

Henry Story <henry.story@bblfish.net>
To: public-ldp-wg@w3.org
LDPRs, LDPCs and the mysterious X

25 January 2013 09:44

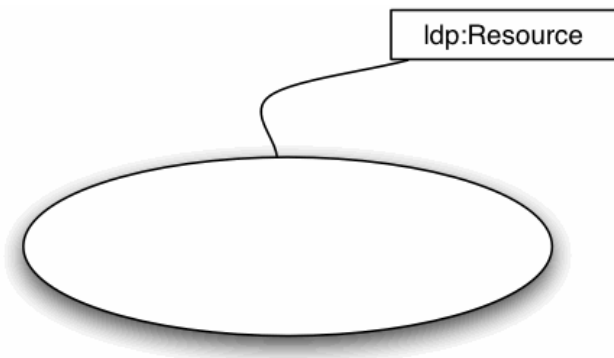


2 Attachments, 16 KB

Let us start with the top of our hierarchy: LDPRs or ldp:Resource.

```
ldp:Resource a rdfs:Class;  
  rdfs:label "LDPR";  
  rdfs:comment ""the class or resources whose behavior is  
    defined in the LDP spec as LDPRs"" .
```

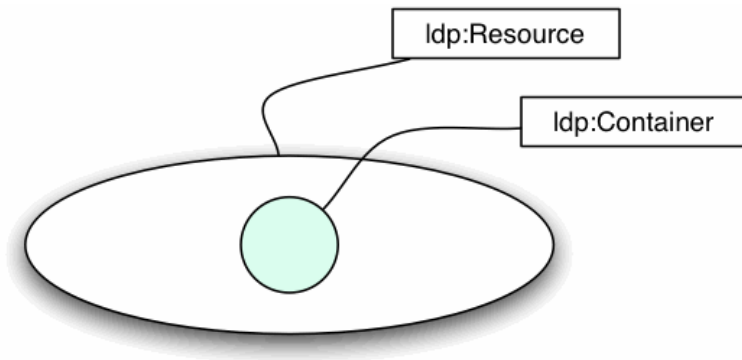
We can draw this set in Venn Diagram fashion as an oval. All ldp:Resources we are to imagine are inside that Oval. Things that are not inside that oval, are things like Cats, numbers such as 42, wine, ...



We then have a different set of things called LDPC or ldp:Container-s . These are all the things that can contain things.

```
ldp:Container a rdfs:Class;  
  rdfs:label "LDPC";  
  rdfs:subClassOf ldp:Resource;  
  rdfs:comment ""  
    The class of resources whose behavior is  
    defined in the LDP spec as LDPCs"" .
```

It is a subclass of ldp:Resource and so that means that any element that is an ldp:Container is an ldp:Resource. In our UML diagram they will therefore be contained in the oval. I'll draw it as a circle here with light green content and place it inside the oval that is ldp:Resource.



So now what remains? We have defined everything currently inside the LDP spec. But from the above we can immediately draw a consequence as to the existence of a third class of things: those things that are NOT LDPCs. That's represented by the space of things that are white in the diagram.

```
_:X a rdfs:Class;
    rdfs:subClassOf ldp:Resource;
    owl:intersectionOf ( [ owl:complementOf ldp:Container ] ldp:Resource ) .
```

So this unnamed `_:X` is a class that is a subclass of LDPR, but does not contain LDPCs. Clearly that must exist, or else there would be no point in creating the class LDPC since it would be identical to the LDPRs.

Also it follows that `_:X` cannot be a subclass of `ldp:Container`, since they have no members in common.

What can we call `_:X`? My thought was that it is that which is contained but which does not contain. In the file system, we tend to call these "files".

Now is it even useful to notice this? Does it have any protocol implications? Well I think it does. For example it explains why ISSUE-45 [1] can give those things a different operational behavior to `ldp:Container`-s with regard to POST. It can do that because there is no overlap between `ldp:Container`-s and `_:X`.

for `ldp:Containers` a POST will create a new resource and add a membership relation to the LDPC. On the other hand for any `_:X` we can give a POST a different interpretation, such as that of APPENDING the triples posted to.

[1] <http://www.w3.org/2012/ldp/track/issues/45>