Am I in Good Shape? Flexible Way to Validate Asset Administration Shell Data Entry via Shapes Constraint Language

IEEE ETFA 2023, Sinaia, Romania
Second Workshop on Implementing Asset Administration Shells (ImplAAS)

German Research Center for Artificial Intelligence (DFKI)
Mohammad Hossein Rimaz
Dr. Christiane Plociennik
Leonhard Kunz
Prof. Martin Ruskowski

Outline

Background

Motivation

Related Work

Workflow & Implementation

Conclusion

Background

Interoperability:

Asset Administration Shell (AAS)

A common structure -> Metamodel

Standardized representation formats -> JSON, XML, RDF, …

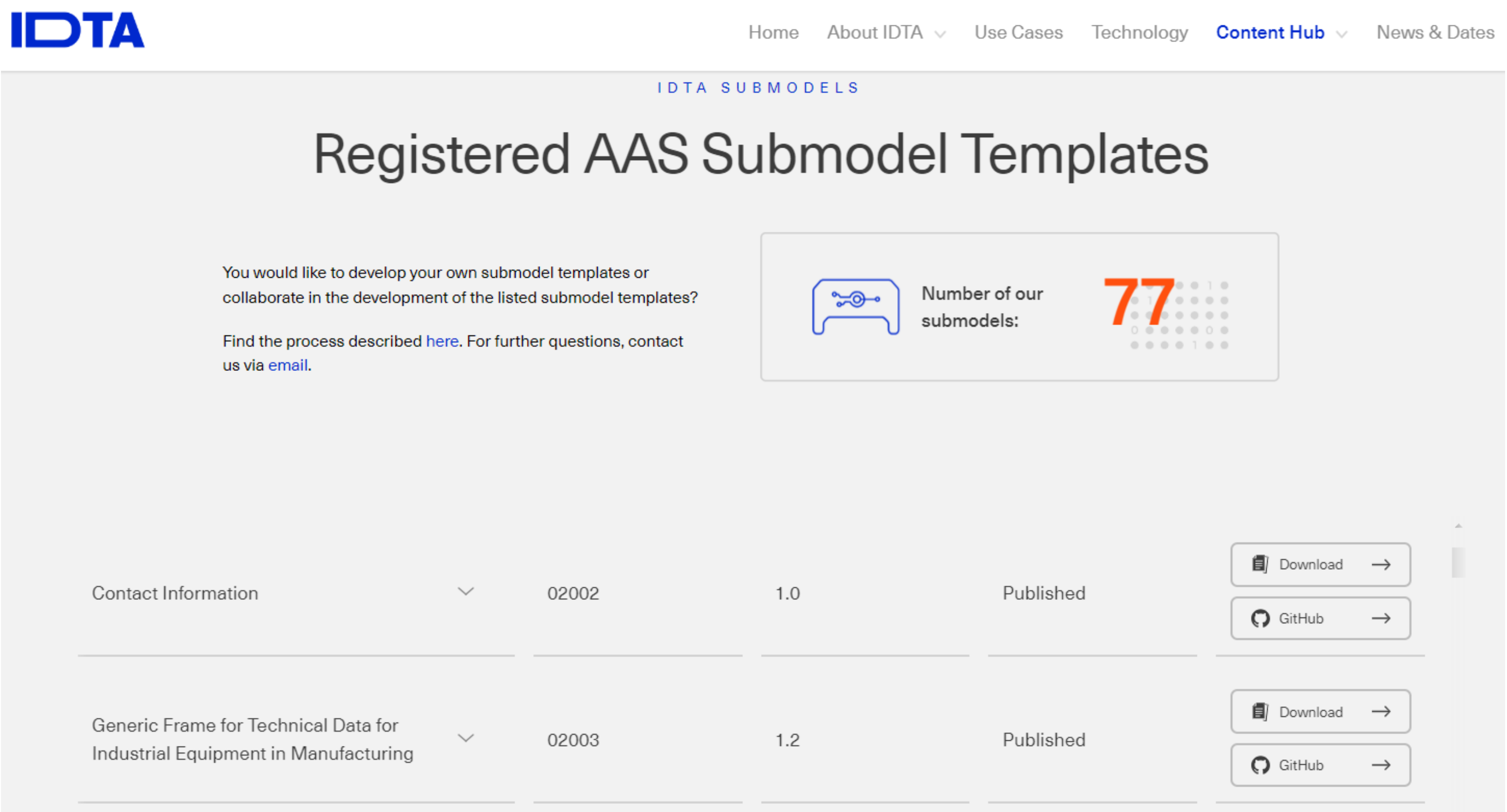
Standardized application programming interface (API) -> REST API

Standardized content -> Submodel Templates

Submodel Templates

Standardized content:

- Submodel Templates



IDTA

Home | About IDTA | Use Cases | Technology | Content Hub | News & Dates

IDTA SUBMODELS

Registered AAS Submodel Templates

You would like to develop your own submodel templates or collaborate in the development of the listed submodel templates?

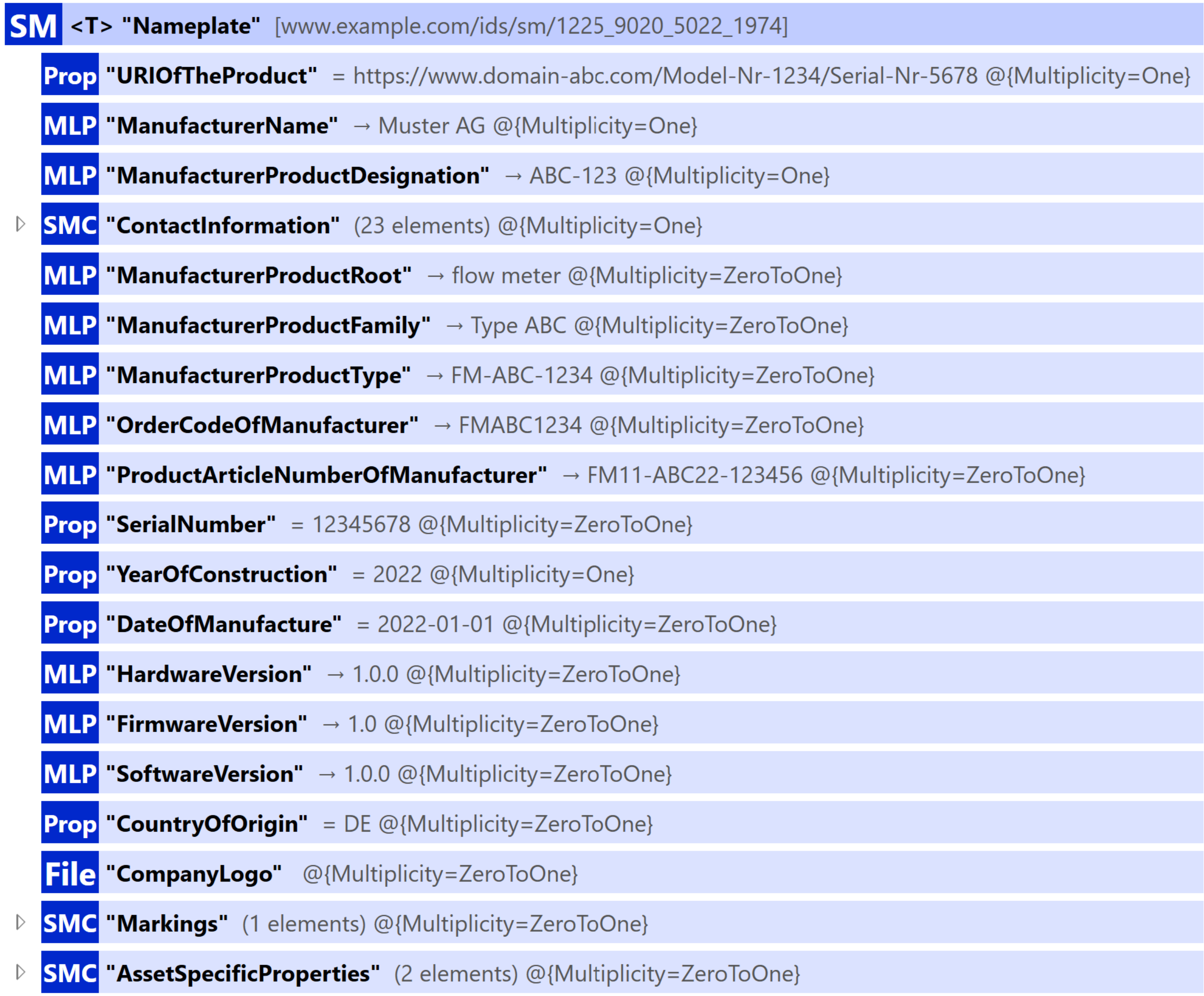
Find the process described here. For further questions, contact us via email.

Number of our submodels: 77

Contact Information	02002	1.0	Published	Download → GitHub →
Generic Frame for Technical Data for Industrial Equipment in Manufacturing	02003	1.2	Published	Download → GitHub →

https://industrialdigitaltwin.org/en/content-hub/submodels

Submodel Templates: Nameplate, Contact Information, …



SM <T> "Nameplate" [www.example.com/ids/sm/1225_9020_5022_1974]

Prop "URIOfTheProduct" = https://www.domain-abc.com/Model-Nr-1234/Serial-Nr-5678 @{Multiplicity=One}

MLP "ManufacturerName" → Muster AG @{Multiplicity=One}

MLP "ManufacturerProductDesignation" → ABC-123 @{Multiplicity=One}

▷ SMC "ContactInformation" (23 elements) @{Multiplicity=One}

MLP "ManufacturerProductRoot" → flow meter @{Multiplicity=ZeroToOne}

MLP "ManufacturerProductFamily" → Type ABC @{Multiplicity=ZeroToOne}

MLP "ManufacturerProductType" → FM-ABC-1234 @{Multiplicity=ZeroToOne}

MLP "OrderCodeOfManufacturer" → FMABC1234 @{Multiplicity=ZeroToOne}

MLP "ProductArticleNumberOfManufacturer" → FM11-ABC22-123456 @{Multiplicity=ZeroToOne}

Prop "SerialNumber" = 12345678 @{Multiplicity=ZeroToOne}

Prop "YearOfConstruction" = 2022 @{Multiplicity=One}

Prop "DateOfManufacture" = 2022-01-01 @{Multiplicity=ZeroToOne}

MLP "HardwareVersion" → 1.0.0 @{Multiplicity=ZeroToOne}

MLP "FirmwareVersion" → 1.0 @{Multiplicity=ZeroToOne}

MLP "SoftwareVersion" → 1.0.0 @{Multiplicity=ZeroToOne}

Prop "CountryOfOrigin" = DE @{Multiplicity=ZeroToOne}

File "CompanyLogo" @{Multiplicity=ZeroToOne}

▷ SMC "Markings" (1 elements) @{Multiplicity=ZeroToOne}

▷ SMC "AssetSpecificProperties" (2 elements) @{Multiplicity=ZeroToOne}

Related Work

Open-Source AAS hosting solutions only works with JSON [1]

Semantic Digital Twin and RDF format also explored [2]

Converting RDF-based model to AAS, leverage semantic interoperability [3]

Finding appropriate Submodel for application scenario via RDF, SPARQL [4]

[1] Jacoby, Michael, Michael Baumann, Tino Bischoff, Hans Mees, Jens Müller, Ljiljana Stojanovic, and Friedrich Volz. "**Open-Source Implementations of the Reactive Asset Administration Shell: A Survey**." *Sensors* 23, no. 11 (2023): 5229.
[2] Lim, Mei Qi, Xiaonan Wang, Oliver Inderwildi, and Markus Kraft. "**The world avatar—A world model for facilitating interoperability.**" In *Intelligent Decarbonisation: Can Artificial Intelligence and Cyber-Physical Systems Help Achieve Climate Mitigation Targets?*, pp. 39-53. Cham: Springer International Publishing, 2022.
[3] Rongen, Sjoerd, Nikoletta Nikolova, and Mark van der Pas. "**Modelling with AAS and RDF in Industry 4.0."** *Computers in Industry* 148 (2023): 103910.
[4] Bouter, Cornelis, Monireh Pourjafarian, Leon Simar, and Robert Wilterdink. "**Towards a comprehensive methodology for modelling submodels in the industry 4.0 asset administration shell**." In *2021 IEEE 23rd Conference on Business Informatics (CBI)*, vol. 2, pp. 10-19. IEEE, 2021.

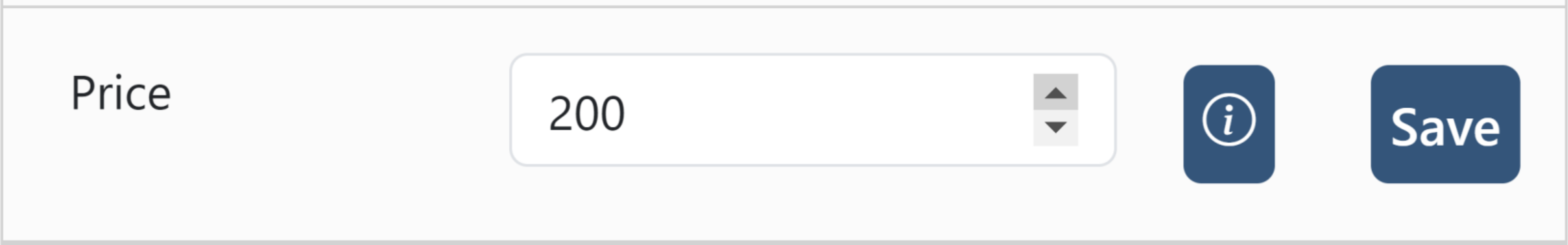
Why?

What about the data?

Heterogenous data source: Human, Machines, ..

Poor data quality -> No interoperability

Price: two hundred, 200, 200 $, 200 €, 200,00 , 200.00



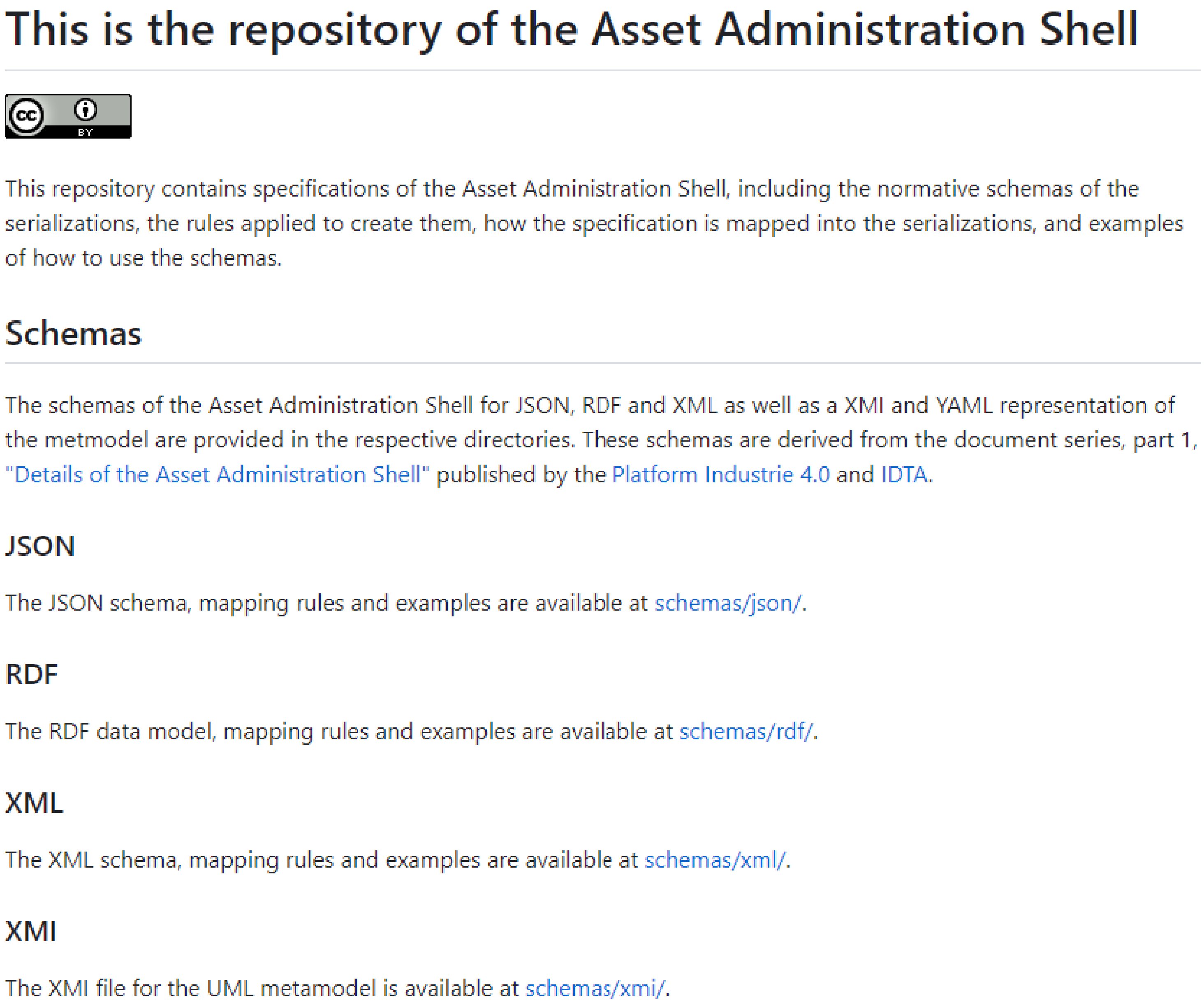
Price 200

Schema for Metamodel

Metamodel Compliance

Check JSON/XML/RDF …

Why not go one step further?



This is the repository of the Asset Administration Shell

This repository contains specifications of the Asset Administration Shell, including the normative schemas of the serializations, the rules applied to create them, how the specification is mapped into the serializations, and examples of how to use the schemas.

Schemas

The schemas of the Asset Administration Shell for JSON, RDF and XML as well as a XMI and YAML representation of the metmodel are provided in the respective directories. These schemas are derived from the document series, part 1, "Details of the Asset Administration Shell" published by the Platform Industrie 4.0 and IDTA.

JSON

The JSON schema, mapping rules and examples are available at schemas/json/.

RDF

The RDF data model, mapping rules and examples are available at schemas/rdf/.

XML

The XML schema, mapping rules and examples are available at schemas/xml/.

XMI

The XMI file for the UML metamodel is available at schemas/xmi/.

https://github.com/admin-shell-io/aas-specs

XML Schema, JSON schema, SHACL

All for checking constraints

JSON, XML mostly suitable for context-free constraints

Co-occurrence (cross-fields) and conditional constraints not so trivial.

External expansions: XML Schematron[1]

Specification expansion: If-then-else added in JSON Schema draft 7.

[1]: https://www.schematron.com/

How?

Custom rules

+ Easier to use

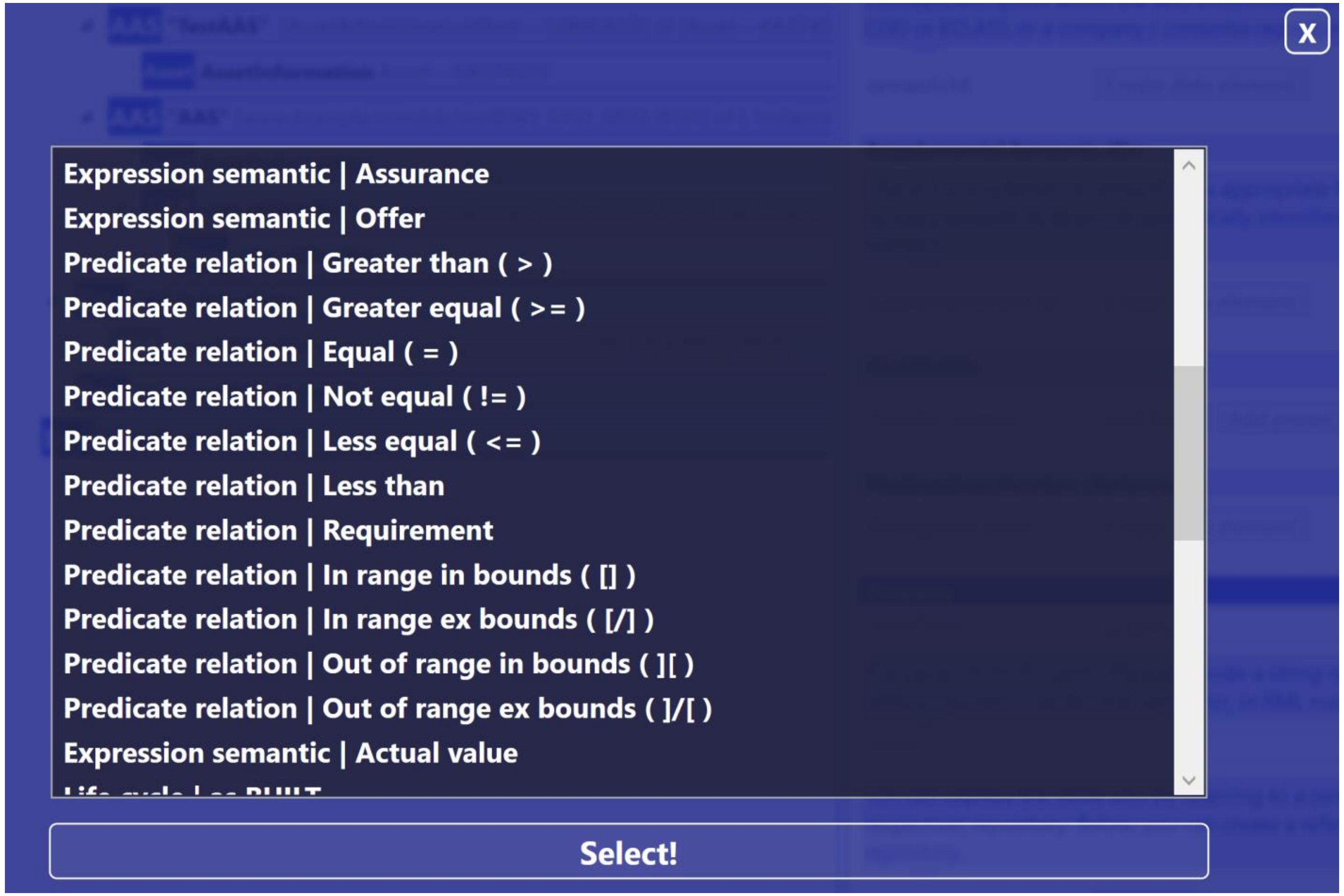
- Implementation

JSON/XML schema

+ Standard approach

+ Existing validators

- Suitable for context-free rules



Expression semantic | Assurance
Expression semantic | Offer
Predicate relation | Greater than (>)
Predicate relation | Greater equal (>=)
Predicate relation | Equal (=)
Predicate relation | Not equal (!=)
Predicate relation | Less equal (<=)
Predicate relation | Less than
Predicate relation | Requirement
Predicate relation | In range in bounds ([])
Predicate relation | In range ex bounds ([/])
Predicate relation | Out of range in bounds ( ][ )
Predicate relation | Out of range ex bounds ( ]/[ )
Expression semantic | Actual value
Life cycle | as BUILT

AASX Package Explorer – Preset Qualifiers

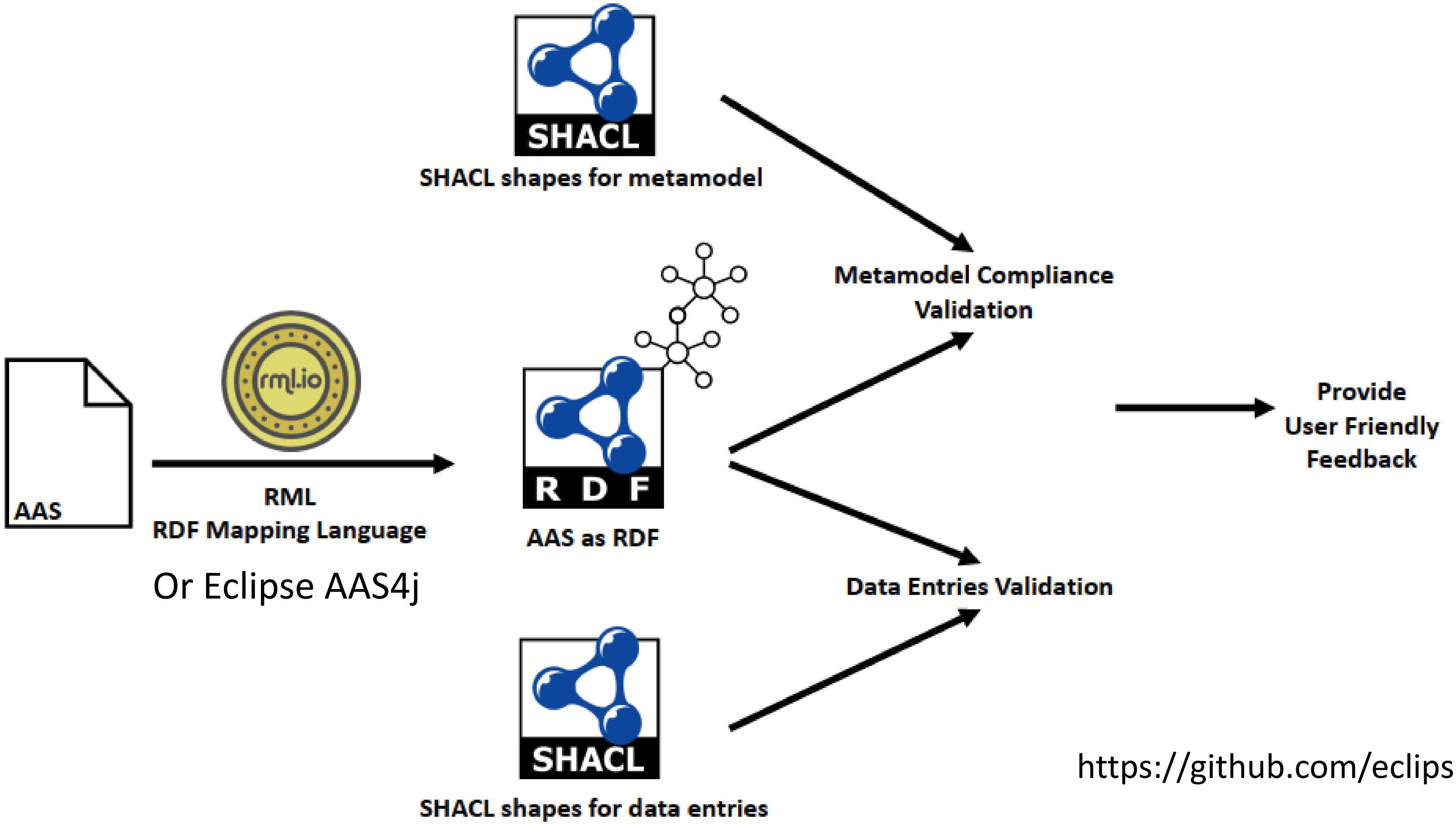
Our Workflow

Shape Constraint Language (SHACL)

+ Standard
+ Existing validators
+ Standard validation result output
+ Leverage Semantic Web tools

  (Query, Reasoning, …)

Our Workflow



SHACL shapes for metamodel

AAS

RML
RDF Mapping Language

Or Eclipse AAS4j

AAS as RDF

Metamodel Compliance Validation

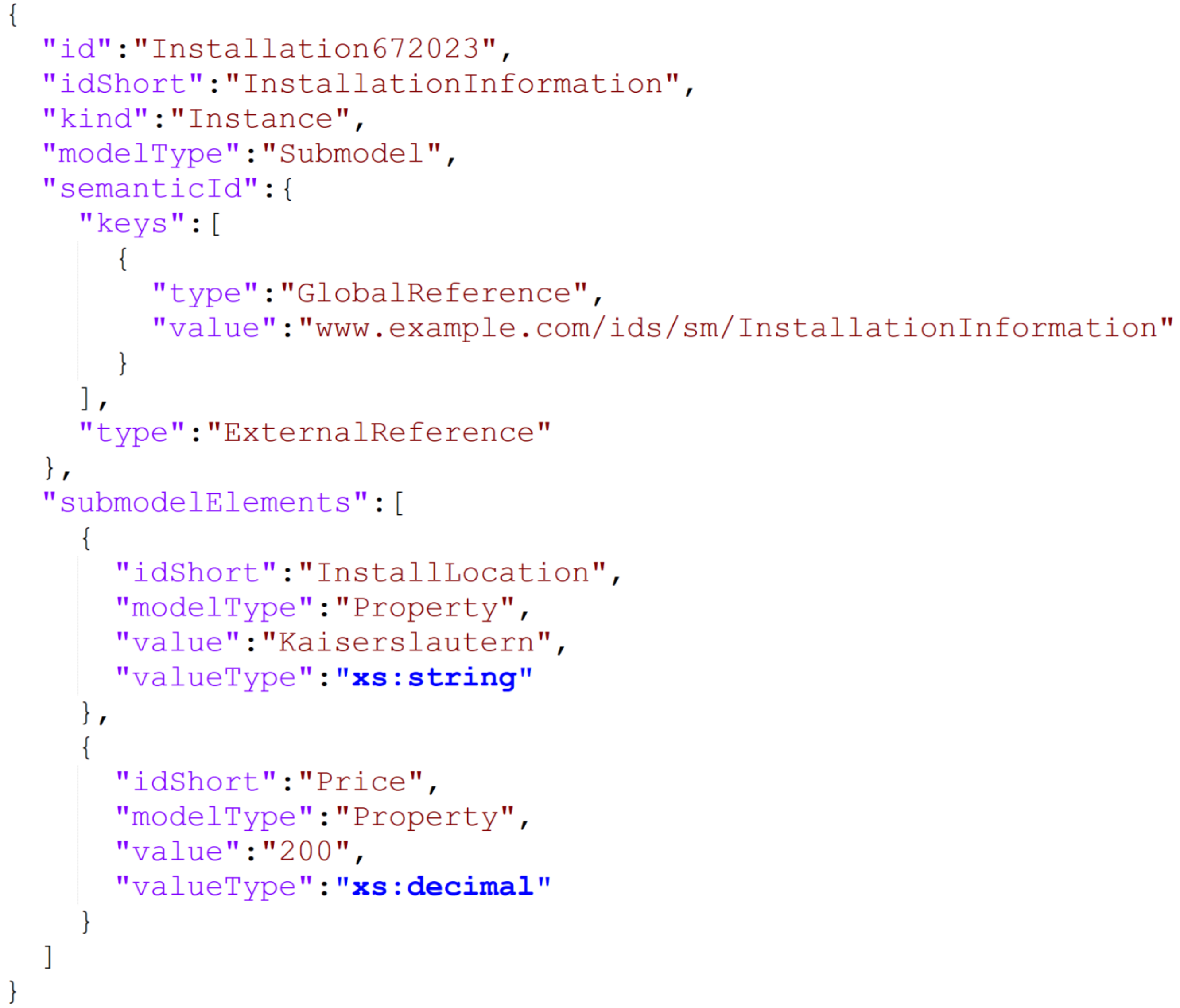
Data Entries Validation

SHACL shapes for data entries

Provide User Friendly Feedback

https://github.com/eclipse-aas4j/aas4j

JSON Representation



```
{
  "id":"Installation672023",
  "idShort":"InstallationInformation",
  "kind":"Instance",
  "modelType":"Submodel",
  "semanticId":{
    "keys":[
      {
        "type":"GlobalReference",
        "value":"www.example.com/ids/sm/InstallationInformation"
      }
    ],
    "type":"ExternalReference"
  },
  "submodelElements":[
    {
      "idShort":"InstallLocation",
      "modelType":"Property",
      "value":"Kaiserslautern",
      "valueType":"xs:string"
    },
    {
      "idShort":"Price",
      "modelType":"Property",
      "value":"200",
      "valueType":"xs:decimal"
    }
  ]
}
```

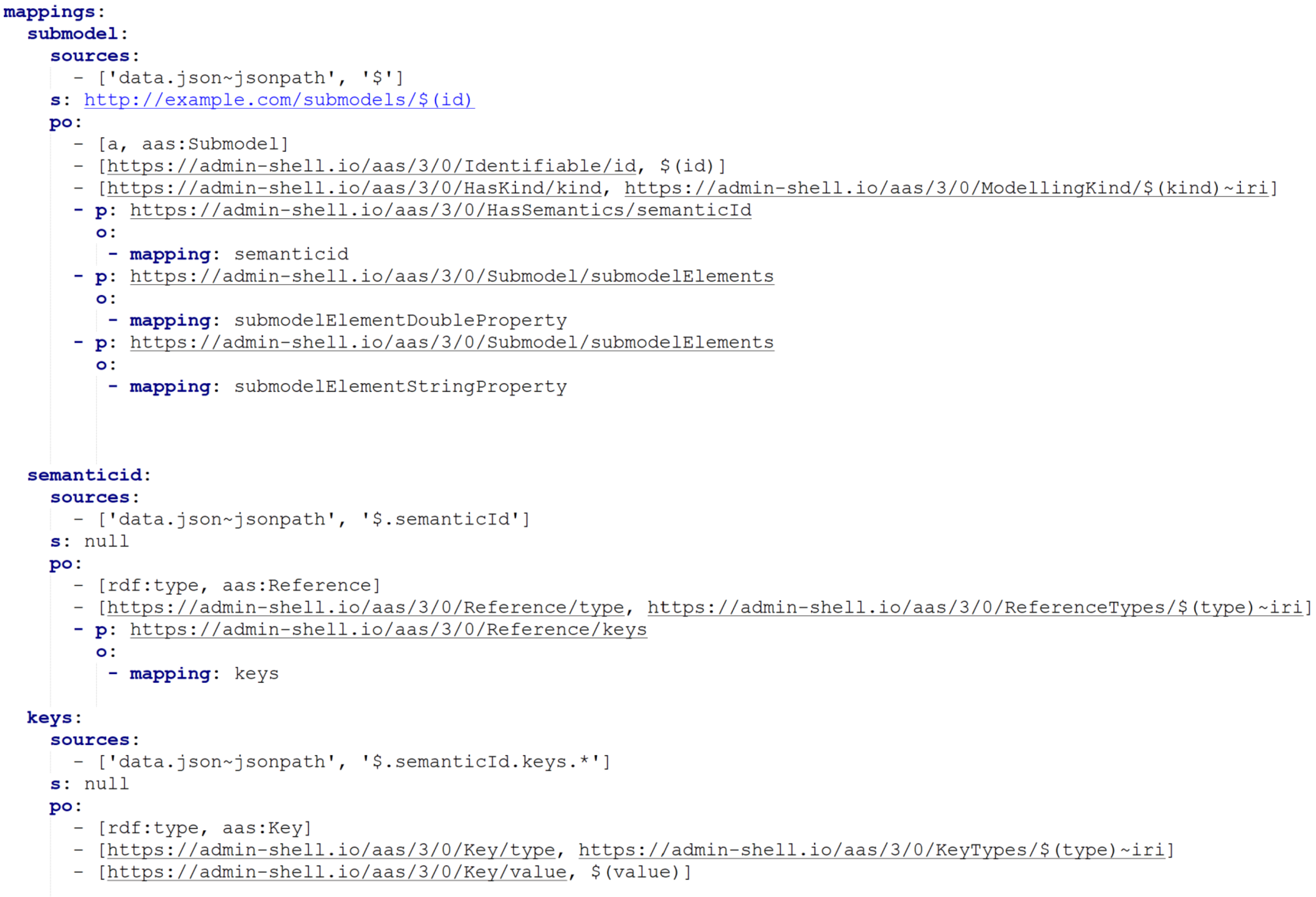
RML as a shortcut

RDF Mapping language (RML) rules to map a datasource (CSV, JSON, XML) to RDF triples.

How to generate <Subject, Predicate, Object>

YARRML

Human readable

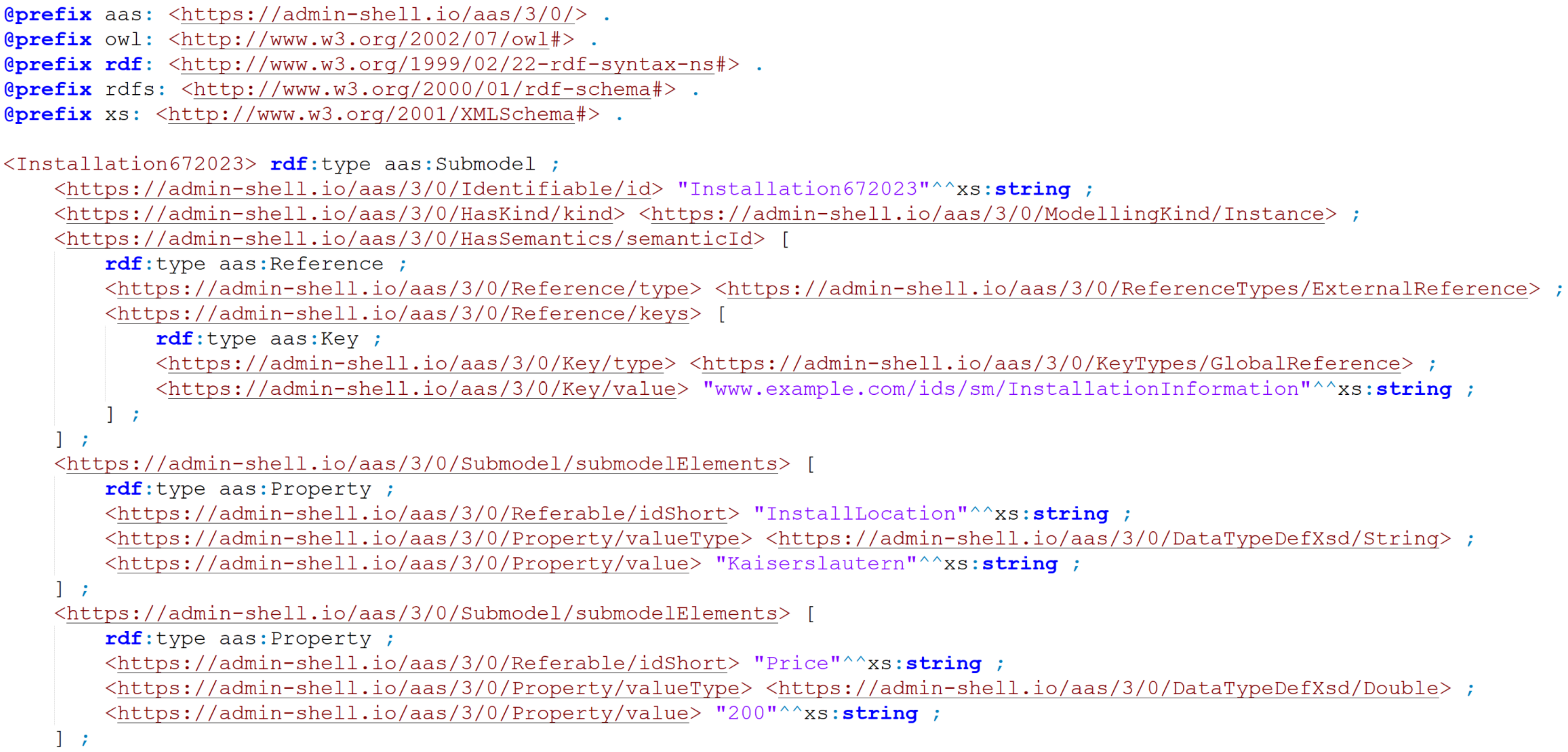


```
mappings:
  submodel:
    sources:
      - ['data.json~jsonpath', '$']
    s: http://example.com/submodels/$(id)
    po:
      - [a, aas:Submodel]
      - [https://admin-shell.io/aas/3/0/Identifiable/id, $(id)]
      - [https://admin-shell.io/aas/3/0/HasKind/kind, https://admin-shell.io/aas/3/0/ModellingKind/$(kind)~iri]
      - p: https://admin-shell.io/aas/3/0/HasSemantics/semanticId
        o:
         - mapping: semanticid
      - p: https://admin-shell.io/aas/3/0/Submodel/submodelElements
        o:
         - mapping: submodelElementDoubleProperty
      - p: https://admin-shell.io/aas/3/0/Submodel/submodelElements
        o:
         - mapping: submodelElementStringProperty


  semanticid:
    sources:
      - ['data.json~jsonpath', '$.semanticId']
    s: null
    po:
      - [rdf:type, aas:Reference]
      - [https://admin-shell.io/aas/3/0/Reference/type, https://admin-shell.io/aas/3/0/ReferenceTypes/$(type)~iri]
      - p: https://admin-shell.io/aas/3/0/Reference/keys
        o:
         - mapping: keys

  keys:
    sources:
      - ['data.json~jsonpath', '$.semanticId.keys.*']
    s: null
    po:
      - [rdf:type, aas:Key]
      - [https://admin-shell.io/aas/3/0/Key/type, https://admin-shell.io/aas/3/0/KeyTypes/$(type)~iri]
      - [https://admin-shell.io/aas/3/0/Key/value, $(value)]
```

RDF Representation

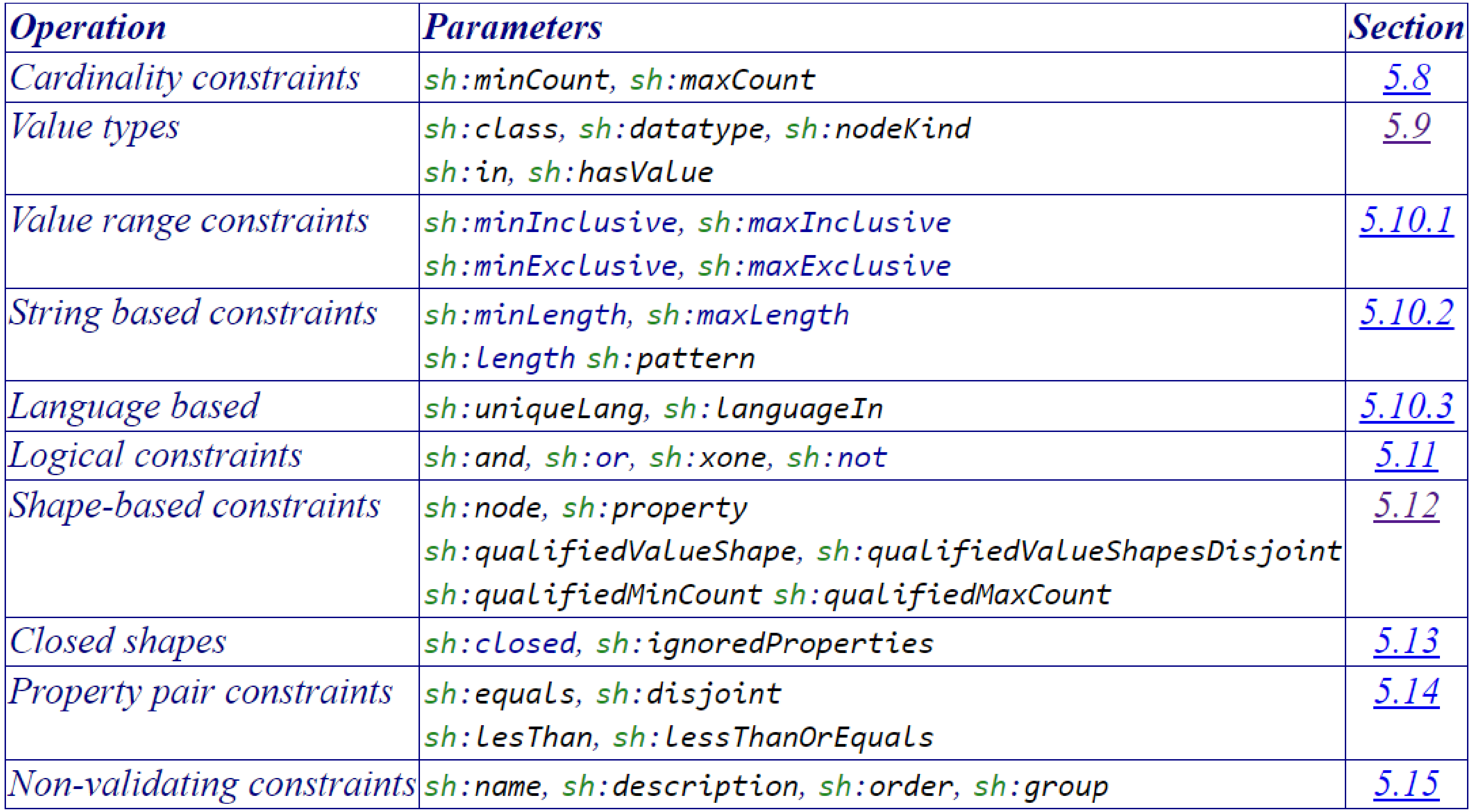


```
@prefix aas: <https://admin-shell.io/aas/3/0/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema#> .

<Installation672023> rdf:type aas:Submodel ;
    <https://admin-shell.io/aas/3/0/Identifiable/id> "Installation672023"^^xs:string ;
    <https://admin-shell.io/aas/3/0/HasKind/kind> <https://admin-shell.io/aas/3/0/ModellingKind/Instance> ;
    <https://admin-shell.io/aas/3/0/HasSemantics/semanticId> [
        rdf:type aas:Reference ;
        <https://admin-shell.io/aas/3/0/Reference/type> <https://admin-shell.io/aas/3/0/ReferenceTypes/ExternalReference> ;
        <https://admin-shell.io/aas/3/0/Reference/keys> [
            rdf:type aas:Key ;
            <https://admin-shell.io/aas/3/0/Key/type> <https://admin-shell.io/aas/3/0/KeyTypes/GlobalReference> ;
            <https://admin-shell.io/aas/3/0/Key/value> "www.example.com/ids/sm/InstallationInformation"^^xs:string ;
        ] ;
    ] ;
    <https://admin-shell.io/aas/3/0/Submodel/submodelElements> [
        rdf:type aas:Property ;
        <https://admin-shell.io/aas/3/0/Referable/idShort> "InstallLocation"^^xs:string ;
        <https://admin-shell.io/aas/3/0/Property/valueType> <https://admin-shell.io/aas/3/0/DataTypeDefXsd/String> ;
        <https://admin-shell.io/aas/3/0/Property/value> "Kaiserslautern"^^xs:string ;
    ] ;
    <https://admin-shell.io/aas/3/0/Submodel/submodelElements> [
        rdf:type aas:Property ;
        <https://admin-shell.io/aas/3/0/Referable/idShort> "Price"^^xs:string ;
        <https://admin-shell.io/aas/3/0/Property/valueType> <https://admin-shell.io/aas/3/0/DataTypeDefXsd/Double> ;
        <https://admin-shell.io/aas/3/0/Property/value> "200"^^xs:string ;
    ] ;
```

SHACL Core Components

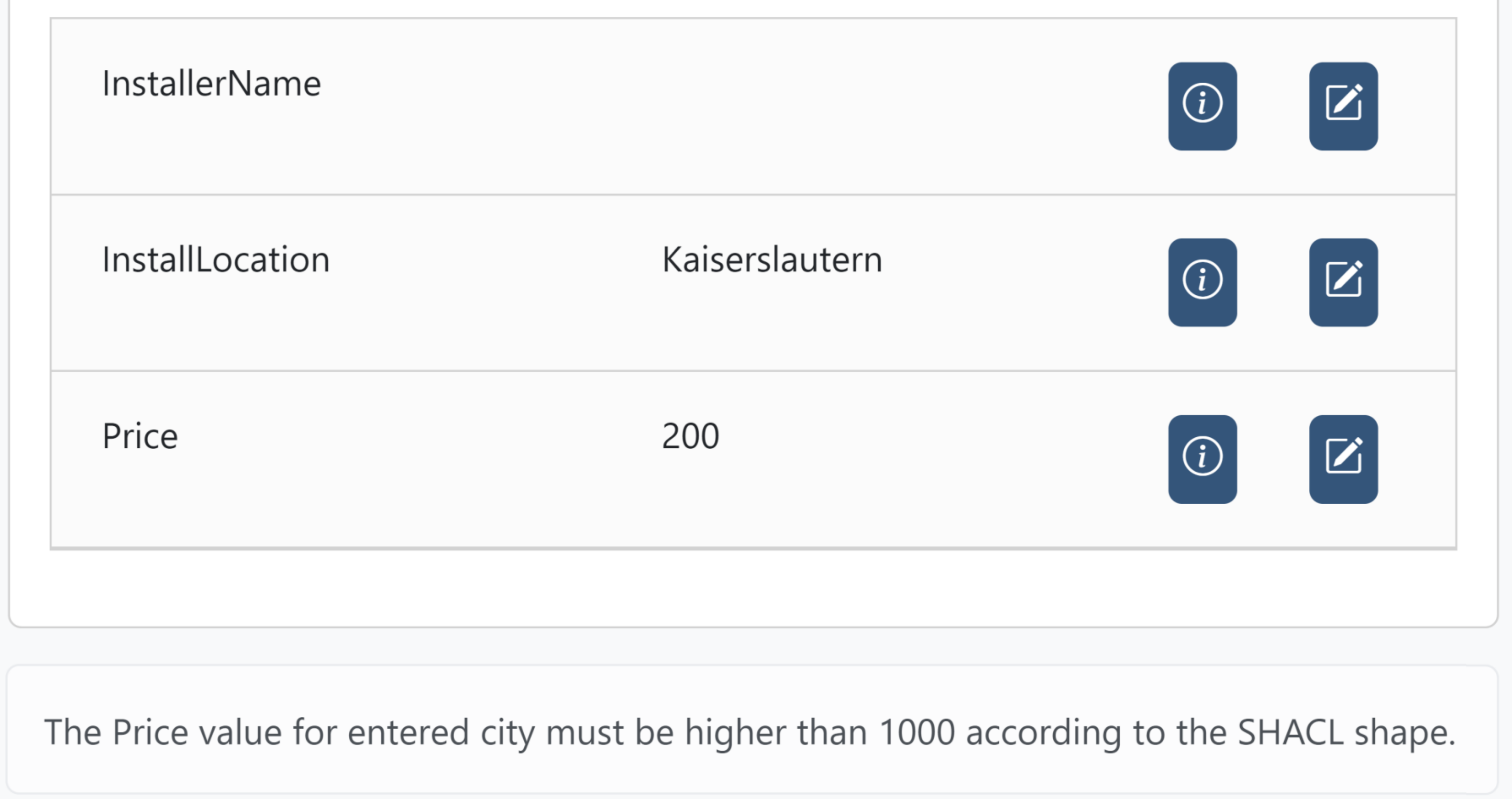
Table 5.3: SHACL core constraint components



Operation	Parameters	Section
Cardinality constraints	sh:minCount, sh:maxCount	5.8
Value types	sh:class, sh:datatype, sh:nodeKind sh:in, sh:hasValue	5.9
Value range constraints	sh:minInclusive, sh:maxInclusive sh:minExclusive, sh:maxExclusive	5.10.1
String based constraints	sh:minLength, sh:maxLength sh:length sh:pattern	5.10.2
Language based	sh:uniqueLang, sh:languageIn	5.10.3
Logical constraints	sh:and, sh:or, sh:xone, sh:not	5.11
Shape-based constraints	sh:node, sh:property sh:qualifiedValueShape, sh:qualifiedValueShapesDisjoint sh:qualifiedMinCount sh:qualifiedMaxCount	5.12
Closed shapes	sh:closed, sh:ignoredProperties	5.13
Property pair constraints	sh:equals, sh:disjoint sh:lesThan, sh:lessThanOrEquals	5.14
Non-validating constraints	sh:name, sh:description, sh:order, sh:group	5.15

https://book.validatingrdf.com/bookHtml011.html
https://www.utwente.nl/en/ces/sal/exams/digital-exams/Linked-Data-and-Semantic-Web/ldsw-lecture7.pdf

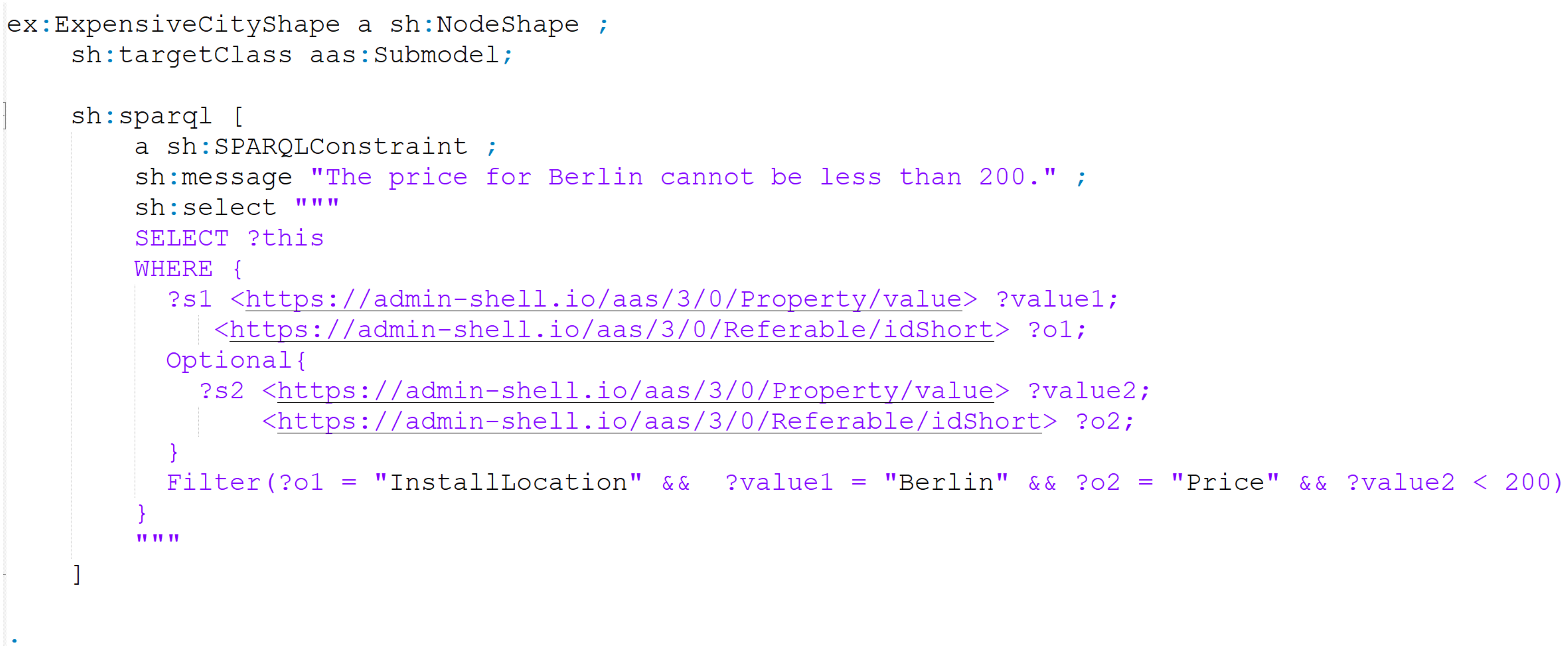
Cross-Fields Constraints



InstallerName		ⓘ	✎
InstallLocation	Kaiserslautern	ⓘ	✎
Price	200	ⓘ	✎

The Price value for entered city must be higher than 1000 according to the SHACL shape.

Example 1

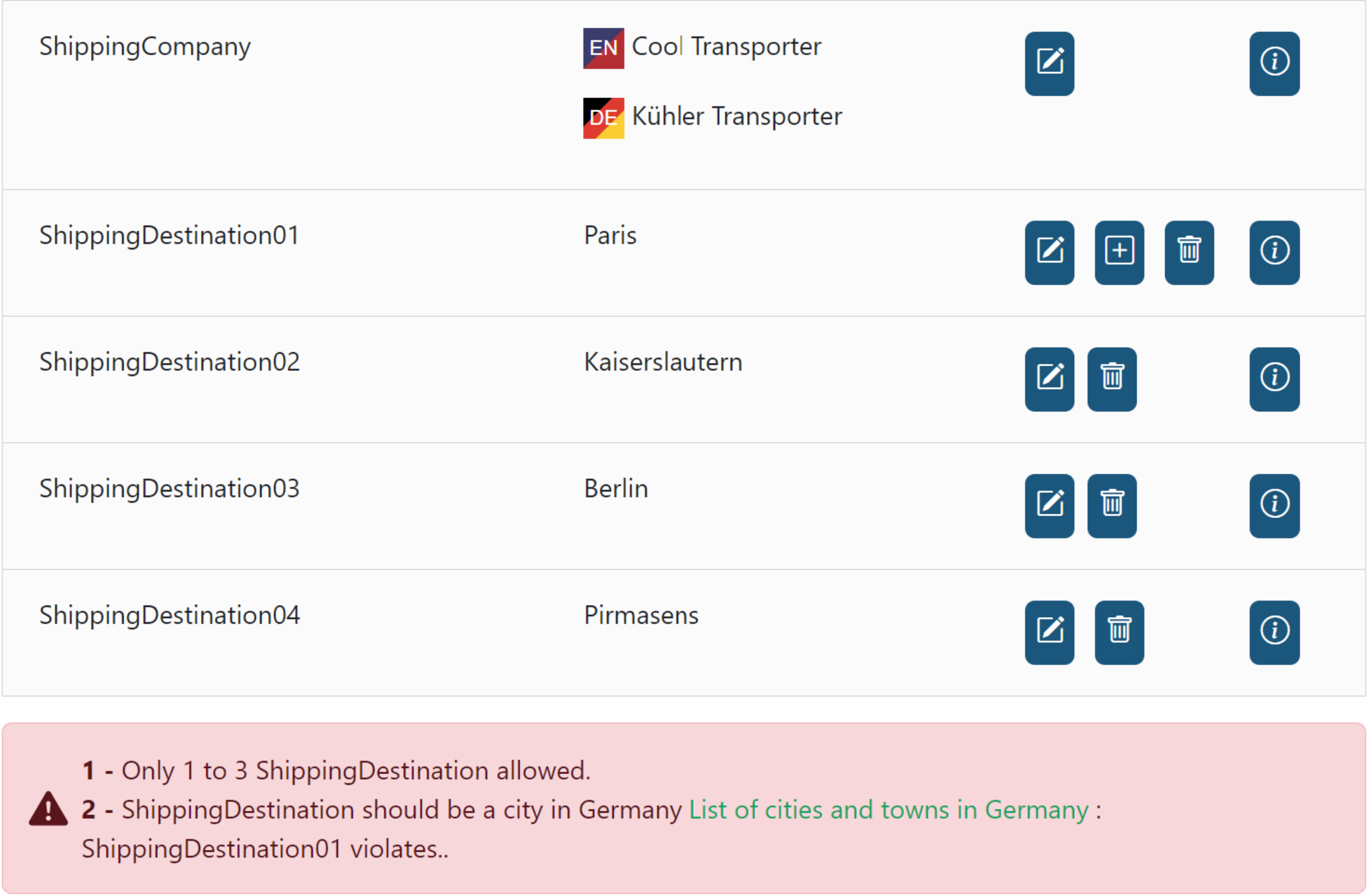


```
ex:ExpensiveCityShape a sh:NodeShape ;
    sh:targetClass aas:Submodel;

    sh:sparql [
        a sh:SPARQLConstraint ;
        sh:message "The price for Berlin cannot be less than 200." ;
        sh:select """
        SELECT ?this
        WHERE {
          ?s1 <https://admin-shell.io/aas/3/0/Property/value> ?value1;
              <https://admin-shell.io/aas/3/0/Referable/idShort> ?o1;
          Optional{
            ?s2 <https://admin-shell.io/aas/3/0/Property/value> ?value2;
                <https://admin-shell.io/aas/3/0/Referable/idShort> ?o2;
          }
          Filter(?o1 = "InstallLocation" &&  ?value1 = "Berlin" && ?o2 = "Price" && ?value2 < 200)
        }
        """
    ]

.
```

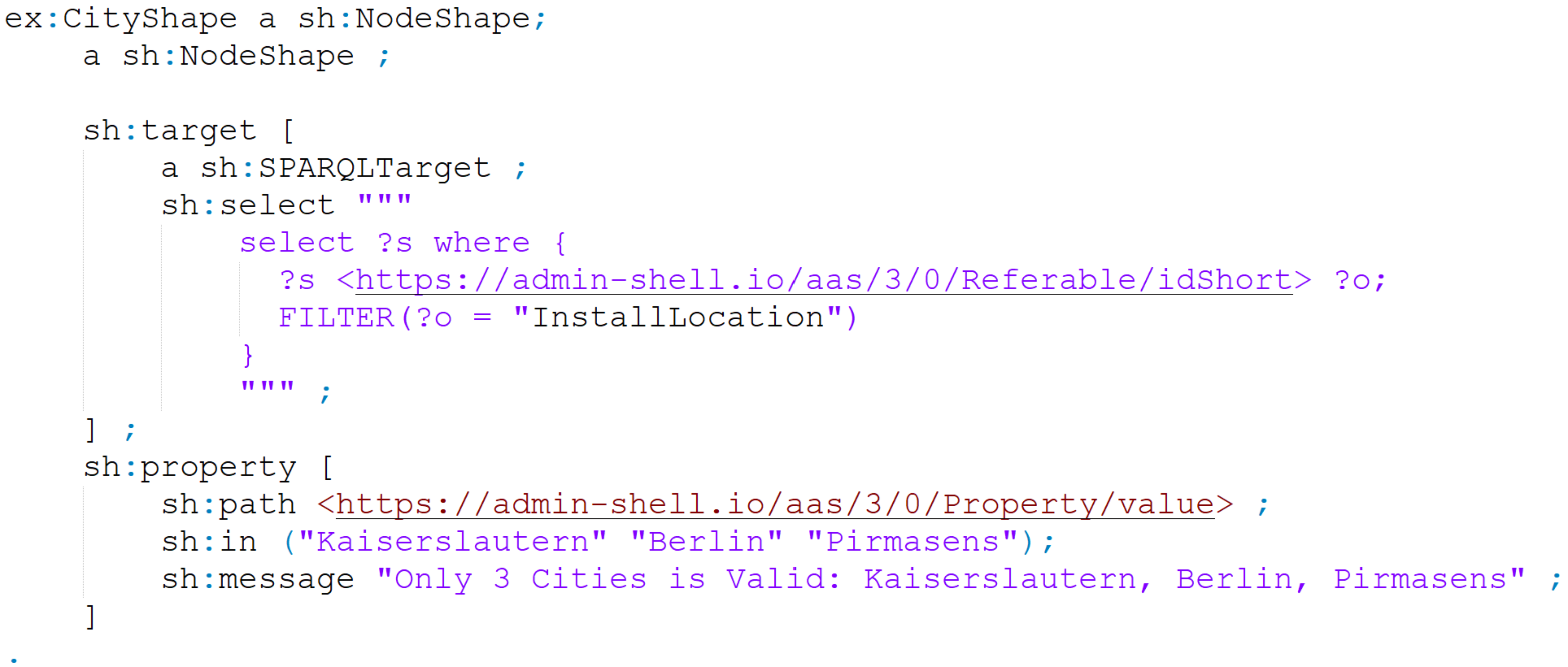
User Friendly



ShippingCompany	EN Cool Transporter DE Kühler Transporter
ShippingDestination01	Paris
ShippingDestination02	Kaiserslautern
ShippingDestination03	Berlin
ShippingDestination04	Pirmasens

⚠ **1 -** Only 1 to 3 ShippingDestination allowed.
2 - ShippingDestination should be a city in Germany List of cities and towns in Germany : ShippingDestination01 violates..

Example 2



```
ex:CityShape a sh:NodeShape;
    a sh:NodeShape ;

    sh:target [
        a sh:SPARQLTarget ;
        sh:select """
            select ?s where {
              ?s <https://admin-shell.io/aas/3/0/Referable/idShort> ?o;
              FILTER(?o = "InstallLocation")
            }
            """ ;
    ] ;
    sh:property [
        sh:path <https://admin-shell.io/aas/3/0/Property/value> ;
        sh:in ("Kaiserslautern" "Berlin" "Pirmasens");
        sh:message "Only 3 Cities is Valid: Kaiserslautern, Berlin, Pirmasens" ;
    ]
.
```

Where?

- Where to put this?
  - Flexible metamodel:
    - Qualifier
    - Extension
    - As a Property in Submodel
    - New Element
    - External mapping between Semantic ID and SHACL schema

Limitations

RML was a shortcut: only basic information of Properties

AAS structure is nested, RML cannot handle recursively nested structures

- Possible solutions:
  - RML Fields [1]

Custom parsers: Eclipse AAS4j, …

[1] Delva, Thomas, Dylan Van Assche, Pieter Heyvaert, Ben De Meester, and Anastasia Dimou. "Integrating Nested Data into Knowledge Graphs with RML Fields." In *KGWC2021, the Knowledge Graph Construction*, vol. 2873. 2021. [Link]

Takeaways

Data quality assurance <-> interoperability

Decision makers -> propose a location

Submodel Template creators :

- JSON Schema
- SHACL
- Let community decide

ReCircE Project

ReCircE

This work is funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection, project ReCircE, grant number 03EN2353B.

Bundesministerium
für Umwelt, Naturschutz
und Reaktorsicherheit

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

CIRCOON
Circular Economy Services

Fraunhofer
IWKS
Fraunhofer Research Institution for Materials Recycling
and Resource Strategies IWKS

GreenDelta

Reproducibility

bit.ly/etfa23a

Slides

Code (RML rules, SHACL examples)

Contact:

Mohammad Hossein Rimaz

hossein.rimaz@dfki.de

Thanks for your attention.

smartFactory KL®

DFKI | IFS Innovative Fabriksysteme