

DIL demonstration

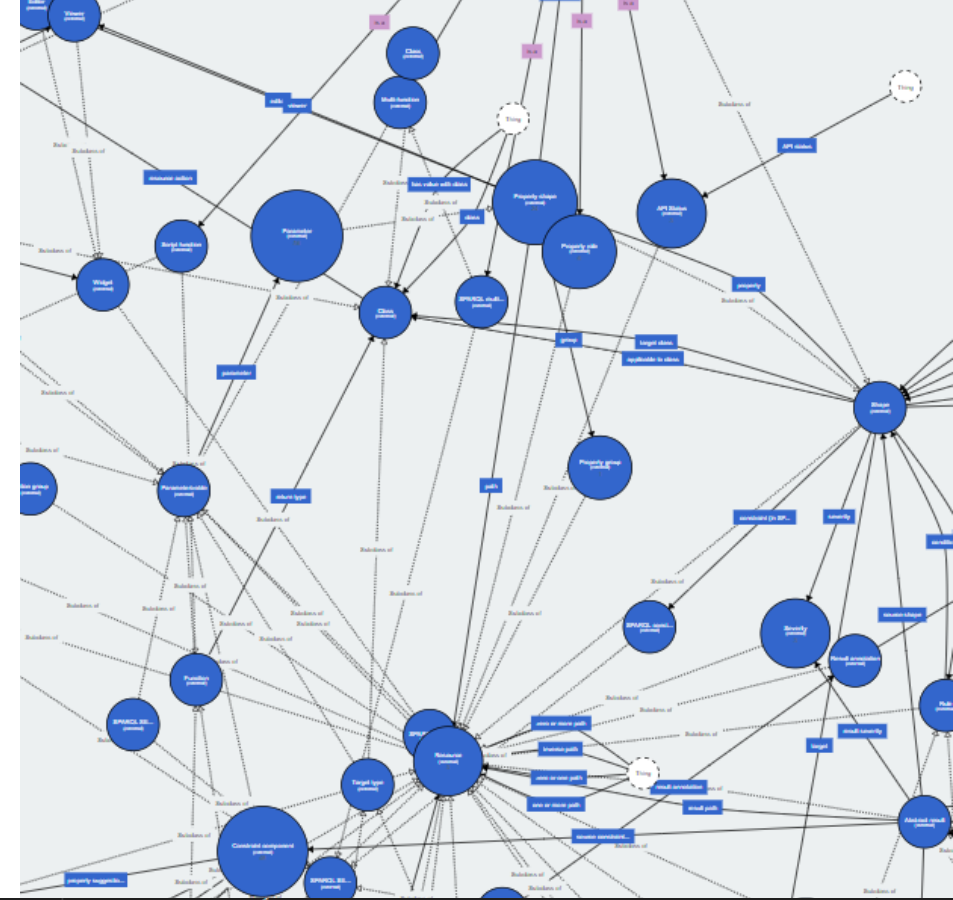
DIL project meeting

21 december 2023

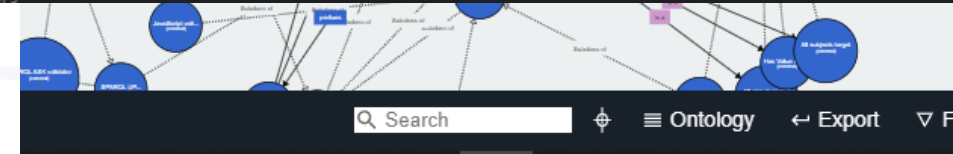


Input: the DCSA ontologie

- Ontology representation of DCSA model as provided by Jan and Eliana
- Files:
 - dil-dtt.ttl
 - dcsa_lists.ttl
- Inhoud:
 - 17 owl classes
 - 17 node shapes, 20 property shapes
 - 8 concept schemes and ~80 concepts (codes)
 - Namespace used:
 - <https://identifier.connekt.nl/dil/ont/dil-dtt/>
 - <https://identifier.connekt.nl/dil/ont/dil-dtt-lists/>
- Ontologie in STH geladen zonder aanpassingen

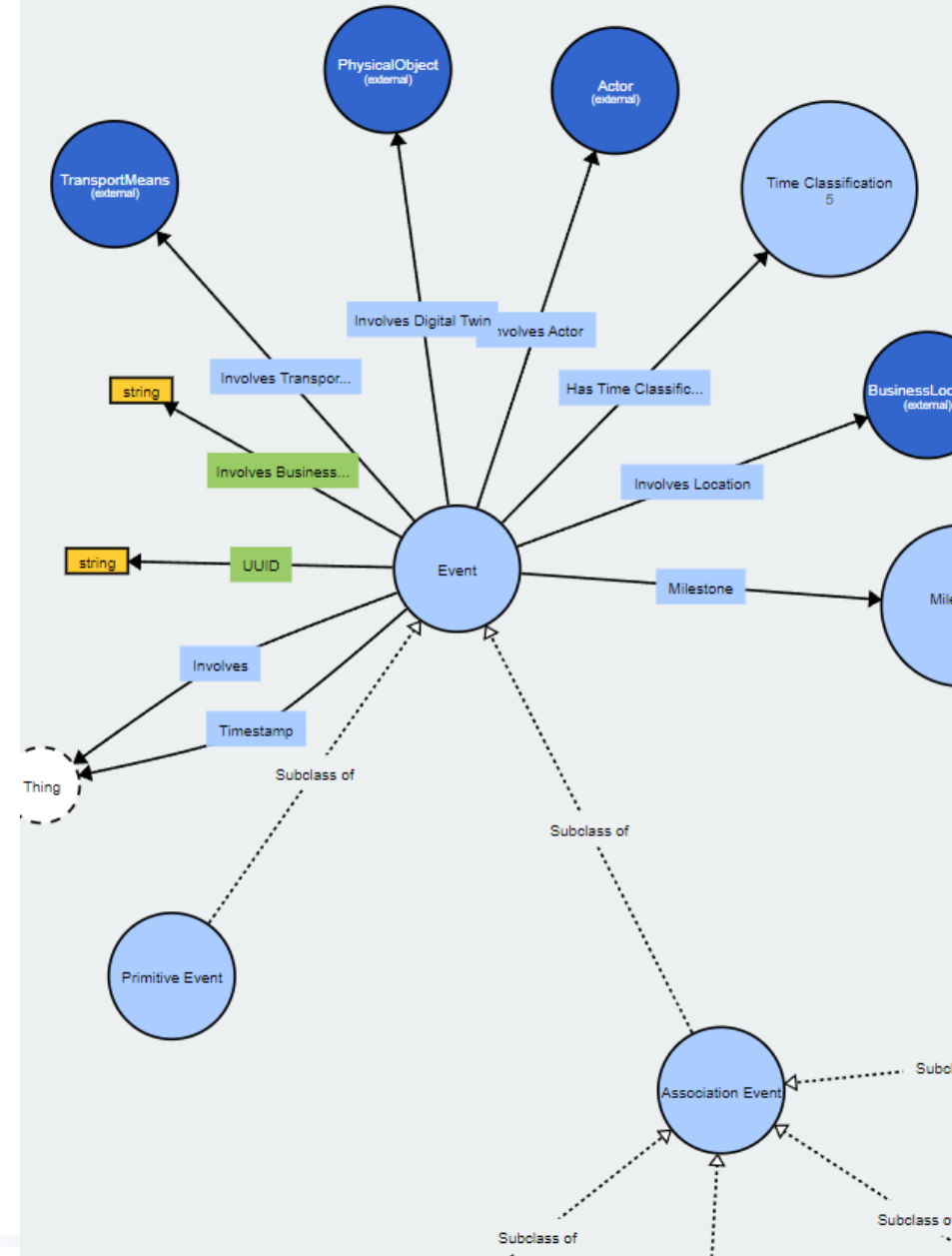


```
231 .sh:name "Document:UUID"@en ;
232 .
233 dil-dtt:TransportEvent
234 .a owl:Class ;
235 .a sh:NodeShape ;
236 .rdfs:comment "An event that pertains to a transport journey"@en ;
237 .rdfs:label "Transport event"@en ;
238 .rdfs:subClassOf dil-dtt:DCSAEvent ;
239 .sh:property dil-dtt:TransportEvent-atKindOfFacility ;
240 .sh:property dil-dtt:TransportEvent-eventKind ;
241 .sh:property dil-dtt:TransportEvent-perTransportMode ;
242 .#
243
```



Input: the FEDeRATED ontologie

- Verschillende modules:
 - Event ontologie
 - Digital Twin ontologie
 - Location ontologie
 - ...
- Namespaces:
 - federated: <https://ontology.tno.nl/logistics/federated/Event#>
 - ...
- Ontologie in STH geladen zonder aanpassingen
- Root class voor events: federated:PrimitiveEvent



Inputs imported into STH

**Semantic Treehouse**

[Specifications](#)

[Codelists](#)

[Validator](#)

[Issues](#)

[Groups](#)

[People](#)

[Organizations](#)

[Uploads](#)

tree view

DIL :

public

ONTOLOGY

DCSA ontology

Ontology representation of DCSA model as provided by Jan and Eliana

v0.1

Graph view • Export ttl 

ONTOLOGY

DIL alignment

v0.1

Graph view • Export ttl 

ONTOLOGY

DCSA lists

Ontology representation of DCSA lists as provided by Jan and Eliana

v0.1

Graph view • Export ttl 

FEDeRATED ontology :

public

ONTOLOGY

FEDeRATED - Event ontology

v0.1 - 2023-07-18

Graph view • Export ttl 

ONTOLOGY

FEDeRATED - Classifications

v0.1

Graph view • Export ttl 

ONTOLOGY

FEDeRATED - DigitalTwin

v0.1

Graph view • Export ttl 

ONTOLOGY

FEDeRATED - Location

v0.1

Graph view • Export ttl 

Event use cases

- Specify event message types
- Combining generic wrapper and domain specific payloads
 - Generic: FEDeRATED event
 - Domain specific: DCSA payload
- Explicitly **model the payload** for these event types; part of the demo are #4 (ETA) and #7 (Gate out)
- Output schema's** for different syntaxes (including XML, JSON and RDF)
- Schema can be used in API specs to **configure data space connectors**

		Event Message				
		Generic BDI Wrapper		Domain Specific Payload Data		
#	Event	Format	Data	Format	Data	Sender
1	Pallet is loaded into container into truck A	BDI	Event ID, Event Type: Loading Event Object: Pallet, Pallet ID Object : Container, Container ID Location: Warehouse A, Transport Means: Truck Time	GS1	EPCIS event Product data GTIN	Warehouse A
2	Estimated time of arrival of truck A at port A	BDI	Event ID, Event Type: ETA Object : Container, Container ID Location: Port A, Transport Means: Truck Time	OTM	OTM event Truck + Trailer ID Vehicle Data, Cargo Data Route Data, Shipment Data	Transporter A
3	Truck has arrived at port A, gate in	BDI	Event ID, Event Type: Gate in Object : Container, Container ID Location: Port A, Transport Means: Truck Time	DCSA	DCSA event Transport Journey Equipment Journey Shipment Journey	Port A, Terminal A
4	Estimated time of arrival of vessel at port B	BDI	Event ID, Event Type: ETA Object : Container, Container ID Location: Port B, Transport Means: Vessel Time	DCSA	DCSA event Transport Journey, Itinerary Equipment Journey Shipment Journey	Shipper
5	Estimated discharge time of container	BDI	Event ID, Event Type: ETA Object : Container, Container ID Location: Port B Time	DCSA	DCSA event Transport Journey Equipment Journey Shipment Journey	Port B, Terminal B
6	Customs cleared, ready for pickup	BDI	Event ID, Event Type: Customs Cleared Object : Container, Container ID Location: Port B Time	DCSA	DCSA event Transport Journey, Customs Data Equipment Journey Shipment Journey	Port B, Terminal B
7	Gate out, actual time of departure of truck B	BDI	Event ID, Event Type: Gate Out Object : Container, Container ID Location: Port B, Transport Means: Truck Time	DCSA	DCSA event Transport Journey Equipment Journey Shipment Journey	Port B, Terminal B
8	Estimated time of arrival of truck B at warehouse B	BDI	Event ID, Event Type: ETA Object : Container, Container ID Location: Warehouse B, Transport Means: Truck Time	OTM	OTM event Truck + Trailer ID Vehicle Data, Cargo Data Route Data, Shipment Data	Transporter B
9	Pallet unloaded from container at warehouse B	BDI	Event ID, Event Type: Unloading event Object: Pallet, Pallet ID Location: Warehouse B Time	GS1	EPCIS event Product Data GTIN	Warehouse B

Supplier

Warehouse A

Transporter A

Port A

Shipper

Port B

Transporter B

Cargo Owner

Warehouse



















1

2

3

4

5

6

7

8

9

innovation

Modelled event types (#4 and #7)

← → ↺ 🏠 dil.semantic-treehouse.nl/specifications



Semantic
Treehouse

🔗 Specifications

☰ Codelists

✓ Validator

🔍 Issues

👤 Groups

👤 People

📁 Organizations

📤 Uploads

Specifications ⋮

Event flow ⋮

public

MESSAGE

1. Loading Event

Pallet is loaded into container into truck A

v0.1

MESSAGE

2. ETA

Estimated time of arrival of truck A at port A

v0.1

MESSAGE

3. Gate in

Truck has arrived at port A, gate in

v0.1

MESSAGE

4. ETA

Estimated time of arrival of vessel at port B

v0.1

Tree view

MESSAGE

5. EDT

Estimated discharge time container

v0.1

MESSAGE

6. Customs cleared

Customs cleared, ready for pickup

v0.1

MESSAGE

7. Gate out

Gate out, actual time of departure of truck B

v0.1

Tree view

MESSAGE

8. ETA

Estimated time of arrival of truck B at warehouse B

v0.1

MESSAGE

9. Unloading

Pallet unloaded from container at warehouse B

v0.1

Event type #4 | ETA Event

FEDerATED

DCSA

4. ETA

0.1

CTRL+K

1...1 ETA Event

1...1 UUID

1...1 Event Type

1...1 Has Time Classification

1...1 Container

1...1 Container Number

1...1 Location

1...1 Vessel

0...1 Vessel Name

1...1 Timestamp

0...1 Transport Journey

1...1 Datetime

1...1 Event ID

1...1 TimelineStatus

1...1 Transport event kind

1...1 At kind of facility

0...1 Per transport mode

0...1 Equipment Journey

0...1 Shipment Journey

0...n Involves Business Identifier

0...n Involves Actor...

0...n Involves Transport Means...

Add all

Add next level descendants

Edit element

Usage notesSyntax bindingElement mappingDevelopmentTechn. details

Artefact

Element: ETA_Event (v0.1)

Property

Target class

<https://ontology.tno.nl/logistics/federated/Event#PrimitiveEvent>

Target basedatatype

Json pointer

Part of message

ETA_Event (4. ETA v0.1)

Parent element

Sequence number

Is visible

Is attribute

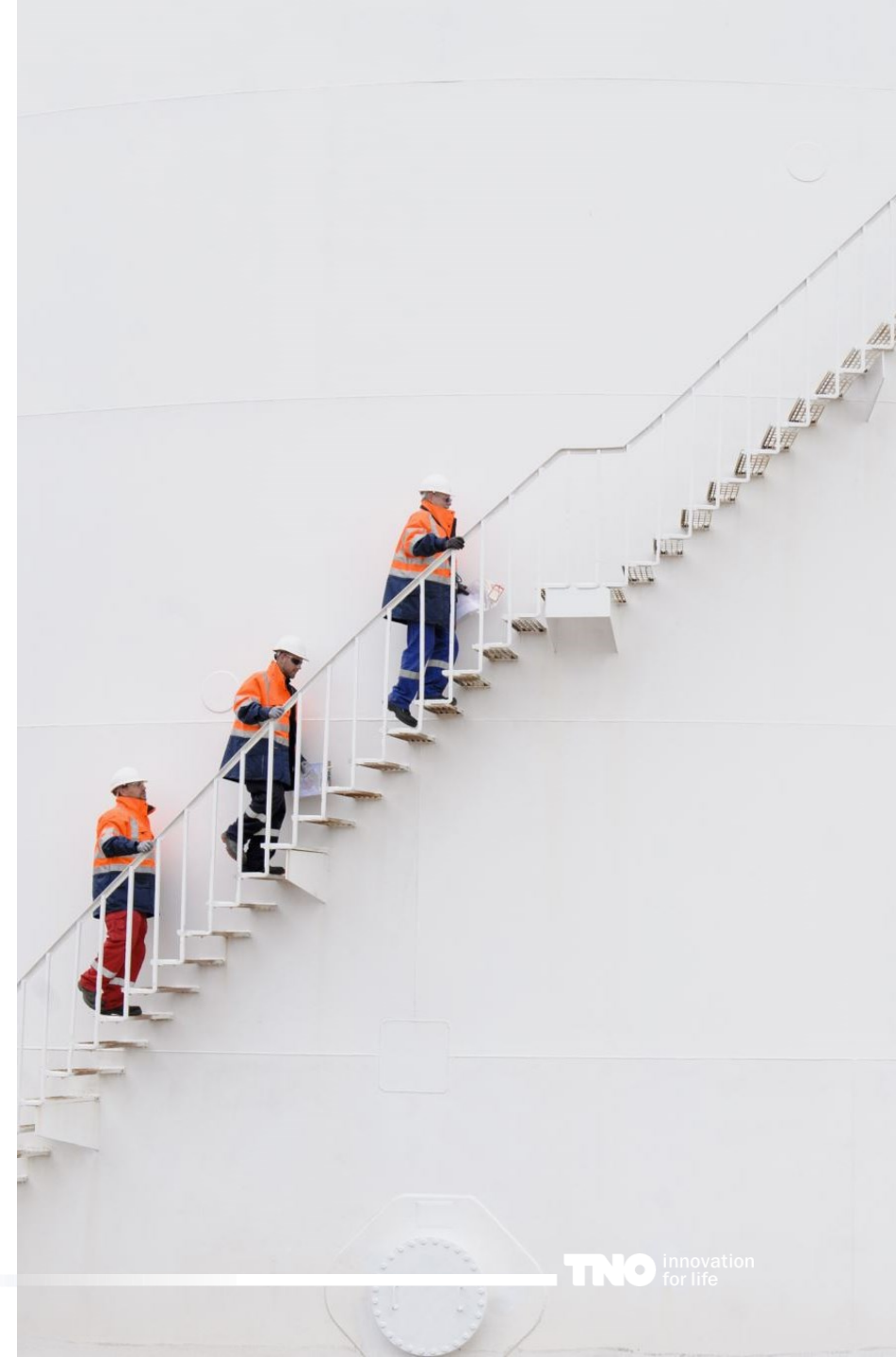
Is reference

Predecessor

Copy is processed

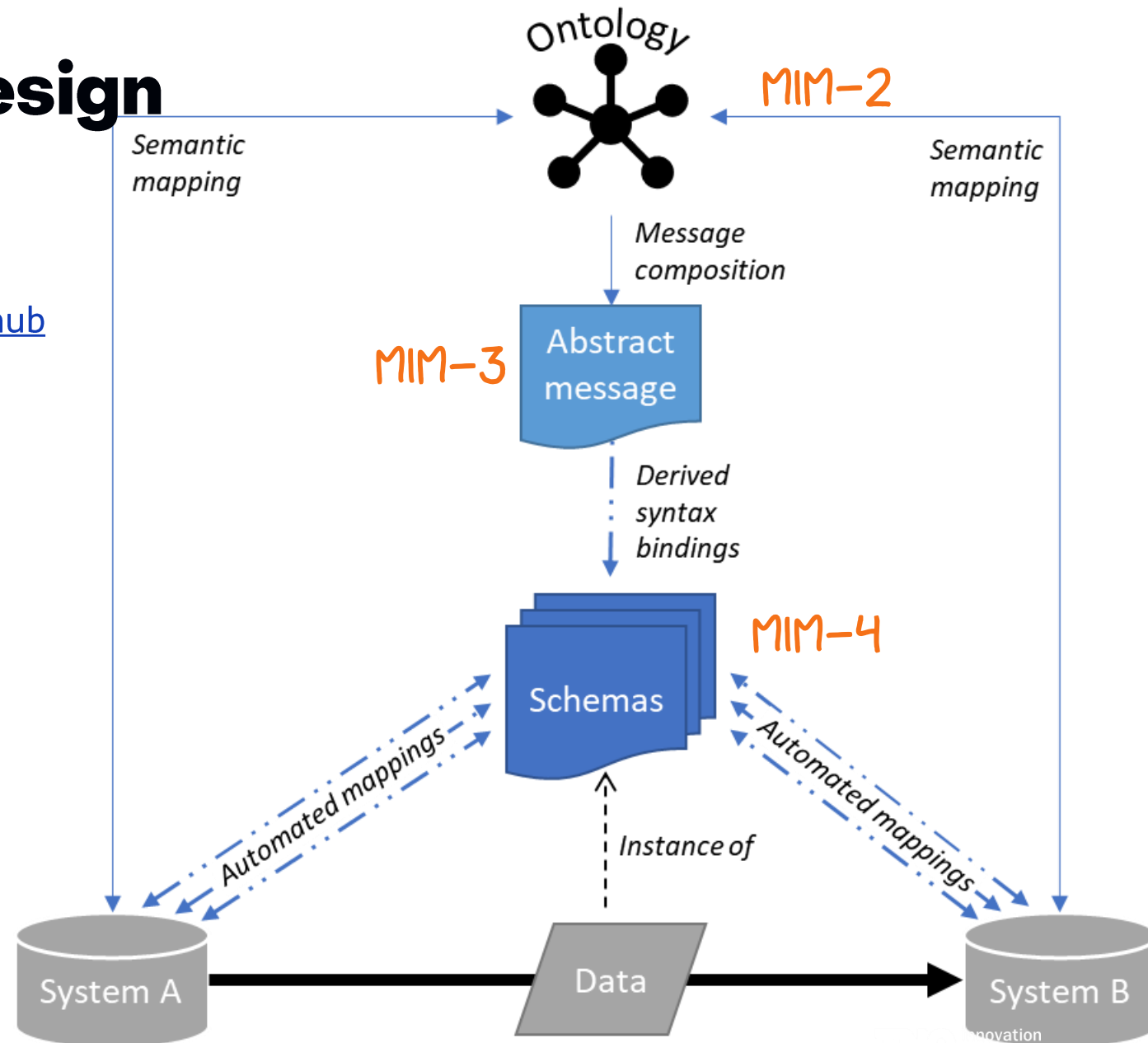
Next steps

- Write down (3-pager) what we realized
- Prepare for demonstration of the storyline for:
 - January: meeting with broader audience
 - February: Data Sharing Festival
- Finalize helm charts



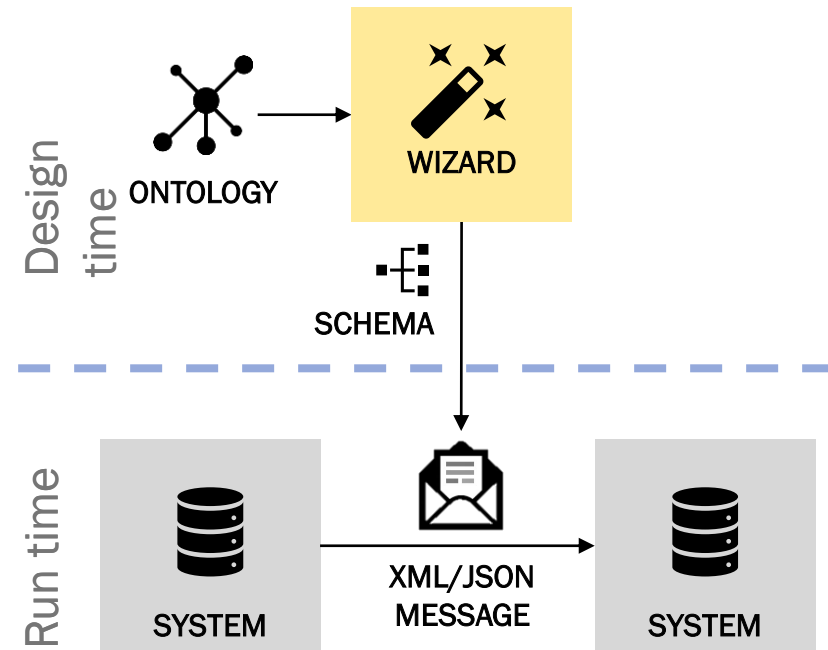
Ontology-driven API design method & tool

- FIT wizard artikel
- <https://www.semantic-treehouse.nl/vocabulary-hub>



Traditional approach

- Generate schemas for traditional message based data exchange
- Resulting schemas can be used for:
 - API specifications (e.g. OpenAPI)
 - Data validation
- The Wizard keeps link between schema elements and the classes/properties from ontology



Towards linked data

- The link between schema elements and ontology classes/properties is used to generate a mapping specification
- Mapping is formalized in RDF Mapping Language (RML)
- An RML engine is used to transform message data to triples according to the ontology
- This approach allows for a large set of heterogeneous data sources being mapped to the same ontology

