

1

2

3

4

# **EbXML Registry Security Proposal**

6

## **Technical Architecture Security Team**

8

**May 10, 2001**

### **Status of this Document**

11

12 There are three categories of ebXML deliverables:

- 13     o Technical Specifications conform to the ebXML Requirements
- 14         document.
- 15     o Technical Reports are either guidelines or catalogues.
- 16     o White Papers constitute a snapshot of on-going work within a
- 17         Project Team.

18

19 This White Paper represents a report that has been approved by the  
20 Technical Architecture Security Team and has been accepted by the ebXML  
21 Steering Committee.

22

23 The material in this document constitutes a snapshot of on-going work  
24 within this Project Team.

25 Distribution of this document is unlimited.

26

27 ***This version:***

28 <http://www.ebxml.org/specs/secREG.pdf>

29

30 ***Latest version:***

31 <http://www.ebxml.org/specs/secREG.pdf>

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

---

## 47 **Authors**

48 o Krishna Sankar [ksankar@cisco.com]

49 o Farrukh Najmi [Farrukh.Najmi@east.sun.com]

50 **Contributors**

- 51 o Munter, Joel D [joel.d.munter@intel.com]
- 52 o Maryann Hondo [mhondo@us.ibm.com]
- 53 o Nieman, Scott [Scott.Nieman@NorstanConsulting.com]
- 54
- 55

56 **Abstract**

57

58 This document is a draft proposal whose purpose is to solicit additional  
59 input and convey the security aspects of the ebXML Registry.

60 **Referenced Documents**

61

62 EbXML Technical Architecture Risk Assessment [secRISK]

63

64

64

64 **Table of Contents**

65 1. Business Problem(s).....5  
 66 1.1. Authentication.....5  
 67 1.2. Integrity.....5  
 68 1.3. Confidentiality.....5  
 69 1.4. Authorization.....6  
 70 General.....6  
 71 2. Requirements.....7  
 72 3. ebXML Registry Security.....9  
 73 3.1. Security rules.....9  
 74 3.2. Security Info Model.....10  
 75 3.3. Security Info Model.....11  
 76 3.4. Security Processing.....13  
 77 3.4.1. Authentication.....13  
 78 3.4.2. Examine Transaction Rights on Object Request  
 79 (Authorization).....13  
 80 3.4.3. Registry Bootstrap.....13  
 81 3.4.4. Content Submission - processing done by the Registry Client  
 82 13  
 83 3.4.5. Content Submission - processing done Registry Service14  
 84 3.4.6. Content Delete/Deprecate - processing done by the Registry  
 85 Client 14  
 86 3.4.7. Content Delete/Deprecate - processing done Registry Service  
 87 14  
 88 4. Technical Details, DTDs, Business messages & Examples**Error! Bookmark not**  
 89 **defined.**  
 90 5. Issues & Ideas.....15  
 91 5.1. Issues.....15  
 92 5.2. Phase 2.....15  
 93  
 94  
 95

## 95 1. Business Problem(s)

96

97 Note: This version (0.003) is a very preliminary version. It is more an  
98 aggregation of the ideas and has an engineering bias. It needs a lot of  
99 rewrite to make it into a specification. Let us start from the ideas and ...

100 I am expecting comments from all with improvements and ideas.

101 How can we make this simpler yet extensible and secure ?

102

### 103 1.1. Authentication

104 The ebXML Registry is being used by businesses for various activities  
105 including publishing information, discovery, ad-hoc query, drill down etc.  
106 Authentication is required to identify the ownership of content as well as  
107 for identifying what "privileges" an entity can be assigned to with respect  
108 to the objects in the registry.

109 In addition, organizations might want to create private spaces for their  
110 partners and the access to these private spaces needs the authentication of  
111 users as well.

### 112 1.2. Integrity

113 The ebXML Registry is global and distributed, which contains information  
114 about capabilities, business process definitions and other XML documents.  
115 The integrity of the registry content is of great importance to those who  
116 refer to and use these documents for mission-critical business  
117 applications.

118 It is expected that most business registries do not have the resources to  
119 validate the voracity of the content submitted to them. The minimal  
120 integrity that the registry must provide is to ensure that content  
121 submitted by a Submitting Organization (SO) is maintained in the registry  
122 without any tampering en-route or within the registry. Furthermore, the  
123 registry should make it possible to identify the SO for any registry  
124 content unambiguously.

### 125 1.3. Confidentiality

126 The registry should provide capabilities for organizations to publish  
127 information, which are seen only by their partners. We cannot assume that  
128 all published information is public.

129 There should be capabilities to publish information to be viewed by a  
130 subset of users - for example the organization's partners.

131 There are two types of confidentiality needs.

132 1. "On the wire" confidentiality that ensures that content cannot be  
133 read on its way to the registry

134 2. "In registry" confidentiality that ensures that content is only  
135 visible to authorized parties (e.g. the partners of the SO)

#### 136 **1.4. Authorization**

137 An issue related to the confidentiality and integrity is the appropriate  
138 access to the data, or authorization. The information publishers should be  
139 able to define who can access and do what with their data. The registry should  
140 provide authorization mechanisms to achieve this.

#### 141 **General**

142 There need to be security around the registry as well as individual  
143 security around the documents.

144

## 144 2. Requirements

145

146 The ebXML Registry security requirements are derived from the business  
147 problems in the previous section:

148

- 149 1. The registry security system should have user level security
- 150 2. The registry also should have document level authorization security
- 151 3. The registry must support a set of default document level authorization  
152 security policies
- 153 4. The registry should allow the default document level authorization  
154 security policies to be customized by publisher of that document
- 155 5. The authorization policies (for example role based access control)  
156 should be granular to specify and limit access at the content (or  
157 object) level as well as at the operation (or method) level
- 158 6. The Registry Service should enforce access control policies when  
159 servicing client requests
- 160 7. All users who access the registry should be authenticated using standard  
161 schemes
  - 162 a. This does not preclude a guest level access which could be used by  
163 users who are not authenticated
  - 164 b. The guest level access, if present, should be the least secure  
165 mode
  - 166 c. The guest level access, if present, should not get any privileges  
167 by default, which means the default privilege should be no access  
168 to the guest level.
- 169 8. The main function of the authenticator is to ensure that only known  
170 entities can access the registry
- 171 9. The registry authentication service should be able to be boot-strapped  
172 (including adding credentials, profiles et al) in a secure way
- 173 10. The Registry authorization scheme should be able to provide, at a  
174 minimum, the following roles (REF : ISO/IEC 11179):
  - 175 a. RegistrationAuthority(RA) - Organization authorized to register  
176 data; usually the owner of the registry
  - 177 b. ResponsibleOrganization(RO) - Organization Responsible for the  
178 contents ; usually the one which signed the content
  - 179 c. Submitting Organization (SO) - One which submits content incl  
180 update, delete etc - ie one that has content submission and  
181 content life cycle management authorization ; this could be many  
182 entities including individuals and departments inside an  
183 organization
  - 184 d. Guest - a user who has some set of minimum capabilities

- 185 11. The authorization scheme should be flexible enough to have public and  
186 private areas within the registry
- 187 12. The security system should not prevent the registry from being a  
188 completely private registry
- 189 13. In order to avoid authentication for every message/interaction, a  
190 session based security scheme could be used
- 191 a. If a session-based scheme is used, the session should not be  
192 permanent.
- 193 b. It is RECOMMENDEED that the session time-out be configurable by  
194 the Registry Administrator
- 195 14. The security system should be able to prevent registry spoofing i.e.  
196 prevent an entity from posing as the intended registry when its not the  
197 intended registry
- 198 15. The security mechanism should be able to prevent the so-called "man-in-  
199 the-middle" attack, the "replay" attack and denial of service attack.
- 200 16. Messages between Registry clients and service need to be confidential
- 201 17. Registry content may be confidential and disclosed only to authorized  
202 parties
- 203 18. Contents may not be visible to registry if registry is not trusted or  
204 there is no need for the registry to see the contents.  
205 For example, if the content contains sensitive information like user  
206 names and passwords, the SO can encrypt the contents. They can still  
207 be kept in the registry but the registry would not be able to "see"  
208 them
- 209 Meta data is always visible to the registry.
- 210 19.  
211



211

212 **3. ebXML Registry Security**

213 **3.1. Security rules**

214 Release 1 will employ credential-based authentication (digital certificates  
 215 and signatures), simple default role based access control and message level  
 216 confidentiality and encryption.

217 These are the security rules, which will be implemented in Release 1.

- 218 • Authentication is required on a per request basis

219 Which means from a security point of view, all messages are independent;  
 220 there is no concept of a session or a long-standing conversation ; there  
 221 is the concept of a multi-message conversation

- 222 • Default Access Control Policies

- 223 o For Release 1, the philosophy is "Any known entity can publish and  
 224 anyone can view"

- 225 o So, the following roles will be built-in the registry:

226

Role	Default Permissions	ISO 11179 Cross Reference
ContentOwner	* implying all methods on ONE ManagedObject (full permissions to ONE object - the one the entity created)	Submitting Organization (SO)
RegistryAdministrator	* implying all methods on ALL ManagedObjects (full permissions to <b>ALL</b> objects in the Registry)	RegistrationAuthority(RA)
RegistryGuest	All getXXX methods on ALL ManagedObjects (read-only access to all content)	Guest
		ResponsibleOrganization(RO)  This is derived from the signature of the content. There are no specific registry permissions for the ResponsibleOrganization

227

- 228 • At the time of content submission, the registry will assign the default  
229 ContentOwner role to the Submitting Organization (SO) as authenticated  
230 by the credentials in the submission message
  - 231 o In Release 1 it will be the DN as identified by the certificate
- 232 • All requests performing sensitive operations are signed
  - 233 o Which means all non-getXXXX messages will need signature
- 234 • All content must be signed
- 235 • For Release 1, clients need not use certificates and will have the  
236 default RegistryGuest privileges
- 237 • Furthermore, in Release 1, the role based access control and access  
238 control policies are not visible outside the registry
  - 239 o Which means the clients will not be able to submit custom access  
240 control policies
  - 241 o In short, for Release 1 :
    - 242 ▪ The Registry Service by default establishes the access  
243 policies
    - 244 ▪ Only the SO and the Registry administrator have access to  
245 all methods and the clients can access the getXXX methods
    - 246 ▪ Anyone can publish content, but needs authentication
    - 247 ▪ Anyone can access the content and no authentication is  
248 required
- 249
- 250 • Release 1 will rely on TRP for message level authentication,  
251 confidentiality & integrity
- 252 • Registry is trusted to see all content
- 253 • There are no negative access control attributes
- 254

### 255 **3.2. Interaction with ebXML TRP**

256

257 The ebXML Registry security involves interactions with the message  
258 layer.

259 In case of ebXML TRP, the following interactions are involved:

260 a) Authentication

261 The TRP has the semantics and syntax for signing the message header.  
262 The registry will use the certificate DN from the signature to  
263 authenticate the user.

264 b) Integrity

265 The TRP has the semantics and syntax for signing the message  
 266 payload. All submitted contents should be signed (as defined in TRP)  
 267 and the Registry will store the signature as a part of the content.  
 268 When a client requests a content, the registry will also send the  
 269 signature. This way, the client can verify the integrity of the  
 270 content.

271 **3.3. Security Info Model**

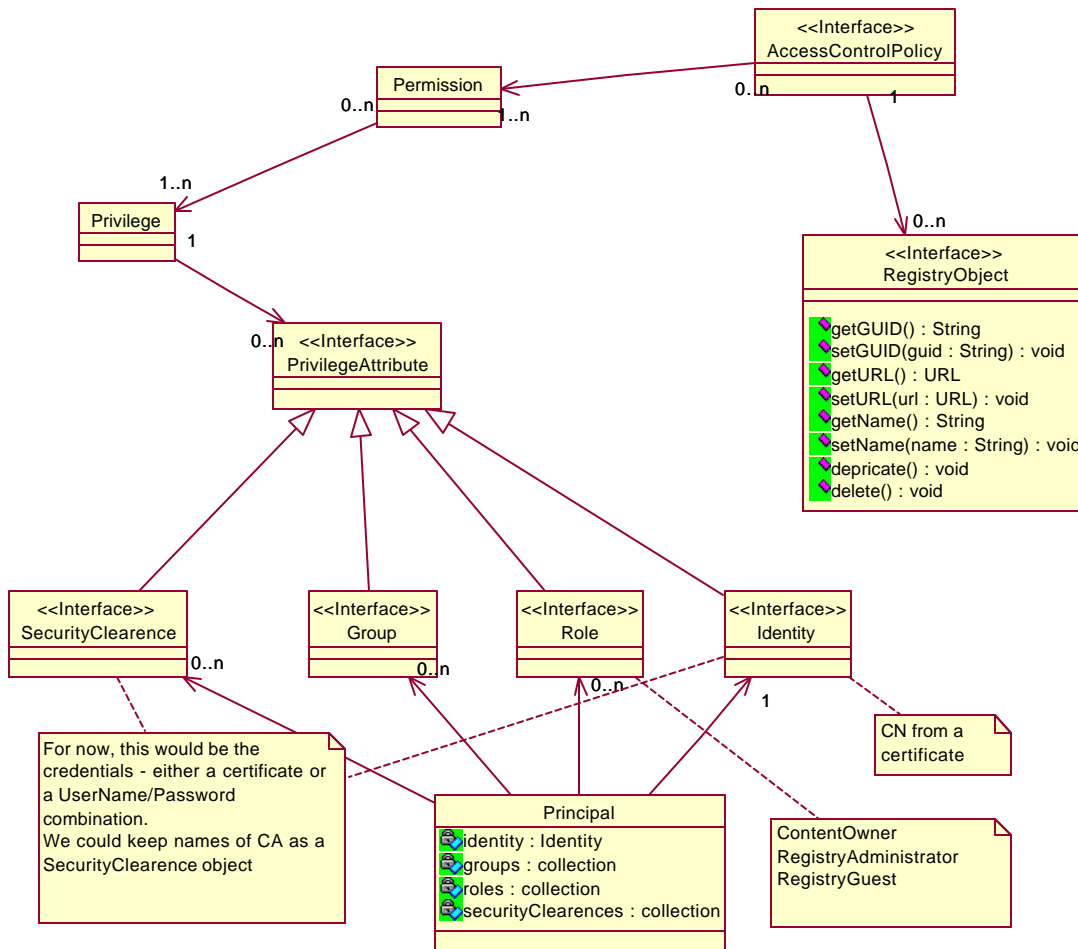
272

273 The security model is based on two goals - simplicity from a client's  
 274 point of view and extensibility for future enhancements.

275 The following figure shows the info model, which contains the security  
 276 related objects. The figure is for reference only. For more detail, please  
 277 refer to the Registry Information Model document.

278

279



280

281

282 The AccessControlPolicy is the the top-level security object. It ties together  
283 the permission object with an instance of a Registry object. The permission  
284 object also contains the methods (of the RegistryObject), which the privilege  
285 object can access.

286 Notes:

287 The actual method names are static and well known.

288 One permission Object is associated with one infoObject. However, an  
289 InforObject will be associated with many permission objects.

290 For example, each infoobject will be associated with three permission objects  
291 which have the attributes

292 {Role = RegistryAdministrator, methods = \*},

293 (Identity = <the DN of the SO>, methods = \*},

294 {Role = RegistryGuest, methods = "getGUID", "getName", "getURL" }

295 A privilege object contains many Privilege Attributes. A Privilege Attribute  
296 can be a Security Clearance, a group, a role, or an identity. This association  
297 enables one, the flexibility to have object access control policies based on a  
298 role, an identity or a group or a securityclearance or even better all of the  
299 above !

300 While privileges deal with groups, roles et al, the permissions deal with the  
301 methods of an object and tie them to privileges. The permission is an "and"  
302 operation (or a cumulative) . i.e. an entity can access the method of a  
303 RegistryObject only if it has all the privileges as detailed by the privilege  
304 object.

305 On the other hand, the AccessPolicy is an "or" operation. If an entity has  
306 "any" of the permissions, it can perform the method as detailed by the  
307 permission object.

308 An Identity usually is the DN in a certificate. It could be username/password  
309 as well.

310 The SecurityClearance object could keep the CA names, root certificates, et  
311 al. A SecurityClearance could be the traditional operations like Read, Create,  
312 Update, and Delete.

313 The group object is not used for now.

314 The role names are ContentOwner, RegistryAdministrator, RegistryGuest.

315 The Principal object is an entity, which has an identity, and optionally a set  
316 of role memberships, group memberships or security clearances. The  
317 authenticator will work against a principal.

318

### 319 **3.4. Security Processing**

320 This section provides a blueprint for how security processing may be  
321 implemented in the registry. It is meant to be illustrative not prescriptive.  
322 Registries may choose to have different implementations as long as they  
323 support the default security roles and authorization rules described in this  
324 document.

#### 325 **3.4.1. Authentication**

326

- 327 1. As soon as a message is received, the first work is the authentication.  
328 A principal object is created.
- 329 2. If the message is signed, it is verified (including the validity of the  
330 certificate) and the DN of the certificate becomes the identity of the  
331 principal. Then the Registry is searched for the principal and if found,  
332 the roles, groups and the securityclearances are filled in.
- 333 3. If the message is not signed, an empty principal is created with the  
334 role RegistryGuest. This step is for symmetry and to decouple the rest  
335 of the processing.
- 336 4. Then the message is processed for the command and the objects it will  
337 act on

338

#### 339 **3.4.2. Examine Transaction Rights on Object Request (Authorization)**

340 For every object, the access controller will iterate thru all the  
341 AccessControlPolicy objects with the object and see if there is a chain  
342 thru the permission objects to verify that requested method is permitted  
343 for the Principal. If any of the permission objects which the object is  
344 associated with has a common role, or identity, or group with the  
345 principal, the action is permitted.

#### 346 **3.4.3. Registry Bootstrap**

347 When a registry is newly created, a default Principal object should be  
348 created with the identity of the Registry Admin's certificate DN with a  
349 role RegistryAdmin. This way, any message signed by the Registry Admin will  
350 get all the privileges.

#### 351 **3.4.4. Content Submission - processing done by the Registry Client**

352 The Registry client has to sign the contents before submission - otherwise  
353 the content will be rejected.

**354 3.4.5. Content Submission - processing done Registry Service**

- 355 1. Like any other request, the client will be first authenticated. In this  
356 case, the Principal object will get the DN from the certificate.
- 357 2. As per the request in the message, the info Object will be created.
- 358 3. The next step is to create the default permission objects
- 359 a. If required, a permission object is created associating the  
360 RegistryObject methods with the Privilege object pointing to the  
361 RegistryAdministrator role with \* as the method name
- 362 b. An AccessControlPolicy object is created with the permission and  
363 the GUID of the new content.
- 364 c. If a principal with the identity of the SO is not available, an  
365 identity object with the SO's DN is created
- 366 d. A principal with this identity is created
- 367 e. A second permission object is created associating this identity  
368 with the with \* as the method name
- 369 f. A third permission object is created associating the RegistryGuest  
370 role with the with the getName, getURL and getUID as the method  
371 names
- 372 g. Then two more AccessControlObjects are created tying in all the  
373 permission objects with the GUID of the newly created object

**374 3.4.6. Content Delete/Deprecate - processing done by the Registry Client**

375 The Registry client has to sign the payload (not entire message) before  
376 submission, for authentication purposes; otherwise, the request will be  
377 rejected

**378 3.4.7. Content Delete/Deprecate - processing done Registry Service**

- 379 1. Like any other request, the client will be first authenticated. In this  
380 case, the Principal object will get the DN from the certificate. As  
381 there will be a principal with this identity in the Registry, the  
382 principal obj will get all the roles from that object
- 383 2. As per the request in the message (delete or deprecate), the appropriate  
384 method in the info Object will be accessed.
- 385 3. The access controller performs the authorization by iterating thru the  
386 permission objects associated with this object
- 387 4. As the Registry had created an AccesssControlPolicy object which has the  
388 permission object associating this identity and with the method names \*,  
389 the action will be permitted.

390

391

392

393

## 393 5. Issues & Ideas

### 394 5.1. Issues

395

- 396 o Trust relationship between distributed registries - Not on Release 1
- 397 o Session and auth tokens exchange - Not in Release 1
  - 398 o Session based interaction
  - 399 o Sessions as short-lived certificates (?)
- 400 o Do we need a userid/password based authentication or can a certificate
- 401 based authentication suffice - No
- 402 o Should we allow Object retrieval via HTTP GET?
- 403 o How to deal with expiration of a certificate associated with submitted
- 404 content
- 405 o What objects are persistent and which are transient. It is hard to grasp
- 406 when the security objects, like permissions or principals are created
- 407 and when they go away (which can be a security issue in itself).
- 408 o Develop a CPP for this. The CPP could define the different roles and
- 409 also demonstrate the security needed at each level....for example the
- 410 "reader" role would not need any security on its request message, as
- 411 opposed to the "document owner" role needing authentication. Then we
- 412 will abstract the security interactions to different roles and provide a
- 413 CPP for it.

414

### 415 5.2. Phase 2

416

- 417 o Define interface to submit custom Access Control Policies
- 418 Identity and Role based authorization
- 419
- 420 o Registry may not be trusted to view all content
- 421 o Trust relationship between distributed registries
- 422 o Session and auth tokens exchange
  - 423 o Session based interaction
  - 424 o Sessions as short-lived certificates
- 425 o Do we need a userid/password based authentication or can a certificate
- 426 based authentication suffice?
- 427 o

428

429

429 **Copyright Statement**

430 Copyright © UN/CEFACT and OASIS, 2001. All Rights Reserved.

431

432 This document and translations of it MAY be copied and furnished to others,  
433 and derivative works that comment on or otherwise explain it or assist in its  
434 implementation MAY be prepared, copied, published and distributed, in whole or  
435 in part, without restriction of any kind, provided that the above copyright  
436 notice and this paragraph are included on all such copies and derivative  
437 works. However, this document itself MAY not be modified in any way, such as  
438 by removing the copyright notice or references to ebXML, UN/CEFACT, or OASIS,  
439 except as required to translate it into languages other than English.

440

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

443

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

449