-contained root registry

In this option information is stored at a unique location in a central data store, the *Root Registry*.

Information Consumer	The consumer of information obtains all information through the root registry, as there is no other source of information.
Information Provider	The provider of information manages the information directly in the root registry. The management of information is federated, meaning that each data owner manages only the data sets that it owns.

Table 17 – Self-contained root Registry roles

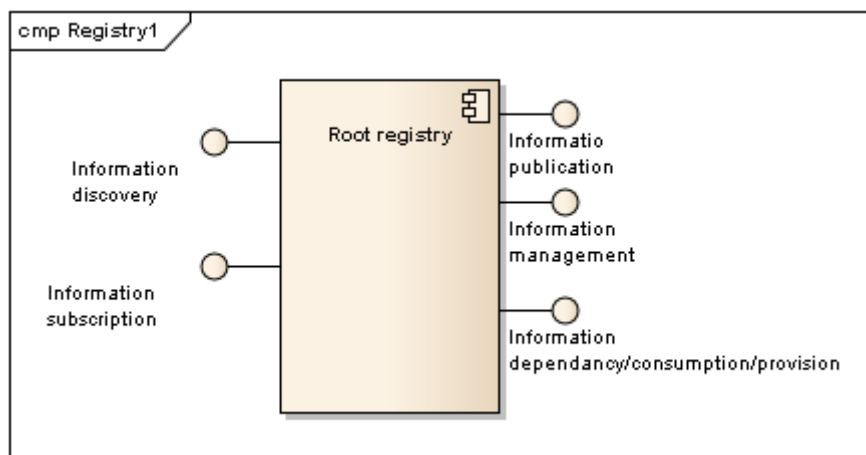


Figure 42 – Self-contained root Registry

##### 2. Root registry with external reference

In this option information is stored at distributed locations, the root registry and some external repositories where the information stored in the root registry refers to. The sharing of information between repositories is done via links. Certain entries in the root registry will contain pointers (i.e. URLs) to specific entries in an external reference repository.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

Information Consumer	The information consumer addresses the root registry for all queries related to service information. The root registry will return information that contains references (i.e. links) to entries in an external reference repository. The consumer will have to go to the external reference system to complete the information set received by the root registry.
Information Provider	The information provider maintains information in two different data stores. There will be policies that ensure that at least certain information is defined in the root registry and that at least the most commonly used values are directly available at the root registry without the need to go to a second repository to complement the information.

Table 18 – Root registry with external reference roles

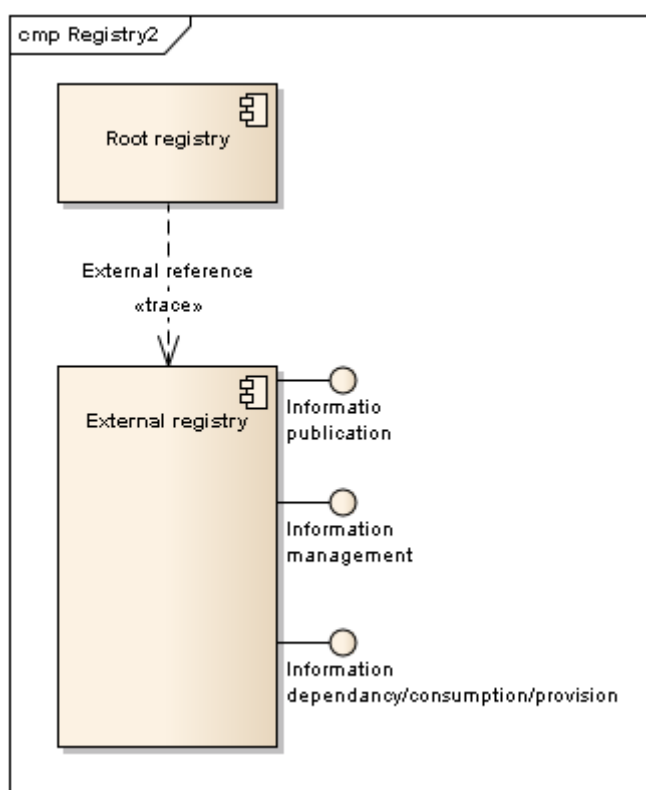


Figure 43 – Root registry with external reference

### 3. Root registry with provider affiliate

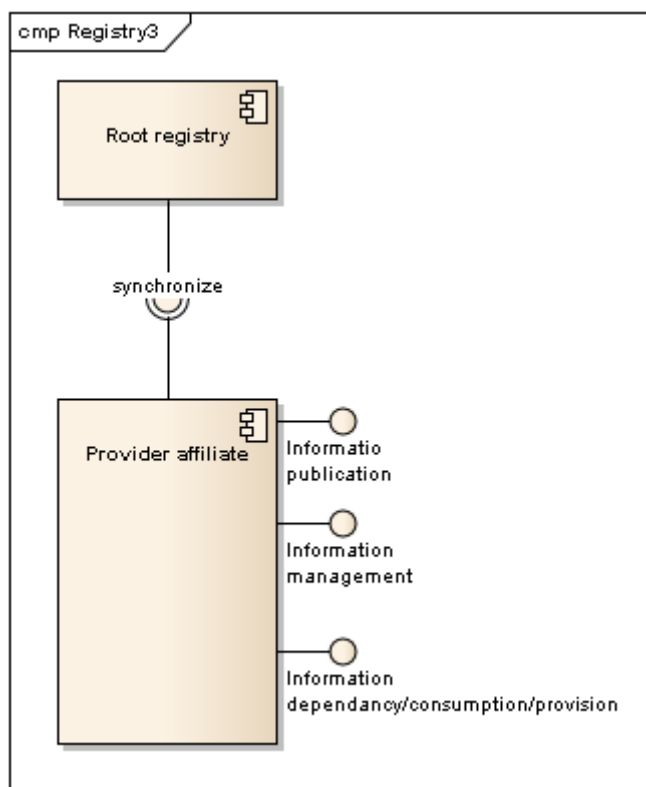
In this option information is stored at two locations, the root registry and an affiliate registry. The affiliate registry is the master or provider of information that is then replicated into the root registry. The information is replicated from the affiliate registry to the root registry via an *Importing Service*. The importing service replicates information by 1) requesting information from the affiliate registry at regular intervals and publishing this to the root registry or 2) using the subscription service to update information in the root registry whenever information changes on the affiliate registry. The scope or subset of information exchanged between registries is determined in the importing service.

It should be taken into account the propagation of access rights to information (e.g. if information is restricted to a group, when replicated to another repository this restriction should remain).

The affiliate registry should comply with certain standards and policies prescribed by the root registry to ensure an efficient management of information (e.g. preventing the creation of duplicate keys to information).

Information Consumer	The main information gate to service related information is the root registry. However, in this case the information consumer has also the option to address the external affiliate registry for the same information.
Information Provider	The provider of information manages the information directly in the affiliate registry. The information provider is prevented from doing updates directly on the root registry (for the information replicated) in order to maintain consistency.

**Table 19 – Root registry with provider affiliate roles**



**Figure 44 – Root registry with provider affiliate**

#### 4. Root registry with consumer affiliate

In this option information is stored at two locations, the root registry and an affiliate registry. The root registry is the master or provider of information that is then replicated into the affiliate registry. The information is replicated from the root registry to the affiliate registry via an *Exporting Service*. The exporting service replicates information by 1) requesting information from the root registry at regular intervals and publishing this to the affiliate registry or 2) using

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

the subscription service to update information in the affiliate registry whenever information changes on the root registry. The scope or subset of information exchanged between registries is determined in the exporting service.

It should be taken into account the propagation of access rights to information (e.g. if information is restricted to a group, when replicated to another repository this restriction should remain).

The affiliate registry should comply with certain standards and policies prescribed by the root registry to ensure an efficient management of information (e.g. preventing the creation of duplicate keys to information).

Information Consumer	The main information gate to service related information is the root registry. However, in this case the information consumer has also the option to address the external affiliated registry for the same information.
Information Provider	The provider of information manages the information directly in the root registry. The information provider is prevented from doing updates directly on the affiliate registry (for the information replicated) in order to maintain consistency.

Table 20 – Root registry with consumer affiliate roles

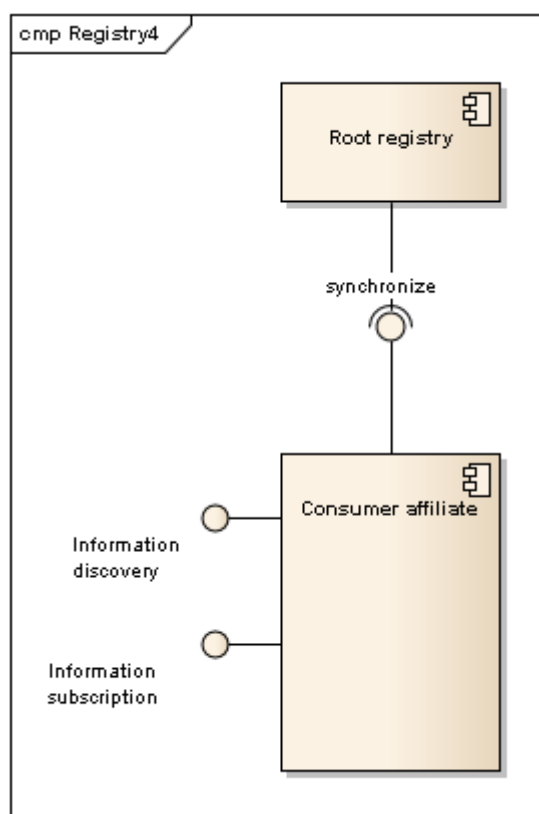


Figure 45 – Root registry with consumer affiliate

### 2.3.5.1.2 Option pros and cons

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

The table below provides pros and cons of registry architectural options. However all options are kept as the final architecture will certainly be a combination of them. Depending on the nature of data stored in the registry, a particular option may be more appropriate. The data sensitivity or sovereignty may also be a criterion to select a particular option where a delegation to an affiliate registry can be put in place.

Option	Pros	Cons
Self-contained root registry	<p>Easy to manage (don't require database synchronization for adding metadata, back-up is centralized).</p> <p>Security policy for the registry itself is centralized (this is more suitable for sensible metadata i.e. security policies).</p> <p>Metadata are not duplicated.</p>	<p>Availability may be limited by the number of possible concurrent access.</p> <p>Require a contingency plan and a contingency location.</p> <p>Not very flexible when some metadata require to be store locally for sovereignty reason.</p>
Root registry with external reference	<p>Metadata are not duplicated.</p> <p>Data storage is more distributed so the global resilience is higher.</p>	<p>Administration task is more complex.</p> <p>Broken references need to be addressed.</p>
Root registry with provider affiliate	<p>Publication traffic is distributed all over the various databases. It is suitable for metadata that need frequent updates.</p> <p>Data storage is distributed so the global resilience is higher.</p>	<p>Metadata are duplicated.</p> <p>Require metadata synchronization between root and affiliate databases.</p>
Root registry with consumer affiliate	<p>Querying traffic is distributed all over the various databases. It is suitable for metadata that need frequent query.</p> <p>Data storage is distributed so the global resilience is higher.</p>	<p>Metadata are duplicated.</p> <p>Require metadata synchronization between root and affiliate databases.</p>

**Table 21 – Registry architectural options Pros and Cons**

### 2.3.5.1.3 SWIM-TI Registry Architectural choice

At the time being, the most appropriate option for realizing the SWIM-TI Registry Functional Block into a SWIM-TI Infrastructure System is:

Root registry with consumer affiliate

**Table 22 – SWIM-TI Registry Architectural choice**

This is specified in SWIM-TI TS (ref. [13]).

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

### 2.3.5.1.4 Registry Policies Management Architectural options

Security policies require to be managed in a single place. The rationale of this choice is the specific high sensitivity of the security related information. Consequently the centralized option of the registry architecture is preferred to manage this kind of information. The other options where information is distributed over many places are not recommended.

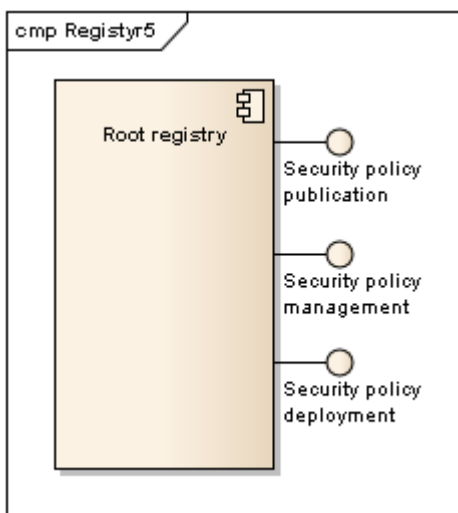


Figure 46 – Registry Architectural options

## 2.3.5.2 Messaging FB Architectural options

### 2.3.5.2.1 OSI standard model

Data exchanges are often modelled according to the OSI standard model (ref [23]).

This model is useful as a framework to structure the standards and technologies of the SWIM-TI Messaging and their interdependencies, their composability and their interactions.

The SWIM-TI Messaging applies to the layers 4 to 7 of the OSI model depicted below:

OSI Model			
	Data unit	Layer	Function
Host Layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Inter-host communication, managing sessions between applications
	Segments	4. Transport	End-to-end connections, reliability and flow control □
Media Layers	Packet/Datagram	3. Network	Path determination and logical addressing □
	Frame	2. Data link	Physical addressing □
	Bit	1. Physical	Media, signal and binary transmission

Table 23 – OSI standard model

Some orthogonal aspects, such as management and security, involve every layer. According to this and in order to achieve the Systems Interconnection, SWIM-TI Messaging deals with the layers [4...7] and the combination of different technologies for each of them enable the achievement of its functional requirements.

For the achievement of its functional requirements, the SWIM-TI Messaging also relies on the services provided by the Layer 3 and has awareness of the standards and technologies related to the Interface with Layer 3.

Layer 3 masks the existence of the Layer 1 and the Layer 2 for the SWIM-TI Messaging. Layer 1 and Layer 2 are therefore not relevant for the SWIM-TI Messaging.

ASSUMPTION	SWIM-TI assumes an IP Network as base upon which SWIM Technical Infrastructure is built/deployed.
ASSUMPTION	SWIM-TI manages layers [4...7] in a standard OSI Model.

### 2.3.5.2.2 IP Network base: design rules

The Systems of Systems (SoS) nature of SWIM requires optimum usage of network resources by following some of the following rules:

- Limit bandwidth usage and adapt communication to available bandwidth.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

127 of 284

- Use compression to minimize bandwidth usage.
- No IP fragmentation and adapt to path MTU when UDP/IP is used.
- Reuse network capabilities such as PIM-SM when multicast is requested for optimum use of the underlying network.
- Favour filtering at source level to avoid unnecessary usage of network resources.

### 2.3.5.2.3 Routing

#### 2.3.5.2.3.1 Overview

Routing can be managed at different OSI layers and with each OSI layer Routing may be supported by multiple technologies. Hence, architectural options for each of them are provided.

Each SWIM-TI SWIM Profile chooses a layer and technology or a combination of multiple layers and multiple technologies for the realization of the Routing.

#### 2.3.5.2.3.2 Constraints

The mapping of the SWIM-TI Routing function onto concrete technology, may introduce constraints as the concrete technology may have specific limitations that do not support all of the SWIM-TI Routing functionality.

#### 2.3.5.2.3.3 Delivery options

##### 2.3.5.2.3.3.1 Overview

Each of the delivery methods can be mapped onto one or more technologies at distinct layers in the OSI model:

Above Layer 3  
Layer 3  
Layer 2

Technology in Layer 3 and Layer 2 are provided by the network level. As Layer 2 is not in scope of the SWIM-TI, Layer 2 is not taken into consideration and no options are provided for Layer 2.

Technology above Layer 3 is typically provided through transport (Layer 4) and through a network overlay (typically Layer 7).

Multiple distinct technologies in distinct layers can be combined.

##### 2.3.5.2.3.3.2 Unicast

###### Above Layer 3:

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) both provide support for Unicast type of delivery for which they rely on IP in Layer 3.

###### Layer 3:

IP provides support for Unicast addressing and routing.

##### 2.3.5.2.3.3.3 Anycast

###### Above Layer 3:

Anycast can be provided through an overlay network (typically Layer 7).

Anycast over the Internet using IP in Layer 3 is problematic (scalability in routers).

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



Anycast over the Internet using an overlay network provides flexibility and scalability.

UDP (Layer 4) provides support for Anycast.

It relies on a configuration of routing in IP in Layer 3

TCP (Layer 4) can provide support for Anycast

It relies on a configuration of routing in IP in Layer 3.

TCP is stateful and Anycast can only be used to find and establish the session whereafter any further communication requires use of Unicast.

Layer 3:

Anycast at Layer 3 can be provided both through IPv4 and IPv6.

In IPv4 there is no explicit notion of Anycast.

The Anycast behavior is a matter of configuration of the IP-routing of Unicast addresses.

In IPv6 the notion of Anycast is made explicit.

Syntactically the Anycast addresses are as in IPv4: Unicast addresses. The Anycast behavior is a matter of configuration of the IP-routing of Anycast addresses

Both in IPv4 and IPv6, support for Anycast relies on the ability for multiple distinct interfaces to have the same IP address. Typically a router will then see multiple paths to the same IP address and will select the one that is at the shortest distance in number of hops. Use of hops to determine the closest instance is not always the most appropriate: for instance from a perspective of latency.

#### 2.3.5.2.3.3.4 Multicast

Above Layer 3:

Multicast can be provided through an overlay network (typically Layer 7).

Multicast over Internet using IP Multicast in Layer 3 is problematic (not supported)

Multicast over Internet using an overlay network provides flexibility and scalability but may be less efficient than IP Multicast in Layer 3

Multicast over Internet using an overlay network combined with using IP Multicast in Layer 3 is possible and allows to use the optimum characteristics of both but this increases complexity.

Reliable multicast.

Reliable multicast protocols typically rely on UDP in Layer 4

UDP (Layer 4) provides support for Multicast.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

129 of 284

It relies on IP Multicast in Layer 3

#### Layer 3:

Multicast at Layer 3 is most commonly implemented using IP multicast, for one-to-many communication over an IP infrastructure in a network. In IP multicast the implementation of the multicast concept occurs at the IP routing level, where routers create optimal distribution paths for datagrams sent to a multicast destination address.

For instance, IP multicast scales to a larger receiver population by not requiring prior knowledge of who or how many receivers there are. Multicast uses network infrastructure efficiently by requiring the source to send a packet only once, even if it needs to be delivered to a large number of receivers. The nodes in the network take care of replicating the packet to reach multiple receivers only when necessary.

- **Multicast ASM**

The original vision for multicast in RFC 1112 supported both one-to-many and many-to-many communication models and has come to be known as Any-Source Multicast (ASM). It should be noted that the many-to-many model introduces complexity within the network (sources discovery is achieved by the network).

ASM is used for information exchanges whose origin can be centralized in some way and to be shared among many SWIM Nodes

- **Multicast SSM**

The Source-Specific Multicast (SSM) model has been developed to only support the one-to-many model distribution with a low-level of complexity within the network. The network does not need to discover the source: it is now under the responsibility of the application.

Both IPv4 and IPv6 support IP Multicast.

#### 2.3.5.2.3.3.5 Broadcast

#### Above Layer 3:

Broadcast can be provided through an overlay network (typically Layer 7).

Broadcast is usually only permitted in a LAN and not authorized in a WAN when using the technology options of Layer 4 and lower Layers.

The reach of a broadcast in an overlay network can be controlled more flexibly than a broadcast in Layer 4 or Layer 3 or Layer 2

UDP (Layer 4) provides support for Broadcast.

It relies on IP Broadcast in Layer 3

#### Layer 3:

IPv4 has the notion of broadcast addresses.

IPv6 does not include the notion of broadcast addresses: it has been superseded by multicast addresses.

#### 2.3.5.2.3.3.6 Layer 3 comparison

The table below presents a comparative between the above mentioned delivery options at Layer 3:

Option	Pros	Cons
--------	------	------

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

130 of 284

Option	Pros	Cons
Unicast	Reliability Extended Security Support Simplicity	Scalability Efficiency
Multicast ASM	Scalability Efficiency	Security Support limited to UDP Reliability Complexity associated to source discovery More vulnerable to DOS attacks
Multicast SSM	Scalability Efficiency No source discovery (simpler than ASM)	Security Support limited to UDP Reliability Must support IGMP v3/MLD v2

Table 24 – Routing Layer 3 Pros and Cons

#### 2.3.5.2.3.4 Organisation

##### 2.3.5.2.3.4.1 Distributed Routing

In a distributed organization form of routing functions, every instance of a MSG FB can find and deliver a message to any other instance of a MSG FB.

##### 2.3.5.2.3.4.2 Centralized Routing

In a centralized organization form of routing functions, every instance of a MSG FB can only deliver a message to any other instance of a MSG FB through a central MSG FB.

##### 2.3.5.2.3.4.3 Federated Routing

In a federated organization form, a group of interconnected centralized routing functions (through interconnected MSG FBs) collaborate to share messages beyond the routing function they provide for their directly connected senders and receivers. In a federation, every individual centralized routing function can forward message from a local sender to federation members making them available for remote receivers.

Such a federated organization of centralized routing functions is named Federated Routing.

In order to establish a Federated Routing in the SWIM-TI, every centralized routing function has to act in accordance with a federation policy document, which specifies exact rules of inter centralized routing function collaboration for one particular federation. Such federation agreements are publicly available in the SWIM registry and so are the rules of routing.

The figure below depicts Federated Routing.

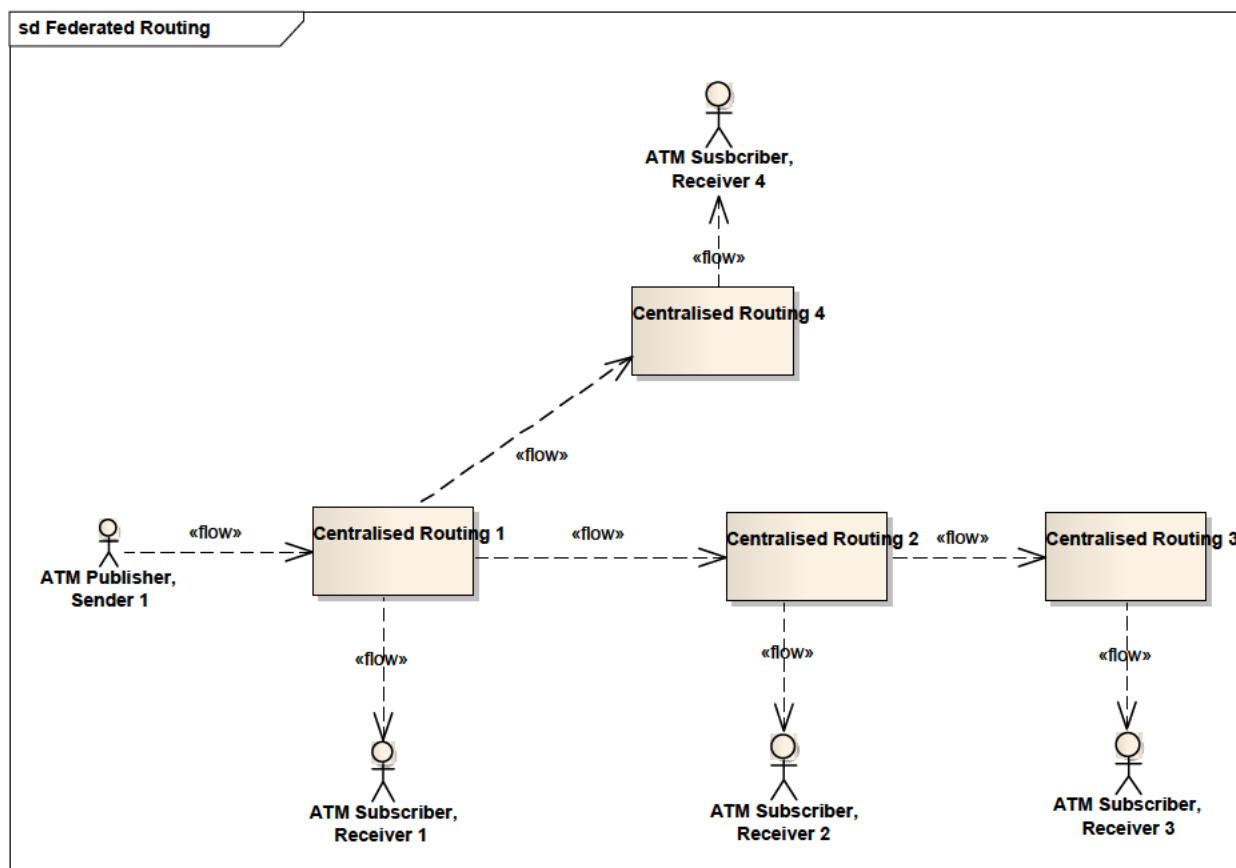


Figure 47 – Federated Routing: example

In the figure above, the flows between centralized routing functions are subject to the routing function. When a centralized routing function receives a message as part of publication process, it delivers it to its local subscribers (if applicable) and also forwards to other Federated Routing members according to routing patterns specified in Federated Routing policy document.

A Federated Routing policy is specified using taxonomy of common purpose condition rules and routing patterns listed above.

The addition of all the centralized routing functions together represents a Federated Routing.

### 2.3.5.2.3.5 AMQP routing

#### 2.3.5.2.3.5.1 AMQP v1.0

##### 2.3.5.2.3.5.1.1 Introduction

AMQP v1.0 is explicitly conceived to support a scalable network of cooperating heterogeneous AMQP intermediaries through which a message is routed from source to destination and the standard explicitly describes the minimal rules that apply to intermediaries.[59][60]

##### 2.3.5.2.3.5.1.2 Openness

AMQP v1.0 imposes almost no restrictions/constraints on the key elements of routing:

- addressing and address space are totally unconstrained.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- collection and distribution of information to support the routing decisions are totally unconstrained. Anything between manual static configuration and dynamical information gathering is possible.
- no routing algorithms are mandated.
- no protocols specific to routing are defined.

### 2.3.5.2.3.5.1.3 Addressing

The AMQP v1.0 protocol provides placeholders for addressing:

- per message addressing: the "to" field and the "reply-to" field

"to" the address of the node the message is destined for:

The to field identifies the node that is the intended destination of the message. On any given transfer this might not be the node at the receiving end of the link.

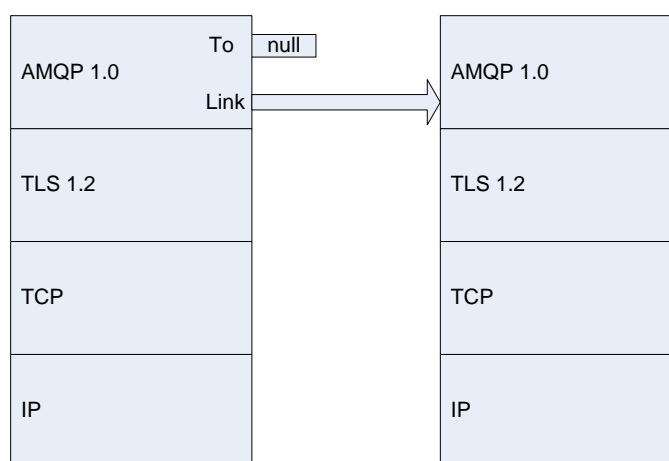
"reply-to" the node to send replies to

The address of the node to send replies to.<sup>42</sup>

- per link addressing: the "address" field for both "source" and "target"

In its simplest form, the AMQP V1.0 nodes of message source and message destination can directly reach each other and do not use routing other than IP routing:

In such case, the "to" and "reply-to" fields are not used and the "service" end-point is hosted at the link "address".

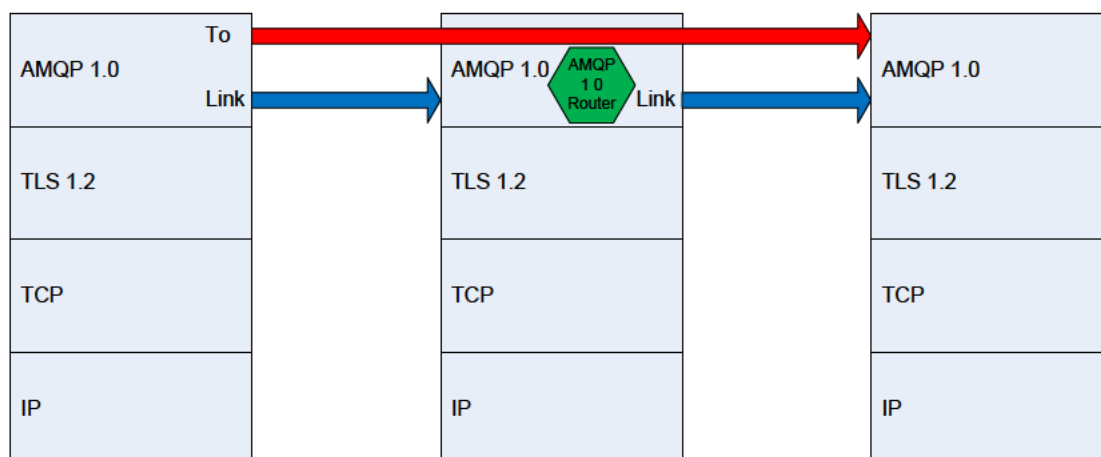


Typically, one such "service" end-point is hosted on a "Message Broker" and the latter uses its own proprietary addressing format.

In more complex forms, where effective routing is required:

<sup>42</sup> The red coloured text is copied as is from the AMQP v1.0 specification.

The “to” and optionally the “reply-to” fields are used to allow the routing to identify the destination of a message.



The red arrow represents end-to-end communication which is routed through one intermediary that contains an AMQP 1.0 Router function. The blue arrows represent point-to-point communication.

Additionally, intermediaries that perform routing functions can use annotations to perform routing decisions.

#### 2.3.5.2.3.5.1.4 Considerations

The openness of AMQP v1.0, allows for any basic (centralized, federated, distributed) organisation form of routing as well as any hybrid combination of these organisation forms. Such hybrid combinations can be composed to virtually any level of complexity and numbers (scalability).

However in the context of more complex organisation forms, this openness requires the explicit definition and agreement of/on a common addressing format and routing semantics and routing related behaviour by the involved parties.

It is assumed that many routing use cases can be satisfied with a single particular well-chosen definition of an addressing format and routing behaviour.

Hence the work at OASIS (currently in draft) on an extension to the AMQP v1.0 standard that targets to provide such a definition that is commonly understood and supported by all AMQP v1.0 components: this extension is usually referenced as “Global Addressing” or “AMQP Addressing”.

Such extension of the AMQP v1.0 standard will not constrain the existing openness of AMQP v1.0 but will standardize a single profile out of the entire solution space for those who find the specification suitable.

#### Broker:

To avoid terminology confusion/ambiguity: a “broker” in the AMQP 1.0 context is an intermediary infrastructure that can host any function and that makes such functions accessible to an AMQP 1.0 peer via the AMQP 1.0 core protocol. Examples of such functions: Queues and Topics but also Routers. The AMQP 1.0 core protocol has no understanding of these or other functions and hence does not limit such functions in any way.

Participants in a communication using AMQP 1.0 can send messages to each other directly without needing a “broker” infrastructure.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Participants in a communication using AMQP 1.0 can also send messages to each other using functionality offered by an intermediate "broker" infrastructure.

Although each "broker" product typically uses its own semantics for Queue and Topic concepts, many "brokers" offer some sort of Queue and Topic function in a highly similar manner.

One typical model is a centralized model with a single "broker" infrastructure.

Participants in a communication using AMQP 1.0 can for instance directly send messages to/receive messages from a Queue or Topic function on the "broker". In a single "broker" model there is no need for a Router function between "brokers".

In other more complex models participants in a communication using AMQP 1.0 cannot reach each other directly. In such case they may for instance rely on a multiple "broker" model that provides Router functionality on one or more "brokers".

The sender delivers a message to a Router function on a "broker" it can directly reach. Note that the message is delivered to a Router function, not to a Queue nor a Topic function. The Router will analyse the To field and decide where to forward the message. This forwarding can be to the Router function on another "broker" which is "closer" to the target or directly to the target - e.g. a Queue function or a Topic function - if that target is directly reachable.

If multiple distinct addressing formats are used then, in order to be able to ensure effective communication, the AMQP 1.0 Router must understand multiple such formats. If all participants would use a single addressing format, then the AMQP 1.0 Router has to understand only one such format.

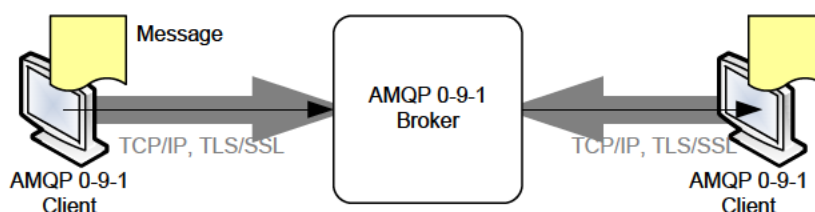
Creation of standardized addressing profile for AMQP 1.0, could simplify the configuration of an AMQP 1.0 Router function but it is not necessary to allow routing between participants in a communication using the AMQP 1.0 core protocol and when relying on complex architecture using "brokers" from distinct vendors.

### 2.3.5.2.3.5.2 AMQP V0.9.1

#### 2.3.5.2.3.5.2.1 Transport Level Considerations

All AMQP 0-9-1 Clients, both message producer and message consumers, have to first establish a TCP/IP connection to the AMQP 0-9-1 Broker. This connection may be protected by TLS/SSL to assure integrity and confidentiality of the transmitted messages.

The AMQP Broker must have a globally reachable IP address. The AMQP clients may be behind NAT and their IP address may change. Unless the Mobile IP is implemented the AMQP clients will have to re-establish the TCP/IP connection whenever the IP address changes. This does not cause any message loss.



Once the TCP/IP connection is established, the AMQP messages can be exchanged in both directions.

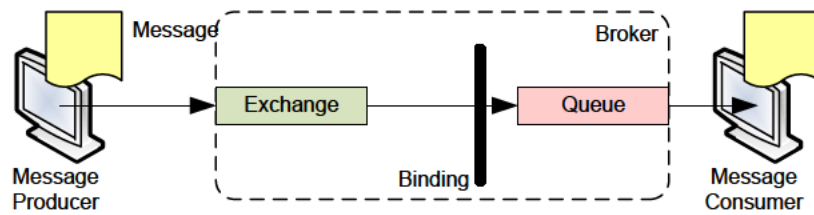
#### 2.3.5.2.3.5.2.2 Broker Level Considerations

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

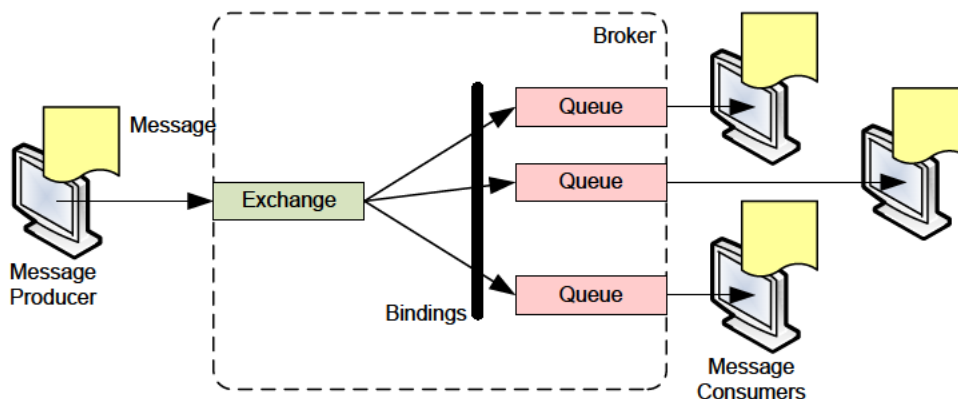
The AMQP 0-9-1 messaging model consists of three basic entities.



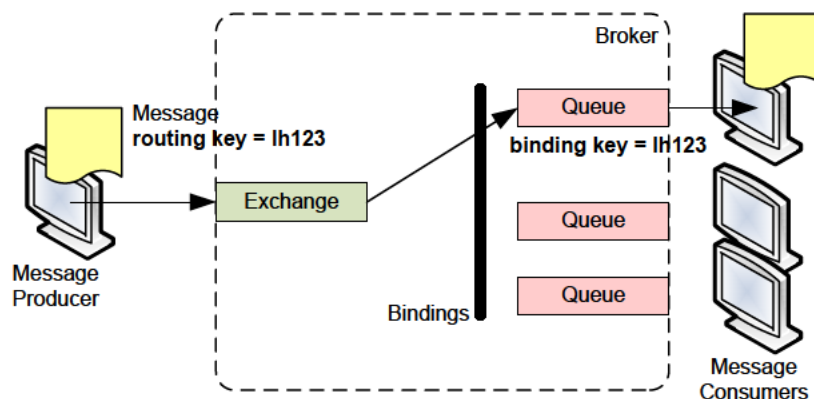
- The *Queue* is a named buffer that holds messages on behalf of a set of consumer applications.
- The *Exchange* is a message routing agent which receives messages and routing information (routing key) and distributes the messages to outbound queues.
- The *Binding* specifies routing arguments that tell the exchange service which messages the queue should get.

The AMQP 0-9-1 specification defines three message exchange types. The exchange types differ only in the algorithm used to determine which queues should receive a message.

The simplest exchange type is a *Fan-out exchange*, which sends each message to every queue bound to the exchange.



A *Direct exchange* sends a message to a queue if the message's routing key is identical to the binding key for the queue. Producers assign each message a target identifier (e.g. aircraft or flight id) and the consumers bind its queue using their assigned identifier (e.g. LH123). Multiple bindings between the same queue and exchange are possible.



founding members



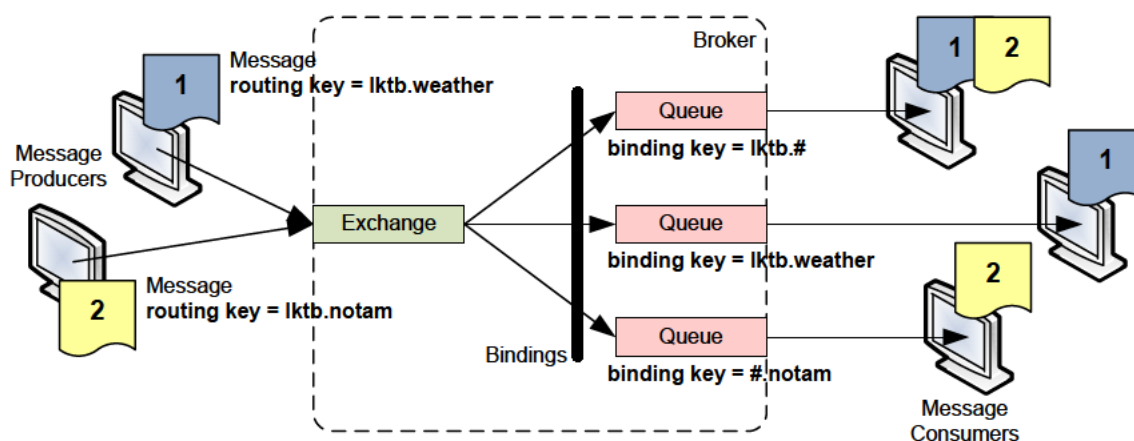
Avenue de Cortenbergh 100 | B -1000 Bruxelles  
 www.sesarju.eu



A *Topic exchange* is similar to a Direct exchange, but uses keys that contain multiple words separated by a "." delimiter. A message producer might create messages with routing keys like `lktb.notam`, `lktb.weather`, `lkpr.notam`, and `lkpr.weather`.

Binding keys for a Topic exchanges can include wildcard characters: a "#" matches one or more words, a "\*" matches a single word. Typical bindings use binding keys like `#.notam` (all NOTAMs), `lktb.#` (all information from LKTb airport), or `lkpr.weather` (all LKPR airport weather items).

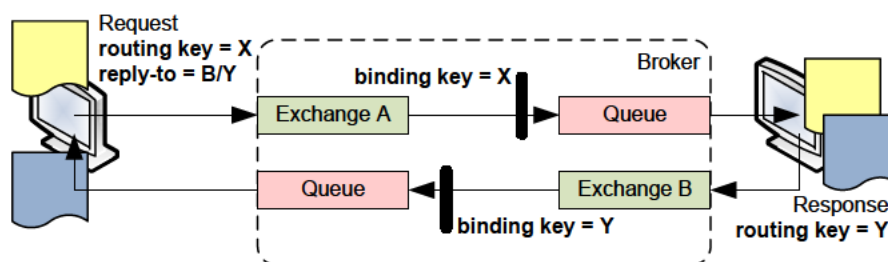
The exchange routes messages to the relevant queue or queues, depending on matches between the routing and binding keys.



### 2.3.5.2.3.5.2.3 Application Level Considerations

The AMQP 0-9-1 contains provisions for more complex message exchange pattern.

- the "reply-to" field may be used to specify routing of the response message in the Request-Response MEP as shown in the following figure.



### 2.3.5.2.3.5.2.4 Broker-to-Broker Considerations

The AMQP 0-9-1 defines the client-to-broker communication only. To create a multi-vendor network two brokers need to be "federated" with each other broker using AMQP 0-9-1 or even other messaging protocol (e.g. AMQP 1.0). A function that federates between the Brokers is not part of the AMQP 0.9.1 standard.

The principle of such function that federates between the Brokers is documented below.

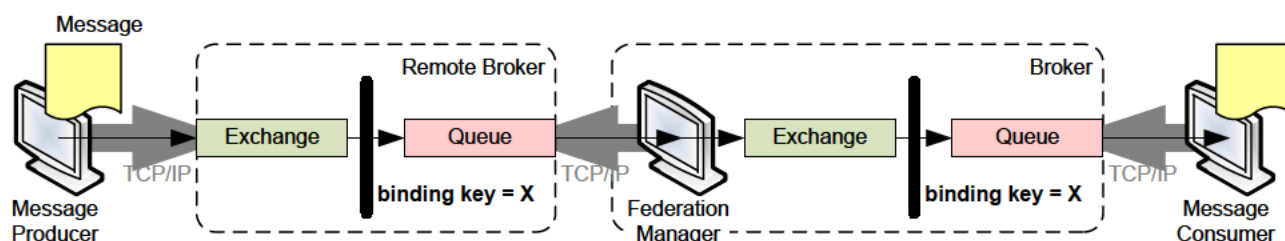
To receive messages from another Remote Broker, the Broker needs to implement a function that federates between the Brokers: this function acts as a standard AMQP 0-9-1 (AMQP 1.0) client towards the Remote Broker and relays messages between the two brokers. The Remote Broker

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Binding can be either static or created dynamically by the function that federates between the Brokers, when the Message Consumer binds its Queue.



Federation Manager depicted in the picture is not part of the AMQP 0.9.1 and it is not part of SWIM-TI TAD. It is the name proposed by 9.19 to group federation related features.

### 2.3.5.2.3.6 DDS routing

DDS routing is described as part the DDSI specification (see [57]). DDS routing is based on a discovery protocol that can be configured generally as a mix of Vector-Distance and Link-State information sharing. The DDSI standard defines a discovery protocol that acts as a network control plane to elaborate routes from publishers to subscribers. Then the publishers are able to route the outgoing messages to the appropriate destinations using multicast or unicast IP address of the subscriber. TCP, UDP or reliable UDP transport layer is used depending on the required quality of service.

The following discovery configurations are possible:

1. UDP/IP multicast discovery(multicast-capable network)

DDSI participants discover each other by means of the 'Simple Participant Discovery Protocol' (SPDP). This protocol is based on periodically sending a message containing the specifics of the participant to a set of known addresses. By default, this is a standardized multicast address (IPv4 239.255.0.1 or IPv6 ff02::ffff:239.255.0.1) that all DDSI implementations listen to. Both data and control traffic use multicast transport. All DDSI participants shall use either IPv4 or IPv6 addresses. When reliable QoS are used a DDSI participant can advertise a unicast address to optimise the retransmission of missed packets when a few subscribers are affected.

2. UDP/IP unicast discovery

In addition to the multicast address DDSI participants advertise a unicast address. The SPDP message contains the unicast and multicast addresses at which the participant can be reached. Typically, each participant has a unique unicast address, which in practice means all participants on a machine have a different UDP/IP port number in their unicast address. Each DDSI participant publishes a list of unicast addresses of other participants it knows. The complete topology is built step by step.

3. TCP/IP discovery

In case of non-capable multicast network, it is possible to configure DDS routing to use TCP/IP connections. It requires that each DDSI participant statically knows a subset of the addresses of other participants. The complete DDS domain topology is built step by step. That mode is fully compliant with Vector-distance model.

Configuration 1) and 2) can be mixed within a single DDS model with some participants using multicast discovery and others using unicast.

The following transport configurations are possible:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

1. UDP/IP multicast

The publisher sends an UDP datagram to a multicast address each of the subscribers is listening to. The routing is delegated to the network layer. Messages may be lost or delivered out of order.

2. UDP/IP reliable multicast

The publisher sends an UDP datagram to a multicast address each of the subscribers is listening to. The routing is delegated to the network layer. A loss detection and retransmission mechanism ensures the message delivery to all the subscribers.

3. UDP/IP unicast

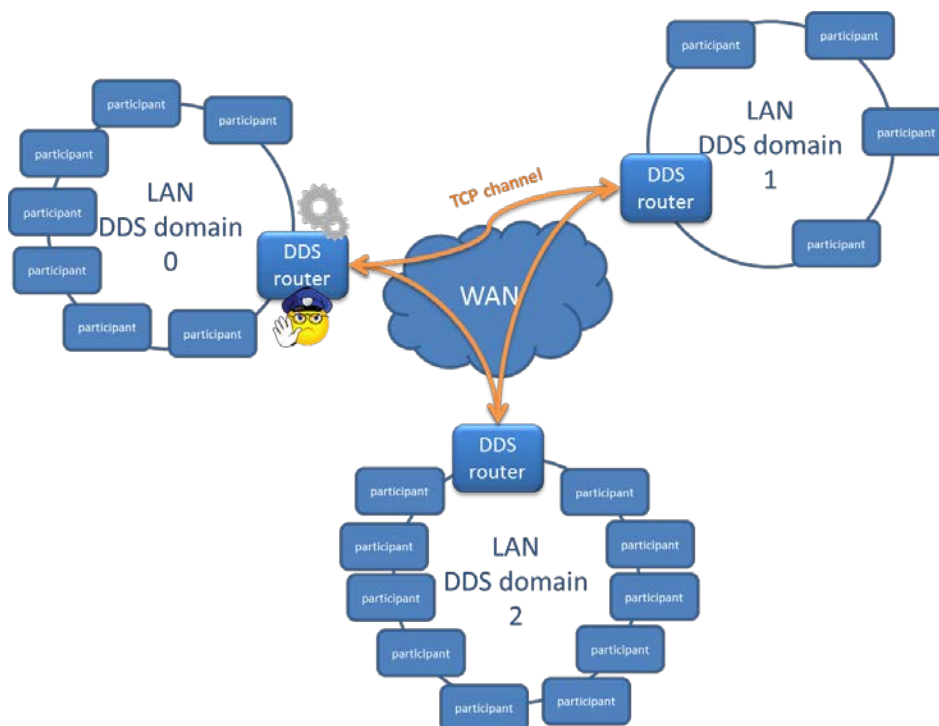
The publisher sends an UDP datagram to a list of unicast addresses, one address per subscriber. The routing is performed by the DDS layer.

In addition to the DDSI standard, some initiatives (e.g. [58]) have been carried out by DDS community to define and deploy DDS routing service over a WAN. However these works are not yet standardized. A new software component called DDS router is introduced at the edge of each DDS domain. A TCP channel is used to interconnect two DDS domains over the WAN.

A DDS router is responsible for traversing appropriately NAT (between local LAN DDS domain and WAN) and firewalls. It plays also the role of:

- Data filtering: a DDS domain can specify which topics are sent or received. Security policy can be enforced to prohibit data leakage outside of the domain whatever is the publishing application.
- Data transformation: DDS types can be automatically transformed into a common type model so that data are understood whatever the specific DDS domain is. The common type model can be defined in a way that it supports type versioning. This kind of data transformation prevents each DDS domain to be upgraded to the new version at exactly the same time.

Figure below summarize the DDS routing service functions.



DDS routing service is fully compatible with the FO-overlay network-architecture described in appendix I of the TAD.

### 2.3.5.2.3.7 SOAP routing: WS-Addressing

By default SOAP uses the address of the transport, such as HTTP, as the address of the end destination of the messages.

This method is typical and suffices for many use cases.

This method does not allow for routing above IP level.

There are however use cases, whereby this typical method does not suffice.

Examples:

- the end destination is not exposed on the transport protocol (e.g. HTTP)
- the source does not know how to directly reach the end destination
- the request is initiated on a HTTP end destination but the destination for the reply is not reachable via HTTP but via another protocol

WS-Addressing allows to decouple the address of the end destination of the message from the address used at transport.

The target address at transport could be that of an intermediate, which will examine the address of the end destination of the message in the WS-Addressing header in the SOAP envelope and determine where to send the message to next using a transport protocol: this can be another intermediate or the end destination.

### 2.3.5.2.3.8 WS-N routing

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

### 2.3.5.2.3.8.1 Considerations

Both WS-BaseNotification and WS-BrokeredNotification are composed by two main parts: subscriptions handling/management and notifications handling (i.e. publication and synchronous and asynchronous receiving). Routing described hereafter does not address “*the routing of subscription handling request/reply messages*” but solely “*Notifications routing*”. It is however anticipated that any technical view and solution for notifications routing rely on properly managed subscriptions: in the topic-based publish/subscribe pattern these represent what the subscribing entities are interested to, so any routing mechanism shall take them into account in order to properly route the notifications concerning a given topic.

The message routing discussed hereafter consists in the routing of WS-N Notifications (“*Notifications routing*”). NotificationBroker, NotificationProducer and NotificationConsumer exchange *Notify Message* defined as follows:

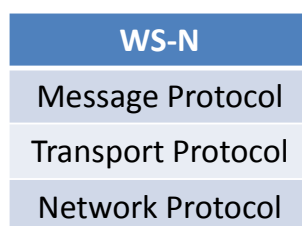
- *wsnt:Notify (Notify Message)*, that contains one or more *wsnt:NotificationMessage*.
- *wsnt:NotificationMessage*, that contains (apart of optional meta-data) one and only one *Message* element containing the actual information to be shared (e.g. Weather data, D-NOTAM, etc. - application-specific Notification content).

Notifications routing may be applied at *wsnt:Notify* level (i.e. the same routing rule is applied for all the contained *wsnt:NotificationMessage*) or at *wsnt:NotificationMessage* level (for the different *wsnt:NotificationMessage* different routing rules are applied).

WS-BaseNotification includes also the concept of *raw Notification* where the application-specific content is exchanged not enveloped in the *Notify Message*. This mainly implies that optional meta-data included in *wsnt:NotificationMessage* (i.e. *wsnt:SubscriptionReference*, *wsnt:Topic* and *wsnt:ProducerReference*) will not be shared.

As already introduced above, the concept of routing can exist at each layer of the stack whereupon WS-N is based. The notion of routing is thus relative to the point of view. WS-Notification family of standards are independent of the underlying stack of protocols (e.g. SOAP over HTTP, SOAP over JMS, SOAP over AMQP) used to distribute notifications.

In the figure below an overview of the layers composing a generic WS-N protocols stack is provided:



**Figure 48 – WS-N stack of protocols**

WS-N layer includes WS-BaseNotification and WS-BrokeredNotification specifications together with any domain-specific extensions (e.g. Policies, etc.) of the standards.

Message Protocol layer represents the layer at which WS-N “Messages” (e.g. *wsnt:NotificationMessage*, *wsnt:Notify*, application-specific content, etc.) are encoded according to a given protocol (i.e. SOAP).

Transport Protocol layer represents the layer at which encoded WS-N “Messages” are exchanged between entities using specific transport (e.g. HTTP, AMQP, DDS).

Network Protocol layer represents the IP layer used by the upper layer to exchange data over the network (e.g. TCP/IP, UDP)<sup>43</sup>

The first point of view to be analysed when describing WS-N routing is the WS-N layer where the description of notifications routing technical view are underlying transport independent. This point of view is introduced in §2.3.5.2.3.8.2. Based on the needs and considerations applying to this layer, in §2.3.5.2.3.8.4 protocols stack layered views are provided in order to introduce different possible stacks WS-N may be based upon. For each stack of protocols, it is then briefly introduced the routing at each layer of the stack and how each layer may contribute to the realization of the routing needs at WS-N layer (§2.3.5.2.3.8.2).

#### 2.3.5.2.3.8.2 WS-N layer point of view

WS-BaseNotification specifies how at WS-N layer the notifications are routed from the *NotificationProducer* to the *NotificationConsumer*. At subscription time the *NotificationConsumer* provides the *NotificationProducer* with the *ConsumerReference* representing the endpoint the *NotificationProducer* shall route notifications. Depending on the use of pull or push messaging, the endpoint refers to a *PullPoint* or to consumer's *NotificationConsumer* interfaces respectively. This is transparent to the *NotificationProducer*.

WS-N *ConsumerReference* is specified as an xml element of WS-Addressing `wsa:EndpointReferenceType` type. This requires and enables WS-Addressing routing at message protocol layer.

Summarizing, in case WS-BaseNotification is used, it is not needed to detail additional routing needs and capabilities at WS-N layer. The notifications routing between *NotificationProducer* and *NotificationConsumer* specified in the standard is complete (from WS-N layer point of view and the anticipated needs) and it relies on the routing capabilities at the layers below WS-N, i.e. WS-Addressing, transport and network protocol layers. Specific scenarios, use cases and constraints may provide routing needs that may be covered by lower layers such as transport protocol.

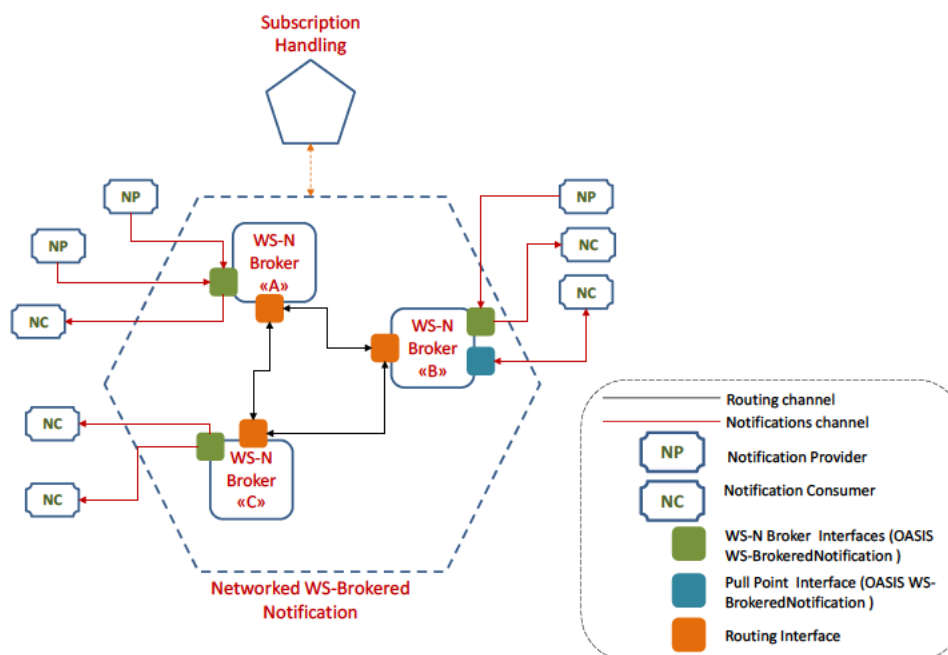
According to actions (e.g. subscription) performed on Subscription Handling the notifications are routed from the publisher to the interested Subscriber. Those entities rely respectively on *NotificationProducer* and *NotificationConsumer* to interact with the Networked WS-Brokered Notification. The latter represents one or more *NotificationBroker(s)* capable to route published notifications to the interested consumers. For simplicity<sup>44</sup>, it is assumed that a given *NotificationProducer/NotificationConsumer* relies on a single Notification Broker to publish/receive notifications: the *NotificationBroker* used by the *NotificationProducer* and the *NotificationBroker* used by the *NotificationConsumer* represent their "access point" to the Networked WS-Brokered Notification<sup>45</sup>

In the figure below, the networked WS-Brokered Notification is exploded depicting several WS-N brokers together with Notifications Producers and Notification Consumers using them.

<sup>43</sup> Same of the protocols provided above as examples for each layer of the stack (e.g. HTTP), in a wider view could be considered as belonging to one or more layer. For instance, HTTP protocol could be considered also as a message protocol. The terminology used in this context is based on the need to clearly distinguish the different layers WS-N is based upon.

<sup>44</sup> Depending on how the Topic namespace is concretely instantiated over the Networked WS-Brokered Notification, a *NotificationProducer/NotificationConsumer* may directly interact with different *Notification Brokers*. For instance, only *Notification Broker "WSN-B1"* may be able to route notifications on "Topic A" whereas only *Notification Broker "WSN-B2"* may be able to route notifications on "Topic B".

<sup>45</sup> It is important to clarify that in some cases/deployment the network of WS-N brokers may consist of a single broker serving multiple *NotificationProducers* and *NotificationConsumers*. In that case, the existence of the broker introduces decoupling between *NotificationProducers* and *NotificationConsumers* and similar considerations provided above for notifications routing in the case of WS-BaseNotification apply also in this case.



**Figure 49 – WS-BrokeredNotification Notifications and Routing channels**

Green rectangles, conformant to OASIS WS-N interfaces (WSDLs), allow Notification Provider (NP) and Notification Consumer (NC) respectively to publish and receive notifications. Blue rectangles, conformant to OASIS WS-N interfaces (WSDLs), allow NC to synchronously pull available notifications. Last but not least, amber rectangles allow the propagation of subscriptions, topic creation and notifications across the brokers. OASIS WS-N specifications do not mandate how routing between brokers has to be realized. Furthermore there are no standard based and (C)OTS vendor independent solutions implementing routing channels between WS-N brokers. WS-N and in general the technologies composing the possible stack of protocols whereupon WS-N is based, do not prevent to choose one routing mechanisms from a community of interest and to apply it for the routing at network of WS-N brokers level. This openness allows to design routing mechanisms and rules but, on the other side, it is reasonable to assume that they will not be provided out-of-box by currently available (C)OTS.

In which cases a network of WS-N brokers may introduce benefits? A number of use cases may be identified according to different needs: performances, scalability, partitioning (at different levels), business/stakeholder constraints of governance and operation of brokers, etc.

Some form of proxying of topics by creating local duplicates combined with forwarding of publications on topics at WS-N level in a network of WS-N brokers, can bring benefits if there are many NotificationConsumers at a particular geographical location that is constraint (e.g. in cost, in bandwidth) in its connectivity with the geographic location of the NotificationProducer. In such case republishing the published notifications only once from the NotificationProducer's local broker to the local broker of the NotificationConsumers and having the duplication performed inside the broker of the local consumers, can hugely increase scalability and performance of the entire network (with regarding to the case where no intermediaries exist between the NotificationProducer and NotificationConsumers).

An effective routing protocol between WS-N brokers should minimize the propagation of the information by avoiding to share information that are not strictly required by a given broker to properly serve its attached NPs and NCs. At the same time the routing protocol should maximize the overall scalability (time and space) without breaking the end-to-end security between NotificationProducers and NotificationConsumers.

Possible drawbacks or specific cases above approach should be able to handle are the following:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

1. When the Topic Namespace is more structured and/or the adopted Topic Expression (see WS-Topics) includes hierarchy, it could be complex to evaluate what information has to be propagated to whom. In some cases it could be more “easy” just to propagate all the information to all the networked brokers (or a sub-set of them selected according to such rule).
2. Topic Namespace changes dynamically (e.g. new topics are added).
3. When NCs provide filters on Topic expression at subscription phase (“I’m interested in all the child topics of the root topic “Troot””), depending on if the Topic Namespace changes dynamically or not, it could be complicated to select only the “relevant” information to be propagated.
4. NCs may be dynamically added to a specific broker or the existing one may subscribe to other topics.

Summarizing, it is required to specify a routing protocol able to dynamically discover who is interested to what and to propagate accordingly all the required information. Possible pros and cons are the following:

- Pros:
  - effective use of resources (e.g. network).
- Cons:
  - May be result in a complex routing mechanism if content based filters have to be evaluated. A possible solution could be to propagate the notifications and to apply the filters at destination side (the broker serving the NotificationConsumer applies the filter).
  - May result in a complex routing mechanism if it is expected to adapt to all the aspects that may evolve dynamically.

A different approach could be to propagate all the information (topic created, subscriptions and notifications, etc.) to all the brokers demanding to them to discard/store those information according to attached NCs (active or planned subscriptions). Of course in this case key cons is the inefficient use of network resources.

A possible variant of above approaches (that could be combined) could be not to propagate/route notifications when they are produced by the NotificationProducers but caching them at NotificationProducer’s local broker side and then (periodically or according to other rules) propagate them to the NotificationConsumer’s local broker. The latter may cache them and notify attached NotificationConsumer according to such rules.

Furthermore, due to such hierarchy in topic namespaces or geographical partitioning, the WS-N brokers may be not all organized as peers but on a hierarchical architecture. This approach may be combined with the other introduced above in order to cover specific needs or to mitigate some cons.

Independently of what (or a combination of above) approach is chosen it is required to discover available WS-N brokers (peers). Depending on the dynamic evolution of the networked brokers (one can be added, another could be removed) the discovery may be:

- Handled at configuration time of each broker, or
- Dynamically at run-time.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



### 2.3.5.2.3.8.3 Security Considerations

As for all routing mechanisms (due to the existence of one or more intermediaries) security of exchanged information has to be taken into account when specifying a routing protocol. WS-Notification specifications adopt WS-Addressing that can be used to attach “security properties” to the exchanged messages. To complement that, message level security mechanisms (e.g. encryption) can be then applied to the notification payload. This will support NP-NC end to end security. Additional security controls may be applied between NP/NC and the Broker: in this case transport level security mechanisms (WS-N operations over SOAP over HTTPs) may be enough.

The routing protocol shall be able to propagate notifications without violating end to end security between NPs-NCs. Furthermore authenticity, confidentiality, integrity and availability at routing protocol peers level are required. Only trusted brokers can join the protocol and only intended recipient brokers are allowed to consumer information being propagated.

Summarizing:

- NP-NC End to End security: message level security handled by NP and NC. The routing protocol shall be able to propagate secured notifications without violate security.
- NP-Broker and NC-Broker security: transport level security.
- Routing protocol security: depending on the specific technology message or transport (or a combination of both) level can be applied.

### 2.3.5.2.3.8.4 Protocols stack layered view

In §2.3.5.2.3.8.2 the routing and possible information to be propagate from WS-N point of view have been introduced. In this paragraph possible stacks of protocols are introduced identifying routing capabilities at each layer of the stack. The analysis is then specialized depending on the channels (routing and notifications channels introduced in §2.3.5.2.3.8.2) where such stack is adopted. As anticipated in §2.3.5.2.3.8.2, the subscription handling/management channels are not analysed in this context.

OASIS WS-N family of standards, are transport independent and delivered using formal languages: WSDL (only Port Types) and XSDs. This openness is constrained by formal artefacts from the WS-N standards and by the use of SOAP message protocol in second layer of the stack independently of the transport and network protocols used. This allows standard based routing relying on WS-Addressing.

Possible SOAP based WS-N stacks with specific Transport and Network protocols, are reported in the figure below.

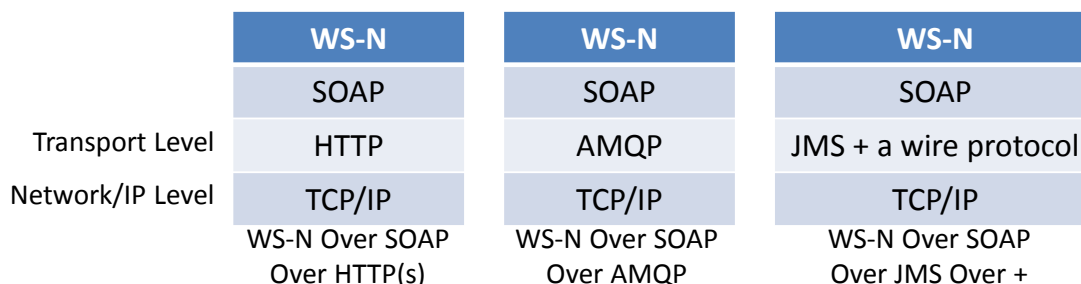


Figure 50<sup>46</sup>

<sup>46</sup> JMS is a standard API that is typically associated with a wire protocol such as OpenWire, T3, T3S and AMQP 1.0.

From the transport layer point of view, only AMQP (refer to §2.3.5.2.3.5.1) and JMS (in combination with a wire protocol) based stacks offer routing mechanisms. Even if some of the available wire protocols for JMS are based on open standards there are no standard mapping between JMS and any of the available wire protocols (there are on-going activities at OASIS on the JMS over AMQP v1.0 mapping).

Table below summarizes this analysis.

	Message Layer	Transport Layer	Network Layer
WS-N over SOAP over HTTP	WS-Addressing	No	IP routing.
WS-N over SOAP over AMQP	WS-Addressing	Yes	IP routing.

**Table 25**

The different options have to be evaluated for each of the notifications channels and according to the WS-N level routing needs introduced in §2.3.5.2.3.8.2. In principle all the stacks introduced above may be adopted in one or more of the identified channels. Two main constraints from WS-N standard exist: *NotificationConsumer::Notify* , *PullPoint::GetMessages* and *PullPoint::Notify* rely on WS-Addressing (*WS-N ConsumerReference*). This implies that on those channels SOAP based WS-N stacks have to be used.

Furthermore, it seems to be suffice to have SOAP over HTTP stack for the request/reply channels *PullPoint::GetMessages* and *NotificationBroker::GetCurrentMessage*. This because it is assumed that there will be no specific routing needs for those channels. In case specific routing needs are identified also other stacks (e.g. SOAP over AMQP) may be used on those channels.

On the other hand, SOAP over AMQP stack could be used in both *NotificationBroker::Notify* and *NotificationConsumer::Notify* channels to decouple NP/NC from the NotificationBroker and relying on SOAP+AMQP routing capabilities. Furthermore, an additional use case for using this stack on *NotificationConsumer::Notify* channel is based on firewall issues experimented by 14.2.9 when asynchronous notifications occur over (SOAP over) HTTP requests (see also activity IT3.1-01).

For what concerns the “routing channels” (communication between WS-N brokers) introduced in §2.3.5.2, independently of the routing rules at WS-N layer, these inter-broker channels may be based on one of the stacks introduced above including the ones non-SOAP based. However, in such hierarchical architecture of the network of broker, some SOAP layer routing capabilities (i.e. WS-Addressing) may be useful.

The constraints and the requirements on the “routing channels” are really different with respect to “notifications channels” discussed above. In particular some aspects as scalability are really key at this level. Inter-broker channels should be realized using stacks of protocols that provide: (i) good scalability, (ii) efficient delivery and discovery and (iii) rich set of routing capabilities at each of layer of the stack.

According to this analysis, in the figure below the stacks of protocols fitting the purpose are provided.

	WS-N	WS-N	WS-N	WS-N
	SOAP	SOAP	SOAP	SOAP
Transport Level	AMQP	DDS	DDS	DDS
Network/IP Level	TCP/IP	TCP/IP	UDP Unicast	UDP Multicast
	WS-N Over SOAP Over AMQP	WS-N Over SOAP Over DDS over TCP	WS-N Over SOAP Over DDS over UDP unicast	WS-N Over SOAP Over DDS over UDP multicast

**Figure 51**

SOAP over AMQP stack has been already introduced above. SOAP over DDS stacks provide routing capabilities at both transport and network level as described in §2.3.5.2.3.6.

The selection of one of those stacks (that in such solution may be composed) shall be based on detailed design and design decisions of the solution implementing WS-N routing needs described in §2.3.5.2.3.8.2.

For all the stacks discussed in this paragraph it is important to notice that only SOAP over HTTP and SOAP over JMS bindings have been standardized.

From the above analysis it becomes apparent that for all the notifications and inter-brokers routing channels the only complete standardized stacks are those from the WS-N over SOAP over HTTP family. If due to well documented reasons it could be required to evaluate additional options (by relaxing the constraint on the use of complete standardized stacks) all the stacks introduced above should be evaluated in detail and then complemented with SESAR specifications. The aim is on one hand to fill the gap of missing complete standardized stacks and on the other hand to properly compose standard technologies to have standard based WS-N protocols stack. If this will be accepted, independently of the channels (notifications or inter-brokers routing) several constraints introduced above should be re-evaluated.

### 2.3.5.2.3.9 Other considerations

#### 2.3.5.2.3.9.1 Information sharing

##### 2.3.5.2.3.9.1.1 Overview

The routing function requires information to make appropriate decisions.

There are three basic options to obtain this information:

- 1) Fixed local configuration
- 2) Dynamic information sharing in the routing function
- 3) Interaction between the routing function and the run-time registry

Hybrid constructions are possible whereby the routing function draws some information from a Run-time Registry and other information from other sources such as a fixed local configuration and/or dynamic information sharing in the routing function

##### 2.3.5.2.3.9.1.2 Dynamic information sharing in the routing function: Vector-distance and Link-state

A commonly used manner to categorise routing is the manner of how the information on the routing topology of a distributed system is shared.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

In a Vector-distance model, each instantiation of a routing function will share its own knowledge on the routing topology with its neighbours that are within a certain distance (hops).

A receiving instantiation of a routing function will use this information to adapt/improve its own knowledge on the routing topology which it will share in turn with its neighbours.

When each instantiation of a routing function does that, all information of each instantiation of a routing function will reach each other instantiation of a routing function.

In a Link-state model, each instantiation of a routing function will share its knowledge on its neighbours with every other instantiation of a routing function. A receiving instantiation of a routing function will use this information to construct a complete view on the routing topology.

The Vector-distance model has various drawbacks such as slow convergence and instability (loops).

The Link-state model converges significantly faster than the Vector-distance model, generates much more traffic and requires much more computing resources.

Both Vector-distance model and Link-state model do not scale well, hence other models are used for larger routing topologies.

#### 2.3.5.2.3.9.1.3 Run-time registry

Information for use by the routing function can be shared through a Run-time Registry.

The messaging can, for instance, query Run-time Registry for information relevant for making a routing decision using criteria/conditions. The Run-time Registry answers are used by the messaging which does the routing accordingly. There is a typical SOA Run-Time Registry use case: "service lookup". The aim is to retrieve the service endpoint of such service according to such parameters.

The role of a Run-time Registry can go further than that of "service lookup" in supporting routing decisions. However gathering information in a Run-time Registry in a dynamic environment and keeping this information relevant can become very challenging.

#### 2.3.5.2.3.9.2 Addressing

An addressing scheme determines the expression of the destination where a message is to be delivered.

Addressing schemes vary and are typically meant to support a particular set of requirements that routing needs to be able to deal with:

- scalability
- mobility
- reliability
- performance
- multi-homing

Typically a single addressing scheme will not be able to satisfy all such requirements: it will enable one or more requirements but may inhibit another.

Typically and in its simplest form, addressing schemes are categorised in hierarchical and flat addressing schemes.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

In a hierarchical addressing scheme:

- scalability is well supported: with limited aggregated knowledge it is possible to select a path towards to the destination of a message. Because an explicit path must not be known for each possible destination, hierarchical addressing schemes are suitable for high scalability.
- mobility is not well supported: moving in an hierarchical addressing scheme requires a change of address.
- reliability: can be supported through multiple routes but not necessarily within the required latency.
- performance: can be supported through aggregation of multiple paths.
- multi-homing: is typically a problem in a hierarchical addressing scheme.

Typically a flat addressing scheme, will exhibit the opposite pros and cons compared to a hierarchical addressing scheme.

- scalability is very limited: a specific path for every possible destination that needs to be reachable, needs to be known, hence such scheme is not scalable.
- mobility is well supported: moving in a flat addressing scheme, does not require a change of address
- reliability is well supported : through aggregation of multiple paths and with almost immediate effect
- performance: can be supported through aggregation of multiple paths.
- multi-homing is well supported

Above simple addressing schemes may be combined with various techniques to compensate for their weaknesses and extend their scope. Example: the creation of an extra level of abstraction through naming. A naming layer can abstract from the changes in a hierarchical addressing scheme due to mobility.

Particular difficulties arise when addressing schemes are used to overload different semantics in the same address. E.g. an address serves for identity, naming and routing.

#### 2.3.5.2.3.9.3 Algorithms: static versus dynamic

The level of freshness of the information that is required to make routing decisions, can vary widely depending on the context.

On one side, it can be highly relevant to know the dynamics of a network such as the amount of traffic, the detection of failures and the knowledge of ad-hoc configuration changes in order to adapt the decision making process.

In other cases, it can be sufficient to update the information to make routing decisions, at a low frequency and in a synchronized manner between all involved parties.

#### 2.3.5.2.3.9.4 Datagram versus virtual circuit

A datagram is entirely self-standing and contains all the addressing information.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

For each datagram from the same source to the same destination, the routing decision at an intermediate node can be different.

Pros: unavailability, overload can be handled dynamically.

Cons: datagrams can arrive at the destination out of order.

Virtual circuit:

A fixed route between source and destination is set up through all intermediaries before any information is effectively sent. The messages are no longer entirely self-standing but contain a short-cut to identify the pre-established path, which will be understood and used by all intermediaries.

Pros: in order arrival, less time needed for the routing decision on each message.

Cons: less flexibility to handle overload, unavailability, etc. in a dynamic manner.

### 2.3.5.2.3.9.5 Mobile IP

At the time of writing, there could be confusion in the matter of mobility and Aircraft.

It is assumed that Ground will never initiate and Air-Ground communication. The Aircraft will always initiate the Air-Ground communication. Following such initiation within the AMQP Session, bidirectional communication can take place.

It is not assumed that the IP-address of the Aircraft will remain fixed.

It is assumed that the IP-address assigned to the Aircraft at a given point, will enable to access any IP-address on the Ground network with a world-wide scope. If the IP-address assigned to the Aircraft changes, then the Aircraft will re-initiate the AMQP connection and the message flow will continue.

Hence, based on these assumptions there is no need in the Purple Profile for specific support for maintaining reachability of the Aircraft in case of moving network attachments.

Notwithstanding above assumptions, it is useful to take mobility into consideration from a functional point of view as well as from a technical point of view and to provide at least the structure for capturing such requirements. From technical view, Mobile IP provides the means for a mobile device that attaches to varying networks during its moves, and therefore potentially using changing IP-addresses, to remain nevertheless reachable on a fixed IP address.

## 2.3.5.2.4 Distribution

### 2.3.5.2.4.1 Mediator

In the functional descriptions related to the distribution function, the term Mediator is used. The Mediator is required for various MEPs to ensure the decoupling of time, space and/or synchronization.

The functions offered by the Mediator can be found in the literature and/or in implementations under various names such as broker, event service and message bus or data bus.

As the semantics of such terminology is varying to a large extent, the term Mediator or Intermediary will be used preferably and combined with a description of the functions it offers in the context of its use.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

150 of 284

#### 2.3.5.2.4.2 Message queuing

The technology related to Message queuing, targets to provide support for the entire domain of asynchronous messaging.

Only one family of technology that supports message queuing has been retained: AMQP.

Message queuing has been popular and successful. However it was mostly based on proprietary wire protocols and has been suffering from that in a context of heterogeneous technology.

AMQP is an open standardized wire protocol that is particularly suited for reliable message queuing

From that family two distinct specifications are included:

AMQP 0.9.1

AMQP 1.0

#### 2.3.5.2.4.3 Publish/Subscribe MEPs and Observer MEPs

Three families of technology have been retained to support the Publish/Subscribe MEPs and the Observer MEPs. Each of these technologies has its specificities, which make them more suitable/appropriate for one context or another.

- ISO/IEC and OASIS AMQP,
- OMG DDS (see also chapter 2.3.5.2.4.6),
- OASIS Web Services Notification.

#### 2.3.5.2.4.4 Fanout

A specific subscription scheme exists for AMQP 0.9.1 that sends all messages to all subscribers:

This technique is called Fanout Exchange and only exists in the AMQP parlance before the OASIS AMQP 1.0 standard

A Fanout Exchange delivers a message unconditionally to all queues connected to the Fanout Exchange, hence to all subscribers associated with the connected queues.

This should be considered as a particular case of Topic-based subscription scheme:

- Each Fanout Exchange gets a unique name.
- The publisher decides to which Fanout Exchange a message is published.
- The subscriber decides to which Fanout Exchange a queue is connected.
- At a higher level of abstraction the name of the Fanout Exchange should hence be considered to be a Topic. There is no subscription to a queue: queues are used to transport the messages. This should not be confused with the notion of Topic Exchange which is also defined in the same specific technology.

#### 2.3.5.2.4.5 Push/Pull

Noteworthy aspects that depend on the technology used to implement the Push or Pull:

- Scalability limitations:
  - o Immediate closure of a connection keeps resource consumption low but is costly to recurrently set up,

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- o Keeping a connection open for reuse reduces overhead and latency but increases resource consumption.
- Security limitations:
  - o Security policies that may disallow particular connection patterns by lower level protocols,
  - o Risk of voluntary and involuntary spamming and DoS.

#### 2.3.5.2.4.6 Data Centric Publish Subscribe

The Data-Centric Publish-Subscribe (DCPS) is targeted toward the efficient delivery of the proper information to the proper recipients<sup>47</sup>. It provides the application with a data-centric information model and is responsible for controlling the lower level layer of the DDS infrastructure targeted toward the efficient and reliable delivery of the information to its intended recipients

In DCPS, data can be accessed in 2 ways:

- Wait based (synchronous calls)
- Listener based (asynchronous callbacks)

DCPS has sophisticated filtering e.g. Topic, Content-FilteredTopic or MultiTopic.

DCPS data model captures relationship between data using keys for example embedded in the data itself. Within the definition of the Topic Type, one or more data elements can be chosen to be a "Key" for the type. The DDS middleware will use the Key to sort incoming data. By specifying one data element to be a Key, an application can then retrieve data from DDS that matches a specific key. The use of keys also supports scalability.

DCPS is configurable via many QoS policies.

#### 2.3.5.2.4.7 Web Services

W3C states in <http://www.w3.org/TR/ws-arch/> chapter 3.1.3 Relationship to the World Wide Web and REST Architectures:

We can identify two major classes of Web services:

- REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations; and,
- arbitrary Web services, in which the service may expose an arbitrary set of operations.

Both classes are suitable for synchronous messaging and both are included.

The class of arbitrary Web services is based on the use of SOAP. SOAP 1.2 allows to be used in a manner consistent with REST. The use of SOAP is explicitly targeted to support a use that is not consistent with REST.

#### 2.3.5.2.5 Message filtering

##### 2.3.5.2.5.1 Filtering in a Publish/Subscribe environment using a queue-based Mediator

---

<sup>47</sup> See [DDS \(Wikipedia\)](#) and references therein for an introductory explanation of DDS.



Concretely and as an example, the Message Filtering function can deal with the process of selecting messages for reception and processing in a Publish/Subscribe environment that uses a queue-based messaging system as Mediator.

In a Publish/Subscribe environment, there are two common forms of selecting the filtering criteria: topic-based and content-based.

In a topic-based system, messages are published to "topics". Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe, and all subscribers to a topic will receive the same messages. The publisher is responsible for defining the classes of messages to which subscribers can subscribe.

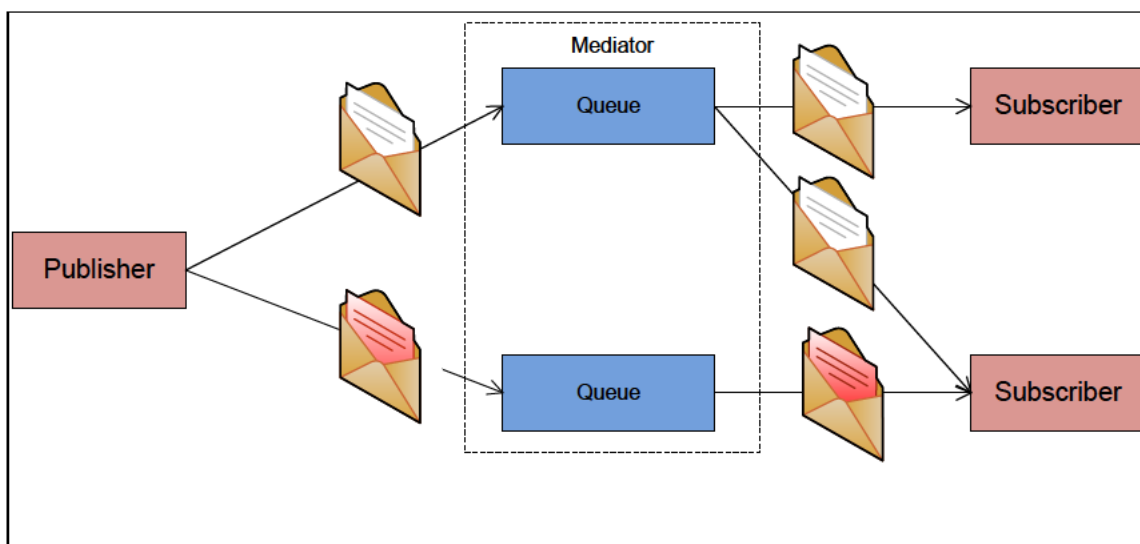


Figure 52 – Topic-based Filtering

In a content-based system, messages are only delivered to a subscriber if the attributes or content of those messages match constraints defined by the subscriber. The subscriber is responsible for classifying the messages. Ideally, none of the SWIM-TI functions should get into the payload of the message, so this content-based filtering should be done via message attributes or metadata.

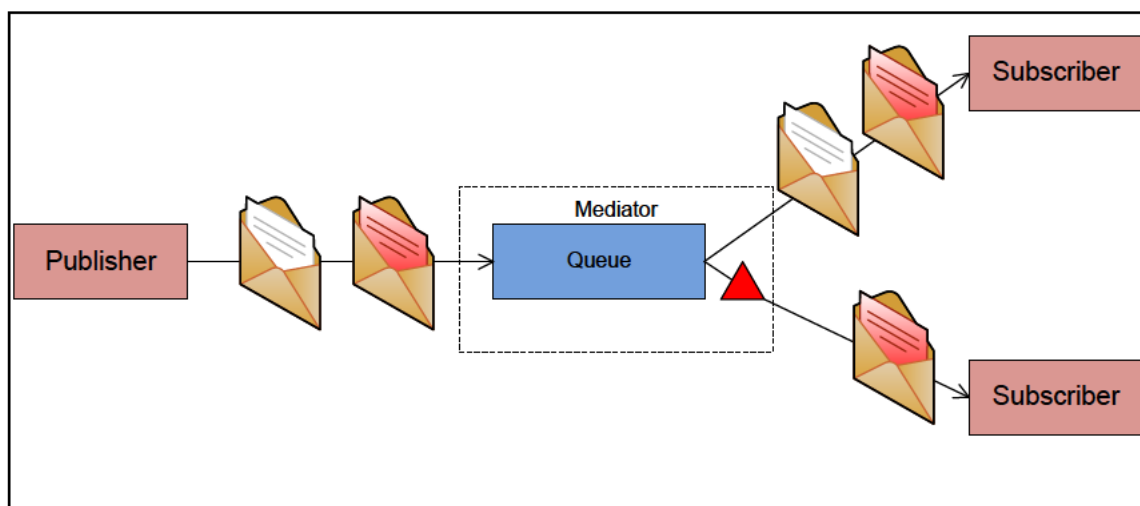


Figure 53 – Content-based Filtering

There is also the possibility to support a hybrid of the two, where publishers sent messages to specific topics, and subscribers register content-based subscriptions to one or more topics.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Some systems support a hybrid of the two; publishers post messages to a topic while subscribers register content-based subscriptions to one or more topics.

### 2.3.5.2.6 Data Management

#### 2.3.5.2.6.1 Examples of Data Format Transformations

Data Format Transformation	From format	To format
XML2XML	XML according to specific XSDs	XML according different specific XSDs
XML2String	XML according to specific XSDs	ASCII / Unicode string
XML2Binary	XML according to specific XSDs	Sequence of octets
XML2Base64String	XML according to specific XSDs	Base64 String
Formatted ASCII / Unicode string to Binary	Formatted ASCII / Unicode string	Sequence of octets
XML2CompBinary	XML according to specific XSDs	Sequence of octets representing compressed XML.
Formatted ASCII / Unicode string to Compressed Binary	Formatted ASCII / Unicode string	Sequence of octets representing compressed String.
Binary2Binary	Binary	Sequence of octets representing the compressed Binary.

Data encapsulation refers to sending data where the data is encapsulated with successive layers of control information before transmission across the network from a service consumer SWIM node to service provider. The reverse of data encapsulation is de-capsulation, which refers to the successive layers of where control data being removed at the receiving end of a network.

When a SWIM node sends a message, the message will take the form of a packet. Each OSI (Open System Interconnection) model layer adds a header to the packet. The packet is then covered with some control information directing it onward to a destination.

Similarly, the message in the packet is encapsulated with some information such as the address of next node, protocol information, the type of data and the source and destination addresses.

#### 2.3.5.2.6.2 Example of Data Encapsulation combined with Protocol Bridge

In the example below a transformation of the messaging protocol (Protocol Bridge) as well as an encapsulation of the format of the data takes place between the message originator and a Publisher.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

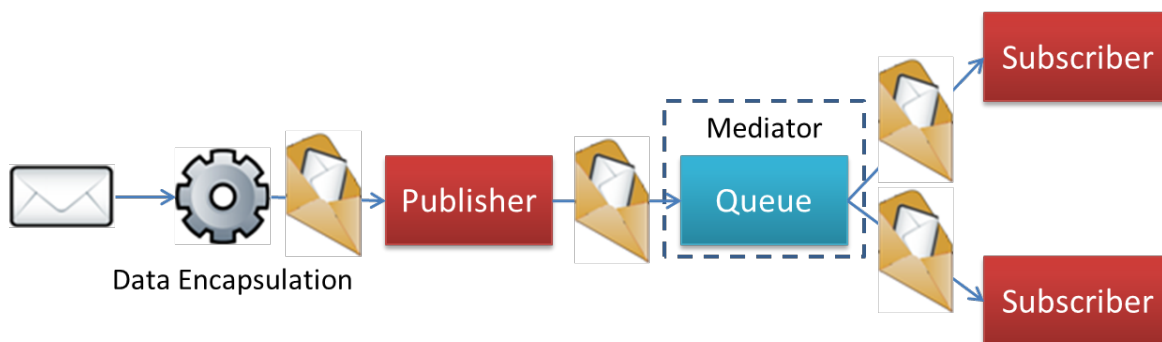


Figure 54 – Protocol Bridge with Data Encapsulation

### 2.3.5.2.6.3 Example of Data Format Transformation combined with Protocol Bridge

In the example below a transformation of the messaging protocol as well as a transformation of the format of the data takes place between the message originator and a Publisher.

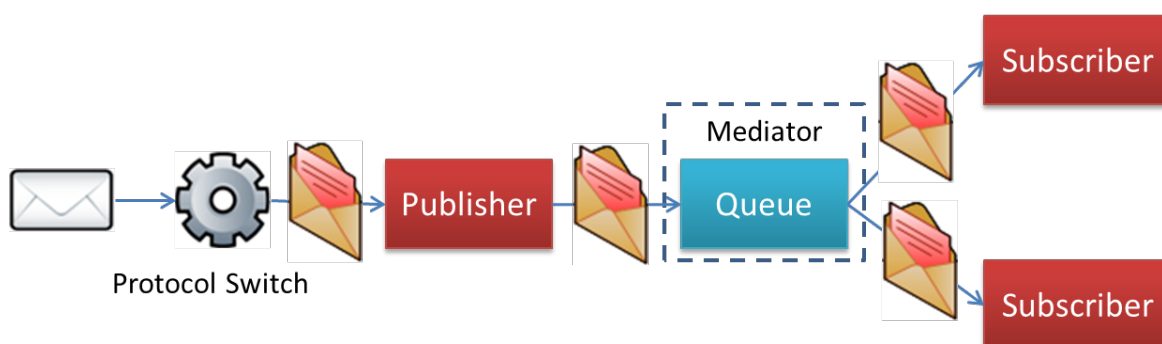


Figure 55 –Protocol Bridge with Data Format Transformation

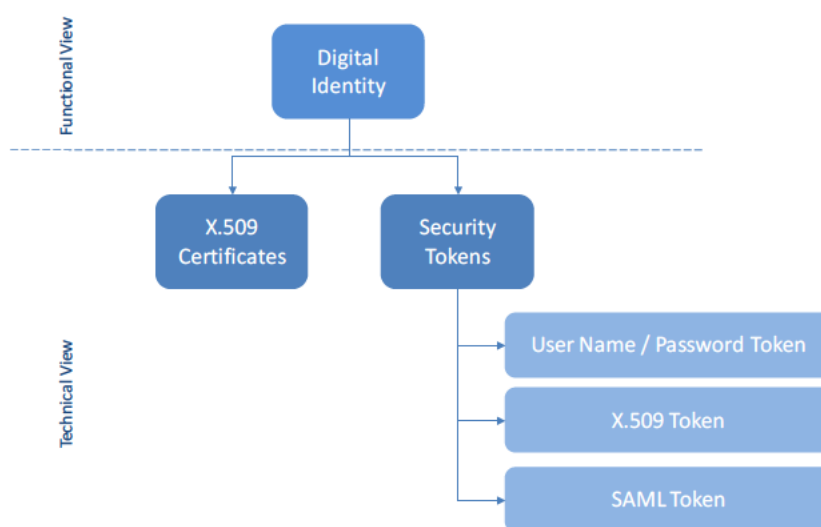
### 2.3.5.3 Security FB Architectural Options

Architectural options for Security FB exceed the local SWIM-TI Profile Instance and are understood at SWIM-TI overall level. SWIM-TI SEC FB provides technical functions as defined in 2.2.1.3. Options related to Brokered Authentication and Authorization are expressed in this section.

#### 2.3.5.3.1 Identity Management and Authentication

In this section the technical views concerning the Identity Management and Authentication are described. The technical views aim at detailing how Authentication and Identity Management elements from the functional view are properly instantiated at the technical level in order to implement the brokered authentication design pattern.

Conceptual difference and relationships between identity at logical and technical views are depicted in the figure below.



**Figure 56 – Identity at functional and technical views**

In the functional view, Authentication and Identity Management functions deal with Digital Identity as the means to claim for such identity (and related attributes) and the means to verify such identity. The Digital Identity may include additional information used for describing entity claims that can be then used at provider side to perform authorization decisions of authenticated entity.

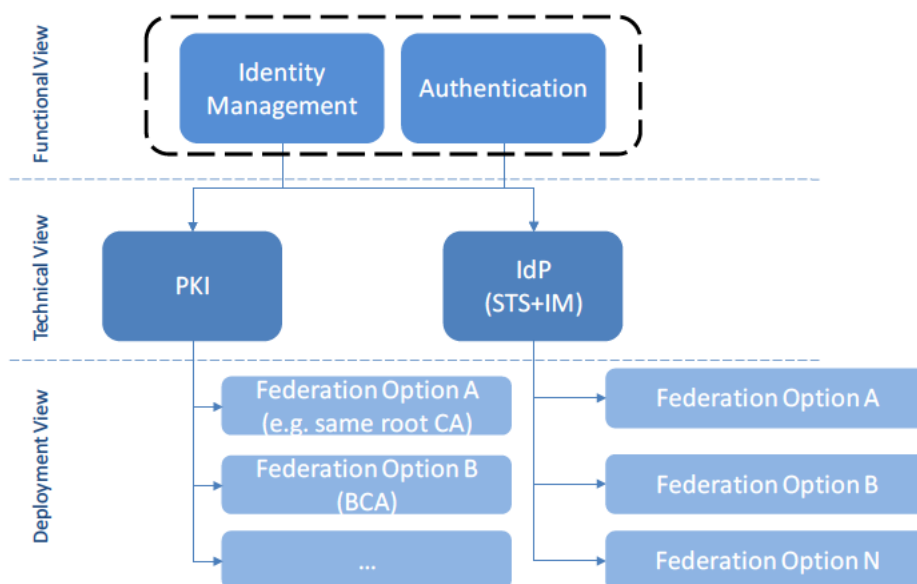
At technical view, the Digital Identity element can be one of the following types depending of the technical architecture and technology adopted:

- a. X.509 Certificates (related to a Public Key Infrastructure, PKI)
- b. Security Tokens (related to a Security Token Infrastructure, STI)
  - b.1 User Name / Password
  - b.2 SAML tokens
  - b.3 X.509 token

It is important to take into account that the target of SWIM-TI functional view are the ATM information exchanges between ATM application but also at SWIM-TI level there could be interaction not strictly ATM specific that will be secure: SWIM-TI Authentication aims at allowing authenticated ATM exchanges but at technical level and inside SWIM-TI there could be interactions (e.g. consumption of the Security Token Service) that will be also authenticated. So when referring to SWIM-TI authentication we refer to the enabling functionalities for authenticated ATM information exchanges,

then in the technical and deployment view those are detailed describing all the technical elements contributing to build that function (e.g. authentication towards STS).

According to the identified types of Digital Identity at the technical level, there could be two different technical views (see figure below): one based on Public Key Infrastructure (PKI) and one based on Security Token Infrastructure (STI). These two views are described in 2.3.5.3.1.1 and 2.3.5.3.1.2.



**Figure 57 – Brokered Authentication (logical) design pattern and related views**

In case “a.” the Identity Management is implemented by related technical functions provided by the PKI and the authentication consists of trust relationships and secure interactions between PKI, consumer side SWIM-TI and provider side SWIM-TI.

In case “b.” the Identity Management (IM) is implemented by related technical functions provided by the STI and the authentication consists of trust relationships and secure interactions between STI, consumer side SWIM-TI and provider side SWIM-TI.

In case “b.” there is or there could be a dependency with the main enabler of case “a.”: the PKI. This is because three reasons:

- When X.509 tokens are used the STI shall be able to properly interact (retrieving and verification) with a PKI that is the enabler managing the X.509 certificates.
- Typically security tokens are digitally signed by issuing STI.
- Independently of the token used, for such interactions (see 2.3.5.3.1.3) composing the case “b.” technical view transport level security may be used (e.g. consumer-STI interaction, ATM interaction using both transport level and message level security) and in that case X.509 certificates are used.

Both STI and PKI manage the identity lifecycle, that can be framed in similar stages to the life cycles of living things:

1. Creation
2. Utilization
3. Termination

Every stage in an identity’s life cycle has scenarios that are candidates for automated management. Finally when the Digital Identity is no longer put to active use, its status might need to be changed or the identity might need to be deleted from the identity store. All events during the life cycle of a Digital

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Identity need to be securely, efficiently, and accurately managed: in SWIM-TI such aspects of the identity management process are governed using the policy based approach.

There are no restrictions on the realization of the Identity Store, except for the fact that it shall be able to support all the kinds of managed Digital Identities. In case “a.” (PKI) the store (or repository) is typically realized using X.500 Directory Access Protocol (DAP) standards or Lightweight Directory Access Protocol (LDAP) standard. In both cases the Identity Store provides the infrastructure for meeting requirements about organization and secure storage of Digital Identities according to the current security policies.

In both technical views (cases “a.” and “b.”) proper Authentication Policies are used. In particular for technical view in case “b.” the policy includes the type of Identity Token that shall be used but also some constraints such as validity interval for the token.

The figure above introduces also, for a given technical view, possible deployment views or options. The main starting point is that such interactions have to occur between entities belonging to different administrative (or security) domains (inter-domains) that both have their own PKI/STI. A solution to allow use/reuse of those elements in cross domain interaction is the Identity Federation or Federated Identity Management.

In case “a.” the federation is based on trust relationships between CAs managing X.509 certificates and Certificate Revocation Lists (CRLs). Refer to 2.4.2.1.1 for further details.

In case “b.” the federation is achieved by federating the several STIs (see 2.4.2.1.2). When X.509 tokens are used, the federation of STIs includes the federation of PKIs they use. This is conceptually the same as case “a.”.

All the aspects concerning the federation (Identity Federation) of such technical views are described in 2.4.2.

### 2.3.5.3.1.1 X.509 Certificates Based Technical View (PKI)

Public Key Infrastructure (PKI) is responsible for signing, emitting and maintaining certificates and revocation lists after verification<sup>48</sup> of requester identity for the benefit of SWIM stakeholders that have not this facility. Two kinds of requests may be invoked. They are Certificate Signing Request (CSR) and Certificate Revocation List (CRL). Certificates are used for digital signature, encryption and authentication.

The Public Key Infrastructure ensures that

- Certification-related transmitted data have not been modified, eavesdropped or stolen during transfer,
- Certification-related transmitted data have been delivered to a known and trusted receiver,
- Certification-related transmitted data comes effectively from a known and trusted issuer in case a client certificate is used (mutual authentication).

A SWIM Certification Authority covers three functions that are:

- The Certificate Management that issues certificates and maintain revocation lists,

---

<sup>48</sup> ITU-T IdM X.1252 define this term as the process or instance of establishing the authenticity of something. Verification of (identity) information may encompass examination with respect to validity, correct source, original, (unaltered), correctness, binding to the entity, etc.

- The Registration Authority (RA) is the administrative function in charge of registration<sup>49</sup> of entities in the Public Key Infrastructure (PKI),
- The Validation Authority (VA) is the function in charge of validation of the certificates.

All together this is named a Public Key Infrastructure (PKI).

A Bridge Certification Authority (BCA) is a Certification Authority in charge of interconnecting trust domains by creating and revoking pair of cross-signed certificates with the different trusted Certificate Authorities. This function provides the capability to authenticate physical or virtual machines, applications or users all over the SWIM-TI, allowing the establishment of certification paths between any of the SWIM stakeholder whatever their respective PKIs are.

In the figure below the technical view is depicted highlighting ATM and SWIM-TI specific interactions and perimeters. In particular the picture represents the case when mutual authentication is required.

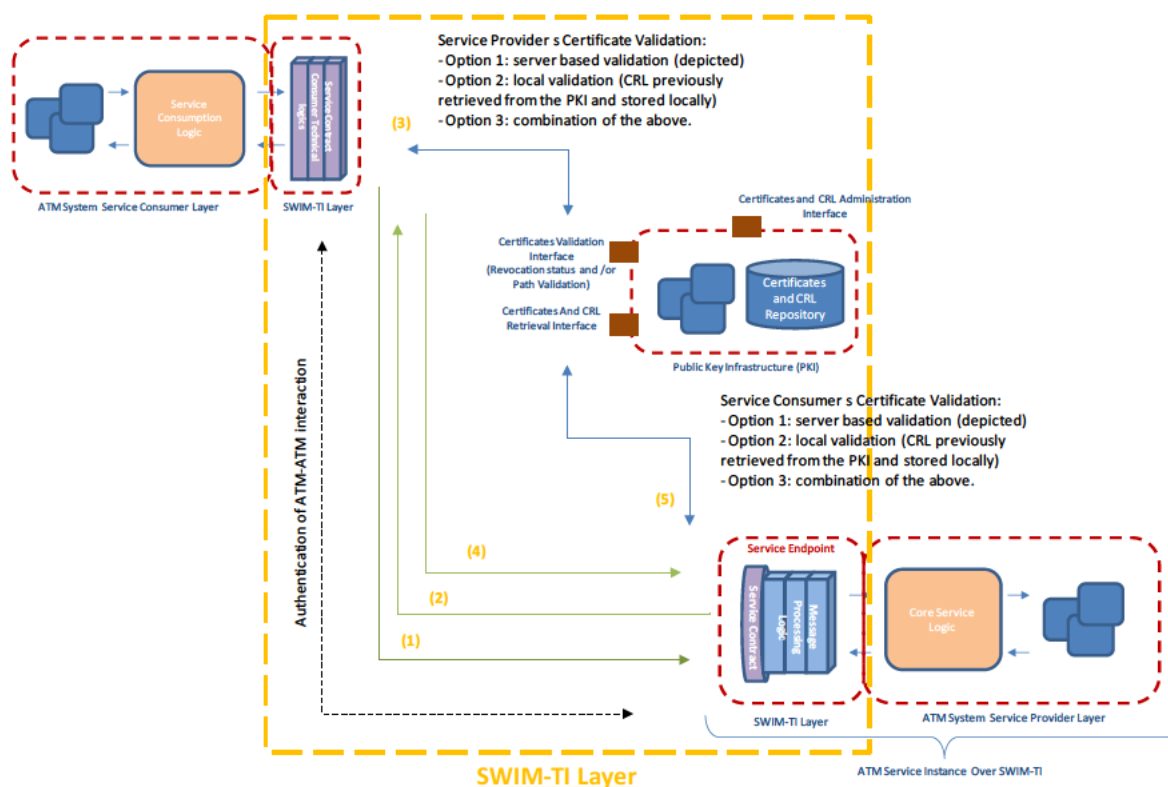


Figure 58 – Brokered Authentication Based on X.509 Certificates

The scenario is the following: two ATM applications are going to interact (ATM service) according to the SRR-MEP and authentication is required and as part of the ATM service contract they rely on X.509 certificates for mutual authentication.

The steps performed during Brokered Authentication using this technical view are the following:

- STEP.0: Agreed Authentication Policy shared.
- STEP.1: SWIM-TI layer on ATM service consumer side requests a protected service.

<sup>49</sup> ITU-T IdM X.1252 define this term as a process in which an entity requests and is assigned privileges to use a service or resource.

- STEP.2: SWIM-TI layer on ATM service provider presents its X.509 certificate.
- STEP.3: SWIM-TI layer on ATM service consumer side verifies the provider certificate by requesting the CA.
- STEP.4: SWIM-TI layer on ATM service consumer side presents its X.509 certificate.
- STEP.5: SWIM-TI layer on ATM service provider side verifies the consumer certificate by requesting the CA.

### 2.3.5.3.1.2 Security Tokens based Technical View (STI)

In the figure below the technical view is depicted highlighting ATM and SWIM-TI specific interactions and perimeters. Furthermore the picture aims at highlighting authentication elements and for simplicity other aspects such as authorization, integrity, etc., are not depicted.

The picture also differentiates (to clarify) between ATM-ATM interactions (dotted red arrows) and SWIM-TI level interactions (dotted black arrows): the main target of the view is to allow authenticated ATM-ATM interactions (for simplicity in the figure just the request part of the ATM Request/Reply is depicted), that at SWIM-TI level is achieved by properly composing interactions such as the authenticated interaction between SWIM-TI consumer side and the STS.

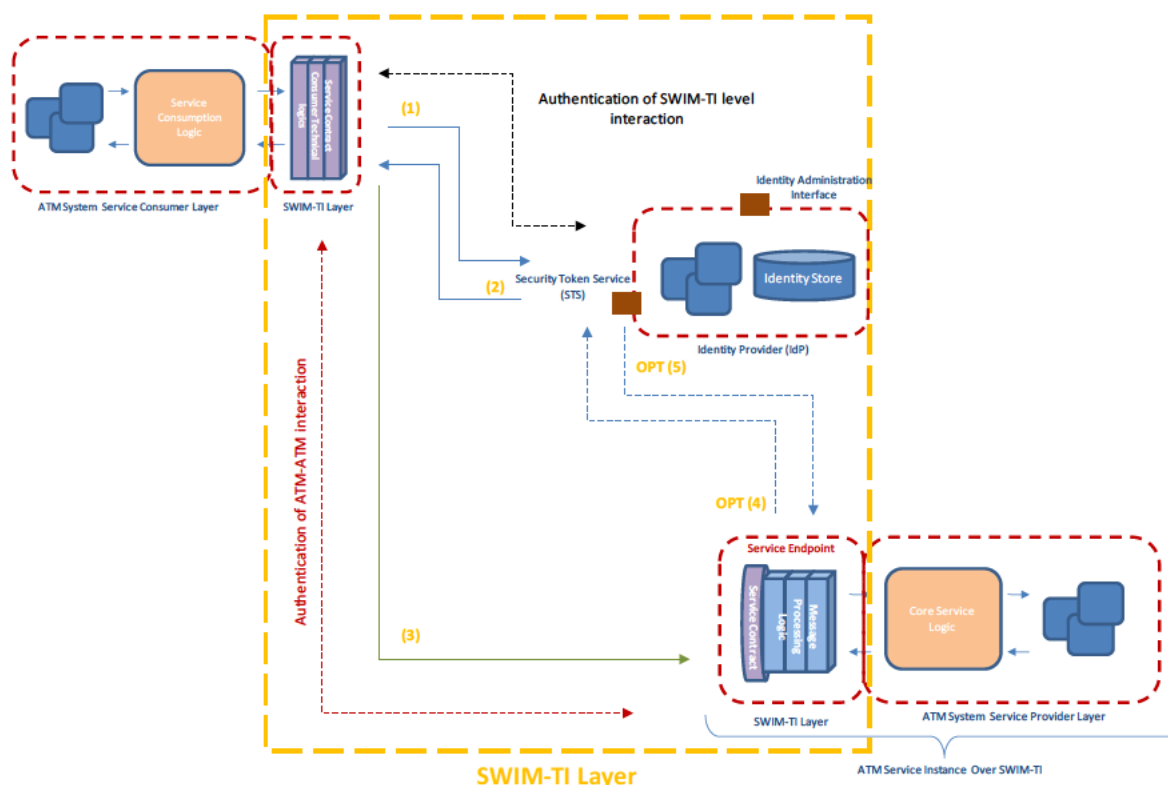


Figure 59 – Brokered Authentication Based On Security Token

The scenario is the following: two ATM applications are going to interact (ATM service) according to the SRR-MEP and authentication is required and as part of the ATM service contract they rely on: STI (in the figure referred to as IdP. In the technical view no considerations about how many IdPs (STIs) are used) and Authentication Policy including the type of token to be used.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



As depicted in the figure the IdP (STI) provides two types of interfaces: one for Identity Administration (create, delete, etc.) and one for token issuing and verification (STS).

The steps performed during Brokered Authentication using this technical view are the following:

STEP.0: Consumer identity available on IdP (STI) side, agreed token type and Authentication Policy shared.

STEP.1: SWIM-TI layer on ATM service consumer side, being demanded to enable authenticated ATM service consumption, submits a request to the IdP STS to retrieve the security token. This interaction is typically authenticated (username and policy, X.509) and the requester may add additional information used by the IdP to assign to it such attributes/properties (used for token retrieving and/or for adding additional information in the token - such as authorization info). This step (and the next one - STEP.2) is not always required: in case of SSO or in general until the token previously retrieved is still valid, the ATM service consumer may reuse token for different consumption requests for the same or different ATM service.

STEP.2: IdP (STI) successfully authenticates the requester and its attributes and then it issues the specific security token (intra-IdP interaction with identity store).

STEP.3: SWIM-TI layer on ATM service consumer side receives the security token and attaches it to the ATM service consumption request.

STEP.4: SWIM-TI layer on ATM service provider side, being demanded to enable authenticated ATM service consumption, verifies the token attached to the consumption request. Typically, this step may just consist of the verification of the digital signature of the token to verify that it has been issued by a trusted IdP (STI). However there could be cases when the IdP (STI) is contacted to verify such identity.

STEP.5: In case the IdP (STI) is contacted to verify such identity, it does that and replies accordingly.

STEP.6: The scenario continues with authorization, etc. and then with the reply.

For completeness, as anticipated before, it is important to highlight that in some cases the above view could be extended introducing also the PKI because: a) transport level security in STEP.1; b) transport level security in STEP.3 (for integrity and confidentiality) and c) use of X.509 tokens. These relationships are discussed in detail in 2.3.5.3.1.3.

There is a variety of security technologies associated with this technical view (just those authentication related):

- WS-Security Framework
  - WS-Security Core Specification.
  - Username Token Profile
  - X.509 Token Profile
  - SAML Token profile
  - WS-Policy
  - WS-SecurityPolicy
  - WS-Trust
- Security Assertion Markup Language (SAML) 2.0

founding members

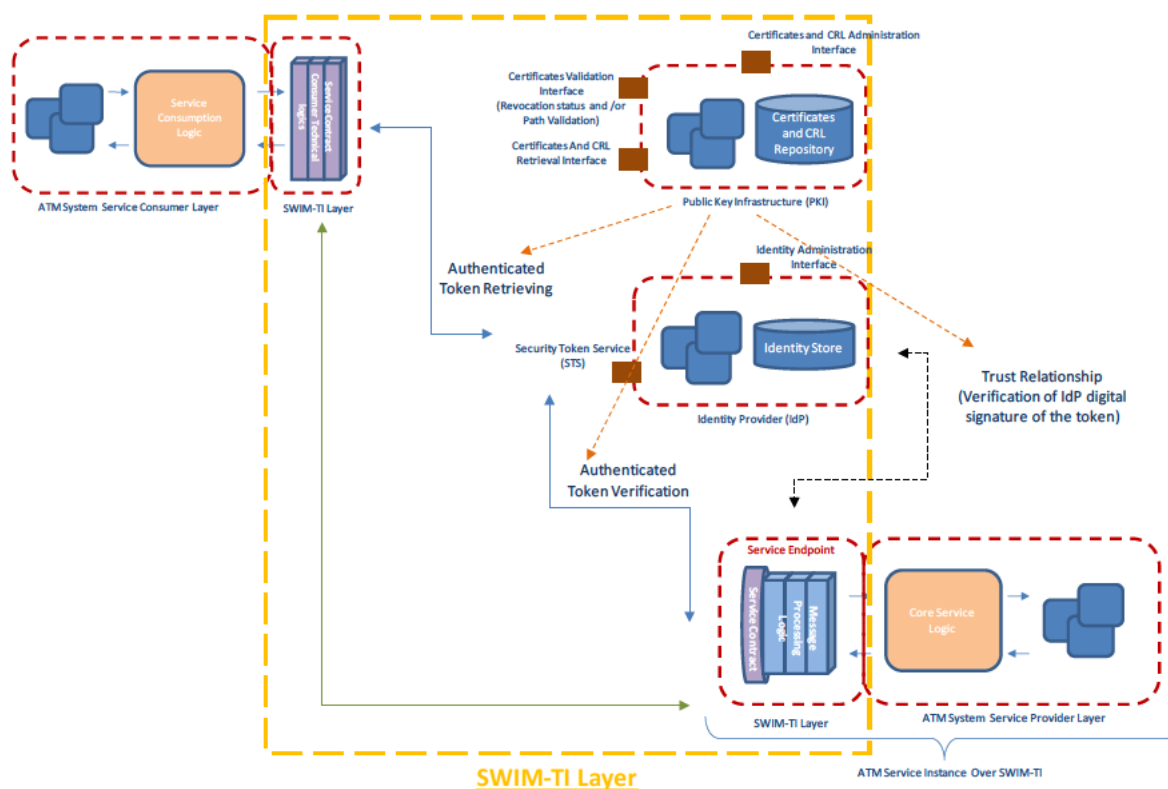


Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

161 of 284

### 2.3.5.3.1.3 Use of PKI in Security Tokens Based Technical View

In the figure below possible relationships between Security Tokens based brokered authentication technical view and the PKI are depicted.



**Figure 60 – Use of PKI in Security Token based Brokered Authentication**

In case “b.” (see 2.3.5.3.1) there is or there could be a dependency with the main enabler of case “a.”: the PKI. This is because three reasons:

- When X.509 tokens are used the STI shall be able to properly interact (retrieving and verification) with a PKI that is the enabler managing the X.509 certificates.
- Typically security tokens are digitally signed by issuing STI.
- Independently of the token used, for such interactions composing the case “b.” technical view transport level security may be used (e.g. consumer-STs interaction, ATM interaction using both transport level and message level security) and in that case X.509 certificates are used.

For completeness, as anticipated before, it is important to highlight that in some cases the above view could be extended introducing also the PKI because: a) transport level security in Brokered Authentication Based On Security Token STEP.1; b) transport level security in Brokered Authentication Based On Security Token STEP.3 (for integrity and confidentiality) and c) use of X.509 tokens.

### 2.3.5.3.2 Authorization

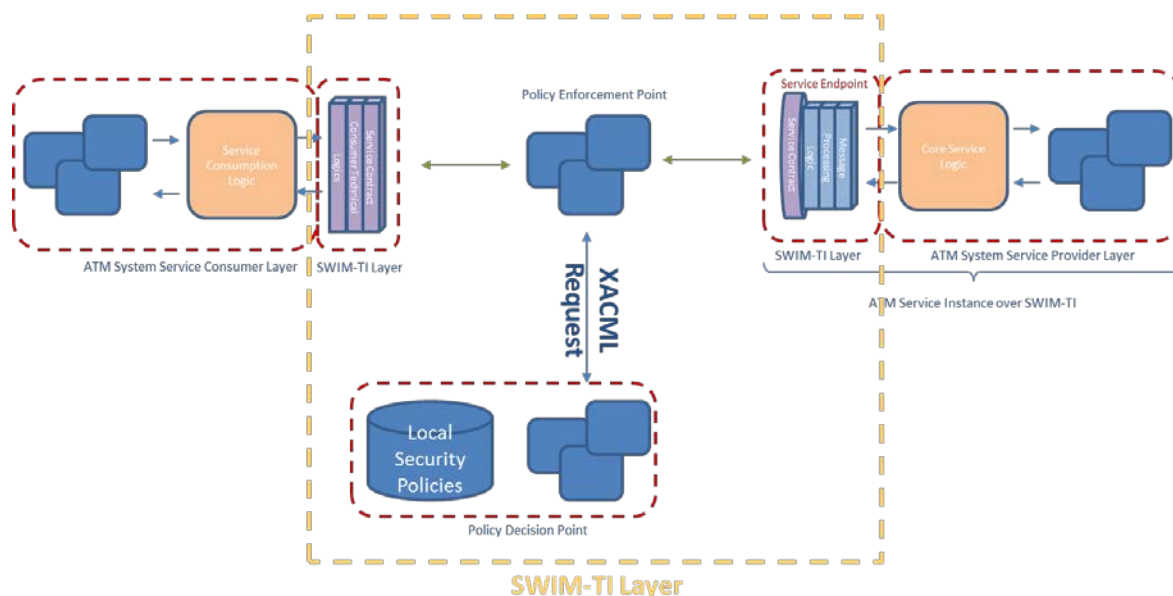
In this section the technical views concerning the authorization are described. Currently only XACML Attribute Based Access Control is described.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

In the figure below the technical view is depicted highlighting ATM and SWIM-TI specific interactions and perimeters.



**Figure 61 – Use of X.509 attribute certificates and XACML request for Authorization**

The scenario is the following: two ATM applications are going to interact (ATM service) according to the SRR-MEP and authorization is required and as part of the ATM service contract they rely on: Attribute Based Access Control supported by X.509 certificates.

As depicted in the figure the Policy Enforcement Point intercepts any incoming request, asks the PDP to check authorization according to attribute values and rejects the incoming request when denied by PDP.

The steps performed during Authorization using this technical view are the following:

- STEP.0: PEP intercepts the incoming request and extracts authorization attributes from the consumer certificate.
- STEP.1: PEP requests the PDP (authorization server) using XACML language.
- STEP.2: PDP validates the attributes against the authorization policy currently in force and responds to PEP
- STEP.3: PEP rejects or forwards the incoming request accordingly.

### 2.3.5.3.3 Option pros and cons

Option	Pros	Cons
Brokered Authentication	<p>Validation and issuing of Digital Identities.</p> <p>Allows to uniquely identify each SWIM-TI Stakeholder.</p> <p>Distributed management of identities.</p> <p>Ability for creating secured domains.</p> <p>Not single point of failure.</p>	-
Authorization	<p>Information is integral.</p> <p>Ability to authenticate information.</p> <p>Only intended recipients consume the intended information.</p>	-

Table 26 – Security FB Architectural Options Pros and Cons

### 2.3.5.3.4 SWIM-TI SEC FB Architectural choice

At the time being, the most appropriate options for securing the SWIM-TI are:

SWIM-TI Profile	Architectural Choice
Blue Profile	<p><b>Authentication</b> – Brokered Authentication</p> <p><b>Authorization</b> – Attribute Based Access Control</p>
Yellow Profile	<p><b>Authentication</b> – Federated Brokered Authentication</p> <p><b>Authorization</b> – Attribute Based Access Control</p>
Purple Profile	<p><b>Authentication</b> – Federated Brokered Authentication</p> <p><b>Authorization</b> – Attribute Based Access Control</p>

Table 27 – SWIM-TI Security FB Architectural choice

This is specified in SWIM-TI Profiles TS (ref. [13]).

## 2.3.5.4 Supervision FB Architectural Options

At the time being there's not a unique Architectural option for any of the available SWIM-TI Profiles, as the SOV is understood to be managed at local SWIM-TI Profile instance level and not considered to be standardized. However, studies were conducted and are still kept in Appendix D for further evaluations.

### 2.3.5.4.1 Option pros and cons

N/A

### 2.3.5.4.2 SWIM-TI SPV FB Architectural choice

N/A

### 2.3.5.5 Shared Object FB Architectural Options

For a better understanding, a concrete realization of the shared object for Flight Objects will be used in order to focus on the first operational usage based on Flight Objects.

#### 2.3.5.5.1 Architectural options

##### 2.3.5.5.1.1 SWIM FO/IOF General Context

The main challenges facing the distribution of Flight Objects are related to the Systems of Systems nature of the European ATM, the use of a Wide Area Network (WAN), the use of multiple DDS vendors and the impact of High Availability and redundancy solutions on network resource usage.

Flight Object management within the Blue profile is based on patterns defined within ED-133 specification.

A Flight Object (FO) is decomposed into multiple **Clusters** (13 in [32]). Clusters can be sent in any order, and only updated clusters are published. Flight Objects are published using OMG Data Distribution Service (DDS). ED-133 specification defines a special DDS topic to allow consistent management of Clusters releases. The **Summary** Topic contains Cluster releases. A Summary data sample is sent whenever one or more Clusters are sent. Additionally, each Flight Data Manager/Publisher (FDMP) performs a periodic publication of all summaries of Flight Objects under its responsibility.

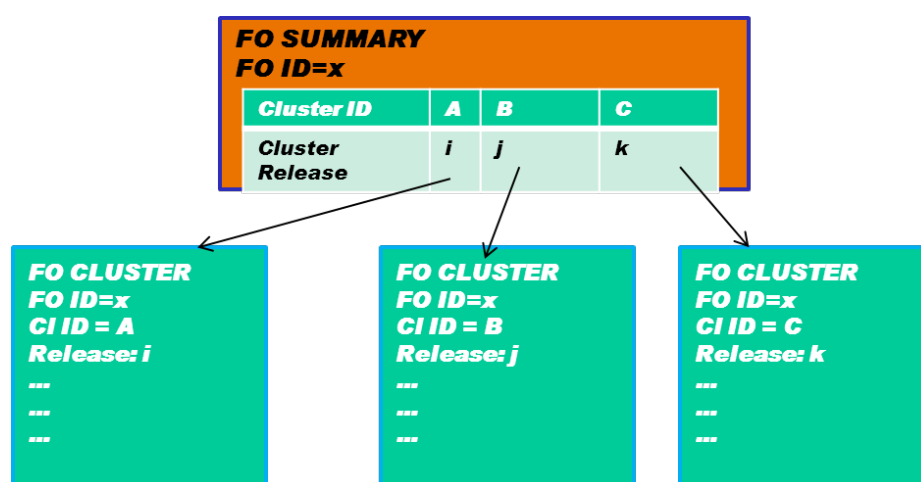


Figure 62 – Overview of Flight Object data model

##### 2.3.5.5.1.1.1 System of Systems

The System of Systems nature of European ATM implies a large number of stakeholders requesting sharing of Flight Objects; making **scalability** of high importance. Involved stakeholders belong to different ownership domains which increase importance of **security** and increase the complexity of the system and network architecture

##### 2.3.5.5.1.1.2 Wide Area Networks (WAN)

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Exchange of Flight Objects occurs over Wide Area Networks with **limited bandwidth**. Current SESAR VPN provides at best 2 Mb/s connections compared to over 100Mb/s common on Local Area Networks (LAN).

A variety of communication equipment is used within WANs which may imply **various sizes for Protocol Data Units** (PDU).

While multicast technologies are very widespread in Local Area Networks, there is **no trivial support for multicast** in a WAN environment. Some emerging multicast protocols driven by Video on Demand market, such as Source-Specific Multicast (SSM), seem best suited for WANs.

Securing access to Local Area Networks often involves firewalls and often involves use of Network Address Translation (NAT) techniques to hide local addressing schemes from outsiders. It is important for SWIM-TI communication infrastructure to support **NATing** and **Firewall traversal**.

The following approach needs to be followed when dealing with WANs:

- Limit bandwidth usage.
- Use compression.
- No IP fragmentation.
- Reuse network capabilities (PIM-SM, SSM, and ASM).
- Favour filtering at source level.

#### 2.3.5.5.1.1.2.1 Limit bandwidth usage

Bandwidth with the reliability required by SWIM is a scarce resource and it is often very expensive. Available bandwidth shall only be used when absolutely necessary. Favour local reconstitution of data when possible.

#### 2.3.5.5.1.1.2.2 Use compression

Overhead induced by compression of data is often acceptable because of the benefits of less data transiting on the WAN.

#### 2.3.5.5.1.1.2.3 Configurable datagram size

When sending a large amount of data, it is preferable that IP datagrams be of the largest size that does not require fragmentation anywhere along the path from the source to the destination [37]. This is particularly important for multicast communications using UDP.

For security reasons, many firewalls also drop IP fragments. Fragmenting encrypted packets consumes computing resources on IPSec appliances because only reconstituted packets can be decrypted. It is thus necessary to proceed with data fragmentation at the application level when necessary in order to avoid fragmenting IP packets at the network level.

#### 2.3.5.5.1.1.2.4 Reuse network capabilities (PIM-SM, SSM, and ASM)

Transmitting the same data to multiple receivers over the network efficiently requires use of multicasting. Traditionally Wide Area Networks are not multicast friendly; but the development of new techniques such as Protocol Independent Multicast (PIM) coupled with success of many multimedia and triple play applications; makes multicasting over Wide Area Networks popular.

Figure 63 presents an efficient distribution of Flight Objects where an FDMP is publishing a Flight Object that is only delivered by stakeholders within the distribution list.

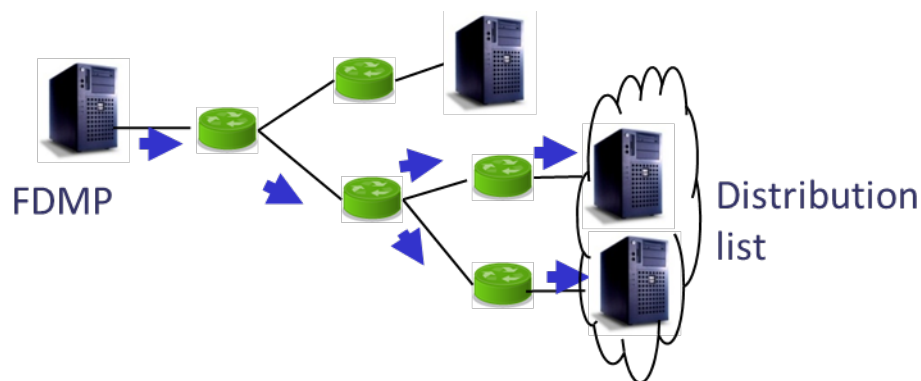


Figure 63 – Efficient use of Network

#### 2.3.5.5.1.1.2.5 Favour filtering at source level

Filtering may be used in multiple forms and in multiple locations in order to only deliver to the application requested/required data. While transparency to receiving applications is always assured, the location where filtering is performed within the network will be a decisive factor on how efficient filtering is. Filtering close to the publisher is far better than filtering at the receiver side as dropping unwanted data after carrying it all the way to the receiver host represents a waste of network resources.

#### 2.3.5.5.1.1.3 Multiple Class of Service (CoS) options for the network

ED-133 specification defines 3 QoS categories ( $d_1$ ,  $d_2$ ,  $d_3$ ) to define priorities for the publication of Flight Objects. ED-133 priorities address mechanisms to provide priority to certain interactions according to some criteria. For correct handling of such classification, the network infrastructure has to provide at least 3 Classes of Service (CoS) to differentiate between the traffic of each category.

There is a widely available standard called **Differentiated Service** (DiffServ<sup>50</sup>) with a relatively straight forward and pragmatic architecture.

DiffServ specifies a mechanism based on Differentiated services Field (DS field) in the IP header for packet classifying and managing network traffic on IP networks.

#### 2.3.5.5.1.1.4 Multiple DDS vendors

The DDS market is very active and multiple vendors already provide industrial grade DDS products. Relying on **standardized wire protocols** such as DDSI ensures interoperability of the SWIM-TI infrastructures.

#### 2.3.5.5.1.1.5 High Availability

Ensuring high availability of SWIM-TI infrastructures and services requires redundancy of data publishers and subscribers. It is necessary to **limit impact of local redundancy** on the other IOP participants. Adding a replica shall not induce reasonably avoidable communication and data transfer on the network.

#### 2.3.5.5.1.2 Current ED-133 approach

The sharing of Flight Objects uses a messaging infrastructure implementing a protocol on top of the OMG DDS. It relies on many DDS QoS such as RELIABILITY, DURABILITY, PRESENTATION, and

<sup>50</sup> <http://datatracker.ietf.org/wg/diffserv/>



DEADLINE. The messaging infrastructure supports message retries, detects all Flight Object releases and delivers to application only the latest coherent set clusters that are published.

ED-133 specification defines 3 categories ( $d_1$ ,  $d_2$ ,  $d_3$ ) to define priorities for the publication of Flight Objects. These categories will have to be relayed somehow to the underlying transport to get help from the network infrastructure in enforcing such priorities.

The following sections analyse some of the DDS QoS currently in use that need to be amended in order to provide true interoperability.

#### 2.3.5.5.1.2.1 DESTINATION\_ORDER

Use of BY\_SOURCE\_TIMESTAMP assumes filtering at the receiver side. When two publishers update the same data instances (same Flight Object), the two publications will be published on the network and a subscriber will receive both publications before deciding which of these two publications is to be retained based on the time stamp of the publisher. ED-133 patterns will make sure that only one publisher (FDMP) is publishing at any time for any Flight Object; but the case may happen when some local replica is starting (see impact of High Availability) at one location and all other previous publishers of a flight object may republish their latest publications for a filtering at the receiver side. This is not efficient as far a network usage is concerned.

As a side effect of relying on BY\_SOURCE\_TIMESTAMP QoS value, all SWIM nodes within the IOP area require to be time synchronized.

#### 2.3.5.5.1.2.2 DURABILITY

The PERSISTENT QoS value is not supported by DDS interoperability protocol (DDSI). Only VOLATILE and TRANSIENT\_LOCAL QoS values are defined by the interoperability protocol.

#### 2.3.5.5.1.2.3 PRESENTATION

ED-133 specification requests the use of GROUP *access\_scope* value for the PRESENTATION QoS to atomic delivery of Flight Object clusters and summary data samples. Since Clusters and Summaries are on different topics, the coherent access shall span multiple DataWriters (in the same Publisher); but this is not covered by the DDS interoperability protocol.

#### 2.3.5.5.1.2.4 PARTITION

The current specification relies on the PARTITION QoS for controlling Flight Object distribution and making sure only stakeholders in the distribution list of a Flight Object only receive it. A DDS Partition is only a logical entity, so communication is possible between DDS entities in hosts not belonging to the same logical partition. The use of DDS partitions may result in multiple publications over the network; as some DDS vendors publish per DDS partition. A Flight Object could then be sent to all hosts within the DDS domain and then filtered-out at destination according to DDS partition; which is not efficient with network usage.

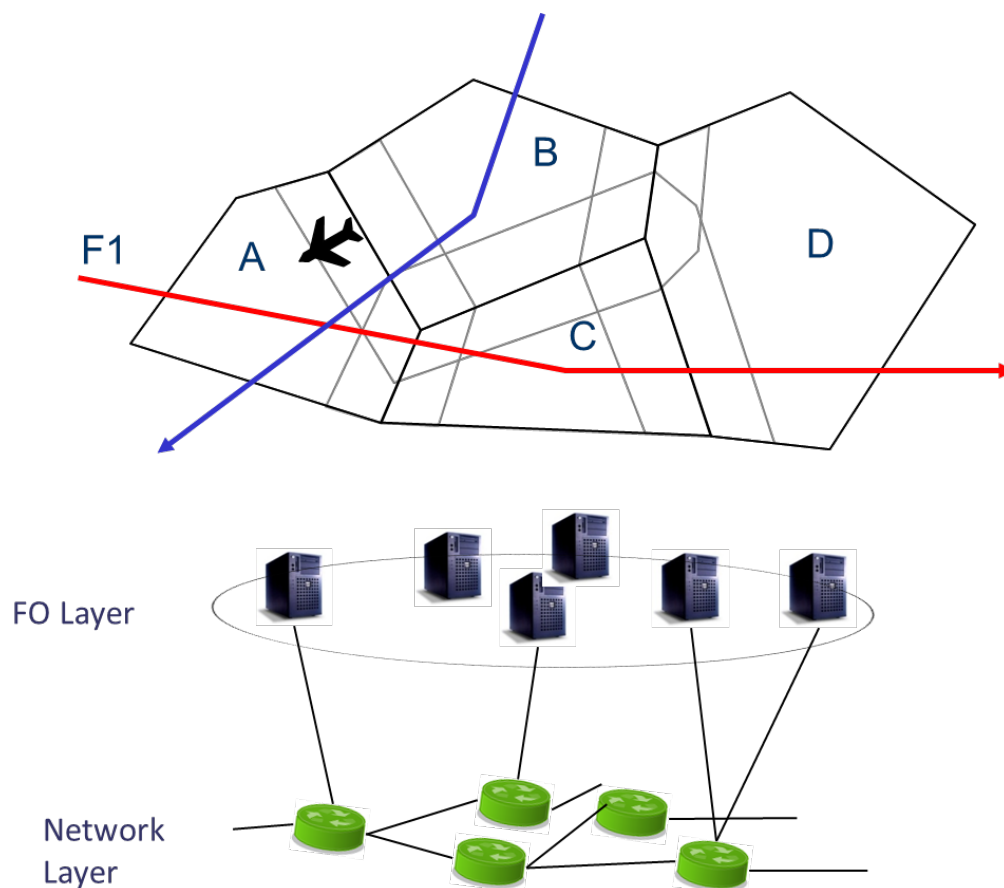
### 2.3.5.5.1.3 FO Overlay Network

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

169 of 284



**Figure 64 – General Architecture of the SWIM FO Overlay Network**

The proposed architecture defines an overlay network for the distribution of Flight Objects within the European ATM. The network layer is under the responsibility of the WAN provider (physical layer).

The FO Layer (FO Overlay) will include software artefacts around the edge of the physical network to enable efficient use of available network resources and to adapt to available communication protocols available for the stakeholders.

The FO Overlay shall be **decentralized** and **secured**.

The FO Layer can be further sub-divided into two planes: a Control Plane and a Data Plane.

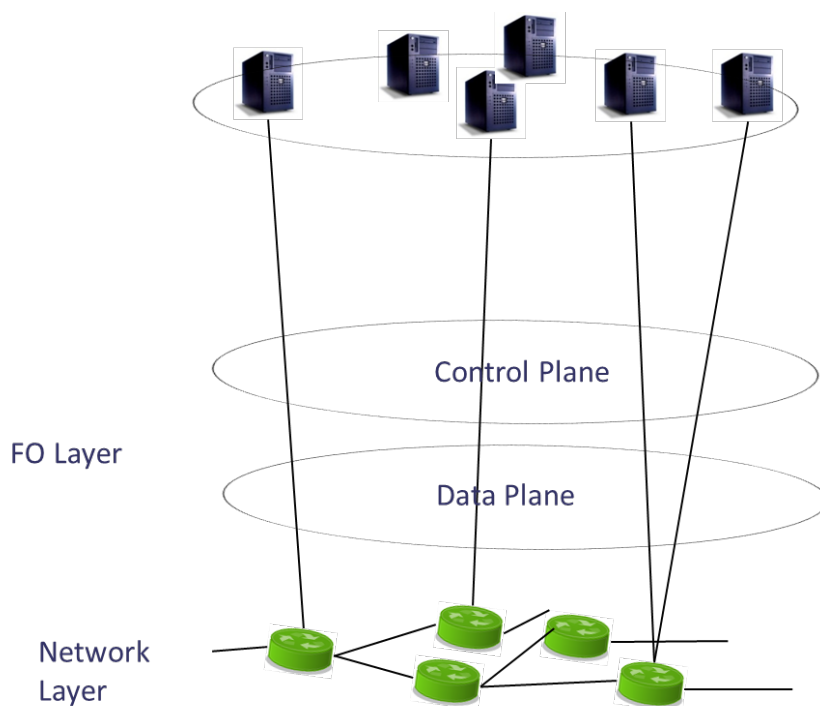


Figure 65 – SWIM FO Overlay network – FO Layer

#### 2.3.5.5.1.3.1 Network Layer

The Network layer is responsible for efficient delivery of Flight Objects within the European ATM.

The network layer shall support **IGMP v3** for IPv4 and **MLDv2** for IPv6 on the ANSPs domains (LANs) and PIM-SM with the two main modes: the traditional Any Source Multicast (ASM) and the Source-Specific Multicast (SSM).

PIM-SSM is easier than PIM-ASM to deploy because it does not require Rendezvous Points (RP). PIM-SSM requires use of IGMPv3 on the receiver side though.

**NOTE:** The Network Layer is the traditional network architecture that also contains its own control, data/forwarding, and management planes; but these are outside of the scope of this document.

In addition to above protocols and services, there is a need for service differentiation in the WAN. Traffic marking/classification by means of IP-layer packet marking (using the DS field) is needed.

#### 2.3.5.5.1.3.2 FO Layer

##### 2.3.5.5.1.3.2.1 Control Plane

The Control Plane is responsible for setting up the right communication path between DataWriters and DataReaders for efficient distribution of flight objects. This takes advantage of available knowledge of the **FO Distribution List**, the **FO Publisher**, **Offered and Requested DDS QoS**, **Participant and Endpoint Discovery protocols**, **communication protocols such as multicast technologies and IGMP versions** for optimum use of network resources.

DDS Discovery protocols are part of this Control Plane. Further requirements will be allocated on the DDS technology to take advantage of the multicast technologies target for Wide Area Networks. As

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

this implies interoperability, a proposal shall be made by SESAR partners involved at the OMG together with some DDS vendors to enhance DDSI specification.

The Control Plane shall take into account available network technologies and protocols (PIM-SSM, PIM-ASM, Unicast UDP/IP and/or TCP/IP) and the FO Distribution List in order to map DDS partitions to multicast routes (if any). For this purpose, the publication of FO\_SUMMARY is considered part of the Control Plan.

FO SUMMARY publication is an important service of the FO Layer. FO summaries are required for the following purposes:

- Improve knowledge of the stakeholders on available Flight Objects. FO summaries convey meta-data on each FO containing the Flight Key, the latest revision and the publisher Identifier.
- Distribute per Flight Object stakeholder interest in receiving all FO data. FO summaries convey meta-data on each FO containing the Distribution List. The Distribution List may be used for DDS Discovery and/or setting up communication paths between FO publisher and members of the distribution list.
- Consistent reconstruction of FO locally. FO summaries convey meta-data on each FO containing the revision of each FO cluster (Flight Object data).
- Supervision information on each FO and its publisher as each FO publisher is publishing periodically FO summaries of all its locally managed FOs.

#### 2.3.5.5.1.3.2.2 Data Plane

The Data Plane constitutes the 'Fast Path' within the FO Overlay Network. This takes advantage of the work done by the Control Plane to deliver Flight Objects to stakeholders within the Distribution List in a network efficient manner, i.e. only send to interested nodes.

Once multicast routes established by the control plane, Flight Object data (FO Clusters) will be published in a one-to-many multicast from the FDMP to all members of the distribution list.

The dynamic nature of the distribution list does not affect network configuration and the delivery of FO data through SSM multicast is very efficient network-wise (only nodes in the distribution list will receive the network packets).

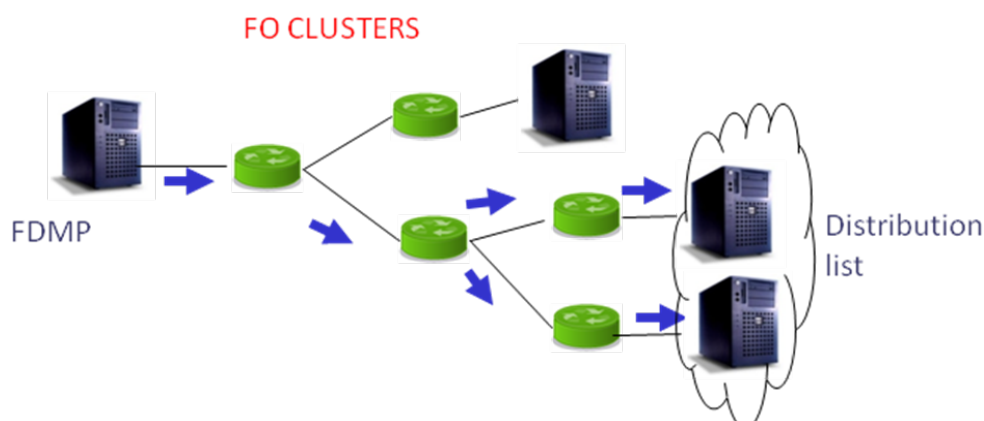


Figure 66 – FO CLUSTERS distribution

2.3.5.5.1.3.2.3 Architectural Elements

The proposed architecture can be described as Hierarchical P2P architecture. Three kinds of nodes are defined: FO Node, FO Router, and FO Broker. These are roles within the FO Overlay network that software applications can take. These can be deployed on separate computing resources or collocated and sharing common hardware. An FO Node is within a SWIM Node; while FO Broker and FO Router may be deployed within Common Configuration Capabilities under the responsibility of ANSPs and/or WAN provider. The FO Broker and/or FO Router may be on a SWIM Node already hosting an FO Node (A SWIM Node may have all the roles at the same time).

'FO Brokers' nodes are part of the control plane as they contribute to DDS discovery and are responsible for the global FO Summary publication/forwarding to all the stakeholders.

'FO Router' relies on the control plane for setting up the data plane for efficient use of network resources. 'FO Router' nodes are responsible for exchanging Flight Objects via the WAN reusing available network protocols.

'FO Node' is within a SWIM Node hosting FO publishers, consumers and/or users. 'FO Nodes' are the basic nodes within the architecture. FO Nodes only discover other FO Nodes under the same FO Router. Exchanges with 'external' FO Nodes on other FO Routers are not seen and no global addressing scheme is required.

Collocating an FO Broker and FO Router within a single node constitutes a Delegate as in BU-2 activity.

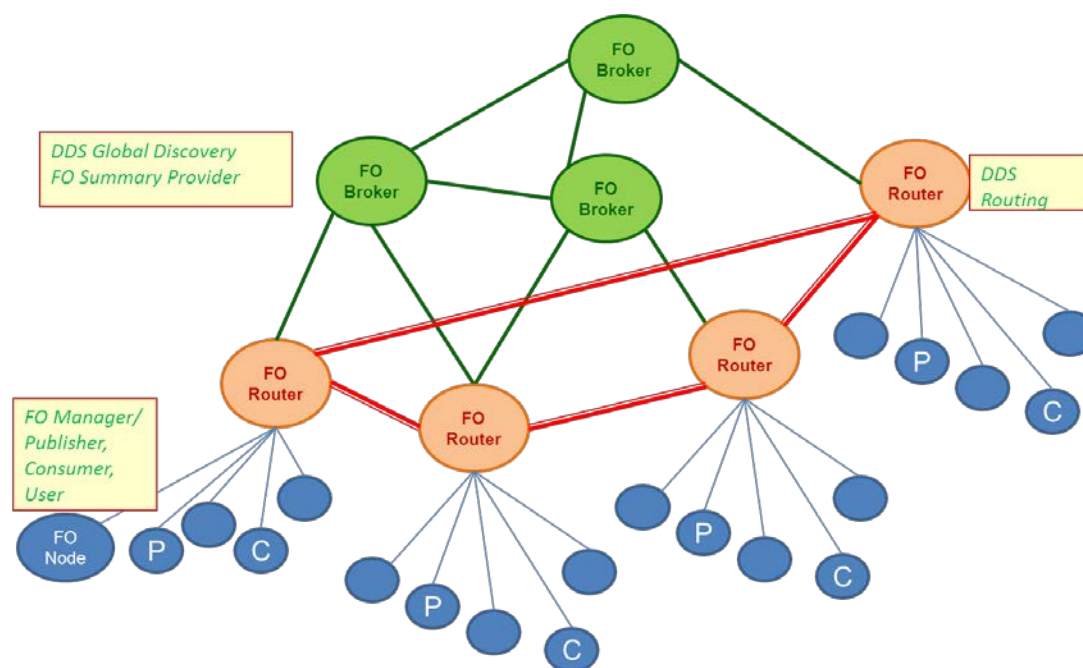


Figure 67 – FO Overlay Architecture

2.3.5.5.1.3.2.3.1 FO Router

'FO Router' nodes are responsible for exchanging Flight Objects via the WAN reusing available network protocols. The FO Router is mainly a 'DDS router' and acts as a gateway between the SWIM Nodes within the ANSP LANs and the WAN.

The Gateway relies on network protocols available locally and may rely solely on existing Simple Participant and Endpoint Discovery Protocols.

founding members



Inter FO Router communication will make the best use of the available WAN network, i.e. network efficient. This may rely on PIM-ASM and/or PIM-SSM multicasting or even unicast (TCP/UDP over IP).

Each FO Router will communicate with one or more FO Brokers for discovery of other FO Routers and for the subscription to FO Summary publications.

A global addressing scheme is required for inter-FO Router communication.

#### 2.3.5.5.1.3.2.3.2 FO Broker

'FO Broker' node responsibilities include DDS Discovery, Distribution of FO Summaries, and setting up subscriptions according to FO Distribution List.

Being decentralized, the overall architecture relies on many FO Broker nodes. Deployment of the FO Broker nodes is subject to availability of many-to-many communication or not in the WAN.

When many-to-many communication is available, FO Brokers may be collocated with the FO Routers and all FO Brokers will perform DDS discovery and FO summary publications. When no many-to-many is available (only PIM-SSM or unicast), some FO Broker nodes will be dedicated for DDS discovery (for initial bootstrap) and for the publication of FO Summaries to stakeholders not in the distribution list through one or more FO Router nodes.

To ease bootstrapping, a connection to an FO Broker shall enable discovery of other Brokers and DDS participants.

#### 2.3.5.5.1.3.2.3.3 FO Node

'FO Nodes' behind the same FO Router rely on available DDS discovery protocol (Simple Participant and Endpoint Discovery protocol) and will rely on available multicast on LAN. All communication (publication/subscription) with external FO Nodes goes through the FO Router.

No direct visibility between FO Nodes under different FO routers is required.

### 2.3.5.5.2 Option pros and cons

Option	Pros	Cons
Current ED-133 Approach	<p>Validated in Iteration 1.1.0</p> <p>Interoperability based on already existing DDSI specification.</p> <p>Decentralized architecture.</p>	<p>Suffers several issues when it has to be deployed in the WAN scenario exposed in section 2.3.5.5.1.1 "WIM FO/IOP General Context":</p> <ul style="list-style-type: none"> <li>• No support for Source-Specific Multicast (SSM)</li> <li>• Flat-model with very high exchange of heartbeat information between SWIM Nodes.</li> <li>• Inefficient use of network resources (filtering done at the receiver side)</li> <li>• Relies on not-scalable DDS Discovery protocol</li> <li>• No dynamic adaptation to Path MTU</li> </ul> <p>It requires multicasting for scalability; but it is based on multicast technologies that are local to</p> <p>Unsecured (not resilient to malicious publishers).</p>
FO Overlay Network	<p>Decentralized and secured.</p> <p>Better scalability thanks to hierarchical architecture</p> <p>Efficient use of the network (use of available PIM-SM).</p> <p>Adapt to network protocols and MTU.</p>	<p>Final solution(s) require prototyping.</p> <p>Requires either enhancement of existing DDS standards, or SWIM-TI specific development.</p> <p>Require support for DDS vendors for early implementations of 'not-yet/draft' standards.</p>

Table 28 – Shared Object FB Architectural Options Pros and Cons

### 2.3.5.5.3 SWIM-TI Shared Object FB Architectural choice

Candidate solutions require prototyping by 14.02.09 and a feedback from the two major DDS vendors whose products are currently used by the 14.02.09 prototypes to select one or more viable solutions for a scalable and secure sharing of Flight Objects in European ATM beyond the context of VP-022 validation exercise.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

175 of 284

The following assumptions have been taken with regards to availability or not of a global addressing scheme:

- IPv6 architecture: A global addressing scheme for both unicast and multicast is available; so there is no need for a gateway device/application to perform Network Address Translation (NAT), for example.
- IPv4 architecture: There is no unique/global addressing scheme and each stakeholder has its own local addressing scheme for both unicast and multicast traffic. A dedicated device/gateway may be required to perform NAT, and forwarding of multicast traffic from local multicast address to a global multicast address and vice versa. The gateway is part of the stakeholder's network infrastructure and will not be part of the SWIM/FO layer. An example of such a gateway is an ANSP's firewall put in place to protect the ANSP's network infrastructure and systems/applications.

The total number of stakeholders participating in the IOP area from ED-133 is dimensioned to 50<sup>51</sup>. This performance metric may be used to help in selecting candidate solutions.

The following section presents possible architectural solutions based on available multicast services PIM-SM (ASM, SSM) in the core of the WAN network, the IP version v4/v6, and support for IGMPv2/v3 (IPv4) and MLDv2/v3 (IPv6) by the multicast routers on the ANSP/stakeholder's network.

Proposal	PIM-ASM	PIM-SSM	DDS Discovery	DDS Data	FO Publication	Summary	FO Cluster Publication (Distribution List: DL)
#1 (Dillon Model)	Yes	Yes	Many-to-many (ASM): scalable & interoperable solution from DDS vendors?  / Discovery Gateway	One group per data instance, SSM: DW to DRs	Many-to-many (ASM)  A single ASM group for the whole IOP Area (*,G) will need to be defined for all the FO Summaries.		One-to-many (SSM: FDMP to DL)  One multicast group per FO (SP-IOP- <i>Max_FO_Managed</i> addresses in total)
#2	Yes	Yes	Many-to-many (ASM): scalable & interoperable solution from DDS vendors?  / Discovery Gateway	DDS Partition per DL  Mapping of DDS Partition to a 'Network' partition.  SSM : DW to DRs	Many-to-many (ASM)  A single ASM group for the whole IOP Area (*,G) will need to be defined for all the FO Summaries.		One-to-many (SSM: FDMP to DL)  One multicast group per DDS Partition
#3	No	Yes	Discovery Gateway for DDS Participant & Endpoint Discovery (SSM and/or Unicast)	DDS Partition per DL  Mapping of DDS Partition to a 'Network' partition.  SSM : DW to DRs	One-to-many (SSM)  • FDMP to DL  • FO Summary Gateway to all (at least for those not in DL)		One-to-many (SSM: FDMP to DL)  One multicast group per DDS Partition

Table 29 – Summary of Architectural Proposals with impact on DDS

**NOTA**

<sup>51</sup> This is the number of locations or SWIM nodes. Each node may host multiple DDS nodes, due to redundancy for example. Each DDS node will contain multiple DataReaders and DataWriters.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



1. The proposals introduce the notion of work sets (or 'per instance subscription') with the help of the information provided within FO summary publication (distribution list). This does not require introducing the concept at the OMG DDS object model; but require support from DDS implementations (with possible extensions to DDSI) for fine grained control over joining and leaving multicast groups.
2. Current ED-133 approach requires sending of an FO summary (meta-data) with the FO clusters (FO data). First FO data for a member in the distribution list that has not yet joined the multicast group (the join will be performed after receipt of the FO Summary) will not reach the member. The member will request a republication of the FO data. This contradicts the original goal of only sending the FO data once; but this will only happen when a member is added to a distribution list. A way to avoid this situation is to either make the publishers (FDMP) publish their intentions first before publishing FO summary and data (similar to the FO summary but with no publication of FO data); or to add the work set concept to DDS.

For a detailed analysis of the candidate solutions refer to Appendix I.

### 2.3.5.5.3.1 Considerations for High Availability Architecture

The following section analyse impact of High Availability solutions to overall performance of the SWIM infrastructure for the distribution of Flight Objects.

Adding redundancy at a specific location (or SWIM node) introduces new DDS Readers/Writers to the overall DDS domain(s). Starting-up of a DataReader replica will result in all DataWriters publishing their latest data samples (combination of reliability, durability and destination order QoS) to align the new DataReader. This will result in a high volume of data at start-up to be transmitted over the network (WAN). **Adding a local replica will impact performance of all nodes with DataWriters.**

In order to implement the replica mechanism, data must be available locally (at the master/primary server). Feeding the new data reader with existing data shall be done locally using some **Node/Local-level State Transfer**. Figure 68 presents such architecture:

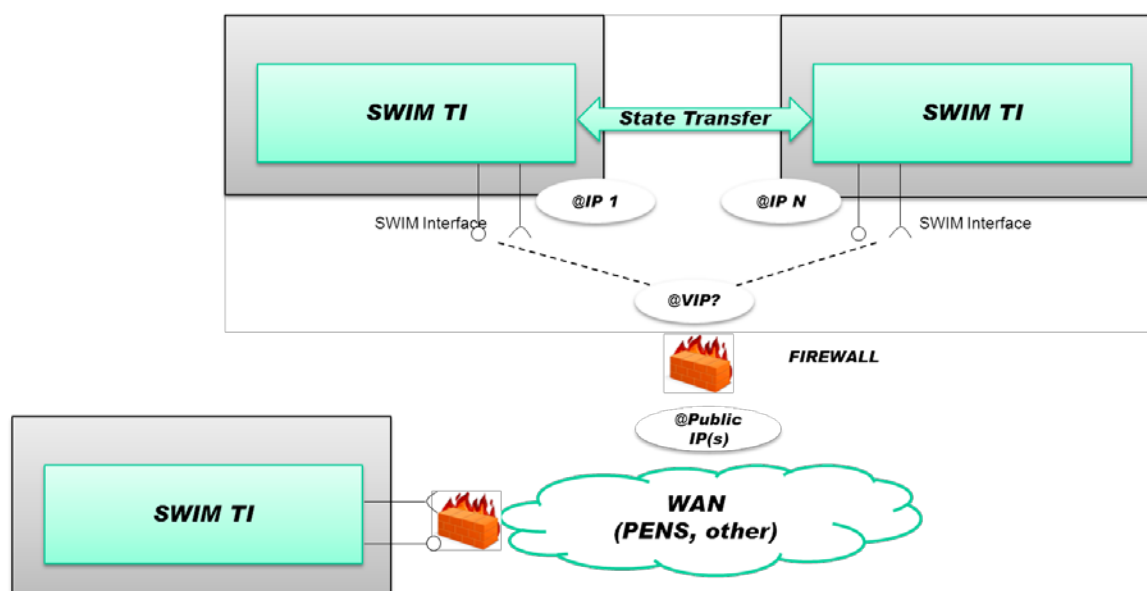


Figure 68 – General Architecture (one HA alternative from BU-09)

Data synchronization from the Wide Area Network may be sometimes necessary. A cold start (a start without prior state) at a certain location may require retrieval of Flight Objects from external publishers over the WAN.

### 2.3.5.5.3.2 OMG DDS Enhancements

The current OMG DDS specification including the interoperability specification (DDSI) shall address the following points.

#### 2.3.5.5.3.2.1 Efficient Discovery

To be able to exchange data samples within a DDS domain, **Readers** and **Writers** need to know about each other's existence and supported communication protocols. The protocol that enables **Participants** to obtain information about the existence and attributes of all the other **Participants** and **Endpoints** in the **Domain** is referred as **Discovery**.

Current DDS interoperability specification defines two discovery protocols, the Simple Participant Discovery Protocol (SPDP) and the Simple Endpoint Discovery Protocols (SEDP), for transparent plug-and-play dissemination of all the information needed to associate matching Writers and Readers.

The two protocols are referred to as the Simple Discovery Protocols. Though simple and efficient, the Simple Discovery Protocols are known to be not very scalable. The DDS Interoperability specification foresees a need to introduce other more scalable discovery mechanisms with more complex hierarchical mechanisms for Ultra-Large Scale systems.

The Simple Discovery Protocols are based on unicast and/or multicasting. Both unicast and/or multicast **Locators** describing available transport, address, and port combinations that can be used to send messages to the Endpoints. The multicast locators do not support multicast technologies such as SSM in use within Wide Area Networks such as PENS.

Some DDS vendors provide other discovery mechanisms with better scalability than the standard simple discovery protocols; but these do not interoperate with other DDS solutions from the other vendors.

#### 2.3.5.5.3.2.2 Limit visibility over Topics

Not all FO Clusters need to be seen by all the nodes! This can be handled by DDS technology provided there is control on which data instances need to be shared with which domain participant. Applications shall be able to specify **instance-level publications and/or subscriptions**.

Instead of using multiple topics, DDS Participants may use a single topic but state which data instances are of interest to them. Solutions shall perform any required filtering at the source level (Writer side) to minimize publications over the network.

#### 2.3.5.5.3.2.3 IP fragmentation

For efficient use of the network, it is important to limit the loss rate of data samples because lost data samples are resent what results in higher bandwidth usage and, therefore, in higher costs. This requires control of Maximum Transmission Unit (MTU) within IP based networks and **avoid IP fragmentation** (for security reasons, many firewalls block IP fragments, losing one single IP fragment on the WAN results in resending the entire UDP datagram). TCP-based protocols are known for their adaptability and avoid IP fragmentation altogether; but this is not the case with UDP-based DDS. It is desirable for DDSI specification to include **Path MTU discovery**<sup>52</sup> protocols.

<sup>52</sup> Such as <https://tools.ietf.org/html/rfc4821> describing a robust method for Path MTU Discovery (PMTUD) that relies on TCP or some other Packetization Layer that operates correctly without ICMP.

#### 2.3.5.5.3.2.4 Bandwidth limitation

It is also required to adapt publication rate to the bandwidth of the Wide Area Network which avoids bursts and subsequent resends of lost data samples. Some DDS vendors already provide some support for bandwidth limitation at the DDS level; but this is not very common.

#### 2.3.5.5.3.2.5 Data Compression

On Wide Area Networks with limited bandwidth (for technical or economic reasons), DDS implementations shall be able to compress data samples before their publications on the network. Some DDS products already support this; but as this is not part of the DDS Interoperability (DDSI) specification, they do not interoperate with other DDS products.

The DDSI specification shall support data compression.

#### 2.3.5.5.3.2.6 Security

Security is vital for large scale and open systems. It is necessary to secure access to Topics and exchange of data samples in an interoperable way (**DDS Security** specification at the OMG).

Firewalls are commonly used within the stakeholders' networks. Participants and Endpoint Discovery protocols shall support **NATing** and **Firewalls traversal** in an interoperable solution, i.e. a stakeholder does not need to be aware of other stakeholders' internal network topologies.

#### 2.3.5.5.3.2.7 Safety

DDS specification defines many QoS for limiting usage or resources. Current resource limits are defined in terms on (maximum) number of samples and instances. This is not sufficient to protect from large samples that may corrupt and/or break DDS applications. It is important to define a **maximum size for data samples** for a safe deployment of DDS-based applications. This will also improve security of the SWIM nodes as buffer overflow techniques may be used to get privileged access to remote nodes.

This does not need to be part of the DDSI specification; it is sufficient to have this supported locally by DDS products.

#### 2.3.5.5.3.2.8 Support for inter-domain multicast protocols

To support commonly used multicast routing protocols for Internet Protocol (IP) networks, such as those PIM<sup>53</sup> Source-Specific Multicast (PIM-SSM) , DDSI shall support use version 3 of the Internet Group Management Protocol (**IGMP v3**).

---

<sup>53</sup> Protocol-Independent Multicast (<http://tools.ietf.org/html/rfc1812>)

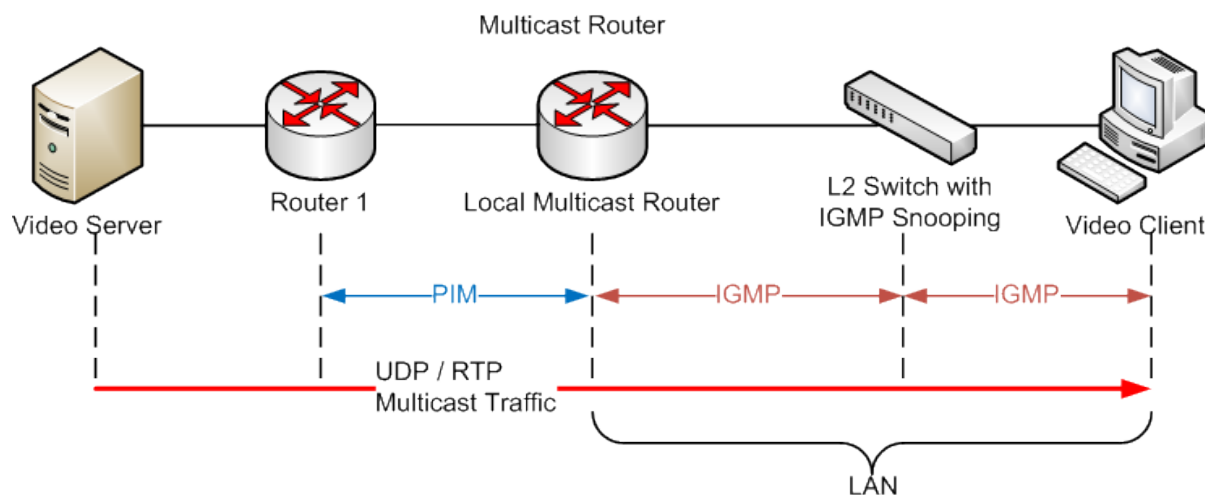


Figure 69 – IGMP basic architecture (source Wikipedia)<sup>54</sup>

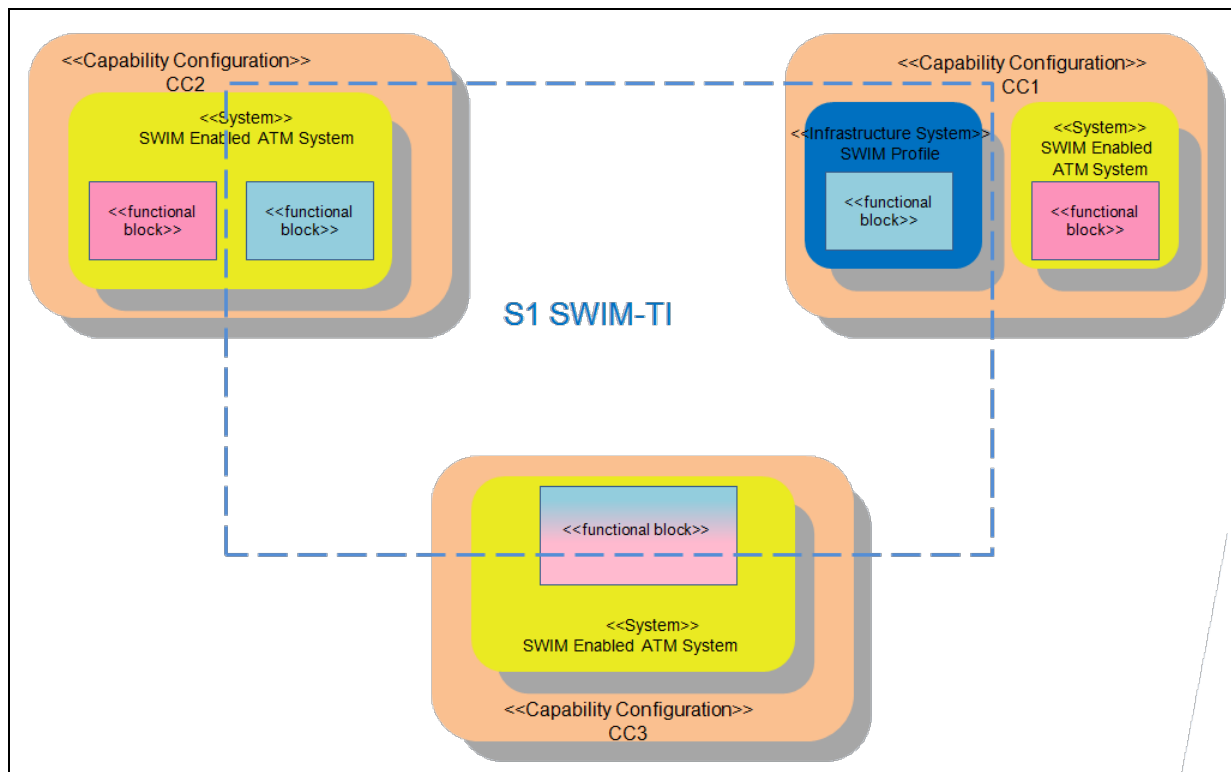
IGMP v3 support brings other requirements where each data writer shall be considered as a separate source for a multicast stream.

DDSI specification shall address both Participant and Endpoint Discovery and Data publication.

<sup>54</sup> [http://en.wikipedia.org/wiki/File:IGMP\\_basic\\_architecture.png](http://en.wikipedia.org/wiki/File:IGMP_basic_architecture.png)

## 2.4 Deployment Options

According to the SWIM ConOps (ref. [15]) and B4.3 ADD (ref. [6]), there are no fixed rules on how the SWIM-TI Functional Blocks are deployed to support an ATM System and so, the design is widely open. The following diagram summarizes the different possibilities<sup>5556</sup>:



**Figure 70 – SWIM-TI and different SWIM-TI Functional Block deployment options**

The diagram indicates the different degrees of evolution of the ATM Systems towards Systems with a service oriented architecture, to which SWIM-TI provides support. At the moment, three different options are foreseen:

**CC1:** There is no integration between Legacy System and SWIM TI. The SWIM Enabled ATM System provides ATM functionality and the SWIM-TI SWIM Profile enables the interchange of info between the SWIM Enabled ATM System and others.

**CC2:** The SWIM Enabled ATM System already incorporates the necessary SWIM-TI Functional Blocks to ensure the interchange of info with other SWIM Enabled ATM Systems. The set of SWIM-TI Functional Blocks need to be coherent to one or more SWIM-TI Profiles.

**CC3:** It represents the most evolved SWIM Enabled ATM System. On it, the communications between its Functional Blocks is already done via a SWIM-like mechanism, implementing a SWIM Profile. It doesn't need specific SWIM-TI Profiles or SWIM-TI Functional Blocks.

Appendix [A] provides some proposals on how to integrate SWIM into the ADD<sup>57</sup>.

<sup>55</sup> The diagram indicates S1 SWIM TI, however, this document focuses on Step 2 and the situation remains.

<sup>56</sup> For clarity, SWIM-TI Shared Infrastructure is not depicted. Its relationship with the Domain CCs doesn't exist, and so, depicting them in this picture doesn't add value.

<sup>57</sup> The integration with the overall European ATM Architecture Design B4.3 is not intended to be a part of a TAD document, but rather of an ADD (and hence, B.4.3 handled). So, appendix [A.3.3] should only be intended as a set of proposals/clarifications.

## 2.4.1 SWIM-TI Messaging

SWIM-TI Messaging is mostly based on COTS products and services, but it is possible that in some particular cases specific software may need to be developed. This depends on the information exchange needs and the SWIM-TI policies defined by the stakeholders.

### 2.4.1.1 Routing

#### 2.4.1.1.1 Multicast: Delegate-based approach

The idea is to partition the network in several Areas, according to geographic and volume of data exchanged between nodes. Since the exchanges between different Areas are expected to be much lower than the internal exchanges, it may be not worth to implement the routing between all the network nodes. Instead of considering this, it may be better to define a delegate for each Area in charge of inter-Area exchanges.

- One delegate by Area  
Areas for sets of nodes are defined, and a delegate is chosen for each Area. Each delegate would be responsible for sending the information to the delegate of the other Areas:

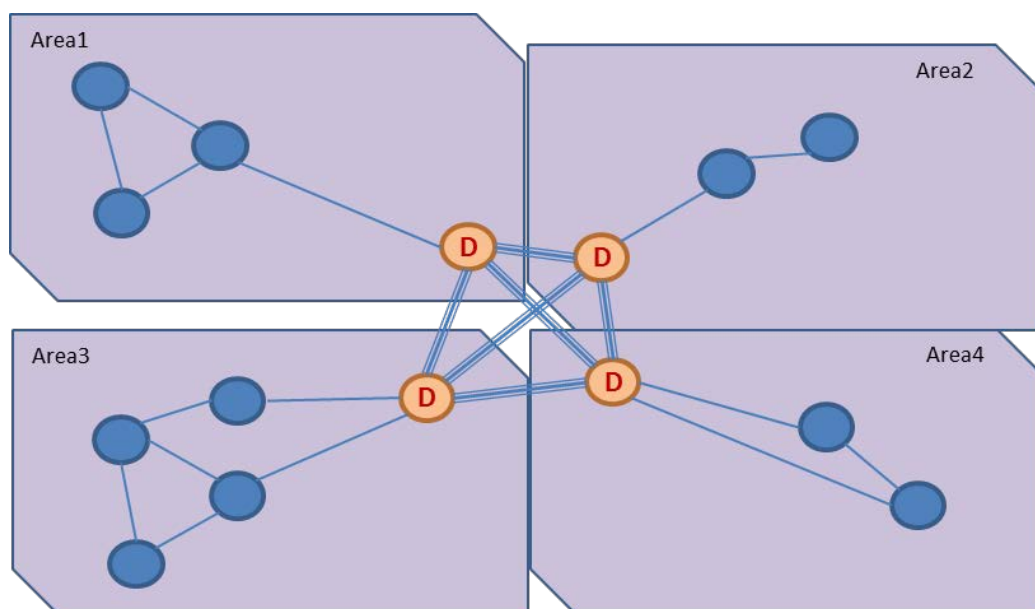


Figure 71 – Delegation by Area

- One delegate by Stakeholder and Area.
- Several delegates (one for each kind of stakeholder) can be defined in each Area:

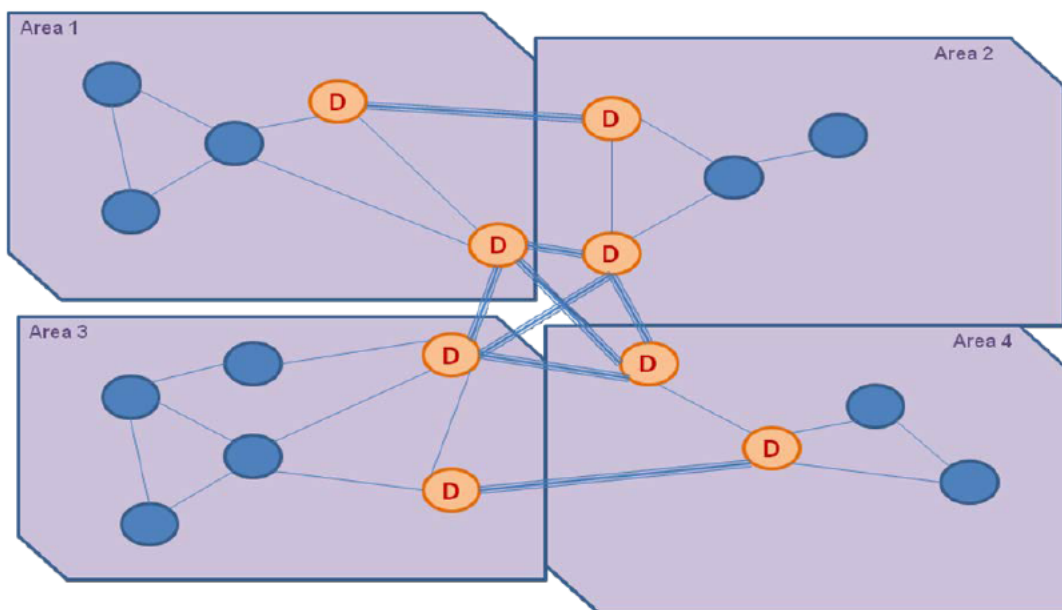


Figure 72 – Delegate by Stakeholder and Area

#### 2.4.1.1.2 Network-based approach

The Network Based approach<sup>58</sup> doesn't make use of the delegate approach relying on underlying network capabilities.

It makes use of both ASM and SSM multicasts, where ASM is used for information exchanges whose origin can be centralized in some way and to be shared among many SWIM Nodes; and SSM for the rest of exchanges. It fully exploits network capabilities though requires IGMP V3 to support SSM; and can be implemented on both IPV4 and IPV6.

The solution defines a control plane for setting up multicast routes and a data Plane for data delivery to only nodes in distribution list.

#### 2.4.1.1.3 Option pros and cons

Regarding Application Level Routing, only one option has been identified for the time being<sup>59</sup>.

The table below presents a comparative between the above mentioned Network Level Routing Multicast Architectural options:

Option	Pros	Cons
No delegated approach	Easier to manage.	Overload of the network.

<sup>58</sup> Dillon's model

<sup>59</sup> Further evolutions of Routing at Application Level could take place and derive in additional Architectural Options to be studied.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Option	Pros	Cons
One Delegate by Area	Scalability. It is possible to deploy new SWIM Nodes without difficulties.  Improvement in the routing performance, delegates know the SWIM Nodes in their Area.	A configuration for defining a delegate is needed and it has to change every time a new area is defined.  Single point of failure for each Area.
One Delegate by Stakeholder and Area	Scalability is much better.  Reduction of the Network Infrastructure bandwidth due to the reduction of exchanges between nodes of different ANSPs.  No single point of failure.	A configuration for defining a delegate is needed and it has to change every time a new area is defined. Can potentially affect the Network Infrastructure and SWIM Node configurations.  The choice of the delegates can have institutional considerations.
Network-based approach	Increased efficiency as it exploits network capabilities.	Requires ASM and thus IGMP v3.

Table 30 – Routing Deployment Options Pros and Cons

## 2.4.2 SWIM-TI Security

In this section, possible deployment options are introduced for the technical view described in 2.3.5.3.

### 2.4.2.1 Identity Management and Authentication

One of the key challenges for SWIM is to allow entities to access resources (services and information) across different security domains finding a balance between governance needs and flexibility of the technical solutions without jeopardize security. For such reason Identity Federation (or Federated Identity Management) Pattern can be taken in consideration as the target solution to realize access management in SWIM: Federation implies delegation of responsibilities honoured through trust relationships between federated parties.

The main starting point is that such interactions have to occur between entities belonging to different administrative (or security) domains (inter-domains) that both have their own PKI/STI. A solution to allow use/reuse of those elements in cross domain interaction is the Identity Federation or Federated Identity.

As defined in 2.3.5.3.1, case "a." relates to X.509 certificates and case "b." to Security Tokens. In case "a." the federation is based on trust relationships between CAs managing X.509 certificates and Certificate Revocation Lists (CRLs).

In case "b." the federation is achieved by federating the several STIs. When X.509 tokens are used, the federation of STIs includes the federation of PKIs they use and this is conceptually the same as case "a."

In a federated security model there is one logical Identity Store for each security domain and all the identity stores are federated (i.e. realize a logical user-identity store) through the federated identity provisioning and identity propagation features

There are three technology elements that are crucial to the concept of federation:

1. A federation protocol that enables parties to communicate.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



2. A flexible trust infrastructure that supports a variety of trust models.
3. An extensible policy management framework that supports differing governance requirements.

### 2.4.2.1.1 X.509 Certificates Based Technical View Deployment Options

A public key infrastructure (PKI) binds public keys to entities, enables other entities to verify public key bindings, and provides the services needed for ongoing management of keys in a distributed system. A PKI is made of Certification Authorities (CA) related by delegation relationship. A single PKI covers one Trust Domain.

A CA performs four basic PKI functions:

- issues certificates (i.e. creates and signs them);
- maintains certificate status information and issues Certificate Revocation Lists (CRL);
- publishes its current (e.g., unexpired) certificates and CRLs, so users can obtain the information they need to implement security services;
- and maintains archives of status information about the expired certificates that it issued.

#### 2.4.2.1.1.1 Hierarchical PKI architecture

Within an enterprise, Certification Authorities are usually organized as a hierarchical tree of related CAs. Root-CA issues certificates to subordinate CAs. Subordinate CAs issue certificates to CAs below them in the hierarchy, or end users.

The CRL is consolidated maintained and issued by the Root-CA. Current CRL is replicated in all subordinate CAs so that verification requests for public keys can be handled properly.

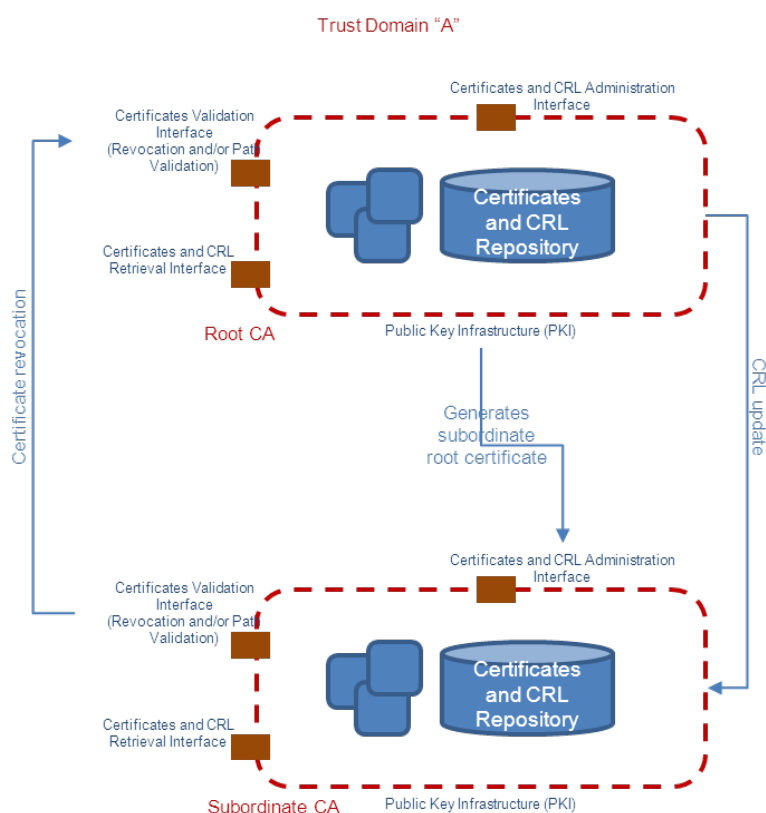


Figure 73 – Certification Authority delegation

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

### 2.4.2.1.1.2 Bridge CA architecture

The Bridge CA architecture is designed to connect enterprise PKIs regardless of the architecture. This is accomplished by introducing a new CA, called a Bridge CA, whose sole purpose is to establish relationships with enterprise PKIs.

The Bridge CA does not issue certificates directly to users. Unlike a root CA in a hierarchy, the Bridge CA is not intended for use as a trust point. All PKI users consider the Bridge CA an intermediary. The Bridge CA establishes peer-to-peer relationships with different enterprise PKIs. These relationships can be combined to form a bridge of trust connecting the trust domains from the different PKIs.

If the trust domain is implemented as a hierarchical PKI, the Bridge CA will establish a relationship with the Root-CA. The CA that enters into a trust relationship with the Bridge is called a principal CA.

In Figure 74 the Bridge CA has established relationships with two enterprise PKIs. The first is “Trust Domain A” CA, and the second is “Trust Domain B” hierarchical PKI. None of the end-users trusts the Bridge CA directly. End-user of “Trust Domain A” trusts the CA A1 that issued their certificates; she trusts the Bridge CA because the Root-CA A issued a certificate to it. End-user of “Trust Domain B” trusts point is the Root-CA B of his hierarchy; he trusts the Bridge CA because the Root-CA B issued a certificate to it. End-user of “Trust Domain A” can use the bridge of trust that exists through the Bridge CA to establish relationships with end-user of “Trust Domain B”.

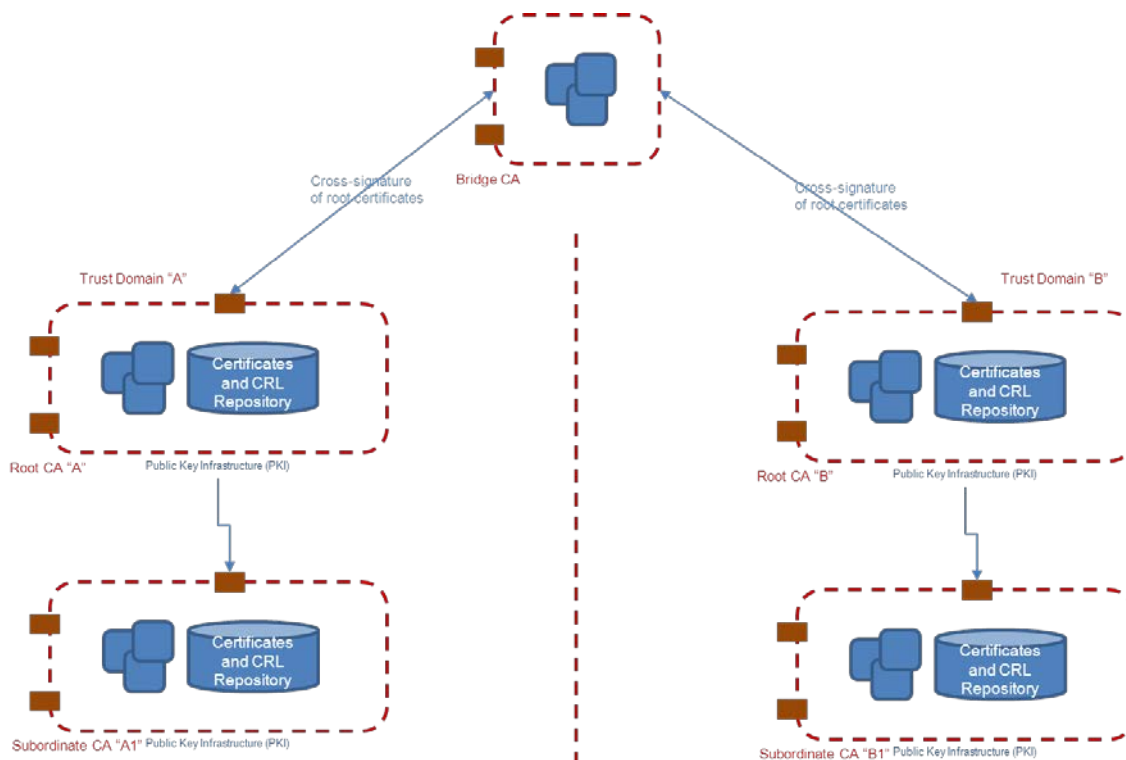


Figure 74 – Bridge CA architecture

In case a principal CA is comprised the trusted relationship with the Bridge CA is revoked. All the other “Trust Domains” can keep going securely without any business interruption.

### 2.4.2.1.1.3 Physical CA architecture

The figure below shows a typical architecture of a Certification Authority. For security reasons, it is preferable to deny all inbound traffic to the CA server and instead let the CA server periodically fetch and process information from external trusted data sources. For that purpose a dedicated protocol named Online Certificate Status Protocol (OCSP) has been defined. That is the reason why VA is also called OCSP-responder.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

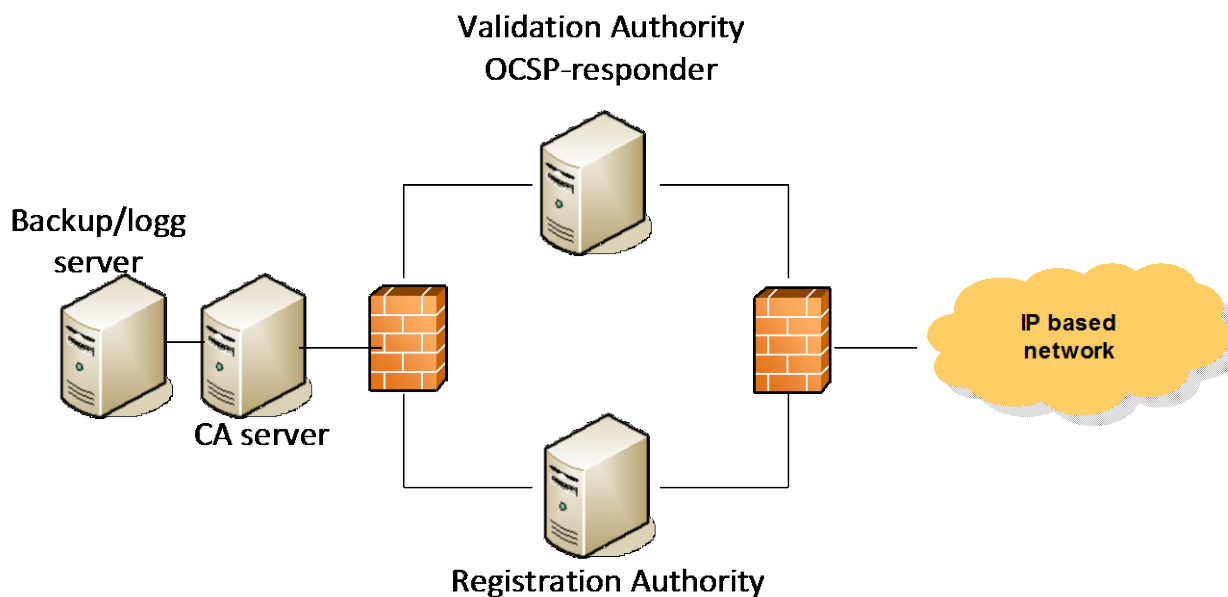


Figure 75 – Physical view of CA

#### 2.4.2.1.2 Security Tokens based Technical View Deployment Options

In this section several options concerning deployment view of the Security Token based Brokered Authentication are introduced. The starting point for deployment options is if the Federated Identity Management is required or not. If consumer and provider rely on the same STI the federation is not required; if they rely on different IdM separately administrated the federation is needed. In the last case there could be different federation options.

In SWIM business environment it is reasonable to assume that an organization (administrative domain) is expected to consume at least one service provided by a different organization: even if the consumer organization may have, for other ATM information exchanges, a single STI trusted by several organizations it is high probable that it is expected to establish a federation with others STI. It is therefore recommended that the federations features are not considered optional.

Hereafter several federation options are documented. They are based on WS and SAML patterns.

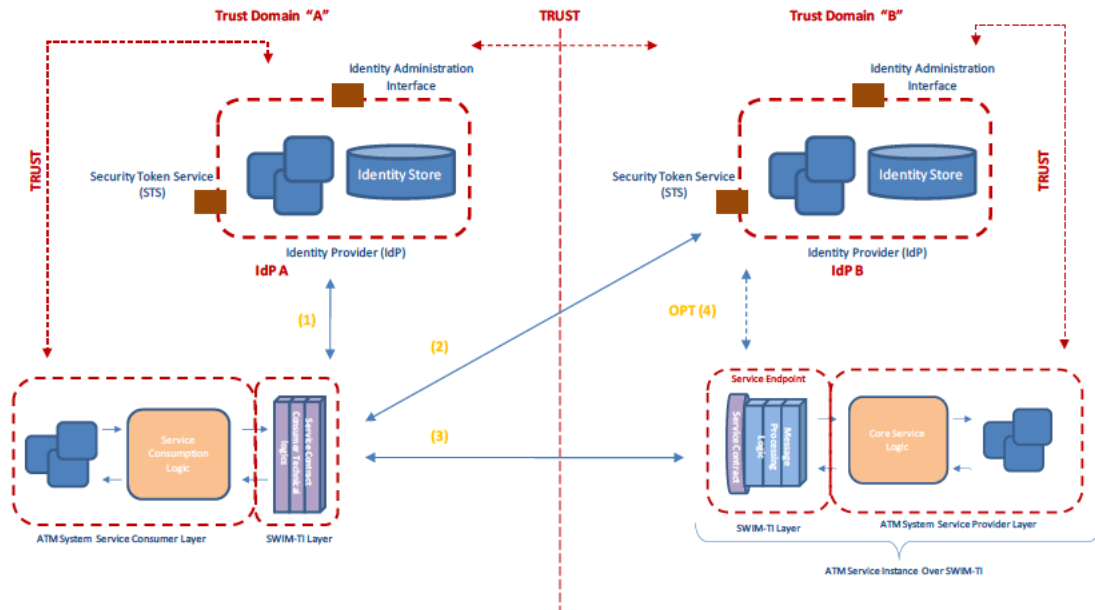


Figure 76 – Identity Federation Option 1

Option 1:

- This is required when the provider only trusts tokens issued by the IdP (STI) it trusts
- This introduces dependencies (and possible constraints) between consumer in one domain and the IdP (STI) in another domain. Furthermore the consumer is expected to authenticate twice (steps 1 and 2).

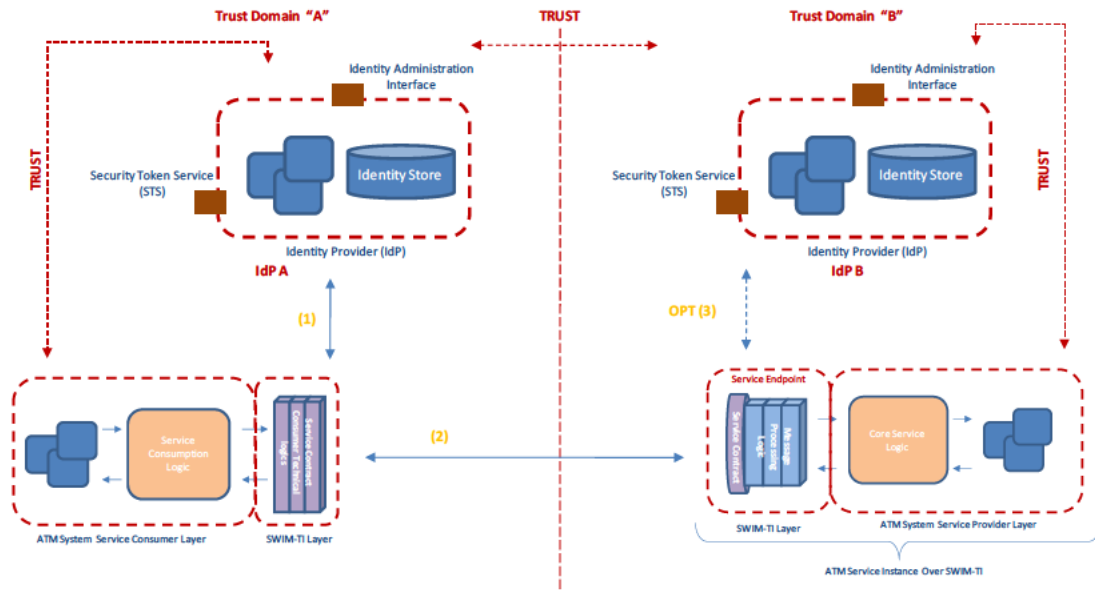


Figure 77 – Identity Federation Option 2

Option 2:

- solved the issues/difficulties identified in option 1.
- This introduces constraints and complexity (identity mapping and propagation) required to establish a trust relationships between two domains: it allows services on Trust Domain B to accept requests with identity information that has not been issued by IdP (STI) B.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

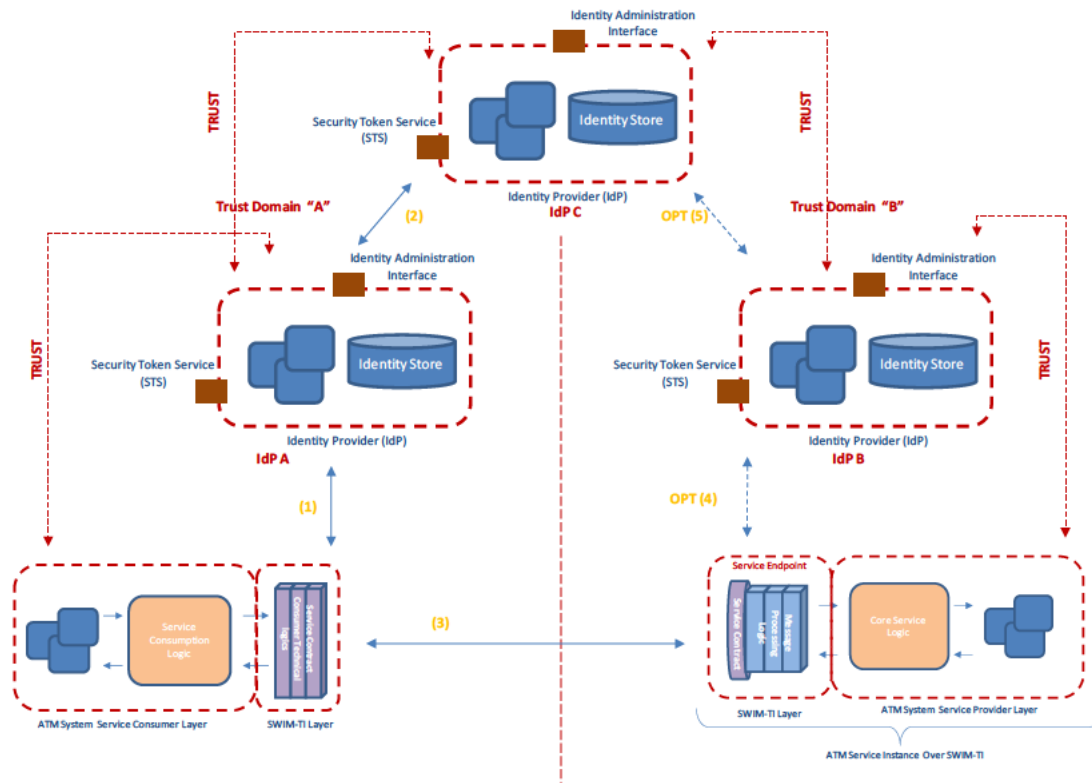


Figure 78 – Identity Federation Option 3

Option 3:

- the trust relationship between Domain A and B is achieved thanks to the fact that both domains trust IdP (STI) C.

The three options above could be properly composed but that introduces possible interoperability (technical and organizational level) issues, increasing of administrative complexity, possible loose of a real (federated) SSO (token issued in such option may be not accepted in other options). In case such possibility is allowed, proper rules governing such federation establishment could help in mitigating technical and organizational issues (e.g. independently of the option adopted, token issued in such option should be trusted also by domain using different options).

The figure below provides possible situation where different SWIM security domains are federated using heterogeneous options.

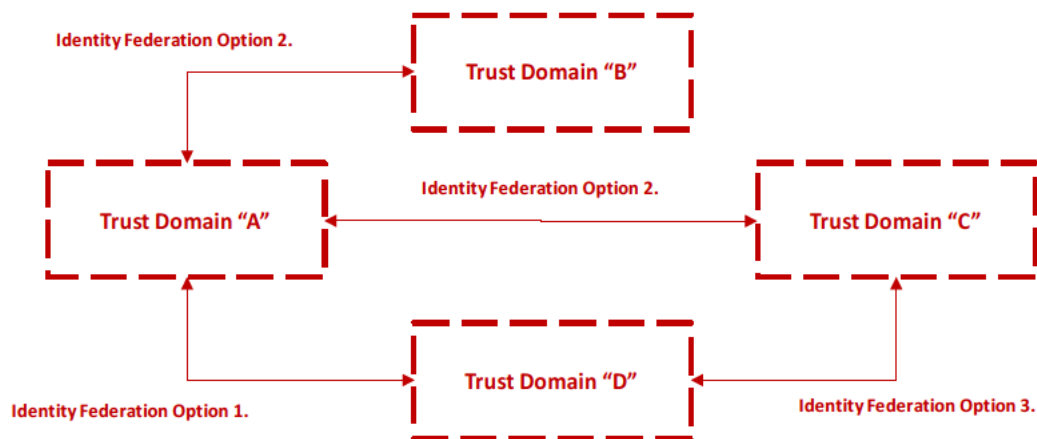


Figure 79 – Federation of SWIM-TI trust domains

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

In the figure above domain A has a trust relationship with domain B and domain C using option 2 whereas the trust relationship with domain D is established using option 1. Furthermore, domain D establishes a trust relationship with domain C using option 3. The possible issues are apparent.

There is a variety of security technologies associated with this technical view (just those authentication related):

- WS-Security Framework
  - WS-Security Core Specification.
  - Username Token Profile
  - X.509 Token Profile
  - SAML Token profile
  - WS-Policy
  - WS-SecurityPolicy
  - WS-Trust
  - **WS-Federation**
- Security Assertion Markup Language (SAML) 2.0
- Liberty Alliance

## 3 Enablers allocation

### 3.1 Allocation of ENs to functional blocks

The current analysis is based in the latest Data Set namely version “Integrated Roadmap – Dataset 16” [55].

Enabler code	Functional block identifier
A/C-57- Onboard migration from existing air-ground data link to air-ground SWIM for AIS/MET services	Messaging Security Supervision Recording
AGSWIM-34-New System AGDLGMS <sup>60</sup>	Messaging Security Supervision Recording
AGSWIM-41-AGDLGMS in support to provide extended OTIS to the aircraft	Messaging Security Supervision Recording
AGSWIM-43-AGDLGMS in support to provide weather information to the aircraft	Messaging Security Supervision Recording
AGSWIM-44-Transmission by AGDLGMS of the airborne Weather information to the meteo system for weather model improvement	Messaging Security Supervision Recording
ER APP ATC 160- ATC to ATC Flight Data Exchange Using Flight Object	Messaging Shared Object
GGSWIM-10c-SWIM Supervision for Step3	Supervision
GGSWIM-51c-SWIM Ground-ground messaging services in Step3	Messaging
GGSWIM-58c-SWIM registry in Step3	Registry
GGSWIM-59c-SWIM security in Step3	Security
SWIM-APS-05a- Provision and Consumption of Flight Object Sharing services for Step 1	Messaging Security Supervision Shared Object Recording

<sup>60</sup> To be noted that the notion of AGDLGMS has been discarded and currently, SWIM-TI A/G communications are expected to be handed by SWIM-TI Purple Profile.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

191 of 284

Enabler code	Functional block identifier
SWIM-APS-05b-Provision and Consumption of Flight Object Sharing services for Step 2	Messaging Security Supervision Shared Object Recording
SWIM-INFR-01a-High Criticality SWIM Services infrastructure Support and Connectivity	Messaging Security Supervision Shared Object Recording
SWIM-INFR-01b-High Criticality SWIM Services infrastructure Support and Connectivity	Messaging Security Supervision Shared Object Recording
SWIM-INFR-05a-General SWIM Services infrastructure Support and Connectivity	Messaging Security Supervision Shared Object Recording
SWIM-INFR-05b-General SWIM Services infrastructure Support and Connectivity	Messaging Security Supervision Shared Object Recording
SWIM-INFR-06b-AIR/GROUND SWIM Services infrastructure Support and Connectivity	Messaging Security Supervision Recording
SWIM-NET-01a - SWIM Network Point of Presence	Not applicable to technical infrastructure
SWIM-SUPT-01a-SWIM Supporting Registry Provisions	Registry
SWIM-SUPT-01b-SWIM Supporting Registry	Registry
SWIM-SUPT-03a-SWIM Supporting Security Provisions	Security
SWIM-SUPT-03b-SWIM Supporting Security	Security
SWIM-SUPT-05a - SWIM Supporting IP Network	Not applicable to technical infrastructure



Enabler code	Functional block identifier
Bridging Provisions	
SWIM-SUPT-06b-SWIM Supporting Supervision	Supervision <sup>61</sup>

**Table 31 – Allocation of ENs to functional blocks**

It is assumed that SWIM-INFR-01a and SWIM-INFR01b are referring Blue Profile. It is assumed that SWIM-INFR-05a and SWIM-INFR-05b are referring Blue and Yellow profiles.

## 3.2 Allocation of ENs to SYS primary projects

The **Table 32 – Allocation of ENs to SYS primary projects** below summarises the allocation of enablers to the WP14 system projects that is P14.02.01 - SWIM Middleware, P14.02.03 - SWIM technical supervision and P14.02.09 - SWIM Platform development and Demonstrator delivery. “P14.02.02 - SWIM Security solutions” is not concerned by the allocation as the project will not provide any prototype development.

The current analysis is based in the latest Data Set “Integrated Roadmap – Dataset 16” [55].

Enabler code	SYS primary project number
A/C-57- Onboard migration from existing air-ground data link to air-ground SWIM for AIS/MET services	Not in WP14 <sup>62</sup>
AGSWIM-34-New System AGDLGMS <sup>63</sup>	Not in WP14 <sup>64</sup>
AGSWIM-41-AGDLGMS in support to provide extended OTIS to the aircraft	Not in WP14 <sup>65</sup>
AGSWIM-43-AGDLGMS in support to provide weather information to the aircraft	Not in WP14 <sup>66</sup>
AGSWIM-44-Transmission by AGDLGMS of the airborne Weather information to the meteo system for weather model improvement	Not in WP14 <sup>67</sup>
ER APP ATC 160- ATC to ATC Flight Data Exchange Using Flight Object	P14.02.09
GGSWIM-10c-SWIM Supervision for Step3	P14.02.03
GGSWIM-51c-SWIM Ground-ground messaging services in Step3	P14.02.09

<sup>61</sup> Please note that only local supervision is addressed and therefore « remote » monitoring is not covered.

<sup>62</sup> 09.19

<sup>63</sup> To be noted that the notion of AGDLGMS has been discarded and currently, SWIM-TI A/G communications are expected to be handed by SWIM-TI Purple Profile.

<sup>64</sup> 09.19

<sup>65</sup> 09.19

<sup>66</sup> 09.19

<sup>67</sup> 09.19

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Enabler code	SYS primary project number
GGSWIM-58c-SWIM registry in Step3	Not addressed <sup>68</sup>
GGSWIM-59c-SWIM security in Step3	P14.02.09
SWIM-APS-05a- Provision and Consumption of Flight Object Sharing services for Step 1	P14.02.09
SWIM-APS-05b-Provision and Consumption of Flight Object Sharing services for Step 2	P14.02.09
SWIM-INFR-01a-High Criticality SWIM Services infrastructure Support and Connectivity	P14.02.09
SWIM-INFR-01b-High Criticality SWIM Services infrastructure Support and Connectivity	P14.02.09
SWIM-INFR-05a-General SWIM Services infrastructure Support and Connectivity	P14.02.09
SWIM-INFR-05b-General SWIM Services infrastructure Support and Connectivity	P14.02.09
SWIM-INFR-06b-AIR/GROUND SWIM Services infrastructure Support and Connectivity	P14.02.01
SWIM-NET-01a - SWIM Network Point of Presence	Not applicable to technical infrastructure
SWIM-SUPT-01a-SWIM Supporting Registry Provisions	Not addressed <sup>69</sup>
SWIM-SUPT-01b-SWIM Supporting Registry	Not addressed <sup>70</sup>
SWIM-SUPT-03a-SWIM Supporting Security Provisions	P14.02.09
SWIM-SUPT-03b-SWIM Supporting Security	P14.02.09
SWIM-SUPT-05a - SWIM Supporting IP Network Bridging Provisions	P14.02.09
SWIM-SUPT-06b-SWIM Supporting Supervision	P14.02.03 <sup>71</sup>

**Table 32 – Allocation of ENs to SYS primary projects**

<sup>68</sup> No system project is currently assuming the implementation of that enabler.

<sup>69</sup> No system project is currently assuming the implementation of that enabler.

<sup>70</sup> No system project is currently assuming the implementation of that enabler.

<sup>71</sup> Please note that only local supervision is addressed and therefore « remote » monitoring is not covered.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

## 4 SWIM Technical Infrastructure Design Principles

This section provides the design principles that have been considered to steer and sometime constraint the SWIM Technical Infrastructure architecture work.

The definition of SWIM, according to SWIM ConOps (ref. [11]) states that “SWIM consists of standards, infrastructure and governance enabling the management of ATM information and its exchange between qualified parties via interoperable services”.

The following chapters summarize the Needs and High Level objectives that raised the need for a System Wide Information Management and the Decisions/Assumptions that support the aforementioned SWIM.

### 1. Wide Access to Information at minimum cost

Increase in capacity is another major objective of SESAR and the overall ATM. The Performance Target (ref. [20]) stresses that “*the future ATM system for 2020 and beyond shall enable a 3-fold increase in capacity which will also reduce delays both on ground and in the air. The functional design of the system will have a coherent system wide information management system.*” This objective will lead to a steady growth in information sharing services, both in terms of number of users and of number of services.

This goal concerns the provision of wide access to ATM information at minimum cost and for a maximum number of stakeholders (Civil and Military).

- to enable maximum of partners to access its information at minimum cost for both Service Provider and Service Consumer;
- To ease up the access to SWIM services for the users – whatever it is a new service or a new user.

When mixing together “wide access” and “low cost” it is obvious that this drives the choice for delivering the SWIM services through Web technologies: web services and requiring minimal or none deployment of SWIM software at the customer site.

### 2. Follow Technological (R)evolutions

One of the biggest technological revolutions (or the biggest) in the last decades was the appearance of the Internet and the subsequent development of Web related technologies.

The evolution of the Web into Web 2.0, further develops the concept of “network as the platform” and empowers the end-user: It is considered strategic to follow and adopt the technologies behind Web 2.0 and to take best advantage of the technical evolution in the definition of the SWIM Technical Infrastructure and associated services. This will support requirements for better security, better data coherence and consistency as well as for inter-operability, maintainability and connectivity.

This implies that the technologies for the delivery of the ATM information should then favour Web related technologies.

### 3. Cost Efficiency

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

195 of 284

Cost efficiency is a major objective for SESAR and the overall ATM. The Performance Target (ref. [20]) stresses that *“in response to the ATM challenges, ATM services shall be provided at a cost to the airspace users which is at least 50% less than today”*.

The continuous improvement sought in cost efficiency imposes to constantly revisit the way the information services are designed, procured, deployed and supported. A number of possible deployment strategies could be imagined:

- Each stakeholder (airline, Airport, ANSP etc.) builds its own local SWIM Technical Infrastructure solution;
- Each stakeholder purchases a SWIM Technical Infrastructure solution from a SWIM-ware vendor;
- Each ATM system vendor builds a SWIM Technical Infrastructure into its turn-key systems;
- European-wide common/shared software procurement (similar to PENS);
- A stakeholder or community of stakeholders outsource the SWIM Technical Infrastructure and possibly application level service provision to a third partner.

This objective is reflected in the SWIM Technical Infrastructure architecture in the following ways:

- SWIM Technical Infrastructure shall be based upon well-recognized or emerging IT standard that are supported by mainstream IT COTS product in the market
- SWIM Technical Infrastructure shall use preferably IT COTS products that implement those IT standards and only require little or no further development/customisation;
- The use of marginal technologies, proprietary solutions or large amounts of software development should be avoided since it would render certain deployment/running strategies impractical (due to their cost).

#### 4. Support Increasing Number of Users and Services

Increase in capacity is another major objective of SESAR and the overall ATM. The Performance Target (ref. [20]) stresses that *“the future ATM system for 2020 and beyond shall enable a 3-fold increase in capacity which will also reduce delays both on ground and in the air. The functional design of the system will have a coherent system wide information management system.”* This objective will lead to a steady growth in information sharing services, both in terms of number of users and of number of services.

This goal drives the SWIM Technical Infrastructure architecture in the sense that it must be scalable.

### 4.1 Design principles

In order to meet the Wide Information Access at low cost for a large and increasing number of Users and Services, the following system design principles are applied in the overall SWIM Technical Infrastructure architecture. These principles assume both ADD (ref. [6]) and SWIM ConOps (ref. [11]) and well known Service Oriented Architecture sources [22].

SWIM-TI Design assumes the 10 Key Principles identified in the SWIM ConOps (ref. [11]) for the System Wide Information Management. These principles are the following:

Principle	Title	Description
-----------	-------	-------------

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

196 of 284

Principle	Title	Description
<b>SWIM ConOps PRINCIPLE-1</b>	Accessibility	ATM stakeholders can directly offer, and consume, ATM information using common service interfaces and network connectivity.
<b>SWIM ConOps PRINCIPLE-2</b>	Equity	No individual stakeholder dominates, or constrains, what may be offered, or consumed, by other stakeholders.
<b>SWIM ConOps PRINCIPLE-3</b>	Flexibility	Capability for adequate, responsive, timely, dynamic and asynchronous changes of providers and users and the information and services they offer and consume.
<b>SWIM ConOps PRINCIPLE-4</b>	Performance	Combined ATM stakeholder and infrastructure provisions must ensure required levels of performance, safety, and resilience, and provide effective incident and evolution management.
<b>SWIM ConOps PRINCIPLE-5</b>	Quality, Integrity & Security	ATM Stakeholders retain responsibility for the quality, integrity, security, and availability of the information, whilst interface and infrastructure technology ensures integrity of exchanges. ATM Stakeholder identification allows information to only be exchanged with appropriate parties and security measures applied.
<b>SWIM ConOps PRINCIPLE-6</b>	Implementation & Evolution	Clear vision and roadmap for operational, technical, and institutional, implementation and evolution; aligned with reduction in the use of individually specialised interfaces and connectivity.
<b>SWIM ConOps PRINCIPLE-7</b>	Cost	Reduced costs for on-going information exchanges, with costs for ATM evolution proportionate to needs, benefits and stakeholder affordability.
<b>SWIM ConOps PRINCIPLE-8</b>	Service orientation	Service orientation methods are expected to be applied to support the ATM stakeholders' use of services to share information.
<b>SWIM ConOps PRINCIPLE-9</b>	Open standards	SWIM is expected to make use of applicable open and internationally recognized standards for the information, the content, the processes, and the provision of services.
<b>SWIM ConOps PRINCIPLE-10</b>	Global applicability	SWIM will need both international and local agreements, to achieve a seamless ATM information environment and therefore adequate governance needs to be established.

Table 33 – SWIM ConOps SWIM 10 Key Principles

Besides this Key Principles, SWIM-TI incorporates/specializes the following Design Principles:

- **Fit for purpose.** SWIM Technical Infrastructure Design and Artifacts must be an answer to business and operational needs. This will mean that solutions provided by SWIM Technical Infrastructure must be well based either in existing needs, either in future needs. This Principle applies not only to the SESAR Programme but also to further evolutions of the SWIM-TI Design after the R&D Stages; **(SWIM-TI Design PRINCIPLE-1)**
- **Aim for the interoperability.** SWIM Technical Infrastructure Design must always handle the interoperability between the existing architectural solutions and the new to be designed; **(SWIM-TI Design PRINCIPLE-2).**

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- **Aim for the functionality / implementation freedom.** SWIM Technical Infrastructure Design must be follow as much as possible functionality description approach, rather than a physical description approach, leaving the implementer the freedom of implementation as long as the requirements are covered and the interoperability maintained; **(SWIM-TI Design PRINCIPLE-3)**.
- **Distributed/Modular design.** SWIM Technical Infrastructure Design must always support a distributed and/or federated approach; **(SWIM-TI Design PRINCIPLE-4)**
- **Scalability.** SWIM Technical Infrastructure Design must ensure the scalability of the proposed architectural Artifacts and Solutions; **(SWIM-TI Design PRINCIPLE-5)**
- **Standards.** SWIM Technical Infrastructure Design must use of standard technologies where applicable. In case of developed software, WP14 is expected to deliver primarily domain independent solutions. If a need for a domain specific software development as part of the SWIM Technical Infrastructure is identified and agreed, it shall be implemented respecting a modular approach (the generic part of the SWIM Technical Infrastructure can be used independently from the domain specific part for ATM software applications which do not need this domain specific part). **(SWIM-TI Design PRINCIPLE-6)**
- **Formal Models.** SWIM Technical Infrastructure Design Artifacts must make use of formal models (such as UML) to express system interfaces; **(SWIM-TI Design PRINCIPLE-7)**
- **Service Orientation.** SWIM Technical Infrastructure Design must apply the concepts behind service-orientation<sup>72</sup> and provide monitoring mechanisms at service level in the design of every component; **(SWIM-TI Design PRINCIPLE-8)**
- **Reuse.** SWIM Technical Infrastructure Design must reuse existing components when possible; I reusability in general should be considered as one of major goals in order to reduce costs; **(SWIM-TI Design PRINCIPLE-9)**

## 4.2 Design Decisions: Architectural Choices vs. Design Principles

It's expected that the Architectural Choices align with the Design Principles in this chapter and with the SWIM ConOps (ref. [11]) SWIM Principles.

The table below justifies the Design Decisions as Architectural Choices via identifying which Design Principles apply to which Architectural choices of those described in chapter 2.2.5:

SWIM-TI FB	Architectural Choice	Design Principles applied
------------	----------------------	---------------------------

<sup>72</sup> Services and service Orientation are defined by WP08 and WPB.

SWIM-TI FB	Architectural Choice	Design Principles applied
SWIM-TI FB Registry	Root registry with consumer affiliate	<p>SWIM ConOps PRINCIPLE-1                      SWIM ConOps PRINCIPLE-2                      SWIM ConOps PRINCIPLE-3                      SWIM ConOps PRINCIPLE-4                      SWIM ConOps PRINCIPLE-5                      SWIM ConOps PRINCIPLE-6                      -                      SWIM ConOps PRINCIPLE-8                      -                      SWIM ConOps PRINCIPLE-10</p> <p>SWIM-TI Design PRINCIPLE-1                      -                      SWIM-TI Design PRINCIPLE-3                      SWIM-TI Design PRINCIPLE-4                      SWIM-TI Design PRINCIPLE-5                      -                      SWIM-TI Design PRINCIPLE-7                      SWIM-TI Design PRINCIPLE-8                      -</p>
SWIM-TI Messaging FB - Blue Profile	<p>Routing Network Level – No delegated approach</p> <p>Limit bandwidth usage</p> <p>Use compression</p> <p>No IP fragmentation</p> <p>Efficient use of the network</p> <p>Favour filtering at source level</p>	<p>SWIM ConOps PRINCIPLE-1                      SWIM ConOps PRINCIPLE-2                      SWIM ConOps PRINCIPLE-3                      -                      SWIM ConOps PRINCIPLE-5                      -                      SWIM ConOps PRINCIPLE-7                      -                      SWIM ConOps PRINCIPLE-9                      SWIM ConOps PRINCIPLE-10</p> <p>SWIM-TI Design PRINCIPLE-1                      SWIM-TI Design PRINCIPLE-2                      SWIM-TI Design PRINCIPLE-3                      -                      -                      SWIM-TI Design PRINCIPLE-6                      SWIM-TI Design PRINCIPLE-7                      -                      SWIM-TI Design PRINCIPLE-9</p>

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

SWIM-TI FB	Architectural Choice	Design Principles applied
SWIM-TI Messaging FB - Yellow Profile	Routing Application Level – Broker Federation	<b>SWIM ConOps PRINCIPLE-1</b> <b>SWIM ConOps PRINCIPLE-2</b> <b>SWIM ConOps PRINCIPLE-3</b> <b>SWIM ConOps PRINCIPLE-4</b> - - <b>SWIM ConOps PRINCIPLE-7</b> <b>SWIM ConOps PRINCIPLE-8</b> <b>SWIM ConOps PRINCIPLE-9</b> <b>SWIM ConOps PRINCIPLE-10</b>  <b>SWIM-TI Design PRINCIPLE-1</b> <b>SWIM-TI Design PRINCIPLE-2</b> <b>SWIM-TI Design PRINCIPLE-3</b> - <b>SWIM-TI Design PRINCIPLE-5</b> <b>SWIM-TI Design PRINCIPLE-6</b> <b>SWIM-TI Design PRINCIPLE-7</b> <b>SWIM-TI Design PRINCIPLE-8</b> -
SWIM-TI Messaging FB - Purple Profile	Routing Application Level – Broker Federation	<b>SWIM ConOps PRINCIPLE-1</b> <b>SWIM ConOps PRINCIPLE-2</b> <b>SWIM ConOps PRINCIPLE-3</b> <b>SWIM ConOps PRINCIPLE-4</b> - - <b>SWIM ConOps PRINCIPLE-7</b> <b>SWIM ConOps PRINCIPLE-8</b> <b>SWIM ConOps PRINCIPLE-9</b> <b>SWIM ConOps PRINCIPLE-10</b>  <b>SWIM-TI Design PRINCIPLE-1</b> <b>SWIM-TI Design PRINCIPLE-2</b> <b>SWIM-TI Design PRINCIPLE-3</b> - <b>SWIM-TI Design PRINCIPLE-5</b> <b>SWIM-TI Design PRINCIPLE-6</b> <b>SWIM-TI Design PRINCIPLE-7</b> <b>SWIM-TI Design PRINCIPLE-8</b> -

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)



SWIM-TI FB	Architectural Choice	Design Principles applied
Security FB - Blue Profile	<p>Access Control – Federated Brokered Authentication</p> <p>Data Protection – Data Origin Authentication and Confidentiality Ensuring</p>	<p><b>SWIM ConOps PRINCIPLE-1</b>  <b>SWIM ConOps PRINCIPLE-2</b>  <b>SWIM ConOps PRINCIPLE-3</b>  <b>SWIM ConOps PRINCIPLE-4</b>  <b>SWIM ConOps PRINCIPLE-5</b>                      -                      -                      -  <b>SWIM ConOps PRINCIPLE-9</b>  <b>SWIM ConOps PRINCIPLE-10</b></p> <p><b>SWIM-TI Design PRINCIPLE-1</b>  <b>SWIM-TI Design PRINCIPLE-2</b>  <b>SWIM-TI Design PRINCIPLE-3</b>  <b>SWIM-TI Design PRINCIPLE-4</b>  <b>SWIM-TI Design PRINCIPLE-5</b>  <b>SWIM-TI Design PRINCIPLE-6</b>  <b>SWIM-TI Design PRINCIPLE-7</b>                      -  <b>SWIM-TI Design PRINCIPLE-9</b></p>
Security FB – Yellow Profile	<p>Access Control – Federated Brokered Authentication</p> <p>Data Protection – Data Origin Authentication and Confidentiality Ensuring</p>	<p><b>SWIM ConOps PRINCIPLE-1</b>  <b>SWIM ConOps PRINCIPLE-2</b>  <b>SWIM ConOps PRINCIPLE-3</b>  <b>SWIM ConOps PRINCIPLE-4</b>  <b>SWIM ConOps PRINCIPLE-5</b>                      -                      -                      -  <b>SWIM ConOps PRINCIPLE-9</b>  <b>SWIM ConOps PRINCIPLE-10</b></p> <p><b>SWIM-TI Design PRINCIPLE-1</b>  <b>SWIM-TI Design PRINCIPLE-2</b>  <b>SWIM-TI Design PRINCIPLE-3</b>  <b>SWIM-TI Design PRINCIPLE-4</b>  <b>SWIM-TI Design PRINCIPLE-5</b>  <b>SWIM-TI Design PRINCIPLE-6</b>  <b>SWIM-TI Design PRINCIPLE-7</b>                      -  <b>SWIM-TI Design PRINCIPLE-9</b></p>

SWIM-TI FB	Architectural Choice	Design Principles applied
Security FB – Purple Profile	<p>Access Control – Federated Brokered Authentication</p> <p>Data Protection – Data Origin Authentication and Confidentiality Ensuring</p>	<p>SWIM ConOps PRINCIPLE-1 SWIM ConOps PRINCIPLE-2 SWIM ConOps PRINCIPLE-3 SWIM ConOps PRINCIPLE-4 SWIM ConOps PRINCIPLE-5 - - - SWIM ConOps PRINCIPLE-9 SWIM ConOps PRINCIPLE-10</p> <p>SWIM-TI Design PRINCIPLE-1 SWIM-TI Design PRINCIPLE-2 SWIM-TI Design PRINCIPLE-3 SWIM-TI Design PRINCIPLE-4 SWIM-TI Design PRINCIPLE-5 SWIM-TI Design PRINCIPLE-6 SWIM-TI Design PRINCIPLE-7 - SWIM-TI Design PRINCIPLE-9</p>
Supervision FB	- <sup>73</sup>	-
Recording FB	- <sup>74</sup>	-
Shared Object FB	<p>Decentralized and secured</p> <p>Hierarchical P2P architecture</p>	<p>SWIM-TI Design PRINCIPLE-1 SWIM-TI Design PRINCIPLE-2 SWIM-TI Design PRINCIPLE-4 SWIM-TI Design PRINCIPLE-5 SWIM-TI Design PRINCIPLE-6</p>

Table 34 – Architectural Choices vs. Design Principles

<sup>73</sup> Decision not taken at Iteration 3.1

<sup>74</sup> Decision not taken at Iteration 3.1

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

## 5 References<sup>75</sup>

### 5.1 Applicable Documents

- [1] EUROCONTROL ATM Lexicon  
<https://extranet.eurocontrol.int/http://atmlexicon.eurocontrol.int/en/index.php/SESAR>
- [2] SESAR PMP, Edition 03.00.02
- [3] SESAR Architecture and Technical Systems Strategy, Edition 02.00.00, 15/05/2011
- [4] SESAR System Thread Guidance, Edition 04.00.00, 20/08/2014
- [5] SESAR Technical Architecture Description template, Edition 03.00.00, 08/05/2012
- [6] B.04.03-D98 ADD SESAR 2020 Transition edition, v00.04.02, 09/02/2016

### 5.2 Reference Documents

The following documents were used to provide input / guidance / further information / other:

- [7] Study on SWIM Civil-Military Interoperability, D1: Characteristics of Military ATM and AD/C2 systems and the justification for their interoperability with SWIM, v0.85, 10/04/2012
- [8] Study on SWIM Civil-Military Interoperability, D2. Target SWIM Interoperability Concept and Architecture, v0.96, 10/04/2012
- [9] 14.02.09-D03, SWIM Technical Infrastructure Definition, Edition 00.01.02, 19/09/2011
- [10] 14.01.03-D39-, SWIM Profiles Final, Edition 00.01.00
- [11] 08.01.01-D42, SWIM Concept of Operations, Edition 00.04.05, 06/05/2014
- [12] 08.01.01-D53, SWIM Registry Concept of Operations V2, Edition 00.02.02
- [13] 14.01.04-D44-001, SWIM-TI Technical Specifications Catalogue
- [14] D3, The ATM Target Concept, September 2007
- [15] 08.03.01-D14, Concept of Operations for SWIM Supervision, Edition 02.00.01
- [16] B4.3-D81, SESAR Working Method on Services, Edition 2013 00.04.01
- [17] 14.01.02-D04, Ground/Ground Technology & Service Option Survey (Step2), Edition 00.01.00
- [18] 14.01.02-D07, Ground/Ground Technology & Service Option Survey - Final Report (Step2), Edition 00.01.00
- [19] 08.03.10-D64 ISRM Service Portfolio, Edition 00.07.01
- [20] D2, The Performance Target, December 2006
- [21] 14.01.02-D03, SWIM Context Definition, v00.01.00, 24/11/2010
- [22] "Improve your SOA project plans",  
<http://www.ibm.com/developerworks/webservices/library/ws-improvesoa/> , 24/07/ 2004
- [23] [http://en.wikipedia.org/wiki/OSI\\_model](http://en.wikipedia.org/wiki/OSI_model)

- [24] <http://www.enterpriseintegrationpatterns.com/MessageRoutingIntro.html>
- [25] [http://soapatterns.org/candidate\\_patterns/federated\\_enterprise\\_authentication](http://soapatterns.org/candidate_patterns/federated_enterprise_authentication)
- [26] [http://soapatterns.org/design\\_patterns/data\\_origin\\_authentication](http://soapatterns.org/design_patterns/data_origin_authentication)
- [27] P14.01.03 D031 SWIM Architectural Definition for Iteration 2.0
- [28] MSG FB Doc -  
[https://extranet.sesarju.eu/WP\\_14/Project\\_14.01.04/Other%20Documentation/Execution%20Phase/Iteration%202.1/TechnicalActivities/Messaging%20improvements\(IT2.1-40\)/INT14.01.34-MSG%20FB%20Description.doc](https://extranet.sesarju.eu/WP_14/Project_14.01.04/Other%20Documentation/Execution%20Phase/Iteration%202.1/TechnicalActivities/Messaging%20improvements(IT2.1-40)/INT14.01.34-MSG%20FB%20Description.doc)
- [29] SEC FB Doc -  
[https://extranet.sesarju.eu/WP\\_14/Project\\_14.01.04/Other%20Documentation/Execution%20Phase/Iteration%202.1/TechnicalActivities/Security%20Improvements\(IT2.1-5\)/INT14.01.34-SEC%20FB%20Description.doc](https://extranet.sesarju.eu/WP_14/Project_14.01.04/Other%20Documentation/Execution%20Phase/Iteration%202.1/TechnicalActivities/Security%20Improvements(IT2.1-5)/INT14.01.34-SEC%20FB%20Description.doc)
- [30] Data Distribution Service for Real-time Systems (DDS), v1.2, January 2007,  
<http://www.omg.org/spec/DDS/1.2>.
- [31] DDS Interoperability Wire Protocol, V2.1, OMG Standard document formal/2010-11-01,  
November 2010, <http://www.omg.org/spec/DDS-RTPS/2.1>.
- [32] Eurocae WG59, ED-133 Flight Object interoperability specification, June 2009.
- [33] A DDS Discovery Protocol  
based on Bloom filters, Master Thesis  
, Javier Sánchez Monedero,, Granada, 14th September 2009  
, [http://dtstc.ugr.es/tl/pdf/pf/PFM\\_jsanchez\\_electronic-EN.pdf](http://dtstc.ugr.es/tl/pdf/pf/PFM_jsanchez_electronic-EN.pdf)
- [34] Architecture de Communication à QoS Garantie pour la Simulation Distribuée  
, Akram HAKIRI, Thèse, 13/07/2012  
, <http://thesesups.ups-tlse.fr/1665/1/2012TOU30055.pdf>,
- [35] Scaling the Data Distribution Service to Global Networks, Angelo Corsaro, Ph.D., PrismTech,  
Sara Tucci-Piergiovanni, Ph.D., University of Rome "La Sapienza",  
<http://www.omg.org/news/meetings/GOV-WS/pr/rte-pres/ultra-large-scale-dds.pdf>.
- [36] Scaling DDS to Millions of Computers and Devices, Rick Warren, RTI,  
<http://www.omg.org/news/meetings/realtime2011/presentations/ScalingDDS.pdf> .
- [37] C. Kent and J. Mogul. Fragmentation Considered Harmful. Proceedings of Frontiers in  
Computer Communications technology, ACM SIGCOMM'87, Stowe, Vermont, August, 1987.
- [38] OASIS Reference Architecture Foundation for Service Oriented Architecture, Version 1,  
December 2012.
- [39] SOA Patterns, <http://arnon.me/soa-patterns/>
- [40] SOA Design Patterns, <http://www.soapatterns.org/>
- [41] ORACLE, Securing the SOA Landscape ,  
<http://www.oracle.com/technetwork/articles/soa/ind-soa-6-security-1980268.html>
- [42] Service Technology Magazine, Securing the SOA Landscape ,  
<http://servicetechmag.com/l74/0713-3>

- [43] Open Group, Open Enterprise Security Architecture (O-ESA) Standard ,  
<https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?publicationid=12380>
- [44] Service Orientation, Service Layers ,  
[http://serviceorientation.com/soamethodology/service\\_layers](http://serviceorientation.com/soamethodology/service_layers)
- [45] OASIS, Glossary for the OASIS SecurityAssertion Markup Language, 2005
- [46] IETF Internet Security Glossary RFC 2828
- [47] IETF Terminology for Policy-Based Management RFC 3198
- [48] IETF A Framework for Policy-based Admission Control RFC 2753
- [49] Distributed Systems, Principles and Paradigms, Second Edition, Andrew S. Tanenbaum, Maarten
- [50] Distributed Systems, Concepts and Design, Fifth Edition, George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair
- [51] The Many Faces of Publish/Subscribe, PATRICK TH. EUGSTER, PASCAL A. FELBER, RACHID GUERRAOUI, ANNE-MARIE KERMARREC
- [52] ICAO Document 9694
- [53] SESAR, The ATM Deployment Sequence, DLM-0706-001-02-00, Document D4, 09/01/2008
- [54] ATM Master Plan 1.1
- [55] WPB.01 Integrated Roadmap, Dataset 00.00.16.
- [56] P14.01.03-D35-SWIM Architectural Definition for Iteration 3.0, 00.01.01, 22/02/2014
- [57] [Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol](#), formal/2014-09-01
- [58] Deploying DDS on a WAN: The DDS routing service  
<https://community.rti.com/content/presentation/deploying-dds-wan-and-gig-dds-router>  
[http://www.omg.org/news/meetings/GOV-WS/pr/rte-pres/DDS\\_RoutingService\\_RTEW09.pdf](http://www.omg.org/news/meetings/GOV-WS/pr/rte-pres/DDS_RoutingService_RTEW09.pdf)
- [59] [OASIS Advanced Message Queuing Protocol \(AMQP\) Version 1.0](#)
- [60] [ISO/IEC 19464:2014 Advanced Message Queuing Protocol \(AMQP\) v1.0 specification](#)
- [61] [OASIS WS-BaseNotification, WS-BrokeredNotification and WS-Topics 1.3.](#)
- [62] ITU-T X.1252 Baseline identity management terms and definitions ([http://www.itu.int/SG-CP/example\\_docs/ITU-T-REC/ITU-T-REC\\_E.pdf](http://www.itu.int/SG-CP/example_docs/ITU-T-REC/ITU-T-REC_E.pdf))
- [63] WPB.04.01, D136 European ATM Architecture (EATMA) Guidance Material v6, edition 00.06.04.
- [64] NATO Architecture Framework, v3.1, 01 March 2010

## Appendix A P14.01.03 TAD Integration with B4.3 ADD

The integration with the overall European ATM Architecture Design B4.3 (see [6]) is not intended to be a part of a TAD document, but rather of an ADD (and hence, B.4.3 handled). So, this appendix should only be intended as a set of proposals/clarifications.

### A.1 ADD Main Concepts/Stereotypes

The following architectural elements are identified in the ADD (ref. [6]) and used to describe the overall European ATM:

1) A **Capability Configuration (CC)** is a combination of Human Resources and Systems configured to provide a Capability derived from operational and/or business need(s) of a stakeholder type. A CC presents to ATM stakeholders what type of systems and/or actors to deploy in order to provide the required capability.

2) A **Resource Interaction** reflects an information exchange need between Capability Configurations (CCs). A Resource Interaction carries (conveys is the term used in NAF) information elements, directly between CCs or through an Infrastructure CC.

3) An **Infrastructure System** is part of an **Infrastructure CC** and provides a set of functionalities that can be used across several domains. Their purpose is to provide a standardized way to support interactions between Domain Systems.

4) Within a CC, a **Domain System** is a System that provides a set of ATM functionalities which are closely linked to the domain's business needs. A **Functional Block (FB)** represents a grouping of functionalities within a Domain System.

5) A **System Port** is an interface (logical or physical) provided by a Domain or an Infrastructure System. System Ports are connected to each other by System Port Connectors. A System Port Connector can only connect ports of the same type, i.e. the same port must exist on both sides/systems. System Ports fulfil technical requirements which might be relevant to realize different Resource Interactions and thus are reusable.

6) System Ports implement Protocols. A **Protocol** is a Standard for communication, i.e. a Protocol always has a relation to a Standard. Protocols may be composite (i.e. arranged in a stack). Lower layers of the protocol stack are provided by Infrastructure systems while the upper layers are implemented at the Domain Systems.  
With the SWIM there will be an intermediate SWIM layer introduced providing SWIM services.

In the architecture framework, the external interfaces of all the Domain Systems have to be defined by means of System Ports to ensure the ATM interoperability. The ADD will describe the information exchanges between Capability Configurations by means of Resource Interactions.

The interfaces between Domain Systems within the same CC, and the interfaces between the Functional Blocks, are not in the scope of the ADD. They are a responsibility of the federating projects and should be documented in the TADs, and described in the different Technical Specification documents (TS) and/or in the Interface Requirements Specification (IRS).

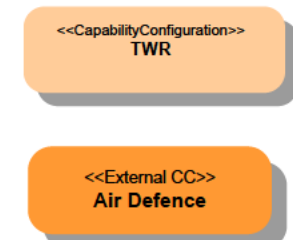
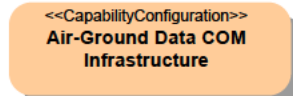

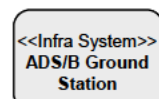
The following table summarizes both architectural elements and stereotypes used (those that are of interest for the integration of P14.01.03; for an extensive description, please refer to the ADD (ref [6]):

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

206 of 284

Concept	Definition	Comment	ADD Symbol (example)
Actor	An actor is an aspect of a person or organisation that enables them to fulfil a particular function.	Each actor can perform one or more roles	
Capability	Capability is the ability of one or more of the enterprise's resources to deliver a specified type of effect or a specified course of action to the enterprise stakeholders.	This is a business concept (WPB.04.01)	Please refer to WPB.04.01 documentation
<b>Capability Configuration</b>	A <b>Capability Configuration (CC)</b> is a combination of Human Resources and Systems configured to provide a Capability derived from operational and/or business need(s) of a stakeholder type.  An external CC is a CC that does not belong to SESAR scope.	CCs represent the nature of the normally instantiated deployment combinations for the different generic stakeholder types. The deployment for a given CC at different locations depends on the stakeholder environment.	
<b>Infrastructure Capability Configuration</b>	An <b>Infrastructure Capability Configuration (Infra CC)</b> is a CC combining Infrastructure Systems.	Examples of Infrastructure Capability Configuration are Communication Infra CC, Surveillance Infra CC...	
<b>Domain System</b>	A <b>Domain System</b> is a System that provides a set of ATM functionalities which are closely linked to the domain's business needs.	A Domain System is something that provides functionality that is directly linked to the business of the given domain. As such it usually will not make sense to use the same system in a different domain.	
<b>Infrastructure system</b>	An <b>Infrastructure System</b> is a System providing a collection of functionalities which are agnostic to the ATM business processes.	An Infrastructure System provides functionalities that are needed throughout different (business) domains. They address crosscutting	

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Concept	Definition	Comment	ADD Symbol (example)
		<p>concerns.</p> <p>An Infrastructure System provides port to port connectivity to support multiple business domains by combining physical elements.</p>	
<b>Functional Block</b>	<p>A <b>Functional Block</b> (FB) represents a grouping of functions that are assembled to support or perform one or more Operational Activities.</p>	<p>Functional blocks can be re-used in multiple Domain Systems or can be specific.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p>&lt;&lt;functional block&gt;&gt;  <b>Flight Management</b></p> </div>
<b>System Port</b>	<p>A <b>System Port</b> is an interface provided by a System.</p> <p>System Ports are connected to each other by System Port Connectors. A System Port Connector can only be made between ports of the same type, i.e. the same port must exist on both sides/systems.</p> <p>The System Ports are linked by System Port Connectors that are realising the Resource Interactions between Capability Configurations. One Resource Interaction could be realized by multiple System Port Connectors.</p>		

Table 35 – ADD Main Concepts/Stereotypes



## A.2 P14.01.03 Main Architectural Elements

As described throughout the document, the main architectural elements to be provided by SWIM-TI are the following:

### 1. Functional Entities:

- A **SWIM-TI Function** is a technical activity which is specified in context of the SWIM-TI. It represents the **lowest level of decomposition** of the SWIM-TI and aims for the completeness (its definition is complete and self-contained).
- **SWIM-TI Functional Block** represents a **logical aggregation** of functions within an instance of the SWIM-TI that are assembled to assist in the conducting of one or more SWIM-TI Activities.
- **SWIM-TI Functional Block** is a **logical grouping of functions** used by other SWIM-TI Functional Blocks (shared or not) in order to ensure the correct behaviour of the SWIM-TI.

A Functional Block is a particular SWIM-TI Functional Block in which the set of functions that are aggregated by it are specified to provide support to the SWIM-TI rather than to provide support to an ATM System.

Functional Blocks are to be used by one or more SWIM-TI Functional Blocks in order to support their functions. A given Functional Block can be used by all or by a sub-group of SWIM-TI Functional Blocks.

### 2. Technical Entities:

- A **SWIM-TI Infrastructure System** is a System providing a collection of SWIM-TI functionalities.
- A **SWIM-TI Profile** is a SWIM-TI Infrastructure System that groups a coherent, appropriately-sized set of SWIM-TI Functional Blocks for a given set of technical constraints/requirements that permit a set of stakeholders to realize Information sharing.

It also defines the mandated open standards and technologies required to realize this coherent grouping of middleware functions/services.

A SWIM-TI Profile is a concrete **group of SWIM-TI Functional Blocks**. For each SWIM-TI Functional Block, a SWIM-TI Profile Instantiation derived from the **SWIM-TI Profile Descriptor** will define a concrete set of requirements.

Each SWIM-TI Profile Instantiation can be understood as a **specific instance** of the SWIM-TI FB decomposition.

- A **SWIM-TI System Port** is a **System Port** provided by SWIM-TI that represents the interface between SWIM-TI Infrastructure Systems.

According to this and in order to summarize, SWIM-TI TAD provides:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

209 of 284

- A) **A set of SWIM-TI Functional Blocks** that describe all the possible functions that are to be handled by the Technical Infrastructure.
- B) **A set of SWIM-TI Profiles (SWIM-TI Infrastructure System)** that, for each of them, particularize the above mentioned set of SWIM-TI Functional Blocks.
- C) **A set of SWIM-TI Infrastructure Systems** realizing **SWIM-TI Functional Blocks** that describe the set of functions that are needed by the SWIM-TI Functional Blocks but are not intended to be included in the SWIM-TI Profiles (for different reasons).
- D) **A set of SWIM-TI System Ports.**

The integration with the ADD should be, according to these definitions, straightforward as the functional and technical entities are aligned.

## A.3 P14.01.03 Proposals

To understand this section, the following statements are to be assumed:

- P14.01.03 (and so WP14 in general) doesn't create/modify Systems.
- P14.01.03 (and so WP14 in general) are only to provide the design of the Technical Infrastructure, but the way this is integrated into a System needs to remain as open as possible (refer to SWIM-TI Design Principles).

### A.3.1 SWIM-TI Profiles and ADD integration

As described in the ADD (ref. [6]), there are several options to integrate SWIM-TI Functional Blocks/functions to the ADD<sup>76</sup>:

---

<sup>76</sup> Even if the figure presents Step1 SWIM-TI, it also applies to Iteration 3.1.

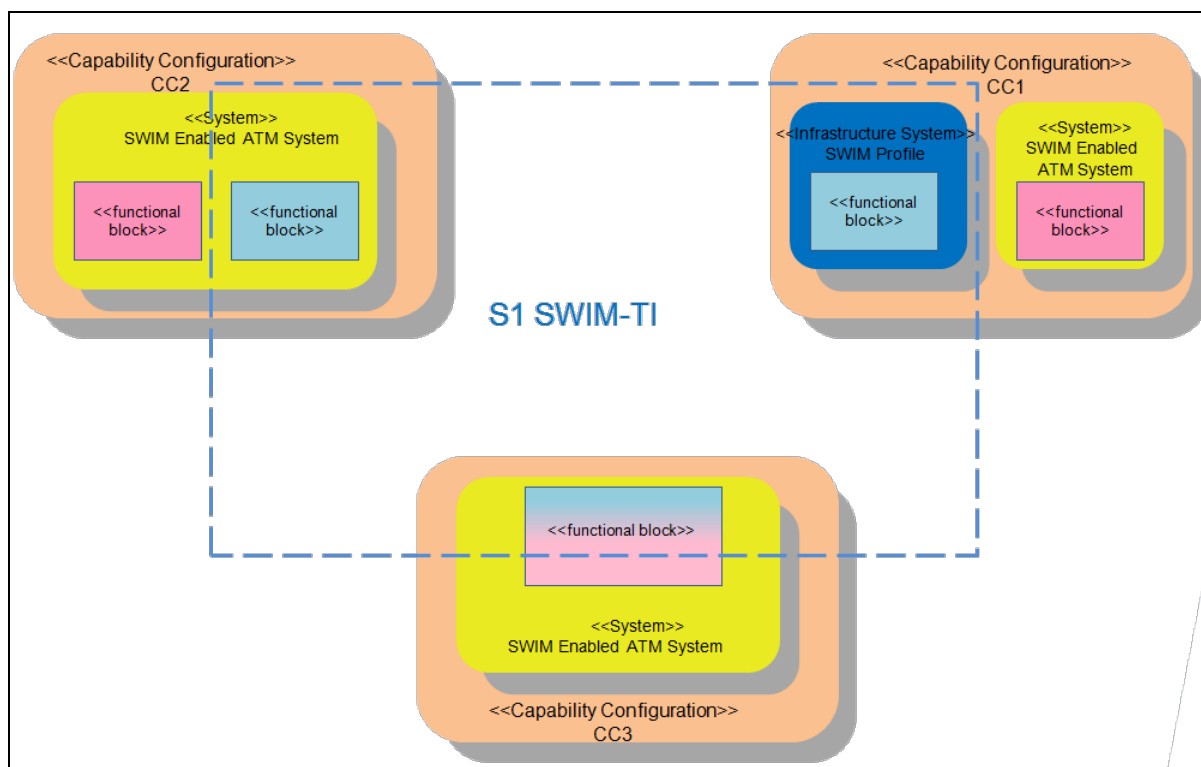


Figure 80 – SWIM-TI and different SWIM-TI Functional Block deployment options

CC2 and CC3 don't provide much discussion: for CC2, the SWIM Enabled ATM System will incorporate among its FB those related to the SWIM-TI in order to support one or more SWIM-TI profiles; for CC3, this is not even needed, as the SWIM Enabled ATM System are already supporting the requirements associated for realizing a concrete SWIM-TI profiles.

In the case of CC1 and having in mind the notion of SWIM-TI Node as "pure logical" or aggregator, the following diagram would provide an answer to integrating one or more (in the figure two) SWIM-TI Profiles for supporting the interoperability of a Capability Configuration.

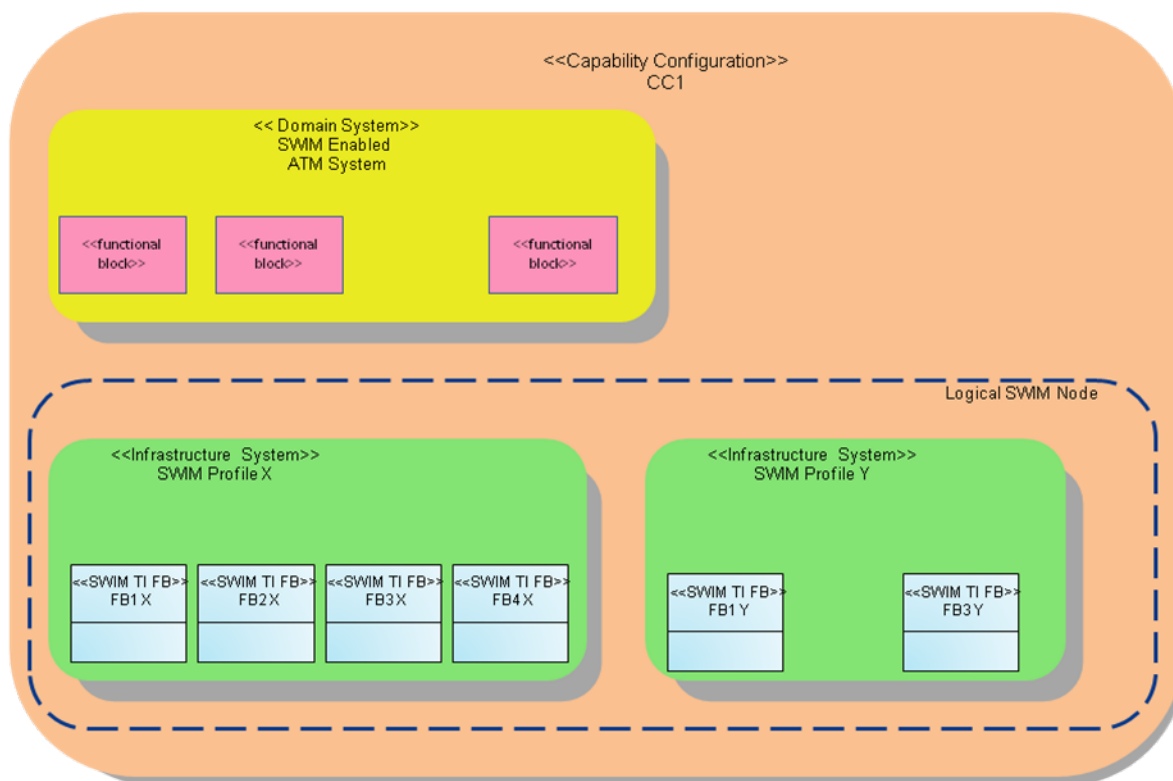


Figure 81 – Example: Two SWIM-TI Profiles supporting one CC

### A.3.2 SWIM-TI Infrastructure Systems and ADD integration

The case of the SWIM-TI Infrastructure Systems is even more complicated, as is not clear who will handle the SWIM-TI Functional Block.

In the definition of SWIM-TI Functional Block is clear the fact that is “used by other SWIM-TI Functional Blocks” and so, it will need to communicate with them and to define the interfaces presented in chapter 2.2.3.

The following figure provides an example of how a SWIM-TI Functional Block could be integrated in the ADD (via creating an Infrastructure System that would support it).

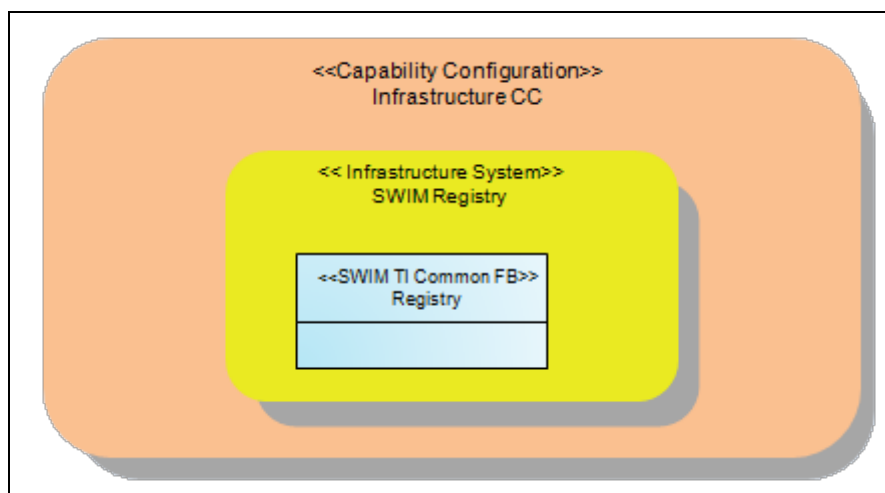


Figure 82 – Example: SWIM-TI Functional Block in the ADD

To be highlighted that allocating a SWIM-TI Functional Block to an Infrastructure System doesn't assume that the System is neither centralized nor distributed. Both of them are deployment options that can be supported.

### A.3.3 SWIM-TI Support Infrastructure proposals

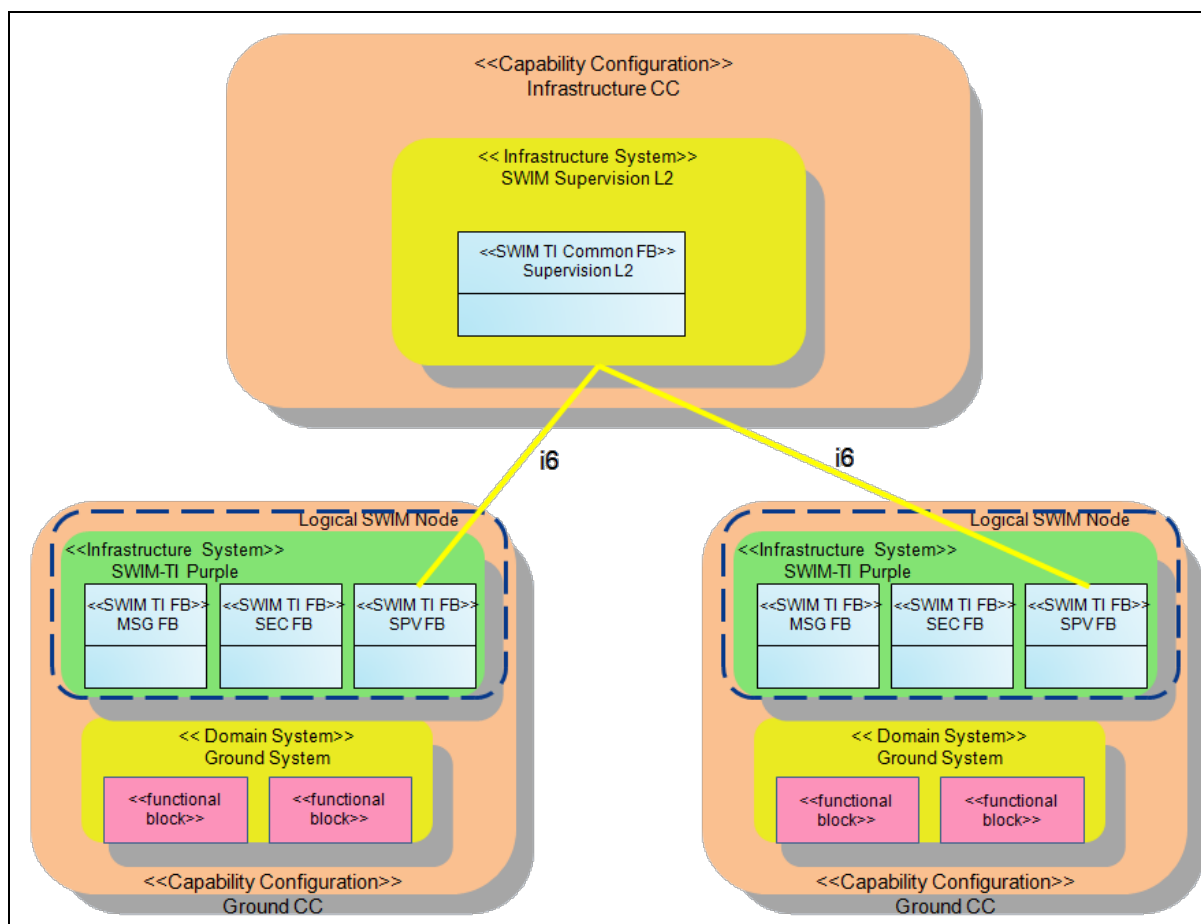
Additional to those already included in the SWIM-TI Functional Blocks, P14.01.03 has identified the following candidates for Infrastructure Systems:

All these proposals need to be further analyzed from different points of view such as Governance, System of Systems, SOA, etc.

- **Supervision Infrastructure System**

Supervision FB analysis proposes L2 and L3 hierarchical SWIM Supervisions. Both SWIM-TI L2 Supervision and SWIM-TI L3 Supervision would be materialized via SWIM-TI Functional Blocks and via Infrastructure Systems. Appendix D describes the different SWIM Supervision elements identified in previous stages.

Note that interface i6 would need to be defined and that the numbering continues the set of interfaces previously identified in chapter 2.2.3.



**Figure 83 – Supervision Infrastructure System**

- **Access Point Infrastructure System.**

P14.01.03 D031 (ref. [27]) included an appendix in which the needs and first proposals for solution for the integration of the Aircraft into the SWIM-TI network were described. Appendix B of this document tackles the process followed to integrate A15 into SWIM-TI TAD.

Within the analysis and further deployment options, the possibility of a Ground Capability Configuration unable to communicate via SWIM-TI with an air Capability Configuration due to the lack of common/interoperable SWIM-TI profiles in both of them was identified. To fix this eventuality, the concept of Access Point was developed. The Access Point Infrastructure System manages the switch from a SWIM-TI Profile to another. This could be extended to other SWIM-TI Profiles.

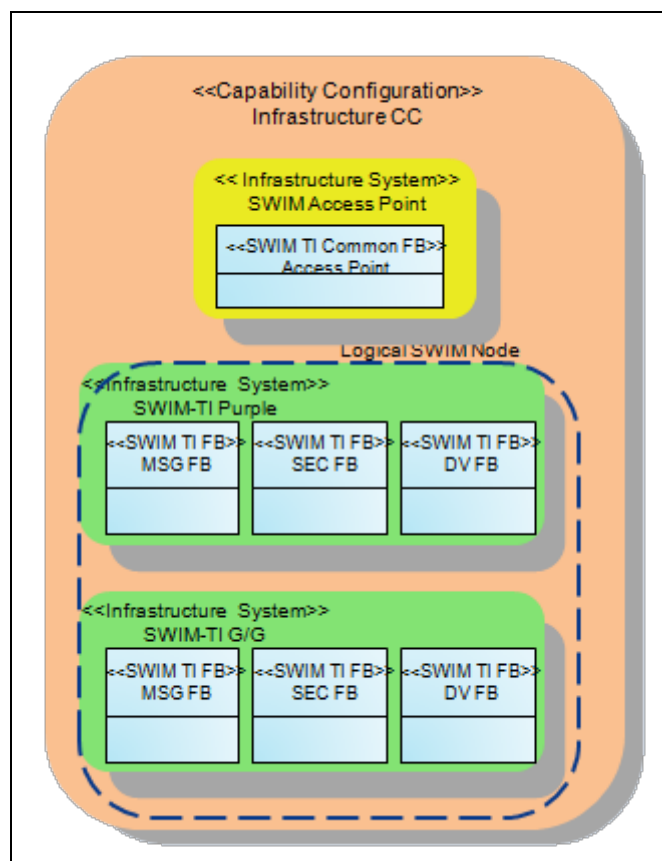


Figure 84 – Access Point Infrastructure System

- **Military Access Point Infrastructure System**

The study on SWIM Civil-Military Interoperability performed in previous iteration via “D1 Characteristics of Military ATM and AD/C2 systems and the justification for their interoperability with SWIM” (ref. [7]) and “D2 Target SWIM Interoperability Concept and Architecture” (ref. [8]) resulted in a proposal for an integration approach via a Military SWIM-TI gateway due to several reasons (see Appendix C).

Architecturally, this approach could be realized by the creation of an Infrastructure System that would support the different SWIM-TI Profiles defined/needed to enable the access from the military systems to the civil systems’ desired information and vice-versa complying with the identified security and performance restrictions.

The following figure presents the architecture of such Infrastructure System.

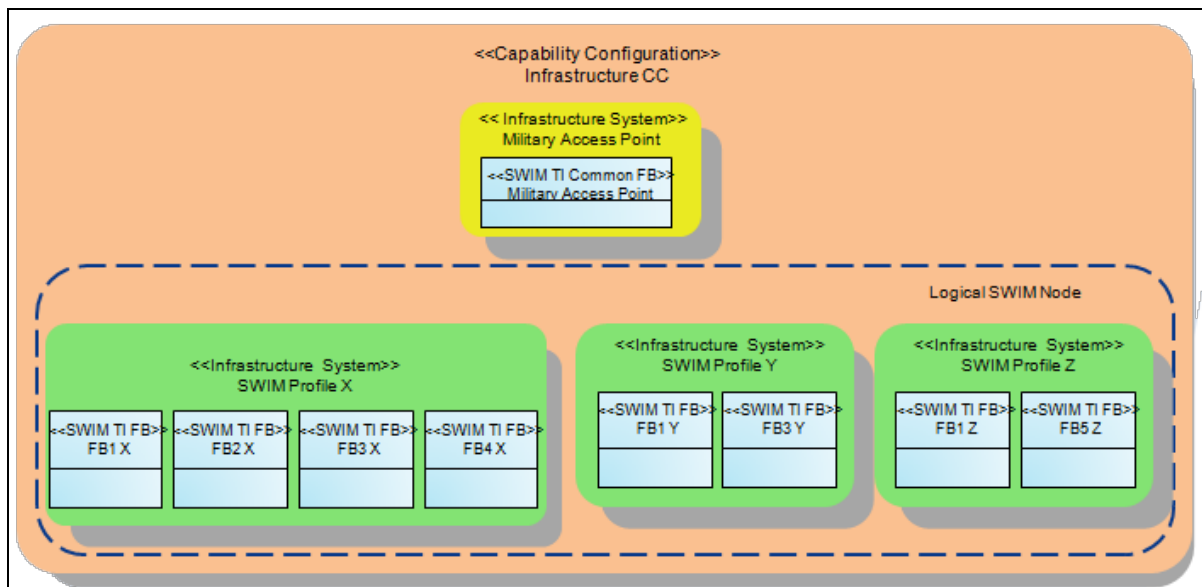


Figure 85 – Military Access Point Infrastructure System



## Appendix B Process to integrate A15 and results

P14.01.03 D031 (ref. [27]) included an appendix in which the needs and first proposals for solution for the integration of the Aircraft into the SWIM-TI network were described.

Being recognized that the aforementioned appendix covered much more than the SWIM-TI scope (in fact, partners from different domains, operational and technical participated in the elaboration of the document), the process for integrating it into the SWIM-TI design based the analysis in the identified requirements as well as in the identified functions.

The integration consisted of an analysis of the coverage by the current SWIM-TI Functional Blocks of the functions defined by A15. In the event that the coverage wasn't complete, it could derive in the need for defining SWIM-TI FB to support the A/G communications.

The following table presents the coverage matrix:

A15 A/G functions	Messaging FB	Security FB
Message Routing and Distribution	X	
Message Filtering	X	
Messaging Protocol Switch	X	
Data Encapsulation	X	
Message Encryption		X
Message Signature		X
Data Validation	X	
Data Format Change	X	

Table 36 – A15 functions vs. SWIM-TI FB

Also due to this approach, a new SWIM-TI Profile is proposed<sup>77</sup>, the one that supports the specific requirements for Messaging FB and Security FB for the SWIM-TI A/G.

### B.1 Support of the Deployment Choice

Moreover, the integration of the functions into the TAD should support all the deployment options presented in this figure:

<sup>77</sup> Already identified as Purple Profile

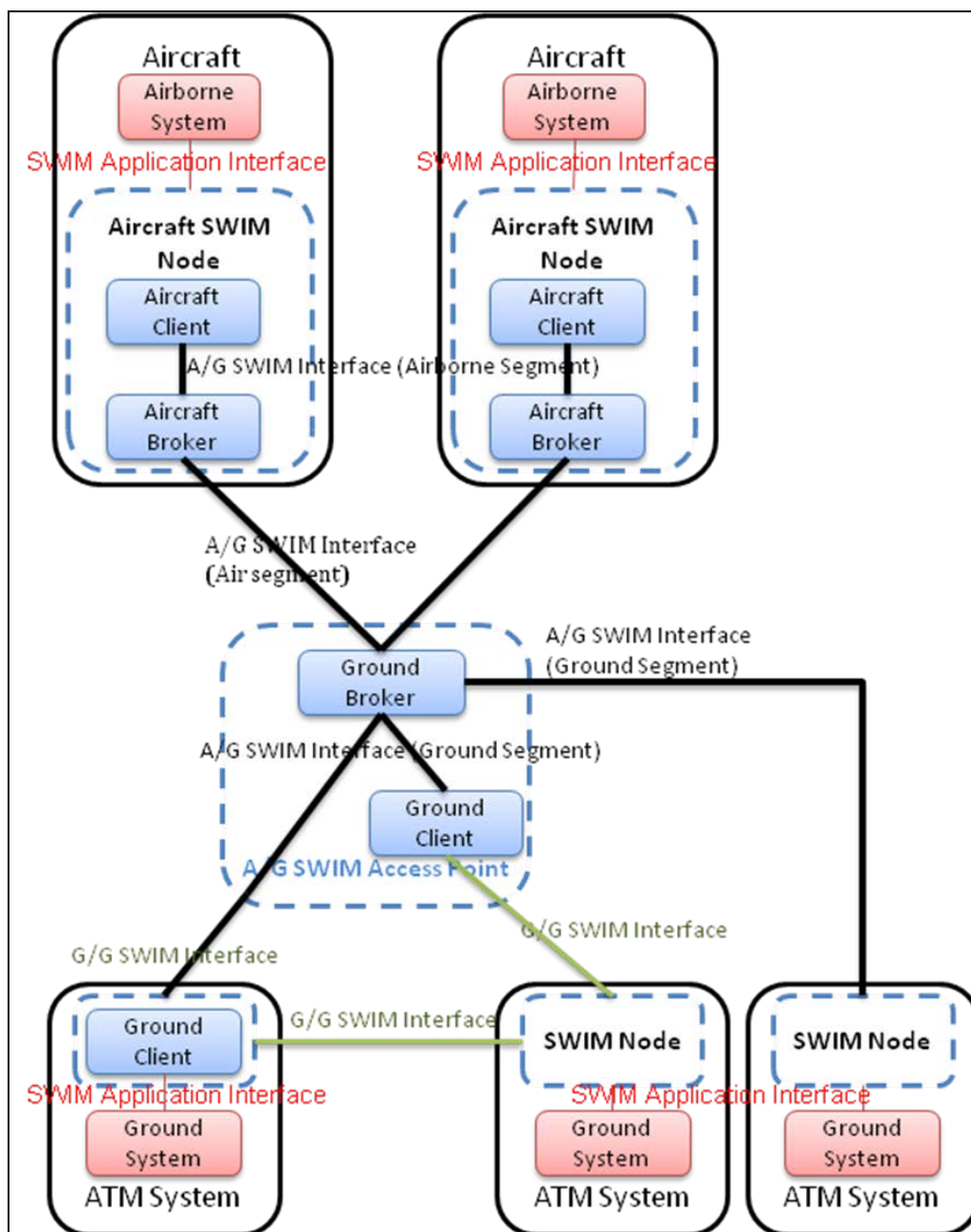


Figure 86 – A15 SWIM-TI A/G Deployment Options

In the figure above, the following architectural elements were used:

- **Aircraft SWIM Node** that aggregates Aircraft Broker (deals with Message Routing and Distribution functions) and Aircraft Client (deals with Message Filtering, Messaging Protocol Switch and Data Encapsulation functions).
- **A/G SWIM Access Point** (logical element that will be composed of at least a Ground Broker to deal with Message Routing and Distribution functions).
- **Ground Client** (deals with Message Filtering, Messaging Protocol Switch and Data Encapsulation functions).

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

- **SWIM Node** that represents a logical SWIM-TI Node (as defined in this document).

Also several interfaces were identified:

- **A/G SWIM Interface (Airborne Segment)**
- **A/G SWIM Interface (Air Segment)**
- **A/G SWIM Interface (Ground Segment)**
- **G/G SWIM Interface**

All these interfaces are supported in the integration by either by SWIM-TI A/G Profile, either SWIM-TI A/G Profiles, either legacy integrations.

Basically, the deployment choice aggregates three different scenarios. The following figures present how these scenarios would be supported by the integration approach followed:

- **Ground System implements the SWIM-TI A/G Profile<sup>78</sup>**

In this case, there's just a need for both counterparts (Capability Configuration in the ground and Capability Configuration in the air) to implement the **SWIM-TI A/G Profile** in order to be able to communicate between them via SWIM-TI.

---

<sup>78</sup> SWIM-TI Purple Profile

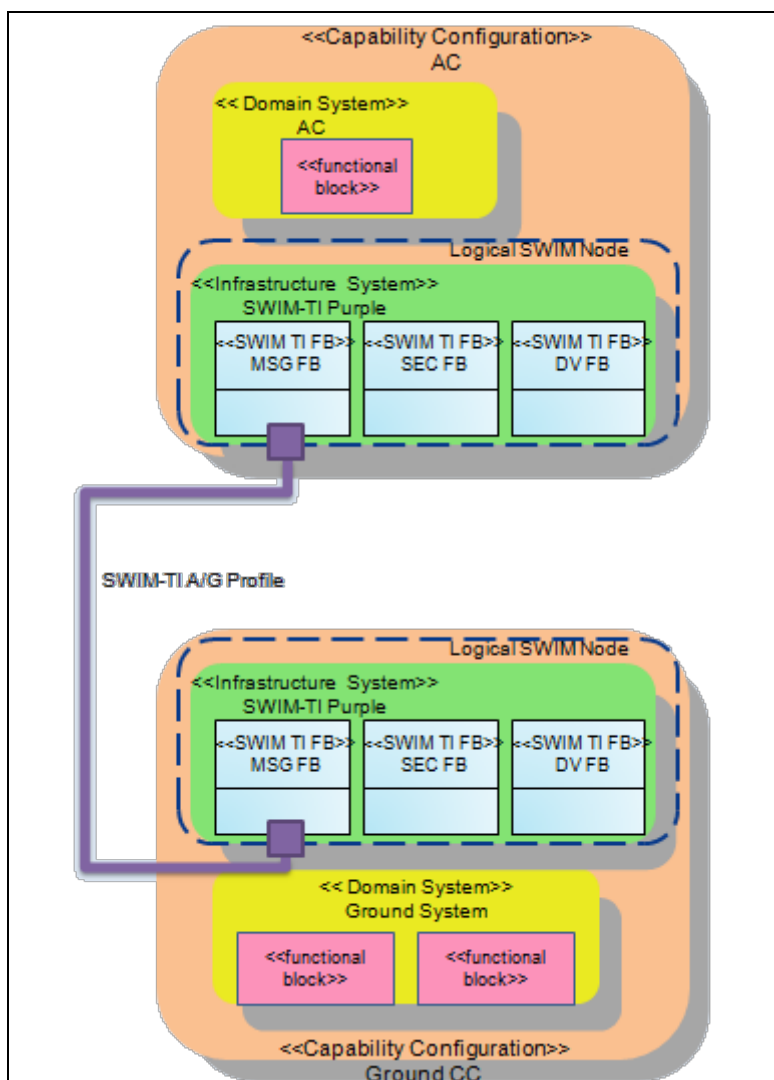


Figure 87 – Ground System implements the SWIM-TI A/G Profile

For the sake of legibility, the following figure doesn't present all the details of the figure above, but it aims at representing how it will be implemented with several AC CC and several Ground CC:

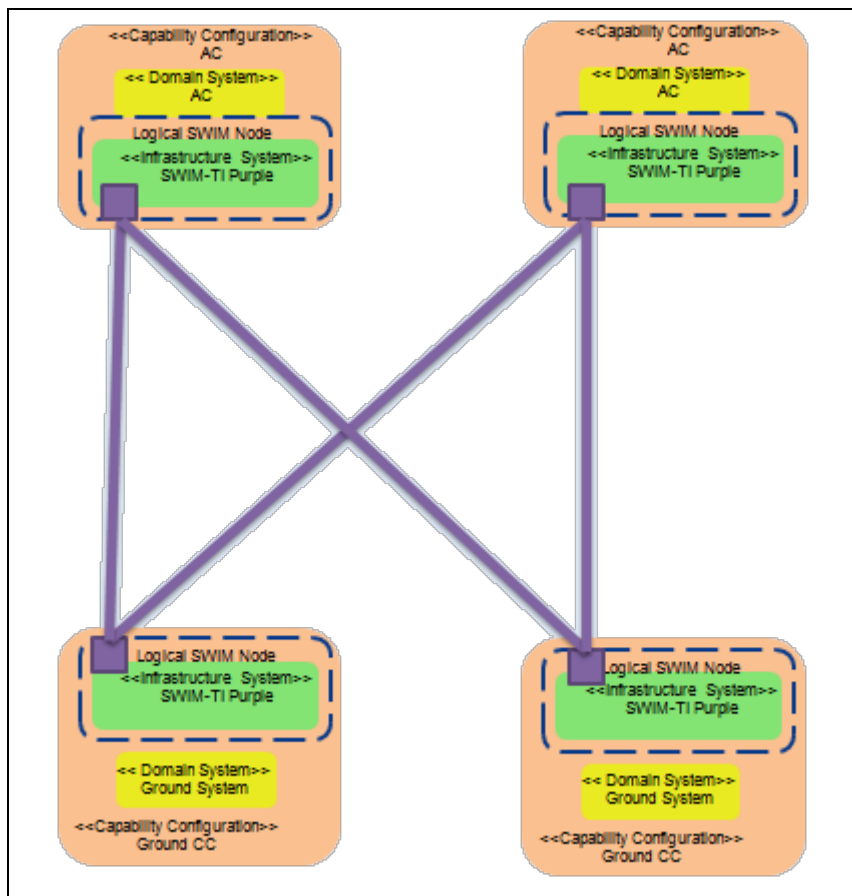


Figure 88 – Several Ground System implements the SWIM-TI A/G Profile

- Ground System doesn't implement the SWIM-TI A/G Profile<sup>79</sup> but implements other SWIM-TI G/G Profiles

In this case, a third party<sup>80</sup> Capability Configuration needs to grant access to the Ground Capability Configuration to the Air Capability Configuration. To access the third party Capability Configuration, SWIM-TI G/G Profiles are used.

<sup>79</sup> SWIM-TI Purple Profile

<sup>80</sup> In the figure, an Infrastructure CC has been used as third party; however, this role can be handled by other Domain CC. This is out of WP14 scope.

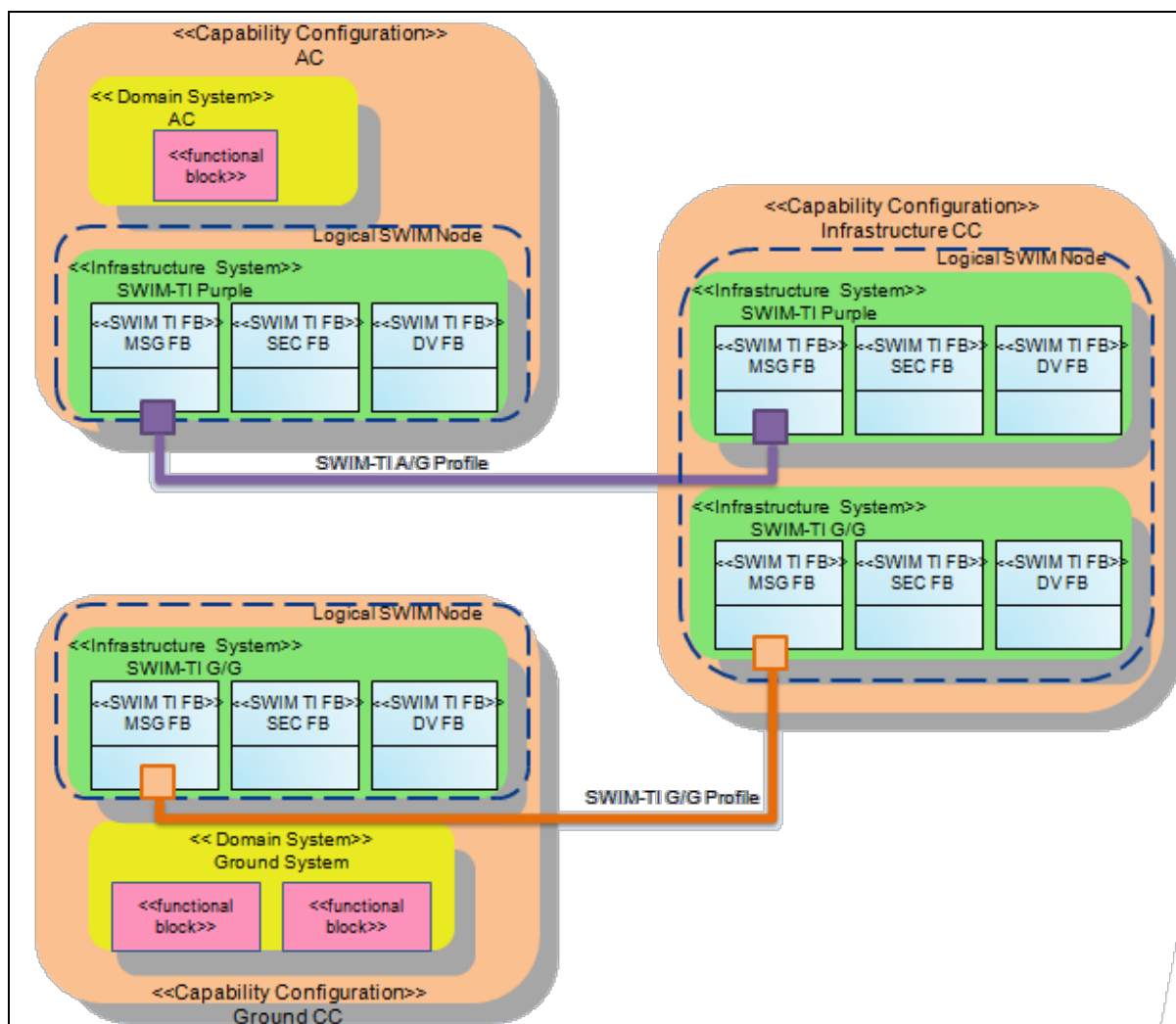


Figure 89 – Ground System doesn't implement the SWIM-TI A/G Profile

For the sake of legibility, the following figure doesn't present all the details of the figure above, but it aims at representing how it will be implemented with several AC CC and several Ground CC:

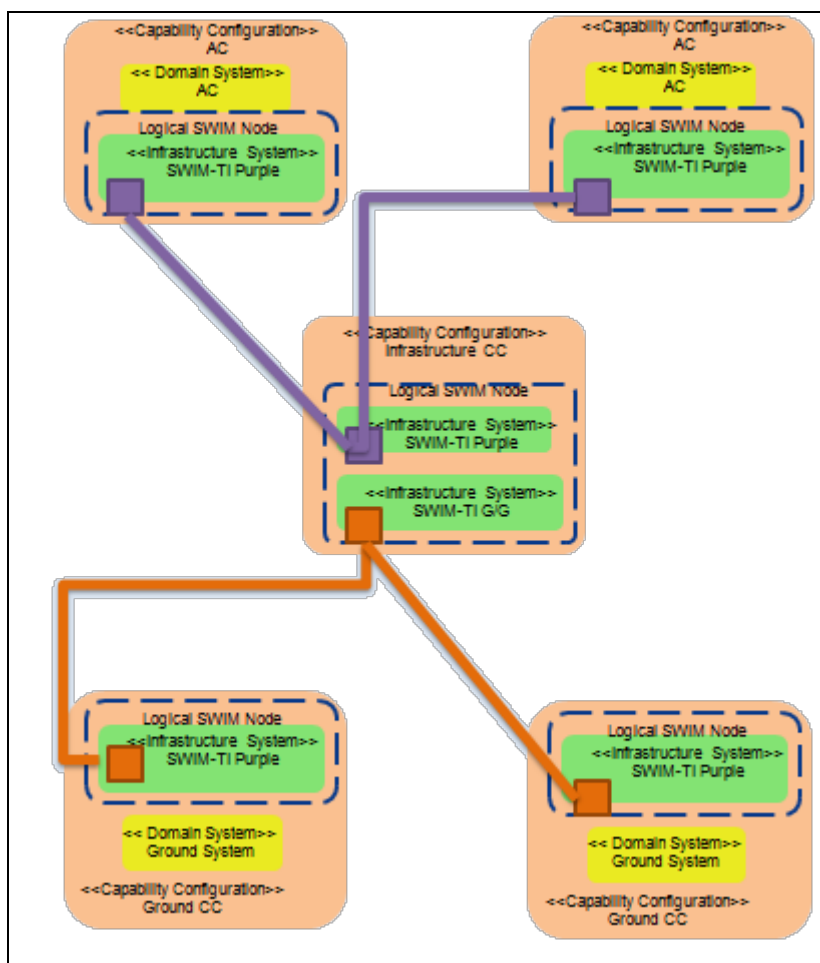


Figure 90 – Several Ground System doesn't implement the SWIM-TI A/G Profile

- **Ground System doesn't implement the SWIM-TI A/G Profile<sup>81</sup> and doesn't implement other SWIM-TI G/G Profiles**

In this case, a third party<sup>82</sup> Capability Configuration needs to grant access to the Ground Capability Configuration to the Air Capability Configuration. To access the third party Capability Configuration, legacy integration is used.

<sup>81</sup> SWIM-TI Purple Profile

<sup>82</sup> In the figure, an Infrastructure CC has been used as third party; however, this role can be handled by other Domain CC. This is out of WP14 scope.

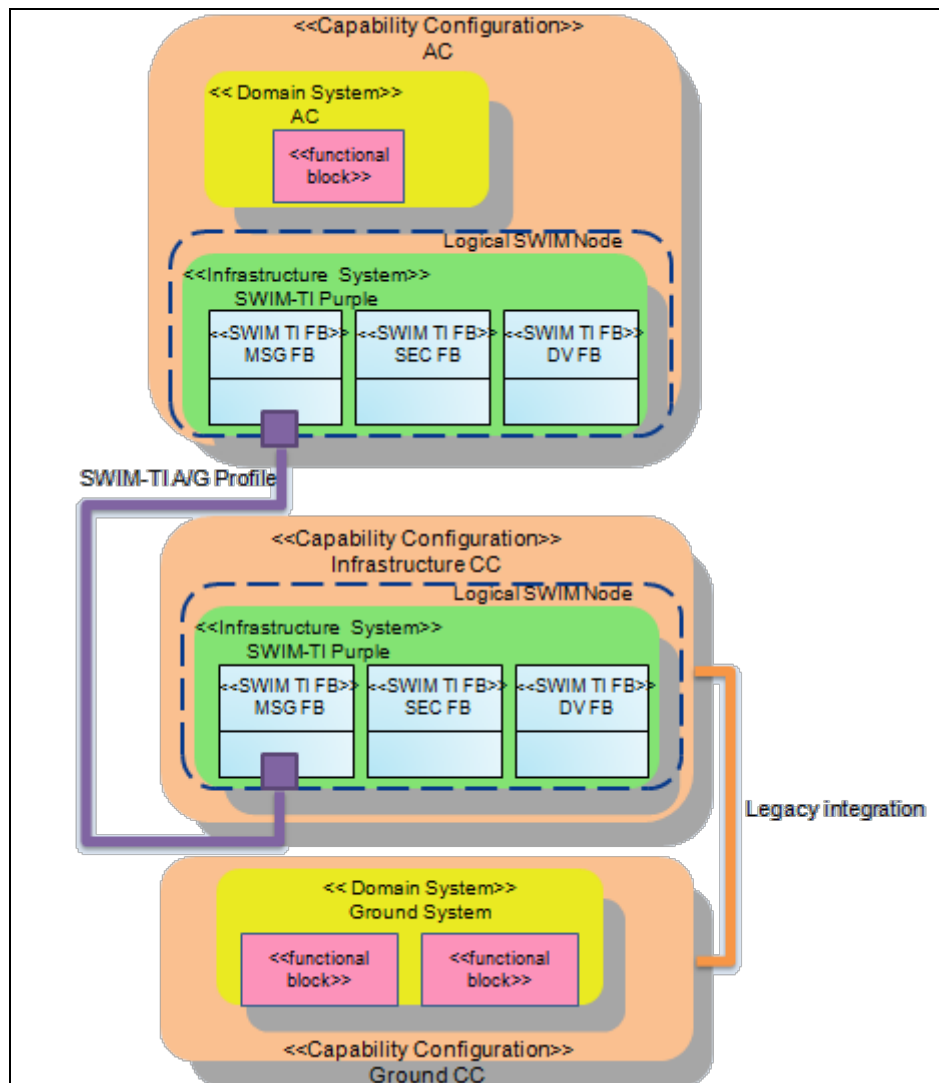


Figure 91 – Ground System doesn't implement the SWIM-TI A/G Profile (legacy integration)



## Appendix C Process to integrate Military Systems into SWIM-TI

### C.1 Military inputs

In addition to the input documents listed in section 1.3, it has been performed an analysis of the following inputs:

- Study on SWIM Civil-Military Interoperability, D1: Characteristics of Military ATM and AD/C2 systems and the justification for their interoperability with SWIM [7].
- Study on SWIM Civil-Military Interoperability, D2. Target SWIM Interoperability Concept and Architecture [8].

It results from these studies that civil SWIM connection with military systems is highly desirable and should be beneficial from both parties. Military systems vary tremendously one country from the others in their implementation. Even the coordination between civil and military controllers varies in the way of operating.

Furthermore it may be noted that ongoing programs exist to modernise some of the existing military systems. Though the ACCS program tends to standardize the systems, additional functions have been implemented for a number of nations. Similarly experimental tests are carried on operational procedures, for OAT control for example.

From this analysis we derived the following assumptions to build the architecture of civil/military interoperability using SWIM:

ASM-0001	SWIM-TI is not used to exchange information directly between two military systems whatever kind they are.
ASM-0002	SWIM-TI shall not connect directly to a military aircraft.
ASM-0003	Classified data are not going through SWIM-TI.

ASM-0001 is induced by the need to maintain the connection between military entities in case of crisis.

ASM-0002 is induced by the fact that military aircraft are considered as weapon system and therefore ASM-0001 applies.

ASM-0003 follows the security rules on classified data.

Proved secured connection between military world and ground SWIM requires additional security solutions that are more expansive than standard security solutions planned to be implemented in civil ground SWIM. The military security must be based on self-defence rather than on trust on the civil one. In case of crisis it must be possible to disconnect all civil/military links in a quick and reliable manner. Considering these constraints leads to build an architecture where the number of physical connections between civil and military world are very limited. Several options are considered in D2 ([8]).

- External adaptors for individual systems  
It is made of a separate device on the perimeter network between the military system and SWIM. It performs protocol translation to allow the military system to continue to use its

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

225 of 284

proprietary or legacy interfaces and it is also able to enforce an appropriate security policy at the domain boundary.

- **Direct integration**  
It is made of an implementation of SWIM compatible middleware on a military system that enables the system to communicate natively with the SWIM network. To deal with security issues, an additional firewall/gateway between the civil/military world would be required.
- **Faked devices**  
It allows dealing with cases whereby the flexibility of the older legacy systems is so limited that the gateway devices must be built to look exactly as a standard device of the legacy system to connect to SWIM.
- **Integrated gateway solutions**  
The implementation of an integrated gateway solution allows the provision of SWIM interoperability for multiple systems through a common interface, similar to a large scale external adaptor.

As national sovereignty and classification of information, lead most of the military systems to exchange on a dedicated national network, the Nations should prefer the latter option. Moreover different forms of gateways that all provide the functionality of a standard SWIM Node at the civil side and appear in different forms at the military side.

The context will determine which option will be deployed.

- Amongst others, factors such as the trust domain, the cost, the planning cycle, international objectives as well as national objectives are part of this context.
- Hence, for instance, at least one gateway could exist per country or multiple gateways of different types could exist per country to deal with variety/heterogeneity of military equipments.

## C.1.1 Architectural options

A Military SWIM-TI gateway is composed of:

- a “civil” SWIM node instantiating all the SWIM profiles required to support ground SWIM services offered by civil systems and services exposed by military systems,
- a military gateway performing the data filtering and the protocol mediation to accommodate military standards,
- a hardware/software diode that secures military network from attacks coming from civil world.

The security architecture expects that all control of the gateway is fully and exclusively on the military side provided the SWIM interoperability requirements are satisfied. E.g. policy definition and enforcement points are on military side.

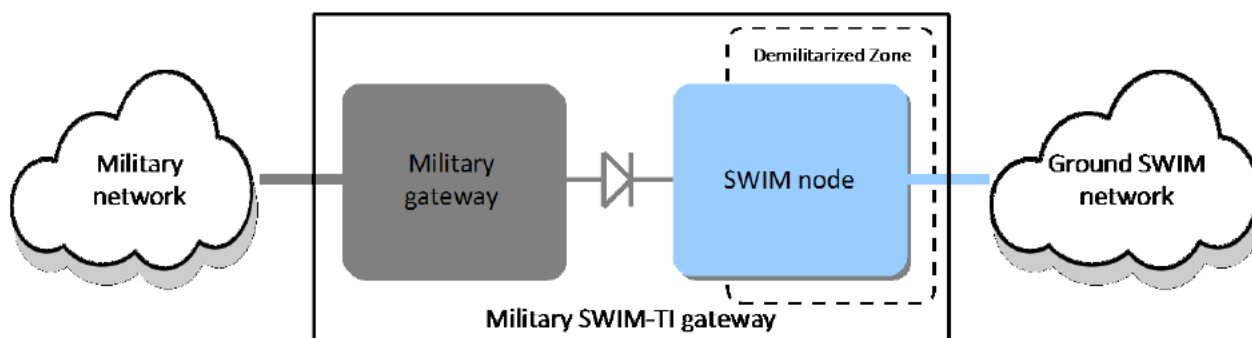


Figure 92 – Military SWIM-TI gateway

On the other side of the military network could be connected a single system, several systems of an Air Army or all the Air military systems of a nation. It may happen that a NATO military gateway could be reused by several NATO systems.

### C.1.2 Military specific constraints

The studies also show some requirements or constraints on the military SWIM-TI gateway. These constraints are listed in the table below. They will be derived in SWIM-TI requirements if needed in the iteration 2.1.

Both from a perspective of service application domains (e.g. Environment domain requirements) as well as from an infrastructure perspective (e.g. clean architectural structuring of MEP, Top-down architecture of security related entities) the current level of specifications in SWIM is not sufficient. The military specifications are therefore provisional and may evolve when more information/specifications become available.

Category	Constraint description
Time	<p>It may be necessary to assess whether ICAO Annex 2 section 3.5.3 (applicable to military systems) implies the need for time synchronization. This assessment is not limited to the Civil/Military interoperability but needs to be clarified at the Civil side as well.</p> <p>SWIM-TI does not provide any time synchronization service.</p>

Category	Constraint description
Security access	<p>The protection of information provided by the Military applies both during the transfer on the SWIM network as well as in the systems where it is processed. The SWIM TI security scopes the transfer part only.</p> <p>For various types of information non-authorized users should be prevented to learn from the information or patterns in the information. This is limited to a system category (ATC en-route system, Airport system, ...). It is not possible to address the level of the human end-user.</p> <p>For various types of information only a subset of the available information should be made available to authorized users and only when relevant and what is relevant on a need-to-know base.</p> <p>It is required that Publish/Subscribe MEP is used with Security capabilities. This might be an issue e.g. with respect to the actual status of security e.g. in DDS. However the current specifications do not exclude this security to be limited to the transport level only.</p> <p>The security mechanism offered by DDS are limited to network security up to now. A standardization activity is on going to fill this gap before the deployment of DDS related SWIM profile.</p> <p>The military side must be able to receive and decrypt confidential information that is sent to the military side from SWIM.</p> <p>The meta-data related to description of military services should not be accessible (for instance in the registry) to non-authorized users. Access to the description of military services in the registry has to be secured.</p>
Security integrity	<p>It must be possible to check the integrity of the information sent from SWIM as well as information sent to SWIM. Integrity checking includes the ability to check the origin of the information. In particular it should not be possible for a non-military user to be able to pretend to be a military user.</p> <p>A typical example of integrity violation consists of spoofing.</p>
Cyber security	<p>Whereas the ultimate responsibility for protection against attack, lies at the military side including the gateway, it is strongly recommended to deploy architectural devices in the SWIM TI to reduce the risk of an attack and to mitigate the impact of an attack.</p> <p>Typical examples of attack are DoS and/or DDoS. An example of an architectural device is the segmentation through compartmentalisation into a Core SWIM and an Access SWIM.</p> <p>Network segmentation is already in place in civil SWIM to segregate safety critical exchanges from others. (ATC/PENS, AMHS, ...).</p>
Accreditation	<p>A security accreditation (or Certification) is proposed by D2. This could be linked to the Certification-NFR. The scope of the systems subjected to accreditation is not clear : only the gateway or also entities in the SWIM TI.</p> <p>This accreditation is not in the scope of the SWIM node. This is under responsibility of the whole gateway considered as a military sub-system.</p>

Table 37 – Military constraints

## Appendix D SWIM-TI Supervision

### D.1 SWIM-TI Supervision Hierarchy

The core of SWIM-TI infrastructure is the SWIM Node, where SWIM-TI Supervision is identified as one of its functionalities. It's important to highlight that a SWIM Node is a logical aggregation of SWIM functionalities (not having a concrete physical meaning, i.e. each of the functionalities can be deployed in different physical assets, being the logical aggregation of the aforementioned SWIM Node).

As described in SWIM Supervision ConOps ([15]), SWIM Supervision it's intended to be described from different perspectives: the L1 one (in which the SWIM TI L1 Supervision deals with the functions of one SWIM node<sup>83</sup>) and the L2 one (in which the SWIM TI L2 Supervision deals with the functions of one or more SWIM TI L1 Supervisions<sup>84,85</sup>).

#### D.1.1 SWIM-TI L1 Supervision

SWIM-TI L1 Supervision is the Supervision that deals with the functionalities/capabilities that, through a SWIM-TI Node, enable a Stakeholder to access to the SWIM-TI.

##### D.1.1.1. Architectural options

The following options are considered as different ways for a SWIM-TI L1 Supervision to be applied:

- One SWIM Node providing access to the SWIM-TI to a System. One SWIM-TI L1 Supervision associated to that SWIM Node.

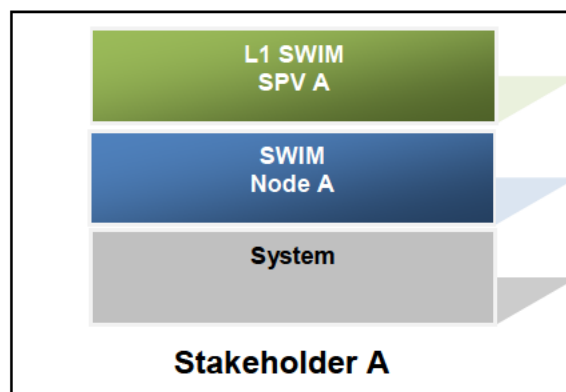


Figure 93 – One SWIM Node for SWIM-TI L1 Supervision

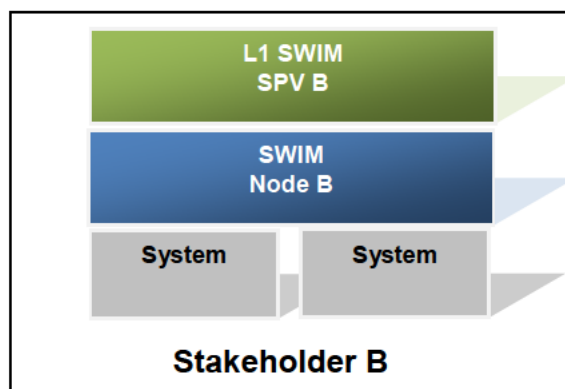
This option considers a SWIM-TI L1 Supervision associated to a single SWIM-TI Node. This SWIM-TI Node integrates, into SWIM TI, only one ATM System. The Stakeholder A has an ATM System integrated into SWIM infrastructure through the SWIM Node A, supervised by its SWIM-TI L1 Supervision.

<sup>83</sup> L1 is often associated with the Local level, being this "local" concept frequently associated to the point of access from a concrete stakeholder to the SWIM TI. As a reminder, the SWIM Node concept doesn't impose deployment options,

<sup>84</sup> Initially this concept was associated to the FAB and the sub-regional concept. However, FAB has a very concrete meaning and SWIM TI Supervision aims at a more general (less constrained) concept. This doesn't prevent the fact that if, at a certain point a L2 SWIM Technical Supervision is needed for a FAB, L2 would be applied.

<sup>85</sup> Initially a L3 SWIM TI Supervision was foreseen, dealing with the management of the whole SoS SWIM TI, however, this concept isn't mature enough and it's not included in the SWIM ConOps ([12]).

- One SWIM Node providing access to the SWIM-TI to more than one ATM Systems. One SWIM-TI L1 Supervision associated to that SWIM Node.

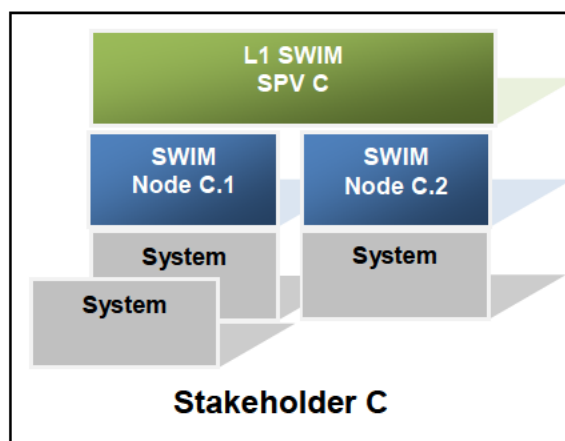


**Figure 94 – One SWIM Node for one or more connected ATM Systems and one SWIM-TI L1 Supervision**

This option considers a SWIM-TI L1 Supervision associated to a single SWIM-TI Node. This SWIM-TI Node integrates, into SWIM TI, more than one ATM System (e.g. a stakeholder has its internal middleware that enables the internal information exchanges and has deployed only one SWIM-TI Node to enable the information exchange between different Stakeholders in the SoS). The Stakeholder B has two ATM Systems integrated into SWIM-TI through the SWIM Node B, supervised by its SWIM-TI L1 Supervision.

Each SWIM Node could integrate, into SWIM Infrastructure, one or more ATM Systems. This assumption has been made to avoid any restriction of the SWIM Design.

- More than one SWIM Nodes for one or more connected ATM Systems and one SWIM-TI L1 Supervision associated to that SWIM Node.



**Figure 95 – One or more SWIM Node for SWIM-TI L1 Supervision**

This option considers a SWIM-TI L1 Supervision associated to several SWIM-TI Nodes. The same stakeholder accesses to SWIM-TI through several SWIM Nodes (e.g. a stakeholder has decided to apply a distributed pattern to the architecture of its Systems and enables the information exchanges through SWIM from several access points, one per system. However, it also intends to manage all the access from the same centralized SWIM-TI L1 Supervision).

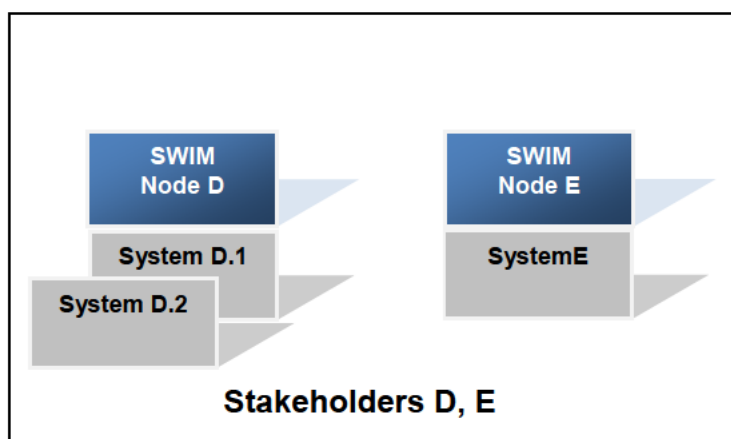
This is the case of a system, connected to others by SWIM, with a SWIM Local SPV and whose nodes aren't centralized, that can rely on only one Local SWIM SPV. This has to be flexible and to satisfy the needs of different stakeholders.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

- No SWIM-TI L1 Supervision associated to the SWIM Node/s.



**Figure 96 – Different SWIM Node deployments without SWIM-TI L1 Supervision**

This option considers those Stakeholders that for concrete reasons don't intend to identify a SWIM-TI L1 Supervision associated to their SWIM-TI Nodes. It doesn't prevent the existence of internal solutions to cope with these Stakeholders' management needs.

## D.1.2 SWIM-TI L2 Supervision

SWIM-TI L2 Supervision is the Supervision that owns the ability of managing<sup>86</sup> different SWIM TI L1 Supervisions.

The scope of SWIM-TI L2 Supervision is to coordinate<sup>87</sup> different SWIM-TI L1 Supervisions.

### D.1.2.1. Architectural options

The following options are considered as different ways for a SWIM-TI L2 Supervision to be applied:

- SWIM-TI L2 Supervision dealing with one or more SWIM-TI L1 Supervisions (regardless of how these SWIM-TI L1 Supervisions are).

SWIM-TI L2 Supervision deals with all aspects concerning the supervision of a set of sites (e.g. Aerodrome, TMA, ACC, APP), through the collection of the supervision messages/information/data relevant to the SWIM Technical Infrastructure.

<sup>86</sup> The specific meaning of what "managing" means is described in the functional chapter. In this case, it only refers to the elements that it manages.

<sup>87</sup> As the previous note, the specific meaning of "coordination" is described in the functional chapter.

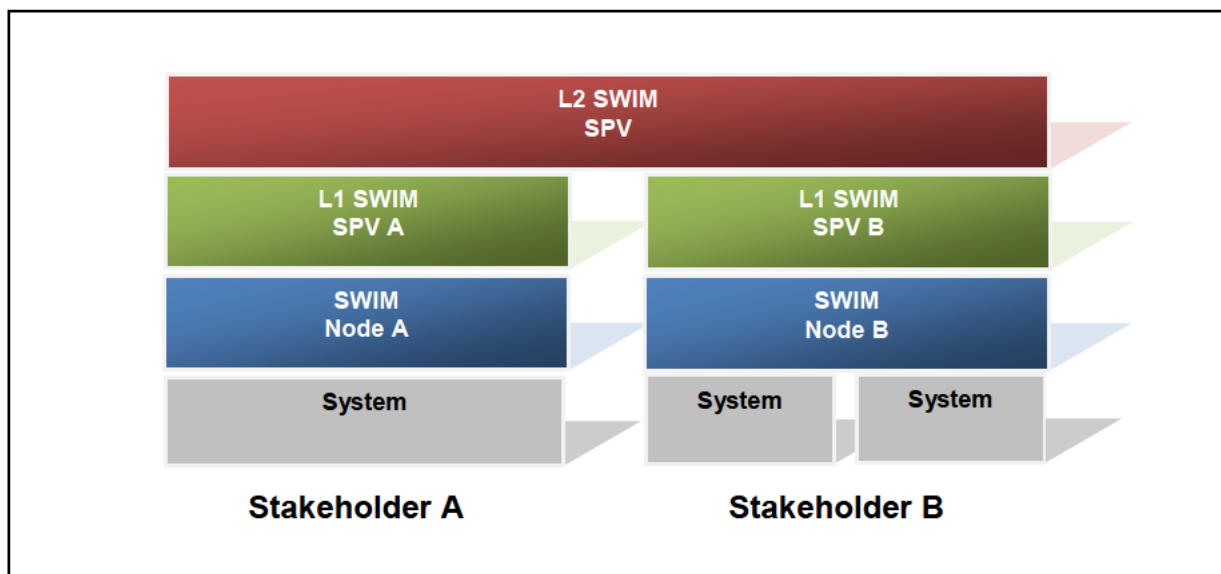


Figure 97 – SWIM-TI L2 Supervision

- No SWIM-TI L2 Supervision.

The SWIM Node (independently if it implements a SWIM-TI L1 Supervision) doesn't share any SWIM-TI information with other SWIM Nodes.

## D.2 SWIM-TI Supervision Services

As stated in the SWIM-TI Supervision Functional Decomposition, SWIM-TI SPV will provide the following major functionalities:

- Configuration Management
- Fault Management
- Performance Management
- Security
- Legal Recording
- Safety

These functionalities could be exposed as services to other SWIM-TI Supervisions, in order to enable the relationship between them. Depending on the different relationships (e.g. between two SWIM-TI L1 SPV or between L1 and L2), there would be different roles that will characterize the access to the services (enabling, disabling the access to them).

The roles for accessing to the services are proposed in the SWIM Supervision ConOps ([15]). Also following what is stated in 8.3.1 Concept of Operations SWIM Supervisor is responsible for:

- the management and supervision of SWIM Infrastructure and its SWIM Nodes locally if L1 is local or remotely controlled if it is partially unmanned or unmanned. In case of fault or bad conditions, he will take the appropriate recovery actions.
- configuration and performance management (monitor SLAs)
- security aspects (SWIM infrastructure administration, providing user rights)

Table 38 – SWIM-TI Supervision responsibilities

This option is stated in SWIM Supervision ConOps as:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu



	Level 1	Level 2	Level 3
Fault Management	✓		
Performance Mgmt	✓		
Configuration Mgmt	✓		
Security Mgmt	✓		
Safety Mgmt	✓		
Legal recording	✓		

**Table 39 – SWIM-TI Supervision L1 responsibilities**

Where only SWIM-TI Supervision is implemented at Level L1 and the access to the exposed services is only allowed between different L1 SWIM-TI Supervision (e.g. when there are several SWIM-TI Nodes under the same ANSP responsibility).

However, and according to the possible hierarchy described in this appendix:

SWIM-TI Supervision could deal with the Supervision of a L2 Level (relationship between different SWIM-TI Nodes)

L2 SWIM Supervision deals with all aspects concerning the supervision of a set of sites (e.g. Aerodrome, TMA, ACC, APP), through the collection of the supervision messages/information/data relevant to the SWIM Technical Infrastructure.

For these cases where also L2 is considered<sup>88</sup>, the roles for accessing the services are as follows:

	Level 1	Level 2	Level 3
Fault Management	✓	<i>Only monitoring</i>	
Performance Mgmt	✓	✓	
Configuration Mgmt	✓	✓	
Security Mgmt	✓	✓	
Safety Mgmt	✓	✓	
Legal recording	✓	✓	

**Table 40 – SWIM-TI Supervision L1 + L2 responsibilities**

Where the Fault Management functionality associated to SWIM-TI L2 Supervision is described as “Only Monitoring”. As described, Fault Management deals with Control Functions (the one that assumes the responsibility of managing the Lifecycle for the supervised entities). The characterization of the role as “Only Monitoring” means that from the set of services defined (related with Fault

<sup>88</sup> Note that what is stated in 8.3.1 Concept of Operations refers also to L3 Supervision, but since this supervision is not foreseen, hasn't been included in the table.

Management) only those that doesn't deal with managing the Lifecycle will be accessible for SWIM-TI L2 Supervision (e.g. those that request the status of a supervised entity).

Each of this services will be further described by the activities of Service Identification that aligns with the method defined in Working Method on Services ([16]).



## Appendix E Communication related Ontology, Terminology, Relationships and Semantics

### E.1 Classification

#### E.1.1 Introduction

Communication is a key capability of the SWIM-TI. There are several issues related to communication middleware that jeopardise a shared understanding:

- There exists no single authoritative classification system for communication middleware.
- In literature and in communication standards and technologies, different terms are used for overlapping and/or identical semantics and identical terms are used with different semantics.
- Some definitions and/or specifications are vague, thus leaving room for diverging interpretations.
- Concrete communication standards and technologies do not typically refer or map their capabilities to a shared or shareable classification system.

Because of those issues, the interpretation and understanding of WP14 text related to the communication middleware may vary strongly between individuals each having their specific background.

The purpose of this Appendix is not to provide the ultimate exhaustive ontology for communication middleware but it targets at identifying and clarifying key structuring elements for communication middleware to be used in a consistent manner by the Stakeholders that contribute and review the WP14 deliverables as well as to support the integration of WP14 activities with other WP activities and to support and to facilitate the assimilation of the content of WP14 deliverables by any other Stakeholders.

The focus of this Appendix is on the elements in communication middleware that are relevant for the SWIM-TI to date: only these elements are elaborated. The structure is intended to allow further elaboration for other elements in case that should become necessary at some point in time. Quotations refer to [49], [50] and [51].

#### E.1.2 Key characteristics

##### E.1.2.1 Decoupling

The notion of decoupling to characterise communication is common across textbooks and articles.

Three criteria (using the naming by Eugster) are recurring.

- Time decoupling,
- Space decoupling,
- Synchronization decoupling.

Other criteria such as semantic decoupling can be observed but their use is less wide spread.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

235 of 284

## Time decoupling

The notion of time decoupling also appears under the name temporal decoupling. The semantics of time decoupling is used in a uniform manner.

Eugster et al:

Time decoupling: The interacting parties do not need to be actively participating in the interaction at the same time.

Tanenbaum and Van Steen:

Temporal coupling means that processes that are communicating will both have to be up and running.

Coulouris et al:

Time uncoupling ..., the sender and receiver(s) do not need to exist at the same time to communicate.

## Space decoupling

The notion of space decoupling also appears under the name referential decoupling. The semantics of distinct instances, typically share:

The sender of a message does not need to know the receiver(s) of the message and the receiver of a message does not have to know the sender of the message.

Slight variations of semantics can exist:

In some definitions space decoupling also includes the absence of knowledge about the number of participants.

Eugster et al:

Space decoupling: The interacting parties do not need to know each other.

Tanenbaum and Van Steen:

In referentially decoupled systems, processes do not know each other explicitly.

...

This is also referred to as being decoupled in space, or referentially decoupled.

Coulouris et al:

Space uncoupling, in which the sender does not know or need to know the identity of the receiver(s), and vice versa.

## Synchronization decoupling

The notion of synchronization decoupling also appears under the names flow decoupling and thread decoupling. The semantics are not always the same.

Eugster et al:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

Synchronization decoupling: Publishers are not blocked while producing events, and subscribers can get asynchronously notified (through a callback) of the occurrence of an event while performing some concurrent activity.

Tanenbaum and Van Steen:

"synchronous communication": the sender is blocked until its request is known to be accepted. There are 3 possible synchronization points ranging from delivery to the middleware over delivery to the intended recipient to reception of a response to a request.

"Asynchronous communication": a sender continues immediately after it has submitted its message for transmission.

## E.1.2.2 Persistence

Tanenbaum and Van Steen:

"With persistent communication, a message that has been submitted for transmission is stored by the communication middleware as long as it takes to deliver it to the receiver"

"with transient communication, a message is stored by the communication system only as long as the sending and receiving application are executing"

## E.1.2.3 Discrete/Streaming

Tanenbaum and Van Steen:

Discrete communication: each message forms a complete unit of information

Streaming communication: multiple related messages. The relationship is through the order of sending or is temporal.

## E.1.2.4 Classifications

### Synchronization and persistence

Tanenbaum and Van Steen:

	Synchronous	Asynchronous
Persistent		Message Queuing
Transient	RPC	

### Time and Space

Tanenbaum and Van Steen:

	Temporal (time) coupled	Temporal (time) decoupled
Referential (space) coupled	Direct, analogous to transient message oriented communication	Mailbox
Referential (space) Decoupled	Meeting oriented	Generative Communication

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

237 of 284

## E.1.2.5 Cardinality

## E.1.2.6 Coordination

### MEP

The MEPs are described extensively in the body of this document.

## E.1.3 High level structure

### E.1.3.1 RPC

Tanenbaum and Van Steen

allowing programs to call procedures located on other machines.

Coulouris and al.

in RPC, procedures on remote machines can be called as if they are procedures in the local address space.

### E.1.3.2 Message oriented

Wikipedia

“message passing sends a message to a process (which may be an actor or object) and relies on the process and the supporting infrastructure to select and invoke the actual code to run”.

#### Message oriented Transient

Tanenbaum and Van Steen

Communication that conforms to the characteristics of message passing and that requires sender and receiver to be active at the same time.

This category is further sub-divided in synchronous and asynchronous.

#### Message oriented Persistent / Message Queuing

Tanenbaum and Van Steen

Message-queuing systems provide extensive support for persistent asynchronous communication. The essence of these systems is that they offer intermediate-term storage capacity for messages, without requiring either the sender or receiver to be active during message transmission.

Coulouris and al.

A message queue is a concept that provides space and time uncoupling between participants.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

238 of 284

## E.1.3.3 Streaming

Tanenbaum and Van Steen

Streaming reflects a form of communication based on a continuous flow of messages subject to various timing constraints.

## E.2 Specific notions

### E.2.1 Multicast

The notion of Multicast encompasses different technologies or types of technologies that may be interlinked but that are different. The semantics of the terminology related to Multicast can vary significantly amongst suppliers and textbook authors and it can therefore be a major source of ambiguity and confusion.

The interpretation of the notion Multicast as a pattern varies between a form:

- of 1 to many communication only
- of 1 to many communication as well as many to many communication

Within the ISO OSI model:

- Multicast occurs at distinct OSI layers:
  - above Layer 3: multicast in an overlay network
  - Layer 3: network layer (e.g. IP Multicast)
  - Layer 2: data link layer (e.g. Ethernet Multicast)
- multiple distinct forms of Multicast at distinct layers can be combined (e.g. Universal Multicast)

A set of varying terminology is used in the context of multicast in an overlay network.

- an overlay network:
  - a virtual network on top of an existing network,
  - typically provides functionality that is not available in an existing network,
  - can provide Multicast:
    - where an existing network does not support multicast and/or specific features are not available in the existing network such reliability and ordering,
    - moves the dependency from routers to hosts.
- a range of terminology can be found, for which the exact scope or the border between category and instance are often unclear:
  - end system multicast (ESM),
  - overlay multicast (OM),
  - application layer multicast (ALM),
  - application level multicast (ALM),
  - host level multicast,
  - end-host multicast.

Multicast can be supported directly by a network technology or can be emulated.

- in one context the notion of Multicast refers to sending of a message through a single operation only,
- in another context, the notion of Multicast also allows implementation through a series of unicasts.

Various levels of Quality of Service are associated with Multicast:

- levels (Coulouris et al. Distributed Systems, Concepts and Design)
  - no explicit mention in particular whereby it may not be clear what is implied,

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

- reliable #1 (Coulouris et al. Distributed Systems, Concepts and Design):
- integrity: message delivered is the same as the one sent and no messages are delivered twice,
- validity: any outgoing message is eventually delivered,
- agreement: a message will be delivered to all members of a group or to none.
- reliable #2 (Tanenbaum et Van Steen, Distributed System, Principles and Paradigms):
  - weak: a message will be delivered to all current members of a group,
  - atomic: a message will be delivered to all current members of a group or to none. The messages will be delivered in the same order to all member of a group.
- ordered:
  - even with support of the Network, the messages can arrive in a different order than sent.
  - various sublevels of ordering can be distinguished (FIFO, causal, totally)
- a range of specific protocols has been defined to deal with various QoS:
  - for instance a protocol built on top of UDP over IP Multicast, to provide reliability and/or ordering
  - example DDSI/RTPS

Every mention of the notion Multicast, requires an explicit qualification on all the elements above to ensure a consistent interpretation.

## E.2.2 Message Broker

As discussed in the overall Ontology and terminology, the notion Broker is generic.

In the context of communication middleware the notion of Broker can be defined as a third party that provides a service that facilitates and/or allows the communication between multiple participants in a communication.

- The presence of a third party is often not required to allow communication between multiple participants in a communication. In such cases, the introduction of a third party often serves one or more forms of decoupling between the participants in a communication.
- In some cases, the presence of a third party is required to allow communication between multiple participants in a communication.

There exists a large amount terminology related to the notions Broker and Messaging.

- The semantics can be different case by case and there exists no overall standardized terminology.
- The terminology is typically local to a specific solution. For instance the authors of textbooks tend to limit the functionality covered by a Message Broker to Message transformation and Event mediation in a Publish/Subscribe context.
- While some suppliers include typically much more functionality under this notion. For instance IBM includes a significant amount of functionality typically based on Enterprise Architecture Integration patterns in the Websphere Message Broker product.

To allow for an unambiguous requirement in such context, it is necessary to identify and define the functions that can be covered by the notion of Broker in the context of Messaging. Each use of the term Message Broker, is then accompanied with an explicit list of functions that are implied it its semantics.

- Message transformation (narrowest interpretation, Coulouris et al. and Tanenbaum et al.),
- Event mediation in Publish/Subscribe (Tanenbaum et al.),
- Routing (eapatterns, [http://en.wikipedia.org/wiki/Message\\_broker](http://en.wikipedia.org/wiki/Message_broker)),
- Protocol translation (eapatterns),
- Persistence (Queuing Layer, ActiveMQ),
- Filtering (Websphere Message Broker),

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

240 of 284



- Aggregation (Websphere Message Broker, [http://en.wikipedia.org/wiki/Message\\_broker](http://en.wikipedia.org/wiki/Message_broker)),
- Collection (Websphere Message Broker),
- Enrichment (Websphere Message Broker),
- Bridging of messaging across security realms (Microsoft),
- Message encapsulation.

## E.2.3 Message Broker versus Enterprise Service Bus

Not only is the notion of Message Broker vague, there is additionally a significant amount of confusion around the concepts Message Broker versus Enterprise Service Bus (ESB). There is no uniform understanding of the exact difference between them:

- According to some, the difference between both consists of a central engine that does everything in a Message Broker versus a central engine in an ESB with a much lesser scope and distribution of functionality on autonomous and separately deployable components (<http://www.mulesoft.com/resources/esb/enterprise-application-integration-eai-and-esb>)
- According to others the key difference lies in accountability (<http://www.udidahan.com/2011/03/24/bus-and-broker-pubsub-differences/>).
- Wikipedia confirms the confusion ([http://en.wikipedia.org/wiki/Enterprise\\_service\\_bus#Ambiguous\\_use\\_of\\_the\\_term\\_ESB\\_in\\_commerce](http://en.wikipedia.org/wiki/Enterprise_service_bus#Ambiguous_use_of_the_term_ESB_in_commerce))
- IBM states that their Message Broker is an ESB : "WebSphere® Message Broker is an Enterprise Service Bus (ESB) built for universal connectivity and transformation in heterogeneous IT environments" (<http://www.ibm.com/developerworks/downloads/ws/wmb/>).
- Illustrative for the confusion:
  - In December 2012, IBM WebSphere Enterprise Service Bus is discontinued and transferred to WebSphere Message Broker (<http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpatem&supplier=897&letternum=ENUS212-522>)
  - As of mid 2013, IBM WebSphere® Message Broker has itself become "IBM® Integration Bus". The latter is qualified as ESB

## E.2.4 Messaging bus

In some documentation related to DDS, the term "Messaging bus" can be found. The notion Messaging bus is however nowhere defined.

A similar term can be found in an article at IBM, that seems to be entirely unrelated (<http://java.boot.by/ibm-257/ch04s02.html>):

"The service integration bus is sometimes referred to as the messaging bus if it is used to provide the messaging system for JMS applications using the default messaging provide."

Unless an explicit definition is referenced the notion of "Messaging bus" must not be used

## E.2.5 Terminology to use

Considering the confusion and the absence of a standardized nomenclature, any use of terminology such a Message Broker, Enterprise Service Bus, Messaging bus alone is meaningless.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

241 of 284

Whenever it is necessary to use such term, it will be accompanied by a description of the functions it offers as well as a technical architecture view that allows to understand its specificities.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

242 of 284

## Appendix F AMQP v1.0

### F.1 The problem/need

#### F.1.1 Proprietary transport protocols for “Asynchronous Messaging”

The term “Asynchronous Messaging” captures messaging that is characterised by time decoupling of both sender and receiver. Depending on the author, the notion “Asynchronous Messaging” may additionally include synchronization decoupling at the message sender side.

“Asynchronous Messaging” systems are widely used and successful. They support a wide/flexible range of Message Exchange Patterns (MEPs) as well as Quality of Service but are typically using proprietary transport protocols for messaging.

Examples:

- OpenWire for ActiveMQ,
- T3/T3S for Oracle Weblogic.

The protocols in the examples above are currently retained in the FAA SWIM for the Publish/Subscribe (terminology used by FAA) style of MEPs.

Because of their proprietary nature and limited support, these protocols are not retained for use in SESAR SWIM.

For the sake of clarity it is necessary to point out that according to the classification by Patrick Eugster et Al. in the “Many faces of Publish/Subscribe” that the MEPs provided by FAA do not qualify as “Publish/Subscribe” MEPs because of the synchronization coupling at the receiver side, but rather as a “Message Queuing” MEP and alike.

#### F.1.2 Standards-based transport protocols are not appropriate for “Asynchronous Messaging”

##### F.1.2.1 Introduction

A number of standardized transport protocols exist and could be used for “Asynchronous Messaging” but their specificities limit their applicability.

##### F.1.2.2 X.400

The X.400 standard is convenient for a wide range of use cases.

From a functional point of view, X.400 can be used natively and/or combined with other protocols, to provide high reliability and transaction.

The X.400 standard has seen significant adoption. Examples:

- X.400 was the native communication protocol between Microsoft Exchange Servers.
- Also, X.400 is the base for the AMHS standard.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

243 of 284

The major draw-back of X.400 is its increasing obsolescence and fading adoption:

- While support for X.400 is still available in current Exchange Server platforms, since Microsoft Exchange Server 2007 it is no longer the native messaging protocol.
- In June 2012, Eurocontrol asked Gartner about the status and prospects for X.400. From Gartner's point of view, X.400 is not a choice for the future:
  - Support for X.400 is available on all major platforms and will likely remain supported on those platforms for a significant time. However no evolution/extension of support to new platforms should be expected.
  - Gartner searched its own archives and found not a single request re. X.400 from any of their customers in the 10 years before June 2012. Gartner indicated this to be strong signal of the obsolescence.
  - Gartner indicated that finding X.400 expertise is difficult today and will become much more difficult in future.

### F.1.2.3 DDS-I over RTPS (DDS Interoperability Wire Protocol (DDSI-RTPS))

DDSI-RTPS has a focused domain of applicability: data centricity. It has native support and is suitable for MEPs of the Publish/Subscribe style. However DDSI-RTPS does not (yet) standardize the MEPs of the Request/Reply style. Nevertheless, similar to what is applicable to the SOAP standard MEPs, any MEP can be emulated on top of another MEP and in the case of DDSI-RTPS requires a complex construction on top of a Publish/Subscribe style MEP.

An example of an appreciation by the CEO of RTI, a prominent vendor of DDS based solutions, of the ability of emulation of some behaviours over DDS which are considered not worth the investment can be found at (<http://electronicdesign.com/embedded/what-s-difference-between-dds-and-amqp>). This consideration is not limited to DDS technology but applies in general to emulation of behaviours on top of the natively supported behaviours of technologies.

DDSI over RTPS cannot provide strong consistency. The highest Quality of Service that can be made available is eventual consistency. Eventual consistency is weaker than strong consistency.

### F.1.2.4 SOAP over HTTP

The binding of SOAP over HTTP is standardized for 2-way Request/Reply style MEPs and 1-way Fire & Forget style MEPs but it is only suitable for those cases where time-coupling is acceptable or required.

Time decoupled messaging can be provided technically and in a standardized manner but only through a complex construction on top of SOAP over HTTP using a composition with WS-Notification Pull Point. For time decoupled messaging between a message producer and a message receiver this means emulation of a Request/Reply style MEP or a Fire & Forget style MEP on top of a Publish/Subscribe style MEP.

For instance in case of an emulation of a time decoupled Request/Reply style MEP 2 Pull points in a Publish/Subscribe style MEP can be used :

- The Service Consumer sends the Request message to a Topic and from there the Request message is forwarded to a Service Provider related Pull point. The Service Provider related Pull point is read by the Service Provider at any time the Service Provider is ready to do so.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

244 of 284

- The Service Provider sends the Reply message to a Topic and from there the Reply Message is forwarded to a Service Consumer related Pull point. The Service Consumer related Pull point is read by the Service Consumer at any time the Service Consumer is ready to so.

Above example is high level and schematic: the effective use entails more complexity such as definition, coordination and management (CRUD of permanent and/or short-lived topics) related to the topics, and the operations related to the subscription management.

Composition of distinct standards of the WS-\* family can provide reliability and transaction. However the use of such constructs is marginal and further increases the complexity.

### F.1.2.5 SOAP over Email

The W3C Note, SOAP Version 1.2 Email Binding, 3 July 2002 describes a method to use email as a complementary binding to the SOAP over HTTP binding, that can be useful for cases where the SOAP over HTTP binding is not appropriate: the domain of “asynchronous messaging”. The Note does not address concrete email protocols such SMTP but stays at a higher level of abstraction.

The W3C Recommendation, SOAP Version 1.2 Part 0: Primer (Second Edition) 27 April 2007, has a section that deals with SOAP over email (SOAP Over Email) and therein provides examples including SMTP. However these examples are explicitly stated to reflect only how the SOAP Binding Framework could be used and not to reflect a standard.

Solutions can be found that provide for SOAP over Email claim

(SMTP in particular). However the use of the SOAP over SMTP binding is marginal. Also, as the reliability features of SMTP are very limited, the SOAP over SMTP requires reliability to be handled at message level protocol rather than at transport level.

The limitation to SOAP 1.2, the absence of a normative W3C text and limited practical use render SOAP over Email not appropriate.

### F.1.2.6 SOAP over JMS

The W3C Recommendation, SOAP over Java Message Service 1.0, 16 February 2012, describes a binding that is appropriate for the domain of “asynchronous messaging”. It also supports both SOAP 1.1 and SOAP 1.2.

However the specification explicitly states that the wire protocol for SOAP/JMS messages is out of scope. Absence of a binding to a standardized wire protocol renders the specification inappropriate.

## F.2 Candidate standards

### F.2.1 AMQP v1.0

OASIS in October 2012 and ISO/IEC standard as of May 1<sup>st</sup>, 2014.

### F.2.2 XMPP

IETF specification, proposed standards RFC 6120, RFC 6121 and RFC 6122 published in March 2011

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

## F.2.3 MQTT 3.1.1

OASIS standard as of November 13th, 2014.

## F.3 Previous work on AMQP in SESAR

### F.3.1 WP14.1.2

Analysis work has been performed in WP14 before Iteration 2.0 on various messaging technologies. This included the status of AMQP 1.0 as it was known at that time: it was not yet an OASIS standard. At that time no strong recommendations could be given in the matter of AMQP 1.0.

### F.3.2 Other WP

AMQP 0.9.1 has been adopted by WP9.19.

AMQP 0.9.1 is a pre-version 1.0 specification by amqp.org.

Despite being a pre-version 1.0, a consensus was found amongst distinct organisations: AMQP 0.9.1 was effectively supported by multiple players and is used operationally.

## F.4 What is new

A number of significant events have taken place since the previous analysis work has been performed in WP14 and the selection of previous versions of AMQP in other WPs.

- From a standardization point of view:
  - The OASIS standard
  - The ISO/IEC JTC1 standard
- Commercial Adoptions:
  - Microsoft “Windows Azure Service Bus” general availability on May 23<sup>rd</sup>, 2013
  - SwiftMQ as of version 9.0.0
  - IBM “MQ Light” general availability with MQ Light as September 30th 2014.
    - AMQP v1.0 versus MQTT by IBM at <https://developer.ibm.com/messaging/2014/02/14/mq-light-wire-protocol/>
- Open source Adoptions:
  - Apache (Qpid, ActiveMQ and Apollo)
  - HornetQ in its version 2.4 published in December 2013
- Intentions

## F.5 Outlook

### F.5.1 Extensions

The AMQP 1.0 standard issued by OASIS bears the term “core” in its name.

Microsoft has published an overview of ongoing work on extensions for AMQP v1.0 at <http://msopentech.com/blog/2013/06/28/extensions-and-binding-updates-for-business-messaging-open-standard-spec-oasis-amqp/>.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

246 of 284

Public information on the progress of this work at OASIS can be found at <https://lists.oasis-open.org/archives/amqp-comment/> and trace of these activities can be found at OASIS itself.

Of the published extensions 2 have a particular interest:

- Global Addressing
- Management

## F.5.2 Bindings

Amongst the targeted bindings, a working group at OASIS works on standardization of JMS over AMQP v1.0.

## F.5.3 Intentions

The IBM “Statement of Direction” linked with WebShpere MQ Version 8 as of April 22th, 2014, can be interpreted as a possibility for inclusion of AMQP v1.0 in a future version of WebShpere MQ.

## F.6 Comparison between AMQP 0-9-1 and AMQP 1.0

This section provides a basic comparison of the two versions 0-9-1 and 1.0 of the Advanced Message Queueing Protocol (AMQP). The comparison has been performed along following criteria:

- Scope and Features
  - What is the scope of each protocol version?
  - Which features are provide by each protocol version?
- Adoption of the standards
  - How are the protocol versions adopted by the industry and by open source solutions?
- Standardization
  - Which standards are applicable for the two protocol versions?

### F.6.1 Comparing Scope and Features

This section tries to compare the general scope of the protocol versions and their features. At a high level, the following table expresses that the two versions have a different scope: while both versions specify an interoperability wire protocol, the version 0-9-1 in addition defines broker behaviour. With respect to the different scope, there existed some “religious” discussions in the internet some years ago; however, these discussions have ceased since quite some time now.

#### F.6.1.1 General Scope

Category	Description	AMQP 0-9-1	AMQP 1.0
Interoperability wire protocol for messaging	The version specifies a wire protocol for messaging	Yes	Yes

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

247 of 284

Category	Description	AMQP 0-9-1	AMQP 1.0
Broker behaviour	The version includes specifications for broker behaviour	Yes	No
Portability API	The version includes specifications for a application layer interface	No	No
Extensibility	The version is designed for compatibility with future enhancements	Yes	Yes
Backward compatibility	The version is backward compatible	"Backwards compatible with 0-9 Not compatible with 0-8"	No
Forward compatibility	The version supports forward compatibility	No	Yes <sup>89</sup>

## F.6.1.2 Individual Features

With regards to the feature comparison, the two AMQP versions are only comparable to a certain extent, because their scope is different. A detailed comparison of individual features has not been performed.

## F.6.2 Comparing the Adoption of Standards

This section provides a comparison of the two protocol versions by illustrating their adoption in different areas. The aspect of adoption of a standard provides valuable information about the standard.

### F.6.2.1 Broker Adoption

The table below lists broker solutions supporting the AMQP standards.

Category	Description	AMQP 0-9-1	AMQP 1.0
----------	-------------	------------	----------

<sup>89</sup> see for example <http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf> (section 2.8.11) and <https://www.oasis-open.org/committees/amqp/charter.php> (section "Interoperability")



Category	Description	AMQP 0-9-1	AMQP 1.0
Commercial solutions		(IIT Software, SwiftMQ Router) <sup>90</sup>	Microsoft Azure Service Bus Microsoft Service Bus for Windows Server IBM MQ Light IBM MQ Light for Bluemix Red Hat Fuse Red Hat A-MQ 6.1 IIT Software, SwiftMQ Router
Open source solutions		RabbitMQ Apache QPID	Apache ActiveMQ Apache QPID Apache Apollo HornetQ (Red Hat MRG) <sup>91</sup> (RabbitMQ) <sup>92</sup>

## F.6.2.2 Client Adoption

The table below lists client solutions supporting the AMQP standards

Category	Description	AMQP 0-9-1	AMQP 1.0
Commercial solutions		RabbitMQ JMS Client	Service Bus .NET API IIT Software, SwiftMQ AMQP 1.0 Java Client

<sup>90</sup> Only partial AMQP 0-9-1 support: "implements ... the most relevant parts of the AMQP 0.9.1 specification." (see <http://swiftmq.com/products/index.html>)

<sup>91</sup> No support for transactions

<sup>92</sup> An "experimental prototype" with limited functionality, see <https://github.com/rabbitmq/rabbitmq-amqp1.0>

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

Category	Description	AMQP 0-9-1	AMQP 1.0
Open source solutions		RabbitMQ: - Java Client - Client for Scala - .NET Client - C Client - and many more: see <a href="http://www.rabbitmq.com/devtools.html">http://www.rabbitmq.com/devtools.html</a>  MuleSoft Anypoint AMQP Connector	Apache QPID Proton  Node-Amqp10 Node.js  Amqp.Net Lite 1.0.0
Programming Language/Run-time Environment	Supported programming languages	Java, C#, C/C++, Ruby, Python, Perl, PHP, Erlang, Haskell	Java, C#, C/C++, Ruby, Python, Perl, .Net

### F.6.2.3 Supporting Tools

The table below lists additional relevant information about standard adoption.

Category	Description	AMQP 0-9-1	AMQP 1.0
Network tracing	Network tracing tools supporting the standard.	Wireshark development release 1.99.2 since 2015-02-20	Wireshark since 20-01-2014

### F.6.3 Comparing the Standardization

This section lists standards related to the individual AMQP versions.

Category	Description	AMQP 0-9-1	AMQP 1.0
Core Standards	Standards defining the protocol	amqp.org November 13th, 2008 (AMQP 0-9-1 Specification) <sup>93</sup>	ISO/IEC 19464:2014 May 1st, 2014 (AMQP 1.0 ISO Standard) <sup>94</sup>  OASIS October 29th, 2012 (AMQP 1.0 OASIS Standard) <sup>95</sup>

<sup>93</sup> AMQP 0-9-1 Specification: <http://www.amqp.org/specification/0-9-1/amqp-org-download>

<sup>94</sup> AMQP 1.0 ISO Standard: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=64955](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=64955)

<sup>95</sup> AMQP 1.0 OASIS Standard <http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf>

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

250 of 284

Category	Description	AMQP 0-9-1	AMQP 1.0
Additional Standards	Additional standards related to the protocol versions	None	JMS over AMQP 1.0 (pending OASIS) Global Addressing (pending OASIS) SOAP over AMQP 1.0 (pending OASIS) Management (pending OASIS)

## F.6.4 Additional Information

The table below provides a collection of references used for the comparison.

Protocol	Category	Reference	Comment	Date
<b>Specifications</b>				
AMQP 0-9-1		<a href="http://www.amqp.org/specification/0-9-1/amqp-org-download">http://www.amqp.org/specification/0-9-1/amqp-org-download</a>	AMQP 0-9-1 Specification	
AMQP 1.0		<a href="http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf">http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf</a>	AMQP 1.0 OASIS Standard	
AMQP 1.0		<a href="http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=64955">http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=64955</a>	AMQP 1.0 ISO Standard	
AMQP 1.0		<a href="http://www.amqp.org/sites/amqp.org/files/amqp.pdf">http://www.amqp.org/sites/amqp.org/files/amqp.pdf</a>	Original AMQP 1.0 Specification	Oct 07, 2011
<b>Charter</b>				
AMQP 1.0		<a href="https://www.oasis-open.org/committees/amqp/charter.php">https://www.oasis-open.org/committees/amqp/charter.php</a>	OASIS AMQP 1.0 Technical Committee Charter	Oct 05, 2012
<b>Support</b>				
AMQP 0-9-1	RabbitMQ	<a href="http://lists.rabbitmq.com/pipermail/rabbitmq-announce/2010-August/000028.html">http://lists.rabbitmq.com/pipermail/rabbitmq-announce/2010-August/000028.html</a>	AMQP 0-9-1 in RabbitMQ 2.0.0	Aug 25, 2010
AMQP 1.0	RabbitMQ	<a href="https://github.com/rabbitmq/rabbitmq-amqp1.0">https://github.com/rabbitmq/rabbitmq-amqp1.0</a>	AMQP 1.0 in RabbitMQ Experimental and limited	

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

251 of 284

Protocol	Category	Reference	Comment	Date
AMQP 0-9-1	SwiftMQ Swiftlet	<a href="http://www.swiftmq.com/products/router/swiftlets/sys_amqp/0_9_1_support/index.html">http://www.swiftmq.com/products/router/swiftlets/sys_amqp/0_9_1_support/index.html</a> <a href="https://lists.oasis-open.org/archives/amqp/201204/msg00037.html">https://lists.oasis-open.org/archives/amqp/201204/msg00037.html</a> <a href="http://www.swiftmq.com/products/index.html">http://www.swiftmq.com/products/index.html</a>	AMQP 0-9-1 SwiftMQ 9.1.0 limited	April 23, 2012
AMQP 1.0	SwiftMQ Swiftlet	<a href="https://www.amqp.org/node/83">https://www.amqp.org/node/83</a> <a href="http://www.theserverside.com/news/thread.tss?thread_id=63459">http://www.theserverside.com/news/thread.tss?thread_id=63459</a> <a href="http://www.swiftmq.com/products/releases/notes/v900/index.html">http://www.swiftmq.com/products/releases/notes/v900/index.html</a> <a href="http://www.swiftmq.com/products/router/swiftlets/sys_amqp/features/index.html">http://www.swiftmq.com/products/router/swiftlets/sys_amqp/features/index.html</a>	AMQP 1.0 SwiftMQ 9.0.0	Jan 11, 2012
AMQP 0-9-1	Apache ActiveMQ	<a href="http://activemq.2283324.n4.nabble.com/jira-Closed-AMQ-5332-Support-AMQP-0-9-1-td4686862.html">http://activemq.2283324.n4.nabble.com/jira-Closed-AMQ-5332-Support-AMQP-0-9-1-td4686862.html</a>	No AMQP 0-9-1 in ActiveMQ	Nov 1, 2014
AMQP 1.0	Apache ActiveMQ	<a href="http://activemq.apache.org/activemq-580-release.html">http://activemq.apache.org/activemq-580-release.html</a>	AMQP 1.0 in ActiveMQ 5.8	Feb 12, 2013
AMQP 0-9-1	Apache Qpid	<a href="http://svn.apache.org/viewvc/qpid/trunk/qpid/specs/amqp0-9-1.stripped.xml?view=log&amp;pathrev=829944">http://svn.apache.org/viewvc/qpid/trunk/qpid/specs/amqp0-9-1.stripped.xml?view=log&amp;pathrev=829944</a>	AMQP 0-9-1 in Qpid 0.6	Oct 26, 2009
AMQP 1.0	Apache Qpid	<a href="https://issues.apache.org/jira/browse/QPID-4368">https://issues.apache.org/jira/browse/QPID-4368</a>	AMQP 1.0 in Qpid 0.20	Nov 28, 2012
AMQP 0-9-1	Red Hat Enterprise MRG	<a href="https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/1.3/pdf/Programming_in_Apache_Qpid/Red_Hat_Enterprise_MRG-1.3-Programming_in_Apache_Qpid-en-US.pdf">https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/1.3/pdf/Programming_in_Apache_Qpid/Red_Hat_Enterprise_MRG-1.3-Programming_in_Apache_Qpid-en-US.pdf</a> <a href="https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/2/html/single/Messaging_Programming_Reference/index.html#AMQP_0_10">https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/2/html/single/Messaging_Programming_Reference/index.html#AMQP_0_10</a> <a href="https://rhn.redhat.com/errata/RHSA-2014-0441.html">https://rhn.redhat.com/errata/RHSA-2014-0441.html</a>	No AMQP 0-9-1 in Enterprise MRG 1.3, 2.0 - 2.3, 2.5.  Unknown for Enterprise MRG 1.2, 1.1 and 1.0 which are no longer supported	

Protocol	Category	Reference	Comment	Date
AMQP 1.0	Red Hat Enterprise MRG	<a href="https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/3/html/3.0_MRG_Messaging_Release_Notes/chap-Introducing_Messaging_3.html#Red_Hat_Enterprise_Messaging_3.0">https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_MRG/3/html/3.0_MRG_Messaging_Release_Notes/chap-Introducing_Messaging_3.html#Red_Hat_Enterprise_Messaging_3.0</a>  <a href="https://access.redhat.com/support/policy/updates/mrg/">https://access.redhat.com/support/policy/updates/mrg/</a>	AMQP 1.0 in Enterprise MRG 3.0 limited (without transactions)	Sep 24, 2014
AMQP 1.0	Red Hat A-MQ	<a href="http://www.redhat.com/en/about/press-releases/red-hat-takes-guesswork-out-cloud-hybrid-and-internet-things-integration">http://www.redhat.com/en/about/press-releases/red-hat-takes-guesswork-out-cloud-hybrid-and-internet-things-integration</a>  <a href="https://www.jboss.org/products/amq/overview/">https://www.jboss.org/products/amq/overview/</a>  <a href="http://www.jboss.org/products/amq/download/">http://www.jboss.org/products/amq/download/</a>  <a href="https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_A-MQ/6.1/html/Release_Notes/FMQReleaseNotesNew.html">https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_A-MQ/6.1/html/Release_Notes/FMQReleaseNotesNew.html</a>	AMQP 1.0 in A-MQ 6.1	Apr 14, 2014
AMQP 1.0	Red Hat Fuse	<a href="http://www.redhat.com/en/about/press-releases/red-hat-takes-guesswork-out-cloud-hybrid-and-internet-things-integration">http://www.redhat.com/en/about/press-releases/red-hat-takes-guesswork-out-cloud-hybrid-and-internet-things-integration</a>  <a href="https://www.jboss.org/products/fuse/overview/">https://www.jboss.org/products/fuse/overview/</a>	AMQP 1.0 in Fuse 6.1	Apr 14, 2014
AMQP 1.0	Wireshark	<a href="https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=9612">https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=9612</a>  <a href="https://code.wireshark.org/review/gitweb?p=wireshark.git;a=history:f=epan/dissectors/packet-amqp.c;hb=8def8ef9c40189472a46d9b1ad95289780e09af5">https://code.wireshark.org/review/gitweb?p=wireshark.git;a=history:f=epan/dissectors/packet-amqp.c;hb=8def8ef9c40189472a46d9b1ad95289780e09af5</a>  <a href="https://code.wireshark.org/review/gitweb?p=wireshark.git;a=blob:f=epan/dissectors/packet-amqp.c;hb=8def8ef9c40189472a46d9b1ad95289780e09af5">https://code.wireshark.org/review/gitweb?p=wireshark.git;a=blob:f=epan/dissectors/packet-amqp.c;hb=8def8ef9c40189472a46d9b1ad95289780e09af5</a>	AMQP 1.0 in WireShark	Jan 20, 2014
AMQP 0-9-1	Wireshark	<a href="http://comments.gmane.org/gmane.comp.networking.rabbitmq.general/38319">http://comments.gmane.org/gmane.comp.networking.rabbitmq.general/38319</a>	AMQP 0-9-1 in WireShark development release	Feb 5, 2015
<b>Wiki</b>				
AMQP 0-9-1 & 1.0		<a href="http://en.wikipedia.org/wiki/Advanced_Messaging_Queueing_Protocol">http://en.wikipedia.org/wiki/Advanced_Messaging_Queueing_Protocol</a>	Overview	

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

253 of 284

Protocol	Category	Reference	Comment	Date
<b>Discussion</b>				
AMQP 0-9-1 & 1.0		<a href="http://kellabyte.com/2012/10/20/clarifying-amqp/">http://kellabyte.com/2012/10/20/clarifying-amqp/</a>	AMQP 0-9-1 vs. 1.0 comparison / clarification	Oct. 20, 2012
AMQP 0-9-1 & 1.0		<a href="http://www.rabbitmq.com/resources/AMQP-F2FPresentation.pdf">http://www.rabbitmq.com/resources/AMQP-F2FPresentation.pdf</a>	This is a neutral document and provides a lot of insight on the targets set of AMQP 1.0. Most of its contents is still applicable.	

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

254 of 284

## Appendix G REST

### G.1 The problem/need

The REST-style has become increasingly popular in the last years. The REST-style is being applied by some ATM Stakeholders. The REST-style is already supported to some extent by one existing binding but too limited to cover the mainstream understanding of the REST-style.

### G.2 Considerations

#### G.2.1 Standard

There is no such thing as a REST standard: it is a way of using protocols, in particular the HTTP protocol. The REST architectural style has first been described by Roy Fielding in [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm).

Various notions are used such as REST-style, RESTful, REST, REST-compliant or REST API. There is no formal ontology and/or nomenclature.

#### G.2.2 REST-style accessibility

Amongst some part of the developer community the REST-style is considered to be very accessible and hugely simpler than SOAP.

SOAP has had its growing pains: at the inception the S stood for Simple but that has been abandoned. Still today to be able to use the SOAP standards in an interoperable way, other standards must be applied onto the SOAP standards, the WS-I Profiles. As long as everyone sticks to that, it is mostly ok. If not, it becomes difficult to very difficult to troubleshoot, in particular if a development framework and/or runtime framework deals with parts or all of the SOAP handling and compositions of WS-\*.

#### G.2.3 REST style versus SOAP

##### G.2.3.1. The more difficult things

The ease to perform interactions using REST-style is because what was/is (more or less) defined to be the REST-style does not provide any consensus, guidance and/or standardization for the more difficult things like atomicity, reliability, message level security, etc.

SOAP based Web Services contain standardized elements for these more difficult things such as WS-AtomicTransaction, WS-ReliableMessaging, WS-SecureConversation, WS-Security, WS-Policy, WS-Trust and WS-Federation.

None of such functionality is standardized for a REST style. This means:

- either such functionality is missing and then the use of REST is only suitable for simple contexts,
- or such functionality is effectively provided via the REST style too but the learning curve is steep and the level of reuse is very low because each and every provider can provide such

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

255 of 284

functionality in another manner. In this case the use of the REST style can become much more expensive and much more complex than the use of the SOAP-based Web Services.

To be noted that according to some interpretations, functionality such as the one covered by WS-AtomicTransaction, WS-ReliableMessaging and WS-SecureConversation is considered to be stateful and thus goes against the principles of REST which requires statelessness. However examples can be found of emulation of the SOAP based WS-AtomicTransaction in the REST area by using a particular REST resource as a transaction coordinator.

## G.2.3.2. Service description

WSDL:

The services exposed via SOAP can be described via WSDL. Versions WSDL 1.1 and WSDL 2.0 are current and valid standards. The description language not only describes the mapping between the abstract service model and the concrete technology but also describes the security controls.

WSDL 2.0 can be used to describe REST-style services to a significant extent (<http://www.ibm.com/developerworks/webservices/library/ws-restwsdl/>). There are some limitations however. Some examples can be found under the heading WSDL 2.0 in <http://bitworking.org/news/125/REST-and-WS>.

WADL:

An equivalent to some extent of WSDL for the REST-style is called WADL. Sun Microsystems has made an attempt to have WADL standardized by W3C (<http://www.w3.org/Submission/wadl/>) but to no avail so far and currently without any perspective of becoming a standard (see <http://www.w3.org/Submission/2009/03/Comment:> "As of today, W3C has no plans to take up work based on this Submission").

It is particularly noteworthy that WADL does not have a capability to describe any of the security controls.

WADL is like WSDL, also XML based but allows to reference a JSON grammar.

RSDL, Swagger:

RSDL and Swagger are examples of other initiatives of a description language for services exposed in a RESTful manner. Their level of uptake is not clear (e.g. WADL is natively supported by soapUI, Swagger is only supported in soapUI through a plug-in written by a third party)

Interesting links:

comparison of API-Blueprint, RAML and Swagger at [http://www.slideshare.net/SmartBear\\_Software/api-strat-2014metadataformatsshort](http://www.slideshare.net/SmartBear_Software/api-strat-2014metadataformatsshort)

There is a significant amount of debate in the matter of how to describe REST-style services.

That debate is not only limited to which formal description language to use, but even includes the question whether a formal description language is needed for a REST-style service and whether it does not go against the principles of the REST-style.

A number of key elements of a contract that are formalised through a WSDL in a SOAP context, are also present in a REST-style only in a different manner. Examples:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

256 of 284



- the HTTP methods are equivalent to the operations and do not need a separate static declaration because they are fixed and known.
- the message format is determined by the MIME-type and can be discovered dynamically.

### G.2.3.3. Service discovery

SOAP is linked with the notion of a registry and the UDDI protocol.

There is not direct equivalence for UDDI related to the REST-style. There are some initiatives to try to squeeze WADL into a registry via a UDDI interface and/or extend the UDDI data model but there are no clear indications of any uptake (<http://apachejuddi.blogspot.be/2014/01/uddi-as-registry-for-rest-services.html>).

### G.2.3.4. Performance

A significant argument in the comparison between a REST-style and a SOAP-based style, is the scalability/performance advantage of the REST-style. This argument is mainly related to caching abilities.

- The use of the HTTP POST method by SOAP does not allow use of caching.
- The use of the HTTP GET method in the REST-style, allows for sophisticated cache patterns: locally, on intermediaries and/or at the side of the service provider. In the latter case, caching can still be useful when using a conditional get and the return a HTTP 304 Not Modified reply instead of the effective transfer of a representation of a resource.

Nevertheless the SOAP 1,2 HTTP binding as well as WSDL 1.1 allow use of the HTTP GET method to profit from caching opportunities.

## G.2.4 Security

### G.2.4.1. SOAP based Web Services

The security configuration can be formalised very precisely through WSDL based on WS-Policy. Knowledgeable code generators can automatically generate provider code and configuration as well as consumer code and configuration from such WSDLs to such extent that no more action is needed by either the service provider or the service consumer.

The portfolio of defined security options for SOAP based Web Services is wide enough to cover most cases (both at transport level as on message level) in a highly interoperable manner.

### G.2.4.2. REST-style

WSDL 2.0 + WS-Policy, allow to precisely describe the transport level security (implemented for instance via SSL/TLS or HTTP Basic Authentication), for the REST-style. Mainstream code generators (e.g. in the Microsoft and Java worlds) can deal with such description. The WADL format does not allow for such transport level security description. Proprietary extensions to WADL have been noted that do allow for such transport level security description.

There is no known method to precisely describe the message level security for the REST style which is understood by mainstream code generators.

### Network level security

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

A simple way to delegate part of the difficulty related to security controls in a REST-style, is to use a VPN at network level such as based on IPsec. Although this can address some security needs such as Integrity and confidentiality, and partially authenticity, the controls are weak because they do not protect the traffic in any way above the IP layer.

### Transport level security

Another way to delegate part of the difficulty related to security controls in a REST-style, is to use security controls at transport level. These controls are typically based on SSL/TLS and/or HTTP AUTH (Basic or Digest).

The strength of the security provided by SSL/TLS is significantly better than network level security (not protected above the transport level) as well as the scope (also mutual authenticity and non-repudiation).

The use of HTTP AUTH on top of Network level security, does not enhance significantly the strength but brings the access control closer to the application.

### Message level security

#### XML format

If the payload is represented through an XML format, the following standards could be used:

- XML Digital Signature
- XML Encryption

These standards are already used in some bindings. They could simply be added to the binding that is targeted to support the REST-style.

#### JSON format

If the payload is represented through a JSON format, some standardization at IETF is underway but not completed for functionality equivalent to XML Digital Signature and XML Encryption.

- JWS: JSON Web Signature (<http://tools.ietf.org/html/draft-jones-json-web-signature-04>)
- JWE: JSON Web Encryption (<https://tools.ietf.org/html/draft-ietf-jose-json-web-encryption-34>)

JSON encoding is not explicitly allowed in the YP but it is not explicitly disallowed either. The draft standards JWS and JWE are not yet in the Yellow Profile either. They could simply be added to the binding that is targeted to support the REST style. In such case, it would be necessary to explicitly add a reference to JSON encoding too.

Moreover, a framework has been defined (JOSE, see <http://jose.readthedocs.org/en/latest>) that includes above but also various other elements such as the encoding of Tokens (JWT) and Keys (JWK). Reference is made to a standardized register of algorithms (JWA).

The main issue with the JSON elements above, is that all of this is currently being standardized at IETF and none of above has reached the level of standard at this time.

### Identity Provision

OAuth 2 and OpenId are strongly linked with JOSE. SAML 2.0 can be used with the REST-style.

Some protocols are user centric and less oriented/suitable for machine to machine communication.

Also as SAML 2.0 relies on sessions and cookies, these are considered stateful and thus not conforming the REST-style

## Appendix H Baseline and Step1 Enablers Allocation

SWIM-TI TAD is being produced in a manner that the latest version deprecates the previous ones<sup>96</sup>. However, the continuous Change Requests and Update Campaigns to the Datasets aims at modifying not only the Enablers (and OIs) allocated to the latest Step, but also to the previous ones. In order to not lose track of such mentioned “deprecated” but “updated” Enablers, this chapter aims at analyzing Baseline and Step 1 Enablers included in Dataset 11.

### H.1 Allocation of Baseline and Step 1 ENs to functional blocks, projects and assessment

The current analysis is based in the latest Data Set available in the SESAR Programme (<https://www.atmmasterplan.eu/enablers>), namely version “v003.11 - Dataset 11”.

An enabler related to the definition of data or service model performed by WP8 should be allocated to system projects outside the WP14. In that case it is marked ‘Not applicable to the technical infrastructure’.

Some of these enablers were written during SESAR Definition Phase and hasn’t evolved since then; however, due to the evolution of the Programme, some of them need to be updated/rewritten. P14.01.03 provides the assessment below to be taken into account by B4.3, as responsible for the overall Dataset.

---

<sup>96</sup> For Iteration 3.1 and precedents, is the case. This doesn’t mean that in further iterations this approach won’t be modified.

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	AAMS-06b	ASM support systems enhanced to exchange static data and airspace usage data with NM systems in AIXM format	Not Applicable to the technical Infrastructure	Not in WP14	Not Applicable to the technical Infrastructure
1	AIMS-14	Set up a digital data chain to ensure the Aeronautical Information data provision into on-board avionic systems	Not Applicable to the technical Infrastructure	Not in WP14	The Enabler needs to be more concrete. Is not clear if talks about Services, Data, Technical, Operational or Infrastructure... or all of them.
1	METEO-01	Enhanced ATM decision making facing adverse MET conditions at the Airport.	Not Applicable to the technical Infrastructure	Not in WP14	When referring to "The provision will be done in a SWIM compliant manner" needs to be specified if the Data will be in the AIRM or if it means that the transport will be done via SWIM. In any case, the enabler is not SWIM-TI.

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	METEO-03	Provision and monitoring of accurate real-time weather information at the Airport.	Not Applicable to the technical Infrastructure	Not in WP14	When referring to "The provision will be done in a SWIM compliant manner" needs to be specified if the Data will be in the AIRM or if it means that the transport will be done via SWIM. In any case, the enabler is not SWIM-TI.  OI IS-0901-C? Not clear
1	METEO-04b	Provision and use of MET information services relevant for Airport and TMA related operations, Step 1	Not Applicable to the technical Infrastructure	Not in WP14	When referring to "The provision will be done in a SWIM compliant manner" needs to be specified if the Data will be in the AIRM or if it means that the transport will be done via SWIM. In any case, the enabler is not SWIM-TI.

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	METEO-05b	Generate and provide MET information relevant for En-route / Approach related operations, Step 1	Not Applicable to the technical Infrastructure	Not in WP14	When referring to "The provision will be done in a SWIM compliant manner" needs to be specified if the Data will be in the AIRM or if it means that the transport will be done via SWIM. In any case, the enabler is not SWIM-TI.
1	METEO-06b	Generate and provide MET information relevant for Network related operations, Step 1	Not Applicable to the technical Infrastructure	Not in WP14	When referring to "The provision will be done in a SWIM compliant manner" needs to be specified if the Data will be in the AIRM or if it means that the transport will be done via SWIM. In any case, the enabler is not SWIM-TI.
1	MIL-0502	Support MIL-0501 with ground-ground COM interface for interconnection of military systems to PENS	Not Applicable to the technical Infrastructure	Not in WP14	Don't know if this is an Enabler at all....
1	SWIM-APS-01a	Provision of Aeronautical Information services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Not Applicable to the technical Infrastructure

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	SWIM-APS-02a	Consumption of Aeronautical Information services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Not Applicable to the technical Infrastructure
1	SWIM-APS-03a	Provision of ATFCM Information Services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Not Applicable to the technical Infrastructure
1	SWIM-APS-04a	Consumption of ATFCM Information Services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Not Applicable to the technical Infrastructure
1	SWIM-APS-05a	Provision and Consumption of Flight Object Sharing services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Not Applicable to the technical Infrastructure
1	SWIM-APS-06a	Provision of Airport Ground Sensor Meteorological Information Services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Needs to be clarified how this "Being part of SWIM environment means", because it might mean that this EN needs to be traced towards a Profile one.
1	SWIM-APS-07a	Stakeholder systems consumption of Meteorological Information services for Step 1	Not Applicable to the technical Infrastructure	Not in WP14	Needs to be clarified how this "Being part of SWIM environment means", because it might mean that this EN needs to be traced towards a Profile one.

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	SWIM-GOV-05a	SWIM Framework	Not Applicable to the technical Infrastructure	Not in WP14	WP08? Is not clear who will take the role of leading the implementation of this EN.
1	SWIM-GOV-10a	Stakeholder Institutional arrangements for the linkage of the SWIM services supporting networks	Not Applicable to the technical Infrastructure	Not in WP14	SWIM Ens need to be traced towards this one. This EN enables SWIM-TI ENs.
1	SWIM-INFR-01a	High Criticality SWIM Services infrastructure Support and Connectivity.	Messaging Security Policy Enforcement Supervision Shared Object Recording	P14.02.09	This EN maps towards Blue Profile  There is a need for an EN mapping towards Purple Profile!
1	SWIM-INFR-05a	General SWIM Services infrastructure Support and Connectivity.	Messaging Security Policy Enforcement Supervision Recording	P14.02.09	This EN maps towards Yellow Profile.  There is a need for an EN mapping towards Purple Profile!
1	SWIM-NET-01a	SWIM Network Point of Presence	Not Applicable to the technical Infrastructure (Network Infrastructure)	Not in WP14	SWIM Ens need to be traced towards this one. This EN enables SWIM-TI ENs.



STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	SWIM-STD-01a	ATM Information Reference Model for Step 1	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-02a	Information Services Reference Model for step 1	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-03a	Use of Standard for SWIM - Recording Services and QoS	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-04a	Use of Standard for SWIM - governance standard for service interfaces.	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-05a	Use of Standard for SWIM Flight Information Content in Services/Exchanges	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-06a	Use of Standard for SWIM Traffic Flow Services Content	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
1	SWIM-STD-07a	Use of Standard for SWIM Airspace Management Services Content	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-08a	Use of Standard for SWIM Aeronautical Information Services Content	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-STD-09a	Use of Standard for SWIM Metrological Information Services Content	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.
1	SWIM-SUPT-01a	SWIM Supporting Registry Provisions	Registry	P14.02.09	
1	SWIM-SUPT-03a	SWIM Supporting Security Provisions	Security	P14.02.09	
1	SWIM-SUPT-05a	SWIM Supporting IP Network Bridging Provisions	Security	P14.02.09	
Baseline	GGSWIM-11	ATM Information based on a common Aeronautical Information Exchange Model (AIXM)	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.

STEP	CODE	Enabler Title	Functional block identifier	SYS Primary Project	ENs Assessment
Baseline	GGSWIM-26a	Provision and use of Ground-ground data services for Network Operations Planning	Messaging Security Policy Enforcement Supervision Shared Object Recording	P14.02.09	Profiles
Baseline	GGSWIM-49	Ground-ground data communications services for airspace reservation/availability	Messaging Security Policy Enforcement Supervision Shared Object Recording	P14.02.09	Profiles
Baseline	GGSWIM-52	Provision and use of ground-ground data communications services for aeronautical information- EAD	Messaging Security Policy Enforcement Supervision Shared Object Recording	P14.02.09	Profiles
Baseline	GGSWIM-53	A common Aeronautical Information Conceptual Model (AICM)	Not Applicable to the technical Infrastructure (Data + Services)	Not in WP14	Needs to be traced towards SWIM ENs.WP08.

**Table 41 – Allocation of Baseline and Step 1 ENs to functional blocks, SYS Primary Projects and Assessment**

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

## Appendix I FO Overlay Network Candidate Architectures

The following section details some candidate architectures for the implementation of the FO Overlay Network.

### I.1 Proposal 1: The Dillon<sup>97</sup> Model

This proposal, aka the Dillon's model, with no gateway and more efficient as it is entirely based on underlying network capabilities. This approach requires support from COTS providers to take into account emerging multicast technologies as existing middleware applicable standards/products assume existence of 'multicast' with PIM-ASM on the WAN (PENS).

The alternative model exploits IOP Manager/Publisher role for Flight Objects (FOs) defined by ED-133 and can be further generalised as an extension for DDS interoperability specification (DDSI).

The proposed solution makes use of both ASM and SSM multicasts, where ASM is used for the summary information to be shared within the whole IOP area; and SSM for the publication of the actual (flight) objects. It fully exploits network capabilities though requires IGMP V3 to support SSM; and can be implemented on both IPV4 and IPV6.

The solution defines a control plane for setting up multicast routes and a data Plane for data delivery to only nodes in distribution list.

#### I.1.1 Control Plane

FO summaries requires global sharing within the whole IOP Area, so will be distributed based on ASM multicast. Each Flight Data Manager/Publisher (FDMP) will publish FO summaries for the flight objects it is managing in a many-to-many manner.

A single ASM group for the whole IOP Area ( $*,G$ ) will need to be defined for all the FO Summaries.

In addition to the distribution list (DL) included in each FO summary, the FDMP shall include an additional information ( $source,G$ ) where FO clusters (Flight Object data) will be published. The *source* refers to the FDMP IP address and *G* to a group multicast address within the SSM range unique to the FO within the context of the source (FDMP).

A typical usage scenario is when an IOP participant detects from the FO summary that is part of the distribution list, it shall join the ( $source,G$ ) group included in the summary in order to receive flight object data publications. Once a participant detects it is no longer part of the distribution list it shall leave the ( $source,G$ ) group. Group membership operations are defined in IGMP V3 protocol.

ED-133 specification defines *SP-IOP-Max\_FO\_Managed* managed flight objects per FDMP, so a range of *SP-IOP-Max\_FO\_Managed* addresses are to be allocated to each FDMP for publication of flight object data in a one-to-many SSM multicast. SSM multicast only requires that *G* is unique within the context of the *source*; allowing the sharing of the *SP-IOP-Max\_FO\_Managed* group addresses by all IOP Area participants. This will simplify configuration of FDMPs.

Current value of *SP-IOP-Max\_FO\_Managed* is 3800 which makes the range of multicast addresses to be allocated for FO distribution reasonable and with no impact on routing tables within the network routers.

<sup>97</sup> First proposed by Patrick Dillon, Network Expert (patrick.dillon@thalesgroup.com)

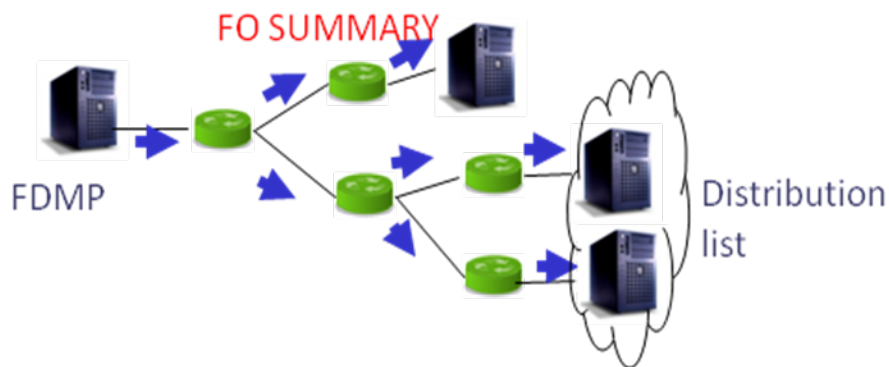


Figure 98 – FO SUMMARY distribution

## I.1.2 Data Plane

Once multicast routes established by the control plane, Flight Object data (FO Clusters) will be published in a one-to-many multicast from the FDMP to all members of the distribution list.

The dynamic nature of the distribution list does not affect network configuration and the delivery of FO data through SSM multicast is very efficient network-wise (only nodes in the distribution list will receive the network packets).

## I.1.3 Advantages

The above solution provides the following advantages:

- No need for a Gateway or Application/SWIM-level DDS router.
- Applies to all PENS network, no need for Areas
- Efficient, only nodes in distribution list receive traffic
- Easy-configuration as the range of multicast groups can be shared by all nodes

## I.1.4 Disadvantages

The main disadvantage of this solution is related to the current version DDS interoperability (DDSI) standard that does not support, in an interoperable manner, SSM multicast. Proposals shall be made to the major DDS vendors by the industrial partners to enhance the current standard and/or define portable DDS APIs for multicast group membership.

## I.1.5 Follow-up

The following activities are foreseen to implement an efficient delivery of flight objects:

- Involve DDS vendors for an easier and portable way of configuring network partitions.
- Study how this configuration fits with NAT (IPV4) between ASNPs addressing plans and PENS addressing plan.
- Assess further generalisation for DDSI protocol itself and make the (*source,G*) information part of the DDS endpoints avoiding the addition of this addressing information to the FO summaries.

## I.1.6 DDS QoS

### I.1.6.1. DURABILITY

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

269 of 284

As only TRANSIENT\_LOCAL and VOLATILE QoS values are allowed by the DDS interoperability specification, no other value is to be used for the DURABILITY QoS. Use TRANSIENT\_LOCAL at DataWriter, and VOLATILE to DataReader so backup replica use locally available state for recovery.

Use of VOLATILE on the DataReader will make sure the DDS COTS products in use will not request from DataWriters submission of previous publications through the network. The DataReader being a late-joiner with VOLATILE value for DURABILITY QoS will only get newest publications.

## I.1.6.2. PARTITION

Multicast communication within a DDS domain ensures efficient use of network resources. It is important for the network infrastructure to support IGMP v3 in the case of IPv4 and MLD v2 for IPv6 for efficient multicasting over a WAN.

Assigning multicast groups per distribution list within a DDS domain is efficient at the network level as this avoids delivering Flight Object data to nodes not in the distribution list; but does not take into account dynamic nature of distribution lists.

The current use of PARTITION QoS does not work; so the PARTITION QoS shall not be used unless it is possible to use 'Network' partitioning to map 'logical' partitions to physical network domains (groups).

## I.1.6.3. TRANSPORT\_PRIORITY

The OMG DDS specification defines a policy called TRANSPORT\_PRIORITY to be used as a hint to the infrastructure as to how to set the priority of the underlying transport used to send the data.

The SWIM technical infrastructure shall make use of TRANSPORT\_PRIORITY to correctly support the 3 defined categories (d\_1, d\_2, d\_3) for Flight Object Publications.

## I.1.6.4. Support for Work sets

The OMG DDS provides a global data space where all participants have a view of all available data. Exchanging all available data between all DDS entities is not desirable given the number of Flight Objects that may exist in the European ATM at one time. There is a need to implement some concept of a **Work set**, or **Data Instance partitioning** where only some data instances (Flight Objects) in a given topic that are of interest to a stakeholder shall be sent to the stakeholder.

Additionally, Flight Object clusters shall only be conveyed to locations hosting members of the distribution list.

## I.1.6.5. Local Recovery

When starting additional server replicas locally, always use locally available Flight Objects and only recover existing Flight Objects via Wide Area Network when strictly necessary.

Recovery via SWIM (WAN) is only to be performed on explicit requests by applications.

## I.2 Proposal 2

This proposal is an option on the Dillon Model, where a DDS Partition is used per Distribution List. This avoids the problem of publishing the same data sample per partition as done by several DDS vendors.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

This is only efficient if filtering is performed at source-level by mapping each DDS partition to a 'Network' partition<sup>98</sup>.

$$partition = f(DL)$$

The function  $f()$  for use to compute the partition name has then to be standardized within the IOP Area since the DDS partition name is to be inferred locally based only on the distribution list (DL).

An example of such a function can be the following:

$$f : hash ( sort ( unique (DL) ) );$$

First  $unique()$  function removes all duplicates within the DL, then  $sort()$  sorts the resulting list alphabetically. Finally  $hash()$  function generates a unique hash for the sorted list (ex: SHA-1, simple concatenate function ...) to reduce the size of partition.

Once a proper mapping has been established so DDS partitions are allocated to multicast groups, the SWIM TI shall be responsible for creating/deleting the subscriber with the DDS partitions upon appearance/disappearance of the local stakeholder in the distribution list.

DDS COTS are required to support allocating multicast groups to DDS partitions and perform any required join/leave operations compliant to source-specific multicast, i.e. join/leave (S,G) where S is the DataWriter's IP address and G the multicast group.

## I.3 Proposal 3

This proposal assumes no availability of ASM within the WAN; so no many-to-many communication is available.

This uses the same approach as proposal 2 for using DDS partitions; but requires specific SWIM Nodes dedicated for DDS Participant & Endpoint Discovery and for the publication of FO Summaries to those members not in the distribution list.

Members with the distribution list will receive FO Summaries from the FO publishers.

---

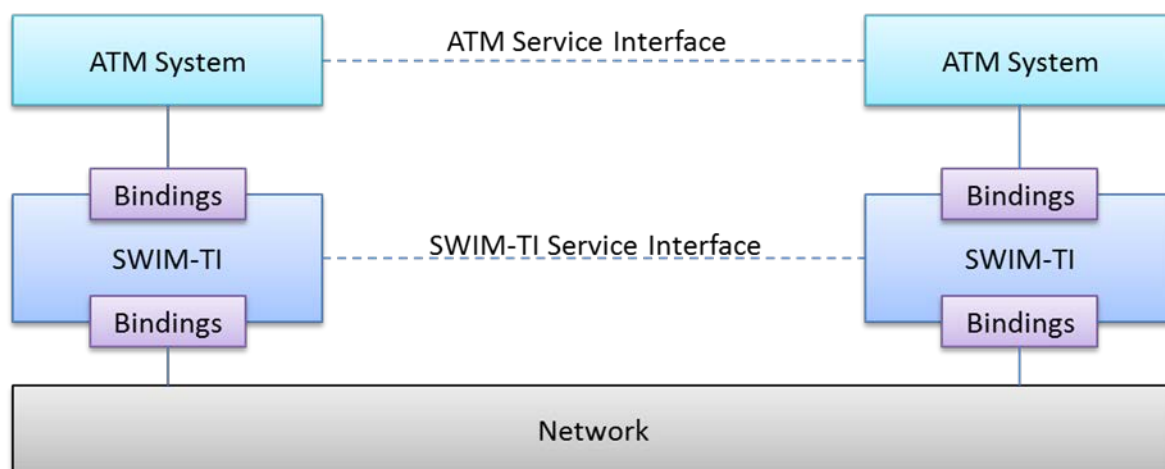
<sup>98</sup> OpenSplice DDS, for example, defines a network partition by one or more unicast, multicast or broadcast IP addresses. Users will then have to define a mapping between a network partition and a DDS partition-topic combination.

## Appendix J Interface Evolution

This section sets out different types of interface evolution that can potentially occur and analyses for each of them the possible impact in the current SWIM-TI and the possible mitigations and solutions that can be adopted for supporting the SWIM inherent heterogeneity.

### J.1 Scope

The following diagram provides an overview of the key components of SWIM of interest for the Interface Evolution:



**Figure 99 – Key components for Interface Evolution**

The ATM Service Interface defines the information exchanged between ATM Systems in terms of operations, messages and data types for each ATM Service. The SWIM-TI Service Interface does the same but for SWIM-TI internal purposes.

These two interfaces put a specific service definition on the wire using the concrete technologies and protocols defined on the different SWIM Profiles. Following characteristics are defined in each interface:

- Protocol Stack
- Message Exchange Pattern (MEP)
- Fault Handling
- Encoding
- Security
- Contract
- Interoperability (if applicable)



This section addresses the Interface Evolution of the ATM Service Technical Interface from the SWIM-TI perspective, analysing the impact on the SWIM-TI Bindings defined on the different SWIM Profiles.

## J.2 Objective

The main objective of this appendix is to provide the SWIM-TI Interface Evolution Strategy. To achieve this, this section defines a framework for analysing the impact of the different types of Interface Evolution that can occur at the ATM Service Interface layer in order to:

- Extract the necessary requirements for each SWIM Profile to implement the Interface Evolution Strategy Defined.
- Provide a set of recommendations from the SWIM-TI point of view for the ATM Service designers.

### Proposal from SOAP and DDS families of bindings interface evolution analysis:

Interface evolution activities focus on evolution of ATM Service STDD (Service Technical Design Description) identify how SWIM-TI may support different strategies by identifying technical enablers, rules and recommendations. To specify versioning and evolution strategies of ATM Service, implies to govern the evolution of the ATM service STDD. P14.01.03 and P14.01.04 can facilitate and "enable" evolution strategies by:

- defining recommendations/proposals and rules (based on J.4 Interface Evolution Techniques) and template (like the Table 55: Service Evolution Description Template) to be applied to properly manage evolution of ATM service STDD. Then, it will be an assumption that someone responsible for the evolution strategy of such ATM service STDD will use/refine/govern such rules. Some rules may be SWIM-TI Profiles Interface Bindings independent whereas other may be binding specific due to particular standards adopted in that binding. For instance rules on XSD modelling techniques to achieve minor version compatibility are only applicable to interface bindings using XML/XSD.
- offering to the system/application layer (both consumer and provider/producer sides) one or more techniques, as introduced and being detailed in J.4.2 Major Versioning Compatibility Techniques for major versions compatibility.

Furthermore, ATM service implementations versioning is not addressed. In particular for a given version of the STDD, a stakeholder may plan different versions of the service implementation. According to the "Contract first" (STDD) approach, changes on service implementations are not expected to impact technical interoperability (the STDD version is the same) if what specified in the STDD is properly used as reference by both provider and consumer.

The "object under evolution" is the STDD which is mainly composed by the following parts:

- a) "*Applicable Service Name and Versioning*", that includes naming, versioning, status and reference concerning the ATM Service (and its related SDD) to which the Service Technical Design applies to.
- b) "*Service Technical Interfaces*", that includes the description of the technical interfaces of the service. This is the part where the link with chosen SWIM-TI interface bindings is provided.
- c) "*Service Levels And Design Decisions*", that includes the description of the service levels and any relevant design decisions taken during technical design.

Interface evolution in scope of IT3.1-04 activity will focus on evolution of only STDD "Service Technical Interfaces" part. This part is composed by different elements and evolution of one or more of those elements may be reflected in compatible or incompatible changes. The key element is anyway the chosen SWIM-TI interface bindings.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

273 of 284

According to that, Table 55: Service Evolution Description Template could be used by ATM services STDD architects as template to document required information (versioning schema, naming rules, etc.) by reporting rules and recommendations identified in the context of activity IT3.1-04.

## J.3 Concepts

This chapter describes the key concepts that are going to be used for designing the Interface Evolution approach for the SWIM-TI.

### J.3.1 Service Technical Interface

A Service Technical Interface in the SWIM-TI is defined as a pair of a SWIM-TI Binding (or SWIM-TI Network Binding) and a specific instantiation of the contract(s) used. A change of any element of the Binding or a change in any of the artefacts defining the contract(s) (e.g. XML Schema, WSDL, IDL...) results in an evolution of the Interface.

### J.3.2 Compatibility

Whenever a change is made in an ATM Service Technical Interface, there will be an impact on every ATM System that has any form of dependency with that service. The measure of the impact is directly related to how compatible the updated ATM Service Technical Interface is with respect to the previous version.

There can be three types of compatibility with respect to an evolution:

- **No Compatibility:** The new version of the service is not able to continue to support the consumers of the previous version, forcing them to update their systems in order to be able to consume the updated service.
- **Backwards Compatibility:** The new version of the service continues to support the consumers of the previous version. This means that the update made to the service does not oblige the existing consumers to update their systems, as they can continue consuming the service with the previous version of the interface.
- **Forward Compatibility:** in this case, the design of the service interface is made in such a way that it can accommodate foreseen or unforeseen changes in the future without affecting the current or new consumers.

An ATM Service Interface is composed of a number of elements that are subject to change and evolution. Depending on which aspect evolves and in which way it evolves, the change can be categorized in the following categories:

- **Compatible Change:** When a change made on a service interface does not affect the existing consumers, the change can be categorized as compatible change. A compatible change is directly related to backwards compatibility, and could potentially be forward compatible but not necessarily.
- **Incompatible Change:** If after a change made on a service interface, the current consumers do not longer support the new version, the change can be categorized as an incompatible change. These types of changes are critical when specifying an Interface Evolution Strategy. These types of changes are directly related to the No Compatible changes described above.

## J.3.3 Versioning

The SWIM-TI recommends that ATM Service Interfaces use the “significance of change” versioning approach (if an alternative versioning scheme is used, it should be clearly defined). This versioning approach describes at least two levels of significance (major and minor) that are represented by two numbers separated by a dot in the following way “major.minor” (e.g. 1.3, 2.0, etc.), being the non-decimal part the major version of the interface and the decimal part the minor version.

This versioning approach can be extended to more levels of significance, by adding dot-number pairs (e.g. major.minor[.build[.revision]]). The semantic meaning of further levels of significance is not standardized in this document and is left up to the implementer to define as it suits the needs of their interface evolution.

- **Release of a Minor Version:** Minor releases usually reflect a Compatible Change that does not affect the current consumers of the service. The SWIM-TI recommends using minor versions strictly for compatible changes.
- **Release of a Major Version:** Major releases affect core service functionality and usually backwards compatibility cannot be guaranteed. This is usually a result of one or more Incompatible Changes included in the release.

## J.3.4 Strategies

There is no one-size-fits-all strategy for managing the interface evolution. The strategy definition is a governance-related phase in the overall service design lifecycle. This section introduces the three most commonly used Interface Evolution Strategies:

- **Strict Strategy:** Any compatible or incompatible change in the service will result in a new Major Version of the service interface. This approach does not support backward or forward compatibility.
  - Pros
    - Full control of the evolution of the service
    - No need to concern on the impact of change
    - Strict and static governance
  - Cons
    - Forces consumers to adapt to change as soon as it occurs may result in non-adaptable stakeholders to become isolated.
    - Heavy coupling between providers and consumers slows down evolution.
- **Flexible Strategy:** Any compatible change results in a new Minor Version of the service and any incompatible change results in a new Major Version. The service interface is designed to support backwards compatibility but not forwards compatibility has to be reached.
  - Pros
    - Change responsive
    - No consumer isolation in case of Minor Version change, as they can still use the previous version of the service.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

275 of 284

- Cons
  - The need of governance for regulating to which extent to provide backwards compatibility defining the types of changes are to be supported as compatible and which not (some changes can be compatible changes, but for governance reasons, they can be set as changes that force a new Major Version).
- **Loose Strategy:** Any incompatible change results in a new version of the service, and the service interface is designed to support both backwards and forwards compatibility.
  - Pros
    - Extreme flexibility
  - Cons
    - Vague and overly coarse service interfaces increments complexity
    - The need of extra governance effort

**Note:** Due to the vague definition of the services using the Loose Strategy, it will not be considered for any further analysis.

## J.3.5 Change Types

This document considers several areas of interest where interface changes could occur and should be considered:

- Data Model
- Messages
- Interface Operations
- Bindings

Changes in Data Model and in the Interface Operations could theoretically be versioned independently, but the preferred approach is to version at interface level, where any change in the Data Model and/or the Interface Operations result in a new Minor or Major Version of the service.

The following sections describe the most common changes that can occur within these two areas.

### J.3.5.1 Data Model Changes

For the Data Model evolution, the four types of changes are considered. These changes cover the simple or complex Data Elements that form the parameters and messages of the operations of an interface:

- Addition of a new Data Element
- Removal of an existing Data Element
- Renaming of a Data Element

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

276 of 284

- Modification of a Data Element

### J.3.5.2 Message Changes

The following changes are considered for the message evolution of an interface.

- Addition/removal of parameters/attributes in the input
- Addition/removal of parameters/attributes in the output
- Change order of parameters in the input/output
- Change cardinality of parameters in the input
- Change cardinality of parameters in the output

### J.3.5.3 Interface Operations Changes

For the Interface Operations evolution, the following types of changes are considered:

- Addition of a new Operation
- Removal of an existing Operation
- Renaming of an Operation
- Change of MEP
- Addition of a fault message to an Operation

### J.3.5.4 Binding Changes

The following changes affect the Binding definition.

- Changes of Protocols
- Changes to MEP
- Changes to Fault Handling
- Changes to Encoding
- Changes to Security

## J.4 Interface Evolution Techniques

This section introduces several techniques available that minimize the impact on consumer side of the evolution of an interface.

### J.4.1 Minor Versioning Compatibility Techniques

The following techniques are used for maintaining Backwards Compatibility between Minor Versions of a specific Interface. These techniques are used within a Flexible Strategy, to convert the types of changes described in section J.3.5 into Compatible Changes that result in a Minor Version update.

#### J.4.1.1 Optional Data Elements

By making a Data Element optional, it can be either added or omitted by the service provider depending on the version of the service. This is useful for both addition and removal of Data Elements.

For the addition of a new Data Element, this element is added as optional and the service provider will include it or request it depending on the consumer implementation.

In case of Removal of an existing Data Element, the behaviour is exactly the opposite of the case described above.

The main issue of this technique is that, in case of being used several times without forcing a Major Version update, the Data Model can become confusing and difficult to understand.

#### J.4.1.2 Selectable Data Elements

This technique allows the service providers to support a set of Data Elements to use from a list of supported and compatible Data Elements. This technique cannot be applied when the Data Elements to select from are of different types.

This technique is especially useful for the Renaming of a Data Element. In this case, the service provider can choose the new name Data Element for the consumers of the newest service minor version or the applicable one in case the consumer invokes an older version of the interface. This also applies when the selectable Data Element is part of the service request.

The main issue of this technique is that, in case of being used several times without forcing a Major Version update, the Data Model can become confusing and difficult to understand.

#### J.4.1.3 Flexible Data Types

This technique is used mainly for the Modification of a Data Element and basically refers to two possible types of modification:

- Type
- Cardinality

For Type modifications, the approach is to use Data Types with a wider scope that covers the possible modification. In case the Data Type change is not compatible, this technique could not be used.

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

278 of 284

For Cardinality modifications, the approach is to use the cardinality range that covers the needs of all the supported service versions.

#### J.4.1.4 Wildcards

The use of wildcards in the definition of the Data Model provides extreme flexibility to the data model. This allows the Data Model to grow (or decrease) in terms of Data Elements, making it possible to add and delete Data Elements without compromising the Data Model. This extreme flexibility comes at the cost of vague and imprecise Data Model that provides few or none information, so its usage is not recommended unless strictly necessary.

#### J.4.1.5 Minor Version Techniques to Change Types Relationship

The following table summarizes which techniques can be applied to Data Model and Message definition changes while ensuring backwards compatibility between minor versions:

	Technique	Optional Technique
<b>Addition of a new Data Element</b>	Optional Data Element	Wildcards
<b>Removal of an existing Data Element</b>	Optional Data Element	Wildcards
<b>Renaming of a Data Element</b>	Selectable Data Element	
<b>Modification of a Data Element</b>	Flexible Data Types	
<b>Addition of a new parameter/attribute in the input</b>	Optional Data Element	Wildcards
<b>Removal of a new parameter/attribute in the input</b>	Optional Data Element	Wildcards
<b>Renaming of a parameter/attribute in the input</b>	Selectable Data Element	
<b>Modification of a parameter/attribute in the input</b>	Flexible Data Types	

Table 42 – Techniques to assure backwards compatibility between minor versions

## J.4.2 Major Versioning Compatibility Techniques

The following section describes techniques that allow service providers to simultaneously maintain in operation multiple Major Versions of an Interface, the pros and cons of each option are reviewed.

### J.4.2.1 Static Binding

This is the simplest solution, but the less flexible one. The service consumers are aware of the location of the compatible service providers and adapt their systems for consuming the required services. Service providers simultaneously expose different endpoints for the different versions of the services they provide. If a Service Provider updates the service interface version, the bound Service Consumers will have to:

- Update their version implementation, or
- Find another Service Provider (can be the same provider in a different endpoint) that still offers the same service version and configure its implementation to access that endpoint

The picture below describes this:

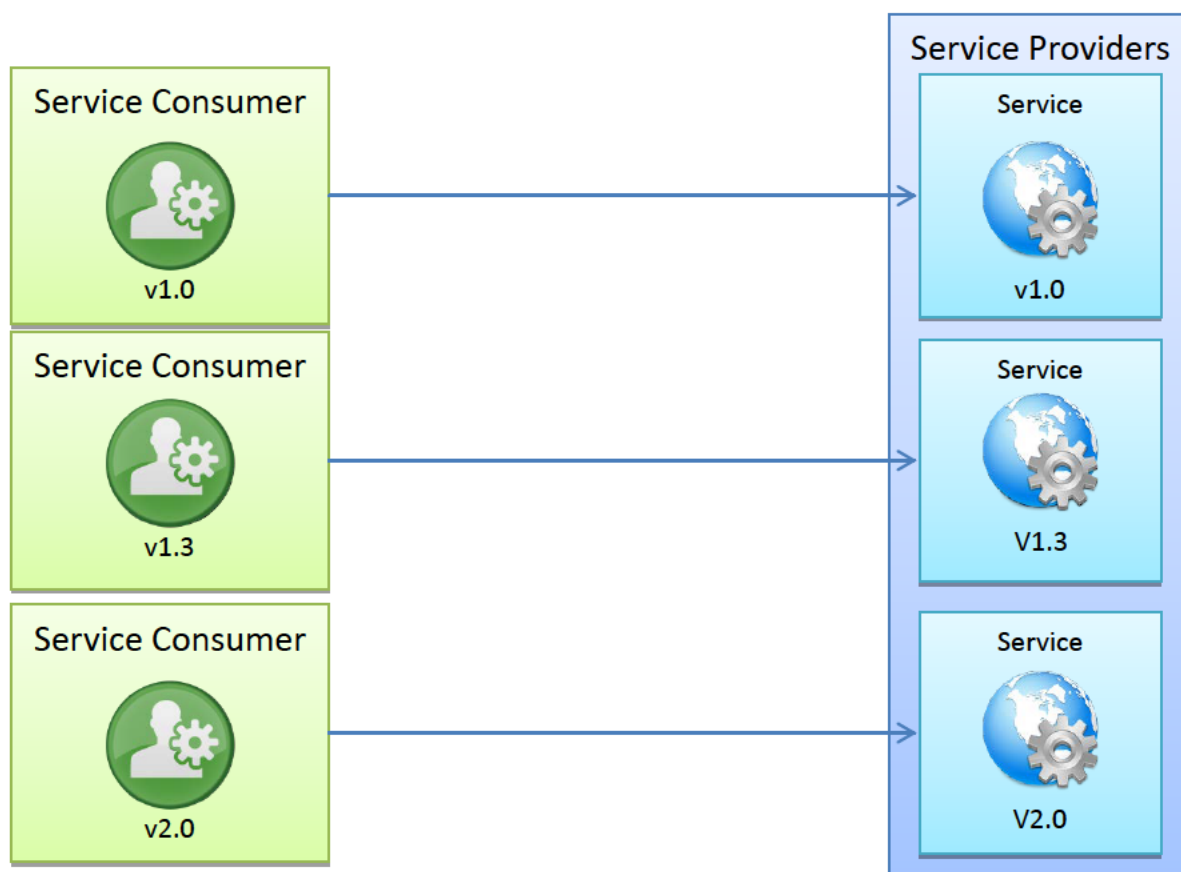


Figure 100 – Static Binding

This option has the following pros and cons:

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
www.sesarju.eu

280 of 284



- Pros:
  - Simple client implementation
  - Better performance, as there is no need of mediation and/or intermediaries
- Cons:
  - No Flexibility
  - Consumers need to be aware of the different service provider endpoints or discover them in certain way (e.g. Registry)
  - There is the need of defining a mechanism for informing consumers of interface service updates
  - Major Version change in a service imply a change in the consumer implementation, either for adapting to the new version or for reconfiguring its system to access other compatible service version

### J.4.2.2 Adapter

This technique consists in the introduction of an adapter that is able to retro-fit newer Major Versions of a service to the previous Major Versions. This technique is not mutually exclusive with the previous ones and can coexist with them. In case of a new Major Version of a Service, the Service Provider should develop the adapter to the previous version and could either maintain the previous endpoint (transparent to Service Consumer), or force Service Consumers to configure their implementations to a new endpoint. The following figure describes this technique:

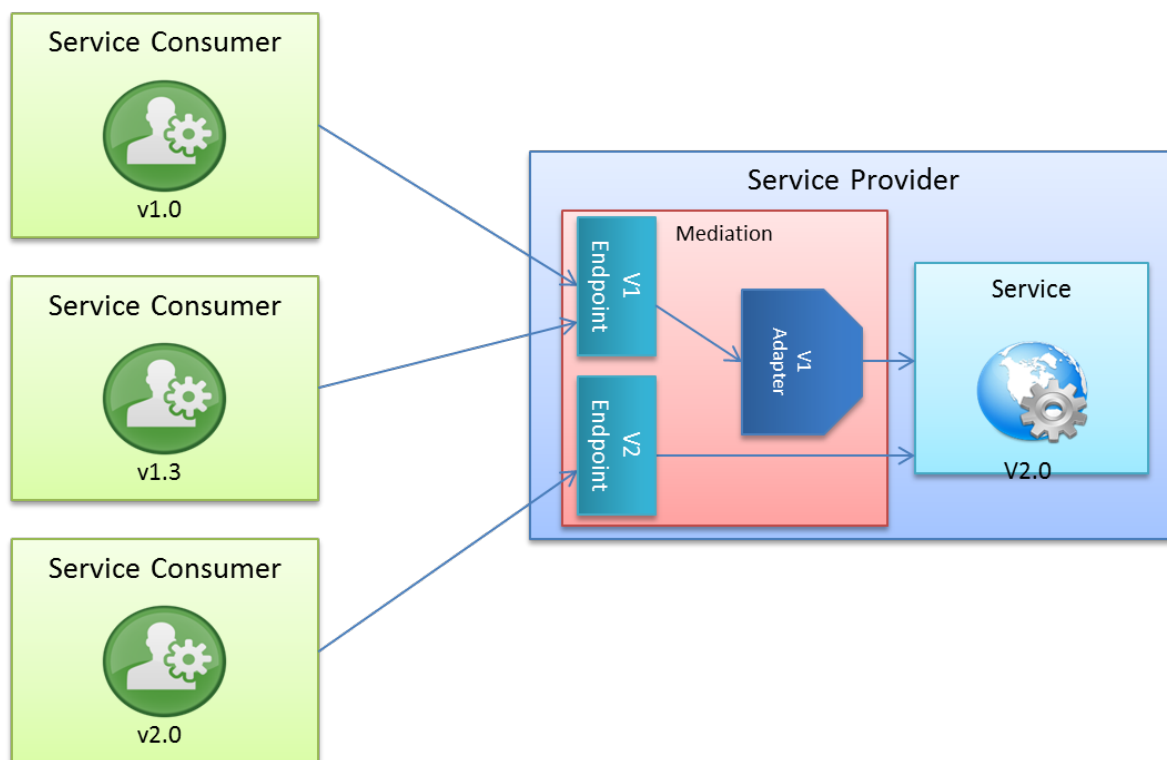


Figure 101 – Adapter

This technique has the following pros and cons:

- Pros:
  - Flexible solution for consumers
  - Responsive to Major and Minor Versions update (no need to update on consumers)
  - Allows Service Providers to have one single implementation of a service
- Cons:
  - Adapter could potentially be complex
  - Increased performance costs
  - Not always applicable

### J.4.2.3 Default Values

Defining default values on new Data Elements allows Service Providers to assign a value to a Data Element that is not provided by service requests of previous versions. Analysing during service design the possibility of assigning default values for new Data Elements facilitates dealing with backwards incompatible changes and enables the use of Adapter technique.

## J.5 SWIM-TI Interface Evolution Strategy Definition Needs

The following questions need to be answered for designing the SWIM-TI Interface Evolution strategy, taking in account the information presented above:

- There is the need to define what types of Interface Evolution are supported as Minor Version and which ones imply a Major Version update. This should be done on a Service basis.
- There is the need to define a retirement strategy for outdated Service versions so Service Consumers with less adapting capabilities can prepare in advance the update their implementation without hampering the natural evolution pace of the Interfaces
- There is the need to define how Service Consumers are informed of Interface Major and Minor updates independently of their need update/configure their systems or not because of the change
- There is the need to define how Service Consumers discover or become aware of the most suitable endpoint for their implementation

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

283 of 284

**-END OF DOCUMENT-**

founding members



Avenue de Cortenbergh 100 | B -1000 Bruxelles  
[www.sesarju.eu](http://www.sesarju.eu)

284 of 284