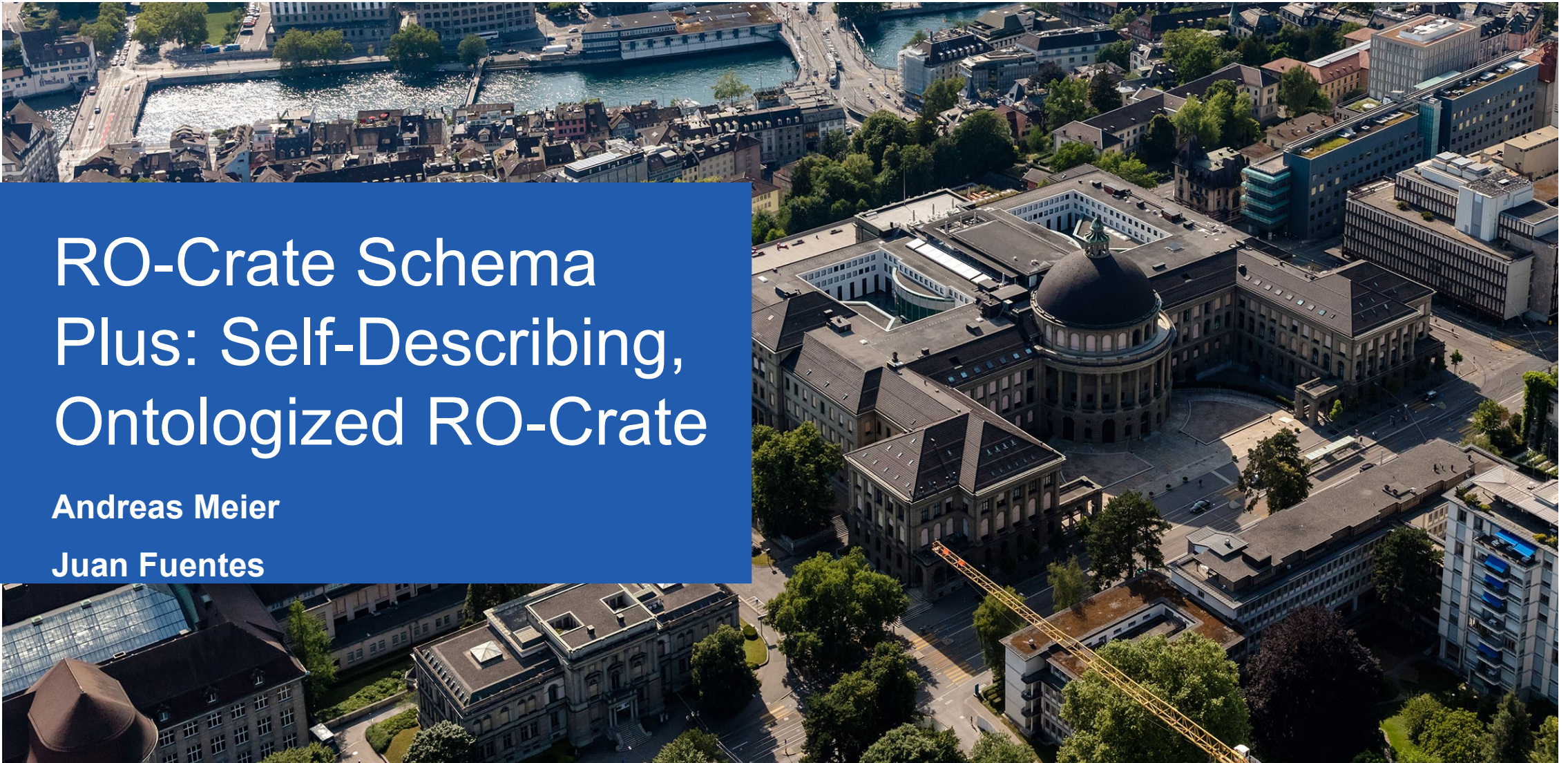


RO-Crate Schema Plus: Self-Describing, Ontologized RO-Crate

Andreas Meier

Juan Fuentes



Background

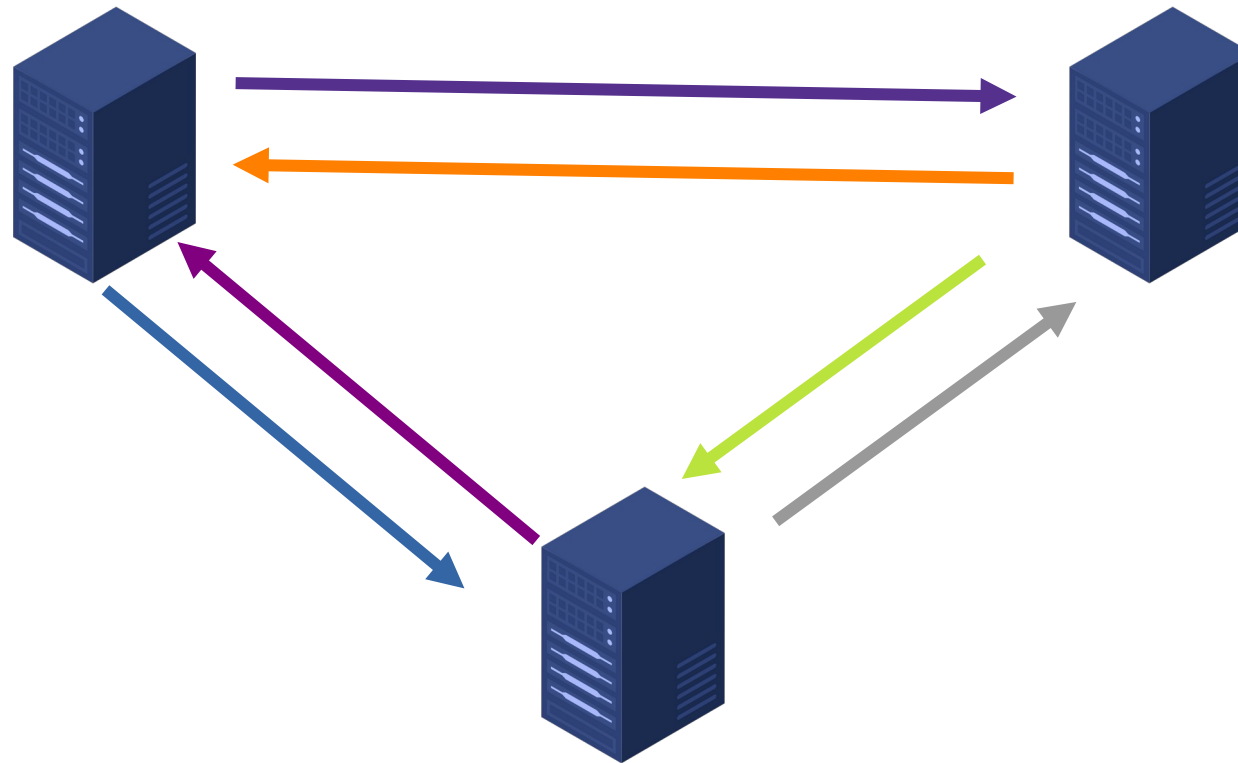
- Ongoing interoperability projects in the ETH domain, particularly [ETH ORD Measure 2](#), a domain-wide effort to improve interoperability in the ETH research data management ecosystem
- Deals with ELNs, LIMSs, data repositories and analysis software
- Objects graphs are of particular interest
- Object graph: linking of metadata entities to other metadata entities and datasets, keeping context intact

Goals

- One common, machine-to-machine interchange format containing:
 - Metadata
 - Schema describing the metadata
 - Ontological annotations
 - Bundled ontologies to guard against link rot

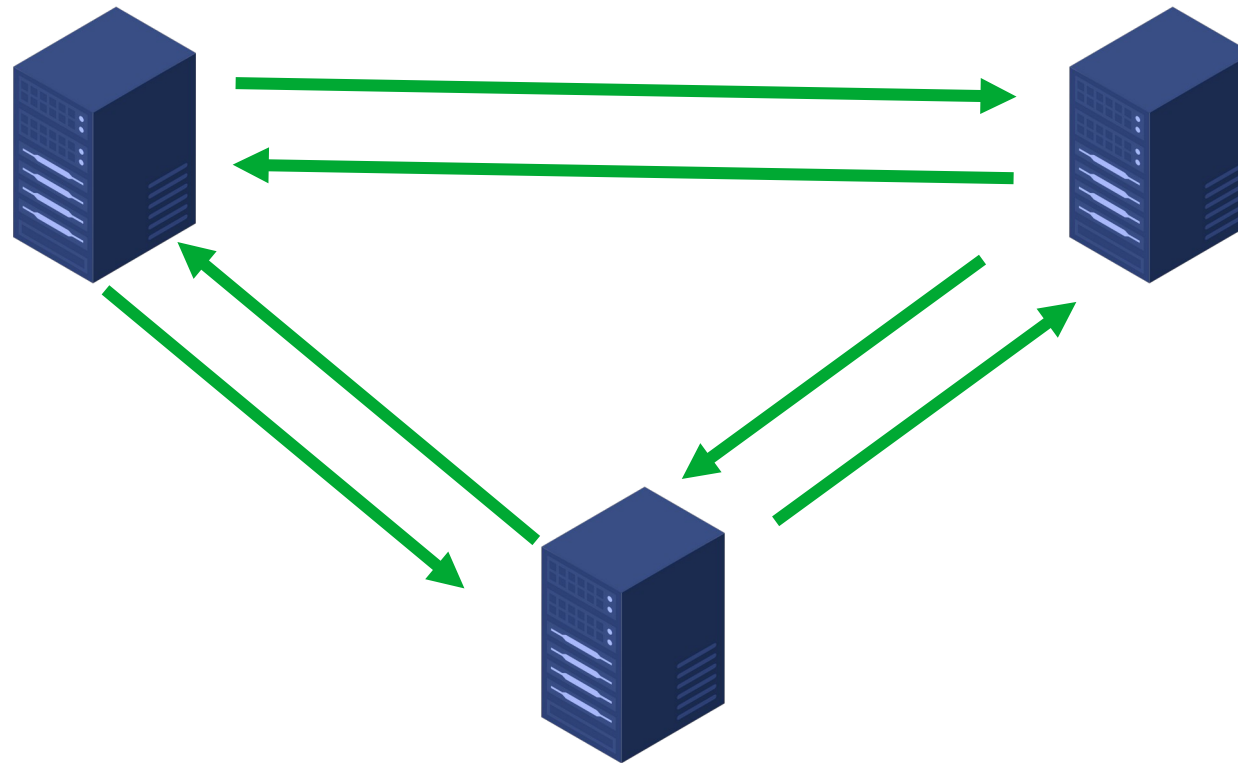
Number of integrations – Problem

- Exchanging metadata with RO-Crate often uses different exchange formats leading to repeated integrations: 3 systems → 6 integrations, 4 systems → 12 integrations, 5 systems → 20 integrations
- **Growth n^2-n ?**



Number of integrations – Idea

- Common metadata exchange format though the use of RO-Crate.



What is RO-Crate?

- Research Object Crate
- A zip/directory with metadata and research data
- **Manifest with JSON-LD** (JSON for linking data, graph format)
- Prefers schema.org types
- Otherwise freeform

<https://www.researchobject.org/ro-crate/specification/1.1/data-entities>

```
<RO-Crate root>/  
|  ro-crate-metadata.json  
|  cp7glop.ai  
|  lots_of_little_files/  
|    | file1  
|    | file2  
|    | ...  
|    | file54
```

An example *RO-Crate JSON-LD* for the above would be as follows:

```
{ "@context": "https://w3id.org/ro/crate/1.1/context",  
  "@graph": [  
    {  
      "@type": "CreativeWork",  
      "@id": "ro-crate-metadata.json",  
      "conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"},  
      "about": {"@id": "./"}  
    },  
    {  
      "@id": "./",  
      "@type": [  
        "Dataset"  
      ],  
      "hasPart": [  
        {  
          "@id": "cp7glop.ai"  
        },  
        {  
          "@id": "lots_of_little_files/"  
        }  
      ]  
    }  
  ]  
}
```

Why use RO-Crate?

- Research Object Crate is a common format
- It's gaining traction
- Community extensions (profiles) are possible

Limitations: Extending RO-Crate for interoperability

- RO-Crate cannot ship custom schemas for metadata
 - No standard way to ship metadata with schemas
 - No official libraries with support for this
- But RO-Crate allows the definition of profiles: Conventions used in a subset of RO-Crates
- Our work aims to provide a profile and software libraries to handle this use case

How do we express schemas and ontologies?

- We build on RDF (Resource Description Framework) technologies
- RDF can be expressed as JSON-LD
- Concretely:
 - RDFS (RDF Schema) for describing the schema
 - OWL (Web Ontology Language) for annotating types and properties
 - OWL for expressing cardinality information
- Our work aims to provide a profile and software libraries to handle this use case

How do we express schemas and ontologies? - Example

- RDFS Classes and RDFS Properties for expressing tabular data
OWL Annotation on these to express ontological annotation

Type			
	Property Type A	Property Type B	Property Type C
ENTRY A			
ENTRY B			
ENTRY C			

```
"@id" : "PUBLICATION_CREATOR",
"@type" : "rdfs:Class",
"owl:restriction" : [ {
  ...
} ],
"rdfs:subClassOf" : {
  "@id" : "openBIS:Object"
},
"owl:equivalentClass" : [ {
  "@id" : "http://datacite.org/schema/kernel-4#Creator"
}, {
  "@id" : "https://schema.org/Creator"
}, {
  "@id" : "https://schema.org/author"
} ]
},
```

Where can you find our work? - Libraries

We have a website: <https://researchobjectschema.github.io/ro-crate-schema-web/>

Here you can find links to the specifications and implementations of libraries

So far, there Java and Python libraries available as a preview

There's a SHACL validation for the manifest

The profile in use: openBIS RO-Crate API

openBIS is offering an RO-Crate API that handles RO-Crates following our schema

- export of RO-Crate
- validation of RO-Crate internals
- import of RO-Crate

This is currently in development for interoperability with various ETH Domain projects, most strongly Paul Scherrer Institute's SciCat and SciLog
The API is based on [an openAPI specification](#).

Schema.org types are handled! As long as the other types are described, schema.org are imported alongside them. This also holds for only using schema.org types.

Questions

Acknowledgment

This work is supported by the ORD Program of the ETH Domain.