

2013 年第 3 期（总第 18 期）

# 数字图书馆标准规范

## 跟踪扫描

主办单位：中国科学院国家科学图书馆

2013 年 4 月

为传播科学知识，促进业界交流，  
特编译《标准规范跟踪扫描》，仅供个人  
学习、研究使用。

## 目 录

【标准规范报道】 .....	1
1、NISO发布描述与识别电子期刊的推荐实践 .....	1
2、PROV——一个关于来源交换的框架 .....	2
3、IMS全球学习组织宣布关于数字内容和应用程序集成的开放标准要求 .....	3
4、IEEE物联网 (IoT) 研讨会提供跨行业合作机会并识别标准化缺口 .....	4
【标准规范推介】 .....	5
一、关联数据平台用例与需求 .....	5
二、IMS学习工具互操作性 (LTI) 安全版本 2.0 公共草案 .....	20

## 【标准规范报道】

### 1、NISO 发布描述与识别电子期刊的推荐实践

#### 确保学术期刊的长期在线可访问性（即使标题和出版商发生变化）的推荐

巴尔的摩，马里兰州——2013 年 3 月 27 日——美国国家信息标准协会 (NISO) 宣布发布一个新的推荐实践：PIE-J：电子期刊的描述与识别 (NISO RP-16-2013)。该推荐实践旨在为出版商和平台提供商在电子期刊的描述（尤其是标题描述、ISSN 的精确使用以及引文实践领域）上提供指导，也为了解决一些期刊、馆藏和图书馆电子资源中长期存在的问题。除了推荐外，文档还包含了以各种出版商的在线期刊平台的截图展现的优良实践的大量实例；对标题历史以及 ISSN 信息的获取有益的资源讨论；国际标准期刊编号 (ISSN) 的概述及其正确使用的要点；数字对象标识符 (DOI®)、CrossRef 注册代理的解释说明以及期刊标题管理 DOIs 的使用技巧；相关标准和推荐实践的回顾。

“引文形成了许多学术研究的基础。除非期刊网站精确地和一致地列出其所发布的内容的标题，否则用户访问其所需要的内容将会大大减少，” Cindy Hepfer（布法罗纽约州立大学继续电子资源管理与编目图书馆员兼 NISO PIE-J 工作组联合组长）解释道，“例如，许多电子期刊出版商和整合者如今将最初以早期的标题发布的数字内容换做现在的标题放在网站上，使用现在的 ISSN，因此严重地妨碍了研究者发现或识别当前所寻找的内容的能力。PIE-J 计划旨在解决这些问题。”

“电子期刊的出版商和供应商对其网站的多样化设计引以为豪，” Bob Boissy (Springer 客户发展与战略联盟经理兼 NISO PIE-J 工作组联合组长) 说道，“然而这些网站如何呈现、识别以及将他们所列出出版物关联在一起可使得终端用户容易地或令人沮丧地、更甚完全徒劳地发现及使用文章。PIE-J 推荐实践指南的应用将促成研究者、作者、图书馆员、在线供应商和出版商更好地发现和访问对其有益的东西。”

“PIE-J 推荐实践提供了一个清晰且简洁的指南列表，出版商可容易地实施，以促进其电子期刊内容的长期访问，” Todd Carpenter (NISO 执行主任) 声明，“该建设性意见将有助于出版商进行原生数字内容的描述并支持最初只以印刷形式出版的期刊内容的继续数字化。”

PIE-J 推荐实践草案以及推荐概述手册可在 NISO PIE-J 工作室网站 [www.niso.org/workrooms/piej/](http://www.niso.org/workrooms/piej/) 中获得。

(编译自 :NISO Publishes Recommended Practice on Presentation and Identification of E-Journals.

[http://www.niso.org/news/pr/view?item\\_key=a4cb17fa3cd324c142e9a33dc74e577663578](http://www.niso.org/news/pr/view?item_key=a4cb17fa3cd324c142e9a33dc74e577663578)

359. [2013-03-27])

(岳增慧编译, 吴贝贝校对)

## 2、PROV——一个关于来源交换的框架

W3C 来源工作小组同时发布了 13 个文档, 定义网络上交换来源的一个框架。它是一个对 PROV 的完整的稳定的定义, 包含 4 个推荐文档。

13 个文档是巨大的, 这是因为 PROV 分成了很多块, 可以适用于特定的团体和用法。在这里, 提供一些关于 PROV 框架的指导和不同文档的角色。

### 核心: 一个数据模型

PROV 的中心是一个数据模型, PROV-DM, 定义了描述来源的一个词表。这些术语顾及了来自数据、过程和代理视角的来源描述。PROV-DM 可以通过多样的序列化技术写出来。PROV 定义了 3 个序列。

1. PROV-O 是一个不太重要的 OWL2 本体, 设计用于关联数据和语义网应用。
2. PROV-N 是一个严密的句法, 针对人类消费。
3. PROV-XML 是一个本地的 XML 体系, 特别为 XML 团体服务。

使用这些序列, 在应用程序之间可以揭示和交换来源。PROV-DM 和它的序列在设计的时候考虑到了扩展性。目前已有 PROV-O 的几个延伸, 这是为特定的团体设计的。

### 支持验证

PROV-DM 和相关联序列的目的是为了在写来源的时候考虑灵活性。因为 PROV 逐渐被使用, 需要考虑灵活性。然而, 也意识到一些用户想要一个指南来确保他们的来源是持续的, 就像有 HTML 验证器, 也要提供 PROV 验证器。PROV Constraints 定义了一系列的约束, 可以被用来实现验证器。PROV Constraints 由在 PROV Sem 中定义的一个正式的语义支持。

### 数据模型延伸

在多样化的应用中可以看到为来源作模型的两个应用案例。一个案例是聚集信息成为集合/词典类型结构 (例如一个带有文档的文件夹), 另一个是将多样化的来源追踪汇集起来。PROV-Dictionary 和 PROV-Links 提供定义构造来帮助构建模型。

### 获取来源

最终, 一旦为来源建立模型, 就能够容易地揭示它。PROV-AQ 定义了如何使用它已存在的网络结构, 如关联标头, 使来源可获取。PROV-AQ 设计的一个关键部分是使它独立于任何序列化格式, 因此可以使用最符合需要的格式。

### 都柏林核心集

都柏林核心集 (DC) 是最广泛发布的词汇之一, 它的很多术语与来源有关。与 DC 团体合作, 我们已经定义了 DC 与 PROV-O (PROV-DC) 之间的映射。这意味着支持 PROV 的应用程

序可以容易地使用来源,该来源已被揭示为DC。

## 概要

PROV 提供了以互操作的方式写出、验证和交换来源信息的一个框架。已经有超过 60 种的工具支持 PROV, 期望未来有更多。

(编译自: PROV – A Framework for Provenance Interchang.

<http://www.w3.org/blog/SW/2013/03/21/prov-a-framework-for-provenance-interchange/>. [2013-03-21])

(吴贝贝编译, 王妍校对)

### 3、IMS 全球学习组织宣布关于数字内容和应用程序集成的开放标准要求

随着教育发展为电子课本、数字视频与交互数字格式, IMS 成员 K-12 区和高等教育机构率先倡导基于集成的开放标准。

2013 年 2 月 19 日, IMS 全球学习组织 (IMS Global) 在美国孤星大学的季度会议上宣布了一个简要文件的可用性, 该文件给一些地区和国家提供了关于如何实现基于数字课程资料集成的开放标准的公共指导, 该数字课程资料以 IMS 标准为基础。

简要的 4 页文档——数字内容和与企业学习平台应用程序集成的开放标准要求, 意味着国家和 K12 校区希望向数字教育演化的途径能够落实到位, 该数字教育有来自教育产业的基层支持。文件也为使用开放标准的企业系统提供集成数字内容和应用程序的指导, 如学习管理系统、教学管理系统、教育资源门户、内容管理系统、学生系统等。最重要的是, 通过各种广泛的平台和内容来源, IMS 已经颁布了超过 140 个的一致性认证。

“我们一直致力于向国家、地区、大学、系统和专业协会宣传, 在今天有可能实现创新数字内容与应用程序的快速、低成本的集成, 而不依赖于任何供应商、基金会或政府项目的 API,” 非盈利的 IMS 全球学习组织的行政长官 Rob Abel 说, “例如, 许多国家都对实施关于数字课程集成的开放标准政策感兴趣, 但可能不清楚在广泛和日益庞大的教育技术供应商之中在今天有可用的解决方案。”

“尽管有许多有趣的倡议出现, 如共享学习协作和学习注册, 但需要被解决的关键问题是数字课程的快速、低成本集成, 使得老师和学生能够无缝使用,” GA 福赛斯郡学区的首席技术和信息官 Bailey Mitchell 评论道, “IMS 认证产品是实现目标的方法, 以便于地区和供应商都不需要创建高代价的自定义方法。”

数字内容和应用程序的IMS标准目前被超过 60 个国家、地区、高等教育机构和系统所倡导。IMS支持者的完整列表: <http://www.imsglobal.org/membersandaffiliates.html>

关于IMS全球学习组织 (IMS Global) [WWW.imsglobal.org](http://www.imsglobal.org)

IMS Global 是能够合理衡量并提高教育参与和成就的提升技术的非营利组织。IMS 成员

本作品采用[知识共享署名-非商业性使用-禁止演绎 2.5 中国大陆许可协议](#)进行许可。

是重要的供应商、机构和政府组织,它们能够使教育的未来通过互操作性和采用计划来协调。IMS 赞助商的学习影响力:一个全球研判计划与会议来确认在教育访问、负担能力和质量方面的创新技术影响力。

(编译自:IMS Global Learning Consortium Announces Open-Standards Requirements for Digital Content and Application Integration.

<http://www.imsglobal.org/pressreleases/pr130219.html>. [2013-02-19])

(王妍编译,岳增慧校对)

#### 4、IEEE 物联网 (IoT) 研讨会提供跨行业合作机会并识别标准化缺口

2013 年 3 月 28 日,专业化组织 IEEE 宣布将在 2013 年 4 月 12 日在中国深圳将举办一个物联网研讨会。该研讨会将为与会者提供识别有关物联网的合作机会和标准化缺口的方法。另外,该研讨会将会帮助产业在物联网市场持续增长,为物联网行业的共识发展利用 IEEE 的价值和平台,并协助创建一个充满活力的物联网生态系统。

“从手上到家、到办公室,随着我们的连接设备成为我们日常生活不可分割的一部分,计算机和网络技术正扮演着越来越重要的角色,”IEEE-SA 企业咨询集团(CAG)副主席、IEEE 标准协会标准委员会(SASB)的成员、IEEE 802.3 以太网工作组副主席和美国博通公司的高级技术总监 Wael William Diab 说,“IEEE-SA 侧重于培养跨行业创新和识别物联网技术对工业和人类的价值与利益,以及发展中必不可少的网络标准,例如 IEEE 802 族”

期望在 2020 年,有 500 亿到 1000 亿的事物被电子链接,IEEE-SA 出版了许多与物联网相关的标准,并有侧重其未来的几个发展计划。物联网将通过电子链接每天出现的数以亿计的“事物”和对象来支持机器间交流的方式,它是全球公认的网络。

2013 年 3 月 9 日,在美国德克萨斯州奥斯汀的 SXSW 互动会议的一个小组中,物联网的未来被来自 IEEE-SA、博通、意法微电子和 Telx 集团的远见者所讨论。组员们探索了互连的动态性和电子设备的聚合,并且强调了需要一个更跨学科的方式来设计、建造、发展、交流与隐私。

“做好准备:物联网将要面对人联网,即普遍的连通性与嵌入式智能能够使环境了解我们,并迎合我们的需求与习惯当最大化能源效率时来确保我们舒适,”IEEE SASB 的成员、IEEE-SA 企业咨询集团(CAG)的成员、IEEE P1901.2 工作组的副主席、IEEE PHYMAC 子工作组的主席,与意法微电子进行工业和能源转换部门的市场发展总监 Oleg Logvinov 说,“这些是近几年来你所听到的“智能家居”和“智能城市”。IEEE-SA 在朝着集中网络标准 IEEE 1905.1-2013 这一目标已经迈出了第一步——集中数字家庭异构技术网络的标准。我们也在寻找应用程序级的聚合来将娱乐、能源互联网与电气车辆集合起来。”

关于 IEEE 标准协会

IEEE 标准协会，一个全球公认的标准制定机构，通过开放获取方式来发展共识标准，该方式促进工业发展并汇集了一个广泛的利益相关者团体。IEEE 标准设定规范和基于当前科技知识的最佳案例。IEEE-SA 有一系列超过 900 个的现行标准，500 多个标准在发展中。

### 关于 IEEE

IEEE，一个大的全球技术性专业化组织，它致力于为造福人类推进技术。通过其高度被引用的出版物、会议、技术标准以及专业和教育活动，IEEE 在从航天系统、计算机和电信到生物学工程、电力和消费性电子产品等一系列各种各样的领域中备受信任。

(编译自：IEEE INTERNET OF THINGS (IoT) WORKSHOP TO PROVIDE CROSS-INDUSTRY COLLABORATION OPPORTUNITIES AND IDENTIFY STANDARDIZATION GAPS.

[http://standards.ieee.org/news/2013/iot\\_china.html](http://standards.ieee.org/news/2013/iot_china.html). [2013-03-28])

(王妍编译，岳增慧校对)

## 【标准规范推介】

### 一、关联数据平台用例与需求

#### 摘要

一组促进简单读写关联数据架构的用户故事、用例、场景介绍与需求，基于 HTTP 访问网络资源（使用 RDF 描述其状态）。

#### 1 范围和动机

#### 2 本文档的组织

用例以一种叙事风格来获取，描述了从用户故事中得到的行为或行为集。它们用来自典型用户故事的具体例子进行润色。贯穿始终的目标是避免协议（尤其是 HTTP 协议）细节，以及任何特定词汇（可能被 LDP 规范所介绍）的使用。

本文档的组织如下：

- **用户故事**获取源自用户或应用视角下的系统需求声明。它们通常是轻量级的、非正式的、可以从一行到一个或两个段落（有时被描述为“叙事诗”）。每一个用户故事的分析将展现许多（功能）用例以及其他非功能需求。
- **用例**通常是获取并建模功能需求。用例描述了不同条件下系统的行为，对谁对系统做了什么、出于何种目的进行编目，但并不关心系统设计或实施。每一个用例由一个参考数进行标识，以避免其他文件的交叉引用；本文档中的用例索引是基于 rdb2rdf 用例。很多类型可以用来获取用例，从一个简单的叙述到一个结构性的描述，先前/发送条件还有 POWDER 中的逐步行为。用例扮演着车轮中枢的角色，轮辐



支持需求分析、基于场景的评估、测试以及与非功能或质量需求的整合。

- **场景**更关注于静止的、代表行动中用例的单一实例。场景既可以是轻快地叙事，也可以是像相互作用图解一样正式地建模。每个用例都应该包括至少一个主场景以及其他可能的备选场景。
- **需求**列出了非功能或质量需求，以及它们源自的用例。它也列出了源自用例的功能需求。在此阶段也可明确地将一些用例视为非需求。

### 3 用户故事

#### 3.1 维护社会联系信息

我们许多人都拥有多个电子邮箱账户，包括了与我们联系的个人和组织的信息——姓名、电子邮箱地址、电话号码、及时信息身份等。当某人的电子邮箱地址或电话号码发生了变化（或者他们获取了一个新的），如果我们可以将该信息在一处更新且其所有的副本都可以自动被更新，那么我们的生活将会更加的简单。换句话说，那些副本将被关联到一些“联系”的定义。也许有好的理由（像离线邮件处理）来保存一个联系的本地副本，但是理想情况下任何副本将仍被关联到一些中央“master”。

然而，就“联系”格式达成一致还不够。即便我们所有的电子邮件提供商对一个联系的格式达成了一致，我们仍然需要使用每个供应商的自定义接口来更新或替换供应者的副本，或者我们将不得不同意一种方法使得每个电子邮件提供商关联到“master”。如果我们抛开个人利益向外看，如果个人或组织能够公开其联系信息，以便我们能与其关联，会更加有用。

无论哪种情况，起作用的是对资源的共识、需要的一些格式以及这些资源的访问指导。这将支持如何获取一个联系的链接，以及如何使用那些链接来进行联系（包括读取、更新和删除它），还有如何简单地创建一个新联系、将其添加入我的联系中，以及当删除一个联系时，它将如何从我的联系列表中移除。能够添加一些在初始设计时没有考虑到的、有关我的联系的专用数据也将是很好的。理想情况下我们想消除联系的多个副本，有关我的联系的附加有效信息可能会存储在单独的服务器上，需要一种简单方法将此信息链接回联系。无论是否一个联系集合是我自己的、被组织共享的或者电子邮件供应商（或一个电子邮件供应商中的一个电子邮件账户）已知所有联系，如果它们都以几乎同样的方式工作那该多好。

#### 3.2 跟踪个人和商业关系

在我们的日常生活中，我们处理与不同组织的不同关系，他们也都有关于我们的数据。然而，任何一个组织都不可能拥有关于我们的所有信息。他们通常让我们访问信息（至少其中一些），许多通过网站，那里我们被一些字符串唯一标识——账号、用户名等。我们必须使用它们的应用程序来与跟我们有关的数据进行交互，然而，我们必须使用它们给我们的标识符。如果我们想要构建任何自己的整体画面（更准确地讲，收集所有其具体化的关于我们的数据），我们作为人类必须使用其自定义应用程序来发现数据、复制它并按我们的需要来组织它。

如果至少数据的网络寻址部分可被一致性地关联,因此不是维护不同格式的各种标识符,也不是必须为每个相应的自定义程序人工提供标识符,我们可为其从根本上建立一套书签,不是更简单吗?当我们想要检查或改变其内容,有个它们都支持的单一一致性应用接口,不是更简单吗?当然是这样的。

我们的关联集可能会是一个简单集合。任何单一组织所持有的信息可能会是简单数据与其他数据集合的混合,例如,一个银行账户余额以及一个历史交易集合。我们银行可很容易地拥有一个其客户集合的每个成员的账户集合。

### 3.3 系统和软件开发工具集成

系统和软件开发工具通常来自不同类型的供应商,建立在不同的架构和技术之上。这些工具都是为了满足特定域场景(建模、设计、需求等)而特制的。通常工具厂商视与其他工具的整合为不可避免的,而不是为其终端用户提供附加值。事后更多考虑的是这些工具的数据——如人物、项目、客户报告的问题和需要——如何与企业 and 外部应用(管理诸如客户、业务优先级和市场趋势的数据)进行整合和关联。此问题可通过标准化一小套工具或来自单一供应商的一套工具被独立出来,但这极少发生,如果发生也是仅在小型机构中。由于这些组织机构在规模和复杂性上都在增长,因此它们需要协同外包发展以及拥有自己的一套工具和流程的不同内部其他组织。需要对通过多工具跨角色、任务和数据处理的更完整的业务流程(系统和软件开发流程)提供更好的支持。这里有几个:

- 为每个应用程序设置一个 API,然后,在每个应用程序中,实施“粘合代码”,利用其它应用程序的 API 把它们集成起来。
- 设计一个存储多应用程序数据的单一数据库,对该数据库实现每个应用程序。在软件开发工具业务中,这些数据库通常被称为“仓储”。
- 实现一个中央“集线器”或“总线”,利用前面描述的 APIs 协调整体业务流程。

可以公平地说,虽然每种方法都有其追随者并可取得某些成功,但它们中没一个是完全令人满意的。关联数据作为一种应用集成技术而使用拥有强大的吸引力<sup>OSLC</sup>。

### 3.4 图书馆关联数据

W3C 图书馆关联数据工作组在其用例报告中有许多用例。这些参考用例聚焦于提取和关联不同来源的图书馆数据的必要性。这些可提供一致的格式以及改进或更新数据的方法的用例变体,将使得有效地共享此数据以及不需要重复全部提取和导入数据就能产生增量更新的方法变得更简易。

“数字对象簇”包括了许多相关用例:

- 分组:这应该“允许终端用户在由于某些原因整合在一起的网络上定义资源组。资源间存在的关系通常是未指明的。组群中一些资源也许不被定义了该组群的机构所控制。”
- 富集:“使得终端用户能够将资源关联起来。”

- 浏览：“支持终端用户浏览所属的组群及资源。”
- 重用：“用户应该具备利用其全部或部分元数据重用全部或部分集合的能力，在关联网络的其他部分。”

“集合”簇也包括了许多相关用例：

- 集合级描述：“作为一个整体为集合提供相关附属元数据，相比于数据项级描述。”
- 集合发现：“启用创新集合发现，如识别一个特定信息源被发现的或移动设备应用的物理集合的最近位置……基于集合级描述。”
- 社区信息服务：社区特定集合的识别与分类。

### 3.5 直辖市运行监测

跨越不同的城市、城镇、国家以及不同的直辖市，有越来越多的服务被产生和消耗大量信息的直辖市管理和运行。该信息被用于帮助监测服务、预测问题及处理物流。为了有效和高效地收集、生产和分析这些所有数据，需要一个松耦合标准数据源基本集。需要一种从不同的监测服务组中公开数据的简单、低成本的方法，可被容易地集成到检查和分析数据的直辖市其他系统中。所有这些服务都关联到且依赖于其他数据和服务，因此拥有一个简单且可伸缩的关联模型是关键。

### 3.6 医疗保健

医生给患者分析、诊断以及提出治疗方案，需要大量复杂、变化且不断增长的知识。该知识需要来自大量资源，包括医生自身的学科知识、咨询其他专业医学团队、公共健康资源、食品与药品监管机构以及其他医学研究和建议知识库。

诊断患者的病情需要患者的药物当前数据以及病历。此外，这些药物的近期医药报告与患者的数据关联到了一起。如果患者有药物过敏现象，这些医生需要将此信息发布到一个适当监管源中。其他医学专业人员需要访问药物的已被证实的和新出现的效果。类似的，如果新疫情具备地理分布模式，那么该信息需要迅速达到一个分布式的和多样化的医学信息系统。同样，新疫情信息反馈给这些管理机构，包括症状额外的细节以及病因，对于产生对未来事件最有效的治疗是关键性的。

### 3.7 广播中的元数据富化

当广播者对元数据富化显示出兴趣时，有许多不同的用例：

- 通过链接事实、事件、地点和人物来丰富档案或新闻元数据
- 丰富由自动抽取工具产生的元数据，如身份识别等
- 丰富分类表或枚举列表中术语的定义

这旨在更有效地信息管理和数据/内容挖掘（如果你找不到你的内容，就像你不曾拥有它，要么重建要么重新获取它，这样做很没有经济效益）。

然而，非常需要促进与其他数据源关联以及解决发现、自动化操作和消除歧义等问题的对策。广播者将要面对的其他重要问题是关联数据的编辑质量、其持久性和使用权。

### 3.8 基础设施数据的聚合与混搭

对于基础设施管理（如存储系统、虚拟机环境以及类似 IaaS 和 PaaS 概念），提供一个不同来源信息被有效地聚合、过滤和可视化的环境是非常重要的。特别地，需要考虑以下用例：

- 虽然一些数据源是基于关联数据的，然而其他则不是，且聚合和混搭必须跨越这些不同来源交叉工作。
- 数据源和整合/过滤数据流的消费者不必实现关联数据本身，它们也许是现成的组件，如构成可视化的仪表板框架。
- 此场景的简单版本是拉式基础的，从数据源请求数据。在更高级的设置中，在架构上不发生大的改变的情况下，应该转向推送式交互模型，数据源推送通知给用户，数据源提供用户可订阅的不同的服务（如“信息消息”或“仅重要提醒”）。

在该场景中，重要因素是具有允许容易聚合和过滤的抽象，独立于被组合的源内部数据模型，可被用于拉式交互以及推送式交互。

### 3.9 低端设备间 RDF 数据负载的共享

围绕语义网降维思想的许多项目需要能够跨给定网络节点成员进行 RDF 负载传输。该传输是在一个受限环境下完成的，依据带宽、节点所采用的本地语义作用域以及节点计算能力。在 P2P 模式下，每个节点既能作为数据消费者又可作为数据提供者，为自己提供数据或作为其他数据传输中介（通常在网状网络中）。

RDF 数据的任意负载的转移可通过容器机制来实现，对其添加或移除 RDF 三元组集。目前“SemanticX0”项目使用了命名图和图存储协议来跨节点创建/删除/复制图，但这（几乎）依仗三元组存储的使用。不幸的是，三元组存储是相当苛刻的软件，并不总是在有限硬件上起作用。一些通用的轻型列存储 REST-like 交互备份会是更好的方法。

### 3.10 二进制资源与元数据的共享

如果关联数据包含以非 RDF 格式表达的资源信息（它们是二进制的，如图片、视频或 XML 格式的人类可读文档），那么这些非 RDF 格式应该像将资源关联起来的 RDF 关系一样简单地发布到关联数据服务器中。关联数据服务器因此应该也允许非关联数据源的发布，且使其容易地发布和编辑那些资源的元数据。

资源有两部分——图像和图像信息（也许在图像文件中，但为其通用性，最好是置于图像之外）。图像信息是至关重要的。它是图像数据及其他数据的复合项（图像的应用元数据）。

### 3.11 数据目录

资产描述元数据模式 (ADMS) 提供了描述语义资产存储库内容的数据模型，但是当建立这些存储库联盟来满足资产重用需要的时候，这留下了许多开放性的挑战。这些包括访问和查询单个存储库以及不必检索所有内容就可以有效地检索更新的内容。因此，我们选择有效利用数据仓库集成方法来构建融合方案。这使得我们能够应对源技术的异质性并受益于它所提

供的优化性能，鉴于单个存储库不会经常发生变化。利用数据仓库，联盟需要：

- 理解数据，如理解它们的语义描述以及其他系统。
- 从不同存储库无缝交换语义资产元数据
- 保持自我更新

存储库所有者可以以关联数据兼容方式来保持其存储库描述的 de-referenceable URIs 及所含资产。ADMS 为数据的有意义交换提供了必要的模型。然而，这留下了没有完全解决的数据有效访问的挑战。

### 3.12 受限的设备和网络

来自物联网(WoT)中资源受限设备的信息已被视为许多领域的驱动力，从智能城市到环境监测到实时追踪。这些设备产生的信息数量成指数增长，需要以一种系统的、标准的和划算的方法来被访问和整合。通过使用网络上相同的标准，与应用程序进行整合将是资源受限设备之间简化的以及高层次的交互，抽象掉不均匀性，将成为可能。即将到来的IoT/WoT标准，如6LowPAN——资源受限设备IPv6——以及受限应用协议(CoAP)，为受限设备的使用提供一个UDP上的HTTP减缩版，已处在一个成熟的阶段。现在下一步是支持也在资源受限设备上的RESTful接口，秉承关联数据原则。由于有限的可用资源，无论是在设备还是网络（例如带宽、内存）一个基于SPARQL更新的解决方案目前看来不认为有用的和/或可行的。一种基于[HTTP-CoAP Mapping](#)的方法将使得受限设备能够直接参与基于关联数据的环境中。

### 3.13 支持科学进程的服务

许多科学领域如今都包括了网络数据密集型方法的分支，如生物信息学、天文学。为支持这些新方法，我们指望超越由科学工作流系统所提供的已建立的平台，从实验记录中获取、协助和保存完整生命周期，通过本地信任共享、分析、传播（包括实验数据的出版“超越PDF”）与重用。

- 聚合，特别是服务之间交换的研究对象(ROs)以及汇集了工作流、数据集、注释和出处的客户端。为此我们使用了一个 RDF 模型。虽然一些聚合内容使用了 RDF 进行编码且关联数据源数量不断增加，然而其他则不是；虽然一些被本地存储在 RO 之内，其他则是远程的（在这两种情况下，通常是缘于资源的大小或访问政策）。
- 分布和关联的服务。一些也许是集中化的（如出版物），其他的也许是本地化的（如每个实验室）。我们需要可简单地和容易地整合入用于科学的各种软件和数据的小型服务。
  - 收集和公开用于存储、修正、探索和重用的 ROs 的基础服务。
  - 为 ROs 提供附加值的服务，如无缝导入/导出科学工作流系统、自动稳定性评价或推荐（且因此与检索/存储/修正/ROs 的基础服务相互作用）。

### 3.14 项目成员信息：信息演化

人物与项目的信息随着角色的变化、组织的变化以及联系细节的变化而发生改变。发现

项目的当前状态对于使得人们联系到合适角色中的合适的人来说是重要的。它也可用于回顾，看一看过去什么人扮演了什么样的角色。

关联数据平台的一个作用就是把管理此类信息交给项目团队自己，而不是从中央网站管理员那里获取更新。

其实现可依靠：

- 每个人物和项目的资源描述
- 描述项目中角色/成员关系的容器资源

为了保留项目历史，资源的旧版本，包括容器资源，应该被保留。因此有必要既处理特定项目又要有一个“当前”概念。

访问信息包括两方面：

- 访问“当前”状态，不考虑资源描述的版本
- 访问历史状态，通过访问资源描述的特定版本

### 3.15 云基础架构管理套件

云计算为给用户提供了云基础架构(IaaS)管理的远程访问提供了 API 支持。基础架构包括系统、计算机、网络、磁盘等。整体结构可看作主要分层的，(云包括系统，系统包括机器等)，辅以交叉链接(如多台机器连接至网络)。

IaaS 场景在生命周期管理与发现、处理非瞬时变化、历史捕捉与查询方面提出了特定的需求。

- 云基础架构的许多方面与生命周期是相关联的，如计算机可在运行和关闭之间转换。这通过 API 应是可管理的，应包括客户端如何发现动态生命周期选项且从而通过应用程序来帮助操纵。
- 常有的情况是，到达一个新的生命周期状态并不是瞬时的。客户端需要监测这类变化的通用机制。
- 检索资源生命周期中所有事件的工具是有用的。
- 查询提供了 API 后询问资源以及寻找应用程序新切入点的方法。

基础设施管理可被视为资源底图的操作处理。

## 4 用例

以下用例都是源于上述的一个或多个用户故事。这些用例在不同的场景中被进行了详细探究，每一个都由单一用户故事例证的一些关键方面所驱动。它们所含的例子仅仅是为了起到说明的作用，而不应该被规范地理解。

### 4.1 用例：管理容器

许多用户故事引入将容器作为创建和管理应用环境中资源的一种机制的思想。被分组在同一容器中的资源通常是属于相同的应用。容器由 URI 进行标识。容器的属性也可代表该容器的功能可见性，使得客户端能够决定他们能对该容器可进行什么其他操作。这些操作可包

括特定应用服务（可通过交换 RDF 文档引起）的描述。

- 提供“提供……资源的访问指导”（功能可见性）（源自用户故事，[Maintaining Social Contact Information](#)）。

#### 4.1.1 主要场景：创建容器

在LDP服务器中创建一个新的容器资源。在[Services supporting the process of science](#)中，[Research Objects](#)是将科学调查中数据、方法与人物集合在一起的资源的语义丰富聚合体。一个基本 workflow 研究对象将被创建用以整合[scientific workflows](#)及其产物。研究对象以空容器的形式开始，在项目整个生命周期中将不断有 workflow、数据集、结果和其他数据加入其中。

##### 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 备选场景：创建一个嵌套容器

嵌套容器的动机源自[System and Software Development Tool Integration](#)用户故事。OSLC变更管理词汇允许故障报告有被成员谓语句`oslc_cm:attachment`引用的附件。“顶级容器”包含问题，且每一个问题资源都拥有其自己的附件资源容器。

##### EXAMPLE 2

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

## 4.2 用例：管理资源

该用例讨论了资源的管理生命周期，且关注于资源所有权。管理资源的责任与其容器有关。例如，一个容器可从客户端接受一个产生新资源的请求。该用例聚焦于容器环境下的资源创建与删除，以及通过在容器间移动资源来实现所有权转移的可能性。资源的所有权应当一直保持明确；以此方式管理的资源都不应该被多个容器所拥有。

一旦一个新资源被创建，它应该被一个 URI 所标识。客户端可将建立可引用 URIs 的职

责交给其数据的容器。容器是此接口终端的一项自然选择，缘于其将已有关于所含资源的一些特定应用知识。虽然服务器拥有资源命名的终极控制，一些应用可请求获得更多的命名控制权，也许能提供一个更加人工可读的 URI。LDP 服务器可支持 Atom Publishing Protocol slug header 等东西，来传达用户定义的命名“线索”。

- 资源的非重复：“消除多副本”，表示在单一位置的资源（源自[#Maintaining Social Contact Information](#)）。
- 资源的分布：关联数据“可存储在独立的服务器中”（源自[#Maintaining Social Contact Information](#)）。
- 一致、全局的命名：资源应被“一致地关联，……而不是以不同的格式保持不同的标识符”（源自[#Keeping Track of Personal and Business Relationships](#)）。

#### 4.2.1 主要场景：创建资源

资源以在容器中被创建而开始其生命期。源自用户故事，[Maintaining Social Contact Information](#)，应该可“容易地创建一个新联系且将其添加入我的联系中”。这暗示了资源创建与应用环境紧密相连。新资源在一个表示“我的联系”的容器中被创建。资源的生命周期与其容器的生命周期相关联。因此，例如，如果“我的联系”被删除了，那么用户可合理地认为其中的所有联系也都被删除了。

联系的细节被 RDF 描述及其属性来获取，包括“名字、邮件地址、电话号码、即时信息身份等”。该描述可包括非标准 RDF；“初始设计不考虑的我的联系的相关数据”。下列 RDF 可被用于描述使用 FOAF 词汇的联系信息。联系在这里被定义了一种可被作为单一单元来创建和更新的资源的 foaf:PersonalProfileDocument 所代表，即使它可描述附属资源，如下面的 foaf:Person。

##### 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 备选场景：删除资源

删除资源及其所有属性。如果资源归属于一个容器中，它将被从该容器中删除，然而此被删除资源的其他关联可作为虚引用而保留。在资源是容器的情况下，服务器也可删除任何或全部所含的资源。在通用实践中，已删除的资源不能被复原。然而也有边缘情况，需要有



限的复原。最佳实践表明,“Cool URIs 不发生改变”,意味着已删除的 URIs 不应该再生。

#### 4.2.3 备选场景: 移动所含资源

许多资源也许拥有超出其容器中成员资格生命期之外的值。这暗示了添加引用参考来修订容器成员资格的方法。在其他容器中克隆容器成员以供使用,导致了信息的重复以及维护问题;网络实践旨在鼓励一种资源的创建,该资源可根据需要被在许多地方进行引用。所有权的变更也许——或也许不——意味着 URI 的改变,取决于特定服务器命名政策。虽然为资源分配一个新 URI 是不允许的,但是可以指示资源已随着适当的 HTTP 响应而发生移动。

#### 4.3 用例: 检索资源描述

访问当前的资源描述,包含该资源的属性以及相关资源的链接。此表示可包括不能被直接访问的相关资源的描述。依靠该应用,服务器可以附加三元组来富化检索到的 RDF。例子包括了增加链接、sameAs 闭包和类型闭包。HTTP 响应应该也包括版本信息(如最后更新或实体标签),以便于后续更新可确保其被应用于正确的版本。

- 合适地使用标准词汇以获得“资源的共识”(源自[Maintaining Social Contact Information](#))。
- “可伸缩关联模型是关键”(源自[#Municipality Operational Monitoring](#))。

##### 4.3.1 主要场景

用户故事[Project Membership Information](#)讨论了人物和项目信息的表示。它呼吁“为每一个人物和项目进行资源描述”使得项目团队能检查这些资源所持有的信息。下例说明了基于[Epimorphics organizational ontology](#)的组织结构所持有的信息的种类。

注意下例定义了 LDP 服务器所主持的 2 种资源。这些资源的表示可包括相关资源的描述,如 <http://www.w3.org/>, 但其受到不同权力的影响,因此在此处不能从 LDP 服务器中获取。

##### 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 备选场景：检索一个非文档资源的描述

在许多情况下，所感兴趣的事情并不总是可解析的事情。下例展示了 FOAF profile 可如何被用于区分人物和 profile；前者是后者的主题。这引起了客户端应怎样处理此类非文档资源的问题。在本例中，HTTP 协议要求在从服务器请求 URI 之前去除离散部分。其结果是一个 profile 的可解析 URI。

#### 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> .
```

### 4.4 用例：更新现有资源

变更 LDP 资源的 RDF 描述，潜在地移除或重写现有数据。这允许应用程序通过添加与其他资源的附加链接来富化资源的表示。

- 非限制词汇：应该可“能够添加……特定应用数据”至资源（源自[#Maintaining Social Contact Information](#)）。

#### 4.4.1 主要场景：富化

这与用户故事[Metadata Enrichment in Broadcasting](#)相关，并基于[BBC Sports Ontology](#)。关联数据的资源中心视角为替换或重写资源及其数据提供了一个自然粒度。最简单的更新类别是用一种新的表示来简单替代已知现有资源。下述例子中有两个独特资源；体育赛事和相关奖项。资源的粒度将允许用户在不破坏事件信息的情况下更换奖项信息。

#### 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 .
```

我们可以对事件进行延伸描述，通过下述内容替代上述描述来关联至金牌得主。

#### 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" .
    ] .
```

#### 4.4.2 备选场景：资源的选择性更新

这与用户故事[Data Catalogs](#)相关，并基于[Data Catalog Vocabulary](#)。目录由下述RDF模型进行描述。

##### 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.
```

一个目录可包含多数据集，因此当关联至新数据集时，更简单也更合适的是仅选择性添加新的数据集链接。将一个的dc:title添加至数据集时可使用[Talis changeset](#)。以下更新是为目录定向添加一个附加数据集。

##### 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#>.

<changeset>
  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 .
  ] .
```

#### 4.5 用例：确定资源是否已发生变化

应该可检索资源（如最后更新或实体标签）的版本信息，但不需要下载资源的表示。此信息可与该资源之前持有的信息进行比较，以决定其是否发生了改变。此版本信息也可被用于后续的条件请求中，以确保它们仅在版本未发生变化时被应用。

##### 4.5.1 主要场景

基于用户故事，[Constrained Devices and Networks](#)，一个 LDP服务器可被配置成基于[Web of Things](#)的CoAP代理。作为CoAP资源的观察者，LDP服务器将其兴趣点登记注册，以便于当传感器发现变化时能及时被通知到。LDP的客户端可询问服务器来判断状态是否已发生改变。

在此例中，有关传感器和相应传感器读数的信息可被表示为RDF资源。下述第一个资源，表示以[Semantic Sensor Network](#)本体描述的传感器。

## 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> .
  ] .

```

随着测度的进行,传感器的值在实时变化。LDP 客户端可询问以下资源来判断其是否已发生改变,而不必非要下载 RDF 表示。由于不同的传感器属性被分散表示(独立的 RDF 表示),因此它们可独立地发生变化。

## EXAMPLE 12

```

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

<http://example.com/energy-management#property_1> :hasMeasurementPropertyValue ◇ .
◇ a :FrequencyValue;
  :hasQuantityValue "50"^^xsd:float.

```

## 4.6 用例:聚合资源

能够管理资源集合有个要求。集合的概念与容器的概念既有联系又有区别。这些集合是(弱)聚合体,与资源的生命周期管理无关,又与资源及其容器之间的所有权不同。然而,容器的复合物也许会映射为支持容器及其所含内容导航的集合。需要能够通过添加和删除成员属性来创建集合。资源可能属于多集合,或不属于任何集合。

- 资源描述是一个“简单数据与集合的混合体”(源自[#Keeping Track of Personal and Business Relationships](#))。
- 相对URIs: 应该可在不破坏内部链接的情况下为集合整体“运送RDF的有效负荷”(源自[Constrained Devices and Networks](#))。

## 4.6.1 主要场景:为集合添加资源

该例源自[Library Linked Data](#)和LLD-UC,尤其是[Subject Search](#)。

在<http://example.com/concept-scheme/subject-heading>中有一个现有集合,定义了主题集合。该集合被定义为 skos:ConceptScheme,客户端希望将一个新概念插入该结构中,将通过 skos:inScheme link 与集合相关。新的主题,“外太空探索”,不一定非要被容器所有。下述 RDF 将被加入到集合的(数据项级)描述。

## 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.

```

## 4.6.2 备选场景:为多集合添加资源

逻辑上,一个资源不应被多个容器所有。然而,它可以是定义了一种聚合体弱化版的多

集合的成员。由于这只是集合的 RDF 描述的一项操作，它应可添加相同的资源至多集中。

作为一个医学术语的机器可读集合。[SNOMED](#)本体在[healthcare](#)中起着重要作用。SNOMED CT允许带有多个父概念的概念不要落入晶格。在下例中，同样的概念也许会受到不同父概念的影响。此例使用了skos:narrowerTransitive来消除干扰概念。

#### 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 用例：过滤资源描述

该用例通过动态地排除特定（成员）属性，扩展了检索资源的 RDF 描述的常规行为。对于容器来说，经常需要能够读取一个排除了容器成员的集合或数据项级描述。

#### 4.7.1 主要场景：检索集合级描述

该场景是基于[Library Linked Data](#)，使用都柏林核心元数据计划[Collection-Level](#)描述。集合可指任何物理的或数字的条目的聚合体。本场景涵盖了客户端可如下例所示的请求一个集合级描述，而不需要下载集合中条目的全列表的情况。

#### 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 备选场景：检索集合的数据项级描述

该用例场景，也是基于[Library Linked Data](#)，聚焦于获取由集合聚合的资源的数据项级描述。最简单的场景是集合成员在单一表示内返回，以便客户端可以通过跟随这些链接来研究数据。不同的应用可使用不同的成员谓词来获取该聚合体。下例使用了rdfs:member，但是许多不同的成员谓词可通用，包括RDF列表。数据项级描述可通过使用书目记录功能需求([FRBR](#)) [ontology](#)来获取。

## 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 O 2 B" ;
  frbr:creator <#Vonnegut_Kurt>;
  frbr:manifestation <ebook21279>.
```

集合可能非常巨大，因此需要一些手段来限制由 LDP 服务器返回的 RDF 表示的规模（如页码）。

#### 4.8 用例：管理媒体资源

应当可容易地将非 RDF 媒体资源添加到接受他们的容器中。媒体资源可在容器生命周期内被更新和移动。

##### 4.8.1 主要场景：访问媒体资源

从用户故事[Sharing Binary Resources and Metadata](#)中，应当可容易地将非RDF媒体资源添加到接受他们的容器中。客户端以容器所接受的媒体类型将一个非RDF表示提交到该容器中。该容器创建一个URI来表示此媒体资源，并创建一个从容器到此新URI的链接。该媒体资源可拥有一个明确的媒体类型表示。应当可找到关于此资源的元数据，并可对其以通常方式进行访问和编辑。

此例使用了媒体资源本体来描述将媒体资源添加到集合中。

## EXAMPLE 17

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

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

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

##### 4.8.2 备选场景：媒体资源附件

一个资源可能有多种译文；就像你可以同时拥有代表同样事情的 PDF 和 JPEG。一个用户正在试图创建一种带有显示有误机件的附加图的工作指令。对于用户和工作指令系统来说，这两种产物被作为一个集合来进行管理。一个单一请求可创建工作指令、附件及其之间的关系。当用户稍后检索工作指令时，他们希望单一请求默认检索包括所有附件在内的工作指令。当用户更新工作指令（如将它标记为完成）时，他们只想要更新适当的工作指令，而不包括其附件。用户可在其生命周期内添加、删除、替换工作指令的附件。

## 5 需求

## 5.1 功能需求

1. 创建容器, 源自[Use Case: Manage containers](#)
2. 创建嵌套容器, 源自[Use Case: Manage containers](#)
3. 创建资源 (容器内部), 源自[Use Case: Manage containers](#)
4. 删除资源, 源自[Use Case: Manage containers](#)
5. 移动所含资源, 源自[Use Case: Manage containers](#)
6. 检索资源描述, 源自[Use Case: Retrieve resource description](#)
7. 检索非文档资源的描述, 源自[Use Case: Retrieve resource description](#)
8. 富化 (现有资源的取代性更新), 源自[Use Case: Update existing resource](#)
9. 资源的选择性更新, 源自[Use Case: Update existing resource](#)
10. 确定资源是否已发生变化, 源自[Use Case: Determine if a resource has changed](#)
11. 为集合添加资源, 源自[Use Case: Aggregate resources](#)
12. 为多集合添加资源, 源自[Use Case: Aggregate resources](#)
13. 检索集合级描述, 源自[Use Case: Filter resource description](#)
14. 检索集合的数据项级描述, 源自[Use Case: Filter resource description](#)
15. 访问媒体资源, 源自[Use Case: Manage media resources](#)
16. 媒体资源附件, 源自[Use Case: Manage media resources](#)

## 5.2 非功能需求

1. 提供资源访问指导, 源自[Use Case: Manage containers](#)
2. 资源非复制, 源自[Use Case: Manage containers](#)
3. 资源分布, 源自[Use Case: Manage containers](#)
4. 一致的、全局命名, 源自[Use Case: Manage containers](#)
5. 使用适当的标准词汇, 源自[Use Case: Retrieve resource description](#)
6. 可伸缩关联模型, 源自[Use Case: Retrieve resource description](#)
7. 无限制词汇, 源自[Use Case: Update existing resource](#)
8. 资源描述是“简单数据与集合的混合”, 源自[Use Case: Aggregate resources](#)
9. 能够共享集合的相关URIs, 源自[Use Case: Aggregate resources](#)

(编译自: Linked Data Platform Use Cases and Requirements.

<http://www.w3.org/TR/2013/WD-ldp-ucr-20130131/>. [2013-01-31])

(岳增慧编译, 吴贝贝校对)

## 二、IMS 学习工具互操作性 (LTI) 安全版本 2.0 公共草案

### 1 概述

本作品采用[知识共享署名-非商业性使用-禁止演绎 2.5 中国大陆许可协议](#)进行许可。

LTI 定义了工具使用者、工具提供者与终端用户之间的大量交互。每个交互都需要某种安全形式，但 LTI 仅提供一些路径。没有涉及到的交互是工具使用者或提供者的责任，如下所示。整个 LTI 规范的概述以及相关文档的完整列表，参见 LTI 实施指南[LTI, 12IMG]文件。

不同的交互总结如下表：

交互	覆盖范围
工具使用者 $\leftrightarrow$ 终端用户	工具使用者的责任
工具提供者 $\leftrightarrow$ 终端用户	工具提供者的责任
工具提供者 $\rightarrow$ 工具使用者网络服务	LTI 网络服务的安全部分
工具使用者 $\rightarrow$ 工具提供者直接 POST	LTI 消息安全（在将来使用——不是 LTI2.0 中）
工具使用者 $\rightarrow$ 通过终端用户浏览器的工具提供者 POST	LTI 消息安全
工具使用者 $\rightarrow$ 工具提供者网络服务	没有涉及——LTI 没有提供或需要任何工具提供者提供的网络服务，但是由于我们已经在工具配置文件中放入一个占位符来声明服务，所以注意到没有预先定义的安全很重要。

由于每个交互都有大量的可能选项，所以每个系统会通过它们各自的配置文件来定义它们所支持的安全配置文件。

在调度过程中，工具提供者会决定安全选项的最佳匹配集，并将其包含到工具代理服务的 POST 中。

### 1.1 文件结构

该文件的结构是：

2. 消息安全——LTI 消息以及它们如何被保护的描述
3. 网络服务安全——LTI 网络服务以及安全服务如何被规定的描述

## 2 消息安全

LTI 消息安全定义了工具使用者与工具提供者之间我们如何保护 POST 数据。这两者都应用到消息中，该消息在系统之间被直接发送以及通过终端用户的浏览器发送。

关于 POST 自身的细节，参阅 LTI 消息框架[LTI, 12MSF]。此文件说明了与组件相关的安全，该组件被应用到 post，尤其是在文件中被引用的 LTI 安全配置文件对象的使用与定义。

下表列出了已知的消息安全配置文件。

**Table 2.1 Known messaging security profiles.**

Profile Name
<code>lti_oauth_hash_message_security</code>



此配置文件依赖于在工具使用者与工具提供者之间的共享密钥。对于一个被注册的工具，该共享密钥被设置为一个系统级密钥，它在 TP 和 TC 之间建立一个广泛的信任关系。对于没被注册的工具，可能有一个系统级的密钥或者是被应用到每个单独的正在被启动的 basic\_lti\_link 的密钥。此部分没有描述哪个密钥与哪个消息相配。

## 2.1 lti\_oauth\_hash\_message\_security 配置文件

OAuth 安全是一个旨在保护 POST 和 GET 请求的安全机制。此部分仅适用于保护启动以及正被序列化和使用 POST 发送的其他消息。

OAuth 网站包含了 OAuth 1.0 规范和实现 OAuth 安全的示例源代码。OAuth 详细说明了如何构建一个基本的信息字符串，然后使用密钥来签名字符串。然后签名作为 POST 请求的一部分被发送，并被使用 OAuth 的生产者验证。

每个 OAuth 规范，在签名过程都会产生大量值，该值被添加到启动请求包括 oauth\_consumer\_key:

```
□□□  
  
  oauth_consumer_key=b289378-f88d-2929-ctools.umich.edu  
  oauth_signature_method=HMAC-SHA1  
  oauth_timestamp=1244834250  
  oauth_nonce=1244834250435893000  
  oauth_version=1.0  
  oauth_signature=Xddn2A%2BjzwjgBIVYkvigaKxCdccc%3D  
  oauth_callback=about:blank  
□
```

Figure 2.1 □ Launch request with oauth\_consumer\_key.

使用 OAuth 签名消息的重要值是 oauth\_consumer\_key 以及 oauth\_consumer\_secret。oauth\_consumer\_key 在消息中被传递为纯文本，并识别哪个使用者在发送允许生产者查找合适的验证密钥。oauth\_consumer\_secret 被用来签名消息。oauth\_callback 不在仅签名场景中使用，所以如果你的 OAuth 库需要它，你可以将它设置为任何值，例如“about:blank”。也要注意：Launch\_presentation\_return\_url 与 oauth\_callback 的作用不同。

由于我们在仅签名场景中使用 OAuth，所以不需要 oauth\_token。

工具提供者必须至少支持 HMAC-SHA1 签名方法将 OAuth 数据传递为 POST 数据。

为了尽可能多地支持 OAuth 客户库与方法，工具提供者被鼓励要尽可能多地支持 OAuth 签名类型，也要在消息头、查询参数形式的 URL 或 POST 参数中支持接收 OAuth 数据。

收到 POST 之后，工具提供者就要使用共享密钥来执行 OAuth 验证。它必须已经在本地存储了与 oauth\_consumer\_key 相联系的工具使用者的关系。时间戳也应该在特定的时间间隔被验证。时间间隔应该是由工具提供者定义的，但应该很小。它确实依赖于工具使用者与工具提供者同步的时间。

工具提供者应该记录接收到的特定场合，并仅允许使用一次任何特定场合。与时间戳相

结合, 他们仅使用此方法来记录与他们可接受的时间间隔相等的时间段。推荐的做法是有 90 分钟的时间间隔, 以便于你能够记录 90 分钟。

### 2.1.1 为 LTI 启动构建 `oauth_consumer_secret`

由于 LTI 支持两种不同和独特的密钥, 每次启动被选择的 `oauth_consumer_key` 都必须恰当, 依赖于 `oauth_consumer_secret` 的范围。当使用 LMS-wide 密钥时, `oauth_consumer_key` 必须与此工具设置的 `system-wide` 键相匹配。当一个 `basic_lti_link` 有一个资源级别的 `oauth_consumer_secret` 时, 就必须使用资源级别的 `oauth_consumer_key`。

## 3 网络服务安全

LTI 定义了一组网络服务, 该服务由工具使用者 (TC) 提供, 被工具提供者使用 (TP), 以及由 TP 提供, 被 TC 使用。网络服务需要得到保护以免受到未经授权的访问。此部分会详细说明与获取这些网络服务相关的安全性问题。

下表列出了已知的网络服务安全配置文件。

**Table 3.1 Known web service security profiles.**

Profile Name
<code>lti_oauth_body_hash_ws_security</code>

### 3.1 `lti_oauth_body_hash_ws_security` 网络服务安全配置文件

此配置文件适用于 LTI 1.1 “普通的 XML” (POX) 服务以及 LTI 2.0、更新的 JSON-LD (REST) 服务。

对于 POX 和 REST 两者来说, 使用 OAuth 签名的消息能确保消息的完整性, 并且建立呼叫系统的身份。https 的使用被推荐给所有的网络服务连接, 因为响应数据没有被签名, 因此避免中间人攻击的唯一方法是使用 ssl。

消息的主体是 XML 或 JSON, 它们遵循特定需求服务操作所需的模式, 并且消息使用 `oauth_consumer_key` 和 `oauth_consumer_secret` 来签名, 两者被用来执行特定用户/课程/资源的工具启动。在发送给工具代理终端的最初 POST 请求案例中, 在工具注册请求中从 TC 传递给 TP 的 `oauth_consumer_secret` 是暂时的。

使用 OAuth 来签名主体的步骤在网站上描述：  
[http://oauth.googlecode.com/svn/spec/ext/body\\_hash/1.0/oauth-bodyhash.html](http://oauth.googlecode.com/svn/spec/ext/body_hash/1.0/oauth-bodyhash.html)

LTI 进一步定义了 OAuth 参数是如何被使用的, 如下:

- `oauth_consumer_key`: 在工具配置过程中工具提供者获得的 “proxyToolGuid”
- `oauth_signature_method`: 此参数的值应该是 “HMAC-SHA1”。所有使用这些服务的信息必须不能使用 `x-www-form-encoded` 的内容形式, 这一点很重要。服务将合法拒绝任何带有此内容类型的请求, 按照 `oauth_body_hash` 签名规范, 它明令禁止 `oauth_body_hash` 与 `x-www-form-encoded` 数据在任何请求中的结合。

这些服务也强调, 所有的 OAuth 参数作为授权头标的一部分被发送。特别是来自请求

URL 和 POST 主体的 OAuth 参数不会被处理。

oauth\_body\_hash 通过使用主体内容的 SHA-1 散列值被计算,并添加到授权头标中。所有的 OAuth 参数、HTTP 方法和 URL 像其他任何 OAuth 签名请求一样来被签名。除了构建主体散列值,实际的 POST 数据并没有参与 oauth\_signature 的计算。

大多 OAuth 库可以产生和验证这些消息的签名,大多数的库已经在授权头标中支持发送 OAuth 参数。

下面显示的是签名请求的示例。授权头标的换行是为了读值更加容易。oauth\_signature 对于如下数据是无效的——它仅仅是一个示例签名。

```
POST http://www.example.com/ims/lis/Result/sourceidGUID HTTP/1.0
Host: 127.0.0.1:80
Content-Length: 757
Authorization: OAuth realm="",
  oauth_version="1.0",
  oauth_nonce="29f90c047a44b2ece73d00a09364d49b",
  oauth_timestamp="1313350943",
  oauth_consumer_key="lmsng.school.edu",
  oauth_body_hash="v%2BxFnmDSHV%2Fj29qhxLwkFILrtPo%3D",
  oauth_signature_method="HMAC-SHA1",
  oauth_signature="8auRpRdPY2KRXUrOyz3HKCs92y8%3D"
Content-type: application/ld+json
{
  "@context" : "http://www.imsglobal.org/impurl/lti/v1/ctx/Outcomes",
  "@id" : "urn:ims:lis/Result/5b485a90-de60-11e0-9572-0800200c9a66",
  "resultScore" : {
    "@type" : "decimal"
    "@value" : "0.83"
  }
}
```

Figure 3.1 A sample signed request.

注意: 此安全配置文件需要 TC 与 TP 的时钟同步。使用可配置的时间间隔可以调整稍微偏离的时钟,但是设置的间隔太大是不提倡的。

(编译自: IMS Learning Tools Interoperability (LTI) Security Version 2.0 Public Draft.

<http://www.imsglobal.org/lti/v2p0pd/ltiSECv2p0pd.html>. [2012-11-01])

(王妍编译, 岳增慧校对)