Web-services policy language use-cases and requirements

Working draft 01, 7 March 2003

Document identifier: wd-xacml-wspl-use-cases-01.doc

Location: http://www.oasis-open.org/committees/xacml/docs/

Send comments to: xacml-comment@lists.oasis-open.org

Editors:

Tim Moses, Entrust (tim.moses@entrust.com)

Contributors:

Anne Anderson, Sun Microsystems
Simon Godik, Overxeer

Abstract:

This working draft defines use-cases for negotiating a variety of forms of policy in the Web-services architecture. Its purpose is to identify the policy requirements of the Web-services application domain and the shortcomings of XACML when applied to that domain.

Status:

This version of the specification is a working draft of the committee. As such, it is expected to change prior to adoption as an OASIS standard.

If you are on the xacml@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the xacml-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to xacml-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Table of contents

1. Introduction

XACML is potentially well suited to serve the policy needs of the Web-services application domain. This document explores the requirements for XACML when used in that domain, in order to identify XACML's shortcomings.

Several aspects of policy are considered, including: cryptographic-security policy, authentication policy, authorization policy, privacy policy, reliable-messaging policy and transaction policy.

2. Use-cases

2.1. Use-case 1: Submit request

Use-case 1 is shown in Figure 1. In this case, Consumer submits a service request to Provider. If the service request conforms with Provider's policy for requests, then Provider accepts the request. Otherwise, it returns a fault status. Optionally, in the fault case, it returns its policy for requests of the type.

This use-case applies to situations in which Provider imposes requirements on the form of acceptable service requests and/or is willing to accept service requests of a certain form. This situation exists, for instance, where Provider requires Consumer to authenticate itself, or allows Consumer to confidentiality-protect submitted data.

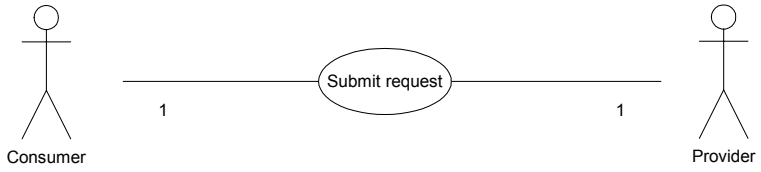
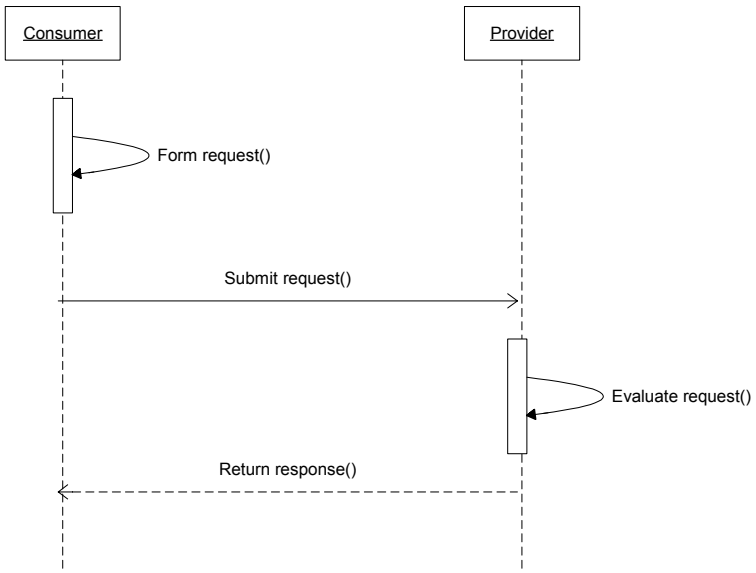


Figure 1 - Use-case 1

79 The corresponding sequence diagram is shown in Figure 2.



80

81 **Figure 2 - Use-case 1 sequence**

82 1. Consumer forms a service request in compliance with its own policy for the request type.

83 2. Consumer sends the request to Provider.

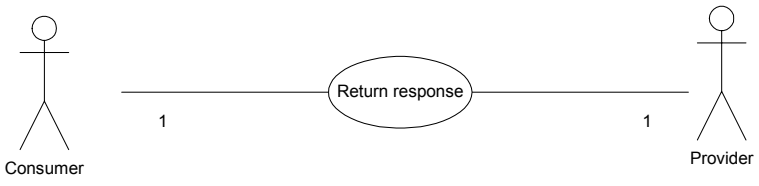
84 3. Provider tests the request against its policy for the request type.

85 4. If the request satisfies Provider's policy, then Provider accepts the request and (optionally)
86 returns a response. If the request does not satisfy Provider's policy, then Provider returns a
87 fault status and, optionally, its policy for requests of the type.

88 2.2. Use-case 2: Return response

89 Use-case 2 is shown in Figure 3. In this case, Provider returns a service response to Consumer. If
90 the service response conforms with Consumer's policy for responses, then it accepts the response.
91 Otherwise, it discards the response.

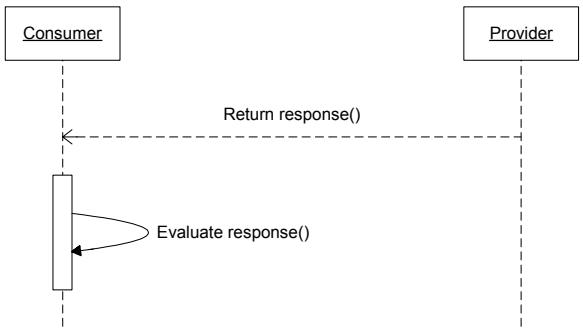
92 This use-case applies to situations in which Consumer imposes requirements on the form of
93 acceptable service responses and/or is willing to accept service responses of a certain form. This
94 situation exists, for instance, where Consumer requires Provider to commit to certain of the
95 contents of the response by signing them.



96

97 **Figure 3 - Use-case 2**

98 The corresponding sequence diagram is shown in Figure 4.



99

100 **Figure 4 - Use-case 2 sequence**

101 1. Provider returns a response.

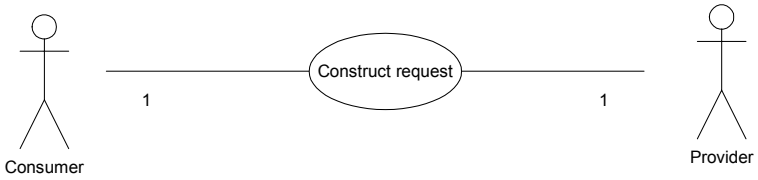
102 2. Consumer tests the response against its policy for responses of the type. If the response
103 satisfies its policy, then it accepts the response. Otherwise, Consumer discards the response.

104 2.3. Use-case 3: Construct request

105 Use-case 3 is shown in Figure 5. In this case, Consumer forms a request that it knows will be
106 accepted by Provider because it conforms with Provider's policy for requests of the type.

107 This use-case applies to situations in which Consumer cannot form an acceptable service request
108 by trial and error. Rather it must form a service request that it can be certain is acceptable to
109 Provider. Therefore, Provider describes in its policy the functions that it insists on performing and
110 the functions that it is willing and able to perform. There may be differential costs associated with
111 alternative functions. Therefore, Provider may wish to indicate which of the alternative functions it
112 prefers to perform.

113 This situation exists, for instance, where Consumer's policy requires that certain contents be
114 encrypted, while Provider's policy requires that certain other contents be "in the clear". Consumer
115 is able to form a request in which information that is required to be encrypted is encrypted, and
116 information that is required to be "in the clear" is "in the clear".



117

118 **Figure 5 - Use-case 3**

The corresponding sequence diagram is shown in Figure 6.

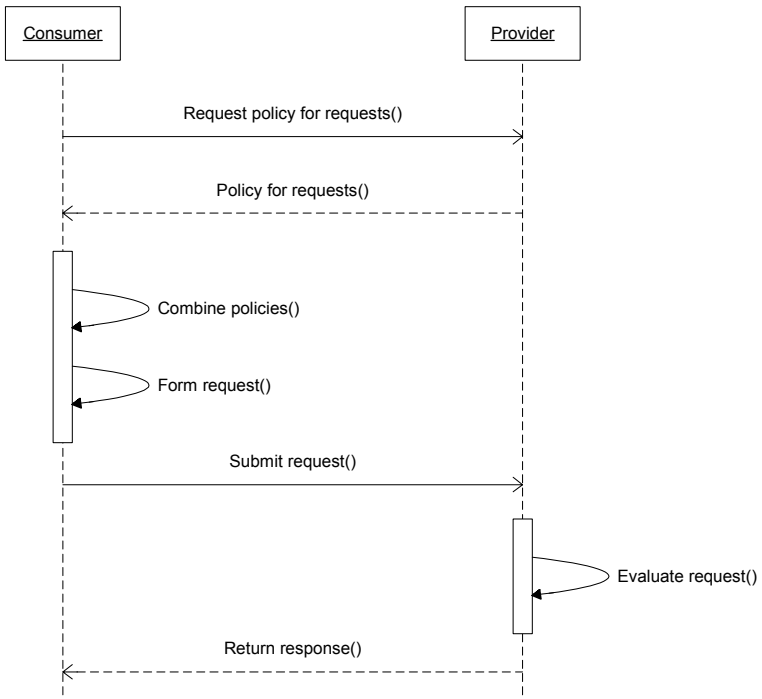


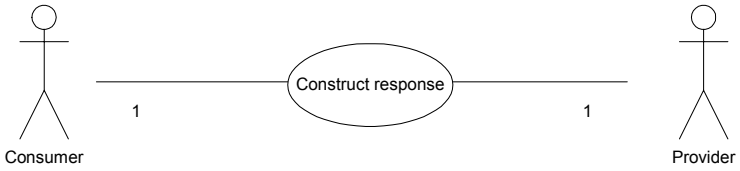
Figure 6 - Use-case 3 sequence

1. Consumer requests Provider's policy for requests.
2. Consumer obtains Provider's policy for requests.
3. Consumer combines Provider's policy for requests with its own.
4. Consumer forms the request in conformance with the combined policy for requests.
5. Consumer sends the request for service to Provider.
6. Provider verifies that the request satisfies its policy for requests.
7. If it does, then it accepts the request and (optionally) returns a response. Otherwise, it returns a fault status.

2.4. Use-case 4: Construct response

Use-case 4 is shown in Figure 7. In this case, Provider forms a response that it knows will be accepted by Consumer, because it conforms with Consumer's policy for responses.

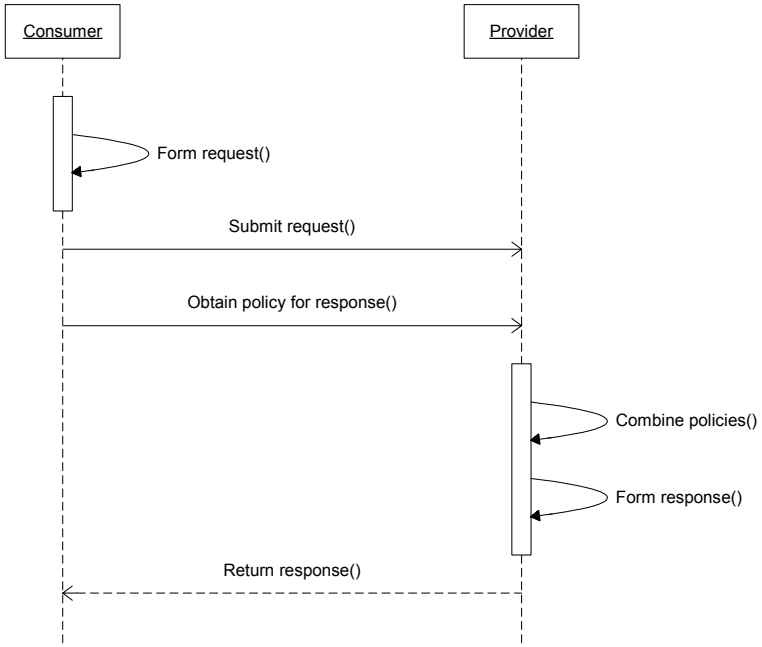
This use-case applies to situations in which Provider cannot form an acceptable response by trial and error. Rather it must form a service response that it can be certain is acceptable to Consumer. This situation exists, for instance, where Provider's policy requires that ceratin contents be encrypted, while Consumer's policy requires that certain other contents be "in the clear". Provider is able to form a response in which information that is required to be encrypted is encrypted, and information that is required to be "in the clear" is "in the clear".



139

140 **Figure 7 - Use-case 4**

141 The corresponding sequence diagram is shown in Figure 8.



142

143 **Figure 8 - Use-case 4 sequence**

144 1. Consumer forms the request.

145 2. Consumer sends the request for service to Provider.

146 3. Provider obtains Consumer's policy for responses.

147 4. Provider combines Consumer's policy for responses with its own.

148 5. Provider forms a response in conformance with the combined policy for responses.

149 6. Provider returns the response to Consumer.

150 2.5. Use-case 5: Disclose confidential data

151 Use-case 5 is shown in Figure 9. In this case, the Provider discloses to Recipient data provided to
152 it by Consumer, in conformance with its own and Consumer's policy for disclosure.

This use-case applies when Consumer provides confidential information, including (but not limited to) personal information, and Provider has to pass certain parts of the confidential information to another entity, not governed by Provider.

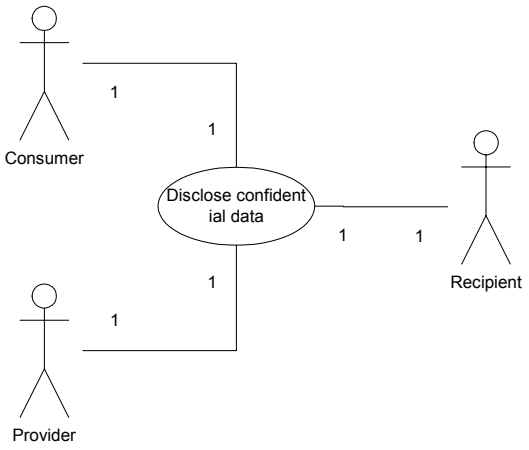


Figure 9 - Use-case 5

The corresponding sequence diagram is shown in Figure 10.

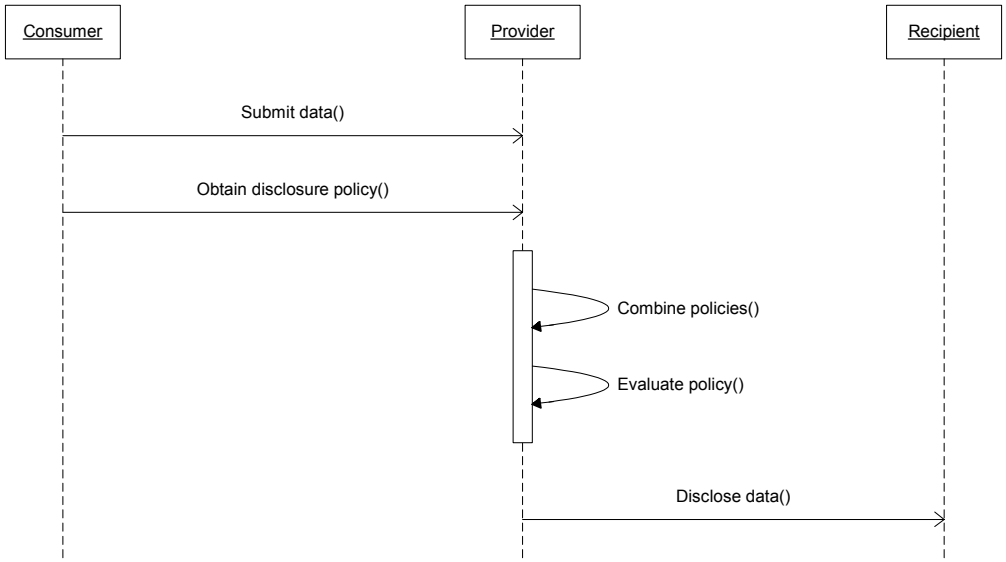


Figure 10 - Use-case 5 sequence

1. Consumer submits data to Provider.
2. Provider obtains Consumer's policy for disclosure.
3. Provider combines Consumer's policy for disclosure with its own.
4. Provider evaluates its own and Consumer's policy for disclosure.
5. If the policy is satisfied, then Provider discloses the data to Recipient. Otherwise, it does not.

2.6. Use-case 6: Intermediary request

Use–case 6 is shown in Figure 11. In this case, Consumer sends a service request to Intermediary. Intermediary forwards a modified request to Provider. Intermediary modifies Provider's policy for requests to express its additional policy requirements.

This use-case applies when Intermediary must examine or modify certain parts of the service request, but Provider is unaware of Intermediary's requirements. This situation exists, for instance, if Intermediary routes the request according to certain of its contents and neither Consumer nor Provider are aware of the algorithm or data requirements of Intermediary, and therefore, Consumer may encrypt the information required by Intermediary for the ultimate recipient, thereby making it unavailable to Intermediary.

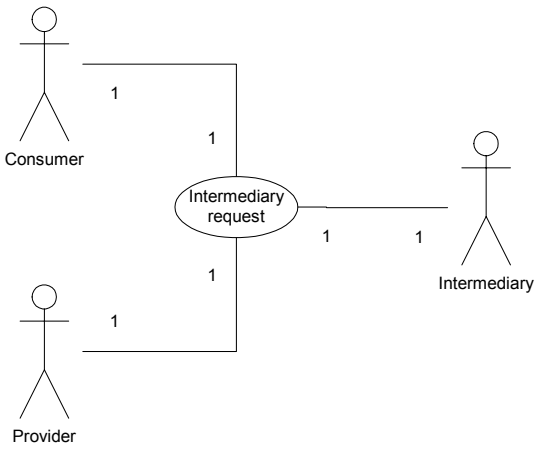
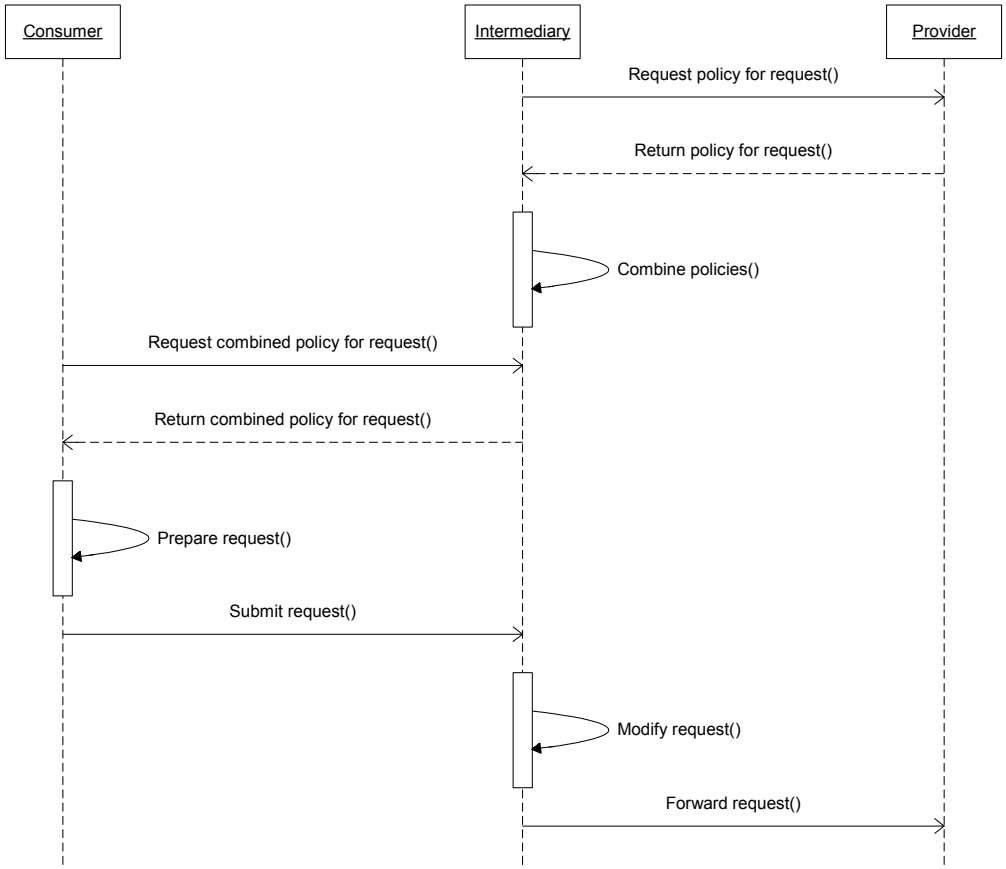


Figure 11 - Use-case 6

178 The corresponding sequence diagram is shown in Figure 12.



179

180 **Figure 12 - Use-case 6 sequence**

181 1. Intermediary requests policy from Provider.

182 2. Provider returns policy to Intermediary.

183 3. Intermediary combines Provider's policy with its own.

184 4. Consumer requests policy from Intermediary.

185 5. Intermediary returns policy to Consumer.

186 6. Consumer prepares a request in conformance with policy.

187 7. Consumer submits a conformant request to Intermediary.

188 8. Intermediary modifies the request.

189 9. Intermediary forwards the request to Provider.

190 Note: Consumer does not have to be aware that the policy provided by Intermediary is the result of
191 combining Intermediary's policy with that of Provider.

2.7. Use-case 7: Intermediary response

Use-case 7 is shown in Figure 13. In this case, Provider sends a service response to Intermediary. Intermediary sends a (potentially) modified response to Consumer. Intermediary modifies Consumer's policy for responses to express its additional policy requirements.

This use-case applies when Intermediary must examine or modify certain parts of the service response, but Consumer is unaware of Intermediary's requirements. This situation exists, for instance, if Intermediary routes the response according to certain of its contents and neither Consumer nor Provider are aware of the algorithm or data requirements of Intermediary, and therefore, Provider may encrypt the information required by Intermediary for the ultimate recipient, thereby making it unavailable to Intermediary.

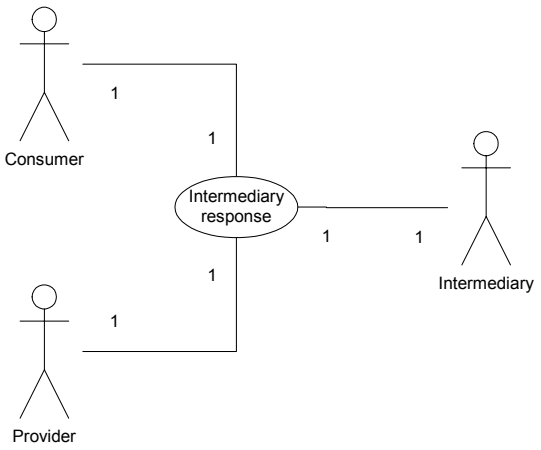
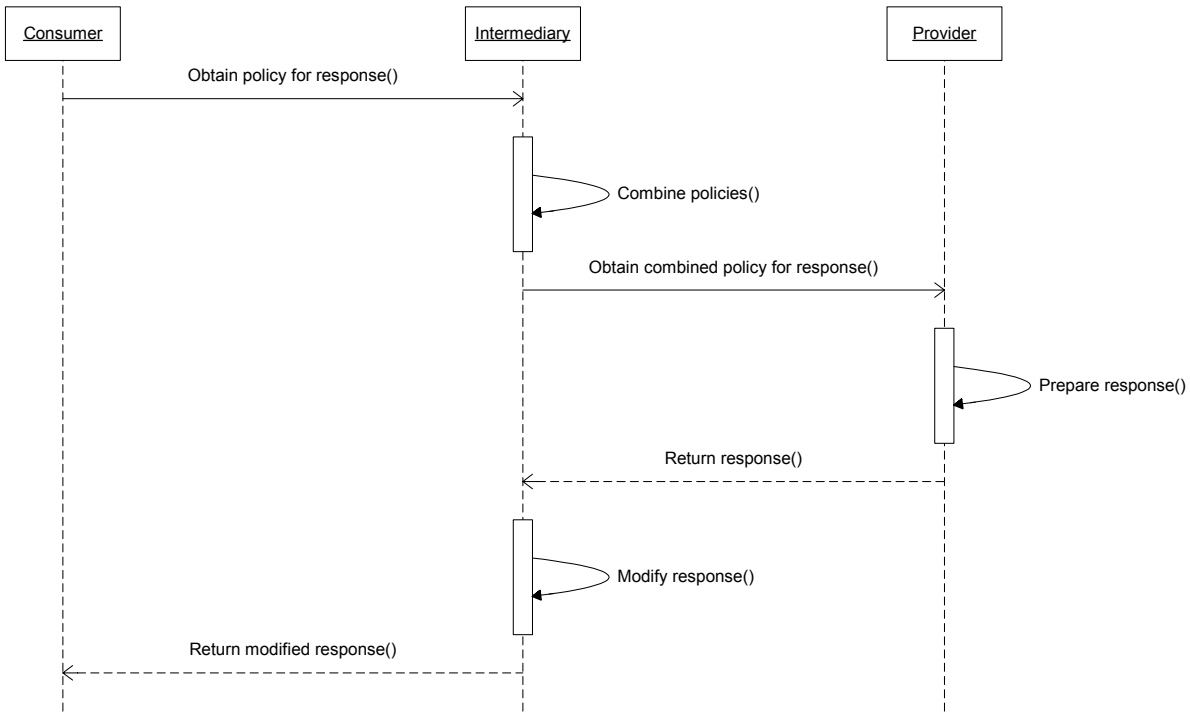


Figure 13 - Use-case 7

204 The corresponding sequence diagram is shown in Figure 14.



205

206 **Figure 14 - Use-case 7 sequence**

207 1. Intermediary obtains policy from Consumer.

208 2. Intermediary combines Consumer's policy with its own.

209 3. Provider obtains policy from Intermediary.

210 4. Provider prepares a response in conformance with policy.

211 5. Provider returns response to Intermediary.

212 6. Intermediary modifies the response.

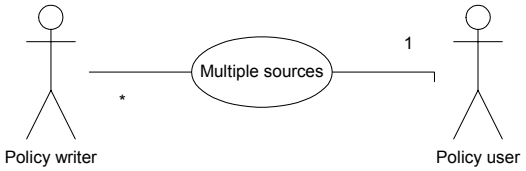
213 7. Intermediary returns the response to Consumer.

214 2.8. Use-case 8: Multiple sources

215 Use-case 8 is shown in Figure 15. In this case, the complete policy associated with a particular
216 operation (whether request or response) is formed by combining policies from a number of sources.

217 This use-case applies, for instance, when the policy applicable to a request is defined at both the
218 departmental and corporate levels. Either the policies may be combined or the evaluation results
219 may be combined. Combination may be performed by the policy user or by another actor.

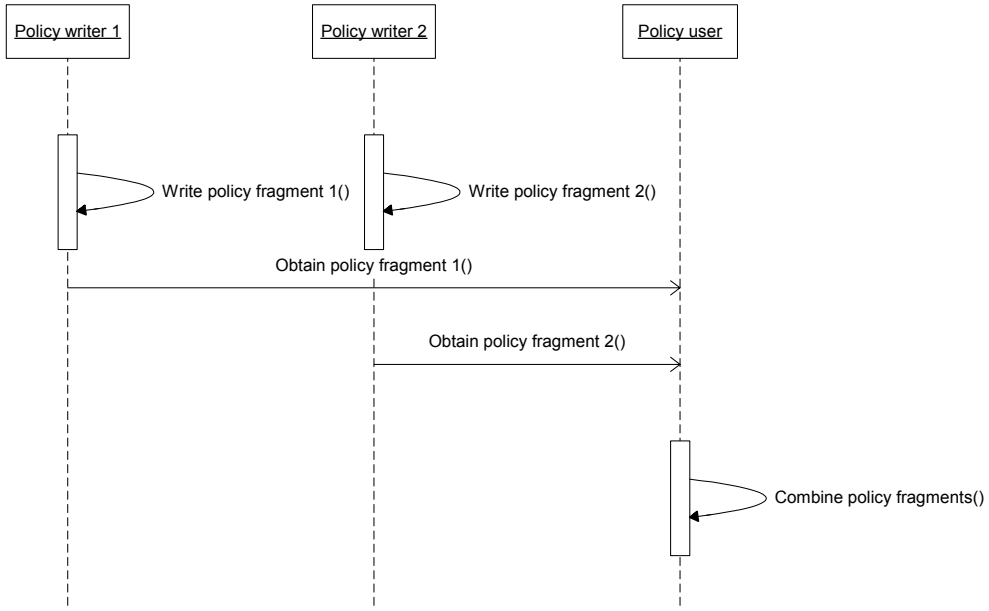
220 Policy fragments may be referenced by name.



221

222 **Figure 15 - Use-case 8**

223 The corresponding sequence diagram is shown in Figure 16.



224

225 **Figure 16 - Use-case 8 sequence**

226 1. Policy writer 1 prepares policy fragment 1.

227 2. Policy writer 2 prepares policy fragment 2.

228 3. Policy user obtains policy fragment 1.

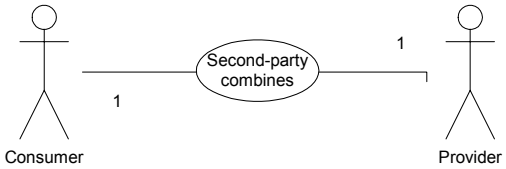
229 4. Policy user obtains policy fragment 2.

230 5. Policy user combines policy fragment 1 and policy fragment 2.

231 2.9. Use-case 9: Second party combines

232 Use-case 9 is shown in Figure 17. In this case, the combined policy associated with a service
233 request is formed by Provider and then returned to Consumer.

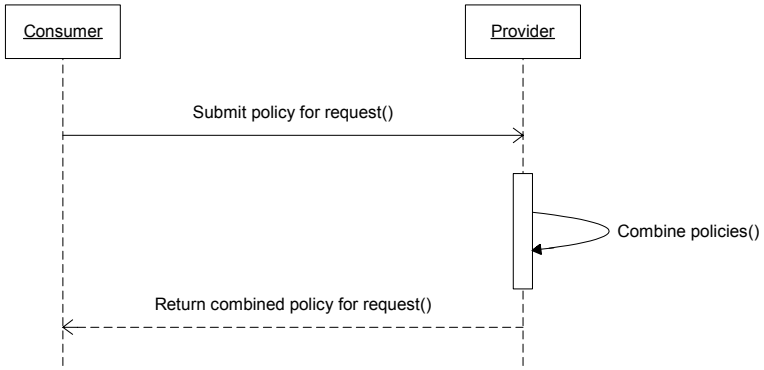
234 This use-case applies when Provider is unwilling to reveal its policy, for instance, if it wishes to
235 ensure that it preferred options are used by Consumer.



236

237 **Figure 17 - Use-case 9**

238 The corresponding sequence diagram is shown in Figure 18.



239

240 **Figure 18 - Use-case 9 sequence**

241 1. Consumer sends policy for request to Provider.

242 2. Provider combines Consumer's policy for request with its own.

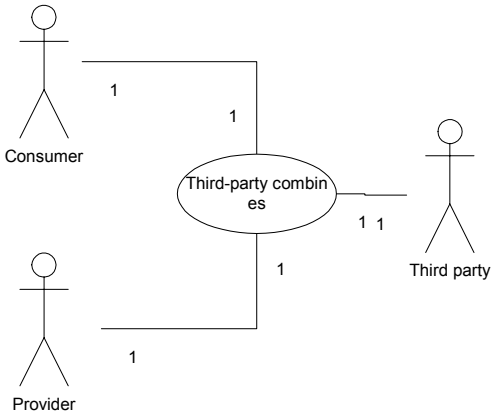
243 3. Provider returns the combined policy to Consumer.

244 4. Consumer submits a request that conforms with the combined policy.

245 2.10. Use-case 10: Third party combines

246 Use-case 10 is shown in Figure 19. In this case, the combined policy associated with a service
247 request is formed by a third party and then returned to Consumer.

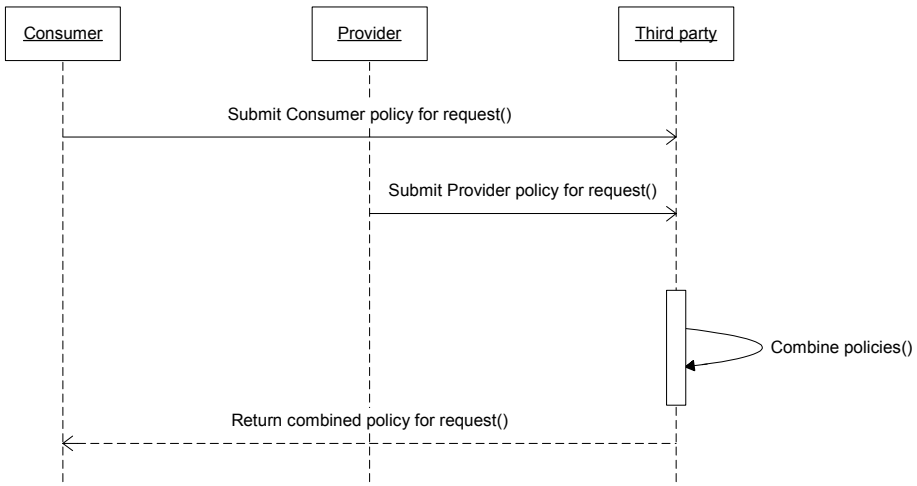
248 This applies when neither Consumer nor Provider wishes to reveal its policy to the other.



249

250 **Figure 19 - Use-case 10**

251 The corresponding sequence diagram is shown in Figure 20.



252

253 **Figure 20 - Use-case 10 sequence**

254 1. Consumer sends policy for request to Third party.

255 2. Provider sends policy for request to Third party.

256 3. Third party combines Consumer's policy for request with Provider's policy for request.

257 4. Third party returns the combined policy to Consumer.

258 5. Consumer submits a request that conforms with the combined policy.

259 3. Policy communication

260 In all use-cases, policy instances may be communicated in any one of a number of ways. For
261 instance:

In the case of simple service provision, where Consumer sends an isolated service request to Provider, Provider may publish its policy in one or more of a number of ways: by WSDL, by HTTP, by LDAP or by SQL or SAML request/response.

In the case of complex service provision, the Provider and Consumer may communicate their policies to one another in a negotiation phase by including them as SOAP headers.

4. Language support

The policy language has to support alternative combinations of requirements, which gives rise to the need for logical combining operations, such as OR and AND. Support for cryptographic-security requirements gives rise to the need for integer comparison operations, such as greater-than and less-than, and set operations, such as subset and superset, over XML nodes and resource identifiers.

It must also be possible to indicate operations that must not be performed.

5. Requirements

5.1. R1 – Evaluates to Boolean

In order to support use-cases 1,2 and 5, it must be possible to evaluate an instance of policy to produce a Boolean result. A TRUE result indicates that the requested action conforms with policy. A FALSE result indicates that it does not.

5.2. R2 – Amenable to combining

In order to support use-case 5, it must be possible to combine the results of evaluation of two or more policies. In order to support use-cases 3, 4, 6, 7, 8, 9 and 10, it must be possible to combine and reduce two or more policies to derive a set of instructions (see R3).

5.3. R3 – Clear semantics

In order to support use-cases 3 and 4, it must be possible to derive from a policy instance a set of instructions for producing a request that conforms with the policy.

5.4. R4 – Common data-types

In order to support multiple policy types in an efficient and interoperable manner, a common set of data-types must be defined. This must include integers, XML nodes and resource identifiers.

5.5. R5 – Extensible data-types

In order to address unforeseen applications, it must be possible to extend the set of built-in data-types.

5.6. R6 - Common operators

In order to support multiple policy types in an efficient and interoperable manner, a common set of operators must be defined. These must include logical operators (including NOT), integer comparison operators and set operators.

5.7. R7 – Extensible operators

In order to address unforeseen applications, it must be possible to extend the set of built-in operators.

5.8. R8 – Multiple enforcement points

In order to support multiple policy types, each with a distinct enforcement point, it must be possible to target a policy instance at a specific enforcement point and message type, and for that enforcement point to be able to identify and obtain the piece of a policy instance that is appropriate to it. Enforcement points must, at least, include: cryptographic-security, authentication, authorization, privacy, reliable-messaging and transactions.

5.9. R9 – Multiple bindings

It must be possible to convey policy instances in a number of different protocols, including: WSDL, SOAP, LDAP, HTTP and SQL and SAML attribute request/response.

5.10. R10 – Preferences

It must be possible for a Web-services end-point to indicate its order of preference amongst a mutually-acceptable set of optional functions.

5.11. R11 – Capabilities

It must be possible for a Web-services end-point to indicate operations that it is capable of performing, as well as operations that it insists upon performing.

5.12. R12 – Specified order

It must be possible for a Web-services end-point to indicate the order in which it will perform operations.

5.13. R13 – Policy identified by name

It must be possible to reference a policy instance by name.

5.14. R14 – Attributes identified by name

It must be possible to reference attributes in a policy instance by name.

5.15. R15 – Attributes identified by location

It must be possible to reference attributes in a policy instance by location.

5.16. R16 – Behaviour in event attributes are unavailable

It must be possible to specify in a policy instance behaviour in the event that referenced attributes cannot be evaluated.

Appendix A. Notices

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's procedures with respect to rights in OASIS specifications can be found at 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 implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.