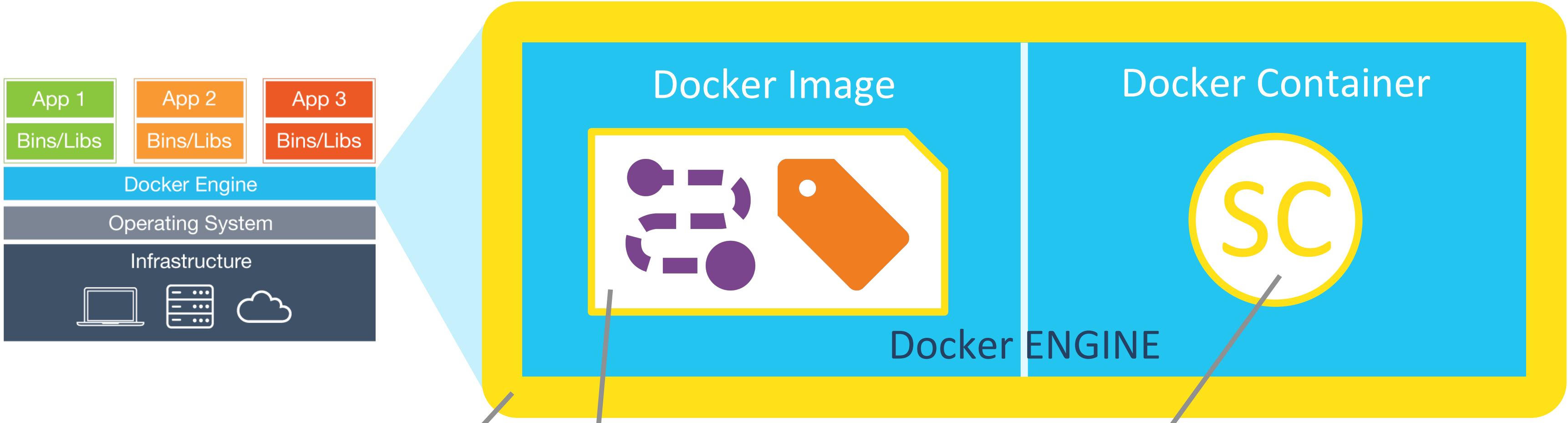
CRC CENTER for RESEARCH COMPUTING

UNIVERSITY of NOTRE DAME

”Smart Containers - Demo”

James Sweet, Charles Vardeman II

Smart Containers IN PRACTICE: Anatomy of a Smart Container



1 Python wrapper is added to standard Docker container

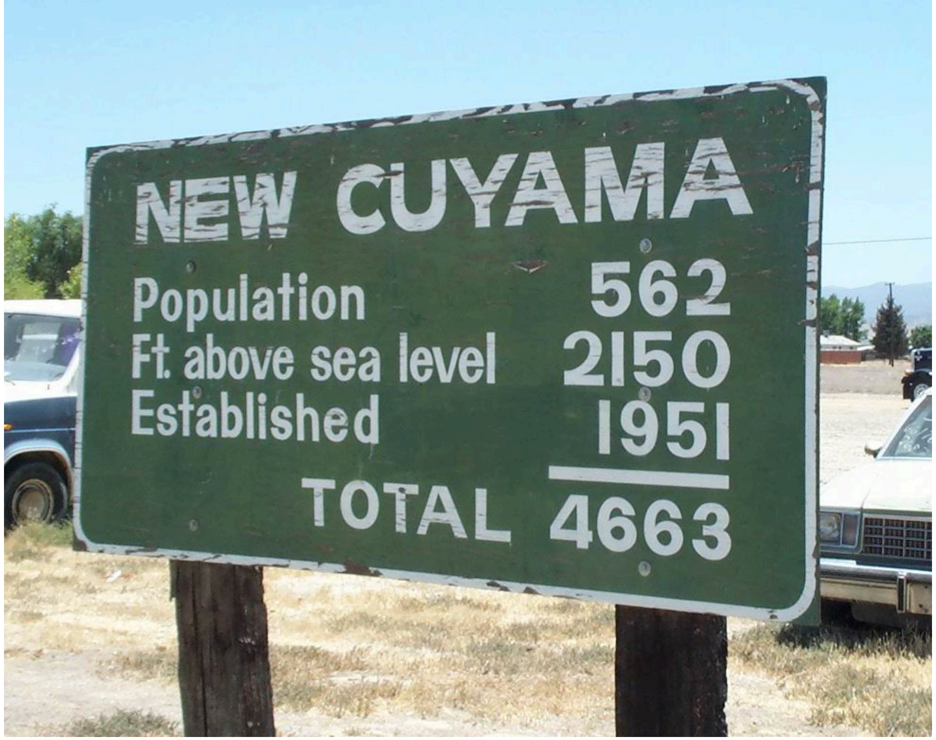
2 Provenance and metadata are written directly to image label

- Machine-readable
- Enables discovery in large repositories

3 Container is provisioned as a “Smart Container”:

- API to write metadata
- Metadata storage and standardization
- Specification of data location

Observation: Semantic Web Technologies provide a Standardized way to provide *Context*



Source: Picture, Mike Gogulski (CC BY 2.5).
Krzysztof Janowicz, Frank van Harmelen, James A. Hendler, and Pascal Hitzler. "Why the Data Train Needs Semantic Rails." AI Magazine, 2014. http://corescholar.libraries.wright.edu/cse/169/.

> “A major paradigm shifts introduced by the Semantic Web is to focus on the creation of smart data instead of smart applications. The rationale behind this shift is the insight that smart data will make future applications more (re)usable, flexible, and robust, while smarter applications fail to improve data along the same dimensions…”

Source: Krzysztof Janowicz, Frank van Harmelen, James A. Hendler, and Pascal Hitzler. “Why the Data Train Needs Semantic Rails.” AI Magazine, 2014. http://corescholar.libraries.wright.edu/cse/169/.

How can containers be used as *Scientific Tools* such that we can provide *Context?*

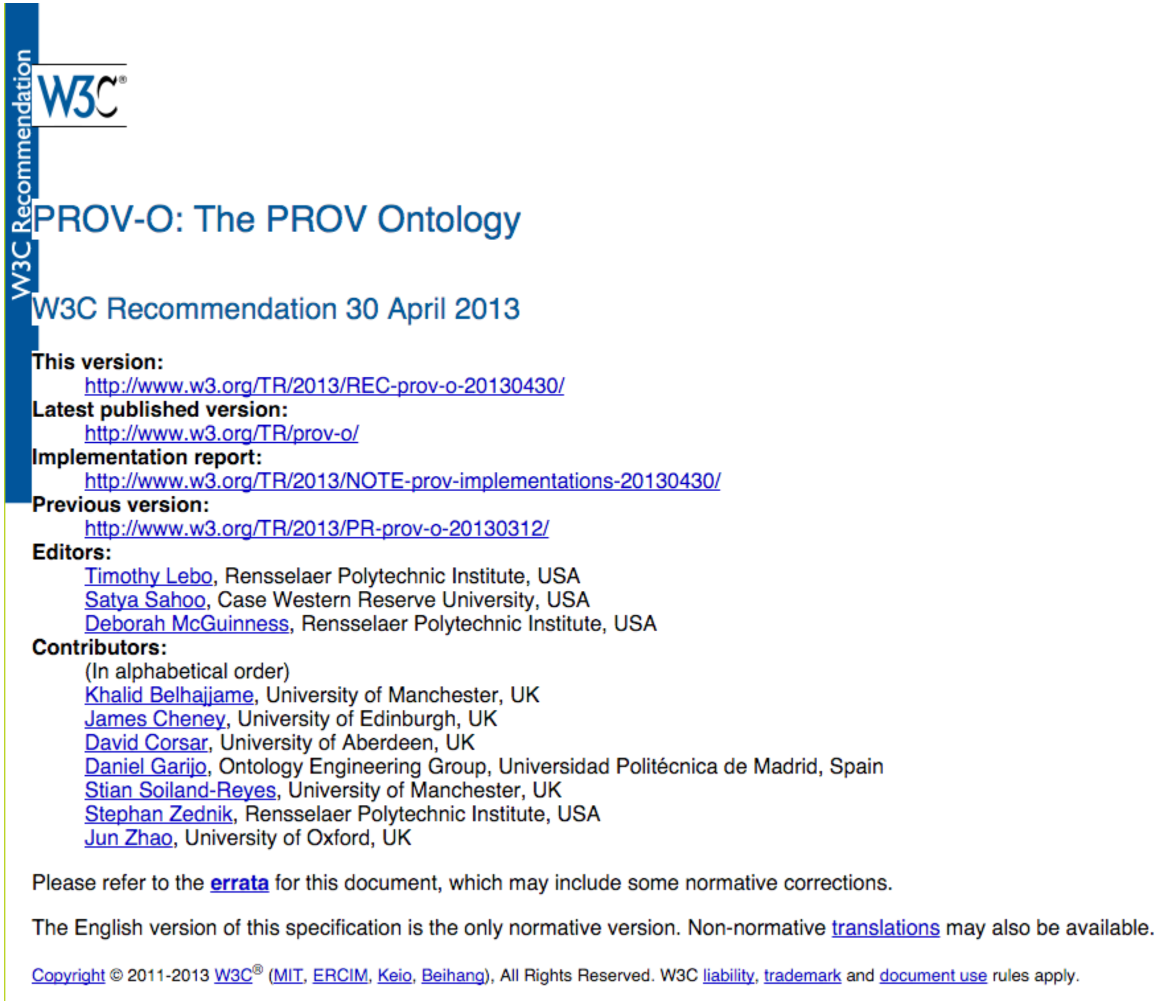
Smart Containers is a Transparent drop-in replacement for the Docker python API and command line interface which embeds linked data within the container.

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.

Source: Tim Berners-Lee, 2006-07-27, https://www.w3.org/DesignIssues/LinkedData.html

Ontology Design Pattern: “reusable successful solution to a recurrent [ontology] modeling problem”

Source: Eva Blomqvist, Pascal Hitzler, Krzysztof Janowicz, Adila Krisnadhi, Tom Narock, and Monika Solanki. “Considerations Regarding Ontology Design Patterns.” *Semantic Web* 7, no. 1 (2015): 1–7.



W3C Recommendation

PROV-O: The PROV Ontology

W3C Recommendation 30 April 2013

This version:
http://www.w3.org/TR/2013/REC-prov-o-20130430/
Latest published version:
http://www.w3.org/TR/prov-o/
Implementation report:
http://www.w3.org/TR/2013/NOTE-prov-implementations-20130430/
Previous version:
http://www.w3.org/TR/2013/PR-prov-o-20130312/
Editors:
Timothy Lebo, Rensselaer Polytechnic Institute, USA
Satya Sahoo, Case Western Reserve University, USA
Deborah McGuinness, Rensselaer Polytechnic Institute, USA
Contributors:
(In alphabetical order)
Khalid Belhajjame, University of Manchester, UK
James Cheney, University of Edinburgh, UK
David Corsar, University of Aberdeen, UK
Daniel Garijo, Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
Stian Soiland-Reyes, University of Manchester, UK
Stephan Zednik, Rensselaer Polytechnic Institute, USA
Jun Zhao, University of Oxford, UK

Please refer to the **errata** for this document, which may include some normative corrections.

The English version of this specification is the only normative version. Non-normative translations may also be available.

Source: https://www.w3.org/TR/prov-o/

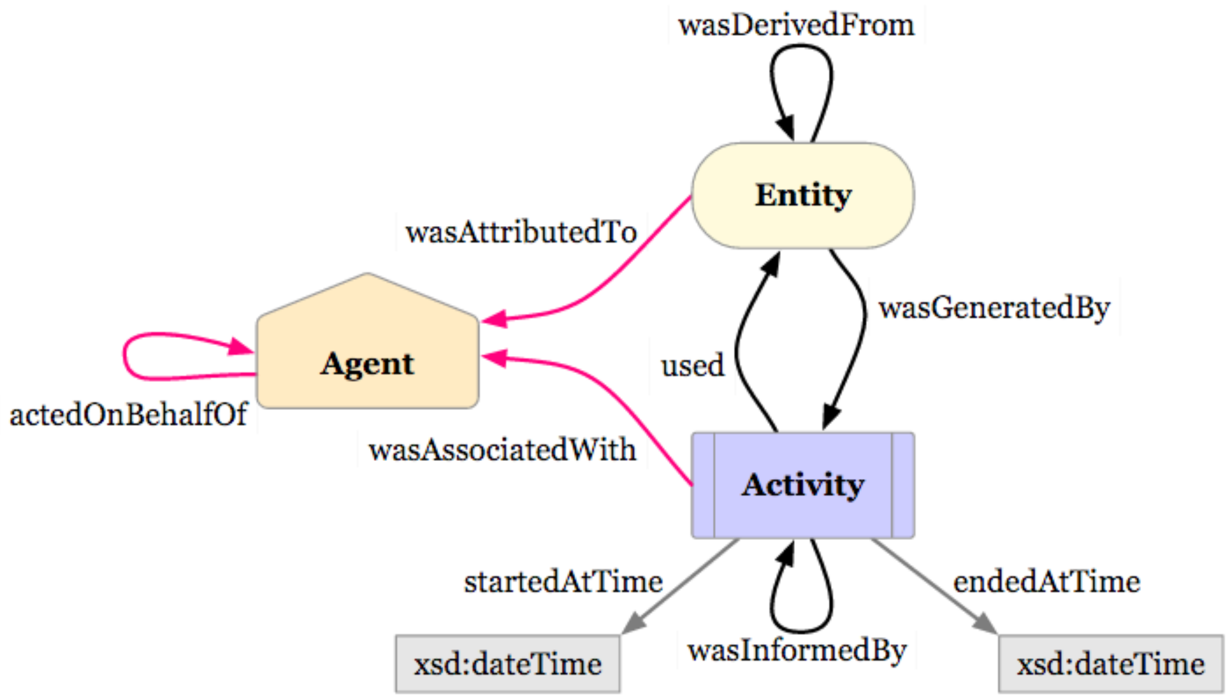


Figure 1. The three Starting Point classes and the properties that relate them.
The diagrams in this document depict Entities as yellow ovals,
Activities as blue rectangles, and Agents as orange pentagons.
The responsibility properties are shown in pink.

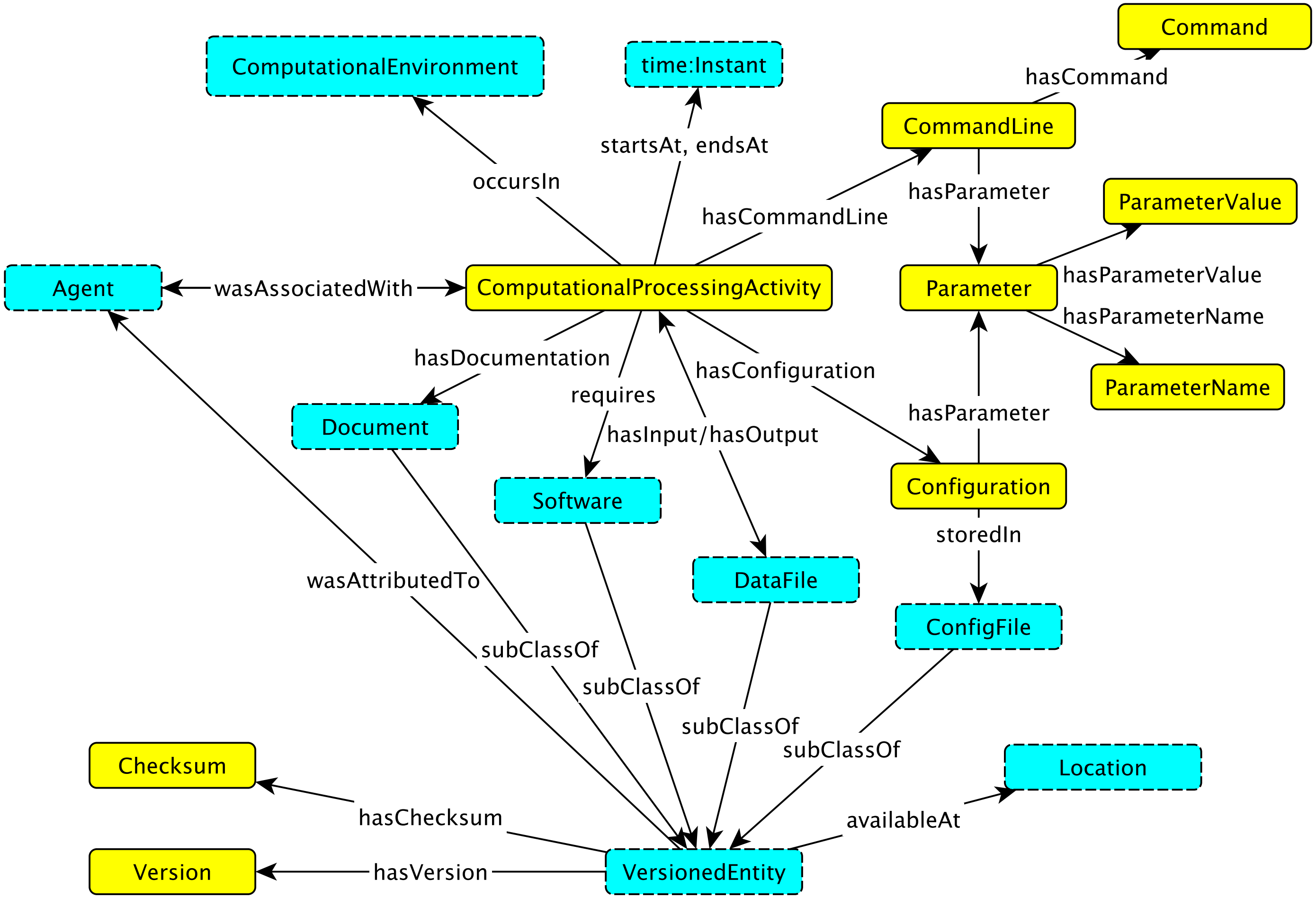
Source: https://www.w3.org/TR/prov-o/

ORCID ID’s are Smart Containers preferred method for defining human agents (Prov:Person) vs. a Prov:SoftwareAgent (i.e. Docker)

```
curl --header "Accept: text/turtle" -L http://orcid.org/0000-0003-4091-6059
```

```
<http://orcid.org/0000-0003-4091-6059>
      a                prov:Person , foaf:Person ;
      rdfs:label        "Charles Vardeman II" ;
      foaf:account       <http://orcid.org/0000-0003-4091-6059#orcid-id> ;
      foaf:based_near    [ a               gn:Feature ;                    gn:countryCode    "US" ;
gn:parentCountry  <http://sws.geonames.org/6252001/>                    ] ;
      foaf:familyName    "Vardeman II" ;
      foaf:givenName     "Charles" ;
      foaf:publications  <http://orcid.org/0000-0003-4091-6059#workspace-works> .
```

Observation: Activity is very general. We need a *Specialization* of activity that models *Computational Activity*

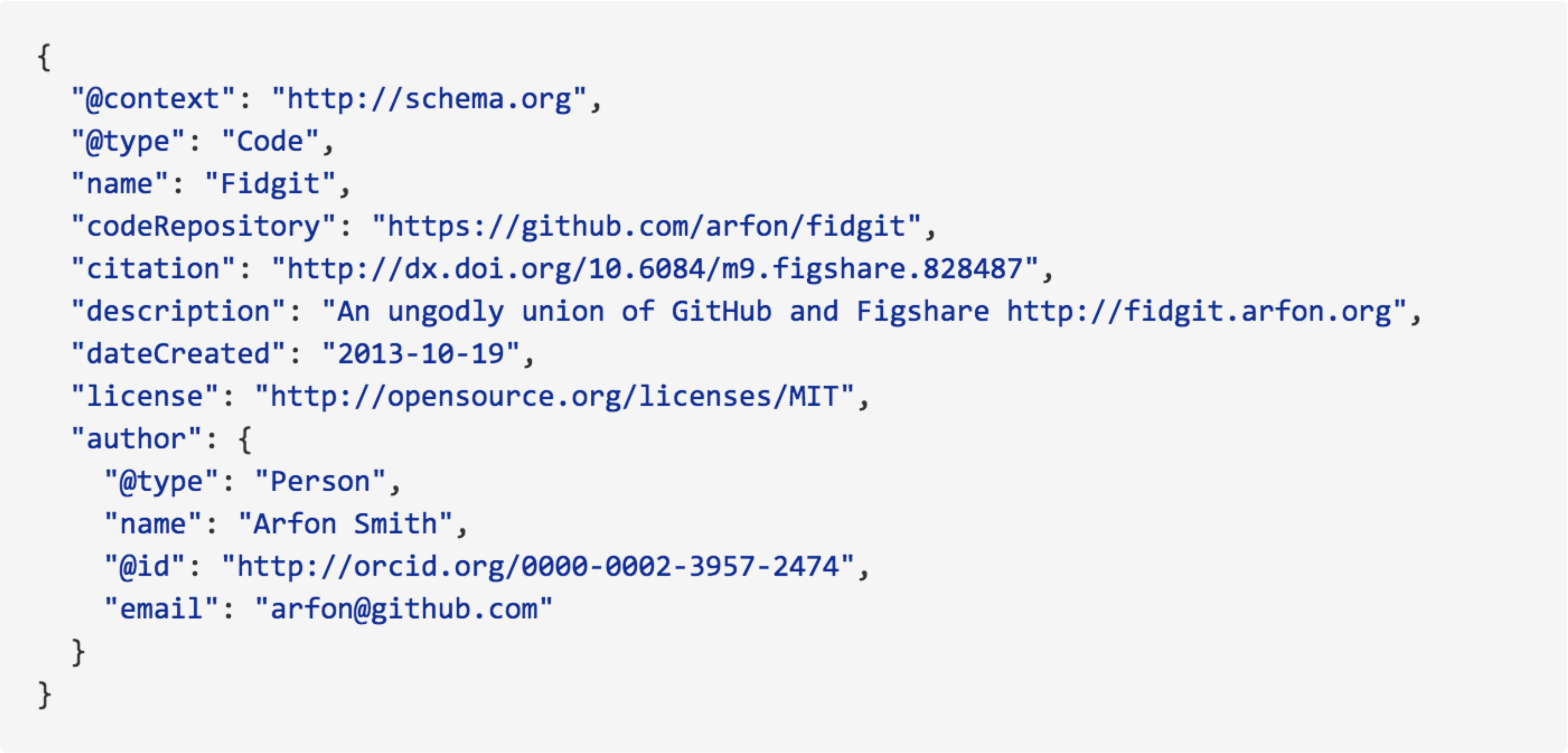


Source: https://github.com/Vocamp/ComputationalActivity

Observation: **Computational Activities** can be connected to *provision* a **Computational Environment**

We can describe the provenance of Docker as a *provider* of a Computational Environment that then instantiates Computational Activities

CodeMeta Effort



```
{
  "@context": "http://schema.org",
  "@type": "Code",
  "name": "Fidgit",
  "codeRepository": "https://github.com/arfon/fidgit",
  "citation": "http://dx.doi.org/10.6084/m9.figshare.828487",
  "description": "An ungodly union of GitHub and Figshare http://fidgit.arfon.org",
  "dateCreated": "2013-10-19",
  "license": "http://opensource.org/licenses/MIT",
  "author": {
    "@type": "Person",
    "name": "Arfon Smith",
    "@id": "http://orcid.org/0000-0002-3957-2474",
    "email": "arfon@github.com"
  }
}
```

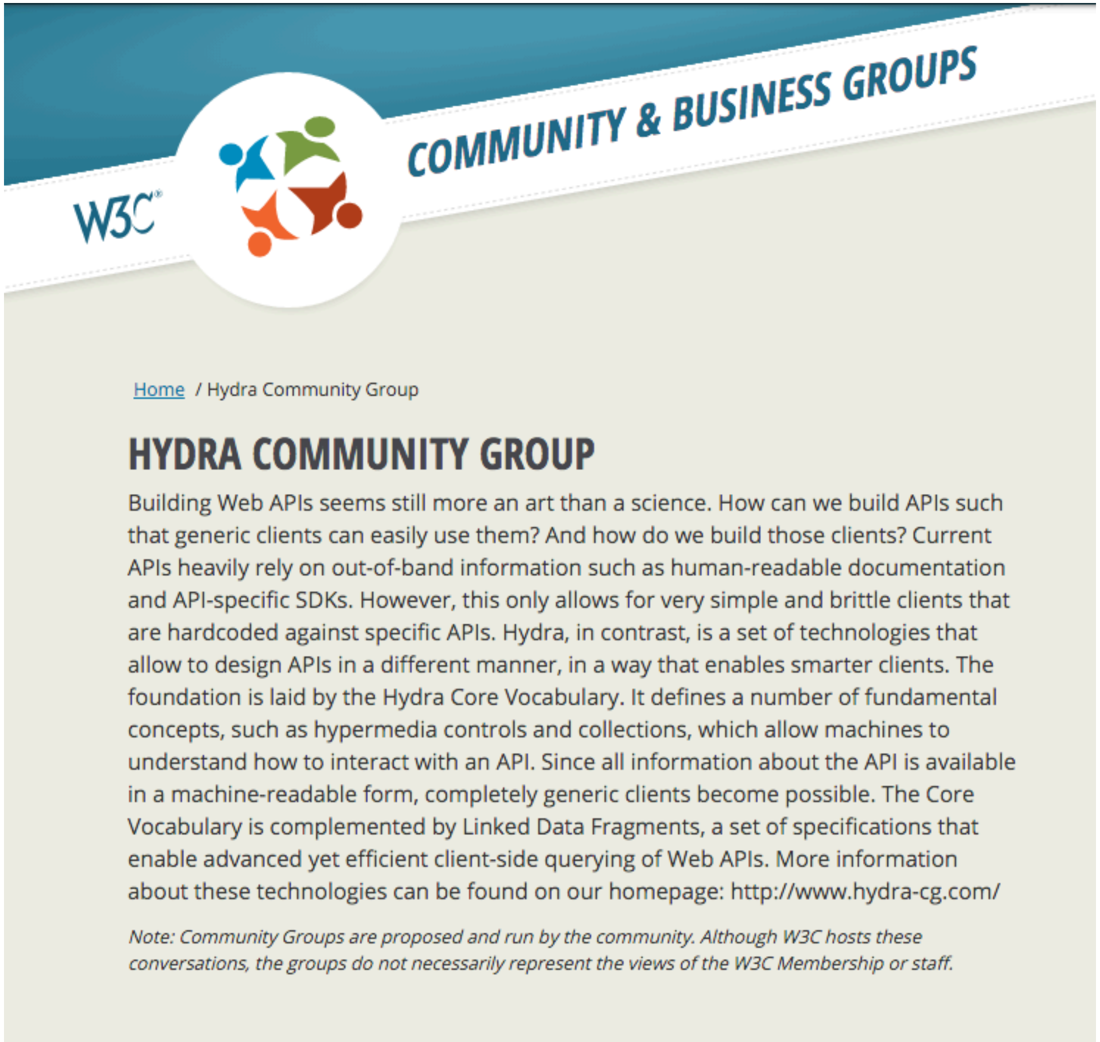
https://github.com/codemeta/codemeta

https://codemeta.github.io/

The provenance information is saved as in the *Docker image* label key/value in using JSON-LD

Demo of command line tool

How do we ”tell” the *machine* where to look for the data and how to access the data?



Home / Hydra Community Group

HYDRA COMMUNITY GROUP

Building Web APIs seems still more an art than a science. How can we build APIs such that generic clients can easily use them? And how do we build those clients? Current APIs heavily rely on out-of-band information such as human-readable documentation and API-specific SDKs. However, this only allows for very simple and brittle clients that are hardcoded against specific APIs. Hydra, in contrast, is a set of technologies that allow to design APIs in a different manner, in a way that enables smarter clients. The foundation is laid by the Hydra Core Vocabulary. It defines a number of fundamental concepts, such as hypermedia controls and collections, which allow machines to understand how to interact with an API. Since all information about the API is available in a machine-readable form, completely generic clients become possible. The Core Vocabulary is complemented by Linked Data Fragments, a set of specifications that enable advanced yet efficient client-side querying of Web APIs. More information about these technologies can be found on our homepage: http://www.hydra-cg.com/

Note: Community Groups are proposed and run by the community. Although W3C hosts these conversations, the groups do not necessarily represent the views of the W3C Membership or staff.

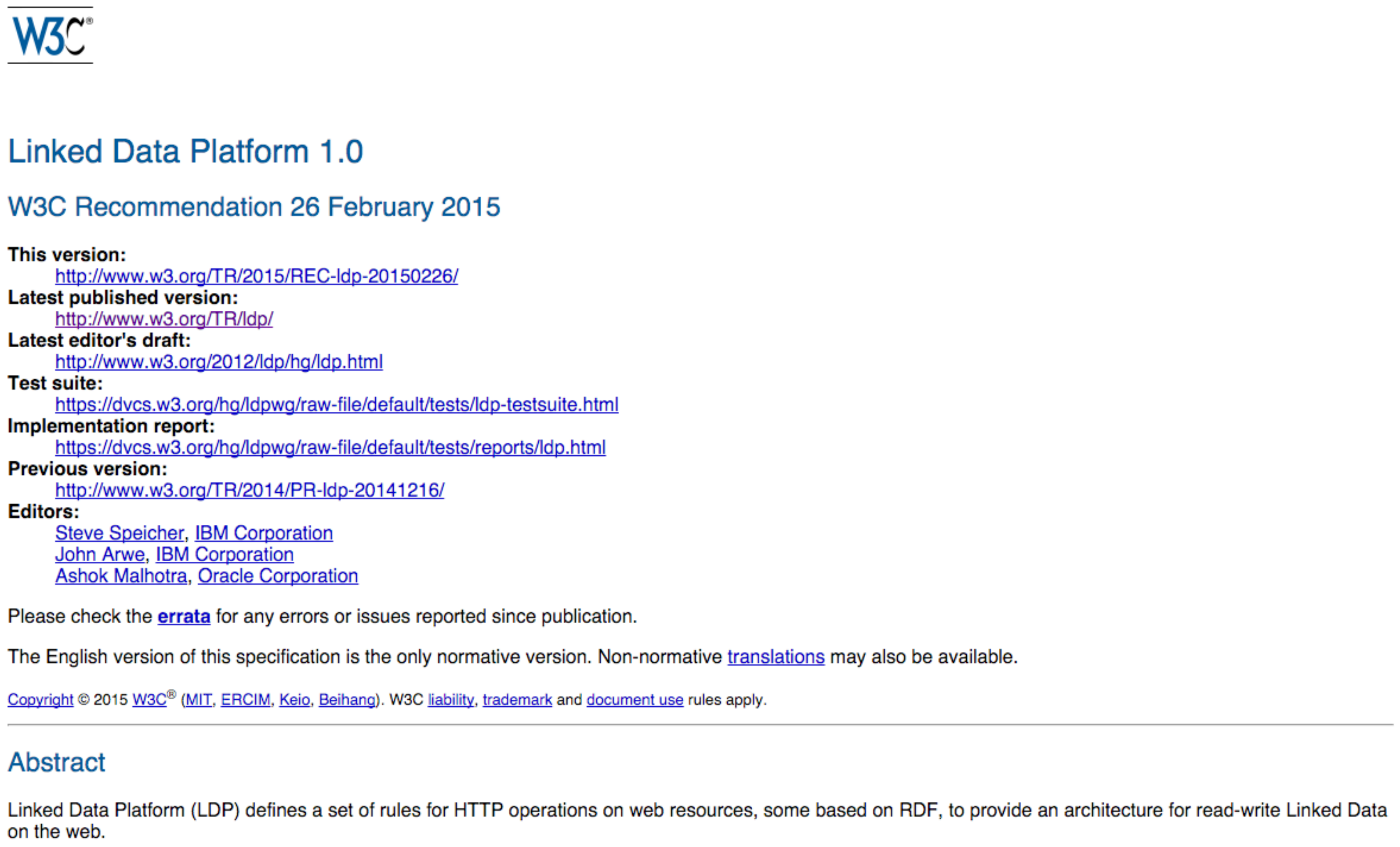
Source: https://www.w3.org/community/hydra/

Container entry-point saved using the Hydra-Core Vocabulary specification the *Docker image in* JSON-LD so clients know how to *access* a running container.

Demo Linked-Data Fragments

http://client.linkeddatafragments.org/

What about the Data?



W3C

Linked Data Platform 1.0

W3C Recommendation 26 February 2015

This version:
http://www.w3.org/TR/2015/REC-ldp-20150226/
Latest published version:
http://www.w3.org/TR/ldp/
Latest editor's draft:
http://www.w3.org/2012/ldp/hg/ldp.html
Test suite:
https://dvcs.w3.org/hg/ldpwg/raw-file/default/tests/ldp-testsuite.html
Implementation report:
https://dvcs.w3.org/hg/ldpwg/raw-file/default/tests/reports/ldp.html
Previous version:
http://www.w3.org/TR/2014/PR-ldp-20141216/
Editors:
Steve Speicher, IBM Corporation
John Arwe, IBM Corporation
Ashok Malhotra, Oracle Corporation

Please check the **errata** for any errors or issues reported since publication.

The English version of this specification is the only normative version. Non-normative translations may also be available.

Abstract

Linked Data Platform (LDP) defines a set of rules for HTTP operations on web resources, some based on RDF, to provide an architecture for read-write Linked Data on the web.

Source: https://www.w3.org/TR/ldp/

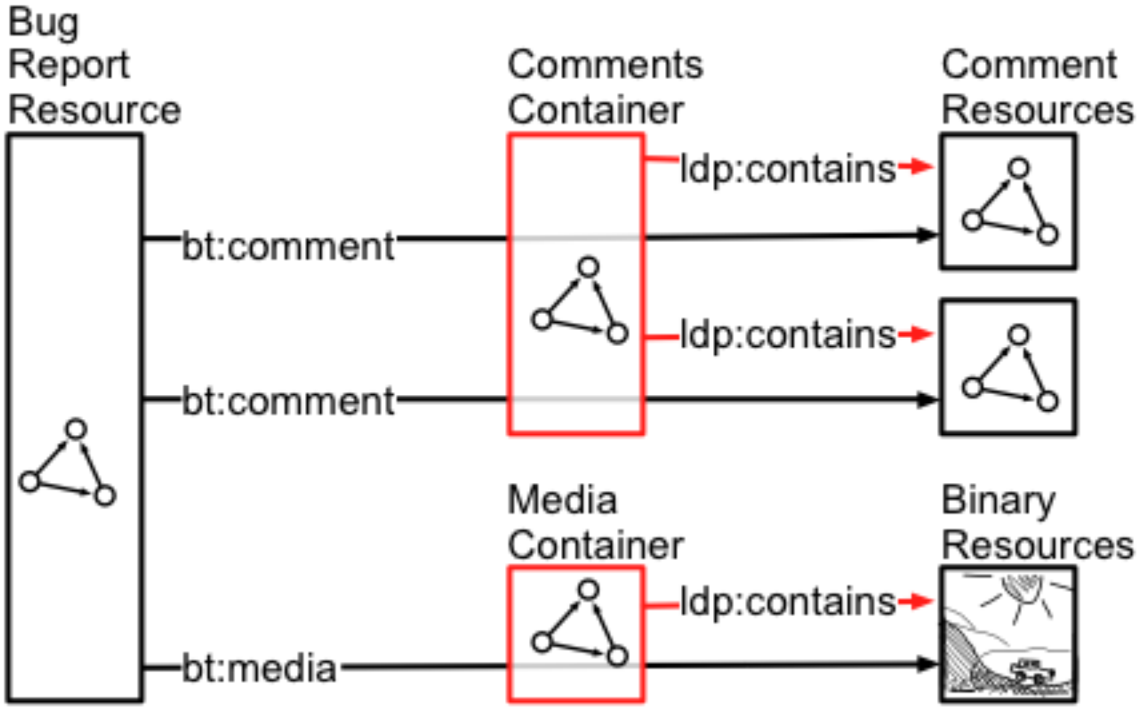


Fig. 4 Managing multiple facets of a Bug with two Direct Containers.

Source: https://www.w3.org/TR/ldp-primer/

Smart Containers organize data inside the container as using a linked data platform specification in a consistent location (/data) as a ldp:container. The provenance information is also provided as a ldp resource.

Thank You

https://github.com/crcresearch/smartcontainers