Scope, Technical Requirements, Approaches, and Recommendations

Granular Identifiers and Metadata for CCSS (GIM CCSS) Project

Daniel R Rehak

V 1.0
2013-09-12

Contents

Executive Summary

This document outlines key technical issues, requirements and solution approaches for the various aspects of the *Granular Identifiers and Metadata for CCSS (GIM CCSS) Project*. It presents recommendations for the technical solution. All recommendations are summarized below.

The recommendations focus on:

- Creating a **digital representation** of the Common Core State Standards (CCSS) based on a well-defined information architecture.
- Assigning, using and managing **identifiers** associated with statements within the CCSS.
- Providing **metadata** describing the statements within the CCSS.

The recommendations address the core *technical* use cases:

- **Linking** – how identifiers of standards statements are used to link standards statements to representations in external systems.
- **Processing** – how the machine-readable form of the CCSS will be used.
- **Metadata** – how metadata describing the CCSS will be used.

These use cases support of the broader requirement of identifying and representing concepts within a CCSS statement.

The document uses key concepts in the recommended solution:

- Standards **statements** that express concepts – the information model treats all types of statements the same.
- **Identifiers** that label statements and other concepts – from an identifier you cannot discern what is being identified.
- A **digital representation** for the model of the standards statement comprised of the description of the standard and its **metadata** – the metadata is solely provenance information.

This document is part of a collection that details the entire solution, including:

- Expressing the information model in JSON and XML.
- APIs.
- Recommendations for publication and deployment.
- Use and best practices.

Summary Recommendations

Technical Use Cases

Supported Use Cases: identifiers and linking, processing the digital representation, metadata.

Scope Recommendation: Supporting the linking use cases is in scope.

Scope Recommendation: Documenting how to use links in other system, e.g., the approach for using links in Learning Registry paradata or LRMI, is out of scope. The ongoing discussions of how to reference statements is best handled in the fora for these other systems as it has no significant impact on the structure of the identifiers for the statements or other aspects of the technical solution. Best practice documentation will include guidelines for common link encodings, e.g., in JSON, XML, or micro formats that can be used by other systems.

Scope Recommendation: Supporting the processing use cases is in scope.

Scope Recommendation: Supporting the provenance metadata use cases is in scope.

Scope Recommendation: Alignment of the digital representation with LRMI and schema.org (i.e., encoding the statement characteristics using schema.org micro format tags within an HTML web page holding the statement text) so that a search engine can be used to discover a standards statement in the “open web” is out of scope.

Scope Recommendation: Representing scope and sequence is out of scope.

Scope Recommendation: Developing end user tools is out of scope.

Scope Recommendation: Use cases not included or subsumed from the above are out of scope for the current effort.

Digital Representation Coverage

The standards, extensions and other data that need to be modeled and supported in the digital representation.

Recommendation: Develop an open, extensible model for the digital representation and infrastructure to support existing and future (including non CCSS) standards and extensions.

Recommendation: Use formal vocabularies, either existing when available or standard/community specific.

Recommendation: Use more general vocabularies instead of more specific vocabularies when appropriate. Use DC Terms, SKOS, schema.org, LRMI, CEDS, etc., when appropriate.

Standards Models

The features of models to represent statements.

Recommendation: Supporting the technical use cases requires a richer standards model than currently used for the digital representation of CCSS.

Metadata Models

Provenance metadata for standards and their digital representation.

Recommendation: Include provenance metadata for both the original source standard and the encoding of the digital representation as separate entities in the information model.

Recommendation: Model metadata using an existing metadata model. Use the Dublin Core vocabulary (consistent with schema.org and LRMI).

Recommendation: Include metadata at the statement level in the digital representation and information model. Include metadata for the standard as a whole in the digital representation and information model.

Versioning

Supporting versions of standards and statements, identifiers linking to versions and metadata versioning.

Recommendation: Include version information at any level in the digital representation and information model. Support version information for the standard as a whole in the digital representation and information model.

Recommendation: Represent versions in the metadata. Use relationships to relate different versions.

Recommendation: Represent language translations of a standard or statement as a type of version.

Recommendation: Provide a distinct identifier for each version of a statement. Do not include an additional "most recent" version identifier.

Recommendation: Do not include versioning of metadata in the digital representation or the information model; versioning is only for the source statement as a whole or the digital representation as a whole, not its metadata.

Licensing

Assigning open licenses to the standards and statements.

Recommendation: Include licensing metadata for both the original source statement and the encoding of the digital representation as separate entities in the metadata part of the digital representation and information model.

Recommendation: Include licenses at the statement level in the digital representation and information model. Include licenses for the standard as a whole in the digital representation and information model.

Recommendation: Release all developed software under the Apache 2.0 license.

Recommendation: Release the conceptual model and all documentation under the Creative Commons Attribution 3.0 Unported license (CC BY 3.0).

Information Architecture

Conceptual design of the technical solution: digital representation, vocabularies, identifiers.

Recommendation: Use a rich information model that has built-in flexibility and extensibility. Develop corresponding best practices of how to use the information model (all or parts) to encode CCSS and granular standards statements, including how to use classification, vocabularies and relationships. The entire information model need not be used to encode CCSS standards and statements.

Recommendation: For any vocabulary-based attribute in a standards statement, support associating the vocabulary definition with the value.

Recommendation: Model vocabularies as first class objects in the digital representation and information model; each vocabulary is defined as an object in the digital representation, either through a reference to the digital form of an external vocabulary or as a model of the vocabulary value space.

Recommendation: Base the information model of an internal vocabulary on an existing, well-defined model, e.g., SKOS, Zthes.

Recommendation: Do not mandate an internal form or technical solution for the data store.

Recommendation: For data exchange, use JSON as the primary encoding of the digital representation. Provide RESTful service interfaces to return either JSON or XML encodings. Consider adding in JSON-LD contexts, identifiers and typing.

Recommendation: Encode the narrative text of the standards statement in HTML5. Use a limited subset of HTML5.

Recommendation: Allow multiple identifiers per statement.

Recommendation: Provide sustainable, basic identifier management system.

Recommendation: Use opaque identifiers for statements and all other elements of the information model.

Recommendation: Use an identifier system that provides structured attributes and meets the infrastructure and deployment criteria. Base the identifier system on the PILIN guidelines.

Services

Access to the digital representation.

Recommendation: Provide an access only fully RESTful API.

Recommendation: Do not include a query interface; use the query language of any selected data store.

Recommendation: Engage the community in developing an open source standards browser.

Workflow

Creating the digital representation.

Recommendation: Use the *Encoded Workflow*. It has less impact on developers and requires less project work. Future transition to a Born Digital Workflow is possible. Publication in different forms from the digital representation is still possible.

Infrastructure and Operational Systems

Options for a deployed, sustained, operation infrastructure and standards statement distribution.

Recommendation: Deploy a system using a statement publishing agent that includes a data store, access, integration with an identifier management system and integration with 3rd party systems.

Recommendation: Use an existing publishing agent if one is available that provides the required capabilities.

Recommendation: Support multiple standards per publishing agent. Enable multiple publishing agents.

Support

User and technical support items.

Project Work Item: Develop selected support items in conjunction with content group after the digital representation and infrastructure choices have been finalized.

Sustainment

Sustaining and supporting the digital representation and any operational deployment after project completion.

Recommendation: To the extent possible, separate sustainment of the statements and their digital representation (edits, versions) from any operational system.

Recommendation: Minimize required new sustainment operations. Rely on or leverage existing ongoing activities.

Project Work Item: Develop sustainment plan.

Introduction

This document outlines key technical issues, requirements and solution approaches for the various aspects of the *Granular Identifiers and Metadata for CCSS (GIM CCSS) Project*. It presents recommendations for the technical solution.

The overall project objective is to "*develop a free, open, and faithful digital representation of the Common Core State Standards at a level of granularity that provides the necessary support for instructional, assessment and professional development implementation by states, districts, publishers, technology companies and others*."

This document addresses technical issues surrounding:

- Creating a **digital/machine readable/processible representation** of the Common Core State Standards (CCSS) based on a well-defined information architecture.
- Assigning, using and managing **identifiers** assigned to granular statements within the CCSS.
- Providing **metadata** describing the granular statements within the CCSS.

Topics covered include:

- **Technical Use Cases** – outlining the key technical problems to be solved:
    - **Linking** Use Cases – how identifiers of standards statements are used to link standards statements to representations in external systems.
    - **Processing** Use Cases – how the machine readable form of the CCSS will be used.
    - **Metadata** Use Cases – how metadata describing the CCSS will be used.
- **Digital Representation Coverage** – scoping which standards, extensions and other data needs to be supported by the digital representation and identifiers.
- **Standards Models** – organizing the structure of the CCSS and other standards models to support use and processing.
- **Metadata Models** – representing metadata for CCSS statements.
- **Versioning** – managing different versions of CCSS statements.
- **Licensing** – applying license terms to the digital representation.
- **Information Architecture** – creating a digital representation of the CCSS that is also capable of representing other similar standards.
- **Services** – providing interfaces and APIs for other systems to access the digital representation of the CCSS.
- **Workflow** – supporting the process of developing, publishing and managing the digital representation of the CCSS.
- **Infrastructure and Operational Systems** – deploying, distributing or running systems and software for managing the digital representation of the CCSS and identifiers, including integration with other systems, along with distributing the digital representation of the CCSS.
- **Support** – providing support and documentation for end users of the digital representation.
- **Sustainment** – addressing how to sustain any ongoing technical work and deployed systems.

Each of the topics is presented in a different section below. Many are interdependent. The presentation repeats some of the issues and may address some points in a different way under different topics.

The document includes a glossary of key terms and references to external sources, along with a list of related project documents describing other parts of the technical solution.

The next section defines some key assumptions and terms.

Key Terms and Assumptions

Understanding key terms: granular statements, identifiers, metadata.

As part of level setting, the following key assumptions govern the remainder of the document. These are expressed through key terms: granular statements, identifiers, metadata. These terms are used as described herein in the project documents.

Standards are written in terms of **statements**. Statements may combine multiple concepts. Granular statements decompose and catalog other statements to uncouple or identify existing concepts. Except for decomposing and cataloging the concepts within a statement, there is nothing that fundamentally differentiates a granular statement from any other (more or less granular) statement, and cataloging of concepts is applicable to all statements. The technical approach does not define what is a granular statement, it only defines how to represent a (any) statement, including references to its decomposition and the concepts contained within the statement.

Throughout "statement" refers to both original (CCSS) and more granular statements. The "granular" qualifier is omitted throughout unless essential in differentiating a granular statement from an existing CCSS statement.

Standards statements are identified. **Identifiers** are labels ("names") attached to a standards statement or labels attached to other data items. There is nothing that differentiates an identifier for an existing standards statement from the identifier attached to a granular statement; one cannot discern granularity by examining any identifier.

Corollary: there is no concept of a "granular identifier". Identifiers refer to things that may be granular or not. The identifiers themselves are not (more or less) granular.

Note: The document uses a formal set of terms to describe identifiers, based on formal model for identifiers and identifier systems. Specific terms are defined in the glossary. Model details can be found in the PILIN reference in the list of references.

The digital representation of a standards statement includes a description of the statement e.g., its classification within the overall standard (e.g., grade, topic, strand), along with information about its provenance. Collectively, all of the information about the statement *could* be treated as **metadata**. Herein, the description of the statement and its metadata are treated as two independent concepts. Thus the digital representation contains two parts: one describes the nature of the standard in a machine readable form without any description of its provenance; the other is the metadata that is used to detail the provenance of the statement. By definition, metadata for a standards statement has no educational meaning and is only provenance data, whose characteristics are domain independent and are similar to those used to catalog any other resource.

Technical Use Cases

Supported Use Cases: identifiers and linking, processing the digital representation, metadata.

The technical approach and priorities are driven by the different use cases. There are three major sets of technical use cases:

- Linking Use Cases – how identifiers of standards statements are used to link CCSS statements into representations in external systems.
- Processing Use Cases – how the digital representation of the CCSS will be used.
- Metadata Use Cases – how metadata describing the CCSS will be used.

Note: The technical use cases underlie the core use case of taking an existing CCSS statement and identifying the existing granular concepts. Making a granular statement requires that the technical approach supports granular statements, but as stated, there is nothing fundamentally special about a granular statement (representation or processing). Thus the use cases focus on the technical criteria for all statements.

Linking Use Cases
Identifiers of the standards statements are used in other systems, i.e., they are *external* links to standards statements:

- An RTTA assessment item links to the granular CCSS statement being assessed.
- An RTTA assessment result links to the granular CCSS statement being assessed.
- The LRMI metadata for a learning resource links to the CCSS statement covered in the learning resource.
- The Learning Registry paradata about a learning activity links to the CCSS statement covered in the learning activity.

Analysis: Each of these can be addressed by storing the identifier of the statement in the other system, e.g., creating a link. There are no particular requirements beyond having identifiers for the statements; the form of the identifier does not matter as long as it can be encoded in the other system's representation. This linking does NOT, per se, require a digital representation of the statements, e.g., it could be sufficient for the link to resolve to a section in the text of the standard or to a simple file that is digitally "opaque" and only meaningful to a human reader. Thus linking is uncoupled from representation.

Scope Recommendation: Supporting the linking use cases is in scope.

Scope Recommendation: Documenting how to use links in other system, e.g., the approach for using links in Learning Registry paradata or LRMI, is out of scope. The ongoing discussions of how to reference statements is best handled in the fora for these other systems as it has no significant impact on the structure of the identifiers for the statements or other aspects of the technical solution. Best practice documentation will include guidelines for common link encodings, e.g., in JSON, XML, or micro formats that can be used by other systems.

Processing Use Cases
Using a digital representation of the CCSS statements (including the core CCSS tree of statements, anchor standards, granular statements and state 15% extensions):

- A user or system uses the identifier of a statement to access and return the full description of the statement, i.e., link resolution. This only requires that the identifier is associated with a location holding the statement and that there is a resolver. Resolution does not require a comprehensive digital representation of statements, e.g., it could be sufficient to return just the text of the statement for human consumption. However, it does require that identifier resolution and managed storage of statements be part of a sustained infrastructure.
- A user or system uses an *existing* 3rd party identifier of a statement to access and return the statement (multiple identifiers for statements exist and are in use). New identifiers for granular statements must coexist with the existing identifiers. This requires that the identifier solution support multiple identifiers.
- Authors and editors use the digital representation as the definitive source of the statements, i.e., authoring/editing is done by working on the digital representation and all presentation (e.g., XML, PDF, HTML) of the standard and its statements is generated from the authoritative digital source. This requires

authoring tools, but complexity of representation is driven by the other processing use cases. See the section on *Workflow* for more details on authoring.

- States develop extensions of, or revisions to, CCSS statements (i.e., the 15% rule). Consumers want to access the original statements and these extensions in the same format. This requires the representation and dissemination model support statements from different sources.
- A State publishes their standard that includes those statements adopted from CCSS and their local modifications. This requires that the representation be capable of indicating modifications and that users can identify the collection of statements adopted by a state.
- Developers produce updates and revisions to individual statements and consumers need to be able to differentiate these changes (in additions to extensions). This requires that the representation support version and provenance metadata throughout the graph of statements.
- Users encode, access and use all types of standards (CCSS statements, state 15% extensions, anchor standards, other extensions or granular statements, non CCSS statements, state standards, district guidelines, objectives) in a common way. This requires that the representation supports encoding a variety of types of statements.
- Users navigate between different related statements in any way: up or down the tree, across facets, across related statements. This requires a rich representation of relationships between statements.
- Users browse and view the standards based on facets, not following the current hierarchy or any predefined organization. This requires a faceted representation and a faceted viewer. This could also enable comparison of different statements.
- Users identify the existing concepts with a statement. This supports alignment of resources to statements.
- Users perform gap analysis and comparisons across multiple standards (CCSS, others, state extensions to CCSS). If there are multiple standards encoded in the same digital store, or if there are local extensions, then an analysis could be performed to determine where there are gaps between standards or extensions. This requires a rich, structured representation.
- Consuming systems access and retrieve all or part of the digital representation for 3rd party use, processing and storage. This requires the ability to access or publish the representation for external consumers.
- Users search for statements that have particular characteristics. This requires a search capability combined with an adequate representation to support indexing or taxon/faceted classification. Providing search tools is out of scope for the current effort, but the representation needs to support search.
- Users develop educational scope and sequence for learning resources related to standards, and develop curricular maps. Representing scope and sequence is out of scope, but the digital representation needs to enable 3rd party development of scope and sequence, curricular maps, etc.

Analysis: The digital representation and underlying information model needs to be rich enough to support the encoding of different types of statements, their maintenance and relationships, in support of a variety of uses. The representation should enable unanticipated uses, but should not include capabilities that can be layered onto the model or developed by 3rd parties or through follow-on work. The digital representation needs to be coupled with a sustainable identifier system, including resolution and support for multiple identifiers per statement.

Scope Recommendation: Supporting the processing use cases is in scope.

Metadata Use Cases

How metadata describing the CCSS will be used:

- Users access information about a particular standard or statement, e.g., status, version, when created, date of publication, history of activities.
- Users access information about the digital form of a standard or statement, e.g., status, when it was encoded, who did the encoding, licensing.

Analysis: Provence metadata can be modeled using traditional cataloging metadata, e.g., Dublin Core. This assumes there is no need to have additional cataloging or provenance metadata (the digital representation of a statement is essentially a custom form of metadata describing the educational nature [facets] of the statement, but as noted, this is not considered metadata herein).

Scope Recommendation: Supporting the provenance metadata use cases is in scope.

Scope Recommendation: Alignment of the digital representation with LRMI and schema.org (i.e., encoding the statement characteristics using schema.org micro format tags within an HTML web page holding the statement text) so that a search engine can be used to discover a standards statement in the "open web" is out of scope.

Scope Recommendation: Representing scope and sequence is out of scope.

Scope Recommendation: Developing end user tools is out of scope.

Scope Recommendation: Use cases not included or subsumed from the above are out of scope for the current effort.

Digital Representation Coverage

The standards, extensions and other data that need to be modeled and supported in the digital representation.

Standards Coverage
The current CCSS covers two standards, ELA and Math. Additional standards are under development. States have the option, as part of adoption, to change a portion of a standard, by extension, elimination, or replacement (the 15% rule). Other standards exist.

A digital representation may be designed to hold:

- CCSS ELA and Math standards only.
- Additional specific standards.
- Other future standards (through an open model of what can be included).
- Other existing standards (through an open model of what can be included).
- The standards as is without modification or extension.
- Any state or other jurisdictional changes to any part of the standards.

Analysis: Any option except that of only a single versioned, granular extensions to the current as is CCSS leads to providing a general open model that can support a full range of additions and extensions. Use cases require supporting CCSS core standards, anchor statements, state extensions, other standards and extensions. Different infrastructure options can be used to store multiple standards in a single digital store or to allow multiple stores with one standard per store. These options are independent of the overall coverage. Decisions on representations and coverage, however, should not be made on infrastructure choices; the different infrastructure options can be adapted to any of the coverage options listed.

Recommendation: Develop an open, extensible model for the digital representation and infrastructure to support existing and future (including non CCSS) standards and extensions.

Vocabularies
Descriptions of statements rely on vocabularies of terms, e.g., grade level, relationships, classification facets. Vocabularies may be:

- Existing formal, externally defined generic (i.e., non educational) vocabularies, e.g., SKOS relationships, DC terms.
- Existing formal, externally defined educational vocabularies, e.g., DoED education level.
- Standard-specific or community-specific vocabularies, e.g., CCSS classifications.
- Open vocabularies (no formal definition of the list of terms).

Alternatively, descriptions may be natural language statements (i.e., not encoded with vocabulary terms) or classifications may be hard coded into the structure of the information model.

Analysis: Natural language statements are flexible but add complexity to discovery (they require full-text indexing and search) and introduce the potential for inconsistencies. Hard coding vocabularies limits modeling to the predefined choices. Using vocabularies provides flexibility but requires that they be modeled and represented.

Recommendation: Use formal vocabularies, either existing when available or standard/community specific.

Several vocabularies are widely used to support the modeling of provenance metadata and educational characteristics, including:

- DC Terms for metadata
- SKOS for knowledge representation
- Schema.org for objects (includes)
- LRMI for educational resources
- CEDS 2.0 / CEDS 3.0 for educational characteristics

Analysis: Existing vocabularies cover many of the key concepts needed to describe a standards statement, and some are widely used in other domains. Using general vocabularies from other domains increases interoperability.

Recommendation: Use more general vocabularies instead of more specific vocabularies when appropriate. Use DC Terms, SKOS, schema.org, LRMI, CEDS, etc., when appropriate.

Technical Work Item: Identify vocabularies for the different specific elements of the information model that are vocabulary based.

Technical Work Item: Define and model CCSS-specific classification vocabularies.

Standards Models

The features of models to represent statements.

Model Features

There are several existing models of standards and statements, including CCSS, Ed-Fi, SLC, CEDS, SIF, ASN, ACARA. The models all share a number of similarities but all have some different characteristics (◘ indicates a discernible feature of the current CCSS model):

- All support multiple standards. ◘
- All models are rooted at a standard (each of the multiple standards are independent). ◘
- All distinguish between the model of the standard and the model of the statement.
- All have identifiers at the statement level. ◘
- Some represent a standard as a directed tree, each statement has a single parent and a statement is a refinement of its parent. ◘
- Some represent the tree through "has-parent" relationships. ◘
- Some represent the tree through "parent-of" ("has children") relationships.
- Some allow multiple relationships between statements.
- Some allow relationships between versions or between other standards.
- Some represent a standard as a directed graph; each statement can have multiple parents.
- Some represent a standard using a fixed taxon structure used to classify the levels in the tree/graph. ◘
- Some represent a standard using an arbitrary taxon structure.
- Some have a mixture of an arbitrary taxon structure with a set of fixed terms.
- Some have additional attributes describing the educational characteristics of the statement.
- Some have standards statements that are text only.
- Some have standards statements that include images or other media.
- Some support standards statements in multiple languages.
- Some include extensible attributes describing educational characteristics of the standards statements.
- Some include notes and annotations about the standards statements.
- Some have provenance metadata.
- Each has a different set of metadata.
- Some support versioning. ◘
- Some support versioning at any level in the tree.
- Some support versioning only for terminal nodes in the graph. ◘
- Some distinguish metadata for the source standard or statement from the metadata about the digital representation.
- Some are based on a relational model (SQL).
- Some are based on a document markup language model (XM). ◘
- Some are based a semantic model (RDF).
- Some support formal (externally defined) vocabularies for values of educational characteristics and taxon facets.
- Some deployments provide services and APIs for access and retrieval of statements.
- Some deployments provide mechanisms for dissemination of the complete standard.

Note: The above list may not be exhaustive.

CCSS Model Features

The CCSS model supports a subset of the general list of features above.

Note: There is very little formal documentation of the CCSS model; primarily the project summary web page. The following is from attempting to reverse engineer the model from the publically available documentation and XML instances.

CCSS uses the following fixed taxon vocabulary to classify statements, which is framework specific:

Math	**ELA/Literacy**
Initiative	Initiative
Framework	Framework
Set	Set (optional)
Grade	Strand+Domain
Domain	Grade
Cluster	Standard
Standard	Component (optional)
Component (optional)	

The CCSS model supports multiple standards through separate XML documents and indicates the standard (and that it is a CCSS standard) by convention through encoding the CCSS-specific vocabulary for the standard designation in the first two parts of the `<StandardCode>` element, i.e., *Initiative* + *Framework*.

The CCSS model is rooted at the standard. By convention, *Initiative* is the top level in all `<StandardCode>` elements. Multiple roots are permitted.

The CCSS model distinguishes between the model of the standard and the model of the statement.

The retrievable CCSS XML instances do not include any information about the standard as a whole. There is no data describing the entire standard. The retrievable XML document for the full standard is a (PSVI unordered) list of individual `<LearningStandardItem>` elements for *Standards* and *Components* included in a single `<LearningStandards>` element that is otherwise empty.

The retrievable CCSS XML instance includes only the individual standards statements for *Standards* and *Components*. The model is capable of representing the other levels in the standard, but these are not included in any available XML instance of a full standard. Manual retrieval of the XML for each level in the hierarchy does not return any additional information about that level – it returns only the terminal *Standards* and *Components* for the requested level in the classification hierarchy.

The CCSS model is a directed tree (hierarchy) with the levels in the tree defined by the explicit taxon path shown above.

- In the CCSS XML, the position in the tree for a `<LearningStandardItem>` element is specified by:
    - The level from the root (numeric)
    - The taxon name

  These two elements appear to be redundant – knowing one defines the other – but there is no automatic enforcement of the relationship.
- The `<StatementCode>` element encodes the full path of taxon names from root to the `<LearningStandardItem>` element. The level could be inferred from the path (although some statements do not align, and grade is problematic for mutli-grade statements/clusters). The path would also be redundant w.r.t. the defined relationships if the entire hierarchy were present in the digital representation; querying the entire graph returns the path.

The CCSS model incorporates grade level in both the taxon structure and as an explicit element in the model.

Introducing finer-grained granular statements than *Component* would require additions to the taxon structure.

In the CCSS XML instance each element has a single parent and an element is a refinement of its parent.

The CCSS XML instance uses "child-of" relationships to represent the tree. Even though the parent identifiers are present for all *Standards* and *Components*, the parent elements are only present in the available XML instances for *Components* (i.e., links to their parent *Standards*). The retrievable XML instance does not contain any elements above the *Standards*.

From the digital representation you can:

- Find the *Standard* for a *Component* (search of the XML for the parent identifier).
- Find the list of all *Components* for a *Standard* (search for all items with the same parent identifier).
- Obtain the identifier of the immediate parent of a *Standard* (*Cluster*, *Grade*), but the corresponding `<LearningStandardItem>` element for the parent is not present.

There is no apparent mechanism to walk the tree from root to leaf or leaf to root through the XML.

A separate, nonpublic, list of all GUIDs does exist, but this is not tied to the XML instances.

The CCSS model is capable of holding other relationships, but none appear to be used and there is no definition of acceptable values for `RelationType`:

- Other (non authoritative) XML instances could include multiple relationships between statements, i.e., model a directed graph.
- Other (non authoritative) XML instances could include relationships between versions or between other standards.
- Other (non authoritative) XML instances could include multiple relationships between different standards.

The CCSS model has multiple identifiers for statements at all levels in the hierarchy:

- The CCSS-curated dot notation identifier.
- The CCSS-curated locator, encoded as a web-resolvable URL form of the dot notation identifier, homed to the corestandards.org
- A non resolvable GUID generated using the DCE UUID algorithm (RFC 4211, but algorithm specific options are not available) (unique only within the DCE context).

Both the dot notation identifier and locator are based on the open taxon vocabulary and are transparent identifiers. Manual curation is used to ensure uniqueness and that the locator resolves to the appropriate statement. The locator and dot notation may be derived from each other and both are equally machine readable. There does not appear to be any mechanism to resolve a GUID to return the `<LearningStandardItem>` or to return a locator of the `<LearningStandardItem>` (with the exception of searching the XML for the GUID). As noted, not all elements of the hierarchy are included in the retrievable XML and thus the complete set of associations between GUIDs and `<LearningStandardItem>` for all statements in the standard are not discernible from the XML.

The CCSS model does not assign identifiers to concepts not in the hierarchy.

The standards statement is text only. The statement may include HTML markup. Supported HTML elements or HTML version cannot be discerned from the retrievable XML instances.

The entire `<LearningStandards>` element is defined in some natural language identified through an `xml:lang` attribute. The XML statement text does not include a definition of the language used in the statement. Identifying the `xml:lang` attribute of the entire collection is the only way to discern the language of a statement.

The CCSS model does not include additional characteristics describing educational aspects standards statements.

The CCSS model does not include notes or annotations about the derivation or rationale for standards statements.

The CCSS model does not include provenance metadata, either for the source CCSS or the XML encoding (the documentation indicates that some metadata exists but this is not disseminated).

The CCSS model does not include any licensing statements, either for the source CCSS or the XML encoding.

The CCSS model digital representation is XML. There is no public statement of the formal definition/schema used. The XML documents do not include any schema (XSD) or any `DOCTYPE`. There is no mechanism to automatically validate any document semantics, e.g., any element value may be an arbitrary type, the vocabularies are not enforced.

Without a defining schema, the XML is extensible, and any other (non authoritative) XML instances could include other elements, redefine or delete elements.

It is not discernible if the digital representation of the CCSS standards is the XML encoding or if there is some other underlying digital representation. The nature of the actual digital store (e.g., file, database) is also not discernible.

Most of the XML elements are collections, permitting multiple values. With the exception of `<GradeLevel>`, this feature does not appear to be used.

The vocabularies and semantics for the taxon, grade, statement codes, etc., are not formally defined.

- The complete set of values is not in any digital representation.
- There is no automatic enforcement that values come from controlled vocabularies.
- There is no definition of how to map the source statement to the taxon facets.
- There is no definition of how to extend the vocabularies and taxons to support more fine-grained statements.

The CCSS model supports encoding of other complete standards (e.g., state standards) at the top-level of the hierarchy. There is no documentation on how to support local extensions at any other level in the hierarchy.

The CCSS model supports versioning of the leaves of the hierarchy – *Standard* or *Component*.

- Version information is encoded (only) in the terminal part of the dot notation identifier and URL-encoded locator. It is not directly encoded as a model attribute or as a metadata attribute.
- There is no direct mechanism to determine all versions of a *Standard* or *Component*.
- There is no direct mechanism to determine the most current version of a *Standard* or *Component*.
- There is no mechanism to describe versioning at any level except for a *Standard* or *Component*.

The corestandards.org web site offers basic services to return the HTML or XML representation of a statement given a known URL-encoded locator. There are no other (e.g., RESTful, WS*) interfaces. There is no resolver for GUIDs.

Analysis: The CCSS model has some flexibility in modeling (e.g., complete standards, other relationships, open taxon structure, multiple values, open vocabularies, no schema). But it is also brittle (e.g., versions are limited to leaves, extensions are only for whole standards, identifiers are tied to the taxons, GUIDs are not resolvable). It is inconsistent with respect to when a concept is modeled through an open structure and when it is fixed and encoded directly in the model. There are some apparent redundancies, when combined with the lack of formally published semantics, make it easy to author invalid statements and difficult to automatically detect errors.

In general, there is no clear rationale for the differences between the various standards models listed above; design choices are typically not available in the public documentation of a model.

In addition to the preceding analysis, more detailed analyses of specific model features are presented in subsequent sections, along with specific recommendations that are based on project goals, use cases and common design principles, e.g., consistency, simplicity, openness. These recommendations are combined to produce the information model.

Recommendation: Supporting the technical use cases requires a richer standards model than currently used for the digital representation of CCSS.

Metadata Models

Provenance metadata for standards and their digital representation.

As described under Key Terms and Assumptions, metadata is about the provenance and description of the standard or standards statement. It is educationally agnostic.

The creation of the source standard and statements and the encoding in the digital representation are two separate processes. They have separate provenances and their metadata, albeit similar, is curated separately. There are multiple existing metadata models, e.g., LOM, Dublin Core, LRMI, schema.org.

Analysis: Metadata models such as LOM and Dublin Core are sufficient to express provenance. LRMI and schema.org encoding of metadata within web pages is currently out of scope.

Recommendation: Include provenance metadata for both the original source standard and the encoding of the digital representation as separate entities in the information model.

Recommendation: Model metadata using an existing metadata model. Use the Dublin Core vocabulary (consistent with schema.org and LRMI).

Metadata may be applied to:

- The whole standard.
- Individual standards statements within the whole.

Analysis: Standards may be created as a whole or updated and extended at the statement level. Thus different parts of the tree of standards statements may have different metadata. Versioning is recommended to be modeled at the statement level. Consistency in modeling also dictates metadata also be modeled at the statement level.

Recommendation: Include metadata at the statement level in the digital representation and information model. Include metadata for the standard as a whole in the digital representation and information model.

Versioning

Supporting versions of standards and statements, identifiers linking to versions and metadata versioning.

Standards and Statement Versioning
Standards and standards statements will evolve over time, e.g., additions, deletions, deprecations, fixing errors. Versioning options (for both the source statement and digital representation) include:

- Once published, a version of the standard is immutable; any change is a new version of the whole standard.
- Individual statements within the whole may be changed; each statement is versioned independently. Statement versions are uncoupled from the overall standard version.

Analysis: Workflow processes determine when a number of individual statement version changes trigger a global version change and how a versioned source statement propagates to produce a versioned digital representation. The current CCSS model supports versioning for the standard as a whole and statement-level versioning only for leaf nodes in the tree and encoding of versioning in the URLs.

Recommendation: Include version information at any level in the digital representation and information model. Support version information for the standard as a whole in the digital representation and information model.

Version information needs to be included in the information model. Options include:

- Encode the versioning information as a unique version number facet.
- Encode the versioning information within the metadata of the statement.
- Treat versioning as a relationship between two instances of a standard or a statement, modeled through a relationship facet between instances. The relationship may be expressed directly in the information model or within the metadata.

Analysis: Including versioning as an explicit facet is straightforward, but produces a somewhat brittle model. Versioning information is not a descriptive characteristics, thus it is metadata. If relationships are part of the digital representation, supporting versioning through relationships is a consistent approach.

Recommendation: Represent versions in the metadata. Use relationships to relate different versions.

Translated Standards
The statement text is expressed in a particular natural language. The standard (or statement) may be translated into a different language. Options include:

- A translation is a separate instance of a standard or statement.
- A translation is a type of version of a standard or statement.

Analysis: Within the library community, translations are typically modeled as versions (see FRBR).

Recommendation: Represent language translations of a standard or statement as a type of version.

Identifiers Linked to Versions
Identifiers link to a statement. There are different options for managing identifiers for versions of statements.

- An identifier for a single version of the statement that is immutable. A new version is given a new identifier. Resolving the identifier returns the associated version of the statement. The user must check to determine if the returned statement is the most current version or check its status.
- An identifier is for the most recent version of the statement. A user may need a mechanism to determine if there are older versions.
- There are multiple identifiers per statement, one that always resolves to the most recent version and one for every version that is immutable.

Analysis: Identifying a known version directly is required; this implies an identifier per version. Determining if there is a newer version of a statement may be complex, indicating that having a link to "most recent" may be required. But obtaining an older version from a specific version is equally complex. Use cases for "most recent" versus "specific version" are unclear. Any solution for linking identifiers to versions introduces complexity somewhere in the overall solution. Having only the "most recent" version identifier does not seem to be useful.

Recommendation: Provide a distinct identifier for each version of a statement. Do not include an additional "most recent" version identifier.

Metadata Versioning
Metadata about a statement may change independently from the statement itself. Such changes are limited to correcting errors in the metadata or adding additional information to the metadata – any change to the statement itself requires a new set of metadata for the revised statement.

When a change to the metadata is made:

- Metadata may be changed directly without any recording of version history.
- Metadata may be fully versioned.

Analysis: Versioning metadata adds complexity and there are no requirements to track versions of metadata.

Recommendation: Do not include versioning of metadata in the digital representation or the information model; versioning is only for the source statement as a whole or the digital representation as a whole, not its metadata.

Licensing

Assigning open licenses to the standards and statements.

The granular statements and the digital representation are to be licensed under an open license. The license is still TBD.

The creation of the statements and encoding in the digital representation are two separate processes. The statements and their digital representation may have separate provenance and thus may have different licenses. Further, a granular statement or an extension may have a different license from its parent or source statement. Licenses, terms, attribution, etc., are typical provenance metadata.

Analysis: Metadata and provenance models include necessary license attributes. Separate metadata for the source and digital representation supports typical workflow and the technical use cases.

Recommendation: Include licensing metadata for both the original source statement and the encoding of the digital representation as separate entities in the metadata part of the digital representation and information model.

Licenses may be applied to:

- The whole standard.
- Individual standards statements within the whole.

Analysis: Standards may be created as a whole or updated at the statement level. A granular statement may have a different license from its parent source statement. Thus licenses are applied at the statement level. Consistency with other parts of the information model also dictate that licenses be applied at the statement level.

Recommendation: Include licenses at the statement level in the digital representation and information model. Include licenses for the standard as a whole in the digital representation and information model.

Project Work Item: Select licenses and rights statements for statements.

Recommendation: Release all developed software under the Apache 2.0 license.

Recommendation: Release the conceptual model and all documentation under the Creative Commons Attribution 3.0 Unported license (CC BY 3.0).

Project Work Item: Determine if the specification for the digital representation requires a specification license with patent terms (versus a document © license).

Information Architecture

Conceptual design of the technical solution: digital representation, vocabularies, identifiers.

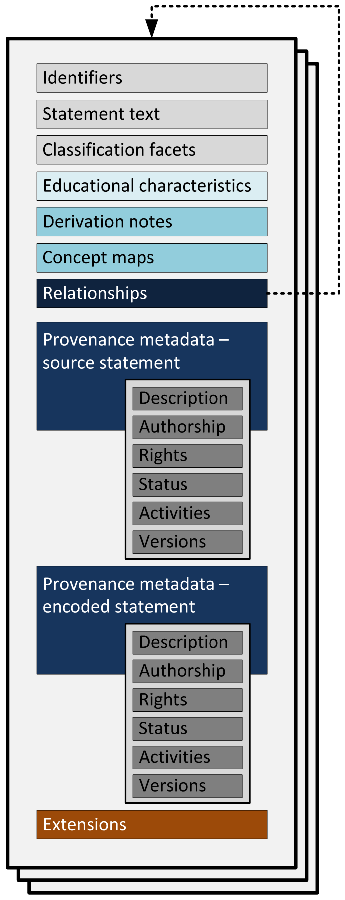
The conceptual design includes:

- The model of standards and standards statements, supporting the digital representation.
- The description of vocabularies used within statements.
- A data store and encoding for holding the digital representation of the statements and vocabularies.
- An identifier management system for creating, managing and resolving identifiers.

Information Model

The information model includes (but is not limited to):

- The identifier(s) assigned to the standards statement.
- The encoding of the natural language text of the standards statement (possibly including media elements).
- The classification facets/taxon for the standards statement.
- Educational descriptors for the standards statement.
- Notes about the derivation of the standards statement.
- Concept maps identifying existing concepts within the standards statement.
- Provenance metadata for the original source statement, including (but not limited to):
  - General description (e.g., title)
  - Authorship
  - History/Activities
  - Versioning
  - Licensing, Rights and Attribution
  - Language
  - Status/State
- Provenance metadata for the digital encoding of the statement, elements as above.
- Relationships between this standards statement and other statements.
- Optional extensions.



In support of the technical use cases, the information model provides:

- Multiple arbitrary identifiers and locators per statement.
- Multiple descriptive notes.
- Multiple classifiers per statement.
- Verifiable vocabulary-based classification.
- Metadata, versioning, language translation and licensing information at any level in the hierarchy of statements.
- Multiple relationships between a statement and another statement in any standard. Relationships between statements support representing both tree and graph structures. In the representation of a complete standard, the relationships must produce a rooted graph.

The information model does not distinguish between the description of a standard and the description of a standards statement. In other models these are two distinct entities, but their descriptions are similar. These models hold metadata only for the standard. The other difference is that the standard is the root of a tree of relationships, but any of the models that support multiple standards has multiple roots; there is no ability to have structure about a single standard. The distinction between standard and statement is convenient for the end user but can be modeled as just another classification facet. The use cases and other requirements dictate that metadata is stored for all statements, eliminating the need for the distinction between standard and statements. Any element needed for a standard is needed for a statement and vice versa. Thus the information model has only a single key concept of statement.

The information model does not define a fixed set of classification facets, a fixed relationship structure or a fixed set of descriptors. These are modeled as vocabularies.

Analysis: The information model encompasses the attributes from the models used in CCSS, CEDS 2.0, CEDS 3.0, Ed-Fi, SLC, ASN, ACARA. Standards statements in any of these models could be mapped to the information model. The model provides broad coverage, aims to be simple, consistent and extensible by describing concepts through modeled vocabularies versus fixed, hard-coded attributes.

Recommendation: Use a rich information model that has built-in flexibility and extensibility. Develop corresponding best practices of how to use the information model (all or parts) to encode CCSS and granular standards statements, including how to use classification, vocabularies and relationships. The entire information model need not be used to encode CCSS standards and statements.

Vocabularies
All standards models use one or more vocabularies, as described in the sections on *Digital Representation Coverage* and *Standards Models*. Vocabularies may be defined externally or may be specific to an encoding of a standard, including vocabularies used in extensions.

The range and source of vocabulary values used within the model or the description of a standard may be:

- Defined in the narrative description of the model, e.g., a statement that the value of grade level comes from the CEDS vocabulary.
- Defined in the specification of digital model, e.g., define a "gradelevel" type as the value space for the grade level entry. The value space may be:
  - Defined externally, e.g., a link/URI to an external vocabulary.
  - Defined inline in the model, e.g., the Ed-Fi XML XSD defines a "gradelevel" type.
- Defined within the entry, e.g., each vocabulary item includes the vocabulary namespace and vocabulary value as a pair. The value space may be:
  - Defined externally.
  - Defined inline in the model.

Analysis: Digital representations of vocabularies provide means to validate values and add more semantics to the overall representation of a statement. Using externally defined vocabularies promotes reuse of the vocabulary in different standards without replicating the definition.

Recommendation: For any vocabulary-based attribute in a standards statement, support associating the vocabulary definition with the value.

Recommendation: Model vocabularies as first class objects in the digital representation and information model; each vocabulary is defined as an object in the digital representation, either through a reference to the digital form of an external vocabulary or as a model of the vocabulary value space.

Recommendation: Base the information model of an internal vocabulary on an existing, well-defined model, e.g., SKOS, Zthes.

Data Store and Digital Encoding
The digital representation of the information model for standards statements and vocabularies needs to be stored within a processing and retrieval system (a data store) and instances of statements need to be exchanged with other systems.

Representation and encoding options include:

- A (normalized) entity-relationship (E-R) model for use in a relational database.
- A document markup language schema.
- A document-oriented model.
- A semantic model and ontology.

- Embedded micro format data in a web page.

The table below describes the data model definition, type of data store, structure of a statement instance and interfaces for each of the options.

Model	Model	Data Store	Statement Instance	Interface
Entity-Relationship	SQL data definitions and data types	RDBMS	normalized relational table(s)	SQL queries and service interface
Document Markup	XSD or other schema definition language	XML database or file system	XML document	service interface
Document Oriented	informal model or JSON Schema	NoSQL database	JSON document	RESTful service interface
Semantic	OWL or other ontology language	Triple store	RDF (encoded in XML)	SPARQL queries and service interface
Embedded	Micro formats	CMS or file system	RDFa micro formats embedded in HTML5	HTTP

Analysis: A rigid, fixed E-R/relational model does not align with the variability across different standards statements and the scope of multiple instances, versioning, and relations required by the technical use cases. Standards are too variable to be easily modeled as normalized tables; the resulting schema is complex and difficult to use. Sematic models are rich and powerful, but require specialized expertise to develop, maintain and use; there is little community expertise in technologies such as RDF and SPARQL. NoSQL and JSON representations are more forward looking, flexible and align with newer developments. XML documents and XSDs/schema definitions are a traditional encoding, but require a fixed model structure. Embedded micro format data present a different approach where data is stored *within* the statement text and surrounding descriptive web page content, rather than the narrative and statement data being encoded in the representation. Transformations from one encoding to another are generally feasible, and the internal store need not use the same storage representation as the data exchange representation.

Recommendation: Do not mandate an internal form or technical solution for the data store.

Recommendation: For data exchange, use JSON as the primary encoding of the digital representation. Provide RESTful service interfaces to return either JSON or XML encodings. Consider adding in JSON-LD contexts, identifiers and typing.

Recommendation: Encode the narrative text of the standards statement in HTML5. Use a limited subset of HTML5.

Technical Work Item: Define the JSON/JSON-LD encoding of the information model and vocabularies.

Identifier Model

Identifiers resolvable as locators are required to meet the technical use cases:

- Each statement requires at least one “unique” identifier.
- Multiple identifiers for an existing statement, from different sources, are in use.
- A service is needed that takes an identifier and returns a locator used to access the digital representation of the statement.
- Identifier resolution needs to be a sustainable service.

Analysis: Identifier systems can be designed to support multiple identifiers and support alignment of identifiers. Identifier systems can be used to support storage of statements within multiple data stores.

Recommendation: Allow multiple identifiers per statement.

Recommendation: Provide sustainable, basic identifier management system.

Identifiers may be:

- Opaque – having no semantic meaning.
- Transparent – having a discernible human understandable meaning and representation.

Analysis: In terms of the information model and services, transparent identifiers are not different from opaque identifiers – they provide identity and access to services such as finding a location or retrieving the attributes associated with the identifier. For the identifier system, the transparency is generally meaningless, semantics is neglected in resolution. Transparent identifiers (e.g., the CCSS taxon path identifiers) are easy for humans to read and understand but are brittle – any change in the model structure or data may invalidate an identifier. Similarly encoding data such as versions in an identifier makes the identifiers brittle. Incorporating complex information models such as concept mapping within an identifier make them difficult to construct and interpret. Identifier systems can be used to include attributes about the identifier object in the identifier data without exposing it in a transparent identifier.

Recommendation: Use opaque identifiers for statements and all other elements of the information model.

Recommendation: Use an identifier system that provides structured attributes and meets the infrastructure and deployment criteria. Base the identifier system on the PILIN guidelines.

Technical Work Item: Select/Specify an identifier system design.

Services

Access to the digital representation.

Service interfaces provide access to a data store of standards statements to retrieve one or more statements.

Service interfaces may be used to:

- Retrieve a single statement given its identifier.
- Retrieve a part of a statement given its identifier, e.g.:
    - The statement text.
    - The statement taxons, concepts or educational characteristics (i.e., not the metadata).
    - The statement metadata (source statement or digital encoding).
- Retrieve the collection of statements at any level in the hierarchy.
- Retrieve the collection of statements below a statement.
- Retrieve all parents (to the root) of a statement.
- Retrieve a collection of related statements.
- Perform create, update and delete operations on the data store.
- Query to find statements that match the query parameters.

Service interfaces can be provided through APIs built using service patterns. APIs can be based on:

- RESTful services.
- Web Services (using WS* standards).

Analysis: RESTful interfaces are consistent with existing approaches to accessing educational data. WS* is the traditional approach to providing enterprise APIs. Statements and their representations are curated through a well-managed workflow process and updated very infrequently; retrieval is the most common process. Query interfaces are powerful, but complex to develop and use and are often provided as part of the underlying data store. Developing an ad-hoc query interface is complex; if provided, the native query language of the data store should be used.

Recommendation: Provide an access only fully RESTful API.

Recommendation: Do not include a query interface; use the query language of any selected data store.

Recommendation: Engage the community in developing an open source standards browser.

Technical Work Item: Design a RESTful access API (GET only). Include capabilities to walk the tree, e.g., retrieval at any level in the tree. Use HTTP methods, headers and responses (versus URI encoding) to specify what to retrieve (full/partial, JSON/XML/HTML/PDF encoding) and describe responses. Follow the SLC API guidelines to design the API. Possibly include URL-encoded access in addition to using HTTP methods.

Workflow

Creating the digital representation.

The digital representation of a standard or a statement is produced and promulgated through the workflow process. There are two major alternative processes.

Encoded Workflow
The standard or statement is created (including newly created, updated, revised) through any current "off line" development and release process; this is the "source" statement. Statement developers may use any tools. They may use any process to release a draft standard or statement, accept feedback and publish in a non-machine readable form. At any point in the statement development workflow, the source statement is released to be encoded into the digital representation and released in digital form. This begins a separate encoding workflow. Publication of the digital form and traditional publication of the source statement are independently curated actions.

Analysis: The encoded workflow process does not require any substantial changes to the current development process for statements or any significant new tools to produce the source statement. It does require the separate encoding process. The source statement and the digital form may be out of sync. The source statement is always authoritative.

Born Digital Workflow
Statement developers are given a set of tools to create a standard or statement directly in digital form. Once created, it is immediately available in digital form and may be published in a traditional form from the digital source. Workflow processes are used to manage the source statement approval process, e.g., the digital form may be marked as "in development" and not available to external users until approved from the release process for the digital form.

Analysis: The born digital workflow requires new tools, processes and the ability to manage both the workflow for source statement development and approval and the workflow for digital release in a single platform. It enables authoritative publication in different forms from a single common source.

Recommendation: Use the *Encoded Workflow*. It has less impact on developers and requires less project work. Future transition to a Born Digital Workflow is possible. Publication in different forms from the digital representation is still possible.

Technical Work Item: As part of documenting best practices, in conjunction with the content working group, include a description of the encoding workflow that takes a source statement and produces the digital representation.

Infrastructure and Operational Systems

Options for a deployed, sustained, operation infrastructure and standards statement distribution.

Deployment Infrastructure
Potential project elements can be split into four groups:

- The source granular statements.
- Digital representation of the full CCSS (original plus granular statements) along with required metadata, vocabularies and identifiers for the standards statements.
- Open source software to support any of the required use cases.
- A deployed, supported operational system that any organization can use.

These four elements are independent:

- The source granular statements can be released and used independently of their digital representation or any deployed infrastructure.
- The digital data can be released and used in some other deployment that is independent from any operational system deployed through this project.
- The software can be used by others to support other standards represented in the same information model or to deploy a system that parallels one deployed from this project.
- The operational system can be used to support any standards represented in the same information model.

Setting up an operational system to support the processing use cases requires a commitment to sustainment along with a determination of the services provided. An operational system may include different capabilities:

- An accessible digital store of the statements.
- A set of services used to operate on the representation.
- An identifier management system.
- Integration with 3rd party systems, e.g., Assessment Consortia, Learning Registry.

The simplest approach is to deploy an operational system with all of the capabilities listed above. Software would be deployed, and statements would be published. Identifiers would be created and “homed” to the selected identifier management system.

It is possible to deploy only an identifier management system without setting up a data store. Any 3rd party could then deploy their own instance of the data store (using the provided open source or other software). The digital representation of the statements would have identifiers that are created when the digital encoding is created and which are homed to and managed by the identifier management system. Additionally, creating extensions or versions of statements by 3rd parties require that they have access to the identifier management system. The identifier management system could be deployed specifically for this effort, or an existing 3rd party identifier management system could be used.

It is also possible to distribute the digital representation with “non homed” identifiers, i.e., identifiers that are unique but that do not resolve and are not managed. Any 3rd party would then need to “home” the identifiers as part of deploying a data store and an identifier management system. Creating extensions or versions of statements by 3rd parties requires that their identifiers be unique within the context used for the CCSS statement identifiers or that they be in a different context.

To distribute the digital representation without deploying any software (data store or identifier management system) the identifiers must be designed to support rehoming and distributed resolution. Providing persistent identifiers requires a set of identifier services – either through a set of centrally provisioned services or through service peers. Any peered service requires a management plan to coordinate peers, and should provide a “resolver of last resort”.

Analysis: Distributing data and software without deployment is technically feasible. The solution is more complex than providing a single deployment coupled with an identifier management system.

Recommendation: Deploy a system using a statement publishing agent that includes a data store, access, integration with an identifier management system and integration with 3rd party systems.

Deployment Design
The typical workflows of publishing and accessing a standards statement are illustrated in Figure 1 (the diagram is illustrative, and does not show all potential workflows or use cases). These include:

- Statement identification.
- Statement publication (including identifier assignment or homing).
- Statement access (retrieving statements and resolving identifiers).
- Integration with the Learning Registry (submitting information about statements to the Learning Registry).
- Integration with external systems (such as the RTTA assessment consortia for purposes such as item alignment).
- Statement discovery through interfaces built using the Learning Registry.

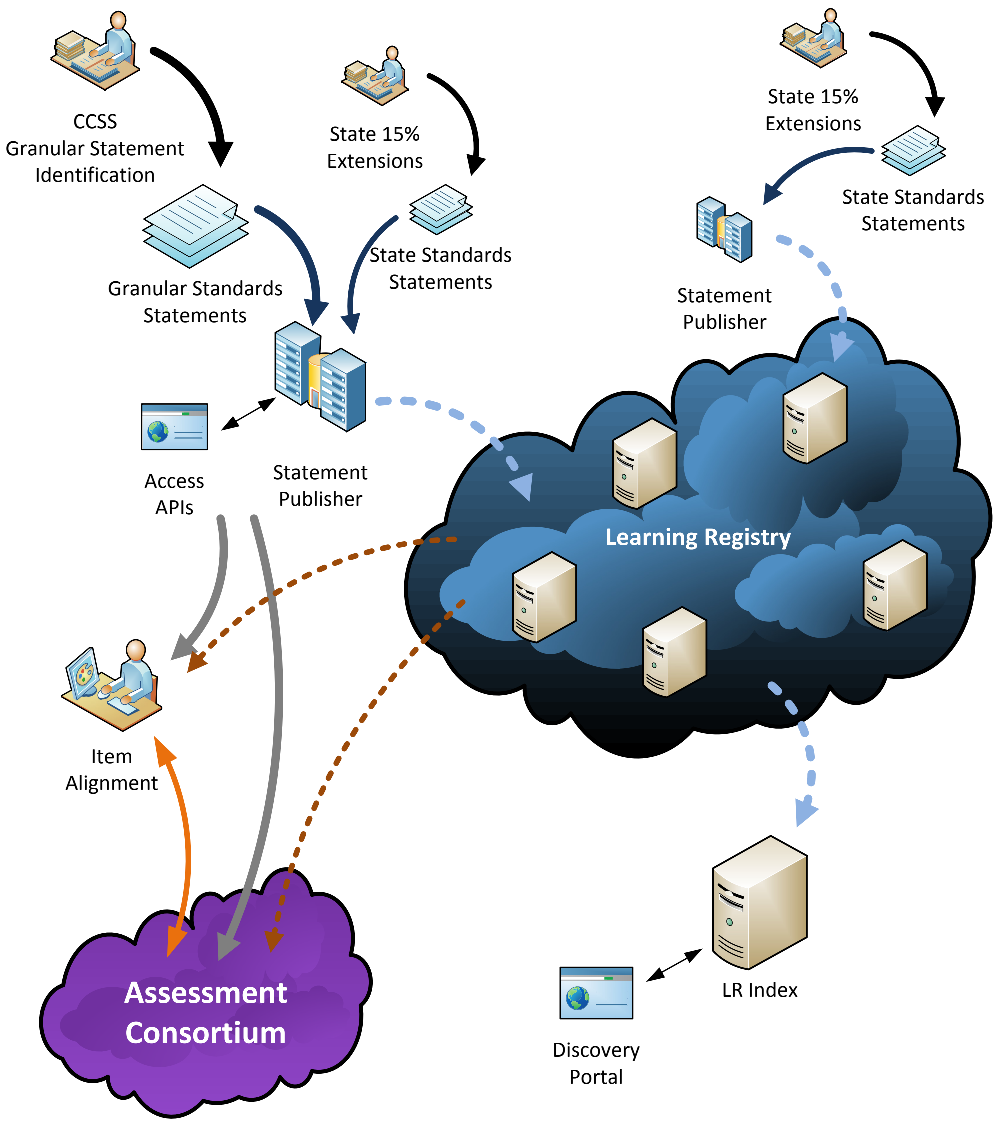


Figure 1: Deployment Model

Various use cases and workflows are supported by this deployment model. Key workflows are illustrated in Figure 2. Numbers in the text # correspond to steps in the diagram.

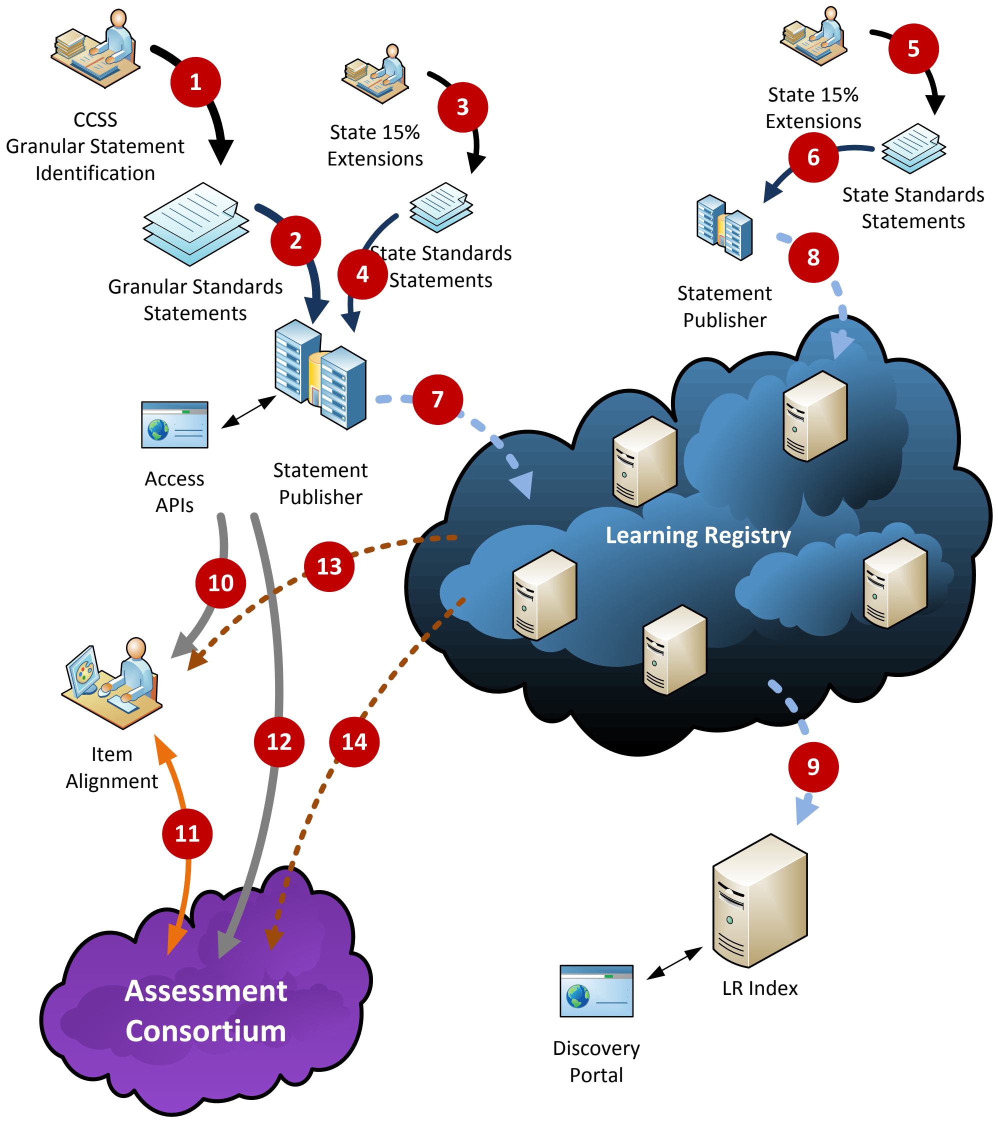


Figure 2: Workflows

Creating granular statements:

- Granular statements and concepts are identified from the source CCSS statements 1.
- Statements (both the description of the standards statement and its metadata) are documented in a digital form in terms of the information model.
- After an approval process (not illustrated), the granular statements are transferred to the publishing agent for publication and distribution 2.

Creating state extensions to CCSS:

- State extensions statements are developed 3.
- After an approval process (not illustrated), the state standards statements are transferred to the publishing agent for publication and distribution 4.

As illustrated in this path, the state could use the same publishing agent as this project. Alternatively, the same workflow could be used with a different publishing agent **5** & **6**. The deployment model permits multiple, independent publishing agents. The choice of publishing agent to use is made by the statement developer.

Other organizations (not illustrated) could use the same workflow to create, identify and publish statements.

Publishing statements:

- Given a set of statements from some source, a publishing agent:
  - Encodes the statements in the digital representation.
  - Augments the metadata with information about the digital encoding.
  - Homes or assigns identifiers used in resolution to retrieve a statement via the associated identifier management system.
  - Publishes the statements to its digital store where they are accessible by others through the APIs and resolvable from the homed or assigned identifiers.
  - Submits records into the Learning Registry, on behalf of the source of the statement, indicating that the new statements have been published **7**.
    - The agent may submit information just about the statements.
    - The agent signs the submission with the key of the statement source, indicating that the statement submission is authentic.
    - In addition to the statements, the agent may take parts of the statements, such as the relations, and publish those as additional paradata assertions about the statements.
- Once published, someone can access the statements directly from the publishing agent though its APIs or distribution mechanism.
- Once submitted to the Learning Registry, someone can discover the statements through the Learning Registry.
- Once resolvable identifiers are defined, an identifier can be used as a link in an external system.

Multiple publishing agents may exist **8**, each differentiated by the services offered (e.g., some publishing agents may be closed and provide services only to known clients, others may be open and publish statements for any client, they may offer different APIs and services).

Updating statements:

- The update process (not illustrated) is identical to the creation and publications steps outlined above. Appropriate metadata, relationships and versioning information are used to tag the standards statements, either when created or when published.

Discovering statements:

- From the Learning Registry, a tool such as the LR Index can monitor the Learning Registry for new instances of standards statements (submitted from publishing agents) **9**.
- The LR Index uses its policies to determine which statements are included in its index (e.g., only CCSS statements and state statements).
- The LR Index provides a discovery portal for users to search for statements matching specific criteria.

Aligning assessment items to statements:

- The item alignment process can retrieve statements (all or some) from the publishing agent through its API **10**.
- Once alignment is complete, the item alignment data can be stored by the assessment consortium **11** with the item linked to the statement.
- Alternatively, the assessment consortium can retrieve statements directly from the publishing agent **12** and pass these to the alignment process **10**:
  - The assessment consortium can hold a copy of the standards statements.
  - The assessment consortium can use the same identifier management system as the publishing agent to provide resolution, but internally, links within the assessment consortium item bank and results could resolve to the locally held copy of the statements.

- Accessing a single publishing agent only provides the statements held by that agent. The assessment consortium could connect directly to the Learning Registry **13** & **14** and find statements from multiple publishing agents that match the consortium's policies.

Aligning resources to statement:
Any 3rd party can access the statements from the publishing agents or from the Learning Registry, align learning resources to statements, and submit the alignments to the Learning Registry through paradata assertions (not illustrated). The alignment workflow is similar to the assessment item workflow described above.

Software Design
The recommended approach of using a statement publishing agent requires a data store and associated services. Options include:

- Using an existing publishing agent that provides the required capabilities.
- Using a bespoke open source publishing platform and data store.

Both of these need to be coupled to an identifier management system.

In both cases, the software must provide:

- A digital store for holding instances of standards statements that utilize all or part of the information model (must support the profile of the information model used by this project).
- A mechanism to publish statements to the digital store.
- A set of APIs to access and retrieve statements.
- A separable, sustainable, identifier management system.

As noted, for integration with other systems, the software should publish to the Learning Registry authoritative records (on behalf of the organization that created the source statement) that the statement has been added to the data store. Other systems that access the Learning Registry can use this information to obtain records of statements from multiple sources.

Analysis: An existing agent may provide the least cost solution, but may require extensions and modifications to support the requirements, plus licensing and service level agreements. Bespoke development generally implies increased cost and time, but will provide the precise features needed, resulting in an open source system.

Recommendation: Use an existing publishing agent if one is available that provides the required capabilities.

The information model supports a shared representation of multiple standards. Three major software design alternatives are possible for the data store:

- The software stack can support multiple standards (multi-tenant storage).
- The software stack can support only one standard (with or without extensions) – a software instance is deployed for each standard, without any identifier management system (virtualized storage, non resolvable identifiers).
- The software stack can support only one standard and uses a shared identifier management system.

Analysis: Storing all standards in the same representation in a single deployed publishing system may enable simpler cross-standard queries and processing at the cost of a slightly more complex representation. Multiple publishing instances with single-standard hosting are simpler but requires multiple deployments and may make cross-standard processing difficult. The issues around a shared identifier management system or multiple identifier management system are the same as outlined above. Establishing multiple deployments, one per standard, is no different than letting multiple parties deploy multi-standard publishing agents.

Recommendation: Support multiple standards per publishing agent. Enable multiple publishing agents.

Support

User and technical support items.

Potential user and technical support includes:

- The formal description/model of the digital representation and associated items.
- Best practices for using the digital representation, including descriptions and examples:
    - Encoding statements into the digital representation.
    - Assigning provenance metadata and curation activities to statements.
    - Assigning identifiers to statements.
    - Publishing statements.
    - Developing vocabularies and encoding them in the digital representation.
    - Versioning of statements.
    - Extensions to statements.
    - Modeling of other standards.
    - Using identifiers in external systems.
- Workflow processes to encode source statements in the digital representation.
- Publishing the digital representation.
- Managing identifiers and the identifier systems.
- Service/API documentation including examples of how to use service APIs and interfaces.

Analysis: The definitive set of support items depends on the selected infrastructure and deployment choices. Development of support items must be done in conjunction with the content working group.

Project Work Item: Develop selected support items in conjunction with content group after the digital representation and infrastructure choices have been finalized.

Sustainment

Sustaining and supporting the digital representation and any operational deployment after project completion.

At overall project completion, the complete CCSS (original statements and granular statements) will exist in digital form. The following **may** need to be sustained and supported over time:

- The standard statements (a document with the statement, the document distribution process, statement maintenance and upgrades, encoding workflow process).
- The open licensed digital representation of the standard, including identifiers (digital representation distribution).
- The description of the digital representation, identifier and metadata solution and supporting materials (technical documents, the document distribution process).
- Open source software (the software distribution package, the software distribution process, software maintenance).
- An operational deployment, possibly including:
    - A digital store.
    - Service interfaces to the digital store.
    - An identifier system.

A sustainment plan is required that addresses:

- What is distributed or deployed.
- The information/document distribution model.
- The operational system deployment model.
- A support and operations plan, including:
    - Who provides support and operations
    - User support and training requirements
    - Capacity plans
    - Service agreements
- The financial/business model for sustainment.
- The management model for deployment, management, maintenance, monitoring, upgrades, community engagement, etc.

Analysis: The definitive set of items to be sustained depends on the selected infrastructure and deployment choices. Sustainment is also dependent on the expected sustainment period. The details of the information model need to be capable of supporting the digital representation over the sustainment period.

Recommendation: To the extent possible, separate sustainment of the statements and their digital representation (edits, versions) from any operational system.

Recommendation: Minimize required new sustainment operations. Rely on or leverage existing ongoing activities.

Project Work Item: Develop sustainment plan.

Glossary

Association Data: *Association data* presents the association between the *name* and the *thing* identified in an *identifier.* [PILIN]

Authority: An *authority* for a component is a party responsible for maintaining the component. [PILIN]

Context: A *context* differentiates *labels* used for distinct purposes and with different authorities. The combination of a *label* and a *context* for the label gives a *name*, and the same label can be used in different contexts to give different names; any label is necessarily *unique* in its context. Contexts impose *policies.* Contexts may be identified by one or more *context identifiers*. The identifier for a context has a name in a context of context identifiers. There is a defined context instance of "known naming systems", preventing infinite recursion of contexts. [PILIN]

Data Store: A *data store* is a tool for the storage and management of data. The data store exposes services allowing access to that data by other parties. [PILIN]

Digital Representation: The digital form of a *standards statement*, including both provenance metadata and the description of all other characteristics of the statement.

Granular Identifier: This concept does not exist within the technical approach. All identifiers are equal; none is more or less granular than any other.

Granular Statement: A *standards statement* derived from the decomposition or mapping of an existing standards statement to identify existing concepts within the statement for the purpose of aligning learning resources and assessment items to part of a statement.

Identifier: An identifier is the association of a *name* with a *thing*. A name may only be associated with one thing at any time, and the name is said to *identify* the thing. [PILIN]

Identifier Management System: An *identifier management system* is a collection of definitions, *information models*, *policies*, and *data stores*, used to manage *identifiers*. An identifier management system has a defined operating scope, and an *authority* acting as its owner. [PILIN]

Identify: A *thing* is *identified* by a *name* if the name and the thing together form an *identifier*. The name is associated with the thing identified, although that is not the only form of association possible in an *identifier management system.* [PILIN]

Information Model: An *information model* is a model of *things* in a domain, their properties, and the relations between them. The choice of what things to *identify* in an *identifier management system* is informed by an information model. [PILIN]

Label: A *label* is a symbol that can potentially be used as a *name*. In an *identifier management system*, labels are typically strings. [PILIN]

Locator: A locator is a string giving the location of a digital object in a *data store*, and can be used as a retrieval key to gain access to the object. A URL is an example of a locator, although not all http: URIs are locators. A locator is specific to a *data store,* and cannot be used to access an instance of the digital object in a different *data store*. A locator can be used as an *identifier*; but it will usually not be *persistent*. Persistent identifiers often *resolve* to the current locator(s) of the *thing identified*. This uncouples the persistent identification of a resource from the current retrieval key for the resource. [PILIN]

Name: A *name* is the association of a *label* with a *context*. [PILIN]

Metadata: Descriptive cataloging and provenance information about a *standards statement*. The type of information is not unique to a standards statement.

Opaque: An *identifier* is *opaque* if there is a direct relationship between the *name* and a relevant fact about the *thing identified*, but the relationship cannot be inferred by inspection. [PILIN]

Persistent: A component is *persistent* if it is managed and maintained for a defined timespan. Normally when an identifier is called persistent, persistence of association is meant. [PILIN]

Publish: The process of making the digital representation of a standards statement available on the Internet for others to use.

Policy: A *policy* is a set of rules regarding components. [PILIN]

Provenance: *Provenance* is the history of how a thing has been managed over time. Data documenting the provenance of a digital object are part of the metadata for that object. [PILIN]

Provenance Metadata: See *Metadata*. Provenance metadata is used in the project documents to identify a subset of information about a *standards statement*, since its entire description could be considered metadata.

Resolve: An identifier is *resolved* by providing information on how to access the *thing* it identifies. This information is the *resolution* of the identifier. An identifier is Internet-Resolvable if the information on how to access the thing identified can be requested and consumed through a well-defined Internet application protocol, and Web-Resolvable if that protocol is a defined web application layer protocol. Resolution in general operates on *association data,* stored, managed and maintained with the identifier in an *identifier management system*. [PILIN]

Source Statement: An original *standards statement*, in natural language form.

Standards Statement: The narrative statement for a single educational standard that defines "what students should understand and be able to do." Within CCSS, a standards statement or the components of a standards statement are the most granular statements.

Statement: See *Standards Statement*.

Thing: A *thing* is what anything that can be talked about is; in particular, it includes whatever can be *identified* with an *identifier*. [PILIN]

Transparent: An *identifier* is semantically *transparent* if there is a direct relationship between the *name* and a relevant fact about the *thing identified*, and the relationship can be inferred by inspection. [PILIN]

Unique: A component is *unique* if there exists one and only one instance of the component within a given scope. [PILIN]

Note: Most terms come from the PILIN Glossary, as noted. Some PILIN definitions have been simplified. Words in italics in a gloss are defined elsewhere in the glossary.

References

[ACARA] *Australian Curriculum RDF Specification*, Report prepared by Education Services Australia Ltd Version 3.0, 2011–06–27

[ACARA Tech] *ACARA Technical Information*
[http://www.australiancurriculum.edu.au/Technical/Introduction]

[ASN] *ASN Application Profile*
[http://standards.jesandco.org/wiki/ASN_Application_Profile]

[CEDS 2.0] *Common Educational Data, Standards Domain Entity Schema*, V 2.0
[https://ceds.ed.gov/domainEntitySchema.aspx]

[CEDS 3.0] *Common Educational Data Standards, CEDS Elements*, Version 3 (Draft)
[https://ceds.ed.gov/elements.aspx?v=3&ex=%28Draft%29]

[CCSSO] *Common Core State Standards Official Identifiers and XML Representation*
[http://www.corestandards.org/common-core-state-standards-official-identifiers-and-xml-representation]

[CCSSO ELA] *English Language Arts Standards*, Common Core State Standards Initiative
[http://www.corestandards.org/ELA-Literacy.xml]

[CCSSO Math] *Standards for Mathematical Practice*, Common Core State Standards Initiative
[http://www.corestandards.org/Math.xml]

[DC] *DCMI Specifications*, Dublin Core Metadata Initiative
[http://dublincore.org/specifications/]

[Ed-Fi] *ed-fi Core XML Schema Reference Documentation*
[http://www.ed-fi.org/technical-documentation/]

[FRBR] *Functional Requirements for Bibliographic Records*
[http://www.ifla.org/publications/functional-requirements-for-bibliographic-records]

[JSON] *The application/json Media Type for JavaScript Object Notation (JSON),* IETF RFC 4627, July 2006.
[http://tools.ietf.org/html/rfc4627]

[JSON LD] *JSON-LD Syntax 1.0*
[http://json-ld.org/spec/latest/json-ld-syntax/]

[LRMI] *Learning Resource Metadata Initiative, The Specification*
[http://www.lrmi.net/the-specification]

[Learning Registry Paradata] *Learning Registry Paradata Specification*
[https://docs.google.com/document/d/1IrOYXd3S0FUwNozaEG5tM7Ki4_AZPrBn-pbyVUz-Bh0/edit]

[PILIN] *PILIN Ontology for Identifiers and Identifier Services*
[http://resolver.net.au/hdl/102.100.272/ G9JR4TLQH]

[PILIN Glossary] *PILIN Glossary*
[http://resolver.net.au/hdl/102.100.272/HHYMV8JQH]

[schema.org] *Schema.org*
[http://schema.org/]

[SKOS] *SKOS Simple Knowledge Organization System, Reference*
[http://www.w3.org/TR/skos-reference/]

[SIF] *Learning Standards: SIF/CEDS Schema Profile*, May 2012

[SLC] *SLC Data Store Logical Model,* Shared Learning Collaborative
[http://dev.slcedu.org/docs/sli-data-store-logical-model]

[SLC API] *Application Development Using the SLC REST API*, Shared Learning Collaborative
[http://dev.slcedu.org/docs/application-development-using-sli-rest-api]

[ZTHES] *The Zthes Specifications for Thesaurus Representation, Access and Navigation*
[http://zthes.z3950.org/]

GIM CCSS Project Documentation

Scope, Technical Requirements, Approaches, and Recommendations [this document]

Learning Standards Digital Representation Specification: JSON Serialization Schemata

Learning Standards Digital Representation Specification: XML Serialization Schemata

Learning Standards Digital Representation Specification: RESTful API

Learning Standards Digital Representation Specification: Publishing Agent Design and Requirements

Change Log

Date	Version	Author	Notes
20121119	0.2	DR	Initial limited release for baseline requirements discussion with management team.
20121127	0.3	DR	Partial unedited draft for management team.
20121129	0.5	DR	Complete draft, unedited. Reorganized sections. Restructured to focus on alternatives and recommendations. Release to management group.
20121203	0.6	DR	Complete draft, edited. Release to working group for review.
20130102	0.7	DR	Release incorporating feedback for review. Update terminology. Added use cases. Add vocabulary descriptions. Add deployment system design.
20130110	0.75	DR	Added Executive Summary.
20130206	0.8	DR	Minor editorial changes. Draft for Public review.
20130206	0.9	DR	Minor editorial updates. Internal working version, not for release.
20130912	1.0	DR	V1.0 Public release