

# YADIS Specification

Version 0.82  
16 January 2006

## Table of Contents

0 Introduction.....	1
1 Scope.....	2
2 Normative References.....	2
2.1 Requests for Comments.....	2
2.2 W3C Recommendations.....	2
2.3 OASIS Specifications.....	2
3 Terms and Definitions.....	3
3.1 Implementer Terms.....	3
3.2 End-user terms.....	4
3.3 Definitions from other specifications.....	4
4 Abbreviations.....	4
5 Conventions.....	4
6 The YADIS Protocol.....	5
6.1 Overview of the YADIS Protocol.....	5
6.2 Protocol Specification.....	5
7. The YADIS Resource Descriptor.....	7
7.1 Overview of the YADIS Resource Descriptor.....	7
7.2 A simple YADIS Resource Descriptor.....	7
7.3. Element definitions.....	8
7.4 Other parts of a YADIS Resource Descriptor.....	9

## 0 Introduction

YADIS is yet another decentralized identity interoperability system. It uses URLs as identifiers and provides a mechanism for determining the identity services available using a given identifier.

This draft Version 0.82 has been prepared by the YADIS project. For more information on the YADIS project, see the [YADIS wiki](#) and [YADIS Overview](#).)

The next draft, Version 0.9, will be for implementors.

If you are implementing YADIS, please add a page to the YADIS wiki by adding a link at [YADIS Implementations](#). When you discover problems with this Version 0.9, please add a page to the YADIS wiki by adding a link at [Change Proposals](#).

Please send your comments on this text to the editor, [Joaquin Miller](#).

This text is made available under [Attribution-ShareAlike 2.5](#) when attributed to the YADIS project.

## 1 Scope

This YADIS specification provides:

- A general purpose identifier for persons and any other entities, which can be used in a variety of identity services.
- A syntax for a resource description document identifying services available using that identifier and an interpretation of the elements of that document.
- A protocol for obtaining that resource description document, given that identifier.

Together these enable coexistence and interoperation of a rich variety of identity services using a single identifier. The identifier uses a standard syntax and a well-established namespace; it requires no additional namespace administration infrastructure.

This specification is intended to be used to implement software for requesting a resource description document and to implement software for providing a resource description document or a locator for that document. The document syntax and interpretation and the protocol are specified to a level of detail sufficient to enable the determination of the compliance of any implementation to this specification.

## 2 Normative References

The following specifications contain provisions which, through reference in this text, constitute provisions of this YADIS Specification. At the time of publication, the editions indicated were valid, with the exception of those in Clause 2.3. All specifications are subject to revision, and parties to agreements based on this Specification are encouraged to investigate the possibility of applying the most recent edition of the specifications listed below.

### 2.1 Requests for Comments

<a href="#">RFC 2119</a>	Key words for use in RFCs to Indicate Requirement Levels, March 1997.
<a href="#">RFC 2616</a>	Hypertext Transfer Protocol -- HTTP/1.1, June 1999.
<a href="#">RFC 3896</a>	Uniform Resource Identifier (URI): Generic Syntax, January 2005.
<a href="#">RFC 3987</a>	Internationalized Resource Identifiers (IRIs), January 2005.

### 2.2 W3C Recommendations

<a href="#">HTML 4.01</a>	HTML 4.01 Specification, 24 December 1999.
<a href="#">XHTML 1.0</a>	The Extensible HyperText Markup Language (Second Edition), 1 August 2002.

### 2.3 OASIS Specifications

The following documents have not been adopted by OASIS. They are listed here for two reasons:

- These documents were used in preparing this YADIS Specification.
- When and if OASIS adopts such specifications, it is expected that this YADIS specification will conform to them and will at that time reference them here as normative references.

<a href="#">XRI Syntax</a>	Extensible Resource Identifier (XRI) Syntax V2.0, Committee Specification, 14 November 2005.
<a href="#">XRI Resolution</a>	Extensible Resource Identifier (XRI) Resolution V2.0, Working Draft 09, 10 November 2005. (See the <a href="#">YADIS wiki</a> for the current draft.)

## 3 Terms and Definitions

### 3.1 Implementer Terms

YADIS User	A person using a YADIS ID as a personal identifier.
YADIS ID	A personal identifier used in one or more YADIS Identity Services. A YADIS ID may be a URL or an XRI resolvable to a URL. A YADIS ID may be used in the YADIS Protocol to obtain a YADIS Resource Descriptor. (See RESTful YADIS URL, Hosted YADIS URL, and Delegating YADIS URL below.)
YADIS URL	A YADIS ID, if it is a URL, otherwise the URL to which that YADIS ID resolves.
YADIS Page	In case there is a document that can be retrieved with an HTTP GET request to the YADIS ID, that document.
YADIS Identity Service	A computer software process (or system of processes) that provides one or more services located using the YADIS Protocol.
YADIS Resource Descriptor	A document identifying the identity services that are available using a YADIS ID.
Resource Descriptor URL	A URL that locates a YADIS Resource Descriptor.
Agent	A computer software process (or system of processes) acting on behalf of a party.
YADIS User Agent	An agent acting on behalf of a YADIS User.
Relying Party	A party responsible for a Relying Party Agent and on whose behalf that Agent acts. A Relying Party is relying on the services provided by a YADIS Identity Service, in particular on data provided by that service concerning the person identified by a YADIS ID.
Relying Party Agent	A role to be fulfilled by an agent that uses a YADIS ID (and the data accessible using that YADIS ID) provided by a YADIS User Agent. The Relying Party Agent discovers the services available for a YADIS ID using the YADIS Protocol, and modifies its own behavior accordingly.
Identity Consumer	Used to mean Relying Party Agent.
Identity Server	A role to be fulfilled by a program that hosts one or more Hosted YADIS IDs. The Identity Server may or may not be located by the same URL as the YADIS ID.
RESTful YADIS ID	A YADIS ID which, in the YADIS Protocol, itself returns a YADIS Resource Descriptor. LID identifiers are REST-full YADIS IDs.
Hosted YADIS ID	A YADIS ID which, in the YADIS Protocol, does not itself return a YADIS Resource Descriptor, but instead specifies an Identity Server, but no delegate YADIS ID. OpenID URLs are either Hosted YADIS URLs, or Delegating YADIS IDs.
Delegating ID URL	A YADIS ID which, in the YADIS Protocol, does not itself return a YADIS Resource Descriptor, but which specifies an Identity Server and a delegate YADIS ID. OpenID URLs are either Hosted YADIS URLs, or Delegating YADIS URLs.

Note: YADIS IDs are intended to be used not only by persons but also by other parties, such as clubs, crowds, businesses and governments. This terminology does not attempt to be general in this way, but instead focuses on use by persons. YADIS IDs are also intended for use to identify agents of any such parties, such as RFID tags and software processes. The YADIS Protocol is intended to be used by any software agent wishing to discover the services available for use with any YADIS ID.

### 3.2 End-user terms

My ID	An identifier a YADIS User uses to identify an individual, such as themselves or somebody else. Same as YADIS ID.
My page	The document located by My URL. Same as YADIS Page.

### 3.3 Definitions from other specifications

This specification makes use of the following terms as defined in the indicated specifications:

IRI	<a href="#">RFC 3987</a>
scheme	<a href="#">RFC 3986</a> 3.1
URL	<a href="#">RFC 3986</a> 1.1.3
URI	<a href="#">RFC 3986</a>
XRI	<a href="#">XRI Syntax</a>

## 4 Abbreviations

These abbreviations are use in this specification:

YRD	YADIS Resource Descriptor
-----	---------------------------

## 5 Conventions

The key words and word pairs, 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL', in this document are to be interpreted as described in RFC 2119.

## 6 The YADIS Protocol

### 6.1 Overview of the YADIS Protocol

The purpose of the YADIS protocol is to enable a Relying Party, which has been offered a YADIS ID, to obtain a YADIS Resource Descriptor that describes the services available using that YADIS ID.

#### 6.1.1 Obtaining the YADIS Resource Descriptor

When a User provides a YADIS ID to a Relying Party, that Relying Party will want to discover which services are available using that YADIS ID. (Is it a LID or OpenID URL? What authentication methods are available?)

To do that, the Relying Party makes an HTTP request using the YADIS URL. This request may take any one of several forms, discussed in Clause 6.2.2 of this document.

In response to the request, the Relying Party obtains either:

1. A YADIS Resource Descriptor.
2. A URL that locates a YADIS Resource Descriptor.
3. Some other response; in this case the identifier is not a YADIS ID or there has been a failure.

If the response is 2, the Relying Party uses that URL to obtain the YADIS Resource Descriptor.

The Resource Descriptor lists the services available using that YADIS ID, including services that can authenticate the User.

For a description of the YADIS Resource Descriptor, see Clause 7, YADIS Resource Descriptor.

### **6.1.2 Authentication**

Authentication is performed using one or more Identity Services. The Relying Party uses the information in the YADIS Resource Descriptor to choose an Identity Service suitable to its purposes, and uses that Service to authenticate the user.

### **6.1.3 Other services**

Identity Services offer other services, in addition to authentication. These services are identified in the YADIS Resource Descriptor and each Service operates according to the specification of the particular Identity Service providing that Service.

Example: LID defines a RESTful protocol that allows the structured query of profile information about the YADIS User. LID Profile Exchange can be offered as a YADIS Service.

## **6.2 Protocol Specification**

### **6.2.1 YADIS URL**

The scheme name of a YADIS URL must be HTTP or HTTPS.

### **6.2.2 Alternatives**

A Relying Party MAY use the YADIS URL to make an HTTP GET request. This request MAY return an HTML document. If it does, either that document itself or the HTTP response-headers will contain a Resource Descriptor URL giving the location of the YADIS Resource Descriptor. The Relying Party then obtains the YADIS Service Descriptor using that URL.

The YADIS Protocol also includes two alternatives:

The Relying Party MAY first issue an HTTP HEAD request. In that case, the Resource Descriptor URL MAY be returned in an HTTP response-header.

The Relying Party MAY include in the HTTP GET request an Accept request-header asking for the YADIS Resource Descriptor to be returned. In that case the YADIS Resource Descriptor MAY be returned in response to that request, instead of an HTML document.

Note: If the resource located by a YADIS ID will be returning a YADIS Resource Descriptor document in response to an Accept request-header, all responses to a request using the YADIS URL need to include a Vary: Accept response-header to indicate to intervening caches that differing values of the Accept header may return different responses. This header needs to be present even in the case where the HTML page is returned (instead of a YADIS Resource Descriptor document).

The following Clauses specify the steps of the YADIS Protocol.

### **6.2.3 Initiation**

The YADIS Protocol is initiated by the Relying Party with an initial HTTP request using the YADIS URL.

This request MUST be either a GET or a HEAD request.

A GET or HEAD request MAY include an HTTP Accept request-header (HTTP 14.1) specifying MIME type, application/xrds+xml.

### **6.2.4 Response**

The response to the initial request MUST comply with the HTTP protocol.

The response MUST be one of:

1. An HTML document with a <head> element which includes a <meta> element with http-equiv attribute, X-YADIS-Location
2. An HTML document with HTTP response-headers which include an X-YADIS-Location response-header
3. HTTP response-headers only, which MAY include an X-YADIS-Location response-header
4. A document of MIME type, application/xrds+xml

Note: Since the response to the initial request MUST comply with the HTTP protocol, if the request is a GET and does not include an 'Accept: application/xrds+xml' header, the response MUST be of type 1 or 2. This is required because the Relying Party MAY omit the Accept and so might only look for the X-YADIS-Location.

### **6.2.5 Document**

If the response is a document of MIME type, application/xrds+xml, that document MUST be a YADIS Resource Descriptor (see Clause 7).

### **6.2.6 Resource Descriptor URL**

If the response includes an X-YADIS-Location response header or an HTML <meta> element whose http-equiv attribute is X-YADIS-Location the value MUST be an absolute URL. That URL is a YADIS Resource Descriptor URL; it MUST locate a YADIS Resource Descriptor.

The Relying Party MUST issue an HTTP GET to retrieve that YADIS Resource Descriptor document.

### **6.2.7 Second Request**

Unless a Relying Party using an HTTP Accept request-header obtains the YADIS Resource Descriptor in response to that request, that Relying Party MUST examine the response to determine if it contains a YADIS Resource Descriptor URL in either the HTTP response-headers or the HTML <head> element and request the document as specified in Clause 6.2.6.

Unless a Relying Party using an HTTP HEAD request obtains a YADIS Resource Descriptor URL in response to that request, that Relying Party MUST then issue an HTTP GET request and examine the response to determine if it contains a YADIS Resource Descriptor URL in either the HTTP response-headers or the HTML <head> element and request the document as specified in Clause 6.2.6.

### **6.2.8 Termination**

When a YADIS Resource Descriptor is returned to the Relying Party the YADIS Protocol terminates.

Note: Further steps depend on the service specifications obtained from the YADIS Resource Descriptor and the intent of the Relying Party; they are outside of the scope of the current YADIS Specification. For example, a Relying Party may decide, in the next step, to authenticate the user agent through OpenID and then perform a LID Profile Exchange.

If none of the requests succeed in obtaining a YADIS Resource Descriptor then the URL used in the initial request is not a YADIS URL or there has been a failure.

## **7. The YADIS Resource Descriptor**

### **7.1 Overview of the YADIS Resource Descriptor**

Note: We considered providing both a text and an XML-based format, in order to make it as easy as possible for Relying Parties to use service information available through YADIS IDs. However, we came to the conclusion that requiring both a text and an XML based format provides only small additional value, and using an XML based format enables use of standard parsing tools and allows service specifications to include additional information without upsetting parsers that do not understand those extensions.

The YADIS Resource Descriptor is based on a simple, extensible XML document called an Extensible Resource Descriptor (abbreviated XRD). The format of XRD documents is being specified by the XRI Technical Committee of OASIS (see the XRI Resolution 2.0 specification and Clause 7.5 of this specification, Schemas for the YADIS Resource Descriptor).

The YADIS Resource Descriptor provides a list of identifiers of Services. These are the Services that know the User identified by the YADIS ID used to obtain the YADIS Resource Descriptor; these services act on behalf of that YADIS User. In the case of some Services, additional data is included for use by the Relying Party in making a request to that Service. Such additional data is not specified in the YADIS Specification but is specified in the definition of that Service.

The Resource Descriptor also enables the User to specify which of the Services it prefers be used.

## 7.2 A simple YADIS Resource Descriptor

Here is an example of a small YADIS Resource Descriptor:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*($v*2.0)">
  <XRD>
    <Service>
      <Type>http://lid.netmesh.org/sso/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/sso/1.0</Type>
    </Service>
  </XRD>
</xrds:XRDS>
```

This Resource Descriptor specifies two services.

## 7.3. Element definitions

### 7.3.4 XRDS

A YADIS Resource Descriptor document consists of an XRD container (an XRDS element) with a YADIS Resource Descriptor (an XRD element).

#### 7.3.1 XRD

A YADIS Resource Descriptor is an XRD element containing a sequence of Service elements. Each Service element is a service description.

The order of the service descriptions is not significant.

A YADIS Resource Descriptor document **MUST** include at least one XRD element.

If more than XRD element is included in a YADIS Resource Descriptor document, a Relying Party **MUST** examine at least the *last* XRD element. YADIS does not specify the meaning of additional XRD elements in a YADIS Resource Descriptor.

#### 7.3.2 Service

An XRD element in a YADIS Resource Descriptor **MAY** contain one or more Service elements, each describing a YADIS service.

Each YADIS Service element consists of one or more Type elements plus optional elements.

Note: We encourage identity protocol designers to be brief.

Note: A YADIS Resource Descriptor that has no Service elements describing a YADIS service is permitted.

### 7.3.3 Type

Each YADIS Service element MUST contain at least one Type element.

Each Type element MUST contain an identifier of some version of some service. This service identifier MUST be a URI or XRI.

For each service identified by a Type element there SHOULD be a service specification document. It is RECOMMENDED that the service identifier be a URL that can be opened in a standard web browser to display that service specification document.

It is RECOMMENDED that each service identifier include an explicit version identifier, in order to facilitate the evolution of the service in the future.

## 7.4 Other parts of a YADIS Resource Descriptor

Here is an example of a larger YADIS Resource Descriptor:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*(($v*2.0)"
  xmlns:openid="http://openid.net/xmlns/1.0">
  <XRD>
    <Service priority="10">
      <Type>http://openid.net/signon/1.0</Type>
      <URI>http://www.myopenid.com/server</URI>
      <openid:Delegate>http://jimmy.myopenid.com/</openid:Delegate>
    </Service>
    <Service priority="30">
      <Type>http://openid.net/signon/1.0</Type>
      <URI>http://www.example.com/openid</URI>
      <openid:Delegate>
        http://users.example.com/james.s
      </openid:Delegate>
    </Service>
    <Service priority="50">
      <Type>http://openid.net/signon/1.0</Type>
      <URI>http://www.livejournal.com/openid/server.bml</URI>
      <openid:Delegate>
        http://www.livejournal.com/users/jimmy/
      </openid:Delegate>
    </Service>
    <Service priority="20">
      <Type>http://lid.netmesh.org/sso/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/sso/1.0</Type>
    </Service>
  </XRD>
</xrds:XRDS>
```

### 7.4.1 URI element

A YADIS Service element MAY contain one or more URI elements. Each URI element specifies a resource that will provide the service.

Each URI element MUST contain a URI that resolves to a resource providing the service (or services) specified by the Type element(s) of that Service element.

That URI MUST be an absolute URL, not a relative URL, URL reference, or other URI.



Note: It is our intention that this element may contain any URI or may contain an IRI, as specified by RFC 3987. This will be specified in a later version of this specification.

If a Relying Party chooses a Service element that contains one or more URI elements, that Relying Party MUST recognize and use a URI element of that Service element.

If there is more than one URI element, the URIs in those elements MUST be equivalent for the purpose of using the service. A Relying Party MAY attempt to use any one or all of the URIs.

If one or more URI elements has a priority attribute, a Relying Party MAY use the priority values as specified in Clause 2.4.3 of the XRI Resolution 2.0 specification.

The order of URI elements is not significant.

The URI element is OPTIONAL. If a URI element is provided, the service specifies meaning of that element and the protocol to use with it.

#### **7.4.2 Priority attribute**

The OPTIONAL priority attribute of the Service element allows the User to specify preferences for the Service to be used. The example above indicates that the User prefers the OpenID protocol using the server, <http://www.myopenid.com/server>, and that the last choice is LID version 1.0. The LID SSO service does not contain a URI element because in LID SSO version 1, the LID URL itself identifies the identity service.

In keeping with the goal of ease of implementation, a Relying Party MAY ignore the priority attribute. A Relying Party that recognizes and uses the priority attribute in one or more Service elements MUST follow the specification of priority attributes in Clause 2.4.3 of the XRI Resolution 2.0 specification.

#### **7.4.3 Other elements in a Service element**

YADIS does not specify any other elements for a Service element. A Service MAY define and use other elements.

Note: This allows a Service element to indicate information that is specific to that service. As examples, the OpenID specification defines the optional Delegate element, which specifies a URL by which the OpenID server knows the User; the TypeKey specification defines the MemberName element.

If a Relying Party intends to chose a particular service, that Relying Party MUST recognize and use all elements specified by a service specification for inclusion in a Service element for that service.

A Relying Party MAY ignore all other elements in a Service element. A Relying Party MAY recognize and use any elements in a Service element, which are specified by the XRD schema.

A Relying Party using a YADIS Resource Descriptor MAY ignore any element of that document except for the elements specified here and any elements specified by a service specification.

Any Relying Party not using a particular service MAY ignore any element of YADIS Resource Descriptor that is specified by the service specification of that service. The use of elements specified by a service specification is determined by that service specification.

#### **7.4.4 Other elements in an XRD**

A Relying Party MAY ignore all elements in an XRD element other than the Service elements.

A YADIS service MUST NOT require the use of a Service element that does not identify that YADIS service in a Type element of that Service element.

A Relying Party MAY recognize and use any element specified by the XRD schema for inclusion in an XRD element.

A YADIS service MUST NOT require the use of any elements of an XRD other than the Service elements that identify the services of that YADIS Service.

## 7.4.5 Other services

The examples so far have been for single sign-on services. A YADIS service MAY offer other services. A YADIS service is NOT REQUIRED to offer a single sign-on service.

For example, the following Resource Descriptor specifies a number of LID services available at the resource identified by the user's YADIS ID:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*(Sv*2.0)">
  <XRD>
    <Service priority="10">
      <Type>http://lid.netmesh.org/sso/2.0</Type>
    </Service>
    <Service priority="20">
      <Type>http://lid.netmesh.org/sso/1.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/post/sender/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/post/receiver/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/relying-party/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/traversal/2.0</Type>
    </Service>
    <Service>
      <Type>http://lid.netmesh.org/format-negotiation/2.0</Type>
    </Service>
  </XRD>
</xrds:XRDS>
```

## 7.4.6 Other XRDSs and other elements in the XRDS

The XRDS schema permits more than one XRD in an XRDS and permits other elements in an XRDS, in addition to XRD elements. Such documents are valid YADIS Resource Descriptors.

A Relying Party MUST examine the *last* of the XRD elements in an XRDS returned in response to a request for a YADIS Resource Descriptor document.

A Relying Party MAY ignore all other elements in an XRDS.

## 7.5 Schemas for the YADIS Resource Descriptor

The schemas of the YADIS Resource Descriptor document are the XRDS and XRD schemas contained in this clause.

A YADIS Resource Descriptor document MUST conform to the XRDS schema. A YADIS Resource Descriptor is an XRD element; it MUST be contained in an XRDS element and MUST conform to the XRD schema.

It is our intention to reference the XRDS and XRD standards when and if they have been adopted by OASIS. In the meantime, we list the schemas here.

### 7.5.1 XRDS Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="xri://$xrds" elementFormDefault="qualified"
```

```

xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xrds="xri://$xrds">
  <!-- Utility patterns -->
  <xs:attributeGroup name="otherattribute">
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:attributeGroup>
  <xs:group name="otherelement">
    <xs:choice>
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:group>
  <!-- Patterns for elements -->
  <xs:element name="XRDS">
    <xs:complexType>
      <xs:sequence>
        <xs:group ref="xrds:otherelement" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="xrds:otherattribute"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

## 7.5.2 XRD Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="xri://$xrd*($v*2.0)" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xrd="xri://$xrd*($v*2.0)"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <!-- Utility patterns -->
  <xs:attributeGroup name="otherattribute">
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:attributeGroup>
  <xs:group name="otherelement">
    <xs:choice>
      <xs:any namespace="##other" processContents="lax"/>
      <xs:any namespace="##local" processContents="lax"/>
    </xs:choice>
  </xs:group>
  <xs:complexType name="URIpattern">
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attributeGroup ref="xrd:otherattribute"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="Stringpattern">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attributeGroup ref="xrd:otherattribute"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <!-- Patterns for elements -->
  <xs:element name="XRD">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="xrd:Query" minOccurs="0"/>

```

```

    <xs:element ref="xrd:Expires" minOccurs="0"/>
    <xs:element ref="xrd:Synonym" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="xrd:XSynonym" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="xrd:ProviderID"/>
    <xs:element ref="xrd:Service" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="xrd:otherelement" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="xrd:id"/>
  <xs:attribute ref="xrd:version"/>
  <xs:attributeGroup ref="xrd:otherattribute"/>
</xs:complexType>
</xs:element>
<xs:element name="Query" type="xrd:Stringpattern"/>
<xs:element name="Expires">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:dateTime">
        <xs:attributeGroup ref="xrd:otherattribute"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="Synonym" type="xrd:URIpattern"/>
<xs:element name="XSynonym" type="xrd:URIpattern"/>
<xs:element name="ProviderID" type="xrd:URIpattern"/>
<xs:element name="Service">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="xrd:Type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:MediaType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:Pattern" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:URI" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="xrd:ProviderID" minOccurs="0"/>
      <xs:group ref="xrd:otherelement" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="xrd:otherattribute"/>
  </xs:complexType>
</xs:element>
<xs:element name="Type" type="xrd:URIpattern"/>
<xs:element name="MediaType" type="xrd:Stringpattern"/>
<xs:element name="Pattern" type="xrd:Stringpattern"/>
<xs:element name="URI" type="xrd:URIpattern"/>
<xs:attribute name="priority" type="xs:nonNegativeInteger"/>
<xs:attribute name="version" type="xs:string" fixed="2.0"/>
<xs:attribute name="id" type="xs:ID"/>
</xs:schema>

```