



Services et protocoles du registre ebXML

Ébauche 03, 2 mai, 2005

Identifiant du document

regrep-rs 3.0-cd-02_fin_fr

Hyperlien

<http://www.oasis-open.org/committees/regrep/documents/3.0/specs/regrep-rs-3.0-draft-03.pdf>

Rédacteurs

Nom	Affiliation
Sally Fuger	Membre à titre individuel
Farrukh Najmi	Sun Microsystems
Nikola Stojanovic	RosettaNet

Collaborateurs

Nom	Affiliation
Diego Ballve	Membre à titre individuel
Ivan Bedini	France Telecom
Kathryn Breininger	The Boeing Company
Joseph Chiusano	Booz Allen Hamilton
Peter Kacandes	Adobe Systems
Paul Macias	LMI Government Consulting
Carl Mattocks	Membre à titre individuel
Matthew MacKenzie	Adobe Systems
Monica Martin	Sun Microsystems
Richard Martell	Galdos Systems Inc.
Duane Nickull	Adobe Systems
Goran Zugic	ebXMLsoft Inc.

12

13 **Résumé**

14 Le présent document définit les services et les protocoles qui s'appliquent à un registre ebXML.

15 Un autre document, soit *Modèle d'information du registre ebXML* [ebRIM], définit les types de
16 métadonnées et de contenu qui peuvent être stockés dans un registre ebXML.

17 **Statut**

18 Le présent document se traduit par une spécification provisoire approuvée par le Comité
19 technique du registre OASIS/ebXML.

20 Les membres du Comité sont invités à présenter leurs commentaires au sujet de cette
21 spécification par le biais de la liste regrep@lists.oasis-open.org. Les autres intervenants, s'ils le
22 désirent, sont priés de s'inscrire à la liste regrep-comment@lists.oasis-open.org et de s'en servir
23 pour formuler leurs commentaires. Pour s'inscrire, prière de faire parvenir un courriel à [regrep-](mailto:regrep-comment-request@lists.oasis-open.org)
24 comment-request@lists.oasis-open.org, en y indiquant le mot « inscription » dans le corps du
25 texte.

26 Pour savoir si des brevets déposés pourraient être essentiels dans l'implantation de la
27 spécification, et pour savoir s'il y a eu des offres par rapport aux conditions de délivrance de ces
28 brevets, prière de consulter la section sur les droits de propriété intellectuelle. Celle-ci figure sur la
29 page Web du Comité technique du registre OASIS/ebXML ([http://www.oasis-](http://www.oasis-open.org/committees/regrep/)
30 [open.org/committees/regrep/](http://www.oasis-open.org/committees/regrep/)).

31

32

33

34 This translated document is provided by the Ministère des Services gouvernementaux du Québec
35 as an informational service to the global community. This is an unofficial, non non-normative
36 translation of the official document,

37 OASIS/ebXML Registry Information Model v3,0, located at

38 <http://www.oasis-open.org/committees/download.php/11383/regrep-3.0-cd-01.zip> ,

39 © copyright OASIS 2005.

40 This translation is published with acknowledgement of and in agreement with terms specified in
41 the OASIS Translation Policy. Neither OASIS nor the Ministère des Services gouvernementaux du
42 Québec assume responsibility for any errors contained herein.

44 Ce document est une traduction fournie par le Ministère des Services gouvernementaux comme
45 un service pour l'information publique à l'échelle internationale. Cette traduction est non-officielle, elle n'a
46 pas un caractère normatif en relation avec le document OASIS/ebXML Registry Information Model v3,0
47 , localisé à

48 <http://www.oasis-open.org/committees/download.php/11383/regrep-3.0-cd-01.zip> ,

49 © copyright OASIS 2005.

50 Cette traduction est publiée en accord avec les Règles d'OASIS relatives à la traduction. Ni
51 OASIS ni le Ministère des Services gouvernementaux n'assument de responsabilité pour toute
52 erreur qui pourrait s'y trouver.

53

54 Le travail de traduction a été effectué par une professionnelle de la traduction, madame Carole
55 LeBel, carole.lebel@sympatico.ca , avec le soutien de Patrice di Marcantonio et Richard Parent
56 pour la terminologie spécialisée.

Table des matières

57		
58	1 Introduction.....	13
59	1.1 Public cible.....	13
60	1.2 Terminologie.....	13
61	1.3 Conventions notationnelles.....	13
62	1.3.1 Diagrammes UML.....	13
63	1.3.2 Nom général de champ d'identifiant.....	13
64	1.3.3 Constantes.....	14
65	1.3.4 Texte en caractères gras.....	14
66	1.3.5 Exemples de valeur.....	14
67	1.4 Conventions liées au schéma XML.....	14
68	1.4.1 Schémas définis par le registre ebXML.....	14
69	1.4.2 Schémas utilisés par le registre ebXML.....	15
70	1.5 Acteurs de registre.....	16
71	1.6 Cas d'utilisation du registre.....	17
72	1.7 Architecture de registre.....	17
73	1.7.1 Applications de registre.....	17
74	1.7.1.1 API d'application.....	18
75	1.7.2 Interfaces de services de registre.....	18
76	1.7.3 Interface de services : rattachements de protocole.....	18
77	1.7.4 Authentification et autorisation.....	18
78	1.7.5 Registre de métadonnées et référentiel de contenu.....	18
79	2 Protocoles de registre.....	19
80	2.1 Demandes et réponses.....	19
81	2.1.1 Type-de-demande-de-registre.....	19
82	2.1.1.1 Syntaxe.....	19
83	2.1.1.2 Paramètres.....	20
84	2.1.1.3 Renvois.....	20
85	2.1.1.4 Exceptions.....	20
86	2.1.2 Demande-de-registre.....	20
87	2.1.3 Type-de-réponse-de-registre.....	21
88	2.1.3.1 Syntaxe.....	21
89	2.1.3.2 Paramètres.....	21
90	2.1.4 Réponse-de-registre.....	22
91	2.1.5 Liste-d'erreurs-de-registre.....	22
92	2.1.5.1 Syntaxe.....	22
93	2.1.5.2 Paramètres.....	22
94	2.1.6 Erreur-de-registre.....	22
95	2.1.6.1 Syntaxe.....	22
96	2.1.6.2 Paramètres.....	23
97	3 Rattachement de protocole SOAP.....	24
98	3.1 Interfaces de service du registre ebXML : Définition abstraite.....	24
99	3.2 Rattachement du protocole SOAP propre aux interfaces de services du registre ebXML.....	25
100	3.3 Interfaces de service du registre ebXML : Service SOAP.....	25
101	3.4 Appariement entre les exceptions et les erreurs SOAP.....	26

102	4 Rattachement de protocole HTTP.....	27
103	4.1 URL fondé sur l'encodage d'un RPC.....	27
104	4.1.1 Paramètres URL standard.....	28
105	4.1.2 Rattachement de Gestionnaire-d'interrogation.....	28
106	4.1.2.1 Exemple de demande Obtenir-l'Objet-de-registre.....	29
107	4.1.2.2 Exemple de réponse Obtenir-l'Objet-de-registre.....	29
108	4.1.2.3 Exemple de demande Obtenir-l'Item-de-référentiel.....	29
109	4.1.2.4 Exemple de réponse Obtenir-l'Item-de-référentiel.....	29
110	4.1.3 Interface HTTP liée au Gestionnaire-de-cycle-de-vie.....	29
111	4.2 URL défini par le demandeur.....	30
112	4.2.1 Syntaxe de l'URL défini par le demandeur.....	30
113	4.2.2 Attribution d'un URL à un Objet-de-registre.....	31
114	4.2.3 Attribution d'un URL à un Item-de-référentiel.....	31
115	4.3 URL par chemin d'accès.....	31
116	4.3.1 Métaphore de fichiers et de dossiers.....	31
117	4.3.2 Chemin d'accès d'un Objet-de-registre.....	32
118	4.3.2.1 Exemple de chemin d'accès.....	32
119	4.3.3 Appariement entre un URL et des objets.....	32
120	4.3.4 Appariement entre un URL et un objet unique.....	33
121	4.3.5 Appariement entre un URL et des objets multiples.....	33
122	4.3.6 Inscription au répertoire.....	33
123	4.3.7 Contrôle d'accès dans la hiérarchie de Paquets-de-registre	34
124	4.4 Algorithme de définition URL.....	34
125	4.5 Motifs de sécurité.....	35
126	4.6 Gestion des exceptions.....	35
127	5 Protocoles de gestion du cycle de vie.....	36
128	5.1 Protocole de demande d'inscription d'objet.....	36
129	5.1.1 Demande-d'inscription-d'objet.....	36
130	5.1.1.1 Syntaxe.....	36
131	5.1.1.2 Paramètres.....	37
132	5.1.1.3 Renvois.....	37
133	5.1.1.4 Exceptions.....	37
134	5.1.2 Génération de l'identifiant unique.....	37
135	5.1.3 Attribut identifiant et références d'objet.....	37
136	5.1.4 Piste de vérification.....	38
137	5.1.5 Exemple de Demande-d'inscription-d'objet.....	38
138	5.2 Protocole de mise à jour d'objet.....	38
139	5.2.1 Demande-de-mise-à-jour-d'objet.....	38
140	5.2.1.1 Syntaxe.....	38
141	5.2.1.2 Paramètres.....	39
142	5.2.1.3 Renvois.....	39
143	5.2.1.4 Exceptions.....	39
144	5.2.2 Piste de vérification.....	40
145	5.3 Protocole d'approbation d'objet.....	40
146	5.3.1 Demande-d'approbation-d'objet.....	40
147	5.3.1.1 Syntaxe.....	40
148	5.3.1.2 Paramètres.....	41

149	5.3.1.3 Renvois.....	41
150	5.3.1.4 Exceptions.....	41
151	5.3.2 Piste de vérification.....	41
152	5.4 Protocole de déclassement d'objet.....	41
153	5.4.1 Demande-de-déclassement-d'objet.....	41
154	5.4.1.1 Syntaxe.....	41
155	5.4.1.2 Paramètres.....	42
156	5.4.1.3 Renvois.....	42
157	5.4.1.4 Exceptions.....	42
158	5.4.2 Piste de vérification.....	42
159	5.5 Protocole de rétablissement d'objet.....	42
160	5.5.1 Demande-de-rétablissement-d'objet.....	43
161	5.5.1.1 Syntaxe.....	43
162	5.5.1.2 Paramètres.....	43
163	5.5.1.3 Renvois.....	44
164	5.5.1.4 Exceptions.....	44
165	5.5.2 Piste de vérification.....	44
166	5.6 Protocole de suppression d'objet.....	44
167	5.6.1 Demande-de-suppression-d'objet.....	44
168	5.6.1.1 Syntaxe.....	44
169	5.6.1.2 Paramètres.....	45
170	5.6.1.3 Renvois.....	45
171	5.6.1.4 Exceptions.....	45
172	5.7 Contrôle de version géré par registre.....	46
173	5.7.1 Ressources placées sous contrôle de version.....	46
174	5.7.2 Versionnage et identification d'objet.....	46
175	5.7.3 ID logique.....	46
176	5.7.4 Établissement de la version.....	47
177	5.7.4.1 Établissement de la version d'un Objet-de-registre.....	47
178	5.7.4.2 Établissement de la version d'un Item-de-référentiel.....	47
179	5.7.5 Versionnage d'un Objet-extrinsèque et d'un Item-de-référentiel.....	47
180	5.7.5.1 Objet-extrinsèque et Item-de-référentiel partagé.....	48
181	5.7.6 Versionnage et objets composés.....	48
182	5.7.7 Versionnage et références.....	48
183	5.7.8 Versionnage et piste de vérification.....	49
184	5.7.9 Association inter-versions.....	49
185	5.7.10 Suppression de version entamée par une application.....	49
186	5.7.11 Suppression de version engagée par le registre.....	49
187	5.7.12 Verrouillage et modifications simultanées.....	49
188	5.7.13 Création d'une version.....	50
189	5.7.14 Annulation du versionnage.....	50
190	6 Protocoles de gestion d'interrogation.....	51
191	6.1 Protocole d'interrogation ad hoc.....	51
192	6.1.1 Demande-d'interrogation-ad-hoc.....	51
193	6.1.1.1 Syntaxe.....	51
194	6.1.1.2 Paramètres.....	52
195	6.1.1.3 Renvois.....	52

196	6.1.1.4 Exceptions.....	52
197	6.1.2 Réponse-à-une-interrogation-ad-hoc.....	53
198	6.1.2.1 Syntaxe.....	53
199	6.1.2.2 Paramètres.....	53
200	6.1.3 Interrogation ad hoc.....	53
201	6.1.3.1 Syntaxe.....	53
202	6.1.3.2 Paramètres.....	53
203	6.1.4 Option-de-réponse.....	54
204	6.1.4.1 Syntaxe.....	54
205	6.1.4.2 Paramètres.....	54
206	6.2 Soutien d'interrogation itérative.....	55
207	6.2.1 Exemple d'itération d'interrogation.....	55
208	6.3 Soutien d'interrogation stockée.....	55
209	6.3.1 Inscription d'une interrogation stockée.....	56
210	6.3.1.1 Déclaration de paramètres d'interrogation.....	56
211	6.3.1.2 Paramètres de contexte canonique.....	57
212	6.3.2 Appel d'une interrogation stockée.....	57
213	6.3.2.1 Établissement des paramètres d'appel d'interrogation.....	57
214	6.3.3 Réponse à un appel d'interrogation stockée.....	58
215	6.3.4 Contrôle d'accès à une interrogation stockée.....	58
216	6.3.5 Interrogation canonique : obtention de l'objet d'utilisateur d'une application.....	58
217	6.4 Syntaxe de requête SQL.....	58
218	6.4.1 FBN propre à la grammaire de la syntaxe de requête (non normative).....	59
219	6.4.2 Schéma relationnel propre aux requêtes SQL.....	60
220	6.4.3 Résultats de requête SQL.....	60
221	6.5 Syntaxe de filtre d'interrogation.....	60
222	6.5.1 Structure de filtre d'interrogation.....	61
223	6.5.2 Éléments d'interrogation.....	61
224	6.5.3 Éléments de filtre.....	62
225	6.5.3.1 Type-de-filtre.....	63
226	6.5.3.2 Type-de-filtre-simple.....	63
227	6.5.3.3 Filtre-booléen.....	64
228	6.5.3.4 Filtre-à-point-flottant.....	64
229	6.5.3.5 Filtre-entier.....	64
230	6.5.3.6 Filtre-date-et-heure.....	65
231	6.5.3.7 Filtre-de-chaîne.....	65
232	6.5.3.8 Filtre-composé.....	65
233	6.5.4 Éléments d'interrogation concaténée.....	66
234	6.5.5 Éléments de branche.....	67
235	6.6 Exemples d'interrogation.....	67
236	6.6.1 Interrogations de nom et de description.....	67
237	6.6.2 Interrogations de Classification.....	68
238	6.6.2.1 Extraction de Schème-de-classification.....	68
239	6.6.2.2 Extraction des enfants d'un Nœud-de-classification spécifié.....	68
240	6.6.2.3 Extraction des objets classifiés par un Nœud-de-classification.....	68
241	6.6.2.4 Extraction des Classifications qui classifient un objet.....	69
242	6.6.3 Interrogations d'Association.....	69
243	6.6.3.1 Extraction de l'ensemble des Associations dont la source constitue un objet spécifié.....	69

244	6.6.3.2 Extraction de l'ensemble des Associations dont la cible représente un objet spécifié.....	70
245	6.6.3.3 Extraction des objets associés fondés sur le type d'Association.....	70
246	6.6.3.4 Interrogation complexe d'Association.....	70
247	6.6.4 Interrogations de Paquet-de-registre	70
248	6.6.5 Interrogations de Lien-externe.....	71
249	6.6.6 Interrogations d'Événement-vérifiable.....	72
250	7 Protocoles de notification d'événement.....	73
251	7.1 Cas d'utilisation.....	73
252	7.1.1 Nouveau Profil de protocole de collaboration.....	73
253	7.1.2 Disponibilité d'un nouveau service.....	73
254	7.1.3 Surveillance du téléchargement d'un contenu.....	73
255	7.1.4 Surveillance des changements de prix.....	73
256	7.1.5 Assurance de la concordance entre les duplications et l'objet source.....	73
257	7.2 Événements de registre.....	73
258	7.3 Abonnement à des événements.....	74
259	7.3.1 Sélection d'événement.....	74
260	7.3.2 Action de notification.....	74
261	7.3.3 Autorisation d'Abonnement.....	75
262	7.3.4 Quota d'Abonnements.....	75
263	7.3.5 Expiration d'un Abonnement.....	75
264	7.3.6 Rejet d'un Abonnement.....	75
265	7.4 Désabonnement à des événements.....	75
266	7.5 Notification d'événement.....	76
267	7.6 Extraction d'événement.....	76
268	7.7 Élagage d'événement.....	76
269	8 Services de gestion de contenu.....	77
270	8.1 Validation de contenu.....	77
271	8.1.1 Validation de contenu : cas d'utilisation.....	77
272	8.1.1.1 Validation des profils de conformité de HL7.....	77
273	8.1.1.2 Validation des processus d'affaires.....	77
274	8.1.1.3 Validation des documents d'affaires UBL.....	78
275	8.2 Catalogage de contenu.....	78
276	8.2.1 Découverte fondée sur le contenu : cas d'utilisation.....	78
277	8.2.1.1 Détection de tous les CPP qui jouent un rôle d'« acheteur ».....	78
278	8.2.1.2 Détection de tous les schémas XML qui utilisent un espace de nommage précis.....	78
279	8.2.1.3 Détection de toutes les descriptions WSDL ayant un rattachement de protocole SOAP.....	78
280	8.3 Service abstrait de gestion de contenu.....	79
281	8.3.1 Modèle d'appel incorporé.....	79
282	8.3.2 Modèle d'appel découplé.....	80
283	8.4 Protocole de service de gestion de contenu.....	81
284	8.4.1 Type-de-demande-de-service-de-gestion-de-contenu.....	82
285	8.4.1.1 Syntaxe.....	82
286	8.4.1.2 Paramètres.....	82
287	8.4.1.3 Renvois.....	82
288	8.4.1.4 Exceptions.....	82
289	8.4.2 Type-de-réponse-de-service-de-gestion-de-contenu.....	83
290	8.4.2.1 Syntaxe.....	83

291	8.4.2.2 Paramètres.....	83
292	8.5 Publication et configuration d'un service de gestion de contenu.....	83
293	8.5.1 Services multiples de gestion de contenu et fichiers multiples de contrôle d'appel.....	85
294	8.6 Appel d'un service de gestion de contenu.....	85
295	8.6.1 Algorithme de résolution propre aux services et aux fichiers de contrôle d'appel.....	85
296	8.6.2 Piste de vérification et contenu catalogué.....	86
297	8.6.3 Intégrité référentielle.....	86
298	8.6.4 Gestion d'erreur.....	86
299	8.7 Protocole de validation de contenu.....	86
300	8.7.1 Demande-de-validation-de-contenu.....	87
301	8.7.1.1 Syntaxe.....	87
302	8.7.1.2 Paramètres.....	87
303	8.7.1.3 Renvois.....	88
304	8.7.1.4 Exceptions.....	88
305	8.7.2 Réponse-de-validation-de-contenu.....	88
306	8.7.2.1 Syntaxe.....	88
307	8.7.2.2 Paramètres.....	88
308	8.8 Protocole de catalogage de contenu.....	89
309	8.8.1 Demande-de-catalogage-de-contenu.....	89
310	8.8.1.1 Syntaxe.....	89
311	8.8.1.2 Paramètres.....	90
312	8.8.1.3 Renvois.....	90
313	8.8.1.4 Exceptions.....	90
314	8.8.2 Réponse-de-catalogage-de-contenu.....	90
315	8.8.2.1 Syntaxe.....	90
316	8.8.2.2 Paramètres.....	90
317	8.9 Exemple typique : service de catalogage XML canonique.....	91
318	8.10 Service de catalogage de contenu XML canonique.....	92
319	8.10.1 Publication du service de catalogage de contenu XML canonique.....	92
320	9 Gestion de registres en coopération.....	93
321	9.1 Cas d'utilisation axés sur des registres en coopération.....	93
322	9.1.1 Références d'objet entre registres.....	93
323	9.1.2 Interrogations fédérées.....	93
324	9.1.3 Mise en cache locale des données d'un autre registre.....	93
325	9.1.4 Déplacement d'objet.....	94
326	9.2 Fédérations de registres.....	94
327	9.2.1 Métadonnées de fédération.....	94
328	9.2.2 Interrogations locales et interrogations fédérées.....	95
329	9.2.2.1 Interrogations locales.....	95
330	9.2.2.2 Interrogations fédérées.....	95
331	9.2.2.3 Appartenance à de multiples fédérations.....	96
332	9.2.3 Opérations fédérées de gestion du cycle-de-vie.....	96
333	9.2.4 Fédérations et mise en cache locale des données à distance.....	96
334	9.2.5 Mise en cache de métadonnées de fédération.....	96
335	9.2.6 Synchronisation de l'heure entre les registres homologues.....	97
336	9.2.7 Fédérations et sécurité.....	97
337	9.2.8 Protocoles de gestion du cycle-de-vie de fédération.....	97

338	9.2.8.1 Union avec une fédération.....	97
339	9.2.8.2 Création d'une fédération.....	97
340	9.2.8.3 Sortie d'une fédération.....	98
341	9.2.8.4 Dissolution d'une fédération.....	98
342	9.3 Duplication d'objet.....	98
343	9.3.1 Cas d'utilisation propres à la duplication d'objet.....	99
344	9.3.2 Interrogations et duplicatas.....	99
345	9.3.3 Opérations liées au cycle-de-vie et duplicatas.....	99
346	9.3.4 Duplication d'objet et registres fédérés.....	99
347	9.3.5 Création d'un duplicata local.....	100
348	9.3.6 Duplication transactionnelle.....	100
349	9.3.7 Tenue à jour des duplicatas.....	100
350	9.3.8 Gestion du cycle-de-vie des duplicatas locaux.....	100
351	9.3.9 Suivi de la localisation d'un duplicata.....	101
352	9.3.10 Références d'objet distantes par rapport à un duplicata.....	101
353	9.3.11 Suppression d'un duplicata local.....	101
354	9.4 Protocole de déplacement d'objet.....	101
355	9.4.1 Demande-de-déplacement-d'objet.....	103
356	9.4.1.1 Paramètres.....	104
357	9.4.1.2 Renvois.....	104
358	9.4.1.3 Exceptions.....	104
359	9.4.2 Demande-d'acceptation-d'objet.....	104
360	9.4.2.1 Paramètres.....	104
361	9.4.2.2 Renvois.....	105
362	9.4.2.3 Exceptions.....	105
363	9.4.3 Déplacement d'objet et Références-d'objet distantes.....	105
364	9.4.4 Notification de déplacement d'objet auprès du détenteur-de-registre-de-destination.....	105
365	9.4.5 Notification d'inscription permanente d'objet auprès du registre-source.....	106
366	9.4.6 Détention d'objet et réaffectation de détenteur.....	106
367	9.4.7 Déplacement d'objet et délais d'attente.....	106
368	10 Sécurité du registre.....	107
369	10.1 Cas d'utilisation liés à la sécurité.....	107
370	10.1.1 Gestion de l'identité.....	107
371	10.1.2 Sécurité de message.....	107
372	10.1.3 Sécurité d'item de référentiel.....	107
373	10.1.4 Authentification.....	107
374	10.1.5 Autorisation et contrôle d'accès.....	107
375	10.1.6 Piste de vérification.....	107
376	10.2 Gestion de l'identité.....	108
377	10.3 Sécurité de message.....	108
378	10.3.1 Sécurité de la couche de transport.....	108
379	10.3.2 Sécurité de message SOAP.....	108
380	10.3.2.1 Signature de message de demande.....	108
381	10.3.2.2 Signature de message de réponse.....	108
382	10.3.2.3 Exigences liées à l'Information-de-clé.....	109
383	10.3.2.4 Validation de signature de message.....	109
384	10.3.2.5 Exemple de signature de message.....	109

385	10.3.2.6 Message comprenant un Item-de-référentiel : exemple de signature.....	110
386	10.3.2.7 Sécurité de message SOAP et protocole HTTP/S.....	112
387	10.3.3 Confidentialité des messages.....	112
388	10.3.4 Exigences liées à la distribution des clés publiques.....	112
389	10.4 Authentification.....	113
390	10.4.1 Registre à titre d'autorité d'authentification.....	113
391	10.4.2 Autorité d'authentification externe.....	113
392	10.4.3 Gestion de session authentifiée.....	113
393	10.5 Autorisation et contrôle d'accès.....	113
394	10.6 Piste de vérification.....	114
395	11 Profil SAML de registre.....	115
396	11.1 Terminologie.....	115
397	11.2 Cas d'utilisation liés au profil SAML.....	115
398	11.2.1 Registre à titre de participant au protocole d'accès unifié.....	116
399	11.3 Rôle du registre par rapport aux protocoles SAML.....	116
400	11.3.1 Rôle de fournisseur de service.....	116
401	11.3.1.1 Exigences liées au rôle de fournisseur de service.....	116
402	11.4 Interface SAML de registre.....	117
403	11.5 Exigences liées au profil SAML de registre.....	117
404	11.6 Opération d'accès unifié.....	118
405	11.6.1 Acteurs de scénario.....	118
406	11.6.2 Opération d'accès unifié – demandeur HTTP non authentifié.....	118
407	11.6.2.1 Séquence de scénario.....	118
408	11.6.3 Opération d'accès unifié – demandeur HTTP authentifié.....	120
409	11.6.4 Opération d'accès unifié – demandeur SOAP non authentifié.....	120
410	11.6.4.1 Séquence de scénario.....	120
411	11.6.5 Opération d'accès unifié – demandeur SOAP authentifié.....	121
412	11.6.5.1 Séquence de scénario.....	122
413	11.6.6 Règles de génération de l'élément <samlp:Demande-d'authentification>.....	123
414	11.6.7 Règles de traitement de l'élément <samlp:Réponse>.....	123
415	11.6.8 Appariement entre un sujet et un utilisateur.....	123
416	11.7 Utilisateurs externes.....	124
417	12 Support de langue naturelle.....	125
418	12.1 Terminologie.....	125
419	12.2 Support de langue naturelle et messages de protocole de registre.....	125
420	12.3 Gestion du support de langue naturelle dans les Objets-de-registre.....	126
421	12.3.1 Jeu de caractères d'une Chaîne-en-une-langue-donnée.....	127
422	12.3.2 Langage d'une Chaîne-en-une-langue-donnée.....	127
423	12.4 Support de langue naturelle et items de référentiel.....	127
424	12.4.1 Jeu de caractères d'un item de référentiel.....	127
425	12.4.2 Langage d'un item de référentiel.....	127
426	13 Conformité.....	129
427	13.1 Profils de conformité.....	129
428	13.2 Grille de caractéristiques.....	129
429	14 Références.....	131
430	14.1 Références normatives.....	131

431	14.2 Références informatives.....	132
432		

Index des diagrammes

Diagramme 1 : Aperçu simplifié de l'architecture du registre ebXML.....	17
Diagramme 2 : Demandes et réponses des protocoles de registre.....	19
Diagramme 3 : Exemple de hiérarchie de Paquets-de-registre	32
Diagramme 4 : Exemple d'inscription au répertoire.....	34
Diagramme 5 : Protocole de demande d'inscription d'objet.....	36
Diagramme 6 : Protocole de mise à jour d'objet.....	38
Diagramme 7 : Protocole d'approbation d'objet.....	40
Diagramme 8 : Protocole de déclassement d'objet.....	41
Diagramme 9 : Protocole de rétablissement d'objet.....	43
Diagramme 10: Protocole de suppression d'objet.....	44
Diagramme 11 : Protocole d'interrogation ad hoc.....	51
Diagramme 12 : Hiérarchie du type de filtre.....	63
Diagramme 13 : Service de validation de contenu.....	77
Diagramme 14 : Service de catalogage de contenu.....	78
Diagramme 15 : Service de gestion de contenu : modèle d'appel incorporé.....	80
Diagramme 16 : Service de gestion de contenu : modèle d'appel découplé.....	81
Diagramme 17 : Configuration de service de catalogage.....	84
Diagramme 18 : Protocole de validation de contenu.....	87
Diagramme 19 : Protocole de catalogage de contenu.....	89
Diagramme 20 : Exemple de catalogage de CPP par le service de catalogage XML canonique.....	91
Diagramme 21 : Références d'objet entre registres.....	93
Diagramme 22 : Fédérations de registre.....	94
Diagramme 23 : Exemple de métadonnée de fédération.....	95
Diagramme 24 : Duplication d'objet.....	99
Diagramme 25 : Déplacement d'objet.....	101
Diagramme 26 : Protocole de déplacement d'objet.....	102
Diagramme 27 : Exemple représentatif de protocole d'accès unifié SAML.....	116
Diagramme 28 : Opération d'accès unifié – demandeur HTTP non authentifié.....	118
Diagramme 29 : Opération d'accès unifié – demandeur SOAP non authentifié.....	120
Diagramme 30 : Opération d'accès unifié – demandeur SOAP authentifié.....	121

1 Introduction

434

435 Un registre ebXML représente un système d'information qui gère de manière sûre tout type de contenu
436 ainsi que les métadonnées normalisées qui le décrivent.

437 Le registre ebXML offre un ensemble de services qui favorisent le partage de contenu et de métadonnées
438 entre des entités organisationnelles dans un environnement fédéré. Il est possible de déployer un tel
439 registre dans un serveur d'applications, un serveur Web ou un autre conteneur de services quelconque.
440 Les applications PEUVENT se servir du registre comme site Web public, semi-public ou privé.

441 Le présent document définit les services fournis par un registre ebXML de même que les protocoles
442 utilisés par les applications de ce registre pour assurer une interaction entre elles-mêmes et ces services.

443 Un autre document, soit *Modèle d'information du registre ebXML* [ebRIM], établit les types de
444 métadonnées et de contenu qui peuvent être stockés dans tout registre ebXML.

1.1 Public cible

445

446 La présente spécification s'adresse à la communauté de développeurs de logiciels, soit :

- 447 • les réalisateurs des services du registre ebXML;
- 448 • les réalisateurs des applications du registre ebXML.

1.2 Terminologie

449

450 Les mots clés « DOIT », « NE DOIT PAS », « REQUIS », « DEVRA », « NE DEVRA PAS »,
451 « DEVRAIT », « NE DEVRAIT PAS », « RECOMMANDÉ », « PEUT » et « OPTIONNEL », et leurs formes
452 au pluriel, qui figurent dans le présent document doivent être interprétés tels qu'ils sont décrits dans la
453 RFC 2119 [RFC 2119] de l'Internet Engineering Task Force (IETF).

454 Le terme « *item de référentiel* » (repository item) désigne un contenu (par exemple, un document XML ou
455 une DTD) qui loge dans un référentiel pour y être stocké et sauvegardé. Chaque item de référentiel est
456 décrit par une instance d'Objet-de-registre. Cet Objet-de-registre catalogue l'Item-de-référentiel qui
457 dispose de métadonnées.

1.3 Conventions notationnelles

458

459 On se sert des conventions ci-dessous dans tout le document pour déterminer les structures de données
460 utilisées. Aussi, on a recours aux conventions de formatage de texte suivantes pour faciliter la lisibilité :

1.3.1 Diagrammes UML

461

462 On utilise des diagrammes de langage de modélisation unifié [UML] pour décrire des concepts de
463 manière concise. Ces diagrammes ne visent pas à transmettre des exigences quelconques en matière
464 d'implantation ou de méthodologie.

1.3.2 Nom général de champ d'identifiant

465

466 Les listes peuvent renfermer des valeurs qui font référence aux objets du registre ebXML par leur attribut
467 identifiant. Ces valeurs d'identifiant détectent seulement les objets qui se trouvent dans le registre ebXML.
468 À des fins de commodité et pour assurer une meilleure lisibilité, on remplace ces valeurs clés par des
469 variables textuelles importantes de façon qu'elles représentent les valeurs d'identifiant. Par exemple, le
470 nom général de la liste ci-dessous évoque le seul identifiant défini pour un objet qui caractérise un
471 exemple de Service :

472

473

```
<rim:Service id="{EXAMPLE_SERVICE_ID}">
```

474 1.3.3 Constantes

475 Les valeurs de constante sont toujours imprimées en format Courier New font, peu importe si elles
476 sont établies par le présent document ou un document référencé.

477 1.3.4 Texte en caractères gras

478 On a recours à un texte en caractères gras dans les listes pour mettre en évidence les aspects
479 qui sont les plus pertinents par rapport aux questions abordées. La liste ci-dessous englobe un
480 exemple de valeur pour la place d'attribut de localisateur-de-contenu. Cet exemple doit être en
481 italique s'il s'agit de l'élément le plus important sur lequel le lecteur devrait diriger principalement
482 son attention.

```
483  
484 <rim:Slot name="urn:oasis:names:tc:ebxml-  
485 regrep:rim:RegistryObject:contentLocator">  
486 ...  
487 </rim:Slot>
```

488

489 1.3.5 Exemples de valeur

490 Les exemples de valeur sont en *italique*. La liste suivante comporte un exemple de valeur qui
491 s'applique à la place d'attribut de localisateur-de-contenu et qui figure en italique :

492

```
493 <rim:Slot name="urn:oasis:names:tc:ebxml-  
494 regrep:rim:RegistryObject:contentLocator">  
495 <rim:ValueList>  
496 <rim:Value>http://example.com/myschema.xsd</rim:Value>  
497 </rim:ValueList>  
498 </rim:Slot>
```

499

500 1.4 Conventions liées au schéma XML

501 La présente spécification utilise des documents de schéma conformes à XML Schema du W3C
502 [Schema1] ainsi qu'un texte normatif pour décrire la syntaxe et la sémantique des objets d'encodage XML
503 et des messages de protocole. En cas de différence entre les documents de schéma du registre ebXML
504 et les listes de schéma dans la spécification, les documents de schéma ont prépondérance sur les listes.
505 Il est important de signaler que dans certains cas, le texte normatif de la spécification impose des
506 contraintes qui vont au-delà de celles mentionnées dans les documents de schéma.

507 Dans la présente spécification, on se sert de préfixes traditionnels d'espace de nommage XML de
508 manière à ce qu'ils correspondent à leurs espaces de nommage respectifs, comme le démontre le
509 tableau ci-dessous. On agit ainsi peu importe si une déclaration d'espace de nommage figure ou non
510 dans les exemples pertinents. L'utilisation de préfixes d'espace de nommage dans les documents
511 d'instance n'est pas normative. Cependant, à des fins d'uniformité et d'intelligibilité, on DEVRAIT y avoir
512 recours dans de tels documents.

513 1.4.1 Schémas définis par le registre ebXML

514

Préfixe	Espace de nommage XML	Commentaires
rim:	urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0	Il s'agit de l'espace de nommage du <i>Modèle d'information du registre ebXML</i> [ebRIM]. Le préfixe s'élide généralement dans les éléments textuels relatifs à ce modèle.

Préfixe	Espace de nommage XML	Commentaires
rs:	urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0	Il s'agit de l'espace de nommage du registre ebXML qui définit les types de base des demandes et réponses propres aux services du registre [ebRS]. Le préfixe s'élide généralement dans les éléments textuels liés aux protocoles du registre ebXML.
query:	urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0	Il s'agit de l'espace de nommage qui concerne les interrogations particulières au registre ebXML, et dont on se sert dans les protocoles d'interrogation entre les applications et le Gestionnaire-d'interrogation [ebRS].
lcm:	urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0	Il s'agit de l'espace de nommage qui se rattache à la gestion du cycle de vie du registre ebXML, et qu'on utilise dans les protocoles de gestion de cycle de vie entre les applications et le Gestionnaire-de-cycle-de-vie [ebRS].
cms:	urn:oasis:names:tc:ebxml-regrep:xsd:cms:3.0	Il s'agit de l'espace de nommage qui est propre aux services de gestion du contenu du registre ebXML, et auquel on a recours dans les protocoles de gestion de contenu entre le registre et les services de gestion de contenu enfichable [ebRS].

515

516 1.4.2 Schémas utilisés par le registre ebXML

517

Préfixe	Espace de nommage XML	Commentaires
saml:	urn:oasis:names:tc:SAML:2.0:assertion	Il s'agit de l'espace de nommage qui s'applique à la déclaration SAML 2.0 [SAMLCore]. Le préfixe s'élide généralement dans les éléments textuels particuliers aux déclarations SAML.
samlp:	urn:oasis:names:tc:SAML:2.0:protocol	Il s'agit de l'espace de nommage qui se rattache aux protocoles SAML 2.0 [SAMLCore]. Le préfixe s'élide généralement dans les éléments textuels liés aux protocoles XML.
ecp:	urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp	Il s'agit de l'espace de nommage propre au profil de client/proxy de SAML 2.0. Cet espace est spécifié dans le présent document ainsi que dans un schéma [SAMLECP-xsd].
ds:	http://www.w3.org/2000/09/xmldsig#	Il s'agit de l'espace de nommage qui se rapporte à XML Signature [XMLSig].
xenc:	http://www.w3.org/2001/04/xmlenc#	Il s'agit de l'espace de nommage qui est associé au chiffrement XML [XMLEnc].

Préfixe	Espace de nommage XML	Commentaires
SOAP-ENV:	http://schemas.xmlsoap.org/soap/envelope	Il s'agit de l'espace de nommage qui concerne SOAP 1.1 [SOAP1.1].
paos:	urn:liberty:paos:2003-08	Il s'agit de l'espace de nommage qui est lié au PAOS (inversion de SOAP) de Liberty Alliance.
xsi:	http://www.w3.org/2001/XMLSchema-instance	Il s'agit de l'espace de nommage qui est défini dans la spécification XML Schema [Schema1] du W3C pour le balisage relatif aux schémas et figurant dans les instances XML.
wsse:	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	Il s'agit de l'espace de nommage qui est établi par la spécification Web Services Security: SOAP Message Security 1.0 [WSS-SMS]. Le registre se sert de cet espace pour garantir la transmission de messages SOAP.
wsu:	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	Il s'agit de l'espace de nommage qui est déterminé par la spécification Web Services Security: SOAP Message Security 1.0 [WSS-SMS]. Le registre a recours à cet espace pour garantir la transmission de messages SOAP.

518

519 1.5 Acteurs de registre

520 La présente section décrit les divers acteurs qui interagissent avec le registre.

Acteur	Description
Opérateur de registre	Il s'agit d'un organisme qui exploite un registre ebXML et en assure la disponibilité des services.
Administrateur de registre	Il s'agit d'un utilisateur privilégié du registre à qui il incombe d'exécuter les tâches administratives nécessaires à l'exploitation continue du registre. Cet utilisateur joue un rôle semblable à celui d'un « super-utilisateur », c'est-à-dire qu'il est autorisé à effectuer <i>n'importe quelle</i> action.
Utilisateur anonyme de registre	Il s'agit d'un utilisateur dont l'identité est inconnue du registre. Il possède des privilèges plutôt limités dans l'utilisation du registre.
Utilisateur inscrit	Il s'agit d'un utilisateur dont l'identité est connue du registre et qui est considéré par celui-ci comme un utilisateur autorisé.
Demandeur	Il s'agit d'un utilisateur qui présente un contenu ou des métadonnées au registre. Il DOIT être un utilisateur inscrit.
Application de registre	Il s'agit d'un programme informatique qui interagit avec le registre par le biais de protocoles propres à celui-ci.

521

522 **1.6 Cas d'utilisation du registre**

523 Le registre ebXML, une fois déployé, assure la prestation de services de gestion de contenu et de
524 métadonnées génériques. À ce titre, il soutient le recours à un ensemble ouvert et large de cas
525 d'utilisation. Voici quelques cas courants abordés par le registre ebXML :

- 526 • Registre de services Web : Favorise la publication, la gestion, la découverte et la réutilisation de
527 descriptions de services Web dans WSDL, ebXML, CPPA et d'autres applications;
 - 528 • Registre de vocabulaire contrôlé : Facilite la publication, la gestion, la découverte et la réutilisation de
529 vocabulaires contrôlés, dont les taxinomies, les listes de codes, les composants élémentaires ebXML,
530 le schéma XML et le schéma UBL;
 - 531 • Registre de processus d'affaires : Permet la publication, la gestion, la découverte et la réutilisation de
532 spécifications de processus d'affaires, comme ebXML BPSS et BPEL;
 - 533 • Référentiel de dossiers médicaux électroniques;
 - 534 • Référentiel de systèmes d'information géologique : Favorise le stockage de données relatives à ces
535 systèmes et provenant de capteurs.
- 536

537 **1.7 Architecture de registre**

538 Le diagramme ci-dessous donne un aperçu simplifié de l'architecture du registre ebXML.

539

Diagramme 1 : Aperçu simplifié de l'architecture du registre ebXML

541 **1.7.1 Applications de registre**

542 Une application de registre représente un programme informatique qui interagit avec un registre par le
543 biais de protocoles propres à celui-ci. Elle PEUT prendre la forme d'une interface utilisateur graphique
544 (GUI), d'une assistance-logiciel ou d'un agent. L'application de registre favorise généralement l'accès à un
545 registre au moyen du protocole SOAP 1.1 avec pièces jointes.

546 Une application de registre peut être exécutée sur un système d'application ou peut se traduire par un
547 point de service Web hébergé dans un serveur. Elle peut aussi être accessible par un navigateur. Dans
548 un cas ou dans l'autre, l'application de registre interagit avec le registre au moyen de protocoles propres à

548 celui-ci.

549 **1.7.1.1 API d'application**

550 Toute application de registre PEUT accéder directement à une interface de registre. De façon subsidiaire,
551 elle PEUT avoir recours à une API d'application de registre comme API Java pour registres XML (JAXR)
552 pour accéder au registre. Les API d'application comme [JAXR] offrent une convivialité pour la
553 programmation et sont, en général, particulières à un langage de programmation quelconque.

554 **1.7.2 Interfaces de services de registre**

555 Le registre ebXML regroupe les interfaces de services suivantes :

- 556 • Interface de Gestionnaire-de-cycle-de-vie : Présente une collection d'opérations en vue de la gestion
557 de bout en bout du cycle de vie des métadonnées et du contenu du registre. Ces opérations
558 renferment la publication, la mise à jour, l'approbation et la suppression de métadonnées et de
559 contenu;
- 560 • Interface de Gestionnaire-d'interrogation : Présente une collection d'opérations en vue de la
561 découverte et de l'extraction de métadonnées et de contenu du registre.

562 La description WSDL de l'interface de services de registre [RS-Interface-WSDL] donne une définition
563 abstraite (neutre par rapport aux protocoles) de cette interface en format WSDL.

564 **1.7.3 Interface de services : rattachements de protocole**

565 La présente spécification définit le rattachement de protocole concret suivant qui s'applique aux interfaces
566 de services génériques du registre ebXML :

- 567 • Rattachement de protocole SOAP : Permet à une application de registre d'accéder au registre par le
568 protocole SOAP 1.1 avec pièces jointes. La description WSDL des rattachements de services de
569 registre [RS-Bindings-WSDL] définit le rattachement des interfaces de services génériques de registre
570 vers le protocole SOAP en format WSDL;
- 571 • Rattachement de protocole HTTP : Permet à une application de navigateur d'accéder au registre au
572 moyen du protocole HTTP 1.1.

573 **1.7.4 Authentification et autorisation**

574 Toute application de registre DEVRAIT être authentifiée par le registre pour que l'on puisse établir
575 l'identité qui s'y rattache. Il s'agit généralement de l'identité de l'utilisateur propre à l'application de registre.
576 Une fois cette identité révélée, le registre DOIT vérifier l'autorisation et le contrôle d'accès avant
577 d'autoriser le traitement d'une demande de l'application de registre.

578 **1.7.5 Registre de métadonnées et référentiel de contenu**

579 Un registre ebXML désigne un registre de métadonnées ainsi qu'un référentiel de contenu. L'implantation
580 d'un registre ebXML se fait habituellement par le recours à une forme quelconque de stockage
581 permanent, comme une base de données; c'est ainsi que l'on procède au stockage des métadonnées et
582 du contenu du registre. Sur le plan de l'architecture, le registre est différent du référentiel. L'accès au
583 registre ainsi qu'au référentiel s'effectue toutefois au moyen des opérations déterminées par les interfaces
584 de services de registre.

2 Protocoles de registre

585

586 Le chapitre 2 présente les protocoles de registre qui sont gérés par les interfaces de services de registre.
587 De manière précise, il donne un aperçu des formes générales d'échange de messages qui s'appliquent à
588 l'ensemble des protocoles de registre.

2.1 Demandes et réponses

589

590 Les messages propres aux demandes et réponses de registre découlent des types communs décrits
591 dans le schéma XML du [RR-RS-XSD]. L'application de registre envoie à un registre un élément
592 provenant du **Type-de-demande-de-registre**, tandis que le registre génère un élément qui adhère au
593 **Type-de-réponse-de-registre** ou en tire son origine, comme l'illustre le diagramme ci-dessous.

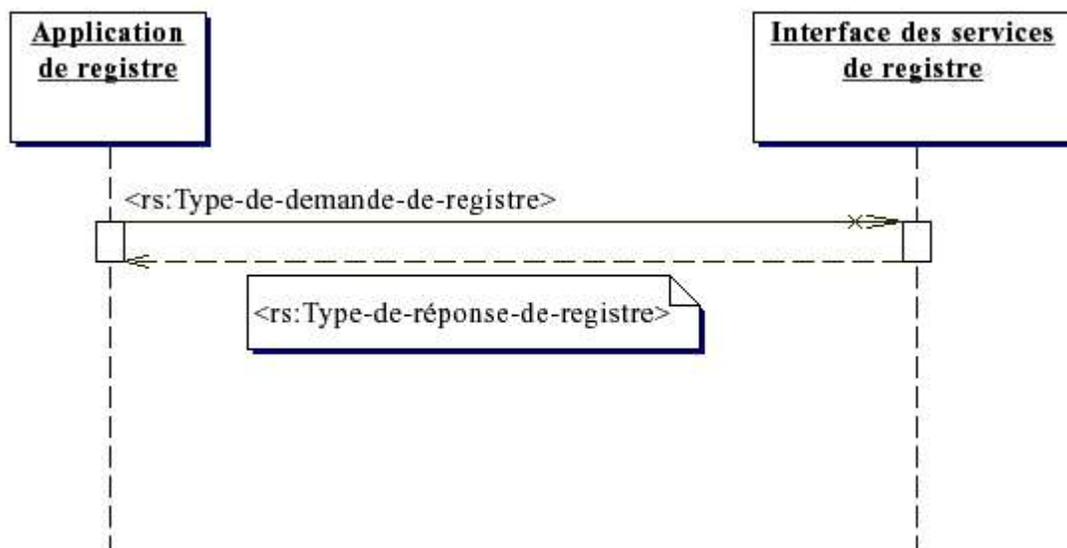


Diagramme 2 : Demandes et réponses des protocoles de registre

595

596 Dans la présente section, les mentions textuelles des éléments et des types sont indiquées à l'aide d'un
597 préfixe d'espace de nommage. Les conventions de préfixe d'espace de nommage sont définies dans
598 l'introduction.

599 Chaque demande de registre est atomique : elle est fructueuse ou bien elle échoue dans son intégralité.
600 En cas de réussite, le registre envoie à l'application une Réponse-de-registre qui confirme la « réussite »
601 de l'opération. En cas d'échec, il lui fait plutôt parvenir une Réponse-de-registre qui comporte un message
602 d'« échec ». S'il y a une réponse immédiate à une demande asynchrone, le registre transmet à
603 l'application une Réponse-de-registre qui comprend l'énoncé « non disponible ». Il y a échec si au moins
604 une condition d'erreur est signalée pendant le traitement des objets inscrits. Les messages
605 d'avertissement n'entraînent pas l'échec de la demande.

2.1.1 Type-de-demande-de-registre

606

607 On se sert du Type-de-demande-de-registre comme type de base commun pour l'ensemble des
608 messages de demande de registre.

2.1.1.1 Syntaxe

609

```
610 <complexType name="RegistryRequestType">
611   <sequence>
612     <!-- every request may be extended using Slots. -->
```

```

613     <element maxOccurs="1" minOccurs="0" name="RequestSlotList"
614 type="rim:SlotListType"/>
615   </sequence>
616   <attribute name="id" type="anyURI" use="required"/>
617   <!--Comment may be used by requestor to describe the request. Used in
618   VersionInfo.comment-->
619   <attribute name="comment" type="string" use="optional"/>
620 </complexType>
621 <element name="RegistryRequest" type="tns:RegistryRequestType"/>

```

622 2.1.1.2 Paramètres

- 623 ▪ **commentaire** : Ce paramètre permet au demandeur de préciser une valeur de place
624 d'attribut qui décrit l'action découlant de la demande. Il est utilisé par la caractéristique
625 « contrôle de version géré par registre » du registre.
- 626 ▪ **identifiant** : Ce paramètre spécifie un identifiant de demande dont la réponse homologue
627 se sert pour assurer une corrélation entre la demande et elle-même. On PEUT également
628 avoir recours à ce paramètre pour établir une corrélation entre une demande et une autre
629 demande connexe. La valeur du paramètre identifiant DOIT se soumettre aux mêmes
630 contraintes que la valeur de l'attribut identifiant en ce qui concerne le type <rim:Type-
631 d'identifiable>.
- 632 ▪ **Liste-de-places-d'attribut-de-demande** : Ce paramètre désigne expressément une
633 collection d'instances de Place-d'attribut. Un Type-de-demande-de-registre PEUT
634 englober des Places-d'attribut sous la forme d'un mécanisme d'extensibilité qui permet
635 d'ajouter des attributs à la demande en tant que Places-d'attribut. Les registres qui ne
636 peuvent pas accueillir de Place-d'attribut DOIVENT discrètement faire abstraction du
637 recours à des places d'attribut propres à l'implantation d'un registre. Aussi, ce même
638 recours PEUT ne pas être interfonctionnel dans les implantations de registre.
639

640 2.1.1.3 Renvois

641 Toutes les Demandes-de-registre renvoient une réponse qui provient du type de base commun propre au
642 Type-de-réponse-de-registre.

643 2.1.1.4 Exceptions

644 Les exceptions suivantes s'appliquent à l'ensemble des demandes de protocole de registre :

- 645 ▪ **Exception-d'autorisation** : Cette exception démontre que le demandeur a tenté
646 d'effectuer une opération qu'il n'était pas autorisé à faire;
- 647 ▪ **Exception-de-demande-invalide** : Cette exception signale que le demandeur a tenté
648 d'exécuter une opération qui était invalide sur le plan de la sémantique;
- 649 ▪ **Exception-de-validation-de-signature** : Cette exception révèle que la signature relative
650 à la demande n'a pu être validée;
- 651 ▪ **Exception-de-délai-d'attente** : Cette exception indique que le délai de traitement de la
652 demande a excédé une limite précise par rapport au registre;
- 653 ▪ **Exception-de-capacité-non-gérée** : Cette exception prouve que le registre n'a pu
654 accueillir la capacité nécessaire à l'exécution de la demande.

655 En plus des exceptions ci-dessus, certaines autres sont définies par le [WSS-SMS] et DOIVENT être
656 renvoyées par une demande de protocole de registre si des erreurs se produisent pendant le traitement
657 de l'élément en-tête SOAP <wsse:Security>.

658 2.1.2 Demande-de-registre

659 Une Demande-de-registre constitue un élément dont le type de base désigne le Type-de-demande-de-
660 registre. Il n'y a aucun autre élément ou attribut à part ceux décrits dans celui-ci. Tout registre PEUT se

661 servir de l'élément de Demande-de-registre pour gérer des demandes de registre propres à l'implantation.

662 2.1.3 Type-de-réponse-de-registre

663 Le Type-de-réponse-de-registre est utilisé à titre de type de base commun pour l'ensemble des réponses
664 de registre.

665 2.1.3.1 Syntaxe

```
666 <complexType name="RegistryResponseType">
667   <sequence>
668     <!-- every response may be extended using Slots. -->
669     <element maxOccurs="1" minOccurs="0" name="ResponseSlotList"
670 type="rim:SlotListType"/>
671     <element minOccurs="0" ref="tns:RegistryErrorList"/>
672   </sequence>
673   <attribute name="status" type="rim:referenceURI" use="required"/>
674   <!-- id is the request id for the request for which this is a
675 response -->
676   <attribute name="requestId" type="anyURI" use="optional"/>
677 </complexType>
678 <element name="RegistryResponse" type="tns:RegistryResponseType"/>
```

679 2.1.3.2 Paramètres

- 680 ▪ **statut** : On se sert de l'attribut statut pour déterminer le statut de la demande. La valeur
681 de cet attribut DOIT se traduire par une référence à un Nœud-de-classification dans le
682 Schème-de-classification canonique de Type-de-statut-de-réponse, comme le décrit le
683 [ebRIM]. Tout registre DOIT pouvoir accueillir les types de statut, tel que le définit le
684 Schème-de-classification canonique de Type-de-statut-de-réponse. On peut procéder à
685 l'extension de ce schème en y ajoutant des Nœuds-de-classification.
686 On a établi les valeurs canoniques suivantes pour le Schème-de-classification de Type-
687 de-statut-de-réponse :
 - 688 • **Réussite** – Ce statut illustre la réussite de la demande;
 - 689 • **Échec** – Ce statut signifie que la demande a avorté. Dans un tel cas, au moins une
690 erreur DOIT être inscrite dans la Liste-d'erreurs-de-registre ou renvoyée sous la
691 forme d'une erreur SOAP;
 - 692 • **Non disponible** – Ce statut signifie que la réponse n'est pas encore disponible. Cette
693 situation peut prévaloir si le Type-de-réponse-de-registre désigne une réponse
694 immédiate à une demande asynchrone dont la réponse réelle n'est pas encore
695 disponible.
- 696 ▪ **identifiant-de-demande** : Ce paramètre met en évidence l'identifiant de la demande
697 pour laquelle il s'agit d'une réponse. Il s'apparie à la valeur de l'attribut identifiant du Type-
698 de-demande-de-registre homologue.
- 699 ▪ **Liste-de-places-d'attribut-de-réponse** : Ce paramètre désigne expressément une
700 collection d'instances de Place-d'attribut. Un Type-de-réponse-de-registre PEUT englober
701 des Places-d'attribut sous la forme d'un mécanisme d'extensibilité qui permet d'ajouter
702 des attributs dynamiques en tant que Places-d'attribut. Les applications de registre qui ne
703 peuvent pas accueillir de Place-d'attribut DOIVENT discrètement faire abstraction du
704 recours à des places d'attribut propres à l'implantation d'un registre. Aussi, ce même
705 recours PEUT ne pas être interfonctionnel dans les implantations de registre.
- 706 ▪ **Liste-d'erreurs-de-registre** : Ce paramètre fait référence à une collection optionnelle
707 d'éléments d'Erreur-de-registre et s'applique si au moins une erreur est détectée pendant
708 le traitement d'une demande par le registre. Le paramètre est décrit plus en détail dans la
709 section 6.9.4.

710 2.1.4 Réponse-de-registre

711 La Réponse-de-registre représente un élément dont le type de base est caractérisé par le Type-de-
712 réponse-de-registre. Il n'y a aucun autre élément ou attribut à part ceux décrits dans celui-ci. De
713 nombreux protocoles de registre se servent de la Réponse-de-registre comme réponse.

714 2.1.5 Liste-d'erreurs-de-registre

715 Une Liste-d'erreurs-de-registre se traduit par une collection optionnelle d'éléments d'Erreur-de-registre et
716 s'applique si au moins une erreur est détectée pendant le traitement d'une demande par le registre.

717 2.1.5.1 Syntaxe

```
718 <element name="RegistryErrorList">  
719   <complexType>  
720     <complexContent>  
721       <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
722         <sequence>  
723           <element ref="rs:RegistryError" maxOccurs="unbounded"/>  
724         </sequence>  
725         <attribute name="highestSeverity" type="rim:referenceURI" />  
726       </restriction>  
727     </complexContent>  
728   </complexType>  
729 </element>
```

730 2.1.5.2 Paramètres

- 731 ▪ *le-plus-grave* : Ce paramètre désigne le Type-d'erreur propre à l'Erreur-de-registre la plus
732 grave dans la Liste-d'erreurs-de-registre. Les valeurs particulières à ce paramètre sont
733 définies par Type-d'erreur dans la section .
 - 734 ▪ *Erreur-de-registre* : Une Liste-d'erreurs-de-registre comporte au moins une Erreur-de-
735 registre. Ce type d'erreur fait référence à un message d'erreur ou d'avertissement qui
736 apparaît pendant le traitement d'une demande par le registre. L'Erreur-de-registre est
737 décrite dans la section 2.1.6.
- 738

739 2.1.6 Erreur-de-registre

740 Une Erreur-de-registre révèle un message d'erreur ou d'avertissement qui apparaît pendant le traitement
741 d'une demande par le registre.

742 2.1.6.1 Syntaxe

```
743 <element name="RegistryError">  
744   <complexType>  
745     <simpleContent>  
746       <extension base="string">  
747         <attribute name="codeContext" type="string" use="required"/>  
748         <attribute name="errorCode" type="string" use="required"/>  
749         <attribute default="urn:oasis:names:tc:ebxml-  
750 regrep:ErrorSeverityType:Error" name="severity" type="rim:referenceURI" />  
751       >  
752         <attribute name="location" type="string" use="optional"/>  
753       </extension>  
754     </simpleContent>  
755   </complexType>  
756 </element>
```

757 **2.1.6.2 Paramètres**

- 758
- 759
- 760
- 761
- 762
- 763
- 764
- 765
- 766
- 767
- 768
- 769
- 770
- 771
- 772
- 773
- 774
- 775
- 776
- 777
- 778
- 779
- 780
- 781
- 782
- 783
- 784
- *contexte-de-code* : Cet attribut définit une chaîne qui révèle un texte contextuel fournissant des détails additionnels au code-d'erreur. Par exemple, si celui-ci se traduit par une Exception-de-demande-invalide, le contexte-de-code PEUT justifier l'invalidité de la demande.
 - *code-d'erreur* : Cet attribut détermine une chaîne qui signale l'erreur détectée. Les implantations DOIVENT établir l'attribut en fonction de l'exception ou de l'erreur, comme le définit la présente spécification (par exemple, l'Exception-de-demande-invalide).
 - *gravité* : Cet attribut précise le degré de gravité de l'erreur détectée. La valeur de l'attribut gravité DOIT citer un Nœud-de-classification dans le Schème-de-classification canonique de Type-de-gravité-des-erreurs, comme le décrit le [ebRIM]. Tout registre DOIT pouvoir accueillir les types de gravité des erreurs, conformément au Schème-de-classification canonique de Type-de-gravité-des-erreurs. On peut procéder à l'extension de ce schème en y ajoutant des Nœuds-de-classification.
On a défini les valeurs canoniques suivantes pour le Schème-de-classification de Type-de-gravité-des-erreurs :
 - **Erreur** – Une Erreur désigne une erreur fatale détectée par le registre pendant le traitement d'une demande. Tout registre DOIT renvoyer un statut d'échec dans la Réponse-de-registre si une erreur a été détectée au cours du traitement d'une demande.
 - **Avertissement** – Un Avertissement constitue une erreur non fatale détectée par le registre pendant le traitement d'une demande. Tout registre DOIT renvoyer un statut de réussite dans la Réponse-de-registre si seuls des avertissements ont été émis au cours du traitement d'une demande et si aucune erreur n'a été détectée.
 - *localisation* : Cet attribut met en évidence une chaîne qui spécifie l'endroit dans le code où l'erreur s'est produite. Les implantations DEVRAIENT faire ressortir la piste de journalisation, le module de code ou le numéro de ligne propres à l'endroit dans le code où l'erreur a été détectée.

3 Rattachement de protocole SOAP

785

786 Le chapitre 3 définit le rattachement de protocole SOAP pour les interfaces de services du
787 registre ebXML. Ce rattachement favorise l'accès au registre au-delà du protocole SOAP 1.1 avec pièces
788 jointes. Le rattachement de protocole SOAP est décrit dans son intégralité dans les fichiers de
789 description WSDL suivants :

- 790 • Interfaces de service du registre ebXML : Définition abstraite [RR-INT-WSDL];
- 791 • Interfaces de service du registre ebXML : Rattachement du protocole SOAP [RR-SOAPB-WSDL];
- 792 • Interfaces de service du registre ebXML : Service SOAP [RR-SOAPS-WSDL].

3.1 Interfaces de service du registre ebXML : Définition abstraite

793

794 Dans le [RR-INT-WSDL], chaque interface de services de registre s'apparie à un type-de-port WSDL
795 abstrait comme suit :

- 796 • On définit un type-de-port pour chaque interface de services :

797

```
798 <portType name="QueryManagerPortType">  
799 ...  
800 </portType>  
801 <portType name="LifeCycleManagerPortType">  
802 ...  
803 </portType>
```

804

- 805 • Dans chaque type-de-port, on établit une opération pour chaque protocole géré par l'interface de
806 services :

807

```
808 <portType name="QueryManagerPortType">  
809 <operation name="submitAdhocQuery">  
810 ...  
811 </operation>  
812 </portType>
```

813

- 814 • Dans chaque opération, on détermine pour le protocole connexe le message de demande et de
815 réponse en tant qu'intrant et extrant de l'opération :

```
816 <portType name="QueryManagerPortType">  
817 <operation name="submitAdhocQuery">  
818 <input message="tns:msgAdhocQueryRequest" />  
819 <output message="tns:msgAdhocQueryResponse" />  
820 </operation>  
821 </portType>
```

822

- 823 • Pour chaque message utilisé dans le cadre d'une opération, on établit un élément de message qui cite
824 l'élément correspondant au message de demande ou de réponse propre aux protocoles de registre.
825 Ce message provient du schéma XML et s'applique à l'interface de services de registre ([RR-LCM-
826 XSD] et [RR-QM-XSD]) :

827

```
828 <message name="msgAdhocQueryRequest">  
829 <part element="query:AdhocQueryRequest"  
830 name="partAdhocQueryRequest" />  
831 </message>  
832 <message name="msgAdhocQueryResponse">  
833 <part element="query:AdhocQueryResponse"  
834 name="partAdhocQueryResponse" />
```

835

```
</message>
```

836 3.2 Rattachement du protocole SOAP propre aux interfaces de 837 services du registre ebXML

838 Dans le [RR-SOAPB-WSDL], on définit un rattachement de protocole SOAP pour les interfaces de
839 services de registre, comme suit :

- 840 • Pour chaque type-de-port qui correspond à une interface de services de registre et est établi dans le
841 [RR-INT-WSDL], on détermine un élément de <rattachement> appelé « rattachement <Nom-
842 d'interface-de-service> »;
- 843 • L'élément de <rattachement> cite le type-de-port défini dans le [RR-INT-WSDL] par le biais de son
844 attribut type;
- 845 • L'élément d'extension <SOAP:rattachement> se sert du style « document »;
- 846 • On établit un élément d'opération pour chaque protocole défini par rapport à l'interface de services. Le
847 nom de l'opération se rattache au message de demande de protocole;
- 848 • L'élément d'extension <SOAP:opération> dispose d'éléments d'intrant et d'extrant qui, eux, renferment
849 des éléments <SOAP:corps> caractérisés par le recours=« littéral ».

850

```
851 <binding name="QueryManagerBinding"  
852 type="interfaces:QueryManagerPortType">  
853 <soap:binding style="document"  
854 transport="http://schemas.xmlsoap.org/soap/http"/>  
855 <operation name="submitAdhocQuery">  
856 <soap:operation soapAction="urn:oasis:names:tc:ebxml-  
857 regrep:wSDL:registry:bindings:3.0:QueryManagerPortType#submitAdhocQuery"/  
858 >  
859 <input>  
860 <soap:body use="literal"/>  
861 </input>  
862 <output>  
863 <soap:body use="literal"/>  
864 </output>  
865 </operation>  
866 </binding>
```

867

868 3.3 Interfaces de service du registre ebXML : Service SOAP

869 Dans le [RR-SOAPS-WSDL], on présente un modèle non normatif de service WSDL qui est caractérisé
870 par l'utilisation du rattachement de protocole SOAP à partir des interfaces de services de registre, comme
871 suit :

- 872 • Un seul élément de service définit le service SOAP concret du registre ebXML. Le modèle de service a
873 comme nom « Service-SOAP-du-registre-ebXML »;
- 874 • L'élément de service comprend une définition de port, où chaque port se rattache à l'une des
875 interfaces de services déterminées pour le registre. Chaque port englobe un URL HTTP pour être
876 accessible et est spécifié par l'attribut localisation de l'élément <SOAP:adresse>;
- 877 • Chaque définition de port cite également un élément du rattachement de protocole SOAP décrit dans
878 la section antérieure.

879

```
880 <service name="ebXMLRegistrySOAPService">  
881 <port binding="bindings:QueryManagerBinding" name="QueryManagerPort">  
882 <soap:address location="http://your.server.com/soap"/>  
883 </port>  
884 <port binding="bindings:LifecycleManagerBinding"  
885 name="LifecycleManagerPort">
```

886
887
888
889

```
<soap:address location="http://your.server.com/soap"/>  
</port>  
</service>
```

890 3.4 Appariement entre les exceptions et les erreurs SOAP

891 Les protocoles de registre définis dans la présente spécification englobent la spécification liée aux
892 exceptions que tout registre DOIT renvoyer si certaines conditions exceptionnelles font surface pendant le
893 traitement du message de demande de protocole. Tout registre DOIT renvoyer les exceptions qui figurent
894 dans les messages de protocole de registre à titre d'erreurs SOAP, comme il est décrit dans la présente
895 section. De plus, les registres DOIVENT être conformes au [WSI-BP] au moment de la génération d'une
896 erreur SOAP. Enfin, ils NE DOIVENT PAS signer les messages d'erreur SOAP qu'ils renvoient.

897 Le tableau ci-dessous donne des renseignements détaillés sur la façon dont un registre DOIT assurer
898 l'appariement entre les exceptions et les erreurs SOAP.
899

Élément d'erreur SOAP	Description	Exemple
code-d'erreur	Il DOIT y avoir un code-d'erreur et il DOIT s'agir du nom de l'exception définie par le préfixe URN : urn:oasis:names:tc:ebxml-regrep:rs:exception:	<i>urn:oasis:names:tc:ebxml-regrep:rs:exception:ObjectNotFoundException</i>
chaîne-d'erreur	Il DOIT y avoir une chaîne-d'erreur, et celle-ci DEVRAIT présenter des renseignements qui précisent la nature de l'exception.	<i>Object with id urn:freebxml:registry:demoDB:Extrinsic Object:zeusDescription not found in registry</i>
détail	Il DOIT y avoir au moins un élément de détail. Celui-ci DEVRAIT faire ressortir la piste de journalisation, le module de code ou le numéro de ligne à l'endroit où l'exception a été détectée dans le code. Si d'autres exceptions sont imbriquées dans cette exception, elles DEVRAIENT figurer dans le registre à titre d'éléments de détail imbriqués dans l'élément de détail de premier niveau.	
acteur-d'erreur	Il DOIT y avoir au moins un acteur-d'erreur. Le premier acteur-d'erreur DOIT constituer l'URL de base du registre.	<i>http://example.server.com:8080/oma r/registry</i>

Tableau 1 : Appariement entre une exception de registre et une erreur SOAP

4 Rattachement de protocole HTTP

Le chapitre 4 définit le rattachement de protocole HTTP qui s'applique aux interfaces abstraites de services du registre ebXML. Ce rattachement favorise l'accès au registre au-delà du protocole HTTP 1.1.

L'interface HTTP offre de nombreuses méthodes d'accès aux Objets-de-registre et aux Items-de-référentiel par le biais du protocole HTTP. Les voici :

- URL fondé sur l'encodage d'un RPC : Permet à une application d'accéder à des objets au moyen d'un URL fondé sur l'encodage d'un appel de procédure à distance (RPC) vers une interface de registre à titre de demande de protocole HTTP;
- URL défini par le demandeur : Permet à une application d'accéder à des objets par un URL défini par le demandeur;
- URL par chemin d'accès : Permet à une application d'accéder à des objets en se servant d'un URL par chemin d'accès qui découle de l'appartenance de l'objet à une hiérarchie de Paquets-de-registre.

Chacune des méthodes ci-dessus comporte ses avantages et ses inconvénients. De plus, elles conviennent toutes mieux à des cas d'utilisation différents, comme l'illustre le tableau suivant :

Méthode d'accès HTTP	Avantages	Inconvénients
URL fondé sur l'encodage d'un RPC	<ul style="list-style-type: none">• L'URL est constant et déterministe.• Le demandeur n'a pas à attribuer un URL de manière explicite.	<ul style="list-style-type: none">• L'URL entraîne un long processus et n'est pas facile à retenir.
URL défini par le demandeur	<ul style="list-style-type: none">• L'URL est très convivial.• Le demandeur peut attribuer n'importe quel URL.• L'URL est constant et déterministe.	<ul style="list-style-type: none">• Le demandeur doit attribuer un URL de manière explicite.• Le registre doit renfermer des ressources additionnelles.
URL par chemin d'accès	<ul style="list-style-type: none">• Le demandeur n'a pas à attribuer un URL de manière explicite.• Il s'agit d'un URL intuitif qui s'inspire d'une métaphore bien connue de fichiers et de dossiers.	<ul style="list-style-type: none">• L'URL n'est PAS constant et déterministe.• Il faut classer les objets de façon qu'ils constituent des éléments de paquet-de-registre.

Tableau 2 : Comparaison des méthodes d'accès HTTP

4.1 URL fondé sur l'encodage d'un RPC

La méthode de l'URL fondé sur l'encodage d'un RPC qui est utilisée pour l'interface HTTP assure l'appariement entre le protocole HTTP et les opérations définies par les interfaces abstraites de registre. Cet appariement s'effectue au moyen d'un style d'appel de procédure à distance. Cette méthode établit la façon dont on se sert des paramètres URL pour déterminer les paramètres d'interface, de méthode et d'appel nécessaires à l'appel d'une opération à partir d'une interface de registre, comme l'interface de Gestionnaire-d'interrogation.

La méthode de l'URL fondé sur l'encodage d'un RPC établit également la manière dont on a recours à une réponse HTTP pour transmettre la réponse générée par l'opération stipulée dans la demande.

926 **4.1.1 Paramètres URL standard**

927 Le tableau ci-dessous présente les paramètres URL que peuvent accueillir les URL fondés sur l'encodage
 928 d'un RPC. Tout registre PEUT implanter aussi d'autres paramètres URL. Il est important de mentionner
 929 que les noms de paramètre URL DOIVENT être traités par le registre de manière insensible à la casse,
 930 alors que les valeurs de paramètre DOIVENT l'être de façon sensible à la casse.

Paramètre URL	Requis	Description	Exemple
interface	OUI	Définit l'interface de services qui représente la cible de la demande.	Gestionnaire-d'interrogation
méthode	OUI	Définit la méthode (opération) comprise dans l'interface qui constitue la cible de la demande.	Obtenir-l'Objet-de-registre
paramètre-<clé>	NON	Définit les paramètres cités qui doivent faire l'objet d'un appel de méthode. Il est important de signaler que certaines méthodes nécessitent le recours à des paramètres particuliers.	param-id= <i>urn:freebxml:registry:demoD B:ExtrinsicObject:zeusDescri ption</i>

Tableau 3 : Paramètres URL standard

931 **4.1.2 Rattachement de Gestionnaire-d'interrogation**

932 Tout registre DOIT pouvoir accueillir un rattachement HTTP propre à l'URL fondé sur l'encodage d'un
 933 RPC, et ce rattachement DOIT être lié à l'interface de services de Gestionnaire-d'interrogation. Pour faire
 934 de l'interface de Gestionnaire-d'interrogation sa cible, le paramètre *interface* de l'URL DOIT constituer un
 935 « Gestionnaire-d'interrogation ». De plus, les paramètres URL ci-dessous sont définis par l'interface HTTP
 936 de Gestionnaire-d'interrogation.

937

Méthode	Paramètre	Valeur de renvoi	Type de demande HTTP
Obtenir-l'Objet-de-registre	identifiant	Il s'agit de l'Objet-de-registre qui s'apparie à l'identifiant précisé.	GET (obtenir)
Obtenir-l'Item-de-référentiel	identifiant	Il s'agit de l'Item-de-référentiel qui s'apparie à l'identifiant spécifié. On doit mentionner qu'un Item-de-référentiel peut se traduire par un contenu arbitraire (par exemple, une image GIF).	GET (obtenir)

Tableau 4 : URL fondé sur l'encodage d'un RPC : méthodes liées au Gestionnaire-d'interrogation

938

939 Il est important de spécifier que dans les exemples ci-dessous, on a omis d'insérer les déclarations
 940 relatives à l'espace de nommage afin de préserver de l'espace. Il faut aussi mentionner que certaines
 941 lignes peuvent être regroupées en raison d'un manque d'espace.

942 **4.1.2.1 Exemple de demande Obtenir-l'Objet-de-registre**

943 L'exemple ci-dessous illustre une demande Obtenir-l'Objet-de-registre.

944

```
945 GET /http?interface=QueryManager&method=getRegistryObject&param-  
946 id= urn:freebxml:registry:demoDB:ExtrinsicObject:zeusDescription  
947 HTTP/1.1
```

948

949 **4.1.2.2 Exemple de réponse Obtenir-l'Objet-de-registre**

950 L'exemple ci-dessous démontre un Objet-extrinsèque, qui constitue une sous-classe concrète d'Objet-de-
951 registre renvoyée comme réponse à un appel de méthode Obtenir-l'Objet-de-registre.

952

```
953 HTTP/1.1 200 OK  
954 Content-Type: text/xml  
955 Content-Length: 555  
956  
957 <?xml version="1.0"?>  
958 <ExtrinsicObject  
959   id =  
960   "urn:freebxml:registry:demoDB:ExtrinsicObject:zeusDescription"  
961   objectType="{OBJECT_TYPE}">  
962   ...  
963 </ExtrinsicObject>
```

964

965 **4.1.2.3 Exemple de demande Obtenir-l'Item-de-référentiel**

966 L'exemple ci-dessous montre cette fois une demande Obtenir-l'Item-de-référentiel.

967

```
968 GET /http?interface=QueryManager&method=getRepositoryItem&param-  
969 id= urn:freebxml:registry:demoDB:ExtrinsicObject:zeusDescription  
970 HTTP/1.1
```

971

972 **4.1.2.4 Exemple de réponse Obtenir-l'Item-de-référentiel**

973 Dans l'exemple ci-dessous, on présume que l'item de référentiel constitue un Profil de protocole de
974 collaboration (CPP), comme le définit la Spécification de Profil et d'Entente de protocole de
975 collaboration ebXML [ebCPP]. L'item de référentiel pourrait renvoyer n'importe quel type de contenu (par
976 exemple, une image GIF).

977

```
978 HTTP/1.1 200 OK  
979 Content-Type: text/xml  
980 Content-Length: 555  
981  
982 <?xml version="1.0"?>  
983 <CollaborationProtocolProfile>  
984   ...  
985 </CollaborationProtocolProfile>
```

986

987 **4.1.3 Interface HTTP liée au Gestionnaire-de-cycle-de-vie**

988 Le mécanisme de l'URL fondé sur l'encodage d'un RPC qui découle du rattachement de protocole HTTP

989 ne peut pas accueillir l'interface de Gestionnaire-de-cycle-de-vie parce que les opérations qui se
990 rattachent à celle-ci nécessitent HTTP-POST, qui est déjà géré par le rattachement de protocole SOAP.

991 **4.2 URL défini par le demandeur**

992 Tout demandeur PEUT déterminer un ensemble allant d'aucun à plusieurs URL définis par le demandeur
993 dans le cas d'un Objet-de-registre ou d'un Item-de-référentiel. Les applications PEUVENT ensuite se
994 servir de ces URL pour accéder à l'objet en question au moyen de la demande GET (obtenir) du
995 protocole HTTP. Les URL définis par le demandeur constituent une solution de rechange aux URL fondés
996 sur l'encodage d'un RPC établis par le rattachement de protocole HTTP pour l'interface de Gestionnaire-
997 d'interrogation. L'avantage des URL définis par le demandeur est qu'il est possible d'accéder aux objets
998 par le biais d'un URL efficace et facile à retenir pour l'utilisateur. Par contre, pour avoir recours à un
999 URL défini par le demandeur, le demandeur doit le spécifier. De plus, cet URL nécessite la présence dans
1000 le registre de ressources de stockage additionnelles.

1001 Les exemples ci-dessous permettent de comparer l'URL défini par le demandeur avec l'URL défini par le
1002 rattachement de protocole HTTP pour l'interface de Gestionnaire-d'interrogation.

1003 On trouve ci-dessous un exemple d'URL défini par le rattachement de protocole HTTP pour l'interface de
1004 Gestionnaire-d'interrogation. Cet URL permet d'accéder à un Objet-de-registre qui se traduit par un Objet-
1005 extrinsèque décrivant une image GIF.

```
1006  
1007  
1008 http://localhost:8080/ebxmlrr/registry/rest/?interface=QueryManager&metho  
1009 d=getRegistryObject&param-  
1010 id=urn:freebxml:registry:demoDB:ExtrinsicObject:zeusDescription  
1011
```

1012
1013 Il est possible d'accéder au même Objet-de-registre (Objet-extrinsèque) par le biais de l'URL défini par le
1014 demandeur suivant :

```
1015  
1016  
1017 http://localhost:8080/ebxmlrr/registry/rest/pictures/nikola/zeus.xml  
1018
```

1019
1020 On trouve ci-dessous un exemple d'URL défini par le rattachement de protocole HTTP pour l'interface de
1021 Gestionnaire-d'interrogation. Cet URL favorise l'accès à un item de référentiel qui constitue une
1022 image GIF.

```
1023  
1024  
1025 http://localhost:8080/ebxmlrr/registry/rest/?interface=QueryManager&metho  
1026 d=getRepositoryItem&param-  
1027 id=urn:freebxml:registry:demoDB:ExtrinsicObject:zeusDescription  
1028
```

1029
1030 On peut accéder au même item de référentiel au moyen de l'URL défini par le demandeur suivant :

```
1031  
1032  
1033 http://localhost:8080/ebxmlrr/registry/rest/pictures/nikola/zeus.jpg  
1034
```

1035

1036 **4.2.1 Syntaxe de l'URL défini par le demandeur**

1037 Tout demandeur DOIT spécifier un URL défini par le demandeur en le désignant comme un suffixe d'URL
1038 qui se rattache à l'URL de base du registre. Le suffixe de l'URL propre à un URL défini par le demandeur
1039 DOIT être unique dans l'ensemble des URL de ce genre qui s'appliquent à tous les objets d'un registre.

1040 Voici l'illustration de l'utilisation des différents URL :

- 1041 • **URL de base de registre** : <http://localhost:8080/ebxml/registry>
- 1042 • **Préfixe implicite d'URL pour HTTP** : <http://localhost:8080/ebxml/registry/http>
- 1043 • **Suffixe d'URL défini par le demandeur** : /pictures/nikola/zeus
- 1044 • **URL complet** : <http://localhost:8080/ebxmlrr/registry/http/pictures/nikola/zeus>

1045 **4.2.2 Attribution d'un URL à un Objet-de-registre**

1046 Tout demandeur PEUT attribuer au moins un URL défini par le demandeur à un Objet-de-registre.

1047 Le demandeur PEUT agir ainsi en utilisant une place-d'attribut canonique dans l'Objet-de-registre. Celle-ci
1048 a l'appellation suivante :

1049

1050

```
1051 urn:oasis:names:tc:ebxml-regrep:rim:RegistryObject:locator
```

1052

1053 Chaque valeur de la collection propre à la place-d'attribut désigne expressément un suffixe de l'URL défini
1054 par le demandeur pour l'Objet-de-registre en question. Le registre DOIT renvoyer l'Objet-de-registre au
1055 moment où l'application HTTP transmet une demande GET (obtenir) de HTTP dont l'URL s'apparie à
1056 n'importe quel URL mentionné dans la place-d'attribut localisation (le cas échéant) de cet Objet-de-
1057 registre.

1058 **4.2.3 Attribution d'un URL à un Item-de-référentiel**

1059 Tout demandeur PEUT attribuer au moins un URL défini par le demandeur à un Item-de-référentiel.

1060 Le demandeur PEUT agir ainsi en ayant recours à une place-d'attribut canonique dans l'Objet-extrinsèque
1061 relatif à l'Item-de-référentiel. Cette place-d'attribut porte le nom suivant :

1062

1063

```
1064 urn:oasis:names:tc:ebxml-regrep:rim:RegistryObject:contentLocator
```

1065

1066 Chaque valeur de la collection propre à la place-d'attribut spécifie un suffixe de l'URL défini par le
1067 demandeur pour l'Item-de-référentiel associé à l'Objet-extrinsèque. Le registre DOIT renvoyer l'Item-de-
1068 référentiel au moment où l'application HTTP transmet une demande GET (obtenir) de HTTP dont l'URL
1069 s'apparie à n'importe quel URL figurant dans la place-d'attribut *localisateur-de-contenu* (le cas échéant)
1070 de l'Objet-extrinsèque relatif à cet Item-de-référentiel.

1071 **4.3 URL par chemin d'accès**

1072 Le mécanisme de l'URL par chemin d'accès permet aux applications HTTP d'accéder à des Objets-de-
1073 registre et à des Items-de-référentiel par le biais d'un URL provenant de la hiérarchie de Paquets-de-
1074 registre de ces Objets-de-registre ou Items-de-référentiel.

1075 **4.3.1 Métaphore de fichiers et de dossiers**

1076 La classe Paquet-de-registre, comme le définit le [ebRIM], favorise l'organisation structurelle d'objets
1077 selon une hiérarchie de Paquets-de registre. Dans ces circonstances, un Paquet-de-registre joue un rôle
1078 semblable à celui d'un dossier à l'intérieur de la métaphore de fichiers et de dossiers souvent présente
1079 dans les systèmes-de-fichiers de la plupart des systèmes d'exploitation. Dans la même veine, les

1080 éléments d'un Paquet-de-registre jouent un rôle similaire à celui des fichiers compris dans un dossier,
1081 dans la métaphore de fichiers et de dossiers.

1082 Dans le cadre de la métaphore de fichiers et de dossiers, un demandeur produit un Paquet-de-registre
1083 dans le but de créer l'équivalent fonctionnel d'un dossier, et il conçoit un Objet-de-registre pour créer
1084 l'équivalent fonctionnel d'un fichier. Aussi, il ajoute un Objet-de-registre de façon qu'il représente un
1085 élément de Paquet-de-registre, afin de créer l'équivalent fonctionnel propre à l'ajout d'un fichier dans un
1086 dossier.

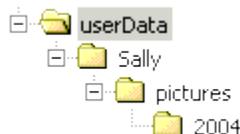
1087 4.3.2 Chemin d'accès d'un Objet-de-registre

1088 Chaque Objet-de-registre dispose d'un *chemin d'accès* implicite. Celui-ci représente une structure de
1089 chemin semblable à la structure de chemin d'accès Unix. Le chemin d'accès est constitué de segments
1090 de chemin d'accès. De façon analogue au chemin d'accès Unix, le dernier segment de chemin d'accès
1091 désigne l'Objet-de-registre, tandis que les segments antérieurs représentent les Paquets-de-registre qui
1092 figurent dans la hiérarchie des éléments de l'Objet-de-registre. Chaque segment se traduit par le nom du
1093 Paquet-de-registre ou de l'Objet-de-registre. L'attribut *nom* est de type Chaîne-en-toutes-langues; par
1094 conséquent, le segment de chemin s'apparie au nom d'un objet dans un paramètre de lieu en particulier.

1095 4.3.2.1 Exemple de chemin d'accès

1096 Voici un exemple où un registre dispose d'une hiérarchie de Paquets-de-registre et a recours au nom des
1097 objets du paramètre de lieu « en_US » :

1098



1099

Diagramme 3 : Exemple de hiérarchie de Paquets-de-registre

1100 Partons du principe que le Paquet-de-registre « 2004 » possède un Objet-extrinsèque appelé « baby.gif »
1101 pour un item de référentiel, et que celui-ci constitue une photographie en format GIF. Dans cet exemple,
1102 les chemins d'accès qui s'appliquent aux divers objets du paramètre de lieu « en_US » sont illustrés dans
1103 le tableau suivant :

1104

Nom de l'objet	Chemin d'accès
userData	/userData
Sally	/userData/Sally
pictures	/userData/Sally/pictures
2004	/userData/Sally/pictures/2004
baby.gif	/userData/Sally/pictures/2004/baby.gif

Tableau 5 : Exemples de chemin d'accès

1105 Il est important de mentionner que dans l'exemple ci-dessus, on présume que le Paquet-de-registre
1106 nommé « userData » (données-d'utilisateur) constitue un paquet racine (qui ne figure pas dans un autre
1107 Paquet-de-registre).

1108 4.3.3 Appariement entre un URL et des objets

1109 Toute application de registre PEUT accéder aux Objet-de-registre et aux Items-de-référentiel par la
1110 demande GET (obtenir) de HTTP. Elle agit ainsi en utilisant les formes précises URL qui s'inspirent du
1111 chemin d'accès pour les objets cibles. La présente section décrit la façon dont un registre résout des
1112 URL par chemin d'accès qui sont spécifiés par une application HTTP.

1113 Le registre DOIT traiter chaque segment de chemin du début à la fin du chemin en question. Et, pour
1114 chaque segment de chemin, il DOIT assurer l'appariement entre le segment et l'attribut valeur d'une
1115 Chaîne-en-une-langue-donnée dans l'attribut nom d'un Objet-de-registre. Pour tous les segments de

1116 chemin, sauf le dernier, l'Objet-de-registre apparié DOIT se traduire par un Paquet-de-registre. Le dernier
1117 segment de chemin, lui, PEUT s'apparier à n'importe quel Objet-de-registre, dont un Paquet-de-registre.
1118 S'il y a omission d'appariement pour un segment de chemin quelconque, il est impossible de résoudre
1119 l'URL par la méthode de l'URL par chemin d'accès. Au moment de l'appariement d'un segment autre que
1120 le premier, le registre DOIT voir à ce que l'Objet-de-registre apparié constitue un élément du Paquet-de-
1121 registre qui s'apparie au segment antérieur.

1122 **4.3.4 Appariement entre un URL et un objet unique**

1123 Si un URL par chemin d'accès s'apparie à un objet unique, il y a deux réponses possibles :

1124

- 1125 • Si la forme précise URL ne se termine pas par le caractère « / » ou si le dernier segment ne
1126 s'apparie pas à un Paquet-de-registre, le registre DOIT envoyer comme réponse un
1127 document XML qui se traduit par la représentation XML de l'Objet-de-registre s'appariant au
1128 dernier segment. Si le dernier segment s'apparie à un Objet-extrinsèque et si l'URL désigne
1129 expressément le paramètre GET (obtenir) de HTTP nommé « Obtenir-l'Item-de-référentiel » et
1130 ayant la valeur « VRAI », le registre DOIT retourner comme réponse l'item de référentiel qui se
1131 rattache à l'Objet-extrinsèque;
- 1132 • Si la forme précise URL se termine par le caractère « / » et si le dernier segment s'apparie à un
1133 Paquet-de-registre, le registre DOIT envoyer comme réponse un document HTML. Celui-ci doit
1134 constituer l'inscription au répertoire (section 4.3.6) de l'ensemble des Objets-de-registre qui
1135 représentent les éléments du Paquet-de-registre s'appariant au dernier segment.

1136

1137 **4.3.5 Appariement entre un URL et des objets multiples**

1138 Tout registre DOIT renfermer une inscription partielle dans le répertoire d'un Paquet-de-registre si un URL
1139 par chemin d'accès s'apparie à des objets multiples.

1140 Un URL par chemin d'accès peut s'apparier à des objets multiples :

1141

- 1142 • s'il existe dans le même Paquet-de-registre des objets multiples qui ont le même nom;
- 1143 • si le segment en question renferme des caractères de remplacement comme « % » ou « ? » en
1144 vue de l'appariement entre les noms des objets multiples au sein du même Paquet-de-registre. Il
1145 est important de mentionner que les caractères de remplacement doivent comporter
1146 l'encodage URL, comme le définit le protocole HTTP. Par exemple, le caractère « % » a comme
1147 encodage « %25 ».

1148

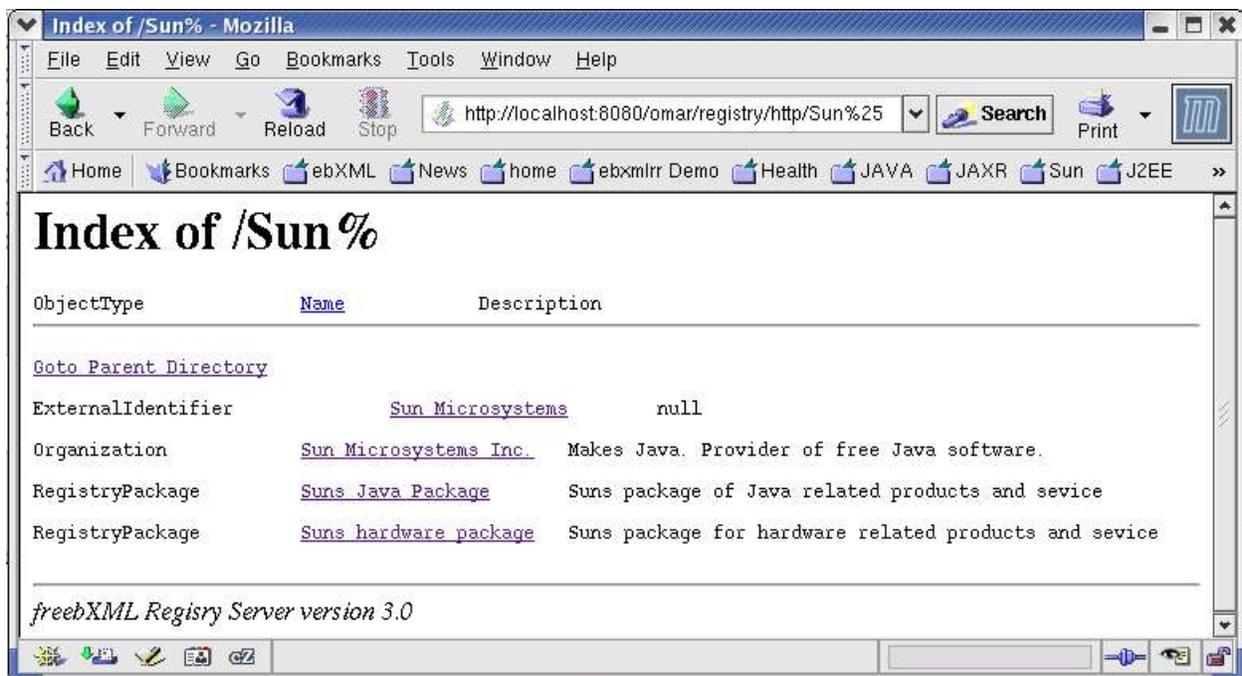
1149 **4.3.6 Inscription au répertoire**

1150 Tout registre DOIT renvoyer une inscription au répertoire à titre de réponse dans certaines circonstances,
1151 comme il a été décrit précédemment. Cette inscription DOIT présenter une liste d'objets qui figurent dans
1152 un Paquet-de-registre en particulier.

1153 Tout registre DEVRAIT structurer une inscription au répertoire de façon que chaque item du répertoire
1154 donne de l'information sur un Objet-de-registre dans le Paquet-de-registre. Les registres PEUVENT
1155 formater leur page d'inscription au répertoire d'une manière qui leur est propre. On laisse toutefois
1156 entendre qu'ils DEVRAIENT la formater en tant que page HTML comprenant au minimum les attributs
1157 type-d'objet, nom et description pour chaque Objet-de-registre énuméré dans l'inscription au répertoire.

1158 Le Diagramme 4 donne un exemple non normatif d'inscription au répertoire qui assure l'appariement entre
1159 l'ensemble des objets racines dont le nom commence par « Sun » (path /Sun%25).

1160



1161

Diagramme 4 : Exemple d'inscription au répertoire

1162 4.3.7 Contrôle d'accès dans la hiérarchie de Paquets-de-registre

1163 Il est important qu'un système de fichiers permette de contrôler qui peut ajouter des fichiers et des sous-
1164 dossiers dans un dossier. Il en est de même pour le mécanisme de l'URL par chemin d'accès.

1165 Tout demandeur PEUT attribuer une règle contextualisée de contrôle d'accès à un Paquet-de-registre
1166 pour créer l'équivalent fonctionnel de l'affectation du contrôle d'accès à un dossier dans la métaphore de
1167 fichiers et de dossiers. La règle contextualisée de contrôle d'accès DEVRAIT permettre d'utiliser l'action
1168 de *référencer* pour contrôler qui peut ajouter des Objets-de-registre à titre d'éléments du dossier, comme
1169 le décrit le [ebRIM].

1170 4.4 Algorithme de définition URL

1171 Le rattachement de protocole HTTP peut accueillir de multiples mécanismes de définition d'URL HTTP;
1172 par conséquent, tout registre DEVRAIT implanter un algorithme afin de déterminer le mécanisme adéquat
1173 de rattachement de protocole HTTP pour définir un URL quelconque.

1174 La présente section présente un algorithme non normatif de définition URL que tout registre DEVRAIT
1175 utiliser pour établir les divers mécanismes de rattachement de protocole HTTP qui devraient servir à la
1176 définition d'un URL HTTP.

1177 Dès la réception d'une demande GET (obtenir) de HTTP, un registre DEVRAIT d'abord vérifier si l'URL en
1178 question constitue un URL fondé sur l'encodage d'un RPC, ce qu'il PEUT faire en s'assurant que le
1179 paramètre URL d'*interface* est mentionné dans l'URL. Si tel est le cas, le registre DEVRAIT définir
1180 l'URL en ayant recours à la méthode de l'URL fondé sur l'encodage d'un RPC, comme le définit la
1181 section 4.1. Si, cependant, ce paramètre n'est pas évoqué, le registre DEVRAIT se servir de la méthode
1182 de l'URL défini par le demandeur pour voir à ce qu'il soit possible de définir l'URL. S'il est toujours
1183 impossible de le définir, le registre DEVRAIT veiller à ce qu'il le soit par le biais de la méthode de
1184 l'URL par chemin d'accès. Si, malgré tout, il est encore impossible de résoudre l'URL, le registre devrait
1185 renvoyer une erreur HTTP 404 (non trouvé), comme le stipule le protocole HTTP.

1186 **4.5 Motifs de sécurité**

1187 Tout registre DOIT appliquer l'ensemble des règles de contrôle d'accès, dont la restriction relative aux
1188 activités de lecture, au moment du traitement d'une demande propre au rattachement de protocole HTTP
1189 d'une interface de services. Cette responsabilité sous-entend que les registres NE DOIVENT PAS définir
1190 un URL par rapport à un Objet-de-registre ou à un Item-de-référentiel si l'application n'est pas autorisée à
1191 lire cet objet.

1192 **4.6 Gestion des exceptions**

1193 Si une méthode d'interface de services produit une exception, celle-ci DOIT être signalée dans une
1194 `Liste-d'erreurs-de-registre` et renvoyée à l'application dans la réponse HTTP à la
1195 demande HTTP.

1196 En cas d'erreur, le message et le code d'état HTTP DEVRAIENT correspondre aux erreurs détectées et
1197 signalées dans la `Liste-d'erreurs-de-registre`. Par exemple, si celle-ci démontre qu'aucun objet
1198 n'a été trouvé et, par conséquent, ne peut être renvoyé, le code d'erreur adéquat DEVRAIT être le code
1199 d'erreur 404. De plus, le message « Exception-liée-à-un-objet-non-trouvé » devrait être joint à ce code. Le
1200 document [RFC 2616] renferme une liste détaillée des codes d'état HTTP. L'appariement entre les
1201 exceptions propres au registre et les codes d'état HTTP n'a pas encore été établi.

1202 5 Protocoles de gestion du cycle de vie

1203 Le chapitre 5 définit les protocoles que peut accueillir l'interface de services de gestion du cycle de vie du
1204 registre. Ces protocoles fournissent la fonctionnalité dont les Applications-de-registre ont besoin pour
1205 gérer le cycle de vie des Objets-de-registre et des Items-de-référentiel du registre.

1206 Le schéma XML qui s'applique aux protocoles de gestion du cycle de vie est décrit dans le [RR-LCM-
1207 XSD].

1208 5.1 Protocole de demande d'inscription d'objet

1209 La Demande-d'inscription-d'objet permet à une Application-de-registre d'inscrire au moins un Objet-de-
1210 registre ou Item-de-référentiel.

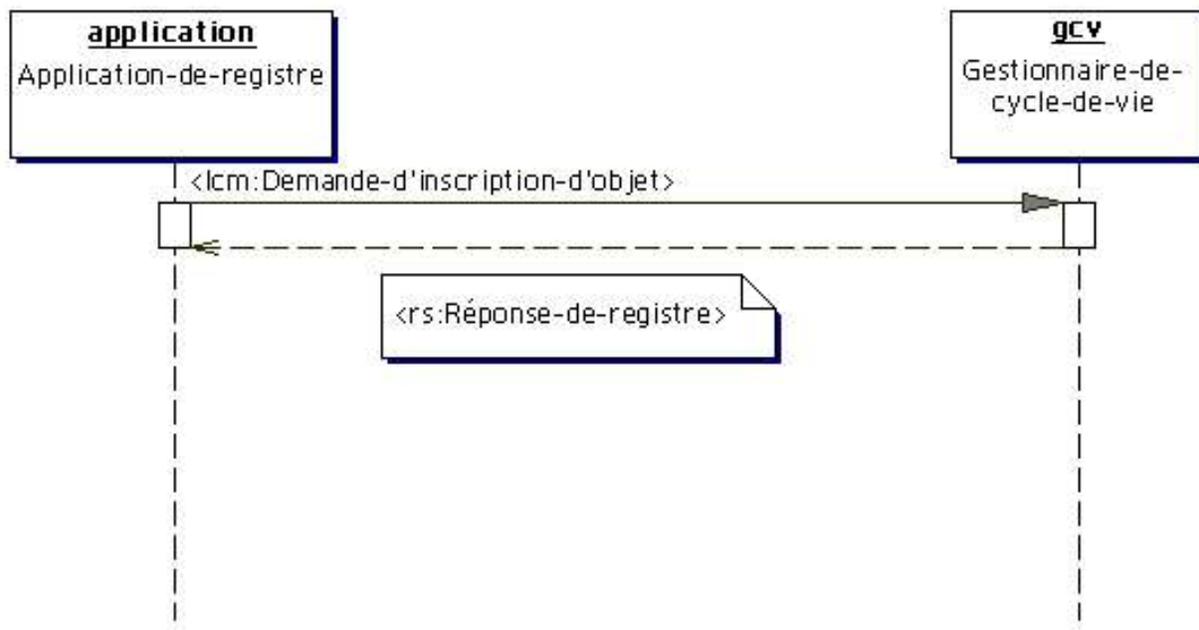


Diagramme 5 : Protocole de demande d'inscription d'objet

1212

1213 5.1.1 Demande-d'inscription-d'objet

1214 Toute application se sert de la Demande-d'inscription-d'objet pour inscrire des Objets-de-registre ou des
1215 Items-de-référentiel dans le registre.

1216 5.1.1.1 Syntaxe

```
1217 <element name="SubmitObjectsRequest">  
1218   <complexType>  
1219     <complexContent>  
1220       <extension base="rs:RegistryRequestType">  
1221         <sequence>  
1222           <element ref="rim:RegistryObjectList"/>  
1223         </sequence>  
1224       </extension>  
1225     </complexContent>  
1226   </complexType>  
1227 </element>
```

1228 **5.1.1.2 Paramètres**

- 1229 *Liste-d'objets-de-registre* : Ce paramètre désigne expressément une collection
1230 d'instances d'Objet-de-registre inscrites dans le registre. Les Objets-de-registre
1231 énumérés dans la liste peuvent constituer de tout nouveaux objets inscrits dans le
1232 registre, ou il peut s'agir d'objets qui existent déjà dans celui-ci. S'il s'agit d'objets déjà en
1233 place, le registre DOIT les gérer de la même façon qu'une Demande-de-mise-à-jour-
1234 d'objet et tout simplement procéder à leur mise à jour.

1235 **5.1.1.3 Renvois**

1236 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

1237 **5.1.1.4 Exceptions**

1238 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4, les
1239 exceptions suivantes PEUVENT être renvoyées :

- 1240 *Exception-liée-à-une-référence-non-résolue* : Démontre que le demandeur a cité un objet
1241 qui n'a pas été résolu pendant le traitement de la demande;
- 1242 *Exception-liée-à-un-item-de-référentiel-non-signé* : Signale que le demandeur a tenté
1243 d'inscrire un Item-de-référentiel qui n'était pas signé;
- 1244 *Exception-liée-à-un-excès-de-quota* : Révèle que le demandeur a essayé d'inscrire un
1245 objet dont le contenu excédait le quota admis par le registre.

1246 **5.1.2 Génération de l'identifiant unique**

1247 Comme le fait valoir le [ebRIM], tous les Objets-de-registre DOIVENT posséder un identifiant unique
1248 quant à la valeur de l'attribut identifiant. L'identifiant d'un Objet-de-registre DOIT représenter un
1249 URN valide et avoir un caractère unique par rapport à l'ensemble des autres Objets-de-registre compris
1250 dans le registre d'origine.

1251 Tout demandeur PEUT, de façon facultative, fournir un attribut identifiant pour des objets inscrits. Si un
1252 demandeur transmet un identifiant, s'il s'agit d'un URN valide et si celui-ci est compatible avec l'identifiant
1253 d'un Objet-de-registre déjà en place dans le registre d'origine, le registre DOIT intégrer la valeur de
1254 l'identifiant fourni par le demandeur et s'en servir à titre de valeur de l'attribut identifiant de l'objet du
1255 registre. Si, par contre, l'identifiant ne constitue pas un URN valide, le registre DOIT renvoyer une
1256 Exception-de-demande-invalide. De plus, si l'identifiant ne concorde pas avec celui d'un Objet-de-registre
1257 qui existe dans le registre d'origine, le registre DOIT renvoyer une Exception-de-demande-invalide en vue
1258 de l'exécution d'une Demande-de-mise-à-jour-d'objet et considérer cette mesure comme une action de
1259 mettre-à-jour propre à une Demande-d'inscription-d'objet.

1260 Si l'application ne transmet aucun identifiant pour un objet inscrit, le registre DOIT générer un identifiant
1261 unique universel. La valeur de tout identifiant généré par le registre DOIT respecter le format d'un URN
1262 désignant un identifiant universel unique de 128 bits pour l'environnement informatique réparti, comme le
1263 stipule l'identifiant unique universel [UUID] :

1264 (Exemple : *urn:uuid:a2345678-1234-1234-123456789012*)

1265 **5.1.3 Attribut identifiant et références d'objet**

1266 Des objets PEUVENT se servir de l'attribut identifiant d'un autre objet pour citer celui-ci. On PEUT utiliser
1267 l'attribut identifiant d'une Demande-d'inscription-d'objet pour faire référence à un objet de cette demande
1268 et citer un objet dans le registre. Le demandeur PEUT attribuer un identifiant à un objet qui figure dans la
1269 Demande-d'inscription-d'objet et qui doit être cité dans le document de demande de façon que cet objet
1270 puisse être référencé dans la demande. Le demandeur PEUT également assigner un URN valide à
1271 l'objet; l'identifiant est alors affecté de manière permanente à l'objet du registre. De façon subsidiaire, le
1272 demandeur PEUT attribuer un identifiant arbitraire qui ne représente pas un URN valide, pourvu que

1273 l'identifiant soit une valeur de tout-URI unique dans le document de demande. Dans un tel cas, l'identifiant
1274 agit à titre de mécanisme de lien au sein du document de demande, mais dès son inscription, il DOIT être
1275 remplacé par un identifiant généré par le registre.

1276 Si un objet de Demande-d'inscription-d'objet doit en citer un autre qui se trouve déjà dans le registre, la
1277 demande PEUT renfermer une Référence-d'objet dont l'attribut identifiant désigne l'identifiant de l'objet du
1278 registre. Cet identifiant est par définition un URN valide. Toute Référence-d'objet PEUT être considérée
1279 comme un mandataire dans la demande relative à un objet du registre.

1280 **5.1.4 Piste de vérification**

1281 Le registre DOIT produire un seul objet d'Événement-vérifiable dont le type-d'événement est créé pour
1282 l'ensemble des Objets-de-registre conçus par une Demande-d'inscription-d'objet.

1283 **5.1.5 Exemple de Demande-d'inscription-d'objet**

1284 L'exemple ci-dessous illustre une Demande-d'inscription-d'objet simple qui est axée sur l'inscription d'un
1285 objet d'Organisation unique dans le registre. On n'y voit pas le message SOAP intégral, l'en-tête de
1286 message et le contenu informationnel additionnel en ce qui concerne les items de référentiel.

```
1287  
1288 <lcm:SubmitObjectsRequest>  
1289   <rim:RegistryObjectList>  
1290     <rim:Organization lid="{LOGICAL_ID}"  
1291       id="{ID}"  
1292       primaryContact="{CONTACT_USER_ID}">  
1293       <rim:Name>  
1294         <rim:LocalizedString value="Sun Microsystems Inc." xml:lang="en-  
1295         US"/>  
1296       </rim:Name>  
1297       <rim:Address city="Burlington" country="USA" postalCode="01867"  
1298       stateOrProvince="MA" street="Network Dr." streetNumber="1"/>  
1299       <rim:TelephoneNumber areaCode="781" countryCode="1" number="123-  
1300       456" phoneType="office"/>  
1301     </rim:Organization>  
1302   </rim:RegistryObjectList>  
1303 </SubmitObjectsRequest>
```

1304 **5.2 Protocole de mise à jour d'objet**

1305 Le protocole relatif à la Demande-de-mise-à-jour-d'objet permet à une application de registre de mettre à
1306 jour au moins un Objet-de-registre ou item de référentiel du registre.

1307

Diagramme 6 : Protocole de mise à jour d'objet

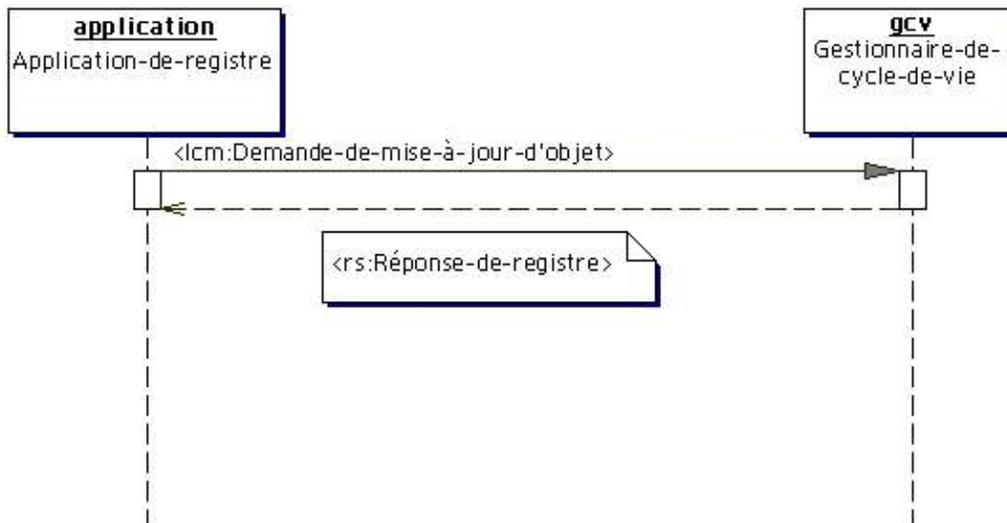
1308

1309 **5.2.1 Demande-de-mise-à-jour-d'objet**

1310 Toute application se sert de la Demande-de-mise-à-jour-d'objet pour mettre à jour des éléments d'Objet-
1311 de-registre ou d'item de référentiel qui existent déjà dans le registre.

1312 **5.2.1.1 Syntaxe**

```
1313 <element name="UpdateObjectsRequest">  
1314   <complexType>  
1315     <complexContent>  
1316       <extension base="rs:RegistryRequestType">  
1317         <sequence>  
1318           <element ref="rim:RegistryObjectList"/>  
1319         </sequence>  
1320       </extension>  
1321     </complexContent>  
1322   </complexType>  
1323 </element>
```



```

1319     </sequence>
1320     </extension>
1321     </complexContent>
1322     </complexType>
1323 </element>

```

1324 5.2.1.2 Paramètres

- 1325 ▪ *Liste-d'objets-de-registre* : Ce paramètre désigne une collection d'instances d'Objet-de-
1326 registre qui subissent une mise à jour dans le registre. Tous les enfants immédiats
1327 d'Objet-de-registre de la Liste-d'objets-de-registre DOIVENT se traduire par des Objets-
1328 de-registre actuels qui figurent déjà dans le registre. Les Objets-de-registre DOIVENT
1329 renfermer tous les attributs nécessaires, même ceux que l'utilisateur n'a pas l'intention de
1330 changer. Tout attribut manquant DOIT être interprété comme une demande
1331 d'établissement de la valeur NUL pour cet attribut. Si un attribut a une valeur par défaut,
1332 celle-ci prévaudra. Si la collection englobe un enfant immédiat d'Objet-de-registre qui
1333 n'existe pas déjà dans le registre, celui-ci DOIT renvoyer une Exception-de-demande-
1334 invalide. Si l'utilisateur souhaite inscrire des objets tant nouveaux que mis à jour, il
1335 DEVRAIT le faire en transmettant une Demande-d'inscription-d'objet. Si un Objet-
1336 extrinsèque est mis à jour et si la Demande-de-mise-à-jour-d'objet ne comprend aucun
1337 Item-de-référentiel, le registre DOIT maintenir tout Item-de-référentiel qui existe déjà et
1338 est associé à l'Objet-extrinsèque original ainsi que l'Objet-extrinsèque mis à jour. Si
1339 l'application souhaite supprimer l'Item-de-référentiel d'un Objet-extrinsèque en place, elle
1340 DOIT le faire au moyen d'une Demande-de-suppression-d'objet et de la fonction portée-
1341 de-suppression=Supprimer-item-de-référentiel-seulement.

1343 5.2.1.3 Renvois

1344 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

1345 5.2.1.4 Exceptions

1346 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4, les
1347 exceptions suivantes PEUVENT être renvoyées :

- 1348 ▪ *Exception-liée-à-une-référence-non-résolue* : Indique que le demandeur a cité un objet
1349 qui n'a pas été résolu pendant le traitement de la demande;

- 1350
- 1351
- 1352
- 1353
- *Exception-liée-à-un-item-de-référentiel-non-signé* : Démontre que le demandeur a tenté d'inscrire un Item-de-référentiel qui n'était pas signé;
 - *Exception-liée-à-un-excès-de-quota* : Révèle que le demandeur a essayé d'inscrire un objet dont le contenu excédait le quota admis par le registre.

1354 5.2.2 Piste de vérification

1355 Le registre DOIT produire un seul objet d'Événement-vérifiable dont le type-d'événement est *mis-à-jour*
 1356 pour l'ensemble des Objets-de-registre mis à jour par une Demande-de-mise-à-jour-d'objet.

1357 5.3 Protocole d'approbation d'objet

1358 Le protocole d'approbation d'objet permet à une application d'approuver au moins un objet déjà inscrit
 1359 d'Objet-de-registre par le biais de l'interface de services de Gestionnaire-de-cycle-de-vie.

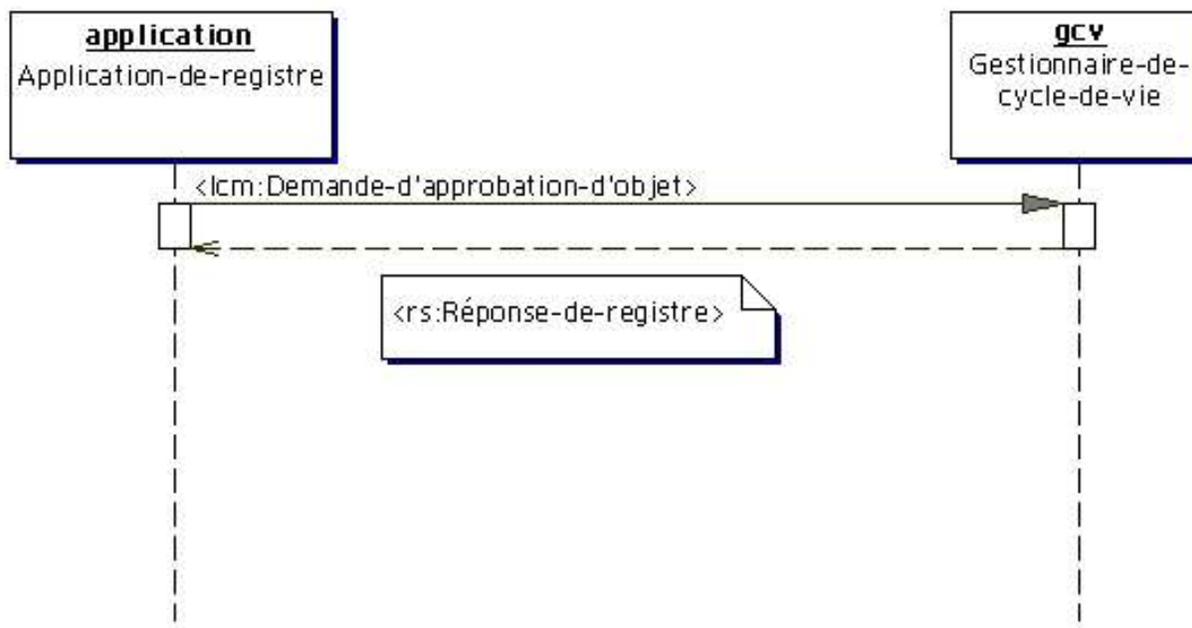


Diagramme 7 : Protocole d'approbation d'objet

1361 5.3.1 Demande-d'approbation-d'objet

1362 Toute application se sert de la Demande-d'approbation-d'objet pour approuver au moins une instance
 1363 d'Objet-de-registre du registre.

1364 5.3.1.1 Syntaxe

```

1365 <element name="ApproveObjectsRequest">
1366   <complexType>
1367     <complexContent>
1368       <extension base="rs:RegistryRequestType">
1369         <sequence>
1370           <element ref="rim:AdhocQuery" minOccurs="0" maxOccurs="1" />
1371           <element ref="rim:ObjectRefList" minOccurs="0"
1372 maxOccurs="1" />
1373         </sequence>
1374       </extension>
1375     </complexContent>
1376   </complexType>
  
```

1378 5.3.1.2 Paramètres

- 1379 ▪ **Interrogation-ad-hoc** : Ce paramètre représente une interrogation. Tout registre DOIT
1380 approuver l'ensemble des objets qui s'apparient à l'interrogation spécifiée, en plus de tout
1381 autre objet établi par les autres paramètres.
- 1382 ▪ **Liste-de-références-d'objet** : Ce paramètre désigne une collection de références aux
1383 instances d'Objet-de-registre du registre. Tout registre DOIT approuver l'ensemble des
1384 objets cités par ce paramètre ainsi que tout autre objet déterminé par les autres
1385 paramètres.

1386 5.3.1.3 Renvois

1387 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

1388 5.3.1.4 Exceptions

1389 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4,
1390 l'exception suivante PEUT être renvoyée :

- 1391 ▪ **Exception-liée-à-un-objet-non-trouvé** : Révèle que le demandeur a demandé un objet qui
1392 n'a pas été trouvé.

1393

1394 5.3.2 Piste de vérification

1395 Le registre DOIT produire un seul objet d'Événement-vérifiable dont le type-d'événement est *approuvé*
1396 pour l'ensemble des instances d'Objet-de-registre approuvées par une Demande-d'approbation-d'objet.

1397 5.4 Protocole de déclassement d'objet

1398 Le protocole de déclassement d'objet permet à une application de déclasser au moins une instance
1399 d'Objet-de-registre déjà inscrite en utilisant l'interface de services de Gestionnaire-de-cycle-de-vie. Une
1400 fois qu'un Objet-de-registre a été déclassé, il est impossible d'inscrire de nouvelles références qui s'y
1401 rattachent (par exemple, de nouvelles Associations et Classifications et de nouveaux Liens-externes). Les
1402 références qui s'associent déjà à un objet déclassé continuent toutefois de fonctionner normalement.

1403

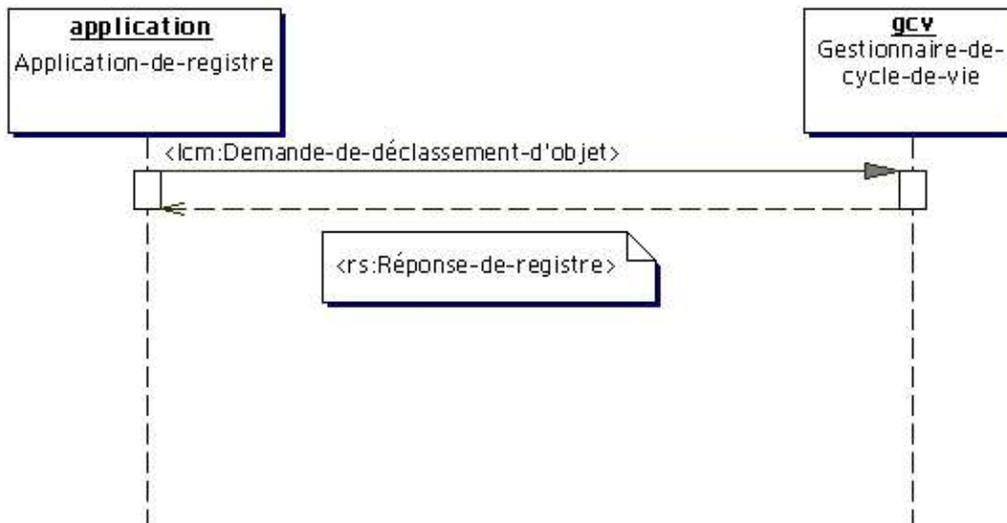
Diagramme 8 : Protocole de déclassement d'objet

1404 5.4.1 Demande-de-déclassement-d'objet

1405 Toute application se sert de la Demande-de-déclassement-d'objet pour déclasser au moins une instance
1406 d'Objet-de-registre du registre.

1407 5.4.1.1 Syntaxe

```
1408 <element name="DeprecateObjectsRequest">
1409   <complexType>
1410     <complexContent>
1411       <extension base="rs:RegistryRequestType">
1412         <sequence>
1413           <element ref="rim:AdhocQuery" minOccurs="0" maxOccurs="1" />
1414           <element ref="rim:ObjectRefList" minOccurs="0"
1415 maxOccurs="1" />
1416         </sequence>
1417       </extension>
```



1418
1419
1420

```

</complexContent>
</complexType>
</element>

```

1421 5.4.1.2 Paramètres

- 1422 ▪ **Interrogation-ad-hoc** : Ce paramètre désigne expressément une interrogation. Tout
- 1423 registre DOIT déclasser l'ensemble des objets qui s'apparient à l'interrogation
- 1424 déterminée, en plus de tout autre objet établi par les autres paramètres.
- 1425 ▪ **Liste-de-références-d'objet** : Ce paramètre spécifie une collection de références aux
- 1426 instances d'Objet-de-registre du registre. Tout registre DOIT déclasser l'ensemble des
- 1427 objets cités par ce paramètre ainsi que tout autre objet précisé par les autres paramètres.

1428 5.4.1.3 Renvois

1429 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

1430 5.4.1.4 Exceptions

1431 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4,

1432 l'exception suivante PEUT être renvoyée :

- 1433 ▪ **Exception-liée-à-une-référence-non-résolue** : Démontre que le demandeur a cité un objet
- 1434 qui n'a pas été résolu pendant le traitement de la demande.

1435 5.4.2 Piste de vérification

1436 Le registre DOIT produire un seul objet d'Événement-vérifiable dont le type-d'événement est *déclassé*

1437 pour l'ensemble des Objets-de-registre déclassés par une Demande-de-déclassement-d'objet.

1438 5.5 Protocole de rétablissement d'objet

1439 Le protocole de rétablissement d'objet de l'interface de services de Gestionnaire-de-cycle-de-vie permet à

1440 une application d'annuler une commande de déclassement d'au moins une instance d'Objet-de-registre

1441 qui a déjà été déclassée. Si un Objet-de-registre est rétabli, il retrouve le statut *inscrit*, et il est alors de

1442 nouveau possible d'inscrire de nouvelles références par rapport à cet objet (par exemple, de nouvelles

1443 Associations et Classifications et de nouveaux Liens-externes).

1444

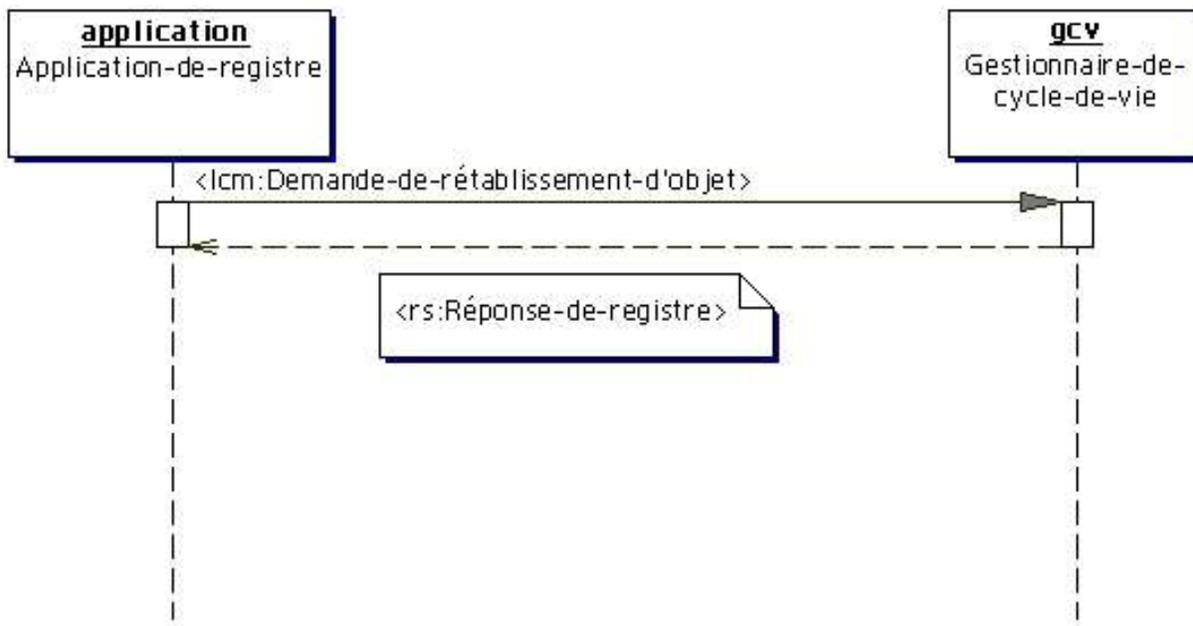


Diagramme 9 : Protocole de rétablissement d'objet

1445 5.5.1 Demande-de-rétablissement-d'objet

1446 Toute application se sert de la Demande-de-rétablissement-d'objet pour rétablir au moins une instance
 1447 d'Objet-de-registre du registre. Celui-ci DOIT négliger discrètement toute tentative de rétablir un Objet-de-
 1448 registre qui n'a pas été déclassé.

1449 5.5.1.1 Syntaxe

```

1450 <element name="UndeprecateObjectsRequest">
1451   <complexType>
1452     <complexContent>
1453       <extension base="rs:RegistryRequestType">
1454         <sequence>
1455           <element ref="rim:AdhocQuery" minOccurs="0" maxOccurs="1" />
1456           <element ref="rim:ObjectRefList" minOccurs="0"
1457 maxOccurs="1" />
1458         </sequence>
1459       </extension>
1460     </complexContent>
1461   </complexType>
1462 </element>
1463 </element>
  
```

1464 5.5.1.2 Paramètres

- 1465 ▪ **Interrogation-ad-hoc** : Ce paramètre désigne une interrogation. Tout registre DOIT
 1466 rétablir l'ensemble des objets qui s'apparient à l'interrogation précisée en plus de tout
 1467 autre objet déterminé par les autres paramètres.
- 1468 ▪ **Liste-de-références-d'objet** : Ce paramètre représente une collection de références aux
 1469 instances d'Objet-de-registre du registre. Tout registre DOIT rétablir l'ensemble des
 1470 objets qui sont cités par ce paramètre ainsi que tout autre objet spécifié par les autres
 1471 paramètres.

1472 **5.5.1.3 Renvois**

1473 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

1474 **5.5.1.4 Exceptions**

1475 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4,
1476 l'exception suivante PEUT être renvoyée :

- 1477 ▪ *Exception-liée-à-une-référence-non-résolue* : Signale que le demandeur a cité un objet
1478 qui n'a pas été résolu pendant le traitement de la demande.

1479 **5.5.2 Piste de vérification**

1480 Le service de registre DOIT produire un seul objet d'Événement-vérifiable dont le type-d'événement est
1481 *rétabli* pour l'ensemble des Objets-de-registre rétablis par une Demande-de-rétablissement-d'objet.

1482 **5.6 Protocole de suppression d'objet**

1483 Le protocole de suppression d'objet permet à une application de supprimer au moins une instance
1484 d'Objet-de-registre ou au moins un item de référentiel en utilisant l'interface de services de Gestionnaire-
1485 de-cycle-de-vie.

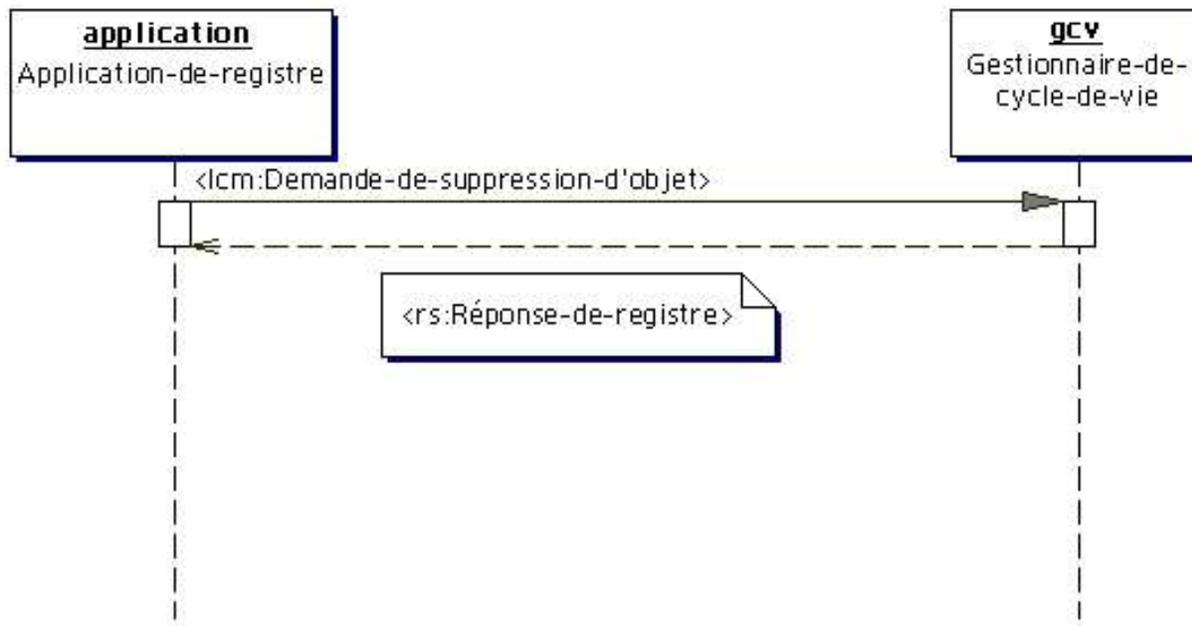


Diagramme 10: Protocole de suppression d'objet

1487 Pour avoir des détails sur le schéma qui se rattache aux documents d'affaires présentés dans ce
1488 processus, prière de consulter le .

1489 **5.6.1 Demande-de-suppression-d'objet**

1490 Toute application se sert de la Demande-de-suppression-d'objet pour supprimer au moins un Objet-de-
1491 registre ou item de référentiel du registre.

1492 **5.6.1.1 Syntaxe**

1493 `<element name="RemoveObjectsRequest">`
1494 `<complexType>`

```

1495 <complexContent>
1496   <extension base="rs:RegistryRequestType">
1497     <sequence>
1498       <element ref="rim:AdhocQuery" minOccurs="0" maxOccurs="1" />
1499       <element ref="rim:ObjectRefList" minOccurs="0"
1500 maxOccurs="1" />
1501     </sequence>
1502     <attribute name="deletionScope"
1503 default="urn:oasis:names:tc:ebxml-regrep:DeletionScopeType:DeleteAll"
1504 type="rim:referenceURI" use="optional"/>
1505   </extension>
1506 </complexContent>
1507 </complexType>
1508 </element>

```

1509 5.6.1.2 Paramètres

1510

- 1511 ▪ **portée-de-suppression** : Ce paramètre détermine la portée de l'impact de la Demande-
1512 de-suppression-d'objet. La valeur de l'attribut portée-de-suppression DOIT constituer une
1513 référence à un Nœud-de-classification dans le Schème-de-classification canonique de
1514 Type-de-portée-de-suppression, comme le décrit le [ebRIM]. Tout registre DOIT pouvoir
1515 accueillir les types de portée-de-suppression, conformément au Schème-de-classification
1516 canonique de Type-de-portée-de-suppression. On peut procéder à l'extension de ce
schème en y ajoutant des Nœuds-de-classification.

1517 On a défini les Nœuds-de-classification canoniques suivants pour le Schème-de-
1518 classification de Type-de-portée-de-suppression :

1519 **Supprimer-item-de-référentiel-seulement** : Cette portée-de-suppression stipule
1520 que le registre DOIT supprimer l'Item-de-référentiel propre aux Objets-
1521 extrinsèques établis, mais NE DOIT PAS supprimer ceux-ci. On assure ainsi la
1522 validité continue des références à ces Objets-extrinsèques. Tout registre DOIT
1523 établir le statut de l'instance d'Objet-extrinsèque de façon qu'elle soit *retirée*.

1524 **Supprimer-tout** : Cette portée-de-suppression souligne que la demande DOIT
1525 supprimer l'Objet-de-registre et l'Item-de-référentiel (le cas échéant) quant aux
1526 objets spécifiés. Il est possible de supprimer tout Objet-de-registre au moyen
1527 d'une Demande-de-suppression-d'objet axée sur la portée-de-suppression
1528 Supprimer-tout seulement si toutes les références (par exemple, les
1529 Associations, les Classifications et les Liens-externes) propres à cet Objet-de-
1530 registre ont été supprimées.

1531 ▪ **Interrogation-ad-hoc** : Ce paramètre désigne expressément une interrogation. Tout
1532 registre DOIT supprimer l'ensemble des objets qui s'appartient à l'interrogation établie, en
1533 plus de tout autre objet déterminé par les autres paramètres.

1534 ▪ **Liste-de-références-d'objet** : Ce paramètre spécifie une collection de références aux
1535 instances d'Objet-de-registre du registre. Tout registre DOIT supprimer l'ensemble des
1536 objets cités par ce paramètre ainsi que tout autre objet établi par les autres paramètres.

1537 5.6.1.3 Renvois

1538 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

1539 5.6.1.4 Exceptions

1540 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4, les
1541 exceptions suivantes PEUVENT être renvoyées :

1542 ▪ **Exception-liée-à-une-référence-non-résolue** : Démontre que le demandeur a cité un
1543 objet qui n'a pas été résolu pendant le traitement de la demande.

- 1544 ▪ **Exception-liée-à-des-références-en-place** : Révèle que le demandeur a tenté de
1545 supprimer un Objet-de-registre alors que les références s'y rattachant existaient encore. Il
1546 est important de mentionner que la suppression d'un Objet-de-registre et de tous les
1547 Objets-de-registre qui y font référence dans le cadre de la même demande est valide.
1548 Dans un tel cas, l'Exception-liée-à-des-références-en-place NE DOIT PAS être rejetée.

1549 **5.7 Contrôle de version géré par registre**

1550 La section 5.7 décrit les caractéristiques du contrôle de version du registre ebXML. Ces particularités sont
1551 fondées sur le [DeltaV]. Le registre ebXML présente une façade simplifiée, qui, elle, fournit un petit sous-
1552 ensemble de la fonctionnalité du [DeltaV].

1553 **5.7.1 Ressources placées sous contrôle de version**

1554 L'ensemble des items de référentiel d'un registre ebXML se traduisent de façon implicite par des
1555 ressources placées sous contrôle de version, comme le définit la section 2.2.1 du [DeltaV]. Il n'est pas
1556 nécessaire de prendre des mesures explicites pour faire de ces items des ressources placées sous
1557 contrôle de version.

1558 Les instances d'Objet-de-registre constituent elles aussi de manière implicite des ressources placées
1559 sous contrôle de version. Un registre peut cependant restreindre ces ressources dans un sous-ensemble
1560 de classes d'Objet-de-registre à partir de règles propres au registre.

1561 Tout registre qui implante la caractéristique de contrôle par une version DEVRAIT au minimum faire des
1562 types suivants des ressources placées sous contrôle de version :

- 1563 ▪ Nœud-de-classification;
- 1564 ▪ Schème-de-classification;
- 1565 ▪ Organisation;
- 1566 ▪ Objet-extrinsèque;
- 1567 ▪ Paquet-de-registre;
- 1568 ▪ Service.

1569 On a choisi la liste ci-dessus de façon à exclure tous les types composés et à inclure la plupart des autres
1570 types d'Objet-de-registre pour lesquels des cas d'utilisation connus nécessitent un versionnage.

1571 **5.7.2 Versionnage et identification d'objet**

1572 Chaque version d'Objet-de-registre représente un objet unique et, à ce titre, possède sa propre valeur
1573 unique pour son attribut identifiant, comme le [ebRIM].

1574 **5.7.3 ID logique**

1575 Toutes les versions d'un Objet-de-registre quelconque correspondent logiquement au même objet, et
1576 chacune d'elles est désignée sous l'appellation d'Objet-de-registre *logique*. Un Objet-de-registre logique
1577 constitue une structure arborescente dont les nœuds sont des versions spécifiques de l'Objet-de-registre.

1578 Une version précise d'Objet-de-registre logique est appelée *instance d'Objet-de-registre*.

1579 Toute instance d'Objet-de-registre DOIT disposer d'un *ID logique* pour favoriser la reconnaissance de ses
1580 propres éléments dans un Objet-de-registre logique en particulier. Il est important de souligner que cette
1581 situation est contraire à celle propre à l'attribut *identifiant*, qui, lui, DOIT être unique pour chaque
1582 version du même Objet-de-registre logique. Toute application peut évoquer l'Objet-de-registre logique de
1583 manière indépendante par rapport aux versions, en utilisant son propre ID logique.

1584 Tout Objet-de-registre se fait attribuer un ID logique par le biais de l'attribut *ID logique* de la classe
1585 d'Objet-de-registre. Si celui-ci est affecté par le demandeur, il doit y avoir une garantie qu'il s'agit là d'un
1586 URN unique à l'échelle mondiale. Tout registre DOIT accepter un ID logique valide transmis par un
1587 demandeur. Si le demandeur ne spécifie aucun ID logique, le registre DOIT en attribuer un, et la valeur de
1588 l'attribut ID logique DOIT être identique à celle de l'attribut identifiant de la première version (qui a été

1589 créée à l'origine) de l'Objet-de-registre logique.

1590 **5.7.4 Établissement de la version**

1591 Tout registre ebXML peut accueillir un versionnage indépendant des métadonnées d'Objet-de-registre et
1592 du contenu d'item de référentiel. Par conséquent, il est important de détenir de l'information sur chaque
1593 version d'une instance d'Objet-de-registre et de son item de référentiel s'il s'agit d'une instance d'Objet-
1594 extrinsèque.

1595 **5.7.4.1 Établissement de la version d'un Objet-de-registre**

1596 Tout Objet-de-registre DOIT posséder un attribut information-sur-la-version dont le type se traduit par la
1597 classe Information-sur-la-version définie par le [ebRIM]. L'attribut information-sur-la-version présente de
1598 l'information sur la version propre à l'instance d'Objet-de-registre en question. Tout registre DOIT faire en
1599 sorte que deux versions du même Objet-de-registre n'aient pas la même valeur pour les attributs
1600 information-sur-la-version et nom-de-la-version.

1601 **5.7.4.2 Établissement de la version d'un Item-de-référentiel**

1602 Si un Objet-de-registre se traduit par un Objet-extrinsèque qui dispose d'un item de référentiel connexe,
1603 l'établissement de la version de cet item de référentiel est indépendant de celui de la version de l'Objet-
1604 extrinsèque.

1605 Tout Objet-extrinsèque ayant un item de référentiel connexe DOIT posséder un attribut information-sur-la-
1606 version-du-contenu dont le type est la classe Information-sur-la-version définie par le [ebRIM]. L'attribut
1607 information-sur-la-version-du-contenu présente l'information sur la version qui s'applique à l'instance
1608 d'item de référentiel en question.

1609 Tout Objet-extrinsèque ne disposant d'aucun item de référentiel connexe NE DOIT PAS avoir d'attribut
1610 information-sur-la-version-du-contenu.

1611 Tout registre DOIT faire en sorte que deux versions du même Objet-extrinsèque aient la même valeur
1612 pour les attributs information-sur-la-version-du-contenu et nom-de-la-version; en effet, de multiples
1613 versions d'Objet-extrinsèque PEUVENT avoir en commun la même version d'Item-de-référentiel.

1614 **5.7.5 Versionnage d'un Objet-extrinsèque et d'un Item-de-référentiel**

1615 Il est possible de mettre à jour et, par conséquent, de versionner de manière indépendante tout Objet-
1616 extrinsèque et son item de référentiel connexe.

1617 Tout registre DOIT maintenir une arborescence de version distincte pour n'importe quel Objet-extrinsèque
1618 et son item de référentiel connexe, comme il a été décrit antérieurement.

1619 Le Tableau 6 démontre toutes les combinaisons possibles pour le versionnage d'un Objet-extrinsèque et
1620 de son item de référentiel. On a éliminé les combinaisons invalides ou impossibles ainsi que celles qui ne
1621 nécessitent aucune prise de mesure. Les seules combinaisons qui exigent un versionnage figurent dans
1622 les cases ombrées. On ne retrouve parmi ces combinaisons que deux cas uniques (dénommés « cas A »
1623 et « cas B »). Il faut mentionner qu'il est impossible de versionner un item de référentiel sans versionner
1624 également son Objet-extrinsèque.

1625

Existence d'un Objet- extrinsèque	Existence d'un Item-de- référentiel	Mise à jour d'un Objet-extrinsèque	Mise à jour d'un Item-de- référentiel	Commentaire
Non	Non			Ne rien faire
Non	Oui			Impossible

Oui	Non	Non	Non	Ne rien faire
		Non	Oui	Impossible
		Oui	Non	Versionner l'Objet-extrinsèque (cas A)
		Oui	Oui	Impossible
Oui	Oui	Non	Non	Ne rien faire
		Non	Oui	Impossible
		Oui	Non	Versionner l'Objet-extrinsèque (cas A)
		Oui	Oui	Versionner l'Objet-extrinsèque et l'Item-de-référentiel (cas B)

Tableau 6 : Versionnage d'un Objet-extrinsèque et d'un Item-de-référentiel

1626

1627 5.7.5.1 Objet-extrinsèque et Item-de-référentiel partagé

1628 Tout Objet-extrinsèque et son item de référentiel sont versionnés de façon indépendante (cas B); par
 1629 conséquent, la version de l'item de référentiel peut être la même pour des versions multiples de l'Objet-
 1630 extrinsèque. Dans un tel cas, l'attribut information-sur-la-version-du-contenu DOIT être le même pour ces
 1631 versions d'Objet-extrinsèque.

1632 5.7.6 Versionnage et objets composés

1633 Si un registre crée une nouvelle version d'Objet-de-registre, il DOIT copier tous les objets composés¹ sous
 1634 la forme de nouveaux objets composés dans la nouvelle version. Cette exigence s'explique par le fait que
 1635 chaque version représente un objet unique et que les objets composés, par définition, ne peuvent pas
 1636 être partagés dans l'éventail d'objets multiples. Plus particulièrement, chaque nouvelle copie d'objet
 1637 composé DOIT posséder un nouvel identifiant puisqu'il s'agit d'un objet différent de l'objet composé
 1638 d'origine de la version antérieure.

1639 Les registres NE DOIVENT PAS versionner d'objet composé.

1640 5.7.7 Versionnage et références

1641 Toute référence d'objet à partir d'un Objet-de-registre cite une version particulière de l'Objet-de-registre
 1642 référencé. Si un registre crée une nouvelle version d'Objet-de-registre cité, il NE DOIT PAS transférer de
 1643 références propres à d'autres objets à partir de la version antérieure jusqu'à la nouvelle version de l'objet
 1644 référencé. Les applications qui souhaitent toujours citer les versions les plus récentes d'un objet
 1645 PEUVENT se servir de la caractéristique de notification d'événement pour mettre à jour des références au
 1646 moment de la création de nouvelles versions. Elles pourront ainsi référencer en tout temps la version la
 1647 plus récente.

¹ Les types d'objet composé figurent dans le Diagramme 1 du *Modèle d'information du registre ebXML [ebRIM]* à titre de classes ayant une composition ou une relation via un « losange plein » avec le type d'Objet-de-registre.

1648 On fait face à un cas particulier si une Demande-d'inscription-d'objet ou une Demande-de-mise-à-jour-
1649 d'objet renferme un objet versionné par le registre et si la demande englobe d'autres objets qui
1650 référencent cet objet. Dans un tel cas, le registre DOIT mettre à jour l'ensemble des références des objets
1651 inscrits à l'objet versionné de façon que ces objets citent, à partir de là, la nouvelle version de l'objet créé
1652 par la demande.

1653 **5.7.8 Versionnage et piste de vérification**

1654 Le Schème-de-classification de Type-d'événement canonique utilisé par la caractéristique de piste de
1655 vérification définit d'abord un type d'événement mis à jour et ensuite un type d'événement versionné
1656 comme enfant du Nœud-de-classification de type d'événement mis à jour. Le trait sémantique se traduit
1657 par le fait qu'un type d'événement versionné constitue une spécialisation du type d'événement mis à jour.

1658 Tout registre DOIT se servir du type d'événement mis à jour de l'Événement-vérifiable au moment où il
1659 met à jour un Objet-de-registre sans toutefois créer de nouvelle version.

1660 Tout registre DOIT avoir recours au type d'événement versionné de l'Événement-vérifiable au moment où
1661 il crée une nouvelle version d'Objet-de-registre logique.

1662 Les registres NE DOIVENT PAS utiliser le type d'événement créé de l'Événement-vérifiable au moment
1663 où ils créent une nouvelle version d'Objet-de-registre logique.

1664 **5.7.9 Association inter-versions**

1665 Dans toute branche unique de l'arborescence de versions qui s'applique à un objet en particulier, toute
1666 version supplante de manière implicite la version qui la précède immédiatement. Il peut parfois être
1667 nécessaire de déterminer de façon explicite quelle version supplante quelle autre version du même objet.
1668 Cette caractéristique est particulièrement véridique si deux versions représentent des racines sœurs de
1669 l'arborescence de versions pour le même objet.

1670 Toute application PEUT spécifier une association entre deux versions quelconques d'un objet
1671 d'arborescence de versions en utilisant le type-d'association canonique « Supplante » pour démontrer que
1672 l'objet-source supplante l'objet-cible de l'association.

1673 Les applications NE DOIVENT PAS établir d'association entre deux versions d'un objet en ayant recours
1674 au type-d'association canonique « Supplante » si l'objet-source constitue une version antérieure au sein
1675 de la même branche d'arborescence de versions que celle de l'objet-cible. En effet, une telle situation
1676 violerait l'association « Supplante » implicite entre les deux versions.

1677 Il est important de mentionner que la présente section est équivalente sur le plan de la fonctionnalité aux
1678 éléments de l'ensemble-des-ancêtres-et-des-descendants des caractéristiques de version, comme le
1679 définit le [DeltaV].

1680 **5.7.10 Suppression de version entamée par une application**

1681 Tout registre ebXML PEUT permettre à des applications de supprimer des versions précises d'un Objet-
1682 de-registre. Toute application PEUT supprimer une version antérieure d'un objet quelconque par le biais
1683 de la Demande-de-suppression-d'objet en spécifiant la version à l'aide de son identifiant unique. La
1684 suppression d'un Objet-extrinsèque DOIT entraîner celle de son item de référentiel si celui-ci n'est cité par
1685 aucune autre version.

1686 **5.7.11 Suppression de version engagée par le registre**

1687 Le registre PEUT retrancher les versions antérieures en fonction des règles administratives propres au
1688 registre dans le but de gérer les ressources de stockage.

1689 **5.7.12 Verrouillage et modifications simultanées**

1690 La présente spécification ne définit pas de caractéristiques propres à l'espace de travail et axés sur une
1691 capacité de retour et d'emprunt, conformément au [DeltaV]. Tout registre ebXML PEUT accueillir des
1692 caractéristiques de ce genre selon une méthode particulière à l'implantation.

1693 La spécification ne préconise aucun modèle de verrouillage ou de branchement en particulier. Une
1694 implantation quelconque peut décider d'accueillir un modèle optimiste (sans verrouillage). Subsidièrement
1695 ou en complément, une implantation peut accueillir un modèle de verrouillage qui gère une capacité
1696 explicite d'archivage et d'extraction. Une note technique ou une spécification peuvent éventuellement tenir
1697 compte en partie d'une telle capacité.

1698 **5.7.13 Création d'une version**

1699 Le registre gère automatiquement la création d'une nouvelle version d'Objet-de-registre ou d'item de
1700 référentiel. Tout registre qui peut accueillir une option de versionnage DOIT concevoir de manière
1701 implicite une nouvelle version d'item de référentiel si cet item est mis à jour par le biais d'une Demande-
1702 d'inscription-d'objet ou d'une Demande-de-mise-à-jour-d'objet. Dans un cas semblable, le registre DOIT
1703 également créer une nouvelle version de son Objet-extrinsèque.

1704 Si l'application souhaite seulement mettre à jour et versionner l'Objet-extrinsèque, elle peut agir ainsi en
1705 se servant d'une Demande-de-mise-à-jour-d'objet, sans pour autant transmettre d'item de référentiel.
1706 Dans un tel cas, le registre DOIT attribuer la version d'item de référentiel associée à la version antérieure
1707 de l'Objet-extrinsèque.

1708 **5.7.14 Annulation du versionnage**

1709 Toute application PEUT spécifier un message d'aide *version-ne-pas* en s'appuyant sur chaque Objet-de-
1710 registre au moment d'inscrire ou de mettre à jour un Objet-de-registre quelconque. Les registres NE
1711 DEVRAIENT PAS créer une nouvelle version de cet Objet-de-registre si le message d'aide *version-ne-pas*
1712 a la valeur « VRAI ». Ce message d'aide PEUT être établi comme Place-d'attribut canonique ayant
1713 l'appellation suivante :

1714
1715 `urn:oasis:names:tc:ebxml-regrep:rim:RegistryObject:dontVersion`

1717 La Place-d'attribut de *version-ne-pas* DOIT avoir la valeur « VRAI » ou « FAUX » si celle-ci est précisée.

1718 Toute application PEUT spécifier un message d'aide de contenu-de-version-ne-pas en se fondant sur
1719 chaque Objet-extrinsèque au moment d'inscrire ou de mettre à jour un Objet-extrinsèque qui dispose d'un
1720 item de référentiel. Les registres NE DEVRAIENT PAS créer une nouvelle version de cet item de
1721 référentiel si le message d'aide de contenu-de-version-ne-pas a la valeur « VRAI ». On PEUT établir ce
1722 message d'aide à titre de Place-d'attribut canonique ayant l'appellation suivante :

1723
1724 `urn:oasis:names:tc:ebxml-regrep:rim:RegistryObject:dontVersionContent`

1726 La Place-d'attribut de contenu-de-version-ne-pas DOIT avoir la valeur « VRAI » ou « FAUX » si celle-ci
1727 est déterminée.

1728 Toute application PEUT également définir les Places-d'attribut de *version-ne-pas* et de contenu-de-
1729 version-ne-pas de la Demande-de-registre en utilisant l'élément `<rs:Liste-de-places-d'attribut-de-`
1730 `demande>`. Si ces Places-d'attribut sont spécifiées dans la demande, tout registre DOIT les considérer
1731 comme si elles figuraient dans chaque Objet-de-registre de cette demande. La valeur des Places-
1732 d'attribut qui est déterminée dans la demande a prépondérance sur celle des Places-d'attribut qui est
1733 spécifiée dans les Objets-de-registre de la demande.

1734

6 Protocoles de gestion d'interrogation

1735
1736
1737
1738

Le chapitre 6 définit les protocoles gérés par l'interface de services de Gestionnaire-d'interrogation de registre. Les protocoles de gestion d'interrogation fournissent la fonctionnalité dont les Applications-de-registre ont besoin pour interroger le registre et y détecter les Objets-de-registre et les Items-de-référentiel.

1739
1740

Le schéma XML qui s'applique aux protocoles de gestion d'interrogation est décrit dans le [RR-QUERY-XSD].

1741

6.1 Protocole d'interrogation ad hoc

1742
1743
1744

Le protocole d'interrogation ad hoc de l'interface de services de Gestionnaire-d'interrogation permet à une application d'interroger le registre et d'en extraire les Objets-de-registre ou les Items-de-référentiel qui s'apparient à l'interrogation établie.

1745
1746
1747

Toute application inscrit une interrogation ad hoc dans le Gestionnaire-d'interrogation en transmettant une Demande-d'interrogation-ad-hoc. Celle-ci renferme un sous-élément qui détermine une interrogation dans l'une des syntaxes d'interrogation gérées par le registre.

1748
1749
1750

Le Gestionnaire-d'interrogation fait parvenir comme réponse à l'application une Réponse-à-une-interrogation-ad-hoc. Celle-ci renvoie une collection d'objets qui s'apparient à l'interrogation. Cette collection est susceptible d'être hétérogène selon les options de demande et d'expression d'interrogation.

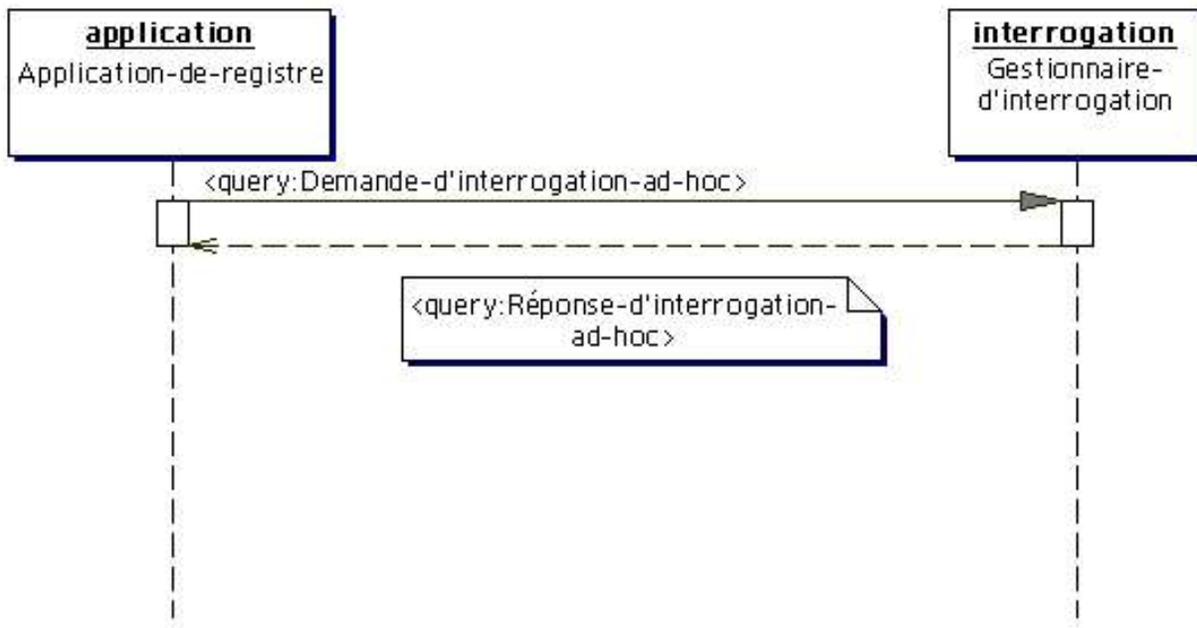


Diagramme 11 : Protocole d'interrogation ad hoc

1751

6.1.1 Demande-d'interrogation-ad-hoc

1752

On se sert de la Demande-d'interrogation-ad-hoc pour inscrire une interrogation dans le registre.

1753

6.1.1.1 Syntaxe

1754
1755
1756
1757
1758

```
<element name="AdhocQueryRequest">
  <complexType>
    <complexContent>
      <extension base="rs:RegistryRequestType">
        <sequence>
```

```

1759         <element maxOccurs="1" minOccurs="1"
1760             ref="tns:ResponseOption"/>
1761         <element ref="rim:AdhocQuery" />
1762     </sequence>
1763     <attribute default="false" name="federated"
1764         type="boolean" use="optional"/>
1765     <attribute name="federation" type="anyURI" use="optional"/>
1766     <attribute default="0" name="startIndex" type="integer"/>
1767     <attribute default="-1" name="maxResults" type="integer"/>
1768 </extension>
1769 </complexContent>
1770 </complexType>
1771 </element>

```

1772 6.1.1.2 Paramètres

- 1773 ▪ **Interrogation-ad-hoc** : Ce paramètre spécifie l'interrogation réelle. Il est décrit en détail
1774 dans la section .
- 1775 ▪ **fédéré** : Ce paramètre optionnel précise que le registre doit traiter l'interrogation à titre
1776 d'interrogation fédérée. Par défaut, sa valeur est FAUX. Cette valeur DOIT justement être
1777 FAUX si le registre achemine une interrogation fédérée vers un autre registre dans le but
1778 d'éviter la présence d'une boucle fermée pendant le traitement de l'interrogation fédérée.
- 1779 ▪ **fédération** : Ce paramètre optionnel établit l'identifiant de la Fédération cible dans le
1780 cadre d'une interrogation fédérée si le registre représente un élément de fédérations
1781 multiples. En l'absence de ce paramètre, tout registre doit acheminer l'interrogation
1782 fédérée vers toutes les fédérations dont il constitue un élément. Cette valeur NE DOIT
1783 PAS être spécifiée au moment où un registre achemine une interrogation fédérée vers un
1784 autre registre dans le but d'éviter la présence d'une boucle fermée au cours du traitement
1785 de l'interrogation fédérée.
- 1786 ▪ **résultats-maximaux** : Ce paramètre optionnel détermine une limite sur le plan du
1787 nombre maximal de résultats que l'application souhaite recevoir après avoir transmis une
1788 interrogation. Si le paramètre n'est pas spécifié, le registre DEVRAIT renvoyer l'ensemble
1789 des résultats. Par contre, si le nombre total de résultats excède une certaine limite propre
1790 au registre, celui-ci DEVRAIT renvoyer un sous-ensemble de résultats respectant cette
1791 limite. La section 6.2.1 présente un exemple à cet égard.
- 1792 ▪ **Option-de-réponse** : Ce paramètre requis permet à l'application de contrôler le format et
1793 le contenu de la Réponse-à-une-interrogation-ad-hoc produite par le registre en réponse
1794 à cette demande. La section 6.1.4 renferme des détails à ce sujet.
- 1795 ▪ **index-de-départ** : On se sert de cette valeur entière optionnelle pour établir les résultats
1796 qui *doivent* être renvoyés à titre de premier résultat au moment de l'itération d'un
1797 ensemble important de résultats. La valeur par défaut correspond à 0; par conséquent,
1798 elle renvoie l'ensemble de résultats à partir de l'index 0 (premier résultat). La section 6.2.1
1799 en donne un exemple.

1800 6.1.1.3 Renvois

1801 Cette demande renvoie une Réponse-à-une-interrogation-ad-hoc. Prière de consulter la section 6.1.2 pour
1802 avoir des détails à ce sujet.

1803 6.1.1.4 Exceptions

1804 En plus des exceptions qui s'appliquent à l'ensemble des demandes définies dans la section 2.1.1.4,
1805 l'exception suivante PEUT être renvoyée :

- 1806 ▪ **Exception-de-requête-invalidé** : Révèle que la sémantique ou la syntaxe de requête est
1807 invalide. L'application doit corriger l'erreur de sémantique ou de syntaxe de requête et
1808 inscrire de nouveau l'interrogation.

1809 6.1.2 Réponse-à-une-interrogation-ad-hoc

1810 Le registre transmet la Réponse-à-une-interrogation-ad-hoc à titre de réponse à une Demande-
1811 d'interrogation-ad-hoc.

1812 6.1.2.1 Syntaxe

```
1813 <element name="AdhocQueryResponse">  
1814 <complexType>  
1815 <complexContent>  
1816 <extension base="rs:RegistryResponseType">  
1817 <sequence>  
1818 <element ref="rim:RegistryObjectList" />  
1819 </sequence>  
1820 <attribute default="0" name="startIndex" type="integer"/>  
1821 <attribute name="totalResultCount" type="integer"  
1822 use="optional"/>  
1823 </extension>  
1824 </complexContent>  
1825 </complexType>  
1826 </element>
```

1827 6.1.2.2 Paramètres

- 1828 ▪ **Liste-d'objets-de-registre** : Il s'agit de l'élément qui renferme les instances d'Objet-de-
1829 registre s'appariant à l'interrogation établie.
- 1830 ▪ **index-de-départ** : On se sert de cette valeur entière optionnelle pour déterminer l'index
1831 qui se rattache au premier résultat de l'ensemble de résultats renvoyés par l'interrogation,
1832 dans l'ensemble intégral de résultats qui s'appariant à celle-ci. La valeur par défaut
1833 correspond à 0. La section 6.2.1 présente un exemple à cet égard.
- 1834 ▪ **nombre-total-de-résultats** : Ce paramètre optionnel précise le nombre total de résultats
1835 qui s'appariant à l'interrogation dans le registre. Si cette valeur n'est pas spécifiée,
1836 l'application devrait présumer qu'il s'agit du nombre de résultats en général. La
1837 section 6.2.1 donne un exemple.

1838 6.1.3 Interrogation ad hoc

1839 Toute application spécifie un élément <rim:Interrogation-ad-hoc> dans une Demande-d'interrogation-ad-
1840 hoc pour déterminer l'interrogation réelle inscrite.

1841 6.1.3.1 Syntaxe

```
1842 <complexType abstract="true" name="AdhocQueryType">  
1843 <complexContent>  
1844 <extension base="tns:RegistryObjectType">  
1845 <sequence>  
1846 <element ref="tns:QueryExpression"  
1847 minOccurs="0" maxOccurs="1" />  
1848 </sequence>  
1849 </extension>  
1850 </complexContent>  
1851 </complexType>  
1852 <element name="AdhocQuery" type="tns:AdhocQueryType"  
1853 substitutionGroup="tns:RegistryObject" />
```

1854

1855 6.1.3.2 Paramètres

- 1856 ▪ **expression-d'interrogation** : Cet élément renferme l'expression d'interrogation réelle. Le

1857 schéma utilisé pour l'expression-d'interrogation est extensible et peut accueillir toute
1858 syntaxe de requête gérée par le registre.

1859 6.1.4 Option-de-réponse

1860 Toute application établit une structure d'Option-de-réponse dans une Demande-d'interrogation-ad-hoc
1861 pour désigner le format des résultats au sein de la Réponse-à-une-interrogation-ad-hoc correspondante.

1862

1863 6.1.4.1 Syntaxe

```
1864 <complexType name="ResponseOptionType">  
1865 <attribute default="RegistryObject" name="returnType">  
1866 <simpleType>  
1867 <restriction base="NCName">  
1868 <enumeration value="ObjectRef"/>  
1869 <enumeration value="RegistryObject"/>  
1870 <enumeration value="LeafClass"/>  
1871 <enumeration value="LeafClassWithRepositoryItem"/>  
1872 </restriction>  
1873 </simpleType>  
1874 </attribute>  
1875 <attribute default="false" name="returnComposedObjects"  
1876 type="boolean"/>  
1877 </complexType>  
1878 <element name="ResponseOption" type="tns:ResponseOptionType"/>
```

1879

1880 6.1.4.2 Paramètres

- 1881
- 1882 ▪ **répondre-avec-objet-composé** : Ce paramètre optionnel détermine si les Objets-de-
1883 registre renvoyés devraient comprendre les objets composés, comme le définit le
1884 Diagramme 1 du [ebRIM]. Le défaut consiste à renvoyer l'ensemble des objets
composés.
 - 1885 ▪ **type-de-retour** : Ce paramètre optionnel d'énumération fait connaître le type d'Objet-de-
1886 registre à renvoyer dans la réponse. Les valeurs de type-de-retour sont les suivantes :
 - 1887 • **Référence-d'objet** – Cette option stipule que la Réponse-à-une-interrogation-
1888 ad-hoc DOIT renfermer une collection d'éléments <rim:Référence-d'objet>.
1889 L'option vise à renvoyer des références aux objets de registre plutôt qu'aux
1890 objets réels;
 - 1891 • **Objet-de-registre** – Cette option précise que la Réponse-à-une-interrogation-
1892 ad-hoc DOIT comprendre une collection d'éléments <rim:Objet-de-registre>;
 - 1893 • **Classe-de-nœud-terminal** – Cette option révèle que la Réponse-à-une-
1894 interrogation-ad-hoc DOIT englober une collection d'éléments qui
1895 correspondent aux classes de nœud terminal, comme le définit le [RR-RIM-
1896 XSD];
 - 1897 • **Classe-de-nœud-terminal-comprenant-un-item-de-référentiel** – Cette
1898 option est la même que l'option Classe-de-nœud-terminal, mais englobe en
1899 plus l'exigence stipulant que la réponse doit comprendre les Items-de-
1900 référentiel, le cas échéant, pour chaque élément <rim:Objet-extrinsèque>
1901 figurant dans la réponse.

1902 Si le « type-de-retour » spécifié ne s'apparie à aucun résultat renvoyé par l'interrogation,
1903 le registre *doit* utiliser le type-de-retour sémantiquement valide qui s'apparie le plus au
1904 résultat.

1905 À titre d'exemple, on peut penser à un cas où une Interrogation-d'organisation est tenue
1906 de renvoyer une Classe-de-nœud-terminal-comprenant-un-item-de-référentiel.

1907
1908
1909

Puisqu'une telle situation est impossible, le Gestionnaire-d'interrogation tiendra plutôt compte de l'option Classe-de-nœud-terminal.

1910 **6.2 Soutien d'interrogation itérative**

1911 La Demande-d'interrogation-ad-hoc et la Réponse-à-une-interrogation-ad-hoc gèrent la capacité d'itérer
1912 un ensemble important de résultats s'appariant à une interrogation logique en favorisant l'inscription de
1913 Demandes-d'interrogation-ad-hoc multiples de façon que chaque interrogation demande un sous-
1914 ensemble différent de résultats dans l'ensemble de résultats. Cette particularité permet au registre de
1915 gérer de manière extensible les interrogations qui s'apparient à un vaste ensemble de résultats. On peut
1916 accéder à la caractéristique d'interrogation itérative au moyen des paramètres d'index-de-départ et de
1917 résultats-maximaux de la Demande-d'interrogation-ad-hoc, et au moyen des paramètres d'index-de-
1918 départ et de nombre-total-de-résultats de la Réponse-à-une-interrogation-ad-hoc, comme on l'a décrit
1919 précédemment.

1920 La caractéristique des interrogations itératives ne constitue pas une capacité véritable de curseur comme
1921 celle propre aux bases de données. Le registre n'est pas tenu de maintenir une cohérence ou un état
1922 transactionnels entre les itérations d'une interrogation. Par conséquent, il est possible d'ajouter de
1923 nouveaux objets dans l'ensemble intégral de résultats entre les itérations, ou bien d'en supprimer qui
1924 existent déjà. Il est donc possible d'ignorer ou de dupliquer un élément d'ensemble de résultats entre les
1925 itérations.

1926 Il est important de mentionner qu'on PEUT implanter une caractéristique d'interrogation itérative qui est
1927 cohérente sur le plan transactionnel, même si cette mesure n'est pas requise.

1928 **6.2.1 Exemple d'itération d'interrogation**

1929 Supposons qu'il y a 1 007 Organisations dans un registre. L'utilisateur souhaite inscrire une interrogation
1930 qui s'apparie à ces 1 007 Organisations. Il désire agir ainsi de manière itérative de façon que les
1931 Organisations soient extraites par groupes de 100. Le tableau ci-dessous met en évidence les paramètres
1932 de la Demande-d'interrogation-ad-hoc et ceux de la Réponse-à-une-interrogation-ad-hoc pour chaque
1933 interrogation itérative dans cet exemple.

1934

Paramètres de la Demande-d'interrogation-ad-hoc		Paramètres de la Réponse-à-une-interrogation-ad-hoc		
index-de-départ	résultats-maximaux	index-de-départ	nombre-total-de-résultats	Nombre de résultats
0	100	0	1 007	100
100	100	100	1 007	100
200	100	200	1 007	100
300	100	300	1 007	100
400	100	400	1 007	100
500	100	500	1 007	100
600	100	600	1 007	100
700	100	700	1 007	100
800	100	800	1 007	100
900	100	900	1 007	100
1 000	100	1 000	1 007	7

1935

1936 **6.3 Soutien d'interrogation stockée**

1937 Le protocole d'Interrogation-ad-hoc permet aux applications d'inscrire des interrogations qui peuvent être
1938 aussi générales ou aussi précises que l'exige le cas d'utilisation. Plus les interrogations sont précises,
1939 plus elles sont complexes. Dans une telle situation, il est souhaitable de masquer la complexité de

1940 l'interrogation de sorte qu'elle passe inaperçue pour l'application qui se sert des interrogations
1941 paramétrées stockées dans le registre. Au moment d'utiliser ces interrogations, l'application est
1942 seulement tenue de révéler l'identité et les paramètres de l'interrogation plutôt que l'expression de
1943 requête.

1944 Les interrogations paramétrées stockées s'avèrent utiles pour les administrateurs de registre puisqu'elles
1945 offrent un mécanisme global aux utilisateurs de registre et favorisent le partage d'un ensemble
1946 d'interrogations courantes.

1947 Les interrogations paramétrées stockées sont également utiles pour les standards verticaux parce que
1948 ceux-ci peuvent définir les interrogations paramétrées propres à un domaine et qu'ils exigent leur
1949 stockage dans le registre.

1950 Les registres ebXML DOIVENT pouvoir accueillir des interrogations paramétrées stockées, comme le
1951 définit la présente section.

1952 **6.3.1 Inscription d'une interrogation stockée**

1953 On procède à l'inscription d'une interrogation stockée par le biais du protocole standard lié à la Demande-
1954 d'inscription-d'objet. L'objet inscrit représente une instance de Type-d'interrogation-ad-hoc.

1955 **6.3.1.1 Déclaration de paramètres d'interrogation**

1956 Au moment d'inscrire une interrogation stockée, le demandeur PEUT déclarer plusieurs paramètres pour
1957 cette interrogation comme il PEUT n'en déclarer aucun. On DOIT déclarer un paramètre en ayant recours
1958 à un nom de paramètre qui commence par le caractère « \$ » et qui est suivi immédiatement d'une lettre
1959 et, ensuite, d'une combinaison quelconque de lettres et de nombres. La FBN suivante définit la façon dont
1960 un nom de paramètre DOIT être déclaré :

1961

```
1962 QueryParameter := '$' [a-zA-Z] ( [a-zA-Z] | [0-9] )*
```

1963

1964 Un paramètre d'interrogation PEUT être utilisé à titre de nom général pour n'importe quelle partie de
1965 l'interrogation stockée.

1966 L'exemple ci-dessous démontre comment une interrogation paramétrée stockée peut être inscrite.

1967

```
1968 <SubmitObjectsRequest>  
1969   <rim:RegistryObjectList>  
1970     <rim:AdhocQuery id="{QUERY_ID}">  
1971       <rim:QueryExpression queryLanguage="{SQL_QUERY_LANG_ID}">  
1972         SELECT * from $tableName ro, Name_ nm, Description d  
1973         WHERE  
1974         objectType = '$objectType'  
1975         AND (nm.parent = ro.id AND UPPER ( nm.value ) LIKE UPPER  
1976         ( '$name' ) )  
1977         AND (d.parent = ro.id AND UPPER ( d.value ) LIKE UPPER  
1978         ( '$description' ) )  
1979         AND (ro.id IN ( SELECT classifiedObject FROM Classification WHERE  
1980         classificationNode IN ( SELECT id  
1981         FROM ClassificationNode WHERE path LIKE '$classificationPath1%'  
1982         ) ) )  
1983       </rim:QueryExpression>  
1984     </rim:AdhocQuery>  
1985   </rim:RegistryObjectList>  
1986 </SubmitObjectsRequest>
```

1987 **Liste 1 : Exemple d'inscription d'interrogation stockée**

1988

1989 L'interrogation ci-dessus utilise les paramètres *\$objectType*, *\$name*, *\$description* et *\$classificationPath1*,
1990 et recherche tous les objets qui s'appartiennent au type-d'objet, au nom, à la description et à la classification.

1991 **6.3.1.2 Paramètres de contexte canonique**

1992 Toute interrogation PEUT renfermer au moins un paramètre de contexte, comme le définit la présente
1993 section. Les paramètres de contexte constituent des paramètres d'interrogation particuliers dont la valeur
1994 n'a pas à être gérée par l'application. La valeur propre à un paramètre de contexte provient plutôt du
1995 registre selon le contexte dans lequel l'interrogation de l'application est traitée.

1996 Au moment du traitement d'une interrogation, tout registre DOIT remplacer l'ensemble des paramètres de
1997 contexte de l'interrogation par la valeur sensible au contexte du paramètre en question. Tout registre
1998 DOIT faire abstraction de toute valeur de paramètre de contexte fournie par l'application.

1999

Paramètre de contexte	Valeur de remplacement
\$currentUser	Doit être remplacé par l'attribut identifiant de l'utilisateur lié à l'interrogation.
\$currentTime	Doit être remplacé par le temps-actuel. La base horaire est la même que celle établie pour l'attribut horodatation de la classe Événement-vérifiable.

2000

2001 **6.3.2 Appel d'une interrogation stockée**

2002 On appelle une interrogation stockée en se servant de la Demande-d'interrogation-ad-hoc et en tenant
2003 compte des contraintes suivantes :

- 2004 • L'élément <rim:Interrogation-ad-hoc> NE DOIT PAS comprendre d'élément <rim:Expression-
2005 d'interrogation>;
- 2006 • La valeur de l'attribut identifiant de l'élément <rim:Interrogation-ad-hoc> DOIT s'apparier à celle de
2007 l'attribut identifiant de l'interrogation stockée;
- 2008 • L'élément <rim:Interrogation-ad-hoc> PEUT renfermer une Place-d'attribut pour chaque paramètre non
2009 contextuel défini pour l'interrogation stockée appelée. Les Places-d'attribut de ce genre présentent la
2010 valeur qui s'applique aux paramètres d'interrogation.

2011 **6.3.2.1 Établissement des paramètres d'appel d'interrogation**

2012 Toute interrogation stockée PEUT être définie à l'aide de plusieurs paramètres comme elle PEUT n'en
2013 nécessiter aucun. Une application peut spécifier plusieurs des paramètres établis pour l'interrogation
2014 stockée, comme elle peut n'en préciser aucun, au moment de l'inscription de la Demande-d'interrogation-
2015 ad-hoc pour l'interrogation stockée. Il est important de mentionner que l'application PEUT déterminer
2016 moins de paramètres que ceux déclarés pour l'interrogation stockée. Enfin, les registres DOIVENT
2017 retrancher tout prédicat de l'interrogation stockée qui renferme des paramètres non inscrits par
2018 l'application pendant l'appel de cette interrogation.

2019 L'application peut essentiellement restreindre ou élargir la spécificité de la recherche en fournissant un
2020 nombre plus élevé ou moins élevé de paramètres.

2021 Les applications précisent un paramètre d'appel d'interrogation en utilisant une Place-d'attribut dont le
2022 nom s'apparie au nom du paramètre et dont la valeur DOIT constituer une seule valeur s'appariant à la
2023 valeur déterminée du paramètre.

2024 Les registres NE DOIVENT tenir compte d'aucun paramètre spécifié par l'application pour une
2025 interrogation stockée qui ne s'apparie pas aux paramètres établis par celle-ci.

2026 La liste ci-dessous présente un exemple de la manière dont l'interrogation stockée évoquée
2027 précédemment est appelée. Elle illustre :

- 2028 • l'interrogation stockée déterminée par la valeur de l'attribut identifiant de l'élément <rim:Interrogation-
2029 ad-hoc>;
- 2030 • la valeur du paramètre \$name produit;
- 2031 • la valeur des autres paramètres établis par l'interrogation, mais non fournis (ce qui démontre que
2032 l'application ne souhaite pas se servir de ces paramètres à titre de critères de recherche).

2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045

```
<AdhocQueryRequest>
  <query:ResponseOption returnComposedObjects="true"
returnType="LeafClassWithRepositoryItem"/>
  <rim:AdhocQuery id="{STORED_QUERY_ID}">
    <rim:Slot name="$name">
      <rim:ValueList>
        <rim:Value>%ebXML%</rim:Value>
      </rim:ValueList>
    </rim:Slot>
  </rim:AdhocQuery>
</AdhocQueryRequest>
```

Liste 2 : Exemple d'appel d'interrogation stockée

2046 6.3.3 Réponse à un appel d'interrogation stockée

2047 Tout registre DOIT transmettre une Réponse-à-une-interrogation-ad-hoc normalisée si une application
2048 appelle une interrogation stockée au moyen d'une Demande-d'interrogation-ad-hoc.

2049 6.3.4 Contrôle d'accès à une interrogation stockée

2050 Une interrogation stockée représente un Objet-de-registre. Comme pour tous les Objets-de-registre,
2051 l'accès à l'interrogation stockée est géré par la règle de contrôle d'accès propre à cette interrogation. On
2052 attribut à une interrogation stockée la règle de contrôle d'accès par défaut qui permet à toute application
2053 de lire et d'appeler cette interrogation. Seuls le détenteur de l'interrogation et l'administrateur de registre
2054 ont comme rôle de mettre à jour ou de supprimer l'interrogation. Le détenteur de l'interrogation peut définir
2055 une règle contextualisée de contrôle d'accès qui réduit la visibilité de cette interrogation et la capacité de
2056 l'appeler. Cette règle peut s'adresser à des utilisateurs, rôles ou groupes précis. Par conséquent, le
2057 détenteur de l'interrogation ou l'administrateur de registre peuvent contrôler *qui* peut appeler *quelles*
2058 interrogations stockées.

2059 6.3.5 Interrogation canonique : obtention de l'objet d'utilisateur d'une 2060 application

2061 Tout registre DOIT pouvoir accueillir une interrogation stockée canonique à l'aide de l'interrogation
2062 suivante :

2063 `id="urn:oasis:names:tc:ebxml-regrep:query:GetCallersUser"`

2064 Cette interrogation DOIT renvoyer l'objet d'utilisateur associé à l'application qui appelle l'interrogation
2065 stockée. L'application NE DOIT fournir aucun paramètre pour cette interrogation. L'interrogation stockée
2066 DEVRAIT se servir du paramètre de contexte canonique `$currentUser`.

2067 Voici un exemple non normatif de requête SQL stockée que tout registre PEUT utiliser pour l'interrogation
2068 stockée canonique en question :

2069

```
<rim:AdhocQuery id="urn:oasis:names:tc:ebxml-  
regrep:query:GetCallersUser">  
  <rim:QueryExpression  
    queryLanguage="urn:oasis:names:tc:ebxml-regrep:QueryLanguage:SQL-92">  
    SELECT u.* FROM User u WHERE u.id = $currentUser;  
  </rim:QueryExpression>  
</rim:AdhocQuery>
```

2077 Il est important de mentionner que tout registre PEUT avoir recours à un filtre d'interrogation stockée
2078 équivalente plutôt qu'une requête SQL stockée.

2079 6.4 Syntaxe de requête SQL

2080 Tout registre ebXML PEUT accueillir le SQL comme syntaxe de requête gérée dans l'élément

2081 <rim:Expression-d'interrogation> de la Demande-d'interrogation-ad-hoc. La présente section présente une
2082 définition normative de la syntaxe SQL qu'un registre ebXML PEUT gérer. Il faut mentionner que le
2083 support de syntaxe SQL d'un registre ne laisse pas entendre que le registre doit utiliser une base de
2084 données relationnelle dans le cadre de son implantation.

2085 La syntaxe SQL de registre constitue un sous-ensemble adéquat de l'énoncé « SÉLECTIONNER »
2086 du SQL de base, comme le définit la norme ISO/IEC 9075:1992 (Langage de bases de données SQL
2087 [SQL]).

2088 Les termes ci-dessous qui figurent entre crochets en chevron sont définis dans le [SQL] ou le [SQL/PSM].
2089 La syntaxe de requête SQL est conforme à la <spécification de requête> et englobe les restrictions
2090 additionnelles suivantes :

2091 37. Une <colonne dérivée> PEUT ne pas avoir de <clause AS>;

2092 38. Une <expression de table> ne renferme pas les clauses optionnelles <clause GROUP BY> et
2093 <clause HAVING>;

2094 39. Une <référence de table> ne peut englober qu'un <nom de table> et un <nom de corrélation>;

2095 40. Une <référence de table> se caractérise par l'absence de l'opérateur AS optionnel entre le <nom de
2096 table> et le <nom de corrélation>;

2097 41. La syntaxe favorise l'utilisation restreinte de sous-requêtes comme suit : la partie droite du <prédicat
2098 dans> peut se limiter à une <spécification de requête>, comme il est décrit ci-dessus.

2099 Comme le définit le [SQL], tout registre DOIT traiter les noms de table et d'attribut de manière insensible à
2100 la casse.

2101 **6.4.1 FBN propre à la grammaire de la syntaxe de requête (non** 2102 **normative)**

2103 La FBN ci-dessous authentifie la grammaire qui s'applique à la syntaxe de requête de registre. Elle agit ici
2104 comme un outil à l'intention des implanteurs. La FBN n'est pas fondée directement sur le [SQL]; par
2105 conséquent, elle figure à titre de syntaxe non normative.

2106

```
query_exp ::= ( query_term ( <UNION> ( <ALL> )? query_term )? )
query_term ::= ( SQLSelect | "(" query_exp ")" )
SQLSelect ::= <SELECT> SQLSelectCols <FROM>
             SQLTableList ( SQLWhere )? ( SQLOrderBy )?
SQLSelectCols ::= ( <ALL> | <DISTINCT> ) * ( ( "*" | SQLLvalueTerm ) )
SQLTableList ::= SQLTableRef ( "," SQLTableRef ) *
SQLTableRef ::= ( <ID> ( <ID> )? )
SQLWhere ::= <WHERE> SQLOrExpr
SQLOrExpr ::= SQLAndExpr ( <OR> SQLAndExpr ) *
SQLAndExpr ::= SQLNotExpr ( <AND> SQLNotExpr ) *
SQLNotExpr ::= ( <NOT> )? SQLCompareExpr
SQLCompareExpr ::= ( SQLIsClause |
SQLSumExpr ( SQLCompareExprRight )? )
SQLCompareExprRight ::= ( SQLLikeClause | SQLInClause
| SQLCompareOp SQLSumExpr )
( <EQUAL> | <NOTEQUAL> | <NOTEQUAL2>
SQLCompareOp ::= | <GREATER> | <GREATEREQUAL>
| <LESS> | <LESSEQUAL> )
SQLFunction ::= ( <UPPER> SQLFunctionArgs )
```

```

SQLFunctionArgs ::= "(" ( SQLSumExpr ( "," SQLSumExpr ) * )? ")"
SQLInClause ::= (<NOT> )?
               <IN> "(" SQLLValueListOrProcedureCall ")"
SQLLValueListOrProcedureCall ::= ( ProcedureCall | SQLLValueList )
ProcedureCall ::= <ID> "(" <STRING_LITERAL> ")"
SQLLValueList ::= SQLLValueElement ( "," SQLLValueElement ) *
SQLLValueElement ::= (<NULL> | SQLSumExpr | SQLSelect )
SQLIsClause ::= SQLColRef <IS> (<NOT> )? <NULL>
SQLLikeClause ::= (<NOT> )? <LIKE> SQLPattern
SQLPattern ::= (<STRING_LITERAL> | "?" | SQLLvalue | SQLFunction )
SQLColRef ::= SQLLvalue
SQLLvalue ::= ( SQLLvalueTerm )
SQLLvalueTerm ::= <ID> (<DOT> idOrStar ) *
idOrStar ::= (<ID> | "*" )
SQLSumExpr ::= SQLProductExpr ( ( "+" | "-" ) SQLProductExpr ) *
SQLProductExpr ::= SQLUnaryExpr ( ( "*" | "/" ) SQLUnaryExpr ) *
SQLUnaryExpr ::= ( "+" | "-" )? SQLTerm
SQLTerm ::= ( "(" SQLOrExpr ")" | SQLColRef
           | SQLLiteral | SQLFunction )
SQLLiteral ::= (<STRING_LITERAL> | <INTEGER_LITERAL>
              | <FLOATING_POINT_LITERAL> )
SQLOrderBy ::= <ORDER> <BY> SQLOrderByList
SQLOrderByElem ::= SQLColRef ( SQLOrderDirection )?
SQLOrderByList ::= SQLOrderByElem ( "," SQLOrderByElem ) *
SQLOrderDirection ::= (<ASC> | <DESC> )

```

2107 6.4.2 Schéma relationnel propre aux requêtes SQL

2108 La définition normative de schéma relationnel qui se traduit par la cible des requêtes SQL de registre se
2109 trouve sur le site Web suivant :

2110 <http://www.oasis-open.org/committees/regrep/documents/3.0/sql/database.sql>

2111 6.4.3 Résultats de requête SQL

2112 Le résultat d'une requête SQL prend la forme d'une collection d'objets de registre. Il ne se traduit jamais
2113 par des attributs partiels. Les objets qui se rattachent à l'ensemble de résultats peuvent être renvoyés en
2114 tant que Références-d'objet, Objets-de-registre ou Classes-de-nœud-terminal selon l'attribut type-de-
2115 retour du paramètre d'option-de-réponse établi par l'application dans la Demande-d'interrogation-ad-hoc.
2116 L'ensemble intégral de résultats est renvoyé à titre d'élément <rim:Liste-d'objets-de-registre>.

2117 6.5 Syntaxe de filtre d'interrogation

2118 La présente section définit de façon normative une syntaxe XML en vue de l'interrogation d'un
2119 registre ebXML appelé « syntaxe de *filtre d'interrogation* ». Tout registre ebXML DOIT pouvoir accueillir
2120 cette syntaxe comme syntaxe d'interrogation gérée dans l'élément <rim:Expression-d'interrogation> de la
2121 Demande-d'interrogation-ad-hoc.

2122 La syntaxe de filtre d'interrogation figure dans le [RR-QUERY-XSD] et provient d'un appariement entre le
2123 [ebRIM] et le schéma XML selon certaines caractéristiques d'appariement.

2124 Le modèle opérationnel de filtre d'interrogation considère le réseau d'Objets-de-registre du registre
2125 comme un document XML virtuel. Une interrogation balaie une partie précise de l'arbre en plus de

2126 retrancher ou de filtrer des objets du document virtuel par le biais d'expressions de filtre. L'interrogation
2127 renvoie enfin une collection d'objets retenus après l'élimination par filtrage de tous les objets qui ne
2128 s'apparient pas aux filtres établis dans cette interrogation.

2129 Contrairement à la syntaxe de requête SQL, la syntaxe de filtre d'interrogation ne peut pas accueillir
2130 d'union entre les classes. Cette situation restreint les capacités expressives de l'interrogation et peut
2131 diminuer en quelque sorte l'efficacité du traitement.

2132 6.5.1 Structure de filtre d'interrogation

2133 L'élément <rim:Expression-d'interrogation> de la Demande-d'interrogation-ad-hoc DOIT renfermer un
2134 élément d'*interrogation* qui découle du type <query:RegistryObjectQueryType>.

2135 Tout élément d'interrogation PEUT comprendre un élément <query:PrimaryFilter> ainsi que des éléments
2136 additionnels de filtre, de branche et d'interrogation, comme l'illustre l'exemple abstrait ci-dessous. Le
2137 schéma normatif ici est défini par le [RR-QUERY-XSD].

```
2138  
2139 <${QueryElement}>  
2140 <PrimaryFilter ... />  
2141 <${OtherFilterElement} ... />  
2142 <${BranchElement} ... />  
2143 <${QueryElement} ... />  
2144 </${QueryElement}>
```

2145

2146 Le rôle des éléments d'interrogation, de filtre et de branche sera précisé ci-dessous.

2147 6.5.2 Éléments d'interrogation

2148 L'élément d'interrogation constitue l'élément de premier niveau de la syntaxe de filtre d'interrogation qui
2149 vise l'interrogation du registre. Le schéma XML du [RR-QUERY-XSD] détermine un élément
2150 d'interrogation pour la classe Objet-de-registre et l'ensemble de ses classes descendantes, comme le
2151 définit le [ebRIM], au moyen de la forme générale suivante :

- 2152 • Chaque classe descendante de la classe Objet-de-registre définit un type-complexe appelé Type-
2153 d'interrogation <classe>. Par exemple, un type-complexe de Type-d'interrogation-d'organisation est
2154 défini pour la classe Organisation dans le [ebRIM];
- 2155 • Le Type-d'interrogation d'une classe descendante de la classe Objet-de-registre DOIT élargir le Type-
2156 d'interrogation pour sa classe maîtresse. Par exemple, le Type-d'interrogation-d'organisation accroît le
2157 Type-d'interrogation-d'objet-de-registre;
- 2158 • La classe Objet-de-registre et chacune de ses classes descendantes définissent un élément appelé
2159 Interrogation <classe> et dont le type est désigné sous le nom de Type-d'interrogation <classe>. Par
2160 exemple, l'élément de l'Interrogation-d'organisation est défini à l'aide du type Type-d'interrogation-
2161 d'organisation.

2162 La classe associée à un élément d'interrogation est dénommée *classe du domaine d'interrogation*.

2163 L'exemple suivant illustre la syntaxe d'interrogation où la classe du domaine d'interrogation constitue la
2164 classe Organisation établie dans le [ebRIM] :

2165

```
2166 <complexType name="OrganizationQueryType">  
2167 <complexContent>  
2168 <extension base="tns:RegistryObjectQueryType">  
2169 ...Relevant Filters, Queries and Branches are defined here...  
2170 </extension>  
2171 </complexContent>  
2172 </complexType>  
2173 <element name="OrganizationQuery" type="tns:OrganizationQueryType"/>
```

2174

2175 Tout élément d'interrogation PEUT disposer d'éléments de filtre, de branche ou d'interrogation
2176 concaténée. Ces éléments sont décrits dans des sections ultérieures.

2177 6.5.3 Éléments de filtre

2178 Tout élément d'interrogation PEUT renfermer au moins un sous-élément de filtre. On se sert d'un élément
2179 de filtre pour *filtrer* ou sélectionner un sous-ensemble d'instances d'une classe particulière du [ebRIM]. La
2180 classe filtrée par un filtre est désignée sous l'appellation de *classe du domaine de filtre*. Un élément de
2181 filtre établit une clause de prédicat restreinte par rapport aux attributs de la classe du domaine de filtre.

2182 Le schéma XML du [RR-QUERY-XSD] spécifie plusieurs éléments de filtre contenus dans une définition
2183 d'élément d'interrogation, comme il peut n'en spécifier aucun. Pour ce faire, il a recours à la forme
2184 générale suivante :

- 2185 • **Filtre-primaire** : Un élément de filtre est établi à l'intérieur d'un Type-d'interrogation-d'objet-de-registre
2186 portant le nom de *Filtre-primaire*. On a recours à ce filtre pour filtrer les instances de la classe du
2187 domaine d'interrogation à partir de la valeur de ses attributs de primitive. La cardinalité de l'élément de
2188 filtre est de zéro ou d'un. L'élément de *Filtre-primaire* est hérité par l'ensemble des Types-
2189 d'interrogation descendants du Type-d'interrogation-d'objet-de-registre;
 - 2190 • **Filtres additionnels** : On se sert des filtres additionnels d'un élément d'interrogation pour filtrer les
2191 instances de la classe du domaine d'interrogation en fonction de la question à savoir si l'instance
2192 candidate de la classe du domaine dispose d'un objet cité qui respecte le filtre additionnel.
2193 On détermine des éléments de filtre additionnel pour les attributs de la classe du domaine
2194 d'interrogation qui répondent à l'ensemble des critères suivants :
 - 2195 • Le domaine d'attribut ne constitue pas un type de primitive (par exemple, une chaîne, un nombre à
2196 point flottant, l'élément date-et-heure, l'élément entier, etc.);
 - 2197 • La classe du domaine d'attribut ne représente pas un Objet-de-registre ou son descendant;
 - 2198 • La classe du domaine d'attribut ne dispose d'aucun attribut de référence (dans le cas contraire, on
2199 doit se servir d'une branche ou d'une sous-interrogation);
- 2200 L'attribut pour lequel le filtre est défini est appelé attribut du domaine de filtre. La classe du domaine
2201 propre à l'attribut du domaine de filtre désigne la classe du domaine de filtre qui s'applique aux
2202 filtres de ce genre. On utilise ce type de filtre pour filtrer les instances de la classe du domaine
2203 d'interrogation en s'appuyant sur les valeurs d'attribut de la classe du domaine de filtre.
- 2204 • L'élément de filtre s'appelle <nom de l'attribut du domaine de filtre>;
 - 2205 • Le type d'élément de filtre se traduit par le type complexe de Type-de-filtre, qui est décrit dans la
2206 section ;
 - 2207 • La cardinalité de l'élément de filtre s'apparie à celle de l'attribut du domaine de filtre de la classe du
2208 domaine d'interrogation.

2209

2210 L'exemple ci-dessous démontre comment le schéma XML du [RR-QUERY-XSD] utilise la forme générale
2211 ci-dessus pour déterminer les filtres du Type-d'interrogation-d'organisation propre à la classe Organisation
2212 traitée dans le [ebRIM].

2213

```
2214 <complexType name="OrganizationQueryType">  
2215   <complexContent>  
2216     <extension base="tns:RegistryObjectQueryType">  
2217       <sequence>  
2218         <element maxOccurs="unbounded" minOccurs="0"  
2219           name="AddressFilter" type="tns:FilterType"/>  
2220         <element maxOccurs="unbounded" minOccurs="0"  
2221           name="TelephoneNumberFilter" type="tns:FilterType"/>  
2222         <element maxOccurs="unbounded" minOccurs="0"  
2223           name="EmailAdresseFilter" type="tns:FilterType"/>  
2224         ...Branches and sub-Queries go here...  
2225       </sequence>  
2226     </extension>  
2227   </complexContent>  
2228 </complexType>
```

2229

2230 Le diagramme de classe UML ci-dessous décrit la structure de la classe de filtre, comme le définit le

2231 schéma XML du [RR-QUERY-XSD]. Il est important de mentionner que les classes dont le nom se
2232 termine par « type » s'apparient aux types-complexes, tandis que que les autres classes de filtre
2233 s'apparient aux éléments établis dans le schéma XML du [RR-QUERY-XSD].

2234

2235

2236

Diagramme 12 : Hiérarchie du type de filtre

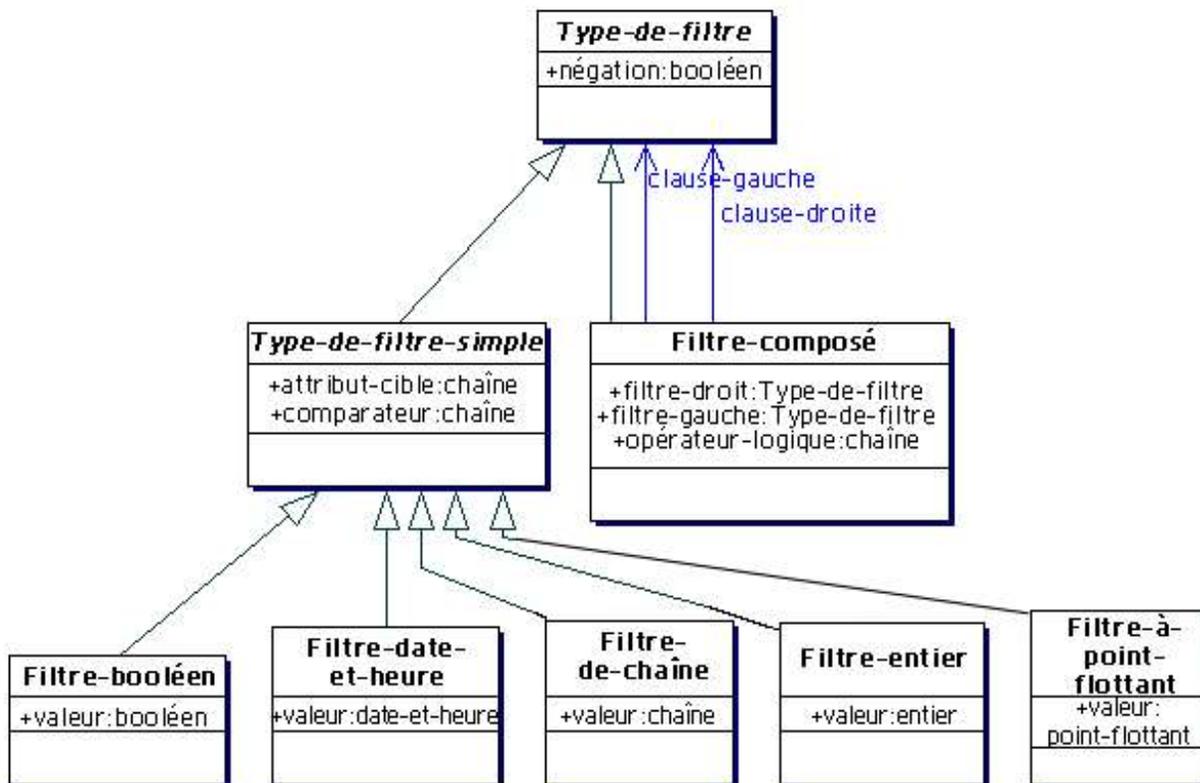
2237

2238 6.5.3.1 Type-de-filtre

2239 Le Type-de-filtre constitue un type-complexe abstrait qui se traduit par le type racine de la hiérarchie
2240 d'héritage propre à l'ensemble des types de filtre.

2241 6.5.3.1.1 Paramètres

- 2242
- 2243
- 2244
- 2245
- **négation** : Ce paramètre précise qu'on DOIT assurer la négation de la valeur booléenne faisant l'objet d'une évaluation par le filtre afin de compléter cette évaluation. Le paramètre est équivalent sur le plan de la fonctionnalité à l'opérateur NON de la syntaxe de SQL.



2246 6.5.3.2 Type-de-filtre-simple

2247 Le Filtre-simple constitue le type de base abstrait propre à plusieurs types de filtre concrets qui sont
2248 définis pour un type de primitive, comme un attribut booléen, un nombre à point flottant, un entier et une
2249 chaîne.

2250 6.5.3.2.1 Paramètres

- 2251
- 2252
- **attribut-de-domaine** : Ce paramètre désigne expressément le nom de l'attribut de primitive de la classe du domaine de filtre. Tout registre DOIT renvoyer une Exception-de-

2253 requête-invalidé si la valeur de ce paramètre ne s'apparie pas au nom de l'attribut de
2254 primitive de la classe du domaine de filtre. Tout registre DOIT exécuter l'appariement des
2255 noms d'attribut de manière insensible à la casse.

2256 ▪ **comparateur** : Ce paramètre spécifie l'opérateur de comparaison entre la valeur de
2257 l'attribut et celle transmise par le filtre. On a fixé les comparateurs suivants :

- 2258 • IE : abréviation de Inférieur-ou-égal-à;
 - 2259 • IA : abréviation de Inférieur-à;
 - 2260 • SE : abréviation de Supérieur-ou-égal-à;
 - 2261 • SA : abréviation de Supérieur-à;
 - 2262 • EA : abréviation de Égal-à;
 - 2263 • DD : abréviation de Différent-de;
 - 2264 • LIKE : identique à l'opérateur LIKE du standard SQL-92; ne DOIT être utilisé que
2265 dans le Filtre-de-chaîne;
 - 2266 • NOTLIKE : identique à l'opérateur NOT LIKE du standard SQL-92; ne DOIT être
2267 utilisé que dans le Filtre-de-chaîne.
- 2268

2269 6.5.3.3 Filtre-booléen

2270 On ne DOIT se servir du Filtre-booléen que pour apparier les attributs de primitive dont le domaine est de
2271 type booléen.

2272 6.5.3.3.1 Paramètres

2273 ▪ **valeur** : Ce paramètre détermine la valeur que l'on DOIT comparer avec la valeur
2274 d'attribut mise à l'essai par le filtre. Il DOIT s'agir d'une valeur booléenne.

2275 L'exemple suivant illustre l'utilisation d'un Filtre-booléen pour en assurer l'appariement avec l'attribut *est-*
2276 *interne* de la classe Schème-de-classification définie dans le [ebRIM] :

```
2277 <BooleanFilter  
2278     domainAttribute="isInternal" comparator="EQ" value="true"/>
```

2279

2280 6.5.3.4 Filtre-à-point-flottant

2281 On ne DOIT utiliser le Filtre-à-point-flottant que pour apparier les attributs de primitive dont le domaine est
2282 de type flottant.

2283 6.5.3.4.1 Paramètres

2284 ▪ **valeur** : Ce paramètre désigne expressément la valeur que l'on DOIT comparer avec la
2285 valeur d'attribut mise à l'essai par le filtre. Il DOIT s'agir d'une valeur flottante.

2286 L'exemple ci-dessous démontre l'utilisation d'un Filtre-à-point-flottant pour en assurer l'appariement avec
2287 un attribut *montant* fictif à point flottant; aucun attribut à point flottant n'a encore été défini pour le [ebRIM].

```
2288 <FloatFilter  
2289     domainAttribute="amount" comparator="GT" value="9.99"/>
```

2290

2291 6.5.3.5 Filtre-entier

2292 On ne DOIT avoir recours au Filtre-entier que pour apparier les attributs de primitive dont le domaine est
2293 de type entier.

2294 6.5.3.5.1 Paramètres

- 2295 ▪ **valeur** : Ce paramètre met en évidence la valeur que l'on DOIT comparer avec la valeur
2296 d'attribut mise à l'essai par le filtre. Il DOIT s'agir d'une valeur entière.

2297 L'exemple ci-dessous révèle l'utilisation d'un Filtre-entier pour en assurer l'appariement avec un attribut
2298 *compte* fictif avec entier; aucun attribut entier n'a encore été défini pour le [ebRIM].

```
2299 <IntegerFilter  
2300     domainAttribute="amount" comparator="LT" value="100"/>
```

2301

2302 6.5.3.6 Filtre-date-et-heure

2303 On ne DOIT avoir recours au Filtre-date-et-heure que pour apparier les attributs de primitive dont le
2304 domaine est de type date-et-heure.

2305 6.5.3.6.1 Paramètres

- 2306 ▪ **valeur** : Ce paramètre présente la valeur que l'on DOIT comparer avec la valeur d'attribut
2307 mise à l'essai par le filtre. Il DOIT s'agir d'une valeur de date-et-heure.

2308 L'exemple ci-dessous illustre l'utilisation d'un Filtre-date-et-heure pour en assurer l'appariement avec
2309 l'attribut *horodotation* de la classe Événement-vérifiable définie dans le [ebRIM], où la valeur
2310 d'horodotation est supérieure à la valeur de date-et-heure établie (ou est plus tardive).

```
2311 <DateTimeFilter  
2312     domainAttribute="timestamp"  
2313     comparator="GT" value="1997-07-16T19:20+01:00"/>
```

2314

2315 6.5.3.7 Filtre-de-chaîne

2316 On ne DOIT utiliser le Filtre-de-chaîne que pour apparier les attributs de primitive dont le domaine est de
2317 type chaîne.

2318 6.5.3.7.1 Paramètres

- 2319 ▪ **valeur** : Ce paramètre spécifie la valeur que l'on DOIT comparer avec la valeur d'attribut
2320 mise à l'essai par le filtre. Il DOIT s'agir d'une valeur de chaîne.

2321 L'exemple ci-dessous démontre l'utilisation d'un Filtre-de-chaîne pour en assurer l'appariement avec
2322 l'attribut *prénom* de la classe Personne définie dans le [ebRIM], où la valeur de prénom s'apparie à la
2323 forme générale déterminée par la valeur.

```
2324 <StringFilter  
2325     domainAttribute="firstName"  
2326     comparator="Like" value="Farid%"/>
```

2327

2328 6.5.3.8 Filtre-composé

2329 On PEUT se servir du Filtre-composé pour désigner une intersection booléenne (ET) ou une union (OU)
2330 entre deux filtres. Il permet à une interrogation d'exprimer une combinaison de clauses de prédicat au
2331 sein d'un filtre d'interrogation.

2332 6.5.3.8.1 Paramètres

- 2333 ▪ **Filtre-gauche** : Ce paramètre définit le premier de deux filtres propres au Filtre-
2334 composé.
- 2335 ▪ **Filtre-droit** : Ce paramètre spécifie le deuxième de deux filtres qui se rattachent au

2336 Filtre-composé.

2337 ▪ **opérateur-logique** : Ce paramètre désigne l'opérateur logique. Il DOIT avoir la valeur
2338 « ET » ou « OU ».

2339 L'exemple ci-dessous caractérise l'utilisation d'un Filtre-composé pour en assurer l'appariement avec
2340 l'attribut *est-interne* de la classe Schème-de-classification définie dans le [ebRIM].

```
2341 <CompoundFilter logicalOperator="AND">  
2342   <LeftFilter domainAttribute="targetObject" comparator="EQ"  
2343     value="{REGISTRY_OBJECT_ID}" type="StringFilter"/>  
2344   <RightFilter domainAttribute="associationType" comparator="EQ"  
2345     value="{HAS_MEMBER_ASSOC_TYPE_NODE_ID}" type="StringFilter"/>  
2346 </CompoundFilter>
```

2347 6.5.4 Éléments d'interrogation concaténée

2348 Tout élément d'interrogation PEUT renfermer au moins un sous-élément d'interrogation concaténée. Les
2349 éléments d'interrogation concaténée ont comme objectif de favoriser le balayage des branches du réseau
2350 de relations défini dans le modèle d'information, et d'élaguer ou de filtrer les branches qui ne sont pas
2351 conformes aux prédicats établis dans l'élément de branche correspondant.

2352 Le schéma XML du [RR-QUERY-XSD] spécifie plusieurs éléments d'interrogation concaténée contenus
2353 dans une définition d'élément d'interrogation, comme il peut n'en spécifier aucun. Pour ce faire, il a
2354 recours à la forme générale suivante :

- 2355 • On définit un élément d'interrogation concaténée pour chaque attribut de la classe du domaine
2356 d'interrogation qui répond à l'ensemble des critères suivants :
 - 2357 • La classe du domaine d'attribut constitue un type de descendant du Type-d'objet-de-registre;
 - 2358 • Cette classe renferme des attributs de référence qui l'associent à une tierce classe au moyen de la
2359 référence;
- 2360 L'attribut pour lequel l'interrogation concaténée est déterminée est désigné sous l'appellation
2361 d'attribut du domaine d'interrogation concaténée. La classe du domaine propre à cet attribut se
2362 traduit par la classe du domaine d'interrogation qui s'applique à l'élément d'interrogation
2363 concaténée.
- 2364 • L'élément d'interrogation concaténée s'appelle interrogation <nom de l'attribut du domaine
2365 d'interrogation concaténée>;
- 2366 • Le type d'élément d'interrogation concaténée s'apparie au Type-d'interrogation particulier à la classe
2367 du domaine qui s'applique à l'attribut du domaine d'interrogation;
- 2368 • La cardinalité de l'élément d'interrogation concaténée s'apparie à celle de l'attribut du domaine
2369 d'interrogation concaténée qui se rattache à la classe du domaine d'interrogation.

2370 L'exemple ci-dessous démontre la façon dont le schéma XML du [RR-QUERY-XSD] se sert de la forme
2371 générale ci-dessus pour établir les éléments d'interrogation concaténée du Type-d'interrogation-
2372 d'organisation propre à la classe Organisation définie dans le [ebRIM].

```
2373  
2374 <complexType name="OrganizationQueryType">  
2375   <complexContent>  
2376     <extension base="tns:RegistryObjectQueryType">  
2377       <sequence>  
2378         ...Filters and Branches go here ...  
2379         <element maxOccurs="1" minOccurs="0"  
2380           name="ParentQuery" type="tns:OrganizationQueryType"/>  
2381         <element maxOccurs="unbounded" minOccurs="0"  
2382           name="ChildOrganizationQuery" type="tns:OrganizationQueryType"/>  
2383         <element maxOccurs="1" minOccurs="0"  
2384           name="PrimaryContactQuery" type="tns:PersonQueryType"/>  
2385       </sequence>  
2386     </extension>  
2387   </complexContent>  
2388 </complexType>
```

2389 6.5.5 Éléments de branche

2390 Tout élément d'interrogation PEUT englober au moins un sous-élément de branche. Les éléments de
2391 branche ressemblent aux éléments d'interrogation concaténée; en fait, ils peuvent comprendre eux aussi
2392 des sous-éléments qui se traduisent par des éléments de filtre et de branche ainsi que des sous-éléments
2393 d'interrogation. Ils sont toutefois différents des éléments d'interrogation en ce sens que leur type ne
2394 constitue pas un type de descendant du Type-d'interrogation-d'objet-de-registre. Les éléments de branche
2395 ont pour but de favoriser le balayage des branches du réseau de relations défini dans le modèle
2396 d'information, et d'élaguer ou de filtrer les branches qui ne sont pas conformes aux prédicats établis dans
2397 l'élément de branche correspondant.

2398 Le schéma XML de [RR-QUERY-XSD] spécifie plusieurs éléments de branche contenus dans une
2399 définition d'élément d'interrogation, comme il peut n'en spécifier aucun. Pour ce faire, il a recours à la
2400 forme générale suivante :

- 2401 • On définit un élément de branche pour chaque attribut de la classe du domaine d'interrogation qui
2402 répond à l'ensemble des critères suivants :
 - 2403 • Le domaine d'attribut ne constitue pas un type de primitive (par exemple, une chaîne, un nombre à
2404 point flottant, l'élément date-et-heure, l'élément entier, etc.);
 - 2405 • La classe du domaine d'attribut renferme des attributs de référence qui l'associent à une tierce
2406 classe au moyen de la référence;
2407 L'attribut pour lequel la branche est déterminée est désigné sous l'appellation d'attribut du domaine
2408 de branche. La classe du domaine propre à cet attribut se traduit par la classe du domaine de
2409 branche qui s'applique à l'élément de branche.
- 2410 • L'élément de branche a comme nom branche <nom de l'attribut du domaine de branche>;
- 2411 • La cardinalité de l'élément de branche s'apparie à celle de l'attribut du domaine de branche qui se
2412 rattache à la classe du domaine d'interrogation.

2413 L'exemple ci-dessous démontre la façon dont le schéma XML du [RR-QUERY-XSD] se sert de la forme
2414 générale ci-dessus pour établir les branches du Type-d'interrogation-d'objet-de-registre propre à la classe
2415 Objet-de-registre définie dans le [eBRIM].

2416

```
2417 <complexType name="RegistryObjectQueryType">  
2418   <complexContent>  
2419     <extension base="tns:FilterQueryType">  
2420       <sequence>  
2421         <element maxOccurs="unbounded" minOccurs="0"  
2422           name="SlotBranch" type="tns:SlotBranchType"/>  
2423         <element maxOccurs="1" minOccurs="0" name="NameBranch"  
2424           type="tns:InternationalStringBranchType"/>  
2425         <element maxOccurs="1" minOccurs="0" name="DescriptionBranch"  
2426           type="tns:InternationalStringBranchType"/>  
2427         ... Relevant Filters, queries go here...  
2428       </sequence>  
2429     </extension>  
2430   </complexContent>  
2431 </complexType>
```

2432

2433 6.6 Exemples d'interrogation

2434 La section 6.6 présente des exemples de syntaxe de requête SQL et de filtre d'interrogation particuliers à
2435 certains cas courants d'utilisation d'interrogation. Chaque exemple révèle la syntaxe de requête SQL de
2436 l'interrogation en question, ce qui est suivi d'une ligne vierge et, ensuite, de la syntaxe de filtre
2437 d'interrogation correspondante.

2438 6.6.1 Interrogations de nom et de description

2439 Les interrogations ci-dessous s'apparient à l'ensemble des instances d'Objet-de-registre dont le nom
2440 englobe le mot « Acme » et dont la description contient le mot « bicyclette ».

```

2441
2442 SELECT ro.* from RegistryObject ro, Name nm, Description d WHERE
2443 nm.value LIKE '%Acme%' AND
2444     d.value LIKE '%bicycle%' AND
2445     (ro.id = nm.parent AND ro.id = d.parent);
2446
2447 <RegistryObjectQuery>
2448   <NameBranch>
2449     <LocalizedStringFilter comparator="Like" domainAttribute="value"
2450       value="%Acme%" xsi:type="StringFilterType"/>
2451   </NameBranch>
2452   <DescriptionBranch>
2453     <LocalizedStringFilter comparator="Like" domainAttribute="value"
2454       value="%bicycle%" xsi:type="StringFilterType"/>
2455   </DescriptionBranch>
2456 </RegistryObjectQuery>
2457
2458

```

2459 6.6.2 Interrogations de Classification

2460 La section 6.6.2 décrit diverses interrogations qui se rattachent aux Classifications.

2461 6.6.2.1 Extraction de Schème-de-classification

2462 L'interrogation ci-dessous est caractérisée par l'extraction de la collection de tous les Schèmes-de-
 2463 classification. Il est important de mentionner que si on le souhaite, cette interrogation peut également
 2464 spécifier des interrogations, des branches et des filtres additionnels à titre de critères de recherche.

```

2465
2466 SELECT scheme.* FROM ClassificationScheme scheme;
2467
2468 <ClassificationSchemeQuery/>
2469

```

2469

2470 6.6.2.2 Extraction des enfants d'un Nœud-de-classification spécifié

2471 L'interrogation ci-dessous met en évidence l'extraction des enfants d'un Nœud-de-classification en
 2472 fonction de l'attribut identifiant du Nœud-de-classification parent.

```

2473
2474 SELECT cn.* FROM ClassificationNode cn WHERE parent = ${PARENT_ID};
2475
2476 <ClassificationNodeQuery>
2477   <PrimaryFilter comparator="Like" domainAttribute="parent"
2478     value="${PARENT_ID}" xsi:type="StringFilterType"/>
2479 </ClassificationNodeQuery>
2480

```

2480

2481 6.6.2.3 Extraction des objets classifiés par un Nœud-de-classification

2482 L'interrogation ci-dessous démontre l'extraction de la collection d'Objets-extrinsèques qui sont classifiés
 2483 en fonction de l'Industrie automobile et de la Géographie du Japon. Il est important de signaler que cette
 2484 interrogation ne s'apparie pas aux Objets-extrinsèques classifiés par les Nœuds-de-classification
 2485 descendants de l'Industrie automobile et de la Géographie du Japon. Cette mesure nécessiterait le
 2486 recours à une interrogation un peu plus complexe.

```

2487
2488 SELECT eo.* FROM ExtrinsicObject eo WHERE
2489     id IN (SELECT classifiedObject FROM Classification
2490           WHERE

```

2490

```

2491         classificationNode IN (SELECT id FROM ClassificationNode
2492                                WHERE path = '\${GEOGRAPHY_SCHEME_ID}/Asia/Japan')
2493     AND
2494     id IN (SELECT classifiedObject FROM Classification
2495            WHERE
2496             classificationNode IN (SELECT id FROM ClassificationNode
2497                                    WHERE path = '\${INDUSTRY_SCHEME_ID}/Automotive'))
2498
2499 <ExtrinsicObjectQuery>
2500 <ClassificationQuery>
2501   <ClassificationNodeQuery>
2502     <PrimaryFilter comparator="EQ" domainAttribute="path"
2503                   value="\${GEOGRAPHY_SCHEME_ID}/Asia/Japan"
2504                   xsi:type="StringFilterType"/>
2505   </ClassificationNodeQuery>
2506 </ClassificationQuery>
2507 <ClassificationQuery>
2508   <ClassificationNodeQuery>
2509     <PrimaryFilter comparator="EQ" domainAttribute="path"
2510                   value="\${INDUSTRY_SCHEME_ID}/Automotive"
2511                   xsi:type="StringFilterType"/>
2512   </ClassificationNodeQuery>
2513 </ClassificationQuery>
2514 </ExtrinsicObjectQuery>

```

2515

2516 **6.6.2.4 Extraction des Classifications qui classifient un objet**

2517 L'interrogation ci-dessous illustre l'extraction de la collection de Classifications qui classifient un objet dont
 2518 l'identifiant s'apparie à \${ID}.

2519

```

2520 SELECT c.* FROM Classification c
2521        WHERE c.classifiedObject = ${ID};
2522
2523 <ClassificationQuery>
2524   <PrimaryFilter comparator="EQ" domainAttribute="classifiedObject"
2525                 value="${ID}" xsi:type="StringFilterType"/>
2526 </ClassificationQuery>

```

2527

2528 **6.6.3 Interrogations d'Association**

2529 La section 6.6.3 décrit les diverses interrogations liées aux Associations.

2530 **6.6.3.1 Extraction de l'ensemble des Associations dont la source constitue un 2531 objet spécifié**

2532 L'interrogation ci-dessous met en évidence l'extraction de la collection d'Associations qui disposent d'un
 2533 objet dont l'identifiant s'apparie à \${SOURCE_ID} et qui ont cet objet comme source.

2534

```

2535 SELECT a.* FROM Association a WHERE sourceObject = ${SOURCE_ID}
2536
2537 <AssociationQuery>
2538   <PrimaryFilter comparator="EQ" domainAttribute="sourceObject"
2539                 value="${SOURCE_ID}" xsi:type="StringFilterType"/>
2540 </AssociationQuery>

```

2541

2542 **6.6.3.2 Extraction de l'ensemble des Associations dont la cible représente un** 2543 **objet spécifié**

2544 L'interrogation ci-dessous illustre l'extraction de la collection d'Associations qui disposent d'un objet dont
2545 l'identifiant s'apparie à \${TARGET_ID} et qui ont cet objet comme cible.

2546

```
2547 SELECT a.* FROM Association a WHERE targetObject = ${TARGET_ID}
```

2548

```
2549 <AssociationQuery>
```

```
2550 <PrimaryFilter comparator="EQ" domainAttribute="targetObject"
```

```
2551 value="${TARGET_ID}" xsi:type="StringFilterType"/>
```

```
2552 </AssociationQuery>
```

2553

2554 **6.6.3.3 Extraction des objets associés fondés sur le type d'Association**

2555 Il est important de sélectionner les Associations dont la valeur de l'attribut type-d'association s'apparie à
2556 celle spécifiée par \${ASSOC_TYPE_ID}. Cette dernière valeur DOIT citer un Nœud-de-classification qui
2557 désigne un descendant du Schème-de-classification de Type-d'association canonique.

2558

```
2559 SELECT a.* FROM Association a WHERE
```

```
2560 associationType = ${ASSOC_TYPE_ID}
```

2561

```
2562 <AssociationQuery>
```

```
2563 <PrimaryFilter comparator="EQ" domainAttribute="associationType"
```

```
2564 value="${ASSOC_TYPE_ID}" xsi:type="StringFilterType"/>
```

```
2565 </AssociationQuery>
```

2566

2567

2568 **6.6.3.4 Interrogation complexe d'Association**

2569 Les diverses formes d'interrogation d'Association peuvent être regroupées en des prédicats complexes.
2570 L'interrogation ci-dessous est axée sur la sélection d'Associations qui s'apparient à un objet-source, à un
2571 objet-cible et à un type-d'association précis.

2572

```
2573 SELECT a.* FROM Association a WHERE  
2574 sourceObject = ${SOURCE_ID} AND  
2575 targetObject = ${TARGET_ID} AND  
2576 associationType = ${ASSOC_TYPE_ID};
```

2577

```
2578 <AssociationQuery>
```

```
2579 <PrimaryFilter logicalOperator="AND" xsi:type="CompoundFilterType">
```

```
2580 <LeftFilter comparator="EQ" domainAttribute="sourceObject"
```

```
2581 xsi:type="StringFilterType" value="${SOURCE_ID}"/>
```

```
2582 <RightFilter logicalOperator="AND" xsi:type="CompoundFilterType">
```

```
2583 <LeftFilter comparator="EQ" domainAttribute="targetObject"
```

```
2584 xsi:type="StringFilterType" value="${TARGET_ID}"/>
```

```
2585 <RightFilter comparator="EQ" domainAttribute="associationType"
```

```
2586 xsi:type="StringFilterType" value="${ASSOC_TYPE_ID}"/>
```

```
2587 </RightFilter>
```

```
2588 </PrimaryFilter>
```

```
2589 </AssociationQuery>
```

2590

2591 **6.6.4 Interrogations de Paquet-de-registre**

2592 L'interrogation ci-dessous met en évidence l'extraction de l'ensemble des Paquets-de-registre qui ont pour

2593 membre l'Objet-de-registre spécifié par \${REGISTRY_OBJECT_ID}.

2594

```
2595 SELECT p.* FROM Package p, Association a WHERE
2596     a.sourceObject = p.id AND
2597     a.targetObject = ${REGISTRY_OBJECT_ID} AND
2598     a.associationType = ${HAS_MEMBER_ASSOC_TYPE_NODE_ID};
```

2599

```
2600 <RegistryPackageQuery>
2601   <SourceAssociationQuery>
2602     <PrimaryFilter logicalOperator="AND" xsi:type="CompoundFilterType">
2603       <LeftFilter comparator="EQ" domainAttribute="targetObject"
2604         value="${REGISTRY_OBJECT_ID}"
2605         xsi:type="StringFilterType"/>
2606       <RightFilter comparator="EQ" domainAttribute="associationType"
2607         value="${HAS_MEMBER_ASSOC_TYPE_NODE_ID}"
2608         xsi:type="StringFilterType"/>
2609     </PrimaryFilter>
2610   </SourceAssociationQuery>
2611 </RegistryPackageQuery>
```

2612

2613 Il est important de mentionner que la valeur \${HAS_MEMBER_ASSOC_TYPE_NODE_ID} représente un
2614 nom général qui s'applique à la valeur de l'attribut identifiant du Nœud-de-classification canonique de
2615 Type-d'association A-pour-membre.

2616 6.6.5 Interrogations de Lien-externe

2617 L'interrogation ci-dessous est axée sur l'extraction de l'ensemble des Liens-externes qui constituent des
2618 Liens-externes pour les Objets-de-registre évoqués par \${REGISTRY_OBJECT_ID}.

2619

```
2620 SELECT el.* From ExternalLink el, Association a WHERE
2621     a.sourceObject = el.id AND
2622     a.targetObject = ${REGISTRY_OBJECT_ID} AND
2623     a.associationType = ${EXTERNALLY_LINKS_ASSOC_TYPE_NODE_ID};
```

2624

```
2625 <ExternalLinkQuery>
2626   <SourceAssociationQuery>
2627     <PrimaryFilter logicalOperator="AND" xsi:type="CompoundFilterType">
2628       <LeftFilter comparator="EQ" domainAttribute="targetObject"
2629         value="${REGISTRY_OBJECT_ID}"
2630         xsi:type="StringFilterType"/>
2631       <RightFilter comparator="EQ" domainAttribute="associationType"
2632         value="${EXTERNALLY_LINKS_ASSOC_TYPE_NODE_ID}"
2633         xsi:type="StringFilterType"/>
2634     </PrimaryFilter>
2635   </SourceAssociationQuery>
2636 </ExternalLinkQuery>
```

2637

2638 Il est important de signaler que la valeur \${EXTERNALLY_LINKS_ASSOC_TYPE_NODE_ID} se traduit
2639 par un nom général propre à la valeur de l'attribut identifiant du Nœud-de-classification canonique de
2640 Type-d'association de Lien-externe.

2641 L'interrogation ci-dessous est caractérisée par l'extraction de l'ensemble des Objets-extrinsèques qui se
2642 rattachent à un Lien-externe déterminé par \${EXTERNAL_LINK_ID}.

2643

```
2644 SELECT eo.* From ExtrinsicObject eo, Association a WHERE
2645     a.sourceObject = ${EXTERNAL_LINK_ID} AND
2646     a.targetObject = eo.id AND
2647     a.associationType = ${EXTERNALLY_LINKS_ASSOC_TYPE_NODE_ID};
```

2648

```
2649 <ExtrinsicObjectQuery>
```

```
2650 <TargetAssociationQuery>
2651   <PrimaryFilter logicalOperator="AND" xsi:type="CompoundFilterType">
2652     <LeftFilter comparator="EQ" domainAttribute="sourceObject"
2653       value="{EXTERNAL_LINK_ID}"
2654       xsi:type="StringFilterType"/>
2655     <RightFilter comparator="EQ" domainAttribute="associationType"
2656       value="{EXTERNALLY_LINKS_ASSOC_TYPE_NODE_ID}"
2657       xsi:type="StringFilterType"/>
2658   </PrimaryFilter>
2659 </TargetAssociationQuery>
2660 </ExtrinsicObjectQuery>
```

2661

2662 6.6.6 Interrogations d'Événement-vérifiable

2663 L'interrogation ci-dessous illustre l'extraction de l'ensemble des Événements-vérifiables particuliers à
2664 l'Objet-de-registre établi par \${REGISTRY_OBJECT_ID}.

2665

```
2666 SELECT ae.* FROM AuditableEvent ae, AffectedObject ao WHERE
2667     ao.eventId = ae.id AND
2668     ao.id = ${REGISTRY_OBJECT_ID}
2669
2670 <AuditableEventQuery>
2671   <AffectedObjectQuery>
2672     <PrimaryFilter comparator="EQ" domainAttribute="id"
2673       value="{REGISTRY_OBJECT_ID}" xsi:type="StringFilterType"/>
2674   </AffectedObjectQuery>
2675 </AuditableEventQuery>
```

2676

2677 **7 Protocoles de notification d'événement**

2678 Le chapitre 7 définit la caractéristique de notification d'événement du registre ebXML d'OASIS.

2679 Cette caractéristique fait en sorte que les registres ebXML d'OASIS peuvent informer leurs utilisateurs ou
2680 ceux d'autres registres de la tenue d'événements susceptibles de les intéresser. Elle permet aux
2681 utilisateurs d'être au courant de la tenue d'événements de registre sans pour autant être obligés
2682 d'interroger le registre à des intervalles réguliers. C'est par cette caractéristique qu'un registre peut
2683 reproduire des changements internes dans d'autres registres dont le contenu pourrait en être touché.

2684 Les registres ebXML gèrent les notifications fondées sur le contenu et où les parties intéressées
2685 manifestent leur intérêt par une interrogation. Ces notifications sont différentes de celles reposant sur des
2686 sujets (on les désigne parfois sous l'appellation de notifications axées sur des thèmes), où les
2687 renseignements sont classés par catégories en fonction du sujet, et où les parties intéressées expriment
2688 leur intérêt à l'égard des sujets prédéfinis.

2689 **7.1 Cas d'utilisation**

2690 La section 7.1 présente des cas d'utilisation qui illustrent différentes méthodes dont les registres ebXML
2691 se servent pour informer leurs utilisateurs ou ceux d'autres registres.

2692 **7.1.1 Nouveau Profil de protocole de collaboration**

2693 Un utilisateur désire savoir à quel moment le Profil de protocole de collaboration (CPP) [ebCPP] d'un
2694 partenaire sera mis à jour ou bien sera supplanté par un autre CPP. Dans une telle situation, l'utilisateur
2695 pourrait vouloir établir une Entente de protocole de collaboration (CPA) [ebCPP] à partir du nouveau CPP.

2696 **7.1.2 Disponibilité d'un nouveau service**

2697 Un utilisateur souhaite savoir à partir de quand un nouveau service de plomberie sera offert dans sa
2698 propre ville, et veut en être avisé aux dix jours. Si un nouveau service est bel et bien offert, l'utilisateur
2699 pourrait tenter d'en savoir davantage sur ce service et le comparer ensuite avec celui déjà offert.

2700 **7.1.3 Surveillance du téléchargement d'un contenu**

2701 Un utilisateur veut savoir à quel moment son CPP [ebCPP] sera téléchargé afin d'évaluer de façon
2702 permanente la réussite de sa récente campagne publicitaire. Il pourrait aussi souhaiter analyser l'identité
2703 des parties intéressées.

2704 **7.1.4 Surveillance des changements de prix**

2705 Un utilisateur désire savoir à quel moment le prix d'un produit qu'il veut acheter descendra sous un seuil
2706 donné. S'il en fait l'achat, il souhaitera également savoir à quel moment ce produit lui sera expédié.

2707 **7.1.5 Assurance de la concordance entre les duplications et l'objet source**

2708
2709 Afin d'améliorer la disponibilité et le rendement liés à l'accès à des objets de registre, un registre local
2710 PEUT dupliquer certains objets hébergés dans un autre registre. Il souhaiterait alors être informé du
2711 moment où l'objet source propre à une duplication serait mis à jour; il pourrait ensuite synchroniser la
2712 duplication avec l'état le plus récent de l'objet source.

2713 **7.2 Événements de registre**

2714 Les activités effectuées dans un registre quelconque entraînent des événements importants. De façon
2715 générale, les événements de registre sont créés au moment où un registre traite des demandes
2716 d'application. De plus, certains de ces événements peuvent survenir quand un opérateur de registre
2717 procède à des mesures administratives. Le [ebRIM] définit la classe Événement-vérifiable, dont les

2718 instances représentent des événements de registre. Au moment où un événement de ce genre se produit,
2719 le registre conçoit une instance d'Événement-vérifiable.

2720 **7.3 Abonnement à des événements**

2721 Un utilisateur PEUT créer un abonnement à un registre s'il souhaite recevoir une notification par rapport à
2722 un type particulier d'événement. Pour ce faire, il doit inscrire une instance d'Abonnement dans un registre
2723 au moyen de la Demande-d'inscription-d'objet. Si un Abonnement est inscrit dans un registre qui ne peut
2724 pas accueillir de notification d'événement, ce registre DOIT renvoyer une Exception-de-capacité-non-
2725 gérée.

2726 La liste ci-dessous présente un exemple d'Abonnement axé sur le recours à une requête SQL prédéfinie
2727 à titre de sélecteur. L'utilisateur recevra une notification par courriel si un Service est créé et si celui-ci est
2728 classé dans la catégorie des services de « plomberie » et est situé dans « Une petite ville ».

2729 La requête SQL qui se trouve dans le sélecteur et est rédigée en langage courant se lit comme suit :

2730 *Trouver tous les Services qui sont créés ET classifiés par un Nœud-de-classification dont le chemin se*
2731 *termine par la chaîne « Plomberie », ET qui sont classifiés par un Nœud-de-classification dont le code*
2732 *renferme la chaîne « Une petite ville ».*

2733

```
2734 <rim:Subscription id="{SUBSCRIPTION_ID}" selector="{QUERY_ID}">  
2735   <!--  
2736     The selector is a reference to a query object that has the  
2737     following query defined  
2738     SELECT * FROM Service s, AuditableEvent e, AffectedObject ao,  
2739     Classification c1, Classification c2  
2740     ClassificationNode cn1, ClassificationNode cn2 WHERE  
2741     e.eventType = 'Created' AND ao.id = s.id AND ao.parent=e.id AND  
2742     c1.classifiedObject = s.id AND c1.classificationNode = cn1.id AND  
2743     cn1.path LIKE '%Plumbing' AND  
2744     c2.classifiedObject = s.id AND c2.classificationNode = cn2.id AND  
2745     cn2.path LIKE '%A Little Town%'  
2746   -->  
2747   <!-- Next endPoint is an email address -->  
2748   <rim:NotifyAction notificationOption="urn:oasis:names:tc:ebxml-  
2749   regrep:NotificationOptionType:Objects"  
2750   endPoint="mailto:farrukh.najmi@sun.com"/>  
2751   <!-- Next endPoint is a service via reference to its ServiceBinding  
2752   object -->  
2753   <rim:NotifyAction notificationOption="urn:oasis:names:tc:ebxml-  
2754   regrep:NotificationOptionType:ObjectRefs"  
2755   endPoint="urn:freebxml:registry:demoDB:serviceBinding:EpidemicAlertListen  
2756   erServiceBinding"/>  
2757 </rim:Subscription>
```

2758

2759 **7.3.1 Sélection d'événement**

2760 Si l'utilisateur ne souhaite être informé que d'événements particuliers susceptibles de l'intéresser, il DOIT
2761 établir une référence vers un objet stocké d'Interrogation-ad-hoc par le biais de l'attribut sélecteur de
2762 l'instance d'Abonnement. L'interrogation détermine si un événement entre en ligne de compte ou non pour
2763 cet Abonnement. Prière de consulter le chapitre 6 pour obtenir des précisions au sujet de la syntaxe de
2764 requête.

2765 **7.3.2 Action de notification**

2766 Au moment de créer un Abonnement, un utilisateur PEUT également en préciser les Actions qui
2767 déterminent ce que le registre doit faire s'il se produit un événement s'appariant à cet Abonnement
2768 (événement d'abonnement).

2769 Un utilisateur PEUT omettre de spécifier une Action dans un Abonnement s'il ne désire aucune
2770 notification de la part du registre. Il PEUT, à des intervalles réguliers, interroger le registre et aller

2771 chercher les notifications en cours.

2772 Le [ebRIM] définit deux méthodes normalisées d'utilisation d'une Action-de-notifier :

- 2773 • Action-de-notifier par courriel : Cette caractéristique permet de transmettre par courriel des
2774 notifications d'événement à un utilisateur ou à un point branché dans le cas d'un composant
2775 logiciel ou d'un agent;
- 2776 • Action-de-notifier par un service : Cette caractéristique permet de transmettre des notifications
2777 d'événement au moyen d'une interface programmatique en appelant un service Web d'écoute
2778 déterminé.

2779 Si le registre peut accueillir une notification d'événement, il DOIT, quelque temps après le traitement
2780 fructueux de chaque interrogation, vérifier l'ensemble des Abonnements enregistrés et actifs, et établir si
2781 des Abonnements quelconques s'apparient à l'événement en question. Si un appariement est détecté, le
2782 registre exécute les actions de notification nécessaires pour l'Abonnement concerné. Tout registre PEUT,
2783 à des intervalles réguliers, procéder à de telles vérifications et aux actions de notification connexes selon
2784 un mode différé qui repose sur des règles propres au registre.

2785 **7.3.3 Autorisation d'Abonnement**

2786 Un opérateur de registre ou un détenteur de contenu PEUVENT se servir de règles contextualisées de
2787 contrôle d'accès pour déterminer les utilisateurs qui ont l'autorisation de créer un Abonnement et les
2788 événements qui s'appliquent ici. Si un utilisateur non autorisé inscrit un Abonnement dans un registre,
2789 celui-ci DOIT renvoyer une Exception-d'autorisation. Si une règle de contrôle d'accès qui régit les
2790 Abonnements devient plus restrictive, il incombe aux implantations de registre de respecter ou non un
2791 Abonnement déjà en place. Les implantations peuvent aussi respecter ou non des Abonnements créés à
2792 partir de la règle antérieure.

2793 **7.3.4 Quota d'Abonnements**

2794 Tout registre PEUT utiliser des règles qui lui sont propres pour établir une limite maximale du nombre
2795 d'Abonnements qu'un utilisateur peut créer. Si un utilisateur autorisé inscrit un nombre plus élevé
2796 d'Abonnements que celui que le lui permet le quota propre au registre, celui-ci DOIT renvoyer une
2797 Exception-liée-à-un-excès-de-quota.

2798 **7.3.5 Expiration d'un Abonnement**

2799 Chaque Abonnement définit un attribut heure-de-début et un attribut heure-de-fin qui en déterminent la
2800 période d'activation. Au-delà de cette période, le registre PEUT renfermer un Abonnement expiré. Un
2801 registre PEUT en tout temps supprimer un Abonnement de ce genre. Dans un tel cas, on DOIT utiliser
2802 l'identité d'un Opérateur-de-registre pour toute interrogation afin d'avoir l'autorisation nécessaire pour
2803 supprimer l'Abonnement d'un utilisateur.

2804 Les registres NE DOIVENT PAS tenir compte des Abonnements expirés quand ils émettent des
2805 notifications d'événement à leurs Abonnements. On PEUT renouveler un Abonnement expiré en en
2806 inscrivant un nouveau.

2807 **7.3.6 Rejet d'un Abonnement**

2808 Tout registre PEUT rejeter un Abonnement si sa gestion s'avère trop coûteuse. Par exemple, un
2809 Abonnement qui souhaite être informé de tout changement apporté à n'importe quel objet pourrait
2810 entraîner des coûts trop élevés pour la plupart des registres. Si un utilisateur autorisé inscrit un
2811 Abonnement dont le traitement est trop coûteux pour un registre, celui-ci DOIT renvoyer une Exception-
2812 liée-à-un-abonnement-trop-coûteux.

2813 **7.4 Désabonnement à des événements**

2814 Un utilisateur PEUT mettre fin à un Abonnement à un registre s'il ne souhaite plus être informé des
2815 événements qui se rattachent à cet Abonnement. Pour ce faire, il doit supprimer l'objet d'Abonnement
2816 correspondant au moyen de la Demande-de-suppression-d'objet du registre.

2817 La règle relative à la suppression d'un objet d'Abonnement est la même que celle qui s'applique à la
2818 suppression de n'importe quel autre objet.

2819 **7.5 Notification d'événement**

2820 Les registres mettent en œuvre les *Actions* propres à un Abonnement afin de transmettre aux abonnés
2821 des renseignements sur les divers événements. Toutefois, peu importe l'action de transférer préconisée,
2822 les registres DOIVENT communiquer les événements liés à tout Abonnement. Ceux-ci sont transmis dans
2823 une instance de Notification, comme l'expose le [ebRIM]. Si on utilise une Action-de-notifier par un
2824 service, la Notification sera transmise à un service de gestionnaire conforme à l'interface d'Application-de-
2825 registre. Si on a plutôt recours à une Action-de-notifier par courriel, la Notification sera envoyée à une
2826 adresse électronique.

2827 La liste ci-dessous illustre un exemple de Notification qui s'apparie à l'exemple d'Abonnement figurant
2828 dans la section 7.3.

2829

```
2830 <rim:Notification subscription="{SUBSCRIPTION_ID}">  
2831 <rim:RegistryObjectList>  
2832 <rim:Service id="f3373a7b-4958-4e55-8820-d03a191fb76a">  
2833 <rim:Name>  
2834 <rim:LocalizedString value="A Little Town Plumbing"/>  
2835 </rim:Name>  
2836 <rim:Classification id="a3373a7b-4958-4e55-8820-d03a191fb76a"  
2837 classifiedObject="f3373a7b-4958-4e55-8820-d03a191fb76a"/>  
2838 <rim:Classification id="b3373a7b-4958-4e55-8820-d03a191fb76a"  
2839 classifiedObject="f3373a7b-4958-4e55-8820-d03a191fb76a"/>  
2840 </rim:Service>  
2841 </rim:RegistryObjectList>  
2842 </rim:Notification>
```

2843

2844 Une Notification PEUT renfermer des Objets-de-registre réels ou des Références-d'objet réelles qui s'y
2845 rattachent. Ceux-ci figurent dans l'élément <rim:Liste-d'objets-de-registre>. Une application PEUT
2846 déterminer si elle préfère recevoir des Objets-de-registre ou bien des Références-d'objet qui s'y
2847 rattachent. Pour ce faire, elle doit se servir de l'attribut option-de-notification de l'Action dans
2848 l'Abonnement même. Le registre PEUT annuler cet attribut à partir de ses propres règles opérationnelles.

2849 **7.6 Extraction d'événement**

2850 Le registre offre l'option de prestation de Notifications selon le transfert dans le style envoi asynchrone
2851 (PUSH); pour ce faire, il a recours aux actions de notifier décrites ci-dessus. Une application PEUT
2852 cependant utiliser aussi le style aller chercher (PULL) afin d'extraire tout événement en cours pour ses
2853 Abonnements. On procède à l'extraction d'événement en se servant du protocole d'Interrogation-ad-hoc et
2854 en interrogeant la classe Notification. Tout registre DEVRAIT mettre en mémoire tampon les notifications
2855 non transmises pendant une certaine période pour permettre aux applications d'aller les chercher. On
2856 PEUT établir cette période en appliquant des règles propres au registre au question.

2857 **7.7 Élagage d'événement**

2858 Tout registre PEUT, à des intervalles réguliers, élaguer des Événements-vérifiables dans le but de gérer
2859 ses ressources. En fait, il s'agit de la responsabilité des registres. Il incombe également aux registres de
2860 déterminer à quel moment doit se faire l'élimination des événements non transmis. Les registres
2861 DEVRAIENT aussi effectuer un élagage en supprimant les renseignements désuets du contenu de leur
2862 piste de vérification. Ils NE DOIVENT toutefois PAS supprimer l'option originale créer un événement au
2863 début de la piste de vérification puisque cette option établit l'identité du détenteur de l'Objet-de-registre.

2864

8 Services de gestion de contenu

2865
2866
2867
2868

Le chapitre 8 décrit les services de gestion de contenu du registre ebXML. Ces services comprennent, sans toutefois s'y limiter, la validation de contenu et le catalogage de contenu. Ils contribuent à améliorer la qualité et l'intégrité des métadonnées et du contenu de registre, et à accroître la capacité pour les applications de les détecter.

2869
2870
2871
2872
2873

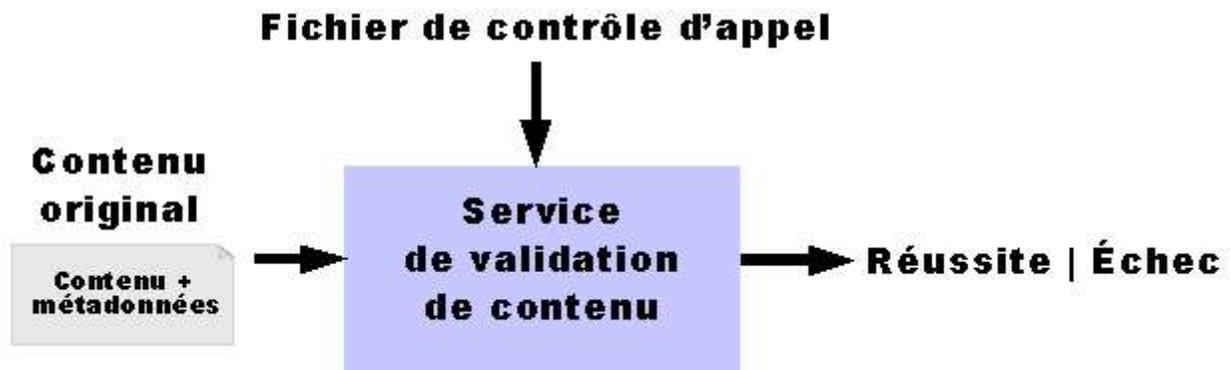
La fonction des services de gestion de contenu de registre repose sur une architecture enfichable qui permet aux applications de publier et de détecter de nouveaux services de ce genre. Ces services constituent des objets de Service conformes à une interface normative de service Web décrite dans le présent chapitre. Les applications PEUVENT configurer un service de gestion de contenu qui a été spécialement conçu pour gérer un type précis de contenu.

2874

8.1 Validation de contenu

2875
2876
2877

La caractéristique de validation de contenu permet d'appliquer des règles de validation qui se rattachent à un domaine en particulier, ce que l'on fait en se fondant sur les métadonnées et le contenu inscrits, et en respectant le contenu.



2879

Diagramme 13 : Service de validation de contenu

2880
2881
2882
2883

Tout registre a recours à au moins un service de validation de contenu pour valider automatiquement les Objets-de-registre et les items de référentiel au moment de leur inscription dans le registre. Les registres DOIVENT rejeter toute demande d'inscription dans son intégralité si elle comprend des données invalides. Si une telle situation se produit, une Exception-de-validation DOIT être renvoyée à l'application.

2884

La caractéristique de validation de contenu contribue à améliorer la qualité des données du registre.

2885

8.1.1 Validation de contenu : cas d'utilisation

2886

Les cas d'utilisation ci-dessous mettent en évidence la caractéristique de validation de contenu.

2887

8.1.1.1 Validation des profils de conformité de HL7

2888
2889
2890
2891

Les normes sur les soins de santé HL7 se servent de la validation de contenu pour appliquer des règles de concordance et effectuer des vérifications sémantiques si un élément de HL7 inscrit un profil de conformité de HL7. HL7 prévoit également avoir recours à cette caractéristique pour améliorer la qualité des autres types d'artefact de HL7.

2892

8.1.1.2 Validation des processus d'affaires

2893
2894
2895

On peut utiliser la validation de contenu pour appliquer des règles de concordance et effectuer des vérifications sémantiques si un processus d'affaires est inscrit dans le registre. Des organisations comme l'UN/CEFACT, l'OAGi et RosettaNet peuvent se servir de cette caractéristique.

2896 **8.1.1.3 Validation des documents d'affaires UBL**

2897 Le comité technique UBL peut avoir recours à la validation de contenu pour appliquer des règles de
2898 concordance et effectuer des vérifications sémantiques si un document d'affaires UBL est inscrit dans le
2899 registre.

2900 **8.2 Catalogage de contenu**

2901 La caractéristique de catalogage de contenu permet de convertir d'une manière sélective les Objets-de-
2902 registre et items de référentiel inscrits en métadonnées définies par le [ebRIM], et de le faire d'une façon
2903 propre au contenu.



2905 **Diagramme 14 : Service de catalogage de contenu**

2906 Tout registre se sert d'au moins un service de catalogage de contenu pour cataloguer automatiquement
2907 les Objets-de-registre et les items de référentiel. Un tel catalogage crée ou met à jour les métadonnées
2908 des Objets-de-registre, comme les instances d'Objet-extrinsèque ou de Classification. Les métadonnées
2909 ainsi cataloguées permettent aux applications de détecter des items de référentiel en fonction de leur
2910 contenu, et de le faire en utilisant les capacités normalisées d'interrogation du registre, opération
2911 désignée sous l'appellation de *découverte fondée sur le contenu*.

2912 Le principal avantage de la caractéristique de catalogage de contenu est qu'elle favorise la découverte
2913 fondée sur le contenu.

2914 **8.2.1 Découverte fondée sur le contenu : cas d'utilisation**

2915 Il existe de nombreux scénarios où la découverte fondée sur le contenu est nécessaire.

2916 **8.2.1.1 Détection de tous les CPP qui jouent un rôle d'« acheteur »**

2917 Une entreprise qui vend un produit en se servant du bon de commande RosettaNet PIP3A4 pourrait
2918 souhaiter détecter les CPP d'autres entreprises, dans la mesure où ces CPP jouent un rôle
2919 d'« acheteur ».

2920 **8.2.1.2 Détection de tous les schémas XML qui utilisent un espace de
2921 nommage précis**

2922 Une application quelconque pourrait désirer trouver dans un registre l'ensemble des documents de
2923 schéma XML qui se servent d'un espace de nommage XML comprenant le mot « oasis ».

2924 **8.2.1.3 Détection de toutes les descriptions WSDL ayant un rattachement de
2925 protocole SOAP**

2926 Une application de registre ebXML pourrait vouloir essayer de détecter l'ensemble des items de référentiel

2927 qui se traduisent par des descriptions WSDL possédant un rattachement défini de protocole SOAP. Il est
2928 important de mentionner que les renseignements liés au rattachement de protocole SOAP représentent le
2929 contenu du document WSDL et non pas les métadonnées.

2930 **8.3 Service abstrait de gestion de contenu**

2931 La section 8.3 décrit en termes abstraits la façon dont le registre peut accueillir des services de gestion de
2932 contenu enfichable et défini par l'utilisateur. Un service de gestion de contenu est appelé à la suite de
2933 l'inscription dans le registre d'un contenu par la méthode normalisée de la Demande-d'inscription-d'objet
2934 et de la Demande-de-mise-à-jour-d'objet. Un appel de Service a lieu pour chaque demande; une
2935 demande peut entraîner de nombreux appels, dont un pour chaque Objet-de-registre pour lequel un
2936 service de gestion de contenu est configuré dans le registre.

2937 Le registre peut exécuter de tels appels de l'une des deux façons suivantes :

2938

- 2939 • **Modèle d'appel incorporé** : Il est possible d'appeler un service de gestion de contenu en
2940 incorporant cet appel au traitement de la Demande-d'inscription-d'objet et de la Demande-de-
2941 mise-à-jour-d'objet. On doit agir ainsi avant d'inscrire le contenu de façon permanente. On
2942 désigne cette opération sous l'appellation de modèle d'appel incorporé;
- 2943 • **Modèle d'appel découplé** : On peut appeler un service de gestion de contenu en découplant cet
2944 appel du traitement de la Demande-d'inscription-d'objet et de la Demande-de-mise-à-jour-d'objet.
2945 On doit agir ainsi après avoir inscrit le contenu de façon permanente. On appelle cette opération
2946 modèle d'appel découplé.

2947

2948 **8.3.1 Modèle d'appel incorporé**

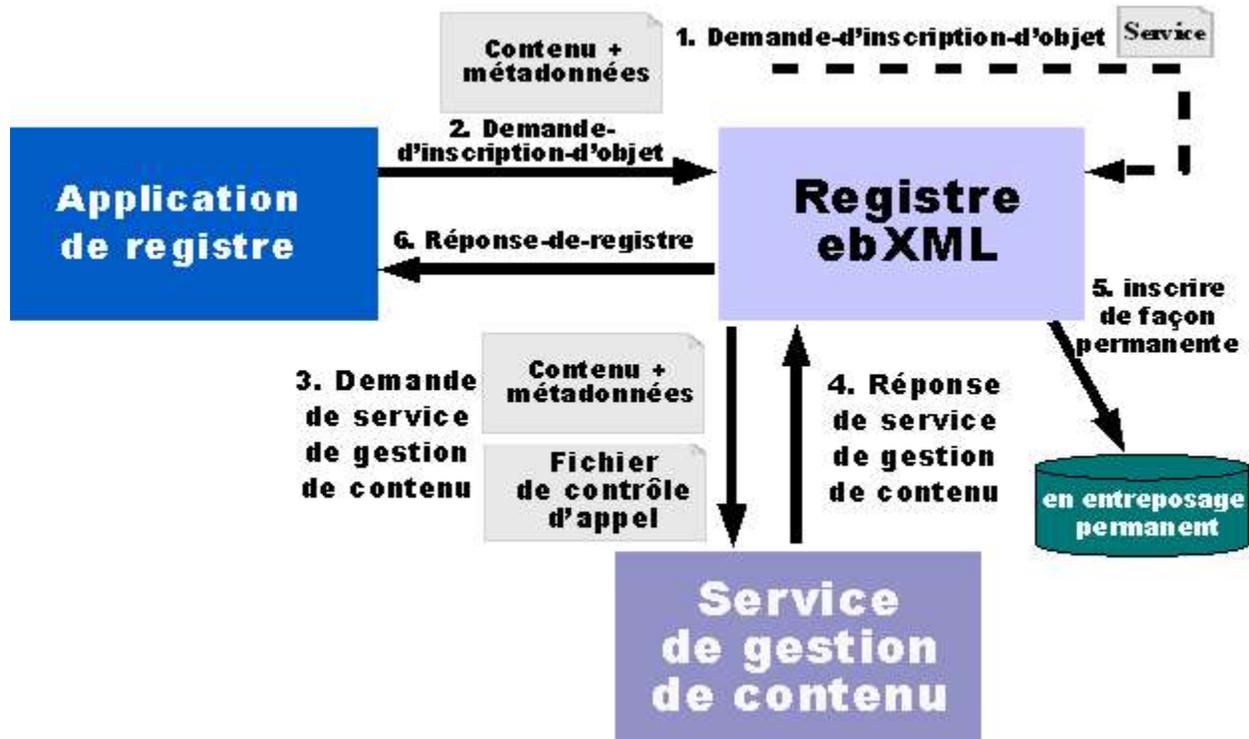
2949 Dans le cadre d'un modèle d'appel incorporé, un registre DOIT appeler un service de gestion de contenu
2950 en incorporant cet appel au traitement de la Demande-d'inscription-d'objet et de la Demande-de-mise-à-
2951 jour-d'objet, et il DOIT agir ainsi avant d'inscrire de façon permanente le contenu de ces demandes.
2952 L'ensemble des métadonnées et du contenu qui ressortent de la Demande-d'inscription-d'objet et de la
2953 Demande-de-mise-à-jour-d'objet originales, ou bien de l'appel d'un service de gestion de contenu
2954 DOIVENT être inscrits de façon permanente sous forme de transaction atomique.

2955 Le Diagramme 15 présente un service abstrait de gestion de contenu et illustre comment il est utilisé par
2956 un registre ebXML au moyen d'un modèle d'appel incorporé. Les étapes sont les suivantes :

2957

- 2958 1. Une application inscrit un service de gestion de contenu S1 dans un registre ebXML. Elle
2959 provient généralement d'une organisation responsable de la définition d'un type précis de
2960 contenu. Par exemple, elle peut provenir de RosettaNet.org et inscrire un service de
2961 validation de contenu en vue de la validation des PIP RosettaNet. L'application se sert de
2962 l'interface normalisée de la Demande-d'inscription-d'objet et de la Demande-de-mise-à-
2963 jour-d'objet pour inscrire le Service. Il s'agit là d'une étape unique qui vise la
2964 configuration dans le registre du service de gestion de contenu;
- 2965 2. Une fois le service de gestion de contenu inscrit, une application potentiellement
2966 différente peut inscrire un contenu dans le registre dont le type d'objet est le même que
2967 celui pour lequel le service de gestion de contenu a été inscrit. L'application a recours à
2968 l'interface normalisée de la Demande-d'inscription-d'objet et de la Demande-de-mise-à-
2969 jour-d'objet pour inscrire le contenu;
- 2970 3. Le registre établit qu'un service de gestion de contenu S1 a été configuré pour le type
2971 d'objet propre au contenu inscrit. Il appelle le service S1 au moyen d'une Demande-de-
2972 service-de-gestion-de-contenu et lui transmet le contenu;

- 2973 4. Le service de gestion de contenu S1 traite le contenu et renvoie une Réponse-de-service-
 2974 de-gestion-de-contenu;
 2975 5. Le registre inscrit ensuite de façon permanente le contenu dans le registre si aucune erreur
 2976 n'a été détectée;
 2977 6. Le registre renvoie une Réponse-de-registre à l'application par rapport à la Demande-
 2978 d'inscription-d'objet et à la Demande-de-mise-à-jour-d'objet (étape n° 2).
 2979
 2980
 2981



2983 **Diagramme 15 : Service de gestion de contenu : modèle d'appel incorporé**

2984 **8.3.2 Modèle d'appel découplé**

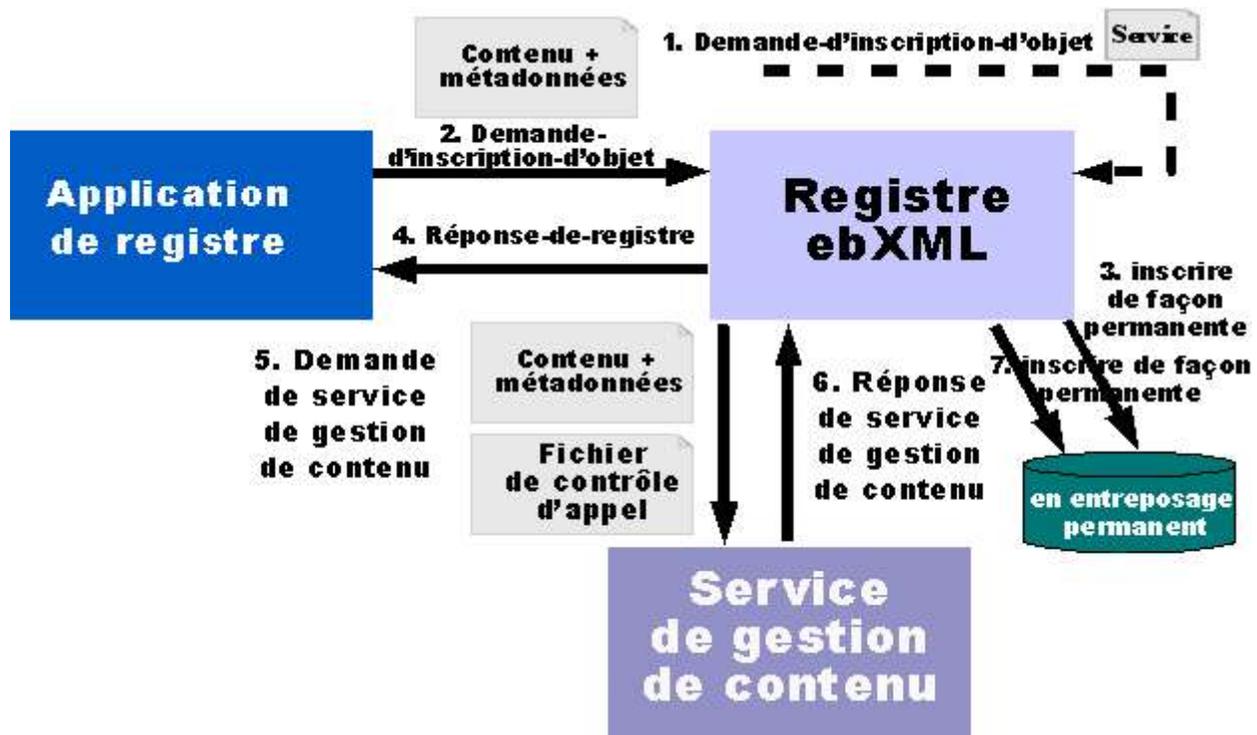
2985 Dans le cadre d'un modèle d'appel découplé, un registre DOIT appeler un service de gestion de contenu
 2986 en découplant ou en distinguant cet appel du traitement de la Demande-d'inscription-d'objet et de la
 2987 Demande-de-mise-à-jour-d'objet. La moindre erreur détectée au cours de l'appel du service de gestion de
 2988 contenu NE DOIT avoir aucun impact sur le traitement de la Demande-d'inscription-d'objet et de la
 2989 Demande-de-mise-à-jour-d'objet originales.

2990 L'ensemble des métadonnées et du contenu de la Demande-d'inscription-d'objet et de la Demande-de-
 2991 mise-à-jour-d'objet originales DOIVENT être inscrits de façon permanente sous forme de transaction
 2992 atomique. Celle-ci doit être découplée des métadonnées et du contenu susceptibles d'être produits par
 2993 l'appel du service de gestion de contenu.

2994 Le Diagramme 16 présente un service abstrait de gestion de contenu et illustre comment il est utilisé par
 2995 un registre ebXML au moyen d'un modèle d'appel découplé. Les étapes sont les suivantes :

- 2996
 2997 1. La première étape est la même que celle du modèle d'appel incorporé (un service de
 2998 gestion de contenu est inscrit);

- 2999 2. La deuxième étape est la même que celle du modèle d'appel incorporé (une application
3000 inscrit un contenu par le biais de la Demande-d'inscription-d'objet et de la Demande-de-
3001 mise-à-jour-d'objet);
- 3002 3. Le registre traite la Demande-d'inscription-d'objet et la Demande-de-mise-à-jour-d'objet,
3003 et les inscrit ensuite de façon permanente de manière qu'elles soient en entreposage
3004 permanent;
- 3005 4. Le registre renvoie une Réponse-de-registre à l'application par rapport à la Demande-
3006 d'inscription-d'objet et à la Demande-de-mise-à-jour-d'objet (étape n° 2);
- 3007 5. Le registre établit qu'un service de gestion de contenu S1 a été configuré pour le type
3008 d'objet propre au contenu inscrit. Il appelle le service S1 au moyen d'une Demande-de-
3009 service-de-gestion-de-contenu et lui transmet le contenu;
- 3010 6. Le service de gestion de contenu S1 traite le contenu et renvoie une Réponse-de-service-
3011 de-gestion-de-contenu;
- 3012 7. Si la Réponse-de-service-de-gestion-de-contenu englobe un contenu produit ou modifié
3013 quelconque, elle est inscrite de façon permanente à titre de transaction distincte de
3014 manière qu'elle soit en entreposage permanent. Si une erreur est détectée pendant l'appel
3015 découplé d'un service de gestion de contenu, elle est enregistrée par le registre d'une
3016 manière propre à celui-ci. Elle NE DOIT PAS être transmise à l'application.
3017



3019 **Diagramme 16 : Service de gestion de contenu : modèle d'appel découplé**

3020 **8.4 Protocole de service de gestion de contenu**

3021 La section 8.4 décrit le protocole de service abstrait de gestion de contenu. Il s'agit du protocole de base
3022 qui s'applique aux autres protocoles concrets, comme le protocole de validation de contenu et le protocole
3023 de catalogue de contenu. Les protocoles concrets sont définis dans une section ultérieure du présent

3024 document.

3025 **8.4.1 Type-de-demande-de-service-de-gestion-de-contenu**

3026 Le Type-de-demande-de-service-de-gestion-de-contenu DOIT constituer le type de base abstrait de
3027 l'ensemble des demandes envoyées d'un registre à un service de gestion de contenu.

3028 **8.4.1.1 Syntaxe**

```
3029 <complexType name="ContentManagementServiceRequestType">  
3030 <complexContent>  
3031 <extension base="rs:RegistryRequestType">  
3032 <sequence>  
3033 <element name="OriginalContent"  
3034 type="rim:RegistryObjectListType"/>  
3035 <element name="InvocationControlFile"  
3036 type="rim:ExtrinsicObjectType" maxOccurs="unbounded" minOccurs="0"/>  
3037 </sequence>  
3038 </extension>  
3039 </complexContent>  
3040 </complexType>
```

3041

3042 **8.4.1.2 Paramètres**

3043 Les paramètres ci-dessous sont soit nouvellement définis pour le type en question, soit hérités. De plus,
3044 ils possèdent une sémantique additionnelle qui va au-delà de celle établie dans la description du type de
3045 base.

- 3046 ▪ *Fichier-de-contrôle-d'appel* : Ce paramètre désigne l'Objet-extrinsèque qui s'applique à
3047 un item de référentiel que l'appelant souhaite déterminer à titre de fichier de contrôle
3048 d'appel. La présente spécification ne précise pas le format de ce fichier. Il DOIT y avoir un
3049 item de référentiel correspondant comme fichier joint de la demande. Cet item de
3050 référentiel DEVRAIT être soumis aux mêmes règles que les fichiers joints d'une
3051 Demande-d'inscription-d'objet et d'une Demande-de-mise-à-jour-d'objet;
- 3052 ▪ *Contenu-original* : Ce paramètre établit les Objets-de-registre qui seront traités par le
3053 service de gestion de contenu. Pour ce qui est des instances d'Objet-extrinsèque
3054 contenues dans le Contenu-original, il PEUT y avoir des items de référentiel à titre de
3055 fichiers joints de la Demande-de-service-de-gestion-de-contenu. La présente
3056 spécification ne détermine pas le format des items de référentiel de ce genre. Ceux-ci
3057 DEVRAIENT être soumis aux mêmes règles que les fichiers joints d'une Demande-
3058 d'inscription-d'objet et d'une Demande-de-mise-à-jour-d'objet.

3059

3060 **8.4.1.3 Renvois**

3061 Cette demande renvoie une Réponse-de-service-de-gestion-de-contenu. La section 8.4.2 renferme des
3062 détails à ce sujet.

3063 **8.4.1.4 Exceptions**

3064 En plus des exceptions renvoyées par les types de demande de base, les exceptions suivantes
3065 PEUVENT l'être également :

- 3066 ▪ *Exception-liée-à-un-item-de-référentiel-manquant* : Signifie que l'appelant n'a pas fourni
3067 d'item de référentiel à titre de fichier joint pour la demande, alors que le Service l'exigeait;
- 3068 ▪ *Exception-liée-à-un-fichier-de-contrôle-d'appel* : Démontre que le Fichier-de-contrôle-
3069 d'appel transmis par l'appelant ne s'apparie pas à celui demandé par le Service;
- 3070 ▪ *Exception-de-contenu-non-géré* : Révèle que le Service en question ne peut pas accueillir

3071 le contenu transmis par l'appelant.
3072

3073 8.4.2 Type-de-réponse-de-service-de-gestion-de-contenu

3074 Le Type-de-réponse-de-service-de-gestion-de-contenu est envoyé par un service de gestion de contenu à
3075 titre de réponse à un Type-de-demande-de-service-de-gestion-de-contenu. Il représente le type de base
3076 abstrait pour l'ensemble des réponses transmises dans un registre à partir d'un service de gestion de
3077 contenu. Il assure le prolongement du Type-de-réponse-de-registre et ne définit aucun nouveau
3078 paramètre.

3079

3080 8.4.2.1 Syntaxe

```
3081 <complexType name="ContentManagementServiceResponseType">  
3082   <complexContent>  
3083     <extension base="rs:RegistryResponseType">  
3084       <sequence>  
3085         </sequence>  
3086       </extension>  
3087     </complexContent>  
3088   </complexType>
```

3089

3090 8.4.2.2 Paramètres

3091 On n'établit aucun nouveau paramètre autre que ceux hérités du Type-de-réponse-de-registre.
3092

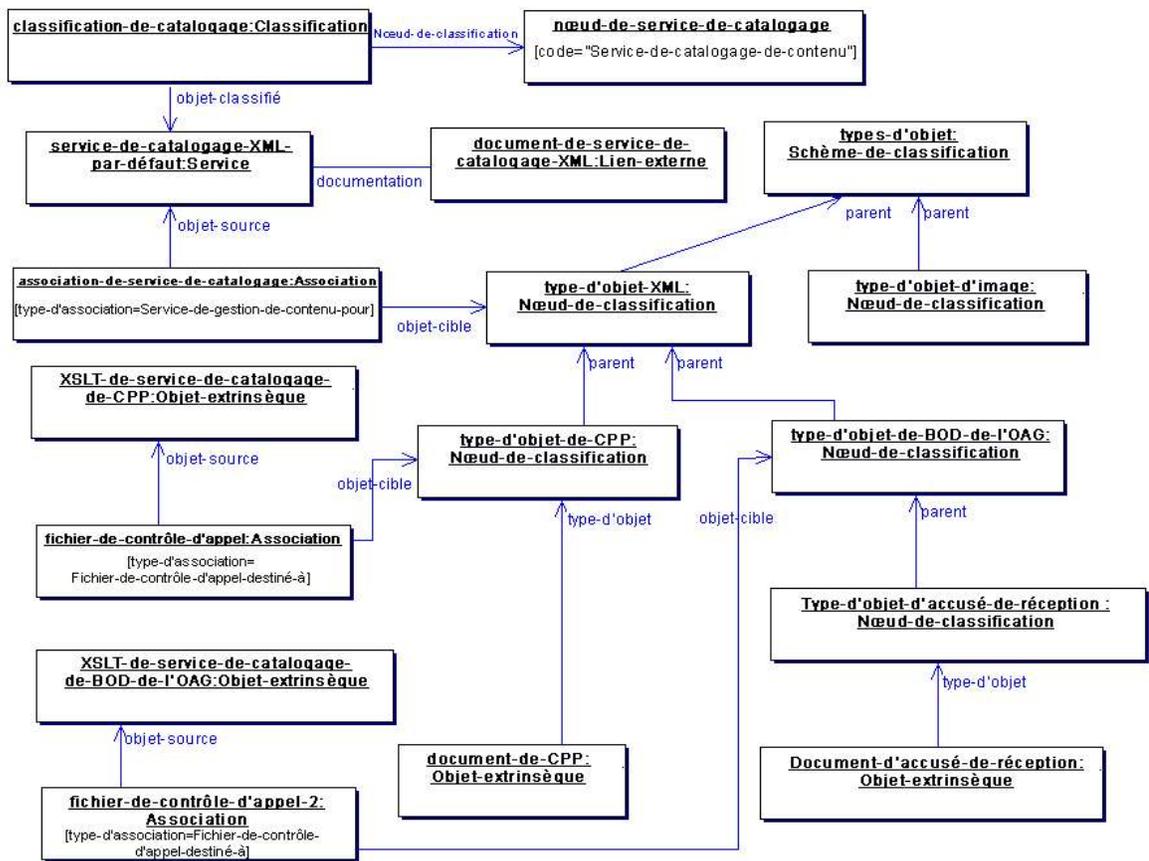
3093 8.5 Publication et configuration d'un service de gestion de contenu

3094 Tout demandeur PEUT inscrire un service de gestion de contenu arbitraire dans un registre ebXML. Ce
3095 service DOIT être publié par le biais de l'interface normalisée de Gestionnaire-de-cycle-de-vie.

3096 Le demandeur DOIT utiliser la Demande-d'inscription-d'objet et la Demande-de-mise-à-jour-d'objet
3097 normalisées pour procéder à certaines publications.

- 3098 ○ C'est ce qu'il DOIT faire pour publier une instance de Service dans le cas d'un service de gestion
3099 de contenu. Ce phénomène est bien illustré dans le Diagramme 17, qui présente un service-de-
3100 catalogage-XML-par-défaut dans le coin supérieur gauche. L'instance de Service DOIT être
3101 caractérisée par la présence d'une Association et d'un Nœud-de-classification dans le Schème-
3102 de-classification de Type-d'objet canonique, comme le définit le [ebRIM]. Le Service DOIT
3103 constituer l'objet-source tandis que le Nœud-de-classification DOIT représenter l'objet-cible. Cette
3104 association favorise le rattachement entre le Service et le Type-d'objet en question. Le type-
3105 d'association propre à l'instance d'Association DOIT être appelé « Service-de-gestion-de-contenu-
3106 pour ». Le Service DOIT être classifié par le Schème-de-classification de Service-de-gestion-de-
3107 contenu canonique, conformément au [ebRIM]. Par exemple, il peut l'être à titre de « Service-de-
3108 validation-de-contenu » ou de « Service-de-catalogage-de-contenu ».
- 3109 ○ L'instance de Service PEUT être classifiée par un Nœud-de-classification selon le Schème-de-
3110 classification de Modèle-d'appel canonique, comme l'explique le [ebRIM], ce qui permet d'établir
3111 si elle a recours au modèle d'appel incorporé ou au modèle d'appel découplé.
- 3112 ○ L'instance de Service PEUT être classifiée par un Nœud-de-classification en fonction du Schème-
3113 de-classification de Modèle-de-gestion-d'erreur canonique, conformément au [ebRIM], ce qui
3114 permet de déterminer si le Service échouera à la première erreur ou s'il l'enregistrera simplement
3115 comme avertissement pour ensuite continuer. La section 8.6.4 renferme des détails à ce sujet.
- 3116 ○ Le demandeur DOIT se servir des deux types de demande ci-dessus pour publier une instance
3117 de Rattachement-de-services qui est comprise dans l'instance de Service et qui DOIT fournir

- 3118 l'adresse-URI au service de catalogage.
- 3119 ○ Le demandeur DOIT faire de même pour publier une instance optionnelle de Lien-externe par
- 3120 rapport au Rattachement-de-services. Cette instance DOIT avoir sa correspondance physique
- 3121 avec une page Web qui décrit :
- 3122 ▪ le format du contenu géré à cataloguer;
 - 3123 ▪ le format du fichier géré de contrôle d'appel.
- 3124 Il est important de mentionner qu'aucun Lien-de-spécification n'est nécessaire puisque la
- 3125 présente spécification [ebRS] est implicite pour les services de catalogage de contenu.
- 3126 ○ Le demandeur DOIT respecter la même règle pour publier au moins un fichier de contrôle d'appel
- 3127 qui prend la forme d'une paire d'Objet-extrinsèque et d'item de référentiel. L'Objet-extrinsèque
- 3128 propre au fichier de contrôle d'appel DOIT englober une Association requise qui dispose d'une
- 3129 valeur de type-d'association citant un Nœud-de-classification descendant du Nœud-de-
- 3130 classification canonique « Fichier-de-contrôle-d'appel-destiné-à ». Ce phénomène est bien illustré
- 3131 par les objets XSLT-de-service-de-catalogage-de-CPP et XSLT-de-service-de-catalogage-de-
- 3132 BOD-de-l'OAG, qui figurent dans le Diagramme 17 (à gauche). Le fichier de contrôle d'appel
- 3133 DOIT désigner l'objet-source, alors que l'objet-cible DOIT se traduire par un Nœud-de-
- 3134 classification du Schème-de-classification de Type-d'objet canonique.
- 3135 ○



3137
3138 **Diagramme 17 : Configuration de service de catalogage**

3139 Le Diagramme 17 donne un exemple de configuration de service de catalogage XML canonique qui se

3140 rattache au type-d'objet d'un contenu XML. On peut utiliser ce service pour tout contenu XML dont

3141 l'attribut type-d'objet garde en mémoire une référence au Nœud-de-classification de Type-d'objet-XML ou
3142 à l'un de ses descendants.

3143 Le diagramme démontre également deux différents fichiers de contrôle d'appel, soit XSLT-de-service-de-
3144 catalogage-de-CPP et XSLT-de-service-de-catalogage-de-BOD-de-l'OAG, dont on peut se servir pour
3145 cataloguer des documents d'objet d'affaires (BOD) de l'OAG et des CPP ebXML, respectivement.

3146 **8.5.1 Services multiples de gestion de contenu et fichiers multiples de** 3147 **contrôle d'appel**

3148 La présente spécification permet aux applications d'inscrire de multiples services de gestion de contenu
3149 du même type (par exemple, la validation et le catalogage) ainsi que de multiples fichiers de contrôle
3150 d'appel pour le même type-d'objet. Les services de gestion de contenu du même type de service qui
3151 s'appliquent au même Type-d'objet sont désignés sous l'appellation de *services homologues de gestion*
3152 *de contenu*.

3153 S'il existe de multiples services de gestion de contenu et fichiers de contrôle d'appel pour le même Type-
3154 d'objet, il DOIT y avoir une association non ambiguë entre un service de gestion de contenu quelconque
3155 et son fichier de contrôle d'appel. Cette caractéristique DOIT être définie par une instance d'Association
3156 qui possède une valeur de type-d'association. Celle-ci DOIT citer un Nœud-de-classification qui se traduit
3157 par un descendant du Nœud-de-classification canonique « Fichier-de-contrôle-d'appel-destiné-à », où
3158 l'Objet-extrinsèque particulier à chaque fichier de contrôle d'appel constitue l'objet-source et où le Service
3159 représente l'objet-cible.

3160 L'ordre d'appel des services homologues de gestion de contenu n'est pas défini. On PEUT toutefois le
3161 déterminer d'une manière propre au registre.

3162 **8.6 Appel d'un service de gestion de contenu**

3163 La section 8.6 décrit la façon dont un registre quelconque appelle un service de gestion de contenu.

3164 **8.6.1 Algorithme de résolution propre aux services et aux fichiers de** 3165 **contrôle d'appel**

3166 Si un registre quelconque reçoit une inscription d'Objet-de-registre, il DOIT utiliser l'algorithme ci-dessous
3167 pour déterminer ou résoudre les services de gestion de contenu et les fichiers de contrôle d'appel dont il
3168 faut se servir pour assurer la gestion dynamique du contenu de cet Objet-de-registre.

3169

- 3170 1. Obtenir l'attribut type-d'objet de l'Objet-de-registre.
- 3171 2. Interroger le registre pour déterminer si le Nœud-de-classification cité par le type-d'objet désigne l'objet-
3172 cible d'une Association dont le type-d'association est *Service-de-gestion-de-contenu-pour*. Si l'Association
3173 recherchée n'a pas été détectée pour ce Nœud-de-classification, répéter cette étape en ayant recours au
3174 Nœud-de-classification parent de celui-ci. Répéter l'étape jusqu'à ce que l'Association recherchée soit
3175 trouvée ou jusqu'à ce que le parent se traduise par le Schème-de-classification. Si l'Association recherchée a
3176 été repérée, effectuer les étapes suivantes pour chaque instance d'Association de ce genre.
- 3177 3. Voir à ce que l'objet-source de l'Association recherchée soit une instance de Service. Si tel n'est pas le cas,
3178 enregistrer une Exception-de-configuration-invalide. Si, cependant, il s'agit bien d'une instance de Service,
3179 l'utiliser comme service de gestion de contenu pour l'Objet-de-registre.
- 3180 4. Interroger le registre pour établir si le Nœud-de-classification de type-d'objet constitue l'objet-cible d'au
3181 moins une Association dont la valeur de type-d'association cite un Nœud-de-classification qui représente un
3182 descendant du Nœud-de-classification canonique *Fichier-de-contrôle-d'appel-destiné-à*. Si l'Association
3183 recherchée n'a pas été détectée pour ce Nœud-de-classification, répéter cette étape en se servant du Nœud-
3184 de-classification parent de celui-ci. Répéter l'étape jusqu'à ce que l'Association recherchée soit trouvée ou
3185 jusqu'à ce que le parent désigne le Schème-de-classification.
- 3186 5. Si l'Association recherchée a été trouvée, s'assurer que son objet-source soit une instance d'Objet-

3187 extrinsèque. Si tel n'est pas le cas, enregistrer une Exception-de-configuration-invalide. Si, par contre, il
3188 s'agit bien d'une instance d'Objet-extrinsèque, utiliser son item de référentiel comme fichier de contrôle
3189 d'appel. S'il existe de multiples Fichiers-de-contrôle-d'appel, ceux-ci DOIVENT tous être révélés au
3190 moment de l'appel du Service.

3191 Cet algorithme favorise l'utilisation de la hiérarchie de type-d'objet en vue de la configuration des services
3192 de gestion de contenu et des fichiers de contrôle d'appel. Cette configuration s'effectue à des degrés
3193 différents de spécificité ou de spécialisation quant au type de contenu.

3194 **8.6.2 Piste de vérification et contenu catalogué**

3195 Le contenu catalogué produit à la suite de l'appel d'un service de gestion de contenu dispose d'une piste
3196 de vérification qui concorde avec les instances d'Objet-de-registre inscrites par les applications de
3197 registre. Celles-ci n'inscrivent toutefois pas de contenu catalogué, ce qui signifie que l'attribut utilisateur
3198 des instances d'Événement-vérifiable quant à un tel contenu cite l'objet de Service du service de gestion
3199 de contenu qui a produit le contenu catalogué. Il s'agit là d'une manière efficace de distinguer le contenu
3200 catalogué de celui inscrit par les applications de registre.

3201 **8.6.3 Intégrité référentielle**

3202 Tout registre DOIT maintenir une intégrité référentielle entre les Objets-de-registre et les items de
3203 référentiel pendant l'appel d'un service de gestion de contenu.

3204 **8.6.4 Gestion d'erreur**

3205 Si le service de gestion de contenu est classifié par le Nœud-de-classification « Arrêt-dès-que-erreur »
3206 selon le Schème-de-classification de Modèle-de-gestion-d'erreur canonique, comme le définit le [ebRIM],
3207 le registre DOIT cesser le traitement de la Demande-d'inscription-d'objet et de la Demande-de-mise-à-
3208 jour-d'objet, et renvoyer le statut « Échec » dès le renvoi d'une première erreur par un appel de service de
3209 gestion de contenu.

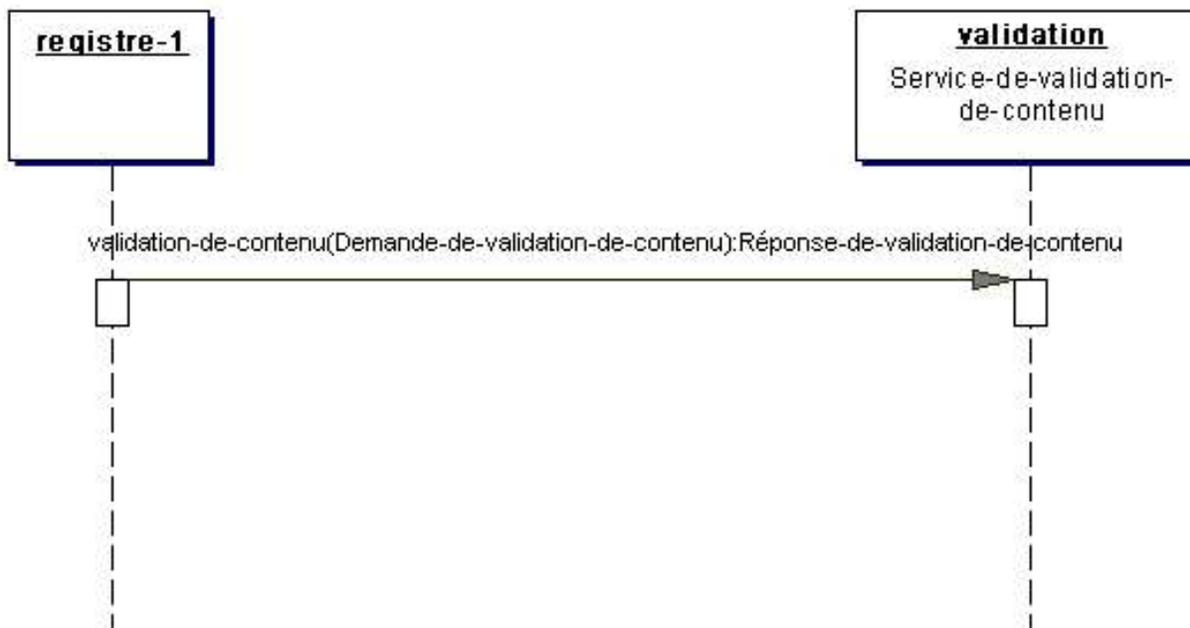
3210 Si le service de gestion de contenu est classifié par le Nœud-de-classification « Enregistrer-l'erreur-et-
3211 continuer » dans le cadre du Modèle-de-gestion-d'erreur, le registre DOIT poursuivre le traitement de la
3212 Demande-d'inscription-d'objet et de la Demande-de-mise-à-jour-d'objet. Le registre DOIT aussi voir à ce
3213 qu'aucune erreur d'appel de service de gestion de contenu ne touche le stockage des Objets-de-registre
3214 et des items de référentiel inscrits. Les erreurs de ce genre DEVRAIENT être enregistrées à titre
3215 d'avertissements et figurer dans la Réponse-de-registre transmise à l'application. Dans un tel cas, si les
3216 métadonnées et le contenu inscrits sont stockés avec succès, le registre DOIT renvoyer une réponse
3217 normale ayant le statut « Réussite » même si des erreurs se sont produites au cours de l'appel
3218 dynamique d'au moins un service de gestion de contenu.

3219 **8.7 Protocole de validation de contenu**

3220 L'interface d'un service de validation de contenu DOIT implanter une seule méthode appelée validation-
3221 de-contenu. Cette méthode accepte une Demande-de-validation-de-contenu à titre de paramètre et
3222 renvoie une Réponse-de-validation-de-contenu en tant que réponse si aucune erreur ne survient.

3223 L'élément de Contenu-original d'une Demande-de-validation-de-contenu DOIT renfermer exactement un
3224 Objet-de-registre qui doit être catalogué. La Réponse-de-validation-de-contenu qui en ressort comprend
3225 l'attribut statut, qui détermine si l'Objet-de-registre (et son contenu) est valide ou non.

3226 Le protocole de validation de contenu ne présente les détails liés à l'implantation pour aucun service
3227 particulier de validation de contenu.



3229

Diagramme 18 : Protocole de validation de contenu

3230 **8.7.1 Demande-de-validation-de-contenu**

3231 On se sert de la Demande-de-validation-de-contenu pour transmettre un contenu à un service de
 3232 validation de contenu de façon qu'il puisse valider l'Objet-de-registre en question et tout contenu connexe.
 3233 L'Objet-de-registre se traduit généralement par un Lien-externe (dans le cas d'un contenu externe) ou un
 3234 Objet-extrinsèque. La Demande-de-validation-de-contenu entraîne l'extension du Type-de-demande-de-
 3235 service-de-gestion-de-contenu du type de base.

3236 **8.7.1.1 Syntaxe**

```

3237 <element name="ValidateContentRequest">
3238   <complexType>
3239     <complexContent>
3240       <extension base="cms:ContentManagementServiceRequestType">
3241         <sequence>
3242           </sequence>
3243       </extension>
3244     </complexContent>
3245   </complexType>
3246 </element>
  
```

3247

3248 **8.7.1.2 Paramètres**

3249 Les paramètres ci-dessous sont soit nouvellement définis pour le type en question, soit hérités. De plus,
 3250 ils possèdent une sémantique additionnelle qui va au-delà de celle établie dans la description du type de
 3251 base.

- 3252 ▪ *Fichier-de-contrôle-d'appel* : Ce paramètre est hérité du type de base et peut n'être pas
 3253 présent. S'il est présent, son format est déterminé par le service de validation de contenu.
- 3254 ▪ *Contenu-original* : Ce paramètre est lui aussi hérité du type de base. Il DOIT renfermer
 3255 exactement un Objet-de-registre (par exemple un Lien-externe ou un Objet-extrinsèque)
 3256 et potentiellement un contenu connexe. La présente spécification n'établit pas le format

3257 du contenu. S'il s'agit d'un Objet-extrinsèque, il PEUT y avoir un item de référentiel
3258 correspondant à titre de fichier joint de la demande, soit le contenu. Cet item DEVRAIT
3259 être soumis aux mêmes règles que les fichiers joints d'une Demande-d'inscription-d'objet
3260 et d'une Demande-de-mise-à-jour-d'objet.
3261

3262 **8.7.1.3 Renvois**

3263 Cette demande renvoie une Réponse-de-validation-de-contenu. La section 8.7.2 renferme des détails à
3264 ce sujet.

3265 **8.7.1.4 Exceptions**

3266 En plus des exceptions renvoyées par les types de demande de base, l'exception suivante PEUT l'être
3267 également :

- 3268 ▪ *Exception-liée-à-un-contenu-invalide* : Signifie que le contenu spécifié est invalide.
3269 L'exception DEVRAIT comprendre suffisamment de renseignements pour permettre à
3270 l'application de déterminer la façon de le rendre valide.
3271

3272 **8.7.2 Réponse-de-validation-de-contenu**

3273 La Réponse-de-validation-de-contenu est transmise par le service de validation de contenu à titre de
3274 réponse à une Demande-de-validation-de-contenu.
3275

3276 **8.7.2.1 Syntaxe**

```
3277 <element name="ValidateContentResponse">  
3278   <complexType>  
3279     <complexContent>  
3280       <extension base="cms:ContentManagementServiceResponseType">  
3281         <sequence>  
3282           </sequence>  
3283         </extension>  
3284       </complexContent>  
3285     </complexType>  
3286   </element>
```

3287

3288 **8.7.2.2 Paramètres**

3289 Les paramètres ci-dessous sont soit nouvellement définis pour le type en question, soit hérités. De plus,
3290 ils possèdent une sémantique additionnelle qui va au-delà de celle établie dans la description du type de
3291 base.

- 3292 ▪ *statut* : Il s'agit d'un attribut hérité. On se sert de cette valeur énumérée pour préciser le
3293 statut de la demande. Voici les valeurs de ce statut :
3294
3295 • Réussite – Ce statut définit la validité du contenu spécifié dans la Demande-de-
3296 validation-de-contenu;
3297 • Échec – Ce statut illustre l'échec de la demande. Si l'erreur renvoyée constitue
3298 une *Exception-liée-à-un-contenu-invalide*, le contenu déterminé dans la
3299 Demande-de-validation-de-contenu était invalide. Si un autre échec quelconque
3300 est détecté au cours du traitement de la demande, une erreur différente PEUT
3301 être renvoyée.
3302

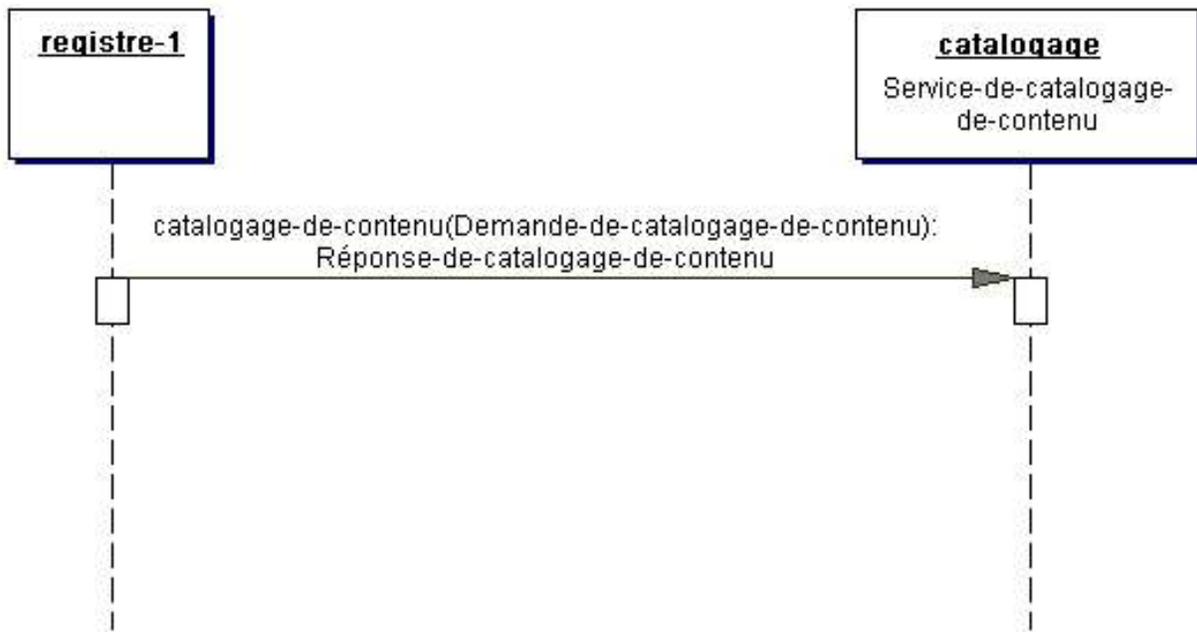
3303 8.8 Protocole de catalogage de contenu

3304 L'interface d'un service de catalogage de contenu DOIT implanter une seule méthode dénommée
3305 catalogage-de-contenu. Cette méthode accepte une Demande-de-catalogage-de-contenu à titre de
3306 paramètre et renvoie une Réponse-de-catalogage-de-contenu en tant que réponse si aucune erreur ne
3307 survient.

3308 La Demande-de-catalogage-de-contenu PEUT renfermer des items de référentiel qui doivent être
3309 catalogués. La Réponse-de-catalogage-de-contenu qui en découle englobe les métadonnées et, le cas
3310 échéant, le contenu qui est produit ou mis à jour par le service de catalogage de contenu à la suite du
3311 catalogage des items de référentiel établis.

3312 Le protocole de catalogage de contenu ne présente les détails liés à l'implantation pour aucun service
3313 particulier de catalogage de contenu.

3314



3315

Diagramme 19 : Protocole de catalogage de contenu

3316 8.8.1 Demande-de-catalogage-de-contenu

3317 On se sert de la Demande-de-catalogage-de-contenu pour transmettre un contenu à un service de
3318 catalogage de contenu de façon qu'il puisse créer des métadonnées sur le catalogage pour l'Objet-de-
3319 registre en question et tout contenu connexe. L'Objet-de-registre se traduit généralement par un Lien-
3320 externe (dans le cas d'un contenu externe) ou un Objet-extrinsèque. La Demande-de-catalogage-de-
3321 contenu entraîne l'extension du Type-de-demande-de-service-de-gestion-de-contenu du type de base.

3322 8.8.1.1 Syntaxe

```
3323 <element name="CatalogContentRequest">
3324   <complexType>
3325     <complexContent>
3326       <extension base="cms:ContentManagementServiceRequestType">
3327         <sequence>
3328           </sequence>
3329       </extension>
3330     </complexContent>
3331   </complexType>
3332 </element>
```

3333
3334

3335 **8.8.1.2 Paramètres**

3336 Les paramètres ci-dessous sont soit nouvellement définis pour le type en question, soit hérités. De plus,
3337 ils possèdent une sémantique additionnelle qui va au-delà de celle établie dans la description du type de
3338 base.

- 3339 ▪ *Fichier-de-contrôle-d'appel* : Ce paramètre est hérité du type de base. S'il est présent,
3340 son format est déterminé par le service de catalogage de contenu.
- 3341 ▪ *Contenu-original* : Ce paramètre est lui aussi hérité du type de base. Il DOIT renfermer
3342 exactement un Objet-de-registre (par exemple, un Lien-externe ou un Objet-extrinsèque)
3343 et potentiellement un contenu connexe. La présente spécification n'établit pas le format
3344 du contenu. S'il s'agit d'un Objet-extrinsèque, il PEUT y avoir un item de référentiel
3345 correspondant à titre de fichier joint de la demande, soit le contenu. Cet item DEVRAIT
3346 être soumis aux mêmes règles que les fichiers joints d'une Demande-d'inscription-d'objet
3347 et d'une Demande-de-mise-à-jour-d'objet.
3348

3349 **8.8.1.3 Renvois**

3350 Cette demande renvoie une Réponse-de-catalogage-de-contenu. La section 8.8.2 renferme des détails à
3351 ce sujet.

3352 **8.8.1.4 Exceptions**

3353 En plus des exceptions renvoyées par les types de demande de base, l'exception suivante PEUT l'être
3354 également :

- 3355 ▪ *Exception-liée-au-catalogage* : Signifie qu'une exception a été détectée dans l'algorithme
3356 de catalogage par rapport au service.
3357

3358 **8.8.2 Réponse-de-catalogage-de-contenu**

3359 La Réponse-de-catalogage-de-contenu est transmise par le service de catalogage de contenu à titre de
3360 réponse à une Demande-de-catalogage-de-contenu.

3361

3362 **8.8.2.1 Syntaxe**

```
3363 <element name="CatalogContentResponse">  
3364   <complexType>  
3365     <complexContent>  
3366       <extension base="cms:ContentManagementServiceResponseType">  
3367         <sequence>  
3368           <element name="CatalogedContent"  
3369             type="rim:RegistryObjectListType"/>  
3370         </sequence>  
3371       </extension>  
3372     </complexContent>  
3373   </complexType>  
3374 </element>
```

3375

3376 **8.8.2.2 Paramètres**

3377 Les paramètres ci-dessous sont soit nouvellement définis pour le type en question, soit hérités. De plus,

3378 ils possèdent une sémantique additionnelle qui va au-delà de celle établie dans la description du type de
3379 base.

- 3380 ▪ *Contenu-catalogué* : Ce paramètre désigne une collection d'instances d'Objet-de-registre
3381 qui ont été créées ou mises à jour à la suite du catalogage dynamique d'un contenu par
3382 un service de catalogage de contenu. Ce service peut entraîner l'ajout de métadonnées,
3383 comme des Classifications, des Identifiants-externes, des noms et des descriptions, à
3384 l'élément de Contenu-catalogué. En cas de modification par la demande de l'item de
3385 référentiel original, il PEUT y avoir un item de référentiel d'accompagnement à titre de
3386 fichier joint du message de réponse en question.

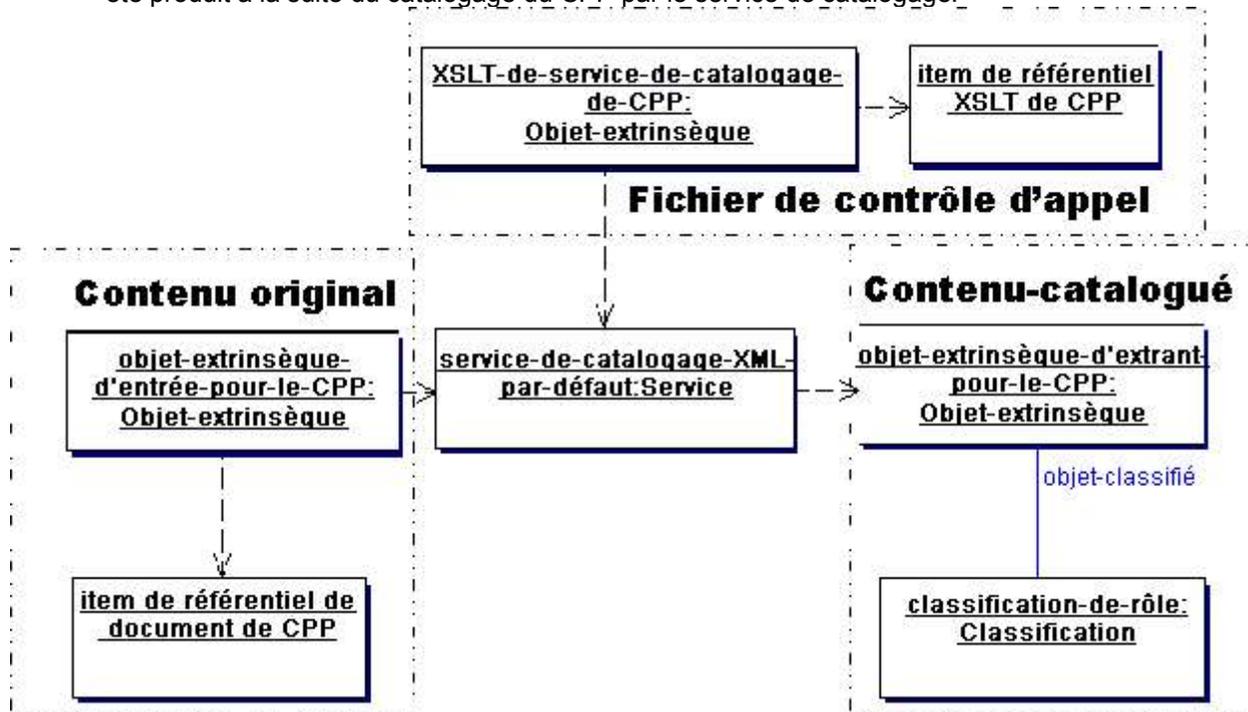
3387

3388

3389 8.9 Exemple typique : service de catalogage XML canonique

3390 Le Diagramme 20 présente une instance UML qui illustre la méthode d'utilisation d'un service de
3391 catalogage de contenu. Il s'agit ici du service de catalogage XML canonique normatif, qui est décrit dans
3392 la section 8.10.

- 3393 ○ Au centre se trouve un service de catalogage de contenu appelé service de catalogage-XML-par-
3394 défaut.
- 3395 ○ À gauche se trouvent un item de référentiel de CPP et son Objet-extrinsèque appelé objet-
3396 extrinsèque-d'entrée-pour-le-CPP, qui est saisi à titre de contenu original pour le service de
3397 catalogage-XML-par-défaut.
- 3398 ○ En haut se trouvent un item de référentiel de feuille de style XSLT et son Objet-extrinsèque qui
3399 est configuré comme fichier de contrôle d'appel pour le service de catalogage-XML-par-défaut.
- 3400 ○ À droite se trouve l'objet-extrinsèque-d'extrant-pour-le-CPP, qui représente l'Objet-extrinsèque
3401 modifié pour le CPP. On peut également voir une classification-de-rôle de Classification, qui
3402 classe le CPP selon l'élément Rôle contenu dans celui-ci. Il s'agit du contenu catalogué qui a
3403 été produit à la suite du catalogage du CPP par le service de catalogage.



3405 **Diagramme 20 : Exemple de catalogage de CPP par le service de catalogage XML canonique**

3406

3407 **8.10 Service de catalogage de contenu XML canonique**

3408 Tout registre ebXML DOIT fournir naturellement le service de catalogage de contenu XML canonique à
3409 titre de service prédéfini qui comprend les contraintes suivantes :

- 3410 • Il y a exactement une instance de Service pour le service de catalogage de contenu XML
3411 canonique;
- 3412 • Le Service constitue un moteur XSLT;
- 3413 • On peut appeler le Service avec précisément un fichier de contrôle d'appel;
- 3414 • Le contenu original du Service DOIT se traduire par un document XML;
- 3415 • Le contenu catalogué du Service DOIT désigner un document XML;
- 3416 • Le fichier de contrôle d'appel DOIT représenter une feuille de style XSLT;
- 3417 • Chaque appel du Service PEUT se faire à l'aide d'un fichier différent de contrôle d'appel (feuille
3418 de style XSLT), selon le type-d'objet de l'Objet-de-registre catalogué. Chaque type-d'objet
3419 DEVRAIT posséder sa propre feuille de style XSLT. Par exemple, les documents de CPP ebXML
3420 DEVRAIENT disposer d'une feuille de style XSLT spécialisée de contrôle d'appel de CPP ebXML;
- 3421 • Le Service DOIT avoir au moins un document XML en entrée sous la forme d'un Objet-de-
3422 registre. Il s'agit généralement d'un Objet-extrinsèque ou d'un Lien-externe;
- 3423 • Le Service PEUT détenir tout au plus un document XML en entrée additionnel qui désigne le
3424 contenu représenté par l'Objet-de-registre (par exemple, un document de CPP ou un profil de
3425 conformité de HL7). La deuxième entrée optionnelle DOIT être citée dans la feuille de style XSLT
3426 par le biais de la fonction « document », à l'aide du nom du document spécifié par l'« item-de-
3427 référentiel » variable, comme dans le « document(\$item-de-référentiel) ». Tout registre DOIT
3428 définir l'« item-de-référentiel » variable au moment où il appelle le service de catalogage XML
3429 canonique;
- 3430 • Le service de catalogage de contenu XML canonique DOIT entraîner l'application de la feuille de
3431 style XSLT à l'égard du document d'instance XML en entrée dans une transformation XSLT, en
3432 vue de la production de la sortie cataloguée.

3433 Le service de catalogage de contenu XML canonique constitue une caractéristique normative requise d'un
3434 registre ebXML.

3435 **8.10.1 Publication du service de catalogage de contenu XML canonique**

3436 Tout registre ebXML DOIT fournir naturellement le service de catalogage de contenu XML canonique à
3437 titre de service prédéfini. Celui-ci DOIT être publié dans le registre dans le cadre de l'initialisation
3438 intrinsèque des données canoniques nécessaires du registre.

3439

9 Gestion de registres en coopération

3440

Le chapitre 9 décrit les capacités et les protocoles qui permettent à de multiples registres ebXML de coopérer dans le but de respecter les cas d'utilisation évolués.

3441

3442

9.1 Cas d'utilisation axés sur des registres en coopération

3443

Ci-dessous se trouve une liste de cas d'utilisation qui mettent en évidence différentes méthodes de coopération entre des registres ebXML.

3444

3445

9.1.1 Références d'objet entre registres

3446

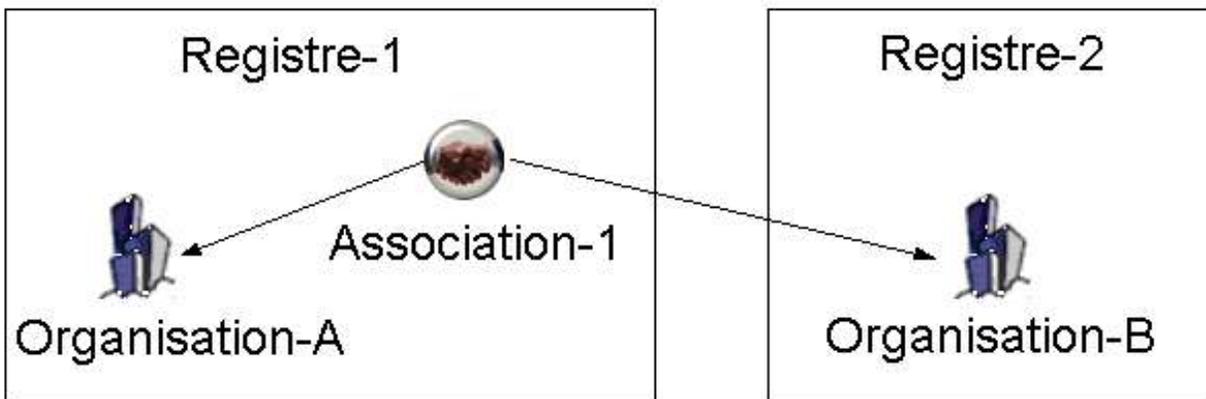
Une organisation demanderesse quelconque souhaite inscrire dans un registre un Objet-de-registre de façon que cet objet en cite un autre qui se trouve dans un autre registre.

3447

3448

À titre d'exemple, on peut mentionner le cas où un Objet-de-registre d'un registre est associé à celui d'un autre registre.

3449



3451

Diagramme 21 : Références d'objet entre registres

3452

3453

9.1.2 Interrogations fédérées

3454

Une application quelconque souhaite émettre une seule interrogation par rapport à de multiples registres et obtenir une seule réponse qui contiendrait des résultats fondés sur l'ensemble des données figurant dans tous les registres. Du point de vue de l'application, elle transmet son interrogation par rapport à un seul registre local qui réunit toutes les données de l'ensemble des registres physiques.

3455

3456

3457

3458

9.1.3 Mise en cache locale des données d'un autre registre

3459

Un registre de destination quelconque désire mettre en cache les données, en partie ou en totalité, d'un autre registre source qui serait disposé à les partager. L'ensemble de données partagé est copié du registre source au registre de destination et peut être vu par les interrogations de celui-ci même si le registre source n'est pas disponible.

3460

3461

3462

3463

On peut souhaiter procéder à la mise en cache locale des données afin d'améliorer la performance et la disponibilité de l'accès aux objets.

3464

3465

En voici un exemple : un Objet-de-registre d'un registre est associé à celui d'un autre registre, et le premier registre met en cache localement le deuxième Objet-de-registre.

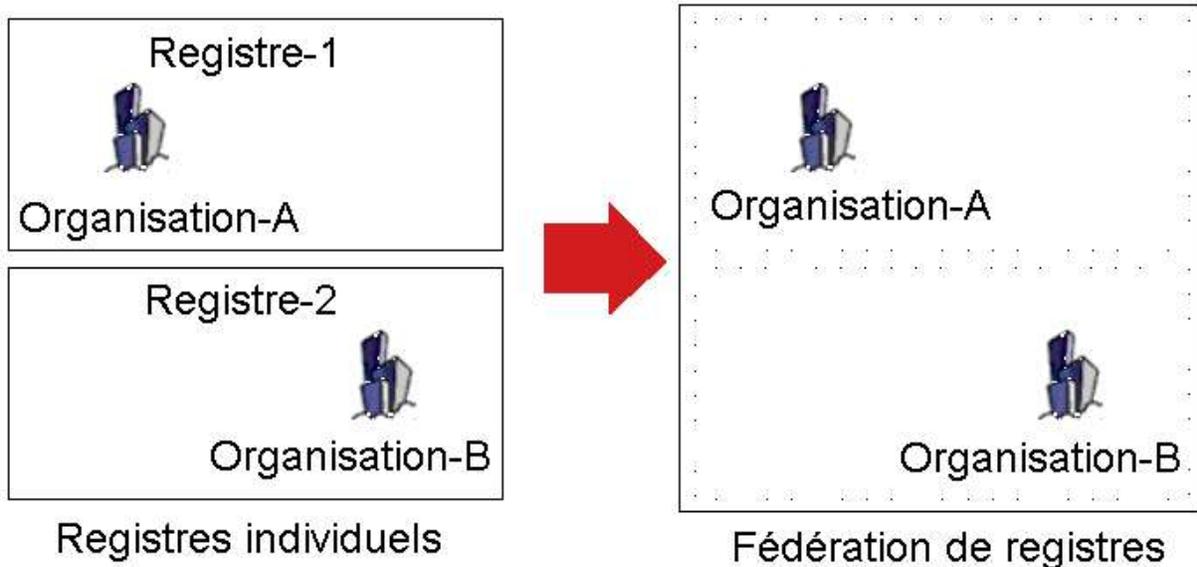
3466

3467 **9.1.4 Déplacement d'objet**

3468 Une organisation demanderesse désire déplacer vers un autre registre ses Objets-de-registre et items de
3469 référentiel à partir du registre où ceux-ci ont été inscrits.

3470 **9.2 Fédérations de registres**

3471 Une fédération de registres désigne un groupe de registres qui ont convenu de leur gré de former une
3472 union basée sur la souplesse d'association. Une fédération de ce genre peut être fondée sur des
3473 particularités et intérêts opérationnels communs que les registres peuvent partager entre eux. Les
3474 fédérations de registre se présentent aux applications de registre comme un seul registre logique.



3476 **Diagramme 22 : Fédérations de registre**

3477 Les fédérations de registre s'inspirent d'un modèle de pairs où tous les registres participants sont égaux.
3478 Chacun de ces registres est désigné sous l'appellation de *registre homologue*. Il n'y a aucune distinction
3479 entre l'opérateur de registre qui a créé une fédération et ceux qui s'y sont joints plus tard.

3480 Tout opérateur de registre PEUT en tout temps concevoir une fédération de registre. Toute fédération
3481 nouvellement créée DOIT avoir exactement un registre homologue représentant le registre exploité par
3482 l'opérateur de registre qui a produit la fédération.

3483 Tout registre PEUT à tout moment décider de se joindre volontairement à une fédération quelconque ou
3484 d'en sortir volontairement.

3485 **9.2.1 Métadonnées de fédération**

3486 Le *Modèle d'information du registre ebXML* définit les classes Registre et Fédération. Les instances de
3487 ces classes et les associations entre ces instances décrivent une fédération et ses membres. De telles
3488 données sur les instances sont appelées métadonnées de fédération. Les classes Registre et Fédération
3489 sont décrites en détail dans le [ebRIM].

3490 Voici un résumé du modèle d'information de la Fédération :

- 3491 ○ Une instance de Fédération constitue une fédération de registre;
- 3492 ○ Une instance de Registre représente un registre qui est membre de la Fédération;
- 3493 ○ Une instance d'Association dont le type-d'association est *a-pour-membre-de-Fédération* désigne
3494 les éléments du registre de la fédération. Elle assure la liaison entre l'instance de Registre et
3495 l'instance de Fédération.

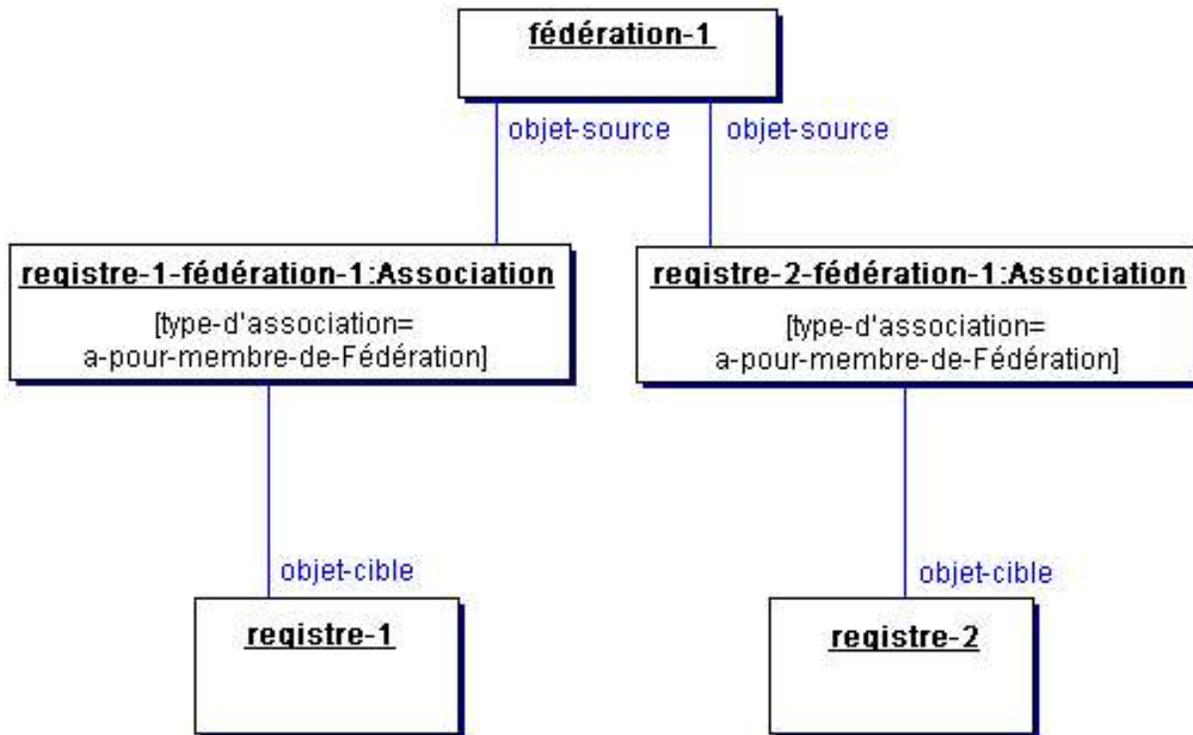


Diagramme 23 : Exemple de métadonnée de fédération

3499 9.2.2 Interrogations locales et interrogations fédérées

3500 Toute fédération se présente aux applications de registre sous la forme de registre logique unifié unique.
 3501 Une Demande-d'interrogation-ad-hoc transmise par une application à un membre de fédération PEUT
 3502 être locale ou fédérée. On ajoute un nouvel attribut booléen appelé *fédéré* à une Demande-
 3503 d'interrogation-ad-hoc pour déterminer si l'interrogation en question est fédérée ou non.

3504 9.2.2.1 Interrogations locales

3505 Si l'attribut fédéré d'une Demande-d'interrogation-ad-hoc a la valeur *FAUX*, l'interrogation en question
 3506 représente une interrogation locale. En l'absence d'un tel attribut, sa valeur par défaut est *FAUX*.
 3507 Toute Demande-d'interrogation-ad-hoc locale n'est traitée que par le registre qui la reçoit. Elle n'est pas
 3508 exploitée pour des données propres à d'autres registres.

3509 9.2.2.2 Interrogations fédérées

3510 Si la valeur de l'attribut *fédéré* d'une Demande-d'interrogation-ad-hoc est *VRAI*, l'interrogation en question
 3511 constitue une interrogation fédérée.

3512 Tout membre de fédération DOIT acheminer n'importe quelle interrogation fédérée qu'elle reçoit vers
 3513 l'ensemble des autres registres de membre de fédération selon la meilleure tentative. S'il n'y a pas de
 3514 portée pour un membre quelconque, peu importe la raison, on PEUT sauter celui-ci.

3515 Si un registre achemine une interrogation fédérée vers d'autres membres de fédération, il DOIT s'assurer
 3516 que l'attribut fédéré a la valeur *FAUX* et que l'attribut *fédération* a la valeur NUL dans le but d'éviter toute
 3517 boucle fermée.

3518 Les interrogations fédérées sont exploitées à partir des données liées à l'ensemble des membres de la
 3519 fédération en question.

3520 Si une application inscrit une interrogation fédérée dans un registre de façon que celle-ci ne spécifie
3521 aucune fédération, et s'il n'existe aucune fédération dans le registre, celui-ci DOIT traiter l'interrogation
3522 comme interrogation locale.

3523 Si une application inscrit une interrogation fédérée qui appelle une interrogation stockée avec paramètres,
3524 le registre DOIT résoudre cette dernière interrogation en fonction de son champ non stocké, et remplacer
3525 toutes les variables par des paramètres fournis par l'utilisateur par rapport à des paramètres contextuels
3526 transmis par le registre. Le registre pourra ensuite acheminer l'interrogation fédérée vers un membre de
3527 fédération.

3528 Si une application inscrit une interrogation itérative fédérée, le registre DOIT se servir de la valeur de
3529 l'attribut *index-de-départ* de la demande originale à titre de valeur du même attribut de la demande
3530 acheminée à chaque membre de fédération. La réponse à la demande originale DOIT se traduire par
3531 l'*union* des résultats découlant de chaque interrogation acheminée. Dans un tel cas, le registre DOIT
3532 renvoyer une valeur de l'attribut *nombre-total-de-résultats* par rapport à la réponse de l'interrogation
3533 fédérée de sorte que cette valeur corresponde au *maximum* de l'ensemble des valeurs de l'attribut
3534 *nombre-total-de-résultats* renvoyées par chaque membre de fédération.

3535 **9.2.2.3 Appartenance à de multiples fédérations**

3536 Tout registre PEUT appartenir à de multiples fédérations. Dans un tel cas, si la valeur de l'attribut *fédéré*
3537 de la Demande-d'interrogation-ad-hoc en question est *VRAI*, le registre DOIT acheminer l'interrogation
3538 fédérée à l'*ensemble* des fédérations dont il fait partie.

3539 De façon subsidiaire, l'application PEUT déterminer l'identifiant d'une fédération en particulier dont le
3540 registre fait partie et le considérer comme la valeur du paramètre *fédération*. Le type de ce paramètre, qui
3541 est tout-URI, établit l'attribut « identifiant » de la fédération recherchée.

3542 En pareil cas, le registre ne DOIT acheminer l'interrogation fédérée que vers la fédération précisée.

3543 **9.2.3 Opérations fédérées de gestion du cycle-de-vie**

3544 La section 9.2.8 renferme des détails sur la façon de créer et de supprimer des fédérations, et la façon
3545 d'adhérer à une fédération et d'en sortir.

3546 On DEVRAIT effectuer l'ensemble des opérations liées au cycle-de-vie pour tout Objet-de-registre au sein
3547 de son propre registre d'origine. On DEVRAIT procéder ainsi en utilisant les opérations établies par
3548 l'interface de Gestionnaire-de-cycle-de-vie. Contrairement aux demandes d'interrogation, les demandes
3549 de gestion du cycle-de-vie ne peuvent accueillir aucune capacité fédérée.

3550 **9.2.4 Fédérations et mise en cache locale des données à distance**

3551 Un membre de fédération n'est pas tenu de maintenir une mémoire cache locale des duplicatas d'Objet-
3552 de-registre et d'item de référentiel qui appartiennent à d'autres membres de la même fédération.

3553 Tout registre PEUT décider de mettre en cache d'une manière locale une partie ou l'ensemble des
3554 données de tout autre registre, peu importe si celui-ci est un membre de fédération ou non. La mise en
3555 cache de données est orthogonale par rapport à une fédération de registre. Elle est décrite dans la
3556 section 9.3.

3557 Il y a par défaut une duplication minimale des membres d'une même fédération; ainsi, l'architecture de
3558 celle-ci réussit bien à réduire l'utilisation de mémoire et de disques pour chaque registre.

3559 La duplication de données est souvent nécessaire à des fins de performance, de montée en échelle et de
3560 tolérance aux pannes.

3561 **9.2.5 Mise en cache de métadonnées de fédération**

3562 Un cas particulier qui s'applique à la mise en cache locale est celui de la mise en cache des instances de
3563 Fédération et de Registre et des Associations connexes qui définissent une fédération quelconque et ses
3564 membres. Les données de ce genre sont désignées sous l'appellation de métadonnées de fédération.
3565 Tout membre d'une fédération est tenu d'assurer pour cette fédération la mise en cache locale des
3566 métadonnées de fédération à partir de la fédération d'origine. La raison qui explique cette exigence cadre

3567 avec un modèle de pairs. De plus, elle favorise une tolérance aux pannes si le registre d'origine d'une
3568 Fédération n'est pas disponible.

3569 Tout membre de fédération DOIT garantir la synchronisation entre les métadonnées de fédération mises
3570 en cache et la copie de base de la Fédération d'origine. Cette synchronisation doit se faire à l'intérieur de
3571 la période déterminée par l'attribut temps-d'attente-pour-la-synchronisation-de-la-duplication de la
3572 Fédération. La synchronisation des métadonnées de fédération mises en cache peut s'effectuer au
3573 moyen d'un appel synchrone ou de la notification d'un événement asynchrone. La caractéristique de
3574 notification d'événement de registre joue ici un rôle important.

3575 **9.2.6 Synchronisation de l'heure entre les registres homologues**

3576 Les membres de fédération ne sont pas tenus de synchroniser leurs horloges systèmes l'un avec l'autre.
3577 Chaque membre DEVRAIT toutefois assurer la synchronisation entre son horloge système et un serveur
3578 d'horloge atomique en fonction du temps d'attente établi par l'attribut temps-d'attente-pour-la-
3579 synchronisation-de-la-duplication de la Fédération.

3580 **9.2.7 Fédérations et sécurité**

3581 En ce qui concerne un registre unique quelconque, les opérations fédérées respectent les mêmes
3582 consignes de sécurité que les opérations normalisées. Cependant, les opérations de fédération
3583 nécessitent souvent une communication de registre à registre. Une telle communication est régie par les
3584 mêmes consignes de sécurité qu'une communication entre une application de registre et un registre. La
3585 seule différence est que le registre demandeur joue le rôle d'application de registre. Aussi, la
3586 communication de registre à registre DEVRAIT se faire dans une voie de communication protégée,
3587 comme HTTP/S. Les membres de fédération DEVRAIENT faire partie de la même fédération SAML si les
3588 registres qui en sont membres implantent le profil SAML de registre décrit dans le chapitre .

3589 **9.2.8 Protocoles de gestion du cycle-de-vie de fédération**

3590 La présente section décrit les diverses opérations qui gèrent le cycle-de-vie d'une fédération et de ses
3591 membres. On exécute ces opérations en utilisant de manière stylisée l'interface normalisée du
3592 Gestionnaire-de-cycle-de-vie d'un registre. Elles constituent des opérations privilégiées. Tout registre
3593 DEVRAIT les restreindre à l'utilisateur de registre qui joue le rôle d'Administrateur-de-registre.

3594 **9.2.8.1 Union avec une fédération**

3595 Les règles suivantes régissent la façon dont un registre s'unit avec une fédération :

- 3596 • Chaque registre DEVRAIT comporter exactement une instance de Registre dont il constitue une
3597 origine. Cette instance est détenue par l'Opérateur-de-registre et peut être transmise dans le
3598 registre par le biais de n'importe quelle méthode propre à l'opérateur. L'instance de Registre NE
3599 DEVRAIT jamais modifier son propre registre d'origine;
- 3600 • Tout registre PEUT présenter une demande d'union avec une fédération en place. Pour ce faire, il
3601 doit inscrire une instance d'association multi-détenteurs qui lie l'instance de Fédération, comme
3602 objet-source, à son instance de Registre, à titre d'objet-cible. Cette opération s'effectue par le
3603 recours à un type-d'association *a-pour-membre-de-Fédération*. Le registre d'origine propre à
3604 l'Association et aux objet de Fédération DOIT être le même.
3605

3606 **9.2.8.2 Création d'une fédération**

3607 Les règles suivantes régissent la manière dont une fédération est créée :

- 3608 • On conçoit une fédération en inscrivant une instance de Fédération dans un registre au moyen
3609 d'une Demande-d'inscription-d'objet;
- 3610 • Le registre dans lequel la fédération est inscrite est appelé *site d'accueil de la fédération*;
- 3611 • Le site d'accueil de la fédération peut ou non être un membre de cette fédération;

3612 • Le site d'accueil de la fédération PEUT renfermer de multiples instances de Fédération.

3613 **9.2.8.3 Sortie d'une fédération**

3614 La règle suivante régit la façon dont un registre quitte une fédération :

3615 Tout registre PEUT en tout temps quitter une fédération en supprimant son instance d'Association *a-pour-*
3616 *membre-de-fédération* qui l'unit à l'instance de Fédération. Cette opération s'effectue au moyen de la
3617 Demande-de-suppression-d'objet normalisée.

3618 **9.2.8.4 Dissolution d'une fédération**

3619 Les règles suivantes régissent la manière dont une fédération est dissoute :

- 3620 • On dissout une fédération en transmettant une Demande-de-suppression-d'objet à son registre
3621 d'origine et en supprimant son instance de Fédération;
- 3622 • La suppression d'une instance de Fédération est gérée par les mêmes règles de contrôle d'accès
3623 que celles qui régissent tout Objet-de-registre;
- 3624 • La suppression d'une instance de Fédération est gérée par les mêmes règles de gestion du cycle-
3625 de-vie que celles qui régissent n'importe quel Objet-de-registre. Cette caractéristique signifie
3626 généralement qu'une fédération NE DOIT PAS être dissoute si elle a des membres de fédération.
3627 Elle PEUT toutefois être déclassée à tout moment, et une fois qu'elle l'est, aucun nouveau
3628 membre ne peut s'y unir.

3629

3630 **9.3 Duplication d'objet**

3631 Les Objets-de-registre d'un registre PEUVENT être dupliqués dans un autre registre. Une copie ainsi
3632 dupliquée d'un objet distant est désignée sous l'appellation de duplicata. L'objet distant PEUT constituer
3633 un objet original ou un duplicata. Le duplicata d'un objet original est appelé duplicata de première
3634 génération. Celui d'un duplicata quelconque est dénommé duplicata de deuxième génération (et ainsi de
3635 suite).

3636 Tout registre qui duplique localement un objet distant est désigné sous l'appellation de registre de
3637 destination pour la duplication. Tout registre qui renferme l'objet distant dupliqué est appelé registre
3638 source pour la duplication.

3639

3640

3641

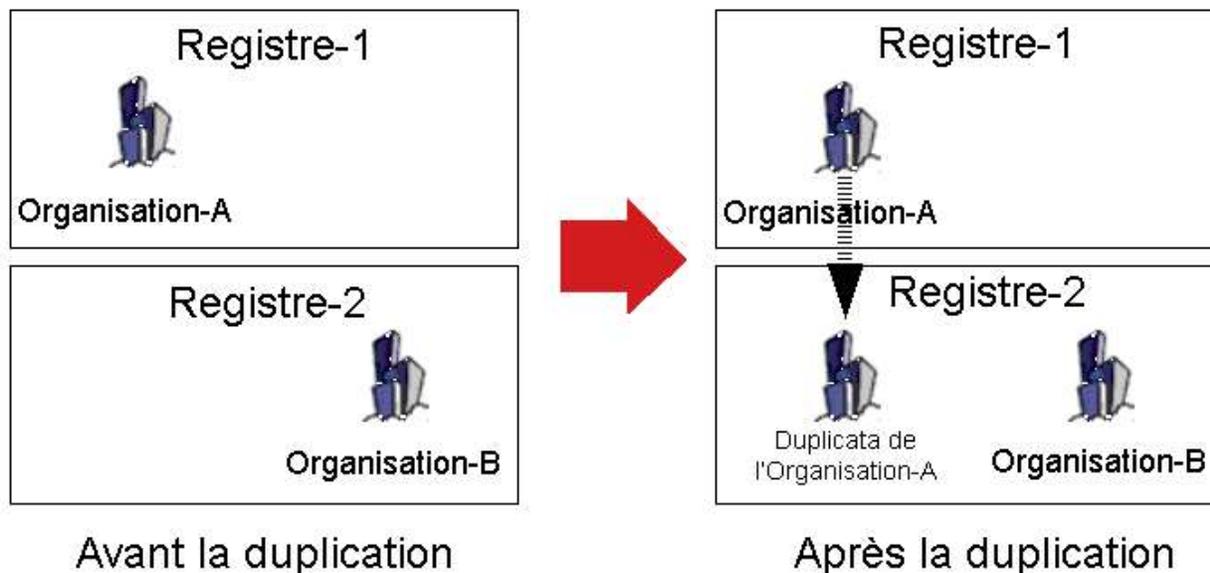


Diagramme 24 : Duplication d'objet

3642

3643

3644 9.3.1 Cas d'utilisation propres à la duplication d'objet

3645 Tout registre PEUT créer un duplicata local d'un objet distant pour diverses raisons. Voici quelques
3646 exemples de cas d'utilisation :

- 3647 ○ Améliorer le temps d'accès et la tolérance aux pannes par la mise en cache locale des objets
3648 distants. Par exemple, un registre PEUT concevoir automatiquement un duplicata local en cas
3649 d'inscription dans le registre d'une Référence-d'objet distante;
- 3650 ○ Accroître la montée en échelle par l'accessibilité aux objets très contestés, comme le schéma du
3651 SCIAN, qui se trouvent dans les multiples duplicatas;
- 3652 ○ Favoriser le recours aux caractéristiques de registres en coopération, comme la topologie
3653 hiérarchique de registre et la mise en cache locale de métadonnées de fédération.

3654 9.3.2 Interrogations et duplicatas

3655 Tout registre DOIT pouvoir accueillir les interrogations d'application de façon à tenir compte du duplicata
3656 local d'un objet distant comme si cet objet était local. Les duplicatas locaux sont pris en considération
3657 selon l'étendue de l'ensemble de données d'un registre quelconque en ce qui concerne les interrogations
3658 locales.

3659 Si une application inscrit une interrogation locale qui extrait un objet distant au moyen de son attribut
3660 identifiant, et si un registre englobe un duplicata local de cet objet, ce registre DEVRAIT renvoyer l'état
3661 défini par le duplicata.

3662 9.3.3 Opérations liées au cycle-de-vie et duplicatas

3663 Les opérations liées au cycle-de-vie qui se rattachent à un objet original DOIVENT être exécutées dans le
3664 registre d'origine de cet objet. Quant aux opérations qui touchent les duplicatas d'un objet original, elles
3665 devraient entraîner une Exception-de-demande-invalide.

3666 9.3.4 Duplication d'objet et registres fédérés

3667 La capacité de duplication d'objet est orthogonale par rapport à la capacité de fédération de registre. On
3668 PEUT dupliquer des objets de n'importe quel registre à n'importe quel autre registre sans pour autant que

3669 ces registres n'aient à faire partie de la même fédération.

3670 **9.3.5 Création d'un duplicata local**

3671 Toute organisation demanderesse peut créer un duplicata au moyen d'une Demande-d'inscription-d'objet
3672 normalisée. Si un registre quelconque reçoit une telle demande comprenant une Liste-d'objets-de-registre
3673 qui, elle, renferme une Référence-d'objet distante, il DOIT concevoir un duplicata pour celle-ci. Dans les
3674 cas de ce genre, le duplicata appartient à l'utilisateur qui a inscrit la Référence-d'objet (par une Demande-
3675 d'inscription-d'objet), alors que l'Objet-de-registre original appartient à son détenteur original.

3676 En plus des organisations demanderesse, tout registre PEUT créer un duplicata d'une manière qui lui est
3677 propre, et le faire dans des situations bien précises.

3678 Pour qu'un duplicata local soit créé, le registre de destination doit lire l'état de l'objet distant du registre
3679 source, après quoi il doit concevoir un duplicata local par rapport à cet objet.

3680 Tout registre DEVRAIT avoir recours à l'interface normalisée de Gestionnaire-d'interrogation pour lire
3681 l'état d'un objet distant (peu importe s'il s'agit d'un objet original ou d'un duplicata). On n'a besoin d'aucune
3682 nouvelle interface de programmation (API) pour la lecture d'un tel objet. La fonctionnalité d'interrogation
3683 ne nécessite aucun enregistrement au préalable; par conséquent, les registres n'ont besoin d'aucun
3684 contrat ou enregistrement au préalable pour lire l'état d'un objet distant.

3685 Dès qu'il a lu l'état de l'objet distant, un registre PEUT adopter des méthodes qui lui sont propres pour
3686 créer un duplicata local de cet objet. Ces méthodes PEUVENT englober l'utilisation de l'interface de
3687 Gestionnaire-de-cycle-de-vie.

3688 On peut distinguer tout duplicata d'un Objet-de-registre quelconque d'un objet original; en fait, l'attribut
3689 d'origine d'un duplicata DOIT pointer vers le registre distant où est stocké l'objet original du duplicata.

3690 **9.3.6 Duplication transactionnelle**

3691 La duplication transactionnelle permet à un registre de dupliquer des événements dans un autre registre
3692 et de le faire de façon systématique sur le plan transactionnel. Cette situation s'applique généralement si
3693 des registres entiers sont dupliqués dans un autre registre.

3694 La présente spécification établit un modèle de duplication davantage basée sur la souplesse
3695 d'association, à titre de solution de rechange à la duplication transactionnelle, pour les raisons suivantes :

- 3696 • La duplication transactionnelle nécessite un couplage serré entre les registres concernés;
- 3697 • La duplication transactionnelle ne constitue pas un cas d'utilisation représentatif des registres;
- 3698 • La duplication basée sur la souplesse d'association, comme le définit la présente spécification,
3699 suffit généralement pour la plupart des cas d'utilisation;
- 3700 • La duplication transactionnelle est très complexe et sujette à des erreurs.

3701

3702 Les implantations de registre ne sont pas tenues d'implanter de duplication transactionnelle.

3703 **9.3.7 Tenue à jour des duplicatas**

3704 Tout registre DOIT assurer la tenue à jour de ses duplicatas en fonction du temps d'attente établi par la
3705 valeur de l'attribut *temps-d'attente-pour-la-synchronisation-de-la-duplication*, qui, lui, est défini par le
3706 registre. Ce phénomène comprend la suppression du duplicata au moment où son objet original est
3707 supprimé de son registre d'origine.

3708 On PEUT tenir à jour les duplicatas en utilisant la caractéristique de notification d'événement du registre
3709 ou en procédant à une interrogation à des intervalles réguliers.

3710 **9.3.8 Gestion du cycle-de-vie des duplicatas locaux**

3711 Les duplicatas locaux représentent des objets en lecture seule. Les actions de gestion du cycle-de-vie ne
3712 sont pas autorisées pour les duplicatas locaux, sauf en ce qui concerne l'action de supprimer, dont on se
3713 sert pour supprimer le duplicata. Toutes les autres actions de gestion du cycle-de-vie DOIVENT être

3714 exécutées pour l'Objet-de-registre original, ce qui doit se faire dans le registre d'origine de cet objet.

3715 9.3.9 Suivi de la localisation d'un duplicata

3716 Pour tout duplicata local d'une instance d'Objet-de-registre distant, il DOIT y avoir exactement une
3717 instance de Référence-d'objet dans le registre local. L'attribut d'origine de la Référence-d'objet propre au
3718 duplicata permet de faire le suivi de la localisation de son origine. Tout Objet-de-registre DOIT
3719 comprendre précisément une origine, qui PEUT être modifiée au moyen du déplacement d'objet, comme
3720 le décrit la section 9.4. Un registre peut ou non effectuer le suivi des changements de localisation pour les
3721 duplicatas qu'il contient.

3722 9.3.10 Références d'objet distantes par rapport à un duplicata

3723 On peut disposer d'une Référence-d'objet distante pour un Objet-de-registre sous la forme de duplicata
3724 d'un autre Objet-de-registre. Dans un tel cas, l'attribut d'origine de cette Référence-d'objet renferme l'URI
3725 de base pour le registre d'origine qui s'applique au duplicata.

3726 9.3.11 Suppression d'un duplicata local

3727 Toute application peut supprimer un duplicata par le biais de la Demande-de-suppression-d'objet. Si un
3728 registre reçoit une telle demande comprenant une Liste-de-références-d'objet qui, elle, renferme une
3729 Référence-d'objet distante, il DOIT supprimer le duplicata local pour cette Référence-d'objet en
3730 présument que l'application a eu l'autorisation nécessaire pour ce faire.

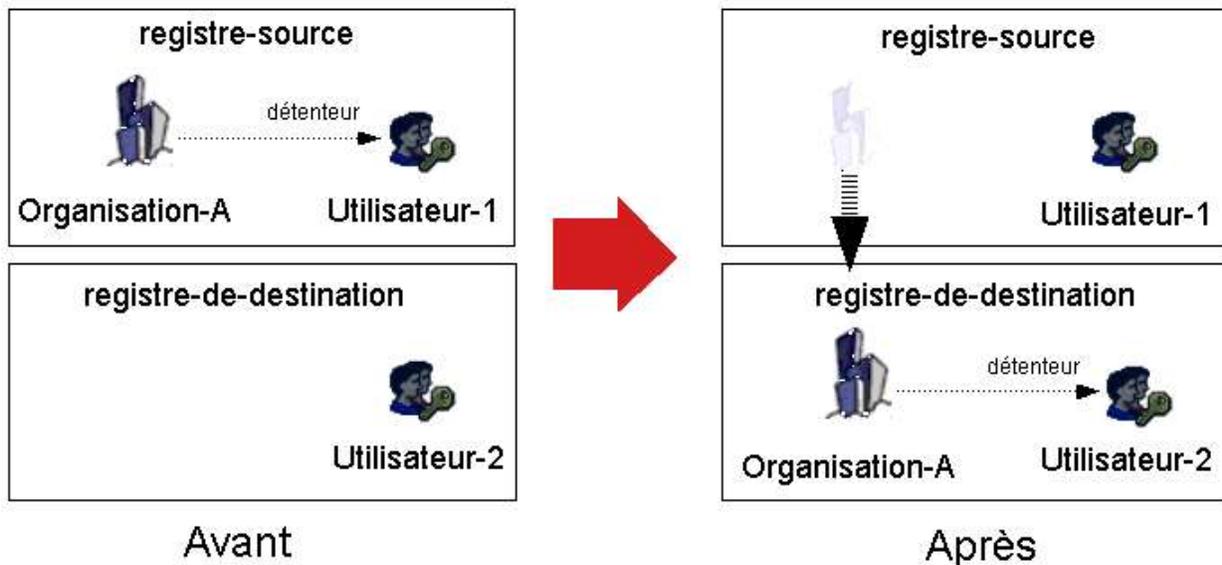
3731 9.4 Protocole de déplacement d'objet

3732 Chaque Objet-de-registre dispose d'un registre d'origine, qui, lui, englobe un utilisateur représentant le
3733 demandeur ou le détenteur de cet objet. Le registre d'origine constitue l'endroit où l'objet a été inscrit au
3734 départ, tandis que le détenteur désigne l'utilisateur qui a inscrit l'objet.

3735 Tout Objet-de-registre PEUT être déplacé d'un registre d'origine vers un autre par le biais du protocole de
3736 déplacement d'objet.

3737 Dans le cadre de ce protocole, le nouveau registre d'origine est désigné sous l'appellation de registre de
3738 *destination*, tandis que l'ancien registre d'origine est appelé registre *source*.

3739



3740

Diagramme 25 : Déplacement d'objet

3741 L'utilisateur du registre source qui détient les objets faisant l'objet d'un déplacement est dénommé

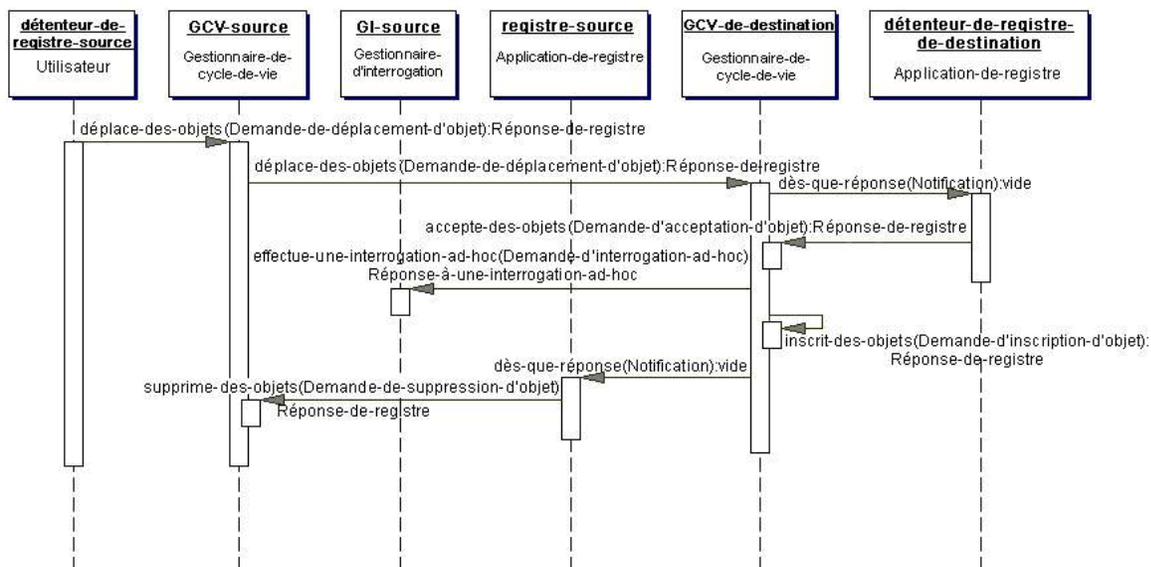
3742 *déteneur-de-registre-source*. L'utilisateur du registre de destination, qui représente le nouveau détenteur
 3743 des objets, est quant à lui appelé *déteneur-de-registre-de-destination*. Le détenteur-de-registre-source et
 3744 le détenteur-de-registre-de-destination désignent souvent le même utilisateur, mais le protocole de
 3745 déplacement d'objet les traite comme deux identités distinctes.

3746 Une méthode particulière d'utilisation du protocole de déplacement d'objet consiste à transférer la
 3747 détention d'Objets-de-registre d'un utilisateur à un autre dans le même registre. Dans un tel cas, le
 3748 protocole est le même à la différence que les registres source et de destination sont identiques.

3749 Voici certains points significatifs à propos du déplacement d'objet :

- 3750 • Le déplacement d'objet n'exige pas que les registres source et de destination fassent partie de la
 3751 même fédération ou que l'un d'eux ait un contrat préalable avec l'autre;
- 3752 • Le déplacement d'objet DOIT assurer la préservation de l'identifiant de l'objet en question. Le
 3753 registre d'origine d'un Objet-de-registre PEUT changer en raison du déplacement d'objet, mais
 3754 son identifiant est toujours le même;
- 3755 • Le Déplacement-d'objet DOIT assurer la préservation de l'intégrité référentielle des Objets-de-
 3756 registre. Les objets déplacés citant un objet qui n'a pas été déplacé DOIVENT favoriser la
 3757 préservation de leur référence. De façon analogue, les objets qui citent un objet déplacé
 3758 DOIVENT également garantir la conservation de leur référence. Ainsi, le déplacement d'un objet
 3759 peut entraîner le transfert de la valeur d'un attribut de référence; une référence locale peut
 3760 devenir une référence distante ou vice-versa;
- 3761 • La Demande-d'acceptation-d'objet ne comprend pas la Liste-de-références-d'objet. Elle n'englobe
 3762 qu'un identifiant-de-transaction non visible qui met en évidence la transaction déplace-des-objets;
- 3763 • Les demandes déterminées par le protocole de déplacement d'objet ne DOIVENT être envoyées
 3764 qu'au registre source ou au registre de destination;
- 3765 • Si un objet est déplacé, le registre-source DOIT enregistrer un Événement-vérifiable de type
 3766 « déplacé ». Dans le cadre des événements déplacés, les URI de base des registres source et de
 3767 destination DOIVENT être enregistrés en tant que Places-d'attribut par rapport à ces événements.
 3768 Voici le nom de ces Places-d'attribut :
 - 3769 o urn:oasis:names:tc:ebxml-regrep:rs:events:sourceRegistry;
 - 3770 o urn:oasis:names:tc:ebxml-regrep:rs:events:destinationRegistry.

3771



3773

Diagramme 26 : Protocole de déplacement d'objet

3774 Le Diagramme 26 met en évidence le protocole de déplacement d'objet. Les participants au protocole
3775 désignent les instances d'Utilisateur de détenteur-de-registre-source et de détenteur-de-registre-de-
3776 destination, ainsi que les interfaces de Gestionnaire-de-cycle-de-vie du registre-source et du registre-de-
3777 destination.

3778 Voici les étapes du protocole :

- 3779 1. Le protocole est entamé par le détenteur-de-registre-source, qui fait parvenir un message de
3780 Demande-de-déplacement-d'objet à l'interface de Gestionnaire-de-cycle-de-vie du registre-
3781 source. Celui-ci DOIT voir à ce que le détenteur-de-registre-source soit autorisé à transmettre
3782 cette demande. L'identifiant de la Demande-de-déplacement-d'objet est utilisé comme identifiant
3783 de transaction pour cette instance du protocole. Le message de cette demande DOIT renfermer
3784 une interrogation ad hoc établissant les objets qui doivent être déplacés;
- 3785 2. Ensuite, le registre-source DOIT transmettre le message de la Demande-de-déplacement-d'objet
3786 à l'interface de Gestionnaire-de-cycle-de-vie du registre-de-destination. Ce message assure la
3787 participation du registre-de-destination au protocole de déplacement. Le registre-de-destination
3788 DOIT stocker les renseignements demandés jusqu'à la fin du protocole ou jusqu'à la fin d'une
3789 période qui suit l'achèvement du protocole et qui est propre au registre;
- 3790 3. Le registre-de-destination DOIT transmettre le message de la Demande-de-déplacement-d'objet
3791 au détenteur-de-registre-de-destination. On PEUT procéder à cette notification au moyen de la
3792 caractéristique de notification d'événement du registre, comme l'explique le chapitre 7. On PEUT
3793 également exécuter la notification en appelant un service d'écoute pour le détenteur-de-registre-
3794 de-destination ou en faisant parvenir un message électronique à celui-ci. C'est ce qui complète la
3795 première phase du protocole de déplacement d'objet;
- 3796 4. Le détenteur-de-registre-de-destination PEUT ensuite envoyer un message de Demande-
3797 d'acceptation-d'objet au registre-de-destination. Cette demande DOIT déterminer la transaction
3798 de déplacement d'objet par le biais de l'identifiant-de-corrélation. La valeur de cet attribut DOIT
3799 constituer l'identifiant de la Demande-de-déplacement-d'objet originale;
- 3800 5. Le registre-de-destination envoie un message de Demande-d'interrogation-ad-hoc au registre-
3801 source. Celui-ci renvoie les objets déplacés sous la forme de Réponse-à-une-interrogation-ad-
3802 hoc. Un nombre élevé d'objets peut entraîner la formulation de multiples Demandes-
3803 d'interrogation-ad-hoc et Réponses-à-une-interrogation-ad-hoc, comme en fait mention la
3804 caractéristique d'interrogation itérative décrite dans la section 6.2;
- 3805 6. Le registre-de-destination inscrit les données déplacées qui établissent l'identité du détenteur-de-
3806 registre-de-destination à titre de détenteur. On PEUT inscrire ces données dans le registre de
3807 destination en se servant de n'importe quelle méthode propre au registre ou d'une Demande-
3808 d'inscription-d'objet. Les conséquences DEVRAIENT toutefois être les mêmes que si on avait
3809 recours à une Demande-d'inscription-d'objet;
- 3810 7. Le registre-de-destination informe le registre-source que les objets déplacés ont été inscrits sans
3811 encombre de façon permanente par la caractéristique de notification d'événement du registre,
3812 comme le décrit le chapitre 7;
- 3813 8. Le registre-source supprime les objets déplacés par le biais de n'importe quelle méthode propre
3814 au registre et de la consignation d'un Événement-vérifiable de type « déplacé ». C'est ce qui
3815 conclut la transaction de déplacement d'objet.

3816 9.4.1 Demande-de-déplacement-d'objet

```
3817 <element name="RelocateObjectsRequest">  
3818   <complexType>  
3819     <complexContent>  
3820       <extension base="rs:RegistryRequestType">  
3821         <sequence>  
3822           <element name="Query" type="rim:AdhocQueryType"/>  
3823           <element name="SourceRegistry" type="rim:ObjectRefType"/>  
3824           <element name="DestinationRegistry" type="rim:ObjectRefType"/>  
3825           <element name="OwnerAtSource" type="rim:ObjectRefType"/>  
3826         </sequence>  
3827       </extension>  
3828     </complexContent>  
3829   </complexType>  
3830 </element>
```

```
3826         <element name="OwnerAtDestination" type="rim:ObjectRefType"/>
3827         </sequence>
3828     </extension>
3829     </complexContent>
3830 </complexType>
3831 </element>
```

3832

3833 9.4.1.1 Paramètres

- 3834 ▪ *identifiant* : L'attribut identifiant fournit l'identifiant de transaction qui s'applique à cette
- 3835 instance du protocole.
- 3836 ▪ *Interrogation-ad-hoc* : Cet élément établit une interrogation ad hoc qui sélectionne les
- 3837 Objets-de-registre déplacés.
- 3838 ▪ *registre-source* : Cet élément désigne expressément la Référence-d'objet qui se rattache
- 3839 à l'instance de Registre de registre-source. La valeur de cet attribut DOIT se traduire par
- 3840 une référence locale si le message est envoyé au registre-source par le détenteur-de-
- 3841 registre-source.
- 3842 ▪ *registre-de-destination* : Cet élément spécifie la Référence-d'objet propre à l'instance de
- 3843 Registre de registre-de-destination.
- 3844 ▪ *détenteur-de-registre-source* : Cet élément détermine la Référence-d'objet propre à
- 3845 l'instance d'Utilisateur de détenteur-de-registre-source.
- 3846 ▪ *détenteur-de-registre-de-destination* : Cet élément précise la Référence-d'objet qui
- 3847 s'applique à l'instance d'Utilisateur de détenteur-de-registre-de-destination.
- 3848

3849 9.4.1.2 Renvois

3850 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

3851 9.4.1.3 Exceptions

3852 En plus des exceptions qui s'appliquent à l'ensemble des demandes, l'exception suivante PEUT être

3853 renvoyée :

- 3854 ▪ *Exception-liée-à-un-objet-non-trouvé* : Signifie que le Registre ou l'Utilisateur en question
- 3855 n'a pas été trouvé dans le registre.
- 3856

3857 9.4.2 Demande-d'acceptation-d'objet

```
3858 <element name="AcceptObjectsRequest">
3859     <complexType>
3860         <complexContent>
3861             <extension base="rs:RegistryRequestType">
3862                 <attribute name="correlationId" use="required" type="{http://
3863 www.w3.org/2001/XMLSchema}anyURI" />
3864             </extension>
3865         </complexContent>
3866     </complexType>
3867 </element>
```

3868

3869 9.4.2.1 Paramètres

- 3870 ▪ *identifiant-de-corrélation* : Ce paramètre fournit l'identifiant de transaction pour cette
- 3871 instance du protocole.
- 3872

3873 **9.4.2.2 Renvois**

3874 Cette demande renvoie une Réponse-de-registre. La section 2.1.4 renferme des détails à ce sujet.

3875 **9.4.2.3 Exceptions**

3876 En plus des exceptions qui s'appliquent à l'ensemble des demandes, l'exception suivante PEUT être
3877 renvoyée :

- 3878 ▪ *Exception-de-demande-invalidé* : Signifie que l'identifiant-de-corrélation spécifié ne
3879 s'apparie pas à une Demande-de-déplacement-d'objet permanente du registre.

3880

3881 **9.4.3 Déplacement d'objet et Références-d'objet distantes**

3882 Le scénario suivant décrit ce qui se produit généralement au moment où une personne passe d'un
3883 registre à un autre :

- 3884 1. Si une personne passe d'une maison à une autre, d'autres personnes peuvent avoir son ancienne
3885 adresse postale;
- 3886 2. Cette personne laisse sa nouvelle adresse à titre d'adresse d'envoi au bureau de poste;
- 3887 3. Le bureau de poste fait parvenir le courrier de la personne à sa nouvelle adresse pendant un
3888 certain temps;
- 3889 4. La demande d'acheminement du courrier expire en définitive, et le bureau de poste n'envoie plus
3890 de courrier à l'ancienne adresse de la personne;
- 3891 5. Pendant la période d'acheminement du courrier, la personne informe les parties intéressées de
3892 son changement d'adresse.

3893 La caractéristique de déplacement d'objet gère un modèle semblable de déplacement d'Objet-de-registre.
3894 Les étapes suivantes décrivent le fonctionnement prévu au moment du déplacement d'un objet.

- 3895 1. Si un Objet-de-registre O1 est déplacé d'un registre R1 à un registre R2, d'autres Objets-de-
3896 registre peuvent avoir des Références-d'objet distantes qui se rattachent à l'Objet-de-registre O1;
- 3897 2. Le registre R1 DOIT créer un Événement-vérifiable de type « déplacé » qui englobe l'URI d'origine
3898 pour le nouveau registre R2;
- 3899 3. Si le registre R1 comprend l'Événement-vérifiable et s'il reçoit une demande d'extraction de
3900 l'Objet-de-registre O1 de la part de l'identifiant, il DOIT envoyer cette demande au registre R2,
3901 extraire de manière transparente l'Objet-de-registre O1 du registre R2 et le transmettre à
3902 l'application. L'Objet-de-registre O1 DOIT renfermer l'URI d'origine par rapport au registre R2
3903 dans l'attribut d'origine optionnel de l'Objet-de-registre. On recommande aux applications de
3904 vérifier l'attribut d'origine de leur Référence-d'objet locale et de le mettre à jour de façon qu'il
3905 s'apparie à la nouvelle valeur de l'URI d'origine de l'objet;
- 3906 4. L'Événement-vérifiable est en définitive effacé après un intervalle propre au registre. Le
3907 registre R1 n'est plus tenu de transmettre des demandes liées à l'Objet-de-registre O1 au
3908 registre R2 qui est transparent pour l'application. Le registre R1 DOIT plutôt renvoyer une
3909 Exception-liée-à-un-objet-non-trouvé;
- 3910 5. Les applications qui s'intéressent au déplacement de l'Objet-de-registre O1 et veulent être
3911 informées de la nouvelle adresse de celui-ci peuvent décider d'en être avisées en ayant au
3912 préalable un abonnement qui utilise la fonctionnalité de notification d'événement du registre. Par
3913 exemple, un registre quelconque qui possède des Références-d'objet distantes par rapport à
3914 l'Objet-de-registre O1 peut concevoir un abonnement pour les événements de déplacement qui
3915 se rattachent à cet Objet-de-registre. Il ne s'agit toutefois pas d'un fonctionnement requis.

3916 **9.4.4 Notification de déplacement d'objet auprès du détenteur-de- 3917 registre-de-destination**

3918 La section 9.4.4 décrit la façon dont le registre-de-destination se sert de la caractéristique de notification

3919 d'événement du registre pour aviser le détenteur-de-registre-de-destination de la tenue d'un événement
3920 déplacé.

3921 Le registre-de-destination DOIT faire parvenir une notification en tenant compte des particularités
3922 requises suivantes :

- 3923 • La notification DOIT représenter une instance d'élément de Notification;
- 3924 • L'instance de Notification DOIT avoir au moins une Place-d'attribut, comme suit :
 - 3925 o La Place-d'attribut DOIT se nommer
 - 3926 urn:oasis:names:tc:ebxml-regrep:rs:events:correlationId.
 - 3927 o La valeur de la Place-d'attribut DOIT se traduire par l'identifiant-de-corrélation pour la
 - 3928 transaction de déplacement d'objet.
 - 3929

3930 **9.4.5 Notification d'inscription permanente d'objet auprès du registre-** 3931 **source**

3932 La section 9.4.5 décrit la façon dont le registre-de-destination se sert de la caractéristique de notification
3933 d'événement du registre pour aviser le registre-source de l'achèvement de l'inscription permanente des
3934 objets déplacés.

3935 Le registre-de-destination DOIT faire parvenir une notification en tenant compte des particularités
3936 requises suivantes :

- 3937 • La notification DOIT représenter une instance d'élément de Notification;
- 3938 • L'instance de Notification DOIT avoir au moins une Place-d'attribut, comme suit :
 - 3939 o La Place-d'attribut DOIT s'appeler
 - 3940 urn:oasis:names:tc:ebxml-regrep:rs:events:objectsCommitted.
 - 3941 o Elle DOIT comporter la valeur *VRAI*.
 - 3942

3943 **9.4.6 Détention d'objet et réaffectation de détenteur**

3944 Tout registre DOIT établir la détention d'un Objet-de-registre quelconque en fonction de l'Événement-
3945 vérifiable le plus récent dont le type-d'événement s'apparie au Nœud-de-classification de Type-
3946 d'événement canonique pour les événements créer ou déplacer.

3947 On fait face à un cas particulier de déplacement d'objet si une Demande-de-déplacement-d'objet propre à
3948 un registre spécifie le même registre à titre de registre-source et de registre-de-destination. La demande
3949 ici vise à changer le détenteur des objets déterminés de façon que le détenteur actuel en devienne un
3950 nouveau.

3951 Dans un tel cas, si l'application ne joue pas le rôle d'Administrateur-de-registre, le protocole exige que le
3952 détenteur-de-registre-de-destination présente une Demande-d'acceptation-d'objet, comme il est décrit
3953 plus haut.

3954 Si, cependant, l'application joue le rôle d'Administrateur-de-registre, le registre DOIT changer le détenteur
3955 de l'objet de sorte qu'il s'agisse de l'utilisateur désigné comme le détenteur-de-registre-de-destination,
3956 sans que celui-ci ne formule de Demande-d'acceptation-d'objet.

3957 **9.4.7 Déplacement d'objet et délais d'attente**

3958 On n'a établi aucun délai d'attente pour le protocole de déplacement d'objet. Les implantations de registre
3959 PEUVENT effacer les transactions incomplètes de déplacement d'objet d'une manière propre au registre,
3960 ce qu'elles peuvent faire à titre de tâche administrative nécessitant le recours à des règles particulières au
3961 registre.

3962

3963

10 Sécurité du registre

3964 Le chapitre 10 décrit les caractéristiques de sécurité du registre ebXML. Le [RFC 2828] comprend un
3965 glossaire des termes de sécurité. La spécification de sécurité du registre incorpore par renvoi les
3966 spécifications suivantes :

- 3967 • [WSI-BSP] WS-I Basic Security Profile 1.0;
- 3968 • [WSS-SMS] Web Services Security: SOAP Message Security 1.0;
- 3969 • [WSS-SWA] Web Services Security: SOAP Messages with Attachments (SwA) Profile 1.0.

3970 Le présent chapitre présente des détails qui sont particuliers au registre, mais qui ne figurent pas dans les
3971 spécifications ci-dessus.

10.1 Cas d'utilisation liés à la sécurité

3972 La section 10.1 décrit les divers cas d'utilisation qui nécessitent le recours à des caractéristiques de
3973 sécurité pour le registre. Les sections ultérieures décrivent, quant à elles, des mécanismes propres au
3974 registre qui favorisent l'application de chacun de ces cas d'utilisation.

10.1.1 Gestion de l'identité

3976 Une organisation quelconque déploie un registre ebXML et doit définir l'ensemble des utilisateurs et des
3977 services qui sont autorisés à se servir des services offerts par le registre. Celui-ci doit toutefois offrir un
3978 mécanisme qui permettra d'enregistrer et, ensuite, de gérer l'identité et le titre associés à ces utilisateurs
3979 et services.
3980

10.1.2 Sécurité de message

3981 Un utilisateur inscrit fait parvenir un message de demande au registre, qui lui envoie ensuite une réponse.
3982 L'utilisateur demande que l'intégrité de son message soit protégée de toute altération (attaque « man-in-
3983 the-middle ») pendant la transmission. Il peut également demander que la communication du message ne
3984 soit pas accessible aux parties non autorisées (confidentialité).
3985

10.1.3 Sécurité d'item de référentiel

3986 Un utilisateur inscrit un item de référentiel dans le registre. Il demande que celui-ci fournisse des
3987 mécanismes de protection de l'intégrité de l'item de référentiel pendant la transmission par fil et qu'il le
3988 fasse tant et aussi longtemps que l'item sera stocké dans le registre. L'utilisateur peut également
3989 demander que le contenu de l'Item-de-référentiel ne soit pas accessible aux parties non autorisées
3990 (confidentialité).
3991

10.1.4 Authentification

3992 Une organisation quelconque qui déploie un registre ebXML demande que celui-ci vérifie le titre donné
3993 par tout utilisateur inscrit qui lui envoie une demande. L'organisation veut ainsi s'assurer que cet utilisateur
3994 est bel et bien inscrit, et déterminer son identité sans ambiguïté.
3995

10.1.5 Autorisation et contrôle d'accès

3996 Une organisation quelconque qui déploie un registre ebXML demande que celui-ci offre un mécanisme
3997 qui protégera ses ressources de tout accès non autorisé. Plus précisément, si un demandeur de registre
3998 envoie une demande au registre, celui-ci limitera les actions de ce demandeur à des actions déterminées
3999 par rapport à des ressources particulières pour lesquelles le demandeur a obtenu une autorisation.
4000

10.1.6 Piste de vérification

4001 Une organisation quelconque qui déploie un registre ebXML demande que celui-ci tienne un journal ou
4002

4003 une piste de vérification pour l'ensemble des actions importantes effectuées par les demandeurs de
4004 registre par rapport à des ressources de registre. Cette particularité présente une forme fondamentale de
4005 non-répudiation, où un demandeur de registre ne peut pas nier qu'il a exécuté des actions enregistrées
4006 dans la piste de vérification.

4007 **10.2 Gestion de l'identité**

4008 Un registre ebXML DOIT fournir un mécanisme de gestion de l'identité qui favorise l'enregistrement et, par
4009 la suite, la gestion d'une identité et d'un titre pour les utilisateurs autorisés du registre.

4010 Si un registre implante le profil SAML de registre, comme le décrit le chapitre 11, la capacité de gestion de
4011 l'identité DOIT être offerte par un service de fournisseur d'identité qui s'intègre dans le registre au moyen
4012 des protocoles de SAML 2.0, tel qu'il est défini dans le [SAMLCore].

4013 Si un registre n'implante pas le profil SAML de registre, il DOIT assurer la fonctionnalité de gestion de
4014 l'identité et d'inscription de l'utilisateur, et le faire d'une manière propre à l'implantation.

4015 **10.3 Sécurité de message**

4016 Tout registre DOIT fournir des mécanismes qui favoriseront l'échange sécurisé de messages entre un
4017 demandeur de registre et le registre lui-même pour assurer l'intégrité des données et des sources,
4018 comme le décrit la présente section.

4019 **10.3.1 Sécurité de la couche de transport**

4020 Tout registre DOIT gérer la communication HTTP/S entre un demandeur HTTP et son rattachement
4021 d'interface HTTP. Tout registre DOIT également gérer cette communication entre un demandeur SOAP et
4022 son rattachement d'interface SOAP si le protocole de transport sous-jacent est le protocole HTTP.

4023 La gestion de la communication HTTP/S DEVRAIT tenir compte des protocoles de transport SSL et TLS.

4024 **10.3.2 Sécurité de message SOAP**

4025 Tout registre DOIT gérer la signature et la vérification de l'ensemble des messages de protocole de
4026 registre (demandes et réponses) entre un demandeur SOAP et son rattachement du protocole SOAP. Les
4027 mécanismes de ce genre DOIVENT être conformes au [WSI-BSP], au [WSS-SMS], au [WSS-SWA] et au
4028 [XMLDSIG]. L'utilisateur est invité à consulter ces spécifications pour avoir des renseignements sur les
4029 mécanismes de sécurité de message.

4030 **10.3.2.1 Signature de message de demande**

4031 Si un utilisateur inscrit envoie un message de demande au registre, il DEVRAIT signer ce message à
4032 l'aide d'une signature de message, ce qui assurera l'intégrité du message en plus de permettre au registre
4033 de procéder à l'authentification et à l'autorisation de la demande. Si le registre reçoit une demande qui ne
4034 comprend pas de signature de message, il DOIT la traiter de manière implicite comme si elle provenait
4035 d'un utilisateur anonyme. Un utilisateur inscrit n'est pas tenu de signer un message de demande à l'aide
4036 de la signature de message si la communication SOAP est assurée par le protocole HTTP/S; en effet, la
4037 sécurité de message est gérée ici par la sécurité de la couche de transport fournie par ce protocole.

4038 Si un utilisateur inscrit fait parvenir au registre un message de demande qui renferme un Item-de-
4039 référentiel à titre de fichier joint SOAP, il DOIT également citer et signer cet item à partir de la signature
4040 de message. Cette opération DOIT être conforme au [RFC 2392] et au [WSS-SWA].

4041 Si le registre reçoit une demande qui englobe un Item-de-référentiel non signé, il DOIT renvoyer une
4042 Exception-liée-à-un-item-de-référentiel-non-signé.

4043 **10.3.2.2 Signature de message de réponse**

4044 Si un utilisateur inscrit envoie un message de demande au registre, celui-ci PEUT utiliser une règle
4045 préétablie de préférence ou une règle par défaut pour déterminer si le message de réponse DEVRAIT
4046 être signé à l'aide d'une signature de message. Si un utilisateur anonyme du registre transmet une

4047 demande, l'autorité d'enregistrement PEUT se servir d'une règle par défaut pour évaluer si la réponse
4048 contient une signature d'en-tête. Un registre ne doit pas signer de message de réponse à l'aide d'une
4049 signature de message si la communication SOAP s'effectue par le protocole HTTP/S; en effet, la sécurité
4050 de message est gérée ici par la sécurité de la couche de transport fournie par ce protocole.

4051 Si un registre fait parvenir à une application de registre un message de réponse signé qui renferme un
4052 Item-de-référentiel à titre de fichier joint SOAP, il DOIT également citer et signer cet item à partir de la
4053 signature de message. Cette opération DOIT être conforme au [RFC 2392] et au [WSS-SWA].

4054 Si l'application de registre reçoit une réponse signée qui comprend un Item-de-référentiel, mais qui n'a
4055 aucune signature d'Item-de-référentiel, elle NE DEVRAIT PAS faire confiance à l'intégrité de la réponse et
4056 DEVRAIT traiter celle-ci comme s'il s'agissait d'une condition d'erreur.

4057 10.3.2.3 Exigences liées à l'Information-de-clé

4058 L'expéditeur d'un message de protocole de registre (demandeur de registre et registre) DEVRAIT
4059 présenter sa clé publique selon l'élément <wsse:Security>. Si cette clé est fournie, elle DOIT faire partie
4060 d'un élément <wsse:BinarySecurityToken> et DOIT être citée à partir de l'élément <ds:KeyInfo> dans la
4061 signature de message. La valeur de l'attribut wsu:Id de l'élément <wsse:BinarySecurityToken> qui
4062 renferme la clé publique de l'expéditeur DOIT être la suivante : **urn:oasis:names:tc:ebxml-**
4063 **regrep:rs:security:SenderCert**. L'élément <wsse:BinarySecurityToken> DEVRAIT comprendre
4064 un certificat X509.

4065 La Liste 3 présente un exemple de signature de message, où, entre autres, l'Information-de-clé est
4066 spécifiée.

4067 10.3.2.4 Validation de signature de message

4068 La validation de signature assure l'intégrité et la sécurité de message et d'Item-de-référentiel connexe, ce
4069 qui touche tant les données que les sources.

4070 Si le registre reçoit une demande qui renferme une signature de message, il DOIT valider cette signature,
4071 comme le définit le [WSS-SMS]. Si la demande englobe un Item-de-référentiel connexe, elle DOIT en
4072 valider la signature, comme en fait mention le [WSS-SWA].

4073 Si le demandeur de registre reçoit une réponse qui comprend une signature de message, il DEVRAIT
4074 valider cette signature, comme le précise le [WSS-SMS]. Si la réponse comporte un Item-de-référentiel
4075 connexe, elle DEVRAIT en valider la signature, conformément au [WSS-SWA].

4076 10.3.2.5 Exemple de signature de message

4077 L'exemple suivant illustre le format d'une signature de message :

```
4078 <soap:Envelope>  
4079 <soap:Header>  
4080 <wsse:Security>  
4081 <wsse:BinarySecurityToken EncodingType="http://docs.oasis-  
4082 open.org/wss/2004/01/oasis-200401-wss-soap-message-security-  
4083 1.0#Base64Binary" ValueType="http://docs.oasis-  
4084 open.org/wss/2004/01/oasis-200401-wss-token-profile-1.0#X509v3"  
4085 wsu:Id="urn:oasis:names:tc:ebxml-regrep:rs:security:SenderCert">  
4086 lui+Jy4WYKGGJW5xM3aHnLxOpGVIpzSg4V486hHFe7sHET/uxxVBovT7JV1A2RnWSW  
4087 kXm9jAEdsm/  
4088 hs+f3NwvK23bh46mNmnCQVsUYHbYAREZpykrd/eRwNgx8T+ByeFhmSviW77n6yTcI  
4089 7XU7xZT54S9  
4090 hTSyBLN2Sce1dEQpQXh5ssZK9aZTMrSFT1NBvNHC3Qq7w0Otr5V4axH3MXffsuI9W  
4091 zxPCfHdalN4  
4092 rLRfNY318pc6bn00zAMw0omUWwBEJZxxBGGUc9QY3VjwNALgGDaEAT7gpURkCI85H  
4093 jdnSA5SM4cY  
4094 7jAsYX/CIpEkRJcBUL1TEFrBZIBYDPzRW1SdsJRJngF7yCoGwJ+/HYOyP8P4OM59F  
4095 Di0kM8GwOE0  
4096 WgYrJHH92qaVhoiPTLi7  
4097 </wsse:BinarySecurityToken>  
4098 <ds:Signature>  
4099 <!--The Message Signature -->
```

```

4100     <ds:SignedInfo>
4101         <ds:CanonicalizationMethod
4102 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
4103             <c14n:InclusiveNamespaces PrefixList="wsse soap"
4104 xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
4105         </ds:CanonicalizationMethod>
4106         <ds:SignatureMethod
4107 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
4108         <ds:Reference URI="#TheBody">
4109             <ds:Transforms>
4110                 <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
4111 c14n#">
4112                     <c14n:InclusiveNamespaces PrefixList=""
4113 xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
4114                 </ds:Transform>
4115             </ds:Transforms>
4116             <ds:DigestMethod
4117 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
4118             <ds:DigestValue>i3qi5GjhHnfoBn/jOjQp2mq0Na4=</ds:DigestValue>
4119             </ds:Reference>
4120         </ds:SignedInfo>
4121         <ds:SignatureValue>PipXJ2Sfc+LTDnq4pM5JcIYt9gg=</ds:SignatureValu
4122 e>
4123         <ds:KeyInfo>
4124             <wsse:SecurityTokenReference>
4125                 <wsse:Reference URI="#urn:oasis:names:tc:ebxml-
4126 regrep:rs:security:SenderCert" ValueType="http://docs.oasis-
4127 open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
4128             </wsse:SecurityTokenReference>
4129         </ds:KeyInfo>
4130     </ds:Signature>
4131 </wsse:Security>
4132 </soap:Header>
4133 <soap:Body wsu:Id="TheBody">
4134     <lcm:SubmitObjectsRequest />
4135 </soap:Body>
4136 </soap:Envelope>

```

4137 **Liste 3 : Exemple de signature de message**

4138 **10.3.2.6 Message comprenant un Item-de-référentiel : exemple de signature**

4139 L'exemple suivant illustre le format d'une signature de message qui signe également l'Item-de-référentiel
4140 qui y est joint :

```

4141
4142 Content-Type: multipart/related; boundary="BoundaryStr" type="text/xml"
4143 --BoundaryStr
4144 Content-Type: text/xml
4145 <soap:Envelope>
4146     <soap:Header>
4147         <wsse:Security>
4148             <wsse:BinarySecurityToken EncodingType="http://docs.oasis-
4149 open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
4150 1.0#Base64Binary" ValueType="http://docs.oasis-
4151 open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
4152 wsu:Id="urn:oasis:names:tc:ebxml-regrep:rs:security:SenderCert">
4153                 lui+Jy4WYKGGJW5xM3aHnLxOpGVIpzSg4V486hHFe7sHET/uxxVBovT7JV1A2RnWSW
4154                 kXm9jAEdsm/
4155                 hs+f3NwvK23bh46mNmCQVsUYHbYAREZpykrd/eRwNgx8T+ByeFhmSviW77n6yTcI
4156                 7XU7xZT54S9
4157                 hTSyBLN2Sce1dEqpQXh5ssZK9aZTMrSFT1NBvNHC3Qq7w00tr5V4axH3MXffsuI9W
4158                 zxPCfHdalN4
4159                 rLRfNY318pc6bn00zAMw0omUWwBEJZxxBGGUc9QY3VjwNALgGDaEAT7gpURkCI85H
4160                 jdnSA5SM4cY

```

```

4161         7jAsYX/CiPEkRJcBULLTEFrBZIBYDPzRWlSdsJRJngF7yCoGWJ+/HYOyP8P4OM59F
4162 Di0kM8GwOE0
4163         WgYrJHH92qaVhoiPTLi7
4164         </wsse:BinarySecurityToken>
4165         <ds:Signature>
4166             <!-- The Message Signature -->
4167             <ds:SignedInfo>
4168                 <ds:CanonicalizationMethod
4169 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
4170                 <cl4n:InclusiveNamespaces PrefixList="wsse soap"
4171 xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
4172                 </ds:CanonicalizationMethod>
4173                 <ds:SignatureMethod
4174 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
4175                 <ds:Reference URI="#TheBody">
4176                     <ds:Transforms>
4177                         <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
4178 c14n#">
4179                             <cl4n:InclusiveNamespaces PrefixList=""
4180 xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
4181                             </ds:Transform>
4182                         </ds:Transforms>
4183                         <ds:DigestMethod
4184 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
4185                         <ds:DigestValue>i3qi5GjhHnfoBn/jOjQp2mq0Na4=</ds:DigestValue>
4186                         </ds:Reference>
4187                     </ds:SignedInfo>
4188
4189             <!--A reference to a RepositoryItem (one for each RepositoryItem)
4190 -->
4191             <ds:SignedInfo>
4192                 <ds:CanonicalizationMethod
4193 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
4194                 <cl4n:InclusiveNamespaces PrefixList="wsse soap"
4195 xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
4196                 </ds:CanonicalizationMethod>
4197                 <ds:SignatureMethod
4198 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
4199                 <ds:Reference URI="cid:${REPOSITORY_ITEM1_ID}">
4200                     <ds:Transforms>
4201                         <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
4202 c14n#">
4203                             <ds:Transform Algorithm="http://docs.oasis-
4204 open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-Content-
4205 Only-Transform" />
4206                             </ds:Transform>
4207                         </ds:Transforms>
4208                         <ds:DigestMethod
4209 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
4210                         <ds:DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</ds:DigestValue>
4211                         </ds:Reference>
4212                     </ds:SignedInfo>
4213
4214                 <ds:SignatureValue>PipXJ2Sfc+LTDnq4pM5JcIYt9gg=</ds:SignatureValu
4215 e>
4216
4217                 <ds:KeyInfo>
4218                     <wsse:SecurityTokenReference>
4219                         <wsse:Reference URI="#urn:oasis:names:tc:ebxml-
4220 regrep:rs:security:SenderCert" ValueType="http://docs.oasis-
4221 open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
4222                         </wsse:SecurityTokenReference>
4223                     </ds:KeyInfo>
4224
4225                 </ds:Signature>
4226             </wsse:Security>

```

```

4227     </soap:Header>
4228     <soap:Body wsu:Id="TheBody">
4229         <lcm:SubmitObjectsRequest/>
4230     </soap:Body>
4231 </soap:Envelope>
4232 --BoundaryStr
4233 Content-Type: image/png
4234 Content-ID: <${REPOSITORY_ITEM1_ID}>
4235 Content-Transfer-Encoding: base64
4236 the repository item (e.g. PNG Image) goes here..

```

Liste 4 : Exemple de signature d'Item-de-référentiel

4238 10.3.2.7 Sécurité de message SOAP et protocole HTTP/S

4239 Si on utilise le protocole HTTP/S entre une application de registre et un registre, on NE DOIT PAS tenir
4240 compte de la sécurité de message SOAP. En voici les règles :

- 4241 • L'application de registre NE DOIT PAS signer le message de demande ou un item de référentiel
4242 quelconque qui se trouverait dans la demande;
- 4243 • Le registre NE DOIT PAS vérifier les signatures de demande ou d'Item-de-référentiel;
- 4244 • Le registre NE DOIT PAS signer le message de réponse ou un item de référentiel quelconque qui se
4245 trouverait dans la réponse;
- 4246 • L'application de registre NE DOIT PAS vérifier les signatures de réponse ou d'Item-de-référentiel.

4247 10.3.3 Confidentialité des messages

4248 Tout registre DEVRAIT pouvoir accueillir le chiffrement des messages de protocole, comme le définit la
4249 section 9 du [WSI-BSP], et s'en servir comme mécanisme de gestion de la confidentialité des messages
4250 de protocole pendant leur transmission sur fil.

4251 Toute application de registre PEUT avoir recours au chiffrement des Items-de-référentiel, comme le
4252 mentionne le [WSS-SWA], et s'en servir comme mécanisme de gestion de la confidentialité des Items-de-
4253 référentiel pendant leur transmission sur fil.

4254 Tout registre DEVRAIT gérer l'inscription des items de référentiel chiffrés.

4255 10.3.4 Exigences liées à la distribution des clés publiques

4256 Le registre et les utilisateurs inscrits DOIVENT s'échanger leurs clés publiques. Cette mesure est
4257 nécessaire pour l'exécution des étapes suivantes :

- 4258 • Authentification mutuelle de l'application de registre et du registre par l'établissement d'une liaison
4259 entre les protocoles SSL et TLS en vue de la sécurité de la couche de transport dans le
4260 protocole HTTP/S;
- 4261 • Validation des signatures de message et d'Item-de-référentiel (ce qui est décrit dans la section);
- 4262 • Déchiffrement des messages chiffrés.

4263 Pour favoriser la sécurité de message, on DOIT respecter les exigences suivantes :

- 4264 1. Un certificat se rattache au registre;
- 4265 2. Un certificat est associé à l'application de registre;
- 4266 3. Une application de registre inscrit son certificat de clé publique dans le registre, ce qui est
4267 généralement fait pendant l'inscription de l'utilisateur et d'une manière propre à l'implantation;
- 4268 4. L'application de registre obtient le certificat de clé publique du registre et le stocke dans sa propre
4269 mémoire locale de clé, ce qui est fait d'une manière propre à l'implantation.

4270

4271 **10.4 Authentification**

4272 Tout registre DOIT pouvoir authentifier l'identité de l'utilisateur associé aux demandes d'application afin de
4273 procéder à l'autorisation et au contrôle d'accès, et de maintenir une piste de vérification de l'accès au
4274 registre. Sur le plan de la sécurité, un service qui permet d'authentifier les demandeurs est désigné sous
4275 l'appellation d'autorité d'authentification.

4276 Tout registre DOIT fournir au moins l'un des mécanismes d'authentification suivants :

- 4277 • Registre à titre d'autorité d'authentification;
- 4278 • Autorité d'authentification externe.

4279

4280 **10.4.1 Registre à titre d'autorité d'authentification**

4281 Tout registre PEUT assurer une capacité d'authentification en jouant le rôle d'autorité d'authentification. À
4282 ce titre, il se sert de l'élément <ds:KeyInfo> dans la signature de message comme titre dans le but
4283 d'authentifier le demandeur en question. Cette fonction nécessite généralement qu'on veille à ce que la
4284 clé publique présentée dans l'élément <ds:KeyInfo> de la signature de message s'apparie à celle d'un
4285 utilisateur inscrit quelconque. On doit également voir à ce que le registre garde un « stockage de clés de
4286 registre » renfermant les clés publiques des utilisateurs inscrits. Les autres détails relatifs au registre à
4287 titre d'autorité d'authentification sont propres à l'implantation.

4288 De façon subsidiaire, si l'application de registre communique avec le registre par le protocole HTTP/S, le
4289 registre DOIT authentifier l'utilisateur d'application de registre si un certificat enregistré est émis au moyen
4290 de l'authentification d'application du protocole SSL. Si le certificat est inconnu pour le registre, celui-ci
4291 DOIT attribuer le mandant de l'Utilisateur-anonyme-de-registre à l'application de registre.

4292 **10.4.2 Autorité d'authentification externe**

4293 Tout registre PEUT également avoir recours à une autorité d'authentification externe pour authentifier les
4294 demandes d'application. Pour utiliser une telle autorité, on doit s'assurer que le registre implante le
4295 profil SAML de registre, comme le décrit le chapitre 11.

4296 **10.4.3 Gestion de session authentifiée**

4297 Aussitôt qu'une demande est authentifiée, tout registre DEVRAIT créer une session authentifiée au
4298 moyen de méthodes propres à l'implantation pour éviter de devoir authentifier de nouveau les demandes
4299 ultérieures transmises par le même demandeur. Si le protocole de transport sous-jacent se traduit par le
4300 protocole HTTP, tout registre DEVRAIT implanter une méthode de gestion de session authentifiée à partir
4301 de la capacité de session HTTP, comme le définit le [RFC 2965].

4302 **10.5 Autorisation et contrôle d'accès**

4303 Dès qu'un registre a authentifié l'identité de l'utilisateur inscrit par rapport à une demande d'application
4304 quelconque, il DOIT procéder à l'autorisation de cet utilisateur et, ensuite, appliquer des règles de contrôle
4305 d'accès en fonction de la décision prise en matière d'autorisation.

4306 L'autorisation et le contrôle d'accès sont exécutés par le registre, qui décide QUI peut effectuer QUELLE
4307 ACTION par rapport à QUELLE RESSOURCE.

- 4308 • QUI : Désigne l'utilisateur défini pendant l'authentification.
- 4309 • QUELLE ACTION : Est établi par la demande de protocole de registre transmise par l'application.
- 4310 • QUELLE RESSOURCE : Regroupe les Objets-de-registre et les Items-de-référentiel sur lesquels
4311 la demande de protocole de registre a des répercussions.

4312 L'autorisation et le contrôle d'accès sont déterminés par la règle de contrôle d'accès qui se rattache à la
4313 ressource sur laquelle l'action exerce un impact.

4314 Tout registre DOIT présenter un mécanisme d'autorisation et de contrôle d'accès qui s'inspire du chapitre

4315 « Modèle d'information de contrôle d'accès » du [ebRIM]. Le modèle d'information de contrôle d'accès
4316 définit une règle de contrôle d'accès par défaut qui DOIT être gérée par le registre. Il établit également un
4317 rattachement au [XACML] qui favorise l'élaboration de règles fines de contrôle d'accès.

4318 **10.6 Piste de vérification**

4319 Une fois l'autorisation vérifiée, tout registre doit appliquer les règles de contrôle d'accès et, le cas échéant,
4320 approuver la demande d'une application qui permettra à celle-ci d'utiliser les services qu'elle a demandés.
4321 Un registre DOIT créer une piste de vérification pour l'ensemble des opérations de Gestionnaire-de-cycle-
4322 de-vie. Il PEUT faire de même pour les opérations de Gestionnaire-d'interrogation. Pour conserver les
4323 ressources de stockage, il PEUT élaguer l'information liée à la piste de vérification qu'il stocke d'une
4324 manière propre à l'implantation. Le registre DEVRAIT procéder à un tel élagage en supprimant les
4325 renseignements désuets du contenu relatif à la piste de vérification. Il NE DOIT toutefois PAS supprimer
4326 l'option originale de création d'événement au début de la piste de vérification puisque cette option
4327 détermine l'identité du détenteur de l'Objet-de-registre.

4328 Le chapitre « Modèle d'information Événement » du [ebRIM] renferme des détails sur la façon dont un
4329 registre conserve une piste de vérification des demandes d'application.

4330

11 Profil SAML de registre

4331 Le chapitre 11 définit le profil SAML de registre que tout registre PEUT implanter pour gérer les protocoles
 4332 SAML 2.0 établis par le [SAMLCore]. Le profil SAML de registre privilégie particulièrement le profil de
 4333 signature unique (SSO) Internet, qui est spécifié dans le [SAMLProf].

11.1 Terminologie

4335 L'utilisateur est invité à consulter le glossaire SAML [SAMLGloss], où il trouvera les divers termes utilisés
 4336 dans le profil SAML de registre. En voici d'ailleurs quelques-uns à des fins de convivialité :

4337

Terme	Définition
Autorité d'authentification	Une autorité d'authentification représente une entité de système (généralement un service) qui permet à d'autres entités de systèmes (habituellement des utilisateurs ou des services) de créer une session authentifiée en prouvant leur identité, ce qu'elles font en transmettant les titres nécessaires (par exemple, le nom d'utilisateur et le mot de passe ou bien le pseudonyme de certificat et le mot de passe). Une autorité d'authentification produit des assertions d'authentification à la suite d'une authentification fructueuse.
Client/proxy	Le client/proxy décrit une application qui fonctionne selon certaines contraintes et qui, par conséquent, ne peut pas gérer le protocole de redirection HTTP. De façon générale, il s'agit d'applications qui ne disposent pas d'un environnement de navigateur Web. Dans le présent document, le meilleur exemple de client/proxy se traduit par une application de registre qui se sert du protocole SOAP pour communiquer avec le registre (demandeur SOAP).
Fournisseur d'identité	Il s'agit d'un type de <i>fournisseur de service</i> qui crée, tient et gère les renseignements sur l'identité des <i>utilisateurs authentifiés</i> (par exemple, les <i>utilisateurs</i>). Un fournisseur de service constitue généralement aussi une autorité d'authentification.
Utilisateur authentifié	Il s'agit d'une entité de système dont l'identité peut être authentifiée. Il s'apparie à l'utilisateur spécifié dans le [ebRIM].
Demandeur SAML	Il s'agit d'une <i>entité de système</i> qui se sert du protocole SAML pour demander des services d'une autre entité de système (une <i>autorité SAML</i> ou un <i>répondeur</i>). On n'a pas recours au terme « application » pour cette notion puisque de nombreuses entités de système agissent simultanément ou en série comme applications et serveurs.
Fournisseur de service	Il s'agit d'un rôle joué par une entité de système dans le cadre duquel elle fournit des services à des utilisateurs authentifiés ou à d'autres entités de système. Le service de registre désigne expressément un fournisseur de service.
Accès unifié	Il s'agit de la capacité de partager une session authentifiée unique dans des applications et services multiples activés par l'accès unifié. Une application peut créer une session authentifiée en étant identifiée à l'aide de toute autorité d'authentification dans le système. L'application peut ensuite effectuer n'importe quelle opération sécurisée au moyen de n'importe quel service activé par l'accès unifié dans le système. Pour ce faire, l'application doit avoir recours à la session authentifiée.
Fermeture unifiée de session	Il s'agit de la capacité de fermer simultanément une session de multiples fournisseurs de service dans un système fédéré.

4338

11.2 Cas d'utilisation liés au profil SAML

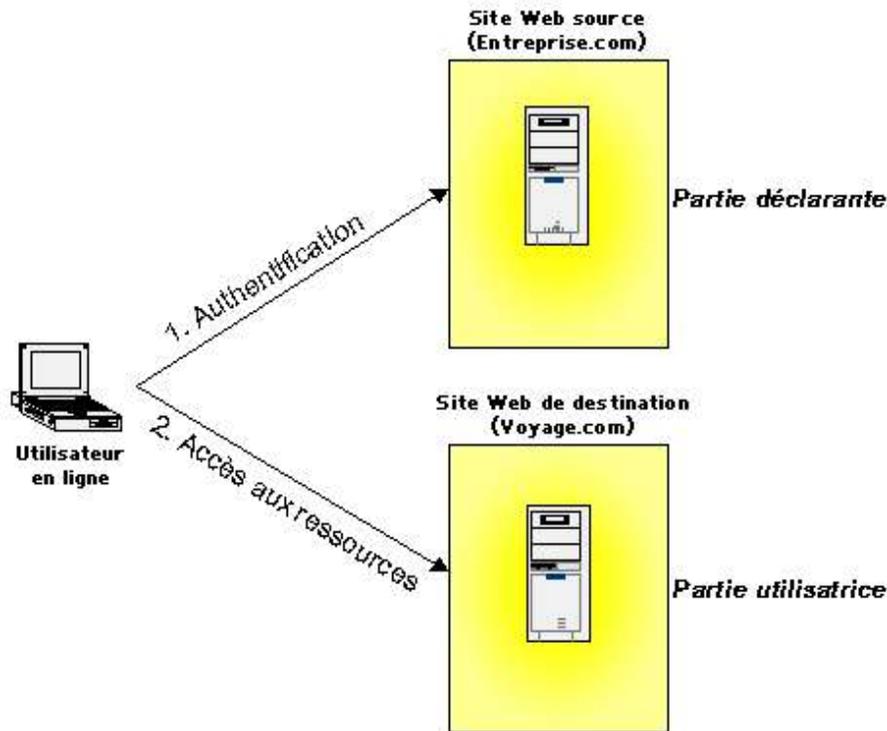
4339 Le profil SAML de registre a comme objectif d'aborder les cas d'utilisation ci-dessous en se servant des
 4340 protocoles définis par le [SAMLCore].
 4341

4342 **11.2.1 Registre à titre de participant au protocole d'accès unifié**

4343 Une grande entreprise déploie un registre ebXML. Elle dispose déjà d'un fournisseur d'identité (par
4344 exemple, un service de gestionnaire d'accès) où elle préserve le titre et de l'information sur les
4345 utilisateurs. Elle possède également déjà une autorité d'authentification (qui peut constituer le même
4346 service que le fournisseur d'identité), dont elle se sert pour authentifier les utilisateurs et activer l'accès
4347 unifié dans l'ensemble de ses applications de service.

4348 L'entreprise souhaite utiliser son fournisseur d'identité pour gérer les utilisateurs de registre et éviter toute
4349 duplication de la base de données sur les utilisateurs qui se trouve dans le fournisseur d'identité du
4350 registre. L'entreprise désire aussi avoir recours à son autorité d'authentification pour authentifier les
4351 utilisateurs de registre. Enfin, elle s'attend à ce que le registre soit pris en considération dans la capacité
4352 du protocole d'accès unifié qui provient de son service d'autorité d'authentification.

4353



4355 **Diagramme 27 : Exemple représentatif de protocole d'accès unifié SAML**

4356 **11.3 Rôle du registre par rapport aux protocoles SAML**

4357 Pour être en accord avec le profil SAML de registre, tout registre ebXML doit jouer le rôle de fournisseur
4358 de service qui s'inspire de sa conformité à l'égard des protocoles SAML 2.0.

4359 **11.3.1 Rôle de fournisseur de service**

4360 En jouant le rôle de fournisseur de service, le registre peut participer aux protocoles SAML. De manière
4361 précise, il peut ainsi utiliser un fournisseur d'identité pour procéder en son nom à l'authentification des
4362 applications.

4363 **11.3.1.1 Exigences liées au rôle de fournisseur de service**

4364 Voici une liste d'exigences qui se rattachent au rôle de fournisseur de service joué par le registre :

- 4365 • Le registre DOIT pouvoir accueillir les protocoles, les messages et les rattachements dont le
4366 fournisseur de service est responsable, comme le définit le profil d'accès unifié Web [SAMLProf].

- 4367 Plus précisément, il DOIT pouvoir lancer un protocole de demande d'authentification et y
4368 participer à l'aide d'un fournisseur d'identité;
- 4369 • Le registre DOIT pouvoir se servir d'un fournisseur d'identité SAML pour authentifier les
4370 demandes d'application;
 - 4371 • Il DOIT pouvoir assurer un contexte de sécurité pour les applications de registre par rapport aux
4372 multiples demandes d'application.
4373

4374 11.4 Interface SAML de registre

4375 Pour être conforme au profil SAML de registre, tout registre ebXML DOIT implanter une nouvelle
4376 interface SAML en plus de ses interfaces de services, comme le Gestionnaire-d'interrogation et le
4377 Gestionnaire-de-cycle-de-vie.

4378 Les détails propres à l'interface SAML de registre ne figurent pas dans la présente spécification. Ils se
4379 trouvent plutôt dans les spécifications SAML 2.0 et DOIVENT pouvoir accueillir les demandes SOAP et
4380 SAML sur HTTP.

4381 Tout registre se sert de son interface SAML pour participer aux protocoles SAML, comme le font les
4382 applications SAML et les fournisseurs d'identité SAML. De manière précise, tout Fournisseur-d'identité a
4383 recours à l'interface de fournisseur de service SAML du registre pour transmettre une réponse à une
4384 demande d'authentification.

4385 11.5 Exigences liées au profil SAML de registre

4386 Pour être conforme au profil SAML de registre, tout registre DOIT implanter un protocole SAML qui pourra
4387 accueillir des échanges de messages particuliers de protocole SAML par le biais de rattachements
4388 déterminés de protocole.

4389 Le Tableau 7 présente la grille des profils SAML, des messages de protocole et de leurs rattachements
4390 que tout registre DOIT pouvoir accueillir pour être conforme au profil SAML de registre.

4391 L'utilisateur peut consulter :

- 4392 • le [SAMLProf] pour avoir une description des profils énumérés;
- 4393 • le [SAMLCore] pour avoir une description des flux de messages énumérés;
- 4394 • le [SAMLBind] pour avoir une description des rattachements énumérés.
4395

Profil	Flux de messages	Rattachement	Exigence relative à l'implantation
Accès unifié Web	<Demande-d'authentification> du Registre au Fournisseur-d'identité	redirection HTTP	DOIT
	<Réponse> du Fournisseur-d'identité au registre	HTTP-POST	DOIT
		artefact HTTP	DOIT
Fermeture unifiée de session	<Demande-de-fermeture-de-session>	redirection HTTP	DOIT
		SOAP	PEUT
	<Réponse-de-fermeture-de-session>	redirection HTTP	DOIT
		SOAP	PEUT

Profil	Flux de messages	Rattachement	Exigence relative à l'implantation
Résolution d'artefact	<Résolution-d'artefact>	SOAP	DOIT
	<Réponse-d'artefact>	SOAP	DOIT
Accès unifié client/proxy	Client/proxy au registre; registre au client/proxy au Fournisseur-d'identité	PAOS	DOIT
	Fournisseur-d'identité au client/proxy au registre; registre au client/proxy	PAOS	DOIT

4396

4397

Tableau 7 : Profils SAML, protocoles et rattachements nécessaires

4398 **11.6 Opération d'accès unifié**

4399 La section 11.6 décrit la séquence d'interaction qui s'applique aux divers types d'opération d'accès unifié.

4400 **11.6.1 Acteurs de scénario**

4401 Voici les acteurs qui figureront dans les différents scénarios d'opération d'accès unifié décrits dans la
4402 section ultérieure :

- 4403 • Demandeur HTTP : Cet acteur représente une application de registre qui accède au registre au
4404 moyen du rattachement de protocole HTTP des protocoles de registre, et qui le fait généralement
4405 en se servant d'un agent d'utilisateur, comme un navigateur Web;
- 4406 • Demandeur SOAP : Cet acteur désigne une application de registre qui accède au registre par le
4407 biais du rattachement de protocole SOAP des protocoles de registre;
- 4408 • Registre : Cet acteur, qui constitue un registre, renferme l'ensemble des interfaces de Registre
4409 comme le Gestionnaire-d'interrogation, le Gestionnaire-de-cycle-de-vie et le fournisseur de
4410 service SAML de registre. Le registre joue un rôle dans les protocoles de registre ebXML ainsi
4411 que dans les protocoles SAML;
- 4412 • Fournisseur-d'identité : Cet acteur se traduit par le Fournisseur-d'identité que le registre utilise
4413 pour procéder aux activités d'authentification en son nom.

4414 **11.6.2 Opération d'accès unifié – demandeur HTTP non authentifié**

4415 Le Diagramme 28 présente une vue d'ensemble de l'opération d'accès unifié qui doit être effectuée si le
4416 demandeur SOAP n'est pas authentifié et s'il accède au registre HTTP au moyen d'un agent d'utilisateur,
4417 comme un navigateur Web.

4418

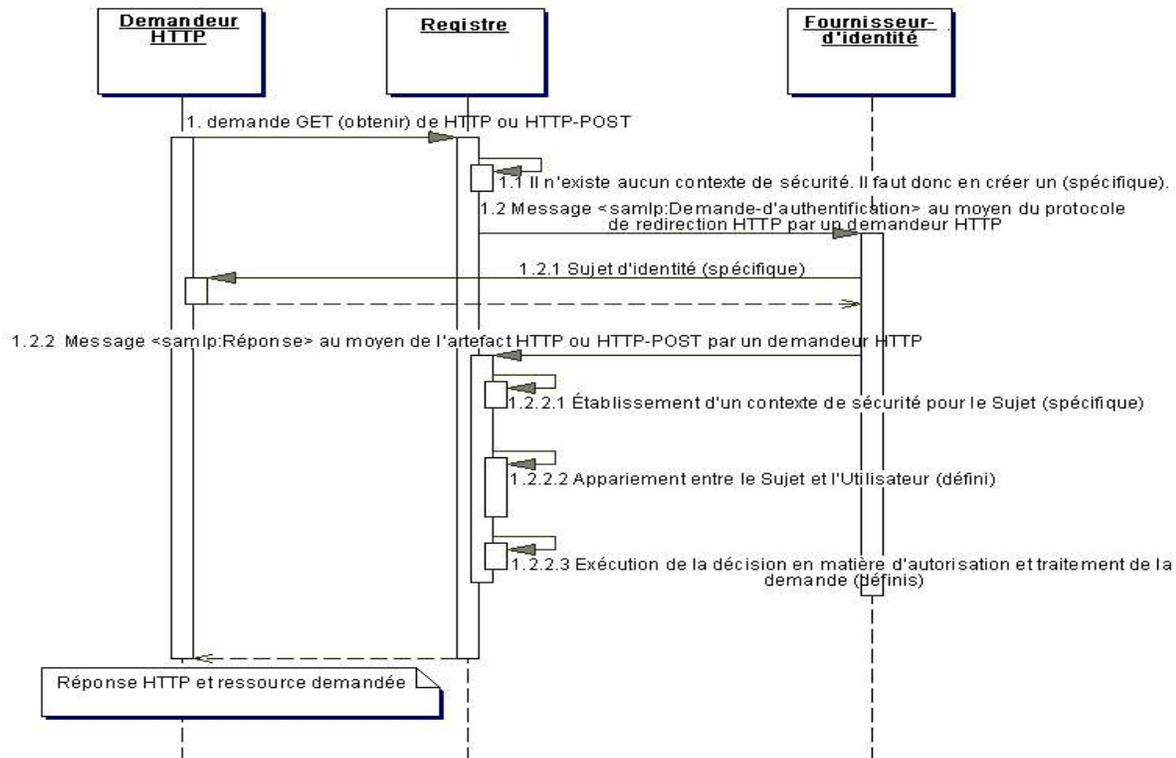
4419

Diagramme 28 : Opération d'accès unifié – demandeur HTTP non authentifié

4420 **11.6.2.1 Séquence de scénario**

4421 Le Diagramme 28 illustre les étapes ci-dessous qui se rattachent à l'opération d'accès unifié.

- 4422 1 Le demandeur HTTP transmet une demande GET (obtenir) de HTTP ou une demande HTTP-POST
4423 à une interface de registre quelconque, comme le Gestionnaire-d'interrogation ou le Gestionnaire-de-
4424 cycle-de-vie.
- 4425 1.1 Le registre vérifie s'il dispose déjà d'un contexte de sécurité qui a été établi pour le sujet associé à
4426 la demande en question. Il s'assure qu'aucun contexte de sécurité n'a été mis en place
4427 antérieurement.



- 4428 1.2 Pour créer un contexte de sécurité, le registre exécute le protocole <samlp:Demande-
 4429 d'authentification> à l'aide du Fournisseur-d'identité. La <Demande-d'authentification> est envoyée
 4430 au moyen du protocole de redirection HTTP par l'agent d'utilisateur (par exemple, un
 4431 navigateur Web) dont se sert le demandeur HTTP.
- 4432 1.2.1 Le Fournisseur-d'identité a recours à des méthodes propres à l'implantation pour définir le sujet.
 4433 Pour ce faire, il doit généralement communiquer avec l'agent d'utilisateur utilisé par le
 4434 demandeur HTTP pour obtenir le titre lié au sujet, après quoi il doit se servir de ce titre pour
 4435 certifier qu'il connaît le sujet. Si la communication est axée sur les protocoles SSL et TLS,
 4436 l'acquisition du titre s'effectue sans la moindre intervention de l'utilisateur et se fait directement
 4437 auprès de l'agent d'utilisateur. On émet l'hypothèse que le Fournisseur-d'identité peut authentifier
 4438 le sujet.
- 4439 1.2.2 Le Fournisseur-d'identité transmet au registre un message <samlp:Réponse> renfermant un
 4440 élément <saml:Déclaration-d'authentification>, ce qu'il fait en utilisant un rattachement SAML
 4441 d'artefact HTTP ou HTTP-POST par le biais de l'agent d'utilisateur.
- 4442 1.2.2.1 Le registre utilise des méthodes propres à l'implantation pour établir un contexte de sécurité par
 4443 rapport au sujet authentifié par le Fournisseur-d'identité. Il agit ainsi en se fondant sur
 4444 l'information qui concerne le sujet et qui est présentée dans le message <samlp:Réponse>. Ces
 4445 mesures peuvent comprendre la création d'une session HTTP pour le demandeur HTTP.
- 4446 1.2.2.2 Le registre apparie l'information sur le sujet qui se trouve dans le message <samlp:Réponse>
 4447 pour en faire une instance <rim:Utilisateur>. Cette opération entraîne l'établissement du
 4448 contexte <rim:Utilisateur> pour le contexte de sécurité.
- 4449 1.2.2.3 Le registre applique ensuite la décision prise en matière d'autorisation à partir de la
 4450 demande HTTP initiale et de l'élément <rim:Utilisateur>. On émet l'hypothèse que cette décision
 4451 visait à favoriser le traitement de la demande. Le registre traite celle-ci, après quoi il renvoie la
 4452 ressource demandée au demandeur HTTP par le biais de la réponse HTTP.
 4453

4454 **11.6.3 Opération d'accès unifié – demandeur HTTP authentifié**

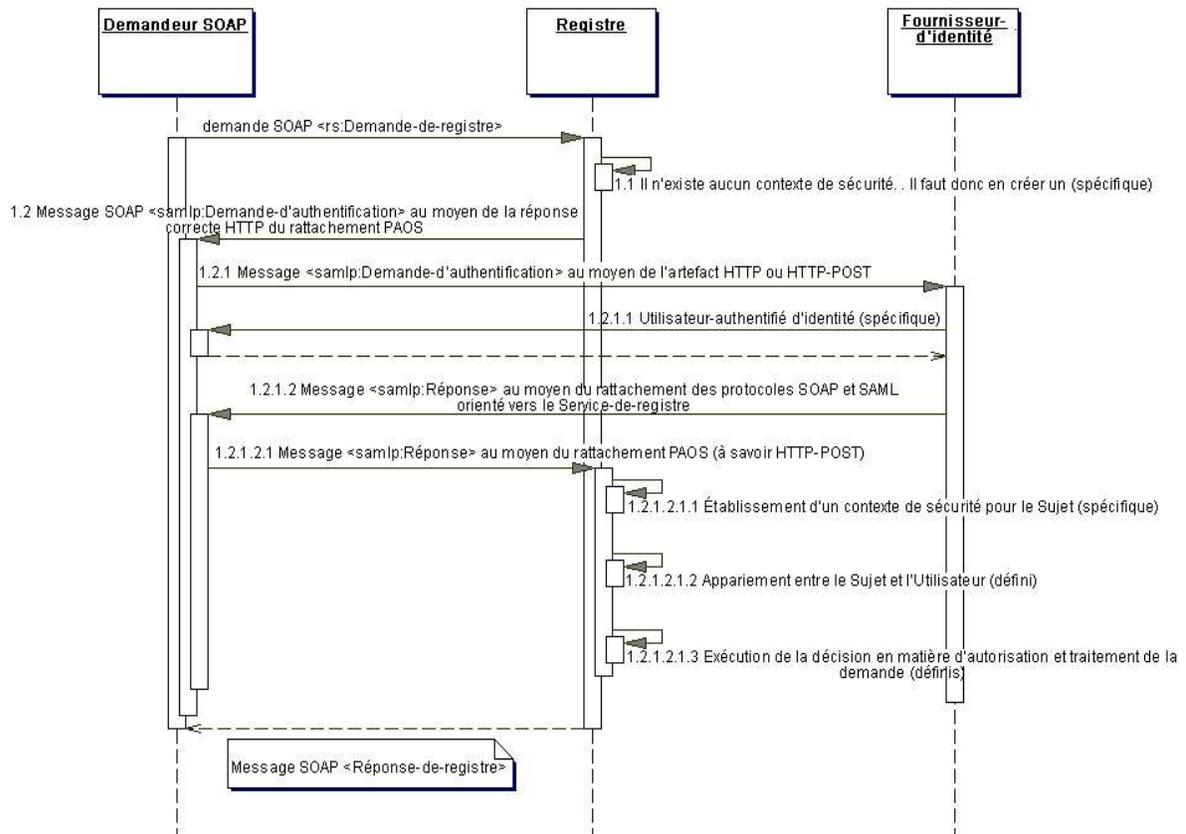
4455 Cette opération s'applique si le demandeur HTTP est d'abord authentifié à l'aide d'un Fournisseur-
4456 d'identité et accède ensuite au registre de HTTP au moyen d'un agent d'utilisateur, comme un navigateur
4457 Web.

4458 On n'a encore défini aucune méthode normalisée d'exécution de déclaration SAML provenant du
4459 demandeur de registre qui est identifié à l'aide d'un Fournisseur-d'identité. Cette authentification s'effectue
4460 dans le cadre du protocole HTTP et s'applique à un fournisseur de service, comme le registre. Tout
4461 registre PEUT gérer ce scénario d'une manière propre à l'implantation. Le Fournisseur-d'identité définit
4462 généralement les méthodes propres à l'implantation.

4463 **11.6.4 Opération d'accès unifié – demandeur SOAP non authentifié**

4464 Cette opération s'applique si un demandeur de registre non authentifié accède au registre par le biais du
4465 protocole SOAP.

4466 Le Diagramme 29 illustre les étapes nécessaires à cet égard.
4467



4469 **Diagramme 29 : Opération d'accès unifié – demandeur SOAP non authentifié**

4470 **11.6.4.1 Séquence de scénario**

4471 Le Diagramme 29 illustre les étapes ci-dessous qui se rattachent à l'opération d'accès unifié.

- 4472 1 Le demandeur SOAP envoie un message SOAP <rs:Demande-de-registre>, tel qu'un message
4473 <lcm:Demande-d'inscription-d'objet>, à une interface de Registre, comme le Gestionnaire-de-cycle-

4474 de-vie. Dans l'en-tête de sa demande, le demandeur SOAP spécifie qu'il est un demandeur
4475 client/proxy, comme le définit le profil de client/proxy dans le [SAMLProf].

4476 1.1 Le registre vérifie s'il dispose déjà d'un contexte de sécurité qui a été établi pour le sujet associé à
4477 la demande en question. Il s'assure qu'aucun contexte de sécurité n'a été mis en place
4478 antérieurement.

4479 1.2 Étant donné que la demande provient d'une application client/proxy, le registre se sert du profil de
4480 client/proxy établi par le [SAMLProf] et transmet au demandeur SOAP un message SOAP
4481 <sampl:Demande-d'authentification> à titre de réponse au message SOAP <rs:Demande-de-
4482 registre>. Le registre agit ainsi en utilisant le rattachement PAOS, conformément au [SAMLBind].
4483 La réponse comporte une réponse HTTP dont le statut est adéquat.

4484 1.2.1 Le demandeur SOAP entame ensuite le protocole <sampl:Demande-d'authentification> à l'aide
4485 du Fournisseur-d'identité. On envoie directement à celui-ci le message <sampl:Demande-
4486 d'authentification> en se servant du rattachement d'artefact HTTP ou HTTP-POST.

4487 1.2.1.1 Le Fournisseur-d'identité a recours à des méthodes propres à l'implantation pour définir le sujet.
4488 Pour ce faire, il doit généralement communiquer avec le demandeur SOAP pour obtenir le titre
4489 lié au sujet, après quoi il doit se servir de ce titre pour certifier qu'il connaît le sujet. Si la
4490 communication est axée sur les protocoles SSL et TLS, l'acquisition du titre s'effectue sans la
4491 moindre intervention de l'utilisateur et se fait directement auprès du demandeur SOAP. On émet
4492 l'hypothèse que le Fournisseur-d'identité peut authentifier le sujet.

4493 1.2.1.2 Le Fournisseur-d'identité transmet au demandeur SOAP un message <sampl:Réponse> qui
4494 renferme un élément <saml:Déclaration-d'authentification>, ce qu'il fait en utilisant un
4495 rattachement des protocoles SOAP et SAML. Le registre figure dans l'en-tête HTTP à titre de
4496 cible définitive de la réponse.

4497 1.2.1.2.1 Le demandeur SOAP transmet au registre un message <sampl:Réponse> qui comprend un
4498 élément <saml:Déclaration-d'authentification>, ce qu'il fait en utilisant un rattachement PAOS
4499 par le biais de HTTP-POST.

4500 1.2.1.2.1.1 Le registre se sert de méthodes propres à l'implantation pour établir un contexte de sécurité
4501 par rapport au sujet authentifié par le Fournisseur-d'identité. Il agit ainsi en se fondant sur
4502 l'information qui concerne le sujet et qui est présentée dans le message <sampl:Réponse>.
4503 Ces mesures peuvent comprendre la création d'une session HTTP pour le
4504 demandeur HTTP.

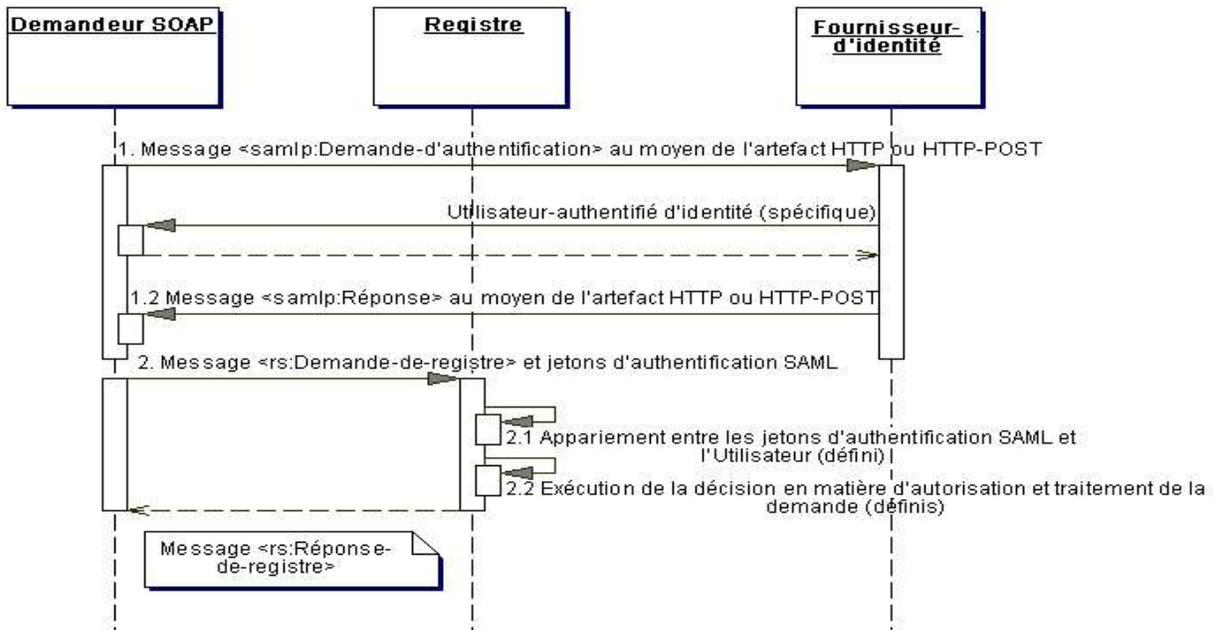
4505 1.2.1.2.1.2 Le registre apparie l'information sur le sujet qui se trouve dans le message
4506 <sampl:Réponse> pour en faire une instance <rim:Utilisateur>. Cette opération entraîne
4507 l'établissement du contexte <rim:Utilisateur> pour le contexte de sécurité.

4508 1.2.1.2.1.3 Le registre applique ensuite la décision prise en matière d'autorisation à partir de la
4509 demande SOAP initiale et de l'élément <rim:Utilisateur>. On émet l'hypothèse que cette
4510 décision visait à favoriser le traitement de la demande. Le registre traite celle-ci, après quoi il
4511 renvoie un message SOAP <rs:Réponse-de-registre> à titre de réponse à la demande SOAP
4512 initiale <rs:Demande-de-registre>.
4513

4514 11.6.5 Opération d'accès unifié – demandeur SOAP authentifié

4515 Cette opération s'applique si le demandeur de registre est d'abord identifié directement à l'aide d'un
4516 Fournisseur-d'identité, et s'il transmet ensuite une demande au registre au moyen du protocole SOAP.
4517

4518 Diagramme 30 : Opération d'accès unifié – demandeur SOAP authentifié



4519 11.6.5.1 Séquence de scénario

4520 Le Diagramme 30 illustre les étapes ci-dessous qui se rattachent à l'opération d'accès unifié.

4521 1 Le demandeur SOAP instaure le protocole <samlp:Demande-d'authentification> directement auprès
 4522 du Fournisseur-d'identité. L'élément <samlp:Demande-d'authentification> est transmis par le biais du
 4523 rattachement d'artefact HTTP ou HTTP-POST.

4524 1.1 Le Fournisseur-d'identité a recours à des méthodes propres à l'implantation pour définir le sujet.
 4525 Pour ce faire, il doit généralement communiquer avec le demandeur SOAP pour obtenir le titre lié
 4526 au sujet, après quoi il doit se servir de ce titre pour certifier qu'il connaît le sujet. Si la
 4527 communication est axée sur les protocoles SSL et TLS, l'acquisition du titre s'effectue sans la
 4528 moindre intervention de l'utilisateur et se fait directement auprès du demandeur SOAP. On émet
 4529 l'hypothèse que le Fournisseur-d'identité peut authentifier le sujet.

4530 1.2 Le Fournisseur-d'identité transmet au demandeur SOAP un message <samlp:Réponse> qui
 4531 renferme un élément <saml:Déclaration-d'authentification>, ce qu'il fait en utilisant un rattachement
 4532 SAML d'artefact HTTP ou HTTP-POST.

4533 2 Le demandeur SOAP envoie un message SOAP <rs:Demande-de-registre>, tel qu'un message
 4534 <lcm:Demande-d'inscription-d'objet>, à une interface de registre, comme le Gestionnaire-de-cycle-
 4535 de-vie. L'élément <soap:En-tête> du message SOAP <rs:Demande-de-registre> englobe des jetons
 4536 d'authentification SAML, comme le définit le [WSS-SAML]. Ces jetons sont fondés sur l'élément
 4537 <samlp:Réponse> qui s'applique pendant l'authentification.

4538 2.1 Le registre assure l'appariement entre les jetons d'authentification SAML découlant de l'élément
 4539 <soap:En-tête> du message <rs:Demande-de-registre>, et une instance <rim:Utilisateur>. Cet
 4540 appariement établit le contexte <rim:Utilisateur> nécessaire à la demande.

4541 2.2 Le registre applique ensuite la décision prise en matière d'autorisation à partir de la
4542 demande SOAP initiale et de l'élément <rim:Utilisateur>. On émet l'hypothèse que cette décision
4543 visait à favoriser le traitement de la demande. Le registre traite celle-ci, après quoi il renvoie un
4544 message SOAP <rs:Réponse-de-registre> à titre de réponse à la demande SOAP initiale
4545 <rs:Demande-de-registre>.
4546

4547 **11.6.6 Règles de génération de l'élément <saml:Demande-** 4548 **d'authentification>**

4549 Les règles suivantes DOIVENT être respectées au moment de la présentation d'un élément
4550 <saml:Demande-d'authentification> par le registre ou l'application de registre :
4551

- 4552 • Tout registre DOIT établir une Règle-d'identifiant-de-nom dans l'élément <saml:Demande-
4553 d'authentification>;
- 4554 • Comme le définit le [SAMLCore], le format de la Règle-d'identifiant-de-nom DOIT être le suivant :
4555 urn:oasis:names:tc:SAML:2.0:nameid-format:persistent. Il est important de mentionner que c'est
4556 l'identifiant permanent qui s'apparie à l'attribut identifiant de l'élément <rim:Utilisateur>.
4557

4558 **11.6.7 Règles de traitement de l'élément <saml:Réponse>**

4559 La section 11.6.7 décrit la façon dont le registre traite l'élément <saml:Réponse> pour en faire un
4560 élément <saml:Demande-d'authentification>.

4561 **Traitement de l'élément <saml:Réponse>**

- 4562 • Traitement de la réponse : Le registre DOIT vérifier l'élément <ds:Signature> propre à l'élément
4563 <saml:Réponse>, s'il y en a un.
- 4564 • Le registre DOIT vérifier l'élément <saml:Statut> lié à l'élément <saml:Réponse> pour s'assurer
4565 qu'il ne comporte pas d'erreur. Si l'élément <saml:Statut> englobe un élément de premier niveau
4566 <saml:Code-de-statut> dont la valeur NE correspond PAS à
4567 urn:oasis:names:tc:SAML:2.0:status:Success, le registre DOIT envoyer une
4568 Exception-d'authentification. Le message de cette exception DEVRAIT comprendre les
4569 renseignements propres au Code-de-statut, au Message-de-statut et aux Détails-de-statut de
4570 l'élément <saml:Statut>.

4571 **Traitement de l'élément <saml:Déclaration>**

- 4572 • Le registre DEVRAIT vérifier l'élément <saml:Déclaration> par rapport aux conditions et respecter
4573 toute condition normalisée définie par le [SAMLCore] et spécifiée, le cas échéant.

4574 **Traitement de l'élément <saml:Déclaration-d'authentification>**

- 4575 • Le registre DOIT examiner l'attribut session-pas-en-date-du-ou-après de l'élément
4576 <saml:Déclaration-d'authentification> pour s'assurer de la validité de la session authentifiée.

4577 **Traitement de l'élément <saml:Sujet>**

- 4578 • Tout registre DOIT apparier l'élément <saml:Sujet> à une instance <rim:Utilisateur>, comme le
4579 décrit la section 11.6.8.

4580 **11.6.8 Appariement entre un sujet et un utilisateur**

4581 Conformément au [SAMLCore], un élément <saml:Réponse> qui se rattache à un élément
4582 <saml:Demande-d'authentification> DOIT englober un élément <saml:Sujet> authentifié au préalable par
4583 le Fournisseur-d'identité. L'élément <saml:Réponse> DOIT également comprendre un élément
4584 <saml:Déclaration-d'authentification> qui affirme que le Fournisseur-d'identité a bel et bien authentifié le
4585 sujet en question.

4586 Le tableau ci-dessous définit l'appariement entre un élément <saml:Sujet> et un élément <rim:Utilisateur>.
4587

Attribut sujet	Attribut utilisateur	Description
contenu de l'Identifiant-de-nom	attribut identifiant	Le format de l'Identifiant-de-nom DOIT se traduire par « urn:oasis:names:tc:SAML:1.1:nameid-format:persistent ».

4588 **Tableau 8 : Appariement entre un sujet et un utilisateur**

4589 Il est important de mentionner qu' au moment d'apparier un sujet à un utilisateur, on NE DEVRAIT tenir
4590 compte d'aucun attribut sujet qui n'est pas précisé ci-dessus. On NE DOIT PAS non plus spécifier un
4591 attribut utilisateur qui ne figure pas ci-dessus.

4592 **11.7 Utilisateurs externes**

4593 Le profil SAML favorise l'inscription des utilisateurs de registre dans un Fournisseur-d'identité qui est
4594 externe par rapport au registre. Ces utilisateurs sont appelés « utilisateurs externes ». Un registre crée de
4595 manière dynamique des utilisateurs de ce genre en appariant un sujet SAML à une instance d'Utilisateur.

4596 Voici certaines restrictions qui se rattachent aux instances d'utilisateur externe :

- 4597 • Les instances d'utilisateur externe sont transitoires du point de vue du registre et NE DOIVENT
4598 PAS être stockées dans celui-ci à titre d'instances d'Utilisateur;
- 4599 • Tout Objet-de-registre DOIT ne citer aucun utilisateur externe à moins qu'il ne soit composé au
4600 sein même d'un tel objet. Par contre, les Objets-de-registre composés, comme les instances de
4601 Classification, peuvent faire référence à leur instance d'utilisateur externe parent;
- 4602 • Étant donné qu'elles sont transitoires, les instances d'utilisateur externe NE DOIVENT PAS
4603 s'apparier à une interrogation de registre en particulier.

4604
4605
4606
4607
4608

12 Support de langue naturelle

4609

4610 Le chapitre 12 décrit les caractéristiques du support de langue naturelle du registre ebXML.

12.1 Terminologie

4611

4612 On se sert des termes suivants pour le support de langue naturelle :

Terme	Description
Jeu de caractères codé	Le jeu de caractères codé représente un appariement entre un jeu de caractères abstraits et un jeu d'entiers [RFC 2130]. ISO-10646, US-ASCII et ISO-8859-1 constituent quelques exemples de jeu de caractères codé.
Schéma d'encodage de caractères	Le schéma d'encodage de caractères désigne un appariement entre un jeu de caractères codé (ou plusieurs jeux) et un jeu d'octets [RFC 2130]. ISO-2022 et UTF-8 représentent quelques exemples de schéma d'encodage de caractères.
Jeu de caractères	<ul style="list-style-type: none">Le jeu de caractères évoque un ensemble de règles en vue de l'appariement entre une séquence d'octets et une séquence de caractères ([RFC 2277] et [RFC 2278]). ISO-2022-JP et EUC-KR sont des exemples de jeu de caractères.Il est possible de trouver une liste de jeux de caractères enregistrés dans le [IANA].

4613

12.2 Support de langue naturelle et messages de protocole de registre

4614

4615

4616 Pour assurer le traitement exact des données sur le plan des services de registre et d'application de
4617 registre, il est essentiel pour le destinataire d'un message de protocole quelconque de connaître le jeu de
4618 caractères qui est utilisé.

4619 Toute application de registre DEVRAIT préciser le paramètre de jeu de caractères dans l'en-tête du MIME
4620 si elle établit un texte XML à titre de Type-de-contenu. Tout registre DOIT déterminer le paramètre de jeu
4621 de caractères dans l'en-tête du MIME s'il spécifie un texte XML à titre de Type-de-contenu.

4622 Voici un exemple de spécification du jeu de caractères dans l'en-tête du MIME :

4623

4624

4625

```
Content-Type: text/xml; charset=ISO-2022-JP
```

4626

4627 Comme le définit le [RFC 3023], si un registre quelconque reçoit un message de protocole sans
4628 paramètre de jeu de caractères, il DOIT utiliser la valeur de jeu de caractères par défaut « US-ASCII ».

4629 De plus, si on se sert d'une entité de demande XML, le paramètre de jeu de caractères est optionnel, et
4630 les services de registre et d'application de registre DOIVENT satisfaire aux exigences énoncées dans la
4631 section 4.3.3 du [REC-XML], qui traite directement de cette contingence.

4632 Si on a recours à un autre Type-de-contenu, l'utilisation du jeu de caractères DOIT se faire conformément
4633 au [RFC 3023].

4634 12.3 Gestion du support de langue naturelle dans les Objets-de- 4635 registre

4636 Le schéma XML du modèle d'information [RR-RIM-XSD] définit l'élément <rim:Type-de-chaîne-en-toutes-
4637 langues> en vue de l'établissement des éléments qui renferment une valeur de chaîne dépendante du
4638 paramètre de lieu.

4639

```
4640 <complexType name="InternationalStringType">  
4641 <sequence maxOccurs="unbounded" minOccurs="0">  
4642 <element ref="tns:LocalizedString"/>  
4643 </sequence>  
4644 </complexType>
```

4645

4646 Tout Type-de-chaîne-en-toutes-langues peut renfermer plusieurs Chaînes-en-une-langue-donnée comme
4647 il peut n'en avoir aucune. Chacune d'elles comprend une valeur de chaîne qui se traduit par un jeu local
4648 précis de langue et de caractères.

4649

```
4650 <complexType name="LocalizedStringType">  
4651 <attribute ref="xml:lang" default="en-US"/>  
4652 <attribute default="UTF-8" name="charset"/>  
4653 <attribute name="value" type="tns:FreeFormText" use="required"/>  
4654 </complexType>
```

4655

4656 Citons à titre d'exemples de ces attributs les attributs « nom » et « description » de la classe Objet-de-
4657 registre qui sont définis dans le [ebRIM], comme il est indiqué ci-dessous.

```
4658 <complexType name="InternationalStringType">  
4659 <sequence maxOccurs="unbounded" minOccurs="0">  
4660 <element ref="tns:LocalizedString"/>  
4661 </sequence>  
4662 </complexType>  
4663 <element name="InternationalString"  
4664 type="tns:InternationalStringType"/>  
4665 <element name="Name" type="tns:InternationalStringType"/>  
4666 <element name="Description" type="tns:InternationalStringType"/>  
4667  
4668 <complexType name="LocalizedStringType">  
4669 <attribute ref="xml:lang" default="en-US"/>  
4670 <!--attribute name = "lang" default = "en-US" form = "qualified" type  
4671 = "language"/-->  
4672 <attribute default="UTF-8" name="charset"/>  
4673 <attribute name="value" type="tns:FreeFormText" use="required"/>  
4674 </complexType>  
4675 <element name="LocalizedString" type="tns:LocalizedStringType"/>
```

4676

4677 Une Chaîne-en-toutes-langues peut gérer de multiples paramètres de lieu dans sa propre collection de
4678 Chaînes-en-une-langue-donnée.

4679 Dans le schéma ci-dessus, une seule instance d'Objet-de-registre peut englober des valeurs pour
4680 n'importe quel élément sensible au support de langue naturelle dans de nombreux paramètres de lieu.

4681 L'exemple ci-dessous présente la façon dont un seul Objet-de-registre peut comprendre des éléments
4682 <rim:Nom> et <rim:Description> sensibles au support de langue naturelle, alors que leur valeur est
4683 précisée dans de multiples paramètres de lieu. Il est important de mentionner que les éléments
4684 <rim:Nom> et <rim:Description> ont comme type l'élément <rim:Type-de-chaîne-en-toutes-langues>.

```
4685 <rim:ExtrinsicObject id="{ID}" mimeType="text/xml">  
4686 <rim:Name>  
4687 <rim:LocalizedString xml:lang="en-US" value="customACP1.xml"/>  
4688 <rim:LocalizedString xml:lang="fi-FI" value="customACP1.xml"/>  
4689 <rim:LocalizedString xml:lang="pt-BR" value="customACP1.xml"/>
```

4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700

```
</rim:Name>
<rim:Description>
  <rim:LocalizedString xml:lang="en-US" value="A sample custom
ACP"/>
  <rim:LocalizedString xml:lang="fi-FI" value="Esimerkki custom
ACP"/>
  <rim:LocalizedString xml:lang="pt-BR" value="Exemplo de ACP
customizado
"/>
</rim:Description>
</rim:ExtrinsicObject>
```

4701

4702 On trouve de l'information sur les paramètres de lieu au niveau des sous-éléments. Par conséquent,
4703 aucun langage ou jeu de caractères n'est associé à une instance précise d'Objet-de-registre.

4704 **12.3.1 Jeu de caractères d'une Chaîne-en-une-langue-donnée**

4705 Le jeu de caractères utilisé par une chaîne propre aux paramètres de lieu (Chaîne-en-une-langue-
4706 donnée) est défini par l'attribut de jeu de caractères. Les applications de registre DEVRAIENT choisir la
4707 valeur UTF-8 ou UTF-16 pour l'attribut jeu-de-caractères des Chaînes-en-une-langue-donnée afin
4708 d'assurer une interopérabilité maximale.

4709 **12.3.2 Langage d'une Chaîne-en-une-langue-donnée**

4710 Le langage PEUT être spécifié dans l'attribut XML:LANG (section 2.12 du [REC-XML]).

4711 **12.4 Support de langue naturelle et items de référentiel**

4712 Une seule instance d'Objet-extrinsèque peut gérer de multiples paramètres de lieu, mais elle est toujours
4713 associée à un seul item de référentiel. Celui-ci PEUT être représenté par un seul paramètre de lieu ou par
4714 de multiples paramètres de lieu. La présente spécification ne spécifie aucune exigence relative au support
4715 de langue naturelle pour les items de référentiel.

4716 **12.4.1 Jeu de caractères d'un item de référentiel**

4717 Si un demandeur inscrit un item de référentiel, il PEUT établir le jeu de caractères dont cet item se sert s'il
4718 utilise l'en-tête de *Type-de-contenu* du MIME propre au MIME multipart qui renferme l'item de référentiel,
4719 comme il est indiqué ci-dessous.

4720
4721
4722
4723

```
Content-Type: text/xml; charset="UTF-8"
```

4724 Les applications de registre DEVRAIENT choisir la valeur UTF-8 ou UTF-16 pour l'attribut jeu-de-
4725 caractères des Chaînes-en-une-langue-donnée pour assurer une interopérabilité maximale. Tout registre
4726 DOIT préserver le jeu de caractères d'un item de référentiel comme il est précisé au départ au moment de
4727 l'inscription de celui-ci dans le registre.

4728 **12.4.2 Langage d'un item de référentiel**

4729 L'en-tête de langage-du-contenu du MIME propre à la partie de corps du MIME qui englobe l'item de
4730 référentiel PEUT déterminer le langage pour un item de référentiel spécifique à des paramètres de lieu.
4731 La valeur de la caractéristique de cet en-tête DOIT être conforme au [RFC 1766].

4732 Le présent document n'établit actuellement que la méthode d'envoi de l'information associée au jeu de
4733 caractères et au langage, de même que la manière dont cette information est stockée dans un registre.
4734 On PEUT toutefois utiliser l'information de langage à titre de l'un des critères d'interrogation, comme
4735 l'extraction des DTD rédigées en français seulement. De plus, de la même façon qu'une application de
4736 registre tient à sélectionner la langue qu'elle préfère pour les messages provenant des services de

4737 registre, on pourrait adopter une procédure de négociation de la langue comme fonctionnalité
4738 additionnelle en vue de la révision éventuelle du document.

4739

13 Conformité

4740 Le chapitre 13 décrit les exigences techniques en matière de conformité qui se rattachent au
4741 registre ebXML. Il est important de mentionner qu'il ne définit aucun test de conformité en particulier
4742 visant la vérification du respect à l'égard des divers profils de conformité.

13.1 Profils de conformité

4744 Le registre ebXML DOIT être conforme à l'un des profils de conformité suivants :

- 4745 • Registre allégé – Le registre doit implanter un ensemble minimal de caractéristiques de base
4746 établies par la présente spécification.
- 4747 • Registre complet – Le registre doit implanter un ensemble de caractéristiques en plus de celles déjà
4748 exigées pour le profil de conformité propre au registre allégé.

13.2 Grille de caractéristiques

4750 Le tableau suivant énumère les exigences en matière d'implantation qui s'appliquent à chaque
4751 caractéristique définie par la présente spécification pour chacun des profils de conformité énumérés ci-
4752 dessus :

Tableau 9 : Grille de conformité aux caractéristiques

Caractéristique	Registre allégé	Registre complet
Rattachement SOAP		
Rattachement de Gestionnaire-d'interrogation	DOIT	DOIT
Rattachement de Gestionnaire-de-cycle-de-vie	DOIT	DOIT
Rattachement de protocole HTTP		
URL fondé sur l'encodage d'un appel de procédure à distance	DOIT	DOIT
URL défini par le demandeur	PEUT	DOIT
URL par chemin d'accès	PEUT	DOIT
Gestionnaire-de-cycle-de-vie		
Protocole d'Inscription-d'objet	DOIT	DOIT
Protocole de Mise-à-jour-d'objet	DOIT	DOIT
Protocole d'Approbaton-d'objet	DOIT	DOIT
Protocole de Déclassement-d'objet	DOIT	DOIT
Protocole de Rétablissement-d'objet	DOIT	DOIT
Protocole de Suppression-d'objet	DOIT	DOIT
Contrôle de version géré par registre	PEUT	DOIT
Gestionnaire-d'interrogation		
Requête SQL	PEUT	DOIT
Filtre d'interrogation	DOIT	DOIT
Interrogation stockée avec paramètres	PEUT	DOIT
Interrogation itérative	PEUT	DOIT
Notification d'événement	PEUT	DOIT
Services de gestion de contenu		

Caractéristique	Registre allégé	Registre complet
Protocole de validation de contenu	PEUT	DOIT
Protocole de catalogage de contenu	PEUT	DOIT
Service de catalogage XML canonique	PEUT	DOIT
Registres en coopération		
Références d'objet distantes	PEUT	DOIT
Interrogations fédérées	PEUT	DOIT
Duplication d'objet	PEUT	DOIT
Déplacement d'objet	PEUT	DOIT
Sécurité du registre		
Gestion de l'identité	DOIT	DOIT
Sécurité de message		
Sécurité de la couche de transport	PEUT	DOIT
Sécurité de message SOAP	DOIT	DOIT
Sécurité d'item de référentiel	DOIT	DOIT
Autorisation et contrôle d'accès		
Règle de contrôle d'accès par défaut	DOIT	DOIT
Règles contextualisées de contrôle d'accès	PEUT	DOIT
Piste de vérification	DOIT	DOIT
Profil SAML de registre	PEUT	DOIT
Support de langue naturelle	DOIT	DOIT

4753

14 Références

14.1 Références normatives

- 4756 [RFC 2119] Bradner, Scott. *RFC 2119 : Les mots-clés à utiliser dans les RFC pour indiquer les niveaux d'expérience*, Internet Engineering Task Force (IETF), mars 1997, <http://www.ietf.org/rfc/rfc2119.txt>.
- 4757
- 4758
- 4759 [ebRIM] Modèle d'information du registre ebXML, version 3.0, <http://www.oasis-open.org/committees/regrep/documents/3.0/specs/regrep-rim-3.0-draft-03.pdf>.
- 4760
- 4761
- 4762 [REC-XML] Recommandation W3C. Langage de balisage extensible (XML), version 1.0, deuxième édition, <http://www.w3.org/TR/REC-xml>.
- 4763
- 4764
- 4765 [RFC 1766] H. Alvestrand. *RFC 1766: Tags for the Identification of Languages*, IETF (Internet Engineering Task Force), 1995, <http://www.cis.ohio-state.edu/htbin/rfc/rfc1766.html>.
- 4766
- 4767
- 4768 [RFC 2130] *RFC 2130: The Report of the IAB Character Set Workshop held 29 February – 1 March, 1996*, IETF (Internet Engineering Task Force), <http://www.faqs.org/rfcs/rfc2130.html>.
- 4769
- 4770
- 4771 [RFC 2277] H. Alvestrand. *RFC 2277 : La politique de l'IETF concernant les jeux de caractères et les langues*, IETF (Internet Engineering Task Force), 1998, <http://www.cis.ohio-state.edu/htbin/rfc/rfc2277.html>.
- 4772
- 4773
- 4774 [RFC 2278] N. Freed et J. Postel. *RFC 2278 : Procédures d'enregistrement des jeux de caractères de l'IANA*, IETF (Internet Engineering Task Force), 1998, <http://www.cis.ohio-state.edu/htbin/rfc/rfc2278.html>.
- 4775
- 4776
- 4777 [RFC 2616] Fielding et al. *RFC 2616 : Protocole de transfert hypertexte – HTTP/1.1*, IETF (Internet Engineering Task Force), 1999, <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.
- 4778
- 4779
- 4780 [RFC 2965] D. Kristol et al. *RFC 2965 : Les mécanismes de gestion d'état HTTP*, IETF (Internet Engineering Task Force), 2000, <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.
- 4781
- 4782
- 4783 [RR-CMS-XSD] ebXML Registry Content Management Services XML Schema, <http://www.oasis-open.org/committees/regrep/documents/3.0/schema/rim.xsd>.
- 4784
- 4785 [RR-LCM-XSD] ebXML Registry LifeCycleManager XML Schema, <http://www.oasis-open.org/committees/regrep/documents/3.0/schema/lcm.xsd>.
- 4786
- 4787 [RR-RIM-XSD] ebXML Registry Information Model XML Schema, <http://www.oasis-open.org/committees/regrep/documents/3.0/schema/rim.xsd>.
- 4788
- 4789 [RR-RS-XSD] ebXML Registry Service Protocol XML Schema, <http://www.oasis-open.org/committees/regrep/documents/3.0/schema/rs.xsd>.
- 4790
- 4791 [RR-QM-XSD] ebXML Registry QueryManager XML Schema, <http://www.oasis-open.org/committees/regrep/documents/3.0/schema/query.xsd>.
- 4792
- 4793 [SAMLBind] S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language (SAML)*, version 2.0, OASIS SSTC, septembre 2004, document ID sstc-saml-bindings-2.0-cd-02, <http://www.oasis-open.org/committees/security/>.
- 4794
- 4795
- 4796
- 4797 Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
- 4798 [SAMLConform] P. Mishra et al. *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML)*, version 2.0, OASIS SSTC, septembre 2004, document ID sstc-saml-conformance-2.0-cd-02, <http://www.oasis-open.org/committees/security/>.
- 4799
- 4800
- 4801

4802		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
4803	[SAMLCore1.0]	E. Maler et al. <i>Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)</i> , OASIS, novembre 2002,
4804		http://www.oasis-open.org/committees/download.php/1371/oasis-sstc-saml-core-1.0.pdf .
4805		
4806		
4807		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
4808	[SAMLProf]	S. Cantor et al. <i>Profiles for the OASIS Security Assertion Markup Language (SAML)</i> , version 2.0, OASIS SSTC, septembre 2004, document ID sstc-saml-profiles-2.0-cd-02,
4809		http://www.oasis-open.org/committees/security/ .
4810		
4811		
4812		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
4813	[SAMLXSD]	S. Cantor et al. <i>SAML protocols schema</i> , OASIS SSTC, septembre 2004, document ID sstc-saml-schema-protocol-2.0,
4814		http://www.oasis-open.org/committees/security/ .
4815		
4816		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
4817	[SAMLXSD]	S. Cantor et al. <i>SAML assertions schema</i> , OASIS SSTC, septembre 2004, document ID sstc-saml-schema-assertion-2.0,
4818		http://www.oasis-open.org/committees/security/ .
4819		
4820		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
4821	[SOAP11]	Note du W3C. Simple Object Access Protocol, mai 2000,
4822		http://www.w3.org/TR/SOAP .
4823	[SwA]	Note du W3C. Protocole SOAP avec pièces jointes, décembre 2000,
4824		http://www.w3.org/TR/SOAP-attachments .
4825	[SQL]	Structured Query Language (FIPS PUB 127-2),
4826		http://www.itl.nist.gov/fipspubs/fip127-2.htm .
4827	[SQL/PSM]	Database Language SQL — Part 4: Persistent Stored Modules (SQL/PSM) [ISO/IEC 9075-4:1996],
4828		ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets .
4829		
4830	[UUID]	DCE 128 bit Universal Unique Identifier,
4831		http://www.opengroup.org/onlinepubs/009629399/apdx.htm#tagcjh_20 .
4832	[WSDL]	Note du W3C. Web Services Description Language (WSDL), version 1.1,
4833		http://www.w3.org/TR/wsdl .
4834	[XML]	T. Bray et al. Langage de balisage extensible (XML), version 1.0, deuxième édition, Consortium W3C, octobre 2000,
4835		http://www.w3.org/TR/REC-xml .
4836		
4837	[XMLDSIG]	La syntaxe et le traitement XML-Signature,
4838		http://www.w3.org/TR/2001/PR-xmlsig-core-20010820/ .
4839	[WSI-BSP]	WS-I Basic Security Profile, version 1.0,
4840		http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html .
4841		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.
4842	[WSS-SMS]	Web Services Security: SOAP Message Security 1.0,
4843		http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf .
4844		
4845	[WSS-SWA]	Web Services Security: SOAP Message with Attachments (SwA) Profile 1.0,
4846		http://www.oasis-open.org/apps/org/workgroup/wss/download.php/10902/wss-swa-profile-1.0-cd-01.pdf .
4847		
4848		Remarque : Cet URL sera mis à jour au moment de l'achèvement du document.

4849 14.2 Références informatives

4850	[ebBPSS]	Spécification d'un schéma de processus d'affaires ebXML,
------	-----------------	--

4851		http://www.ebxml.org/specs .
4852	[ebCPP]	Spécification de Profil et d'Entente de protocole de collaboration ebXML,
4853		http://www.ebxml.org/specs/ .
4854	[ebMS]	Spécification de Service de Messages ebXML, version 1.0,
4855		http://www.ebxml.org/specs/ .
4856	[DeltaV]	RFC 3253: Versioning Extension to WebDAV, IETF,
4857		http://www.webdav.org/deltav/protocol/rfc3253.html .
4858	[XPT]	Langage XML Path (XPath), version 1.0,
4859		http://www.w3.org/TR/xpath .
4860	[IANA]	Keld Simonsen et al. IANA (Internet Assigned Numbers Authority),
4861		Official Names for Character Sets,
4862		http://www.iana.org/ .
4863	[RFC 2392]	E. Levinson. <i>RFC 2392: Content-ID and Message-ID Uniform Resource Locators</i> ,
4864		IETF,
4865		http://www.ietf.org/rfc/rfc2392.txt .
4866	[RFC 2828]	R. Shirey. <i>RFC 2828 : Internet Security Glossary</i> , IETF (Internet Engineering
4867		Task Force), mai 2000,
4868		http://www.cis.ohio-state.edu/htbin/rfc/rfc2828.html
4869	[RFC 3023]	M. Murata. <i>RFC 3023 : Les types de média XML</i> , IETF (Internet Engineering
4870		Task Force), 2001,
4871		ftp://ftp.isi.edu/in-notes/rfc3023.txt .
4872	[SAMLMeta]	S. Cantor et al. <i>Metadata for the OASIS Security Assertion Markup Language</i>
4873		(SAML), version 2.0, OASIS SSTC, septembre 2004, document ID sstc-saml-
4874		metadata-2.0-cd-02,
4875		http://www.oasis-open.org/committees/security/ .
4876	[SAMLGloss]	J. Hodges et al. <i>Glossary for the OASIS Security Assertion Markup Language</i>
4877		(SAML), version 2.0, OASIS SSTC, septembre 2004, document ID sstc-saml-
4878		glossary-2.0-cd-02,
4879		http://www.oasis-open.org/committees/security/ .
4880	[SAMLSecure]	F. Hirsch et al. <i>Security and Privacy Considerations for the OASIS Security</i>
4881		<i>Assertion Markup Language (SAML)</i> , version 2.0, OASIS SSTC,
4882		septembre 2004, document ID sstc-saml-sec-consider-2.0-cd-02,
4883		http://www.oasis-open.org/committees/security/ .
4884	[SAMLTech]	J. Hughes et al. <i>Technical Overview of the OASIS Security Assertion Markup</i>
4885		<i>Language (SAML)</i> , version 2.0,
4886		http://www.oasis-open.org/committees/download.php/7874/sstc-saml-tech-
4887		overview-2.0-draft-01.pdf .
4888	[UML]	Langage de modélisation unifié,
4889		http://www.uml.org ,
4890		http://www.omg.org/cgi-bin/doc?formal/03-03-01 .
4891		

4892

A.Remerciements

4893 À titre de rédacteurs, nous voudrions remercier les membres du Comité technique du
4894 registre ebXML/OASIS de leur contribution. Nous tenons à remercier les membres qui étaient votants au
4895 moment de la publication du présent document. Leur nom figure sur la page titre de celui-ci.

4896 Nos remerciements vont enfin aux personnes suivantes pour leur collaboration et dont les documents ont
4897 été utilisés comme intrants pour les spécifications du registre ebXML/OASIS :

4898

Nom	Affiliation
Aziz Abouelfoutouh	Gouvernement du Canada
Ed Buchinski	Gouvernement du Canada
Asuman Dogac	Université technique du Moyen-Orient, Ankara (Turquie)
Michael Kass	NIST
Richard Lessard	Gouvernement du Canada
Evan Wallace	NIST
David Webber	Membre à titre individuel

4899

B.Avis

4901 OASIS n'a pas de position concernant la validité ou la portée des droits de propriété intellectuelle ou
4902 d'autres droits qui pourraient être revendiqués comme étant propres à l'implantation ou à l'utilisation de la
4903 technologie décrite dans le présent document, ni concernant l'étendue d'une licence qui, selon ces droits,
4904 serait disponible ou non. OASIS n'a tenté d'établir de tels droits d'aucune manière que ce soit. Le site
4905 Web d'OASIS renferme de l'information sur les procédures employées par l'Organisation pour faire
4906 appliquer les droits figurant dans ses spécifications. Il est possible de s'informer auprès du directeur
4907 exécutif d'OASIS par rapport aux revendications des droits ou aux licences mises à la disposition du
4908 public. Il est également possible de connaître auprès de lui le résultat de toute tentative effectuée par les
4909 implémenteurs ou utilisateurs de la présente spécification pour obtenir une autorisation ou une licence
4910 générale leur permettant d'avoir de tels droits propriétaires.

4911 OASIS invite les parties intéressées à lui signaler tout droit d'auteur, tout brevet ou toute demande de
4912 brevet, ou bien tout autre droit propriétaire qui pourrait couvrir une technologie pouvant être nécessaire à la
4913 mise en œuvre de la présente spécification. Les personnes intéressées sont priées de communiquer avec
4914 le directeur exécutif d'OASIS.

4915 **Droit d'auteur © Organisation pour l'avancement des standards d'information structurée [OASIS],**
4916 **2002. Tous droits réservés.**

4917 Le présent document et ses traductions peuvent être copiés et distribués. Tous travaux dérivés apportant
4918 des commentaires, des explications ou une aide pour sa mise en place peuvent également être élaborés,
4919 copiés, publiés et distribués, en totalité ou en partie, sans aucune restriction, à condition que la
4920 déclaration de droit d'auteur ci-dessus et le présent paragraphe soient inclus sur ces exemplaires et dans
4921 les travaux dérivés. Le présent document ne doit cependant pas être modifié de quelque façon,
4922 notamment par la suppression de la déclaration de droit d'auteur ou des références à OASIS, sauf pour
4923 des raisons de développement des spécifications OASIS, auquel cas les procédures pour le droit d'auteur
4924 définies dans le document sur les droits de propriété intellectuelle d'OASIS doivent être respectées ou, le
4925 cas échéant, traduites dans des langues autres que l'anglais.

4926 Les droits limités garantis ci-dessus sont perpétuels et ne seront révoqués ni par OASIS ni par ses
4927 successeurs ou cessionnaires.

4928 Le présent document et les renseignements qu'il contient sont fournis sur une base « TELLE QUELLE ». OASIS
4929 NE RECONNAÎT AUCUNE GARANTIE EXPRESSE OU LÉGALE QUI COMPREND, SANS
4930 TOUTEFOIS S'Y LIMITER, LA GARANTIE QUE L'UTILISATION DES RENSEIGNEMENTS PROPOSÉS
4931 NE COMPROMETTRA PAS DES DROITS OU DES GARANTIES LÉGALES DE COMMERCE OU
4932 L'ADÉQUATION À UN BUT DONNÉ.