

Updating Linked Data practices for FAIR Digital Object principles

Stian Soiland-Reyes, Leyla Jael Castro, Daniel Garijo, Marc Portier, Carole Goble, Paul Groth

The University of Manchester

University of Amsterdam

<https://orcid.org/0000-0001-9842-9718>

[@soilandreyes](#)

First International Conference on FAIR Digital Objects

Leiden, 2022-10-27



This work is licensed under a
[Creative Commons Attribution 4.0 International License.](#)

FAIR Digital Object (FDO) - conceptual view

Rigid Persistent Identifiers

Self-describing digital objects

Distributed architecture

Machine actionable

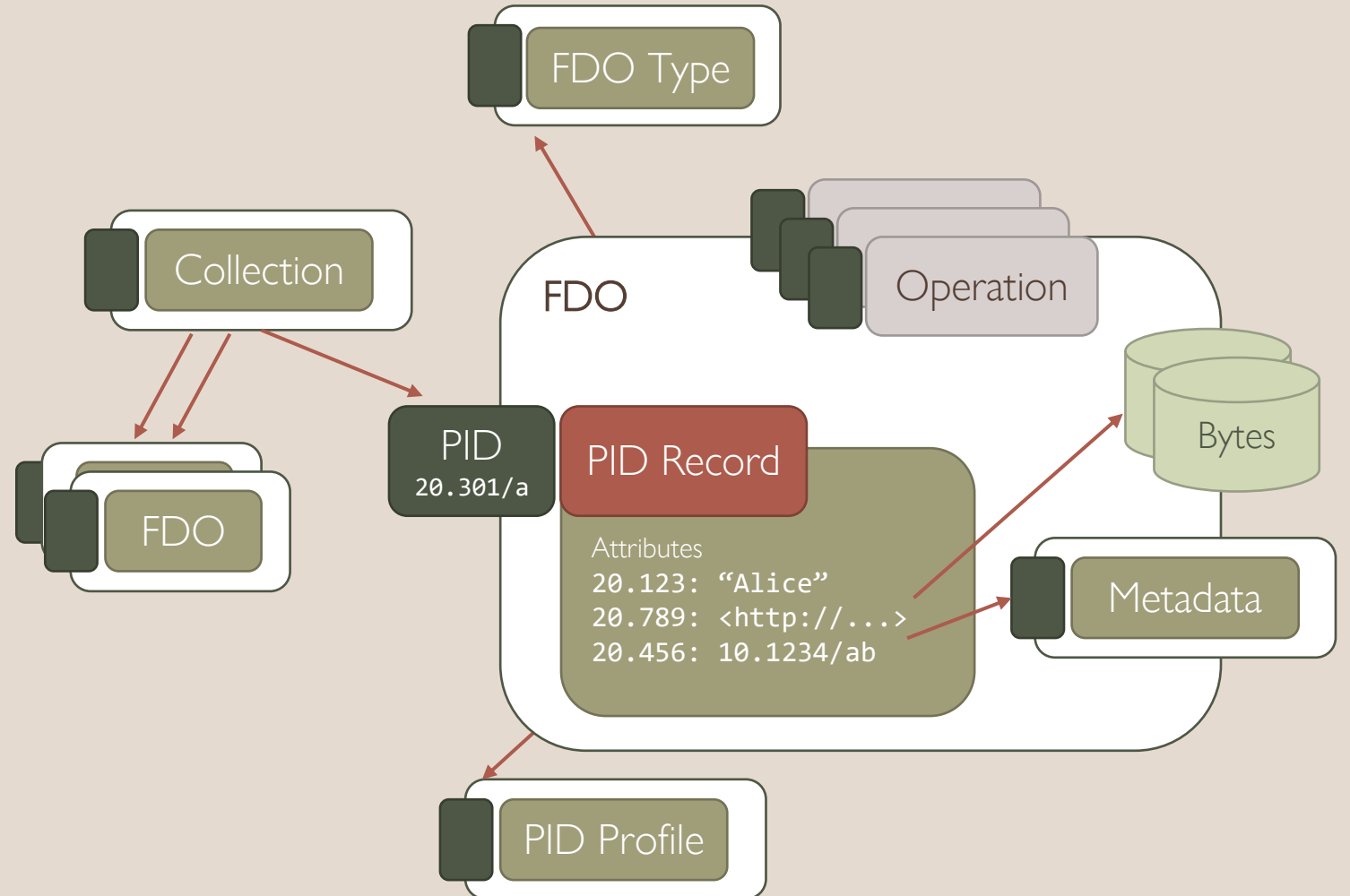
Encapsulation of operations

- CRUD
- Extensible operations

Data/metadata **abstraction**

- Several types of metadata

→ Predictable implementation of FAIR
for **active objects**, not just static data



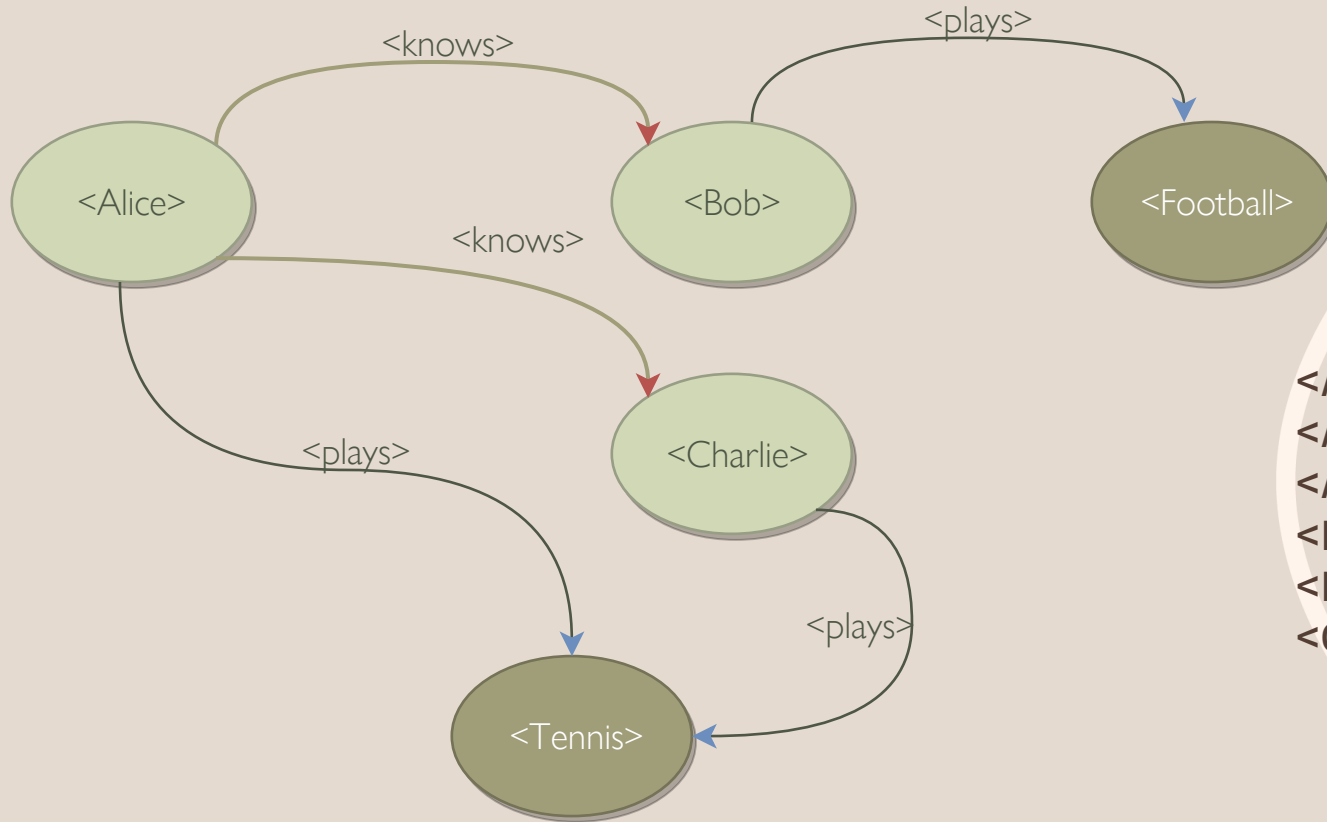
<https://fairdo.org/>
<http://shorturl.at/klrw2>



Linked Data



Triples?

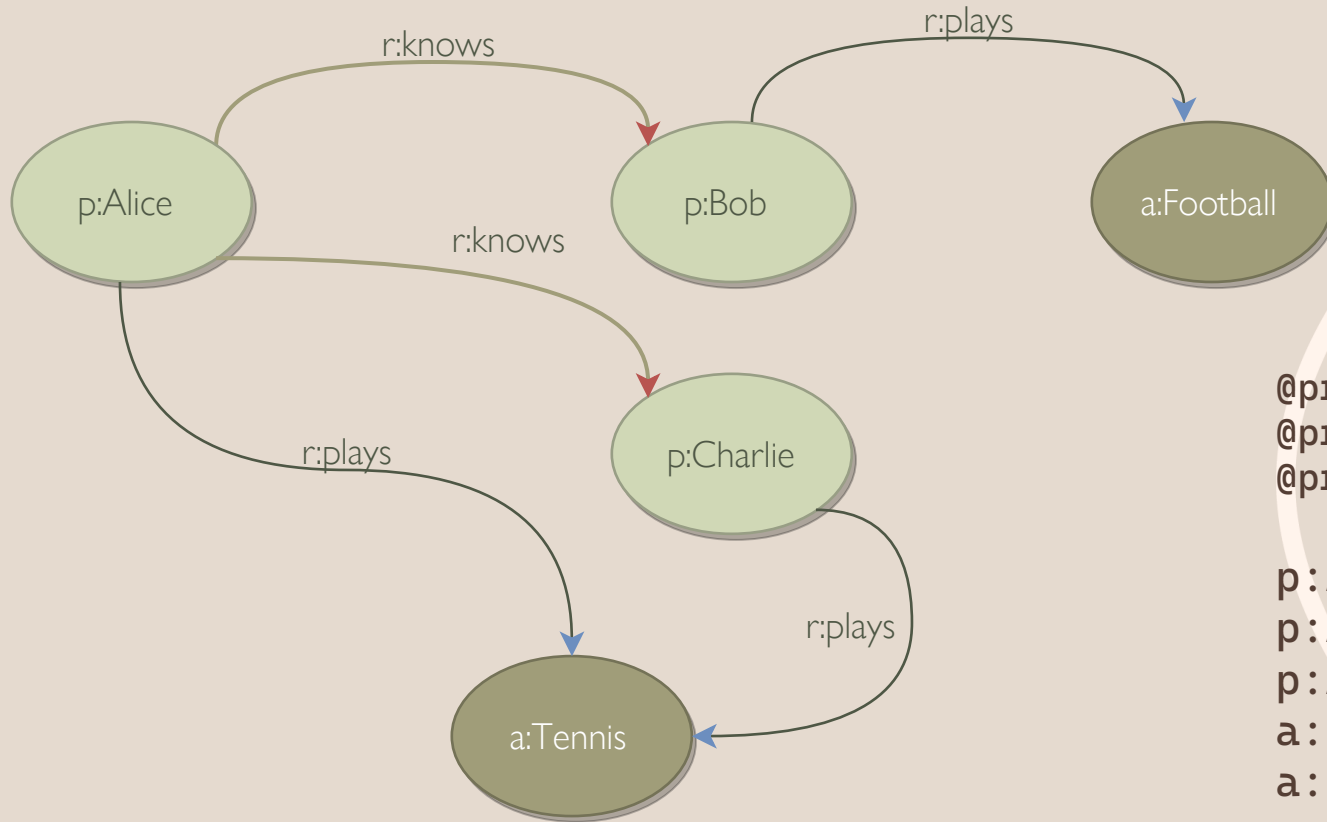


```
<Alice> <knows> <Bob> .  
<Alice> <knows> <Charlie> .  
<Alice> <plays> <Tennis> .  
<Bob> <knows> <Charlie> .  
<Bob> <plays> <Football> .  
<Charlie> <plays> <Tennis> .
```

<http://www.w3.org/TR/rdf11-primer/>

<https://commons.apache.org/proper/commons-rdf/introduction.html>

URI as identifier



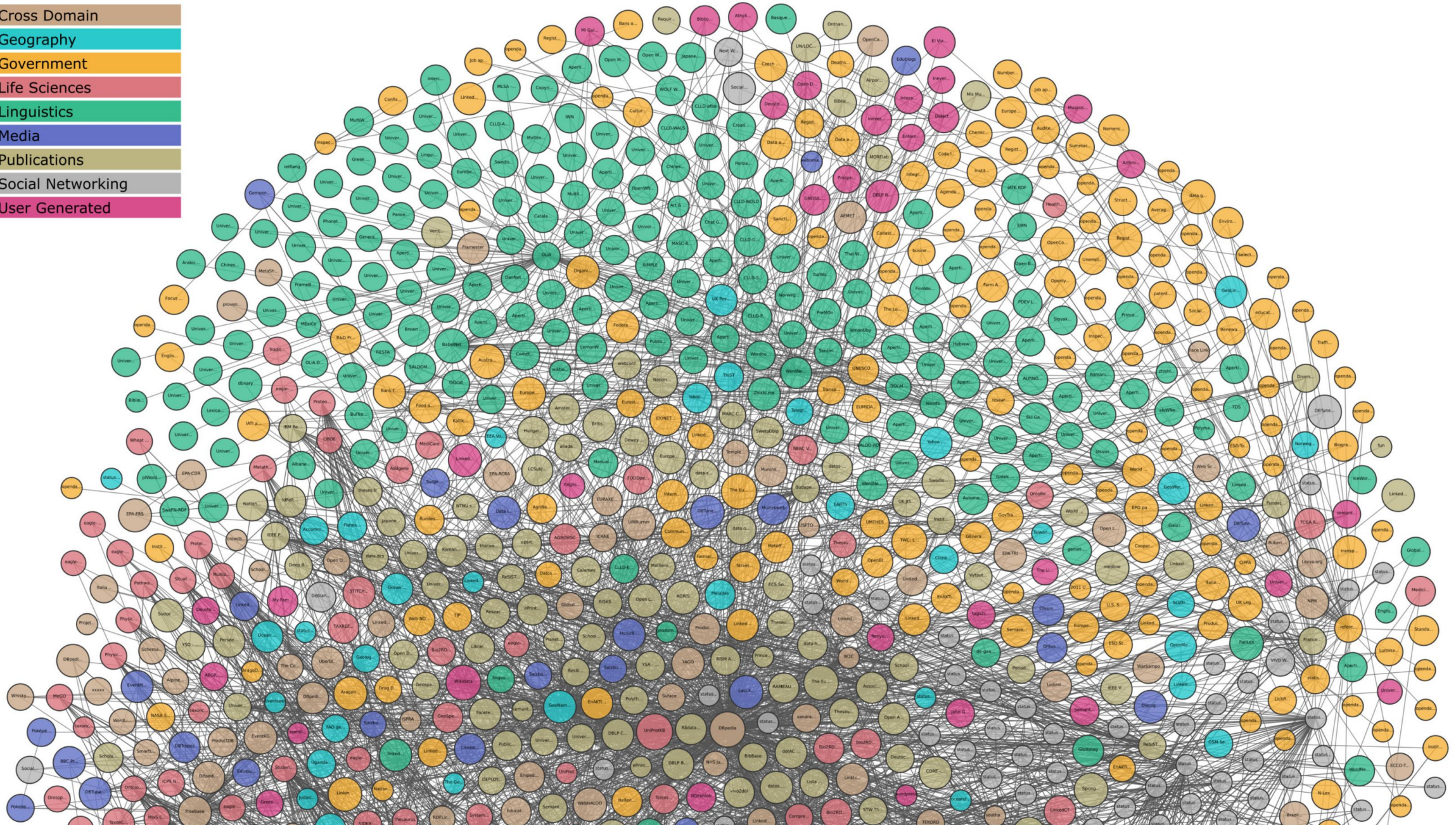
```
@prefix p: <http://example.com/people/>  
@prefix r: <http://example.com/relations/>  
@prefix a: <http://example.org/activities/>
```

```
p:Alice r:knows p:Bob .  
p:Alice r:knows p:Charlie .  
p:Alice r:plays a:Tennis .  
a:Bob r:knows p:Charlie .  
a:Bob r:plays a:Football .  
p:Charlie r:plays a:Tennis .
```

The Linked Open Data Cloud

<https://lod-cloud.net/>

Legend



TURNING LINKED DATA INTO KNOWLEDGE GRAPHS

Mature Linked Data tooling available
Powerful analytics and querying
Merging of graphs
Mappings and import/export
Ontology

<http://www.w3.org/TR/sparql11-overview/>
Adapted from
<https://euclid-project.eu/modules/chapter2.html>
<https://neo4j.com/labs/neosemantics/4.1/import/>

The image shows a SPARQL Query interface with a query editor and a results table. Below the query editor, there is a code block for an RDF import statement and a text block explaining the import. At the bottom, a Neo4j graph visualization shows a central node 'neo4j' connected to three other nodes: 'Neo4j-GraphQL', 'NSMNTX', and 'APOC'.

```
1 PREFIX dbpedia: <http://dbpedia.org/resource/>
2 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
3 PREFIX mo: <http://purl.org/ontology/mo/>
4 PREFIX dc: <http://purl.org/dc/elements/1.1/>
5
6 SELECT ?album ?title
7 WHERE { dbpedia:The_Beatles foaf:made ?album .
8         ?album a mo:SignalGroup .
9         ?album dc:title ?title }
```

Results for your query (100 of ...)

album	title
http://musicbrainz.org/signal-group/2ebf0c97-6ccb-3403-93...	Ultra Rare Trax, Volume 3
http://musicbrainz.org/signal-group/e10467b1-2a90-3923-ae...	Rarities
http://musicbrainz.org/signal-group/928aa116-1489-4929-8b...	The Beatles Collection, Volume 4: Revolver / Let It Be
http://musicbrainz.org/signal-group/d5d3c944-daf9-3224-91...	1962 Live Recordings
http://musicbrainz.org/signal-group/77c4cc64-cece-3458-b0...	Rare Photos & Interview CD, Volume 3
http://musicbrainz.org/signal-group/7b299b16-92ab-362a-b6...	The Recording Sessions, Volume 3
http://musicbrainz.org/signal-group/d9ea0025-57aa-3c94-b8...	I'll Cry Instead

```
CALL n10s.rdf.import.fetch("https://github.com/neo4j-labs/neosemantics/raw/3.5/docs/rdf/nsmntx.ttl", "Turtle");
```

In this case, the imported graph will look something like this, in which the names for labels, properties and relationships are more of the kind you're use to work with in Neo4j:

```
graph LR
  neo4j((neo4j)) -- runsOn --> Neo4jGraphQL((Neo4j-GraphQL))
  neo4j -- runsOn --> NSMNTX((NSMNTX))
  neo4j -- runsOn --> APOC((APOC))
```

Linked Data is familiar to Web Developers

Schema.org



Welcome to Schema.org

Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.

Schema.org vocabulary can be used with many different encodings, including RDFa, Microdata and JSON-LD. These vocabularies cover entities, relationships between entities and actions, and can easily be extended through a well-documented extension model. Over 10 million sites use Schema.org to markup their web pages and email messages. Many applications from Google, Microsoft, Pinterest, Yandex and others already use these vocabularies to power rich, extensible experiences.

Founded by Google, Microsoft, Yahoo and Yandex, Schema.org vocabularies are developed by an open community process, using the public-schemaorg@w3.org mailing list and through GitHub.

<https://schema.org/>

Trends:

Moving away from rigid semantics, more light-weight use

Example-driven documentation

Linking to Web pages with embedded metadata

<https://w3techs.com/technologies/details/da-jsonld>

W³Techs
Web Technology Surveys

advertise here

Home Technologies Reports Sites Quality Users Blog Forum

Technologies

- Content Management
- Server-side Languages
- Client-side Languages
- JavaScript Libraries
- CSS Frameworks
- Web Servers
- Web Panels
- Operating Systems
- Web Hosting
- Data Centers
- Reverse Proxies
- DNS Servers
- Email Servers
- SSL Certificate Authorities
- Content Delivery
- Traffic Analysis Tools
- Advertising Networks
- Tag Managers
- Social Widgets
- Site Elements
- Structured Data
- Markup Languages
- Character Encodings
- Image File Formats
- Top Level Domains
- Server Locations
- Content Languages

Trends

Technology Changes

Comparison

Compare with other

Technologies > Structured Data > JSON-LD

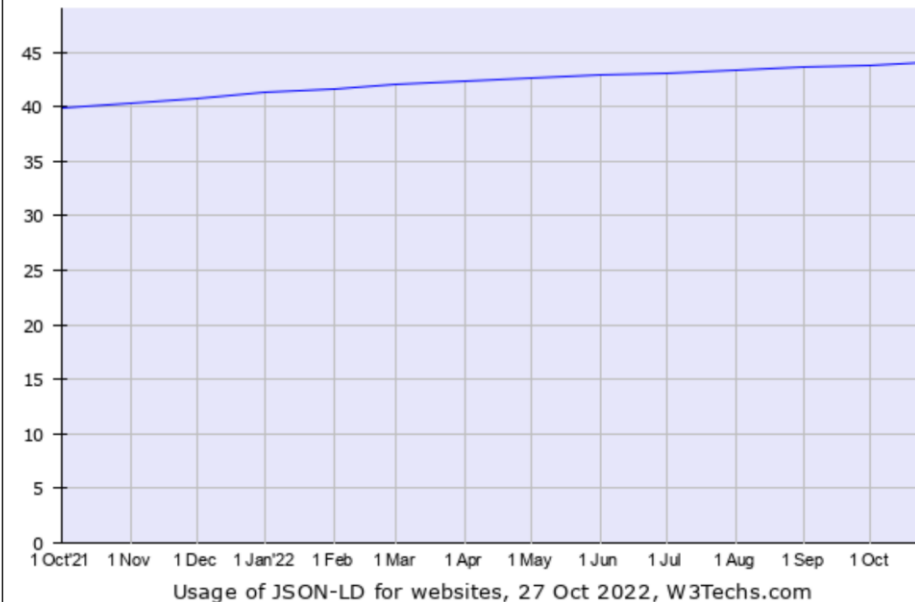
Usage statistics of JSON-LD for websites

These diagrams show the usage statistics of JSON-LD as structured data format on the web. See [technologies overview](#) for explanations on the methodologies used in the surveys. Our reports are updated daily.

JSON-LD is used by 44.1% of all the websites.

Historical trend

This diagram shows the historical trend in the percentage of websites using JSON-LD. Our dedicated trend survey shows more [structured data usage trends](#).



Implementing FDO using RO-Crate

Infrastructure independent –

avoiding repository/service silos

Practical, lightweight, robust

Familiar, developer friendly,

web-native, machine- and human-readable, search engine accessible

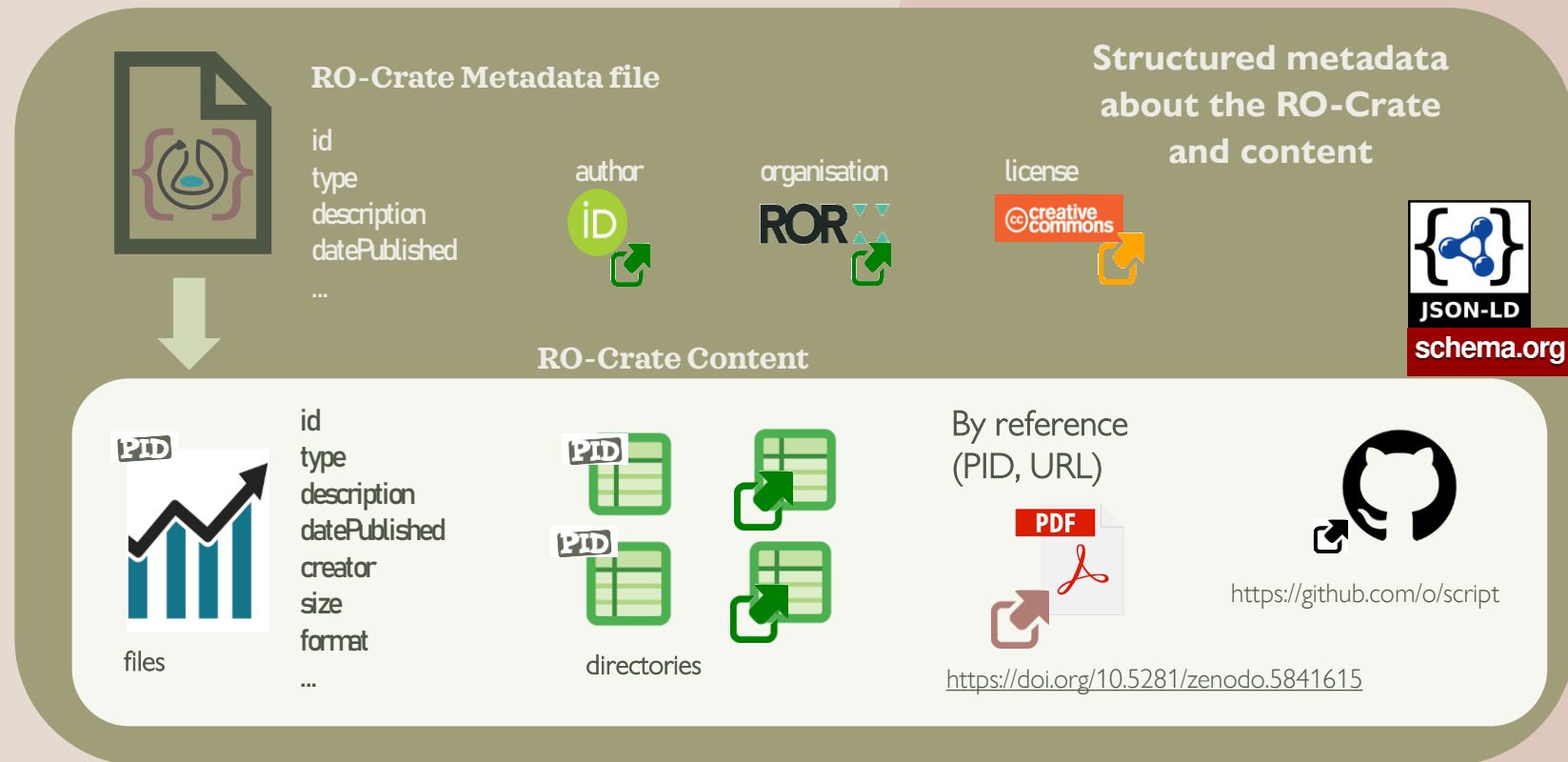
Adoptable Linked Data JSON and PIDs

Embrace diversity, legacy,

unknowns, open-ended, multi-

interpretation, self-describing, interlingua

Adoptable Metadata Profiles



<https://doi.org/10.3233/DS-210053>

<https://www.researchobject.org/ro-crate/>



A stylized, dark grey leaf graphic with several pointed leaves, positioned in the upper left corner of the slide.

FAIR
Signposting

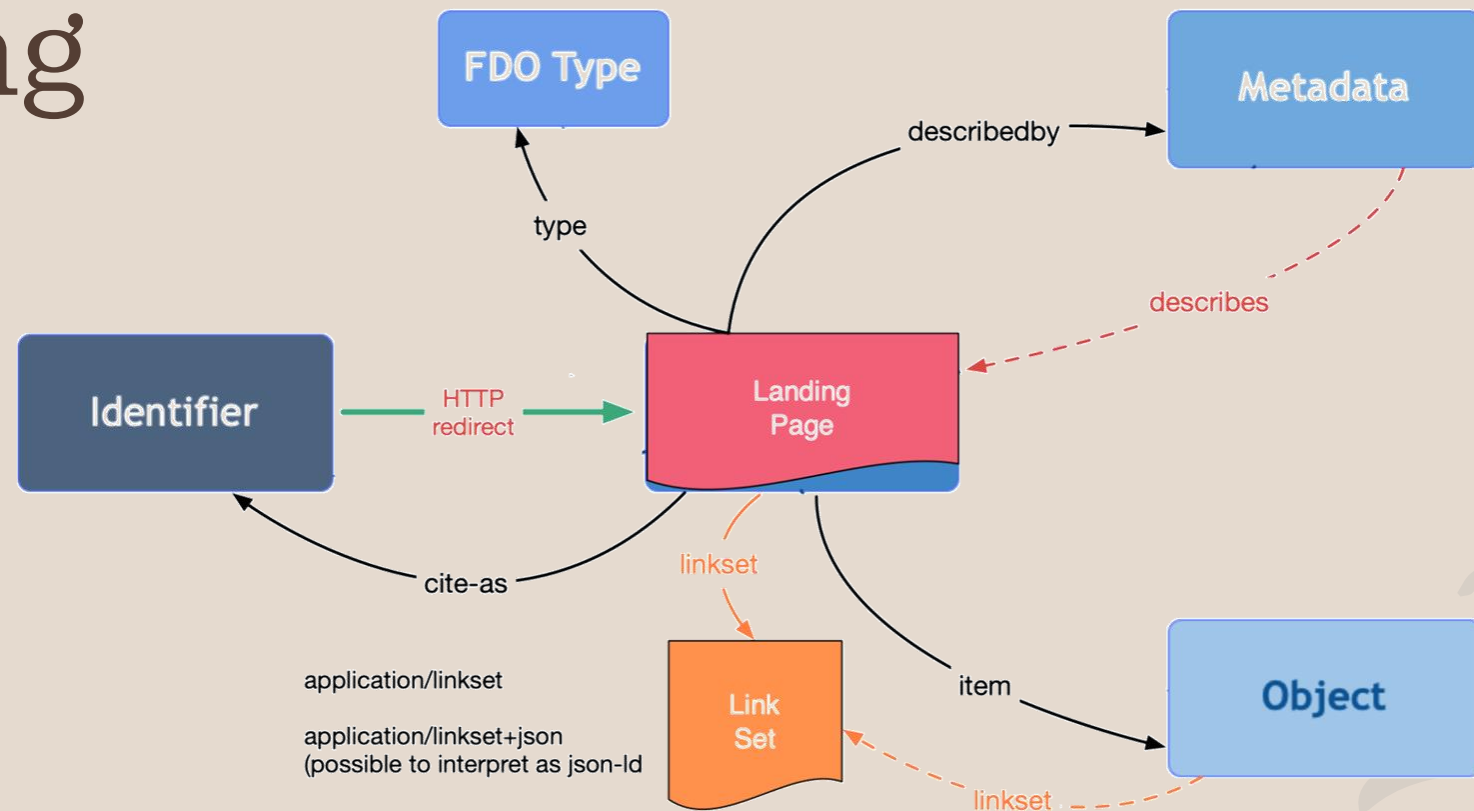
Decorative white lines consisting of overlapping circles and arcs, located in the bottom left corner of the slide.

FAIR Signposting

1.3. Typed Links in the FAIR Signposting Profile

The Relation Types that are used for the FAIR Signposting Profile as a means to meaningfully interlink resources that represent a scholarly artifact on the web are shown in the below table. The general description of their meaning is based on the more formal language used in the specification that define them. Their specific use for the FAIR Signposting Profile is provided in the descriptions of [Level 1](#) and [Level 2](#), below.

Relation Type	Description
<code>author</code>	The target of the link is a URI for an author of the resource that is the origin of the link.
<code>cite-as</code>	The target of the link is a persistent URI for the resource that is the origin of the link.
<code>describedby</code>	The target of the link provides metadata that describes the resource that is the origin of the link.
<code>type</code>	The target of the link is the URI for a class of resources to which the resource that is the origin of the link belongs.
<code>license</code>	The target of the link is the URI of a license that applies to the resource that is the origin of the link.
<code>item</code>	The origin of the link is a collection of resources and the target of the link is a resource that belongs to that collection.



```
(a2a) stain@xena11:~$ signposting https://doi.org/10.48546/workflowhub.workflow.255.1
Signposting for https://workflowhub.eu/workflows/255?version=1
CiteAs: <https://doi.org/10.48546/workflowhub.workflow.255.1>
DescribedBy: <https://workflowhub.eu/workflows/255?version=1> application/ld+json
             <https://workflowhub.eu/workflows/255?version=1> application/vnd.datacite.datacite+xml
Item: <https://workflowhub.eu/workflows/255/ro_crate?version=1> application/zip
(a2a) stain@xena11:~$
```

• <https://signposting.org/FAIR/>

<https://pypi.org/project/signposting/>

What can Linked Data learn from FDO?

Use and indicate **Persistent** identifiers

.. make sure they **resolve!** *No heuristics, please*

Always make **types** explicit

.. and indicate **type system** used

Make Linked Data active objects – Link to API **operations**

What can FDO learn from LD?

Keep it **simple**

Encourage **organic growth**

Legacy systems won't go away (easily)

Consider **consumption** first (e.g. querying)

Exploit **existing tooling**

Establish **best practices** and **shared vocabularies**



Thank you !

