



Telecom SOA Use Cases and Issues

Version 1.0

Committee Draft 01 / Public Review 01

19 October 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/soa-tel/t-soa-uci/v1.0/cd01-pr01/t-soa-uc-pr-01.html>
<http://docs.oasis-open.org/soa-tel/t-soa-uci/v1.0/cd01-pr01/t-soa-uc-pr-01.pdf> (Authoritative)
<http://docs.oasis-open.org/soa-tel/t-soa-uci/v1.0/cd01-pr01/t-soa-uc-pr-01.doc>

Previous Version:

N/A

Latest Version:

<http://docs.oasis-open.org/soa-tel/t-soa-uci/v1.0/t-soa-uc.html>
<http://docs.oasis-open.org/soa-tel/t-soa-uci/v1.0/t-soa-uc.pdf> (Authoritative)
<http://docs.oasis-open.org/soa-tel/t-soa-uci/v1.0/t-soa-uc.doc>

Technical Committee:

OASIS SOA for Telecom (SOA-Tel) TC

Chair(s):

Mike Giordano, giordano@avaya.com

Editor(s):

Enrico Ronco, enrico.ronco@telecomitalia.it

Related work:

This specification replaces or supersedes:

- Not Applicable

This specification is related to:

- Not Applicable

Declared XML Namespace(s):

Not Applicable

Abstract:

This document is the first deliverable produced within the OASIS SOA for Telecom (SOA-TEL) TC and has the objective of collecting potential technical issues and gaps of SOA standards (specified by OASIS and other SDOs) utilized within the context of Telecoms.

All perceived technical issues on SOA standards contained in this document are structured with a description of the context, a use case, and a rationalization of the possible gap within the standard.

Amongst future deliverables of the SOA-TEL TC there is a Requirements specification, which will aim to extend the current core SOA enabling stack (Web Services and/or REST, etc.) in support of Telecom needs on the basis of the issues identified within the present document.

Status:

This document was last revised or approved by the OASIS SOA for Telecom (SOA-Tel) TC on the above date. The level of approval is also listed above. Check the “Latest Version” or “Latest Approved Version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <http://www.oasis-open.org/committees/soa-tel/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/soa-tel/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/soa-tel/>.

Notices

Copyright © OASIS® 2009. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", "SOA-TEL", are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction.....	7
1.1	Terminology.....	7
1.2	Normative References.....	8
1.3	Non-Normative References.....	8
2	Context setting.....	9
3	Issues on Addressing and Notification.....	12
3.1	Transaction Endpoints Specification.....	12
3.1.1	Scenario/context.....	12
3.1.2	Use Case.....	12
3.1.3	Perceived Technical Issue.....	14
3.2	WS-Notification.....	14
3.2.1	Scenario/context.....	14
3.2.2	Use Case (A).....	14
3.2.3	Perceived technical issue (A).....	15
3.2.4	Use Case (B).....	16
3.2.5	Perceived Technical issue (B).....	17
4	Issues on communications protocols.....	18
4.1	SOAP.....	18
4.1.1	Scenario/context.....	18
4.1.2	Use Case.....	18
4.1.3	Perceived Technical issue.....	22
5	Issues on Security.....	25
5.1	SAML Token Correlation.....	25
5.1.1	Scenario/context.....	25
5.1.2	Use Case.....	25
5.1.3	Perceived Technical issue.....	27
5.2	SAML Name Identifier Request.....	28
5.2.1	Scenario/context.....	28
5.2.2	Use Case.....	28
5.2.3	Perceived Technical issue.....	29
5.3	SAML Attribute Management Request.....	29
5.3.1	Scenario/context.....	29
5.3.2	Use Case.....	30
5.3.3	Perceived Technical issue.....	31
5.4	User ID Forwarding.....	32
5.4.1	Scenario/context.....	32
5.4.2	Use Cases.....	32
5.4.3	Perceived Technical issue.....	35
6	Issues on Management.....	37
6.1	Introduction.....	37
6.2	Scenario/context.....	37
6.3	Services exposing Management Interface.....	37
6.3.1	Perceived Technical Issues.....	39

6.4 Metadata in support of Service Lifecycle Management.....	39
6.4.1 Perceived Technical issues.....	42
6.5 Recap of issues and considerations for OASIS SOA-Tel analysis.....	42
7 Issues on SOA collective standards usage.....	44
7.1 Common Patterns for Interoperable Service Based Communications.....	44
7.1.1 Scenario/purpose.....	44
7.1.2 Scenario/context.....	45
7.1.3 Technical Issues/ Solutions:.....	49
8 Conformance.....	50
Appendix A. Acknowledgements.....	51
Appendix B. Web Services Standards Landscape.....	52
Appendix C. Possible workaround related to issue in Section 3.1 “Transaction Endpoints Specification”	53

Table of Figures

Figure 1: Reference Schema to classify SOA subject areas	9
Figure 2: Mapping of received contributions on Reference Schema.....	11
Figure 3: Transaction endpoints scenario.....	13
Figure 4: Transaction endpoints scenario flow	13
Figure 5: Notification Use Case (a) flow	15
Figure 6: Notification use case (b) flow.....	16
Figure 7: "SOAP" use case representation	19
Figure 8: SOAP message, request formulated by the Service Consumer	20
Figure 9: Message needed by the Service Provider (Ultimate SOAP receiver)	21
Figure 10: Message effectively forwarded by the ESB to the appropriate Service Provider	22
Figure 11: Simplified transaction diagram for the "SAML token correlation" use case.....	26
Figure 12: "SAML token correlation" use case: pictorial representation.....	26
Figure 13: "SAML name Identifier request" use case: pictorial representation.....	28
Figure 14: "SAML Attribute Management request" use case: pictorial representation.....	32
Figure 15: User ID Forwarding use case	32
Figure 16: User ID Forwarding – "Customer care" use case	33
Figure 17: User ID Forwarding – "MVNO" use case	35
Figure 18: TM Forum "SDF Service".....	38
Figure 19: Including management capabilities definition in the SDF Service description	38
Figure 20: SDF Reference Model	40
Figure 21: SDF Service lifecycle phases and associated metadata.....	41
Figure 22: SDF Service Metadata (concepts).....	41
Figure 23: Service Lifecycle Management through SDF	42
Figure 24: Real-time communications in the context of an "any" application seamlessly across any device and network.....	45
Figure 25: Sequence diagram example for the Universal Communication Profile case	47

1 Introduction

Service-Oriented Architecture, SOA, is a design approach that divides everyday business applications into individual processes and functions, otherwise termed “service components”. These service components can then be deployed and integrated among any supporting applications and run on any computing platform. SOA enables a business to drive its application architecture by aligning the business processes with the information technology infrastructure. In effect the composite application becomes a collection of services communicating over a message bus via standard interfaces and allowing each component to be incorporated into the business process flow creating loosely coupled reusable component architecture.

The use of SOA architectural concepts allows the developer to create complex and dynamically changing applications reaching out to other component providers, who may be inside the organization or an external third party component provider.

From the perspective of an application developer, SOA is a set of programming models and tools for creating, locating, and building services that implement business processes. SOA presents a programming model to build complex composite services, and at this time the current industry approach uses web service technologies to implement SOA.

The next generation of applications are adopting a composite model where the components that are involved in the application execution path may be obtained from the efforts of multiple providers, each specializing in certain core competencies. These components will need to provide an open standards based interface to the application plane that is consumable by the tooling that the business community is comfortable with using. This makes it easier to combine components into applications to meet the needs of customers, suppliers and business partners.

This approach allows the application service provider to offer complex services, whose behavior can be dynamically managed to offer the optimal experience for the end user. As well as providing a mechanism to develop rapid applications there are also various management and deployment areas that need to be handled in this multi-component multi-vendor model as each component may have specific deployment or management considerations.

The use of SOA technology within the telecommunications area is expanding as by using a standardized interface to the network the telecommunications enablers can be exposed for consumption by the IT applications running in the business plane. These interfaces can be based upon various aspects of SOA, WSDL, Web Services Description Language, a REST, REpresentational State Transfer, model or other technology. In any case the consuming application can use the relevant IT tool set to bring these enablers into the business process to supply a real time communications service component.

Part of the work being undertaken by the OASIS SOA-TEL TC is to understand how SOA-related specifications and standards are used within the scope of the telecommunications environment and determine if there are any issues when used in this manner.

The objective of this deliverable is to identify possible technical issues related to the utilization of current SOA standards and specifications in the context of telecommunications. Such issues or gaps are illustrated by means of specific use cases.

Amongst future deliverables of the SOA-TEL TC there is a Requirements specification, which will aim to extend the current core SOA enabling stack (Web Services and/or REST, etc.) in support of Telecom needs on the basis of the issues identified within the present document.

1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

47 1.2 Normative References

- 48 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
49 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 50 **[WS-I Basic Profile]** WS-I Basic Profile Version 1.0: "Final Material", available at
51 <http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>.
- 52
- 53 **[WSDL 1.1]** W3C Note (15 March 2001): "Web Services Description Language (WSDL)
54 1.1". <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.
- 55
- 56 **[SOAP 1.2]** W3C SOAP v.1.2, available at <http://www.w3.org/TR/soap12-part1/>
- 57
- 58 **[WS-N 1.3]** OASIS Standard, "Web Services Base Notification 1.3 (WS-
59 BaseNotification)", version 1.3, 1 October 2006. [http://docs.oasis-
60 open.org/wsn/wsn-ws_base_notification-1.3-spec-os.htm](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.htm)
- 61
- 62 **[WS-A 1.0]** W3C Web Services Addressing 1.0 – Core W3C Recommendation 9 May
63 2006, <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>
- 64
- 65 **[WS-S 1.1]** OASIS Standard, "Web Services Security specification, version 1.1", 1
66 February 2006. [http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
67 1.0.pdf](http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf) and <http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf>
- 68
- 69 **[SOA RM 1.0]** OASIS Standard, "OASIS Reference Model for Service Oriented Architecture
70 1.0", Oct. 12, 2006. <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>
- 71
- 72 **[SCA Assembly 1.1]** OASIS Committee Draft 03, "Service Component Architecture Assembly
73 Model Specification Version 1.1", Mar. 09, [http://docs.oasis-
74 open.org/opencsa/sca-assembly/sca-assembly-1.1-spec-cd03.html](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec-cd03.html)
- 75
- 76 **[SOA RA 1.0]** OASIS Public Review Draft 01, "Reference Architecture for Service Oriented
77 Architecture 1.0", Apr. 2008, [http://docs.oasis-open.org/soa-rm/soa-
78 ra/v1.0/soa-ra-pr-01.pdf](http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf)
- 79
- 80 **[WSDL 2.0]** W3C Web Services Description Language (WSDL) Version 2.0 Part 0:
81 Primer, [http://www.w3.org/TR/2007/REC-wsdl20-primer-
82 20070626/Recommendation](http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/Recommendation), June 2007
- 83
- 84 **[SAML 2.0]** OASIS Standard, "Assertions and Protocol for the OASIS Security Assertion
85 Markup Language (SAML) V2.0", March. 2005, [http://docs.oasis-
86 open.org/security/saml/v2.0/saml-2.0-os.zip](http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip)
- 87

88 1.3 Non-Normative References

- 89
- 90 **[WS Landscape]** Possible representation of web services specification landscape, available at
91 <http://www.innoq.com>.

2 Context setting

This section provides a rationale for the classification of the issues presented in the document. Literature on SOA is vast, as the theme has acquired increasing importance and relevance over time. Contextualizations and generalizations can be a difficult task, since many perspectives could be taken into account and importance perceptions may vary depending on reader's interests. Nevertheless, Figure 1 is an attempt to provide a context rationalization of items related to SOA: it was built not with the intent of being rigorous, but rather to provide a possible classification schema for the readers of this document.

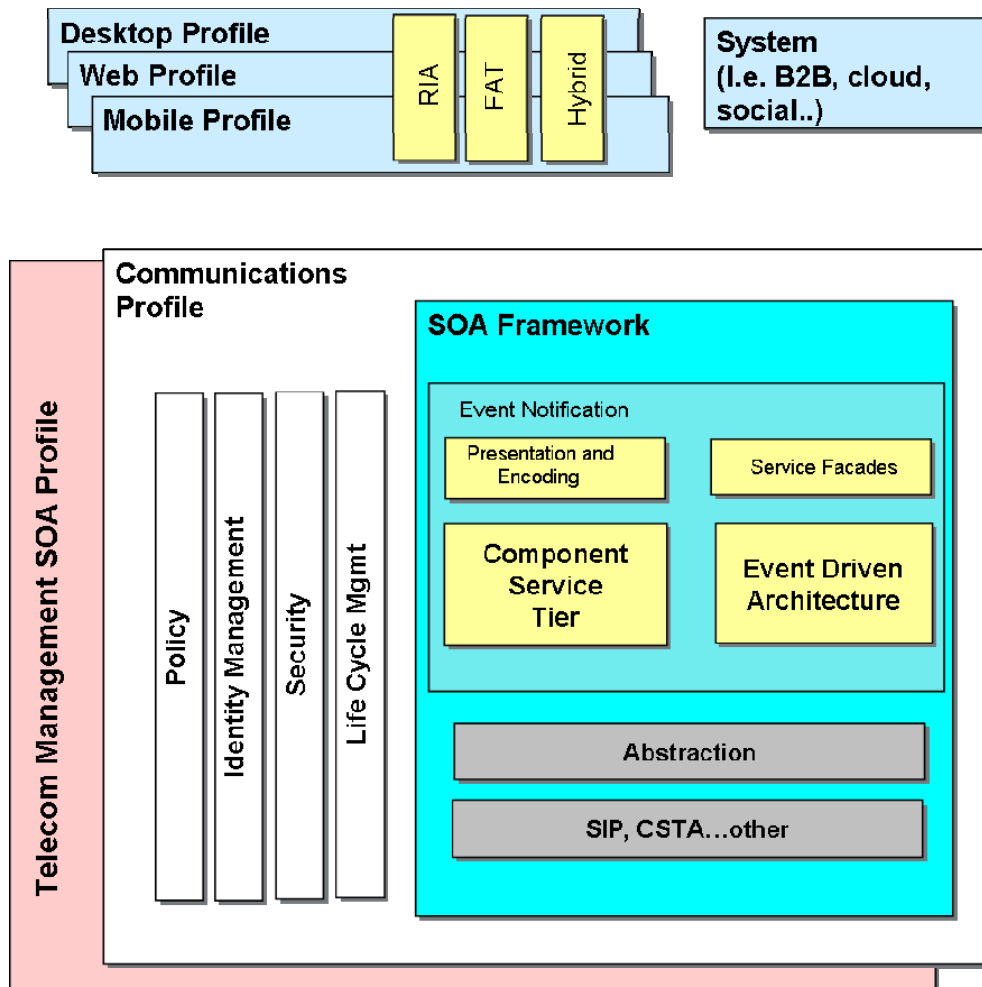


Figure 1: Reference Schema to classify SOA subject areas

The contributions received and analyzed within SOA-TEL on possible issues of SOA standards in the Telecoms context are related to some of the subject areas depicted in Figure 2.

107 The list of received contributions is presented hereafter, while in Figure 2 a mapping of the contributions
108 to the Reference Schema is provided.

- 109 1. **Transaction Endpoints Specification**, related to a possible issue on the W3C WS-Addressing
110 specification; the necessity to specify the endpoint of a final result of a “process/transaction” (i.e.
111 asynchronous response) result should be sent.
- 112 2. **Notification**, related to a possible issue on the OASIS WS-Notification specification; the necessity to
113 specify for the Provider of a notifications service to specify the endpoint to which the Notification
114 should be sent.
- 115 3. **SOAP Protocol** issue, related on a possible issue on the W3C SOAP specification; the necessity for
116 an “intermediate SOAP node” to also cover the role of “SOAP ultimate receiver node”.
- 117 4. **SAML Token Correlation**, related to a possible issue on the OASIS WS-Security specification; the
118 necessity of enabling “correlation” of a security token to another.
- 119 5. **SAML Name Identifier Request**, related to a possible issue on the OASIS SAML specification: the
120 possibility to extend the SAML protocol to enable a Service provider (SP) to register single Users with
121 an Identity Provider (IdP) “on-the-fly”, as the need arises.
- 122 6. **SAML Attribute Management**, related to a possible issue on the OASIS SAML specification: the
123 possibility to extend the SAML protocol to enable a SP (Service Provider) to transmit user attributes
124 to be stored within an IdP (Identity Providers).
- 125 7. **User-ID Forwarding**, related to a possible issue in the OASIS WS-Security specification; the
126 necessity to define a common means to add two (or more) credentials in one message.
- 127 8. **Services exposing Management Interface**, related to possible issues on the OASIS SOA
128 Reference Model (SOA RM) and SOA Service Component Architecture (SCA) Assembly Model; the
129 necessity to specify more than one service interface for a single SOA service.
- 130 9. **Metadata in support of Service Lifecycle Management**, related to the possibility to enrich the
131 OASIS SOA Reference Architecture (SOA RA) with metadata necessary for Service Lifecycle
132 Management identified within the TM Forum SDF program.
- 133 10. **Universal Communications Profile**, related to the specification of a possible common profile for
134 universal interoperability across domains.

135

1. Transaction Endpoints Specification
2. Notification
3. SOAP Protocol
4. SAML Token Correlation
5. SAML Name Identifier Request
6. SAML Attribute Management
7. User-ID Forwarding
8. Services exposing Management Interface
9. Metadata in support of Service Lifecycle Management
10. Universal Communications Profile

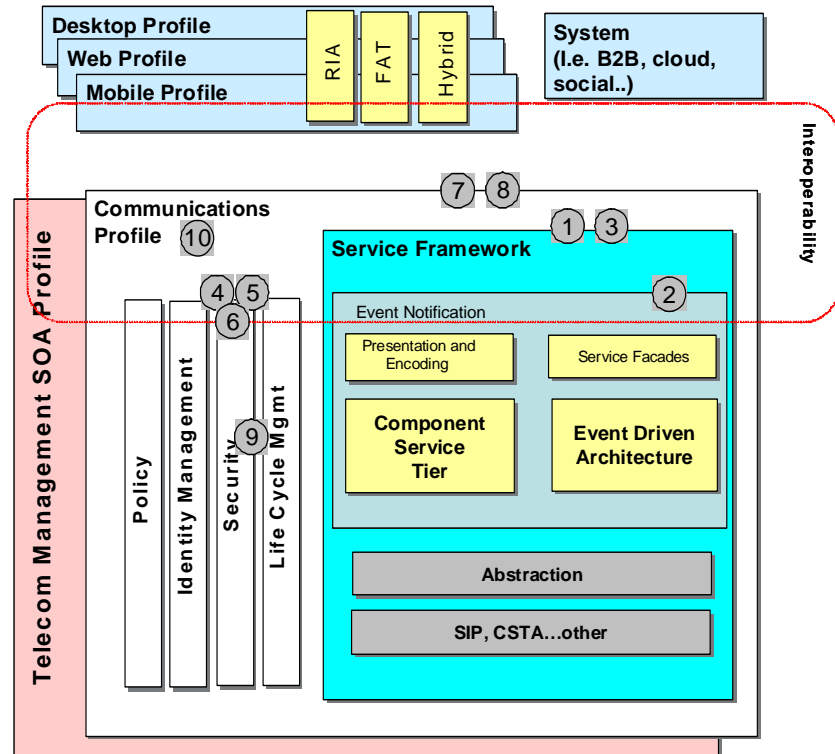


Figure 2: Mapping of received contributions on Reference Schema

136
137
138
139
140
141
142
143
144
145
146
147
148

The document is organized in the following sections:

- Section 3, Issues on Addressing and Notification;
- Section 4, Issues on Communication Protocols;
- Section 5, Issues on Security;
- Section 6, Issues on Management;
- Section 7, Issues on SOA collective standards usage.

All perceived technical issues on SOA standards contained in this document are structured with a description of the context, a use case, and a rationalization of the possible gap within the standard.

149 3 Issues on Addressing and Notification

150 3.1 Transaction Endpoints Specification

151 3.1.1 Scenario/context

152 The issue presented in this section derives from a concrete case, implemented within an operator's SOA
153 Middleware.

154 The operator is in the process of deploying a SOA infrastructure, of which some of the constituting
155 elements are an ESB (Enterprise Service Bus), a BPM (Business Process Manager), some "Service
156 Consumers (systems or applications), some "Service Providers" (systems or applications).

157 An aspect to be considered is that to satisfy performance criteria it has been decided that the ESB must
158 be intrinsically "stateless" (i.e. it must not store any persistence information on destination of incoming
159 service requests).

160 Moreover, the "number" of ESB can vary, i.e. there can be interconnected trunks of different vendors'
161 ESB.

162 3.1.2 Use Case

163 The following Use Case describes the technical problem (Figure 3 and Figure 4). To improve readability
164 the depicted use case presents only one instance of ESB, but the possible solution to the problem must
165 satisfy also the cases of multiple instances of ESB.

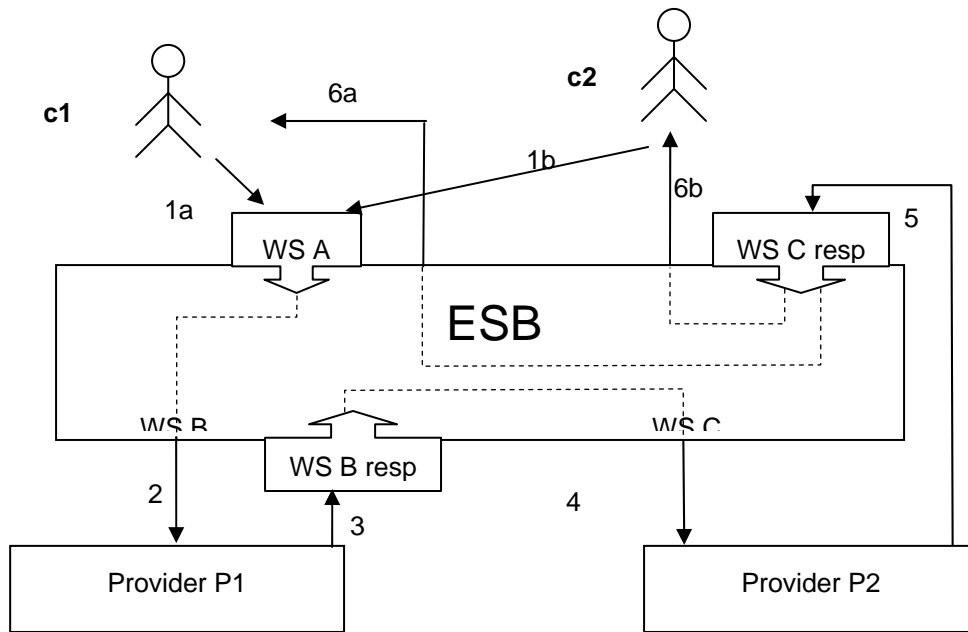
166 A Service Consumer (C1 or C2) invokes a Service, implemented as a Web Service (Web Service A).

167 Such WSA is achieved as an "itinerary" with the composition of more elementary services, provided by
168 Provider P1 and Provider P2.

169 The ESB provides intermediary services for final exposition, enrichment and Data reconciliation and
170 routing.

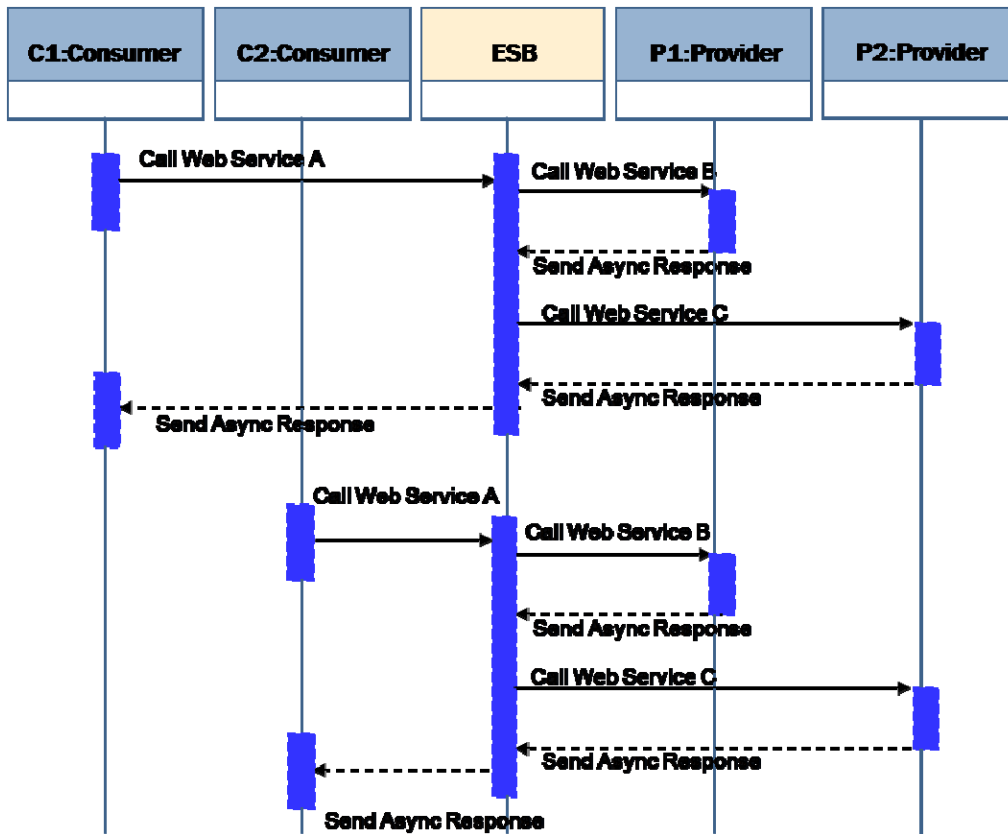
- 171 • Case **A**: C1 is the originator and final receiver.
- 172 • Case **B**: C2 is the originator and final receiver.

173



174
175
176

Figure 3: Transaction endpoints scenario



177
178
179

Figure 4: Transaction endpoints scenario flow

180 Use Case Steps:

181 **Case A**

- 182 • C1 invokes WSA, exposed by ESB.
- 183 • WSA is executed with the internal composition (transparent to C1) and with intermediary services
184 provided by the ESB.
- 185 • At the end of the internal interactions, the ESB forwards the response to C1.

186 **Case B**

- 187 • C2 invokes WSA, exposed by ESB.
- 188 • WSA is executed with the internal composition (transparent to C2) and with intermediary services
189 provided by the ESB.
- 190 • At the end of the internal interactions, the ESB forwards the response to C2.

191 **3.1.3 Perceived Technical Issue**

192 With the current knowledge and expertise, in presence of an ESB offering intermediary services, there is
193 no formal way to specify the endpoint (e.g. C1 or C2) to which the final result of a “process/transaction”
194 (i.e. asynchronous response) result should be sent.

195 Affected specification is W3C **[WS-A]**.

196 **3.2 WS-Notification**

197 **3.2.1 Scenario/context**

198 Event-Driven Architectures are extremely important in environments, like Telecoms, where it is necessary
199 to handle massive network events that have a business value to registered subscribers.

200 Often these solutions rely on proprietary protocols that work against the implementation of SOA
201 principles.

202 There’s a strong technical and business need for a Notify/Subscribe protocol which could be widely
203 adopted and used by Vendors and Telecom Operators. Moreover the protocol should support the
204 presence of intermediaries between the Subscriber and the Notifier.

205 In the following, 2 use cases and related issues are presented, one related to a lack of acceptance of an
206 existing standard by the vendor community, and one on a specific technical issue on existing standards.

207

208 Specifications addressed within this section are:

- 209 • OASIS Web Services Base Notification 1.3 (WS-BaseNotification) **[WS-N]**, OASIS Standard, 1
210 October 2006, http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.htm
- 211 • W3C Web Services Addressing 1.0 **[WS-A]** – Core W3C Recommendation 9 May 2006,
212 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>.

213 **3.2.2 Use Case (A)**

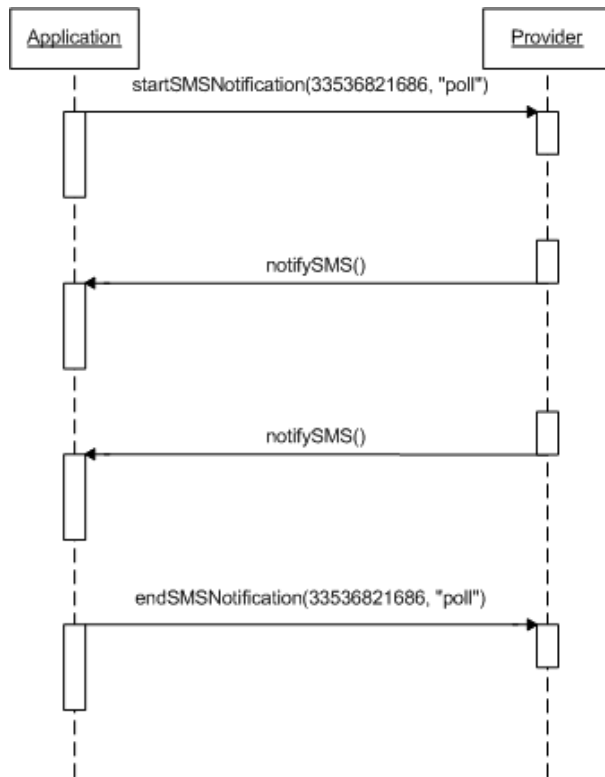
214 The following Use Case describes a technical problem which is common for a Telecom Operator (ref.
215 Figure 5).

216 An Application wants to be notified when a specific “Large Account Mobile Number” receives an SMS with
217 a specific keyword in the message content.

218 Use Case Steps:

- 219 1. The Application informs the Provider that it wants to be notified when the specified Large Account
220 Number “33536821686” receives an SMS containing the word “poll”.
- 221
- 222 2. The Provider notifies the Application when an incoming event from the underlying network
223 responds to the Subscribing criteria.

224 3. The Application informs the Provider that it does not want to be notified anymore when the
 225 specified Large Account Number "33536821686" receives an SMS containing the word "poll".
 226
 227



228
 229
 230 Figure 5: Notification Use Case (a) flow
 231

232 **3.2.3 Perceived technical issue (A)**

233 Currently a commonly used interoperable standard does not exist to address "Notify/Subscribe message
 234 exchanges".

235 The last approved specification, OASIS WS-Notification **[WS-N]**, has been very poorly adopted by the
 236 vendors community and consequently has no interoperability value.

237 The need is that such specification gets endorsed/adopted by the vendor community in order for it to add
 238 value in this specific context.

239
 240 Such lack is perceived as a strong market gap with negative impacts for both Telecom Operators and
 241 Third Parties involved in the development of new services:

- 242 1) Operators are limited in their business development since they must rely on costly proprietary
- 243 solutions and customizations implemented by vendors;
- 244 2) Third Parties, who are typically involved in developing new services for their customers, can not fully
- 245 exploit in their services development the open network infrastructures provided by Telco Operators.

246

247 **3.2.4 Use Case (B)**

248 The following Use Case describes a second technical problem which is common for Telecom Operators
249 (ref. Figure 6).

250 An Application must be notified when a specific “Large Account Mobile Number” receives an SMS with a
251 specific keyword in the message content. There are one or more intermediaries between the Application
252 and the Provider.

253

254 **Use Case Steps:**

255 1. The Application informs the Intermediary that it wants to be notified when the specified Large
256 Account Number “33536821686” receives an SMS containing the word “poll”.

257

258 2. The Intermediary sends the subscription request to the Provider.

259

260 3. The Provider notifies the Intermediary when an incoming event from the underlying network
261 responds to the Subscribing criteria.

262

263 4. The Intermediary sends the notification to the Application.

264

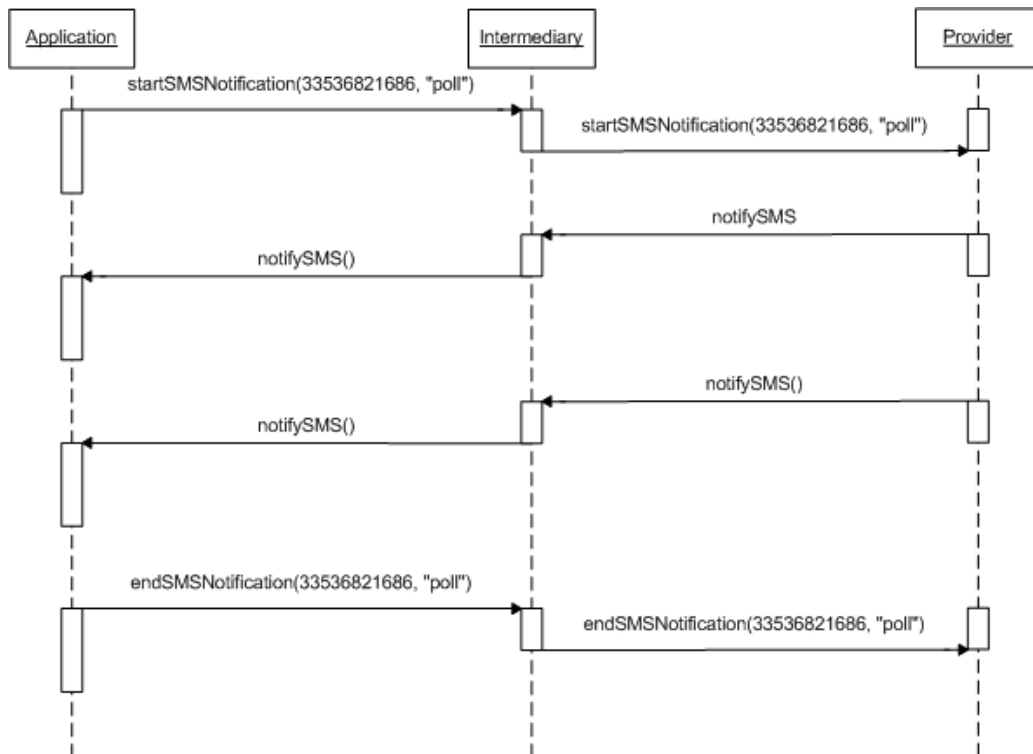
265 5. The Application informs the Intermediary that it does not want to be notified anymore when the
266 specified Large Account Number “33536821686” receives an SMS containing the word “poll”.

267

268 6. The Intermediary sends the “unsubscribe” request to the Provider.

269

270



271

272

273

Figure 6: Notification use case (b) flow

274 **3.2.5 Perceived Technical issue (B)**

275 The last approved specification to support Notify/Subscribe patterns, WS-Notification **[WS-N]**, relies on
276 W3C WS-Addressing **[WS-A]** for the asynchronous delivery of notifications, which means that there is no
277 formal way for the Provider to specify the endpoint to which the Notification should be sent.

278 As an example, in the case illustrated above there is no standard way for the Provider to indicate the
279 original Application as destination of the notification, due to the presence of intermediary (ies) in the path.

280

281 The issue on WS-A impacts thus also the WS-N specification. Refer to Section 3.1 within this document
282 for the technical issues with the WS-A specification.

283 "in presence of intermediary, there is no formal way to specify the endpoint to which the final
284 result of a "process/transaction" (i.e. asynch. response) result should be sent."

285

286 The technical problem here exposed prevents Telecom Operators to develop standardized solutions for
287 the management of "multiple notify/subscribe patterns", and forces to rely on costly customizations and
288 proprietary solutions.

289

290 4 Issues on communications protocols

291 4.1 SOAP

292 4.1.1 Scenario/context

293 The issue presented in this section derives from a concrete case, occurred within the context of the
294 development of a platform for Mobile Virtual Network Operators (MVNOs).

295 This section is related to a possible technical issue within the SOAP 1.2 [**SOAP 1.2**] specification, in
296 particular on the “SOAP Intermediary” and “Ultimate SOAP receiver” concepts.

297 The specification defines the following (within its section 1.5.3):

298

- **Initial SOAP sender**
 - The SOAP sender that originates a SOAP message at the starting point of a SOAP message path.
- **SOAP intermediary**
 - A SOAP intermediary is both a SOAP receiver and a SOAP sender and is targetable from within a SOAP message. It processes the SOAP header blocks targeted at it and acts to forward a SOAP message towards an ultimate SOAP receiver.
- **Ultimate SOAP receiver**
 - The SOAP receiver that is a final destination of a SOAP message. It is responsible for processing the contents of the SOAP body and any SOAP header blocks targeted at it. In some circumstances, a SOAP message might not reach an ultimate SOAP receiver, for example because of a problem at a SOAP intermediary. An ultimate SOAP receiver cannot also be a SOAP intermediary for the same SOAP message (see [2. SOAP Processing Model](#)).

299

300

301 In particular it is stated that

- A **SOAP Intermediary** processes the header of a SOAP message.
- An **Ultimate SOAP receiver** processes the body of a SOAP message and can not also be a SOAP intermediary for the same SOAP message.

305 The issue presented in the following Use Case illustrates the need to have a SOAP Intermediary which
306 must process the body of a SOAP message in addition to its “canonical” role of processing the SOAP
307 message header.

308 The case is included within the activities of deployment of a company-ware SOA infrastructure, of which
309 some of the constituting elements are an ESB (Enterprise Service Bus), some “Service Consumers
310 (systems or applications), some “Service Providers” (systems or applications), a BPM (Business Process
311 Manager), etc.

312 4.1.2 Use Case

313 A Service Consumer C1 (e.g. a CRM application) invokes a Web Service to execute a transaction within a
314 specific business process for the management of Mobile Virtual Network Operators (ref. Figure 7).

315 The access point for the Consumer C1 is the ESB, which exposes such Web Service and moreover
316 executes some of its typical functions such as Data Enrichment and Content Based Routing (CBR).

317

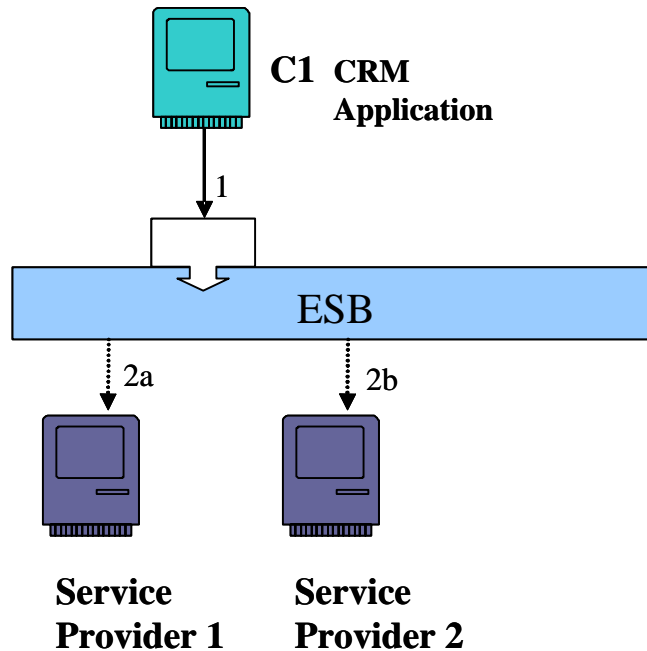


Figure 7: "SOAP" use case representation

318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332

Figure 8 contains the SOAP message which is the request formulated by the Service Consumer (e.g. the CRM application) to the ESB.

The request contains:

- A SOAP Envelope (in **black** color). This is enclosed for completeness but is not subject of discussion within this contribution;
- the SOAP Header, in **red** color;
- The SOAP message Body, in **blue** (and **green**) color.

With reference to the SOAP 1.2 specification, the ESB is a "SOAP Node" (ref. Section 1.5 in the [SOAP 1.2] specification).

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:m0="http://operator/BSS/MVNO/NetProvisioningCustomTypes">
<SOAP-ENV:Header>
<m:Header xmlns:m="http://operator/BSS/MVNO/NetProvisioningHeaderTypes">
  <m:sourceSystem>String</m:sourceSystem>
  <m:businessID>String</m:businessID>
</m:Header>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <m:ActivateLineMessage xmlns:m="http://telecomitalia.it/BSS/MVNO/NetProvisioning">
    <m:Command>
      <m0:description>String</m0:description>
    </m:Command>
    <m:MobilePhoneAccount>
      <m0:telephoneNumber>String</m0:telephoneNumber>
      <m0:ManagedOn>
        <m0:ICCID>String</m0:ICCID>
      </m0:ManagedOn>
    </m:MobilePhoneAccount>
    <m:NetworkProfile>
      <m0:ID>String</m0:ID>
      <m0:TDS>String</m0:TDS>
    </m:NetworkProfile>
    <m:Context>
      <m0:value>String</m0:value>
    </m:Context>
  </m:ActivateLineMessage>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

333

334

335

Figure 8: SOAP message, request formulated by the Service Consumer

336

337 The ESB for this use case must process the body of the SOAP message in order to perform 2 operations:

338

1. "Data Enrichment",

339

The ESB queries a provisioning system to obtain the IMSI of the asset (mobile phone number) in order to add such data to the message: it invokes a Web Service, exposed by that system, which takes in input the ICCD, present in the message, and returns the IMSI.

340

341

342

2. CBR (Content Based Routing)

343

The ESB decides on the final receiver of the SOAP message on the basis of the content of the "Context" field (in **green** in Figure 2).

344

345

Once such tasks are performed, the ESB deletes the "Context" field from the message and subsequently forwards the SOAP message to the selected Service Provider.

346

347

348 **Note:**
 349 The Data Enrichment task is executed with the collaboration of other “Service Providers” (different
 350 than SP1 or SP2), but it is not a subject to be discussed within this contribution: for this reason details
 351 are omitted.
 352
 353 After such tasks are complete, the ESB must forward the SOAP message to the selected Service
 354 Provider, which is the “real” Ultimate SOAP receiver. The message that must be finally sent to the SP by
 355 the ESB is the one depicted in Figure 9.
 356 It is fundamental to state that the Service Provider needs the header present in the SOAP message, e.g.
 357 because the content of the “business ID” field can not be associated to the body of the SOAP message.

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:m0="http://operator/BSS/MVNO/NetProvisioningCustomTypes">
  <SOAP-ENV:Header>
    <m:Header xmlns:m="http://operator/BSS/MVNO/NetProvisioningHeaderTypes">
      <m:sourceSystem>String</m:sourceSystem>
      <m:businessID>String</m:businessID>
    </m:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:ActivateLineMessage xmlns:m="http://operator/BSS/MVNO/NetProvisioning">
      <m:Command>
        <m0:description>String</m0:description>
      </m:Command>
      <m:MobilePhoneAccount>
        <m0:telephoneNumber>String</m0:telephoneNumber>
        <m0:ManagedOn>
          <m0:ICCID>String</m0:ICCID>
          <m0:IMSI>String</m0:IMSI>
        </m0:ManagedOn>
      </m:MobilePhoneAccount>
      <m:NetworkProfile>
        <m0:ID>String</m0:ID>
        <m0:TDS>String</m0:TDS>
      </m:NetworkProfile>
    </m:ActivateLineMessage>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

358
 359 Figure 9: Message needed by the Service Provider (Ultimate SOAP receiver)

360
 361 Nevertheless, given the initial definitions (section 1.5.3 of the SOAP Specification), since the ESB needs
 362 to elaborate the body of the message, it becomes an “Ultimate SOAP receiver” and thus can not be
 363 simultaneously classified as “SOAP Intermediary”.

364 The consequence of this is that the ESB can not forward the header of the SOAP message to the
365 selected Service Provider (i.e. to the “real” Ultimate SOAP receiver).
366 Thus the message really forwarded by the ESB is depicted in Figure 10.
367

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:m0="http://operator/BSS/MVNO/NetProvisioningCustomTypes">
  <SOAP-ENV:Body>
    <m:ActivateLineMessage xmlns:m="http://operator/BSS/MVNO/NetProvisioning">
      <m:Command>
        <m0:description>String</m0:description>
      </m:Command>
      <m:MobilePhoneAccount>
        <m0:telephoneNumber>String</m0:telephoneNumber>
        <m0:ManagedOn>
          <m0:ICCID>String</m0:ICCID>
          <m0:IMSI>String</m0:IMSI>
        </m0:ManagedOn>
      </m:MobilePhoneAccount>
      <m:NetworkProfile>
        <m0:ID>String</m0:ID>
        <m0:TDS>String</m0:TDS>
      </m:NetworkProfile>
    </m:ActivateLineMessage>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

368
369
370 Figure 10: Message effectively forwarded by the ESB to the appropriate Service Provider
371

372 This is a real case faced by the operator, and to overcome the problem some costly ad-hoc
373 developments-customizations were necessary to **re-build / reinsert** the necessary header within the
374 message before the ESB could forward the “complete” message to the final Service Provider.

375 4.1.3 Perceived Technical issue

376 In the SOAP specification the following is stated.

377 -----

378 2.1 SOAP Nodes

379 A SOAP node can be the initial **SOAP sender**, an **ultimate SOAP receiver**, or a **SOAP intermediary**. A
380 SOAP node receiving a SOAP message **MUST** perform processing according to the SOAP processing
381 model as described in this section and in the remainder of this specification, etc.

382
383
384

385 **2.2 SOAP Roles and SOAP Nodes**

386 In processing a SOAP message, a SOAP node is said to act in one or more SOAP roles, each of which is
 387 identified by a URI known as the SOAP role name. The roles assumed by a node MUST be invariant
 388 during the processing of an individual SOAP message. This specification deals only with the processing
 389 of individual SOAP messages. No statement is made regarding the possibility that a given SOAP node
 390 might or might not act in varying roles when processing more than one SOAP message.

391
 392 **Table 2** defines three role names which have special significance in a SOAP message (see **2.6**
 393 **Processing SOAP Messages**).
 394

Table 2: SOAP Roles defined by this specification		
Short-name	Name	Description
Next	"http://www.w3.org/2003/05/soap-envelope/role/next"	Each SOAP intermediary and the ultimate SOAP receiver MUST act in this role.
None	"http://www.w3.org/2003/05/soap-envelope/role/none"	SOAP nodes MUST NOT act in this role.
ultimateReceiver	"http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver"	The ultimate receiver MUST act in this role.

395
 396 In addition to the SOAP role names defined in **Table 2**, other role names MAY be used as necessary to
 397 meet the needs of SOAP applications.

398 -----

399
 400 Due to the fact that the ESB (as a SOAP Node) processes the body of the message, it is classified as
 401 "ultimateReceiver".

402
 403 As a consequence, the ESB can not "Forward" the SOAP Header to the appropriate Service Provider (ref.
 404 Sections 2.7.1 of the SOAP specification) since it has value "ultimateReceiver". The following table
 405 depicts the behavior of the ESB being an ultimateReceiver.
 406

Role		Header block	
Short-name	Assumed	Understood & Processed	Forwarded
next	Yes	Yes	No, unless reinserted
		No	No, unless relay ="true"
user-defined	Yes	Yes	No, unless reinserted
		No	No, unless relay ="true"
	No	n/a	Yes
ultimateReceiver	Yes	Yes	n/a
		No	n/a
none	No	n/a	Yes

407
 408

409 The case presented shows that a SOAP Intermediary (the ESB), which is clearly not the “ultimate
410 receiver” of the SOAP message, is forced to assume the role of “ultimateReceiver” since it processes
411 the body of the message. This prevents the ESB to correctly perform its “proper” intermediary role, since
412 “An ultimate SOAP receiver cannot also be a SOAP intermediary for the same SOAP message”.

413 The perceived technical gap suggested by the operator is that the SOAP specification should be modified
414 in order to enable a SOAP Intermediary node to “forward” the SOAP Header in automatic mode (thus
415 without the Header reinsertion) even if such node performs some processing operation over the body of
416 the SOAP message.

417 Another way of expressing this perceived gap is to state that currently only 3 roles are allowed for a
418 SOAP Node (i.e. initial SOAP Sender, SOAP intermediary, SOAP ultimate receiver – section 2.1 of the
419 SOAP 1.2 specification), while a probable fourth role enabling the simultaneous body processing and
420 header forwarding of a specific SOAP message may be needed.

421 Should the specification already enable this, OASIS SOA-TEL TC suggests to modify them in order to
422 avoid possible ambiguities and misinterpretations.

423

424 5 Issues on Security

425 5.1 SAML Token Correlation

426 5.1.1 Scenario/context

427 The issue presented in this section derives from a concrete case of telecommunications services' sales
428 and post-sales: in particular the activation and provisioning of ADSL service to residential customers.

429 The business process under analysis is complex and necessitates to be orchestrated by a BPM
430 (Business Process Management) application.

431 Such process is a "long-running" type process: in fact one of its tasks requires a human intervention
432 within the central office, which can be executed within hours (or days).

433 This implies that the process must be handled in a different mode from the "security management"
434 perspective.

435 This section addresses potential issues within the OASIS Web Services Security specification, **[WS-S**
436 **1.1]**.

437 5.1.2 Use Case

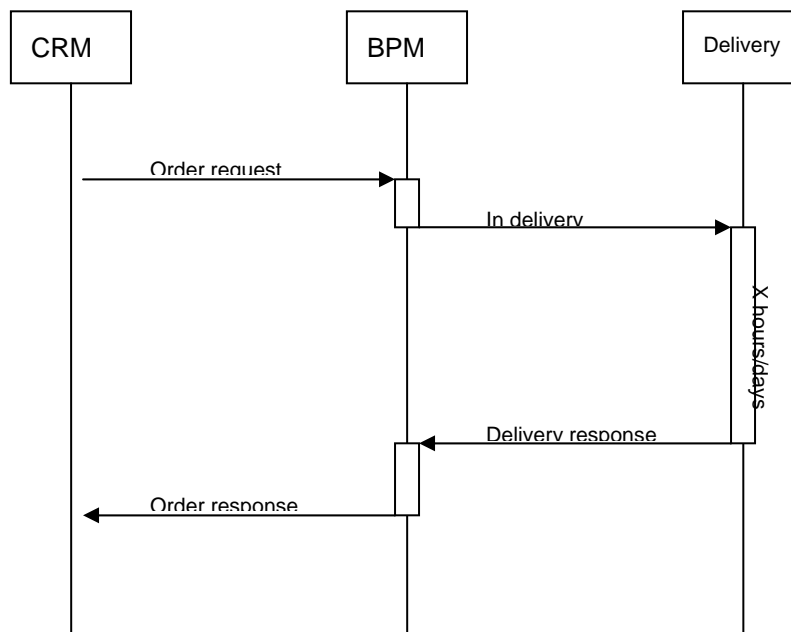
438 A consumer, e.g. a CRM application invokes a service to execute a specific business process, the
439 activation of ADSL services for a residential customer.

440 The BPM application gets in charge of the orchestration/execution of such processes.

441 Given the fact that the process is "long-running", the BPM shall, at a given point, suspend the
442 orchestration/execution of the process until it will receive a specific "activity closure" event from a back
443 office system once the appropriate technician will have terminated his manual tasks.

444 The following schema Figure 11 depicts a simplified transaction diagram, while Figure 12 provides a
445 pictorial representation of the Use Case.

446

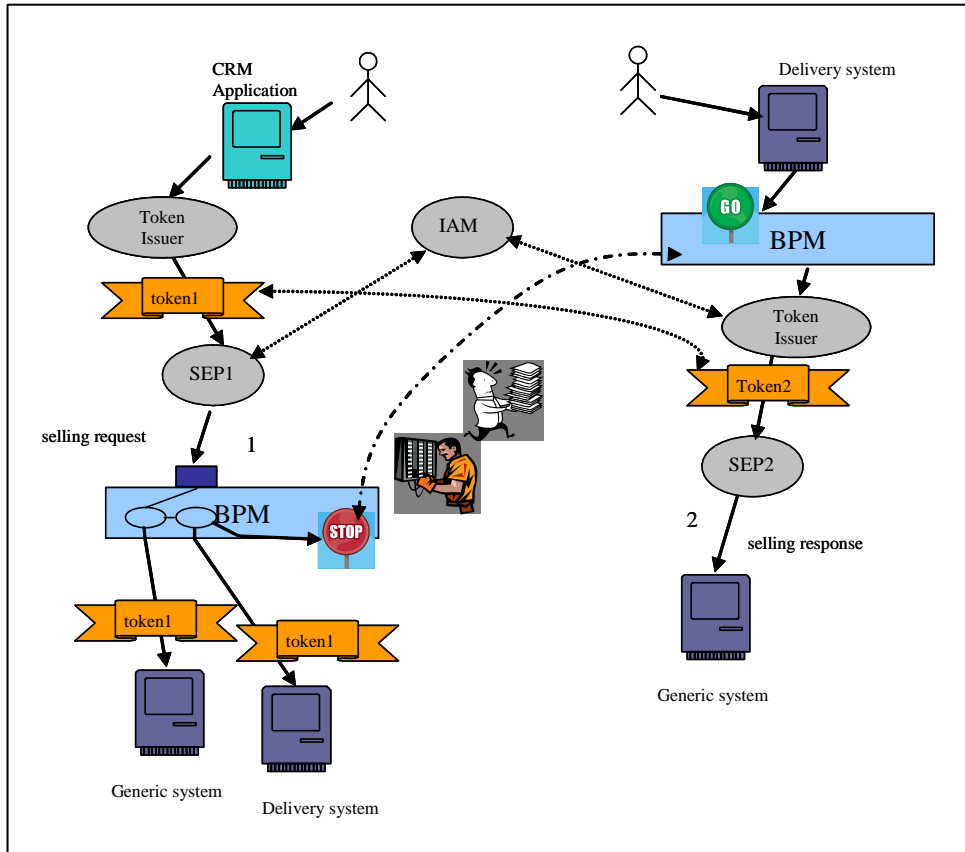


447

448

449
450

Figure 11: Simplified transaction diagram for the “SAML token correlation” use case



451
452

Figure 12: "SAML token correlation" use case: pictorial representation

453

454 **Use Case steps.**

- 455 • The CRM sends an ADSL activation request.
- 456 • The consumer (CRM) provides its credentials to a Token Issuer and requires the generation of a security token, "token1". The token is associated to the initial message and has limited duration, since
- 457 extending it would mean to have a weaker security policy.
- 458
- 459 • The Security Enforcement Point, interacting with the policy decision point (IAM) (Identity Access
- 460 Manager), applies the authentication and authorization policies.
- 461 • The BPM orchestrates the process interacting with the various services exposed by the involved
- 462 systems within the company SOA infrastructure. All interactions are executed with the "token1" as
- 463 security token.
- 464 • When appropriate, the BPM invokes a service exposed by a Delivery system to obtain a physical
- 465 configuration within the central office. At this stage the BPM suspends the execution of the business
- 466 process (the duration of the task may require hours or days), awaiting for the reception of a specific
- 467 "activity closure" event.
- 468 • The Delivery System activates the technical configuration task.
- 469 • A human intervention is performed within the central office.
- 470 • Once this task is terminated, the technician reports the "activity closure" on the Delivery system,
- 471 which generates the "activity closure" event for the BPM.
- 472 • The BPM resumes the suspended process, invoking the "next step" in the ADSL activation process.
- 473 • If the security token "token1" is expired, the BPM requests the Token Issuer to generate a new
- 474 security token, "token2", since the previous is not valid any more.
- 475 • The remaining portion of the process is executed utilizing the new security token, "token2".

476 5.1.3 Perceived Technical issue

477 In the described scenario the issue is related to which credentials (capabilities) must be utilized to
478 generate the security token “token2”.

479 The BPM is responsible for the orchestration/execution of the process, and is the entity which is entitled
480 to request the generation of the new security token “token2”, which is of course different from “token1”.

481 This is a weakening factor for the “security architecture”, since an element of the middleware
482 infrastructure (the BPM) would need to request the generation of security tokens which are not
483 “correlated” (or “directly coupled”) to the real entity which requires the initiation of the business process
484 (i.e. the CRM application, thus the CRM sales representative) and to the business process itself. It is a
485 requirement for the Telecom Operator to reduce such potential security threats.

486 It should be possible for the BPM to request the Token Issuer to generate a new token “associated” to the
487 “token1”, and to maintain evidence of that correlation, in order to authorize the BPM itself, once security
488 checks are validated by the IAM, to invoke all pending services within the second part of the process
489 because such invocations are “really” part of a “security authorized” business process.

490 The WS-Sec specification [WS-S 1.1], in Section 7 - row 824, states that mechanisms for referencing
491 security tokens are defined.

492 In row 870 the following is stated:

493 -----

494 870 /wsse:SecurityTokenReference/@wsse:Usage

495 871 This optional attribute is used to type the usage of the

496 872 <wsse:SecurityTokenReference>. Usages are specified using URIs and multiple

497 873 usages MAY be specified using XML list semantics. **No usages are defined by this**

498 874 **specification.**

499 -----

500

501 Thus, from a syntactical perspective, the specification enables the “correlation” of a security token to
502 another one, but it does not prescribe how such correlation should be formalized.

503

504 Moreover, within non-normative Appendix D “SecurityTokenReference Model”, specific examples of
505 security token referencing are provided, with emphasis of the “signature referencing”.

506 Within this appendix, Row 2413 to 2432 do provide an example of “non-signature references”, but the
507 specification states that

508

509 2430 *This may be an expensive task and in the*

510 2431 *general case impossible as there is no way to know the "schema location" for a specific*

511 2432 *namespace URI.*

512

513 In conclusion, the lack of normative guidelines on how to address this problem is perceived as a strong
514 issue for a Telecom Operator because the “correlation” problem must anyhow be solved, but adopted
515 solutions result to inevitably be proprietary, costly, non-standard, vendor/platform dependent
516 customizations.

517

518 5.2 SAML Name Identifier Request

519 5.2.1 Scenario/context

520 The context of this section is that of a SP (Service Provider) being newly added to the circle of trust of an
521 IdP (identity Provider).

522 Currently, as soon as a SP becomes a member of the circle of trust of an IdP, the SP is forced to import
523 all of the SP's Users into the IdP's databases.

524 The objective of this contribution is to propose a modification to the current SAML V2.0 specification
525 (saml-core-2.0-os.pdf) so that the SP can be enabled to register single Users with the IdP "on-the-fly", as
526 the need arises. Such goal can be achieved with the introduction of a new SAML protocol, named "SAML
527 Name Identifier Request" within the SAML specification.

528 SAML supports SPs to get attributes about Users from an IdP. Regarding name identifiers, the SP usually
529 sends an AuthnRequest to the IdP. Then, the IdP sends an AuthnResponse containing a NameIdentifier
530 ("Subject") back to the SP. However, if a SP is newly added to the circle of trust of an IdP, the IdP will not
531 know of the User identifiers of the SP, which is required in order for the IdP to authenticate the Users of a
532 SP.

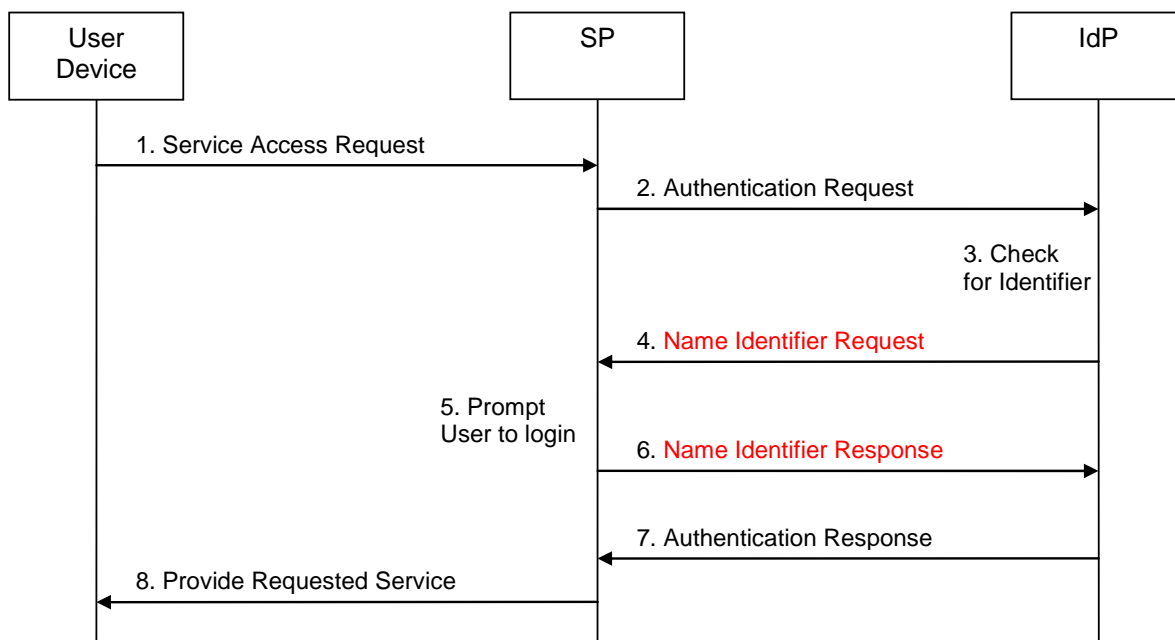
533 The issue highlighted in this section aims at possibly extending the SAML specifications.

534 5.2.2 Use Case

535 A user device, a SP and an IdP are the actors of this use case of the SAML Name Identifier Request
536 mechanism. The SP is new to the circle of trust of the IdP. The IdP does not know a name identifier of the
537 user device. The IdP requests a name identifier from the SP, who sends the desired name identifier to the
538 IdP.

539 Figure 13 provides a high-level message flow illustrating this SAML Name Identifier Request use case.
540 Messages 4 and 6 belong to the SAML Name Identifier Request protocol this contribution is aiming at.
541 These messages are interlaced into the SAML Authentication Request and Response exchange between
542 SP and IdP and are not specified in SAML V2.0 yet (therefore, marked in red):

543



544

545

546

Figure 13: "SAML name Identifier request" use case: pictorial representation

547 The single steps of this use case are as follows:

548

- 549 1) The user requests access to a service offered by a SP. The user device does not include any
550 authentication credentials.
- 551 2) Since access to this service requires the User to be authenticated but the request in step 1 does
552 not include any authentication credentials, the SP sends an Authentication Request to the IdP.
553 This Authentication Request may be passed to the IdP via the user device using redirection.
- 554 3) The IdP checks the Authentication Request received in step 2, and - as the SP is new to the IdP's
555 circle of trust - the IdP determines that it does not have an identifier stored in its database for the
556 User for the given SP.

557 Conventionally, the IdP would respond to the Authentication Request by issuing an error
558 message or a randomly generated identifier. This, however, is problematic: In the former case,
559 the service access request in step 1 breaks down. In the latter case, the SP has to ask the user
560 for his credentials and then send (usually via a backchannel) a message to the IdP indicating that
561 from now on the IdP should use the "real identifier" instead of the random one for the given user
562 (this could be done via the NameIdentifier Management Protocol).

- 563 4) This step is not defined in SAML V2.0: Since the IdP has realized in step 3 that it does not have
564 an identifier for the combination of the User and the SP, the IdP generates a message called
565 Name Identifier Request and sends it to the SP.
- 566 5) Upon receipt of the Name Identifier Request, the SP recognises that the IdP does not have an
567 identifier for the combination of SP and User. Therefore, the SP prompts the User to log in to the
568 SP.
- 569 6) This step is also not defined in SAML V2.0: The SP sends a message called Name Identifier
570 Response to the IdP. This response message includes the identifier for the combination of User
571 and SP that the IdP is to use in any further communication and authentication processes.
- 572 7) On receipt of the Name Identifier Response, the IdP stores the identifier contained in the Name
573 Identifier Response in its database. The IdP sends an Authentication Response to the SP, which
574 uses the identifier received in step 6.
- 575 8) The SP grants the User access to the requested service.

576 **5.2.3 Perceived Technical issue**

577 This contribution aims at introducing a new SAML protocol called SAML Name Identifier Request protocol
578 into the SAML 2.0 specifications.

579 **5.3 SAML Attribute Management Request**

580 **5.3.1 Scenario/context**

581 More and more services and applications are becoming available on the Internet, and many of these
582 services and applications require authentication. With the convergence of telco and Internet domain, the
583 telco has added functionality, namely IDM functions. The telco operator will collaborate with several SPs,
584 that in return depend on the telco's profile and attribute store. This causes a scenario where not the SP
585 manages the attributes, but the telco operated IDM.

586 One approach that has been developed to assist users to access multiple services and applications, each
587 requiring separate authentication procedures, involves the use of identity federation.

588 Security Assertion Markup Language (SAML) is an XML standard for exchanging authentication and
589 authorisation data between security domains. For example, SAML is used for exchanging assertion data
590 between an identity provider (a producer of assertions) and a service provider (a consumer of assertions).

591 The issue highlighted in this section aims at possibly extending the SAML specifications.

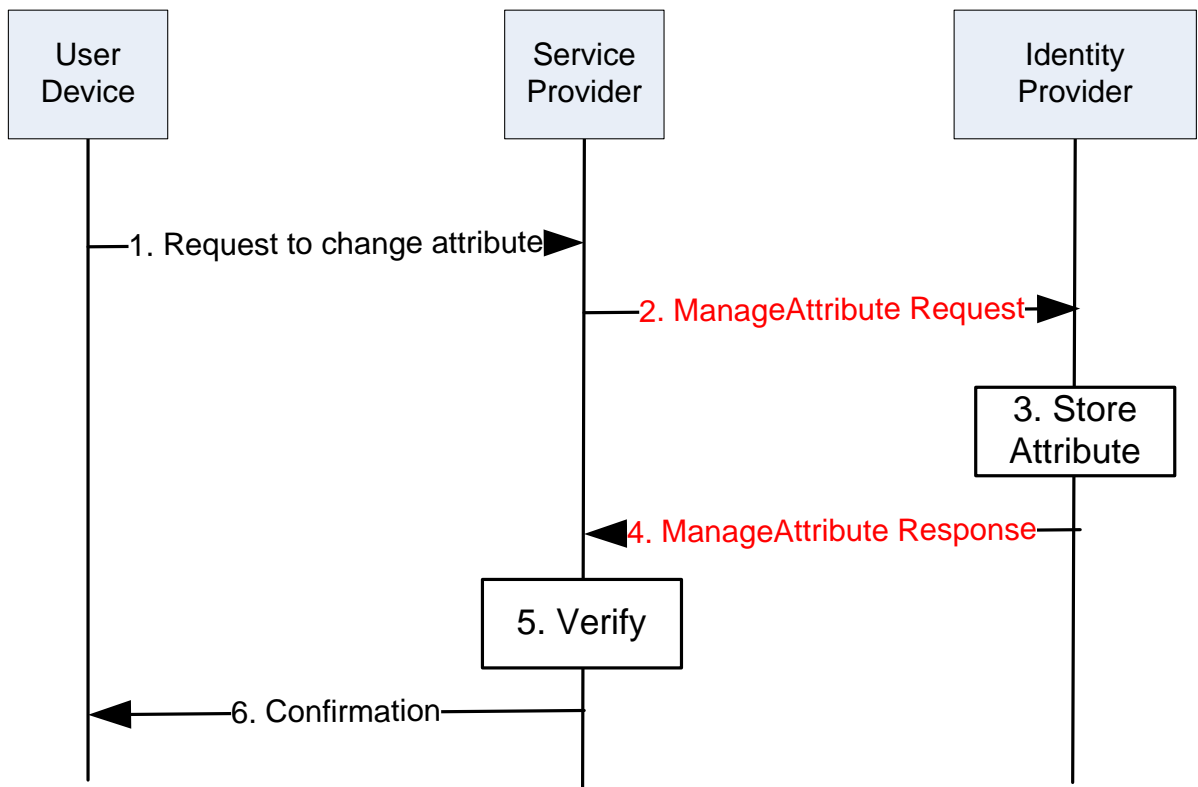
592 **5.3.2 Use Case**

593 A user wishes to use his attribute information across multiple service providers, such attribute information
594 can be layout, preferred email address, etc. Today, these attributes are stored locally at each of service
595 provider. Thus, user will have to enter and changes the same attributes multiple times in order to ensure
596 they are consistent for each of the different service providers the user has an account with, resulting in a
597 bad user experience.

598 The user creates a temporary or transient account. The service provider allows the user to set specific
599 settings like coloring, text size, etc. But he/she does not want to set these setting again each time the
600 user logs in because the service provider will not be able to link the attributes for a user’s temporary
601 account with the user’s permanent account. This is because by the very nature of a temporary or
602 transient account the next time the user logs on to the service provider the user will have a different
603 username and so the service provider will not be able to link the attributes for a user’s temporary account
604 with the user’s permanent account.

605

606 Figure 14 provides a high-level message flow outlining the proposed SAML Attribute Management
607 protocol:



608

609 Figure 14: “SAML Attribute Management request” use case: pictorial representation

610

611

612 The ManageAttribute Request and Response messages are marked in red since the SAML 2.0 does not
613 support such messages yet. The ManageAttribute Request allows the Service Provider to manage
614 attributes stored on the Identity Provider side. As an example, the following XML instance of a
615 ManageAttribut Request asks the Identity Provider to set the value of the “mail” attribute to
616 “trscavo@gmail.com”:

617

618 The following example shows what such a change in the specification would enable to do:

```
619 <samlp:ManageAttributeRequest
620   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
621   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
622   ID="aaf23196-1773-2113-474a-fe114412ab72"
623   Version="2.0"
624   IssueInstant="2006-07-17T20:31:40Z">
625   <saml:Issuer
626     Format="urn:oasis:names:tc:SAML:1.1:nameid-
627   format:X509SubjectName">
628     C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
629   </saml:Issuer>
630   <saml:Subject>
631     <saml:NameID
632       Format="urn:oasis:names:tc:SAML:1.1:nameid-
633   format:X509SubjectName">
634       C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
635     </saml:NameID>
636   </saml:Subject>
637   <saml:AttributeStatement>
638     <saml:Attribute
639       xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
640       x500:Encoding="LDAP"
641       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
642       Name="urn:oid:1.3.6.1.4.1.1466.115.121.1.26"
643       FriendlyName="mail">
644     <saml:AttributeValue
645       xsi:type="xs:string">trscavo@gmail.com</saml:AttributeValue>
646     </saml:Attribute>
647   </saml:AttributeStatement>
648 </samlp:ManageAttributeRequest>
```

649 **5.3.3 Perceived Technical issue**

650 The SAML protocol currently provides two methods that enable *a service provider to retrieve attributes*
651 *relating to a user from identity provider.*:

- 652 • The first method is an attribute push method in which the identity provider can send attribute
653 information within the SAML assertion provided in response to the service provider's user
654 authentication request.
- 655 • The second method is an attribute pull method in which the service provider can use an
656 AttributeAuthority message or an AttributeQuery message to retrieve information regarding user
657 attributes from the identity provider once the user has been authenticated by the identity provider.

658

659 → In both methods described, the service provider can only obtain information relating to the attributes of
660 the user logged into the service provider.

661 → There currently exists no mechanism to enable a service provider to transmit user attributes to be
662 stored at the identity provider. This contribution identifies the use case of such mechanism.

663

664 The issue highlighted in this section aims at possibly extending the SAML specifications.

665 5.4 User ID Forwarding

666 5.4.1 Scenario/context

667 The issue presented in this section derives from a concrete case of activities performed by an operator in
668 order to define and implement a “security architecture” for its SOA middleware infrastructure.

669 This section addresses potential issues within the OASIS Web Services Security specification ([WS-S
670 1.1]).

671 Specifically such issues/limitations are related to the necessity of forwarding the User ID across the SOA
672 Infrastructure.

673 5.4.2 Use Cases

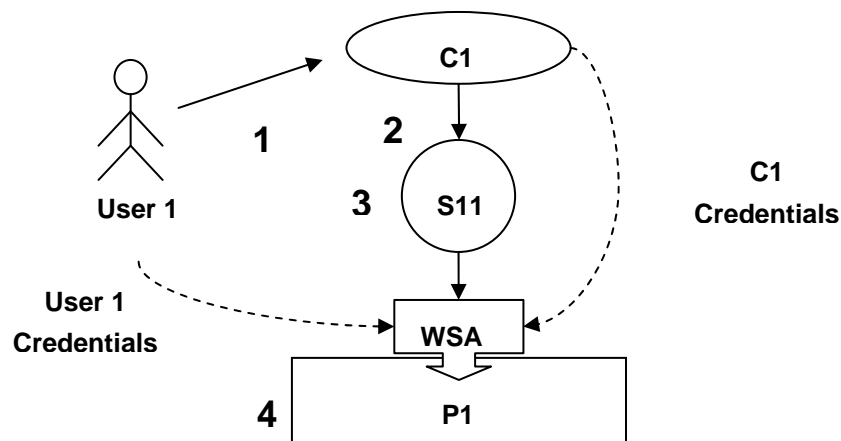
674 In order to better describe the potential technical issues, hereafter a use case is presented (ref. Figure
675 15), with two possible different example scenarios. The use case is that of a Web Service exposed by an
676 Application Provider, and the scenarios are:

- 677 • Customer Care portal accessed by both operator customers and personnel (Call Center Operators),
678 each of them having different “rights” on accessed data.
- 679 • Telco Messenger Service accessed by different MVNOs (Mobile Virtual Network Operators), each of
680 them having different “rights” on accessed data.

681

682 Use case Description

683



684

685

686

Figure 15: User ID Forwarding use case

687

688 1. User 1 accesses a front-end application (C1) using his Credentials (i.e. SSO Token).

- 689 2. C1 invokes a Web Service (WS-A) exposed by P1 and passes the User's credentials (i.e. SAML
 690 Assertion) and its credentials (i.e. X.509 Certificate) for XML Encryption and XML Signature (WS-
 691 Security 1.1).
- 692 3. S1 (Security Enforcement Point) handles the invocation message and enforces the AAA policies:
 693 a. It validates C1 X.509 Certificate.
 694 b. It verifies the XML Encryption and Signature using the public key of C1.
 695 c. It verifies if C1 is authenticated & authorized to access the WS-A (C1 X.509 Certificate).
 696 d. It verifies if the SAML Assertion and User's token are still valid.
 697 e. It verifies if User 1 is authenticated & authorized to access WS-A.
 698 4. P1 (Provider) runs the business logic.

699 **5.4.2.1 Customer Care portal accessed by both operator customers and**
 700 **personnel (Call Center Operators)**

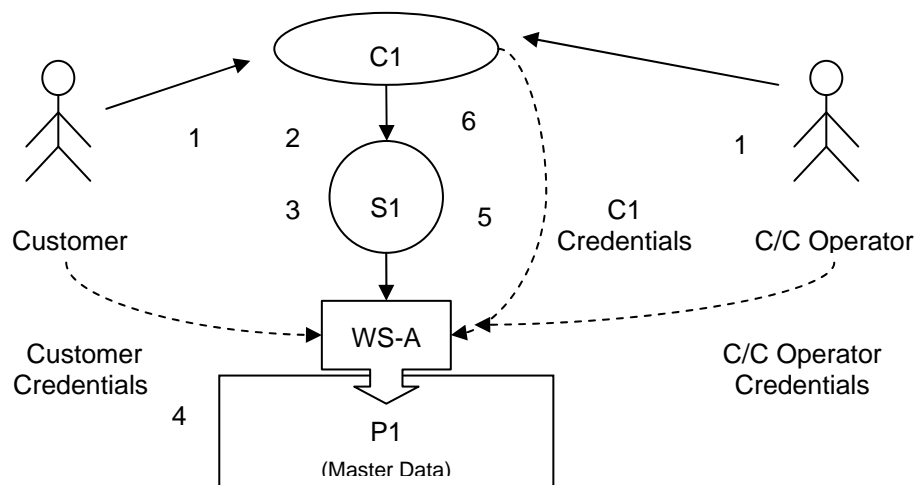
701 C1 is a Portal for Customer Caring that consumes a Web Service (WS-A) for retrieving profile information.
 702 It is used by both Customers (for Self Caring) and Call Center Operators (ref. Figure 16).

703 Some of the available information such as: incoming and outgoing calls, personal information or credit
 704 cards details are ruled by privacy policies.

705 Obviously WS-A and all its operations are accessible by C1 but information provided as result or specific
 706 details depend on the original requester: a Customer could have full access on all information and details
 707 available on its profile while a Call Center Operator could be granted to view only a subset such data (i.e.
 708 partial call numbers, filtered credit cards details, etc.).

709 In the following scenarios C1 invokes WS-A for retrieving the list of incoming call numbers for specific
 710 customers:

711



712

713

714 Figure 16: User ID Forwarding – “Customer care” use case

715

716 **Scenario 1 (Operator's Customers)**

- 717 1) A Customer accesses C1 to view the list of outgoing calls by using his Credentials (i.e. SSO
 718 Token).
- 719 2) C1 invokes a Web Service (WS-A) exposed by P1 passing the Customer's credentials in a SAML
 720 Assertion and using its X.509 Certificate for XML Encryption and XML Signature (WS-Security
 721 1.1).
- 722 3) S1 (Security Enforcement Point) handles the invocation message and enforces the AAA policies:
 723 a. It validates C1 X.509 Certificate,
 724 b. It verifies the XML Encryption and Signature using the public key of C1,

- 725 c. It verifies if C1 is authenticated & authorized to access the WS-A (C1 X.509 Certificate),
- 726 d. It verifies if the SAML Assertion and User's token are still valid,
- 727 e. It verifies if operator Customers is authenticated & authorized to invoke WS-A and what
- 728 level of information could access.
- 729 4) P1 (Provider) runs the business logic.
- 730 5) S1 receives the result from P1 and applies all the privacy policies in order to then return the data
- 731 to C1
- 732 6) C1 shows the entire results to Customers such as:
- 733
- 734 03/27/09 11:39 3355799553 05:37
- 735 03/27/09 12:03 3359955125 10:57.
- 736

Scenario 2 (Call Center Operator)

- 738 1) A Call Center Operator accesses to view the list of incoming call numbers for a specific customer
- 739 by using his Credentials (i.e. SSO Token).
- 740 2) C1 invokes a Web Service (WS-A) exposed by P1 passing the Operator's credentials in a SAML
- 741 Assertion and using its X.509 Certificate for XML Encryption and XML Signature (WS-Security
- 742 1.1).
- 743 3) S1 (Security Enforcement Point) handles the invocation message and enforces the AAA policies:
- 744 a. It validates C1 X.509 Certificate,
- 745 b. It verifies the XML Encryption and Signature using the public key of C1,
- 746 c. It verifies if C1 is authenticated & authorized to access the WS-A (C1 X.509 Certificate),
- 747 d. It verifies if the SAML Assertion and User's token are still valid,
- 748 e. It verifies if C/C Operator is authenticated & authorized to invoke WS-A and what level of
- 749 information could access.
- 750 4) P1 (Provider) runs the business logic.
- 751 5) S1 receives the result from P1 and applies all the privacy policies in order to then return the data
- 752 to C1.
- 753 6) C1 shows the entire results to C/C Operator such as:
- 754
- 755 03/27/09 11:39 3355799XXX 05:37
- 756 03/27/09 12:03 3359955XXX 10:57
- 757

5.4.2.2 Telco Messenger Service accessed by different MVNOs (Mobile Virtual Network Operators)

760 An operator has released a new integration layer called "Services Exposure" (SE) dedicated to supply all
 761 possible services (Telco, OSS and BSS) needed to any MVNO. At the moment the operator has 2 MVNO
 762 customers which consume more or less the same services, but with different policies and SLAs ruled by
 763 specific service contacts (ref. Figure 17).

764 The possibility to uniquely identify the NVNO that is using a service and enforce ad-hoc policies becomes
 765 essential to enable the operator to guarantee those contracts.

766 In addition to that all services exposed by the Service Exposure are potentially consumable by any other
 767 operator application. Therefore the possibility to identify also the application consumer is strong
 768 requirement for an operator.

769 In the following scenario MVNO1 and MVNO2 invoke WS-A to send messages to their customers, but
 770 while MVNO1 can send all types of messages (i.e. SMS, Reliable SMS, MMS, email, etc.), MVNO1 can
 771 send only SMS and MMS:

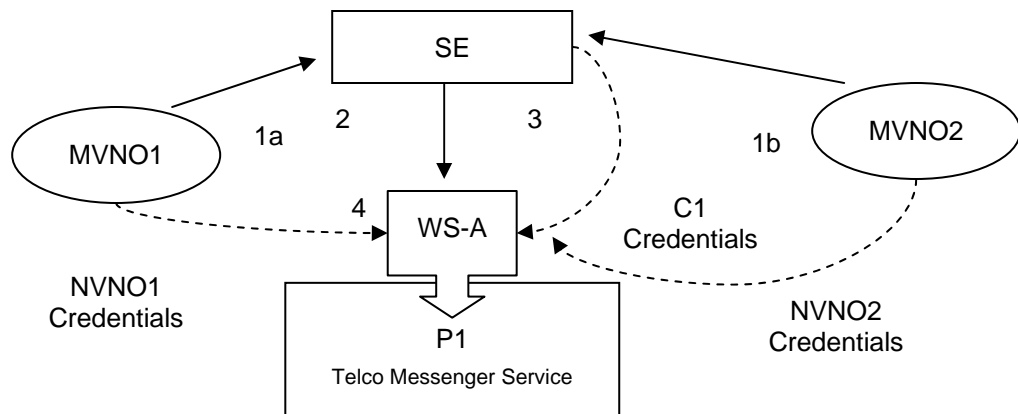


Figure 17: User ID Forwarding – “MVNO” use case

772
773
774
775
776
777
778
779
780
781

- 1) MVNO1 and MVNO2 invoke a service exposed by SE for sending messages.
- 2) SE enforce the AAA policies based on services contracts specific for each MVNOs.
- 3) SE verifies which types of messages MVNO1 and MVNO2 can send.
- 4) SE forwards the invocations to WS-A using its credentials (i.e. X.509 Certificate) and including the MVNO credentials (i.e. SAML Assertion).

5.4.3 Perceived Technical issue

At the moment it seems to be impossible to add two (or more) credentials in one message. OASIS WS-Sec specifications [WS-S 1.1], Section 6, “Security Tokens” rows 717 and 719, may offer a possibility to address the issue.

786
787
788
789
790
791

In row 717 and following it is stated:

717 /wsse:UsernameToken/wsse:Username/@{any}
718 This is an extensibility mechanism to allow additional attributes, based on schemas, to be
719 added to the <wsse:Username> element.

792
793

While in row 791 and following it is stated:

791 /wsse:BinarySecurityToken/@{any}
792 This is an extensibility mechanism to allow additional attributes, based on schemas, to be
793 added.

797

In any case, the solution proposed by specifications is not sufficient because, even allowing the addition of an attribute, e.g. an “Original Requester” in the specific use case, such addition would not solve the issue because it would be anyway necessary to agree the schema (protocol) amongst all actors involved in the SOA infrastructure (provided by different vendors, etc.).

This would inevitably lead to the necessity of a high customization (and consequent expenditure) of the security models.

804 In order to avoid costly, non-standard, vendor/platform dependent customizations and ad-hoc
805 agreements, the operator considers that it is opportune to standardize such “protocol”.
806

807 6 Issues on Management

808 6.1 Introduction

809 The purpose of this section is to introduce to OASIS SOA-Tel TC requirements related to Service
810 Interface cardinality and definition of metadata for Service Lifecycle Management as they emerge from
811 the specification work in TeleManagement Forum Service Delivery Framework (SDF) program
812 (<http://www.tmforum.org/ServiceDeliveryFramework/4664/home.html>).

813

814 This section addresses:

- 815 • potential limitations in the OASIS specifications that have been considered when analyzing the
816 architectural patterns and possible implementations (such as SOA) for SDF's distributed capabilities,
817 specifically OASIS SOA-Reference Model [**SOA RM 1.0**] and SCA Assembly Model [**SCA Assembly**
818 **1.1**].
- 819 • potential updates to OASIS SOA Reference Architecture [**SOA RA 1.0**] as a result of the specification
820 work developed in TM Forum SDF team, specifically:
 - 821 - additional Service Management Interface,
 - 822 - additional metadata for the support of Service Lifecycle Management.

823 6.2 Scenario/context

824 The context from which this proposal originates is the modeling and specification activities that
825 TeleManagement Forum is performing in order to define a Service Delivery Framework. The results are
826 published in TM Forum's SDF Reference Model (TR139v2) and SDF Reference Architecture (TMF061)
827 documents, available to TM Forum's Members.

828

829 The TM Forum SDF objective is to manage end to end the lifecycle of services including cases where
830 services have dependencies they can not manage and cases where services are the result of dynamic
831 and static composition across service ownership/governance domains.

832

833 A Service Delivery Framework must respond to most actual management needs of Service Providers
834 while Services increasingly diversify:

- 835 • manage a Service the same way, whether it comes from network, web or IT resources,
- 836 • manage a Service the same way, whether it is retailed, wholesale or operated in-house,
- 837 • manage compositions of Services when each Service may be owned by separate entities
838 (organizations, Service or Content Providers), including the relationship that must exist among these
839 entities,
- 840 • manage multiple versions of a Service.

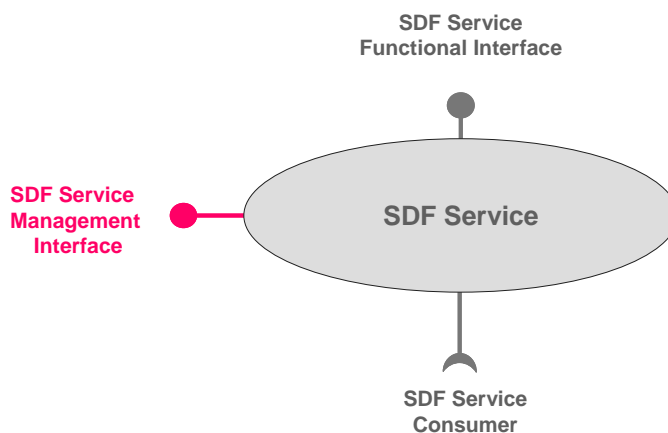
841 6.3 Services exposing Management Interface

842 The complexity of Service Providers business and operations requires a Service to be managed close to
843 the context in which it is used in order to understand who is using the service, eventually change service
844 parameters to adapt to its usage, measure in real-time the quality of each interaction with the service,
845 check on service status, etc.

846 A Service may have multiple capabilities, some of which may be used for functional purposes some for
847 management purposes, depending on the context in which the service is used.

848

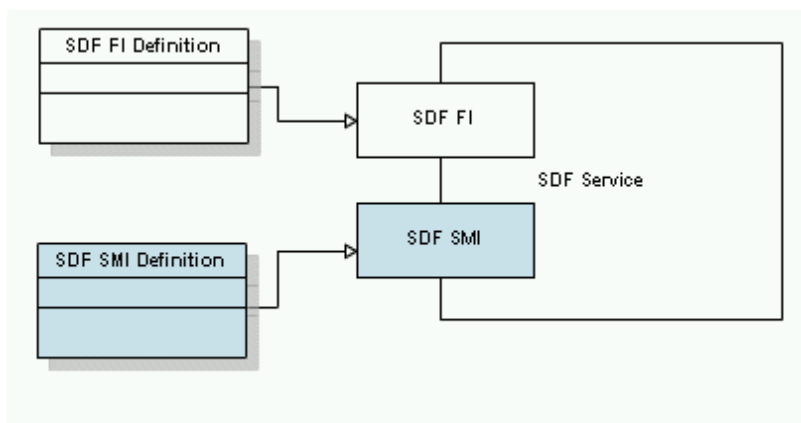
849 To fulfill TM Forum SDF's goal of E2E service lifecycle management, the TM Forum SDF team considers
 850 as Service model one where the Service exposes its manageability capabilities by means of a specific
 851 Interface, following the pattern in Figure 18.
 852



853
 854
 855
 856
 857
 858
 859
 860
 861
 862

Figure 18: TM Forum "SDF Service"

In this model, the SDF Service capabilities are exposed and consumed through the SDF Functional Interfaces (SDF FI) while the management capabilities/operations of the SDF Service are available through the SDF Service Management Interface (SMI). SDF Service may consume other Services through yet another, consumer type, interface (ref. Figure 19).



863
 864
 865
 866
 867
 868
 869
 870
 871
 872

Figure 19: Including management capabilities definition in the SDF Service description

The reasons for the separation and exposure of manageability capabilities at another interface (SMI) are:

- Management capabilities are consumed by other type of (specialized) consumers (e.g. support services) with different policy/security rules than consumers of functional capabilities
- Some higher level operations and business around services can be simplified by ignoring "layers/levels" at which functional capabilities of services may be embedded, and access directly their management capabilities.

873 6.3.1 Perceived Technical Issues

874 The OASIS documentation defines Services in SOA RM and Service Components in SCA as if the
875 cardinality of Service Interface is 1 and only one.

876 -----

877 **[SOA-RM 1.0]:** (Section 3.1) “A service is accessed by means of a service interface (see Section
878 3.3.1.4), where the interface comprises the specifics of how to access the underlying capabilities.”

879 **[SOA-RM 1.0]:** (Subsection 3.3.1.4) “The service interface is the means for interacting with a
880 service.”

881 **[SCA Assembly 1.1]:** “A Service represents an addressable interface of the implementation.”

882 Note – SCA definition for Service may be a consequence of the SOA-RM definition, we do not
883 know

884 -----

885 Moreover, for those implementers who use WSDL to describe services, the W3C **[WSDL 2.0]** primer
886 document, (section 5.4) states that, “wsdl:service specifies only one wsdl:interface ()”.

887 We are aware of the solutions presented by W3C but these solutions are not standardized.

888

889 Following these documents it seems to be impossible to have two or more interfaces for a SOA Service.
890 At the same time, SOA RA document acknowledges that “In fact, managing a service has quite a few
891 similarities to using a service” hinting that a management of a service should happen at an interface. The
892 same document offers though another solution (separation between management services and non-
893 management services) which we will discuss in the next use case.

894 -----

895 **[SOA-RA 1.0]** (3137 – 3140) “In fact, managing a service has quite a few similarities to using a
896 service: suggesting that we can use the service oriented model to manage SOA-based systems
897 as well as provide them. A management service would be distinguished from a non-management
898 service more by the nature of the capabilities involved (i.e., capabilities that relate to managing
899 services) than by any intrinsic difference. “

900 -----

901 Today many management capabilities are bundled with the functional interface of the service description
902 which makes management of services very hard. This situation poses a problem for suppliers who would
903 like to follow a SOA path for their SDF solutions. For example,

- 904 • how can they take already existing SOA Services and make them SDF Services?
- 905 • Can a SOA Service work with a Management Interface and a Functional Interface?

906 In TM Forum, the MTOSI team created multiple (coarse and fine grain) web services as alternative to
907 multiple interfaces (<http://www.tmforum.org/BestPracticesStandards/mTOPMTOSI/2319/Home.html>).

908 There is a need to specify that all these WS-es are related (e.g. allow access and interaction with the
909 same Inventory and its elements).

910 TM Forum SDF team is seeking reconciliation on this matter and asks about possibilities to express the
911 SDF Service and its SMI using SOA Service model.

912 TM Forum SDF team is also seeking alignment of its SMI addition to a Service model with the work
913 developed in OASIS WSDM – MOWs.

914 6.4 Metadata in support of Service Lifecycle Management

915 In TM Forum’s SDF Reference Model (ref. Figure 20) (ref. TM Forum TR 139 v 2) the lifecycle
916 management of an SDF Service is supported by other services created to fulfill the needs of business and
917 operational processes.

918

919

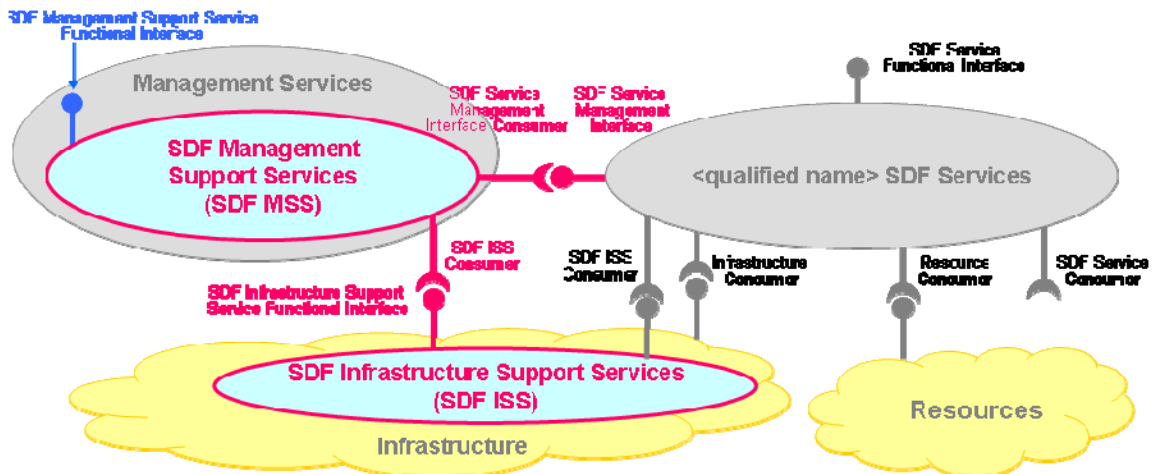


Figure 20: SDF Reference Model

920
921
922
923
924
925
926
927
928
929
930
931

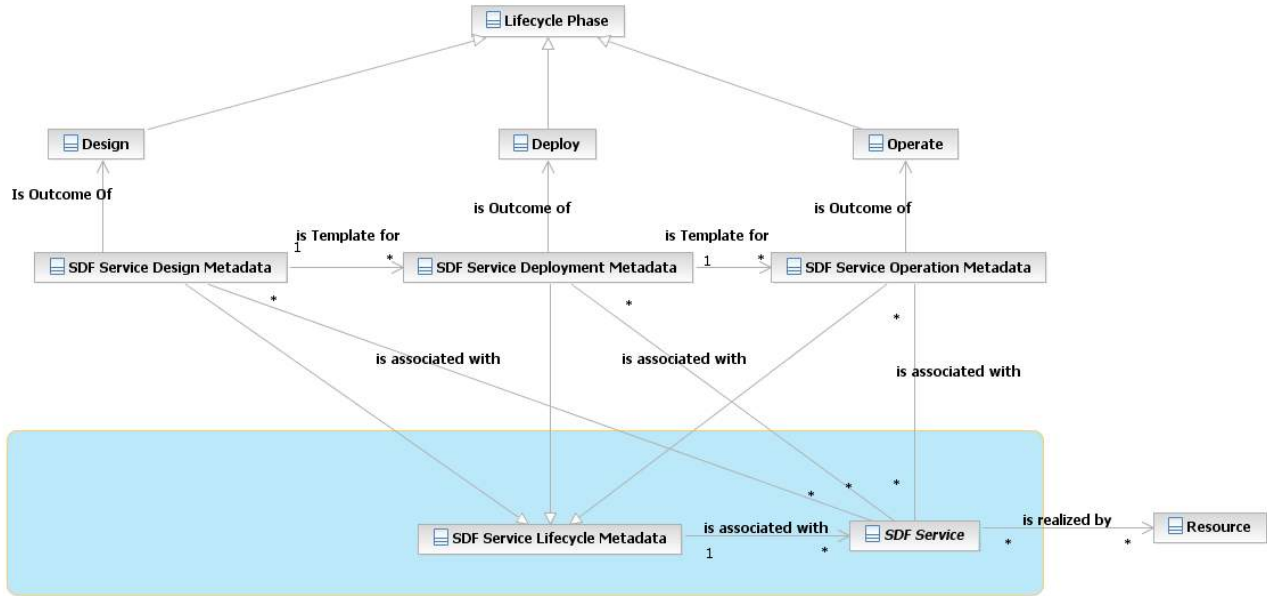
- **SDF Management Support Service (SDF MSS):** An SDF Management Support Service (SDF MSS) consumes the SDF SMI of a SDF Service to manage the SDF Service. Examples of SDF MSS-es are Activation/Configuration, Problem management, Service Quality Management.
- **SDF Infrastructure Support Service (SDF ISS):** An SDF ISS provides reusable functionalities, exposed via functional interface(s), to support the SDF. Examples of possible SDF ISS are: Catalogues, Metadata repository, User Profile.

932 In agreement with the OASIS [SOA RA 1.0] (3137 – 3140) paragraph mentioned in section 6.3.1, SDF
933 RM shows that these supporting services are of the same nature as the SDF Service itself, the only
934 difference is that they “manage” or help in managing the SDF service (e.g. helping is the role of ISS
935 Services). But these services need to be managed at their turn. For this reason, SDF Support Services
936 follow the same pattern as the SDF Service: they have both a **functional and a management interface**.

937 Specialization in supporting and managing a service during its whole lifecycle requires finer granularity
938 knowledge about that service: properties, supported actions or operations, possible states as well as
939 contracts that may govern interactions with the service (including pre and post conditions for these
940 interactions), what is the “architectural” style for service “composability”, what are its dependencies or
941 what is the level of exposure for its functional capabilities.

942 The proposed model for the TMF SDF SDF Service is complemented by additional data representation
943 (metadata) in support of SDF Service lifecycle management (ref. Figure 21 and Figure 22). This new data
944 representation containing information about the service in various phases of its lifecycle, aims at covering
945 current gaps in the information available for the purpose of service management (e.g. what is already
946 covered by the SOA Service description) in the overall context of Service Provider’s business and
947 operations. Moreover, this metadata is dynamic: it may change from one phase to another of the SDF
948 Service lifecycle.

949

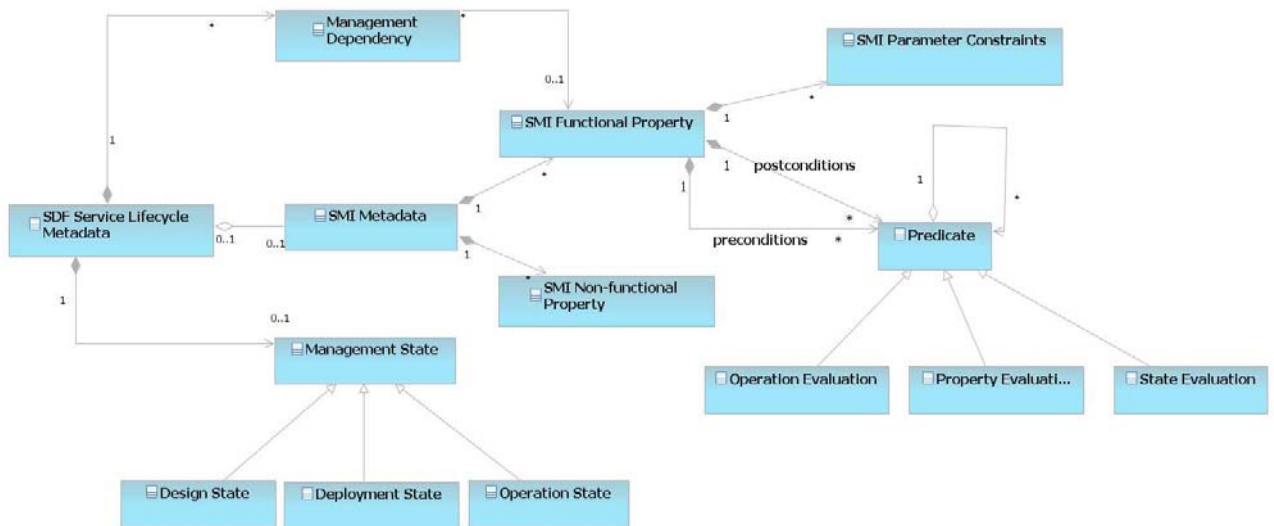


950
951
952
953
954
955
956
957

Figure 21: SDF Service lifecycle phases and associated metadata

The SDF Service Lifecycle Metadata consists at least of:

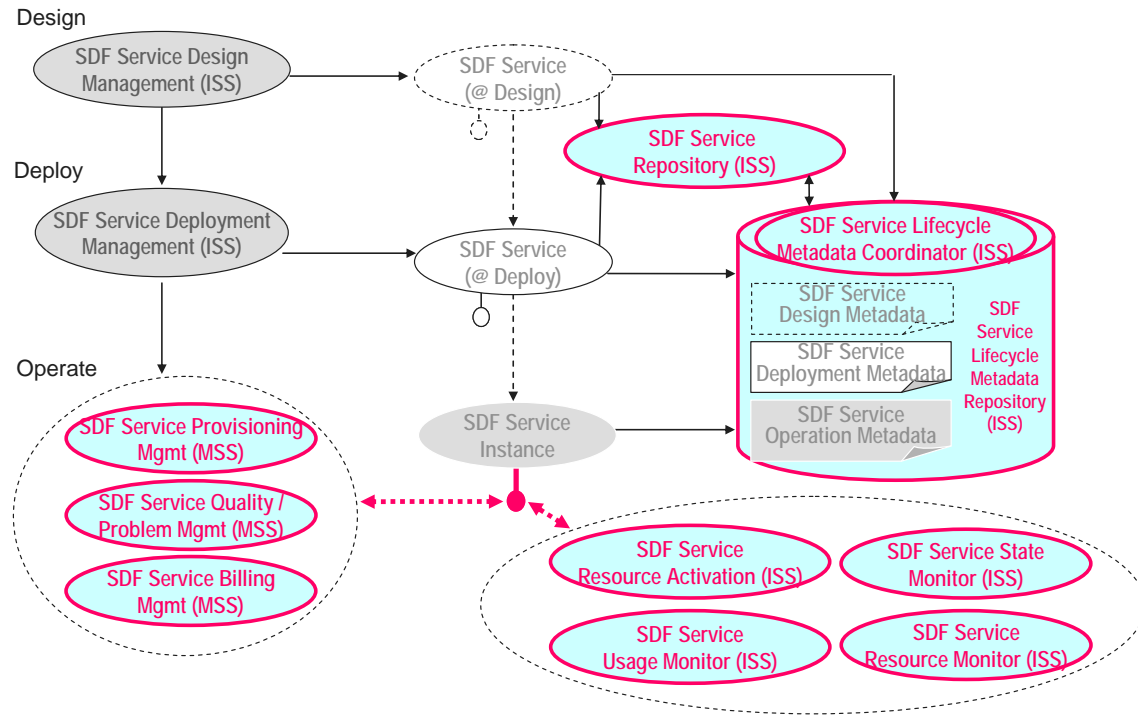
1. **Additional information about the SMI of a SDF Service** (properties, actions);
2. **Management Dependencies of the SDF Service**, including cross-domains dependencies;
3. **Management State** of the SDF Service.



958
959
960
961
962
963
964

Figure 22: SDF Service Metadata (concepts)

The way this metadata is used by SDF Supporting Services to manage an SDF Service during its lifecycle is depicted below (ref. Figure 23).



965
966
967
968

Figure 23: Service Lifecycle Management through SDF

969 6.4.1 Perceived Technical issues

970 The purpose of TM Forum work is not to duplicate existing work but to add to it that part that is necessary
971 for service lifecycle management. The information representation (metadata) that TM Forum SDF team
972 has identified as necessary for SDF Service Lifecycle Management, as well as its evolving nature, do not
973 seem to be modeled in the current SOA Service Description Model and supported by the Management of
974 Services approach described in [SOA –RA 1.0] document. TM Forum SDF Team believes that modeling
975 service dependencies including dependencies across ownership/governance domains is important
976 addition to the SOA RA.

977 TM Forum SDF team is seeking OASIS expert advice on what to do. Can the additional metadata it
978 specifies for the purpose of SDF Service lifecycle management be added to the current [SOA RA 1.0], in
979 respect to the views and the models that are already part of this Reference Architecture?

980 TM Forum SDF team is also seeking OASIS expert advice on aspects such as supporting versioning and
981 compatibility of this metadata, existing architectural patterns for data contribution from various
982 applications/sources/systems and for assurance of cohesiveness across metadata elements and along
983 the phases in the lifecycle of a service.

984 6.5 Recap of issues and considerations for OASIS SOA-Tel analysis

985 TM Forum SDF team is seeking reconciliation on the matter of the additional service management
986 interface and asks about possibilities to express the SDF Service and its Service Management Interface
987 (SMI) in the SOA Service model. TM Forum SDF Team believes that distinguishing the SMI from the
988 Functional Interface of a Service is necessary for the reasons exposed in the use case.

989 What is OASIS's advice on this and how can SDF Service model be realized with current SOA Services
990 Model?

991

992 TM Forum SDF team is also seeking OASIS expert advice on positioning of its SMI addition to a Service
993 model within the work developed in OASIS WSDM – MOWs.

994 TM Forum SDF team is also seeking OASIS expert advice on what should be the relationship between
995 the SDF Reference Model and the SOA Reference Architecture - Service as Managed Entities part.

996 TM Forum SDF team is seeking OASIS expert advice on how to organize and integrate the additional
997 metadata for the purpose of SDF Service lifecycle management in the current [**SOA RA 1.0**] and do so
998 with respect to the views and the models which are already part of this RA.

999 TM Forum SDF team is also seeking OASIS expert advice on aspects such as supporting versioning and
1000 compatibility of metadata, existing architectural patterns for data contribution from various
1001 applications/sources/systems and for assurance of cohesiveness across metadata elements and along
1002 the phases in the lifecycle of a service.

1003

1004 7 Issues on SOA collective standards usage

1005 7.1 Common Patterns for Interoperable Service Based 1006 Communications

1007 7.1.1 Scenario/purpose

1008 There is a growing set of application models that serve a general web and mobile market and
1009 consequently can only expect a web application pattern and can not make any assumptions of the
1010 protocol stack other than IP. These applications are no longer exclusive to the public domain.
1011 Applications in the enterprise are adopting these new computing models, seamlessly moving between
1012 internal and external clouds trying to leverage the elasticity that the model offers and blending application
1013 oriented communications across these boundaries. Such applications are typically designed to support
1014 highly functional virtual and often transient partner/ end user/ customer relationships.

1015 Users in these models expect access to information anytime, anywhere and will expect the enablement of
1016 communications within that context of any application to be delivered in the same way. Ubiquity of
1017 communications as a part of this set of internet type applications, LAN attached or mobile, needs to allow
1018 for interoperation across a definable set of standards and device types in order for it to achieve the same
1019 universality as the supporting application models, bringing seamless communications utility across
1020 different communication domains and applications.

1021 In such models, the application can only make general assumption about the device attributes and
1022 protocol stacks these devices support. Ubiquity of communication within the application model calls for
1023 device information and communications channel setup to be ascertained thru the process of user/ device
1024 connecting to the application. In some situations the application may not be directly involved in setting up
1025 media, in other cases it will either need to participate, at least in part or entirely. An application may even
1026 have to make decisions as to the best choice of path of delivery.

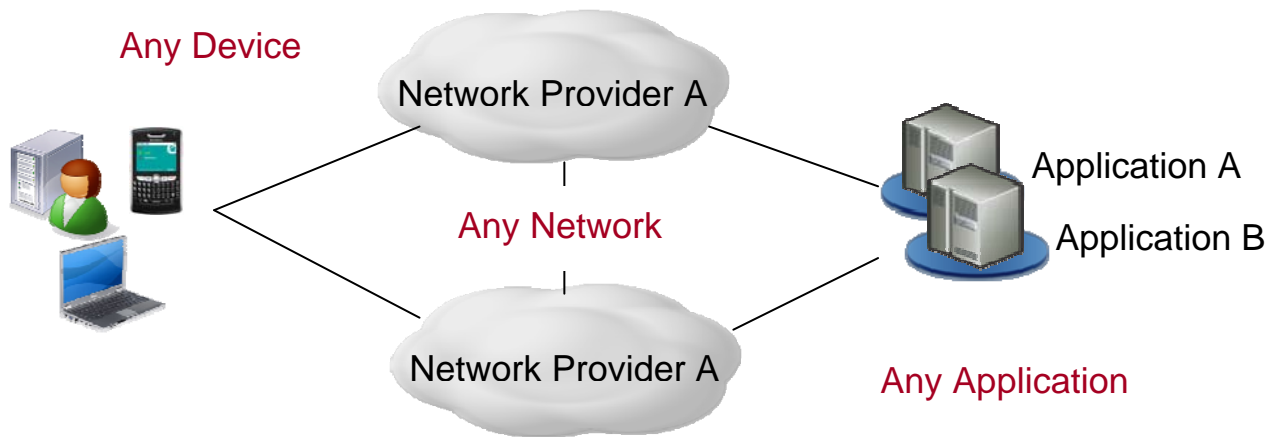
1027 Achieving ubiquitous access to application resources irrespective of network domain is often a function a
1028 combined collection of standards working in unison (i.e. profile) providing consistent patterns to access
1029 applications resources. Consistency in approach across different media and control paths, client types
1030 and application domains is essential to foster larger a eco-system of co-operative applications for the user
1031 across different network and application domains. Hence, the patterns supporting the discovery, setup
1032 and delivery of communications within the context of a set of applications needs to be normalized in order
1033 to enable interoperable solutions across heterogeneous environments.

1034

1035 Enclosed is an example:

- 1036 ○ An Independent collision appraisal company has independent collision agents that broker across
1037 separate suppliers on behalf of many insurance companies, auto suppliers and collision repair
1038 shops. The agents choose which suppliers to use based on their locale and relationships but
1039 these are under a lot of change.
 - 1040 ○ No one company owns and controls the type of agent device.
 - 1041 ○ Agents typically search a few supplier sites for any given situation. They expect to be
1042 able to quickly call and have the context of the part/order be available to any parts
1043 supplier, insurance company and collision shop they use. The agent may further use
1044 media (picture, video) to support and verify the parts needed with the supplier.
 - 1045 ○ The applications from different companies support different service profiles (voice, video,
1046 picture, and data) to deliver the capability. Real Time communications is supported thru
1047 variable means including but not limited to, SIP, Jingle or simply an RTP stream
1048 controlled directly by the application.
 - 1049 ○ A standard means application communications profile needs to be delivered in order to
1050 allow any agent and device to work in the context of a set of independent applications
1051 from different suppliers

1052 The market in general needs a normalized means to establish communications to the endpoint without
 1053 being prescriptive at the endpoint. Applications need greater control over the different choices to be made
 1054 given multiple network paths and options. An application requesting a connection should be able to adapt
 1055 seamlessly to the network environment and protocols used to set up the communications channels. In
 1056 addition, external tools such as BPEL, BPM and ESB should be able to leverage this common foundation
 1057 to incorporate communications processing. This is important for broader adoption of communication as a
 1058 service using well known patterns and skills. Figure 24 depicts the case.
 1059
 1060



1061
 1062
 1063 Figure 24: Real-time communications in the context of an “any” application seamlessly across any device
 1064 and network
 1065

1066 The following is a minimum set of requirements:

- 1067
- 1068 1. **Universal service discovery/ dynamic bindings**
 - 1069 2. **Bi-directional, full duplex control across different modes of communication thru web**
 - 1070 3. **Common support for asynchronous interactions with event subscriptions and**
 - 1071 4. **Means to associate application context with stateful communication interactions (i.e.**
 - 1072 5. **Common communication information model enabling connection negotiation.**
 - 1073 6. **Common patterns for client web services to work within a SIP and XMPP context.**
 - 1074 ○ **Integrated control of media delivery (transport channels and their parameters)**
 - 1075 ○ **Control of communications channel, events for that session**

1076
 1077
 1078
 1079
 1080 Items 1, 2, 3 and 4 above target a common set of web service infrastructure requirements to generically
 1081 set up communications. Items 5 and 6 are essential to handle differences (e.g., between a SIP or Jingle,
 1082 etc based endpoints) thru the service interface.

1083 7.1.2 Scenario/context

1084 This use case involves a simple web application that connects to the site, pulls down a list of people to
 1085 contact and allows the user to click-to-call. Assume a simple model where JavaScript is downloaded to
 1086 the client and sets up the web service call to a communication service with the URI provided. The
 1087 sequence diagram in Figure 25 depicts the case.

1088 The use case defines a simple setup of a voice connection for one side of the connection. More complex
1089 types of communication scenarios (e.g. conferencing, video) and multi-modal interactions (e.g. voice with
1090 chat sessions) should be supported with the same pattern. All applications need a common means to set
1091 up different ports supporting different types (voice, pictures) or multiplex thru one port but can not assume
1092 one standard or protocol stack is at play as they do not know who and what type of device is going to
1093 connect. A server based model implies that communications is handled at the server (i.e. server connects
1094 client A to client B) where as the client model is more p2p. Each mode must be generally supported by
1095 the pattern.

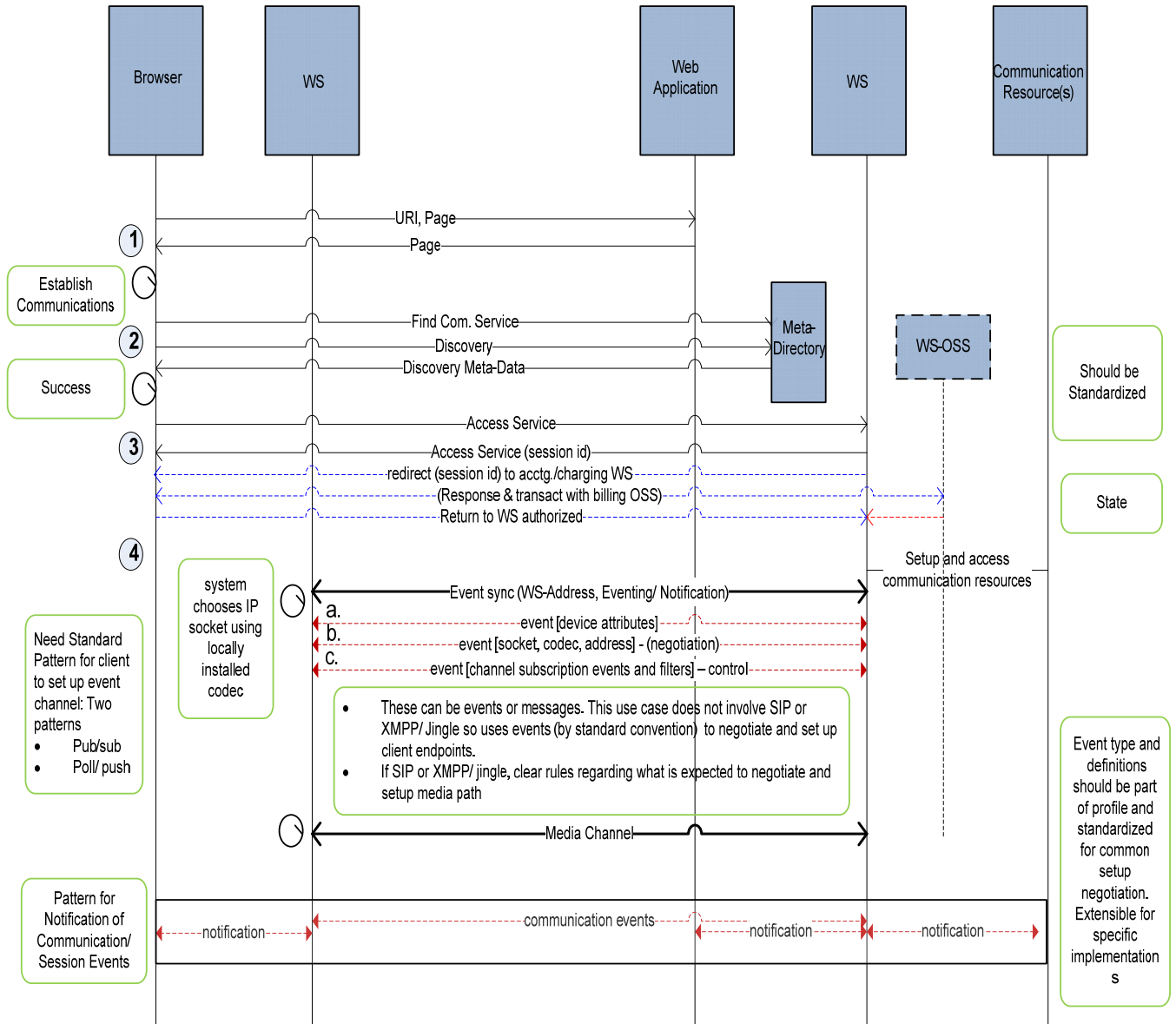
1096 The pattern discussed in this use case can equally be applied to REST type models using Restful API
1097 mechanisms. This use case will confine itself to a web services client/ interaction model. It is important to
1098 understand that whichever programming model used for the application, for generally application
1099 interoperability across domain, the application model for communications needs to be consistent. Lastly,
1100 some of the interface discovery complexity could be handled thru a commonly defined interface used
1101 across vendors. Lack of such an agreed upon model, places more complexity in the meta-data needed to
1102 describe what services handle what type of communications (i.e. voice or video connection, conference,
1103 etc.) and more importantly describing the events types and data structures across the wire. This use
1104 case does not go into detail the interactions for device attribute and/or interface discovery.

1105 **The basic interaction in this use case involves a web service interchange enabling the setup of a**
1106 **communications channel exclusively. In this case we are selecting a communication channel that**
1107 **is a proprietary RTP enabled socket controlled by the application. Hence, events need to be**
1108 **exchanged to inform, negotiate and select the address on each side, the real time protocol used,**
1109 **the codec and other pertinent information. The same negotiation process can be used to select a**
1110 **SIP or XMPP/ Jingle based media channel when device attributes and condition warrant. In this**
1111 **latter case, these protocols would negotiate the information on their own, freeing the service itself**
1112 **from this activity.**

1113 Looking at this pattern we see that the set of requirements for the web services infrastructure (i.e.
1114 standards) within the context of communications is clarified. We need a standard means to establish a
1115 multimedia channel supporting real-time voice and video exclusively thru the web but also allow for
1116 variation to support other approaches. This allows a higher degree of inter-operability across different
1117 business and network domains. The standard pattern promotes common skills, behavior and tool
1118 integration. It fosters development consistency, simplicity driving wider adoption and most important,
1119 allows providers to offer solutions that work in the context of an inter-operable cloud.

1120

1121 Use Case Sequence Diagram:
1122



1123
1124

1125 Figure 25: Sequence diagram example for the Universal Communication Profile case

1126

- 1127 **Use Case Steps:**
- 1128 1. The communication responds back with a session id for the context of the application within a
1129 communication channel.
- 1130 2. A bi-directional web services interface is set up to receive events for this session id.
- 1131 a. Client looks up service meta-data and discovers interface, binding, events and capabilities of
1132 service. (i.e. WS- meta data and WS-policy)¹.
- 1133 b. If there is no clear interface specification (i.e. CSTA, Parlay-x, other) then a very robust meta-
1134 directory and policy infrastructure is needed to support the interface variations across
1135 vendors.
- 1136 c. Connection is attempted. This may trigger events such as subscription authorization or pay-
1137 as-you-go. This results in redirecting to a billing-OSS WS that engages the client over the
1138 event-channel for payment methods and payment completion – leading to a notification and
1139 return to the service-WS for further service delivery/denial².
- 1140 3. Client connect to WS
- 1141 a. Event channel is set up.
- 1142 b. This event channel is overlaid with a subscription interface allowing each side to subscribe
1143 and filter as necessary specific events needed for the communications.
- 1144 i. Model needs to support timely and reliable delivery of events
- 1145 ii. Model needs to support events delivered in specific order
- 1146 4. Client sends event indicating its device characteristics, communication modes (SIP, Jingle, etc.)³.
- 1147 a. Connection is made using “proprietary” socket. Application has designed the separation of
1148 different types (i.e. picture, video, voice) and it manages the parsing and reformatting of each
1149 for the application.
- 1150 i. User is in voice session
- 1151 ii. User is in transmitting pictures
- 1152 b. Server sends event indicating the mode it wishes to use given the device attributes.
- 1153 i. If SIP or XMPP/ Jingle client, negotiation of codec and address via those standards
1154 but information (i.e. session description) is delivered to client application thru the web
1155 service. The application sets up and controls the media, creates SDP response and
1156 defines RTP port
- 1157 c. In this simple case we are using RTP with session description/ negotiation being handled thru
1158 WS event channel.
- 1159 d. Client sends event to WS indicating what connection processing events it is interested in. In
1160 this case it asks for connection, disconnect, hold/resume for picture and mute/un-mute for
1161 events.
- 1162 e. Remote user presses hold for picture. Event is propagated to device and picture transmission
1163 is held
- 1164

¹ Note: IETF work and SIP media and session policies stds (xml-based; can be realized as derived schema of the ws-policy core). Same goes for security policy (though ws-security-policy as it is restricted to only policies for ws-security standards.).

² This step is but an example interaction of several possible generic pre-communication events. In-communication and post-communication events are also conceivable.

³ Note: Any WS-standards here or is it an area that the SOA-TEL TC can develop schema for?

1165 Since service architectures are inherently transport neutral, we can not rely on any underlying means (i.e.
1166 TCP) to manage the session lifecycle. We do not imply any particular means in this example to establish
1167 statefulness at either point across the wire, just a means to set up and convey the information across any
1168 channel.

1169 It is our intention to first look to see if this is a common pattern across all communications services and to
1170 identify the relevant standards that can be used and/or need to extend to support the need. Once
1171 explored for web services we can extrapolate this to a common set of patterns for a broader set of service
1172 interface types.

1173 **7.1.3 Technical Issues/ Solutions:**

1174 The purpose of the above uses case is not to prescribe a solution but what a solution may need to look
1175 like in the context of the problem. The problem is basically that in order to deliver ubiquitous mobility and
1176 interoperability to users, applications can not be bound by a single network provider nor underlying
1177 assumptions on the real-time protocols used. Access to real-time communications needs to be
1178 normalized across set of common access patterns in the context of any given application. The process is
1179 not disjoint; application and communications need to work in context to deliver full effectiveness. Access
1180 to the application resource requires the discovery the right pattern without any pre-defined assumptions
1181 about the underlying network. The application also needs to be able to make decisions as to the best path
1182 in multiple paths exist based on policy, cost, quality and device attributes.

1183 Service orient architectures are in principle about decoupling the underlying transport form the delivery of
1184 the application resource. This principle needs to be hold for access to applications / services and real
1185 time communications used in the context of any application allowing for common access across a broad
1186 set of applications.

1187 **8 Conformance**

1188 The objective of this document is to collect potential technical issues and gaps of SOA standards utilized
1189 within the context of communications service providers, in order to enable subsequent development of
1190 requirements for the solution of such issues.

1191 As such no conformance clauses apply to this document.

1192 **Appendix A. Acknowledgements**

1193 The following individuals have participated in the creation of this specification and are gratefully
1194 acknowledged:

1195

1196 **Participants:**

1197

1198	Mike Giordano	Avaya
1199	Liu Feng	Avaya
1200	Mahalingam Mani	Avaya
1201	Ian Jones	BT
1202	Sami Bhiri	Digital Enterprise Research Institute (DERI)
1203	Paul Knight	Individual
1204	Lucia Gradinariu	LGG Solutions
1205	Orit Levin	Microsoft
1206	Joerg.Abendroth	Nokia Siemens Networks
1207	Christian Guenter	Nokia Siemens Networks
1208	Thinn Nguyenphu	Nokia Siemens Networks
1209	Olaf Renner	Nokia Siemens Networks
1210	Abbie Barbir	Nortel
1211	John Storrie,	Individual
1212	Vincenzo Amorino	Telecom Italia
1213	Luca Galeani	Telecom Italia
1214	Maria Jose Mollo	Telecom Italia
1215	Vito Pistillo	Telecom Italia
1216	Enrico Ronco	Telecom Italia
1217	Federico Rossini	Telecom Italia
1218	Luca Viale	Telecom Italia

Appendix C. Possible workaround related to issue in Section 3.1 “Transaction Endpoints Specification”

1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272

This section is non-normative.

This issue described within Section 3.1 could be solved with the following “workaround” solution, which in any case is not mandatory but exploits some “optional” features of WS-Addressing.

Note:

- This proposal does not require any “persistence” on any intermediary and is fully compliant with WS-Addressing specification.
- The TC asks if, apart from the proposed workaround, there is another standard reference solution for the highlighted problem.

Should there be no other solution apart from the proposed workaround; **the proposal is to extend the WS-Addressing specification in order that the “Message Properties” include a new tag (provisionally named “Final Destination”) to specify the process/transaction result.**

Moreover the proposal is to make the utilization of this new tag as Mandatory whenever it is necessary to specify a “final destination”, i.e. in presence of a non-direct “requester-consumer” situation.

Proposed Workaround:

CASE A:

1. **C1 invokes WS-A** and specifies in the *replyTo* section of the WS-Addressing header the *EPR (Endpoint Reference)* where it wants to receive the asynchronous response (**C1**). (Example: <http://service1.sc.local/response>).
2. The **ESB invokes WSB** and specifies in the *replyTo* section of the WS-Addressing header the *EPR (Endpoint Reference)* where it wants to receive the asynchronous response (Example: <http://service1.esb.local/response>). By doing so it takes the *replyTo* section received by C1 and embeds it in the *referenceParameters* section of *replyTo*. P1 is obliged by WS-Addressing specification to return the *referenceParameters* in the *To* section when sending the asynchronous response.
3. **P1 returns the asynchronous response** to the *replyTo* address (Example: <http://service1.esb.local/response>) specified by the ESB, together with the *referenceParameters* section.
4. The **ESB invokes WSC** and specifies in the *replyTo* section of the WS-Addressing header the *EPR (Endpoint Reference)* where it wants to receive the asynchronous response (Example: <http://service2.esb.local/response>). By doing so it takes the *referenceParameters* section received by WSB and embeds it in the *replyTo* section. P2 is obliged by WS-Addressing specification to return the *referenceParameters* in the *To* section when sending the asynchronous response.

1273

1274 5. **P2 returns the asynchronous response** to the ESB *replyTo* address (Example:
1275 <http://service2.esb.local/response>) specified by the ESB, which includes the *referenceParameters*
1276 section.

1277

1278 6. **The ESB gets the *replyTo* info**, embedded in the *referenceParameters* received from P2, to
1279 address the asynchronous response to **C1**.

1280

1281 **CASE B:**

1282 Same as Case 1 with C2 originator and final destination.