Vlaamse overheid

OSLO Passenger Transport Hubs:

Thematic workshop 2

Welcome!

Thursday 23rd of June 2022
Virtual workshop – Microsoft Teams

We start at 09:05am

Before we start…

Mute yourself during the meeting

Raise your **hand** before speaking. We encourage interaction!

Questions and suggestions can always be communicated via the **chat**.

yes/no questions can be answered via the chat:

Yes = +1
No = - 1
Neutral = 0

Objective

Giving a **summary** of the first thematic workshop

Explanation and elaboration on **discussion points of last session**

Presentation of and gathering of last remarks on **the revised model** for this data standard

Objectives of today

Presentation of the final data model

Summary of the last thematic working group

Presentation and discussion of the data model

Explanation of how to consult the model, application profile and vocabulary

Topics

09:00 - 09:10 AM	Welcome and introduction
09:10 - 09:20 AM	Recap of last workshop
09:20 - 09:50 AM	Passenger Transport Hubs model
9:50 - 10:00 AM	*Break*
10:00 - 10:15 AM	Publication PURL.eu
10:15 - 10:25 AM	Registering issues
10:25 - 10:45 AM	Q&A and next steps

Who is who?

Recap of the last thematic workshop

Topics of the last thematic workshop

Introduction to Unified Modeling Language

- An intro was given about the working of Unified Modeling Language
- Providing a basic knowledge of UML, in order that people understand how to read the model.

Presentation of the model

- The model was presented
- Focused on adjustments that we're made
  - Adjustment of spelling mistakes
  - Translation errors
- Presentation of page on PURL.eu

Discussion points

- Discussion of our views on the discussion points from the business working group.
  - How will we take it into account?
  - Which things are out of scope?
- Brainstorm about possible improvements of and adjustments to the model

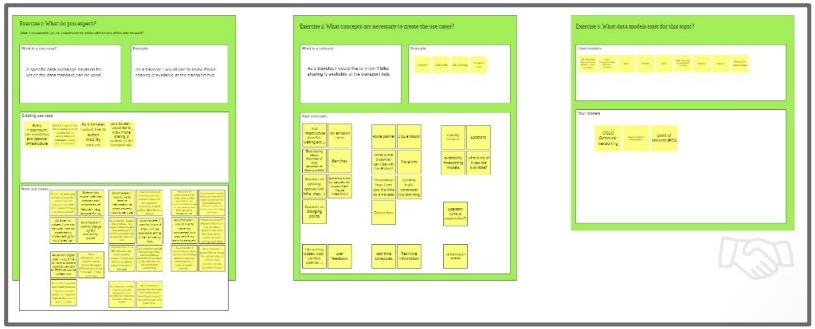
Scope of this project

Develop a semantic framework for mapping Passenger Transport Hubs and sharing data.

Develop a sustainable application profile and vocabulary for Passenger Transport Hubs.

We follow the OSLO methodology, which means:

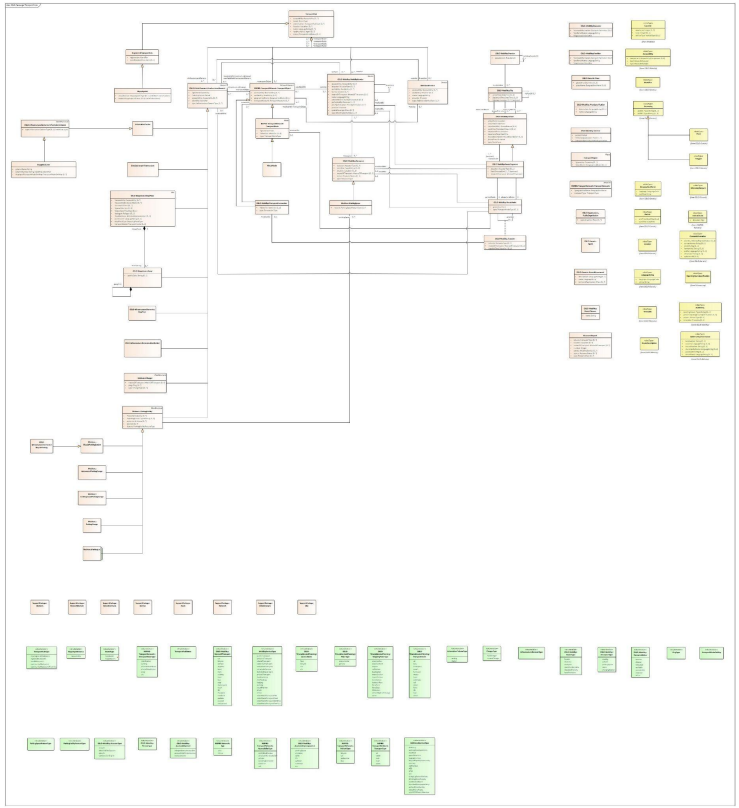
- We start from use cases
- We align as much as possible with existing standards
- We define cases ourselves where necessary



Passenger Transport Hubs model

Vlaanderen
verbeelding werkt

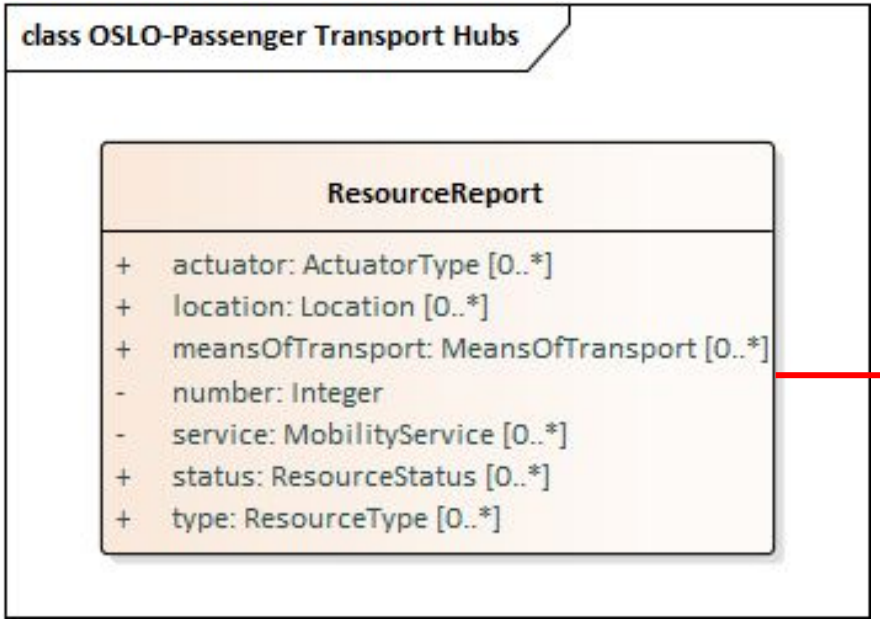
Full model

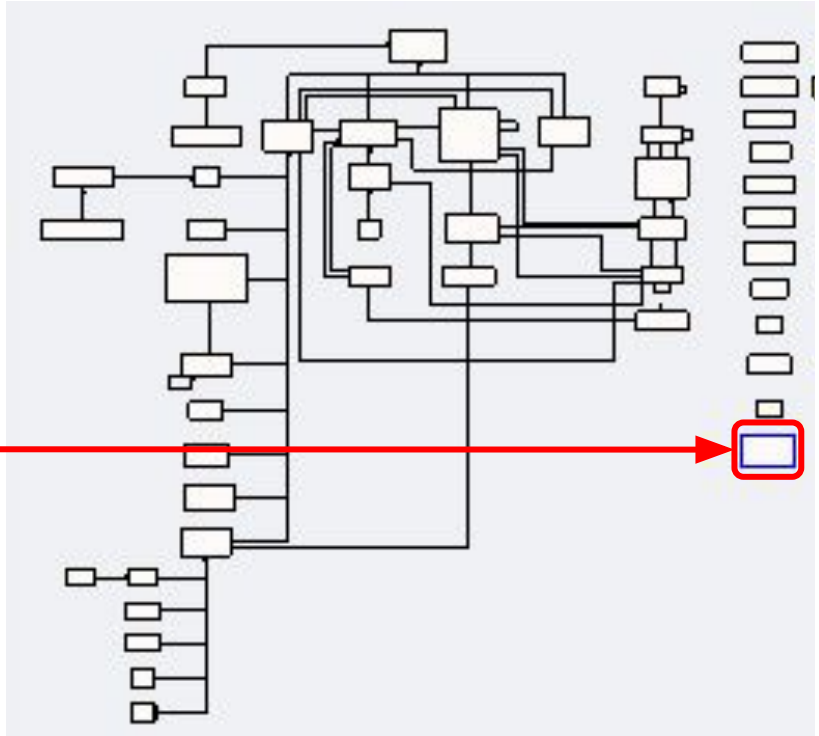


- Semantic description of a Transport Hub.
- Description of the role of a Transport Hub within the creation of a Traveler’s trip.

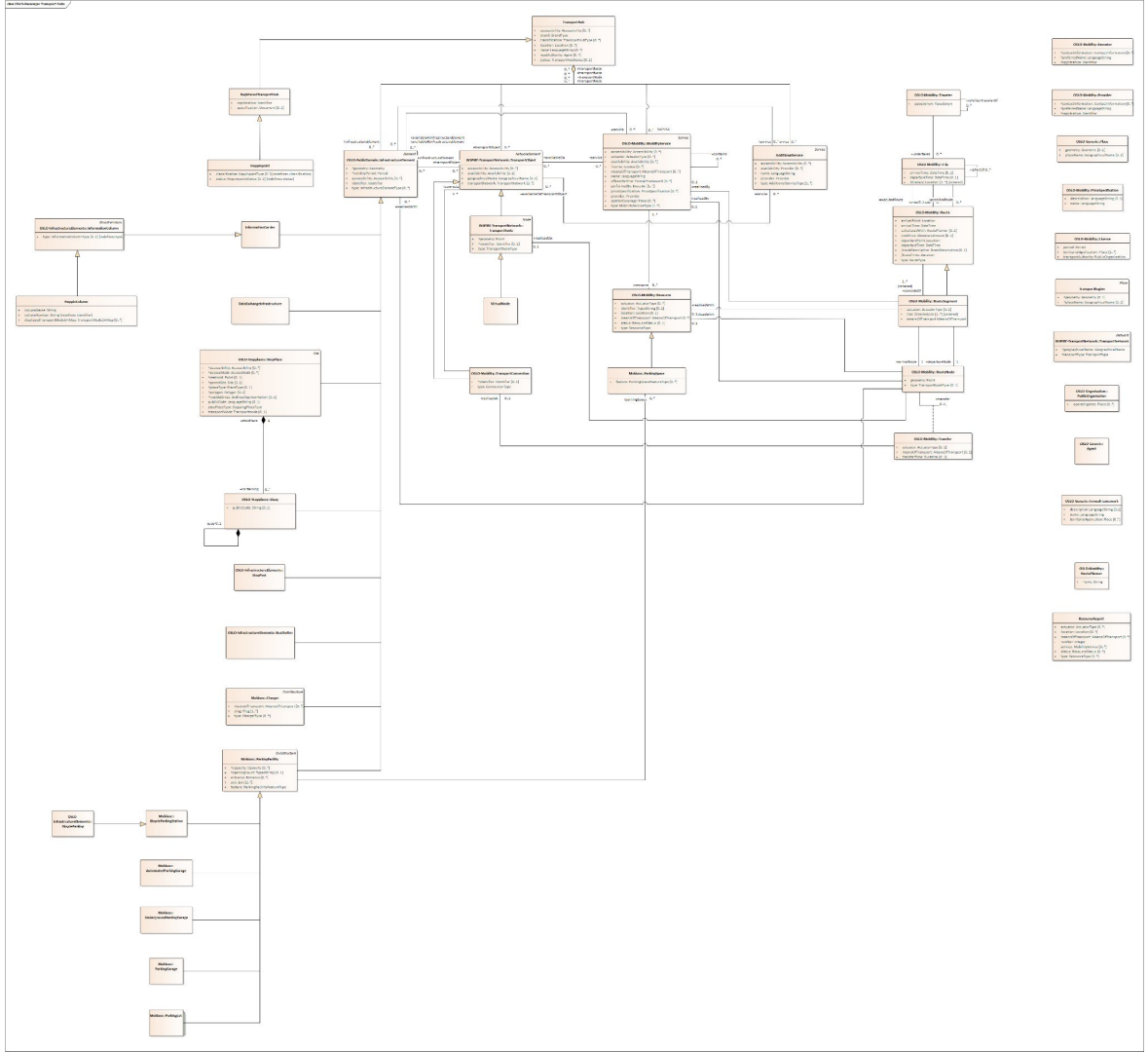
MURAL

Adjustments to the model

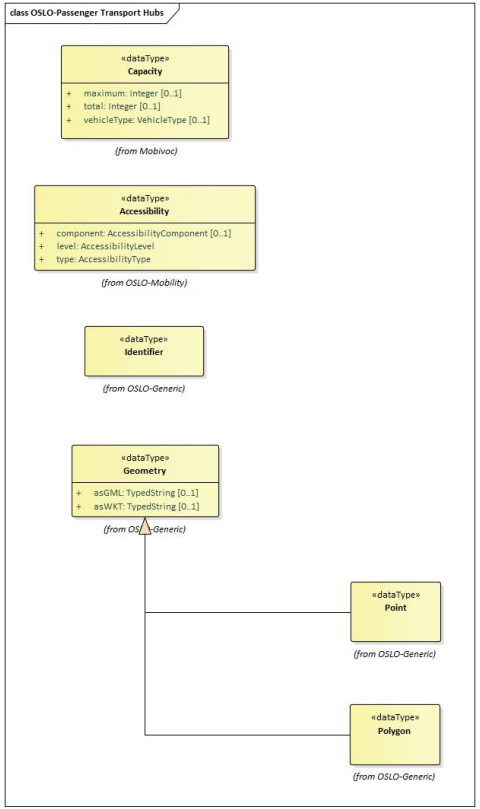


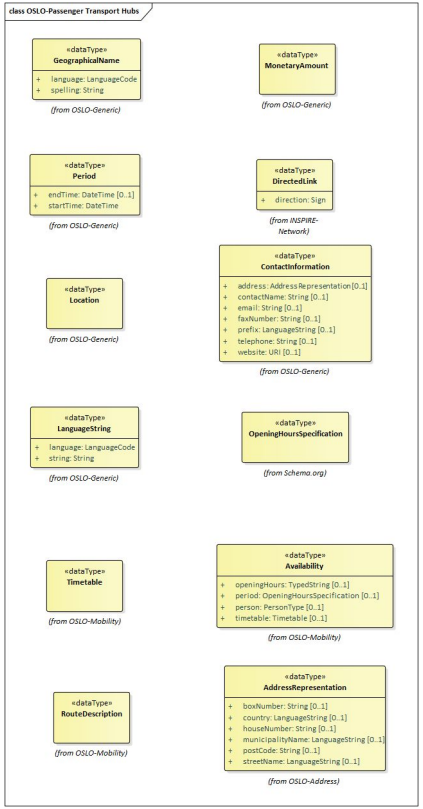


Classes

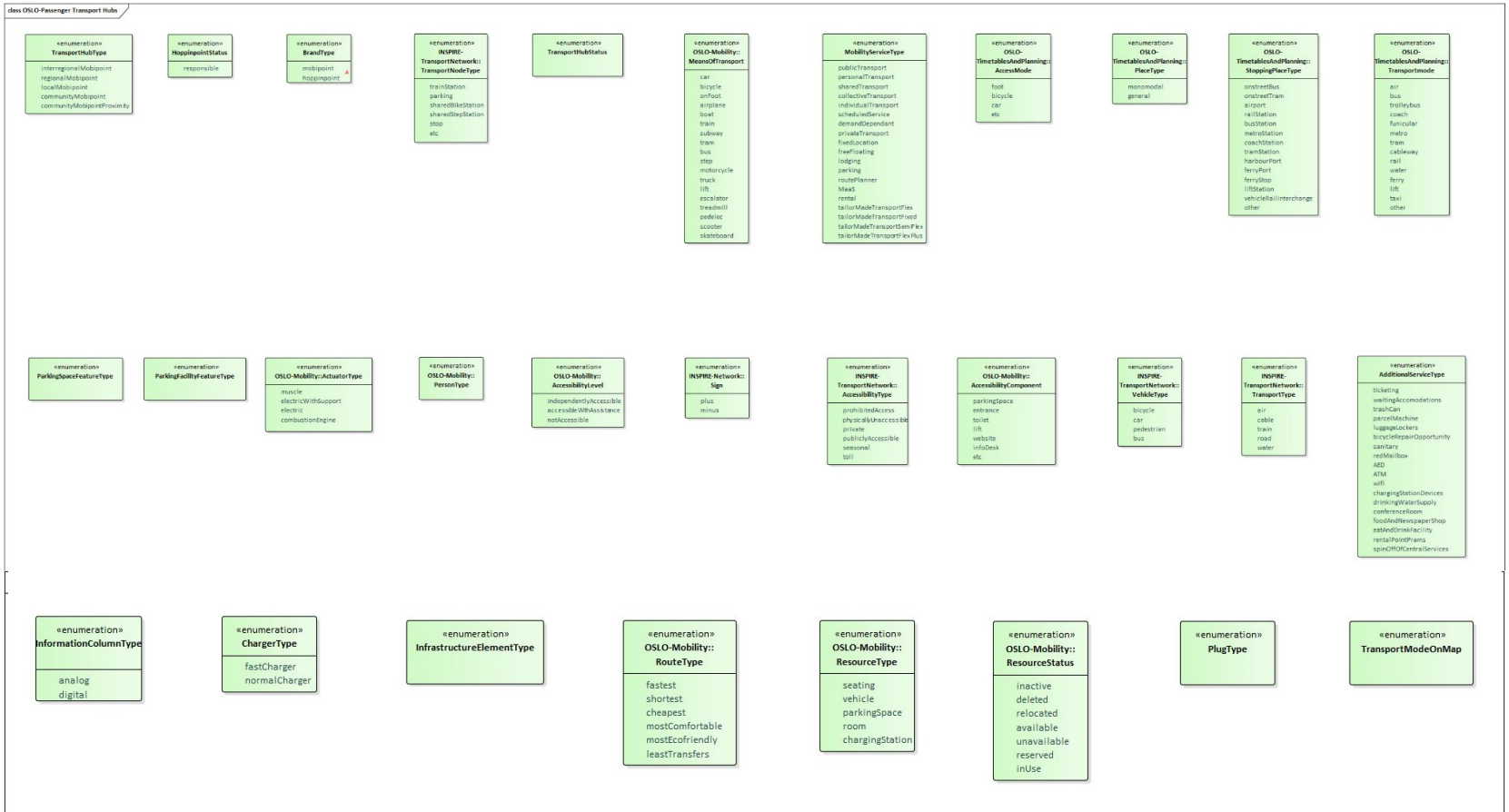


Data types


class OSLO-Passenger Transport Hubs
«dataType»
Capacity
+ maximum: Integer [0..1]
+ total: Integer [0..1]
+ vehicleType: VehicleType [0..1]
(from Mobivoc)
«dataType»
Accessibility
+ component: AccessibilityComponent [0..1]
+ level: AccessibilityLevel
+ type: AccessibilityType
(from OSLO-Mobility)
«dataType»
Identifier
(from OSLO-Generic)
«dataType»
Geometry
+ asGML: TypedString [0..1]
+ asWKT: TypedString [0..1]
(from OSLO-Generic)
«dataType»
Point
(from OSLO-Generic)
«dataType»
Polygon
(from OSLO-Generic)


class OSLO-Passenger Transport Hubs
«dataType»
GeographicalName
+ language: LanguageCode
+ spelling: String
(from OSLO-Generic)
«dataType»
MonetaryAmount
(from OSLO-Generic)
«dataType»
Period
+ endTime: DateTime [0..1]
+ startTime: DateTime
(from OSLO-Generic)
«dataType»
DirectedLink
+ direction: Sign
(from INSPIRE-Network)
«dataType»
Location
(from OSLO-Generic)
«dataType»
ContactInformation
+ address: AddressRepresentation [0..1]
+ contactName: String [0..1]
+ email: String [0..1]
+ faxNumber: String [0..1]
+ prefix: LanguageString [0..1]
+ telephone: String [0..1]
+ website: URI [0..1]
(from OSLO-Generic)
«dataType»
LanguageString
+ language: LanguageCode
+ string: String
(from OSLO-Generic)
«dataType»
OpeningHoursSpecification
(from Schema.org)
«dataType»
Timetable
(from OSLO-Mobility)
«dataType»
Availability
+ openingHours: TypedString [0..1]
+ period: OpeningHoursSpecification [0..1]
+ person: PersonType [0..1]
+ timetable: Timetable [0..1]
(from OSLO-Mobility)
«dataType»
RouteDescription
(from OSLO-Mobility)
«dataType»
AddressRepresentation
+ boxNumber: String [0..1]
+ country: LanguageString [0..1]
+ houseNumber: String [0..1]
+ municipalityName: LanguageString [0..1]
+ postCode: String [0..1]
+ streetName: LanguageString [0..1]
(from OSLO-Address)

Enumerations

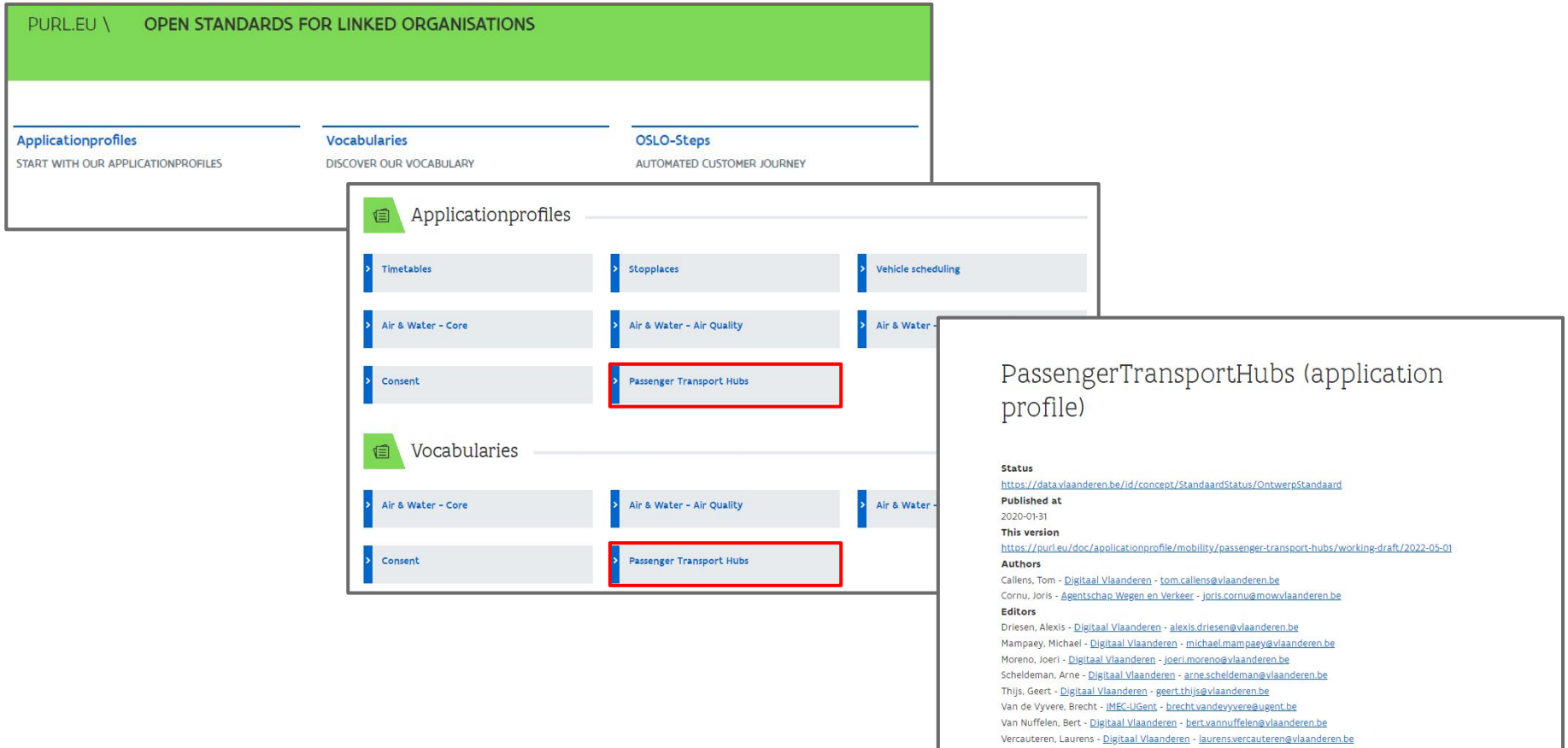


Publication PURL.eu

https://purl.eu/doc/applicationprofile/mobility/passenger-transport-hubs/

PURL.EU

https://purl.eu/



Application profile

https://purl.eu/doc/applicationprofile/mobility/passenger-transport-hubs/

Overview

This document describes the usage of the following entities for a correct usage of the Application Profile:

| AdditionalService | Agent | AutomatedParkingGarage | BicycleParking | BicycleParkingStation | BusShelter | Charger | CivicStructure | DataExchangeInfrastructure | Element | Executor | FormalFramework | HoppinColumn | Hoppinpoint | InformationCarrier | InformationColumn | InfrastructureElement | License | Location | MobilityService | Network | NetworkElement | Node | ParkingFacility | ParkingGarage | ParkingLot | ParkingSpace | PriceSpecification | Provider | PublicOrganisation | Quay | RegisteredPassengerTransportHub | Resource | ResourceReport | Route | RouteNode | RoutePlanner | RouteSegment | Service | Site | StopPlace | StopPost | StreetFurniture | Transfer | TransportConnection | TransportHub | TransportNetwork | TransportNode | TransportObject | TransportRegion | Traveler | Trip | UndergroundParking | VirtualNode |

This document describes the usage of the following datatypes for a correct usage of the Application Profile:

| Accessibility | AddressRepresentation | Availability | Capacity | ContactInformation | DirectedLink | GeographicalName | Geometry | Identifier | LanguageString | Location | MonetaryAmount | OpeningHours | Period | Point | Polygon | RouteDescription | Timetable |

TransportHub

Description

Place where passengers and/or freight are exchanged between vehicles and/or modes of transport.

Usage

Examples for passenger transport are train stations, underground stations, bus stations, airports, ferry terminals, car parks... For freight transport it concerns typically airports, seaports, transhipment terminals. Transport nodes are typically intermodal, e.g. travellers combine the (shared) bicycle with the train for their trip, but intramodal transfers (e.g. from one bus line to another) are also important. Concretely, a Transport interchange includes infrastructure (stations, car parks, charging infrastructure...), (mobility) services (public transport lines, shared bicycles, shops...) and nodes in transport networks (public transport stops, car sharing points...).

Properties

For this entity the following properties are defined: accessibility, brand, classification, infrastructureElement, location, name, roadAuthority, service, service, status, transportObject.

Property	Expected Range	Cardinality	Description	Usage	Codelist
accessibility	Accessibility	0..*	The accessibility of the Passenger Transport Hub.		
brand	BrandType	1	The brand of the Passenger Transport Hub.	Example: Hoppinpoint, Mobipoint,...	
classification	PassengerTransportHubType	0..*	The classification of the Passenger Transport Hub.		
infrastructureElement	InfrastructureElement	0..*	The infrastructure element of the Passenger Transport Hub.		

Vocabulary

https://purl.eu/ns/mobility/passenger-transport-hubs/

3. Overview

This section lists all classes and properties of the vocabularium.

Explore the vocabularium

Classes

DataExchangeInfrastructure | HoppinColumn | Hoppinpoint | InformationCarrier | RegisteredPassengerTransportHub | ResourceReport | TransportHub | TransportRegion | VirtualNode |

Properties

| accessibility | actuator | brand | classification | classification | location | location | meansOfTransport | number | registration | roadAuthority | service | specification | status | status | status | vehicletype |

4. Classes

This section gives a formal definition of every class.

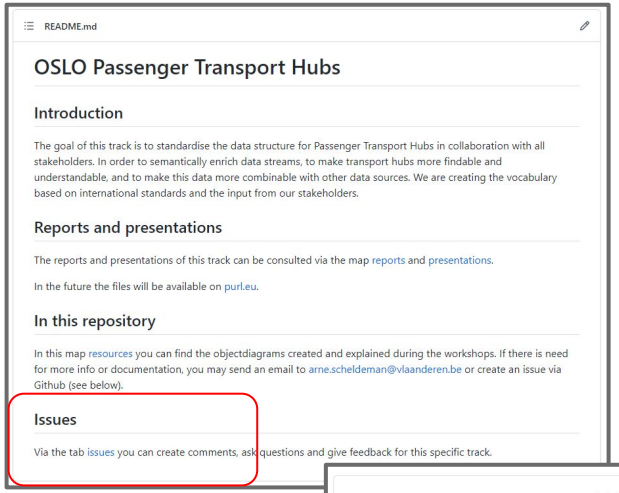
Class *DataExchangeInfrastructure*

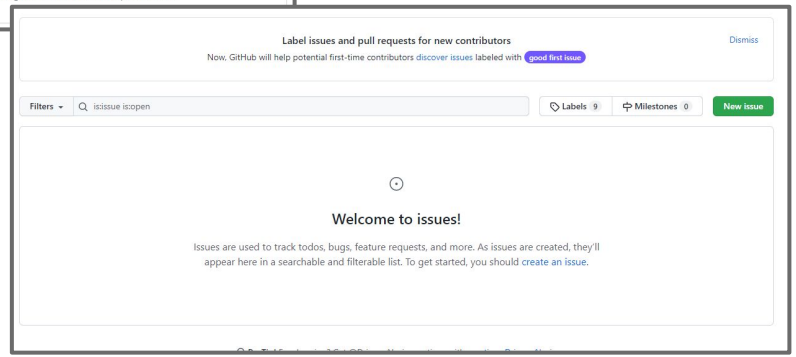
Type	Class
URI	https://purl.eu/ns/mobility/transporthubs#DataExchangeInfrastructure
Specialisation of	https://data.vlaanderen.be/ns/openbaardomein/infrastructuur#Infrastructuurelement
Definition	Physical facilities aimed at exchanging data.
Usage	E.g. servers, cables, modems.

Registering issues

How to register issues?

- Register via [GitHub page](#).





Q&A and Next Steps

Vlaanderen
verbeelding werkt

What do we do…?

Don’t we have to add …?

Can’t we better …?

What about …?

Didn’t we forget …?

Next steps

Process the input.

Distribution of a report from this workshop. Feedback is certainly welcome!

Publication on PURL.eu and start of public review.

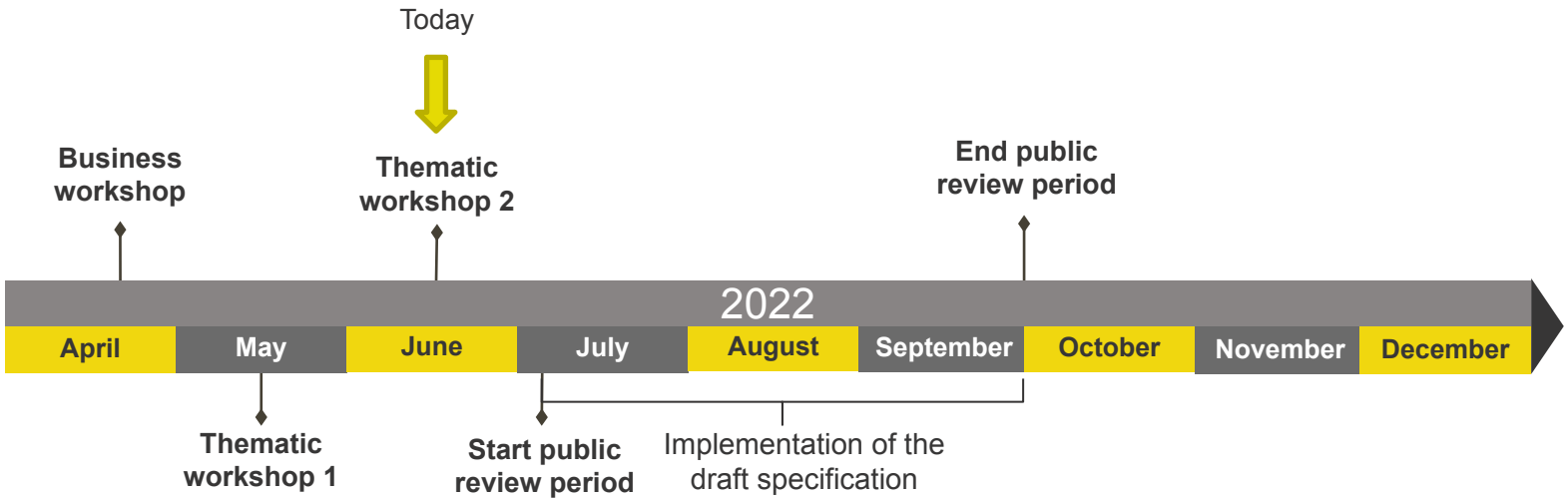
Issue gathering via PURL.eu.

Public review

- Model is stable
  - Possibility for PoC development
- Based on impact of feedback
  - Small semantic adjustments
  - Big semantic adjustments: possibility of additional working group

Planning

Public review within approximately two weeks.



Feedback & cooperation

Feedback can be provided by e-mail to the following persons:

- digitaal.vlaanderen@digitaal.vlaanderen.be
- laurens.vercauteren@vlaanderen.be
- arne.scheldeman@vlaanderen.be
- alexis.driesen@vlaanderen.be

Feedback/input can be provided via GitHub: https://github.com/Informatievlaanderen/OSLOthema-passengerTransportHubs

Through the creation of **issues**

Thank you!

Vlaanderen
verbeelding werkt

LDES data-example

- Assumption: we are going to synchronise our replica-database
  - E.g. by polling the EventStream
- Here: the Evenstream of OSLO PTH → Resource
- To ensure that it is up-to-date (in order that we always know the correct number of available bikes in a train station, e.g. Blue-bikes in this example).

Blue-bike

LDES data-example

- Let's assume that our replica of the Resources database looks like this at **9am**.
- Two bikes are available at that moment:
  - Bike BB-LS-001
  - Bike BB-LS-002
- They are in BicycleParkingStation BPS123 (in Leuven station)
- And provided by MobilityService MS456 (Blue-bike service)


```
{
    "@context": "",
    "@graph": [
        {
            "@id": "http://example.com/id/resource/BB-LS-001",
            "@type": "Resource",
            "type": "Bike",
            "location": "http://example.com/id/bicycle-parking-station/BPS123",
            "identifier": "BB-LS-001",
            "status": "available",
            "service": "http://example.com/id/mobility-service/MS456"
        },
        {
            "@id": "http://example.com/id/resource/BB-LS-002",
            "@type": "Resource",
            "type": "Bike",
            "location": "http://example.com/id/bicycle-parking-station/BPS123",
            "identifier": "BB-LS-002",
            "status": "available",
            "service": "http://example.com/id/mobility-service/MS456"
        }
    ]
}
```

LDES data-example: Evenstream, Node

- When polling the PTH:Resource Eventstream e.g. at **9:15am** we see a new version for bike BB-LS-001
  - (1) The status of the bike changed from “available” to “inUse” at **9:05am**.
- The Evenstream (https://example.com/ldes/mobility/resource) presents updates page by page.
  - (2) Here the update was found on page 5 (the pages are called Nodes in LDES)
- REMARK: For illustrative reasons the page presents only one update, there could have been a lot more.

LDES data-example: Version

- Version is modelled according to OSLO-Generics (based on the PROV-ontology).
  1. The update is of type VersionInValidtime (i.e. the Version of this Resource in real-time).
- It remains a Resource, while the update is a Version and a Resource at the same time. Thus, it has attributes of both.
  2. It has a version id, here implemented as a GUID.
  3. Being a Version, it inherited the attributes of this class, like a pointer to the original Object and the time the Version was created (which is **9:05am**).

LDES data-example: Processing

- After processing the update, we see in our replica that only bike BB-LS-002 is still available
- A count for BicycleParkingStation BPS123 (in Leuven station) would show that the number of available bikes is now 1


```
{
    "@context": "",
    "@graph": [
        {
            "@id": "http://example.com/id/resource/BB-LS-001",
            "@type": "Resource",
            "type": "Bike",
            "location": "http://example.com/id/bicycle-parking-station/BPS123",
            "identifier": "BB-LS-001",
            "status": "inUse",
            "service": "http://example.com/id/mobility-service/MS456"
        },
        {
            "@id": "http://example.com/id/resource/BB-LS-002",
            "@type": "Resource",
            "type": "Bike",
            "location": "http://example.com/id/bicycle-parking-station/BPS123",
            "identifier": "BB-LS-002",
            "status": "available",
            "service": "http://example.com/id/mobility-service/MS456"
        }
    ]
}
```