Validate Graph Data with SHACL

SPHN DCC Training

Philip Krauss, Trivadis - Part of Accenture

Martin Zablocki, Dr.-Ing., Trivadis - Part of Accenture

PHI Data Interoperability Team, SIB Swiss Institute of Bioinformatics

A project of

SAMW ASSM
Schweizerische Akademie der Medizinischen Wissenschaften
Académie Suisse des Sciences Médicales
Accademia Svizzera delle Scienze Mediche
Swiss Academy of Medical Sciences

SIB
Swiss Institute of
Bioinformatics

Agenda

1. Why SHACL?
2. SHACL
3. Validation Report
4. Hands-On with GraphDB
5. SHACLer

POLL 1 (Single Choice)

What do you mainly intend to use SHACL for?

❑ Quality Assurance, ensure data is valid and consistent

❑ Exchange the structure of the data content

❑ Application Integration, where different software components need to function together smoothly

❑ Provide interoperability among independent services and tools

Why describe & validate RDF?

For RDF producers

Understand the contents which is going to be produce

Ensure the expected structure is produced

Ensures data to be valid and consistent

Lowers error rate by preventing bad data from entering the system

Harmonize the structure and meaning across departments and organizations

For RDF consumers

Understand the meaning of the content

Verify the structure before processing it

Query harmonized and well-connected data from different sources

Similar technologies

Technology	Schema
Relational Databases	DDL
XML	DTD, XML Schema, RelaxNG
JSON	JSON Schema
RDF	?

What can be used for RDF?

SHACL - **Sh**apes **C**onstraint **L**anguage

W3C recommendation since July 2017, www.w3.org/TR/shacl

A language for checking RDF Graphs against conditions

Supported by many RDF Triple Stores

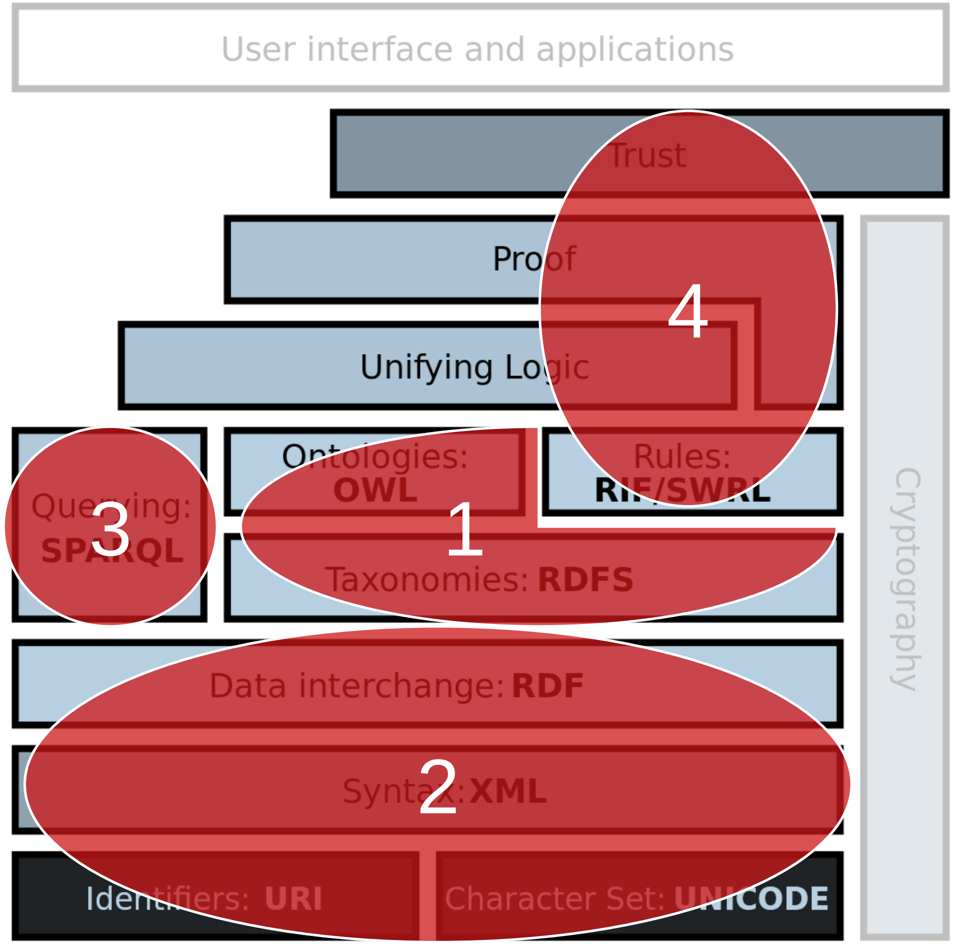
Like RDFS and OWL, SHACL constraints are written in RDF

SHACL Core processors support validation with the SHACL Core Language

SHACL-SPARQL processors support validation with the SHACL-SPARQL Language
(not covered in this tutorial)

Semantic Web Layer Cake

1. Ontologies
2. RDF
3. SPARQL
4. SHACL



Poll 2 (Multiple Choice)

What technologies of the Semantic Web Layer Cake did you use already?

□ Ontologies

□ RDF

□ SPARQL

□ SHACL

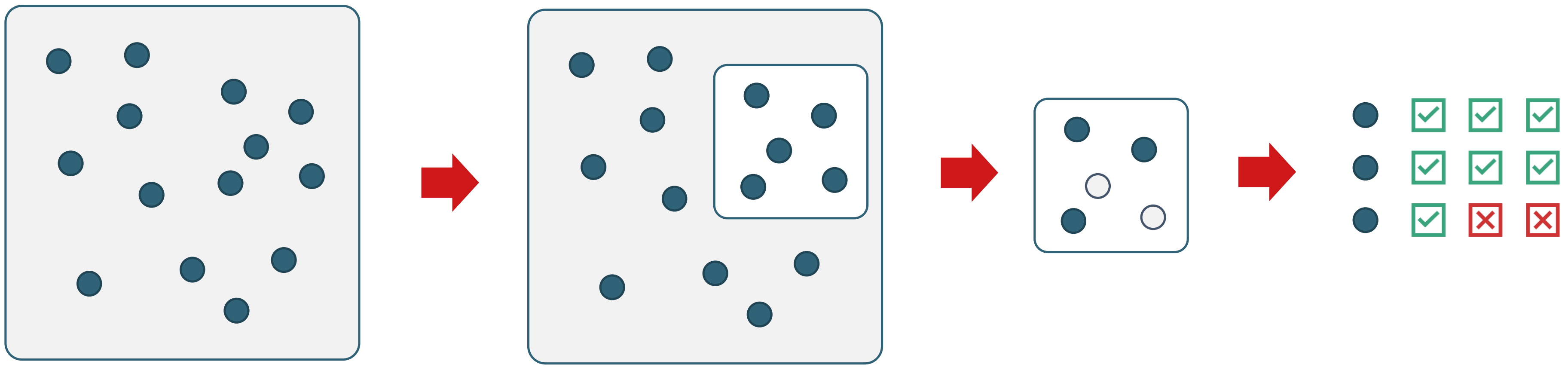
How does it work - Concept

Data Graph

Targets are used to select focus nodes

Filters are used to remove some focus nodes

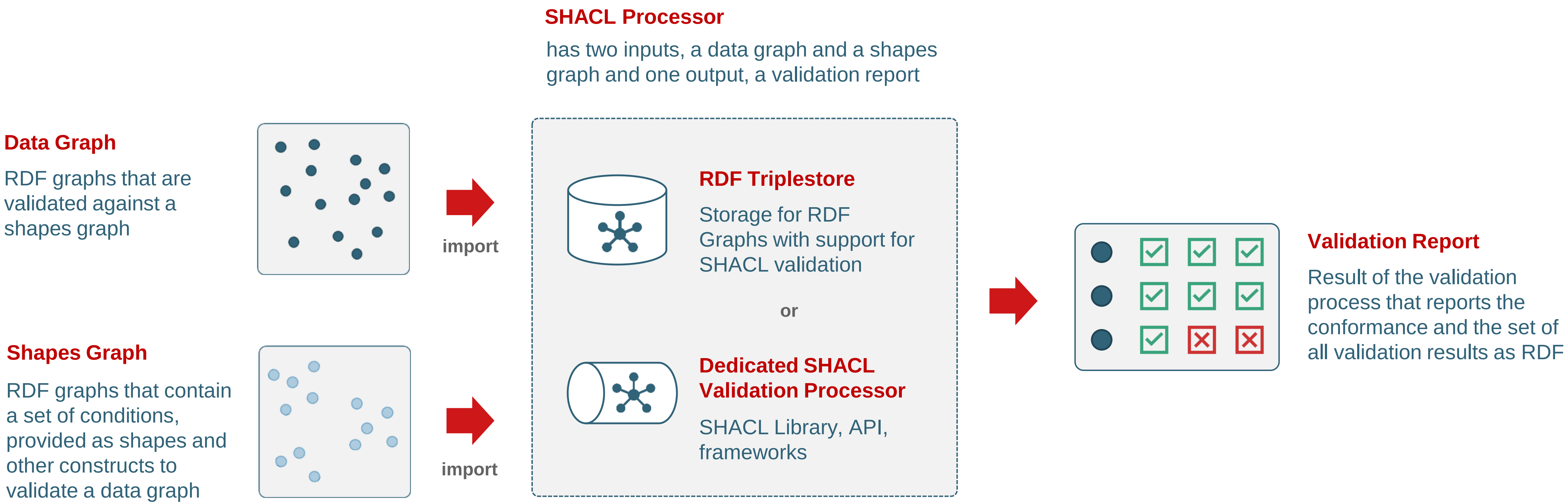
Constraints are checked to generate validation report



*Selection of targets are SHACL node Shape Targets, not subsampled data.

Filters are part of SHACL Advanced Features (SHACL-SPARQL specification)

How does it work - Technology



SHACL Processor

has two inputs, a data graph and a shapes graph and one output, a validation report

Data Graph

RDF graphs that are validated against a shapes graph

Shapes Graph

RDF graphs that contain a set of conditions, provided as shapes and other constructs to validate a data graph

RDF Triplestore

Storage for RDF Graphs with support for SHACL validation

Dedicated SHACL Validation Processor

SHACL Library, API, frameworks

Validation Report

Result of the validation process that reports the conformance and the set of all validation results as RDF

Targets, Shapes, and Constraints

SHACL Part 2

Namespace for SHACL

Within this document, the following namespace prefix bindings are used:

Prefix	Namespace
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
sh:	http://www.w3.org/ns/shacl#
xsd:	http://www.w3.org/2001/XMLSchema#
ex:	http://example.com/ns#
:	http://example.com/ns#

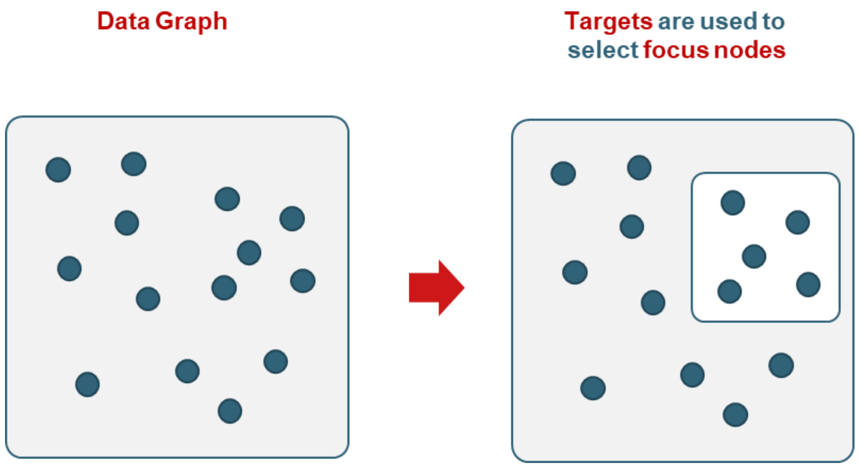
Targets in SHACL

Targets specify what RDF graph nodes are validated against a shape

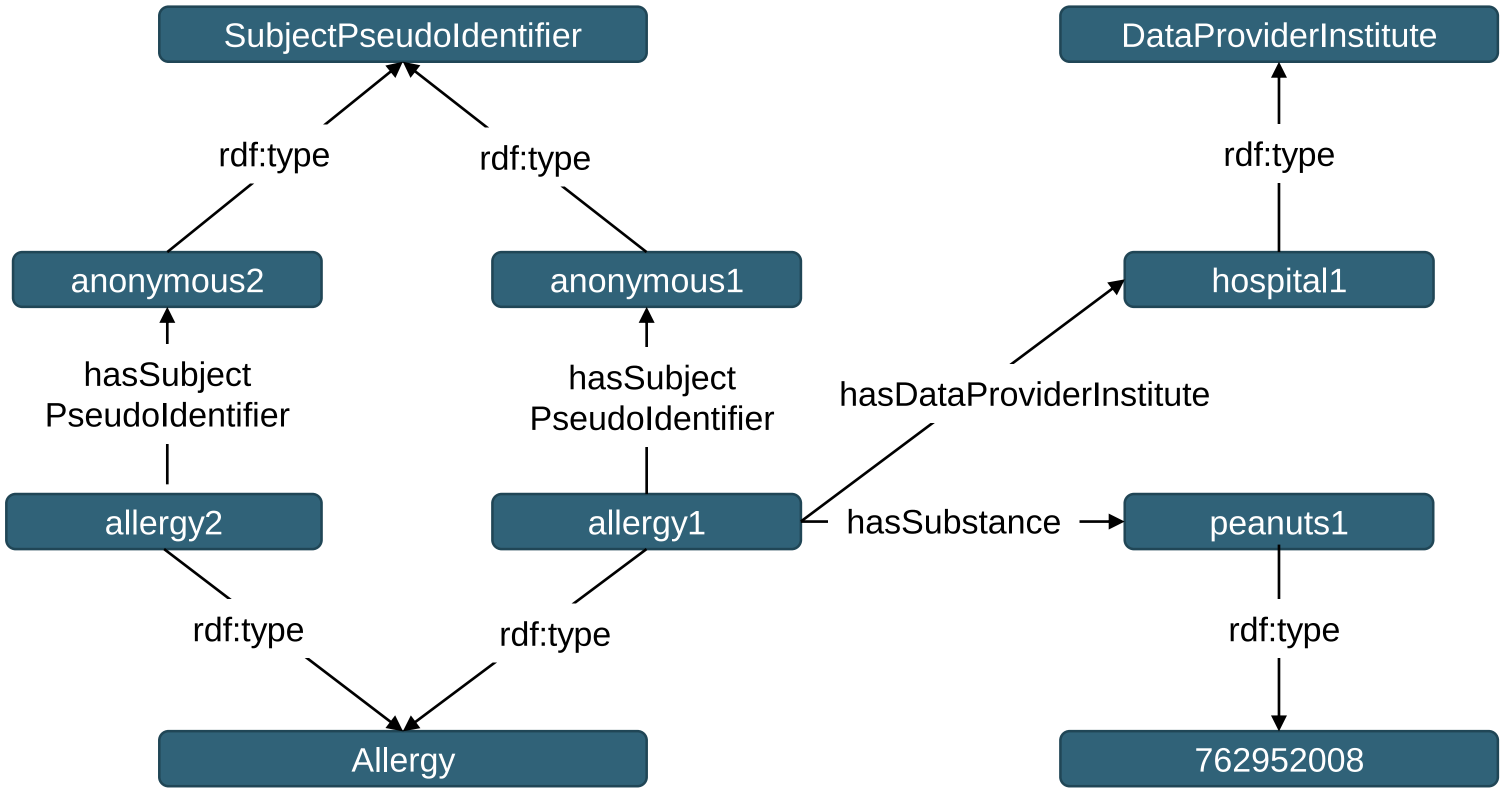
During validation, we refer to the targets as focus nodes

SHACL Core includes the following kinds of targets:

Value	Description
targetNode	Directly points to a specific node
targetClass	All nodes that have a given type
targetSubjectsOf	All subjects of a specific property
targetObjectsOf	All object of a specific property



Example

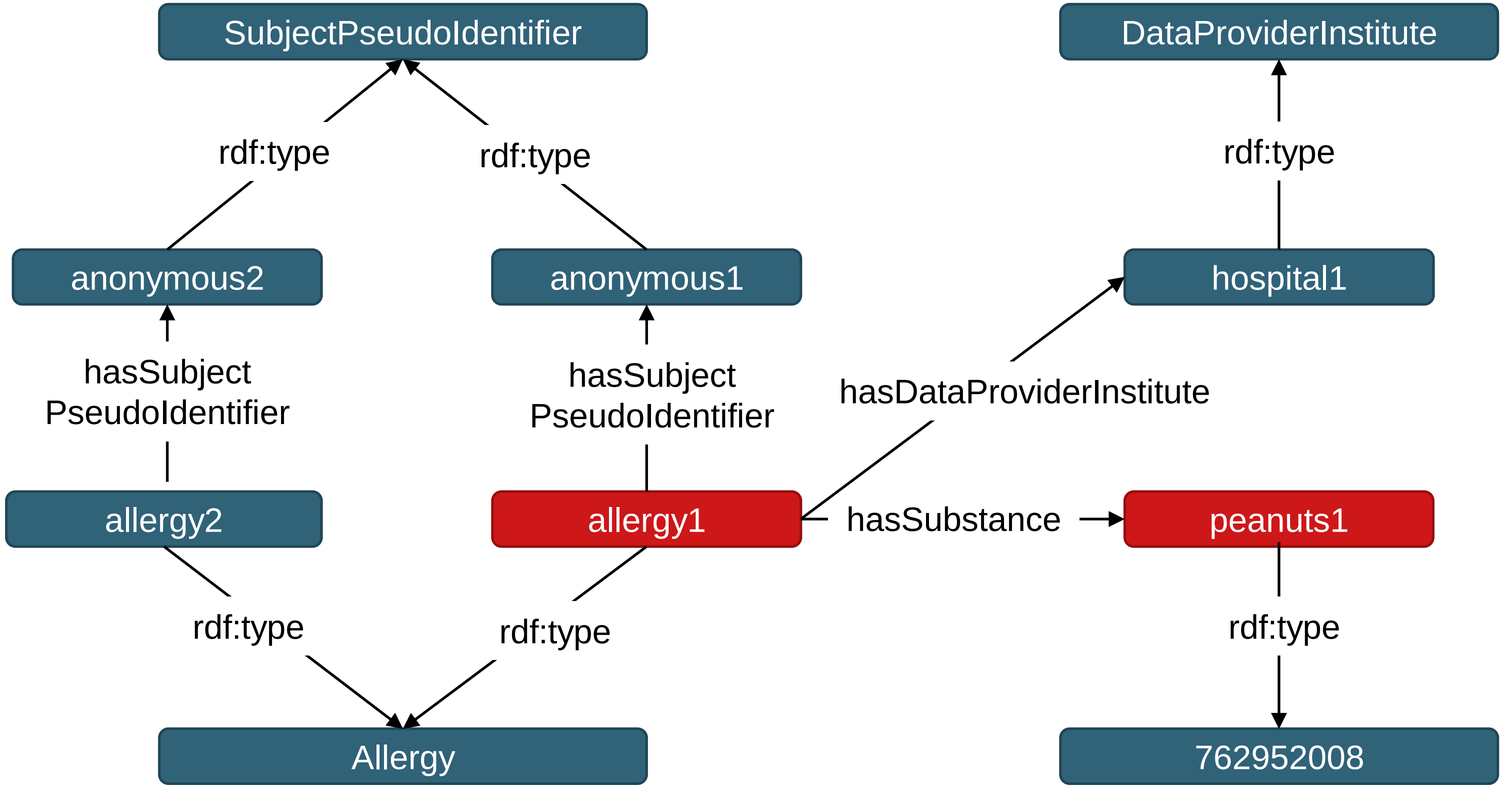


Namespaces have been omitted for a simplified representation

Target: sh:targetNode

SHACL Shape specifying two focus nodes, :allergy1 and :peanuts explicitly.

```
ex:SibShape1
   a sh:NodeShape ;
   sh:targetNode :allergy1,
                 :peanuts1 .
```

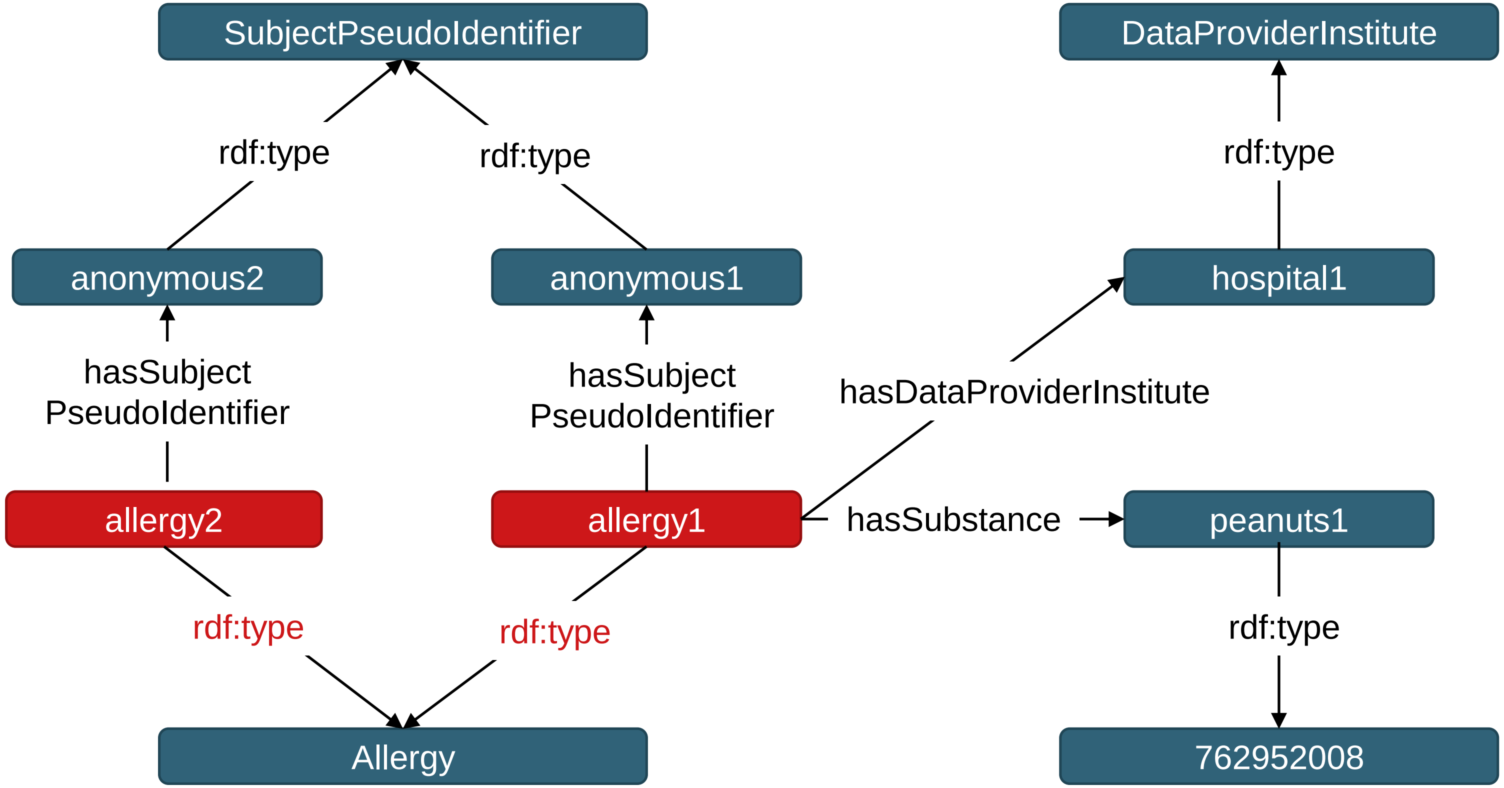


Namespaces have been omitted for a simplified representation

Target: sh:targetClass

SHACL Shape specifying all nodes as focus nodes, which are instances of the class :Allergy

```
ex:SibShape2
   a sh:NodeShape ;
   sh:targetClass :Allergy .
```



Namespaces have been omitted for a simplified representation

Shapes in SHACL

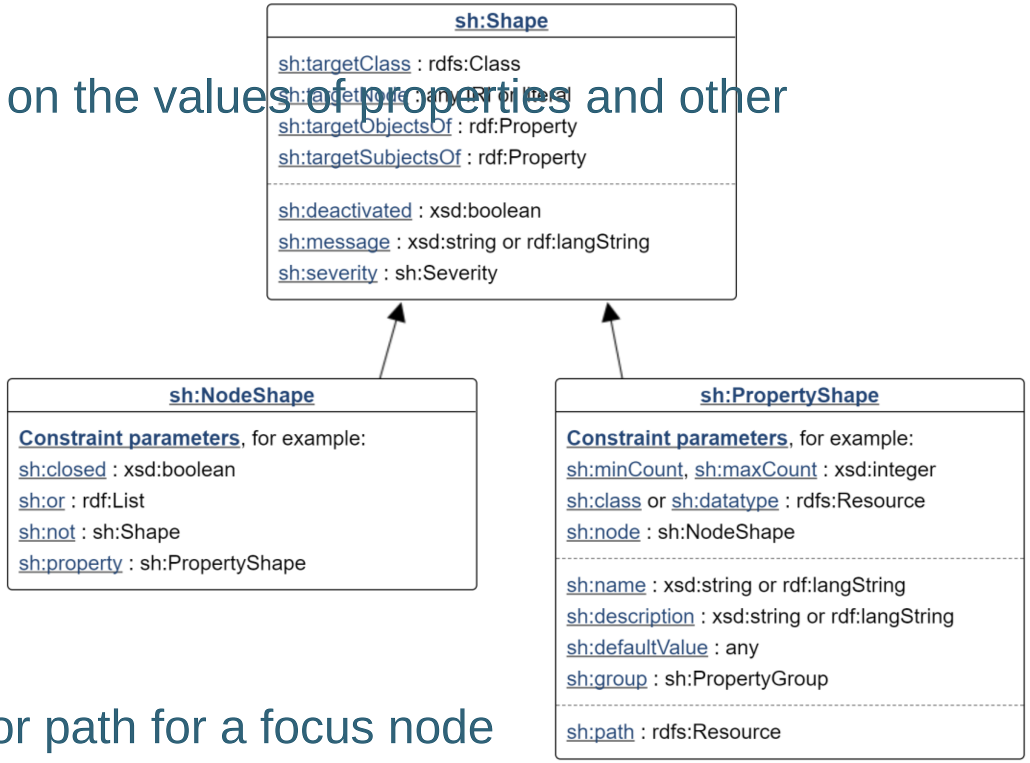
A shape determines how to validate a focus node based on the values of properties and other characteristics of the focus node

Node Shape

- Specifies constraints about the focus node itself
- Each focus node is validated individually,
  to check if its compliant with the node shape

Property Shapes

- Constraints about the value of a particular property or path for a focus node



sh:Shape

sh:targetClass : rdfs:Class
sh:targetNode : any IRI or literal
sh:targetObjectsOf : rdf:Property
sh:targetSubjectsOf : rdf:Property

sh:deactivated : xsd:boolean
sh:message : xsd:string or rdf:langString
sh:severity : sh:Severity

sh:NodeShape

Constraint parameters, for example:
sh:closed : xsd:boolean
sh:or : rdf:List
sh:not : sh:Shape
sh:property : sh:PropertyShape

sh:PropertyShape

Constraint parameters, for example:
sh:minCount, sh:maxCount : xsd:integer
sh:class or sh:datatype : rdfs:Resource
sh:node : sh:NodeShape

sh:name : xsd:string or rdf:langString
sh:description : xsd:string or rdf:langString
sh:defaultValue : any
sh:group : sh:PropertyGroup

sh:path : rdfs:Resource

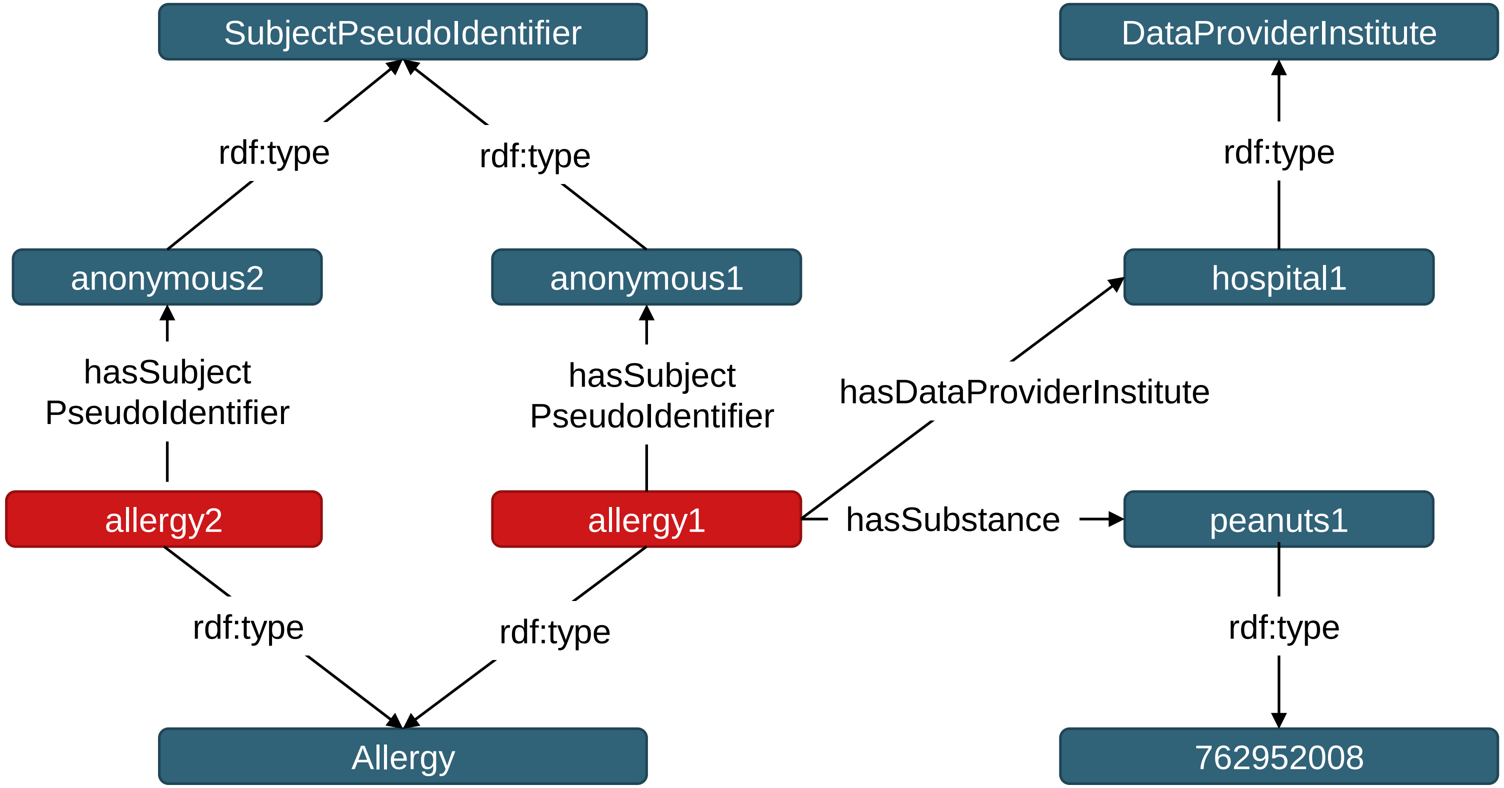
Source: https://www.w3.org/TR/shacl

sh:NodeShape

SHACL Shape specifying all nodes as focus nodes, which are instances of the class :Allergy.

The NodeShape requires the focus node to be a valid IRI.

```
ex:SibShape5
  a sh:NodeShape ;
  sh:targetClass :Allergy ;
  sh:nodeKind sh:IRI.
```



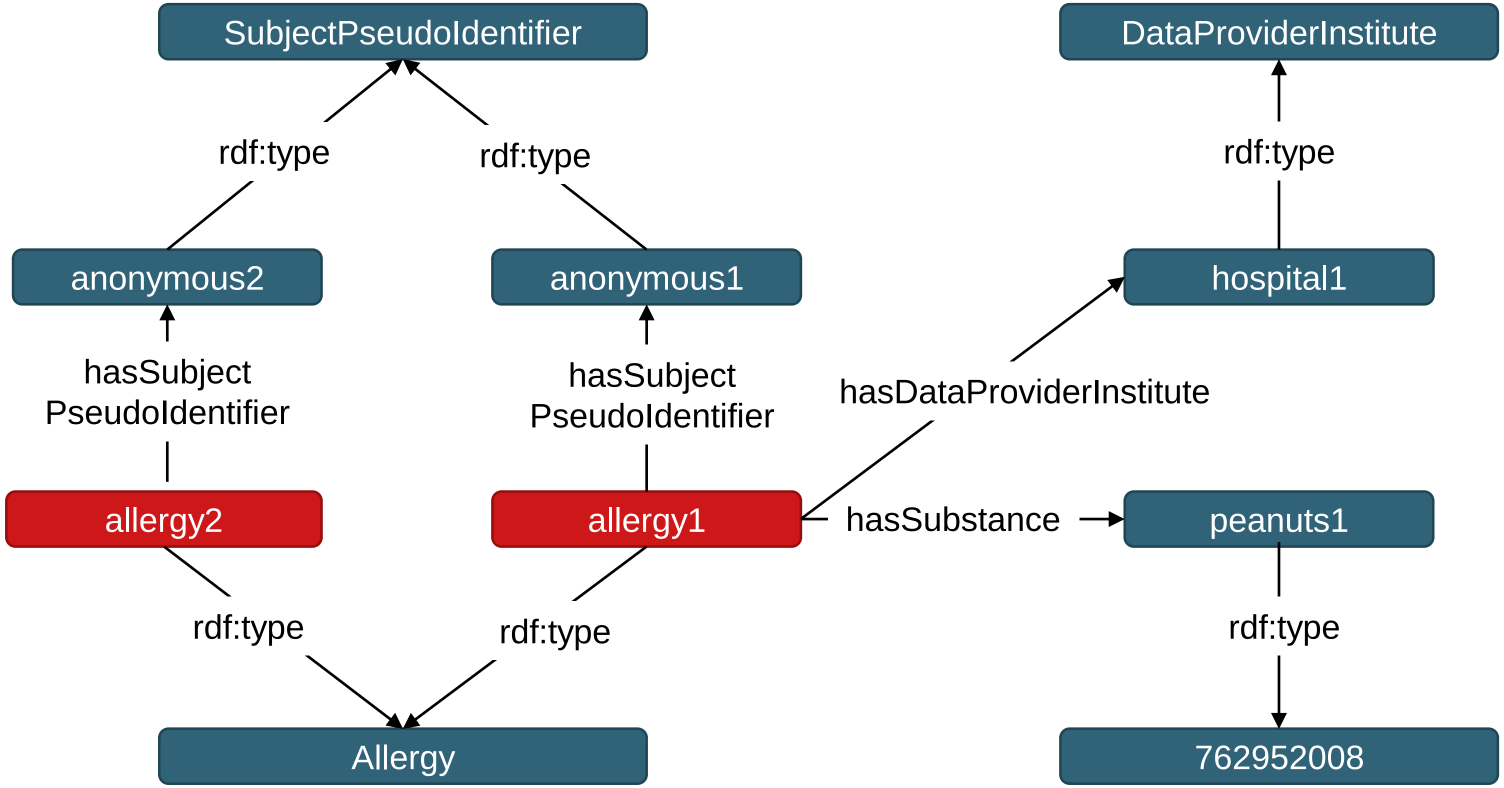
Namespaces have been omitted for a simplified representation

sh:PropertyShape

SHACL Shape specifying all nodes as focus nodes, which are instances of the class :Allergy.

The PropertyShape requires the selected focus nodes to have a :hasDataProviderInstitute property, which refers to a listed IRI (:hospital1 or :hospital3).

```
ex:SibShape6
  a sh:NodeShape ;
  sh:targetClass :Allergy ;
  sh:property ex:SubPropertyShape1 .

ex:SubPropertyShape1
  a sh:PropertyShape ;
  sh:path :hasDataProviderInstitute ;
  sh:in ( :hospital1 :hospital3 ) .
```



Namespaces have been omitted for a simplified representation

sh:PropertyShape (2)

SHACL Shape specifying all nodes as focus nodes, which are instances of the class :Allergy.

The PropertyShape requires the selected focus nodes to have a :hasDataProviderInstitute property, which refers to a listed IRI (:hospital1 or :hospital3).

```
ex:SibShape6
  a sh:NodeShape ;
  sh:targetClass :Allergy ;
  sh:property ex:SubPropertyShape1 .

ex:SubPropertyShape1
  a sh:PropertyShape ;
  sh:path :hasDataProviderInstitute ;
  sh:in ( :hospital1 :hospital3 ) .
```

equivalent representation

SHACL Shape specifying all nodes as focus nodes, which are instances of the class :Allergy.

The PropertyShape requires the selected focus nodes to have a :hasDataProviderInstitute property, which refers to a listed IRI (:hospital1 or :hospital3).

```
ex:SibShape7
  a sh:NodeShape ;
  sh:targetClass :Allergy ;
  sh:property [
     sh:path :hasDataProviderInstitute ;
     sh:in ( :hospital1 :hospital3 )
  ] .
```

Namespaces have been omitted for a simplified representation

SHACL Property Paths

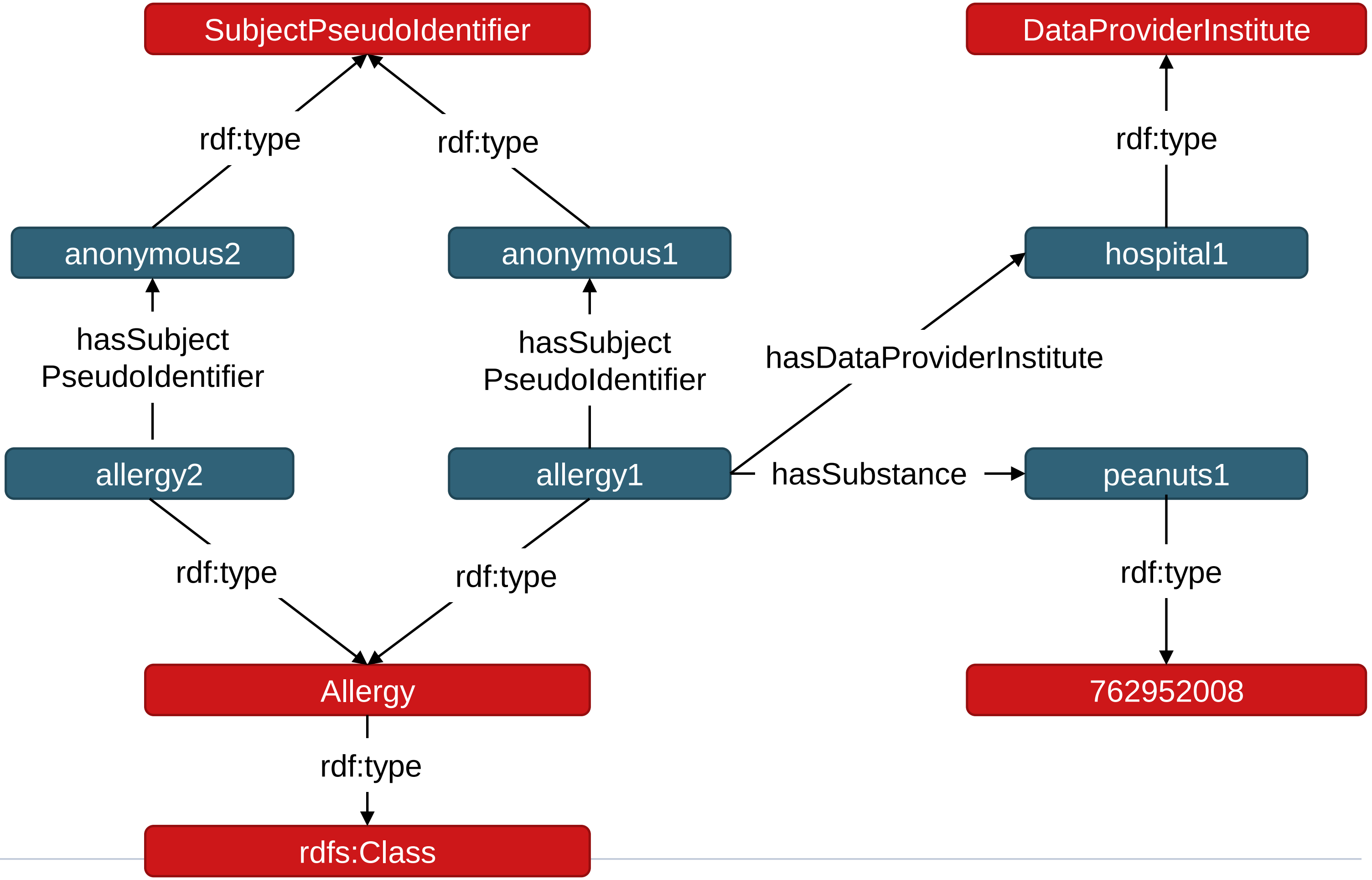
SPARQL Property Paths	SHACL
ex:parent	ex:parent
^ex:parent	[sh:inversePath ex:parent]
ex:parent / ex:firstName	(ex:parent ex:firstName)
rdf:type+	[sh:oneOrMorePath rdf:type]
rdf:type / rdfs:subClassOf*	(rdf:type [sh:zeroOrMorePath rdfs:subClassOf])
ex:father | ex:mother	[sh:alternativePath (ex:father ex:mother)]

SHACL Property Paths Example

SHACL Shape selects nodes, which are used as types. Constraint checks whether they are IRIs.

The Property Path expression follows the rdf:type property recursively

```
ex:SibShape8
  a sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:path [ sh:oneOrMorePath rdf:type;
  sh:nodeKind sh:IRI ] .
```



Namespaces have been omitted for a simplified representation

applicability

SHACL Core Constraints

Constraint Type	Constraints (namespace sh: is omitted for readability)	Node Shape	Property Shape
Value Type	`class, datatype, nodeKind, targetClass`	✓	✓
Cardinality	`minCount, maxCount`	✘	✓
Values	`node, in, hasValue, path`	✓	✓

Value Range	`minInclusive, maxInclusive`
	`minExclusive, maxExclusive`
		✘	✓

String-based	`minLength, maxLength, pattern, languageIn, uniqueLang`	✓	✓
Logical constraints	`not, and, or, xone`	✓	✓
Closed shapes	`closed, ignoredProperties`	✓	✘
Property Pair constraints	`equals, disjoint, lessThan, lessThanOrEquals`	✘	✓
Non-validating constraints	`name, value, defaultValue`	✓	✓
Qualified shapes	`qualifiedValueShape, qualifiedMinCount, qualifiedMaxCount`	✘	✓

Namespaces have been omitted for a simplified representation

SHACL Core Constraints currently in SPHN

Constraint Type	Constraints (namespace sh: is omitted for readability)	SPHN application example
Value Type	`class, datatype, nodeKind, targetClass`	hasBodySiteCode property points to either an instance or subinstance of snomed:123037004
Cardinality	`minCount, maxCount`	Only one SubjectPseudoIdentifier is connected to a BodyHeight
Values	`node, in, hasValue, path`	Death_status is only allowed to be instantiated by the individuals of sphn:Death or sphn:Unknown
Logical constraints	`not, and, or, xone`	Whenever there is are multiple possibilities to express
Closed shapes	`closed, ignoredProperties`	No other properties are allowed to be used in SPHN (is changing)

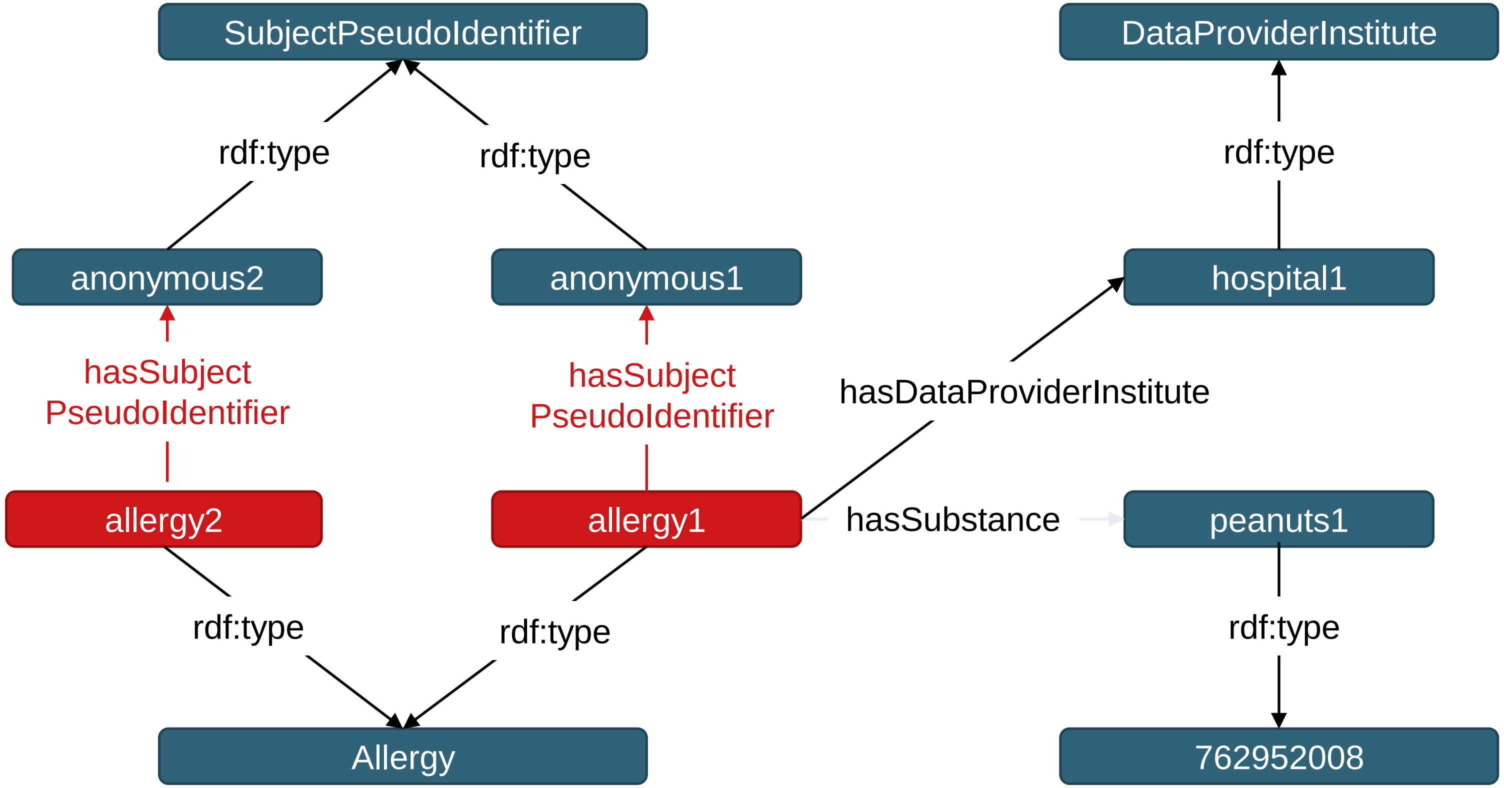
Namespaces have been omitted for a simplified representation

Cardinality

SHACL Shape specifying all nodes as focus nodes, which are instances of the class :Allergy.

The PropertyShape requires the selected focus nodes to have exactly one object with the property :hasDataProviderInstitute .

```
ex:SibShape11
  a sh:NodeShape ;
  sh:targetClass :Allergy ;
  sh:property [
    sh:path :hasSubjectPseudoIdentifier ;
    sh:minCount 1 ;
    sh:maxCount 1
  ] .
```



Namespaces have been omitted for a simplified representation

sh:closed

In RDF, any node can have values for any property

sh:closed specifies that focus nodes only has values for those properties which have been explicitly enumerated via sh:property

Constraint	Description
closed	If true, valid resources must only have values for properties that appear in sh:property
ignoredProperties	Optional list of properties that are also allowed

SHACL Shape allowing only the property :hasDataProviderInstitute and rdf:type to be used.

```
ex:SibShape12
  a sh:NodeShape ;
  sh:closed true ;
  sh:property [
     sh:path :hasDataProviderInstitute
  ] ;
sh:ignoredProperties (rdf:type) .
```

Poll 3 (Multiple Choice)

Which of the following cannot be expressed with SHACL?

❑ Validate the distribution of values

❑ Check the completeness, i.e., whether all nodes are present in the graph

❑ Correct mistakes in the graph, e.g., exclusion of a node based on the validation result

❑ Validation against recursive constraints

❑ Monitoring of changes in graphs

❑ Cardinality of relations

Limitations of SHACL tailored to SPHN

Cannot validate on classes that are "misspelled" e.g.,

```
sphn:Death_Status1  ≠  sphn:Death_Status
```

SPHN Quality Assurance Framework

Consists of SHACL + SPARQL + Execution Framework

Combining validation (SHACL + some SPARQL) with Statistical Queries (SPARQL)

Validation Report

SHACL Part 3

Validation Report

The validation report is the result of the validation process

● ☑ ☑ ☑
● ☑ ☑ ☑
● ☑ ☒ ☒

It reports the conformance (true or false) and a set of validation results

The validation report is again an RDF graph described with the SHACL Validation Report Vocabulary

Example of a validation report where the data graph conforms to the shapes graph

```
[   a sh:ValidationReport ;
    sh:conforms true ;
] .
```

Example of a validation report where the data graph does not conform to the shapes graph

```
[   a sh:ValidationReport ;
    sh:conforms false ;
    sh:result [ a sh:ValidationResult ;  ...] ,
              [ a sh:ValidationResult ;  ...] .
] .
```

Validation Report Example

A Validation Report may provide guidance on how to identify or fix violations in the data graph.

Additional information such as provenance metadata is provided by means of properties as shown below.

Example of a validation report where the data graph does not conform to the shapes graph

```
[  a sh:ValidationReport ;
   sh:conforms false ;
   sh:result [
      a sh:ValidationResult ;
      sh:resultSeverity  sh:Violation ;
      sh:focusNode       :Bob ;
      sh:resultPath      :age ;
      sh:value           "twenty two" ;
      sh:resultMessage   ":age expects a literal of datatype xsd:integer." ;
      sh:sourceConstraintComponent  sh:DatatypeConstraintComponent ;
      sh:sourceShape                :Person-age ;
   ] .
] .
```

The focus node :Bob violates the SHACL Shape „:Person-age“. The property :age is only allowed to have integer values, but a literal “twenty two” was found.

Validation Report Properties

Property Name	Property	Description
Conformance Checking	`sh:conforms`	`false` if the validation produce any results, i.e., a Validation Results, and `true` otherwise
Validation Results	`sh:result`	Each validation produces a `sh:result`
Focus node	`sh:focusNode`	A validation result has exactly one focus node, which was validated and has caused the violation
Path	`sh:resultPath`	Equivalent to the value of `sh:path` of the shape
Value	`sh:value`	RDF term (at most one) that caused the result
Source	`sh:sourceShape`	Shape name that the focus node was validated against
Constraint Component	`sh:sourceConstraintComponent`	Specifiy the constraint component, that caused the result, e.g., the constraint `sh:minCount` has `sh:MinCountConstraintComponent`
Details	`sh:detail`	May link to other violations for that shape
Message	`sh:resultMessage`	Communicate additional textual details to humans
Severity	`sh:resultSeverity`	The severity level of the shape that caused the result

Namespaces have been omitted for a simplified representation

Debugging Shapes

Declaring the Severity of a Shape

How critical is a violation of this shape?

Declaring Messages for a Shape

What information should be shown in case of violation?

Deactivating a Shape

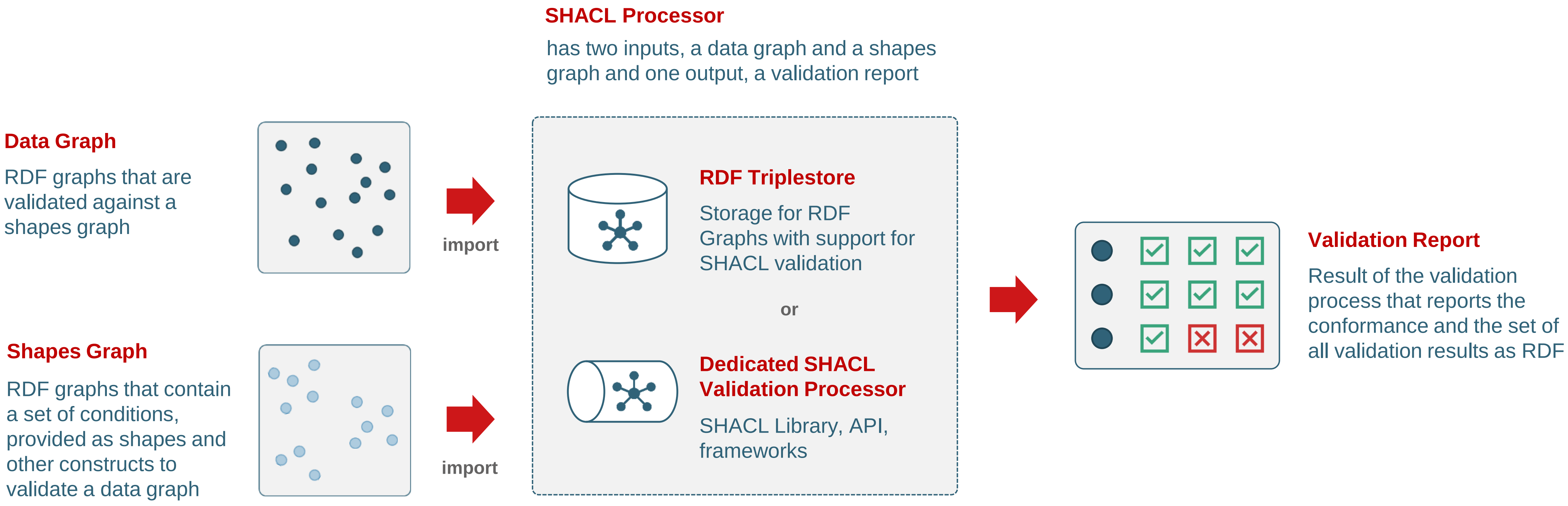
Exclude shapes which should not be applied (tests, application context)

Hands-On with GraphDB

SHACL Part 4

We are doing it. You are invited to repeat it on your own.

What do we need for our Hands-On? (review)



SHACL Processor

has two inputs, a data graph and a shapes graph and one output, a validation report

Data Graph

RDF graphs that are validated against a shapes graph

Shapes Graph

RDF graphs that contain a set of conditions, provided as shapes and other constructs to validate a data graph

RDF Triplestore

Storage for RDF Graphs with support for SHACL validation

or

Dedicated SHACL Validation Processor

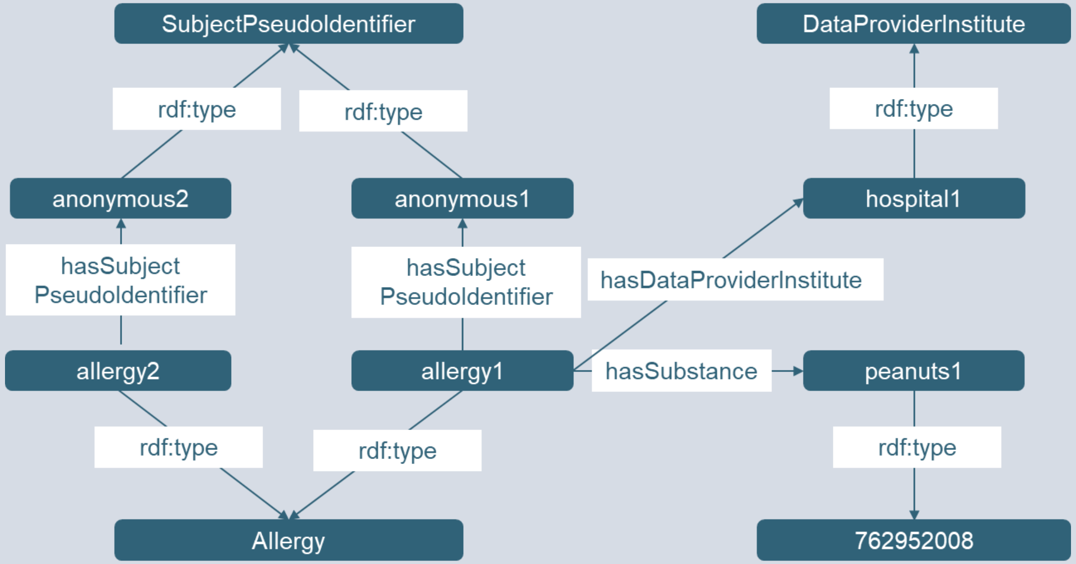
SHACL Library, API, frameworks

Validation Report

Result of the validation process that reports the conformance and the set of all validation results as RDF

Data Graph

```
@prefix allergies: <http://sib.swiss/allergies/> .
@prefix patients: <http://sib.swiss/fictivePatients/> .
@prefix substances: <http://sib.swiss/substances/> .
@prefix sib: <http://sib.swiss/> .
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix snomed: <http://snomed.info/id/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

# types
patients:anonymous1 rdf:type sphn:SubjectPseudoIdentifier .
patients:anonymous2 rdf:type sphn:SubjectPseudoIdentifier .
sib:hospital1 rdf:type sphn:DataProviderInstitute .
allergies:allergy1 rdf:type sphn:Allergy .
allergies:allergy2 rdf:type sphn:Allergy .
substances:peanuts1 rdf:type snomed:762952008 .

# relations to the allergy
allergies:allergy1 sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
allergies:allergy1 sphn:hasDataProviderInstitute sib:hospital1 .
allergies:allergy1 sphn:hasSubstance substances:peanuts1 .
allergies:allergy2 sphn:hasSubjectPseudoIdentifier patients:anonymous2 .
```



RDF Graph in Turtle Syntax (*.ttl)

S

SHACL Shapes which conform with the Data Graph. No Validation Results are produced.

```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex:   <http://sib.swiss/examples#> .
@prefix sib:  <http://sib.swiss/> .

ex:SibShape1_correct
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:nodeKind sh:IRI .

ex:SibShape2_correct
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:property [
    sh:path sphn:hasDataProviderInstitute ;
    sh:in ( sib:hospital1 sib:hospital2 sib:hospital3 )
 ] .

ex:SibShape3_correct
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:path ( sphn:hasSubstance rdf:type ) .

ex:SibShape4_correct
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path sphn:hasDataProviderInstitute ;
   sh:minCount 0;
   sh:maxCount 1
 ] .
```

Modified SHACL Shapes which produce Validation Results.

```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex:   <http://sib.swiss/examples#> .
@prefix sib:  <http://sib.swiss/> .

ex:SibShape1_wrong
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:nodeKind sh:Literal .

ex:SibShape2_wrong
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:property [
    sh:path sphn:hasDataProviderInstitute ;
    sh:in ( sib:hospital2 sib:hospital3 )
  ] .

ex:SibShape3_wrong
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path ( sphn:hasSubstance rdf:type ) ;
   sh:minCount 1
 ] .

ex:SibShape4_wrong
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path sphn:hasDataProviderInstitute ;
   sh:minCount 1;
   sh:maxCount 1
 ] .
```

RDF Graph in Turtle Syntax (*.ttl)

Shape Graph

SHACL Shapes which conform with the Data Graph. No Validation Results are produced.

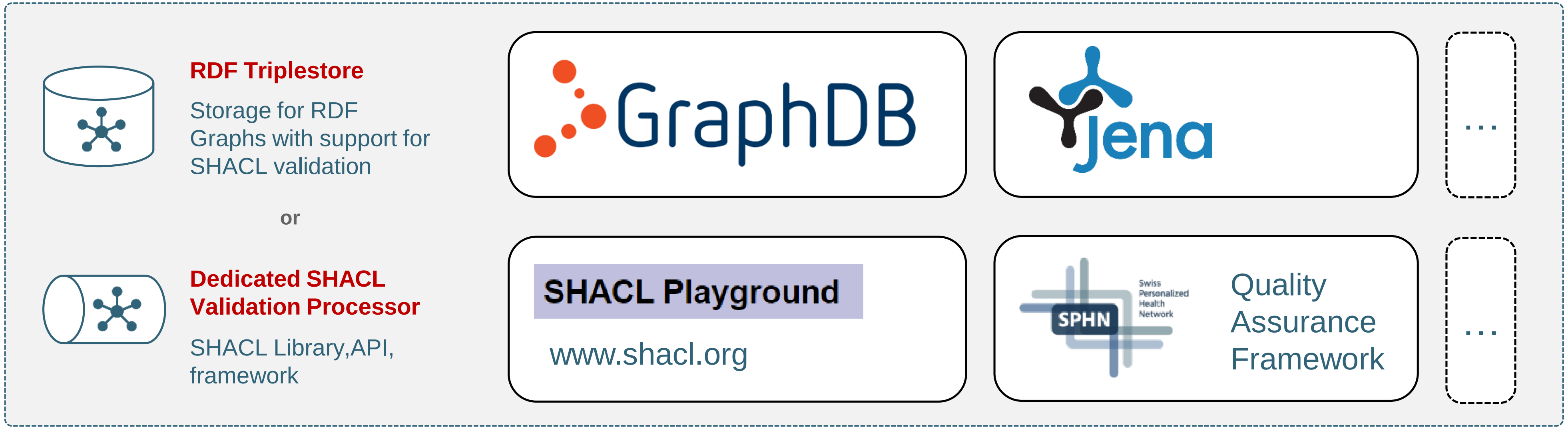
```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex:   <http://sib.swiss/examples#> .
@prefix sib:  <http://sib.swiss/> .

ex:SibShape2_correct
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:property [
    sh:path sphn:hasDataProviderInstitute ;
    sh:in ( sib:hospital1 sib:hospital2 sib:hospital3 )
 ] .
```

Modified SHACL Shapes which produce Validation Results.

```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex:   <http://sib.swiss/examples#> .
@prefix sib:  <http://sib.swiss/> .

ex:SibShape2_wrong
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:property [
    sh:path sphn:hasDataProviderInstitute ;
    sh:in ( sib:hospital2 sib:hospital3 )
  ] .
```

SHACL Processor

SHACL Processor



RDF Triplestore

Storage for RDF Graphs with support for SHACL validation

or

Dedicated SHACL Validation Processor

SHACL Library,API, framework

GraphDB

Jena

…

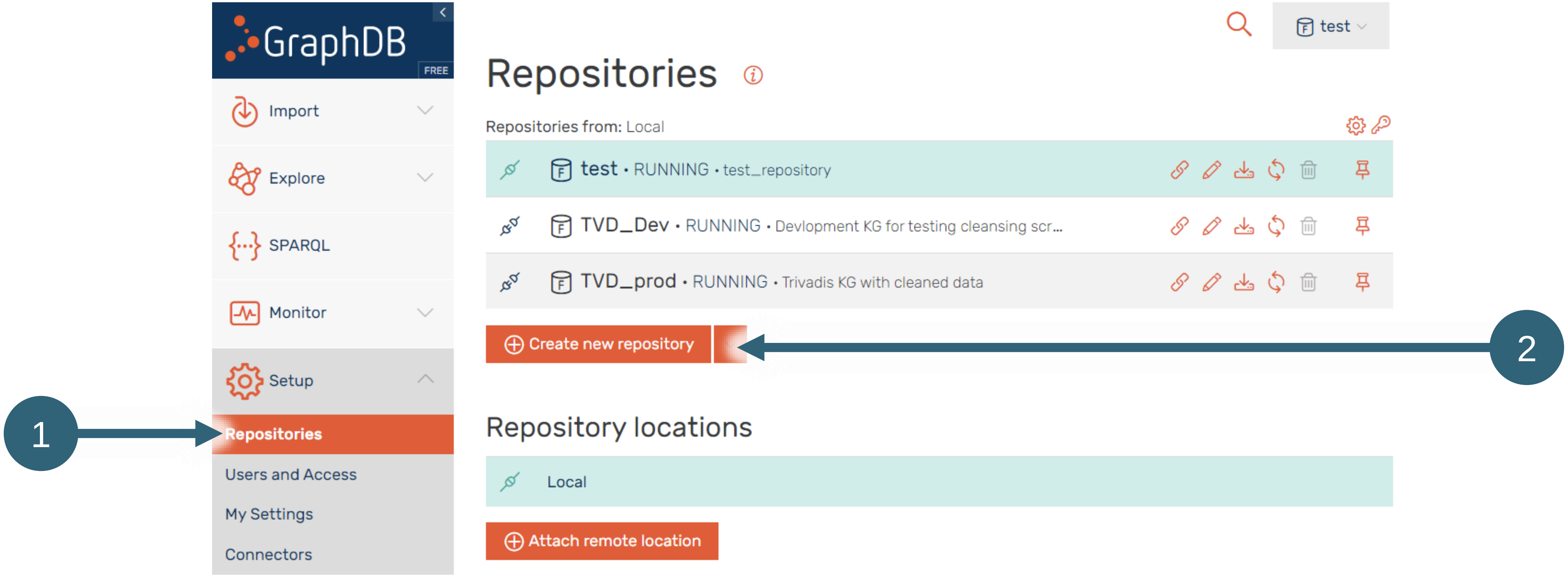
SHACL Playground

www.shacl.org

SPHN Quality Assurance Framework

…

Example with GraphDB Free (1/9)



Repositories

Repositories from: Local

test • RUNNING • test_repository

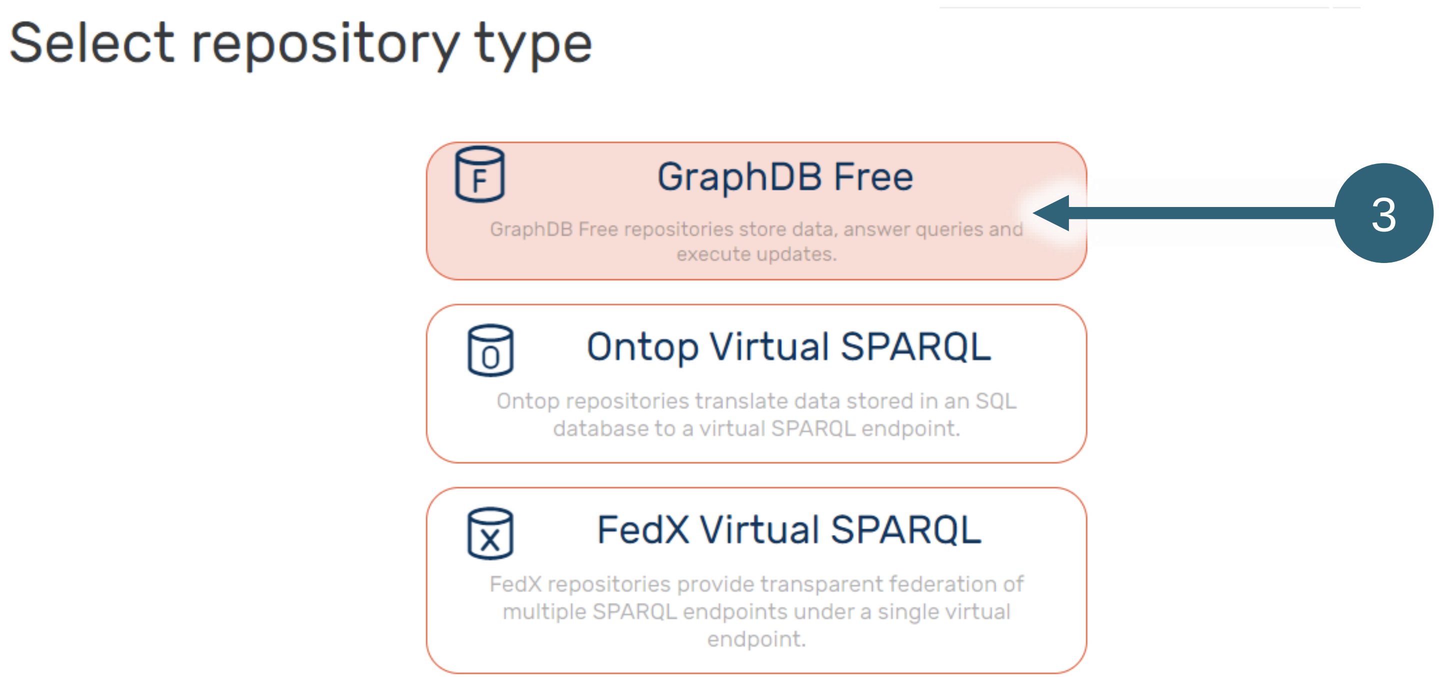
TVD_Dev • RUNNING • Devlopment KG for testing cleansing scr...

TVD_prod • RUNNING • Trivadis KG with cleaned data

Repository locations

Local

Example with GraphDB Free (2/9)



Select repository type

GraphDB Free

GraphDB Free repositories store data, answer queries and execute updates.

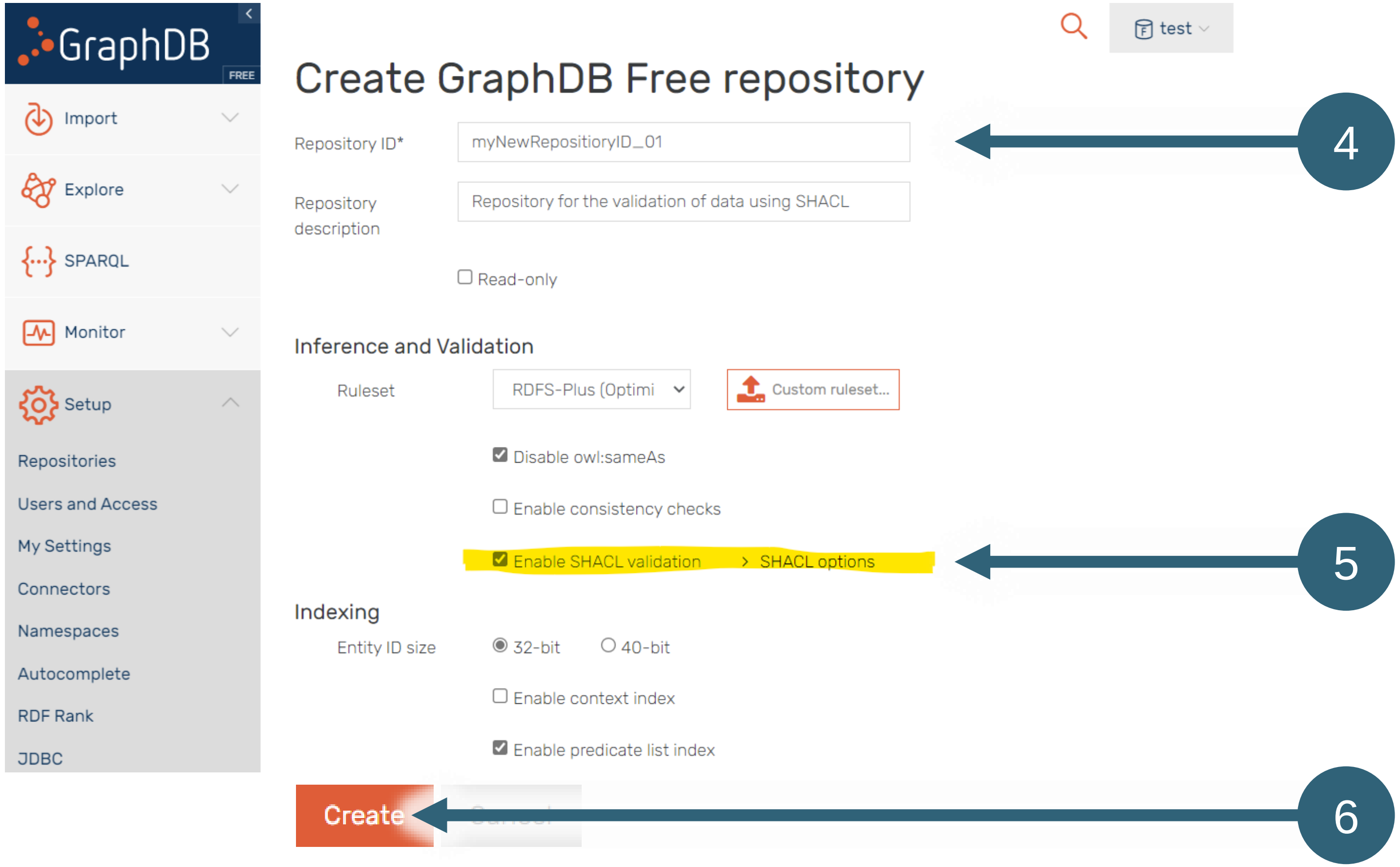
Ontop Virtual SPARQL

Ontop repositories translate data stored in an SQL database to a virtual SPARQL endpoint.

FedX Virtual SPARQL

FedX repositories provide transparent federation of multiple SPARQL endpoints under a single virtual endpoint.

Example with GraphDB Free (3/9)



Create GraphDB Free repository

Repository ID*: myNewRepositioryID_01

Repository description: Repository for the validation of data using SHACL

□ Read-only

Inference and Validation

Ruleset: RDFS-Plus (Optimi — Custom ruleset...

☑ Disable owl:sameAs

□ Enable consistency checks

☑ Enable SHACL validation › SHACL options

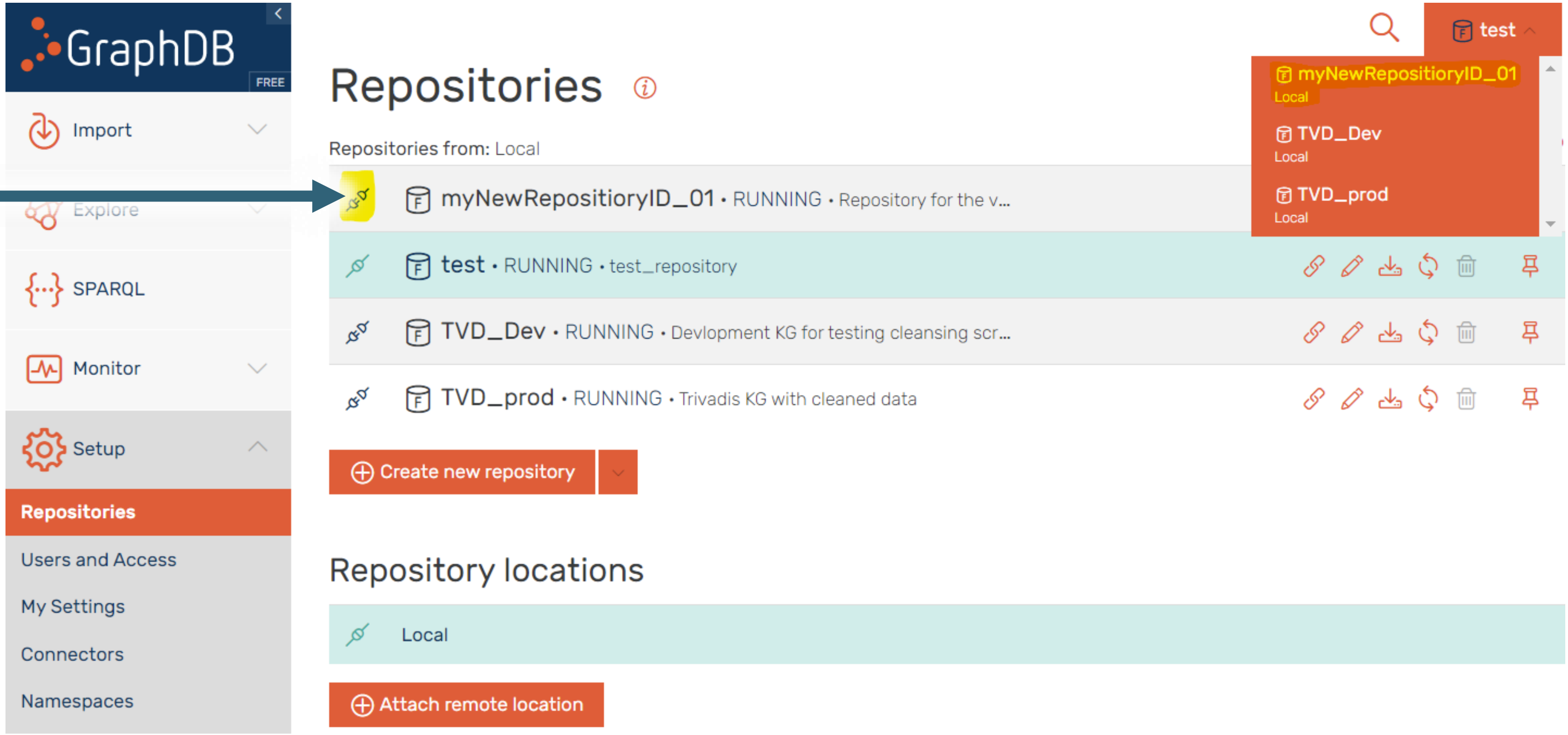
Indexing

Entity ID size: ● 32-bit ○ 40-bit

□ Enable context index

☑ Enable predicate list index

Example with GraphDB Free (4/9)



7

Connect to new repository

GraphDB FREE

Import

Explore

SPARQL

Monitor

Setup

Repositories

Users and Access

My Settings

Connectors

Namespaces

Repositories

Repositories from: Local

myNewRepositioryID_01 • RUNNING • Repository for the v...

test • RUNNING • test_repository

TVD_Dev • RUNNING • Devlopment KG for testing cleansing scr...

TVD_prod • RUNNING • Trivadis KG with cleaned data

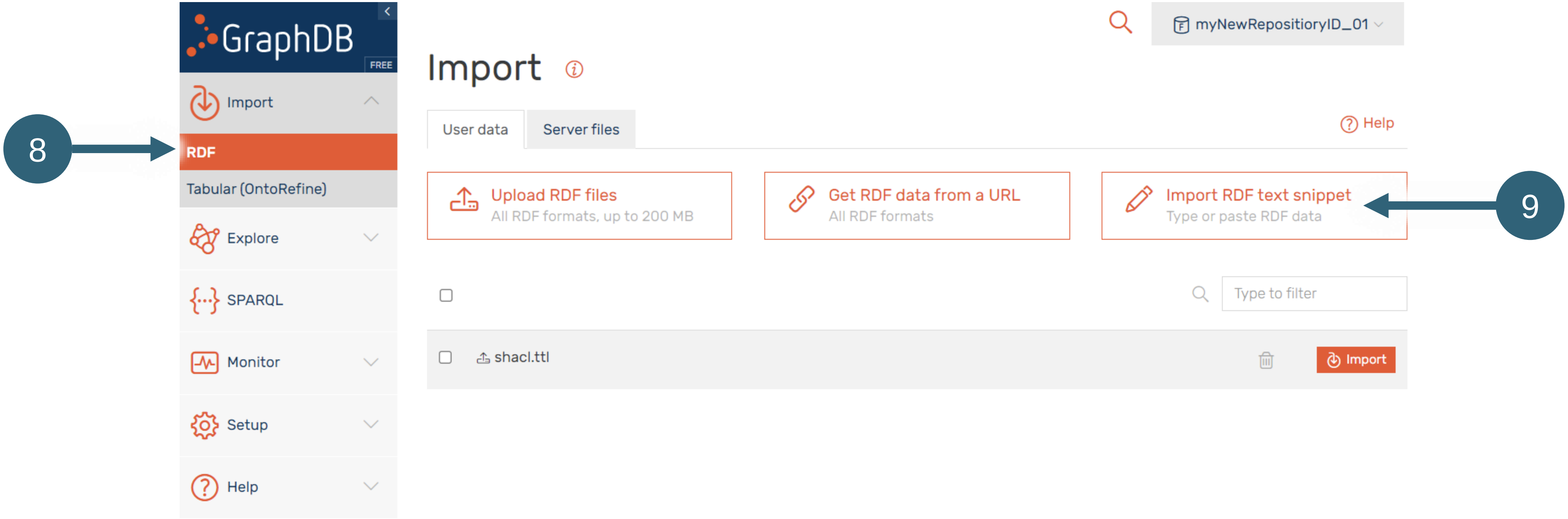
Create new repository

Repository locations

Local

Attach remote location

Example with GraphDB Free (5/9)



Example with GraphDB Free (6/9)

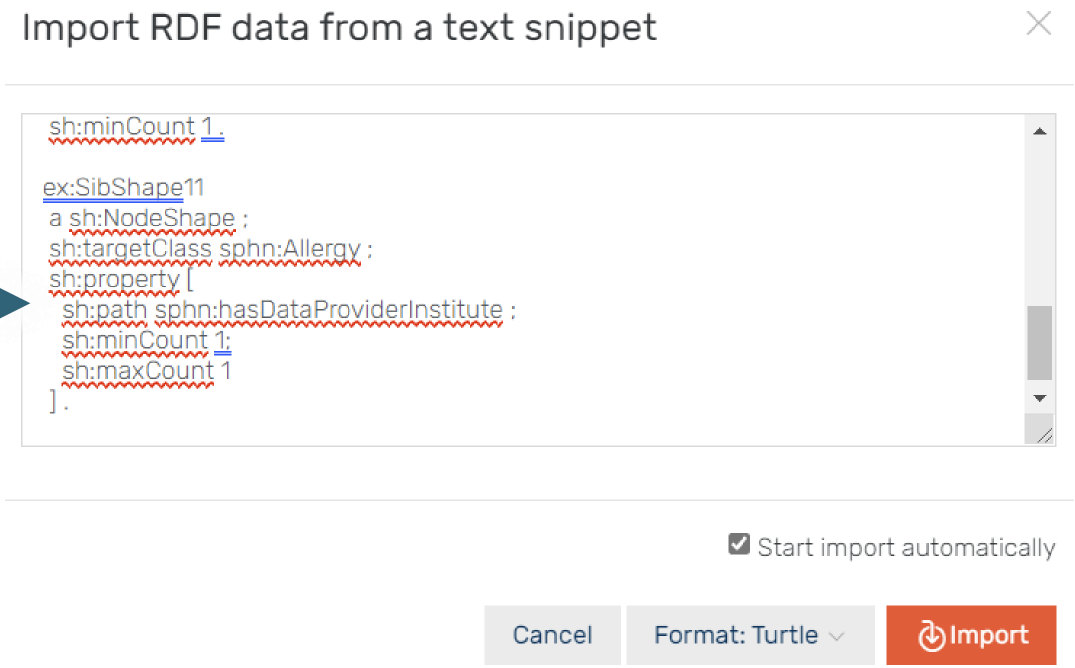
Modified SHACL Shapes which produce Validation Results.

```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex:   <http://sib.swiss/examples#> .
@prefix sib:  <http://sib.swiss/> .

ex:SibShape5
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:nodeKind sh:Literal .

ex:SibShape7
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:property [
    sh:path sphn:hasDataProviderInstitute ;
    sh:in ( sib:hospital2 sib:hospital3 )
  ] .

ex:SibShape9
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path ( sphn:hasSubstance rdf:type ) ;
   sh:minCount 1 .
 ] .

ex:SibShape11
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path sphn:hasDataProviderInstitute ;
   sh:minCount 1;
   sh:maxCount 1
 ] .
```

SHACL Shapes which conform with the Data Graph. No Validation Results are produced.

```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex:   <http://sib.swiss/examples#> .
@prefix sib:  <http://sib.swiss/> .

ex:SibShape5
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:nodeKind sh:IRI .

ex:SibShape7
  a sh:NodeShape ;
  sh:targetClass sphn:Allergy ;
  sh:property [
    sh:path sphn:hasDataProviderInstitute ;
    sh:in ( sib:hospital1 sib:hospital1 sib:hospital3 )
 ] .

ex:SibShape9
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:path ( sphn:hasSubstance rdf:type ) .

ex:SibShape11
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path sphn:hasDataProviderInstitute ;
   sh:minCount 0;
   sh:maxCount 1
 ] .
```

10

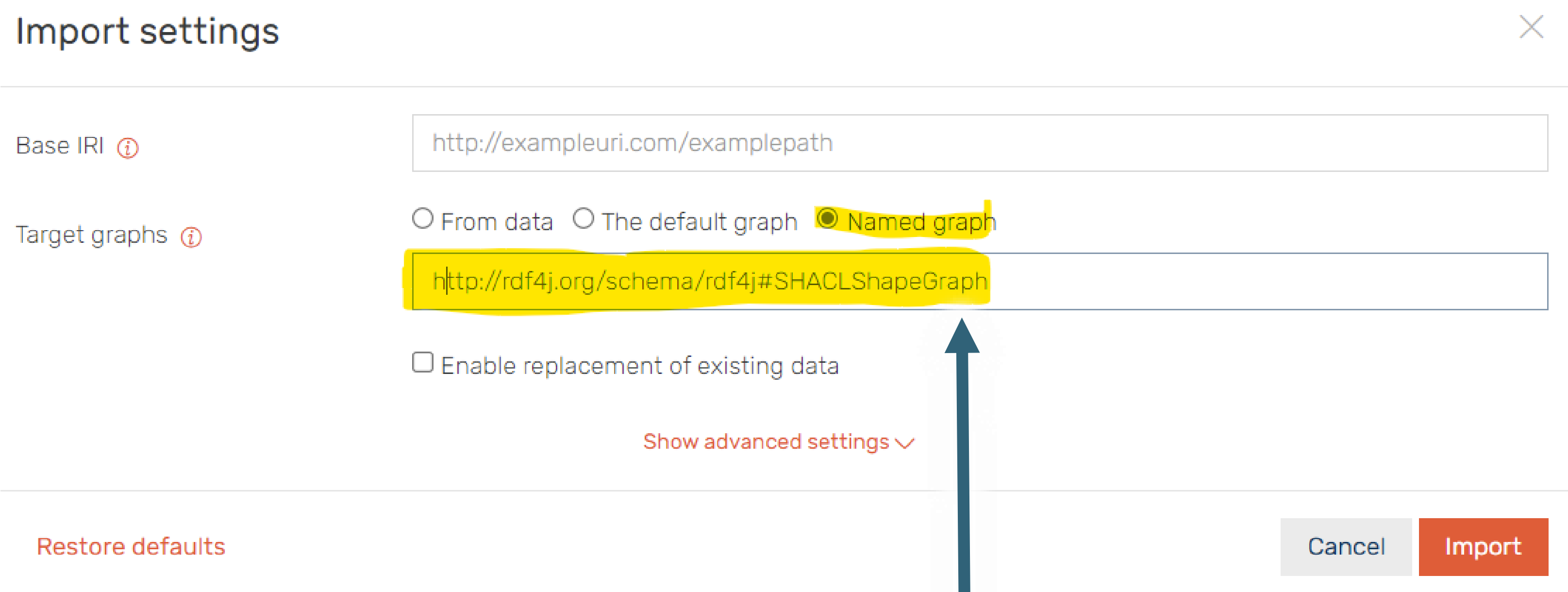


Import RDF data from a text snippet

```
 sh:minCount 1 .

ex:SibShape11
 a sh:NodeShape ;
 sh:targetClass sphn:Allergy ;
 sh:property [
   sh:path sphn:hasDataProviderInstitute ;
   sh:minCount 1;
   sh:maxCount 1
 ] .
```

☑ Start import automatically

Example with GraphDB Free (7/9)



Import settings

Base IRI ⓘ http://exampleuri.com/examplepath

Target graphs ⓘ ○ From data ○ The default graph ● Named graph

http://rdf4j.org/schema/rdf4j#SHACLShapeGraph

☐ Enable replacement of existing data

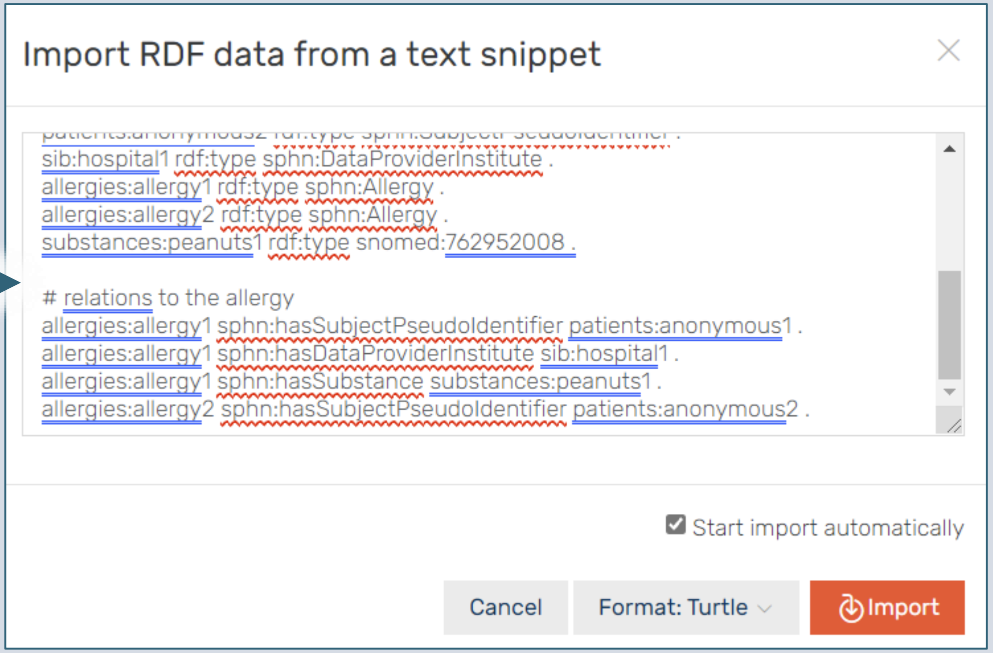
11 Copy

```
http://rdf4j.org/schema/rdf4j#SHACLShapeGraph
```

Example with GraphDB Free (8/9)

```
@prefix allergies: <http://sib.swiss/allergies/> .
@prefix patients: <http://sib.swiss/fictivePatients/> .
@prefix substances: <http://sib.swiss/substances/> .
@prefix sib: <http://sib.swiss/> .
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix snomed: <http://snomed.info/id/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

# types
patients:anonymous1 rdf:type sphn:SubjectPseudoIdentifier .
patients:anonymous2 rdf:type sphn:SubjectPseudoIdentifier .
sib:hospital1 rdf:type sphn:DataProviderInstitute .
allergies:allergy1 rdf:type sphn:Allergy .
allergies:allergy2 rdf:type sphn:Allergy .
substances:peanuts1 rdf:type snomed:762952008 .

# relations to the allergy
allergies:allergy1 sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
allergies:allergy1 sphn:hasDataProviderInstitute sib:hospital1 .
allergies:allergy1 sphn:hasSubstance substances:peanuts1 .
allergies:allergy2 sphn:hasSubjectPseudoIdentifier patients:anonymous2 .
```

12



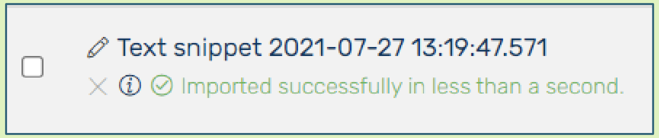
Import RDF data from a text snippet

```
patients:anonymous2 rdf:type sphn:SubjectPseudoIdentifier .
sib:hospital1 rdf:type sphn:DataProviderInstitute .
allergies:allergy1 rdf:type sphn:Allergy .
allergies:allergy2 rdf:type sphn:Allergy .
substances:peanuts1 rdf:type snomed:762952008 .

# relations to the allergy
allergies:allergy1 sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
allergies:allergy1 sphn:hasDataProviderInstitute sib:hospital1 .
allergies:allergy1 sphn:hasSubstance substances:peanuts1 .
allergies:allergy2 sphn:hasSubjectPseudoIdentifier patients:anonymous2 .
```

☑ Start import automatically

Example with GraphDB Free (9/9)

SHACL Shapes which conform with the Data Graph. No Validation Results are produced.



□ Text snippet 2021-07-27 13:19:47.571

Imported successfully in less than a second.

Failed SHACL Validation, Validation Results are produced. (shortened)

```
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

_:node1 a sh:ValidationReport;
  sh:conforms false;
  sh:result _:node2 .

_:node2 a sh:ValidationResult;
  sh:focusNode <http://sib.swiss/allergies/allergy1>;
  sh:sourceConstraintComponent sh:InConstraintComponent;
  sh:sourceShape <http://shape.ontotext.com/node#8057ff97/1>;
  sh:resultPath sphn:hasDataProviderInstitute;
  sh:value <http://sib.swiss/hospital1> .
```

org.eclipse.rdf4j.sail.shacl.GraphDBShaclSailValidationException: Failed SHACL validation

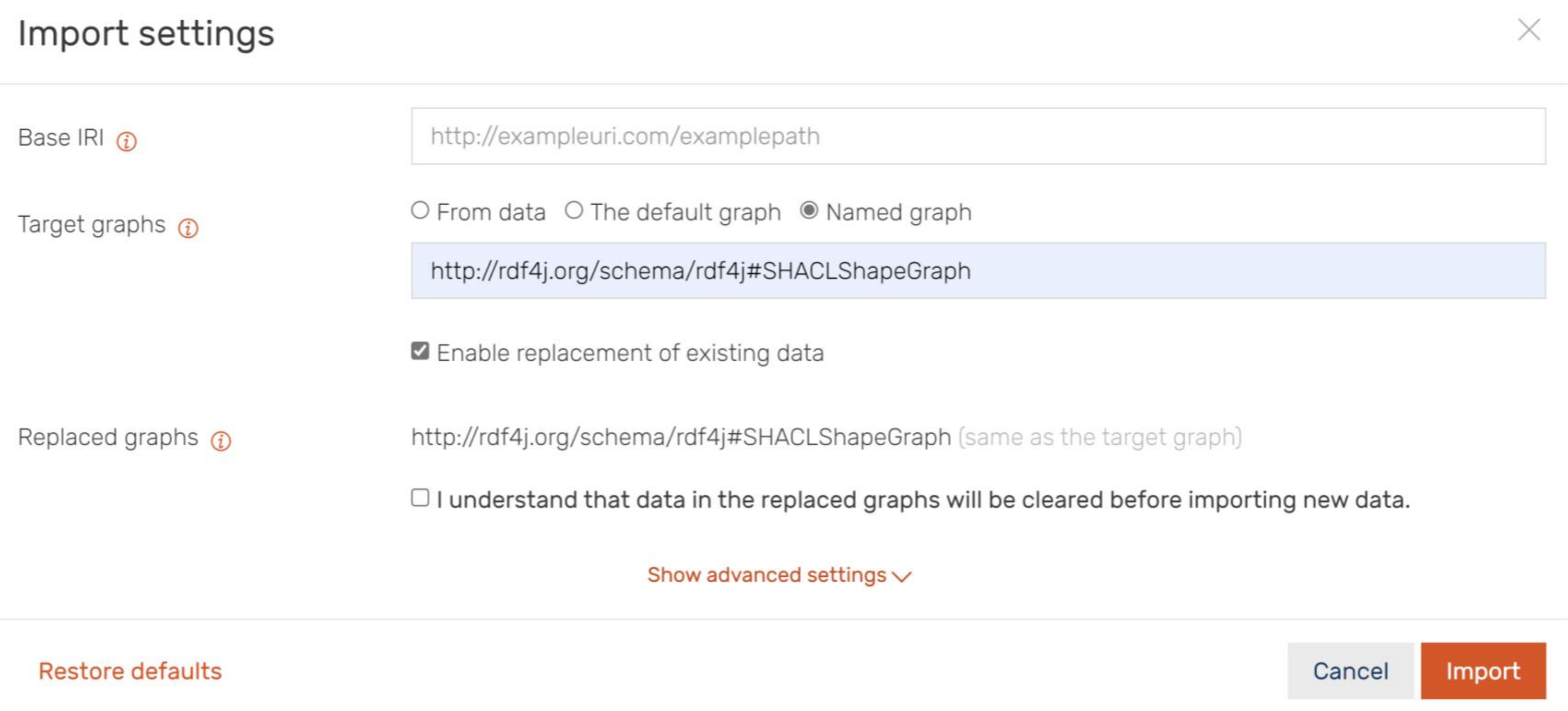
```
@prefix sh: <http://www.w3.org/ns/shacl#> .

_:node1fjvvjh1qx20666 a sh:ValidationReport;
 sh:conforms false;
 <http://rdf4j.org/schema/rdf4j#truncated> false;
 sh:result _:50caaf9e-9f15-47ab-9251-ac8f00953264, _:b8afc68d-ff69-4941-b790-744b9fd808bc,
  _:ef856ec9-f5e4-4e50-83d2-8474ef6a284d, _:57dc6962-12db-49e0-9110-3f937842bd93 .

_:50caaf9e-9f15-47ab-9251-ac8f00953264 a sh:ValidationResult;
 sh:focusNode <http://sib.swiss/allergies/allergy1>;
 sh:value <http://sib.swiss/hospital1>;
 sh:resultPath <https://biomedit.ch/rdf/sphn-ontology/sphn#hasDataProviderInstitute>;
 sh:sourceConstraintComponent sh:InConstraintComponent;
 sh:resultSeverity sh:Violation;
 sh:sourceShape _:a0a2fecd-1f7d-4d83-91b0-bc2492198561-1 .

_:a0a2fecd-1f7d-4d83-91b0-bc2492198561-1 a sh:PropertyShape;
 sh:path <https://biomedit.ch/rdf/sphn-ontology/sphn#hasDataProviderInstitute>;
 sh:in _:567c7179-a5d1-40f0-b366-2b4dcbda0868-2 .

_:567c7179-a5d1-40f0-b366-2b4dcbda0868-2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>
  <http://sib.swiss/hospital2>;
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:node1fjvvjh1qx20667 .

_:node1fjvvjh1qx20667 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://sib.swiss/hospital3>;
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil> .

_:b8afc68d-ff69-4941-b790-744b9fd808bc a sh:ValidationResult;
 sh:focusNode <http://sib.swiss/allergies/allergy1>;
```

Poll 4 (single choice)

Why does the allergy1 example fail on hospital1?

- ❏ hospital1 is not in the list of allowed values
- ❏ hospital1 is wrongly written
- ❏ there was no shape for hospital1

Replacing and Removing Shape Graphs in GraphDB



Import settings

Base IRI ⓘ http://exampleuri.com/examplepath

Target graphs ⓘ ○ From data ○ The default graph ● Named graph

http://rdf4j.org/schema/rdf4j#SHACLShapeGraph

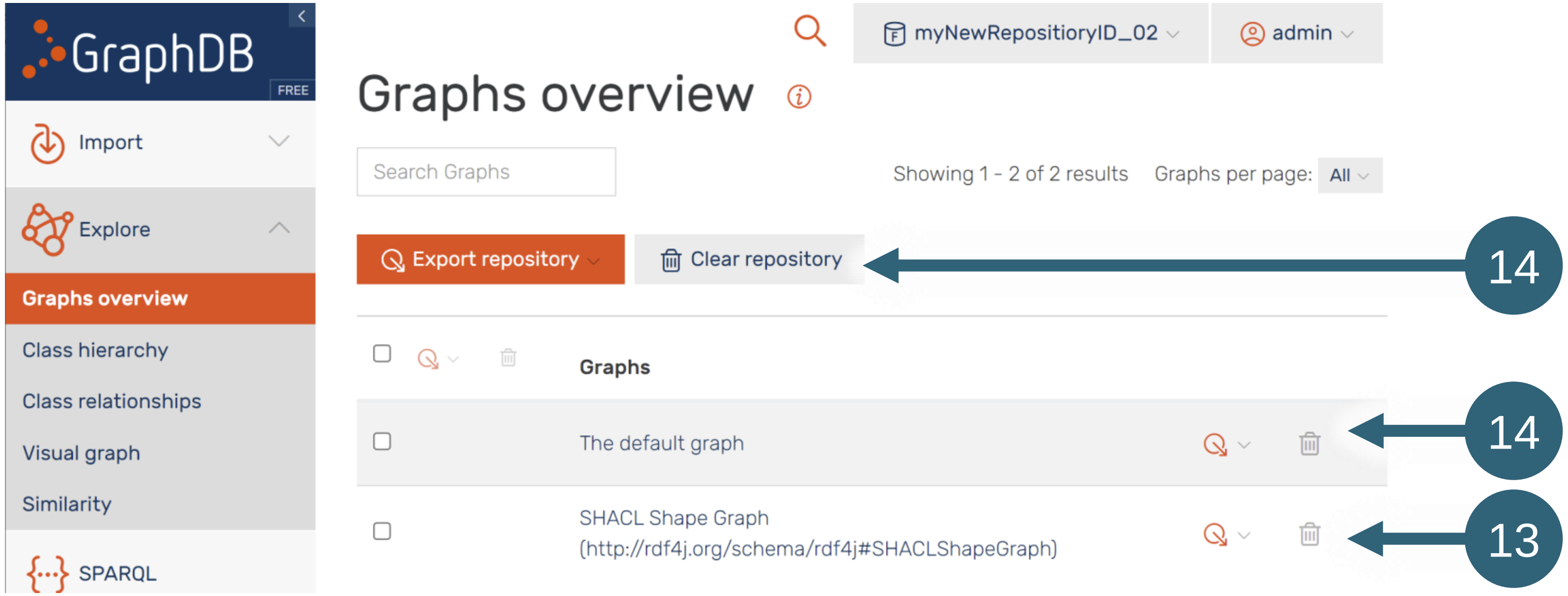
☑ Enable replacement of existing data

Replaced graphs ⓘ http://rdf4j.org/schema/rdf4j#SHACLShapeGraph (same as the target graph)

□ I understand that data in the replaced graphs will be cleared before importing new data.

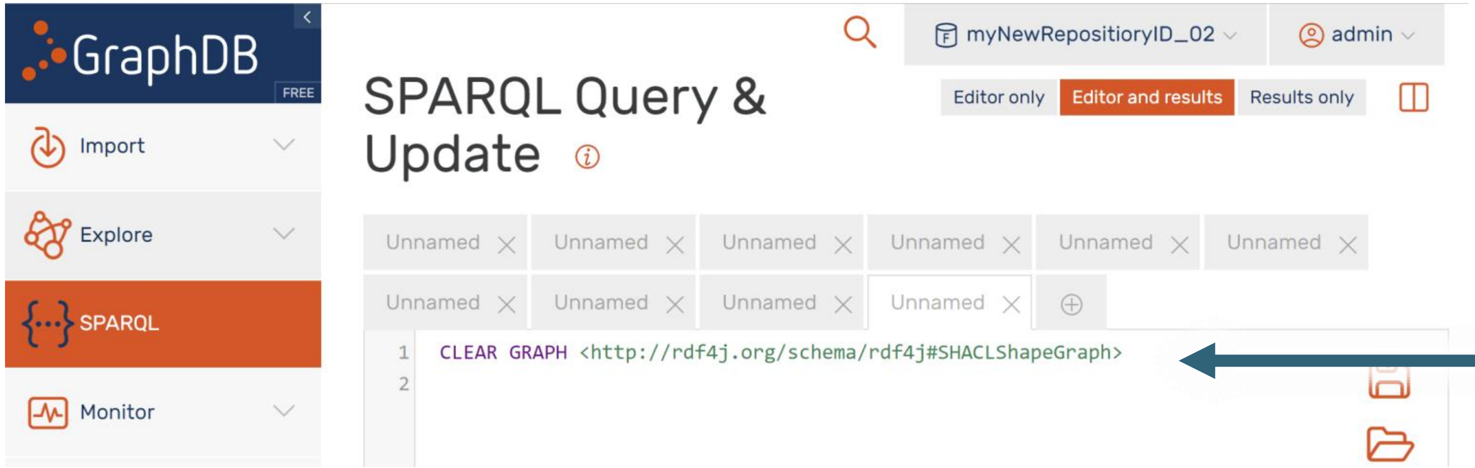
Shape Graphs are stored separately from data, importing a new shape graph by enabling the **Enable replacement of existing data** box option in the Import settings dialog box would **not** work.

Replacing and Removing Shape Graphs in GraphDB



Go to the Graphs overview, remove the Graphs and Clear the repository

Replacing and Removing Shape Graphs in GraphDB



SPARQL Query & Update

```
CLEAR GRAPH <http://rdf4j.org/schema/rdf4j#SHACLShapeGraph>
```

16

Go to the SPARQL Editor in the Workbench.

Clear the RDF Graph for storing shapes by running a CLEAR GRAPH query

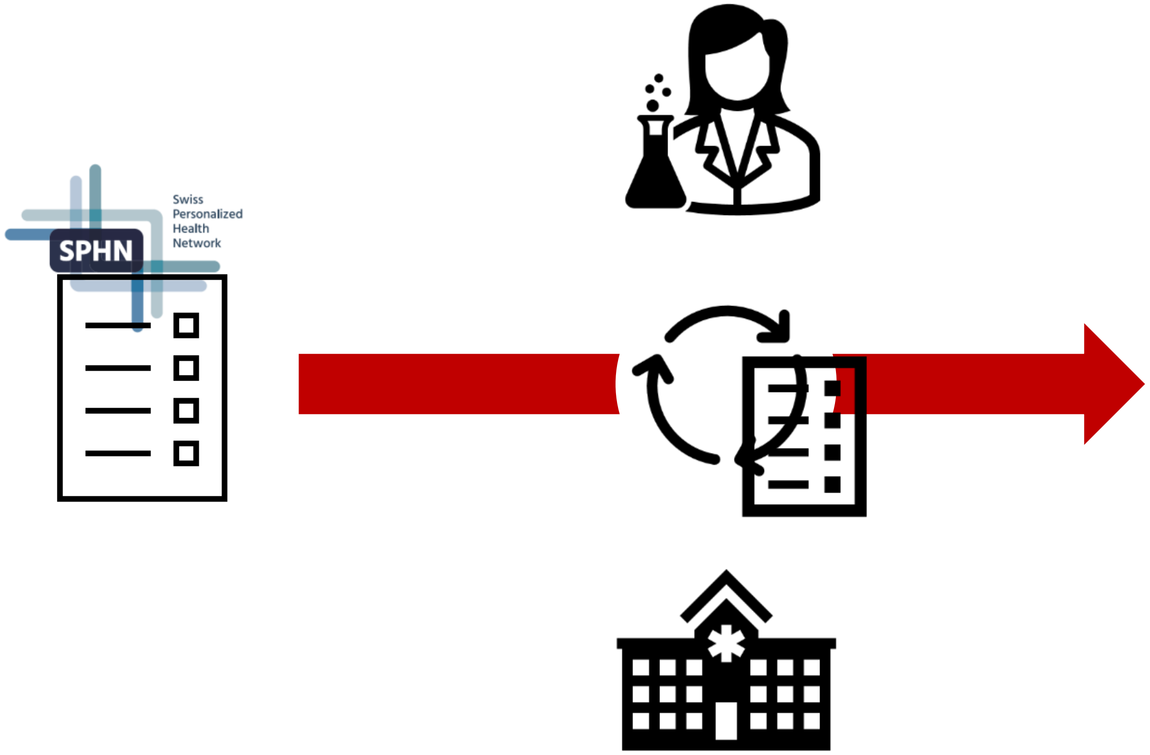
15 Copy

```
CLEAR GRAPH <http://rdf4j.org/schema/rdf4j#SHACLShapeGraph>
```

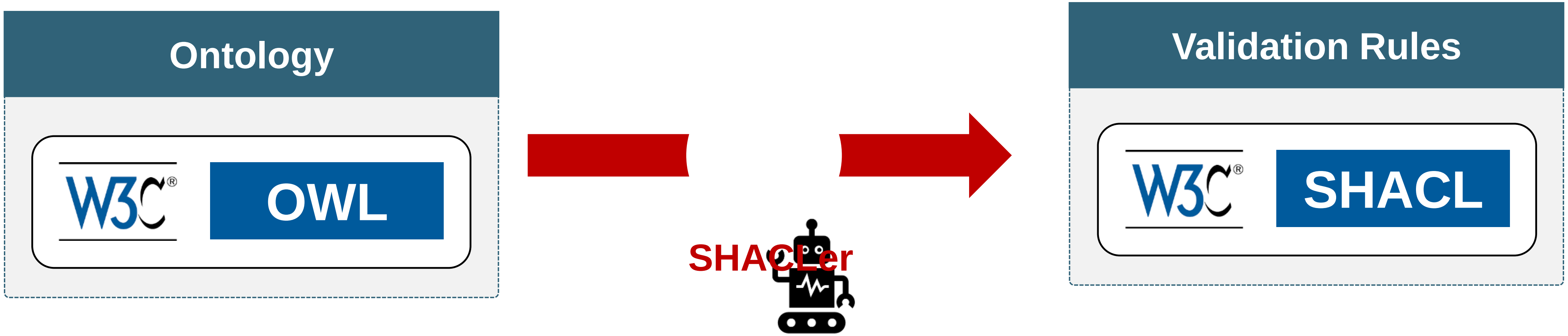
How to SHACLer

SHACL Part 5

Project specific RDF schema



Procedure



Ontology

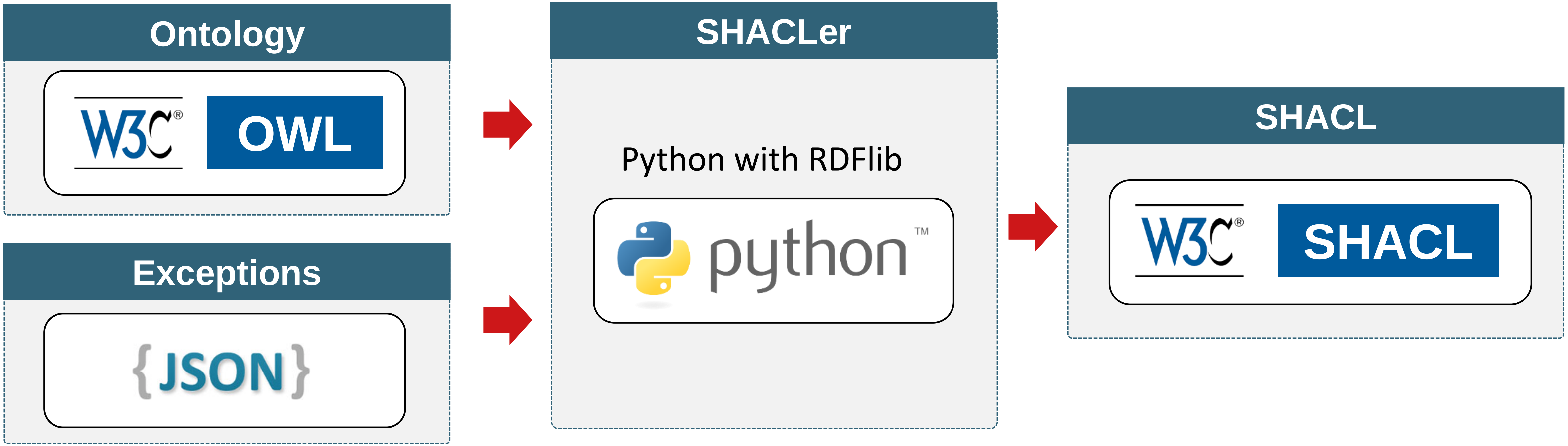
W3C OWL

SHACLer

Validation Rules

W3C SHACL

Technical Overview



Ontology

Exceptions

SHACLer

Python with RDFlib

SHACL

Demo Time!

Input (shortened)

```
###  https://biomedit.ch/rdf/sphn-ontology/sphn#hasMSuffix

sphn:hasMSuffix
    rdf:type owl:DatatypeProperty ;
    rdfs:subPropertyOf sphn:hasTNMClassificationValue ;
    rdfs:domain sphn:TNMClassification ;
    rdfs:range xsd:string ;
    rdfs:comment "suffix associated to the metastasis (M), as defined in the TNM coding system, e.g., 1b" ;
    rdfs:label "has M suffix code" .
```

How to run

```
.\shacl_generator.py -o 'ttl' -d ".\sphn_ontology.ttl" -e
"exceptions.json" "shacl_2021-2.ttl"
```

Output (shortened)

```
constraints:sphnTNMClassification a sh:NodeShape ;
    sh:closed false ;
    sh:ignoredProperties ( rdf:type ) ;
    sh:property
        [ sh:datatype xsd:string ;
            sh:path sphn:hasMSuffix ];
    sh:targetClass sphn:TNMClassification .
```

References

References 1

Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017
https://www.w3.org/TR/shacl/

RDF and Linked Data Validation - ESWC'16 Tutorial, ESWC 2016
https://www.weso.es/RDFValidation_ESWC16/

Validating RDF Data, Jose E. Labra Gayo, Eric Prud’hommeaux, Iovka Boneva, Dimitris Kontokostas (2018)
http://book.validatingrdf.com/

References 2

SHACLer Documentation

https://sphn-semantic-framework.readthedocs.io/en/latest/sphn_framework/dataquality.html?highlight=shacler#data-validation-with-shacl

https://sphn-semantic-framework.readthedocs.io/en/latest/user_guide/data_quality.html?highlight=shacler#data-validation

SHACLer Repository

https://git.dcc.sib.swiss/sphn-semantic-framework/sphn-shacl-generator
(available on request, will be released publicly soon)

SPHN Quality Assurance Framework (available on request)

https://git.dcc.sib.swiss/sphn-semantic-framework/sphn-ontology/-/tree/master/quality_assurance