



XRI Requirements and Glossary

Version 1.0 – 12 June 2003

Document identifier:

xri-requirements-and-glossary-v1.0

Location:

<http://www.oasis-open.org/committees/xri/spec/>

Editors:

Gabe Wachob, Visa International <gwachob@visa.com>
Drummond Reed, OneName <drummond.reed@onename.com>
Marc LeMaitre, OneName <marc.lemaitre@onename.com>
Dave McAlpin, Epok <dave.mcalpin@epokinc.com>
Davis McPherson, Epok <davism@epokinc.com>

Abstract:

This document describes architectural motivations and requirements for development of the Extensible Resource Identifier (XRI) specifications. It also includes a normative glossary of terms used in this document and other XRI deliverables.

Status:

This document is a committee requirements specification. It may be updated periodically on no particular schedule. Send comments to the editors.

Committee members should send comments on this specification to the xri@lists.oasis-open.org list. Others should subscribe to and send comments to the xri-comment@lists.oasis-open.org list. To subscribe, send an email message to xri-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

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

30 Table of Contents

31	1	Introduction	4
32	1.1	Terminology.....	4
33	2	Motivations.....	5
34	2.1	Introduction.....	5
35	2.2	Persistent Identification	7
36	2.3	Human-Friendly vs. Machine-Friendly Identification	7
37	2.4	Cross-Context Identification	9
38	2.5	Resource Attribute and Version Identification	10
39	2.6	Delegation, Federation, and Extensibility.....	10
40	2.7	Security and Privacy	11
41	3	XRI Syntax Requirements.....	12
42	3.1	URI and URN Requirements.....	12
43	3.1.1	URI Conformance	12
44	3.1.2	URN Conformance.....	12
45	3.2	Abstraction and Independence	12
46	3.2.1	Location-Independence	12
47	3.2.2	Application-Independence	13
48	3.2.3	Transport-Independence	13
49	3.2.4	Type-Independence	13
50	3.2.5	Security Method-Independence.....	13
51	3.3	Persistent Identification	13
52	3.3.1	Persistent Identifiers	13
53	3.3.2	Reassignable Identifiers	13
54	3.3.3	Combining Persistent and Reassignable Identifiers	14
55	3.4	Human-Friendly and Machine-Friendly Identification.....	14
56	3.4.1	Human-Friendly Identifiers (HFIs).....	14
57	3.4.2	Machine-Friendly Identifiers (MFIs)	14
58	3.4.3	Combining HFIs and MFIs	14
59	3.4.4	Identifier Mapping	14
60	3.4.5	Explicit Non-Resolvability	14
61	3.4.6	Internationalization.....	14
62	3.4.7	Character Encoding	15
63	3.5	Cross-Context Identification	15
64	3.5.1	Cross-References	15
65	3.5.2	URLs as Cross-References	15
66	3.6	Attribute and Version Identification	15
67	3.6.1	Attribute Identification	15
68	3.6.2	Version Identification.....	15

69	3.7 Authority, Delegation, Federation, & Extensibility	16
70	3.7.1 Unlimited Root Authorities	16
71	3.7.2 Unlimited Topologies	16
72	3.7.3 Unlimited Delegation and Federation	16
73	3.7.4 Scheme Extensibility.....	16
74	3.7.5 Specializations	16
75	3.8 Data Protection and Security	16
76	3.8.1 Identifier Security	16
77	3.8.2 Identifier Privacy	17
78	4 XRI Resolution Requirements	18
79	4.1 Non-Resolvability	18
80	4.2 Semantic Mapping.....	18
81	4.3 Resolution Mechanism-Independence.....	18
82	4.4 Internet Resolution Mechanism.....	18
83	4.5 Unlimited Federation	18
84	4.6 Interoperability of Specializations.....	18
85	4.7 Scalability	18
86	4.8 Redundancy	18
87	4.9 Trusted Resolution	19
88	4.10 Proxy Resolution	19
89	5 Glossary.....	20
90	5.1 Normative Glossary.....	20
91	5.2 Informative Glossary	25
92	6 References.....	26
93	Appendix A. Acknowledgments	27
94	Appendix B. Notices	28
95		

96 1 Introduction

97 This document is divided into four major sections:

- 98 ▪ *Motivations* describes why the XRI TC was chartered and the major problems the XRI
99 specifications are intended to address.
- 100 ▪ *XRI Syntax Requirements* enumerates the requirements for the XRI URI scheme.
- 101 ▪ *XRI Resolution Requirements* enumerates the requirements for XRI resolution.
- 102 ▪ *Glossary* contains a listing of the key terms used in this document and the rest of the XRI
103 TC deliverables.

104 1.1 Terminology

105 The key words **must**, **must not**, **required**, **shall**, **shall not**, **should**, **should not**, **recommended**,
106 **may**, and **optional** in this document are to be interpreted as described in IETF RFC 2119
107 **[Keywords]**.

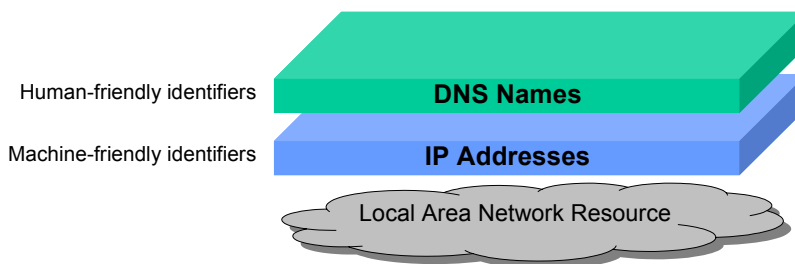
108 Other terms used in this document are defined in the Glossary (Section 5).

109

110 2 Motivations

111 2.1 Introduction

112 Internet architecture today is based primarily on two layers of identifiers, as shown in Figure 1:
113



114

115 *Figure 1: The two layers of Internet identifiers in predominant use today.*

116 The first layer, IP (Internet Protocol) addressing, defines the Internet itself. IP was developed to
117 standardize packet exchange between local area networks, a task that required a layer of globally
118 unique identifiers for every network segment and host. Since the goal was highly efficient packet
119 routing, IP addresses were designed to be very machine-friendly—a series of decimal numbers
120 (IPv4) or hex characters (IPv6) representing fixed-byte addressing segments.

121
122

```
172.14.206.73  
:AE46:83F2::9B15:2287
```

123 A second layer, the DNS (Domain Name System), was subsequently developed to provide a
124 name service for IP hosts. This abstraction layer solved two problems: a) it provided human-
125 friendly identifiers for IP-addressable hosts, making them much easier for people to remember
126 and use, and b) it allowed Internet hosts or users to have a logical identity that transcended a
127 particular IP address.

128
129

```
www.example.com  
mary.smith@example.com
```

130 These two layers of identifiers, when combined with local area network identifiers, can uniquely
131 identify any resource on the Internet. Tim Berners-Lee and other architects took full advantage of
132 this when creating the World Wide Web. They developed an identifier syntax originally called URL
133 (Uniform Resource Locator) and now called URI (Uniform Resource Identifier)¹ that allowed a
134 combination of DNS names, IP addresses, and local identifiers to serve as a hyperlink between
135 resources. The syntactic rules for URI schemes (e.g., HTTP URIs, FTP URIs, email URIs, etc.)
136 were most recently specified in IETF RFC 2396 in August 1998 **[URI]**.²

137
138

```
http://www.example.com/pages/products/widget.html  
mailto:mary.smith@example.com
```

139 The phenomenal success of the Web meant that URIs became the fastest-growing new address
140 in history. As the Web grew, it encountered the problem of links breaking because the resource

¹ The term "URL" is no longer in use by the IETF and W3C. See IETF RFC 3305, "Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations".

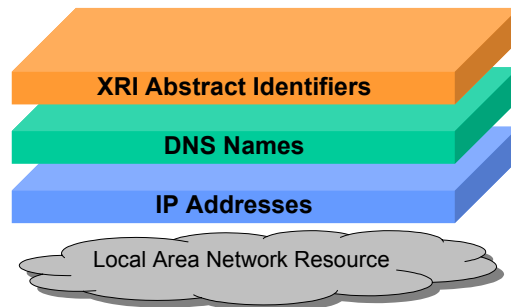
² A revision to the URI specification, RFC2396bis, is under preparation by Roy Fielding. See <http://www.apache.org/~fielding/uri/rev-2002/rfc2396bis.html>.

141 referenced by a URI changed its location on the network. Berners-Lee and others recognized that
142 solving this problem would require another level of abstraction—a layer of persistent URIs that
143 would remain the same even when the resources they referenced changed their locations. They
144 called this new type of location-independent identifier a URN (Uniform Resource Name). The URI
145 scheme for URNs was specified by IETF RFC 2141 in May 1997 [URN].³

```
146 urn:uuid:c2f41010-65b3-11d1-a29f-00aa00c14882  
147 urn:isbn:0-395-36341-1
```

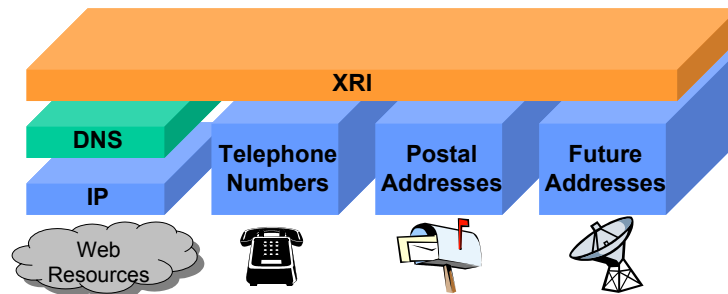
148 Since the completion of the IETF URN work, a number of new technologies have appeared for
149 modeling human semantics and data exchange relationships over the Internet, including the
150 Semantic Web, Topic Maps, Web services, digital identity, and digital rights management. While
151 many of these technologies require persistent identifiers, they have also generated a number of
152 other new requirements for abstract identifiers that are not addressed by URNs. These
153 requirements form the primary motivations for XRIs as discussed in the following sections.

154 The overall goal of the XRI specifications is to establish a standard syntax and resolution protocol
155 for fully abstract identifiers—in short, to enable a third layer of Internet identifiers similar to the
156 DNS naming and IP addressing layers that exist today, as shown in Figure 2:
157



158
159 *Figure 2: XRIs are designed to provide a uniform third layer of abstract identifiers for Internet resources.*

160 The potential for this new layer goes beyond just the Internet. With the growing convergence of
161 the Web with other networks such as wired and wireless phone networks, satellite networks,
162 package delivery networks, etc., an XRI can serve as a true *unified address*—a single abstract
163 identifier that can be resolved (with the appropriate data protections) to any concrete address or
164 attribute associated with the target resource. Unified addresses represent an enormous potential
165 savings in labor—both in people spending time looking up phone numbers, fax numbers, email
166 addresses, etc., and in developers spending time coding and testing routines to locate and verify
167 the current address of a target resource.
168



169
170 *Figure 3: XRIs can serve as true unified addresses across all communications networks.*

³ The full scope of the IETF URN work is summarized at <http://www.ietf.org/html.charters/urn-charter.html>.

171 2.2 Persistent Identification

172 As discussed above, the original motivation for a new layer of abstract identifiers was the need for
173 persistence—the ability to for an identifier to maintain its association with a resource independent
174 of the resource's current location on the network. The requirements for persistent identifiers—
175 URNs—were set forth by the IETF in RFC 1737 [URNReqs].

176 The IETF URN specification [URN] requires *absolute persistence*, i.e., that the entire identifier
177 never be reassigned to another resource for all time. The IETF recognized that such a
178 requirement can be difficult to enforce operationally, since it depends on factors that are not
179 technical in nature (the longevity and business practices of the identifier authority, for example).

180 In practice, many identifiers need only *relative persistence* in one of two ways. First, persistence
181 be required within the context of a top-level authority which may itself have a reassignable
182 identifier such as a DNS name or IP address. This is the case for many URIs within large
183 database-driven web sites.

```
184 http://www.someportal.com/s/19821  
185 http://somenews.com/2010-1071-998513.html  
186 http://www.somestore.com/exec/tg/browse/-/1/002-9387661-7480836
```

187 Secondly, persistence may only be needed for a relative period of time. Even very long-lived
188 identifiers may be reassigned, particularly in fixed address spaces. As a general rule, the
189 frequency of reassignment varies with the type and purpose of the identifier. Postal addresses,
190 for example, are usually very long-lived, lasting for decades or even centuries. By contrast phone
191 numbers and DNS domain names both have typical registration cycles of from one to ten years.
192 At the other end of the spectrum IP addresses may (especially in the case of dynamic IP
193 assignment mechanisms like DHCP) be reassigned to a different computer every online session.

194 Persistence can thus be viewed along a gradient from absolute to relative, and XRI syntax and
195 resolution mechanisms should be designed to accommodate this gradient.

196 *Supporting both absolute and relative persistent identifiers is a key*
197 *motivation of the XRI specifications.*

198 2.3 Human-Friendly vs. Machine-Friendly Identification

199 A second key property of abstract identifiers is their human-friendliness. By this, we mean the
200 ability of a human being to understand, remember, and use an identifier, vs. the ability of a
201 machine to efficiently resolve, cache, and process it. Perhaps the best example of these two
202 polarities is DNS names and IP addresses. DNS names are typically very semantically reflective
203 of the resource they represent.

```
204 www.yahoo.com  
205 www.ibm.com/products  
206 mary.smith@hotmail.com
```

207 IP addresses are just the opposite – they are pure numeric or hexadecimal strings which are
208 generally not semantically reflective of the resource they represent.

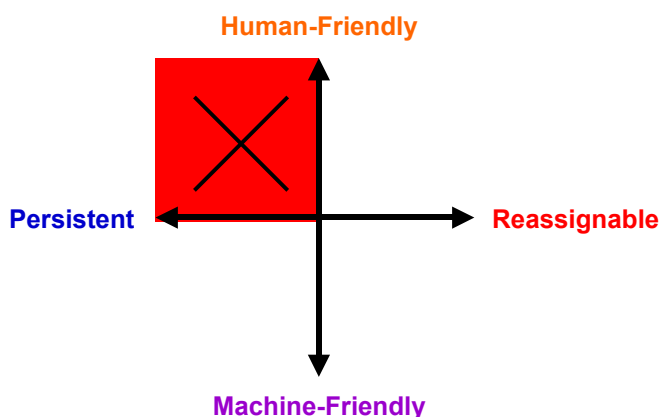
```
209 172.14.206.73  
210 :AE46:83F2::9B15:2287
```

211 As with persistent vs. reassignable identifiers, there is a continuous gradient between human-
212 friendly identifiers (HFIs) and machine-friendly identifiers (MFIs). In fact many composite
213 identifiers, such as postal addresses, are typically a mixture of both HFI and MFI components.⁴

214 `Mary Smith, 4216 Corliss Ave North, Seattle WA 98133-8914`

215 The relationship of the HFI/MFI gradient and the persistent/reassignable gradient can be
216 visualized by the following graph:

217



218

219 *Figure 4: The relationship of the persistent/reassignable gradient and the HFI/MFI gradient.*

220 What this graph illustrates is that while an abstract identifier may theoretically fall anywhere in the
221 spectrum above, in practice there is one quadrant where the two requirements conflict—the
222 intersection of persistent identifiers and HFIs.

223 The reason has nothing to do with technology and everything to do with the nature of human
224 language. People are forever reassigning the meaning of words, names, and phrases. A filename
225 assigned by a user to one file today may be reassigned to another file tomorrow. A domain name
226 registered to one website this month may be reregistered to another next month. A trademark
227 registered by one company this year could be sold to another the next. At the highest level, this
228 constant redefinition of semantic identifiers manifests itself as the slow "semantic drift" of entire
229 languages—the primary reason many dictionaries are republished every year.

230 Semantic drift at any speed makes it difficult for HFIs to remain persistent. This is why most
231 persistent identifiers tend to be partially or entirely MFIs—strings of numbers or "nonsense
232 characters" that are unique but do not carry semantic meaning. Some URN systems, being the
233 most persistent identifiers of all, are excellent examples.

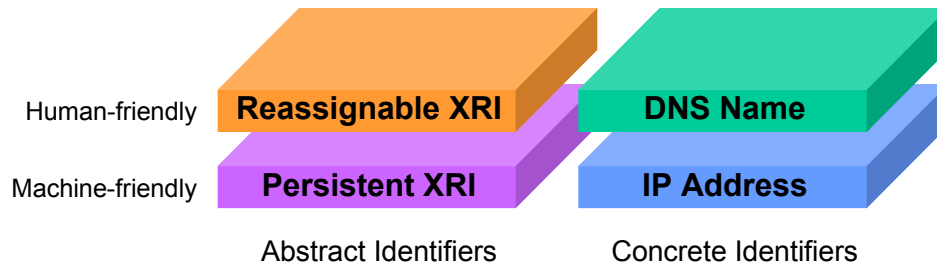
234 `urn:uuid:c2f41010-65b3-11d1-a29f-00aa00c14882`
235 `urn:isbn:0-395-36341-1`
236 `urn:ietf:rfc:2396`

237 Because of this inherent conflict between persistent and human-friendly identifiers, a second key
238 requirement of XRIs is that:

- 239 a. They must support any combination of persistent and reassignable HFIs and MFIs, and
240 b. When a resource needs both a reassignable HFI and a persistent MFI, the XRI specifications
241 must allow the former to be resolved to the latter.

⁴ From an evolutionary standpoint, most early postal addresses consisted entirely of HFI components such as personal names, city names, state/province names, and country names. MFI components including routing numbers and postal codes were added later to support automated mail handling equipment.

242 This second scenario, called *semantic mapping*, mirrors the same two-layer model for abstract
 243 identifiers that DNS names and IP addresses provide for concrete identifiers as shown in Fig. 5.
 244



245
 246 *Figure 5: XRIs can map reassignable HFIs to persistent MFIs the same way DNS names are mapped to IP*
 247 *addresses.*

248 Semantic mapping can solve a wide range of problems relating to human usability of network
 249 resources, ranging from smarter search technologies and simpler security systems to more
 250 intelligent user interfaces and natural language translation applications.

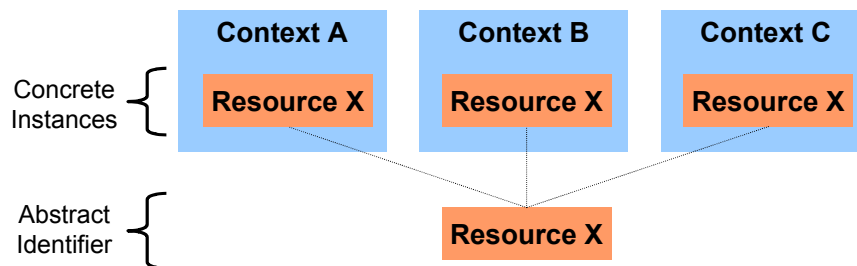
251 *Providing a unified syntax for both HFIs and MFIs and semantic mapping*
 252 *between the two is a key motivation of the XRI effort.*

253 2.4 Cross-Context Identification

254 Another of the key advantages of fully abstract identifiers is that they are very useful for
 255 identifying resources that may have multiple concrete representations in different network
 256 locations. To borrow a real-world example, the English language concept of "President" has a
 257 concrete representation in many different companies. In fact a postal letter can usually be
 258 addressed to the president of a company simply by using the abstract identifier, "President,
 259 [postal address of company]".

260 Yet this same generalization is the exception rather than the rule with network resources. To be
 261 sure, some username conventions like "postmaster", "info", "sales", or "support" are commonly
 262 used to route email messages to those well-known functions of an organization. But few such
 263 conventions exist for Web resources beyond the DNS server for a website having the name
 264 "www" or the home page of a web site having the name "index.htm" or "index.html".

265 It can be very useful to have a standard way of identifying logically equivalent resources across
 266 multiple physical contexts—for example, being able to locate the same file stored on multiple file
 267 servers, or the same invoice stored in multiple accounting systems. It would enable program-
 268 matic querying, indexing, and manipulation of these resources to a much higher degree of
 269 precision that is available today through keyword and other natural language search techniques.
 270



271
 272 *Figure 6: Sharing an abstract identifier across multiple concrete contexts.*

273 Note that doing this in a uniform manner requires a URI syntax that permits combining an
 274 abstract resource identifier created in one context (e.g., a dictionary or taxonomy authority) with a

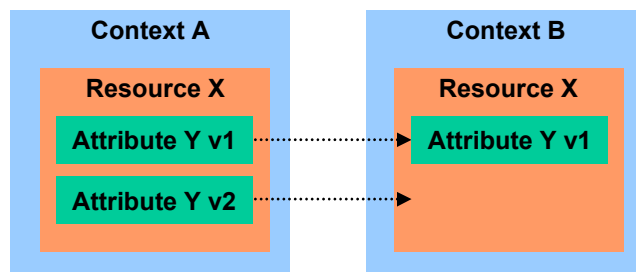
275 concrete identifier that establishes the local context—exactly like combining the term "President"
276 with a concrete postal address. For example, an XRI representing a concept such as
277 "management team" could be combined with the URI of a home page to form a well-known
278 address for this type of resource on any corporate website. This would enable the development of
279 much smarter and more specialized spiders than crawl the Web today.

280 *Providing a standard means for identifying the same abstract resource*
281 *across different concrete contexts is a key motivation of the XRI effort.*

282 2.5 Resource Attribute and Version Identification

283 A corollary to the need for cross-context identification is the need to establish the equivalence of
284 different concrete resources that correspond to the same abstract resource identifier. This is the
285 classic problem of consistency and data synchronization. Solving this problem in a general
286 manner requires not only sharing the same identifier for the abstract resource as a whole, but: a)
287 being able to identify the attributes of the resource down to the lowest level at which consistency
288 is to be maintained, and b) being able to unambiguously identify each version of an attribute.

289 A common example of this problem is the long-promised electronic business card. If an individual
290 were to share copies of an electronic business card with 100 contacts, the same logical resource
291 might be stored in 100 electronic address books somewhere on the network. If the owner of the
292 business card updated a phone number, this new value would need to be synchronized with all
293 100 copies. To do so at the level of the phone number attribute (rather than the entire business
294 card) requires the ability to identify this specific attribute and version.
295



296

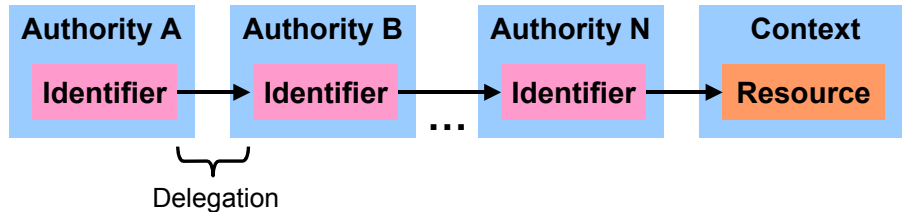
297 *Figure 7: The need to standardize attribute and version identification.*

298 RFC 2396 [URI] establishes a standard delimiter for addressing a fragment of a resource in a
299 URI using the # character. However it does not specify any syntax for expressing either nested
300 attributes or versions. This requires that every web site adopts its own local convention for
301 accomplishing this, a significant hindrance to interoperability when data sharing is desired.

302 *Providing a standard means for addressing attributes and versions of a*
303 *resource is a key motivation of the XRI effort.*

304 2.6 Delegation, Federation, and Extensibility

305 The success of the IP addressing and DNS naming layers has been largely due to their
306 delegation models. Each requires a minimum of centralized control and permits delegation of
307 identifier authority to any depth. These same requirements must be extended to any abstract
308 identifier layer designed for Internet-scale deployment.
309



310

311

Figure 8: The need for delegation to arbitrary depth.

312 The primary effect of this requirement is that the syntax for XRI must be generalized enough to
 313 support extensions or restrictions at any level of delegation. This is similar to the same
 314 generalized purpose of URI syntax: it established a common syntax for all identifier schemes on
 315 the World Wide Web while permitting any number of more specialized schemes to be developed.

316

317

Providing a means for unlimited delegation, federation, and extension of abstract identifiers is a key motivation of the XRI effort.

318 2.7 Security and Privacy

319 A final motivation is the critical role that identifiers play in security and privacy. Security
 320 requirements frequently (but not always) require being able to verify the identity of the parties to a
 321 transaction. At the same time, privacy requirements frequently (but not always) require that the
 322 parties not disclose any more information than necessary for the transaction.

323 With respect to security, XRIs offer the ability to establish and maintain persistent identifiers that
 324 can be essential to maintaining long-term trust relationships. With respect to privacy, XRIs offer
 325 the ability to be free of personally-identifiable information (particularly when compared with the
 326 frequent use of email addresses as personal identifiers).

327 Therefore XRIs can play a key role in supporting both the security and privacy requirements of
 328 many new next-generation Internet applications, particularly those modeling digital identity and
 329 human relationships over the net.

330

331

Supporting the security and privacy requirements of Internet trust infrastructure is a key motivation of the XRI effort.

332 3 XRI Syntax Requirements

333 Following are the requirements for the XRI URI scheme specification.

334 3.1 URI and URN Requirements

335 3.1.1 URI Conformance

336 The XRI specifications for all identifiers must conform to the URI specification documented in
337 IETF RFC 2396 [URI] after canonical transformation. These requirements include:

- 338 • BNF definition of an identifier scheme
- 339 • Hierarchical namespace with absolute and relative identifier forms
- 340 • Global transcribability
- 341 • Escaping and character encoding rules
- 342 • Naming authorities
- 343 • Query components
- 344 • Fragment identifiers

345 Note that this requirement may be modified by 3.4.6, Internationalization, in which the XRI TC
346 shall evaluate whether XRI syntax should be conformant with the Internationalized Resource
347 Identifier specification [IRI].

348 3.1.2 URN Conformance

349 The XRI specifications for an absolute persistent identifier, or for the persistent segments of a
350 relative persistent identifier, must conform to the URN Functional Requirements specificaton,
351 IETF RFC 1737 [URNReqs]. To the greatest extent possible without conflicting with other XRI
352 syntax requirements, they should also conform to the URN Syntax specification, IETF RFC 2141
353 [URN]. These requirements include:

- 354 • Global scope: A URN is a name with global scope which does not imply a location. A
355 URN must be capable of naming any resource in the universe and must have the same
356 meaning everywhere.
- 357 • Global uniqueness: The same URN will never be assigned to two different resources.
- 358 • Persistence: URNs must support identifiers that have a permanent lifetime.
- 359 • Scalability: URNs can be assigned to any resource that might conceivably be available
360 on or off the network, for hundreds of years.
- 361 • Legacy support: URNs must support existing legacy naming systems
- 362 • Extensibility: The URN scheme must permit future extensions to the scheme.
- 363 • Independence: It is solely the responsibility of a naming authority to determine the
364 conditions under which it will issue a URN.
- 365 • Resolution: The URN scheme must support resolution of the URN. However not all
366 URNs must be resolvable.

367 3.2 Abstraction and Independence

368 3.2.1 Location-Independence

369 The XRI specifications must enable identifiers to identify a resource independent of its location on
370 the network (if any). At various points in its lifetime, the same resource may or may not be

371 accessible on the network. It may migrate from one network endpoint to another. It may move
372 from one machine to another. It may be retired and be permanently inaccessible. Consequently, it
373 must be possible to construct a valid XRI that does not reflect an identified resource's location on
374 the network (or lack thereof).

375 **3.2.2 Application-Independence**

376 The XRI specifications must enable identifiers to identify a resource independent of an application
377 that creates or manages the resource. X.500 distinguished names, for example, identify directory
378 entries managed within the X.500 tree but may not be portable to other directory systems. By
379 contrast, most file naming systems are application independent.

380 **3.2.3 Transport-Independence**

381 The XRI specifications must enable identifiers to identify a resource independent of the protocol
382 used to access it on the network (if any). HTTP URLs, for example, are intrinsically bound to a
383 specific transport protocol. While an explicit binding to a transport protocol is appropriate for some
384 URIs, it should be possible to construct XRIs that are completely disassociated from the
385 mechanism or protocol used to transport data representing the identified resource.

386 **3.2.4 Type-Independence**

387 The XRI specifications must enable identifiers to identify a resource independent of the resource
388 type. URIs are often used for purposes beyond the intent of their schemes. HTTP URLs, for
389 example, are frequently used as opaque, unique identifiers that have no association with
390 hypertext (as designators of XML namespaces, for instance). Websites often repurpose email
391 addresses as login IDs, treating them as ubiquitous, globally unique mnemonic handles but
392 completely changing the mailto scheme's intention of identifying an electronic mailing address
393 **[Mailto]**. It is the intent of XRIs to provide an abstraction that allows resources to be identified
394 without respect to their underlying, concrete type. Therefore, XRIs must allow identifiers that have
395 no expressed association with the type of resource they identify.

396 **3.2.5 Security Method-Independence**

397 The XRI specifications must support the ability for an identifier to identify a resource independent
398 of the authentication, authorization, or access control technologies or methods used to ensure the
399 security of the data associated with the resource. An example of where this is not done is SPKI
400 **[SPKI]** and other PKI technologies where the ID is a public key. While XRI syntax should be
401 sufficiently expressive to include security attributes of a resource if desired, it must not be
402 required.

403 **3.3 Persistent Identification**

404 **3.3.1 Persistent Identifiers**

405 The XRI specifications must support the ability for the association between an identifier and the
406 resource it identifies to persist for any desired period of time regardless of changes to the
407 resource, its attributes, or its location on the network (if any). In the case of absolute persistent
408 identifiers this means the identifier will never be reassigned even if the resource becomes
409 unavailable or is no longer in existence.

410 *Note:* This requirement is mutually exclusive with 3.3.2, Reassignable Identifiers.

411 **3.3.2 Reassignable Identifiers**

412 The XRI specifications must support the ability for the association between an identifier and the
413 resource it identifies to be changed. This is often desirable when an identifier semantically reflects

414 characteristics of the resource it identifies. Because semantics and semantic relationships
415 change over time, it should be possible to accommodate these changes within the resource's
416 identifier. For example, if a trademark or trade name is sold from one company to another, it
417 should be possible for identifiers previously associated with resources at the original company to
418 be reassigned to different resources at the new company.

419 *Note:* This requirement is mutually exclusive with 3.3.1, Persistent Identifiers.

420 **3.3.3 Combining Persistent and Reassignable Identifiers**

421 The XRI specifications must enable a composite identifier to combine both persistent identifiers
422 and reassignable identifiers as components of a single expression. In addition, the XRI syntax
423 must be able to: a) distinguish between these two types of components within the expression,
424 and b) permit a portion of the expression to be non-authoritative, e.g., a human-readable
425 comment.

426 **3.4 Human-Friendly and Machine-Friendly Identification**

427 **3.4.1 Human-Friendly Identifiers (HFIs)**

428 The XRI specifications must support the ability to create identifiers optimized for human
429 readability, memorability, and usability.

430 **3.4.2 Machine-Friendly Identifiers (MFIs)**

431 The XRI specifications must support the ability to create identifiers optimized for machine and
432 network efficiency.

433 **3.4.3 Combining HFIs and MFIs**

434 The XRI specifications must enable a composite identifier to combine both HFIs and MFIs as
435 components of a single expression.

436 **3.4.4 Identifier Mapping**

437 The XRI specifications must support the ability for an identifier itself to serve as a resource. The
438 practical effect of this requirement is that it must be possible for an XRI to resolve to an XRI.
439 Consequently, resolution of an XRI may be indirect and iterative. See also 4.2.

440 **3.4.5 Explicit Non-Resolvability**

441 The XRI specifications must support the ability for the syntax of an identifier to express that it is
442 non-resolvable, i.e., that a resolver should not even attempt to resolve it. Such identifiers are
443 useful primarily for cross-references. See Sections 2.4 and 3.5.

444 **3.4.6 Internationalization**

445 The XRI specifications must support the ability for semantic identifiers to represent resources in a
446 user's native language and scripts. The XRI specifications shall follow the guidelines in RFC 2718
447 **[Guidelines]** concerning internationalized character set usage in URIs. They shall further
448 constrain the allowable characters to those valid in XML documents as specified by XML 1.0
449 **[XML]**. The TC shall also evaluate whether XRI syntax should conform to the Internationalized
450 Resource Identifier specification **[IRI]**.

451

452 **3.4.7 Character Encoding**

453 The XRI specifications must use UTF-8 as the encoding mechanism for internationalized
454 character sets. See the IETF Policy on Character Sets [IETFCharsets]. Also see [URI], [IRI], and
455 [Unicode].

456 **3.5 Cross-Context Identification**

457 **3.5.1 Cross-References**

458 The XRI specifications must support the ability to use an identifier in the context of another
459 identifier, i.e., for an XRI to be contained within another XRI). The ultimate resolution (if it is
460 resolvable) of the containing XRI and all contained parts must be the responsibility of the
461 resolvers in the context of the naming authority of the containing XRI. The contained XRI must
462 resolve (if it is resolvable) on its own according to the XRI resolution mechanism. Example:

463 An ISBN XRI contained within an XRI for an order processing form at
464 bookstore.com might resolve the complete XRI to a web page describing
465 the book and providing shopping cart functions to buy the book, put it
466 on a wish list, etc. But the same ISBN XRI contained with a NY Times
467 XRI might resolve to a Web page containing the review of the book in
468 last Sunday's book review section. Meanwhile at the Library of
469 Congress, the same ISBN XRI entered into the card catalog system might
470 resolve to the location in the stacks where copies of the book will be
471 found.

472 **3.5.2 URIs as Cross-References**

473 The XRI specifications must enable other URIs to be one type of identifier that can be used as a
474 cross-reference. In other words, cross-references as defined in 3.5.1 must not be limited to other
475 XRIs, but may contain any fully qualified URI.

476 **3.6 Attribute and Version Identification**

477 **3.6.1 Attribute Identification**

478 The XRI specifications must support the ability to identify attributes of a resource, including
479 nested attributes (attributes that contain other attributes).

480 As noted in the Normative Glossary, an attribute is defined as any data, metadata, or resource
481 that can be identified only in the context of a specific resource, e.g., the age of a person, the
482 weight of a rock, or the diameter of a planet. Note that an attribute in the context of one resource
483 may be a resource itself in another context.

484 **3.6.2 Version Identification**

485 The XRI specifications must support the ability to unambiguously identify a version of a resource
486 or an attribute. Many URIs include version information about a specified resource. However RFC
487 2396 [URI] does not define a standard way to express this version information, so it is generally
488 embedded in the path component in a way that is specific to the URI's top-level naming authority.
489 Consequently, applications processing URIs must deal with versioning on a case-by-case basis.
490 Because "version" is a generic concept applicable to virtually all resources, the XRI specification
491 must define a standard way to express this concept in any XRI.

492 **3.7 Authority, Delegation, Federation, & Extensibility**

493 **3.7.1 Unlimited Root Authorities**

494 The XRI specifications must support the ability for any resource to serve as its own root identifier
495 authority.

496 **3.7.2 Unlimited Topologies**

497 The XRI specifications must support the ability for identifier authorities to be organized in any
498 topology (e.g., centralized, hierarchical, federated, peer-to-peer, or web). All absolute XRIs will be
499 rooted on an identifier authority that establishes rules governing that authority's namespace.
500 Beyond this top-level authority, however, the XRI specification should be as unrestrictive as
501 possible with respect to the organization or topology, if any, reflected in the remaining path.

502 **3.7.3 Unlimited Delegation and Federation**

503 The XRI specifications must support the ability for any identifier authority to delegate to any other
504 identifier authority, and for delegation relationships to change over time.

505 **3.7.4 Scheme Extensibility**

506 The XRI specifications must support the ability for the identifier scheme to be extended without
507 changing the underlying architecture. It should be possible, therefore, for an application
508 processing an XRI to dynamically load appropriate handlers for a particular XRI.

509 RFC 2396 [URI] says, "Many URI schemes include a top hierarchical element for a naming
510 authority, such that the namespace defined by the remainder of the URI is governed by that
511 authority." RFC 2396 treats this top-level element as a special case, allowing URIs in general to
512 be extensible at least with respect to scheme and top-level naming authority. RFC 2396 goes on
513 to define the segments following the authority as simply "data, specific to the
514 authority...identifying the resource within the scope of that scheme and authority."

515 XRIs, in contrast, must treat each segment as a potential naming authority that defines its own
516 namespace and the rules governing that namespace. In other words, XRIs should be extensible
517 at any segment in the same way that URIs are extensible at the top segment. The implication,
518 then, is that syntax rules for XRIs should be as unrestrictive as possible to allow any given
519 identifier authority the widest possible latitude in defining rules and syntax specific to its
520 namespace.

521 **3.7.5 Specializations**

522 The XRI specifications must support the ability for an extension or restriction to the XRI scheme
523 applying to a particular identifier community or namespace to be published as a BNF ruleset or
524 similar specification. Such specializations must not inhibit interoperability of XRI resolvers – see
525 requirement 4.6.

526 **3.8 Data Protection and Security**

527 **3.8.1 Identifier Security**

528 The XRI specifications must not require identifiers that compromise security. Requiring identifiers
529 that contain a login name or password, for example, would potentially compromise the security of
530 the identified resource.

531 **3.8.2 Identifier Privacy**

532 The XRI specifications must not require identifiers that compromise privacy. For example,
533 requiring identifiers that are semantically reflective of the identified resource would potentially
534 compromise that resource's privacy. It must be possible, therefore, to construct an XRI that
535 contains little or no semantic reflection.

536 4 XRI Resolution Requirements

537 4.1 Non-Resolvability

538 The XRI specifications must permit identifiers that are not resolvable. The application that initiated
539 the resolution request must be able to distinguish between the following results:

- 540 • Successful resolution; resolved value is returned.
- 541 • XRI exists but is not resolvable.
- 542 • XRI does not exist (or is a least unknown by the resolver).
- 543 • XRI is explicitly non-resolvable (see 3.4.5).

544 4.2 Semantic Mapping

545 The XRI specifications must enable any XRI to resolve to any other XRI (3.4.4), and specifically
546 an HFI (3.3.1) to resolve to a MFI (3.3.2).

547 4.3 Resolution Mechanism-Independence

548 The XRI specifications must not require the use of a particular resolution mechanism and must be
549 able to be bound to multiple resolution mechanisms.

550 4.4 Internet Resolution Mechanism

551 The XRI specifications must specify at least one resolution mechanism using widely available
552 Internet technologies.

553 4.5 Unlimited Federation

554 The XRI specifications must support the ability for resolution to be delegated across any number
555 of identifier authorities, and for delegation relationships to change over time. Specifically, it should
556 support the ability for an identifier to be resolved privately within a community of interest, and later
557 federated and thus be resolvable with other communities.

558 4.6 Interoperability of Specializations

559 Extensions or restrictions of the XRI specifications (see 3.6.5) shall not inhibit interoperability of
560 XRI resolution mechanisms, i.e., a specialization of the XRI URI scheme must not cause XRI
561 resolvers to fail.

562 4.7 Scalability

563 The XRI specifications must be capable of being implemented globally at very large scale.
564 Identifier authorities are encouraged, but not required, to support scalable naming and resolution.
565 The XRI specification itself must not impede such support.

566 4.8 Redundancy

567 The XRI specifications must enable an identifier to be resolved in such a way that there is no
568 single point of failure.

569 **4.9 Trusted Resolution**

570 The XRI specifications must ensure that the specified resolution mechanism(s) can be extended
571 to provide secure and trusted resolution. Trusted resolution in this context means the relying party
572 (i.e., the party requesting resolution) can trust that the result of resolution is authoritative as
573 defined by the controlling identifier authority. Specification of the actual mechanism for secure
574 and trusted resolution may be out of scope of this TC. Data confidentiality and access control are
575 explicitly outside of the context of this requirement.

576 **4.10 Proxy Resolution**

577 The XRI specifications must not preclude the use of proxy resolution mechanisms that do not
578 reveal intermediate resolution values or other attributes associated with a resource that may
579 compromise security or privacy. For example, if a human-friendly XRI resolves to a persistent XRI
580 (as provided in 3.4.4) which in turn resolves to a requested attribute, the proxy resolution
581 mechanism would not reveal the persistent XRI.

582 **5 Glossary**

583 Terminology related to identifiers, names, addresses, and other semantic and syntactic
584 associations is notoriously difficult because of the levels of abstraction involved. The intent of the
585 XRI TC is not to solve this problem, but simply to use a limited set of commonly accepted terms
586 precisely and uniformly throughout the XRI specifications. (Note that wherever possible, we
587 incorporate terms from Unified Modeling Language (UML) **[UML]**.) These terms and their
588 definitions are presented in the following Normative Glossary.

589 Following this is an Informative Glossary that explains other frequently used terms in this area of
590 work and how they map into the terms chosen for the Normative Glossary.

591 **5.1 Normative Glossary**

592 **Absolute Identifier**

593 An identifier that refers to a resource independent of the context in which the identifier is
594 used or resolved. Mutually exclusive with "Relative Identifier".

595 **Absolute Persistence**

596 A property of an identifier whereby the entire identifier is persistent and will never be
597 reassigned to another resource. A URN is an absolute persistent identifier. Mutually
598 exclusive with "Relative Persistence".

599 **Abstract Identifier**

600 An identifier which is not directly resolvable to a resource, but which must be resolved
601 into a concrete identifier first (or is non-resolvable). A URN is an example of an abstract
602 identifier. Note that some abstract identifiers may not be resolvable at all—see "Non-
603 Resolvable Identifier". Mutually exclusive with "Concrete Identifier".

604 **Attribute**

605 Any data, metadata, or resource that can be identified only in the context of a specific
606 resource. Examples include the age of a person, the weight of a rock, and the diameter of
607 a planet. Attributes always relative to the resource they describe; they exist only in the
608 context of this relationship. In UML terms, this means a resource has a composition
609 relationship with its attributes, i.e., they are parts of only one whole, vs. an aggregation
610 relationship, where a part can be a member of multiple wholes. For example, a person
611 can have only one age (composition), but a person can be a member of multiple
612 workgroups (aggregation).

613 Note that an attribute in the context of one resource may be a resource itself in another
614 context. For example, a phone number is a composite attribute of a phone, however the
615 phone number may itself be a resource aggregated by another resource such as a
616 business card. Attributes can also be nested, i.e., contain other attributes. For example,
617 the phone number attribute of a telephone may in turn contain the attributes country
618 code, area code, number, and extension.

619 **Community**

620 See "Identifier Community".

621 **Composite Identifier**

622 An identifier that consists of a path of two or more components, called segments. See
623 "Path" and "Segment". URIs are composite identifiers – see **[URI]**.

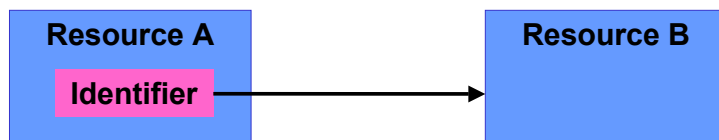
- 624 **Concrete Identifier**
- 625 An identifier which can be directly resolved to a resource, rather than indirectly to another
626 identifier. The HTTP URI of a Web page, the IP address of a host machine, a phone
627 number, and a postal address are all examples of concrete identifiers. All concrete
628 identifiers are resolvable identifiers. Mutually exclusive with "Abstract Identifier".
- 629 **Context**
- 630 See "Identifier Context".
- 631 **Cross-Context Identifier**
- 632 An identifier assigned in one context that is reused in another context. Cross-context
633 identifiers are used primarily to identify logically equivalent resources in different domains
634 or physical locations, for example, the same logical invoice stored in two accounting
635 systems (the originating system and the receiving system), the same logical Web page
636 stored on multiple proxy servers, or the same datatype in two databases or XML
637 schemas.
- 638 **Cross-reference**
- 639 See "Cross-Context Identifier".
- 640 **Delegation**
- 641 See "Identifier Delegation".
- 642 **Domain**
- 643 The set of resources that share a common association. Typically used in networking to
644 mean a zone of control, administration, authority, security, or policy enforcement. For
645 example, a "security domain" is a zone (a network, or collection of machines, or other
646 logical partition) where all entities have a certain level of trust not afforded outside that
647 zone. A "host domain" is a zone where all the resources are physically hosted and
648 administered together. A "legal domain" is a zone where all the resources are under the
649 control of the same resource controller.
- 650 **Federation**
- 651 See "Identifier Federation".
- 652 **HFI**
- 653 See "Human-Friendly Identifier".
- 654 **Human-Friendly Identifier**
- 655 An identifier containing human-readable words or phrases intended to invoke linguistic
656 associations and be easy for people to remember and use. Mutually exclusive with
657 "Machine-Friendly Identifier."
- 658 **Identifier**
- 659 A string of characters that refers to a resource. More specifically, an attribute of a
660 resource (the identifier context) that forms an association with another resource (the
661 identifier target). In UML terms, this means an identifier is an attribute of one object that
662 forms an association with another object. The general term identifier does not specify
663 whether the identifier is concrete or abstract, persistent or reassignable, human-friendly
664 or machine-friendly, absolute or relative, public or private, or resolvable or non-
665 resolvable.
- 666 **Identifier Authority**
- 667 A resource that assigns identifiers to other resources. The term may also refer to the
668 responsible resource controller.

669 **Identifier Community**

670 The set of resources that share a common identifier authority. From a technical
671 perspective, this means the set of resources whose identifiers form a directed acyclic
672 graph or tree.

673 **Identifier Context**

674 The backpointer of an identifier, i.e., the resource of which the identifier is an attribute.
675 Context is always relative to an identifier. Context is the parent resource that assigns the
676 identifier for the target resource. Since multiple resources may assign an identifier for a
677 target resource, the resource can be said to be identified in multiple contexts.
678



Resource B is identified in the context of Resource A

679

680 **Identifier Delegation**

681 The process of an identifier authority assigning an identifier to another identifier authority.

682 **Identifier Federation**

683 The joining of two identifier communities by having an identifier authority in one
684 community delegate to an identifier authority in the other community to create a single
685 community. From a technical perspective, this joins two directed acyclic graphs into one.

686 **Identifier Scheme**

687 The syntactic rules governing the composition of a composite identifier, for example the
688 rules governing delimiters, ordering, and legal characters. The IETF URI specification
689 **[URI]** specifies an overall scheme for the identifiers used in the World Wide Web.
690 Individual URI scheme specifications (including the XRI specification) specify identifier
691 schemes compliant with this specification.

692 **Machine-Friendly Identifier**

693 An identifier that is optimized for efficient machine searching, routing, caching, and
694 resolvability. Mutually exclusive with "Human-Friendly Identifier."

695 **MFI**

696 See "Machine-Friendly Identifier".

697 **Network Resource**

698 A resource that has a digital representation on the network and is addressable in some
699 form. Network resources include hosts, files, directory entries, databases, services, web
700 pages, etc. Mutually exclusive with "Non-Network Resource".

701 **Node**

702 The smallest segment in a path, i.e., the individual points in the directed graph formed by
703 any composite identifier.

704 **Non-Network Resource**

705 A resource that exists independently of the network. Non-network resources include
706 people, organizations, physical objects, and concepts ("car", "flower", "love"). The
707 unambiguous identification of non-network resources can be particularly challenging—
708 see David Booth's paper on this subject, "Four Uses of a URL: Name, Concept, Web
709 Location, and Document Instance" **[Booth]**. Mutually exclusive with "Network Resource".

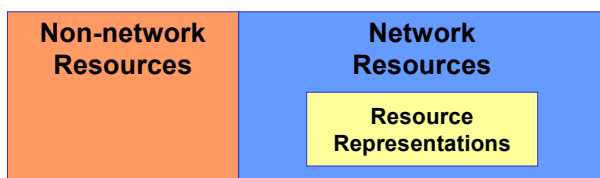
- 710 **Non-Resolvable Identifier**
- 711 An identifier that does not reference a network resource or resource representation, but
712 which exists only to abstractly represent a resource. A non-resolvable identifier is always
713 an abstract identifier and does not have any corresponding data or metadata describing
714 the resource it represents, and thus cannot be resolved in the conventional sense. Non-
715 resolvable identifiers are frequently used as cross-references – see "Cross-Context
716 Identifiers". Mutually exclusive with "Resolvable Identifier."
- 717 **Path**
- 718 Any sequence of segments within a composite identifier.
- 719 **Private Identifier**
- 720 An identifier that is not intended to be shared outside the community in which it is
721 assigned. Private identifiers may contain sensitive data. Mutually exclusive with "Public
722 Identifier".
- 723 **Public Identifier**
- 724 An identifier that is intended for public disclosure and does not contain sensitive data.
725 Mutually exclusive with "Private Identifier".
- 726 **Reassignable Identifier**
- 727 An identifier that may be reassigned from one resource to another. Example: the domain
728 name "business.com" may be reassigned from ABC Company to XYZ Company, or the
729 email address "john@example.com" may be reassigned from John Smith to John Jones.
730 Reassignable identifiers tend to be human-friendly identifiers because they frequently
731 represent the mapping of non-network semantic relationships onto network resources or
732 resource representations.
- 733 **Registration**
- 734 The process of a resource requesting an identifier from an identifier authority and
735 supplying the data and metadata necessary for the identifier authority to resolve the
736 identifier to the target resource. Not all identifiers are registered – for example an
737 authority may unilaterally assign an identifier to a resource without the knowledge or
738 participation of the resource. In addition, not all registered identifiers are selected by the
739 registering resource. In general, human-friendly identifiers are selected by the registering
740 resource and machine-friendly identifiers are selected by the identifier authority.
- 741 **Relative Identifier**
- 742 An identifier that refers to a resource relative to the context in which the identifier is used
743 or resolved. Mutually exclusive with "Absolute Identifier".
- 744 **Relative Persistence**
- 745 A property of an identifier in which only a relative portion of the identifier is persistent, or
746 the identifier is only persistent for a relative period of time. Mutually exclusive with
747 "Absolute Persistence".
- 748 **Representation**
- 749 See "Resource Representation".
- 750 **Resolvable Identifier**
- 751 An identifier that references a network resource or resource representation and therefore
752 can be resolved into data or metadata describing the target resource.
- 753 **Resolution**
- 754 The process of dereferencing an identifier to a set of data and metadata describing the
755 target resource.

756 **Resolver**

757 A resource that offers the service of resolution.

758 **Resource**

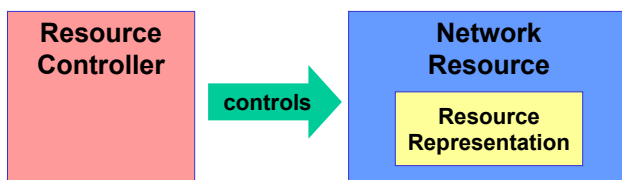
759 As defined in RFC 2396 **[URI]**: "anything that has identity". In Topic Maps **[TopicMaps]**,
760 a resource is the equivalent of a "subject". In UML **[UML]**, a resource is modeled as an
761 "object". Resources are of two types: non-network resources and network resources.
762 Network resources in turn contain a subtype, resource representations. A resource
763 representation may represent either a network resource or a non-network resource.
764



765
766

767 **Resource Controller**

768 A resource responsible for managing a network resource. A resource controller is
769 typically a non-network resource (e.g., a person or an organization), however it may also
770 be a network resource such as an application. In the European Union, a non-network
771 resource controller (a person or an organization) that controls a resource representation
772 containing personally identifiable data is legally referred to as a "data controller".
773



774
775

776 **Resource Representation**

777 A network resource that represents the attributes of another resource. A resource
778 representation may represent either a network resource (such as an application) or a
779 non-network resource (such as a person, organization, object, or concept).

780 **Segment**

781 Any syntactically-defined component of a composite identifier. A segment may consist of
782 a single node or a path of nodes. Segments may also contain other segments. A
783 segment can distinguished by its purpose (e.g., an authority segment, a cross-reference
784 segment) or by its syntactic delimiter(s) (e.g., a slash segment, a dot segment, etc.)

785 **Semantic Mapping**

786 The approach of using a human-friendly reassignable identifier to identify a machine-
787 friendly persistent identifier for the purpose of establishing equivalence of semantic
788 names and concepts across languages, ontologies, communities, etc.

789 **URI**

790 Uniform Resource Identifier. An Internet and Web architecture term for the identifiers
791 used to create the World Wide Web. See **[URI]**.

792 **URN**

793 Uniform Resource Name. An Internet and Web architecture term for persistent identifiers.
794 See **[URN]**.

795 **Version**
796 A state of a resource or an attribute that can be identified apart from other states.
797

798 **5.2 Informative Glossary**

799 The following terms appear frequently in work related to Internet identifiers but were not chosen
800 for the Normative Glossary because they either: a) are too general, or b) overlap with another
801 term judged to be more appropriate for XRI work.

802

803 **Address**

804 A term commonly used to refer to a resolvable identifier, i.e., one that can be resolved
805 into data and metadata describing the target resource.

806 **Digital Identity**

807 A term commonly used to refer to a resource representation that represents a non-
808 network resource, such as a person or an organization.

809 **Directory**

810 A term commonly used for a resource that serves as an index of other resources, i.e.,
811 one specializes in storing and searching identifiers and resource representations. In
812 general every identifier authority maintains some form of directory.

813 **ID**

814 A term typically used for persistent identifiers and/or machine-friendly identifiers.
815 However ID is also shorthand for "identifier" and thus in some contexts may include any
816 type of identifier.

817 **Identity**

818 A very broad term with many connotations in computer science, social science,
819 philosophy, and popular culture. See "Resource" and "Digital Identity."

820 **Locator**

821 A term equivalent to "Concrete Identifier" but which is sometimes also used to mean
822 "Resolvable Identifier".

823 **Name**

824 A term often synonymous with "Identifier" but with many additional connotations about
825 the identifier type. Ironically, two of the most common Internet uses are in direct conflict:
826 a Domain Name Service (DNS) name is a reassignable identifier, while a Uniform
827 Resource Name (URN) is a persistent identifier.

828 **Object**

829 A term used by UML **[UML]** to describe a resource.

830 **Subject**

831 A term used by Topic Maps **[TopicMaps]** that explicitly refers to a non-network resource.
832 To represent a subject on the network, you must use a Topic. See "Topic".

833 **Topic**

834 A term used by Topic Maps **[TopicMaps]** that explicitly refers to the identifier of a non-
835 network resource, called a Subject. See "Subject".

836 6 References

837

838 **[Keywords]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
839 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

840 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, *Uniform Resource Identifiers (URI):
841 Generic Syntax*, <http://www.ietf.org/rfc/rfc2396.txt>, IETF RFC 2396, August
842 1998.

843 **[URN]** R. Moats, *URN Syntax*, <http://www.ietf.org/rfc/rfc2141.txt>, IETF RFC 2141,
844 May 1997.

845 **[URNReqs]** K. Sollins, L. Masinter, *Functional Requirements for Uniform Resource
846 Names*, <http://www.ietf.org/rfc/rfc1737.txt>, IETF RFC 1737, December 1994.

847 **[Mailto]** P. Hoffman, L. Masinter, J. Zawinski, *The mailto URI Scheme*,
848 <http://www.ietf.org/rfc/rfc2368.txt>, IETF RFC 2368, July 1998.

849 **[UML]** Object Management Group, *Unified Modeling Language (UML) Version 1.5*,
850 <http://www.omg.org/technology/documents/formal/uml.htm>, March 1, 2003.

851 **[SPKI]** C. Ellison, *SPKI Requirements*, <http://www.ietf.org/rfc/rfc2692.txt>, IETF RFC
852 2692, September, 1999.

853 **[Guidelines]** L. Masinter, H. Alvestrand, D. Zigmund, R. Petke, *Guidelines for New URL
854 Schemes*, <http://www.ietf.org/rfc/rfc2718.txt>, IETF RFC 2718, November
855 1999.

856 **[XML]** T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, *Extensible Markup
857 Language (XML) 1.0 (Second Edition) W3C Recommendation*,
858 <http://www.w3.org/TR/REC-xml>, October 2000.

859 **[IRI]** M. Duerst, M. Suignard, *Internationalized Resource Identifiers (IRIs)*,
860 <http://www.ietf.org/internet-drafts/draft-duerst-iri-03.txt>, March 2003.

861 **[IETFCharsets]** H. Alvestrand, *IETF Policy on Character Sets and Languages*,
862 <http://www.ietf.org/rfc/rfc2277.txt>, January 1998.

863 **[Unicode]** The Unicode Consortium, *The Unicode Standard, Version v3.0*, Addison-
864 Wesley Pub Co; ISBN: 0201616335, February, 2000.

865 **[Booth]** D. Booth, *Four Uses of a URL: Name, Concept, Web Location, and
866 Document Instance*, [http://www.w3.org/2002/11/dbooth-names/dbooth-
names_clean.htm](http://www.w3.org/2002/11/dbooth-names/dbooth-
867 names_clean.htm), January 2003.

868 **[TopicMaps]** Steve Pepper, Ontopia, *The TAO of Topic Maps*,
869 <http://www.ontopia.net/topicmaps/materials/tao.html>, April, 2002.

870

871 **Appendix A. Acknowledgments**

872 In addition to the editors, the following people made significant contributions to this document:

- 873 • Peter Davis, Neustar
- 874 • Mike Lindelsee, Visa International
- 875 • Nat Sakimura, NRI
- 876 • Dave Wentker, Visa International
- 877 • John Veizades, Visa International

878 Appendix B. Notices

879 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
880 that might be claimed to pertain to the implementation or use of the technology described in this
881 document or the extent to which any license under such rights might or might not be available;
882 neither does it represent that it has made any effort to identify any such rights. Information on
883 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
884 website. Copies of claims of rights made available for publication and any assurances of licenses
885 to be made available, or the result of an attempt made to obtain a general license or permission
886 for the use of such proprietary rights by implementors or users of this specification, can be
887 obtained from the OASIS Executive Director.

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

891 **Copyright © OASIS Open 2003. All Rights Reserved.**

892 This document and translations of it may be copied and furnished to others, and derivative works
893 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
894 published and distributed, in whole or in part, without restriction of any kind, provided that the
895 above copyright notice and this paragraph are included on all such copies and derivative works.
896 However, this document itself does not be modified in any way, such as by removing the
897 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
898 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
899 Property Rights document must be followed, or as required to translate it into languages other
900 than English.

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

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