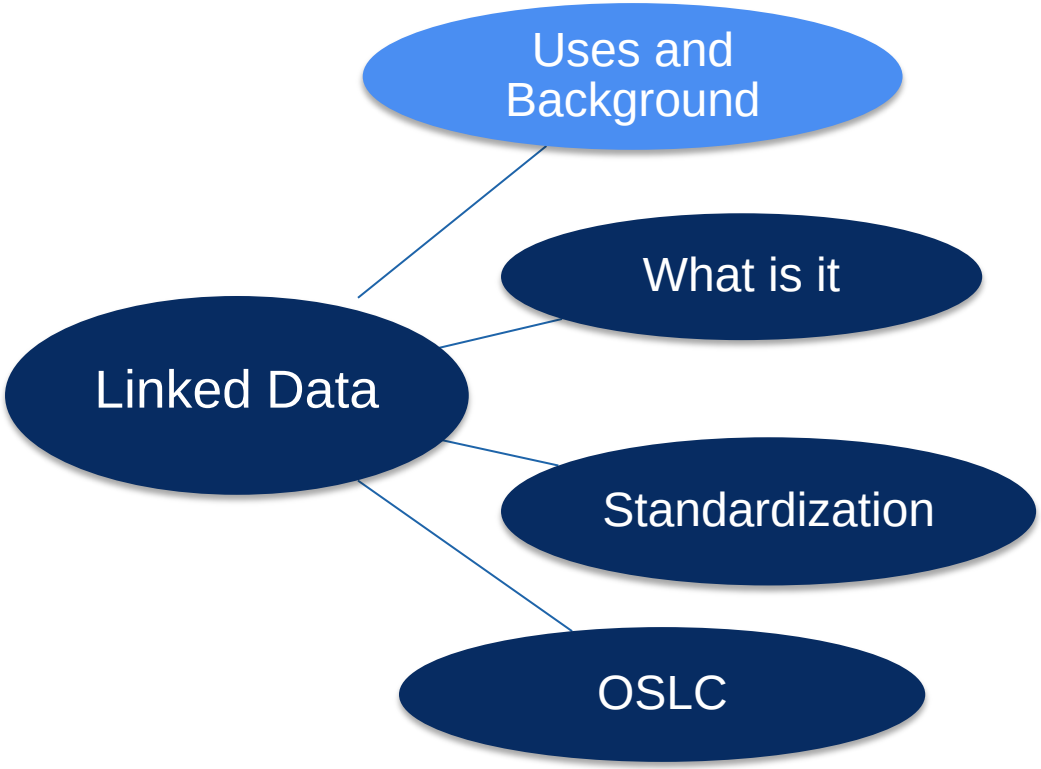
W3C Linked Data Platform and OSLC V3

Steve Speicher
Linked Lifecycle Data and OSLC
Chief Architect
IBM Rational

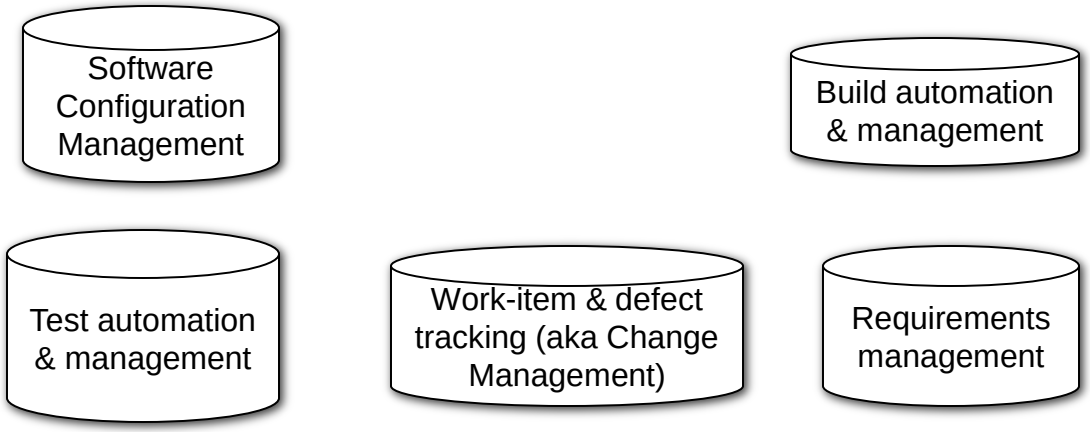
@sspeiche

sspeiche@gmail.com

Agenda



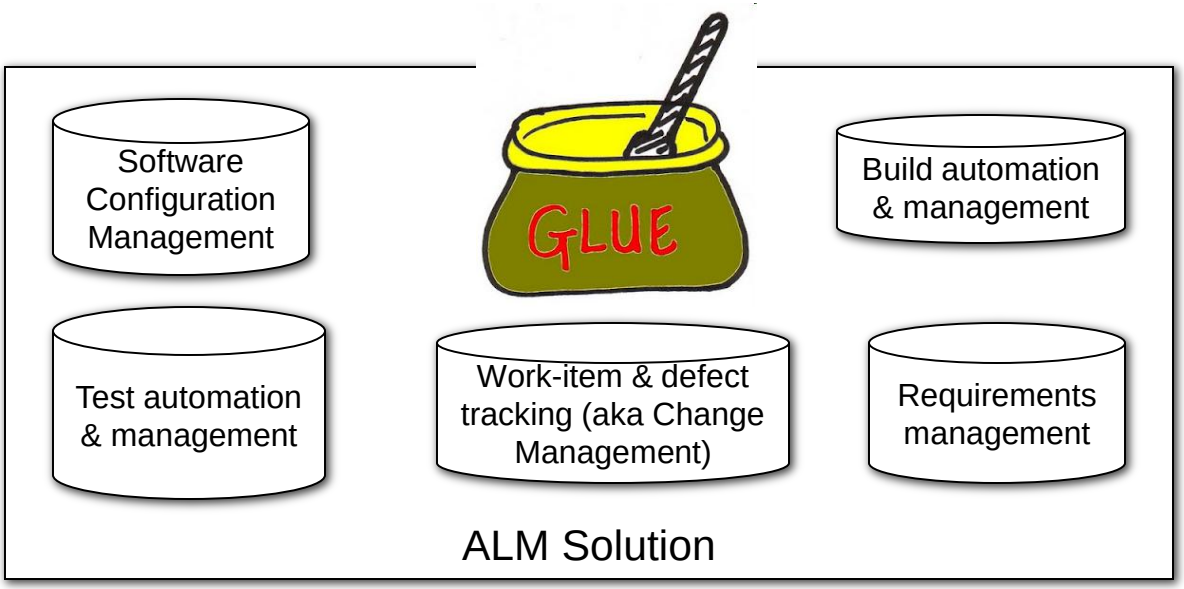
Software Development Tools : Year 2000 - point products



User complaints

- Need tools to work better
- Need tools to work **together**

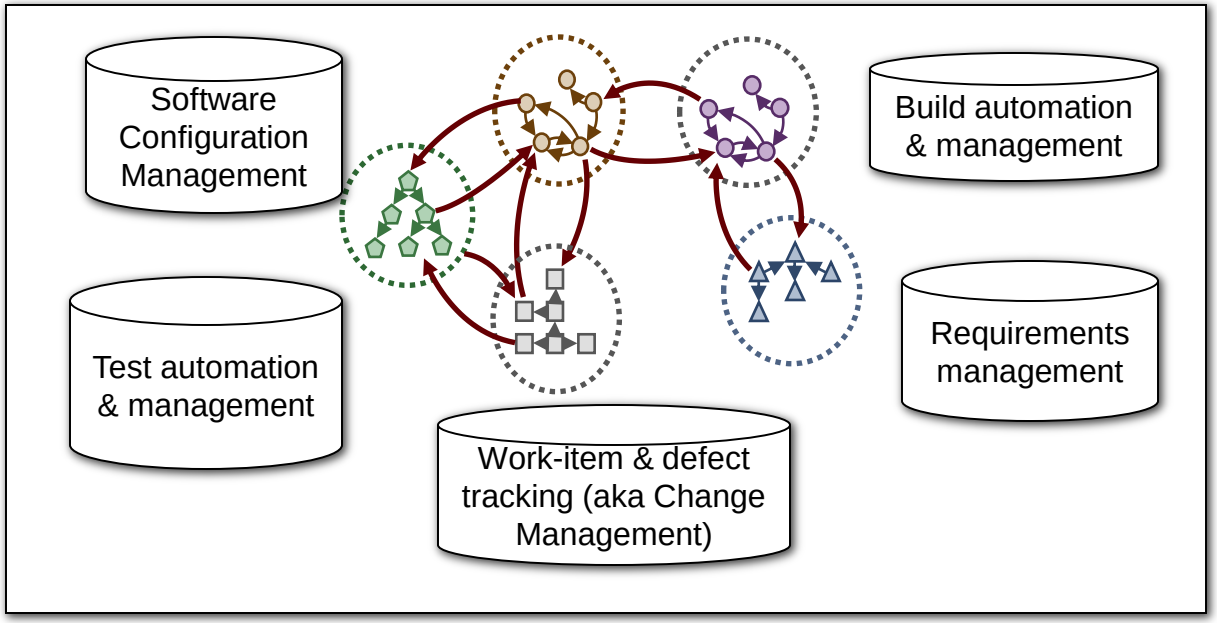
Year 2005 – ALM V1



- Developer complaints
  - Clumsy integrations
  - Want best-of-breed choice
- Management complaints
  - Need governance, metrics, reports, …

- Mixed Tool Environments
  - Open-source offerings with DIY integration
  - 3rd party ALM suites

Year 2010 – Linked Data and OSLC – A major breakthrough



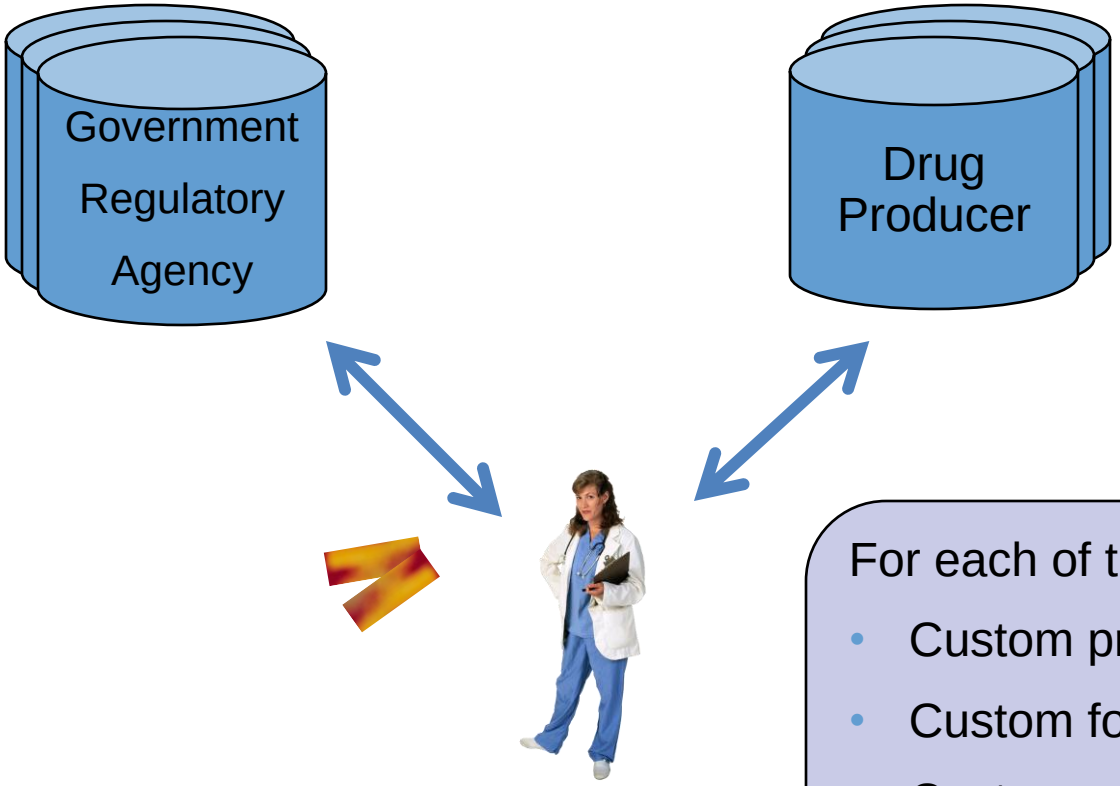
Learned to integrate with open protocols instead of glue

GET MORE DETAILS @ http://open-services.net

Healthcare – Reporting of drug trial and adverse effects

- Often requires custom data collection techniques at many office, clinic and mobile locations
- Need to report to drug producer as well as government regulatory agencies
- Valuable data as input in diagnosing a patient's condition

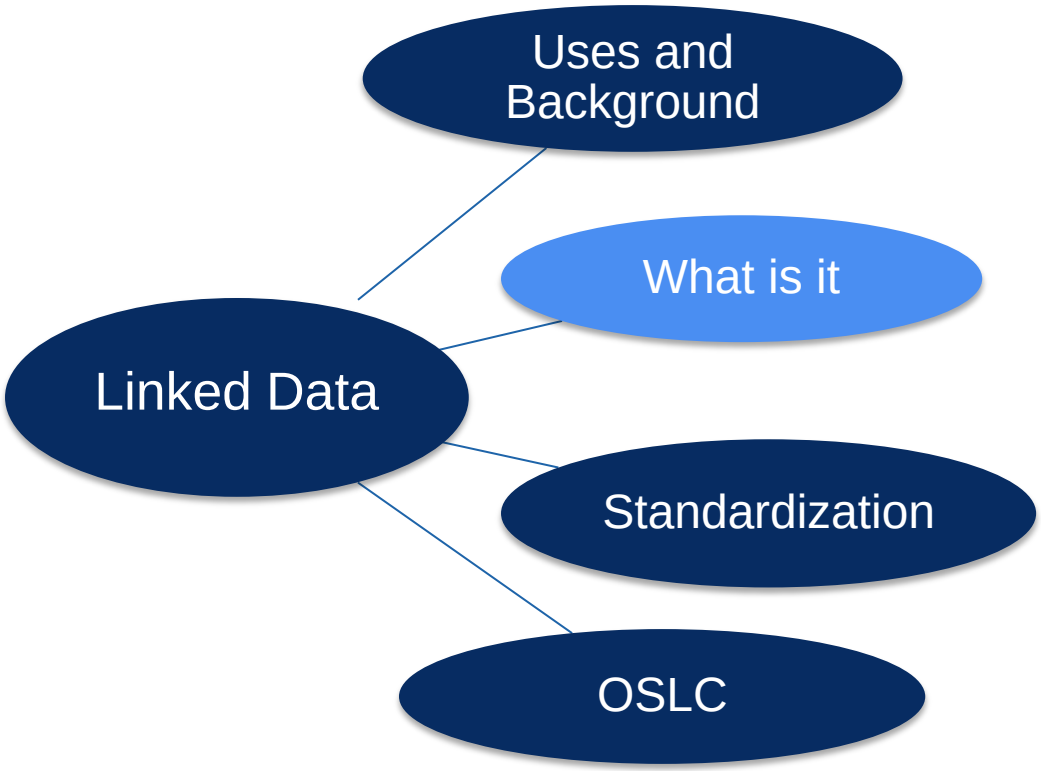
Healthcare – Adverse Event Reporting Landscape



For each of these 3, need:

- Custom protocol
- Custom format
- Custom schema
- Custom security

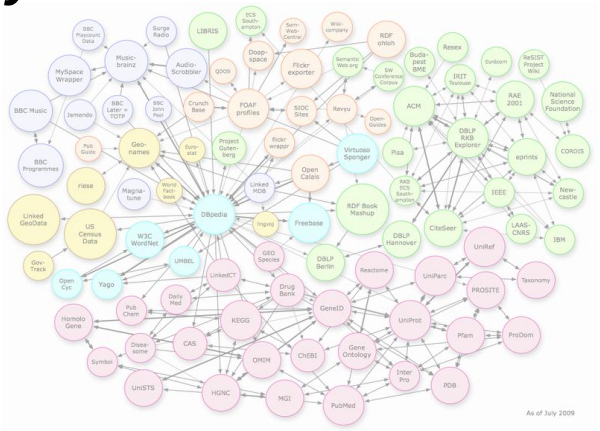
Agenda



Linked Data – Defined by Tim Berners-Lee

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.***

He concludes this with: "**Simple**."



Reference: "Linked Data", Tim Berners-Lee, 2006-07-27

Linked Data – What is it?

TestCase 14 is blocked by Issue 973

http://srv/qm/tc/14	qm:blockedBy	http://apache.org/jira/HTTPCLIENT-973

HTTP GET

Joe is a comm[illegible] for Apache

http://joecoder.me	d[illegible]mitter	http://apache.org

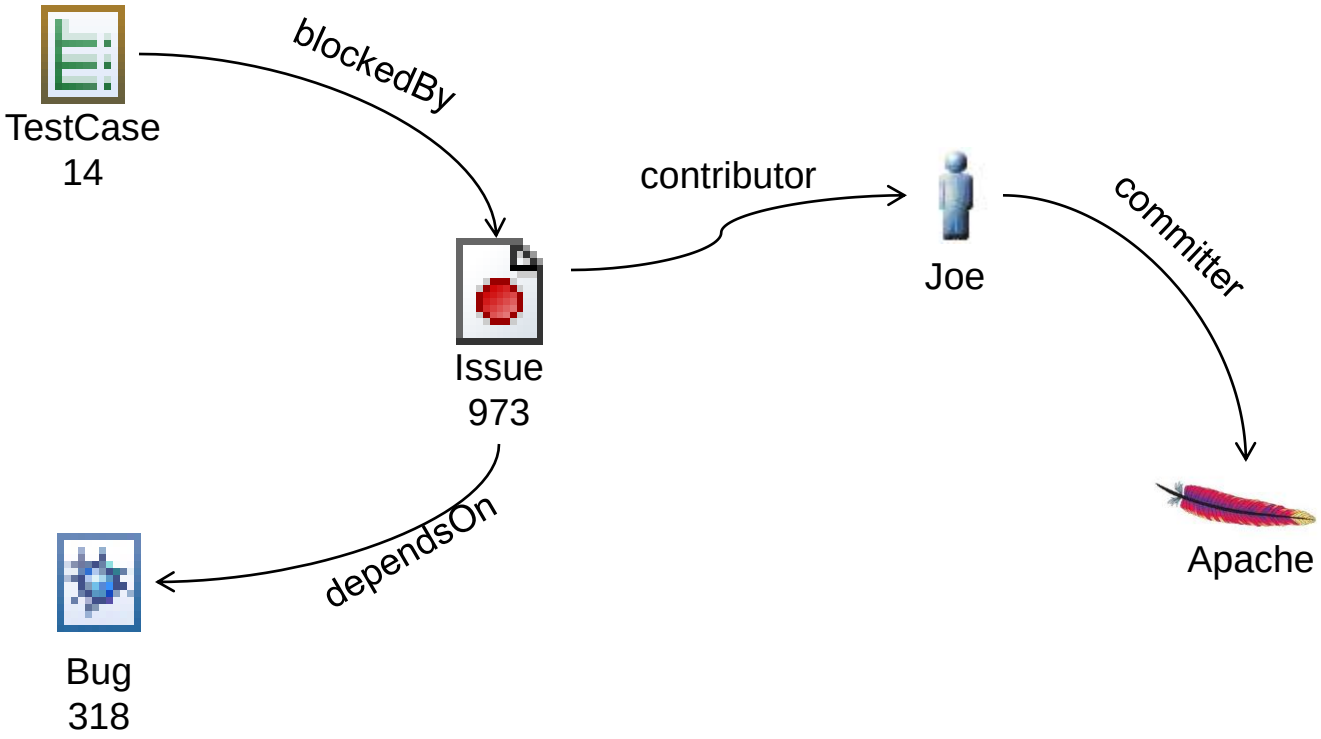
Issue 973 depends on Bug 318

http://apache.org/jira/HTTPCLIENT-973	cm:dependsOn	http://bugs.sun.com/? id=7172318

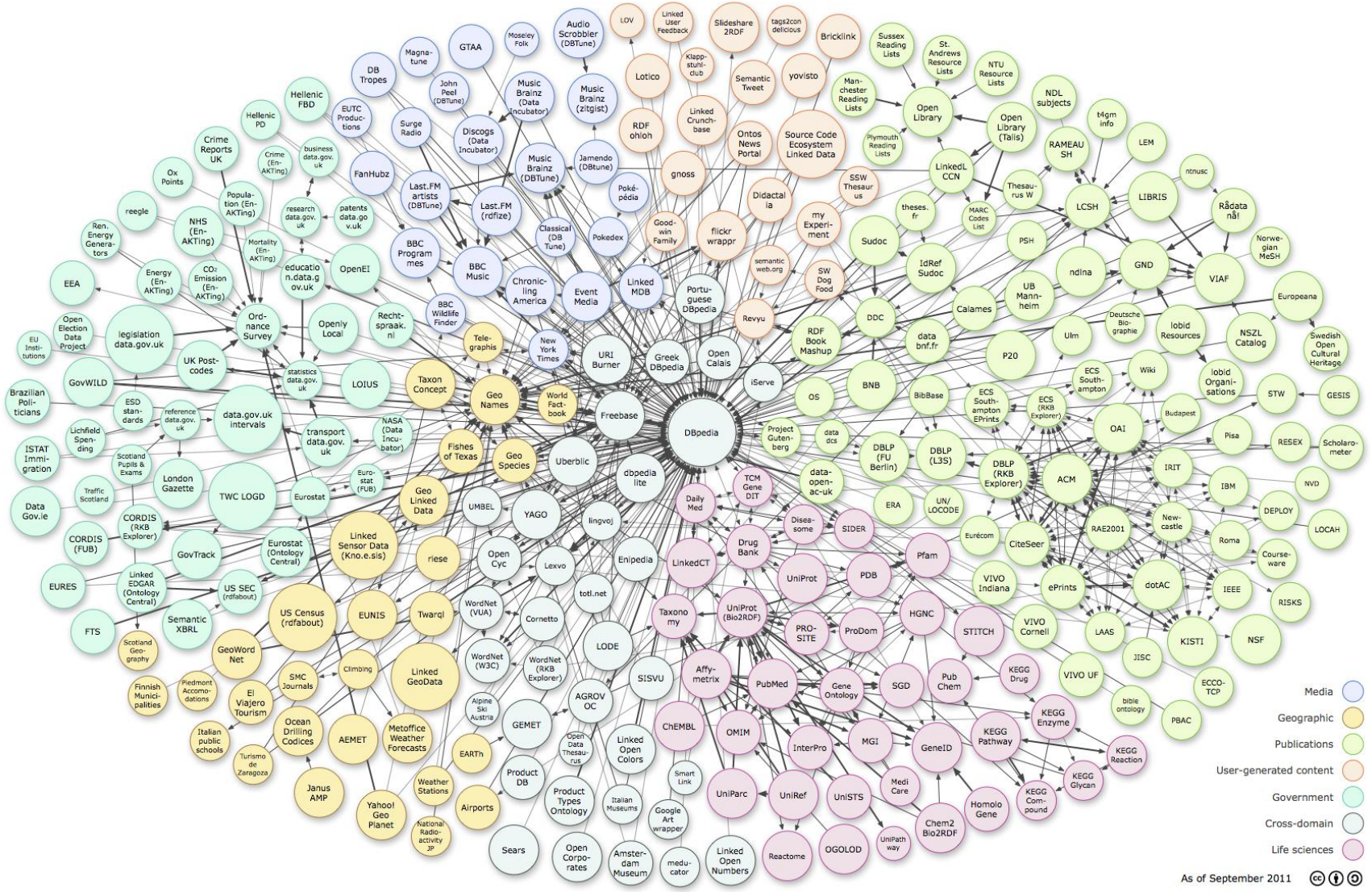
Issue 973 is owned by Joe

http://apache.org/jira/HTTPCLIENT-973	dc:contributor	http://joecoder.me

Linked Data – What is it? Show me a picture

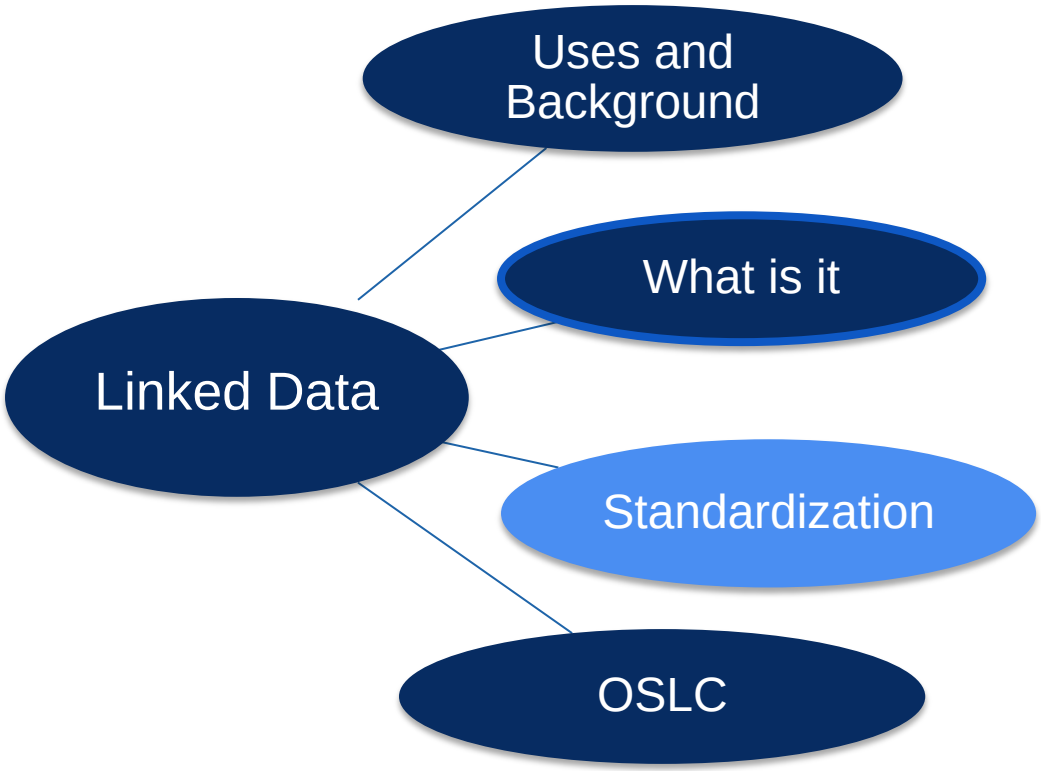


Linked Data – Bridging separate data sources (but with meaning)



Source: http://lod-cloud.net Sept 2011

Agenda



Linked Data – state of the art

- Typically focused on exposing data with relationships
- No support for update and creation of granular data
- Data updated, created and deleted by receiving new dataset dumps
- Often an export transformation and dump of data
- Many different publications on anti-patterns and best practices
  - They don't all align
  - Often different based on specialized usage

- Provides great value for many use cases as shown above
- Interoperable solutions are limited due to no agreement on anti-patterns and best practices
- Leads to most applications support least common denominator or hacks for different situations

Evolution of W3C Linked Data

- W3C Linked Data
  - Four design principles proposed by Tim Berners-Lee in a 2006 article
  - Standardization will accelerate industry adoption and enhance interop

- IBM Sponsored W3C Workshop on Linked Enterprise Data Patterns (Dec 2011)
  - Participating organizations included EMC, Elsevier, IBM, Nokia, Oracle
  - Workshop recommended W3C produce a standard “which defines a Linked Data Platform”, leveraging RDF
- IBM Submitted Linked Data Basic Profile 1.0 proposal to W3C (March 2012)
  - Base the Linked Data Platform foundation on OSLC Core technology and experience
  - Co-submitters: DERI, EMC², ORACLE, redhat, semanticweb.com, Tasktop
  - Supporters: CAMBRIDGE SEMANTICS, SIEMENS

W3C Linked Data Platform (LDP) Working Group

Details @ http://www.w3.org/2012/ldp

- Workgroup membership spans 45 participants from 27 organizations
- IBM and EMC as co-chair
- Working towards defining a clear definition of “Linked Data”, in the form of a W3C Recommendation
  - Based on Tim Berners-Lee four principles of Linked Data
- Chartered to produce a “Linked Data Platform” specification that:
  - HTTP-based (RESTful) application integration patterns using read/write Linked Data
  - Will complement SPARQL and will be compatible with standards for publishing Linked Data, bringing the data integration features of RDF to RESTful, data-oriented software development.

- Published First Public Working Draft on 25 October 2012
- Use Case & Requirements publication pending – editor’s draft
- On target to deliver a W3C Candidate Recommendation in 2013

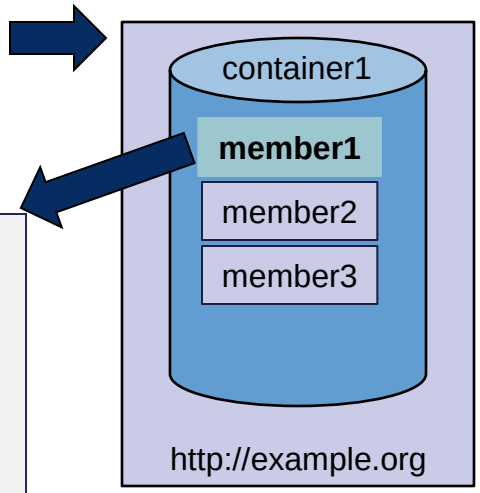
Linked Data Platform - Resource

- What resource formats should be used?
- What literal value types should be used?
- Are there some typical vocabularies that should be reused?
- How is optimistic collision detection handled for updates?
- What should client expectations be for changes to linked-to resources, such as type changes?
- What can servers do to ease the burden of constraints for resource creation?

LDP Resource – GET a simple example

Request

```
GET /container1/member1 HTTP/1.1
Host: example.org
Accept: text/turtle
```



Response

```
Removed HTTP headers to save some space
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ldp: <http://w3.org/ns/ldp#>.

<http://example.org/container1/member1>
   a o:Cash;
   dcterms:title “ACME Bank savings account”;
   o:value 45.00.
```

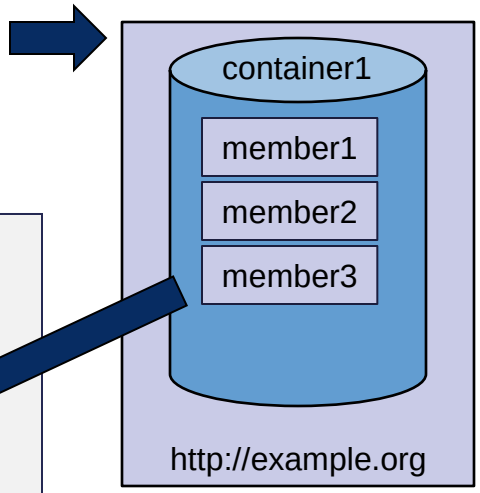
Linked Data Platform - Container

- To which URLs can I POST to create new resources?
- Where can I GET a list of existing resources?
- How is the order of the container entries expressed?
- How do I get information about the members along with the container?
- How do I GET the entries of a large container broken up into pages?
- How can I ensure the resource data is easy to query?

ldp:Container – GET a simple example

Request

```
GET /container1 HTTP/1.1
Host: example.org
Accept: text/turtle
```



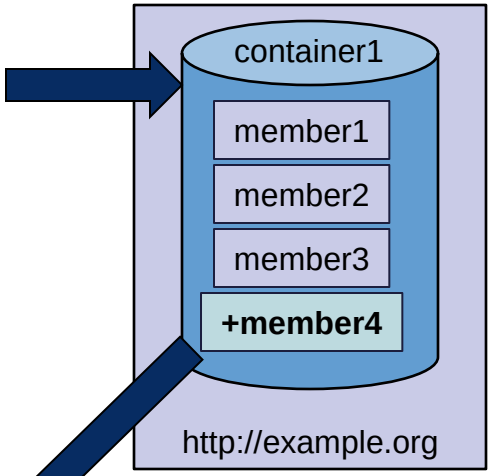
Response

```
Removed HTTP headers to save some space
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ldp: <http://w3c.org/ns/ldp#>.

<http://example.org/container1>
   a ldp:Container;
   dcterms:title "A very simple container";
   rdfs:member
      <http://example.org/container1/member1>,
      <http://example.org/container1/member2>,
      <http://example.org/container1/member3>.
```

ldp:Container – create resource and add

Request

```
POST /container1 HTTP/1.1
Host: example.org
Content-type: text/turtle
Content-length: 324

@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix o: <http://example.org/ontology/>.

<>
   a o:Stock;
   dcterms:title “ACME Co.”;
   o:value 100.00.
```



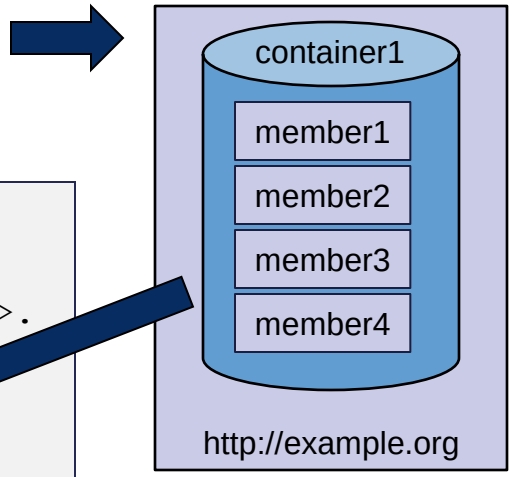
Response

```
HTTP/1.1 201 CREATED

Content-Location: http://example.org/container1/member4
```

ldp:Container – get updated

Request

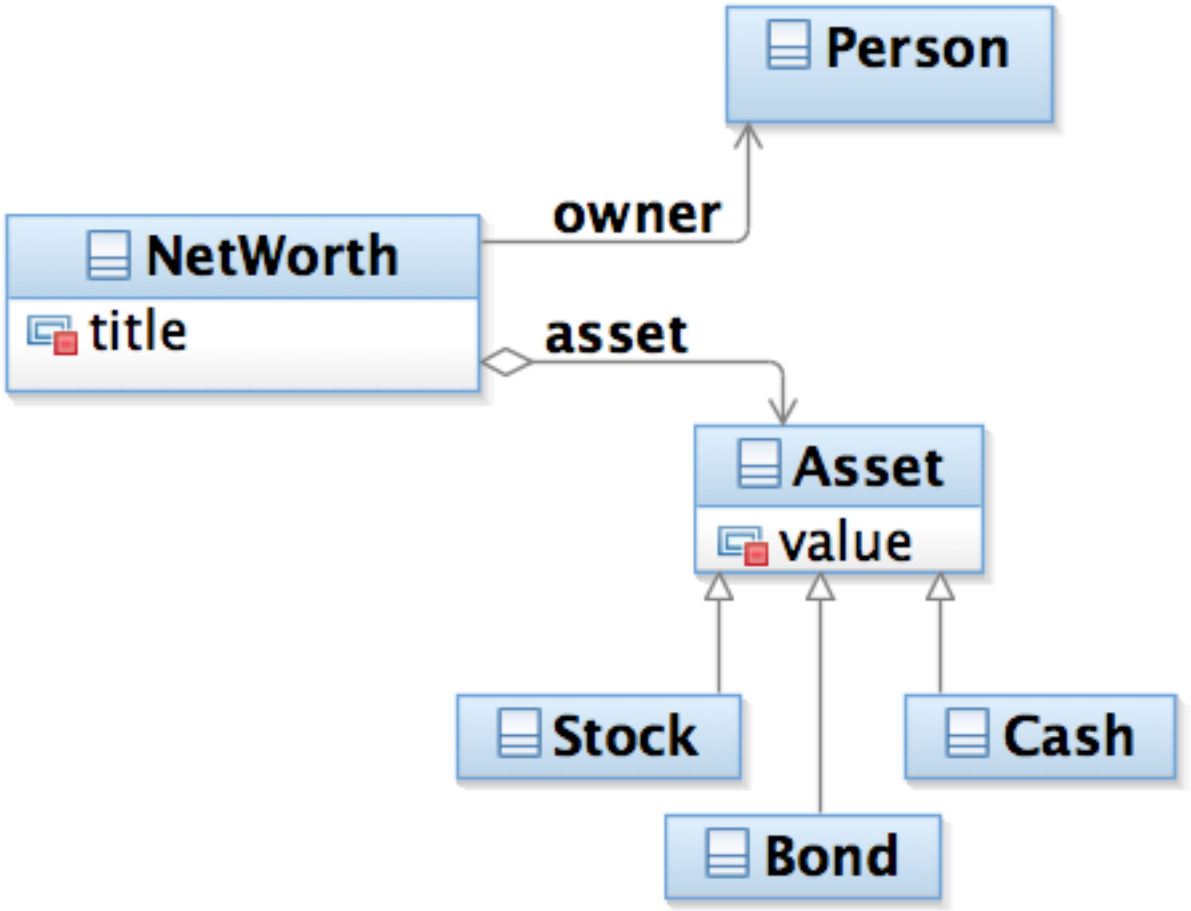
```
GET /container1 HTTP/1.1
Host: example.org
Accept: text/turtle
```



Response

```
Removed HTTP headers to save some space
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ldp: <http://w3c.org/ns/ldp#>.

<http://example.org/container1>
   a ldp:Container;
   dcterms:title "A very simple container";
   rdfs:member
      <http://example.org/container1/member1>,
      <http://example.org/container1/member2>,
      <http://example.org/container1/member3>,
      <http://example.org/container1/member4>.
```

ldp:Container – Net worth example



Disclaimer: picture only in UML, does not follow UML constraints

ldp:Container – slightly more complex example

```
# The following is the representation of
#   http://example.org/netWorth/nw1
@prefix o: <http://example.org/ontology/>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ldp: <http://w3c.org/ns/ldp#>.

<http://example.org/netWorth/nw1>
   a o:NetWorth;
   o:asset
      <http://example.org/netWorth/nw1/assetContainer/a1>,
      <http://example.org/netWorth/nw1/assetContainer/a2>.

<http://example.org/netWorth/nw1/assetContainer>
   a ldp:Container;
   dcterms:title "The assets of JohnZSmith";
   ldp:membershipSubject <http://example.org/netWorth/nw1>;
   ldp:membershipPredicate o:asset.
```

- Membership predicate is something other than rdfs:member (o:asset)
- Subject for members is something other than the container

ldp:Container – Member Information

```
# chopped @prefix
<http://example.org/netWorth/nw1>
   a o:NetWorth;
   o:asset
      <http://example.org/netWorth/nw1/assetContainer/a1>,
      <http://example.org/netWorth/nw1/assetContainer/a2>.

<http://example.org/netWorth/nw1/assetContainer>
   a ldp:Container;
   dcterms:title "The assets of JohnZSmith";
   ldp:membershipSubject <http://example.org/netWorth/nw1>;
   ldp:membershipPredicate o:asset.

<http://example.org/netWorth/nw1/assetContainer/a1>
   a o:Stock;
   o:value 10000.
<http://example.org/netWorth/nw1/assetContainer/a2>
   a o:Bond;
   o:value 20000.
```

- Statements about the member resources included with container representation response

ldp:Container – Only data *about* it, no members

Request

```
GET /netWorth/nw1?non-member-properties HTTP/1.1
Host: example.org
Accept: text/turtle; charset=utf-8
```

Response

```
Removed HTTP headers to save some space
@prefix o: <http://example.org/ontology/>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ldp: <http://w3c.org/ns/ldp#>.

<http://example.org/netWorth/nw1>
   a o:NetWorth.
# Members omitted

<http://example.org/netWorth/nw1/assetContainer>
   a ldp:Container;
   dcterms:title "The assets of JohnZSmith";
   ldp:membershipSubject <http://example.org/netWorth/nw1>;
   ldp:membershipPredicate o:asset.
```


ldp:Container – Paging (first page)

```
# Request-URI: http://example.org/netWorth/nw1?firstPage
# omitted @prefix
<http://example.org/netWorth/nw1>
   a o:NetWorth;
   o:asset
      <http://example.org/netWorth/nw1/assetContainer/a1>,
      <http://example.org/netWorth/nw1/assetContainer/a2>.

<http://example.org/netWorth/nw1/assetContainer>
   a ldp:Container;
   dcterms:title "The assets of JohnZSmith";
   bp:membershipSubject <http://example.org/netWorth/nw1>;
   bp:membershipPredicate o:asset.

<http://example.org/netWorth/nw1/assetContainer?firstPage>
  a ldp:Page;
  ldp:pageOf   <http://example.org/netWorth/nw1/assetContainer>;
  ldp:nextPage <http://example.org/netWorth/nw1/assetContainer?p=2>.

<http://example.org/netWorth/nw1/assetContainer/a1>
      a o:Stock;
      o:value 100.00.
<http://example.org/netWorth/nw1/assetContainer/a2>
      a o:Cash;
      o:value 50.00.
```

- New ldp:Page resource
- Members of containers are paged, not HTTP responses

ldpContainer – Paging (last page)

```
# Request-URI: http://example.org/netWorth/nw1/assetContainer?p=2
# omitted @prefix
<http://example.org/netWorth/nw1>
   a o:NetWorth;
   o:asset
      <http://example.org/netWorth/nw1/assetContainer/a3>.

<http://example.org/netWorth/nw1/assetContainer>
   a bp:Container;
   dcterms:title "The assets of JohnZSmith";
   bp:membershipSubject <http://example.org/netWorth/nw1>;
   bp:membershipPredicate o:asset.

<http://example.org/netWorth/nw1/assetContainer?p=2>
    a ldp:Page;
    ldp:pageOf <http://example.org/netWorth/nw1/assetContainer>;
    ldp:nextPage rdf:nil.

<http://example.org/netWorth/nw1/assetContainer/a5>
   a o:Stock;
   dcterms:title "Big Co.";
   o:value 200.02.
```

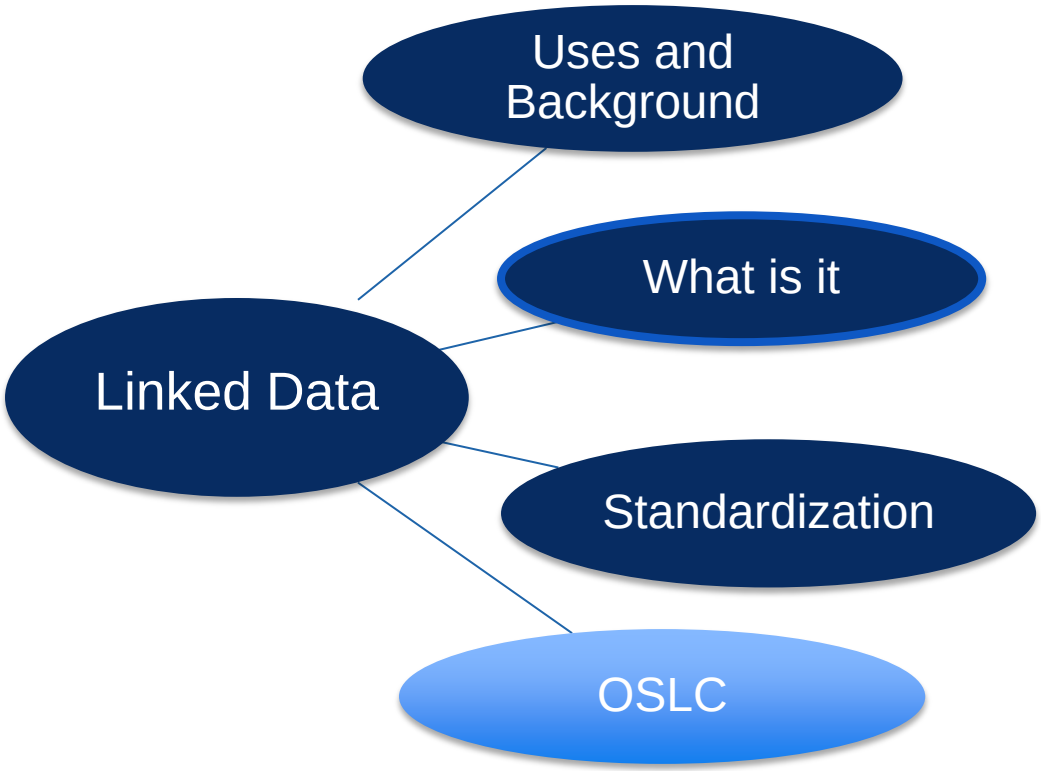
- Last page indicated by rdf:nil

ldp:Container – Paging (ordering)

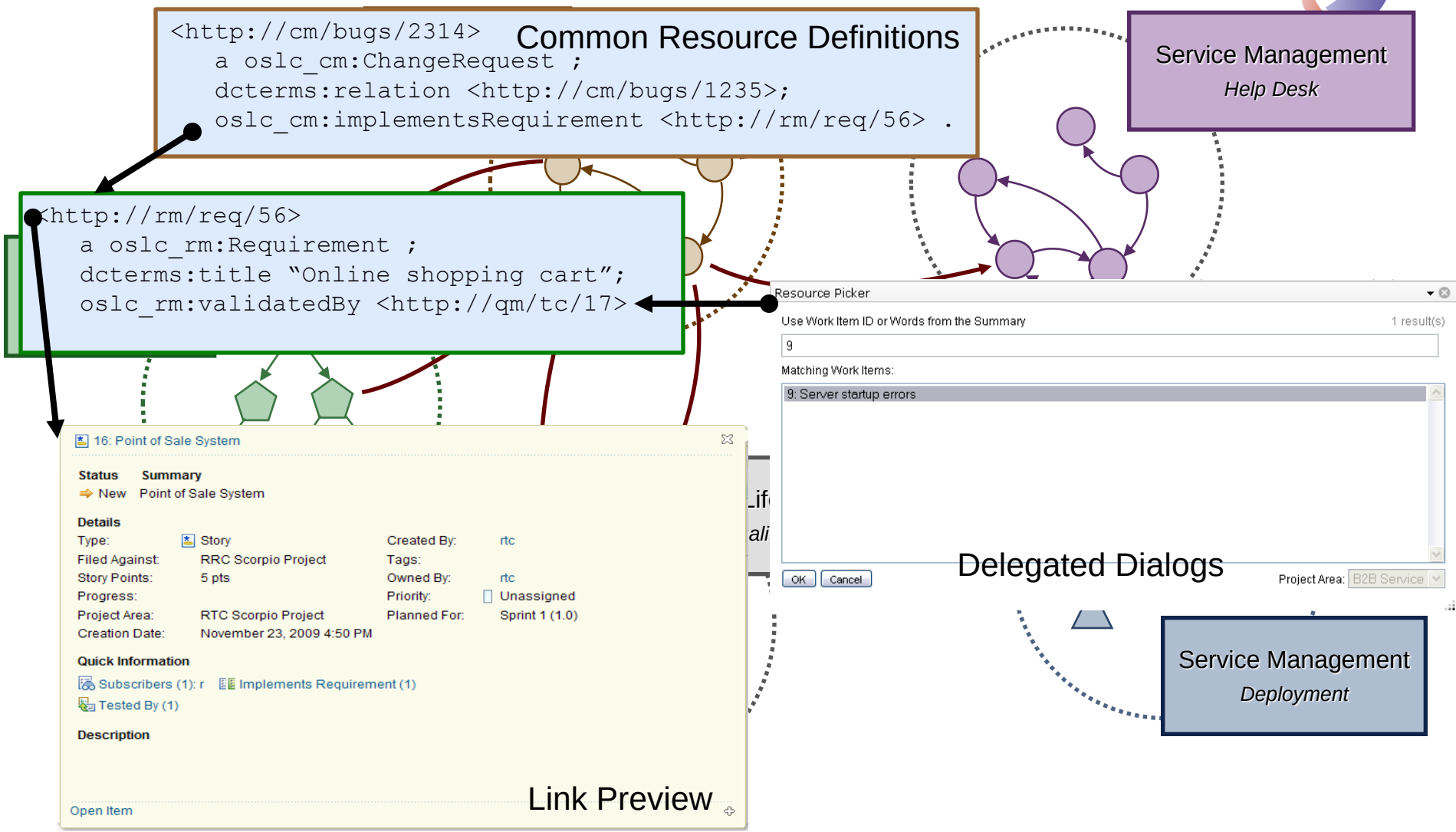
```
# Request-URI: http://example.org/netWorth/nw1/assetContainer?firstPage
# omitted @prefix
<http://example.org/netWorth/nw1>
   a o:NetWorth;
   o:asset
      <http://example.org/netWorth/nw1/assetContainer/a1>,
      <http://example.org/netWorth/nw1/assetContainer/a2>.

<http://example.org/netWorth/nw1/assetContainer>
   a ldp:Container;
   dcterms:title "The assets of JohnZSmith";
   ldp:membershipSubject <http://example.org/netWorth/nw1>;
   ldp:membershipPredicate o:asset.

<http://example.org/netWorth/nw1/assetContainer?firstPage>
  a ldp:Page;
  ldp:pageOf <http://example.org/netWorth/nw1/assetContainer>;
  ldp:containerSortPredicates (o:value).

<http://example.org/netWorth/nw1/assetContainer/a1>
      a o:Stock;
      o:value 100.00.
<http://example.org/netWorth/nw1/assetContainer/a2>
      a o:Cash;
      o:value 50.00.
```

- Order indicated by ldp:containerSortPredicate
- No ordinal predicate introduced, leverage domain model

Agenda



OSLC – Based on Linked Data



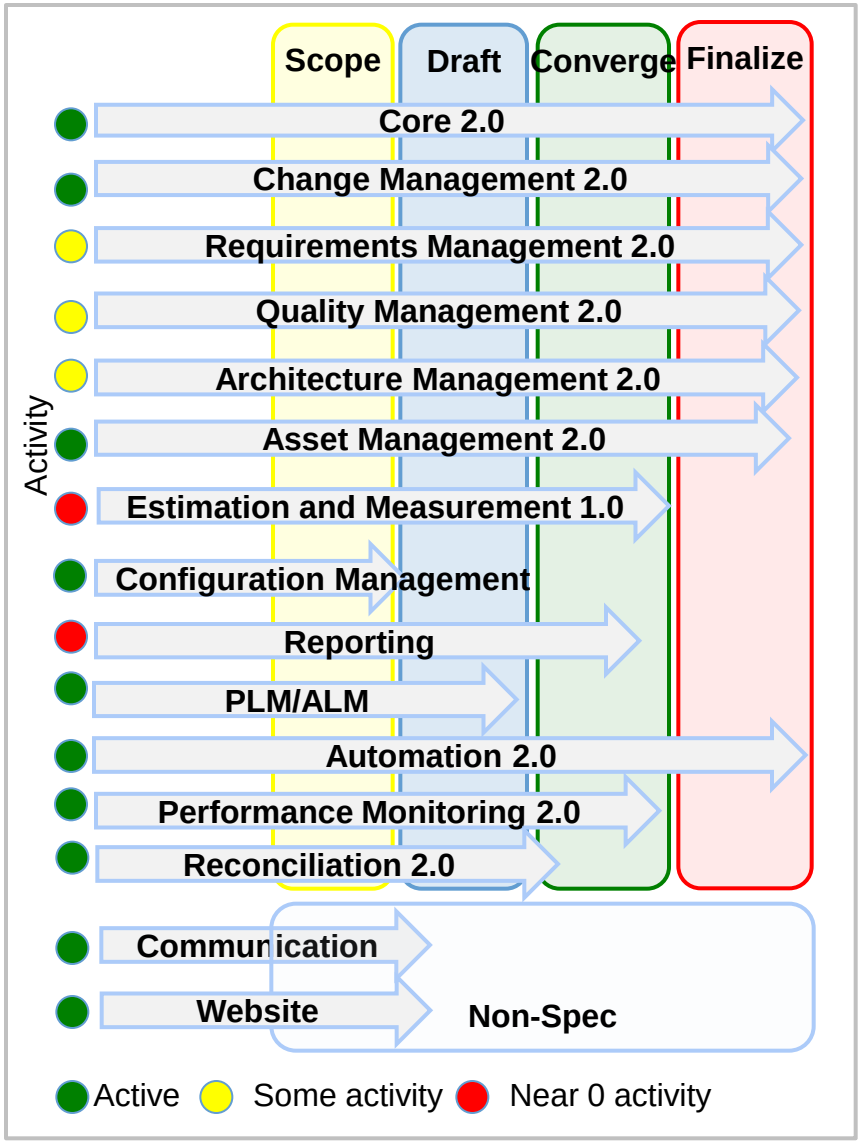
http://open-services.net

Working Group Update

Summary
- Steering Committee new in 2012 already providing key direction
- Some post-2.0 work starting: Core, CM and Architecture
- Re-scoped SCM WG to more general Config Mgmt
- New ISM focused WG: PerfMon and Resource Reconciliation

Participation
- PROMIS Japan Committee on project data using OSLC
- CESAR Interoperability Framework decision to based on OSLC, along with CRYSTAL project
- SPRINT Project plans to base on OSLC

Implementations continue to grow
- Tracking page on oslc.co
- Aided by Eclipse Lyo 1.0 release



Open Source Linked Data projects

Simple yet solid framework for building RESTful Web services based on JAX-RS

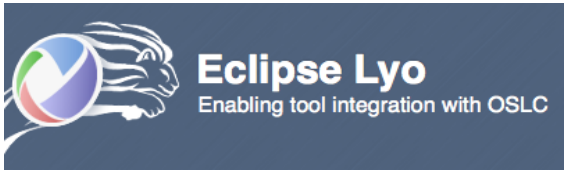
RDF Java API and Fuseki provides REST-style SPARQL HTTP interface

Easily develop semantic web apps by providing tools to manipulate RDF data, create RESTful Web Services

Extend traditional content management systems with semantic services

Marmotta 

Impl of LDP that can be extended and deployed easily by orgs who want to publish data or build custom apps

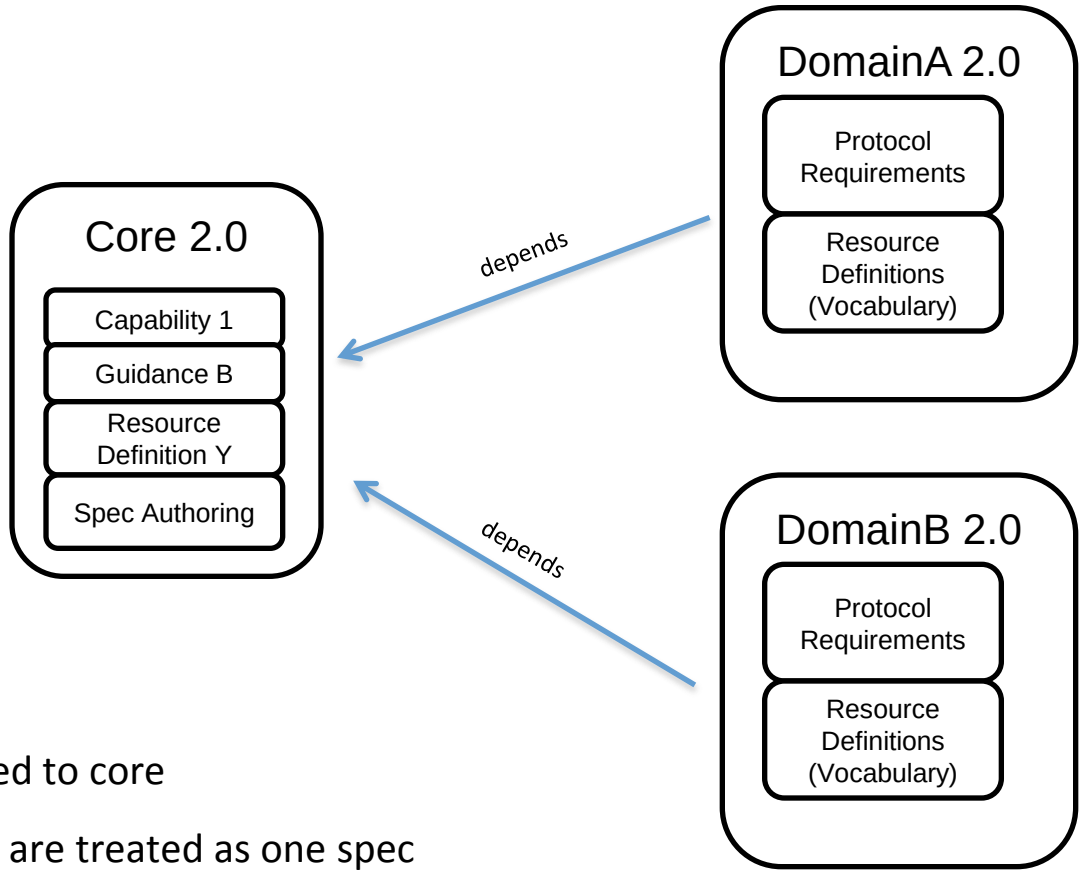


SDK and supporting tools to help the community to adopt OSLC specifications and build compliant tools

Handling Linked Data requests - GET

```
@GET
@Path("{crId}")
@Produces({“application/rdf+xml”, “application/xml”, “application/json”})
public ChangeRequest getChangeRequest(@Context final HttpServletResponse httpServletResponse,
                                      @PathParam(”crId") final String crId)
{
  final ChangeRequest cr = Persistence.getChangeRequest(crId);
  if (cr != null)  {
    cr.setServiceProvider(ServiceProviderSingleton.getServiceProviderURI());
    setETagHeader(getETagFromChangeRequest(cr), httpServletResponse);
    return cr;
  }

  throw new WebApplicationException(Status.NOT_FOUND);
}
```

V2 Spec Structure



- Everything is tied to core
- Domain vocabs are treated as one spec
- Domains are useful to ground discussions with SMEs and learn how to apply common patterns

V2 Spec Issues

- Redundancy -- Same (almost) Core profiles across domains
- Differences -- Different Core profiles across domains
- Core is an abstract base class, need a domain
  - Some cases, there is not domain or a domain variant…require a domain to participate?
- Forces tight coupling of domain specs (vocabularies esp.) to Core
  - Why can't Core evolve and support domain vocabs without having to rev domains everytime?
  - Why do domains need to rev specs to add/adjust vocab terms?

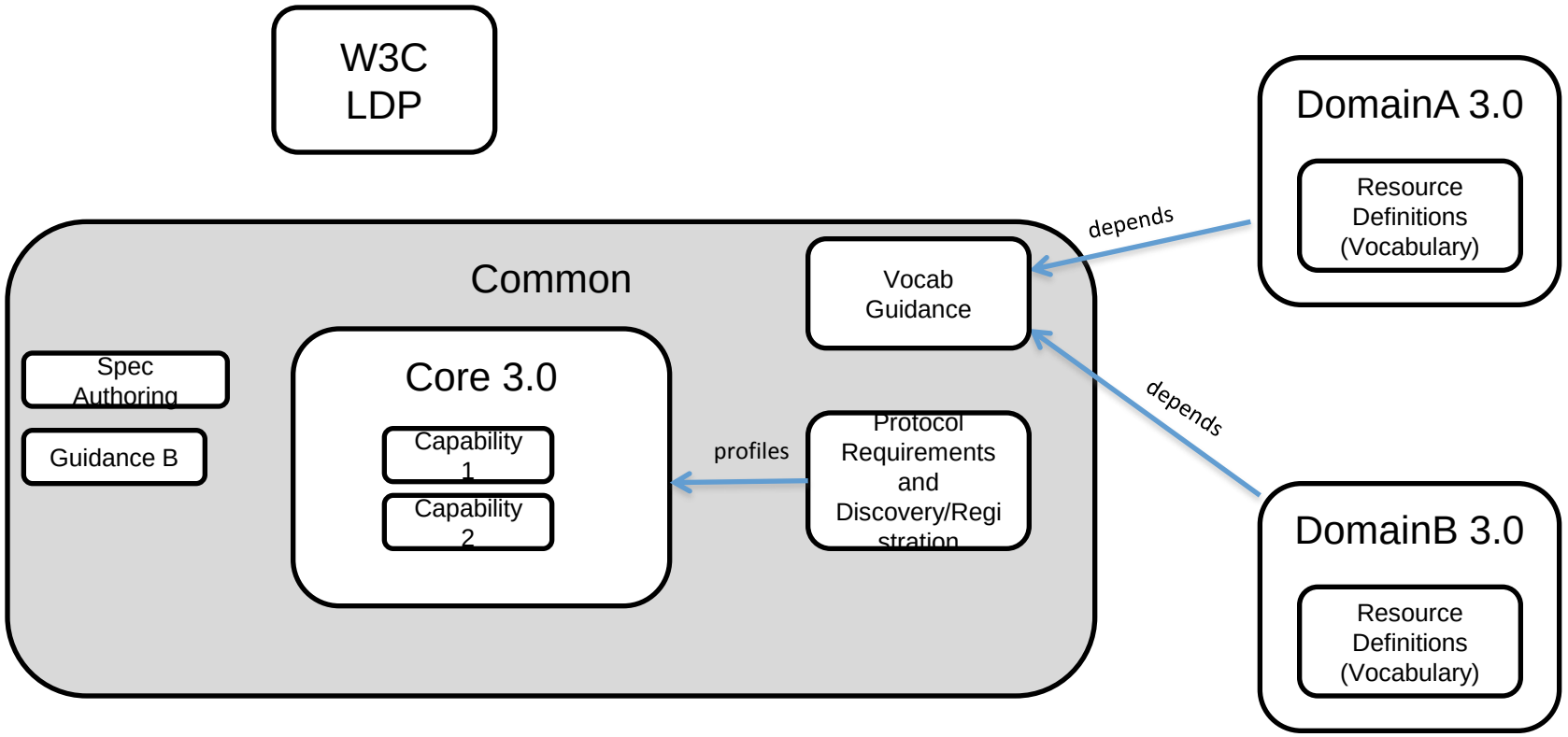
OSLC V3 -- Key Themes

- **Adoption** - evaluate what tasks can be done to help with general adoption of specs - tutorials, SDKs, supporting docs to spec, validating the level of OSLC support through a scale
- **Simplification** - along the same theme of adoption, look for ways to continue to simplify both the number of concepts, model and overall complexity of OSLC, reducing the technical barriers to adoption. This includes alignment with other standards bodies like W3C around linked data patterns.
- **Community Growth** - ways to broaden the community and solidify OSLC as an independent entity opening the door for larger community
- **DevOps** - expanding the reach of ALM into operations which also has some implications of OSLC support strictly within Ops
- **PLM** - exposes specification and guidance on spec usage for PLM
- **ALM continued** - looking at additional scenarios needed. Think of it as a continued evolution of what we have today. This includes cross-cutting capabilities such as partial update.

V3 Spec Goals

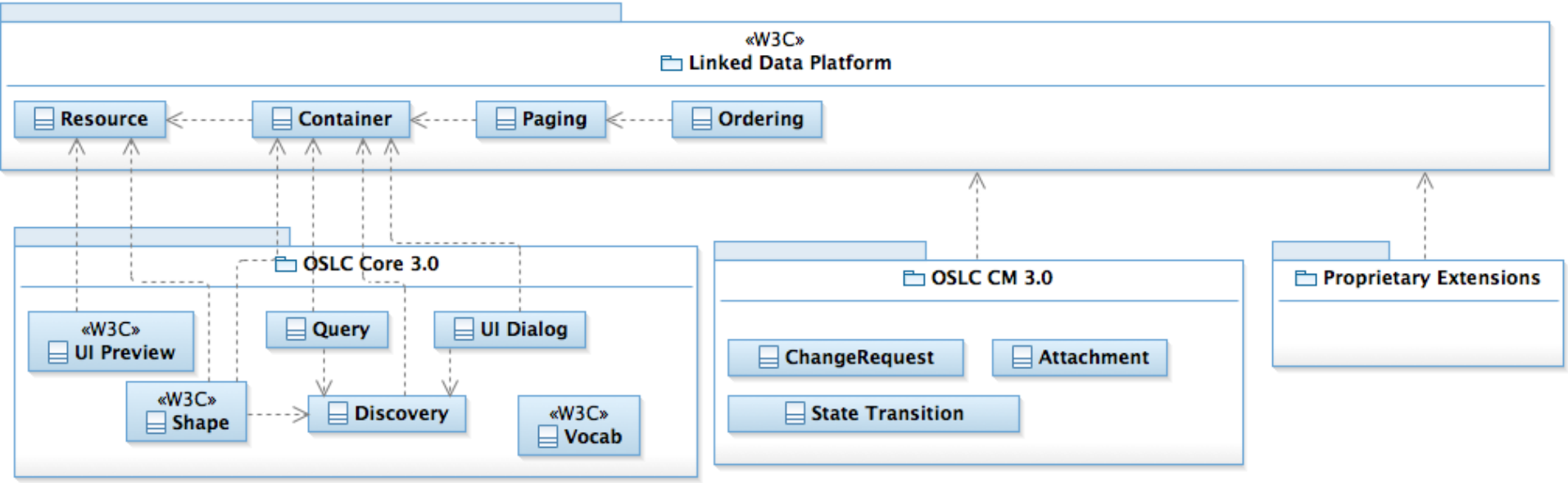
- Collection of independent capabilities
- Specs/capabilities can incrementally be added
- Interesting groupings of these capabilities may make a *profile*
- Capabilities define their own discoverability
  - share a common model when one exists
- DomainX 3.0 = DomainX vocab + Resource Definitions + Core *profile*
- It's ok to share – better reuse of domain vocabs
- Tie everything back to something meaningful
  - Illustrate how spec can be used to satisfy an integration scenario

V3 Spec Structure



- Protocol and Vocabs have different spec lifecycle
- Domains typically don't add protocol or change needs
- "Core" defines how domains or vocabs come into play (Domains are NOT dependent on Core but on vocab guidance)
- Domains may depend on other domain vocabs

V3 Core Capabilities and Domain Sample



- Minimal / no dependency on Core from domain
- Core capabilities can evolve and be additive
- A grouping of capabilities for reuse (profiles)

OSLC Core V3 -- Components

Name	Description
Terminology	A glossary of terminology used throughout the specifications.
Resources Definitions and Operations	Summary of the HTTP and RDF standard techniques and best practices that used, and anti-patterns avoided, when constructing clients and servers that read and write linked data.
Partial update	A partial update (HTTP PATCH) model and representation for web resources.
Containers	The resource that allow new resources to be created using HTTP POST and existing resources to be found using HTTP GET.
Paging	A mechanism for splitting the information in large containers into pages whose representation can be requested incrementally.
Shapes and Validation	A mechanism for describing the properties that a particular type of resource must or may have. Semantics defined using SPARQL ASK.
URL-based Query Model	URL-based query syntax and semantics defined in SPARQL SELECT, binding SPARQL endpoint to Containers.
Links	Rules and guidance around usage of links and statements on links.
Representation	Resource representation guidance on Turtle, JSON and RDF/XML.
Authentication and Security	Requirements and recommendation on authentication models such as OAuth (2.0, 1.0a) and SSO-kind of scenarios.
Error Responses	Resource definition (vocabulary) that can be used as the basis for forming an error response.
Delegated UI DIalogs	Technique where one provider can embed a creation or selection UI from another server using a combination of an HTML <iframe> and JavaScript code.
UI Previews	Technique to show a user in-context information when displaying a link to a resource, and to show more information when the user's mouse lingers over the link.
Discovery	how a client can learn about what a server can do and if it is the right kind of server
Compatibility with Core V2	Considerations around V2 compatibility with V2
Vocabulary	All the terms defined by Core V3
Discussion Resource	Resource definition for a sequence of comments pertaining to the linked to resource.

http://open-services.net/wiki/core/OSLCCoreSpecificationsV3/

Questions?