



Linked Data Platform Use Cases and Requirements

W3C Working Draft 31 January 2013

This version:

<http://www.w3.org/TR/2013/WD-ldp-ucr-20130131/>

Latest published version:

<http://www.w3.org/TR/ldp-ucr/>

Latest editor's draft:

<http://www.w3.org/2012/ldp/hg/ldp-ucr.html>

Previous version:

Editors:

[Steve Battle](#), [Sysemia Limited](#)

[Steve Speicher](#), [IBM Corporation](#)

[Copyright](#) © 2013 [W3C](#)® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Código de campo cambiado

Código de campo cambiado

Código de campo cambiado

Código de campo cambiado

Código de campo cambiado

Abstract

A set of user stories, use cases, scenarios and requirements that motivate a simple read-write Linked Data architecture, based on HTTP access to web resources that describe their state using RDF.

Comentario [M.E.G.1]: Cold start. See (1) in the review report

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <http://www.w3.org/TR/>.

This document was published by the [Linked Data Platform Working Group](#) as a [First Public Working Draft](#). If you wish to make comments regarding this document, please send them to public-ldp@w3.org ([subscribe](#), [archives](#)). All comments are welcome.

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). This document is informative only. W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

- 1. [Scope and Motivation](#)
- 2. [Organization of this Document](#)
- 3. [User Stories](#)
 - [3.1 Maintaining Social Contact Information](#)
 - [3.2 Keeping Track of Personal and Business Relationships](#)
 - [3.3 System and Software Development Tool Integration](#)
 - [3.4 Library Linked Data](#)
 - [3.5 Municipality Operational Monitoring](#)
 - [3.6 Healthcare](#)
 - [3.7 Metadata Enrichment in Broadcasting](#)
 - [3.8 Aggregation and Mashups of Infrastructure Data](#)
 - [3.9 Sharing payload of RDF data among low-end devices](#)
 - [3.10 Sharing Binary Resources and Metadata](#)
 - [3.11 Data Catalogs](#)
 - [3.12 Constrained Devices and Networks](#)
 - [3.13 Services Supporting the Process of Science](#)
 - [3.14 Project Membership Information: Information Evolution](#)
 - [3.15 Cloud Infrastructure Management](#)
- 4. [Use Cases](#)
 - [4.1 Use Case: Manage containers](#)
 - [4.1.1 Primary scenario: create container](#)
 - [4.1.2 Alternative scenario: create a nested container](#)
 - [4.2 Use Case: Manage resources](#)
 - [4.2.1 Primary scenario: create resource](#)
 - [4.2.2 Alternative scenario: delete resource](#)
 - [4.2.3 Alternative scenario: moving contained resources](#)
 - [4.3 Use Case: Retrieve resource description](#)
 - [4.3.1 Primary scenario](#)
 - [4.3.2 Alternative scenario: retrieve description of a non-document resource](#)
 - [4.4 Use Case: Update existing resource](#)
 - [4.4.1 Primary scenario: enrichment](#)
 - [4.4.2 Alternative scenario: selective update of a resource](#)
 - [4.5 Use Case: Determine if a resource has changed](#)
 - [4.5.1 Primary scenario](#)
 - [4.6 Use Case: Aggregate resources](#)
 - [4.6.1 Primary scenario: add a resource to a collection](#)
 - [4.6.2 Alternative scenario: add a resource to multiple collections](#)
 - [4.7 Use Case: Filter resource description](#)

- 4.7.1 Primary scenario: retrieve collection-level description
 - 4.7.2 Alternative scenario: retrieve item-level description of a collection
 - 4.8 Use Case: Manage media resources
 - 4.8.1 Primary scenario: access media resources
 - 4.8.2 Alternative scenario: media-resource attachments
- 5. Requirements
 - 5.1 Functional Requirements
 - 5.2 Non-Functional Requirements
- A. Acknowledgements
- B. Change History
- C. References
 - C.1 Informative references

1. Scope and Motivation

Linked Data was defined by Tim Berners-Lee with the following guidelines [LINKED-DATA]:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs, so that they can discover more things

These four rules have proven very effective in guiding and inspiring people to publish Linked Data on the web. The amount of data, especially public data, available on the web has grown rapidly, and an impressive number of extremely creative and useful “mashups” have been created using this data as result.

There has been much less focus on the potential of Linked Data as a model for managing data on the web - the majority of the Application Programming Interfaces (APIs) available on the Internet for creating and updating data follow a Remote Procedure Call (RPC) model rather than a Linked Data model.

If Linked Data were just another model for doing something that RPC models can already do, it would be of only marginal interest. Interest in Linked Data arises from the fact that applications with an interface defined using Linked Data can be much more easily and seamlessly integrated with each other than applications that offer an RPC interface. In many problem domains, the most important problems and the greatest value are found not in the implementation of new applications, but in the successful integration of multiple applications into larger systems.

Some of the features that make Linked Data exceptionally well suited for integration include:

- A single interface – defined by a common set of HTTP methods – that is universally understood and is constant across all applications. This is in

contrast with the RPC architecture where each application has a unique interface that has to be learned and coded to.

- A universal addressing scheme – provided by HTTP URLs – for both identifying and accessing all “entities”. This is in contrast with the RPC architecture where there is no uniform way to either identify or access data.
- A simple yet extensible data model – provided by RDF – for describing data about a resource in a way which doesn’t require prior knowledge of vocabulary being used.

Comentario [M.E.G.2]: Oranges to apples comparison. See (2) in the review report.

Experience implementing applications and integrating them using Linked Data has shown very promising results, but has also demonstrated that the original four rules defined by Tim Berners-Lee for Linked Data are not sufficient to guide and constrain a writable Linked Data API. As was the case with the original four rules, the need generally is not for the invention of fundamental new technologies, but rather for a series of additional rules and patterns that guide and constrain the use of existing technologies in the construction of a [LINKED-DATA-PLATFORM] to achieve interoperability.

The following list illustrates a few of the issues that require additional rules and patterns:

- What URLs do I post to in order to create new resources?
- How do I get lists of existing resources, and how do I get basic information about them without having to access each one?
- How should I detect and deal with race conditions on write?
- What media-types/representations should I use?
- What standard vocabularies should I use?
- What primitive data types should I use?

Comentario [M.E.G.3]: Right message, but unfair critique. See (3) in the review report.

~~A good~~The goal for the [LINKED-DATA-PLATFORM] ~~would be~~is to define a specification required to allow the definition of a writable Linked Data API equivalent to the simple application APIs that are often written on the web today using the Atom Publishing Protocol (APP). ~~APP, which~~ shares some characteristics with Linked Data, ~~such as the use of HTTP and URLs, but relying on~~ ~~One difference is that Linked Data relies on~~ a flexible data model ~~with based on~~ RDF, ~~which that~~ allows for multiple representations.

2. Organization of this Document

Use-cases are captured in a narrative style that describes a behavior, or set of behaviors drawn from user-stories. They are embellished with concrete examples drawn from representative user-stories. The aim throughout has been to avoid details of protocol (specifically the HTTP protocol), and use of any specific vocabulary that might be introduced by the LDP specification.

Comentario [M.E.G.4]: A rewrite of this paragraph could be included as part of the abstract.

This document is organized as follows:

- **User Stories** capture statements about system requirements written from a user or application perspective. They are typically lightweight and

informal and can run from one line to a paragraph or two (sometimes described as an 'epic') [2]. Analysis of each user story will reveal a number of (functional) use-cases and other non-functional requirements.

- See *Device API Access Control Use Cases and Requirements [DAP-REQS]* for a good example of user stories and their analysis.

Con formato: Sangría: Izquierda: 1,25 cm, Sin viñetas ni numeración

Con formato: Fuente: Cursiva

- **Use Cases** are used to capture and model functional requirements. Use cases describe the system's behavior under various conditions [3], cataloging who does what with the system, for what purpose, but without concern for system design or implementation [4].

Use cases act like the hub of a wheel, with spokes supporting requirements analysis, scenario-based evaluation, testing, and integration with non-functional, or quality requirements.

Con formato: Sangría: Izquierda: 1,25 cm, Sin viñetas ni numeración

Each use case is identified by a reference number to aid cross-reference from other documentation; use-case indexing in this document is based on rdb2rdf use_cases [RDB2RDF-UC].

- A variety of styles may be used to capture use_cases, from a simple narrative to a structured description with actors, pre/post conditions, **and** step-by-step behaviors (as in *POWDER: Use Cases and Requirements [POWDER-USE-CASES]*), and non-functional requirements raised by the use_case. ~~Use cases act like the hub of a wheel, with spokes supporting requirements analysis, scenario-based evaluation, testing, and integration with non-functional, or quality requirements.~~

Con formato: Sangría: Izquierda: 1,25 cm, Sin viñetas ni numeración

Con formato: Fuente: Cursiva

- **Scenarios** ~~are more focused still, representing a single instance of a use case in action~~ are used to model functional requirements of a use case, and focus on a use case in action. Scenarios may range from lightweight narratives as seen in *Use cases and requirements for Media Fragments [MEDIA-FRAGMENTS-REQS]*, to being formally modeled as interaction diagrams. Each use_case ~~should~~ includes at least a primary scenario, and possibly other alternative scenarios.

Con formato: Fuente: Cursiva

- **Requirements** lists functional and non-functional or quality requirements, and the use cases they may be derived from. This approach is exemplified in the *Use Cases and Requirements for the Data Catalog Vocabulary [DCAT-UCR]*. ~~It also lists functional requirements that stem from use-cases. It is also possible at this stage to explicitly identify some use-cases as non-requirements.~~

Con formato: Fuente: Cursiva

3. User Stories

3.1 Maintaining Social Contact Information

Comentario [M.E.G.5]: Very informal language. 1st person used.

Many of us have multiple email accounts that include information about the people and organizations we interact with – names, email addresses, telephone numbers, instant messenger identities and so on. When someone’s email address or telephone number changes (or they acquire a new one), our lives would be much simpler if we could update that information in one spot and all copies of it would automatically be updated. In other words, those copies would all be linked to some definition of “the contact.” There might also be good reasons (like off-line email addressing) to maintain a local copy of the contact, but ideally any copies would still be linked to some central “master.”

Agreeing on a format for “the contact” is not enough, however. Even if all our email providers agreed on the format of a contact, we would still need to use each provider’s custom interface to update or replace the provider’s copy, or we would have to agree on a way for each email provider to link to the “master”. If we look outside our own personal interests, it would be even more useful if the person or organization exposed their own contact information so we could link to it.

What would work in either case is a common understanding of the resource, a few formats needed, and access guidance for these resources. This would support how to acquire a link to a contact, ~~and~~ how to use those links to interact with a contact (including [reading](#), [updating](#), and [deleting](#) it), as well as how to easily [create a new contact](#), add it to my contacts, and when deleting a contact, how it would be removed from my list of contacts. It would also be good to be able to add some application-specific data about my contacts that the original design didn’t consider. Ideally we’d like to eliminate multiple copies of contacts, there would be additional valuable information about my contacts that may be stored on separate servers and need a simple way to link this information back to the contacts. Regardless of whether a contact collection is my own, shared by an organization, or all contacts known to an email provider (or to a single email account at an email provider), it would be nice if they all worked pretty much the same way.

3.2 Keeping Track of Personal and Business Relationships

In our daily lives, we deal with many different organizations in many different relationships, and they each have data about us. However, it is unlikely that any one organization has all the information about us. Each of them typically gives us access to the information (at least some of it), many through websites where we are uniquely identified by some string – an account number, user ID, and so on. ~~We have to use their applications to interact with the data about us, however, and we have to use their identifier(s) for us.~~ If we want to build any semblance of a holistic picture of ourselves (more accurately, collect all the data about us that they externalize), we as humans must use their custom applications [together with the identifiers they have for us](#) to find their data [about us](#), copy it, and organize it to suit our needs.

Would it not be simpler if at least the Web-addressable portion of that data could be linked to consistently, so that instead of maintaining various identifiers in different formats and instead of having to manually supply those identifiers to

Comentario [M.E.G.6]: Mix of issues. Separate. See (9) in the review report..

Comentario [M.E.G.7]: Very informal language. 1st person used.

each one's corresponding custom application, we could essentially build a set of bookmarks to it all? When we want to [examine](#) or [change](#) their contents, would it not be simpler if there were a single consistent application interface that they all supported? Of course it would.

Our set of links would probably be a [simple collection](#). The information held by any single organization might be a mix of simple data and [collections of other data](#), for example, a bank account balance and a collection of historical transactions. Our bank might easily have [a collection of accounts for each member of its collection of customers](#).

Comentario [M.E.G.8]: Not clear. See (10) in the review report.

3.3 System and Software Development Tool Integration

Comentario [M.E.G.9]: Formal language. Impersonal.

System and software development tools typically come from a diverse set of vendors and are built on various architectures and technologies. These tools are purpose built to meet the needs for a specific domain scenario (modeling, design, requirements and so on.) Often tool vendors view integrations with other tools as a necessary evil rather than providing additional value to their end-users. Even more of an afterthought is how these tools' data -- such as people, projects, customer-reported problems and needs -- integrate and relate to corporate and external applications that manage data such as customers, business priorities and market trends. The problem can be isolated by standardizing on a small set of tools or a set of tools from a single vendor, but this rarely occurs and if does it usually does so only within small organizations. As these organizations grow both in size and complexity, they have needs to work with outsourced development and diverse internal other organizations with their own set of tools and processes. There is a need for better support of more complete business processes (system and software development processes) that span the roles, tasks, and data addressed by multiple tools. This demand has existed for many years, and the tools vendor industry has tried several different architectural approaches to address the problem. Here are a few:

- Implement an API for each application, and then, in each application, implement "glue code" that exploits the APIs of other applications to link them together.
- Design a single database to store the data of multiple applications, and implement each of the applications against this database. In the software development tools business, these databases are often called "repositories."
- Implement a central "hub" or "bus" that orchestrates the broader business process by exploiting the APIs described previously.

It is fair to say that although each of those approaches has its adherents and can point to some successes, none of them is wholly satisfactory. The use of [Linked Data](#) as an application integration technology has a strong appeal [OSLC](#).

Comentario [M.E.G.10]: How are LD and EAI related. Introduce OSLC a little bit. See (11) in the review report.

Comentario [M.E.G.11]: Add appropriate informativ reference.

3.4 Library Linked Data

Comentario [M.E.G.12]: Formal language. Impersonal.

The *W3C Library Linked Data ~~working~~ Working group Group* has a number of use cases cited in their *Use Case Report [LLD-UC]*. These referenced use cases focus on the need to extract and correlate library data from disparate sources. Variants of these use cases that can provide consistent formats, as well as ways to improve or update the data, would enable simplified methods for both efficiently sharing this data as well as producing incremental updates without the need for repeated full extractions and import of data.

Con formato: Fuente: Cursiva

Con formato: Fuente: Cursiva

The 'Digital Objects Cluster' contains a number of relevant use-cases:

- Grouping: This should "Allow the end-users to define [groups of resources](#) on the web that for some reason belong together. The relationship that exists between the resources is often left unspecified. Some of the resources in a group may not be under control of the institution that defines the groups."
- Enrichment: "Enable end-users to [link resources together](#)."
- Browsing: "[Support end-user browsing through groups](#) and resources that belong to the groups."
- Re-use: "Users should have the capability to re-use all or parts of a collection, with all or part of its metadata, elsewhere on the linked Web."

The 'Collections' cluster also contains a number of relevant use-cases:

- Collection-level description: "Provide [metadata pertaining to a collection as a whole](#), in contrast to item-level description."
- Collections discovery: "Enable innovative collection discovery such as identification of nearest location of a physical collection where a specific information resource is found or mobile device applications ... based on collection-level descriptions."
- Community information services: Identify and classify collections of special interest to the community.

3.5 Municipality Operational Monitoring

Comentario [M.E.G.13]: Formal language. Impersonal.

Across various cities, towns, counties, and various municipalities there is a growing number of services managed and run by municipalities that produce and consume a vast amount of information. This information is used to help monitor services, predict problems, and handle logistics. In order to effectively and efficiently collect, produce, and analyze all this data, a fundamental set of loosely coupled standard data sources are needed. A simple, low-cost way to [expose data](#) from the diverse set of monitored services is needed, one that can easily integrate into the municipalities of other systems that inspect and analyze the data. All these services have links and dependencies on other data and services, so having a simple and scalable linking model is key.

Código de campo cambiado

3.6 Healthcare

Comentario [M.E.G.14]: Formal language. Impersonal.

For physicians to analyze, diagnose, and propose treatment for patients requires a vast amount of complex, changing and growing knowledge. This knowledge needs to come from a number of sources, including physicians' own subject knowledge, consultation with their network of other healthcare professionals, public health sources, food and drug regulators, and other repositories of medical research and recommendations.

To diagnose a patient's condition requires current data on the patient's medications and medical history. In addition, recent pharmaceutical advisories about these medications are linked into the patient's data. If the patient experiences adverse ~~affects~~ effects from medications, these physicians need to publish information about this to an appropriate regulatory source. Other medical professionals require access to both validated and emerging effects of the medication. Similarly, if there are geographical patterns around outbreaks that allow both the awareness of new symptoms and treatments, this information needs to quickly reach a very distributed and diverse set of medical information systems. Also, reporting back to these regulatory agencies regarding new occurrences of an outbreak, including additional details of symptoms and causes, is critical in producing the most effective treatment for future incidents.

3.7 Metadata Enrichment in Broadcasting

Comentario [M.E.G.15]: Formal language. Impersonal.

There are many different use cases when broadcasters show interest in metadata enrichment:

- enrich archive or news metadata by linking facts, events, locations and personalities
- enrich metadata generated by automatic extraction tools such as person identification, etc.
- enrich definitions of terms in classification schemes or enumeration lists

This comes in support of more effective information management and data/content mining (if you can't find your content, it's like you don't have it and must either recreate or acquire it again, which is not financially effective).

However, there is a need for solutions facilitating linkage to other data sources and taking care of the issues such as discovery, automation, disambiguation, etc. Other important issues that broadcasters would face are the editorial quality of the linked data, its persistence, and usage rights.

3.8 Aggregation and Mashups of Infrastructure Data

Comentario [M.E.G.16]: Formal language. Impersonal.

For infrastructure management (such as storage systems, virtual machine environments, and similar IaaS and PaaS concepts), it is important to provide an environment in which information from different sources can be aggregated,

[filtered](#), and visualized effectively. Specifically, the following use cases need to be taken into account:

- While some data sources are based on Linked Data, others are not, and aggregation and mashups must work across these different sources.
- Consumers of the data sources and aggregated/filtered data streams are not necessarily implementing Linked Data themselves, they may be off-the-shelf components such as dashboard frameworks for composing visualizations.
- Simple versions of this scenario are pull-based, where the data is requested from data sources. In more advanced settings, without a major change in architecture it should be possible to move to a push-based interaction model, where data sources push notifications to subscribers, and data sources provide different services that consumers can subscribe to (such as "informational messages" or "critical alerts only").

In this scenario, the important factors are to have abstractions that allow easy aggregation and filtering, are independent from the internal data model of the sources that are being combined, and can be used for pull-based interactions as well as for push-based interactions.

3.9 Sharing payload ~~Payload of RDF data~~ ~~Data among~~ ~~Among low~~ ~~Low-end~~ ~~End devices~~ ~~Devices~~

Several projects around the idea of [downscaling the Semantic Web](#) need to be able to ship payloads of RDF across the nodes member of a given network. The transfers are done in a constrained context in terms of bandwidth, scope of the local semantics employed by the nodes and computing capabilities of the nodes. In a P2P style, every node has the capability to act either as a data consumer or a data provider, serving its own data or acting as a relay to pass other's data along (typically in mesh networks).

The transfer of an arbitrary payload of RDF data could be implemented through the container mechanism, adding and removing sets of RDF triples to it. Currently, the ["SemanticXO"](#) project uses named graphs and the graph store protocol to create/delete/copy graphs across the nodes but this (almost) imposes the usage of a triple store. Unfortunately, triple stores are rather demanding pieces of software that are not always usable on limited hardware. Some generic REST-like interaction backed up with a lightweight column store would be [a](#) better approach.

3.10 Sharing Binary Resources and Metadata

When publishing datasets about stars one may want to publish links to the pictures in which those stars appear, and this may well require publishing the pictures themselves. Vice versa: when publishing a picture of space we need to know which telescope took the picture, which part of the sky it was pointing at, what filters were used, which identified stars are visible, who can read it, who can write to it, ...

Comentario [M.E.G.17]: Formal language. Impersonal.

Comentario [M.E.G.18]: Expectations not clear. See (12) in the review report.

Comentario [M.E.G.19]: Add appropriate informative reference.

Comentario [M.E.G.20]: Add appropriate informative reference.

Comentario [M.E.G.21]: Informal language. Mixed 1st person and impersonal.

If Linked Data contains information about resources that are most naturally expressed in non-RDF formats (be they binary such as pictures or videos, or human readable documents in XML formats), those non-RDF formats should be just as easy to publish to the Linked Data server as the RDF relations that link those resources up. A Linked Data server should therefore allow publishing of ~~non~~-non-linked data resources too, and make it easy to publish and edit metadata about those resources.

Comentario [M.E.G.22]: Why? See (13) in the review report.

The resource comes in two parts - the image and information about the image (which may be in the image file but is better kept external to it as it's more general). The information about the image is vital. It's a compound item of image data and other data (application metadata about the image) ~~does that~~ are not distinguished from the platform's point-of-view.

Comentario [M.E.G.23]: Not clear. See (14) in the review report.

3.11 Data Catalogs

The *Asset Description Metadata Schema* (ADMS) provides the data model to describe semantic asset repository contents, but this leaves many open challenges when building a federation of these repositories to serve the need of asset reuse. These include accessing and querying individual repositories and efficiently retrieving [updated content](#) without having to retrieve the whole content. Hence, we chose to build the integration solution capitalizing on the Data Warehousing integration approach. This allows us to cope with heterogeneity of sources technologies and to benefit from the optimized performance it offers, given that individual repositories do not usually change frequently. With Data Warehousing, the federation requires one to:

Comentario [M.E.G.24]: Formal language. Mixes 1st person with impersonal.

Comentario [M.E.G.25]: Add appropriate informative reference.

Con formato: Fuente: Cursiva

- understand the data, i.e. understand their semantic descriptions, and other systems.
- seamlessly exchange the semantic assets metadata from different repositories
- keep itself up-to-date.

Comentario [M.E.G.26]: We? Who? Is this talking about a particular solution out there? See (15) in the review report.

Repository owners can maintain de-referenceable URIs for their repository descriptions and contained assets in a Linked Data compatible manner. ADMS provides the necessary data model to enable meaningful exchange of data. However, this leaves the challenge of efficient access to the data not fully addressed.

Related: [Data Catalog Schema and Protocol](#)

Comentario [M.E.G.27]: Contextualize reference verbosely. See (16) in the review report.

3.12 Constrained Devices and Networks

Information coming from resource constrained devices in the Web of Things (WoT) has been identified as a major driver in many domains, from smart cities to environmental monitoring to real-time tracking. The amount of information produced by these devices is growing exponentially and needs to be accessed and integrated in a systematic, standardized and cost efficient way. By using the same standards as on the Web, integration with applications will be simplified and higher-level interactions among resource constrained devices,

Comentario [M.E.G.28]: Formal language. Impersonal.

Comentario [M.E.G.29]: No requirements beyond the LDP protocol itself.

Comentario [M.E.G.30]: Add appropriate informative reference.

abstracting away heterogeneities, will become possible. Up-coming IoT/WoT standards such as [6LowPAN](#) -IPv6 for resource constrained devices- and the [Constrained Application Protocol \(CoAP\)](#), which provides a downscaled version of HTTP on top of UDP for the use on constrained devices, are already at a mature stage. The next step now is to support RESTful interfaces also on resource constrained devices, adhering to the Linked Data principles. Due to the limited resources available, both on the device and in the network (such as bandwidth, energy, [and](#) memory) a solution based on [SPARQL Update](#) is at the current point in time considered not to be useful and/or feasible. An approach based on the [HTTP-CoAP Mapping](#) would enable constrained devices to directly participate in a Linked Data-based environment.

Comentario [M.E.G.31]: Add appropriate informative reference.

Comentario [M.E.G.32]: Add appropriate informative reference.

Con formato: Fuente: Cursiva

Comentario [M.E.G.33]: Add appropriate informative reference.

Con formato: Fuente: Cursiva

Comentario [M.E.G.34]: Add appropriate informative reference.

3.13 Services Supporting the Process of Science

Comentario [M.E.G.35]: Informal language. 1st person used.

Comentario [M.E.G.36]: No requirements beyond the LDP protocol itself.

Many fields of science now include branches with in silico data-intensive methods, e.g. bioinformatics, astronomy. To support these new methods we look to move beyond the established platforms provided by scientific workflow systems to capture, assist, and preserve the complete lifecycle from record of the experiment, through local trusted sharing, analysis, dissemination (including publishing of experimental data "beyond the PDF"), and re-use.

- [Aggregations](#), specifically *Research Objects (ROs)* that are exchanged between services and clients bringing together workflows, data sets, annotations, and provenance. We use an RDF model for this. While some aggregated contents are encoded using RDF and an increasing number are linked data sources, others are not; while some are stored locally "within" the RO, others are remote (in both cases this is often due to size of the resources or access policies).
- Services that are distributed and linked. Some may be centralising for e.g. publication, others may be local, e.g. per lab. We need lightweight services that can be simply and easily integrated into and scale across the wide variety of softwares and data used in science: we have adopted a RESTful approach where possible.
 - Foundation services that collect and expose ROs for storage, modification, exploration, and reuse.
 - Services that provide added-value to ROs such as seamless import/export from scientific workflow systems, automated stability evaluation, or recommendation (and therefore interact with the foundation services to retrieve/store/modify/ROs).

Comentario [M.E.G.37]: Reword. See (20) in the review report.

seeAlso: [Wf4Ever](#)

Comentario [M.E.G.38]: Contextualize reference verbosely. See (21) in the review report.

3.14 Project Membership Information: Information Evolution

Comentario [M.E.G.39]: Formal language. Impersonal.

Information about people and projects changes as roles change, as organisations change and as contact details change. Finding the current state of a project is important in enabling people to contact the right person in the right role. It can also be useful to look back and see who was performing what role in the past.

A use of a Linked Data Platform could be to give responsibility for managing such information to the project team itself, instead of requiring updates to be requested from a centralised website administrator.

This could be achieved with:

- Resource descriptions for each person and project
- A container resource to describe roles/membership in the project.

To retain the history of the project, the old version of a resources, including container resources, should be retained so there is a need to address both specific items and also have a notion of "current".

Access to information has two aspects:

- Access to the "current" state, regardless of the version of the resource description
- Access to historical state, via access to a specific version of the resource description

3.15 Cloud Infrastructure Management

Comentario [M.E.G.40]: Formal language. Impersonal

Cloud operators offer API support to provide customers with remote access for the management of Cloud infrastructure (IaaS). Infrastructure consists of Systems, Computers, Networks, Discs, etc. The overall structure can be seen as mostly hierarchical, (Cloud contains Systems, Systems contain Machines, etc), complemented with crossing links (e.g. multiple Machines connected to a Network).

The IaaS scenario makes specific requirements on lifecycle management and discovery, handling non-instant changes, history capture and query:

- Many aspects of Cloud infrastructure have associated lifecycle, e.g. a Computer can be transitioned between Running and Shutdown. This should be manageable through the API, which should include how a client discovers dynamic lifecycle options and thus help steering through an application.
- It is often the case that attaining a new lifecycle state is not instantaneous. Clients require a universal mechanism for monitoring such changes.
- A facility to retrieve all events in the lifecycle of a resource can be useful.
- Query provides the means to interrogate the resources behind the API, as well as finding new entry points into the application.

Infrastructure management may be viewed as the manipulation of the underlying graph of resources.

4. Use Cases

Comentario [M.E.G.41]: Improve traceability. See (22) in the review report.

The following use-cases are each derived from one or more of the user-stories above. These use-cases are explored in detail through the development of scenarios, each motivated by some key aspect exemplified by a single user-story. The examples they contain are included purely for illustrative purposes, and should not be interpreted normatively.

4.1 Use Case: Manage containers

A number of user-stories introduce the idea of a *container* as a mechanism for creating and managing resources within the context of an application. Resources grouped together within the same container would typically belong to the same application. A container is identified by a URI so is a resource in its own right. The properties of a container may also represent the *affordances* of that container, enabling clients to determine what other operations they can do on that container. These operations may include descriptions of application specific services that can be invoked by exchanging RDF documents.

- Provides "access guidance for ... resources" (affordances) (from user-story, [Maintaining Social Contact Information](#)).

4.1.1 Primary scenario: create container

Create a new container resource within the LDP server. In [Services supporting the process of science](#), [Research Objects](#) are semantically rich aggregations of resources that bring together data, methods and people in scientific investigations. A basic workflow research object will be created to aggregate [scientific workflows](#) and the artefacts that result from this workflow. The research object begins life as an empty container into which workflows, datasets, results and other data will be added throughout the lifecycle of the project.

EXAMPLE 1

```
@prefix ro:      http://purl.org/wf4ever/ro#
@prefix dct:    http://purl.org/dc/terms/
@prefix ore:    http://www.openarchives.org/ore/
```

```
<> a ro:ResearchObject, ore:Aggregation ;
    dct:created "2012-12-01"^^xsd:dateTime .
```

4.1.2 Alternative scenario: create a nested container

The motivation for nested containers comes from the [System and Software Development Tool Integration](#) user-story. The OSLC Change Management vocabulary allows bug reports to have attachments referenced by the membership predicate `oslc_cm:attachment`. The 'top-level-container' contains issues, and each issue resource has its own container of attachment resources, i.e. `:attachments in the case of :issue1234`.

EXAMPLE 2

```
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

Comentario [M.E.G.42]: Add requirement name.

Comentario [M.E.G.43]: Add numbering.

Comentario [M.E.G.44]: Is this supposed to be an action?

Comentario [M.E.G.45]: Describe example (see 4.3.1)

Código de campo cambiado

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

Comentario [M.E.G.46]: Provide better example description (see 4.3.1)

```

@prefix oslc_cm: <http://open-services.net/ns/cm#>.
@prefix : <http://example.org/>.

:top-level-container rdfs:member :issue1234 .

:issue1234 a oslc_cm:ChangeRequest;
  dcterms:identifier "1234";
  dcterms:type "a bug";
  dcterms:related :issue1235;
  oslc_cm:attachments :attachments123.

:issue1235 a oslc_cm:ChangeRequest;
  dcterms:title "a related bug".

:attachments a oslc_cm:AttachmentList;
  oslc_cm:attachment :attachment324, :attachment251.

```

Comentario [M.E.G.47]: Not needed for example.

Comentario [M.E.G.48]: Not needed for example.

Comentario [M.E.G.49]: Would split in two graphs, providing full URIs.

4.2 Use Case: Manage resources

This use-case addresses the managed lifecycle of a resource and is concerned with resource *ownership*. The responsibility for managing resources belongs with-to their container. For example, a container may accept a request from a client to make a new resource. This use-case focuses on creation and deletion of resources in the context of a container, and the potential for transfer of ownership by moving resources between containers. The ownership of a resource should always be clear; no resource managed in this way should ever be owned by more than one container.

Once a new resource has been created it should be identified by a URI. Clients may defer responsibility for establishing dereferenceable URIs to the container of their data. The container is a natural choice for the endpoint for this interface as it will already have some application-specific knowledge about the contained resources. While the server has ultimate control over resource naming, some applications may require more control over naming, perhaps to provide a more human-readable URI. An LDP server could support something like the Atom Publishing Protocol slug header to convey a user defined naming 'hint' [RFC5023].

Comentario [M.E.G.50]: Out of the scope. See (24) in the review document.

- Non-duplication of resources: "Eliminate multiple copies", representing resources in a single place (from [#Maintaining Social Contact Information](#)).
- Distribution of resources: Linked data "may be stored on separate servers" (from [#Maintaining Social Contact Information](#)).
- Consistent, global naming: Resources should be "linked to consistently, ... instead of maintaining various identifiers in different formats" (from [#Keeping Track of Personal and Business Relationships](#)).

Comentario [M.E.G.51]: Add numbering.

Comentario [M.E.G.52]: Add numbering.

Comentario [M.E.G.53]: Add numbering.

4.2.1 Primary scenario: create resource

Resources begin life by being created within a container. From user-story, [Maintaining Social Contact Information](#), It should be possible to "easily create a new contact and add it to my contacts." This suggests that resource creation is closely linked to the application context. The new resource is created in a

container representing "my contacts." The lifecycle of the resource is **then** linked to the lifecycle of its container. So, for example, if "my contacts" is deleted then a user would also reasonably expect that all contacts within it would also be deleted.

Contact details are captured as an RDF description and its properties, including "names, email addresses, telephone numbers, instant messenger identities and so on." The description may include non-standard RDF; "data about my contacts that the original design didn't consider." The following RDF could be used to describe contact information using the FOAF vocabulary [FOAF]. A contact is represented here by a `foaf:PersonalProfileDocument` defining a resource that can be created and updated as a single-unit, even though it may describe ancillary resources, such as a `foaf:Person`, below.

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

EXAMPLE 3

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<> a foaf:PersonalProfileDocument;
    foaf:PrimaryTopic [
        a foaf:Person;
        foaf:name "Timothy Berners-Lee";
        foaf:title "Sir";
        foaf:firstName "Timothy";
        foaf:surname "Berners-Lee";
        foaf:nick "TimBL", "timbl";
        foaf:homepage <http://www.w3.org/People/Berners-Lee/>;
        foaf:weblog
    <http://dig.csail.mit.edu/breadcrumbs/blog/4>;
        foaf:mbox <mailto:timbl@w3.org>;
        foaf:workplaceHomepage <http://www.w3.org/>.
    ]
```

4.2.2 Alternative scenario: delete resource

Comentario [M.E.G.54]: Sub-scenarios overlooked. See (25) in the review report.

Delete a resource and all its properties. If the resource resides within a container it will be removed from that container, however other links to the deleted resource may be left as dangling references. In the case where the resource is a container, the server may also delete any or all contained resources. In normal practice, a deleted resource cannot be reinstated. There are however, edge-cases where limited undelete may be desirable. Best practice states that "Cool URIs don't change" [COOLURIS], which implies that deleted URIs shouldn't be recycled.

Código de campo cambiado

4.2.3 Alternative scenario: moving contained resources

Many **rR** resources may have value beyond the life of their membership in a container. This implies methods to add references to revise container membership. Cloning container members for use in other containers results in duplication of information and maintenance problems; web practice is to encourage the creation of one resource, which may be referenced as many places as necessary. A change of ownership may - or may not - imply a change of URI, depending upon the specific server naming policy. While assigning a

new URI to a resource is discouraged [WEBARCH], it is possible to indicate that a resource has moved with an appropriate HTTP response.

Comentario [M.E.G.55]: Out of the scope. See (26) in the review report.

4.3 Use Case: Retrieve resource description

Access the current description of a resource, containing properties of that resource and links to related resources. The representation may include descriptions of related resources that cannot be accessed directly. Depending upon the application, a server may enrich the retrieved RDF with additional triples. Examples include adding incoming links, `owl:sameAs` closure and `rdf:type` closure. The HTTP response should also include versioning information (i.e. last update or entity tag) so that subsequent updates can ensure they are being applied to the correct version.

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

- Use standard vocabularies as appropriate to enable a "common understanding of the resource" (from [Maintaining Social Contact Information](#)).
- A "scalable linking model is key" (from [#Municipality Operational Monitoring](#)).

Comentario [M.E.G.56]: Add requirement name.

Comentario [M.E.G.57]: Add numbering.

Comentario [M.E.G.58]: Add requirement name.

Comentario [M.E.G.59]: Add numbering.

Comentario [M.E.G.60]: Add functionality.

4.3.1 Primary scenario

The user-story [Project Membership Information](#) discusses the representation of information about people and projects. It calls for "Resource descriptions for each person and project" allowing project teams to review information held about these resources. The example below illustrates the kinds of information that might be held about organizational structures based on the [Epimorphics organizational ontology](#).

Note that the example below defines two resources (shown as separate sections below) that will be hosted on an LDP server based at `http://example.com/`. The representations of these resources may include descriptions of related resources, such as `http://www.w3.org/`, that fall under a different authority and therefore can't be served from the LDP server at this location. Thus, in the representation in Example 4, no information about the role of `http://example.com/member1` is locally defined, therefore it would have to be retrieved by the dereferencing the individual defined for the `org:role` property, which in this case points to another individual hosted at `http://example.com` (`http://example.com/director`) whose contents are defined in Example 5.

EXAMPLE 4

```
@prefix org: <http://www.w3.org/ns/org#> .
@prefix owltime: <http://www.w3.org/2006/time> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://example.com/> .

<member1> a org:Membership ;
  org:member <http://www.w3.org/People/Berners-Lee/card#i> ;
  org:organization http://www.w3.org/> ;
  org:role <director> ;
```

```

    org:memberDuring [a owltime:Interval; owltime:hasBeginning [
        owltime:inXSDDateTime "1994-10-
01T00:00:00Z"^^xsd:dateTime]] .

<http://www.w3.org/> a org:FormalOrganization ;
    skos:prefLabel "The World Wide Web Consortium"@en ;
    skos:altLabel "W3C" .

```

EXAMPLE 5

```

@prefix org: <http://www.w3.org/ns/org#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://example.com/> .

<director> a org:Role ;
    rdfs:label "Director" .

```

4.3.2 Alternative scenario: retrieve description of a non-document resource

In many cases, the things that are of interest are not always the things that are resolvable. The example below demonstrates how a FOAF profile may be used to distinguish between the person and the profile; the former being the topic of the latter. This begs the question as to what a client should do with such non-document resources. In this case the HTTP protocol requires that the fragment part be stripped off before requesting the URI from the server. The result is a resolvable URI for the profile.

EXAMPLE 6

```

@base <http://www.w3.org/People/Berners-Lee/card>
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<> a foaf:PersonalProfileDocument ;
    dc:title "Tim Berners-Lee's FOAF file" ;
    foaf:homepage <http://www.w3.org/People/Berners-Lee/> ;
    foaf:primaryTopic <#i> .

```

Comentario [M.E.G.61]: See 4.3.1 for better example introduction.

Comentario [M.E.G.62]: Describe how the client/server would do with the example below.

4.4 Use Case: Update existing resource

Change the RDF description of a LDP resource, potentially removing or overwriting existing data. This allows applications to *enrich* the representation of a resource by adding additional links to other resources.

- Unrestricted vocabulary: It should be possible be "able to add ... application-specific data" to resources (from [#Maintaining Social Contact Information](#)).

Comentario [M.E.G.63]: Add numbering.

4.4.1 Primary scenario: enrichment

This relates to user-story [Metadata Enrichment in Broadcasting](#) and is based on the [BBC Sports Ontology](#). The *resource-centric* view of linked-data provides a natural granularity for substituting, or overwriting a resource and its data. The simplest kind of update would simply replace what is currently known about a resource with a new representation. There are two distinct resources in the example below; a sporting event and an associated award. The granularity of

the resource would allow a user to replace the information about the award without disturbing the information about the event.

EXAMPLE 7

```
@prefix sport: <http://www.bbc.co.uk/ontologies/sport/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

:mens_sprint a sport:MultiStageCompetition;
  rdfs:label "Men's Sprint";
  sport:award <#gold_medal> .

<#gold_medal> a sport:Award .
```

We can enrich the description as events unfold, linking to the winner of the gold medal by substituting the above description with the following.

EXAMPLE 8

```
@prefix sport: <http://www.bbc.co.uk/ontologies/sport/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:mens_sprint a sport:MultiStageCompetition;
  rdfs:label "Men's Sprint";
  sport:award <#gold_medal> .
<#gold_medal> a sport:Award;
  sport:awarded_to [
    a foaf:Agent ;
    foaf:name "Chris Hoy" .
  ] .
```

Comentario [M.E.G.64]: Introduce example 7, and use full URIs (see section 4.3.1)

Comentario [M.E.G.65]: Make impersonal.

Comentario [M.E.G.66]: Use full URIs (see section 4.3.1)

4.4.2 Alternative scenario: selective update of a resource

This relates to user-story [Data Catalogs](#), based on the [Data Catalog Vocabulary](#). A catalogue is described by the following RDF model.

EXAMPLE 9

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

:catalog a dcat:Catalog ;
  dcat:dataset :dataset/001;
  dcterms:issued "2012-12-11"^^xsd:date.
```

A catalog may contain multiple datasets, so when linking to new datasets it would be simpler and preferable to selectively add just the new dataset links. A

[Talis changeset](#) could be used to add a new `dc:title` to the dataset. The following update would be directed to the catalogue to add an additional dataset.

Comentario [M.E.G.67]: Introduce example and use full URIs (see section 4.3.1)

Comentario [M.E.G.68]: Add appropriate informative reference.

Con formato: Fuente: (Predeterminado) Courier New, 11 pts

EXAMPLE 10

```
@prefix : <http://example.com/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix cs: <http://purl.org/vocab/changeset/schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<change1>
```

```

a cs:ChangeSet ;
cs:subjectOfChange :catalog ;
cs:createdDate "2012-01-01T00:00:00Z" ;
cs:changeReason "Update catalog datasets" ;
cs:addition [
  a rdf:Statement ;
  rdf:subject :catalog ;
  rdf:predicate dcat:dataset ;
  rdf:object :dataset/002 .
] .

```

Comentario [M.E.G.69]: Use full URIs (see section 4.3.1)

Comentario [M.E.G.70]: Missing explicit requirements. See (27) in the review report.

4.5 Use Case: Determine if a resource has changed

It should be possible to retrieve versioning information about a resource (e.g. last modified or entity tag) without having to download a representation of the resource. This information can then be compared with previous information held about that resource to determine if it has changed. This versioning information can also be used in subsequent *conditional* requests to ensure they are only applied if the version is unchanged.

4.5.1 Primary scenario

Based on the user story, [Constrained Devices and Networks](#), an LDP server could be configured to act as a proxy for a CoAP [COAP] based [Web of Things](#). As an observer of CoAP resources, the LDP server registers its interest so that it will be notified whenever the sensor reading changes. Clients of the LDP can interrogate the server to determine if the state has changed.

Comentario [M.E.G.71]: Add functionality.

Código de campo cambiado

Comentario [M.E.G.72]: Add appropriate informative reference.

Código de campo cambiado

In this example, the information about a sensor and corresponding sensor readings can be represented as RDF resources. The first resource below, represents a sensor described using the [Semantic Sensor Network](#) ontology.

Comentario [M.E.G.73]: Add appropriate informative reference.

EXAMPLE 11

```
@prefix : <http://example.com/energy-management/>.
```

```

<> a :MainsFrequencySensor;
rdfs:comment "Sense grid load based on mains frequency";
ssn:hasMeasurementCapability [
  a :FrequencyMeasurementCapability;
  ssn:hasMeasurementProperty <#property_1> .
] .

```

Comentario [M.E.G.74]: Missing namespace prefix.

Comentario [M.E.G.75]: Missing namespace prefix.

Comentario [M.E.G.76]: Add ssn and rdfs prefix mappings.

Comentario [M.E.G.77]: Use full URIs (see section 4.3.1)

The value of the sensor changes in real-time as measurements are taken. The LDP client can interrogate the resource below to determine if it has changed, *without* necessarily having to download the RDF representation. As different sensor properties are represented disjointly (separate RDF representations) they may change independently.

EXAMPLE 12

```
@prefix : <http://example.com/energy-management/>.
```

```

<http://example.com/energy-management#property_1>
:hasMeasurementPropertyValue <> .
<> a :FrequencyValue;

```

Comentario [M.E.G.78]: Missing namespace prefix.

:hasQuantityValue "50"^^xsd:float.

Comentario [M.E.G.79]: Missing namespace prefix.

Comentario [M.E.G.80]: Use full URIs (see section 4.3.1)

4.6 Use Case: Aggregate resources

There is a requirement to be able to manage *collections* of resources. The concept of a collection overlaps with, but is distinct from that of a container. These collections are (weak) aggregations, unrelated to the lifecycle management of resources, and distinct from the ownership between a resource and its container. However, the composition of a container may be reflected as a collection to support navigation of the container and its contents. There is a need to be able to create collections by adding and deleting individual membership properties. Resources may belong to multiple collections, or to none.

- Resource descriptions are a "mix of simple data and collections" (from [#Keeping Track of Personal and Business Relationships](#)).
- Relative URIs: It should be possible to "ship payloads of RDF" for a collection as a whole without breaking internal links (from [Constrained Devices and Networks](#)).

Comentario [M.E.G.81]: Add requirement name.

Comentario [M.E.G.82]: Add numbering.

Comentario [M.E.G.83]: Add numbering.

4.6.1 Primary scenario: add a resource to a collection

Comentario [M.E.G.84]: Misleading example. See (28) in the review report.

This example is from [Library Linked Data](#) and LLD-UC [LLD-UC], specifically [Subject Search](#).

There is an existing collection at `<http://example.com/concept-scheme/subject-heading>` that defines a collection of subject headings. This collection is defined as a `skos:ConceptScheme` and the client wishes to insert a new concept into the scheme. which will be related to the collection via a `skos:inScheme` link. The new subject-heading, "outer space exploration", is not necessarily owned by a container. The following RDF would be added to the (item-level) description of the collection.

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

Con formato: Fuente: 11 pto

Con formato: Fuente: (Predeterminado) Courier New, 11 pto

Con formato: Fuente: 11 pto

EXAMPLE 13

```
@prefix scheme : <http://example.com/concept-scheme/>.
@prefix concept : <http://example.com/concept/>.
```

```
scheme:subject-heading a skos:ConceptScheme.
```

```
concept:Outer+space+Exploration skos:inScheme scheme:subject-
heading.
```

Comentario [M.E.G.85]: Add skos prefix mapping.

4.6.2 Alternative scenario: add a resource to multiple collections

Comentario [M.E.G.86]: Incomprehensible scenario. See (29) in the review report.

Logically, a resource should not be owned by more than one container. ~~however~~However, it may be a member of multiple collections which define a weaker form of *aggregation*. As this is simply a manipulation of the RDF description of a collection, it should be possible to add the same resource to multiple collections.

As a machine-readable collection of medical terms, the [SNOMED](#) ontology is of key importance in [healthcare](#). SNOMED CT allows concepts with more than one parent that don't fall into a lattice. In the example below, the same concept may fall under two different parent concepts. The example uses `skos:narrowerTransitive` to elide intervening concepts.

EXAMPLE 14

```
@prefix : <http://example.com/snomed/>.

:_119376003 a skos:Concept ;
  skos:prefLabel "Tissue specimen"
  skos:narrowerTransitive :TissueSpecimenFromHeart.

:_127462005 a skos:Concept ;
  skos:prefLabel "Specimen from heart"
  skos:narrowerTransitive :TissueSpecimenFromHeart.

:_128166000 a skos:Concept ;
  rdfs:label "Tissue specimen from heart".
```

4.7 Use Case: Filter resource description

This use-case extends the normal behaviour of retrieving an RDF description of a resource, by dynamically excluding specific (membership) properties. For containers, it is often desirable to be able to read a collection, or item-level description that excludes the container membership.

4.7.1 Primary scenario: retrieve collection-level description

This scenario, based on [Library Linked Data](#), uses the Dublin Core Metadata Initiative [Collection-Level](#) description. A collection can refer to any aggregation of physical or digital items. This scenario covers the case whereby a client can request a collection-level description as typified by the example below, without necessarily having to download a full listing of the items within the collection.

EXAMPLE 15

```
@prefix rdf: <rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix : <http://example.org/bookshelf/>.
@prefix dcmitype: <http://purl.org/dc/dcmitype/>.
@prefix cld: <http://purl.org/cld/terms/>.
@prefix dcterms: <http://purl.org/dc/terms/>.

<> dc:type dcmitype:Collection ;
  dc:title "Directory of organizations working with Linked Data"
  ;
  dcterms:abstract "This is a directory of organisations
specializing in Linked Data."
  cld:isLocatedAt <http://dir.w3.org>
  cld:isAccessedVia
<http://dir.w3.org/rdf/2012/directory/directory-
list.xhtml?construct>
```

4.7.2 Alternative scenario: retrieve item-level description of a collection

Comentario [M.E.G.87]: Add appropriate informative reference.

Código de campo cambiado

Con formato: Fuente: (Predeterminado) Courier New, 11 pts

Con formato: Fuente: 11 pts

Comentario [M.E.G.88]: Add skos prefix mapping.

Comentario [M.E.G.89]: Use full URIs (see section 4.3.1).

Comentario [M.E.G.90]: Missing explicit requirements. See (30) in the review report.

Comentario [M.E.G.91]: Improve example introduction.

Comentario [M.E.G.92]: Invalid syntax. Not needed for the example.

Comentario [M.E.G.93]: Use full URIs (see section 4.3.1)

This use-case scenario, also based on [Library Linked Data](#), focuses on obtaining an item-level description of the resources aggregated by a collection. The simplest scenario is where the members of a collection are returned within a single representation, so that a client can explore the data by following these links. Different applications may use different membership predicates to capture this aggregation. The example below uses `rdfs:member`, but many different membership predicates are in common use, including RDF Lists. Item-level descriptions can be captured using the Functional Requirements for Bibliographic Records ([FRBR](#)) [ontology](#).

Código de campo cambiado

Con formato: Fuente: (Predeterminado) Courier New, 11 pts

Comentario [M.E.G.94]: Add appropriate informative reference.

EXAMPLE 16

```
@prefix frbr: <http://purl.org/vocab/frbr/core#>.

<> rdfs:member <#ebooks97>, <#ebooks21279>.

<#work97> a frbr:LiteraryWork;
  dc:title "Flatland: a romance of many dimensions" ;
  frbr:creator <#Abbott_Edwin>;
  frbr:manifestation <ebook97>.

<#work21279> a frbr:LiteraryWork;
  dc:title "2 B R 0 2 B" ;
  frbr:creator <#Vonnegut_Kurt>;
  frbr:manifestation <ebook21279>.
```

Collections are potentially very large, so some means may be required to limit the size of RDF representation returned by the LDP server (e.g. pagination).

Comentario [M.E.G.95]: Add rdfs and dc prefix mappings.

Comentario [M.E.G.96]: Use full URIs for containers and resources (see section 4.3.1)

Comentario [M.E.G.97]: Needs breakdown. See (31) in the review report.

Comentario [M.E.G.98]: Refer to binary or non-RDF resources See (33) in the review report..

Comentario [M.E.G.99]: Missing explicit requirements. See (32) in the review report.

4.8 Use Case: Manage media resources

It should be possible to easily add non-RDF media resources to containers that accept them. Media resources may be updated and removed during the lifecycle of the container.

4.8.1 Primary scenario: access media resources

From the [User-user Story-story Sharing Binary Resources and Metadata](#) it should be possible to easily add non-RDF resources to containers that accept them. Clients submit a non-RDF representation to a container in a media type accepted by that container. The container creates a URI to represent this media resource, and creates a link from the container to the new URI. The media resource may have an explicit [RDF](#) representation of the media type. It should be possible to find the metadata about such a resource and to access and edit it in the usual ways.

This example uses the Ontology for Media Resources to describe a media resource added to a collection [[MEDIAONT](#)].

EXAMPLE 17

```
@prefix ma: <http://www.w3.org/ns/ma-ont#> .

<dataset> a ma:Collection ;
  ma:hasMember <dataset/imagen1.jpg>
```

Con formato: Español (alfab. internacional)

Comentario [M.E.G.100]: Missing namespace prefix.

```
<dataset/image1.jpg> a ma:MediaResource ;
    ma:hasFormat "image/jpeg" .
```

4.8.2 Alternative scenario: media-resource attachments

A resource may have multiple *renditions*; ~~the idea that you can have i.e.,~~ a PDF and a JPEG representing the same thing. A user is trying to create a work order along with an attached image showing a faulty machine part. To the user and to the work order system, these two artifacts are managed as a set. A single request may create the work order, the attachment, and the relationship between them, atomically. When the user retrieves the work order later, they expect a single request by default to retrieve the work order plus all attachments. When the user updates the work order, e.g. to mark it completed, they only want to update the work order proper, not its attachments. Users may add/remove/replace attachments to the work order during its lifetime.

Comentario [M.E.G.101]: Orphan scenario. See (34) in the review report.

5. Requirements

Comentario [M.E.G.102]: Improve traceability. See (35) in the review report.

5.1 Functional Requirements

1. Create Containers, from [Use Case: Manage containers](#)
2. Creation of nested containers, from [Use Case: Manage containers](#)
3. Creation of resources (within a container), from [Use Case: Manage resources](#)
4. Deletion of resources, from [Use Case: Manage resources](#)
5. Moving contained resources, from [Use Case: Manage resources](#)
6. Retrieve resource description, from [Use Case: Retrieve resource description](#)
7. Retrieve description of a non-document resource, from [Use Case: Retrieve resource description](#)
8. Enrichment (substituting update of existing resource), from [Use Case: Update existing resource](#)
9. Selective update of a resource, from [Use Case: Update existing resource](#)
10. Determine if a resource has changed, from [Use Case: Determine if a resource has changed](#)
11. Add a resource to a collection, from [Use Case: Aggregate resources](#)
12. Add a resource to multiple collections, from [Use Case: Aggregate resources](#)
13. Retrieve collection-level description, from [Use Case: Filter resource description](#)
14. Retrieve item-level description of a collection, from [Use Case: Filter resource description](#)
15. Access media resources, from [Use Case: Manage media resources](#)
16. Media-resource attachments, from [Use Case: Manage media resources](#)

5.2 Non-Functional Requirements

1. Provide access guidance to resources, from [Use Case: Manage containers](#)

2. Non-duplication of resources, from [Use Case: Manage resources](#)
3. Distribution of resources, from [Use Case: Manage resources](#)
4. Consistent, global naming, from [Use Case: Manage resources](#)
5. Use standard vocabularies as appropriate, from [Use Case: Retrieve resource description](#)
6. Scalable linking model, from [Use Case: Retrieve resource description](#)
7. Unrestricted vocabulary, from [Use Case: Update existing resource](#)
8. Resource descriptions are a "mix of simple data and collections", from [Use Case: Aggregate resources](#)
9. Relative URIs enabling sharing of collections, from [Use Case: Aggregate resources](#)

A. Acknowledgements

This section is non-normative.

We would like to acknowledge the contributions of user-story authors: Christophe Guéret, Roger Menday, Eric Prud'hommeaux, Steve Speicher, John Arwe, Kevin Page.

B. Change History

This section is non-normative.

- 2012-12-14 - Initial ReSpec'ing framework for [Workgroup working wiki document](#) (SS)
- 2012-12-16 - Pulled in and ReSpec'd content from [Workgroup working wiki document](#) (SS)
- 2012-12-16 - Fixed cross section links and initial pass at biblio refs (SS)

C. References

C.1 Informative references

[COAP]

E Shelby; et al. [Constrained Application Protocol \(CoAP\)](#). IETF Internet Draft, December 2012. URL: <http://tools.ietf.org/html/draft-ietf-core-coap>

[COOLURIS]

Richard Cyganiak; Leo Sauermann. [Cool URIs for the Semantic Web](#). 3 December 2008. W3C Note. URL: <http://www.w3.org/TR/2008/NOTE-cooluris-20081203>

[DAP-REQS]

Robin Berjon et al. [Device API Requirements](#) 15 October 2009. Working Group Note. URL: <http://www.w3.org/TR/2009/NOTE-dap-api-reqs-20091015/>

[DCAT-UCR]

R. Cyganiak; F. Maali. [Use Cases and Requirements for the Data Catalog Vocabulary](#) 16 December 2012. W3C Editor's Draft. URL: <http://dvcs.w3.org/hg/gld/raw-file/default/dcat-ucr/index.html>.

Código de campo cambiado

Código de campo cambiado

[FOAF]

Dan Brickley, Libby Miller. [FOAF Vocabulary Specification 0.98](#). 9 August 2010. URL: <http://xmlns.com/foaf/spec/>

[LINKED-DATA]

Tim Berners-Lee. [Linked Data Design Issues](#). 27 July 2006. W3C-Internal Document. URL: <http://www.w3.org/DesignIssues/LinkedData.html>

Código de campo cambiado

[LINKED-DATA-PLATFORM]

Steve Speicher et al. [Linked Data Platform 1.0](#) 25 October 2012. W3C Working Draft. URL: <http://www.w3.org/TR/ldp/>

[LLD-UC]

D. Vila Suero. [Library Linked Data Incubator Group: Use Cases](#). 25 October 2011. W3C Incubator Group Report. URL: <http://www.w3.org/2005/Incubator/ld/XGR-ld-usecase-20111025/>

Código de campo cambiado

[MEDIA-FRAGMENTS-REQS]

Raphael Troncy; Erik Mannens. [Use cases and requirements for Media Fragments](#). W3C Working Draft 17 December 2009. URL: <http://www.w3.org/TR/2009/WD-media-frags-reqs-20091217>

[MEDIAONT]

WonSuk Lee; et. al. [Ontology for Media Resources 1.0](#). 9 February 2012. W3C Recommendation. URL: <http://www.w3.org/TR/2012/REC-mediaont-10-20120209/>

[POWDER-USE-CASES]

Phil Archer. [POWDER: Use Cases and Requirements](#). 31 October 2007. W3C Note. URL: <http://www.w3.org/TR/2007/NOTE-powder-use-cases-20071031>

Código de campo cambiado

[RDB2RDF-UC]

Eric Prud'hommeaux; Michael Hausenblas. [Use Cases and Requirements for Mapping Relational Databases to RDF](#). 8 June 2010. W3C Working Draft. URL: <http://www.w3.org/TR/2010/WD-rdb2rdf-ucr-20100608/>

Código de campo cambiado

[RFC5023]

J. Gregorio, B. de hOra. [Atom Publishing Protocol](#). IETF RFC 5023. October 2007. URL: <http://www.ietf.org/rfc/rfc5023.txt>

[WEBARCH]

Norman Walsh; Ian Jacobs. [Architecture of the World Wide Web, Volume One](#). 15 December 2004. W3C Recommendation. URL: <http://www.w3.org/TR/2004/REC-webarch-20041215/>

Código de campo cambiado