DigiPass

Know Your Products!

SHACL Workshop

November 12, 2025

Aas, Norway

Harmonization of Advanced Materials Ecosystems serving strategic Innovation Markets to pave the way to a Digital Materials & Product Passport

COORDINATION AND KNOWLEDGE SHARING ACROSS MATERIALS DEVELOPMENT COMMUNITIES (CSA):

DigiPass CSA project has been funded by the European Commission for the programme HORIZON-CL4-2023-RESILIENCE-01, Grant Agreement No **101138510**
WIKKI LIMITED, UK participant in Horizon Europe Project DigiPass, is supported by UKRI grant number 10100819: **DigiPass**

Why

DPP is data

read

write

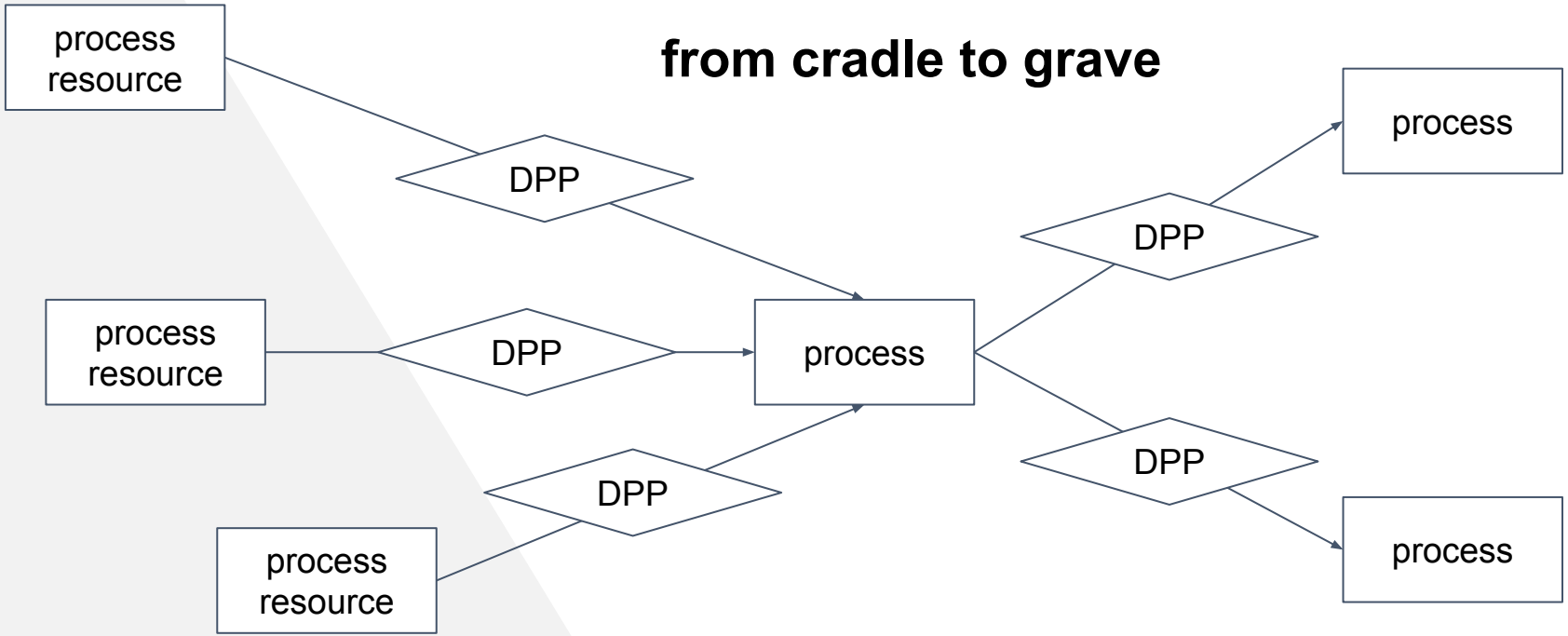
transfer

store

check

validate

Process network/chaining → DPP network/chaining



Data handling is the backbone of any platform

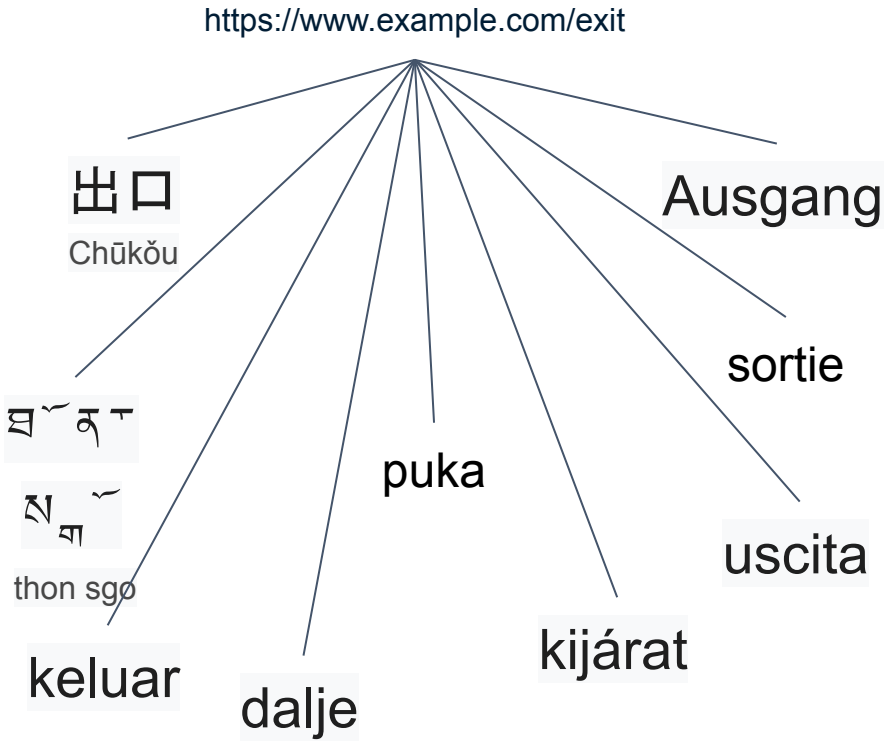
computer readable → interoperability

DPPs are linked data sets

Language independent

IRI

International Resource Identifiers



Program

- Basics -- RDF
- What is SHACL
- SHACL Components
- SHACL example with focus on generating a GUI form
- Quick presentation of exercises
- Useful links

Foundation -- RDF

RDF (**R**esource **D**efinition **F**ramework)

- based on triples **Subject Predicate Object** -- small sentences
- form graphs
  - nodes: subjects and objects
  - arcs: predicates

desirable:

all elements are linked to IRIs,

thus implementing a ontology-domain-specific language

What is SHACL

SHACL stands for **SHA**pe **C**onstraint **L**anguage.

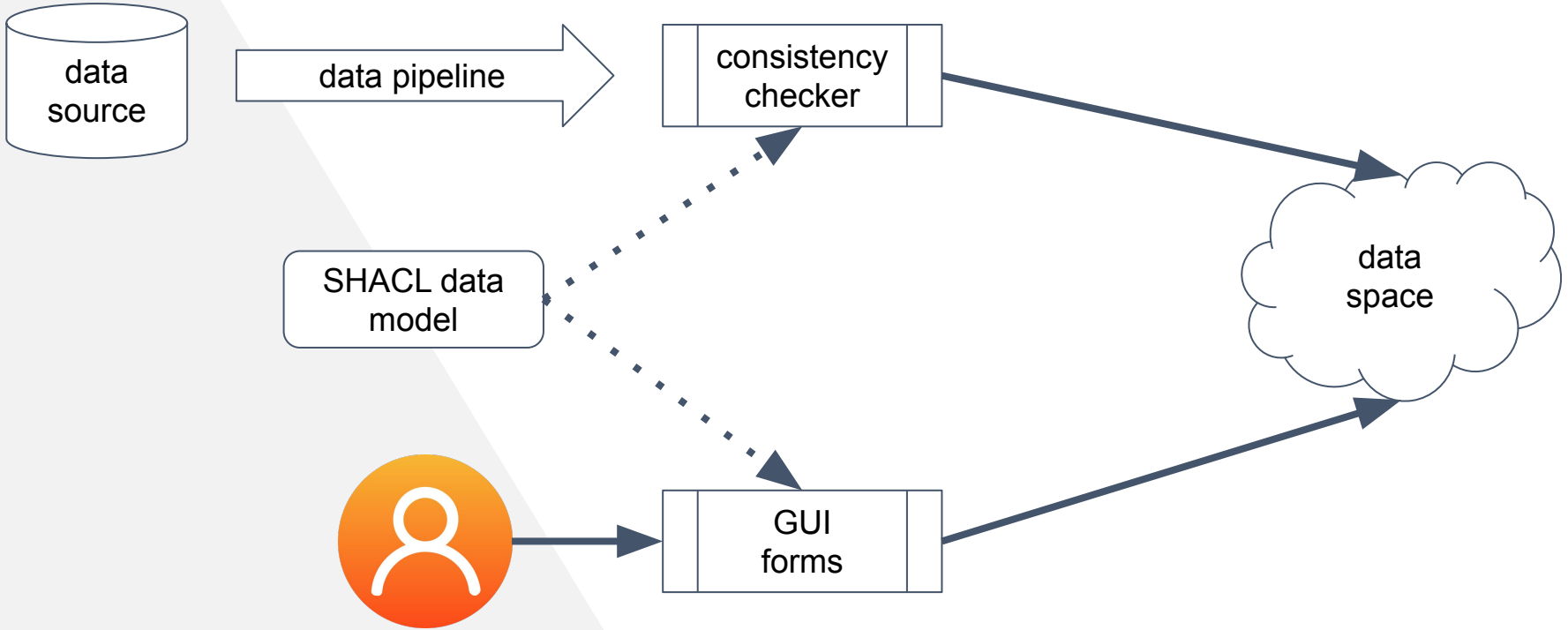
SHACL is designed to describe data models including data type and constraints

SHACL was designed for checking the consistency of data with the data model.

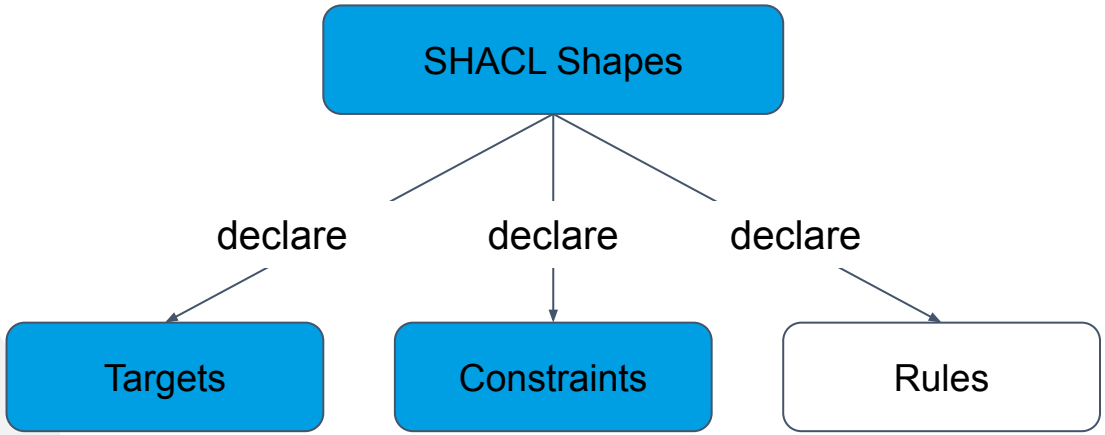
SHACL ensures conformance of RDF data to a defined schema (model) *.

*TopQuadrant -- vocabulary inspired by IBM resource shapes -- architecture & rules inspired by SPIN -- syntax inspired by ShEx

Model for checking → model for GUI input forms



SHACL components



target data to conform to model

how data looks

infer on existing data

*TopQuadrant

A SHACL model is a **shapes graph.**

A **node shape** captures the characteristics of a specific shapes graph node.

A **property shape** captures the characteristic and constraints of focus graph nodes that are reached via a

- a path -- sh:path: ex:parent
- inverse path -- sh:path: [sh:inversePath ex:parent]
- sequence of paths -- sh:path: (ex:parent ex:firstName)
- alternative paths -- sh:alternativePath (ex:pathA | ex:pathB)

Constraint Type	Constraints Component	Node Shape	Property Shape
Value Type	class, datatype, nodeKind, targetClass	yes	yes
Cardinality	minCount, maxCount	no	yes
Values	node, in, hasValue, path	yes	yes
Value Range	minInclusive, maxInclusive, minExclusive, maxExclusive	no	yes
String-based	minLength, maxLength, pattern, languageIn, uniqueLang	yes	yes
Property Pair Constraints	equals, disjoint, lessThan, lessThanOrEquals	no	yes
Logical Constraints	not, and, or, xone	yes	yes
Qualified Shapes	qualifiedValueShape, qualifiedMinCount, qualifiedMaxCount	no	yes
Closed Shapes	closed, ignoredProperties	yes	no
Non-validating Constraints	name, value, defaultValue	yes	yes

SHACL example with focus of generating GUI

turtle

```
# define prefixes to improve readability
@prefix ex: <http://example.org/> .
@prefix sh:      <http://www.w3.org/ns/shacl#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .

# define root shape with the identifier PersonShape
ex:PersonShape a sh:NodeShape ;
  sh:targetClass ex:Person ;
  sh:property [                    # add blank node with properties
    sh:name "first name" ;         # prompt "first name"
    sh:path ex:firstName ;         # applies to all instances of "firstName" in class "Person"
    sh:minCount 1 ;                # minimal number of first names == 1
    sh:datatype xsd:string ;       # data type == string
    ] ;
  sh:property[                     # add blank node with properties
    sh:name "last name" ;          # prompt "last name"
    sh:path ex:lastName ;          # applies to all instances of "lastName" in class "Person"
    sh:minCount 1 ;                # minimal number of last name == 1
    sh:maxCount 1 ;                # maximal number of last name == 1
    sh:datetype xsd:string ;       # data type == string
    sh:maxLength 20] .             # maximal number of characters == 20
```

Retrofitted example from TopQuadrant

turtle

```
# define prefixes to improve readability
@prefix ex: <http://example.org/> .
@prefix sh:      <http://www.w3.org/ns/shacl#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .

# define root shape with the identifier PersonShape
ex:PersonShape a sh:NodeShape ;
  sh:targetClass ex:Person ;
  sh:pattern "^http://mydomain.com/person/" ;
  sh:property ex:Person-first-name ;          # add property node for first name
  sh:property ex:Person-last-name .           # add property node for last name

ex:Person-first-name                          # property node for first name
    sh:name "first name" ;                    # prompt "first name"
    sh:path ex:firstName ;                    # applies to all instances of "firstName" in class "Person"
    sh:minCount 1 ;                           # minimal number of first names == 1
    sh:datatype xsd:string .                  # data type == string

ex:Person-last-name                           # property node for last name
  sh:name "last name" ;                       # prompt "last name"
  sh:path ex:lastName ;                       # applies to all instances of "lastName" in class "Person"
  sh:minCount 1 ;                             # minimal number of last name == 1
  sh:maxCount 1 ;                             # maximal number of last name == 1
  sh:datetype xsd:string ;                    # data type == string
  sh:maxLength 20 .                           # maximal number of characters == 20
```

Quick view on exercises

Workflow

1. Download the exercises as a zip file and unpack it in a suitable location
2. In your favourite browser manoeuvre to the location: https://ulb-darmstadt.github.io/shacl-form/#try-your-own
3. Open your favourite text editor
4. Load in exercise_person_given_name_using_schema.txt
5. Select all and paste into the left widget in the Darmstadt playground
6. Selecting the middle widget shows the form.
7. Try to fill the form. If successful, the result shows in the right widget.
8. You can choose to export the result, which yields a ttl file with the data model.

Person exercise

Download zip file via my webpage:

https://preisig.folk.ntnu.no

or via my github

https://github.com/heinz-preisig/Dissemination/blob/master/README.md

The exercises contain some tips attached to the posed question.

Some reading is required.

Useful links

Main documents

SHACL W3C document The primary SHACL document

SHACL Advanced Features

SHACL core Newest language definition document

SHACL Compact Syntax syntax in ANTLR format

RDF11-concepts RDF base document

RDF 1.1 Turtle The primary Turtle document

SCHEMA ontology A handy ontology for various purposes

Regular Expressions regular expressions

Information on SHACL

SHACL slides TopQuadrant SHACL overview

SHACL Wikipedia English WikiPedia entry for SHACL

SHACL slides Swiss Personalized Health Network

SHACL slides Linköping University

SHACL slides Czech technical University -- RDF validation shows shape and data graph

SHACL slides SHACL by example

Book Validating RDF Data

A different SHACL Wiki Uses a software tool Logseq forming a knowledge graph

GUI form generation DASH

DASH Data Shapes Root webpage

Software

pySHACL A Python validator for SHACL

Relation to Graph Databases

neo4j Validating Neo4j graphs against SHACL

Videos

TopQuadrant Part 1

TopQuadrant Part 2

Video Validation example 1

Video Validation example 2

Cheat sheet

cheat sheet The only cheat sheet found

Know Your Products!

Helmholtz-Zentrum hereon

AC2T research

Fraunhofer ITWM

LUXEMBOURG INSTITUTE OF SCIENCE AND TECHNOLOGY LIST

WIKKI

COMPOSITES UNITED

IRES

ecca prepaintedmetal.eu

NTNU

ARISTOTLE UNIVERSITY OF THESSALONIKI

oet ORGANIC ELECTRONIC TECHNOLOGIES

COORDINATION AND KNOWLEDGE SHARING ACROSS MATERIALS DEVELOPMENT COMMUNITIES (CSA):
DigiPass CSA project has been funded by the European Commission for the programme HORIZON-CL4-2023-RESILIENCE-01, Grant Agreement No **101138510**
WIKKI LIMITED, UK participant in Horizon Europe Project DigiPass, is supported by UKRI grant number 10100819: **DigiPass**