Het publiceren van datasets in het transportdomein voor maximaal hergebruik

Publishing Transport Data for Maximum Reuse

Pieter Colpaert

Promotoren: prof. dr. E. Mannens, dr. ir. R. Verborgh
Proefschrift ingediend tot het behalen van de graad van
Doctor in de industriële wetenschappen

UNIVERSITEIT
GENT

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. R. Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2017 - 2018

ISBN 978-94-6355-040-6
NUR 982, 993
Wettelijk depot: D/2017/10.500/75

Examination board

Prof. Oscar Corcho
Prof. Filip De Turck
Prof. Sidharta Gautama
Dr. ir. Philip Leroux
Dhr. Björn De Vidts

Chair

Prof. Patrick De Baets

Supervisors

Prof. Erik Mannens
Dr. ir. Ruben Verborgh

Preface

> "It is the most beautiful day of the year so far."
> — Jozef Colpaert.

When I started studying engineering, I thought data management was in a far more advanced state. I could not imagine that still, in 2007, manual intervention was needed to find the right data source and to integrate it into your own application. I could also not imagine that it would be illegal to – in your spare time – create a more mobile friendly webpage for accessing the time schedules of the Belgian railway company. Nonetheless, the iRail project still received a cease and desist letter, claiming a breach of *Intellectual Property Rights* (IPR). This sparked my interest in data availability in general, as much as it sparked my personal interest into IPR, as I was desparate to understand whether creating such a website was indeed illegal.

IRail did not stop the project. Instead, a non profit organization was set up in 2010 to foster creativity using mobility data, bringing together a community of enthusiasts – I was one of these – after the story hit the media. The organization released an *Application Programming Interface* (API) which would allow third parties to integrate railway data within their own services. The API is still online as an open-source project, accepting contributions from other transport data enthusiasts. Thanks to the iRail project, I was able to access most interesting research data in primetime. The query logs of this API for instance, would prove themselves priceless.

In 2011, I wrote my master's thesis on extending the Open Data publishing framework *The DataTank* – which I started earlier that year – with a queryable interface over HTTP. The goal of the thesis was to offer a better experience to developers that wanted to use governmental datasets in their own software. While we did

design a query language for small in-memory documents, we did not take into account the scalability of these server interfaces, neither did we study the effects on the information system as a whole. We only tested the overal query execution time, which would show an increasing response time when an increasing amount of datasets would be combined. The querying interface on top of The DataTank later disappeared again from the stable release, yet the ambition to make Open Data more used and useful remained.

Challenged by officials whom I wanted to prove there was indeed commercial value in Open Data, I co-founded the start-up FlatTurtle with Yeri Tiete, the founder of iRail, and Christophe Petitjean, the excentric business owner of rentalvalue that came up with the idea to sell information displays to professional real estate owners. These information displays would show the latest information about for example public transport, weather, news, or internal affairs. With FlatTurtle, we were unable to reuse datasets by relying on basic building blocks of the Web: at the time, for example, the servers were not using proper cache headers, legal conditions were not clear, and identifiers would conflict and change across data updates.

The company did not make me financially rich, yet what I have been able to learn in terms of running a business in this period was invaluable. Furthermore, it taught me how – while there are a lot of people – there is still a limit to the amount of people you can meet in one day. While trying to change the world for the better, whether it is with a product you sell, with a research proposal, or with a general idea – such as the one of Open Data –, your impact will be as big as the quality of your pitch to explain the solution. With this in mind during my time pursuing a PhD, I tried my best when I gave one of the many invited talks explaining Linked Open Transport Data and its importance. Having to explain this subject over and over again influenced the first chapters of this book heavily.

At the same time, the iRail non-profit merged together with other initiatives such as Open Street Map, Creative Commons and Open Access, into Open Knowledge Belgium. Still today I am part of the board of directors of that non profit organization, trying to create a world where

knowledge creates power for the many, not the few.

When starting my PhD in November 2012, I also thought the field of Web Engineering would be in a more advanced state. In the field of Web APIs for example, new vague paradigms are still today popping up without clear comparisons between their advantages and disadvantages. In this PhD, a modest contribution is done to measure data publishing interfaces for the purpose of public transit route planning. Instead of only measuring response times, I measured the impact of this interface on the information system as a whole, measuring cost-efficiency of a server interface, cacheability for a certain mix of queries, and described non-measurable benefits such as flexibility for developers or privacy by design.

Overall over the last four years, I have been happy. Being able to come home in the evening with a feeling that you are contributing to a better world is how I would describe my dream job. When confronted with these kind of life questions late at night in a bar with friends, I would – like a geek does – with great pleasure and in great length explain the Kardashev scale. This scale in 3 levels is a method of measuring a civilization's level of technological advancement. Today, humanity is at level zero, not being able to survive a natural disaster and not being able to be independent for its energy source from the host planet. Crucial to becoming a type 1 civilization – and it is unsure whether humanity will become a type 1 civilization – is to have an information system in which each individual can contribute to the civilization's knowledge and use it to make informed decisions. Such information systems will more than ever in the next years play a crucial role in – just to name a few – education, science, decision making, and politics.

Belgium in particular has been an interesting country to do research into governmental organizations. It is a dense country, where for research into governmental organizations, as in a small geographic area, different governmental levels, from local to European, and companies of different scale can be studied. This also makes using a decentralized approach to data governance a necessity: the Belgian federal government's datasets have to be interoperable with the datasets from

departments and agencies of the Flemish government, as well as with the databases of all local governments. In our work with these organizations, we had an interdisciplinary team in which I worked together with people from among others MICT and SMIT. While it is impossible to mention everyone, I owe a big thank you to two people with whom I without doubt have collaborated the most so far. Nils and Mathias, I have had a blast studying the road to Open Data together. I look forward to, for the next few years, to make the region of Flanders a leading example in real-time Open Data publishing, and to grow our interdisciplinary team of Open Data researchers as there is still a lot of work left undone.

Ruben, it is a true honor to be able to be part of your team. You not only set the bar high for yourself and the team, you also know how to guide the team towards success and impact. I look forward to continue working on Knowledge on Web-Scale under your supervision as a postdoctoral researcher. Erik, thanks for always protecting my back.

To my – old and new, close and distant – colleagues, project partners, and people encountered in local, regional, federal, and European government organizations: thank you for being an infinite source of inspiration. You will undoubtedly recognize parts of this dissertation that were the result of a discussion we may have had or problem you confronted me with.

Mom and dad, you often refer to me as the optimist (I optimistically call it realism). I am happy to be able to live with this trait. As the quote used to introduce this preface goes, I am sure this is not merely caused by genetics, but that this was also caused by the way how I was raised. From my perspective – that is all I can speak for – all went well in this process: thank you!

Finally, Annelies, thank you for reminding me from time to time there is more to life than Open Data.

Pieter

Table of Contents

4. Qualitatively measuring different parameters
    1. Legal interoperability
    2. Technical interoperability
    3. Syntactic interoperability
    4. Semantic interoperability
    5. Querying interoperability
5. Conclusion

CHAPTER 4 — Raising Interoperability of Governmental Datasets

1. Data portals: making datasets discoverable
    1. The DataTank
    2. 5 stars of Open Data Portals
    3. Open Data Portal Europe and the interoperability of transport data
    4. Queryable metadata
2. Open Data in Flanders
    1. A tumultuous background
    2. Discussing datasets at the Department of Transport and Public Works
    3. Challenges and workshops
    4. Recommendations for action
3. Local Decisions as Linked Open Data
    1. Implementation and demonstration
4. Conclusion

CHAPTER 5 — Transport Data

1. Data on the road
    1. Constraints for a large-scale ITS data-sharing system
    2. A use case in Ghent
    3. Access to road networks
2. Public Transit Time Schedoles
    1. Route planning queries
    2. Other queries
    3. Route planning algorithms
    4. Exchanging route planning data over the Web
3. Conclusion

CHAPTER 6 — Public Transit Route Planning Over Lightweight Linked Data Interfaces

1. The Linked Connections framework
2. Evaluation design
3. Results of the overall cost-efficiency test
4. Linked Connections with wheelchair accessibility
    1. Linked Connections with filtering on the client
    2. Linked Connections with filtering on the server and client
5. Wheelchair accessibility feature experiment
    1. Evaluation design
    2. Results
6. Discussion
    1. Network latency
    2. Actual query response times
    3. More advanced federated route planning
    4. Denser public transport networks
    5. Real-time updates: accounting for delays
    6. Beyond the EAT query
    7. On disk space consumption
    8. Non-measurable benefits
7. Conclusion and discussion

CHAPTER 7 — Conclusion

Glossary

API	Application Programming Interface
BGP	Basic Graph Patterns
CORS	Cross Origin Resource Sharing
CPU	Central Processing Unit
CSA	Connection Scan Algorithm
CSV	Comma-Separated Values
DCAT	Data CATalogue vocabulary
DTPW	Flemish Department of Transport and Public Works
EAT	Earliest Arrival Time
GTFS	General Transit Feed Specification
HATEOAS	Hypermedia As The Engine Of Application State
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IIOP	Identifier Interoperability
IMI	Information Modeling and Interoperability
INSPIRE	Infrastructure for Spatial Information in the European Community
IPR	Intellectual Property Rights
ITS	Intelligent Transport Systems
JSON	Javascript Object Notation
LC	Linked Connections

LDF	Linked Data Fragments
MEAT	Minimum Expected Arrival Time
PSI	Public Sector Information
RDF	Resource Description Framework
REST	Representational State Transfer
RSD	Road Sign Database
SOA	Service Oriented Architecture
SIRI	The Service Interface for Real Time Information
TPF	Triple Pattern Fragments
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Samenvatting

De manier waarop reizigers hun routes willen plannen is divers. Enkele voorbeelden: routes berekenen rekeninghoudend met een fysische beperking; het combineren van verschillende transportmiddelen; het in rekening brengen of een eindgebruiker een (plooi-)fiets, wagen of bepaalde abonnementen bezit; of zelfs het berekenen van routes op basis van de mooiste foto's op socialmediakanalen. Eerder dan louter een wiskundig probleem, is routeplanning vandaag een probleem dat afhangt van de beschikbaarheid van gegevens: een beter routeadvies kan gegeven worden als er nog meer datasets worden verwerkt tijdens de query-evaluatie.

Overheidsadministraties beheren datasets die kunnen bijdragen tot zo'n routeplanningsadvies. Vandaag zijn er duidelijke aanwijzingen dat die administraties hun data al beginnen publiceren op opendataportalen. Toch is vandaag de kost om die datasets te hergebruiken in andere systemen te hoog. Dat kunnen we zeggen omdat we simpelweg nog geen bewijs hebben gevonden dat veel datasets worden gebruikt. Hoe kunnen we opendatastrategieën verbeteren en ervoor zorgen dat deze datasets meer gebruikt en nuttiger worden?

Het publiceren van data voor maximaal hergebruik betekent het zoeken naar een lagere kost om die data te integreren binnen systemen van derde partijen. Dit is een automatisatie-probleem: in het ideale geval werkt software geschreven om te werken met een dataset, direct ook met een dataset gepubliceerd door een andere autoriteit. We kunnen die adoptiekost verlagen en het hergebruik automatiseren als we de interoperabiliteit tussen al deze datasets op het Web verhogen. Daarom is de focus van dit doctoraat het bestuderen van databroninteroperabiliteit – elk met hun heterogeniteitsproblemen – op 5 veschillende

lagen. De (I) juridische laag beschrijft de vraag of we volgens de wet en de licenties, de twee datasets mogen samenvoegen. Op de (II) technische laag bestuderen we dan weer de moeilijkheden die gepaard gaan met twee datasets fysisch samen te brengen. De (III) syntactische laag beschrijft of dat het formaat waar de dataset in geserialiseerd is, kan worden gecombineerd met andere bronnen. Verder kan de syntax ook bouwblokken aanbieden om op een gestandardiseerde manier elementen te identifieren of in te delen in een domeinmodel. Dit creëert de basis om te komen tot een hogere (IV) semantische interoperabiliteit, gezien voor dezelfde objecten in de echte wereld de identificatoren gealigneerd kunnen worden.

Het doel van dit doctoraat is het bestuderen hoe de databroninteroperabiliteit verhoogd kan worden, om de hergebruikskost in routeplanningssoftware te verlagen.

Eenmaal twee datasets interoperabel zijn over deze vier lagen, is het niet automatisch zo dat er makkelijk vragen kunnen gesteld worden over beide bronnen. Gezien we databronnen bestuderen gepubliceerd door meerdere autoriteiten, voegen we ook de (V) querying-laag toe. Twee extremen bestaan vandaag om data te publiceren: of de queries worden volledig uitgevoerd op de server van de data-publisher, of enkel een datadump wordt aangeboden, en de queries worden dus volledig uitgevoerd op de infrastructuur van de hergebruiker. De *Linked Data Fragments* (LDF)-as beschrijft een onderzoekskader om een evenwicht te vinden tussen functionaliteiten te voorzien door een data-publicerende server, of functionaliteiten die door de hergebruiker moeten worden geïmplementeerd. In plaats van één datadump aan te bieden, worden gelinkte fragmenten voorgesteld. Hierdoor kunnen programma's via metadata in ieder fragment, meer fragmenten ontdekken.

Op ieder niveau kunnen we vandaag al generieke oplossingen voorstellen om het potentieel hergebruik van data te maximaliseren. De juridische aspecten bijvoorbeeld worden afgehandeld door de *Open Definition*. Deze definitie eist dat een document wordt gemetadateerd met een publieke licentie. De enige

voorwaarden die mogen worden opgenomen, zijn ten eerste dat enerzijds er een verplichting mag zijn om naamsvermelding te eisen, en ten tweede dat er mag geëist worden dat een afgeleid document moet worden gepubliceerd onder dezelfde licentie. De juridische interoperabiliteit stijgt nog als de hergebruiksvoorwaarden zelf ook machineleesbaar zijn, en op hun eigen manier ook worden gepubliceerd voor maximimaal hergebruik.

Het Web is ons wereld-wijd informatiesysteem. Om technische interoperabiliteit te verzekeren, is de *uniforme interface* – een van de REST architecturale beperkingen – die we aannemen HTTP. Maar ook voor de identificatoren verkiezen we web-adressen of HTTP URIS te gebruiken. Op deze manier kunnen dezelfde identificatoren gebruikt worden voor toegang tot verschillende representaties voor hetzelfde object. De identificaties worden een globaal unieke tekenreeks, waardoor identificatieconflicten worden vermeden. Bovendien kunnen verschillende serialisaties met behulp van de *Resource Description Framework* (RDF) elk gegevenselement met deze HTTP URIS annoteren, waardoor ieder element in deze databron automatisch gedocumenteerd wordt.

In 2015 kreeg ik de kans om samen met communicatiewetenschappers de organisatorische uitdagingen te bestuderen bij het Departement voor Mobiliteit en Openbare Werken van de Vlaamse Overheid. Dankzij drie Europese directives (PSI, INSPIRE en ITS) en hun eigen inzicht, konden we al een strategie ontdekken tot het maximaliseren van hergebruik van hun gegevens. Hoe zo een opendatastrategie naar het volgende niveau te tillen, was echter nog onduidelijk. Door 27 data-eigenaars en directeurs te interviewen, kwamen we tot een lijst van voorstellen tot acties overheen alle interoperabiliteitsniveaus.

Geen enkele van de veelgebruikte specificaties vandaag – zoals GTFS of DATEX2 – binnen de transport-wereld, documenteren hun datamodellen met behulp van URI's. Om deze toch bruikbaar te maken binnen de RDF-wereld, bouwde ik zelf mappings en publiceerde deze.

Om routes te plannen over verschillende bronnen bestudeerden we bestaande routeplanningsalgoritmes. Het te selecteren basisalgoritme moet ook een efficiënte

fragmentatie-strategie toelaten. Onze hypothese was dat op deze manier een nieuwe afweging gemaakt kon worden, door een kostenefficiënte interface voorop te stellen – overheids-servers zouden immers niet moeten instaan voor de berekening van een antwoord op eender welke vraag – alsook het toelaten van voldoende flexibiliteit bij hergebruikers. Ik besloot om het Connection Scan Algoritme hiervoor te gebruiken.

In dit boek introduceren we het *Linked Connections* (LC) raamwerk. Een LC-server publiceert een gesorteerde lijst van connecties – koppels van een vertrek en een aankomst – in fragmenten, aan elkaar gelinkt via volgende en vorige pagina links. Het Connection Scan Algoritme kan dan worden geïmplementeerd op de infrastructuur van de data-hergebruiker.

Het nadeel van dit publicatiemechanisme is uiteraard een hoger verbruik van bandbreedte. Dit leidt tot hogere querytijden als de hergebruiker voor de eerste keer zo'n vraag moet beantwoorden, zeker op een traag netwerk. De hergebruiker kan hier echter evengoed een tussenliggende server zijn, dat naar een eindgebruiker toestel een beknopt antwoord stuurt. Als we bestudeerden wat de impact was van een extra functionaliteit op de server, dan merkten we dat bij bijvoorbeeld een rolstoeltoegankelijkheid-filter op de server, zowel de server als het toestel van de eindgebruiker meer werk hadden om de data te verwerken.

We kunnen besluiten dat we met LC een raamwerk hebben gecreëerd waarbij we optimaal omspringen met de vijf databroninteroperabiliteitslagen. Een nieuwe afweging werd gemaakt tussen werk te doen door de server en werk te doen door de hergebruiker, waarbij hergebruik over het informatiesysteem gemaximaliseerd kan worden.

In het algemeen kan ik op basis van dit doctoraat, om een beter opendatabeleid te implementeren, enkele tips meegeven om huidige HTTP-interfaces te verbeteren. (I) Fragmenteer uw datasets en publiceer documenten over het beveiligde HTTPS protocol. De manier waarop fragmenten gekozen moeten worden hangt af van geval tot geval. (II) Wanneer uw publicatie-mechanisme snellere antwoorden moet toelaten, kunnen op de server-side ook

meer fragmenten worden aangeboden, meer filters, geaggregeerde datasets (relevant voor tijdsreeksen), enzovoort. Om de server-schaalbaarheid te optimaliseren, is het belangrijk om (III) de juiste cache-headers te implementeren. Om de vindbaarheid van de data te optimaliseren, raad ik aan om (IV) hypermedia beschrijvingen toe te voegen. (V) Web-adressen of HTTP URI's zijn er dan weer om identificatoren voor zowel dingen als het domeinmodel te documenteren. Voor de juridische interoperabiliteit moet er (VI) een link naar een machineleesbare open licentie toegevoegd worden in het document. (VII) Voeg ook een Cross Origin Resource Sharing HTTP header toe, zodanig dat ook op andere domeinen webtoepassingen uw data kunnen hergebruiken. Ten laatste kan er ook nog (VIII) DCAT-AP metadata voorzien worden, zodat de dataset ook kan worden beschreven in opendataportalen.

Deze aanpak werkt niet enkel voor statische gegevens: documenten op het web kunnen immers veranderen. Het HTTP protocol laat toe om voor een aantal seconden een informatie resource te cachen, of ook te cachen op basis van een zogenoemde ETag. Zelfs als een document slechts enkele seconden wordt gecachet, wint een server aan schaalbaarheid, en kan ook een maximum belasting berekend worden op een backend systeem.

Ondanks de lange tijd dat het Web al meegaat – of toch in termen van digitale ontwikkelingen – zijn er nog steeds organisationele uitdagingen om een wereldwijd informatiesysteem te bouwen voor *iedereen*. Ik hoop dat ik met dit doctoraat een naslagwerk heb geschreven dat kan dienen als input voor standaarden, en als een inspiratie voor mensen die zelf hun eigen opendatasysteem op web-schaal willen bouwen.

Summary

The way travelers want route planning advice is diverse. To name a few: finding journeys that are accessible with a certain disability; combining different modes of transport; taking into account whether the traveler owns a (foldable) bike, car or public transit subscription; or even calculating journeys with the nicest pictures on social network sites. Rather than merely being a mathematical problem, route planning advice became a data accessibility problem. Better route planning advice can only be given when more datasets can be used within the query evaluation.

Public administrations maintain datasets that may contribute to such route planning advice. Today, there is evidence of such datasets being published on Open Data Portals, yet still the cost for adopting these datasets in end-user systems is too high, as there is no evidence yet of wide reuse of these simple datasets. In order to make these datasets more used and useful, how can we leverage Open Data publishing policies?

Publishing data for maximum reuse means persuing a lower cost for adoption of your dataset. This is an automation challenge: ideally, software written to work with one dataset, works as well with datasets published by a different authority. We can lower the cost for adoption of datasets and automating data reuse, when we raise the interoperability between all datasets published on the Web. Therefore, in this PhD we study the interoperability of data sources – each with their hetereneity problems – and introduce *5 data source interoperability* levels. The (I) *legal level* puts forward the question whether we are legally allowed to bring two datasets together. On the (II) *technical level*, we can study whether there are technical difficulties to physically bring the datasets together. The (III) *syntactic*

interoperability describes whether the serializations can be brought together. Moreover, the syntax should provide building blocks to document identifiers used in the dataset, as well as the domain model used. This creates the basis for reaching a higher (IV) *semantic interoperability*, as for the same real-world objects, identifiers can be aligned.

The goal of this PhD is to study how to raise the data source interoperability of public datasets, in order to lower the cost for adoption in route planning services.

As we study data sources published by multiple authorities, and as we still need to be able to evaluate queries over these datasets, we also added the (V) *querying level*. When the other four layers are fulfilled, we can otherwise still not garantuee a cost-efficient way to evaluate queries. Today, two extremes exist to publish datasets on the Web: or the query evaluation happens entirely on the data publisher's interface, or only a data dump is provided, and the query evaluation happens entirely on the infrastructure of a reuser after replicating the entire dataset. The *Linked Data Fragments* (LDF) axis introduces a framework to study the effort done by clients vs. the effort done by servers, and tries to find new trade-offs by fragmenting datasets in a finite number of documents. By following hypermedia controls within these documents, user agents can discover fragments as they go along.

To each of these layers, we can map generic solutions for maximizing the potential reuse of a dataset. As we are working towards Open Data, the legal aspect is covered by the *Open Definition*. This definition requires the data to be accompanied by a public license that informs end-users about the restrictions that apply when reusing these datasets. The only restrictions that may apply are the legal obligations to always mention the source of the original document containing the data, and the legal restriction that when changing this document, the resulting document needs to be published with the same license conditions. The legal interoperability raises when these reuse conditions itself are also machine interpretable, and are in the same way to be published for maximum reuse

as well.

We are using the Web as our worldwide information system. In order to ensure technical interoperability, the *uniform interface* – one of the REST architectural constraints – that we adopt is HTTP. Yet, also for the identifiers, we choose to use HTTP identifiers or URIs. This way, the same identifiers can be used for accessing different serializations and representations for the same object. This also means the identifiers become a globally unique string of characters, and thus avoids identifier conflicts. Furthermore, using the *Resource Description Framework* (RDF), different serializations can annotate each data element with these HTTP URIs as well, which enables identifier reuse and linking across independent data sources.

In 2015, I had the opportunity to study the organizational challenges, together with communication scientists, at the *Flemish Department of Transport and Public Works* (DTPW) of the Flemish government. Three European directives (PSI, INSPIRE, and ITS) extended with own insights, created a clear willingness to publish data for maximum reuse. How to implement such an Open Data strategy in a large organization was still unclear. As we interviewed 27 data owners and directors, we came to a list of recommendations for next steps on all interoperability levels.

None of the common specifications – such as GTFS, TRANSMODEL, and DATEX2 – for describing time schedules, road networks, disruptions, and road events have an authoritative Linked Data approach. For the specific case of public transit time schedules, we used the GTFS specification and mapped the terms within the domain model to URIs for these to become usable in RDF datasets.

For route planning over various sources, we studied the current existing public transit route planning algorithms. The to be selected base algorithm on which other route planning algorithms can be based, needs to work on top of a data model that allows for an efficient fragmentation strategy. Our hypothesis was that this way a new trade-off could be established, putting forward a cost-efficient way of publishing – as governmental organizations cannot afford to evaluate all queries over all

datasets on their servers – as well as leave room for client-side flexibility. For this purpose, we found the *Connection Scan Algorithm* (CSA) to be a good fit.

We introduced the *Linked Connections* (LC) framework. An LC server publishes an ordered list of connections – departure and arrival pairs – in fragments interlinked with next and previous page links. The CSA algorithm can then be implemented on the side of the data consumer. Enabling the client to do the query execution comes with benefits: (I) off-loading server, (II) better user-perceived performance, (III) more datasets can be taken into account, and (IV) privacy by design, as the query itself is never sent to a server.

The drawback of this publishing method is a higher bandwidth consumption, and when the client did not cache any resources yet, the querying – certainly when using a slow network – is slow. However, the clients do not necessarily need to be the end-user devices, also intermediary servers can evaluate queries over the web, and give concise and timely answers to a smaller set of end-users. When studying whether an LC server should now also expose the functionality of wheelchair accessibility, we found that both client and server had more work to process the data.

With LC, we designed a framework with a high potential interoperability of all five levels on the Web. We researched a new trade-off for publishing public transport data by evaluating the cost-efficiency. The trade-off chosen allows for flexibility on the client-side, while offering a cost-efficient interface to data publishers.

In order to achieve a better Web ecosystem for sharing data, we propose a set of minimum extra requirements when using the HTTP protocol. (I) Fragment your datasets and publish the documents over HTTPS. The way the fragments are chosen depends on the domain model. (II) When you want to enable faster query answering, provide aggregated documents with appropriate links (useful for – for example – time series), or expose more fragments on the server-side. For scalability, (III) add caching headers to each document. For discoverability, (IV) add hypermedia descriptions in the document. (V) A web address (URI) per object you describe, as well as HTTP URIS for the domain

model. This way, each data element is documented and there will be a framework to raise the semantic interoperability. For the legal interoperability, (VI) add a link to a machine readable open license in the document. (VII) Add a Cross Origin Resource Sharing HTTP header, enabling access from pages hosted on different origins. Finally, (VIII) provide DCAT-AP metadata for discoverability in Open Data Portals.

This approach does not limit itself to static data. The HTTP protocol allows for caching resources for smaller amounts of time. Even when a document may change every couple of seconds, the resource can still be cached during that period of time, and a maximum load on a back-end system can be calculated.

Despite the old age of the Web – at least in terms of digital technology advances – there are still organizational challenges to overcome to build a global information system for the many. I hope this PhD can be the input for standardization activities within the (public) transport domain, and an inspiration to publishing on Web-scale for others.

Chapter 1
Introduction

> " If you want to go fast, go alone. If you want to go far, go together. "
> — African proverb.

How far do you live from work? Keep the answer to this question in mind. Is the unit of measurement you used to answer this question minutes or kilometers? When asking a certain audience this question, each time, a significant amount of people answered with a distance in kilometers, while others would answer with a distance in time. Now imagine a software program has to calculate the time distance from one point to another for an end-user. Just imagine the amount of datasets that could be used to come up with a good response to that question... For 4 years I have been working in projects that had one goal in common: sharing data for an unknown number of use cases and an unknown number of users.

In this chapter we first discuss the research question. Then we discuss in more detail the projects that contributed to this research and the structure of the rest of this book.

1. RESEARCH QUESTION

I will define *open* in the next chapter.

Transport data is used as a focus, yet there is no clear distinction between transport data and other kinds of data. As an illustration, even datasets like criminality rates could be at some point used to provide a better route planning experience. Only in Chapter 5 and 6 we will dive into the specifics of the transport domain.

I studied *lightweight* interfaces for sharing *open transport datasets*. The term Open Transport Data, on the one hand, entails the goal of maximizing the reuse of your transport datasets. For example, in order to inform commuters better, a public transport agency wants to make sure the last updates about their transit schedules are available in each possible end-user interface. Another example of a clear incentive for governmental organizations in specific to publish data for maximum reuse would be policy decisions: datasets maintained by public administrations should be published "once-only" and become as used and useful as possible, as part of their core task. Publishing data for maximum reuse is an automation challenge: ideally, software written to work with one dataset, works as well with datasets published by a different authority. We can lower the cost for adoption of datasets and automating data reuse, when we raise the interoperability between all datasets published on the

Web.

Lightweight interfaces on the other hand, entails that when publishing the data for maximum reuse – and thus when this data becomes widely adopted – there is not an ever growing publishing cost that comes with this server interface. We observe today that there are two common ways to share transport data. The first way is to provide an export of all facts in one dump, which can be used by reusers to ask any question slowly. The second way is to provide a data service, which can be used by reusers to answer a set of specific questions quickly. The goal of the Linked Connections framework introduced in Chapter 6, is to experiment with the trade-offs between the efforts needed to be done by reusers, questions that can be answered quickly by reusers and the cost-efficiency of the data publishing interfaces when reuse grows.

Hence, the research question of this PhD is: **"how can the data source interoperability of public datasets be raised, in order to lower the cost for adoption in route planning services?"**

In order to have an answer to this question, this book will go broader than merely discussing the technical aspects of datasources. I worked in close collaboration with communication scientists to study the management and publishing of data sources qualitatively as well. This might make this research question atypical for a PhD within software engineering.

2. ASSUMPTIONS AND LIMITATIONS

This work assumes HTTP is the uniform interface for Open Data publishing. Other protocols exist (e.g., ftp or e-mail), yet the scale of the adoption of HTTP servers for Open Data is irreversible. If you would be reading this dissertation at a time in the future where the HTTP protocol is not used any longer (which at the time of writing, I would describe as "unlikely"), I first of all would have to admit that this assumption in my PhD is terribly wrong. However, the experiments described in the next chapters would also work for other protocols: an identifier

strategy would still be needed over this new protocol (Chapter 3 and Chapter 4), as well as a way to fragment these datasets (Chapter 6).

A second assumption is that the datasets that can be published will not have any privacy constraints. In practise, from the moment a dataset may contain something that may identify people, they need to go through a privacy check before they would be able to published publicly. As this is a different research area, we made the assumption that every dataset mentioned in this PhD, has either no person data, or has gone through the necessary checks in order to be disseminated.

A limitation of the current work – yet part of future work – is that we will be focusing on public transport routing and not calculate routes over road networks. Calculating routes over a road network can however be solved in a similar fashion, following the principles described in the conclusion.

Finally, the last limitation is that we describe the cost for adoption, yet do not describe an economical model to calculate such a cost. Instead, we assume that by raising the interoperability, the cost for adoption will lower.

3. THE CHAPTERS AND PUBLICATIONS

When I started research at what was back then still called the MultiMediaLab, I was handed a booklet called “Is This Really Science? The Semantic Webber’s Guide to Evaluating Research Contributions” [1] written by Abraham Bernstein and Natasha Noy. In that booklet, a quote from Ernest Rutherford was used: “All science is either physics or stamp collecting”. This bold statement illustrates a useful distinction between two types of research: one that studies a phenomenon and creates hypotheses about it and the other that catalogues and categorizes observations. The next three chapters will show how we see and categorize the domain of data publishing, in order to see more clearly how we can contribute to this domain. In Chapter 6, we introduce and evaluate Linked Connections, in order to prove that it is indeed more cost-

efficient to host.

I based my dissertation on a collection of papers that I have (co-)authored. Yet, it is also bringing together findings from deliverables from the projects I have been part of, invited talks I have been giving, blog posts I have been writing, and on the side projects I have been side-tracked by out of curiosity. A short overview of the chapters and what they are based on, is given below:

Chapter 2 – Open Data and interoperability

This chapter is based on my explanation on Linked Open Data, which I have been teaching over the course of my research position.

Chapter 3 – Measuring interoperability

This chapter is based on the first journal paper I authored for the Computer journal [2]. In the paper, I tried to quantify the interoperability of governmental datasets, in order to find out which datasets on an Open Data Portal would need more work.

Chapter 4 – Raising interoperability of governmental datasets

This chapter describes three projects which each were valorized in a publication. One project was on creating better Open Data Portals [3], another on an Open Data policy for the DTPW [4], and a third project was on creating a Linked Data strategy for local council decisions as a way to simplify administrative tasks.

Chapter 5 – Transport data

This chapter comes in two parts: a part about data on the road, which is based on a publication on more recent work about real-time parking availabilities [6], and data about public transit route planning, which is based on the related work of the paper introducing Linked Connections [7].

Chapter 6 – Public Transit route planning over lightweight Linked Data interfaces

Finally, the last chapter before the conclusion was based on four papers that also nicely illustrate the thought process over time. My PhD Symposium

paper written back in 2013 [8] illustrated that I wanted to be able to answer *any kind of route planning question* (sic) over the Web of data. In 2015 I published a demo paper [9] that would explain in a proof of concept, that I meant to give the client more freedom to calculate routes the way they like. In 2016, I extended this proof of concept with wheelchair accessibility features [10] and looked into what would be more efficient for the information as a whole. Finally, in 2017 I published the paper evaluating the cost-efficiency of this new lightweight interface [7].

4. INNOVATION IN ROUTE PLANNING APPLICATIONS

In 2016, everyone wanted an app. Not having an app would mean not being able to call yourself a digitally advanced transit agency, even if it already has a website with exactly the same functionality. Tim Berners-Lee, in 1989, concluded his proposal for the Web with the advice that we should focus on creating a better information system, that works for anything and is portable, then again having to work on new fancy graphic techniques and complex extra facilities that do not tackle the root of the problem. Today, his conclusion could not be more on topic.

We do not need a separate route planning app for each transit agency. If you would ask smartphone users, they only need one application that returns a route from one place to another. We can see evidence of public transit authorities that understand this need. Public transit agencies share data among each other to include their data in each of their own app. However, instead of a solution, this now becomes a quadratic problem, in which each agency has to share their data with each other agency that they seem relevant. If we want an app that works world-wide, this approach will not scale. Moreover, we become dependent on the goodwill of the public transit agencies to implement features for specific use cases. Take

for example the ability to take into account your specific set of subscriptions, the ability to take into account wheel chair accessiblity, or the ability to assist you in planning your next multi-day international trip.

At the time of writing, Google Maps is the most popular route planning application.

Organically, we see that public transit companies understand this need, and share their data with Google Maps. While among digital citizens, this move is regarded as a long due step in the right direction, it is still questionable whether only Google Maps should receive this data. Giving one company the monopoly on creating a route planning experience for 100% of the population is not the solution either. Instead, we advocate for Open Data: everyone should be able to create and integrate a route planner in their own service offering.

How exactly such an open dataset should be published is the subject of this book. Datasets need to be integrated in various "views", which all work on top of a similar route planning *Application Programming Interface* (API). However, we should also be able to create different route planning APIs ourselves for different use cases that were not kept in mind by transit agencies when publishing the data. This entails that the raw data should be published – not only the answers to advanced questions –.

5. THE PROJECTS

This book has been written while working on European, Flemish, and bilaterally funded research projects. In November 2012, my first month at the MultiMedia Lab (now Internet and Data Lab), I was tasked with the further development of The DataTank. The DataTank is open source software to open up datasets over HTTP, while also adding the right metadata to these datasets. The first version of The DataTank was further developed thanks to a project at Westtoer, which needed a single point of reference for their tourism datasets. After that, the project for EWI helped further shaping this project as simple data portal software. The DataTank was initially created at Open Knowledge Belgium and iRail (for a background, read the preface) and was installed for the open data portal of

among others, Flanders, region of Kortrijk, Antwerp, and Ghent. Today, research on this project ceased and commercial support is available via third parties.

The first two years, I was funded on projects in the domain of e-government, where there was a need for better data dissemination. In all the aforementioned projects, the 5 stars of Linked Open Data was used as a framework. We would never however reach the 5 stars and would always be confronted with a glass ceiling: why would 3 stars not be enough? In these two years, we thus mainly described data in various formats, not often with well aligned data models, using the DCAT specification, which at the time was still being built.

Apps for Europe was a different kind of project. Its goal was to support governmental organizations with their first steps towards Open Data and organize co-creation events. The project provided me with travel budget to travel to among others Berlin, Manchester, Amsterdam, Paris, and Switzerland. It provided me with the understanding of when developers would start to use governmental datasets, and gave me the first insight in how and why public administrations maintain certain datasets.

In the next projects, we created our own way of evaluating open data policies by introducing a framework to study the data source interoperability. This aligns with the goal to maximize reuse, and thus to provably raise the interoperability of data sources. Times are changing, and instead of e-government, Open Data would now become more eagerly funded under the umbrella of Smart Cities. The first project that was not linked to e-government was the ITS VIS project with its.be, a public-private partnership that works on the European directive regarding *Intelligent Transport Systems* (ITS). It also set up a data portal at data.its.be, yet also put steps in the direction of interoperable semantics by transforming the DATEX2 specification to a Linked Data vocabulary.

In 2015, I had the opportunity to study the organizational challenges, together with communication scientists, at the *Flemish Department of Transport and Public Works* (DTPW) of the Flemish government. Three European directives (PSI, INSPIRE, and ITS) extended with own insights, created a clear willingness to publish data for

maximum reuse. How to implement such an Open Data strategy in a large organization, was still unclear. As we interviewed 27 data owners and directors, we came to a list of recommendations for next steps on all interoperability levels. This project marked the start for many more projects that would need our help for publishing data for maximum reuse, such as the local council decisions as Linked Data project, the Smart Roeselare project, the reuse assessment for the Flemish Institute for the Sea, and finally for the Smart Flanders program.

Table 1: An overview of the projects I was part of from November 2012 until June 2017 when at Internet and Data Lab.

Name	Period	Description and impact
Westtoer tourism	2012	Westtoer advanced The DataTank. Tourism Open Data portal created http://datahub.westtoer.be
An experimental data publishing platform for EWI	2012–2013	Advanced The DataTank with a connection to the popular CKAN, created the DCAT-AP extension. Shaped the current features of The DataTank
Apps for Europe	2012–2014	Advanced The DataTank and was the first project to bring together various … Apps for Ghent is still organized each year
Open Transport Net	2014–2016	A project on DCAT, data portals and Open Transport Data.
Flemish Innovation Study for its.be	2014–2017	Setting up a data portal and advancing the DATEX2 specification towards a Linked Data ontology
An open data vision for the Department of Mobility and Public Works	2015	A first project to methodologically raise the impact and interoperability of datasets to be published at the Flemish government.
Local Decisions as Linked Data	2016	Publishing local council decisions as Linked Data for administrative simplification

The OASIS team	2016–2018	Sharing experience between Ghent and Madrid on publishing Linked Data about public services and transport data.
Smart Roeselare	2017	Supporting the City of Roeselare with their Smart City vision to make more reuse of their data.
Raising the reuse of datasets maintained by the Flemish institute for the Sea	2016–2017	Towards an Open Sea Data innovation lab by making their data used and useful.
Smart Flanders	2017–2020	Supporting the 13 Flemish center cities and Brussels to publish real-time open data.

Each project contributed in their own respect to this dissertation. While some gave me access to research subjects, other projects gave me the freedom to further explore research directions I thought were worth further exploring.

REFERENCES

[1] Bernstein, A., Noy, N. (2014). *Is this really science? the semantic webber's guide to evaluating research contributions*. Technical report.

[2] Colpaert, P., Van Compernolle, M., De Vocht, L., Dimou, A., Vander Sande, M., Verborgh, R., Mechant, P., Mannens, E. (2014, October). *Quantifying the interoperability of open government datasets*. (pp. 50–56). Computer.

[3] Colpaert, P., Joye, S., Mechant, P., Mannens, E., Van de Walle, R. (2013). *The 5 Stars Of Open Data Portals*. In proceedings of 7th international conference on methodologies, technologies and tools enabling e-Government (pp. 61–67).

[4] Colpaert, P., Van Compernolle, M., Walravens, N., Mechant, P. (2017, April). *Open Transport Data for maximizing reuse in multimodal route planners: a study in Flanders*. IET Intelligent Transport Systems. Institution of Engineering and Technology.

[5] Buyle, R., Colpaert, P., Van Compernolle, M., Mechant, P., Volders, V., Verborgh, R., Mannens, E. (2016, October). *Local Council Decisions as Linked Data: a proof of concept*. In proceedings of the 15th International Semantic Web Conference.

[7] Colpaert, P., Verborgh, R., Mannens, E. (2017). *Public Transit Route Planning through Lightweight Linked Data Interfaces*. In proceedings of International Conference on Web Engineering.

[8] Colpaert, P. (2014). *Route planning using Linked Open Data*. In proceedings of European Semantic Web Conference (pp. 827–833).

[9] Colpaert, P., Llaves, A., Verborgh, R., Corcho, O., Mannens, E., Van de Walle, R. (2015). *Intermodal Public Transit Routing using Linked Connections*. In proceedings of International Semantic Web Conference (Posters & Demos).

[10] Colpaert, P., Ballieu, S., Verborgh, R., Mannens, E. (2016). *The impact of an extra feature on the scalability of Linked Connections*. In proceedings of ISWC2016.

Chapter 2
Open Data and Interoperability

> "We should work toward a universal linked information system, in which generality and portability are more important than fancy graphics techniques and complex extra facilities"
> — Tim Berners-Lee (1989).

"Where is this website getting its data from?", I asked myself while I was informed that my bus is arriving in five minutes. We all have used the word data before, yet it remains difficult, even in academic mids, to define what the word precisely means. I have, in vain, been looking for the one definition that would help me study publishing data generically. I have been thinking of my own definitions which, each time, I would gradually decide to move away from, as I would always be able to give a counter example that did not fit the definition. In this chapter, we will not introduce a standard definition for the term "data". Instead, we will talk about the interoperability between two or more datasets from four perspectives: the syntactic perspective, semantic perspective, the legal perspective and the perspective of asking questions perspective. That will introduce and motivate our view – as there is no "one way" to look at this – on data in the large scale context of the Web.

A *datum*, the singular form of the word *data* in Latin, can be translated as "a given". When someone or something makes for example the observation that a certain train will start from the Schaerbeek station, and by recording this observation in this text, we have created a datum. When we have many things that are given, we talk about a dataset, or simply, data.

In English, as well as in this PhD thesis, the word data is also often used as a singular word to refer to the abstract concept of a pile of recorded observations.

Data need a container to be stored in or transmitted with. We can store different observations in a written *document*, which in turn can be published as a book or can be shared online in a Web format. Data can also be stored within a *database*, to then be shared with others through e.g., *data dumps* or *query services*.

1. A DATA FORMAT

In order to store or transit data in a document, we need to agree on a *serialization* first. A simple example of such a serialization is *Comma-Separated Values* (csv) [1], in which the elements are separated by commas. Each line in

the file contains a record and each record has a value for each column. An illustration of how a train leaving from Schaerbeek station would look like in this tabular format, is given in Figure 1.

```
"id","starts from"
"Train P8008","Schaerbeek Station"
```

Figure 1: Example of a CSV file describing a train line that starts in the station of Schaerbeek

This is not the only way in which this data can be serialized into CSV. We can imagine different headings are used, different ways to identify "Train P8008", or even "starts from" and "Train P8008" to switch places. Each serialization only specifies a certain *syntax*, which tells how data elements are separated from each other. It is up to specifications on a higher abstraction layer to define how these different elements will relate to each other, based on the building blocks provided by the serialization format. The same holds true for other serializations, such as the hierarchical *Javascript Object Notation* (JSON) or *Extensible Markup Language* (XML) formats.

As people decide how datasets are shaped, human language is used to express facts within these serializations. Noam Chomsky, who laid the foundations of generative grammar, built models to study language as if it were a mathematical problem. In Chomskian generative grammar, the smallest building block to express a fact one can think of, is a *triple*. Such a triple contains a *subject* (such as "Train P8008"), a *predicate* (such as "starts from"), and an *object* (such as "Schaerbeek Station"). Within a triple, a subject has a certain relation to an object, and this relation is described by the predicate. In Figure 2, we illustrate how our CSV example in Figure 1 would look like in a triple structure.

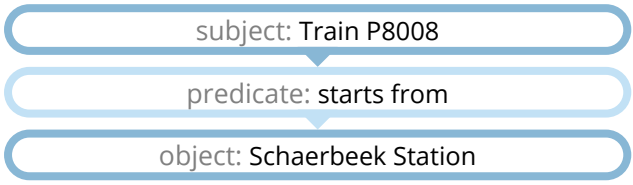

Figure 2: The example in Figure 1 encoded and illustrated as a triple

This triple structure – rather than a tabular or hierarchical data model – helps studying data in its most essential form. It allows to extent the theory we build for one datum or triple, to more data. By re-using the same elements in triples, we are able to *link* and weave *a graph* of connected statements. Different dedicated serializations for triples exist, such as Turtle [2] and N-Triples [3], yet also specifications exist to encode triples within serializations like JSON, CSV, or XML.

JSON, CSV, and XML are at the time of writing popular formats on the Web, that can be intepreted by any modern programming language today. No knowledge is required about these serializations in the remainder of this book.

2. DOCUMENTING MEANING

Let us perform a thought experiment and imagine three triples published by three different authorities. One machine publishes the triple in Figure 2, while two other publish the triples illustrated in Figures 3 and 4 – the serialization used can be any – representing the facts that the train P8008 starts from Schaerbeek Station, Schaerbeek Station is located in Schaerbeek City, and the Belgian singer Jacques Brel was born this city.

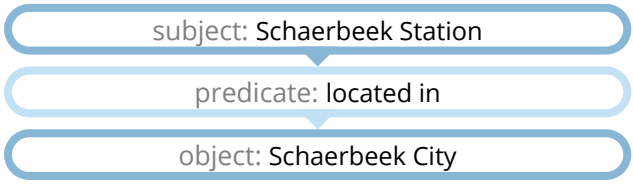

Figure 3: On a second machine, the fact that Schaerbeek Station is located in the city of Schaerbeek is published.

A user agent is software that acts on behalf of a user.

When a *user agent* visits these three machines, it can now answer more questions than each of the machines would be able to do on their own, such as: "What trains leave in the city in which Jacques Brel was born?". A problem occurs however. How does this user agent know whether "Schaerbeek City" and "Schaerbeek" are the same entity?

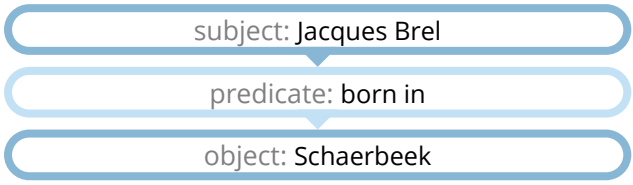

Figure 4: On a third machine, the fact that Jacques Brel, the famous Belgian singer, was born in Schaerbeek is published.

Semantics, in this context, refers to how technology can assist in comparing the meaning between two entities.

Instead of using words to identify things, numeric identifiers are commonly used. This way, every organization can have their context in which entities are described and discussed. E.g., the station of Schaerbeek could be given then identifier *132*, while the city of Schaerbeek could be given the identifier *121*. Yet for an outsider, it becomes unclear what the meaning is of *121* and *132*, as it is unclear where its *semantics* are documented, if documented at all.

Resources can be anything, including documents, people, physical objects, and abstract concepts [4]. Within the *Resource Description Framework* (RDF), they can be identified using a *Uniform Resource Identifier* (URI), or represented by a literal value (such as a date or a string of characters).

Linked Data solves this problem by using Web identifiers, or HTTP *Uniform Resource Identifiers (URIS)* [5]. It is a method to distribute and scale semantics over large organizations such as the Web. When looking up this identifier – by using the HTTP protocol or using a Web browser – a definition must be returned, including links towards potential other interesting *resources*. The triple format to be used in combination with URIS is standardized within the RDF [4]. In Figure 5, we exemplified how these three triples would look like in RDF.

```
<http://phd.pietercolpaert.be/trains#P8008>
<http://phd.pietercolpaert.be/terms#startsFrom>
<http://irail.be/stations/NMBS/008811007> .

<http://irail.be/stations/NMBS/008811007>
<http://dbpedia.org/ontology/location>
<http://dbpedia.org/resource/Schaerbeek> .

<http://www.wikidata.org/entity/Q1666>
<http://www.wikidata.org/entity/P19>
<http://dbpedia.org/resource/Schaerbeek> .
```

Figure 5: The three triples are given a global identifier and are added using RDF's simple N-Triples serialization.

One can use Linked Open Vocabularies to discover reusable URIs [6].

The URIs used for these triples already existed in other data sources, and we thus favoured using the same identifiers. It is up to a data publisher to make a choice on which data sources can provide the identifiers for a certain of entities. In this example, we found WikiData to be a good source to define the city of Schaerbeek and to define Jacques Brel. We however prefer iRail as a source for the stations in Belgium. As we currently did not find any existing identifiers for the train route P8008, we created our own local identifier, and used the domain name of this dissertation as a base for extending the knowledge on the Web.

3. INTELLECTUAL PROPERTY RIGHTS AND OPEN DATA

As *Intellectual Property Rights* (IPR) legislation diverges across the world, we only checked the correctness of this chapter with European copyright legislation [7] in mind.

When a document is published on the Web, all rights are reserved by default until 70 years after the death of the last author. When these documents are reused, modified and/or shared, the conscent of the copyright holder is needed. This conscent can be given through a written statement, but can also be given to everyone at once through a public license. In order to mark your own work for reuse, licenses, such as the Creative Commons

licenses, exist, that can be reused without having to invent the same legal texts over and over again.

Copyright is only applicable on the container that is used for exchanging the data. On the abstract concept of facts or data, copyright legislation does not apply. The European directive on *sui generis* database law [8] specifies that, however, *databases* can be partially protected, if the owner can show that there has been "qualitatively and/or quantitatively a substantial investment in either the obtaining, verification, or presentation of the content of the database" [9]. It allows a database owner to protect its database from (partial) replication by third parties. So, while there is no copyright applicable on data itself, database rights may still be in place to protect a data source. In 2014, the Creative Commons licenses were extended [10] to also contain legal text on the sui generis database law, and would since then also work for datasets.

More information on the definition of Open Data maintained by Open Knowledge International is available at opendefinition.org [11].

Data can only be called *Open Data*, "when anyone is able to freely access, use, modify, and share for any purpose (subject, at most, to requirements that preserve provenance and openness)". Some data are by definition open, as there is no *Intellectual Property Rights* (IPR) applicable. When there is some kind of IPR on the data, an *open license* is required. This license must allow the right to reuse, modify, and share without exception. From the moment there are custom restrictions (other than preserving provenance or openness), it cannot be called "open".

While the examples given here may sound straightforward, these two IPR frameworks are the source of much uncertainty. Take for example the case of the Diary of Anne Frank [12], for which it is unclear who last wrote the book. While some argue it is in the public domain, the organization now holding the copyright states the father did editorial changes, and the father of Anne Frank died much later. For the reason of avoiding complexity when reusing documents – and not only for this reason – it is desired that the authoritative source can verify the document's *provenance* or authors at all time, and a license or *waiver* are included in the dataset's metadata.

When this book would be processed for the data facts

that are stored within this book, what happens to copyright? Rulings in court help us to understand how this should be interpreted. A case in the online newspaper sector, Public Relations Consultants Association Ltd vs. The Newspaper Licensing Agency Ltd [13] in the UK in 2014, interpreted the 5th article of the European directive on copyright in the way that a copy that happens for the purpose of text and data mining is incidental, and no conscent should be granted for this type of copies.

Next, considering the database rights, it is unclear what a *substantial* investment is, regarding the data contained within these documents. One of the most prominent arrests for the area of Open (Transport) Data, was the ruling of the British Horseracing Board Ltd and Others vs. William Hill Organization Ltd [14], which stated that the results of horse races collected by a third party was not infringing the database rights of the horse race organizer. The horse race organizer does not invest in the database, as the results are a natural consequence of holding horse races. In the same way, we argue the railway schedules of a public transport agency are not protected by database law either, as a public transport agency does not have to invest in maintaining this dataset. These interpretations of copyright and sui generis are also confirmed by a study of the EU Commission on intellectual property rights for text mining and data analysis [9].

4. SHARING DATA: A CHALLENGE RAISING INTEROPERABILITY

A dataset is created in order for it to be shared at some point in time. If it is not shared with other people, it will need to be shared with other systems (such as an archive)

This is a use case we came across when visiting the *Flemish Department of Transport and Public Works* (DTPW), discussed in Chapter 4.2.

or shared with your own organization. Take for example a dataset that is created within a certain governmental service, for the specific use case of solving questions from members of parliament. While at first, the database's design may not reflect a large number of use cases, the dataset is not just removed after answering this question. Instead, it will be kept on a hard drive at first, in order to be more efficient when a follow-up question would be asked. The dataset may also be relevant for answering different questions, and thus, a project is created to share this kind of documents proactively [15].

Imagine you are a database maintainer for this project, and someone asks to share the list of the latest additions with you. You set up a meeting and try to agree on terms regarding the legal conditions, you agree on how the data will be sent to the third party, discuss which syntax to use, the semantics of the terms that are going to be shared, and which questions that should be answered over the dataset. The *protocol* that is created can be documented, and whenever a new question needs to be answered, the existing protocol can be reused, or, when it would not cover all the needs any longer, needs to be rediscussed. When now more people want to reuse this data, it quickly becomes untenable to keep having meetings with all reusers. Also vice versa, when a reuser wants to reuse as much data as possible, it becomes untenable to have meetings with all data publishers.

In the previous chapter, we discussed datasets for which the goal was to *maximize the reuse*, which entails maximizing both the number of reusers, as well as maximizing the amount of questions each reuser can solve. In order to motivate data consumers to start reusing a certain dataset, some publishers rely on the intrinsic motivation of citizens [16], yet when performing a cost-benefit analysis, the cost to manually fix the heterogenity of datasets is still too high compared to the benefits of the company itself [17]. In this PhD, we explore the possibilities to lower this cost for adoption for a certain dataset, by lowering the *data source interoperability* [18], which we define as how easy it is to bring two, or more, datasets together.

4.1. Legally

The first level is the legal level: data consumers must be allowed for these two datasets to be queried together. When for a certain dataset a specific one on one contract needs to be signed before it can be used, the cost for adoption for data consumers becomes too high [17]. When two datasets are made available as Open Data and have an Open License attached to it, the interoperability problems will be lower. Even for datasets that are possibly in the *public domain*, the Open Data movement advocates for a clear disclaimer on each dataset with the publisher's interpretation.

4.2. Technically

The second level is the technical level, which entails how easy it is to bring two datasets physically together. Thanks to the Internet, we can acknowledge this is possible today, yet the protocols to exchange data diverge, from e-mail, to the *File Transfer Protocol* (FTP), or the *HyperText Transfer Protocol* (HTTP).

HTTP [19], the protocol that powers the Web, allows actions to be performed on a given resource, called *request methods* or also called HTTP *verbs*. The protocol defines for every method whether it is a *safe* method, defined as a method that does not change anything on the server.

It is up to the application's developers to implement these methods. It is not uncommon that the "safe" property is not respected, resulting in undesired consequences in for example Mendeley back in 2012 [20].

An example of a safe method is GET. By executing a GET request, you can download the representation of a certain resource. The same representation can be requested over and over again with the same result, until the resource changes its state – e.g., when a train is delayed (a change in the real world), or when someone adds a comment to an article using a POST request –. The result of this request can be cached within clients, in servers, or in intermediate caches.

POST requests are not safe. Each time a POST request is executed, a change or side-effect may happen on the server. It is thus not cacheable, as each new request must be able to trigger a new change or must be able to result in a different response.

The protocol has also more than a hand full of other

methods, has *headers* to indicate specifics such as which encoding and format to use, how long the response can be cached, and has response codes to indicate whether the request was succesful. In this PhD, it is not up for debate whether or not to use HTTP: the scale of adoption it has reached at the time of writing makes it the natural choice for the uniform interface. The rationale followed in this paper would also remain valid with other underlying protocols. In 2015, the HTTP/2 protocol [21] was drafted and is today seeking for adoption. It is fully backwards compatible with HTTP/1.1, the currently widely adopted specification.

4.3. Syntactically and semantically

The third kind of interoperability describes whether the serializations of both datasets can be read by a user agent. When deserialized, the data can be accessed in memory. The meaning behind these identifiers can conflict when the same identifier is used for something different. When they do not conflict, there may also be multiple identifiers for the same objects. In both ways, it will lower the *semantic interoperability*.

5. INTEROPERABILITY LAYERS

The term interoperability has been coined in several research papers, both qualitative as quantitative. What the authors of these papers have in common, is that they propose to structure interoperability problems in different categories. For example, back in 2000, the IMI model [22] was introduced, in order to discuss the exchange of object oriented data between software systems. The IMI model has only three levels: the syntax layer, the object layer and the semantic layer. The *syntax layer* is responsible for "dumbing down" object-oriented information into document instances and byte streams. The *object layer*'s goal is to offer applications an object-oriented view on the information that they operate upon. This layer defines how hierarchical elements are structured. Finally, the *semantic*

layer is the layer that makes sure the same meaning is used for the same identifiers. The authors argue that each of these three layers should have their own technology to have a fully interoperable service. Today we indeed see that XML syntax has an object specification called RDF/XML, which standardizes how RDF triples can be contained within such a document.

Interoperability problems were also described as problems that occur while integrating data. The goal of an integration process is to provide users with a unified view on one machine. Four types of heterogeneity are discussed: *Implementation heterogeneity* occurs when different data sources run on different hardware and *structural heterogeneity* occurs when data sources have different data models, in the same way as the object level in the imi model. *Syntax heterogeneity* occurs when data sources have different languages and data representations. Finally, *semantic heterogeneity* occurs when "the conceptualisation of the different data sources is influenced by the designers' view of the concepts and the context to be modelled".

Legal, organizational, political, semantic, and technical are then again the five levels in which Europe categorizes their datasets on the European Union's data portal in order to indicate for what interoperability level this dataset could be used. These levels are intended to discuss how data and knowledge is spread on a policy level within big organizations, such as Europe as a whole.

Finally, in a review on interoperability frameworks, four categories of interoperability are again identified: technical, syntactic, semantic and organizational. The first three are the same as in this paper, yet the *organizational interoperability* focuses on high level problems such as cultural differences and alignment with organizational processes. Within the data source interoperability, we consider the effects from organizational heterogeneity to have affected the semantic interoperability.

6. THE 5 STARS OF LINKED DATA

In order to persuade data managers to publish their data as "Linked Open Data", Tim Berners-Lee introduced a 5 star rating for data in 2009, cfr. Figure 6. The *first star* advocates to make the data available on the web in whatever format, similar to our *technical interoperability* layer. Furthermore, for open data, it also advocates for an *Open License*, similar to the *legal interoperability* layer. The *second star* requires that the dataset is *machine readable*. This way, the data that needs to be reused can be copy pasted into different tools, allowing for the data to be studied in different ways more easily. The *third star* advocates for an *open format*, similar to the syntactic interoperability, making sure anyone can read the data without having to install specific software. The *fourth star* advocates for the use of URIs as a way to make your terms and identifiers work in a distributed setting, and thus allowing a discussion on semantics using the RDF technology. Finally, the *fifth star* advocates for reusing existing URI vocabularies, and to link your data to other datasets on the Web. Only by doing the latter, the Web of data will be woven together. The 5 star system to advocate for Linked Open Data has gained much traction and cannot lack from any introduction to Linked (Open) Data or maintaining datasets on web-scale. We however are cautious using the 5 stars in our own work, as it could give the impression a 100% interoperable 5 star dataset exists and no further investments would be needed at some point to make it better. For realists who rightfully believe a perfect dataset does not exist, wonder why going for 5 star data is needed... Are 3 stars not good enough? When presenting the roadmap as *interoperability layers* however, maintaining a dataset is a never ending effort, where each interoperability can be improved. Raising interoperability is not one-sided: the goal is to be as interoperable as possible with an information system. When the information management system, and the datasets in it do more effort towards interoperability, your dataset can also be made more interoperable over time.

This idea is also put forward by the *World Wide Web Consortium* (W3C) best practices guide for data on the Web [23] of the W3C.

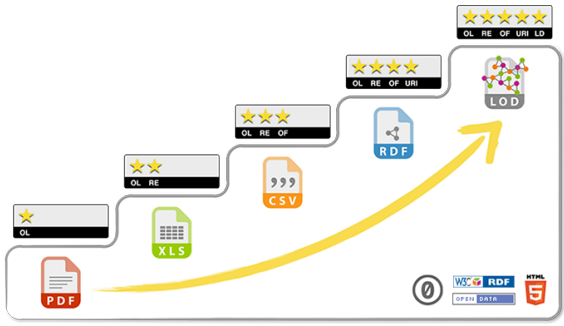

Figure 6: The 5 star scheme towards Linked Data, as used by Tim Berners-Lee to advocate for better data exchange over the Web.

7. INFORMATION MANAGEMENT: A PROPOSAL

Let's create a system to distribute data within our own organization for the many years to come. Our requirements would be that we want our data policy to scale: when more datasets are added, when more difficult questions need to be answered, when more questions are asked, or when more user agents come into action, we want our system to work in the same way. The system should also be able to evolve while maintaining backwards-compatibility, as our organization is going to change over time. Datasets that are published today, should still work when accessed when new personnel is in place. Such a system should also have a low entry-barrier, as it needs to be adopted by both developers of user agents as data publishers.

Tim Berners-Lee created his proposal for Information management on large scale within CERN in 1989 [24]. What we now call "The Web" is a knowledge base with all of mankind's data, which still uses the same building blocks

For an overview of REST, we refer to the second chapter in the PhD dissertation of Ruben Verborgh, a review of REST after 15 years [25], or the original dissertation of Roy Fielding [26], or Fielding's reflections about REST in 2017 [27].

as at the time of Tim Berners-Lee's first experiments. It was Roy Fielding that, in 2000 – 11 years after the initial proposal for the Web –, derived a set of *constraints* [26] from what was already created. Defined while standardizing HTTP/1.1, this set of "how to build large knowledge bases"-constraints is known today as *Representational State Transfer* (REST). As with any architectural style, developers can choose to follow these constraints, or to ignore them. When following these contraints, REST promises beneficial properties to your system, such as a good network efficiency, better scalability, higher reliability, a better user-perceived performance, and more simplicity.

Clients and servers implement the HTTP protocol so that their communication is technically interoperable. Just like Linked Data insists on using HTTP identifiers, REST's *uniform interface* constraint requires that every individual information resource on the Web is accessed through a single identifier – a URI – *regardless* of the concrete format it is represented in. Through a process called *content negotiation*, a client and a server agree on the best representation. For example, when a resource "station of Schaerbeek" is identified by the URI http://irail.be/stations/NMBS/008811007 and a Web browser sends an HTTP request with this URI, the server typically sends an HTML representation of this resource. In contrast, an automated route planning user agent will usually ask and receive a JSON representation of the same resource using the same URI. This makes the identifier http://irail.be/stations/NMBS/008811007 semantically interoperable, since clients consuming different formats can still refer to the same identifier. This identifier is also *sustainable* (i.e., semantically interoperable over time), because new representation formats can be supported in the future without a change of protocol or identifier [25].

In order to navigate from one representation to another, *controls* are given within each representation. Fielding called this *Hypermedia As The Engine Of Application State* (HATEOAS): when a user agent once received a start URL, it would be able to answer its end-user's questions by using the controls provided each step of the way.

8. INTELLIGENT AGENTS

Now, we can create a user agent that provides its end-users with the nearest railway station. A user story would look like this: when you push a button, you should see the nearest station relative to your current location. In a *Service Oriented Architecture* (SOA), or how we would naturally design such an interaction in small-scale set-ups, we expose a functionality on the server, which requires the application to send its current location to the server. A URL of such a request would look like this: http://{my-service}/nearestStation?longitude=3.14159&latitude=51.315. The server then responds with a concise and precise answer. This minimizes the data that has to be exchanged when only one question is asked, as only one station needs to be transferred. Does this advantage weigh up to the disadvantages?

The number of information resources – or documents – that you potentially have to generate on the server, is over a trillion. As it is unlikely that two people – wanting to know the nearest railway station – are at exactly the same locations, each HTTP request has to be sent to the server for evaluation. Rightfully, SOA practitioners introduce rate limiting to this kind of requests to keep the number of requests low. An interesting business model is to sell people who need more requests, a higher rate limit. Yet, did we not want to maximize the reuse of our data, instead of limiting the number of requests possible?

When assuming that a precision of 11m, or 4 decimal places in both longitude and latitude, is enough, then we would still have 6.48×10^{12} URLs exposed for a simple feature.

8.1. Caching for scalability

As there are only 646 stations served by the Belgian railway company, describing this amount of stations easily fits into one information resource identified by one URL. When the server does not expose the functionality to filter the stations on the basis of geolocation, all user agents that want to solve any question based on the location of stations, have to fetch the same resource. This puts the server at ease, as it can prepare the right document once each time the stations' list is updated. Despite the fact that now all 646 stations had to be transferred to the user

For instance, https://api.irail.be/stations

In Computer Science, this is also called the *principle of locality*

agent, and thus consumed significantly more bandwidth, also this user agent can benefit. For example, when soon after, a similar question is executed, the dataset will already be present in the client's cache, and now, no data at all will need to be transferred. This raises the user-perceived performance of a user interface. When now the number of end-users increases by a factor of thousand per second – not uncommon on the Web –, it becomes easier for the server to keep delivering the same file for those user agents that do not have it in cache already. When it is not in cache of the user agent itself, it might already be in an intermediate cache on the Web, or in the server's cache, not leading to the server having to invest in CPU time per user. Caching, one of the REST constraints, thus has the potential to eliminate some network interactions and server load. When exploited, a better network efficiency, scalability, and user-perceived performance can be achieved.

We empirically study the effect of caching on route planning interfaces in Chapter 7.

8.2. The all knowing server and the user agent

In a closed environment – for instance, when you are creating a back-end for a specific app, you assume the information lives in a *Closed World* – a server is assumed to be all knowing. When asking for the nearest station, the server should know all stations in existence and return the closest one. Yet, when I would live nearby the border of France, a server that assumes a Belgian context will not be able to give me a correct answer to this question. Instead, I would have to find a different server that also provides an answer to this similar question. Furthermore, when now, I would love to find the nearest wheelchair accessible station, no answer can be returned, as the server does not expose this kind of functionality. The server keeps user agents "dumb".

On the Web, we must take into account an *Open World Assumption*: one organization can only publish the stations it knows about, or a list of stations they use for their own

Not to mention that the complexities that would arise when datasets would not be interoperable. For instance, if not everyone had the same definition of "a station"

use case. An implication of this [28], is that for user agents, it becomes impossible to get the total number of stations: following links, they can infinitely keep crawling the Web whether there is someone that knows something about a station that was not mentioned before. However, a user agent can be intelligent enough, to, within its own context, decide whether or not the answers it received until now, are sufficient. For instance, when creating a public transport app in Belgium, the app can be given a complete list of transport agencies in Belgium, according to its developer. When this user agent *discovers* and retrieves a list of stations published by all transport agencies, it can assume its list will be sufficient for its use case.

When a user agent is given a start URL, it should be able to follow links to discover other information resources.

8.3. Queryability

For scalability and user-perceived performance, we argued that publishing information resources with fewer server-side functionality is a better way of publishing Open Data. We also argued that in the case of solving all questions on the server, the server pretends to be all knowing, when in fact it is not: it just has a closed world approach. The user agent has to adopt the specific context of the server, and is not able to answer questions over multiple servers at once. A user agent should be able to add its own knowledge to a problem, coming from other data sources on the Web or from the local context it currently has.

It is not because two datasets are legally, technically, syntactically, and semantically interoperable, that a user agent can answer questions over these two datasets without any problem. A query answering algorithm also needs to be able to access the right parts of the data at a certain moment. First, we can make the server expose over a trillion information resources of our data, by answering all specific questions on the server. However, as previously discussed, the questions that can be answered depend on the server infrastructure and the server context. Combining different services like this becomes increasingly difficult. This idea is taken from SOA, and is not a great match for maximizing the reuse of open datasets.

Another option is that the data publisher publishes one

data dump of all data within an organization. While this is preferred over a service when a diverse range of queries is needed, a data dump has clear drawbacks too. For instance, when the dataset would update, the entire dataset needs to be downloaded and reprocessed by all user agents again. Furthermore, for a user agent that only wants to know the nearest station, the *overhead* to download and interpret the entire file becomes too big. The possibility that data is downloaded that will never be used by the user agent becomes bigger too.

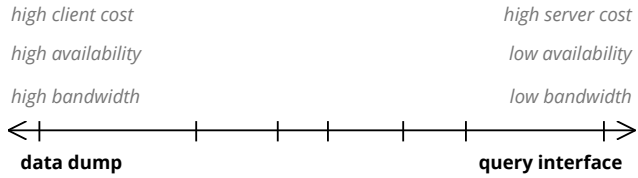

Figure 7: The *Linked Data Fragments* (LDF) idea plots the expressiveness of a server interface on an axis. On the far left, an organization can decide to provide no expressiveness at all, by publishing one data dump. On the far right, the server may decide to answer any question for all user agents.

Instead of choosing between data dumps and query services, we propose to explore options in-between. Only one very simple question, asking for all data, can be answered by the server itself. It is up to the user agent to solve more specific questions locally. When the dataset is split in two fragments, e.g., all stations in the north of the country and all stations in the south of the country, the user agent can, depending on the question that needs to be solved, now choose to only download one part of the dataset. When publishing data in even smaller fragments, interlinked with each other, we help user agents to solve questions in a more timely fashion. The more fragments are published by the server, the more *expressive* we call a server interface. With this idea in mind, *Linked Data Fragments* (LDF) [29], illustrated in Figure 7, were introduced. Publishing data is making trade-offs: based on a publisher's budget, a more or less expressive interface can be installed.

Just like a database that will be prepared to answer certain types of questions, a hypermedia interface can also be modeled for the questions it needs to answer. By providing extra controls and by providing more fragments of the data to be retrieved, the queryability of the interface will raise for particular problems.

9. CONCLUSION

The goal of an *Open Data policy* is to share data with anyone for any purpose. The intention is to *maximize the reuse* of a certain data source. When a data source needs to attract a wide variety of use cases, a cost-benefit analysis can be made: when the cost for adoptions is lower than the benefits to reuse this data, the data is going to be adopted by third parties. This cost for adoption can be lowered by making data sources more *interoperable*.

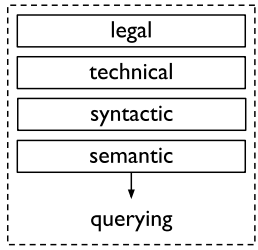

Figure 8: The layers of data source interoperability for allowing user agents to query your data.

In order to make it more feasible for developers to make their app work globally instead of only for one system, we introduced the term *data source interoperability*. We define interoperability as how easy it is to evaluate questions over two data sources. The term can then be generalized by comparing your data source with all other data sources within your organization, or even more generally, all other data sources on the Web. We discussed – and will further study – interoperability on five levels, as illustrated in Figure 8:

1. Are we *legally* allowed to merge two datasets?
2. Are there *technical* difficulties to physically bring the datasets together?
3. Can both *syntaxes* be read by the user agent?
4. Do both datasets use a similar *semantics* for the same identifiers and domain models?
5. How difficult is it to *query* both datasets at once?

We only consider data here that has to be maximally disseminated.

On the legal level, public open data licenses help to get datasets adopted. It is not because the content of a license complies to the Open Definition, that the cost for adoption is minized. Licenses that are custom made may not be as easy to use, as it needs to be checked whether it indeed complies. On the technical level, we are still working on better infrastructure with HTTP/2. On the syntactic level, we are working on efficient ways to serialize triples data facts in data [30], such as with the on-going standardizing work with JSON-LD, on-going research and development within for example HDT, CSV on the Web, or rdf-thrift. On the semantic level, we are looking for new vocabularies to describe, avoid conflicts in identifiers using URIs, and make sure our data terms are documented. And finally, when querying data, we are still working on researching how we can crawl the entire Web with an Open World assumption [28], or how to query the Web using more structured Linked Data Fragments interfaces.

In this chapter we did not mention data quality. One definition defines data quality as the perceived data quality when an end-user wants to use it for a certain use case. E.g., "The data quality is not good, as it does not provide me with the necessary precision or timeliness". However, for other use cases the data may be perfectly suitable. Another definition mentions data quality as how close it corresponds to the real world. Furthermore, is it really a core task of the government to raise the quality of a dataset beyond the prime reason why the data was created in the first place? When talking about Open Data, the goal is to maximize data adoption by third parties. Even bad quality data – whatever that may be – might also be interesting to publish.

Interoperability is a challenge that is hard to advance on your own: also other datasets need to become

interoperable with yours. It is an organizational problem that slowly finds its way into policy making. In the next chapter, we discuss how we can advance interoperability within large organizations.

REFERENCES

[1] Shafranovich, Y. (2005, October). *Common Format and MIME Type for CSV Files*. IETF.

[2] Beckett, D., Berners-Lee, T., Prud'Hommeaux, E., Carothers, G. (2014, February). *RDF 1.1 Turtle – Terse RDF Triple Language*. W3C.

[3] Carothers, G., Seaborne, A. (2014, February). *RDF 1.1 N-Triples – A line-based syntax for an RDF graph*. W3C.

[4] Schreiber, G., Raimond, Y. (2014, February). *RDF 1.1 Primer*. W3C.

[5] Berners-Lee, T., Fielding, R., Masinter, L. (2005, January). *Uniform Resource Identifier (URI): Generic Syntax*. IETF.

[6] Vandenbussche, P.-Y. (2017). *Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web*. Semantic Web 8.3 (pp. 437–452).

[7] European Parliament (2001). *Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society*. EUR-LEX.

[8] European Parliament (1996). *Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases*. EUR-LEX.

[9] Triaille, J-P., de Meeûs d'Argenteuil, J., de Francquen, A. (2014, March). *Study on the legal framework of text and data mining*.

[10] Creative Commons (2013, December). *Creative*

Commons 4.0/Sui generis database rights draft.

[11] Open Knowledge International (2004). *The Open Definition*.

[12] Moody, G. (2016, April). *Copyright chaos: Why isn't Anne Frank's diary free now?*. Ars Technica.

[13] Judgment of the Court (Fourth Chamber) (2014, June). *Public Relations Consultants Association Ltd v Newspaper Licensing Agency Ltd and Others.*. EUR-LEX.

[14] Judgment of the Court (Grand Chamber) (2004, November). *The British Horseracing Board Ltd and Others v William Hill Organization Ltd.*. EUR-LEX.

[15] Research Service of the Flemish Government (). *Overview of datasets of the Flemish Regional Indicators*.

[16] Baccarne, B., Mechant, P., Schuurman, D., Colpaert, P., De Marez, L. (2014). *Urban socio-technical innovations with and by citizens*. (pp. 143–156). Interdisciplinary Studies Journal.

[17] Walravens, N., Van Compernolle, M., Colpaert, P., Mechant, P., Ballon, P., Mannens, E. (2016). *Open Government Data': based Business Models: a market consultation on the relationship with government in the case of mobility and route-planning applications*. In proceedings of 13th International Joint Conference on e-Business and Telecommunications (pp. 64–71).

[19] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Bernsers-Lee, T. (1999, June). *Hypertext Transfer Protocol – HTTP/1.1*. IETF.

[20] Verborgh, R. (2012, July). *GET doesn't change the world*.

[21] Belshe, M., Peon, R., Thomson, M. (2015, May). *Hypertext Transfer Protocol Version 2 (HTTP/2)*. IETF.

[22] Melnik, S., Decker, S. (2000, September). *A Layered Approach to Information Modeling and Interoperability on the Web*. In proceedings of the ECDL'oo Workshop on the Semantic Web.

[23] Farias Lóscio, B., Burle, C., Calegari, N. (2016, August).

Data on the Web Best Practices.

[24] Berners-Lee, T. (1989, March). *Information Management: A Proposal*.

[25] Verborgh, R., van Hooland, S., Cope, A.S., Chan, S., Mannens, E., Van de Walle, R. (2015). *The Fallacy of the Multi-API Culture: Conceptual and Practical Benefits of Representational State Transfer (REST)*. Journal of Documentation (pp. 233–252).

[26] Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.

[27] Fielding, R. T., Taylor, R. N., Erenkrantz, J., Gorlick, M. M., Whitehead E. J., Khare, R., Oreizy, R. (2017). *Reflections on the REST Architectural Style and "Principled Design of the Modern Web Architecture"*. Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (pp. 4–11).

[28] Hartig, O., Bizer, C., Freytag, J.C. (2009). *Executing SPARQL queries over the Web of Linked Data*. In proceedings of International Semantic Web Conference. Springer Berlin Heidelberg.

[29] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P. (2016, March). *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. Journal of Web Semantics.

Chapter 3
Measuring Interoperability

" When you take apart a ship plank by plank and assemble it again, is it still the same ship? "
— Theseus's paradox.

In order to maximize potential reuse, an open data publisher wants its data source to be as interoperable – on all levels – as possible with other datasets world-wide. Currently, there is no way to identify relevant datasets to be interoperable with and there is no way to measure the interoperability itself. In this chapter we discuss the possibility of comparing identifiers used within various datasets as a way to measure semantic interoperability. We introduce three metrics to express the interoperability between two datasets: the identifier interoperability, the relevance, and the number of conflicts. The metrics are calculated from a list of statements which indicate for each pair of identifiers in the system whether they identify the same concept or not. The effort to collect these statements is high, and while not only relevant datasets are identified, also machine-readable feedback is provided to the data maintainer. We will therefore also look at qualitative methods to study the interoperability within a large organization.

When raising the interoperability between two or more datasets, the cost for adoption will lower. A user agent that can process one dataset, will be able to ask questions over multiple without big investments. If only we could become more interoperable with all other datasets online, then our data would be picked up by all existing user agents. Of course, this is not a one way effort: other datasets also need to become more interoperable with ours, and with all other datasets on the Web. It becomes a quadratically complex problem, where each dataset needs to adopt ideas from other datasets, managed by different organizations with different ideas. In this chapter, we will look at ways to measure the interoperability of datasets, which in the same way is also a quadratic problem as each dataset needs to be compared with all other datasets.

How can we measure the impact of a certain technology on the interoperability? A first effort we did in 2014, was comparing identifiers of available open datasets across different cities [1].

1. COMPARING IDENTIFIERS

A simplistic approach would be to classify relations between the identifiers (IDs) of two datasets in four categories. One dataset can contain the same identifier as a dataset in the other. When this identifier identifies the same real-world thing in both datasets, we call this a *correct id match*. When this identifier identifies a different real-world thing, we can call this a false id match or an *id conflict*. In the same way, we have a *correct different id* and a *false different id*.

Table 2: A first dataset about the city of Ghent

id	long	lat	type_sanit	fee	id_ghent
1	3.73	51.06	new_urinal	free	PS_151

In Table 1, we give an example of an open dataset of public toilets in the city of Ghent. In Table 2, a similar dataset about the city of Antwerp is given. An identifier is each element that is not a literal value such as “3.73”. Identifiers may be elements of the data model, as well as real-world objects described within the dataset. In these two tables, we can label some identifiers as conflicting, other elements as correct id matches, others as correct different ids or false different ids. However, the labeling can happen differently depending on the domain expert. Certainly when “loose semantics” are utilized it becomes difficult to label an identifier as identifying “the same as” another identifier.

Table 3: A second dataset, now about the city of Antwerp

id	long	lat	type	fee	description
1	4.41	51.23	urinal	none	Hessenhuis

1.1. An initial metric

In our research paper in 2013 for the sake of simplicity,

we assumed that domain experts would be able to tell whether two identifiers are “the same as” or “not the same as”. When we would have a list of statements, classifying these identifiers in four categories, we would be able to deduct a metric for the interoperability of these two datasets. The first metric we introduced was the ID ratio.

$$ID_{\%} = \frac{\#\,(\text{correct id matches})}{\#\,(\text{real-world concept matches})}$$

Figure 1: The identifier ratio ID%

We apply this formula on top of our two example datasets in Table 1 and Table 2. The correct identifier matches are: “id”, “long”, “lat”, “fee” (4). Conflicts would be the identifier “1” (1). Real-world concept matches would be: “id”, “long”, “lat”, “type/type_sanit”, “fee”, “urinal/new_urinal” (6). An initial metric would thus score these two datasets as 66% interoperable, with 1 conflict.

Of course, this depends on our view of the world and our definitions of the things within this dataset.

In a simple experiment, we asked programmers to give a score for the interoperability between 5 different datasets and a reference dataset. The outcome revealed that there were clear design issues with this metric: when there would be a low amount of real-world matches, the score would be influenced quickly, as the number of samples to be tested is low. A full report, and the research data, on this experiment can be found on a Github repository at pietercolpaert/IIOP-demo.

1.2. Identifier interoperability, relevance, and number of conflicts

There are two problems with ID%. First, The ID% may be calculated for 2 datasets which are not at all relevant to compare. This can lead to an inaccurate high or low interoperability score, as the number of real-world concept matches is low. Second, when different datasets are brought together, some identifiers are used more than others. An identifier which is used once has the same weight as an identifier that is used in almost all facts.

Instead of calculating an identifier ratio, we introduce a *relevance* (ρ) metric. A higher score on this metric would

mean that two datasets are relevant matches to be merged. We now can introduce two types of relevance: the relevance of these two datasets as is ($\rho_{identifiers}$), and the relevance of these two datasets when all identifiers would be interoperable ($\rho_{real\text{-}world}$). We define the ρ as the number of occurences of an identifier when both datasets would be merged and would be expressed in triples. The $\rho_{identifiers}$ in our example would become 8, as 2 times 4 statements can be extracted from a merged table. The $\rho_{real\text{-}world}$ would become 12, as the number of occurences that would be left when the dataset would be 100% interoperable, would be 2 times 6. We then define the *Identifier Interoperability* (IIOP), as the ratio between these two relevance numbers. For this dataset, the IIOP becomes 66%. Coincidentally the same as the ID%, with more rows in the dataset, it would quickly be different. When repeating the same experiment as with the ID%, we now see a credible interoperability metric when comparing the $\rho_{real\text{-}world}$, the number of conflicts and the IIOP.

While these three metrics may sound straightforward, it appears to be a tedious task to gather statements. What is the threshold to decide whether two identifiers are the same as or not the same as [2]? A philosophical discussion arrises – cfr. Theseus's paradox – whether something identified by one party can truly be the same as something identified by someone else. It is up to researchers to be cautious with these statements, as a same as statement may be prone to interpretation.

1.3. The role of Linked Open Data

Conflicts are the easiest to resolve. Instead of using local identifiers when publishing the data, all identifiers can be preceded by a baseURI. E.g., instead of "1", an identifier can contain http://{myhost}/{resources}/1. This simple trick will eleminate the number of conflicts to zero. In order to have persistent identifiers, organizations introduce a URI strategy. This strategy contains the rules needed to build new identifiers for datasets maintained within this organization.

Persistent identifiers are identifiers that stand the test of time. Cool URIS don't change.

Another problem that arose, was that a third party

cannot be entirely sure what was intended with a certain identifier. Linked Data documents these terms by providing a uniform interface for resolving the documentation of these identifiers: the HTTP protocol. This way, a third party can be certain what the meaning is, and when the data is not linked, it can link the data itself more easily by comparing the definitions. The more relevant extra data is provided with this definition, the easier linking datasets should become.

Linked Data helps to solve the semantic interoperability, for which the identifier interoperability is just an indicator. By using HTTP URIs, it becomes possible to avoid conflicts, at least, when an authority does not create one identifier with multiple conflicting definitions and when this definition, resolvable through the URI, is clear enough for third parties not to missinterpret it. The effort it would take for data publishers to document their datasets better and to provide "same as" statements with other data sources is similar to just providing URIs within your data and linking them with existing datasets from the start. However, the theoretical framework built, provides an extra motivation for Linked Data.

2. STUDYING INTEROPERABILITY INDIRECTLY

It is understandable that policy makers today invest time and money in raising interoperability, rather than measuring its current state. When however no globally unique identifiers are used, can we work on semantic interoperability? Even more generally, is interoperability a problem data publishers are worried about at all? Or is their foremost concern to comply to regulations to publish data as is? In this section, we are going to measure interoperability indirectly, by trying to find qualitative answers to these questions by studying the adoption by third parties, studying the technology of published datasets and finally studying the organizations themselves.

Making the cost-benefit analysis for data reuse, when

more third party adoption can be seen, we can also conclude the datasets become more interoperable when the benefits did not change. We can perform interviews of market players and ask them how easy it is to adopt certain datasets today. This is an interesting post-assessment. In a study we executed within the domain of multimodal route planners [3], it appeared that only a limited amount of market players could be identified that reused the datasets. Each time, reusers would replicate the entire dataset locally. We concluded that still only companies with large resources can afford to reuse data. This is again evidence that the cost for adoption, and thus the interoperability problems need to lower.

In this paper [3] we reported on the study we executed

When studying data policies today, we can observe a certain maturity on the five layers of data source interoperability. For example, when a well-known open license is used, we can assume the legal interoperability is higher than with other datasets. When the data is publicly shared on the Web and accessible through a URL, we can say the technical interoperability is high, and when documents and server functionality are documented through hypermedia controls, we can assume the querying interoperability is high. At some level, in order to raise interoperability, we have to make a decision for a certain standard or technology together. From an academic perspective, we can only observe that such a decision does not hinder other interoperability layers. When studying the interoperability, we could then give a higher score to these technologies that are already well adopted. Today, these technologies would be HTTP as a uniform interface, RDF as a framework to raise the semantic interoperability, and the popular Creative Commons licenses for the legal aspect. In the next chapter, we will study organizations based on the reasoning to set up a certain kind of service today. Epistemologically, new insights can be created by the data owner to publish data in a more interoperable way.

With the upcoming HTTP/2.0 standard, all Web-communication will be secure by default (HTTPS). In the rest of this book, we will use HTTP as an acronym for the Web's protocol, yet we will assume the transport layer is secure.

3. QUALITATIVELY MEASURING AND RAISING INTEROPERABILITY ON THE 5 LAYERS

Aspects such as the organizational, cultural, or political environment may be enablers for a higher interoperability on several levels, but should as such not be taken into account for studying data source interoperability itself. In among others the European framework ISA² and EIF, also political interoperability is added.

When quantitatively studying dataset interoperability, we can discuss each of the five layers seperately. In this section we give an overview of all layers and how they can be used in interviews. Depending on the quality that we want to reach in our information system, we may be more strict on each of the levels, and thus for every project, a different kind of scoring card can be created. This is in line with other interoperability studies [4], that overall agree that interoperability is notoriously difficult to measure, and thus qualitatively set the expectation for every project.

3.1. Legal interoperability

Political, organizational, or cultural aspects may influence legal interoperability. When studied on a global scale, there is a political willingness to reach a cooperation on IPR frameworks. The better the overarching IPR framework, such as a concensus on international copyright law, better legal interoperability is achieved. Today, still different governments argue they need to build their own licenses for open data publishing, which again however would lower the legal interoperability. When there is a reason to lower the interoperability, the interviewer should ask whether and why these reasons weigh up to the disadvantages of lower interoperability in the opinion of the interviewee.

When only considering *open* datasets, measuring the interoperability boils down to making sure that a machine can understand what the reuse rights are. The first and foremost measurement we can do, is testing whether we can find an open license attached to the data source. The Creative Commons open licenses for example, each have a URI, which dereference into an RDF serialization which provides a machine readable explanation about the data.

3.2. Technical interoperability

Next, we can check how it is made available. When the dataset can be reached through a protocol everyone speaks, technical interoperability is 100%. Today, we can safely assume everyone knows the basics of the HTTP protocol. When a dataset has a direct URL over which it can be downloaded, we can say it is technically interoperable. The technical interoperability can even raise when we also refer to related different pages in the response headers or body of the response, yet we keep studying these aspects for the querying interoperability layer. An interesting test to publish data in a technically interoperable way can be that given a URL, one is able to download the entire dataset.

3.3. Syntactic interoperability

The syntactic interoperability of two datasets is maximized when both datasets use the same serialization. In the case of Open Data, the syntactic interoperability is not worth measuring, as given a certain library in a certain progamming language, every syntax can just be read in memory without additional costs (e.g., the difference in cost to parse XML vs. JSON will be marginal). Quantitatively studying the syntactic interoperability thus involves studying whether the actual data is also machine readable, in accordance with the third star of Tim Berners-Lee.

However, much again depends on the intent. If it's the core task of a service to build spreadsheets with statistics from various sources, and because of the fact that these statistics are summaries to, for instance, solve parliamentary questions, these spreadsheets in a closed format need to be opened up. Will it be an added value if another government service now also provides a machine readable version of these datasets? Researchers should be careful in what the outcome of blindly measuring interoperability means for the internal government processes. Quick wins are not always the best solution: raising syntactic interoperability – and other kinds of interoperability as well – means changing internal processes and software. A holistic view of processes is

needed. Interviews with data managers may thus be more constructive than merely measuring the syntactic interoperability.

Some syntaxes allow semantic mark-up, while other specifications and standards for specific user agents do not allow a standard way for embedding triples. Still in this case, the identifiers for documents and real-world objects should remain technically interoperable. We therefore could measure how ready a certain government is for *content negotiation* and whether there is a strategy for maintaining identifiers in the long run.

3.4. Semantic interoperability

An indicator for semantic interoperability can be created by comparing identifiers, as shown in the first section. However, we can also qualitatively assess how well semantic interoperability issues are dealt with on a less fundamental level. With a closed world assumption, we can create a contract between all parties that redefines which set of words are going to be used. We can see evidence of this in specific domains where standards are omnipresent. E.g., within the transport domain, DATEX2 and GTFS are good examples. However, from the moment this data needs to be combined within an open world – answering questions over the borders of datasets – these standards fail at making a connection.

Essentially, researching the semantic interoperability boils down to studying how good identifiers are regulated within an information system. Again a qualitative approach can be taken as well. A perfect world does not exist, but we can interview governmental bodies to find out why they made decisions towards a solution, and look whether semantic interoperability was a problem they tried to tackle somehow.

Without RDF there is no automated way to find semantic interoperable properties or entities. Depending on the qualitative study, and whether the semantic interoperability is an issue that will be identified, we can decide to introduce this technology to the organization, if not yet known. Even when a full RDF solution is in place, still the semantic interoperability can thus vary across

datasets.

3.5. Querying interoperability

The LDF axis – illustrated again in figure 2 – was at all time used to discuss the queryability of interfaces. When only a query interface would be offered, we would – as a quick win – also indicate that there is the possibility of offering data dumps. Hypermedia apis – what this axis advocates for – are not yet part of off the shelf products, which make it particularly difficult for organizations to explore other options.

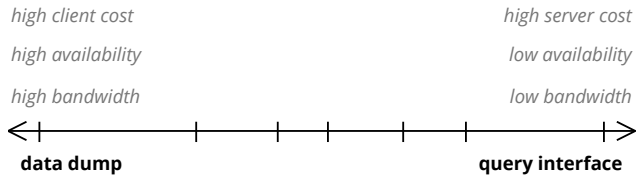

Figure 2: The *Linked Data Fragments* (LDF) idea plots the expressiveness of a server interface on an axis. On the far left, an organization can decide to provide no expressiveness at all, by publishing one data dump. On the far right, the server may decide to answer any question for all user agents.

The best example of full interoperability would be that by publishing the data using the HTTP protocol, the data becomes automatically discovered and used in this application.

4. CONCLUSION

Raising interoperability entails making it easier for all user agents on the Web to discover, access and understand your data. We explored different ways to measure interoperability between two datasets. For Open Data however, we would then need to generalize – or scale up – these approaches to interoperability between a

dataset and “all other possible open datasets”.

A first – not advised due to the investments needed – way is to compare identifiers between these two datasets or systems. When the same identifiers are used over the two datasets, we can assume a high interoperability. However, whether identifiers actually identify the same thing is prone to interpretation. We see this exercise mainly as proof that Linked Data is the right way forward: by using HTTP URIs, we use the *uniform interface* of the Web to document identifiers. Furthermore, using Web addresses instead of local identifiers will make sure another Web framework – RDF – can be used to discuss the relation between different real-world objects. Through comparing identifiers, we showed the importance of RDF to raise the semantic interoperability. Also within the legal, technical, syntactical, and querying interoperability, RDF may play its role. Without RDF, we would have to rely on a different mechanism to retrieve machine readable license information, or we would have to rely on a specification that reintroduces syntax rules for hypermedia.

Only time will tell whether using Web addresses for identifiers – and consequentially RDF – will become the norm for all aspects. Well established standards then will have to evolve to RDF as well.

A second way to measure interoperability between datasets is to study the effects. The fact that Open Data by definition should allow data to be redistributed and mashed up with other datasets, means it becomes hard to automatically count each access to a data fact coming from your original dataset. A successful open data policy can thus be measured by the number of parties that declare that they reuse the dataset. Interviewing the parties that voluntarily declared this fact, may result in interesting insights on how to raise the interoperability. However, this is a post-assessment when an Open Data policy (or a data sharing policy) is in place.

A third way is to interview data owners within an organization on their own vision on Open Data. During the interviews, questions can be asked on why certain decisions have been taken, each time categorizing the answer at an interoperability layer. This is an epistemological approach, in which data owners will get new insights when explaining their vision within the context of the 5 interoperability layers. Depending on the interviewed organizations, different technologies can be assumed accepted. While some organizations will find it

evident to use HTTP as a uniform interface, others may still send data that should be public to all stake holders using a fax machine. For fewer cases – discussed in the next chapter – RDF was evident. Thefore, we need a good mix between desk research on the quality of the data and interviews with data maintainers in order to create a good overview of the interoperability.

In the next chapter we introduce our own context of the organizations we worked with and will choose a qualitative approach to studying interoperability.

REFERENCES

[1] Colpaert, P., Van Compernolle, M., De Vocht, L., Dimou, A., Vander Sande, M., Verborgh, R., Mechant, P., Mannens, E. (2014, October). *Quantifying the interoperability of open government datasets*. (pp. 50–56). Computer.

[2] Halpin, H., Herman, I., J. Hayes, P. (2010, April). *When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web*. In proceedings of the World Wide Web conference, Linked Data on the Web (LDOW) workshop.

[3] Walravens, N., Van Compernolle, M., Colpaert, P., Mechant, P., Ballon, P., Mannens, E. (2016). *Open Government Data': based Business Models: a market consultation on the relationship with government in the case of mobility and route-planning applications*. In proceedings of 13th International Joint Conference on e-Business and Telecommunications (pp. 64–71).

[4] C. Ford, T., M. Colombi, J., R. Graham, S., R. Jacques, D. (1980). *A Survey on Interoperability Measurement*. In proceedings of 12th ICCRTS.

Chapter 4
Raising Interoperability of Governmental Datasets

> "We want to see a world where data creates power for the many, not the few."
>
> — Open Knowledge International.

constantly weighing trade-offs: will you invest more in keeping the history of your dataset? Or will you invest more in creating specific materialized versions of your specific data? As researchers within our team at IMEC, we were able to study different organizations from the inside out to study the barriers to publish data as Open Data. We discuss three clusters of datasets that we have collaborated on to shape the Open Data policy. First, we introduce a proof of concept for Local Council decision as Linked Data, a project that gets further developed in 2017–2018. Then we discuss the datasets of the Department of Transport and Public Works – for whom we did an assessment of their Open Data strategy – and their history. Finally we discuss the governance of data portals and how we can discover datasets through their metadata. We found that raising the interoperability within these organization is not easy, yet at a certain moment we were able to suggest actions to improve on the state of the art. In each of the three use cases we formulate a list – supported by an organization – of next actions to take. Datasets necessary for the use case of route planning often originate from these datasets that initially were not built with route planning in mind. Yet, still, we would like that route planners ideally take into account council decisions in order to update their maps.

As we study datasets governed within Europe, we need to understand the policy background. Three European directives – the *Public Sector Information* (PSI) directive, the *Infrastructure for Spatial Information in the European Community* (INSPIRE) directive, and the *Intelligent Transport Systems* (ITS) directive – regulate how respectively public sector information, geospatial information, and transport data need to be shared. These directives still leave room for implementation.

First, the PSI directive states that each policy document that has been created in the context of a public task, should be opened up. Thanks to this directive, open became the default: instead of having to find a reason to make a document publicly available, now a good reason

needs to be made public in order to keep certain datasets private.

Next, the INSPIRE directive regulates how to maintain and disseminate geospatial data. It defines the implementation rules in order to set up Spatial Data Infrastructure (SDI) for the European Community. INSPIRE has a holistic approach, defining rules for among others metadata, querying, domain models or network services. Although it originated for purposes of EU environmental policies and policies or activities which may have an impact on the environment, the 34 themes in INSPIRE, can be used within other domains as well. Take for example the domain of public transport, where entities such as the "railway station of Schaerbeek" or administrative units such as "the city of Schaerbeek" also need to be described using spatial features.

Finally, the ITS directive focuses on the transport domain itself. It is in its essence not a data sharing directive. It has delegated acts in which the sharing of data in an information system is described.

In this chapter, we will chronologically run through three periods. The first period was when the PSI directive gained traction and when the first Open Data Portals were set up. I just started my research position and was trying to create a framework to study Open Data, its goals, and how we could structure the priorities when implementing such a data portal. Next, we discuss transport datasets within the *Flemish Department of Transport and Public Works* (DTPW). There, the overlap between the three directives becomes apparent. Finally, we discuss an opportunity to publish the content of local council decision as the authoritative source of a variety of facts.

1. DATA PORTALS: MAKING DATASETS DISCOVERABLE

1.1. The DataTank

The first project I helped shaping at IMEC was The DataTank [1]. It is an open-source project that helps, in its essence, to automate data publishing, and bring datasets closer to the HTTP protocol and the Web. It contains tools to republish unstructured data in common Web-formats. Furthermore, it automatically gets the best out of the HTTP protocol, adding:

- An ACCESS-CONTROL-ALLOW-ORIGIN: * header, which allows this resource to be used within a script on a different domain.
- A caching header, which allows clients to know whether the resource updated or not.
- Paging headers when the resource would be too large.
- Content negotiation for different serializations.

In order to allow a quick overview of the data in the resource, the software also provides a best effort HTML visualization of the data.

When creating The DataTank, we built it around the 5 stars of Linked Open Data [2]. For each star, our goal was to provide a HTTP interface that would not overload the back-end systems. The data source interoperability can be lifted on the legal level, where the metadata also includes a URI to a specific license. Also on the technical level, it published a dataset over HTTP and added headers to each response. Furthermore, syntactically, it provides each dataset in a set of common serializations. Semantically, The DataTank can read, but does not require, data in RDF serializations as well. Using the tdt/triples package, one can automatically look up the documentation of a certain entity using HTTP URI dereferencing [3]. This requires the dataset itself to use URIs and an RDF serialization.

1.2. 5 stars of Open Data Portals

When making the requirements analysis of the future roadmap of The DataTank back in 2013, we published the 5 stars of data portals [4]. The idealistic goal was that a data portal should allow the data to be maintained as if it was a common good. The five stars of Open Data Portals are to be interpreted cummulatively and were defined as follows:

★ A dataset registry
: A list of links towards datasets that are openly licensed

★★ A metadata provider
: Make sure the authentic sources inside your organization are adding the right metadata fields (e.g., according to dcat). This list of datasets should in its turn be licensed openly, so that other portals can aggregate or cherrypick datasets.

★★★ A cocreation platform
: Support a conversation about your data.

★★★★ A data provider
: Make sure the resources inside your organization are given a unique identifier and that you have interoperable access to the datasets themself, not only its metadata.

★★★★★ A common datahub
: Have a way for third parties to contribute to your dataset.

The DataTank's functionality focuses on the second star and development on further stars were never persued. Data projects such as Wikidata, Open Corporates, Wikipedia, or Open Street Map come closest to what was envisioned with the fifth star.

1.3. Open Data Portal Europe and the interoperability of transport data

The European Data Portal brings together all data that

The portal is available at https://www.europeandataportal.eu/

is available on national Open Data Portals in Europe. Analysis of the current 11,003 datasets can create an overview of data interoperability in Europe.

On a legal level, we notice that the majority of datasets use standard licenses, which increases legal interoperability. However, the Belgian public transport datasets are still not on the European data portal as they do not have an Open License. Technically, while 85.5% of the datasets are directly accessible through a URL, 1,584 (14.5%) dataset descriptions only link to an HTML page with instructions how to access the dataset. On a semantic level, only 295 (2.7%) datasets use an RDF serialization.

1.4. Queryable metadata

Data describing other datasets, typically maintained by a data portal, need to be available for reuse as well. We can thus apply the theory of maximizing reuse of datasets to DCAT data. The standard license within The DataTank and within CKAN is the Creative Commons Zero waiver, which provides for full legal interoperability with any other dataset. The data is technically available over HTTP and is typically available in one or more RDF serializations. However, we still notice some interoperability issues with data portal datasets today.

CKAN is at the time of writing the most popular data portal software.

One of the problems happens – despite using RDF – on the semantic level. For instance, when harvesting the metadata from European member states' data portal on the European data portal, new URIs are created per dataset instead of reusing the existing ones.

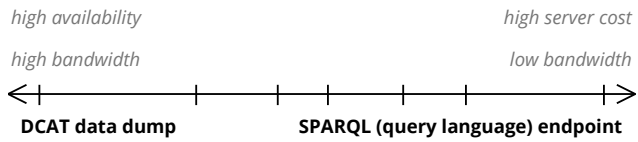

Figure 1: The *Linked Data Fragments* (LDF) axis applied to metadata.

One of the goals behind DCAT is to allow metadatasets

with similar semantics to be maintained decentrally. However, the querying itself still happens centrally. We proposed an in-between solution with TPF. [5] This way, a European data portal would not have to replicate all datasets anymore, but would only be the authoritative registry for national data portals. Thanks to the TPF interface and extra building blocks such as a full text search exposed on the level of each member state, the same user interface would be able to be created. Furthermore, the harvesting that takes place today would be automated using HTTP caching infrastructure.

2. OPEN DATA IN FLANDERS

In 2015, we had the opportunity study a common vision throughout all the suborganizations within the *Flemish Department of Transport and Public Works* (DTPW) in Flanders. We studied the background of several datasets, and without taking a position ourselves, we would interview all data maintainers and department directors on their experience with "Open and Big Data". On the basis of this input, we created recommendations for action that would help forward the Open Data policy within the DTPW.

In this section, we first describe the background of different key datasets in Flanders, to then discuss the specific datasets at the DTPW. We then report on the workshops and the challenges that came out of the interviews and explain our recommendations for action.

2.1. A tumultuous background

In Flanders, the *Road Sign Database* (RSD) project started in 2008 as a result of an incident the year before where a trucker was jammed under a bridge, as the driver was unaware the truck was too high. The investigation after the incident showed there were no traffic signs notifying drivers of a maximum height for vehicles. As a result, the traffic in the city was jammed for several hours. As this was not the first time something like this happened, the RSD was born by ministrial decree, building a complete

central database of traffic signs in Flanders. However, the RSD has been the subject of data management research [6], as the database did not live up to its expectations.

The responsibility to implement this database was split: the Agency for Roads and Traffic (ART) would have to maintain the traffic signs on the regional roads, while the *Flemish Department of Transport and Public Works* (DTPW) would have to maintain the traffic signs on the local roads. A company took 360° pictures of all roads in Flanders, and using these pictures, all road signs were indexed in a register. Two years later, by August 2010, this inventory was complete for all 308 municipalities and the DTPW made the RSD of the municipalities available via a software application. In order to keep the database up to date, the munipalicaties were asked to update it through this application, and requested to sign an agreement. Some municipalities refused to sign the agreement, and others, who signed the agreement, complained this initial application was too complex and untrustworthy. Quickly, less than 30% of the municipalities kept using the RSD [6].

In 2011, for its own needs, ART lauched a roads database instead of the RSD, which now includes specific information on the roads. In the same year, also Google Street View was launched in Flanders, launching an application to view the 360° pictures Google took of all streets, which local councils found easier to use than the RSD application. In March 2013, a new, more stable and user-friendly application of the RSD was launched, yet the adoption of the application remained low. The original cost of the RSD was estimated to be €5 million, yet by Augstus 2010, this estimate has risen to €15 million, and by March 2013, when the new more user-friendly version should have launched, had risen to €20 million. In April 2014, yet another database was launched at the Agency for Geographic Information: the General Information Platform for the Public Domain. The database collects all road works in order not to block roads that are currently part of a deviation. Up to date, local governments are legally bound to filling out numerous database of the higher government, yet in practise, only few databases remain well maintained.

In August 2014, a new minister of mobility was appointed. While first the new minister was looking into making it also legally binding to fill out the database, he later looked at limiting the scope of the RSD to speed signs, defeating its initial purpose, yet making sure third parties would show the right speed limit in in-car navigation systems. He would also demand a study to see what other mobility datasets could be shared as Open Data for the purpose of serving route planner developers, which in its turn funded the research reported in this paper.

This aligns well with the European "Once Only" principle.

In order to off-load local governments, one way that is currently looked into is to make local council decisions available as Linked Data. As part of the PSI directive, local administration need to open up the decisions taken in the local council. The numerous databases local governments have to fill out today, would this way be able to fetch the fragments of the local council decisions they are interested in themselves. Today, this approach is being evaluated. Perhaps this approach can be rolled out over all 308 municipalities. We will discuss this project in more detail in the next section.

In order to lower the cost for adoption for third parties to integrate the data that will be published, we look into raising the interoperability of these data sources. In the next chapter, we describe the method in order to find the challenges the DTPW still sees that need to be overcome in order for all agencies to publish their data for maximized reuse.

2.2. Discussing datasets at the Department of Transport and Public Works

Out of 27 interviews with data owners and directors working in the policy domain of mobility, we collected a list of datasets potentially useful for multimodal route planning. Definitions of *a dataset* however diverged depending on who was asked. Data maintainers often mentioned a dataset in the context of an *internal database*, used to fulfil an internal task, or used to store and share data with another team. When talking to directors, a dataset would be a *publicly communicated dataset*, e.g., a

dataset for which metadata can be found publicly, a dataset that would be discussed in politics, or a dataset the press would write stories about. In other cases, a dataset would exist *informally* as a web page, or as a small file on a civil servant's hard drive.

A data register of the mobility datasets that are part of the Open Data strategy can now be found at http://opendata.mow.vlaanderen.be/. The list consists of *publicly communicated datasets* as well as *informal data sources* published on websites. During the interviews, we were able to gather specific challenges related to specific datasets useful for multimodal route planning, summarized in the following table: for a dataset to be truly interoperable, all boxes need to be ticked.

Table 4: Selection of studied datasets with their interoperability levels as of October 2016

Dataset	legal	tech	syntax	semantic	querying
Traffic Events	open license	yes	XML	no URIs	file
Roads database	open license	yes	XML	no URIs	no
Validated statistics	open license	yes	CSV	no URIs	no
Information websites	no	yes	HTML	no URIs	linked documents
Public Transit timetables	closed license	over FTP	ZIP	no URIs	dump
Road Signs	no license	yes	none	no URIs	no
Address database	open license	yes	XML	PoC	dump and service
Truck Parkings	open license	yes	XML	no URIs	file
Metadata catalogue	open license	yes	XML	yes	dump

Traffic events on the Flemish highways

This dataset is maintained by the Flemish Traffic Center, has an open license and is publicly available. It decribes the traffic events, only on the highways, to which the core tasks of the traffic center is limited. The datasets can be downloaded in XML. For the semantics in this XML, two versions in two different specifications (OTAP and DATEX2) are available, for which the semantics can be looked up manually. The elements described in the files are not given global identifiers however, making it impossible to refer to a similar object in a different dataset. The dataset is small and is published as a dynamic data dump. As the dataset is small enough to be contained in one file, it can be fetched over HTTP regularly, as well as the updates. The HTTP protocol works well for dynamic files, as caching headers can be configured in order not to overload the server when many requests happen in a short time. The file, except for the *semantic interoperability*, thus provides also as a good dataset for federated route planning queries.

Road database for regional roads

The road database for the regional roads is maintained by the ART. It is a geospatial dataset and already has to comply to the INSPIRE directive. Its geospatial layers are thus already available as web services on the geospatial access point of Flanders: http://geopunt.be. The roadmap in 2016 was to also add an open license and to also publish the data as linked files using the TN-ITS project's specification (http://tn-its.eu/).

Validated statistics of traffic congestion on the Flemish highways

Today, validated statistics of traffic congestions on the Flemish highways are published under the Flemish Open Data License by the Flemish Traffic Center. A website was developed, which allows someone that is interested to create charts of the data, as well as export the selected statistics as XLS or CSV. The *legal*, *technical*, and *syntactic interoperability* are thus fully resolved. Yet when looking at

the *semantic interoperability*, no global identifiers are used within the dataset. Furthermore, when looking at the *querying interoperability*, machines are even discouraged from using the files, as a test for whether you are a human (a captcha) is used to prevent machines from discovering and downloading the data automatically. When requesting a csv file, the server generates the csv file with historic data on the fly from the database.

Information Websites

Examples of such datasets are the real-time dataset of whether a bicycle elevator and tunnel is operating (http://fietsersliften.wegenenverkeer.be/), a real-time dataset of whether a bridge is open or not (http://www.zelzatebrug.be/) shows when a bridge north of the city of Ghent will open again (when closed), or a dataset of quality labels of car parks next to highways (http://kwaliteitsparkings.be). The three examples mentioned can be accessed in HTML. Nevertheless, this as well is a valuable resource for end-user applications, as when the page would be openly licensed and when the data would be annotated with web addresses, the data can be extracted and replicated with standard tools and questions can be answered over these different data sources. These three examples are always only *technologically* and *syntactically* interoperable, as they use HTML to publish the data, yet there are no references to the meaning of the words and terms used. Furthermore, there is no open license on these websites, not explicitly allowing reuse of this data. Finally, as the data can easily be crawled by user-agents and thus replicated, we reason that in a limited way, the data would be able to be used in a federated query.

Public transit time tables maintained by De Lijn

Planned timetables, as well as access to a real-time webservice, can be requested through a one-on-one contract. This contract results in an overly complex legal interopability. First, a human interaction needs to request access to the data, which can be denied. Furthermore, in

the standard contract, it is not allowed for a third party to sublicense the data, which makes republishing the data, or a derived product, impossible. The planned timetables can be retrieved in the GTFS specification, which is an open specification, making the dataset *syntactically interoperable*. The identifiers used within this dataset for, e.g., stops, trips, or routes do not have a persistency strategy. Therefore, the *semantic interoperability* cannot be guaranteed. As a dump is provided, potential reusers have access to the entire data source reliably. The querying interoperability could be higher when the dataset would be split in smaller fragments.

Road Sign Database (RSD)

The database, in October 2016, is still only available through a restricted application. It is a publicly discussed dataset, as its creation was commissioned by a decree. On a regional level, the RSD is in reality two data stores: one database for regional road signs, managed by ART, and a database which collected the local road signs, managed by the department itself. Some municipalities would however also keep a copy of their own road signs on a local level, leading to many interoperability problems when trying to sync. Sharing this data with third parties however only happens over the publicly communicated RSD, which is only accessible through the application of the RSD itself.

Address database

A list of addresses is maintained as well by another agency, called Information Flanders. The database has to be updated by the local administrators , just like the RSD. Thanks to the simplicity of the user-interface and the fact that it is mandatory to update the database while changing, removing, or adding addresses, the database is well adopted by the local governments. It is licensed under an open license, and it is published on the Web in two ways: a data dump is updated regularly, and a couple of web services, which work on top of the latest dataset are provided. Currently, Information Flanders is creating a Proof of Concept (PoC) to expose the database as Linked

Data, as such every address will get a URI.

Truck parkings on the highways

This dataset needs to be shared with Europe, which in its turn makes this dataset publicly available at the European Union's data portal (http://data.europa.eu/euodp/en/data/dataset/etpa). The dataset is available publicly, under an open license, as XML, using the DATEX2 stylesheet. The file however does not contain persistent identifiers, thus it is impossible to guarantee the *semantic interoperability*. As with the traffic events, the file allows for querying by downloading the entire file.

Open Data portal's metadata

In order for datasets to be found by, e.g., route planning user agents, they need to be discoverable. The metadata from all datasets in Flanders are available at http://opendata.vlaanderen.be in RDF/XML. The metadata is licensed under a Creative Commons Zero license, and for each dataset and its way to be downloaded (distribution), a URI is available. In order to describe the dataset, the URI vocabulary DCAT is used, which is a recommendation by the European Commission in order to describe data catalogues in an interoperable way. However, within INSPIRE, another metadata standard was specified for geospatial data sources. GeoDCAT-AP is at the time of writing being created to align INSPIRE and DCAT (https://joinup.ec.europa.eu/node/139283). It is thus far, the only dataset that complies in an early form to all the interoperability levels introduced in this paper.

2.3. Challenges and workshops

We organised two workshops: one to validate the outcomes of the interviews with the different governmental organisations, the other to align the market needs with the governmental Open Data roadmap. In the first workshop, we welcomed a representative of each organisation within the DTPW that we had already met during a one on one interview. In the first half, we had an

introductory program where we summarised the basics of an Open Data policy: the open definition, the implementation of the PSI directive in Flanders, and the interoperability layer model. Furthermore we also gave a short summary of the results of the interviews with the market stakeholders. The key challenges were listed, discussed, initially identified by the heads of division of the DTPW. In order to identify these challenges, all interviews were first analysed in search of arguments both pro and con an open data policy. In the second half of this workshop we had three parallel break-out sessions in which we discussed unresolved questions that came out of the interviews. The arguments that returned most often were bundled and summarised into ten key challenges:

Should data publishing be centralised or decentralised within the department and what process should be followed?

This challenge refers to how data should reach the market and the public. A variety of scenarios can be envisaged here, each with benefits and disadvantages. This is not only a very practical challenge, but also one that relates to responsibility, ownership, and the philosophy behind setting up an open data policy. Potential scenarios that may resolve this challenge are also dependent on political decisions and the general vision for data management at the policy level. The workshop showed that a lot of political and related organisational aspects come into play in relation to this challenge. Attention to the balance between what is strategically possible and technically desirable is key in tackling this challenge.

Ensuring reusers interpret the data correctly

This refers to the fact that the context in which data are generated within government needs to be very well understood by potential reusers. Certain types of data require a certain domain expertise to be interpreted in a correct manner. While good and sufficient metadata can partially answer this challenge, in very specific cases a meeting with related data managers from the opening

organisation will be required to avoid misinterpretation.

Acquiring the right means and knowledge on how to publish open data within our organisation

Setting up an open data policy also requires internal knowledge, particularly in larger and complex organisations. This means that the right people need to be identified internally, giving them access to training, while also giving them responsibility and clearing them of other tasks. In other cases, there may be a need to attract external knowledge on the topic that can then be internalized. In any event, proper training (and for example a train-the-trainer programme) is key in developing and executing a successful open data strategy.

Knowing what reusers want

If the goal of opening up is to maximize reuse of data, it is important to understand what potential reusers are looking for, not only in terms of content but also in terms of required standards, channels and interactions. Various forms of interaction can be used to gain this insight and the most appropriate one will depend on the organisations involved and their goals. One-to-one meetings are preferably avoided to alleviate concerns of preferential treatment, but co-creation workshops, conferences, and study days, can be a potential solution to this challenge.

Influencing what reusers do with the data

This is a challenge that governments certainly struggle with: providing open data means giving up control over what happens with that data. The question captured here is how governments can guide, nudge, or steer the reuse of data so that the resulting applications, services, or products still support the policies defined by them. While illegal reuse of data is by definition out of the question, the main challenge here – from the government's perspective – is how to deal with undesired reuse. Again, consultation and dialogue are key: if the market understands the logic behind certain datasets as well as the reasons behind the

government opening them up, undesired reuse becomes less of a potential issue. If on the other hand the market has ideas that government had not anticipated, a dialogue can take place on the practical implications of that reuse.

Supporting evidence based policy-making

Open data does not only serve reuse outside of the opening organisation, but can also be put to use within different departments and divisions for example. The challenge defined here is how to make optimal use of data to shape policies, based on real-life evidence. To resolve this, an internal department or cell that follows up all data-related activities, acts as single point of contact and defines data policies could play a role in examining how data can contribute to policy-making.

Creating responsibility

The main question here is where the role of government stops and to which extent it should further enhance or improve datasets beyond its own purposes. Furthermore, the basic minimum quality should be defined and explicated towards potential reusers. Tackling this challenge means clearly defining a priori where the role of government ends. As this is also a political discussion to some extent, having a clearly-communicated policy is key.

Raising government's efficiency

This challenge deals with the potential gains that open data can mean for the Department as an organisation of organisations, but also for the Department as an actor within the policy domain of Transport and Public Works. Internal processes need to be established to ensure that an efficiency gain at the level of the own organisation is enabled.

Ensuring sustainability once a dataset is published

Next to covering short-term initiative, long-term

processes also need to be set up within the organisation so that the open data policy is sustainable both for the government organisation as well as the outside world. This means a smart design of such processes and guidelines, which are also constantly evaluated and tested against practice.

Ensuring the technical availability of datasets

This final challenge questions the basic guarantees that government should provide towards the publication and availability of the datasets. At which point does this become a service that does not necessarily need to be provided by government for free and what is the basic level of support (e.g., is a paid SLA provided for 24/7 data availability and tech support)? Again this decision is a political one, but clearly communicating to stakeholders and the market what they can expect is most important. This means having an internal discussion to define these policies.

These challenges were discussed in smaller groups during the workshop in order to formulate solutions. By giving answers or providing "ways out" of these questions, the participants were challenged to think together and develop a solution that is carried by everyone in the organisation.

In the second workshop, we invited several market players reusing Flemish Open Data. As a keynote speaker, we invited CityMapper (http://citymapper.com), which outlined what data they need to create a world-wide multimodal route planner.

2.4. Recommendations for action

The three directives (PSI, INSPIRE and ITS) were often regarded as the reference documents to be implemented. The best-practices for PSI, as put forward by the "Interoperability solutions for public administrations, business and citizens" (ISA^2) programme, focus on Linked Data standards for semantic interoperability. However, the INSPIRE directive for geospatial data, brings forward a national access portal for geospatial data services, in which

datasets are made available through services. There is a metadata effort, called GEODCAT-AP, which brings the metadata from these two worlds together in one Linked Data specification. The ITS directive also puts forward their own specifications, such as NETEX at http://netex-cen.eu/, DATEX2 at http://www.datex2.eu/ and SIRI at http://www.siri.org.uk/. These specifications do not require persistent identifiers, and do not make use of URIS for the data model. We advised the department to first comply to the ISA² best practices, as getting persistent, autodocumented identifiers is the only option today to raise the *semantic interoperability* on web-scale. For datasets that already complied to the INSPIRE or ITS directive, the department would also make these available as data dumps (e.g., as with the roads database).

The Flemish government has *style guidelines* for their websites. We advised to implement extra guidelines for the addition of structured data, e.g., with RDFa. Next, a conclusion from the first workshop was to invest in *guidelines* for the creation of databases. This should ensure each internal and externally communicated dataset is annotated with the right context.

In order to overcome the many organisational challenges, recommendations for action were formulated and accepted by the board of directors:

- Keeping a private data catalogue for all datasets that are created (open and non-open);
- All ICT policy documents need to have references to the Open Data principles outlined in the vision document;
- The department of DTPW is responsible for following up these next steps, and will report to the board of directors;
- Opening up datasets will be part of the roadmap of each sub-organisation within DTPW;
- On fixed moments, there will be meetings with the Agency Information Flanders to discuss the Open Data policy.

Finally, also specific recommendations to data owners, as exemplified in the table, were given.

3. LOCAL DECISIONS AS LINKED OPEN DATA

Probably one of the most ambitious project I have been part of must have been on Local Decisions as Linked Open Data [7]. The core task of a local government is to make decisions and document them for the many years to come. Local governments provide the decisions, or minutes, from these meetings to the Flemish Agency for Domestic Governance as unstructured data. These decisions are the authoritative source for – to name a few – the mandates and roles within the government, the mobility plan, street name changes, or local taxes.

Base registries are trusted authentic information sources controlled by an appointed public administration or organization appointed by the government.

The RSD and the address database – as discussed earlier – are good examples of such base registries. As these examples illustrate, maintaining a base registry comes with extra maintenance costs to create the dataset and keep it up to date. Could we circumvent these extra costs by relying on a decentral way of publishing the authoritative ?

In other countries, we see prototyping happening with the same ideas in mind. OpenRaadsInformatie publishes information from 5 local councils in the Netherlands as Open Data, as well as the OParl project for local councils in Germany. Each of these projects use their own style of JSON API. The data from the municipalities is collected through APIs and by scraping websites and transformed to Linked Open Data. According to the Dutch project's evaluation, the lack of metadata at the source causes a direct impact on the cohesion between the different assets because they can't be interlinked. Next, the W3C Open Gov community group is discussing and preparing an RDF ontology to describe, among others, people, organizations, events, and proposals. Finally, in Flanders, the interoperability program of the Flemish Government, "Open Standards for Linked Organizations" also referred to as OSLO², focuses on the semantic level and extends the ISA CORE Vocabularies to facilitate the integration of the Flemish base registries with one another and their implementation in business processes of both the public and private sector.

We interviewed local governments on how they register and publish Local Council Decisions. We then organized three workshops which formulated the input for a proof of concept: two workshops were organized for creating a preliminary domain model, and one workshop was organized to create wireframes on how Local Council Decisions would be created and searched through in an ideal scenario. The domain concepts were formalized into two Linked Data vocabularies: one for the metadata and one for describing public mandates, formalized in https://lblod.github.io/vocabulary. The proof of concept consists of four components:

1. an editor for local decisions,
2. an HTML page publishing service responsible for URI dereferencing,
3. a crawler for local decisions, and
4. two reuse examples on top of the harvested data.

We introduced a virtual local government called VlaVirGem for which we can publish local decisions. The editor at lblod.github.io/editor is a proof of concept of such an editor, which reuses existing base registries. You can choose to fill out a certain template for decisions that often occur, such as the resignation of a local counselor or the installation of a new one. When filling out the necessary fields, the editor will help you: for example, it will autocomplete people that are currently in office. You will then still be able to edit the official document, which contains more information such as links to legal background, context and motivation, and metadata. When you click the publish button, the decision is published as a plain HTML file on a file host. The URIs are created as hash-URIs from the document's URL.

A harvester is then set up using The DataTank. By configuring a rich snippets harvester, HTML files are parsed and some links are followed to discover the next to be parsed document. The extracted triples are republished for both the raw data as an overview of the mandates. This data is the start of two reuse demos at http://vlavirgem.pieter.pm: the first for generating an automatic list of mandates, and the second is a list of local

decisions.

Although Local Council Decisions contain high quality information in the form of non-structured data, the information in the authoritative source for local mandates today does not. In order to reduce the workload to share this information (e.g., a newly appointed counselor) with other governments or the private sector, the local decision can be published as a Linked Open Data document at the source.

4. CONCLUSION

In the previous chapter, I mentioned three ways to measure interoperability. The first method was to quantify the interoperability by measuring the similarity based on user-input. This method remained conceptual and remains untested in a real-world scenario today. On the basis of the projects that got funding over the next years, I hypothesize that for government data today, more obvious big steps towards raising interoperability can be taken that do not require a quantified semantic intoperability approach.

The second way was to study the effects of a publishing strategy. When more reuse could be noticed in services and applications, the better the balance between the cost for adoption and the benefits will be, and thus the more interoperable a dataset is. As from the transport datasets published on the European data portal, only a limited amount of reuse can be found, for which the high impact datasets still have to be discovered. Also at the DTPW our team interviewed reusers, which would indicate that reuse of governmental datasets at this moment was limited [8].

You can see the current reuse cases at the European data portal: https://www.europeandataportal.eu/en/using-data/use-cases

The third way was to qualitatively study organizations on the basis of the interoperability layers. The first approach for this is through desk research. Again, a quick scan through the European or Flemish data portal reveils that the overall interoperability of these datasets is low. A second approach was to study the datasets qualitatively by means of an interview. This is what our team did at the DTPW, which revealed a list of current issues and

recommendations for actions. The list of current issues could be a useful means for comparing the results of these qualitative studies too.

Finally, we elaborated on a proof of concept built for local council decisions as Linked Data. Council decisions annotated with Linked Data have the benefit of less manual work and that civil servants can search easier through current legislation. Our team also noticed a potential quality gain in editing due to correct legal references (even referencing to decisions of their municipality) and the use of qualitative factual data (e.g., addresses linked to the Central Reference Address Database). Finally, there are also efficiency gains in the publication of the decisions that are automatically published on the website of the local government, in the codex, and without additional efforts suitable for reuse by third parties. The Local Decisions as Linked Data today is a typical example of how the Flemish government can stimulate a decentralized data governance, yet offer centralized tools for local governments that are not able to follow up on European data standards.

In the city of Ghent in Flanders for one, it would take more than 2 months to update all route planning systems when updates happen.

How long would it take a local council decision to get into a route planning system? Still today, someone from a local, regional or national government would need to contact route planning organizations and provide them with the right updates in the format they need. It is an engineering problem to automate the process of data reuse on the various interoperability issues, but also a policy problem to bring a better data culture within a large organization. Today, within the Flemish government, we notice a change towards “assisted decentralization”, in which data systems are architectural decentral, but where the regional government provides services to the underlying governments.

In these three use case, I started from the perspective of specific data publisher and worked my way up in the organization to see what would be needed for a higher interoperable dataset. In the next chapter, we will dive deeper into the field of transport data in specific.

REFERENCES

[1] Colpaert, P., Dimou, A., Vander Sande, M., Breuer, J., Van Compernolle, M., Mannens, E., Mechant, P., Van de Walle, R. (2014). *A three-level data publishing portal*. In proceedings of the European Data Forum.

[2] Colpaert, P., Vander Sande, M., Mannens, E., Van de Walle, R. (2011). *Follow the stars*. In proceedings of Open Government Data Camp 2011.

[3] Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R. (2014). *Painless URI dereferencing using The DataTank*. European Semantic Web Conference (pp. 304–309).

[4] Colpaert, P., Joye, S., Mechant, P., Mannens, E., Van de Walle, R. (2013). *The 5 Stars Of Open Data Portals*. In proceedings of 7th international conference on methodologies, technologies and tools enabling e-Government (pp. 61–67).

[5] Heyvaert, P., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R. (2015, June). *Merging and Enriching DCAT Feeds to Improve Discoverability of Datasets*. Proceedings of the 12th Extended Semantic Web Conference: Posters and Demos (pp. 67–71). Springer.

[6] Van Cauter, L., Bannister, F., Crompvoets, J., Snoeck, M. (2016). *When Innovation Stumbles: Applying Sauer's Failure Model to the Flemish Road Sign Database Project*. IGI Global.

[7] Buyle, R., Colpaert, P., Van Compernolle, M., Mechant, P., Volders, V., Verborgh, R., Mannens, E. (2016, October). *Local Council Decisions as Linked Data: a proof of concept*. In proceedings of the 15th International Semantic Web Conference.

[8] Walravens, N., Van Compernolle, M., Colpaert, P., Mechant, P., Ballon, P., Mannens, E. (2016). *Open Government Data': based Business Models: a market consultation on the relationship with government in the case of mobility and route-planning applications*. In

proceedings of 13th International Joint Conference on e-Business and Telecommunications (pp. 64–71).

Chapter 5
Transport Data

> " Logic will get you from A to B; imagination will take you everywhere. "
> — Anonymous.

Route planning applications can better target end-user needs when they have access to a higher quantity of datasets. The algorithms used by these apps need access to datasets that exceed one database and need query functionality that exceed standard query languages. We rethink the access to data by studying how route planning algorithms work. In this chapter, we describe the state of the art of route planning interfaces. Transport data today is published in or a datadump, or a web-service which answers the entire question on the server-side. If we want to advance the state of the art in sharing data for maximum reuse, we will have to establish a new trade-off between client and server effort.

The way travelers want route planning advice is diverse: from finding journeys that are accessible with a certain disability [1], to taking into account whether the traveler owns a (foldable) bike, a car or a public transit subscription, or even calculating journeys with the nicest pictures on social network sites [2]. However, when presented with a traditional route planning HTTP API taking origin-destination queries, developers of, e.g., traveling tools are left with no flexibility to calculate journeys other than the functions provided by the server. As a consequence, developers that can afford a larger server infrasture, integrate data dumps of the timetables (and their real-time updates), into their own system. This way, they are in full control of the algorithm, allowing them to calculate journeys in their own manner, across data sources from multiple authorities.

For instance, trying to answer the question "how long do I have to walk from one point to another?" can take into account the geolocation of the streets, the weather conditions at that time of the day, the steepness of the road, whether or not there is a sidewalk, criminality reports to check whether it is safe to walk through these streets, the wheelchair accessibility or accessibility for the visual imparaired of the road, whether the street is blocked by works at that time, etc. We can imagine the complexities that arise if the user does not only want to walk, but that he also wants to get advice taking different transport modes into account. An *open world* approach is

needed: a certain dataset should be queried with the assumption that it can only answer part of the question, and that a better answer can always be found by using more datasets. [3]

In this chapter, we first scratch the surface of publishing data for route planning on the road as well, by building a proof of concept in collaboration with the city of Ghent. We then discuss the state of the art in the field of public transit route planning, the focus of this book. Finally, we conclude with opportunities to evolve this state of the art with Linked Connections, described in the next chapter.

1. DATA ON THE ROAD

As we have discussed in the previous chapter, it is difficult to draw the line between transport data and non transport data. Even merely administrative datasets such as the alteration of a streetname, may at some point become useful for a route planning algorithm. We first discuss, within the field of *Intelligent Transport Systems* (ITS), what the constraints should be for an information system, in order to share transport data world-wide.

1.1. Constraints for a large-scale ITS data-sharing system

Within the domain of ITS, the ITS directive helped popularizing publicly sharing data. For example, with a delegated regulation elaborating on a European Access Point for Truck Parking Data, it regulates sharing the data through a national access point similar to the INSPIRE directive, a directive for sharing data within the geo-spatial domain.

When creating a system to distribute data within our own domain – for the many years to come – an important requirement is that this data policy needs to scale up efficiently. When more datasets are added, when more difficult questions need to be answered, when more

questions are asked, or when more user agents come into action, we want our system to work without architectural changes. Furthermore, the system should be able to evolve while maintaining backwards-compatibility, as our organization is always changing. Datasets that are published today, still have to work when accessed when new personnel is in place. Such a system should also have a low entry-barrier, as it needs to be adopted by both developers of user agents as well as data publishers.

1.2. A use case in Ghent

The mobility organization of Ghent was given the task to build a virtual traffic centre. The centre should inform Ghentians with the latest up to date information about mobility in the city. As the city is in a transition, banning cars from the city centre, this is a project with high expectations. Information to build this traffic centre comes from existing geospatial layers, containing the on-street parking zones with, among others, their tariffs, all off-street parking lots, all streets and addresses, and all traffic signs. Also real-time datasets are available, such as the sensor data from the induction loops, bicycle counters, thermal cameras, and the availability of the off-site parking lots. Third parties also contribute datasets, such as the public transit time schedules and their real-time updates, and traffic volumes in, to and leaving the city. In order to bring these datasets from various sources together, and analyse them, both the semantics as the queryability is lacking.

The city of Ghent allowed us to define a couple of URIs for parking sites. In this city, a URI strategy is in place to negotiate identifiers since 2016. The URI strategy defines a base URI at "https://stad.gent/id/". Using this strategy, the city introduced URIs for each parking site, similar to the examples above. When we now would point our browser to https://stad.gent/id/parking/P1, we will be directed to a page about this parking space. Furthermore, this identifier is interoperable across different systems, as when we would GET this URI from a computer program, I can negotiate its content type, and request an RDF representation of choice.

As there are only a couple of parking sites at the city of Ghent, describing this amount of parking sites easily fits into one information resource identified by one URL, for instance, http://linked.open.gent/parking/. When the server does not expose the functionality to filter the parking sites on the basis of geolocation, all user agents that want to solve any question based on the location of parking sites, have to fetch the same resource. This puts the server at ease, as it can prepare the right document once each time the parking sites list is updated. Despite the fact that now all parking sites had to be transferred to the user agent, and thus consumed more bandwidth, also the user agent can benefit. When a similar question is executed from the same device, the dataset will already be present in the user agent's cache, and now, no data at all will need to be transferred. This raises the user-perceived performance of a user interface. When now the number of end-users increases by a factor of thousand per second – not uncommon on the Web – it becomes easier for the server to keep delivering the same file for those user agents that do not have it in cache already. When it is not in the user agents own cache, it might already be in an intermediate cache on the Web, or in the server's cache, resulting in less CPU time per user. Caching, another one of the REST constraints, thus has the potential to eliminate some network interactions and server load. When exploited, a better network efficiency, scalability, and user-perceived performance can be achieved.

Without the CORS header, a resource is by default flagged as potentially containing private data, and cannot be requested by in-browser scripts at a different domain. More information at e nable-cors.org

In a test environment, we published a Linked Data document at http://linked.open.gent/parking/, by transforming the current real-time XML feeds using a PHP script. First, this script adds metadata to this document indicating this data has an open license, and thus becomes legally interoperable with other Web resources. Next, we added HTTP headers, indicating that this document can be used for *Cross Origin Resource Sharing* (CORS), as well as an HTTP header that this document can be cached for 30 seconds. Finally, we also added a content negotiation for different RDF representations.

As we would like to solve *Basic Graph Patterns* (BGP) queries, we added the hypermedia controls requested by the *Triple Pattern Fragments* (TPF) specification, which

details how to filter the triples in this document based on their subject and/or predicate and/or object [4]. As the number of data facts that need to be published is small enough, we directed all controls to the main document, and did not expose extra server functionality.

The latest version of the Triple Pattern Fragments hypermedia specification can be found at https://www.hydra-cg.com/spec/latest/triple-pattern-fragments/.

Following these steps, we made the data queryable through clients that can exploit the Triple Pattern hypermedia controls, such as the Linked Data Fragments client available at http://client.linkeddatafragments.org. This client is able to query over multiple Triple Pattern Fragments interfaces at once, and thus answer federated queries by following hypermedia controls. The following query selects the name, the real-time availability, and the geolocation from Open Street Map (Linked Geo Data) of a parking lot in Ghent with more than 200 spaces available: http://bit.ly/2jUNnES. This demonstrates that complex questions can still be answered over simple interfaces. The overall publishing approach is cost-efficient and stays as close as possible to the HTTP protocol as a uniform interface. The only part where an extra technical investment was needed, was in documenting the definitions of the new URIs for the parking sites.

2. PUBLIC TRANSIT TIME SCHEDULES

Computation of routes within road networks and public transport networks are remarkably different: while some speed up methods work well for the former, they do not for the latter [5]. In this chapter, we will focus on public transit route planning, further on referred to as *route planning*.

A public transit network is considered, in its most essential form, to consist of stops, connections, trips, and a transfer graph [6]:

- A stop *p* is a physical location at which a vehicle can arrive, drop off passengers, pick up passengers and depart;

- A connection *c* represents a vehicle driving from a departure stop at a departure time, without intermediate halt, to an arrival stop at an arrival time;
- A trip is a collection of connections followed by a vehicle;
- A transfer is when a passenger changes a vehicle at the same stop, or a nearby stop with a certain path in between.

A network can also optionally contain routes, which is the collection of trips that follow, to a certain extent, the same series of stops. As the number of routes is in many cases much smaller than the number of trips, the name of these routes are, for simplicity, commonly used to inform passengers. In some route planning applications such as Raptor [7] or Trip-based route planning [8], a clustering algorithm may re-cluster the trips into routes favorable for the algorithm.

For future reference, an arrival and a departure are two concepts defined by a location and a timestamp. A departure at a certain stop, linked to an arrival at a different stop can thus create a connection.

2.1. Route planning queries

We differentiate various types of queries over public transport networks [9]:

- An *Earliest Arrival Time* (EAT) *q* is a route planning question with a time of departure, a departure stop, and an arrival stop, expecting the earliest possible arrival time at the destination;
- The *Minimum Expected Arrival Time* (MEAT) is similar to an EAT, yet it takes into account the probability of delayed or canceled connections;
- Given departure stop, a *profile query* computes for every other stop the set of all earliest arrival journeys, for every departure from the departure stop;

- The *multi-criteria profile* query calculates a set of journeys that each are not outbettered in arrival time or number of transfers, also called a set of Pareto optimal journeys.

In route planning software, the EAT problem is, among others, used for mobility studies, to calculate a latest arrival time for profile queries or for preprocessing to speed up other methods. For providing actual route planning advice, a profile query is what most end-user interfaces expect. The same problem solving algorithm can, with minimal adaptations, be used for the latest departure time problem, which requests the last possible time to leave in order to arrive on time at a certain stop.

2.2. Other queries

Within any of the given problems, a user may need extra personal features, such as the ability to request journeys that are wheelchair accessible, journeys providing the most "happy" route [2], journeys where the user will have a seat [1], or journeys with different transfer settings depending on, e.g., reachability by foldable bike or criminality rates in a station.

Unrelated to the previous problems is that end users also still want answers to other questions without a route planning algorithm to be in place, such as getting a list of the next departures and arrivals at a certain stop, all stops a certain trip contains, or a list of all stops in the area.

2.3. Route planning algorithms

Only in the last decade, algorithms have been built specifically for public transit route planning [10]. Two models were commonly used to represent such a network: a time-expanded model and a time-dependent model [11]. In a *time-expanded model*, a large graph is modeled with arrivals and departures as the nodes, and edges to connect a departure and an arrival together. The weights on these edges are constant. In a *time-dependent model*, a smaller graph is modeled in which vertices are physical

stops and edges are transit connections between them. The weights on these edges change as a function of time. On both models, Dijkstra and Dijkstra-based algorithms can be used to calculate routes [11].

Raptor [7] was the first base algorithm to disregard Dijkstra-like graph algorithms and instead exploit the basic elements of a public transit network. It does this by studying the routes within a network. In each round, it computes the earliest arrival time for journeys with a per round increasing number of transfers. It looks at each route in the network at most once per round. Using simple pruning rules and parallelization with multiple cores, the algorithm can be made even faster. It is currently the algorithm behind software like Open Trip Planner, or Bliksemlabs' RRRR software.

Trip-based route planning [8] uses the same ideas as Raptor and works with an array of trips. In a preprocessing phase, it links trips together when there is a possibility to transfer at a certain stop. During query execution, it can take into account the data generated during the preprocessing phase.

Transfer Patterns [10] preprocesses a timetable such that at execution time, it is known which transfers can be used at a certain departure stop and departure time. The preprocessing requires by design more CPU time and memory than the preprocessing of trip-based route planning. *Scalable Transfer Patterns* [12] enhances these results by reducing both the necessary time and space consumption by an order of magnitude. Within companies that have processing power of idle machines to their disposal, the latter is a desired approach, which makes it the current algorithm behind the Google Maps public transit route planner according to its authors.

Finally, the *Connection Scan Algorithm (CSA)* is an approach for planning that models the timetable data as a directed acyclic graph [9]. By topologically sorting the graph by departure time, the shortest path algorithm only needs to scan through connections within a limited time window. CSA can be extended to solving problems where it also keeps the number of transfers limited, as well as calculating routes with uncertainty. The ideas behind CSA can scale up to large networks by using multi-overlay

networks [6].

It is a challenge to compare route planning algorithms as they are often built for answering slightly different questions and are executed on top of different data models. When studying the state of the art in order to find a suitable data publishing model, we were looking for the smallest building block needed for both preprocessors as actual query execution algorithms (with and without preprocessed data).

2.4. Exchanging route planning data over the Web

The *General Transit Feed Specification* (GTFS) is a framework for exchanging data from a public transit agency to third parties. GTFS, at the time of writing, is the de-facto standard for describing and exchanging of transit schedules. It describes the headers of several CSV files combined in a ZIP-file. Through a *calendar.txt* file, you are able to specify on which days of the week a certain entry from *service.txt* is going to take place during a part of the year. In a *calendar_dates.txt* file, you are able to specify exceptions on these *calendar.txt* rules for example indicating holidays or extra service days. Using these two files, periodic schedules can be described. When an aperiodic schedule needs to be described, mostly only the *calendar_dates.txt* file is used to indicate when a certain service is running. A *gtfs:Service* contains all rules on which a *gtfs:Trip* is taking place, and a *gtfs:Trip* is a periodic repetition of a trip as defined earlier. Trips in GTFS contain multiple *gtfs:StopTimes* and/or *gtfs:Frequencies*. The former – mind the difference with a *connection* – describes a periodic arrival time and a departure time at one certain *gtfs:Stop*. The latter describes the frequency at which a *gtfs:Trip* passes by a certain stop. GTFS also describes the geographic shape of trips, fare zones, accessiblity, information about an agency, and so forth.

Other specifications exist, such as the European CEN specification Transmodel, the Dutch specification BISON, the Belgian specification BLTAC and the specification for describing real-time services *The Service Interface for Real*

Time Information (SIRI). They each specify a serialization specific format for exchanging data, and some specify the questions that should be exposed by a server.

Up to date, route planning solutions exist as services, such as Navitia.io or Plannerstack, in end-user applications such as CityMapper, Ally or Google Maps, or as open source, such as Open Trip Planner or Bliksemlabs RRRR. Other common practices include that an agency, such as a railway company exposing a route planner over HTTP themself. Each of these route planners however have the disadvantage that they do not allow querying with an open world assumption: each response is final and is not supposed to be combined with other response documents.

We have given URIs to the terms in the GTFS specification through the Linked GTFS (base URI: http://vocab.gtfs.org/terms#) vocabulary. The definitions of the above terms can be looked up by replacing *gtfs:* by the base URI.

3. CONCLUSION

If we may conclude anything from our first proof of concept within the city of Ghent, it is that the field of transport data is not much different than Open Data in general. If we want automated reuse of datasets within route planners, we still need to get some minimum requirements accepted in the entire transport domain. These requirements are situated on the same interoperability levels those we were already discussing. For the domain specific parts, we also need to get more specifications to become Web specifications and define URIs for their terms.

We have done this for both DATEX2 as GTFS.

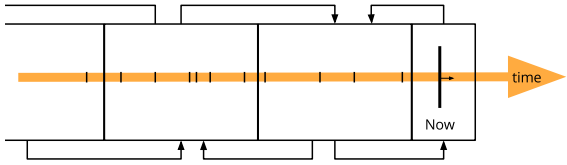

Figure 1: When publishing data in fragments, a generic fragmentation strategy can be thought of for continuously updating data as well.

One challenge which we were able to overcome, is to publish real-time data over HTTP. The approach was again not much different for static documents, only now we need to ensure the latency with the back-end system is minimized, while setting the right caching headers. Real-time data can be put in fragments as well, allowing multiple observations to be stored in a page. This can be annotated by using a named graph and the PROV-O vocabulary. The dataset of parking lots, published as illustrated in Figure 1, is now available at https://linked.open.gent/parking.

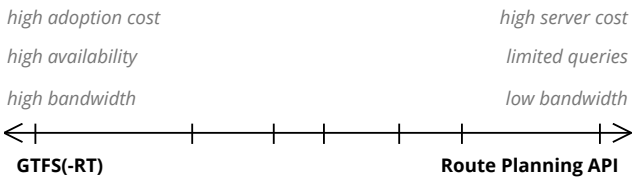

Figure 2: The *Linked Data Fragments* (LDF) axis applied to public transit route planning: on both ends, centralization is key as the client or the server will centralize all data.

Finally, today one can see clear evidence of the two extremes on the LDF axis – applied to route planning in Figure 2 – within public transit timetables. In the next chapter, we will explore different trade-offs on this axis.

REFERENCES

[1] Colpaert, P., Ballieu, S., Verborgh, R., Mannens, E. (2016). *The impact of an extra feature on the scalability of Linked Connections*. In proceedings of ISWC2016.

[2] Quercia, D., Schifanella, R., M. Aiello, L. (2014). *The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city*. In proceedings of The 25th ACM conference on Hypertext and social media (pp. 116–125). ACM.

[3] Colpaert, P. (2014). *Route planning using Linked Open Data*. In proceedings of European Semantic Web Conference (pp. 827–833).

[4] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P. (2016, March). *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. Journal of Web Semantics.

[5] Bast, H. (2009). *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*. (pp. 355–367). Springer Berlin Heidelberg.

[6] Strasser, B., Wagner, D. (2014). *Connection Scan Accelerated*. Proceedings of the Meeting on Algorithm Engineering & Expermiments (pp. 125–137).

[7] Delling, D., Pajor, T., Werneck, R. (2012). *Round-Based Public Transit Routing*. Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12).

[8] Witt, S. (2016). *Trip-Based Public Transit Routing Using Condensed Search Trees*. Computing Research Repository (CORR).

[9] Dibbelt, J., Pajor, T., Strasser, B., Wagner, D. (2013). *Intriguingly Simple and Fast Transit Routing*. Experimental Algorithms (pp. 43–54). Springer.

[10] Bast, H., Carlsson, E., Eigenwillig, A., Geisberger, R., Harrelson, C., Raychev, V., Viger, F. (2010). *Fast routing in very large public transportation networks using*

transfer patterns. Algorithms – ESA 2010 (pp. 290–301). Springer.

[11] Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C. (2008). *Efficient models for timetable information in public transportation systems*. Journal of Experimental Algorithmics (JEA) (pp. 2–4). ACM.

[12] Bast, H., Hertel, M., Storandt, S. (2016). *Scalable Transfer Patterns*. In proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX). Society for Industrial and Applied Mathematics.

[13] Corsar, D., Markovic, M., Edwards, P., Nelson, J.D. (2015). *The Transport Disruption Ontology*. In proceedings of the International Semantic Web Conference 2015 (pp. 329–336).

Chapter 6
Public Transit Route Planning Over Lightweight Linked Data Interfaces

> "The impact of a feature on a Web API should be measured across implementations: measurable evidence about features should steer the API design decision process."
> — Principle 5 of "A Web API ecosystem through feature-based reuse" by Ruben Verborgh and Michel Dumontier.

End-users want to plan public transit journeys based on parameters that exceed a data publisher's imagination. In the previous chapter we have learned that today, on the one hand, data publishers provide, rather expensive to host, route planning APIs, which do not allow a reuser to add functionality to the algorithm. On the other hand, the open data initiative advocates for data publishers to also provide data dumps with real-time changesets, which however is cost intensive to integrate. We want to enable reusers to create route planners with different requirements, as well as enable public transit agencies to publish the data for unlimited reuse. In order to establish a new trade-off on the Linked Data Fragments axis, Linked Connections introduces a light-weight data interface, over which the base route planning algorithm "Connections Scan" [1] can be implemented on the client-side. In this chapter, we report on testing query execution time and CPU usage of three set-ups that solve the Earliest Arrival Time (EAT) problem using query mixes within Belgium: route planning on the server-side, route planning on the client-side without caching, and route planning on the client-side with caching. Furthermore, we study how and where new features can be added to the Linked Connections framework by studying the feature of wheelchair accessibility.

When publishing data for maximum reuse, HTTP caching can make sure that through a standard adopted by various clients, servers, as well as intermediary and neighborhood caches [2], can be exploited to reach the user-perceived performance necessary, as well as the scalability/cost-efficiency of the publishing infrastructure itself. For that purpose, we can take inspiration from the *locality of reference principle*. Time schedules are particularly interesting from both the geospatial perspective as the time locality. Can we come up with a fragmentation strategy for time schedules?

When publishing departure-arrival pairs (*connections*) in chronologically ordered pages – as demonstrated in our 2015 demo paper [3] – route planning can be executed at data integration time by the user agent rather than by the

data publishing infrastructure. This way, *Linked Connections* (LC) allows for a richer web publishing and querying ecosystem within public transit route planners. It lowers the cost for reusers to start prototyping – also federated and multimodal – route planners over multiple data sources. Furthermore, other sources may give more information about a certain specific vehicle that may be of interest to this user agent. It is not exceptional that route planners also take into account properties such as fares [4], wheelchair accessibility [5], or criminality statistics. In this chapter, we test our hypothesis that this way of publishing is more lightweight: how does our information system scale under more data reuse compared to origin-destination APIs?

This rest of the chapter is structured in a traditional academic way, first introducing more details on the Linked Connections framework, to then describe the evaluation's design, made entirely reproducible, and discuss our open-source implementation. Finally, we elaborate on the results and conclusion.

1. THE LINKED CONNECTIONS FRAMEWORK

The time performance of CSA, discussed in the previous chapter, is $O(n)$ with n the number of input connections. These input connections are a subset of the set of connections in the real world. The smaller this subset, the lower the time consumption of the algorithm will be. Lowering the number of connections can be done using various methods involving preprocessing, end-user preferences, multi-overlay networks, geo-filtering, or other heuristics. The CSA algorithm allows for an Open World Assumption while querying. The input connections that are fed in the algorithm are not all connections in existence, and when fed with other input connections, it may find a better journey. Furthermore, the stops and trips are discovered while querying: there is no need to keep a list of all trips and stops in existence. Thanks to the locality,

extensibility and "open world" properties, three properties favorable for a Web environment, CSA was chosen as the base algorithm for our framework.

```
IN: Ordered list of connections C, query q

arrivalTimes[q.departureStop] ← q.departureTime;
arrivalTimes[q.destinationStop] ← ∞;
c ← C.next();
while (c.departureTime <= arrivalTimes[q.destinationStop]) {
  if (arrivalTimes[c.departureStop] < c.departureTime
   AND arrivalTimes[c.arrivalStop] > c.arrivalTime) {
    arrivalTimes[c.arrivalStop] ← c.arrivalTime;
    minimumSpanningTree[c.arrivalStop] ← c;
  }
  c ← C.next();
}
return reconstructRoute(minimumSpanningTree, q);
```

Code snippet 1: Pseudo code solving the earliest arrival time problem using the Connection Scan Algorithm

Also preprocessing algorithms [6] scan through an ordered list of connections to, for example, find transfer patterns [7].

To solve the EAT problem using CSA, timetable information within a certain time window needs to be retrievable. Other route planning questions need to select data within the same time window, and solving these is expected to have similar results Instead of exposing an origin-destination API or a data dump, a Linked Connections server paginates the list of connections in departure time intervals and publishes these pages over HTTP. A public transit timetable, in the case of the base algorithm CSA, is represented by a list of connections, ordered by departure time. Each page contains a link to the next and previous one. In order to find the first page a route planning needs, the document of the entry point given to the client contains a hypermedia description on how to discover a certain departure time. Both hypermedia controls are expressed using the Hydra vocabulary.

http://www.hydra-cg.com/spec/latest/core/

The base entities that we need to describe are connections, which we documented using the LC Linked Data vocabulary (base URI: http://semweb.mmlab.be/linkedco

nnections#). Each connection entity – the smallest building block of time schedules – provides links to an arrival stop and a departure stop, and optionally to a trip. It contains two literals: a departure time and an arrival time. Linked GTFS can be used to extend a connection with public transit specific properties such as a headsign, drop off type, or fare information. Furthermore, it also contains concepts to describe transfers and their minimum change time. For instance, when transfering from one railway platform to another, Linked GTFS can indicate that the minimum change time from one stop to another is a certain number of seconds.

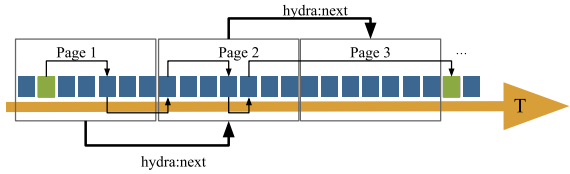

Figure 1: An LC server fragments and publishes a long list of connections, ordered by departure time.

I implemented this hypermedia control as a redirect from the entry point to the page containing connections for the current time. Then, on every page, the description can be found of how to get to a page describing another time range. In order to limit the amount of possible documents, we only enable pages for each X minutes, and do not allow pages describing overlapping time intervals. When a time interval is requested for which a page does not exist, the user agent will be redirected to a page containing connections departing at the requested time. The same page also describes how to get to the next or previous page. This way, the client can be certain about which page to ask next, instead of constructing a new query for a new time interval.

X is a configurable amount

```
{
  "@context":{
    "lc":"http://semweb.mmlab.be/ns/linkedconnections#",
    "hydra":"http://www.w3.org/ns/hydra/core#",
    "gtfs":"http://vocab.gtfs.org/terms#",
    "cc" : "http://creativecommons.org/ns#"
```

```
  },
  "@id":"http://{host}/2017/apr/2",
  "@type":"hydra:PagedCollection",
  "cc:license":
      "http://creativecommons.org/publicdomain/zero/1.0/",
  "hydra:next":"http://{host}/2017/apr/3",
  "hydra:previous":"http://{host}/2017/apr/1",
  "hydra:search":{
    "@type":"hydra:IriTemplate",
    "hydra:template":"http://{host}/{?departureTime}",
    "hydra:variableRepresentation":
        "hydra:BasicRepresentation",
    "hydra:mapping":{
      "@type":"IriTemplateMapping",
      "hydra:variable":"departureTime",
      "hydra:required":true,
      "hydra:property":"lc:departureTimeQuery"
    }
  },
  "@graph":[
    {
      "@id":"http://{host}/a24fda19",
      "@type":"lc:Connection",
      "lc:departureStop":"http://{host}/stp/1",
      "lc:departureTime":"2017-04-02T12:00:00Z",
      "lc:arrivalStop":"http://{host}/stp/2",
      "lc:arrivalTime":"2017-04-02T12:30:00Z",
      "gtfs:trip":"http://{host}/trips/2017/1"
    },
    …
  ]
}
```

Code snippet 2: Example of a Linked Connections server response in JSON-LD. The context is, according the the JSON-LD specification, the part where terms are mapped to URIs. After that, the hypermedia section follows, where the current page is linked to the next page and previous page. Furthermore, it is also described how to query for something starting at a certain departure time. The remainder of the file provides an example of a connection. It describes both a departure and an arrival, and it is given a unique identifier. Furthermore, it is extended with a trip id from the GTFS vocabulary.

A naive implementation of federated route planning on top of this interface is also provided, as a client can be configured with more than one entry point. The client then performs the same procedure multiple times in parallel. The "connections streams" are merged and sorted as they are downloaded. A Linked Data solution is to ensure that the client knows how to link a stop from one agency to a stop from another.

2. EVALUATION DESIGN

We implemented different components in JavaScript for the Node.js platform. We chose JavaScript as it allows us to use both components both on a command-line environment as well as in the browser.

csa.js
: A library that calculates a minimum spanning tree and a journey, given a stream of input connections and a query

Server.js
: Publishes streams of connections in JSON-LD, using a MongoDB to retrieve the connections itself

Client.js
: Downloads connections from a configurable set of servers and executes queries

gtfs2lc
: A tool to convert existing timetables as open data to the Linked Connections vocabulary

The Belgian railway company estimates delays on its network each minute.

We set up 2 different servers, each connected to the same *MongoDB* database which stores all connections of the Belgian railway system for October 2015:

1. A *Linked Connections server* with an NGINX proxy cache in front, which adds caching headers configured to cache each resource for one minute and compresses the body of the response using gzip;

2. A *route planning server* which exposes an origin-destination route planning interface using the csa.js library. The route planning server uses the csa.js library to expose a route planning API on top of data stored in a MongoDB. The code is available at https://github.com/linkedconnections/query-server.

In the *LC without cache set-up*, we introduce a set-up where the LC server is used as a data source by the end-user's machine. A demo of an implementation in a browser can be viewed at http://linkedconnections.org, where the client.js was used as a library in the browser. The *LC with cache set-up* is where one user agent is doing all the querying on behalf of all end-users. Finally, with the *traditional origin-destination approach*, a data maintainer publishes a route planning service over HTTP where one HTTP request equals one route planning question, developed to be able to compare the previous two set-ups.

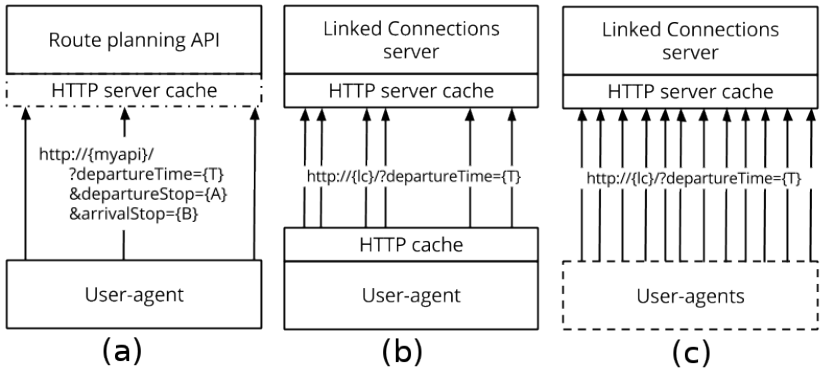

Figure 2: (a) Client-side route planning; (b) Client-side route planning with client-side cache; (c) Server-side route planning

These tools are combined into different set-ups:

Client-side route planning

The first experiment executes the query mixes by using the Linked Connections client. Client caching is disabled, making this simulate the *LC without cache* set-up, where every request could originate from an end-user's device.

Client-side route planning with client-side cache
: The second experiment does the same as the first experiment, except that it uses a client side cache, and simulates the *LC with cache* set-up.

Server-side route planning
: The third experiment launches the same queries against the full route planning server. The query server code is used which relies on the same csa.js library as the client used in the previous two experiments.

In order to have an upper and lower bound of a real world scenario, the first approach assumes every request comes from a unique user agent which cannot share a cache, and has caching disabled, while the second approach assumes one user agent does all requests for all end-users and has caching enabled. Different real world caching opportunities – such as user agents for multiple end-users in a software as a service model, or shared peer to peer neighborhood caching [2] – will result in a scalability in-between these two scenarios.

We also need a good set of queries to test our three set-ups with. The iRail project provides a route planning API to apps with over 100k installations on Android and iPhone [8]. The API receives up to 400k route planning queries per month. As the query logs of the iRail project are published as open data, we are able to create real query mixes from them with different loads. The first mix contains half the query load of iRail, during 15 minutes on a peak hour on the first of October. The mix is generated by taking the normal query load of iRail during the same 15 minutes, randomly ordering them, and taking the half of all lines. Our second mix is the normal query load of iRail, which we selected out of the query logs, and for each query, we calculated on which second after the benchmark script starts, the query should be executed. The third mix is double the load of the second mix, by taking the next 15 minutes, and subtracting 15 minutes from the requested departure time, and merging it with the second mix. The same approach can be applied with limited changes for the subsequent query mixes, which are taken from the next days' rush hours. Our last query mix is 16 times the

query load of iRail on the 1st of October 2015. The resulting query mixes can be found at https://github.com/linkedconnections/benchmark-belgianrail.

These query mixes are then used to reenact a real-world query load for three different experiments on the three different architectures. We ran the experiments on a quad core Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz with 8GB of RAM. We launched the components in a single thread as our goal is to see how fast CPU usage increases when trying to answer more queries. The results will thus not reflect the full capacity of this machine, as in production, one would run the applications on different worker threads.

Our experiments can be reproduced using the code at https://github.com/linkedconnections/benchmark-belgianrail. There are three metrics we gather with these scripts: the CPU time used of the server instance (HTTP caches excluded), the bandwidth used per connection, and the query execution time per LC connection. For the latter two, we use a per-connection result to remove the influence of route complexity, as the time complexity of our algorithm is $O(n)$ with n the total number of connections. We thus study the average bandwidth and query execution time needed to process one connection per route planning task.

3. RESULTS OF THE OVERALL COST-EFFICIENCY TEST

Figure 3 depicts the percentage of the time over 15 minutes the server thread was active on a CPU. The server CPU time was measured with the command pidstat from the sysstat package and it indicates the server load. The faster this load increases, the quicker extra CPUs would be needed to answer more queries.

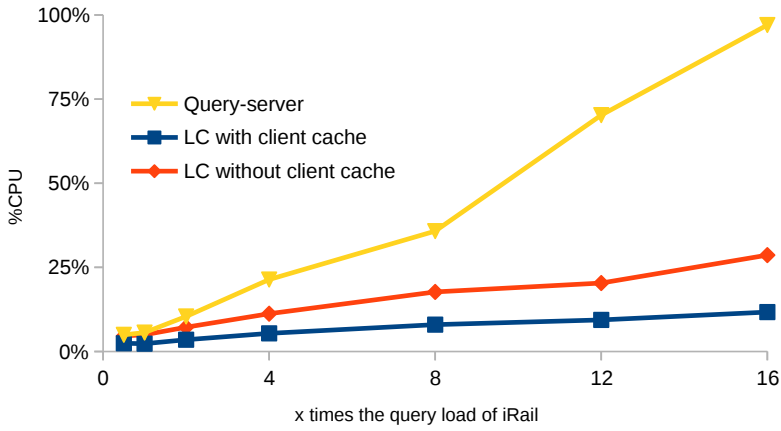

Figure 3: CPU time consumed by the server in 3 different scenarios shows that Linked Connections, even without caching, is more light-weight than the traditional approach.

When the query load is half of the real iRail load on October 1st 2015, we can see the lowest server load is the LC set-up with client cache. About double of the load is needed by the LC without client cache set-up, and even more is needed by the query server. When doubling the query load, we can notice the load lowers slightly for LC with cache, and the other two raise slightly. Continuing this until 16 times the iRail query load, we can see that the load of the query server raises until almost 100%, while LC without cache raises until 30%, while with client cache, 12% is the measured server load.

Table 5: Server CPU usage under increasing query load shows that the Linked Connections server is more cost-efficient: more queries can be answered on one core.

query load	LC no cache	LC with cache	Query server
0.5	4.61%	2.44%	4.95%
1	4.97%	2.32%	5.62%
2	7.21%	3.50%	10.41%
4	11.23%	5.39%	21.37%
8	17.69%	7.98%	35.77%
12	20.34%	9.38%	70.24%
16	28.63%	11.71%	97.01%

In Figure 4, we can see the query response time per connection of 90% of the queries. When the query load is half of the real iRail load on October 1st 2015, we notice that the fastest solution is the query server, followed by the LC with cache. When doubling the query load, the average query execution time is lower in all cases, resulting in the same ranking. When doubling the query load once more, we see that LC with client cache is now the fastest solution. When doubling the query load 12 times, also the LC without client cache becomes faster. The trend continues until the the query server takes remarkably longer to answer 90% of the queries than the Linked Connections solutions at 16 times the query load.

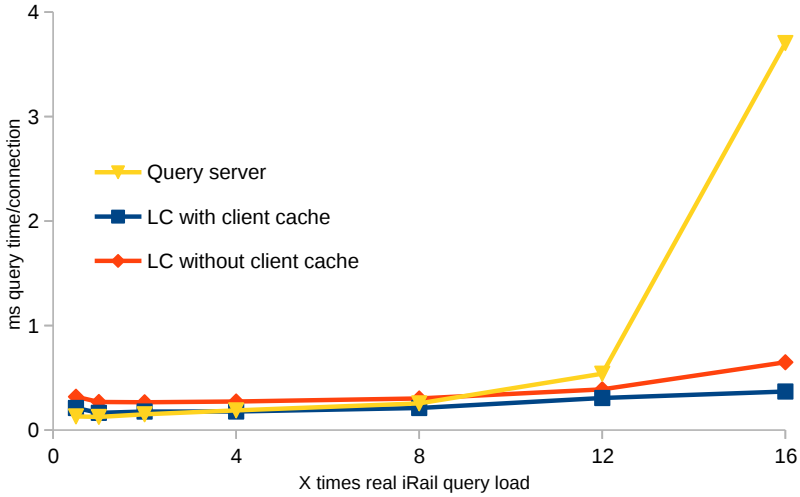

Figure 4: This figure shows the average of the response times divided by the number of connections that were needed to be processed.

The average bandwidth consumption per connection in bytes shows the price of the decreased server load as the bandwidth consumption of the LC solutions are three orders of magnitude bigger: LC is 270B, LC with cache is 64B and query-server is 0.8B. The query server only gives one response per route planning question which is small. LC without client cache has a bandwidth that is three orders of magnitude bigger than the query server. The LC with a client cache has an average bandwidth consumption per connection that is remarkably lower. On the basis of these numbers we may conclude the average cache hit-rate is about 78%.

Table 6: 90% of the journeys will be found with the given time in ms per connection under increasing query load.

query load	LC no cache	LC with cache	Query server
0.5	0.319	0.211	0.130
1	0.269	0.165	0.125
2	0.266	0.177	0.151
4	0.273	0.177	0.188
8	0.302	0.211	0.255
12	0.389	0.307	0.539
16	0.649	0.369	3.703

4. LINKED CONNECTIONS WITH WHEELCHAIR ACCESSIBILITY

In order to study how to add extra features to this framework, we performed another experiment, in order to test the impact of this extra feature on both the client and server. A wheelchair accessible journey has two important requirements: first, all vehicles used for the route should be wheelchair accessible. Thus, the trip (a sequence of stops that are served by the same vehicle), should have a wheelchair accessible flag set to true when there is room to host a wheelchair on board. Secondly, every transfer stop, the stop where a person needs to change from one vehicle to another vehicle, should be adapted for people with limited mobility. As the LC server does not know where the traveler is going to hop on as it only publishes the time tables in pages, it will not be able to filter on wheelchair accessible stops.

When extending the framework of Linked Connections with a wheelchair accessibility feature, there are two possible ways of implementing this:

1. Filter both the wheelchair accessible trips and stops on the client;

2. Only filter the wheelchair accessible trips on the server while filtering the stops on the client.

4.1. Linked Connections with filtering on the client

In a first approach, the server does not expose wheelchair accessibility information and only exposes the Linked Data of the train schedules using the Linked Connections vocabulary. The client still calculates wheelchair accessible journeys on the basis of its own sources. We thus do not extend the server: the same server interface is used as in LC without wheelchair accessibility. The LC client however is extended with two filter steps: the first filter removes all connections from the connections stream whose trip is not wheelchair accessible. The second filter is added to CSA, to filter the transfers stops. When CSA adds a new connection to the minimum spanning tree, it detects whether this would lead to a transfer. CSA can use the information from stops in the Linked Open Data cloud to decide if the transfer can be made and the connection can be added to the minimum spanning tree. To be able to support dynamic transfers times, CSA can also request Linked Data on "transfers" from another source. When a transfer is detected, CSA will add the transfer time to the departure time of the connection. The trips, stops, and transfers in our implementation are simple JSON-LD documents. They are connected to a module called the *data fetcher* that takes care of the caching and pre-fetching of the data.

This solution provides an example of how the client can now calculate routes using more data than the data published by the transit companies. As wheelchair accessibility is specified in the GTFS standard, we can expect that the public transit companies provide this data. In practice we have noticed that the data is mostly not available, which can be considered normal for an optional field. This makes us believe an external organisation that represents the interests of the less mobile people should be able to publish this data.

4.2. Linked Connections with filtering on the server and client

In the Linked Connections solution with filtering on both server and client-side, the server still publishes the time schedules as Linked Data, yet an extra hypermedia control is added to the LC server to enable the trips filter on the resulting connections. The wheelchair accessibility information is directly added to the servers' database, and an index is configured, so that the LC server can query for this when asked.

5. WHEELCHAIR ACCESSIBILITY FEATURE EXPERIMENT

5.1. Evaluation design

The purpose of the evaluation is to compare the two LC solutions based on the scalability of the server-interface, the query execution time of an EAT query, and the CPU time used by the client. For this evaluation, we used the same query mixes as in the previous experiment.

Two LC servers, one for the LC with filtering on the client setup and one for the LC with trips filtering on the server setup. The MongoDBs used by the LC servers are populated with connections of the Belgian railway company of 2015. Also a *transfers, trips and stops resource* as a data source for the wheelchair accessibility information is created.

5.2. Results

Figure 5 contains the CPU load of the server. The server interface with filtering on the server has a similar scalability as without the filter functionality. However, an average raise of 1.24% in processing power needed, can be noticed.

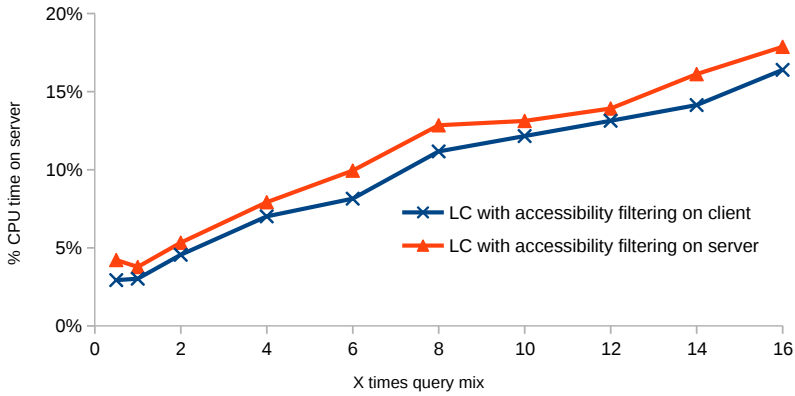

Figure 5: Server CPU load under increasing query load shows that wheelchair accessibility filtering on the server takes more effort for the server under all query loads.

In Figure 6, we can see the CPU load of the client performing the algorithm. When the query load is half of the real iRail load, we notice that the load is 20% for filtering on the client and 21% for trips filtering on the server. Continuously increasing the query load results in an increased client CPU load. The client load of the solution with filtering on the client increases from 27% at query mix 1 to 93% at query-mix 16 while the solution with filtering on the server-side increases from respectively 30% to 93%. For query loads higher than 12 times the iRail load the difference between the two solution becomes less than 1%.

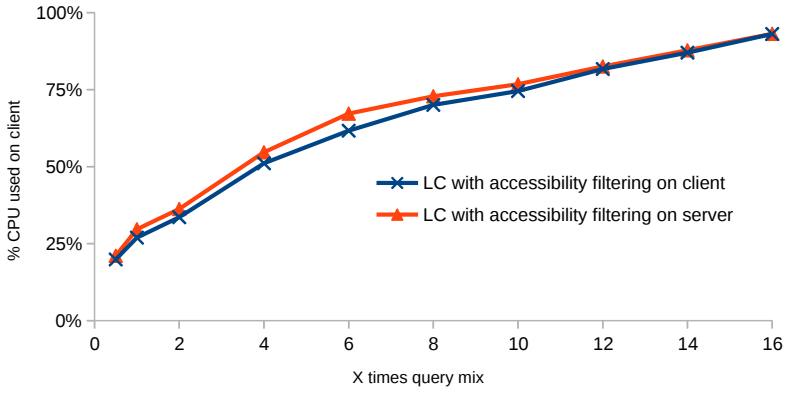

Figure 6: Client CPU load is higher under increasing query load for the LC setup with filtering on both the client and server-side than only on the client-side.

Figure 7 shows the average query response time per connection. We notice that the fastest solution is the solution with filtering on the client for lower query loads, yet for higher query loads, the execution times become comparable.

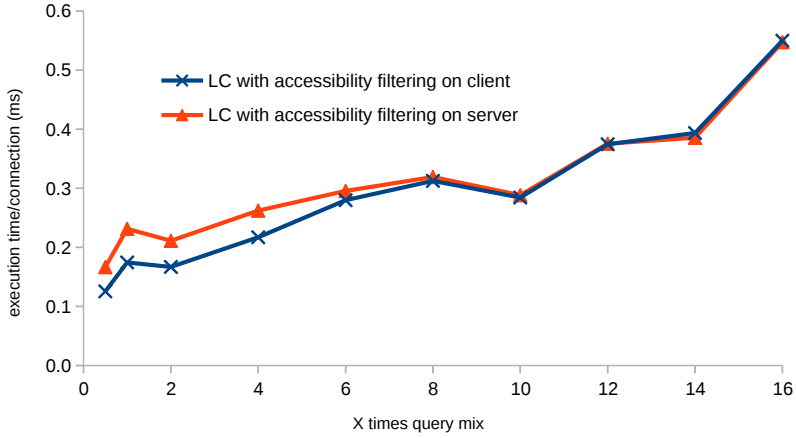

Figure 7: The average time needed to process one connection in milliseconds shows that for the normal query loads the Linked Connections solution with trip filtering on the server is on average slower than the solution with filtering only on the client. When the query load is 8 times the normal query load, we can no longer notice a difference.

When observing half of the normal iRail query load the measured cache rate is 76% and 70% for respectively with filtering on the client as with trip filtering on the server. The cache hit rate slowly decreases to respectively 70% and 64% at query mix 8 and finally to 64% and 61% at the highest query mix 16. When observing all query loads, the cache hit rate measured from the Linked Connections framework with filtering on the client is lower than the framework with filtering on both client and server-side.

When comparing the two LC implementations we can observe that moving the trips filter from the client to the server-side does not cause an improvement of the query execution time. The cache performance is better for the client-side filtering solution because the hybrid solution needs an extra parameter to query the LC server. This results in more unique requests and consequently in a lower cache hit rate. The loss of cache performance increases the number of requests that the LC server needs to handle, which leads to a higher CPU load at the server-side and an increased query execution time. The difference becomes smaller as the query load increases, as more queries can use the already cached Linked Connections documents.

6. DISCUSSION

The results look promising: thanks to a cache hit-rate of 78%, the Linked Connections server is able to handle more queries than the traditional query server approach.

6.1. Network latency

In the evaluation, we tested two extremes: one where the user agent can cache nothing, being an end-user application, and another, where the user agent can cache all end-user questions. The latter can be an intermediate traditional API with a back-end that uses something similar to the Linked Connections client. The cost-efficient interfaces of Linked Connections can this way also be used as a new way to exchange public transit data, while the

end-users still get small responses.

6.2. Actual query response times

The average number of connections in the journeys of the results, multiplied with the query execution time per connection will give an idea of the average waiting time per route planning query. For the Belgian railway system, based on the iRail query logs, we scan an average of 2700 connections per query. For the entire Belgian railway system as input connections, route planning queries will on average, even in the case of 16 times the iRail query load, return results under 2 seconds, again network latency not taken into account.

6.3. More advanced federated route planning

The current approach is simple: when multiple entrypoints are configured, data will be downloaded by following links in both entrypoints. The stream of connections that comes in will be ordered just in time, and provided to the CSA algorithm. Further research and engineering is needed to see how sources would be able to be selected just in time. E.g., when planning a route from one city to another, a planning algorithm may first prioritize a bus route to border stations in the first city, to then only query the overarching railway network, to then only query the bus company in the destination city.

6.4. Denser public transport networks

The number of connections in a densily populated area like Paris or London are indeed much higher than in other areas. In order to scale this up to denser areas, the key would be to split the dataset in smaller regions. This would allow for federated querying techniques to prune the to be queried parts of the Linked Connections datasets. This fragmentation strategy could also allow for parallel processing, limiting the time the client is waiting for data to be downloaded.

A generic approach for fragmentation strategies for

data in ordered ranges has been put in the multidimensional interfaces ontology [9], which takes inspiration from B-trees and R-trees. We leave it as a challenge to future research to apply concepts like multi-overlay networks [1] to Linked Connections.

6.5. Real-time updates: accounting for delays

As the LC pages in the proof of concept are generated from a Mongodb store, updating a departure time or arrival time will result in the new data being queryable online. However, user agents querying pages may experience problems when connections suddenly shift pages. One possible solution to this is to use Memento on top of the HTTP server, making the LC client able to query a fixed version. Another solution would be to keep the updated connection in all pages. When the algorithm detects a duplicate connection, it can take the last version of the object or choose to restart the query execution.

6.6. Beyond the EAT query

The basic CSA algorithm solving the EAT problem can be extended to various other route planning questions for which we refer to the paper in which CSA was introduced [1]. Other queries may for instance include Minimum Expected Arrival Times, a set of Pareto optimal routes [1], isochrone studies, or analytical queries studying connections with a certain property. For all of these queries, it is never necessary to download the entire dataset: only the window of time that is interesting to the query can be used.

6.7. On disk space consumption

GTFS contains rules, while the Linked Connections model contains the result of these rules. In the current implementation, the result of a 2MB GTFS can easily be 2GB of Linked Connections. However, on the one hand, we can come up with systems that generate Linked Connections pages from a different kind of rules

dynamically. On the other hand, keeping an identifier on disk for each connection that ever happened and is planned to happen, is an interesting idea for analytical purposes. E.g., in Belgium, iRail is now using the Linked Connections model to keep a history of train delays and an occupancy score [8] and execute analytical queries over it.

6.8. Non-measurable benefits

While bandwidth, CPU load, and query execution times are measurable, there are also other design aspects to take into account, which are not directly measurable. For instance, in the case where end-user machines execute the route planning algorithm, privacy is engineered by design: a man in the middle would not be able to determine from where to where the end-user is going. As the client is now in control of the algorithm, we can now give personalized weights to connections based on subscriptions the end-user has or we can calculate different transfer times based on how fast you can walk, without this information having to be shared with the public transit agencies.

7. CONCLUSION

We measured and compared the raise in query execution time and CPU usage between the traditional approach and Linked Connections. We achieved a better *cost-efficiency*: when the query-interface becomes saturated under an increasing query load, the lightweight LC interface only reached 1/4th of its capacity, meaning that the same load can be served with a smaller machine, or that a larger amount of queries can be solved using the same server. As the server load increases, the LC solution even – counter-intuitively – gives faster query results.

These result are strong arguments in favor of publishing timetable data in cacheable fragments instead of exposing origin-destination query interfaces when publishing data for maximum reuse is envisioned. The price of this decreased server load is however paid by the bandwidth that is needed, which is three orders of

magnitude bigger. When route planning advice needs to be calculated while on, for example, a mobile phone network, network latency, which was not taken into account during the tests, may become a problem when the cache of the device is empty. An application's server can however be configured with a private origin-destination API, which in its turn is a consumer of a Linked Connections dataset, taking the best of both worlds. When exposing a more expressive server interface, caution is advised. For instance in the case of a wheelchair accessibility feature, the information system would become less efficient when exposing this on the server interface.

Our goal was to enable a more flexible public transport route planning ecosystem. While even personalized routing is now possible, we also lowered the cost of hosting the data, and enabled in-browser scripts to execute the public transit routing algorithm. Furthermore, the query execution times of queries solved by the Linked Connections framework are competitive. Until now, public transit route planning was a specialized domain where all processing happened in memory on one machine. We hope that this is a start for a new ecosystem of public transit route planners.

REFERENCES

[1] Strasser, B., Wagner, D. (2014). *Connection Scan Accelerated*. Proceedings of the Meeting on Algorithm Engineering & Expermiments (pp. 125–137).

[2] Folz, P., Skaf-Molli, H., Molli, P. (2016). *CyCLaDEs: A Decentralized Cache for Triple Pattern Fragments*. The Semantic Web. Latest Advances and New Domains: 13th International Conference (pp. 455–469). Springer International Publishing.

[3] Colpaert, P., Llaves, A., Verborgh, R., Corcho, O., Mannens, E., Van de Walle, R. (2015). *Intermodal Public Transit Routing using Linked Connections*. In proceedings of International Semantic Web

Conference (Posters & Demos).

[4] Delling, D., Pajor, T., Werneck, R. (2012). *Round-Based Public Transit Routing*. Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12).

[5] Colpaert, P., Ballieu, S., Verborgh, R., Mannens, E. (2016). *The impact of an extra feature on the scalability of Linked Connections*. In proceedings of ISWC2016.

[6] Witt, S. (2016). *Trip-Based Public Transit Routing Using Condensed Search Trees*. Computing Research Repository (CORR).

[7] Bast, H., Carlsson, E., Eigenwillig, A., Geisberger, R., Harrelson, C., Raychev, V., Viger, F. (2010). *Fast routing in very large public transportation networks using transfer patterns*. Algorithms – ESA 2010 (pp. 290–301). Springer.

[8] Colpaert, P., Chua, A., Verborgh, R., Mannens, E., Van de Walle, R., Vande Moere, A. (2016, April). *What public transit API logs tell us about travel flows*. In proceedings of the 25th International Conference Companion on World Wide Web (pp. 873–878).

[9] Taelman, R., Colpaert, P., Verborgh, R., Mannens, E. (2016, August). *Multidimensional Interfaces for Selecting Data within Ordinal Ranges*. Proceedings of the 7th International Workshop on Consuming Linked Data.

Chapter 7
Conclusion

> "The future is so bright, we will have to wear shades."
> — Erik Mannens.

Lowering the cost to reuse a dataset can be done by raising its interoperability to other datasets. A developer then has the possibility to automate reuse of more datasets when a user-agent can recognize common elements. In this dissertation, I studied how this data source interoperability of Open (Transport) Datasets can be raised. I introduced five layers of data source interoperability, that can create a framework to solve these questions: legal, technical, syntactic, semantic, and querying. On the one hand, these five layers were used for qualitative research studying public administrations in Flanders. On the other hand, they were used to design a public transport data publishing framework called Linked Connections (LC). With LC, I researched a new trade-off for publishing public transport data by evaluating the cost-efficiency. The trade-off chosen allows for flexibility on the client-side with a user-perceived performance that is comparable to the state of the art. When publishing data in any domain – or when creating standards for publishing data – a similar exercise should be made. I summarize the key take aways in 8 minimum requirements for designing your next HTTP Open Data publishing interface.

The research question as discussed in Chapter 1 was "How can the data source interoperability of Open (Transport) Data be raised?". In Chapter 2, I introduced the theoretical framework to study publishing data for maximum reuse in five data source interoperability layers. Chapter 3 then elaborated on how we can – and whether we should – measure the data source interoperability, based on these layers, and discussed three possible approaches. In Chapter 4, I applied this to three projects, which have been carried out through the course of this PhD, and reasoned that qualitatively studying interoperability would work best at that time for studying maximizing reuse at the Flemish government. In Chapter 5, I then introduced the specifics of data in the transport domain, which sketched the current state of the art. Finally, Chapter 6 introduced the Linked Connections framework, in which the conclusion contains strong arguments – supported by the evaluation – in favor of

Linked Connections for publishing public transport data.

In the upcoming HTTP/2.0 standard, all Web-communication need to be secure (HTTPS). Using the HTTP protocol today thus implies using the secured HTTPS to identify Web-resources. A redirect from a HTTP URL to a HTTPS URL still enables older identifiers to persist.

Lowering the cost for adoption for public datasets is a complex cross-cutting concern. Different parties across different organizations need to – just to name a few – align their vision on Open Data, need to create and accept domain models, need to agree upon legal conditions need to be agreed upon, and need to pick a Linked Data interface to make their data queryable. To that extent, we need to identify the minimum requirements that would lower the cost for adoption across the entire information system on all interoperability levels. In order to achieve a better Web ecosystem for sharing data in general, I summarized a minimum set of extra requirements when using the HTTP protocol to publish Open Data.

1. Fragment your datasets and publish the documents over HTTP. The way the fragments are chosen depends on the domain model.
2. When you want to enable faster query answering, provide aggregated documents with appropriate links (useful for, e.g., time series), or expose more fragments on the server-side.
3. For scalability, add caching headers to each document.
4. For discoverability, add hypermedia descriptions in the document.
5. A web address (URI) per object you describe, as well as HTTP URIs for the domain model. This way, each data element is documented and there will be a framework to raise the semantic interoperability.
6. For the legal interoperability, add a link to a machine readable open license in the document.
7. Add a Cross Origin Resource Sharing HTTP header, enabling access from pages hosted on different origins.
8. Finally, provide DCAT-AP metadata for discoverability in Open Data Portals.

This approach does of course not limit itself to static data. The HTTP protocol allows for caching resources for smaller amounts of time. Even when a document may

change every couple of seconds, the resource can still be cached during that period of time, and a maximum load on a back-end system can be calculated.

While economists promise a positive economic impact from Open Data, I did not yet see proof of this impact – to the extent promised – today. Over the next years, we will need to focus on lowering the cost for adoption if we want to see true economic impact. For data publishers, this entails raising the data source interoperability. For data reusers, this entails automating their clients to reuse these public datasets: when a new dataset becomes available and discoverable, this user agent can automatically benefit from this dataset. Today, this is a manual process, where the user agent has to store and integrate all data locally first, or where the user agent has to rely merely on the expressiveness of a certain data publisher's API. When fragmenting datasets in ways similar to Chapter 6, it becomes entirely up to the HTTP client's cache to decide what data to keep locally, and what data to download just in time.

There are still open research questions questions that we are going to tackle in the years to come. For one, I look forward researching fragmentation strategies within the domains of geo-spatial data. Our hypothesis is that intermodal route planning, combining both modes using road networks and public transport routing, can be achieved when a similar approach for routable tiles can be found. Moreover, solving full-text search by exposing fragments, inspired by how indexes and tree structures are built today, may afford a more optimized information architecture for federated full-text search. Finally, also organizational problems still need to be tackled: what interfaces need to be hosted by whom?

For the next couple of years, I look forward to building and expanding our IDLab team on Linked Data interfaces, and work further on projects with these organizations that I got to know best. A new generation of PhD researchers is already working on follow up projects, building a more queryable future for the many, not the few.