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/).

Table of Contents

1 Introduction

This document is divided into four major sections:

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

1.1 Terminology

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

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

2 Motivations

2.1 Introduction

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

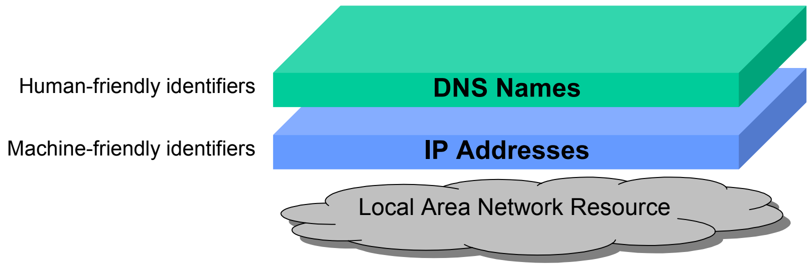


Figure 1: The two layers of Internet identifiers in predominant use today.

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

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

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

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

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

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

The phenomenal success of the Web meant that URIs became the fastest-growing new address 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.

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

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

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

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

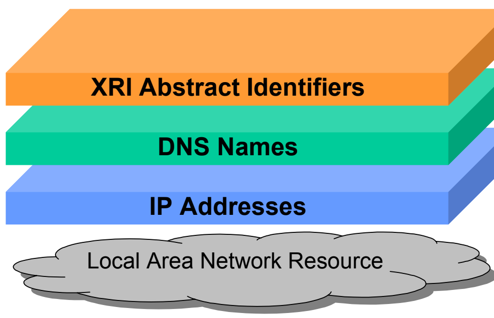


Figure 2: XRIs are designed to provide a uniform third layer of abstract identifiers for Internet resources.

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

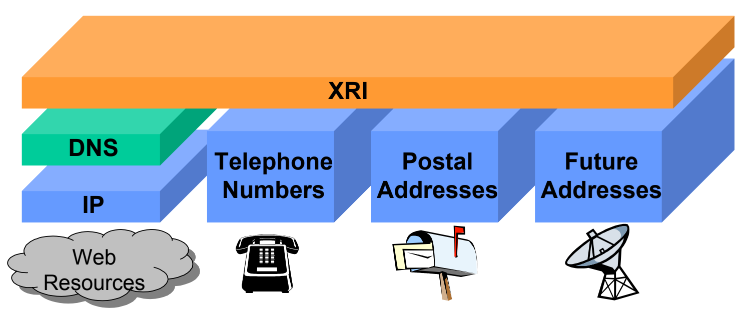


Figure 3: XRIs can serve as true unified addresses across all communications networks.

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

2.2 Persistent Identification

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

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

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

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

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

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

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

2.3 Human-Friendly vs. Machine-Friendly Identification

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

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

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

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

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

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

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

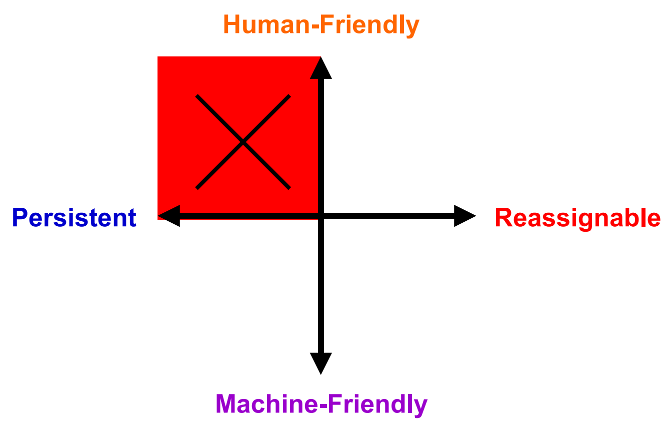


Figure 4: The relationship of the persistent/reassignable gradient and the HFI/MFI gradient.

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

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

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

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

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

a. They must support any combination of persistent and reassignable HFIs and MFIs, and

b. When a resource needs both a reassignable HFI and a persistent MFI, the XRI specifications 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.

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

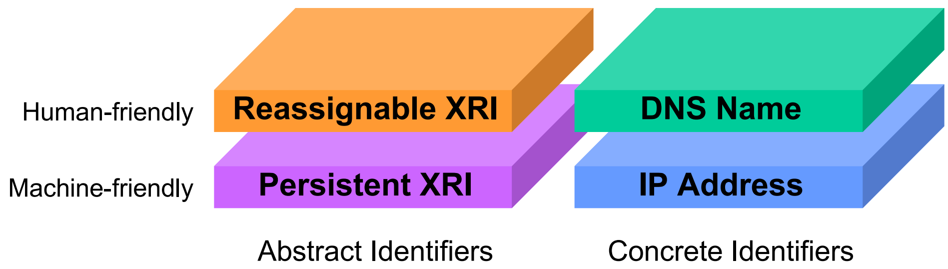


Figure 5: XRIs can map reassignable HFIs to persistent MFIs the same way DNS names are mapped to IP addresses.

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

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

2.4 Cross-Context Identification

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

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

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

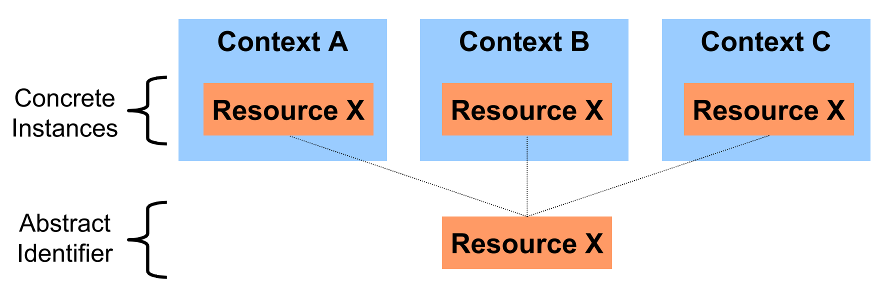


Figure 6: Sharing an abstract identifier across multiple concrete contexts.

Note that doing this in a uniform manner requires a URI syntax that permits combining an abstract resource identifier created in one context (e.g., a dictionary or taxonomy authority) with a concrete identifier that establishes the local context—exactly like combining the term "President" with a concrete postal address. For example, an XRI representing a concept such as "management team" could be combined with the URI of a home page to form a well-known address for this type of resource on any corporate website. This would enable the development of much smarter and more specialized spiders than crawl the Web today.

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

2.5 Resource Attribute and Version Identification

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

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

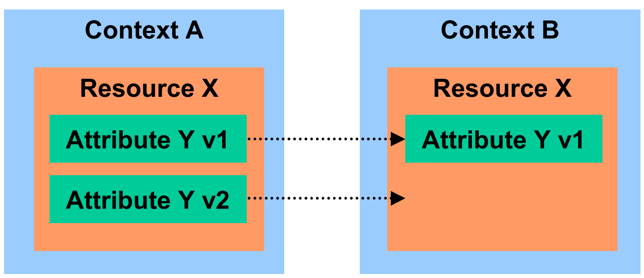


Figure 7: The need to standardize attribute and version identification.

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

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

2.6 Delegation, Federation, and Extensibility

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

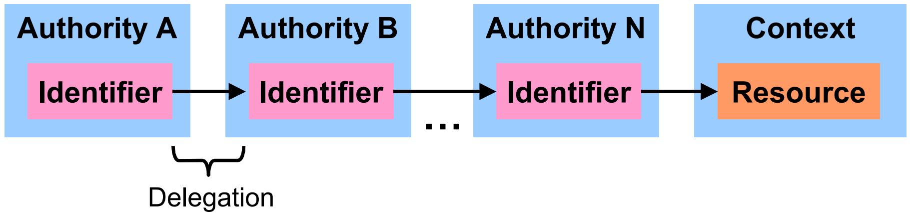


Figure 8: The need for delegation to arbitrary depth.

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

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

2.7 Security and Privacy

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

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

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

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

3 XRI Syntax Requirements

Following are the requirements for the XRI URI scheme specification.

3.1 URI and URN Requirements

3.1.1 URI Conformance

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

- BNF definition of an identifier scheme
- Hierarchical namespace with absolute and relative identifier forms
- Global transcribability
- Escaping and character encoding rules
- Naming authorities
- Query components
- Fragment identifiers

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

3.1.2 URN Conformance

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

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

3.2 Abstraction and Independence

3.2.1 Location-Independence

The XRI specifications must enable identifiers to identify a resource independent of its location on the network (if any). At various points in its lifetime, the same resource may or may not be accessible on the network. It may migrate from one network endpoint to another. It may move from one machine to another. It may be retired and be permanently inaccessible. Consequently, it must be possible to construct a valid XRI that does not reflect an identified resource's location on the network (or lack thereof).

3.2.2 Application-Independence

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

3.2.3 Transport-Independence

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

3.2.4 Type-Independence

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

3.2.5 Security Method-Independence

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

3.3 Persistent Identification

3.3.1 Persistent Identifiers

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

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

3.3.2 Reassignable Identifiers

The XRI specifications must support the ability for the association between an identifier and the resource it identifies to be changed. This is often desirable when an identifier semantically reflects characteristics of the resource it identifies. Because semantics and semantic relationships change over time, it should be possible to accommodate these changes within the resource's identifier. For example, if a trademark or trade name is sold from one company to another, it should be possible for identifiers previously associated with resources at the original company to be reassigned to different resources at the new company.

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

3.3.3 Combining Persistent and Reassignable Identifiers

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

3.4 Human-Friendly and Machine-Friendly Identification

3.4.1 Human-Friendly Identifiers (HFIs)

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

3.4.2 Machine-Friendly Identifiers (MFIs)

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

3.4.3 Combining HFIs and MFIs

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

3.4.4 Identifier Mapping

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

3.4.5 Explicit Non-Resolvability

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

3.4.6 Internationalization

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

3.4.7 Character Encoding

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

3.5 Cross-Context Identification

3.5.1 Cross-References

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

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

3.5.2 URIs as Cross-References

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

3.6 Attribute and Version Identification

3.6.1 Attribute Identification

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

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

3.6.2 Version Identification

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

3.7 Authority, Delegation, Federation, & Extensibility

3.7.1 Unlimited Root Authorities

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

3.7.2 Unlimited Topologies

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

3.7.3 Unlimited Delegation and Federation

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

3.7.4 Scheme Extensibility

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

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

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

3.7.5 Specializations

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

3.8 Data Protection and Security

3.8.1 Identifier Security

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

3.8.2 Identifier Privacy

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

4 XRI Resolution Requirements

4.1 Non-Resolvability

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

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

4.2 Semantic Mapping

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

4.3 Resolution Mechanism-Independence

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

4.4 Internet Resolution Mechanism

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

4.5 Unlimited Federation

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

4.6 Interoperability of Specializations

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

4.7 Scalability

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

4.8 Redundancy

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

4.9 Trusted Resolution

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

4.10 Proxy Resolution

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

5 Glossary

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

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

5.1 Normative Glossary

Absolute Identifier

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

Absolute Persistence

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

Abstract Identifier

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

Attribute

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

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

Community

See "Identifier Community".

Composite Identifier

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

Concrete Identifier

An identifier which can be directly resolved to a resource, rather than indirectly to another identifier. The HTTP URI of a Web page, the IP address of a host machine, a phone number, and a postal address are all examples of concrete identifiers. All concrete identifiers are resolvable identifiers. Mutually exclusive with "Abstract Identifier".

Context

See "Identifier Context".

Cross-Context Identifier

An identifier assigned in one context that is reused in another context. Cross-context identifiers are used primarily to identify logically equivalent resources in different domains or physical locations, for example, the same logical invoice stored in two accounting systems (the originating system and the receiving system), the same logical Web page stored on multiple proxy servers, or the same datatype in two databases or XML schemas.

Cross-reference

See "Cross-Context Identifier".

Delegation

See "Identifier Delegation".

Domain

The set of resources that share a common association. Typically used in networking to mean a zone of control, administration, authority, security, or policy enforcement. For example, a “security domain” is a zone (a network, or collection of machines, or other logical partition) where all entities have a certain level of trust not afforded outside that zone. A "host domain" is a zone where all the resources are physically hosted and administered together. A "legal domain" is a zone where all the resources are under the control of the same resource controller.

Federation

See "Identifier Federation".

HFI

See "Human-Friendly Identifier".

Human-Friendly Identifier

An identifier containing human-readable words or phrases intended to invoke linguistic associations and be easy for people to remember and use. Mutually exclusive with "Machine-Friendly Identifier."

Identifier

A string of characters that refers to a resource. More specifically, an attribute of a resource (the identifier context) that forms an association with another resource (the identifier target). In UML terms, this means an identifier is an attribute of one object that forms an association with another object. The general term identifier does not specify whether the identifier is concrete or abstract, persistent or reassignable, human-friendly or machine-friendly, absolute or relative, public or private, or resolvable or non-resolvable.

Identifier Authority

A resource that assigns identifiers to other resources. The term may also refer to the responsible resource controller.

Identifier Community

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

Identifier Context

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

Resource B is identified in the context of Resource A

Identifier Delegation

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

Identifier Federation

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

Identifier Scheme

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

Machine-Friendly Identifier

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

MFI

See "Machine-Friendly Identifier".

Network Resource

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

Node

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

Non-Network Resource

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

Non-Resolvable Identifier

An identifier that does not reference a network resource or resource representation, but which exists only to abstractly represent a resource. A non-resolvable identifier is always an abstract identifier and does not have any corresponding data or metadata describing the resource it represents, and thus cannot be resolved in the conventional sense. Non-resolvable identifiers are frequently used as cross-references – see "Cross-Context Identifiers". Mutually exclusive with “Resolvable Identifier.”

Path

Any sequence of segments within a composite identifier.

Private Identifier

An identifier that is not intended to be shared outside the community in which it is assigned. Private identifiers may contain sensitive data. Mutually exclusive with "Public Identifier".

Public Identifier

An identifier that is intended for public disclosure and does not contain sensitive data. Mutually exclusive with "Private Identifier".

Reassignable Identifier

An identifier that may be reassigned from one resource to another. Example: the domain name "business.com" may reassigned from ABC Company to XYZ Company, or the email address "john@example.com" may be reassigned from John Smith to John Jones. Reassignable identifiers tend to be human-friendly identifiers because they frequently represent the mapping of non-network semantic relationships onto network resources or resource representations.

Registration

The process of a resource requesting an identifier from an identifier authority and supplying the data and metadata necessary for the identifier authority to resolve the identifier to the target resource. Not all identifiers are registered – for example an authority may unilaterally assign an identifier to a resource without the knowledge or participation of the resource. In addition, not all registered identifiers are selected by the registering resource. In general, human-friendly identifiers are selected by the registering resource and machine-friendly identifiers are selected by the identifier authority.

Relative Identifier

An identifier that refers to a resource relative to the context in which the identifier is used or resolved. Mutually exclusive with "Absolute Identifier".

Relative Persistence

A property of an identifier in which only a relative portion of the identifier is persistent, or the identifier is only persistent for a relative period of time. Mutually exclusive with "Absolute Persistence".

Representation

See "Resource Representation".

Resolvable Identifier

An identifier that references a network resource or resource representation and therefore can be resolved into data or metadata describing the target resource.

Resolution

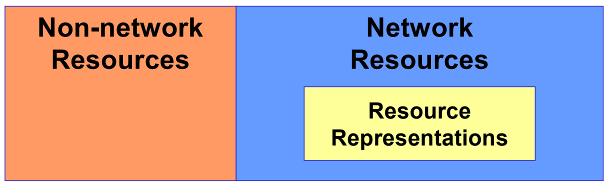
The process of dereferencing an identifier to a set of data and metadata describing the target resource.

Resolver

A resource that offers the service of resolution.

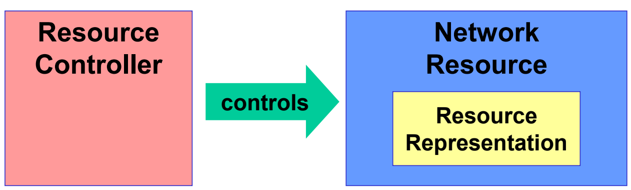
Resource

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



Resource Controller

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



Resource Representation

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

Segment

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

Semantic Mapping

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

URI

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

URN

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

Version

A state of a resource or an attribute that can be identified apart from other states.

5.2 Informative Glossary

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

Address

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

Digital Identity

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

Directory

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

ID

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

Identity

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

Locator

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

Name

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

Object

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

Subject

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

Topic

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

6 References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix A. Acknowledgments

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

- Peter Davis, Neustar
- Mike Lindelsee, Visa International
- Nat Sakimura, NRI
- Dave Wentker, Visa International
- John Veizades, Visa International

Appendix B. Notices