

BS EN 16425:2014



BSI Standards Publication

## Simple Publishing Interface

**bsi.**

...making excellence a habit.™

**National foreword**

This British Standard is the UK implementation of EN 16425:2014.

The UK participation in its preparation was entrusted to Technical Committee IST/43, Information technology for learning, education and training.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2014. Published by BSI Standards Limited 2014

ISBN 978 0 580 78346 3

ICS 35.240.30; 35.240.99

**Compliance with a British Standard cannot confer immunity from legal obligations.**

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2014.

**Amendments issued since publication**

Date	Text affected
------	---------------

---

---

ICS 35.240.30; 35.240.99

English Version

## Simple Publishing Interface

Interface de publication simple

Schnittstelle für einfaches Publizieren (Simple Publishing Interface - SPI)

This European Standard was approved by CEN on 22 May 2014.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION  
COMITÉ EUROPÉEN DE NORMALISATION  
EUROPÄISCHES KOMITEE FÜR NORMUNG

**CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels**

## Contents

Page

Foreword.....	3
1 Scope .....	4
2 Terms and definitions .....	4
3 Requirements and design principles.....	5
3.1 General.....	5
3.2 Syntactic versus semantic interoperability .....	6
3.3 “By reference” and “by value” publishing.....	6
3.4 Flexible application.....	6
3.5 Objectives.....	7
4 SPI Model.....	7
4.1 General.....	7
4.2 Submit a resource.....	8
4.2.1 General.....	8
4.2.2 Resource submission by value .....	9
4.2.3 Resource submission by reference .....	10
4.3 Delete resource .....	12
4.4 Submit metadata .....	12
4.5 Delete metadata .....	14
4.6 Errors .....	14
4.6.1 General.....	14
4.6.1.1 Introduction .....	14
4.6.1.2 Method not supported .....	14
4.6.1.3 Invalid authorization token.....	14
4.6.1.4 Package type not supported .....	14
4.6.1.5 Content type not supported .....	14
4.6.1.6 Deletion not allowed .....	15
4.6.1.7 Invalid identifier.....	15
4.6.1.8 Invalid source location .....	15
4.6.1.9 Schema not supported .....	15
4.6.1.10 Metadata validation failure .....	15
4.6.1.11 Resource validation failure .....	15
4.6.1.12 Resource not retrieved .....	15
4.6.1.13 Overwriting not allowed .....	15
4.6.1.14 Method failure.....	15
4.7 SPI target configurations .....	15
4.8 Authentication.....	16
5 Conclusion .....	17
Bibliography .....	18

## **Foreword**

This document (EN 16425:2014) has been prepared by Technical Committee CEN/TC 353 “Information and Communication Technologies for Learning, Education and Training”, the secretariat of which is held by UNI.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by January 2015, and conflicting national standards shall be withdrawn at the latest by January 2015.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document contains the requirements for the Simple Publishing Interface (SPI), a protocol for storing educational materials in a repository.

This protocol facilitates the transfer of metadata and content from tools that produce learning materials to applications that persistently manage learning objects and metadata, but is also applicable to the publication of a wider range of digital objects.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

## 1 Scope

This European Standard specifies the Simple Publishing Interface (SPI), an abstract protocol for publishing digital content and/or the metadata that describes it into repositories in a way that preserves the references between the two. This protocol is designed to facilitate the transfer of learning materials from tools that produce learning materials to applications that manage learning objects and metadata. It is also applicable to the publication of a wider range of digital objects.

The objectives behind SPI are to develop practical approaches towards interoperability between repositories for learning and applications that produce or consume educational materials. Examples of repositories for learning include educational brokers, knowledge pools, institutional repositories, streaming video servers, etc. Examples of applications that produce these educational materials are query and indexation tools, authoring tools, presentation programs, content packagers, etc.

Whilst the development of the SPI specification draws exclusively on examples from the education sector, it is recognised that the underlying requirement to publish content and metadata into repositories crosses multiple application domains.

This abstract model has been designed to be implemented using existing specifications such as v1.3 Simple Web-service Offering Repository Deposit (SWORD) profile [SWORD], Package Exchange Notification Services [PENS] and the publishing specification that was developed in the ProLearn Network of Excellence [PROLEARN SPI]. The intent of this work is thus not to create yet another specification but to create a model that can be bound to existing technologies in order to make sure that these technologies are used in a way that takes into account requirements specific to the learning domain, where it is necessary to publish both content and metadata that references it in a way that preserves these references.

The SPI model enumerates the different messages that are interchanged when publishing metadata and content.

## 2 Terms and definitions

For the purposes of this document, the following terms and definitions apply and are used to distinguish the requester from the system that publishes an entity (a metadata instance or a learning object):

### 2.1

#### **source**

system that issues a publication request. Alternatively, this system can be labelled as requester

### 2.2

#### **target**

system to which publication requests are sent. This can be a repository component or a middle layer component. Such a middle layer component can fulfil several tasks. It can generate and attach metadata to a resource, disaggregate and publish more granular components or act for instance as an adapter to a third party publishing API (application programming interface)

**NOTE** The terms “client” and “server” have not been used in order to avoid any bias towards an interface that is only applicable in client/server applications. Moreover, the scenarios in which the API is used also envisage a source running on a server (e.g., publishing from within an LMS). In the remainder of this document, the terms “resource”, “digital content”, “learning object” and “educational material” are used interchangeably.

### 3 Requirements and design principles

#### 3.1 General

In this clause, some of the requirements for a publishing API are identified. These requirements stem from different repository architectures where learning resources and metadata instances need to be communicated across system boundaries. SPI enables applications to upload learning resources or metadata to a repository. For example, Figure 1 illustrates how an authoring tool (e.g., OpenOffice) could use SPI to upload a resource directly into a repository. A Learning Management System (LMS) (e.g., Moodle, Blackboard) could enable teachers to publish their materials transparently into a repository. By doing so, materials are simultaneously made available to students and published into a repository where they can be reused.

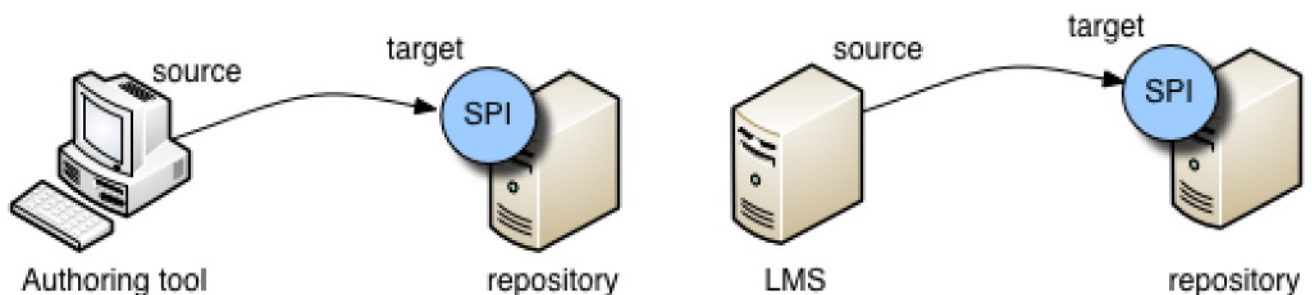


Figure 1 — Example SPI architectures

SPI also enables flexible architectures where a middleware component gathers learning resources or metadata through an SPI interface (from authoring tools or harvesters), applies value adding operations on these, and then stores them into a backend repository. Examples of such operations are disaggregation of material into small reusable components, automatic generation of metadata and validation or translation services.

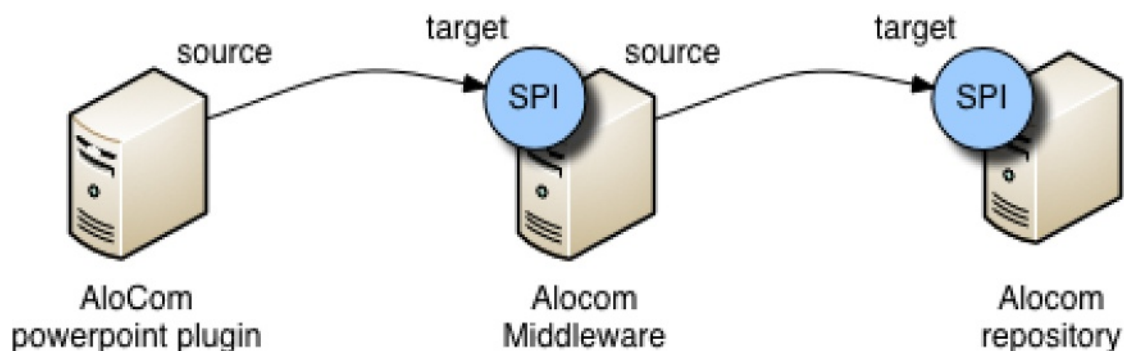


Figure 2 — AloCom architecture

Such architecture has been implemented in the context of the AloCom project (Figure 2). [ALOCOM]. This architecture contains a plug-in for MS PowerPoint, a source that can publish to a middle layer application, which is the target of this publishing operation. Next, the AloCom middleware disaggregates the material into small reusable components such as diagrams, individual slides, etc. and automatically generates metadata for each component. Each individual component is then published by the middleware component into a specialised AloCom repository where individual components are available for reuse. The AloCom middleware acts as a source and the AloCom repository as target.

Interoperability in both publishing steps is important. First, as several applications (not only MS PowerPoint) require publishing access to the middle layer application, the publishing process from within end-user

applications needs standardization. Secondly, the middle layer application shall be interoperable with other repositories, to promote interchangeability of components.

### 3.2 Syntactic versus semantic interoperability

The design of the SPI API is based on the design principles of the simple query interface (SQI) [SQI]. As such a simple set of commands that is extensible and flexible have been defined. By analogy with SQL, this protocol makes the following distinction between semantic and syntactic interoperability:

- syntactic interoperability is the ability of applications to deal with the structure and format of data. For instance, a language such as XML Schema Description (XSD) ensures the syntactic interoperability of XML documents as it allows for the parsing and validation of these documents;
- semantic interoperability refers to the ability of two parties to agree on the meaning of data or methods. When exchanging data, semantic interoperability is achieved when data are interpreted the same way by all the applications involved.

This European Standard tackles semantic interoperability for SPI. Without a binding (e.g., a REST binding) this specification cannot be implemented. A binding for SPI will realise syntactic interoperability.

### 3.3 “By reference” and “by value” publishing

Traditionally, two approaches allow for passing data from a source to a target:

- “by value” publishing embeds a learning object, after encoding, into the message that is sent to a target;
- “by reference” publishing embeds a reference (e.g., a URL) to a learning object to publish into the message that is sent to a target. This is different from publishing metadata in a referatory. Publishing in a referatory involves publishing metadata that contains a reference to the learning object. When publishing a learning object “by reference”, a reference to the learning object is used to fetch the learning object. This reference is not added to the metadata instance that describes the learning object but is used to retrieve the learning object before storing it internally.

“By value” publishing is useful for a standalone, desktop application that cannot be approached by a target in “by reference” mode. In this case, embedding a learning object in a message passed to the target lowers the threshold for pushing a learning object. “By reference” publishing is particularly suited when larger amounts of data need to be published. As embedding large files into a single message may cause degraded performance, a need exists to use a distinct method (e.g., FTP, HTTP, SCP, etc.) for transferring learning objects. Rather than imposing one of these approaches, the publish protocol will be designed to support both of them.

### 3.4 Flexible application

Some aspects of the SPI design follow existing applications and practices within the e-learning domain:

- a learning object referatory manages metadata that refer to learning objects stored on separate systems. Repositories that do not manage learning objects should thus be able to support SPI;
- some applications manage publishing learning objects without the metadata. For instance, PENS enabled applications submit packages to a server without metadata [PENS];
- SPI allows for publishing to repositories that manage both learning objects and metadata.

The MACE architecture for metadata enrichment [MACE] features different content providers that offer their metadata through an OAI-PMH target [OAI-PMH]. A general purpose harvester like the ARIADNE harvester is an example of a component that feeds metadata to a metadata referatory. Standardizing the publishing between the harvester and the metadata repository makes these components interchangeable.



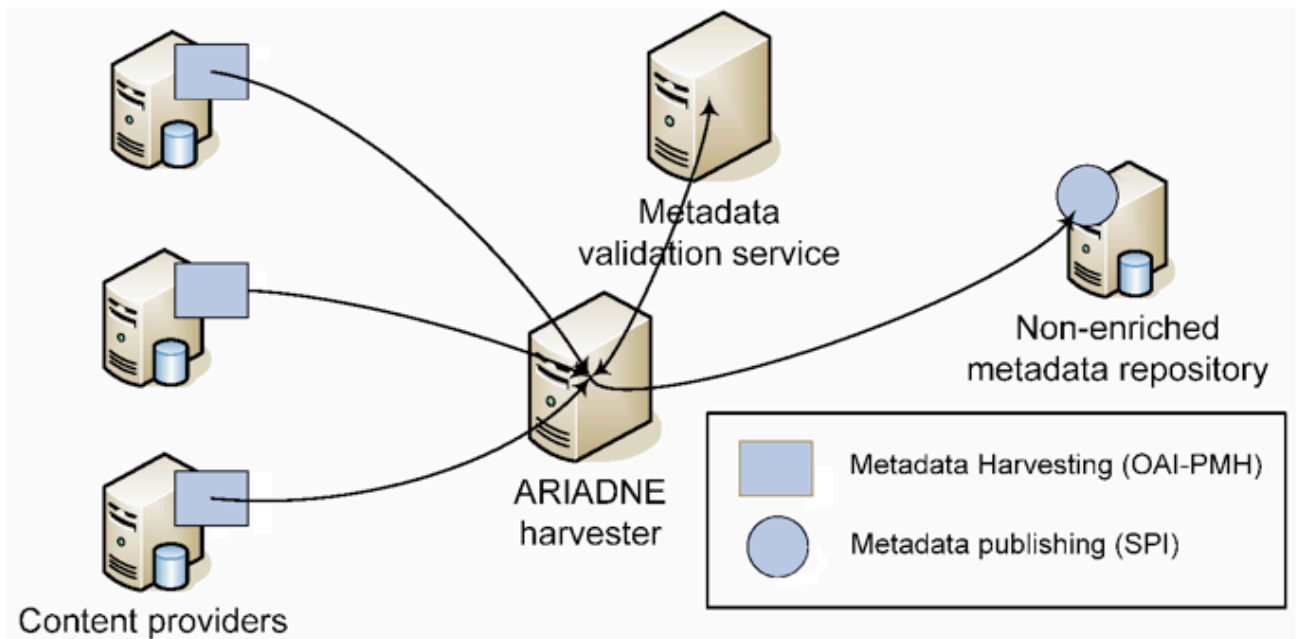


Figure 3 — The MACE harvesting architecture

### 3.5 Objectives

This publishing protocol meets the following objectives:

- SPI enables integrating publishing into authoring environments. This is beneficial for the author workflow, as they do not need to manually upload their learning objects using external publishing applications;
- SPI provides interoperability between applications that publish and applications that manage learning objects and metadata. Doing so, the effort of integrating publishing access into an authoring application can be reused on other learning object repositories, provided that they support SPI.

## 4 SPI Model

### 4.1 General

The model for SPI builds on a separation between data and metadata. The SPI model defines several classes of messages and functional units in a publishing architecture. When binding the specification to a given technology, these concepts are mapped into a concrete specification that can be implemented in a repository and for which conformance can be tested. All messages that are defined by the SPI model contain mandatory (M) and optional (O) elements. Mandatory means that a binding cannot relax this condition. A binding shall implement a mandatory attribute and shall make it mandatory as well. A binding can deal with optional elements in three ways. It can opt not to include the element, it can include the element and make it optional, or it can include the element and make it mandatory. A binding might for instance choose to not support transporting the filename attribute, or an SPI binding can offer support for the filename attribute while still allowing the source to provide a null value for this element. Depending on the choices made when implementing an SPI target, the latter can be configured in different ways and sources shall know the exact configuration of a target in order to be able to use it. As a consequence, the configuration of an SPI target shall be exposed to sources using at least one of the strategies presented in 4.7.

As shown by the class diagram of Figure 4, with SPI, a resource shall have an identifier (that can either be generated by a target or a source). In addition, the resource may have a filename associated. Every resource

can be described by zero, one, or more metadata instances. A metadata instance shall have a metadata identifier that identifies the metadata instance itself and shall have a resource identifier that is equal to the identifier of the resource. The metadata identifier (that can be either generated by the source or the target) enables distinguishing between multiple metadata instances referring to the same resource.



**Figure 4 — Resource and metadata instance**

In this model, a metadata instance shall be connected to a resource. However the resource may be hosted externally. In such a case, ingesting the resource is not part of the publishing scenario. For instance, when applying SPI to a referatory, only the messages described in 4.4 are implemented.

Alternatively, resources can be published without metadata. In this scenario, only the messages described in 4.2 are used. As an example, a single resource can be published to a repository. This scenario also includes the example of a file that consists of both data and metadata packaged in one content package.

Furthermore, this model also deals with a situation where multiple metadata instances describe the same resource.

The SPI model does not include explicit methods for updating resources or metadata instances. However, both metadata and resources can be deleted. Submitting an entity with an identifier that already exists in the target should be treated in one of the following ways by the target:

- the target overwrites the entity;
- the target creates a new version of the entity if it supports versioning;
- the target refuses to update the resource and returns an error.

Through the registry, a target can document which of the three options are supported.

## 4.2 Submit a resource

### 4.2.1 General

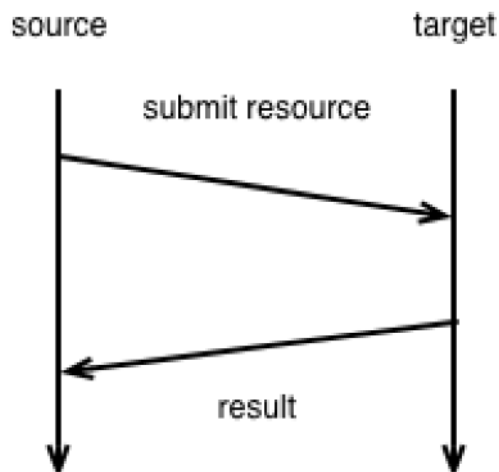
Submitting a resource involves sending a binary stream to a target. Depending on the binding that is used, this byte-stream can be encoded in various ways.

SPI defines two approaches for publishing a resource to a repository: “by value” and “by reference” publishing. As both methods are optional, an implementation can decide:

- to support both methods;
- to support only by reference ingesting of resources;
- to support only by value ingesting of resources; or
- not to support publishing of resources (i.e., only to support metadata publishing).

#### 4.2.2 Resource submission by value

Figure 5 illustrates how messages are interchanged when a “submit resource” request embeds the actual resource to be published. A source first sends a message containing a resource to a target. Then, the target replies either by acknowledging a successful ingest, or by returning an error message (see 4.6).



**Figure 5 — By value publishing of a resource**

The source can send several attributes with the message, which help the target to publish the resource. These attributes are either mandatory (M), which means that a source shall include them in the message, or might be optional (O), meaning that they can be omitted.

**Table 1**

Attribute	Description
Authorization token (O)	A token that enables the target to validate that the source is authorized to create a resource.
Identifier (O)	When this attribute is used, the source is responsible for generating an identifier for the resource. If this attribute is not present, a target shall generate an identifier that it returns through the result. A binding may decide not to offer support for source-generated identifiers by forbidding this attribute. When an identifier already exists in the repository, the target shall overwrite the existing resource or shall indicate that overwriting resources is not allowed or shall create a new version of the resource.
Resource (M)	The resource that will be published on the target.
Package type (O)	Identifies the kind of package (e.g., ADL SCORM, IMS Common Cartridge) that is being transmitted. A binding can for instance adopt the SWORD Content Package Types <sup>1</sup> to encode these. A target can use this attribute to reject the ingestion (for instance when it does not offer support for a particular package type). Furthermore, a target may use this information to unpack this package appropriately.
Content type (O)	Identifies the kind of resource that is being transmitted. A binding can for instance adopt the IANA MIME media types [IANA] to encode the

<sup>1</sup> <http://purl.org/NET/sword-types>.

	content type. In some scenarios or bindings, the source might be unaware of the content-type.
Collection[] (O)	Within a repository, several collections of data can be hosted. A source can publish a resource in multiple collections. A binding can restrict the cardinality of this attribute. When this parameter is omitted, a default collection is assumed by targets that host more than one collection.
Filename (O)	This filename captures the filename of the data that is transported. When no filename is present, a target might generate one, or might not use a filename.

After successful ingestion of a resource, the target shall return a message to the source. This message can be empty, acknowledging that the ingestion was successful, or can optionally contain one attribute. A target returns an error message (see 4.6) to indicate that a resource was not successfully ingested.

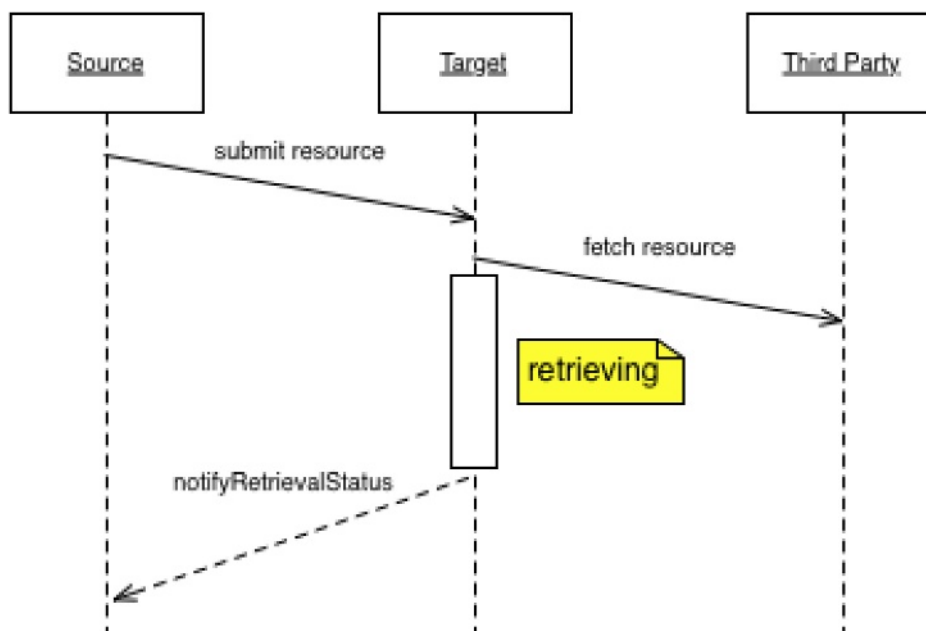
**Table 2**

<b>Attribute</b>	<b>Description</b>
Identifier (O)	The identifier of the resource that was published. Through using this identifier, the resource can be altered or deleted.  This attribute shall be present if the source did not propose an identifier. The target may repeat the identifier that was proposed in the request. If a target ignores the identifier that was proposed by the source, it shall have an identifier in the response.

### 4.2.3 Resource submission by reference

“By value” publishing embeds a learning object in the request, whereas “by reference” publishing embeds a reference and does not deal with the actual transfer of the resource to the target. This clause describes the second scenario. A source can use this approach to submit a reference (e.g., a URL or a resolvable location) to a learning object. The reference is then used by the target to obtain the learning object(s). In the meantime, a source might submit another reference, with a different identifier. Finally, when the target has successfully obtained the object, it sends a *notifyRetrievalStatus* message to the source.

“By reference” submission enables asynchronous handling of resources. A source can send for instance multiple resource messages to a target while it does not await the process of actually fetching the resource. When the target has finished fetching a resource, it will asynchronously notify the source. Simulating this behaviour in “by value” mode, would require the source to set up a thread for each single submit resource message.



**Figure 6 — By reference publishing of a resource**

The following table captures the attributes that are transferred to a target in “by reference” publishing mode.

**Table 3**

Attribute	Description
Authorization token (O)	A token that enables the target to validate that the source is authorized to create a resource.
Identifier (O)	When this attribute is used, the source is responsible for generating an identifier for the resource. If this attribute is not present, a target shall generate an identifier that it returns through the result. A binding may decide not to offer support for source-generated identifiers by forbidding this attribute. When an identifier already exists in the repository, the target shall overwrite the existing resource or shall indicate that overwriting resources is not allowed or shall create a new version of the resource.
Resource Reference (M)	A reference (e.g., a URL) to a location where the resource can be obtained.
Package type (O)	Identifies the kind of package that is being transmitted. A binding can for instance adopt the SWORD Content Package Types <sup>2</sup> to encode these. A target can use this attribute to reject the ingestion (for instance when it does not offer support for a particular package type). Furthermore, a target may use this information to unpack this package appropriately.
Content type (O)	Identifies the kind of content that is being transmitted. A binding can for instance adopt the IANA MIME media types to encode these content types. In some scenarios or bindings, the source might be unaware of the content-type.
Collection (O)	Within a repository, several collections of data can be hosted. A source can indicate that a resource is to appear in multiple collections. A

<sup>2</sup> <http://purl.org/NET/sword-types>.

	binding can restrict the cardinality of this attribute. When this parameter is omitted, a default collection is assumed by targets that host more than one collection.
Filename (O)	This filename captures the filename of the data that is transported. When no filename is present, a target might generate one, or might not use a filename.
Source (O)	This attribute refers to the source, indicating where the target shall deposit the NotifyRetrieval message. If a binding does not offer support for a NotifyRetrievalStatus message, this attribute can be omitted.

After successful ingest of a resource, the target should return a message to the source (NotifyRetrievalStatus). This message acknowledges that the ingestion was successful. A particular binding can decide to omit this acknowledge message.

**Table 4**

Attribute	Description
Identifier (O)	The identifier of the resource that was published. Through using this identifier, the resource can be altered or deleted.  This attribute shall be present if the source did not propose an identifier. The target may repeat the identifier that was proposed in the request.
Resource Reference (M)	This reference shall be equal to the reference that was submitted in the request and enables the source to identify which submission was successful.

### 4.3 Delete resource

The deleteResource message deletes a learning object on the target. A target is free in how to realise a delete: it can issue a real delete or can make the resource not retrievable.

**Table 5**

Attribute	Description
Authorization token (O)	A token that enables the target to validate that the source is authorized to delete this resource.
Identifier (M)	Indicates the resource to be deleted.

A target shall return either a message indicating that the deletion was successful, or that shows an error.

### 4.4 Submit metadata

The SPI model distinguishes between the submission of a metadata instance and the submission of a resource to a repository.

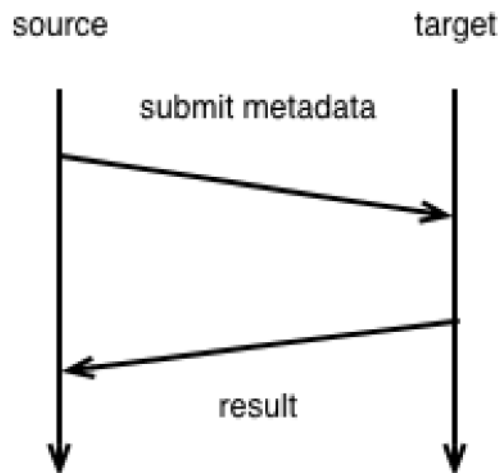
Some protocols do not make a distinction between requests for publishing metadata and requests for publishing resources. When binding SPI to these protocols one can opt to still publish metadata and resources with separate requests or package metadata with the resource and send both with one request to the target.

The SPI model recognizes the following situations:

- metadata can be packaged with the resource. In such a case, only the submit resource part of the model is implemented. SPI does not specify how a target should process a package with metadata inside;

- metadata are dealt with differently and is transported separately with the messages introduced in this clause.

This clause focuses on the second part of this model, where two methods are exposed: one for ingesting the metadata instance and one for ingesting the resource.



**Figure 7 — Publishing of a metadata instance**

Figure 7 illustrates that the process of ingesting a metadata instance into a repository is not very different from ingesting a resource in a “by value” mode. However, as shown in the table below, it requires specific attributes.

**NOTE** Because a resource can be described by several metadata instances, the concept of metadata identifier is introduced.

**Table 6**

<b>Attribute</b>	<b>Description</b>
Authorization token (O)	A token that enables the target to validate that the source is authorized to create a resource.
Metadata Identifier (O)	When this attribute is used, the source is responsible for generating an identifier for the metadata instance. If this attribute is not present, a target shall generate an identifier that it returns through the result. A binding may decide not to offer support for source-generated identifiers by omitting this attribute. When a metadata instance with this identifier already exists in the repository, the target shall overwrite the existing instance or shall indicate that overwriting metadata instances is not allowed.
Identifier (O)	This identifier refers to the resource this metadata instance is referring to.
Metadata instance (M)	The actual metadata instance that is sent to the target.
Metadata schema id (O)	A repository may support different metadata schemas and application profiles of a schema. This attributes enables a source to define what schema (or application profile) the instance validates against.
Collection (O)	Within a repository, several collections of (meta)data can be hosted. A source can indicate that a metadata instance is to appear in multiple collections. A binding can restrict the cardinality of this attribute. When this parameter is omitted, a default collection is assumed by targets that host more than one collection.

After successful ingest of a metadata instance, the target shall return a message to the source. When ingestion was successful, this message acknowledges the success. A success message can optionally contain one metadata identifier attribute. If the ingest was unsuccessful, an error message is returned (see 4.6).

**Table 7**

<b>Attribute</b>	<b>Description</b>
Metadata Identifier (O)	The identifier of the metadata instance that was published. Through using this identifier, the resource can later be manipulated (e.g., deleted). This attribute shall be present if the source did not propose an identifier. The target may repeat the identifier that was proposed in the request.

#### **4.5 Delete metadata**

The deleteMetadata message deletes a metadata instance on the target.

**Table 8**

<b>Attribute</b>	<b>Description</b>
Authorization token (O)	A token that enables the target to validate that the source is authorized to delete this metadata instance.
Metadata Identifier (M)	Indicates the metadata instance to be deleted

A target shall return either a message indicating that the deletion was successful, or that shows an error.

#### **4.6 Errors**

##### **4.6.1 General**

###### **4.6.1.1 Introduction**

This clause lists the various error messages that can be returned as a result of an SPI request.

###### **4.6.1.2 Method not supported**

All methods listed above are optional. A target can therefore indicate that in a particular binding or profile of SPI, a method is not supported.

###### **4.6.1.3 Invalid authorization token**

This error is returned when a target requires authentication and a source is not authorized or when the authentication token is invalid.

###### **4.6.1.4 Package type not supported**

The target does not offer support for the package type that is indicated in a “submit resource by value” or a “submit resource by reference” message.

###### **4.6.1.5 Content type not supported**

The target does not offer support for the content type that is indicated in a “submit resource by value” or a “submit resource by reference” message.



**4.6.1.6 Deletion not allowed**

The target indicates that the resource or metadata instance cannot be deleted.

**4.6.1.7 Invalid identifier**

The (metadata) identifier that is submitted or deleted by the source is invalid. This message can only occur in a submit resource or a submit metadata scenario.

**4.6.1.8 Invalid source location**

The source attribute that is submitted by the source is invalid. This message can only occur in a submit resource by reference scenario.

**4.6.1.9 Schema not supported**

The metadata schema id attribute in a submit metadata request, refers to a schema that is not supported by the target.

**4.6.1.10 Metadata validation failure**

This error can be returned, if a target supports validation of metadata. This error indicates that the metadata that is sent to the target does not validate. For instance the metadata are not well formed or does not validate against the metadata schema.

**4.6.1.11 Resource validation failure**

This error can be returned, if a target supports validation of resources. This error indicates that the resource sent to the target does not validate. For instance if the target detects that a SCORM package is not valid, it can return this error.

**4.6.1.12 Resource not retrieved**

The resource that was sent in a “submit resource by reference” scenario could not be retrieved.

**4.6.1.13 Overwriting not allowed**

With this message, a target indicates that overwriting the resource or metadata instance is not allowed. This message can be returned when submitting a resource or when submitting metadata.

**4.6.1.14 Method failure**

The method failed for a reason not listed above. An optional parameter enables specifying the nature of the error.

**Table 9**

Attribute	Description
message (O)	An explanation of the nature of the failure.

**4.7 SPI target configurations**

An SPI target can be configured in different ways, meaning that it can support different combinations of the features described in this document. In order to connect to an SPI target, it is necessary to know what features are supported - i.e. the target configuration. Ideally this configuration should be documented in a machine-readable format so that it can be used to automatically connect to an SPI target. There exist multiple ways to expose the configuration of a target to client systems:

1. Various protocols offer ways to document the capabilities of a service:
  - In the SRU/W community, the ZeeRex specification is used to explain the capabilities of an SRU/W service [ZEEREX];
  - In SWORD/APP, service documents list the available collections with their capabilities. (e.g., content packages supported, MIME types that are accepted).
2. The IMS LODE Registry [LODE] specification proposes a data model for representing repositories, their collections and the protocols they support.

Whatever the solution adopted to describe the capabilities of an SPI target, it should cover the target properties described in the table below.

**Table 10**

<b><i>SPI Target Property</i></b>	<b>Description</b>
<i>Binding name and version</i>	The SPI binding used by the target.
<i>Accept resource</i>	Does the target support the ingestion of digital resources?
<i>Accept resource by reference</i>	Does the target support the ingestion of digital resources by reference?
<i>Accept resource by value</i>	Does the target support the ingestion of digital resources by value?
<i>Is resource deletion supported</i>	Can resources be deleted?
<i>Package type</i>	Defines the kind of content packages that the target accepts.
<i>MIME type</i>	Enumerates a list of MIME types that can be ingested.
<i>Resource validation</i>	Does the target validate resources?
<i>Accept metadata</i>	The target can ingest metadata as a separate instance.
<i>Is metadata deletion supported</i>	Can metadata be deleted?
<i>Is metadata validation supported</i>	Does the target validate metadata?
<i>Update policy</i>	
<i>Overwrite</i>	Resubmitted resource with an existing resource identifier overwrites the existing resource with this identifier.
<i>Version</i>	Resubmitted resource with an existing resource identifier leads to the creation of a new version of the resource.
<i>Forbid</i>	Resubmitted resource with an existing resource identifier is rejected.
<i>Accept anonymous connection</i>	The target does not require credentials.

#### **4.8 Authentication**

Authentication and authorization is outside the scope of this model. SPI however assumes an authorization token to be created that allows a source to communicate with a target.

A (stateful) authorization token can build on the simple session management concept that is further described in CWA 15454 [SQI]. It is assumed that a session is established before any further communication can take place.

Alternatively, a stateless scenario can be supported. For instance by using HTTP, one could use the basic access authentication schema that concatenates a username and password and encodes it to an authentication token with the base64 algorithm [HTTP-AUTHENTICATION].

## **5 Conclusion**

This European Standard contributes a model for publishing metadata and resources to digital repositories. The model that is presented can be applied to many scenarios:

- publishing from an authoring tool to a repository;
- publishing from an indexation tool to a repository;
- publishing from a metadata harvester to a repository cache;
- publishing to intermediate layers that process content and/or metadata.

Although the model can also be used to synchronize metadata and content between repositories, SPI is not the best way to handle synchronisation or replication. Synchronisation and replication are easier to realise with notification services or harvesting protocols (e.g., OAI-PMH).

## Bibliography

- ALOCOM VERBERT K. DUVAL E. (2007). Evaluating the alocom approach for scalable content repurposing. In EC-TEL, pages 364-377
- IANA IANA MIME MEDIA TYPES. <http://www.iana.org/assignments/media-types/>
- [LODE] IMS Learning Object Discovery & Exchange (LODE). <http://www.imsglobal.org/lode/>
- [MACE] Wolpers, M.; Memmel, M.; Giretti, A. Metadata in Architecture Education – First evaluation Results of the MACE System. Springer Lecture Notes in Computer Science
- OAI-PMH OPEN ARCHIVE INITIATIVE PROTOCOL FOR METADATA HARVESTING. see <http://www.openarchives.org/OAI/openarchivesprotocol.html>
- PENS PACKAGE EXCHANGE NOTIFICATION SERVICES. see <http://www.aicc.org/docs/AGRs/agr011v1.pdf>
- [PROLEARN SPI] Ternier, S.; Massart, D.; Demidova, E.; Olmedilla, D.; Memmel, M. and DUVAL E. D4.8 SPI – The Simple Publishing Interface, [http://my.confolio.org/portfolio/files/Prolearn-KUL/KUL/Participation/WorkPackages/WP\\_04\\_-\\_KUL\\_\\_leader\\_/Deliverable\\_4.8/PROLEARN\\_D04.08.pdf](http://my.confolio.org/portfolio/files/Prolearn-KUL/KUL/Participation/WorkPackages/WP_04_-_KUL__leader_/Deliverable_4.8/PROLEARN_D04.08.pdf)
- SQI SIMON B. MASSART D., VAN ASSCHE F., TERNIER S., DUVAL E., eds. (2005) A simple query interface specification for learning repositories. CEN Workshop Agreement (CWA 15454)
- RFC2119 BRADNER S. Key words for use in RFCs to Indicate Requirement Levels. <http://www.ietf.org/rfc/rfc2119.txt>, March 1997
- SWORD SIMPLE WEB-SERVICE OFFERING REPOSITORY DEPOSIT (SWORD). <http://swordapp.org/>
- ZEEREX ZeeRex: The Explainable Explain Service, <http://zeerex.z3950.org/>
- ATOMPUB Joe Gregorio and Bill de hOra, Atom Publishing Protocol. <http://tools.ietf.org/html/rfc5023>
- HTTP-AUTHENTICATION AUTHENTICATION H.T.T.P. Basic and Digest Access Authentication. <http://tools.ietf.org/html/rfc2617>
- CWA 15454, *A Simple Query Interface Specification for Learning Repositories*



# British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

## About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards-based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

## Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at [bsigroup.com/standards](http://bsigroup.com/standards) or contacting our Customer Services team or Knowledge Centre.

## Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at [bsigroup.com/shop](http://bsigroup.com/shop), where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

## Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to [bsigroup.com/subscriptions](http://bsigroup.com/subscriptions).

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

**PLUS** is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit [bsigroup.com/shop](http://bsigroup.com/shop).

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email [bsmusales@bsigroup.com](mailto:bsmusales@bsigroup.com).

## BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

## Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

## Copyright

All the data, software and documentation set out in all British Standards and other BSI publications are the property of and copyrighted by BSI, or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use. Except as permitted under the Copyright, Designs and Patents Act 1988 no extract may be reproduced, stored in a retrieval system or transmitted in any form or by any means – electronic, photocopying, recording or otherwise – without prior written permission from BSI. Details and advice can be obtained from the Copyright & Licensing Department.

## Useful Contacts:

### Customer Services

**Tel:** +44 845 086 9001

**Email (orders):** [orders@bsigroup.com](mailto:orders@bsigroup.com)

**Email (enquiries):** [cservices@bsigroup.com](mailto:cservices@bsigroup.com)

### Subscriptions

**Tel:** +44 845 086 9001

**Email:** [subscriptions@bsigroup.com](mailto:subscriptions@bsigroup.com)

### Knowledge Centre

**Tel:** +44 20 8996 7004

**Email:** [knowledgecentre@bsigroup.com](mailto:knowledgecentre@bsigroup.com)

### Copyright & Licensing

**Tel:** +44 20 8996 7070

**Email:** [copyright@bsigroup.com](mailto:copyright@bsigroup.com)



...making excellence a habit.™