Open Transport Net

DELIVERABLE

Project Acronym:	**OTN**
Grant Agreement number:	**620533**
Project Full Title:	OpenTransportNet – Spatially Referenced Data Hubs for Innovation in the Transport Section

---

D4.3 DATA HARVESTER

Version: 1.0

Authors:
Pieter Colpaert (iMinds)

Reviewers:
Evangelos Argyzoudis (INTRASOFT)
Tomas Mildorf (UWB)
Karel Charvat (HSRS)
Andrew Stott (CORVE)

Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

This project has received funding from the European Union's Competitiveness and Innovation Framework Programme under grant agreement no. 620533.

Table of Contents

List of Tables

List of Figures

Revision History

doo	Date	Author	Organization	Description
0.1	10/01/2015	Pieter Colpaert	iMinds	First draft
0.2	20/01/2015	Pieter Colpaert	iMinds	Version for internal review
0.3	25/01/2015	Pieter Colpaert	iMinds	Internal review remarks integration
0.4	28/01/2015	Pieter Colpaert	iMinds	Final draft for consortium review
1.0	30/01/2015	Pieter Colpaert	iMinds	Final version

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Executive Summary

In Deliverable 4.2 “Data Collection and sharing plan”, we have introduced the general high level picture in Figure 1. Deliverable 4.2 focused on number 1 and 2 in this illustration, while this deliverable will be about the number 3: the data harvesting.

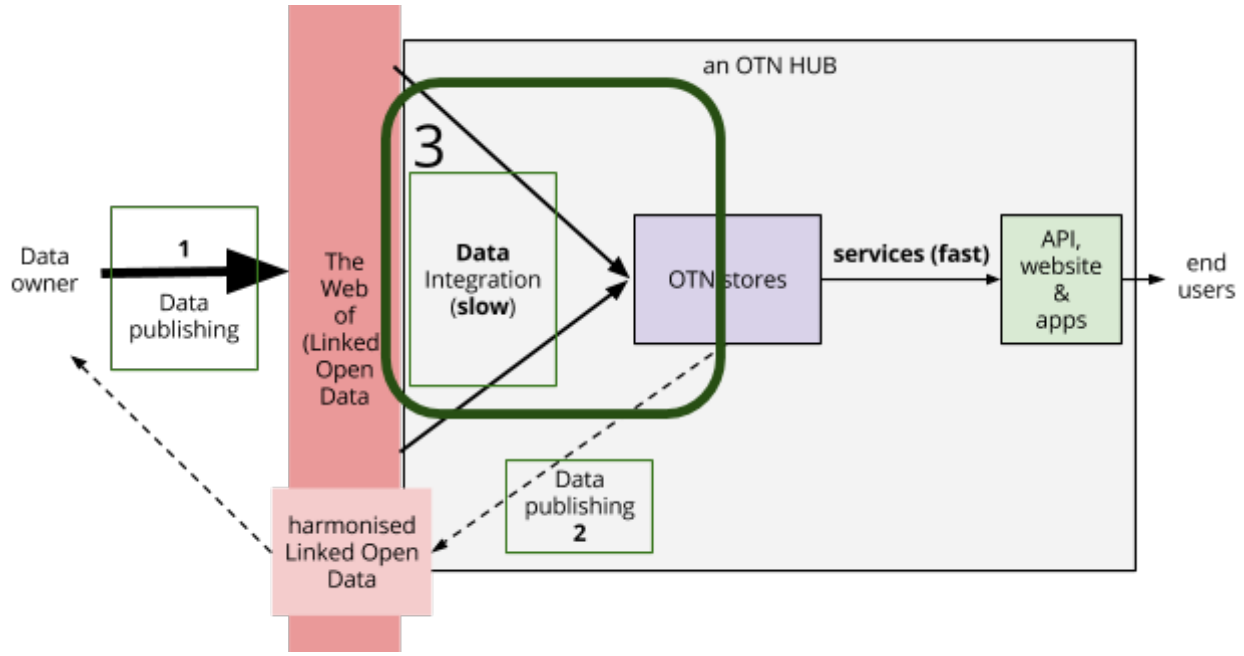

Figure 1: General overview of the data collection and sharing plan

The architecture for the data harvesting is illustrated in figure 2. *tdt/input* stands for an Extract-Transform-Load (ETL) extension of The DataTank which can be found at https://github.com/tdt/input.

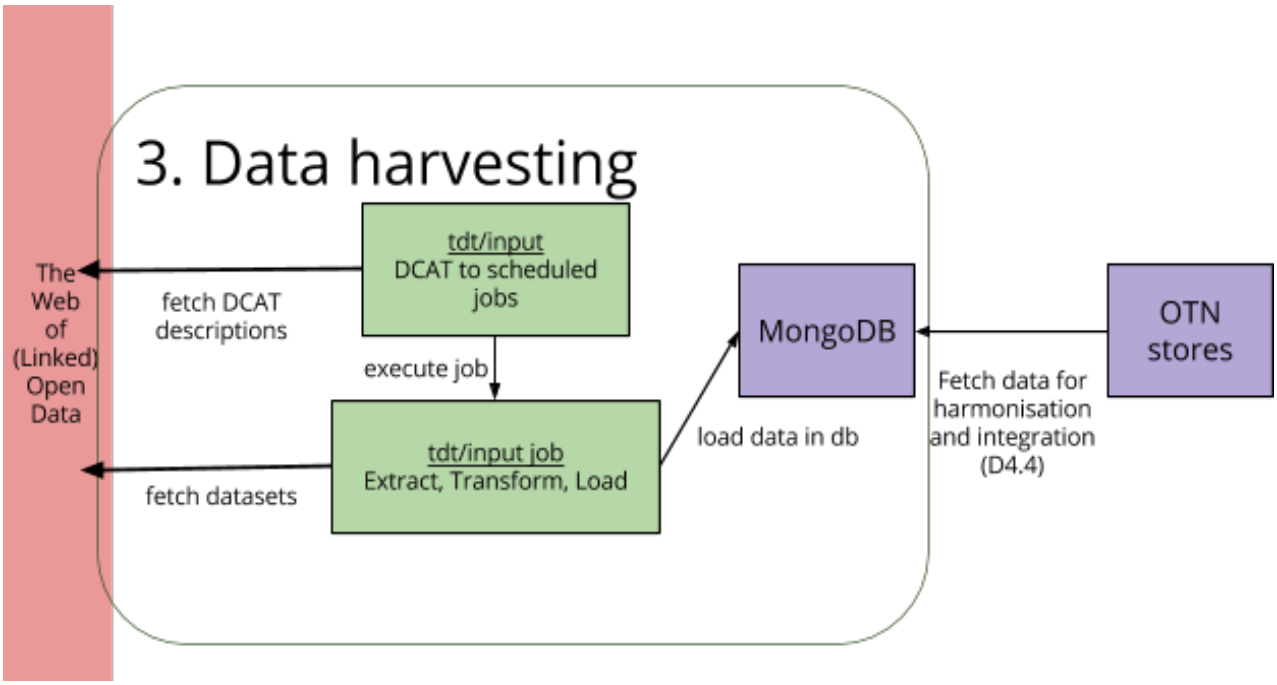

Figure 2: Overview of the data harvesting

In a first step, the DCAT feeds from deliverable 4.2 will be loaded in a new script written to configure tdt/input jobs using DCAT feeds. These DCAT feeds will be fetched each time the scheduler is consulted. When the scheduler notices that a certain job needs to be executed, the data will be refreshed by executing the job. Such a tdt/input jobs runs through 3 stages: (1) first it extract the data from its current format (e.g., it reads an XML file), (2) next, it transforms the data to an internal object, and (3) finally, this object is loaded in a database. In our case, we have chosen for MongoDB for scalability and flexible search capabilities. The metadata of the collections in MongoDB are stored in a new OTN-MP file which can be used by the other tasks.

1 Introduction

Data harvesting, in the context of this deliverable, is the act where a selection of data from various sources is extracted, stored, and made available for internal use through a previously agreed upon system. The benefit of this is that the data is afterwards usable from a query interface without having to download the dataset from the Web. Furthermore, when data changes a lot, different versions can be studied. In the latter case, the harvesting mechanism also works as an archive. In figure 1 we have illustrated the set-up of the project.

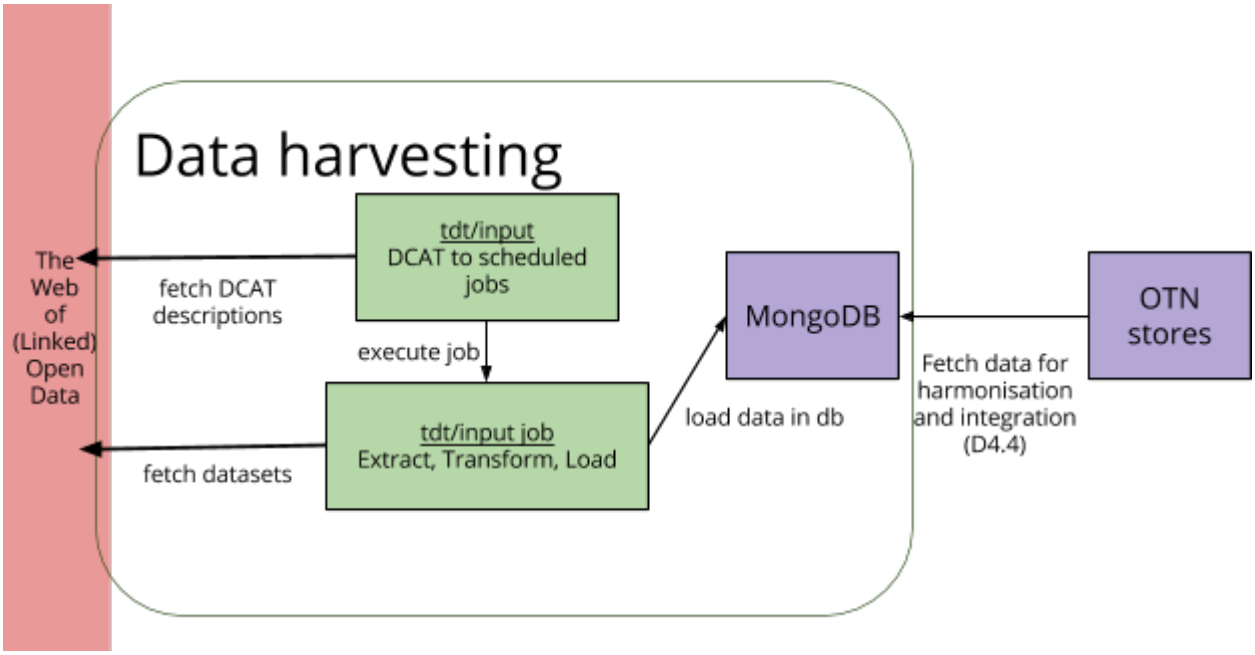

Figure 3: Overview of the data harvesting framework

First, we will describe how different metadata feeds are going to be integrated. Then, we're going to describe how these metadata feeds are going to be used to create harvesting jobs within tdt/input, a The DataTank extension to create Extract-Transform-Load (ETL) workflows.

2 Harvesting the metadata

The metadata from different data portals need to be collected and integrated before we can start harvesting the data itself. In Deliverable 4.2 we have introduced the Open Transport Net Metadata Profile. This profile is an application profile of DCAT and is compliant with the European standard DCAT-AP.

2.1 Themes

The categories under which they are categorised was left open. Yet, we will need one categorisation framework to be able to select the right datasets. For now, we have decided to use the taxonomy used by default within The DataTank: http://ns.thedatatank.com/dcat/themes#Taxonomy.

This taxonomy was created by looking at popular data portals in Belgium, by creating a list of all the themes currently used, and finally, by merging these in one document which can be viewed at https://docs.google.com/spreadsheets/d/1wIvRF4TyyEm0YGRL6jMo-tUDBhOSNmCSpEtQQJovhes/edit. As this list is created with the use case of aggregating Belgian data portals in mind, it may not be perfectly suitable for OTN, yet it is a good start. The themes with their URIs and a link to EuroVoc are listed in Table 1.

Name	new URI	URI eurvoc skos:closMatch	Eurovoc name
Governance and policy	http://ns.thedatatank.com/dcat/themes#Government	http://eurovoc.europa.eu/1172	Government
Mobility	http://ns.thedatatank.com/dcat/themes#Mobility	http://eurovoc.europa.eu/2015	Means of transport
Tourism	http://ns.thedatatank.com/dcat/themes#Tourism	http://eurovoc.europa.eu/4470	Tourism
Financing	http://ns.thedatatank.com/dcat/themes#Financing	http://eurovoc.europa.eu/1018	Public finance
Economy	http://ns.thedatatank.com/dcat/themes#Economy	http://eurovoc.europa.eu/637	Economy
Work	http://ns.thedatatank.com/dcat/themes#Work	http://eurovoc.europa.eu/4543	Work
Sports	http://ns.thedatatank.com/dcat/themes#Sports	http://eurovoc.europa.eu/4245	Sport
Cultural events	http://ns.thedatatank.com/dcat/themes#CulturalEvents	http://eurovoc.europa.eu/214801	Cultural event
Cultural heritage	http://ns.thedatatank.com/dcat/themes#CulturalHeritage		
Media	http://ns.thedatatank.com/dcat/themes#Media	http://eurovoc.europa.eu/216049	mass media
Environment	http://ns.thedatatank.com/dcat/themes#Environment	http://eurovoc.europa.eu/2825	Environment

Legislation	http://ns.thedatatank.com/dcat/themes#Law	http://eurovoc.europa.eu/578	Public law
Safety and justice	http://ns.thedatatank.com/dcat/themes#SafetyAndJustice		
Demography	http://ns.thedatatank.com/dcat/themes#Demography	http://eurovoc.europa.eu/385	Demography
Education	http://ns.thedatatank.com/dcat/themes#Education	http://eurovoc.europa.eu/668	Education
Research	http://ns.thedatatank.com/dcat/themes#Research	http://eurovoc.europa.eu/2914	Research
Living, health, wellbeing, poverty	http://ns.thedatatank.com/dcat/themes#LivingHealthWelfare	http://eurovoc.europa.eu/1004	welfare

Table 1: The themes used within the first iteration of the metadata aggregator

In future implementations, depending on the quality of the first harvested data, we will be able to make a more thorough categorisation within the transport field itself.

The URI dereferencing is done through tdt/triples (http://github.com/tdt/triples), an extension of The DataTank. When Open Transport Net wants to publish their own URIs, these new URIs can be made dereferencable as well.

2.2 DCAT-feed merger

The feed will need to be aggregated in one system, for the purpose of being able to select which datasets need to be harvested (e.g., only these in the domain of transport and mobility). We have written a small script which does that: if there is no theme that can be derived from the feed, the description will be parsed and we will do named entity recognition on dbpedia to find the closest matching theme. Furthermore, the project will try to add spatial attributes to datasets: if a certain dataset is harvested from the city of Antwerp, we will add a property binding this dataset to Antwerp.

The code of this project and instructions can be found at https://github.com/opentransportnet/dcat-merger. This Node.js application allows merging the DCAT information of several sources into one single Turtle (an RDF serialisation format) file. It can be used by editing the config.json and configure the right sources. Be sure to have Node.js installed and to have run “npm install” in the directory.

As a final result, we have a file which describes a data catalogue with datasets from various sources categorized by location and theme. This metadata can be queried for the right datasets that we want in our harvester.

3 The DataTank Input

The DataTank Input, or tdt/input (https://github.com/tdt/input), is a tool which facilitates the Extract-Transform-Load (ETL) lifecycle within a harvester.

First we will describe how to install tdt/input as an extension of tdt/core (https://github.com/tdt/core), then we will explain how to configure jobs. Finally, a small script should be written to be able to transform DCAT data into a job.

3.1 Installation

From the documentation at http://docs.thedatatank.com:

The installation is fairly simple and can be done by telling the core application that tdt/input is now a required package. This can be done by editing the require section of the composer.json file, located in your root application folder.
Note that the commands listed are assumed to be executed from the root folder of the datatank application.

```
"require": {
    ...,
    "tdt/input" : "dev-master"
}
```

After the composer.json changed, you need to let composer know that a new dependency has been added. The following command will download the input package:

```
$ composer update
```

The next thing you need to do is to create the datatables necessary for input to store its information. This can be done by executing the following command:

```
$ php artisan migrate --package=tdt/input
```

Now that everything is set and done, you'll have to let the core application know that it has a package it needs to approach. Unfortunately this has to be done manually in Laravel 4, by adding the service provider to the providers array entry in the app.php file located in the app/config folder.

A snippet is of this file is

```
'providers' => array(

    'Illuminate\Foundation\Providers\ArtisanServiceProvider',
    'Illuminate\Auth\AuthServiceProvider',
    'Illuminate\Cache\CacheServiceProvider',
    'Illuminate\Foundation\Providers\CommandCreatorServiceProvider',
    ...
    'Tdt\Input\InputServiceProvider',
)
```

3.2 Configuring and executing a job

How to manage jobs and execute them can be found at
http://docs.thedatatank.com/4.3/input_management

4 MongoDB

MongoDB is a document store: it stores JSON objects and allows one to query over them. In the next version of tdt/input, current proof of concept available at https://github.com/weopendata/packed-input, there will be a MongoDB loader.

On installation instructions, query instructions and management instructions, we refer to the official documentation at http://mongodb.org.

5 Conclusion

This deliverable focused on the next steps after Deliverable 4.2: Data sharing and collection plan. The metadata that is gathered, can be merged using a project we've introduced on the Open Transport Net github organization. Furthermore, the resulting catalogue can be used to generate harvesting tasks.

As after consultation with the consortium at Issy-les-Moulineau on the 27th of January, fetching data from an external source is not a key functionality required, the next efforts will first focus on getting the right metadata from datasets from the pilot cities which have predefined use cases and indicate a level of quality. When it appears necessary in later phases of the project, the mongodb harvesting architecture may be set up.