Semantic Evaluation at Large Scale Tutorial

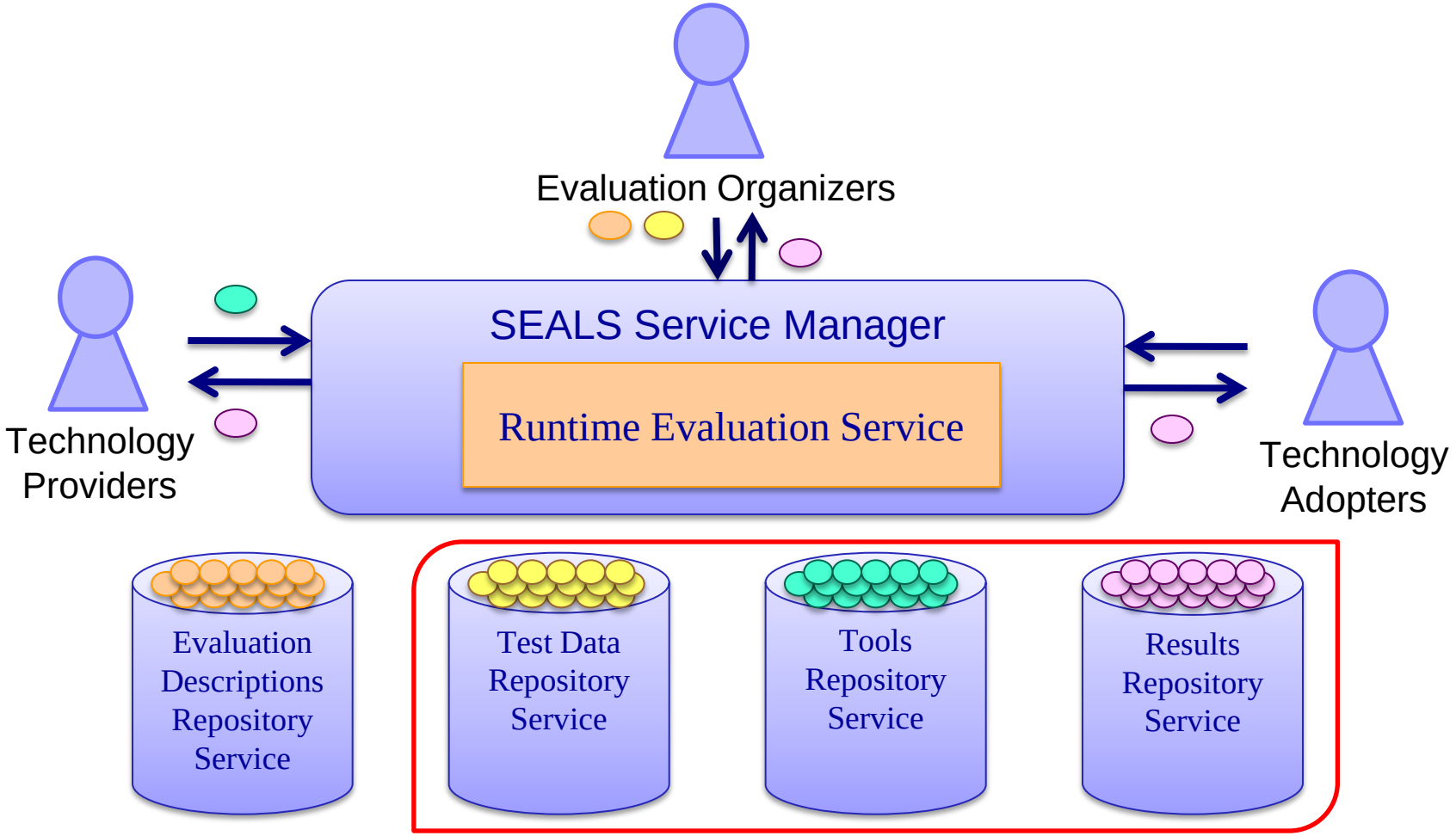
Repositories Management

Adrian Marte, STI Innsbruck
Michael Schneider, FZI Karlsruhe

SEALS Repositories

OVERVIEW

Overall Architecture



SEALS Repositories

- Test Data Repository Service
  - Persistently stores test data sets
  - Stores references to external test data sets
  - Stores synthetic test data generators
  - Generates synthetic test data
- Tools Repository Service
  - Stores the tools involved in an evaluation
- Results Repository Service
  - Stores the raw result of an evaluation
  - Stores interpretations over one or more raw results

SEALS Repositories

ARCHITECTURE

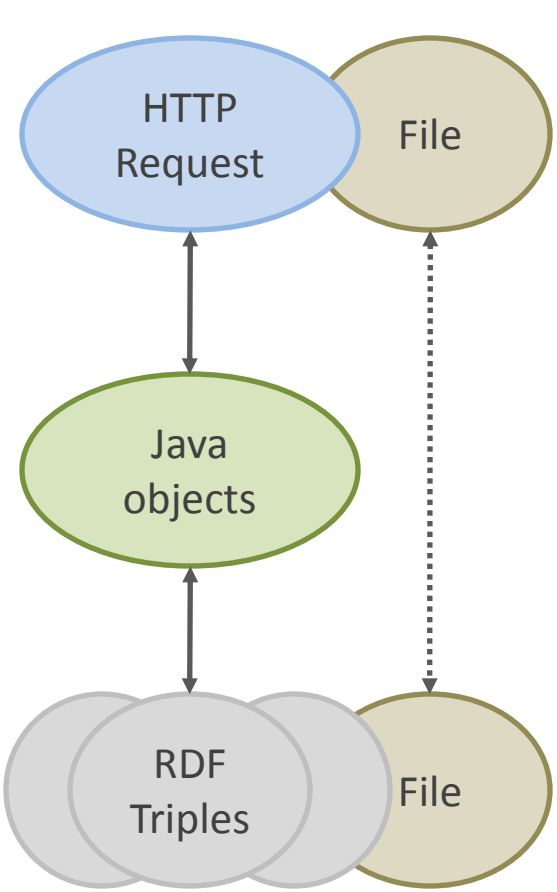
Artifacts and Artifact Versions

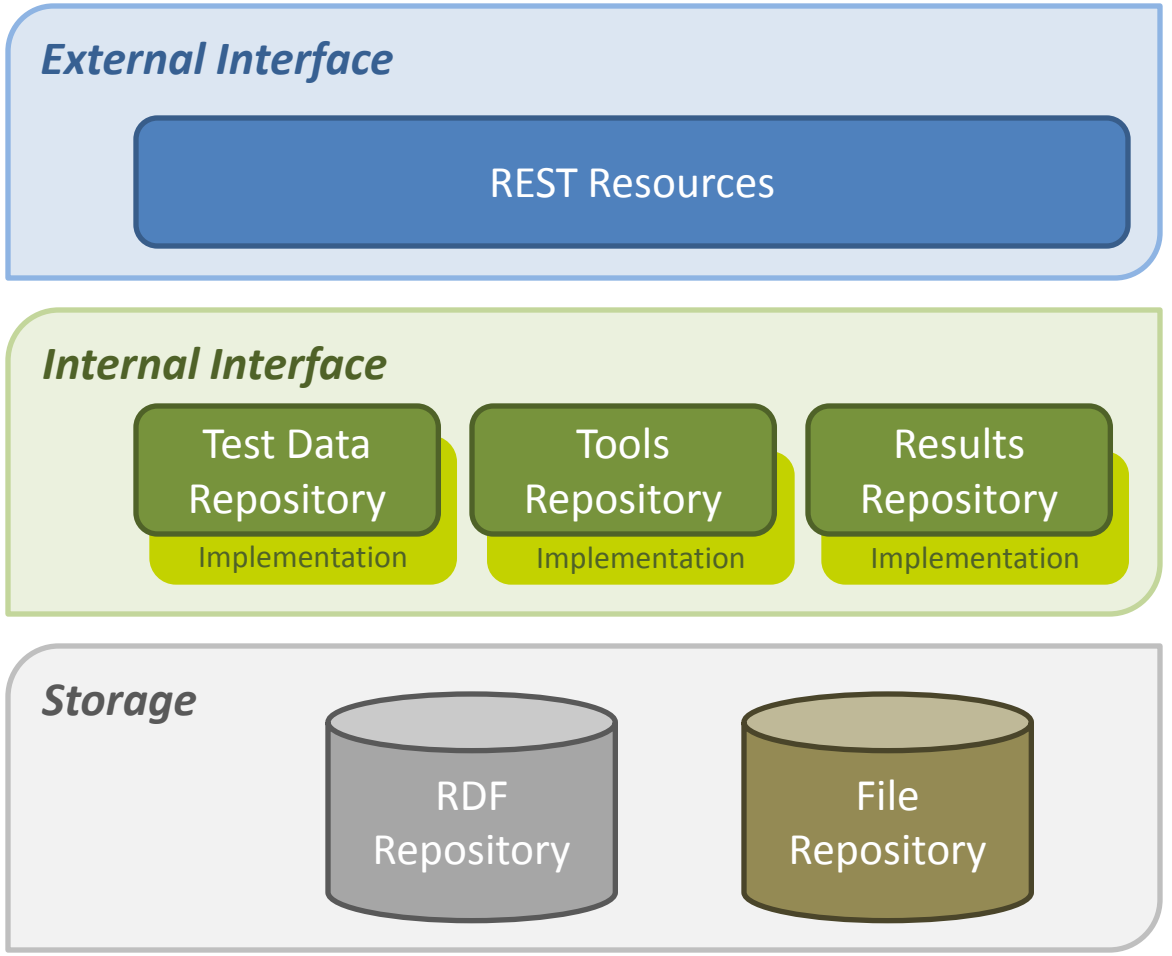
- Repositories store artifacts and artifact versions
  - Artifact
    - Usually a collection of versions
    - Usually does not have data associated with it
    - Test data collection, tool, result, interpretation, etc.
    - E.g., the tool “Protégé” is an artifact
  - Artifact version
    - Concrete version of an artifact
    - Comes with concrete data
    - E.g., the tool version “Protégé 4.1” is an artifact version

Metadata

- Artifacts and artifact versions can be described using the SEALS metadata ontology
- Repositories store data and metadata separately
- Used to identify, interlink and search for artifacts and artifact versions

Repositories Architecture



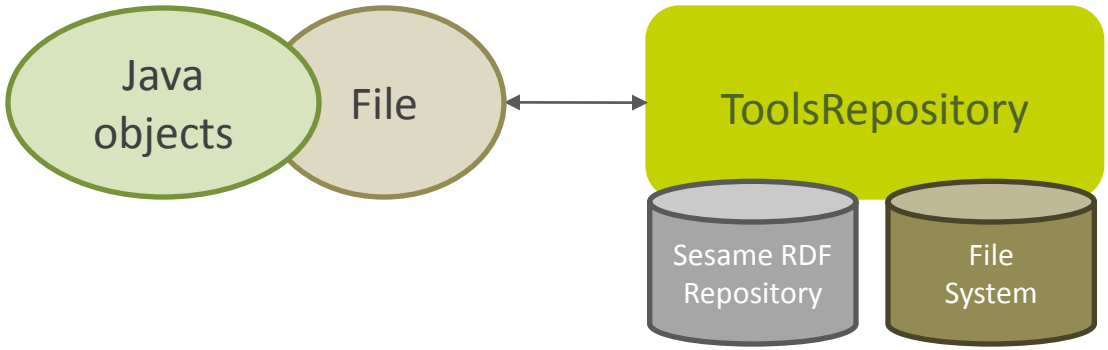


Storage Layer

- *File Repository*
  - Stores submitted tools, generators, test data, etc.
  - Uses underlying file system
  - Allows for storage of many large files
- *RDF Repository*
  - Stores RDF encoded metadata
  - Allows to pose SPARQL queries over metadata
  - Based on Sesame RDF repository
  - Can be accessed remotely over HTTP

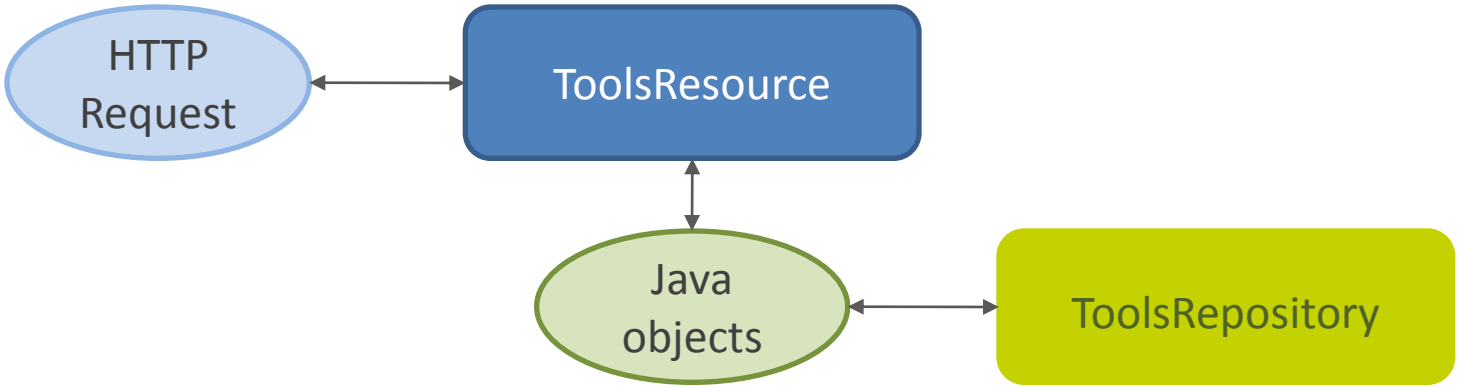
Internal Interface Layer

- Java interfaces for repositories
- Use underlying storage layer
- Allows repository functionality to be exposed using different protocols
- Allows other implementations that use different storage mechanisms



External Layer

- RESTful interface based on Restlet
  - Framework for RESTful applications
  - Open-source
- HTTP resources represent repository functionality
- Dispatch HTTP requests to corresponding repositories
- Use underlying repository interface implementations



SEALS Repositories

MANAGING ENTITIES

Describing Entities

- Metadata is RDF/XML encoded
- Submitted to repository when registering or updating entities in the repository
- Used to identify, interlink and search for entities stored in the repositories

Storing Entities

- Artifact along with its RDF/XML encoded metadata is registered in repository using a HTTP POST request
- Artifact version along with metadata and data (in the form of a ZIP file) is submitted to the resource of the artifact using a HTTP POST request
- GET, PUT, DELETE or POST can be used to retrieve, update, delete or add additional artifacts and artifact versions
- POST is also used to publish an artifact, in which case it is allowed to be used in an evaluation

Accessing Entities

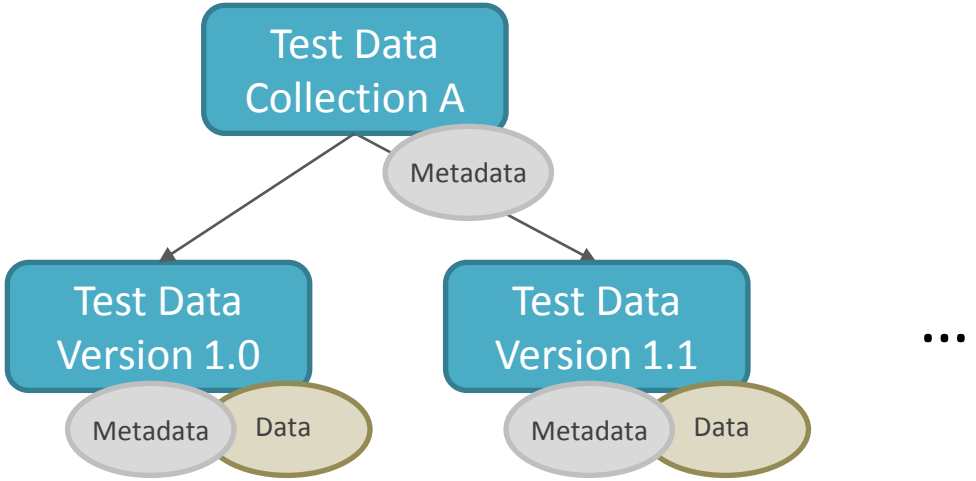
- Repositories create unique HTTP resources for submitted artifacts and artifact versions using the (auto-generated) `dcterms:identifier` value in the metadata
- Identifier is auto-generated by repository if not defined in the metadata
- Each repository defines URL schemes to access and manage the stored entities

SEALS Repositories

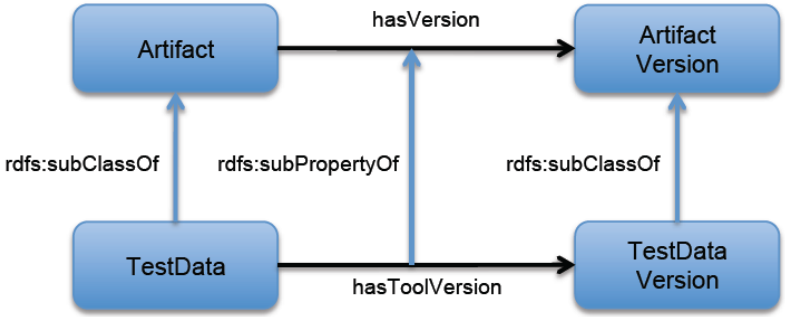
TEST DATA REPOSITORY SERVICE

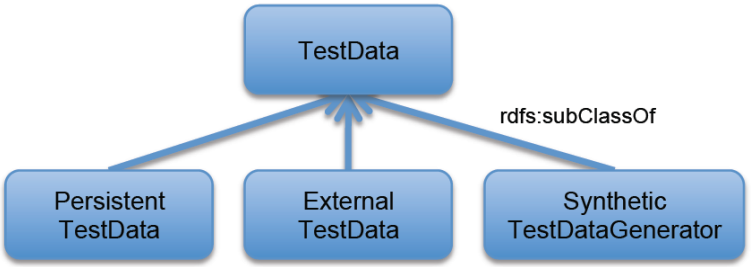
Test Data Repository Service

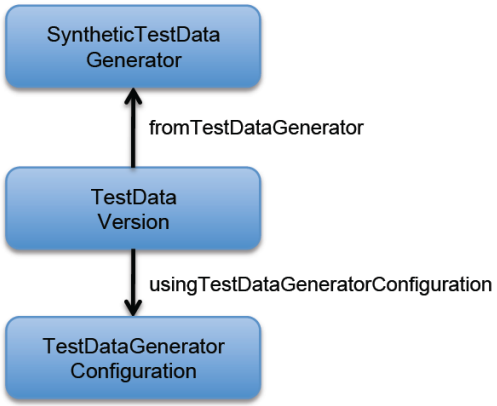
- Manages *test data collections* and *test data versions* used for evaluations
- Stores persistent test data (suites)
- Allows to reference and persist external test data
- Provides means for generating and storing synthetic test data using a user-submitted test data generator



Describing Test Data







Describing Test Data: Example

- Ontology matching test data: conference testsuite

```
<rdf:RDF xmlns:rdfs="..." xmlns:rdf="..." xmlns:dcterms="..."
         xmlns:seals="http://www.seals-project.eu/ontologies/SEALSMetadata.owl#">
    <seals:PersistentTestData>
        <seals:hasName rdf:datatype="&xsd;string">Conference Testsuite</seals:hasName>
        <dcterms:description rdf:datatype="&xsd;string">
                    The conference testsuite.
        </dcterms:description>
    </seals:PersistentTestData>
</rdf:RDF>
```

```
<rdf:RDF xmlns:rdfs="..." xmlns:rdf="..." xmlns:dcterms="..."
         xmlns:seals="http://www.seals-project.eu/ontologies/SEALSMetadata.owl#">
    <seals:TestDataVersion>
        <seals:hasVersionNumber rdf:datatype="&xsd;string">2010</seals:hasVersionNumber>
        <dcterms:description rdf:datatype="&xsd;string">
                    Version of the conference dataset used in 2010.
        </dcterms:description>
    </seals:PersistentTestData>
</rdf:RDF>
```

Accessing Test Data

URI	Op.	Accept	Effect
[domain]/testdata	GET	RDF	Search/browse
[domain]/testdata/persistent	POST	RDF	Register new persistent test data collection
[domain]/testdata/persistent/[collection]	GET	RDF	Retrieve metadata of test data collection
	GET	ZIP	Retrieve current test data version
	PUT	RDF	Update metadata of test data collection
	POST	RDF	Add test data version

Accessing Test Data

URI	Op.	Accept	Effect
[domain]/testdata/persistent/[collection]/[version]	GET	RDF	Retrieve metadata of test data version
	PUT	RDF	Update metadata of test data version
	DELETE		Remove test data version
	POST	RDF	Publish test data version
[domain]/testdata/persistent/[collection]/[version]/pub	GET	RDF	Retrieve metadata of test data version
	GET	ZIP	Retrieve data of publish test data version
	DELETE	ZIP	Unpublish test data version

Accessing Test Data

URI	Op.	Accept	Effect
[domain]/testdata/persistent/[collection] [version]/suite/	GET	RDF	Retrieve suite metadata
[domain]/testdata/persistent/[collection]/ [version]/suite/[suite item]/item/[data item]	GET		Retrieve data item
[domain]/testdata/persistent/[collection]/[version]/ suite/[suite item]/component/[component type]	GET		Retrieve component

Synthetic Test Data Generators

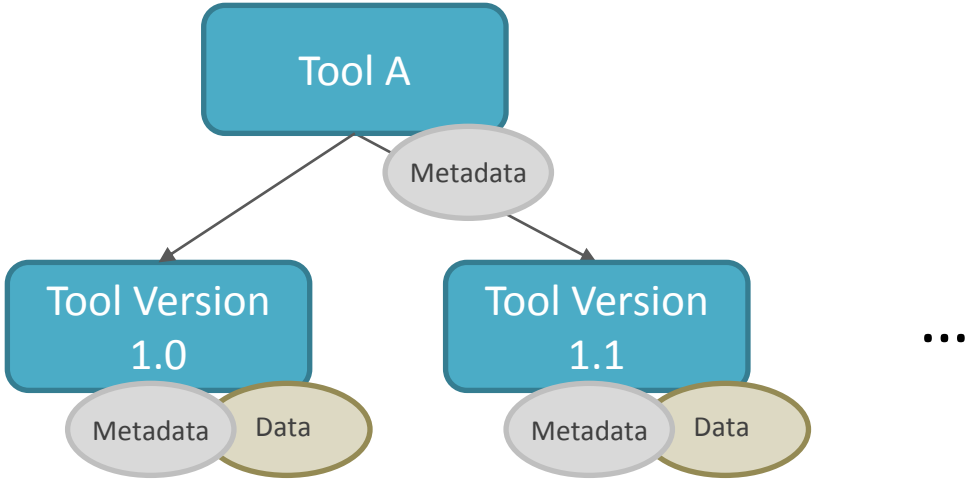
- Test Data Repository Service supports storage and on-the-fly generation of synthetic test data using a user-defined generator
- Synthetic Test Data Generator API enables implementation of a generator

SEALS Repositories

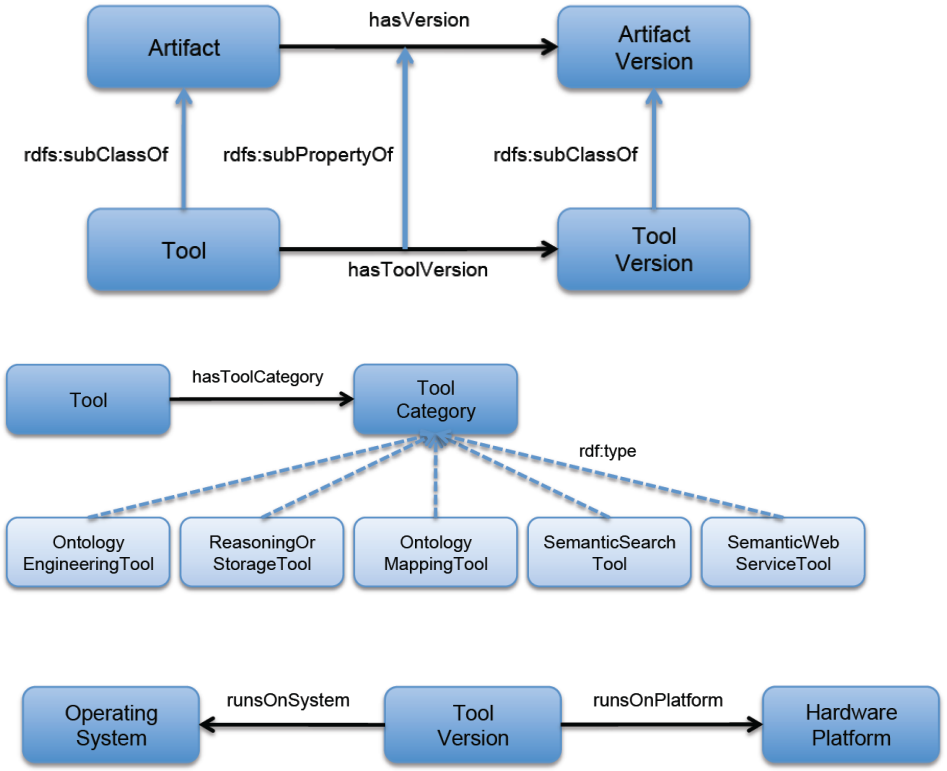
TOOLS REPOSITORY SERVICE

Tools Repository Service

- Manages *tools* and *tool version* that are evaluated on the platform
- Tools are expected to be packaged such that they can be integrated in an evaluation workflow



Describing Tools

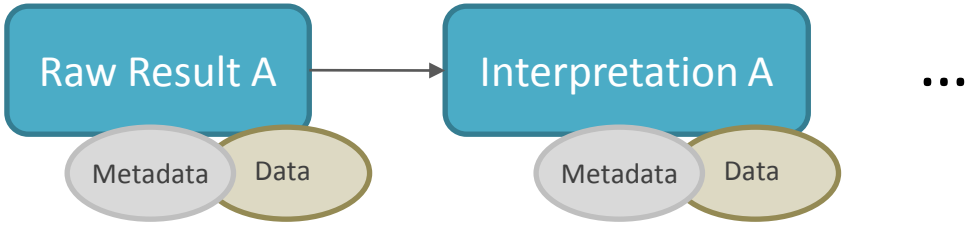


SEALS Repositories

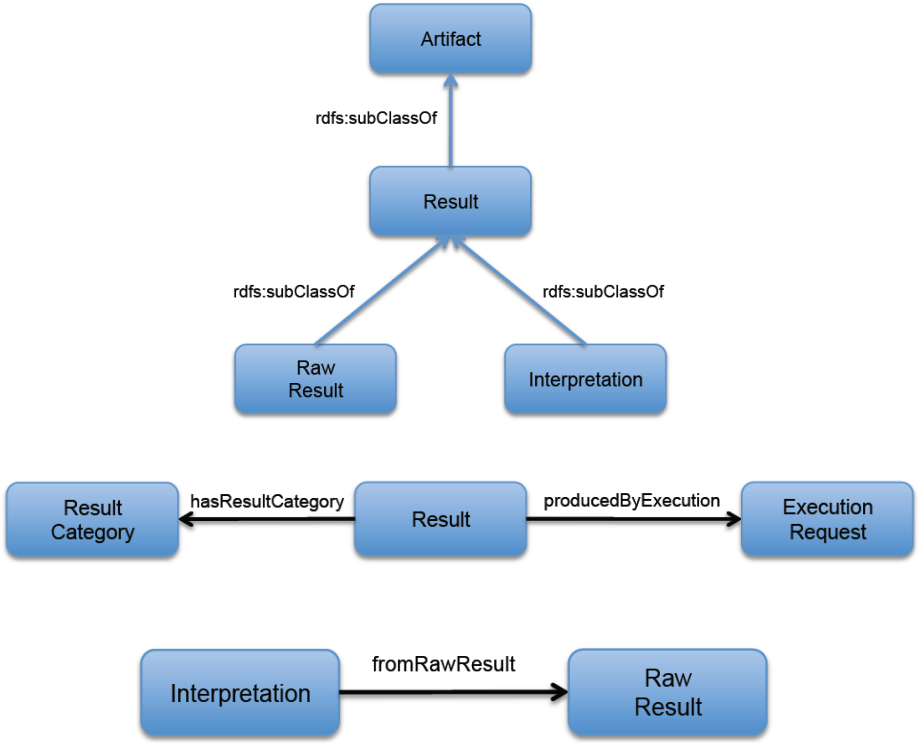
RESULTS REPOSITORY SERVICE

Results Repository Service

- Manages *raw results* and *interpretations* thereof that are created during an evaluation of a tool
- Can be described using suite ontology



Describing Results



THANK YOU!