CityREST: CityJSON in A Database + RESTful Access

Xiaoai Li

Supervisors:
Hugo Ledoux
Jordi van Liempt
Stelios Vitalis

TU Delft

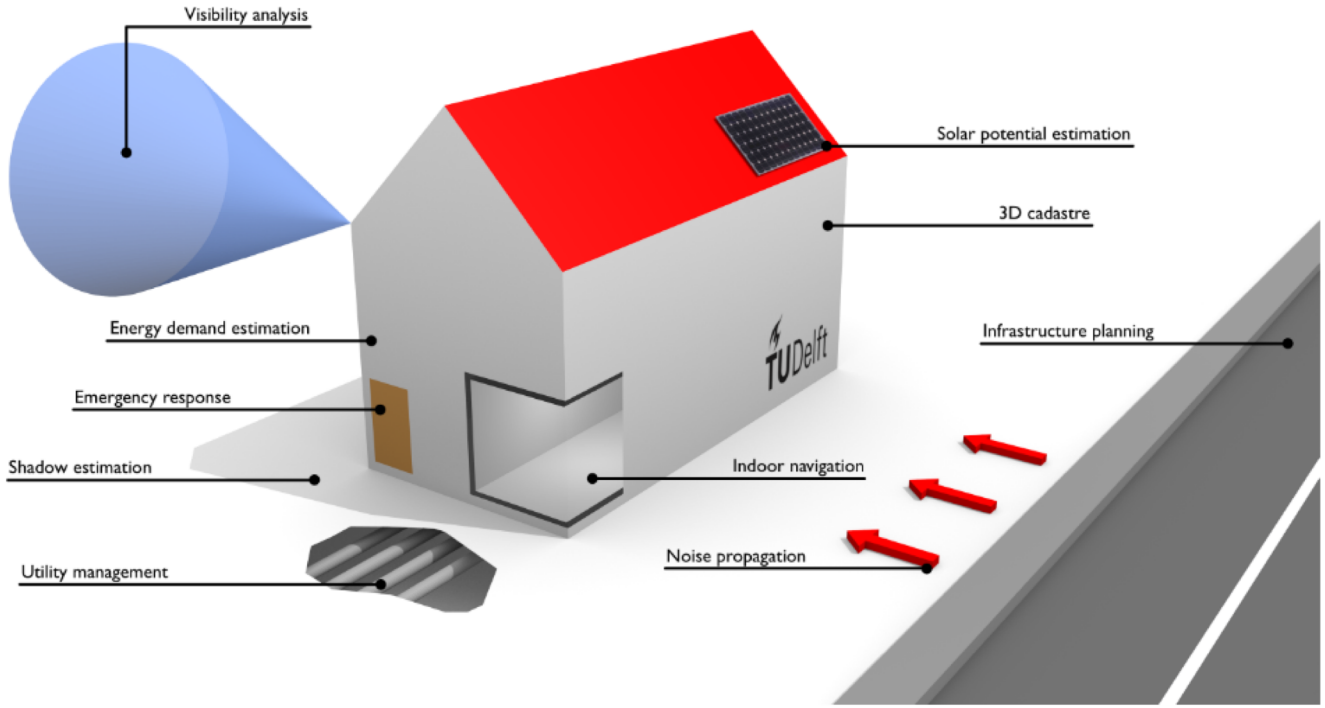
Contents

- Introduction
- Theoretical background and related work
- Methodology
- Implementation
- Benchmarking and results
- Conclusion and future work

TU Delft

Introduction

3D City Models



Applications in 3D city models (Biljecki et al., 2015)

Dissemination of 3D City Models

pdok

3D Topografie

Beschikbare 3D Topografie bestanden

Voor Nederland komen drie 3D Topografie-bestanden beschikbaar als open data: 3D Basisbestand Volledig, 3D Basisbestand Gebouwen en de 3D Hoogtestatistieken Gebouwen. Deze zullen jaarlijks worden geactualiseerd. Meer informatie over 3D Topografie en de voorwaarden waaronder dit bestand beschikbaar gesteld wordt is te vinden op de informatiepagina 3D Basisvoorziening.
De Hoogtestatistieken worden geleverd als één bestand voor heel Nederland. De andere datasets per kaartblad. Het bestandsformaat is CityJSON en wordt in een zip bestand geleverd. Omvang van een kaartbladbestand is ongeveer 200-700 MB.

Werkwijze voor het downloaden van een kaartblad

Selecteer in het drop-down menu het gewenste luchtfotojaar. Momenteel is alleen 2018 beschikbaar.
Selecteer op de kaart het gewenste kaartblad. Inzoomen kan met de scrolfunctie van uw muis of door dubbelklikken op de kaart.
Vervolgens vindt u in de tabel rechts van de kaart of onder de kaart een link om het bijbehorende zipbestand te downloaden.

Luchtfotojaar: 2018

KAARTBLAD:		
INHOUD	FORMAAT	LINK
3D Basisbestand Volledig	CityJson (gezipt)	
3D Basisbestand Gebouwen	CityJson (gezipt)	

LANDSDEKKEND		
INHOUD	FORMAAT	LINK
3D Hoogtestatistieken Gebouwen	GeoPackage 1.2 (gezipt)	Download

Indien u vragen heeft over het product of feedback wilt delen dan kunt u contact opnemen via beheerpdok@kadaster.nl of 088-1834500.

The 3D dataset access through the Dutch geoportal, PDOK

OGC standards for 3D city models

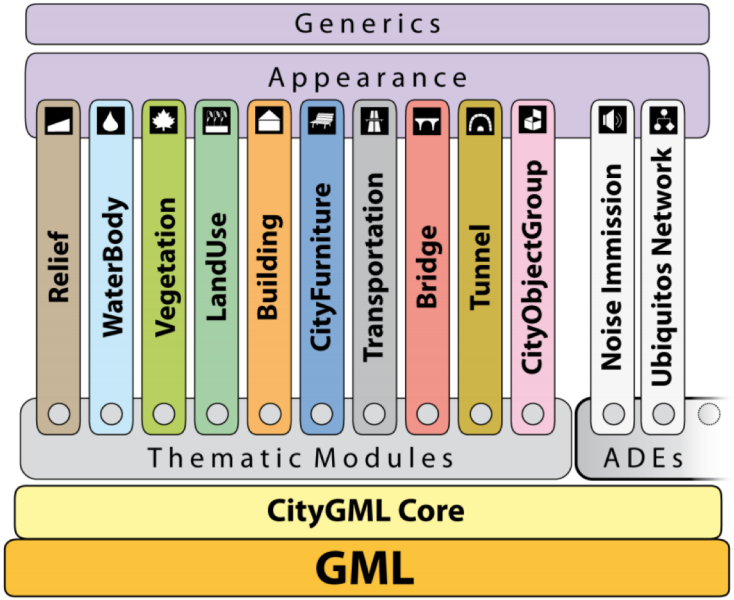


Image source: virtualcitySYSTEMS

CityGML

CityJSON

Encodings for 3D city models

TU Delft

Research Question:

*How to best develop a **RESTful API** for **fast access** to geospatial features in CityJSON and how to properly store CityJSON in a **database** to support the RESTful access?*

TU Delft

Theoretical background and related work

CityJSON

```
{
  "type": "CityJSON",
  "version": "1.0",
  "CityObjects": {},
  "vertices": []
}
```

The minimal valid CityJSON object

```
{
  "id-1": {
    "type": "Building",
    "attributes": {
      "measuredHeight": 22.3,
      "roofType": "gable",
    },
    "children": ["id-2"],
    "geometry": [{...}]
  },
  ...
}
```

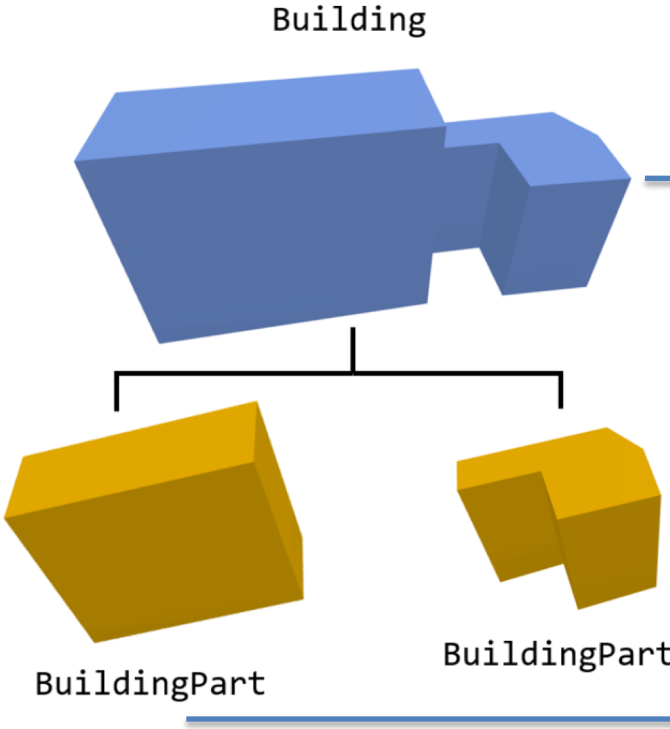
One CityObject

```
{
    "type": "Solid",
    "lod": 2,
    "boundaries": [[
        [[0,3,2,1,22]],
        [[4,5,6,7]],
        [[0,1,5,4]],
        [[1,2,6,5]]
        ]],
    "semantics": {
        "surfaces" : [
        {"type": "RoofSurface" },
        {
         "type": "WallSurface",
         "paint": "blue"
        },
        {"type": "GroundSurface" }
        ],
        "values": [ [0, 1, 1, 2] ]
    }
}
```

One Geometry object

TU Delft

CityJSON




```
"CityObjects": {
    "id-1": {
        "type": "Building",
        "geometry": [],
        "children": [
            "id-2",
            "id-3"
        ]
    },
    "id-2": {
        "type": "BuildingPart",
        "geometry": [...],
        "parents":  ["id-1"]
    },
    "id-3": {
        "type": "BuildingPart",
        "geometry": [...],
        "parents":  ["id-1"]
    }
}
```

TU Delft

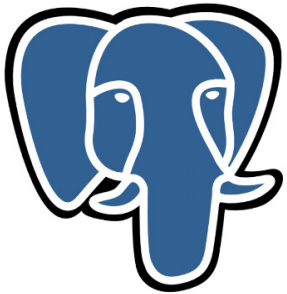
OGC API - Features

Resource	Path	Purpose
Landing page	/	This is the top-level resource, which serves as an entry point.
Conformance declaration	/conformance	This resource presents information about the functionality that is implemented by the server.
API definition	/api	This resource provides metadata about the API itself.
Feature collections	/collections	This resource lists the feature collections that are offered through the API.
Feature collection	/collections/{collectionId}	This resource describes the feature collection identified in the path.
Features	/collections/{collectionId}/items	This resource presents the features that are contained in the collection.
Feature	/collections/{collectionId}/items/{featureId}	This resource presents the feature that is identified in the path

The resources defined by OGC API (Open Geospatial Consortium, 2020)

TU Delft

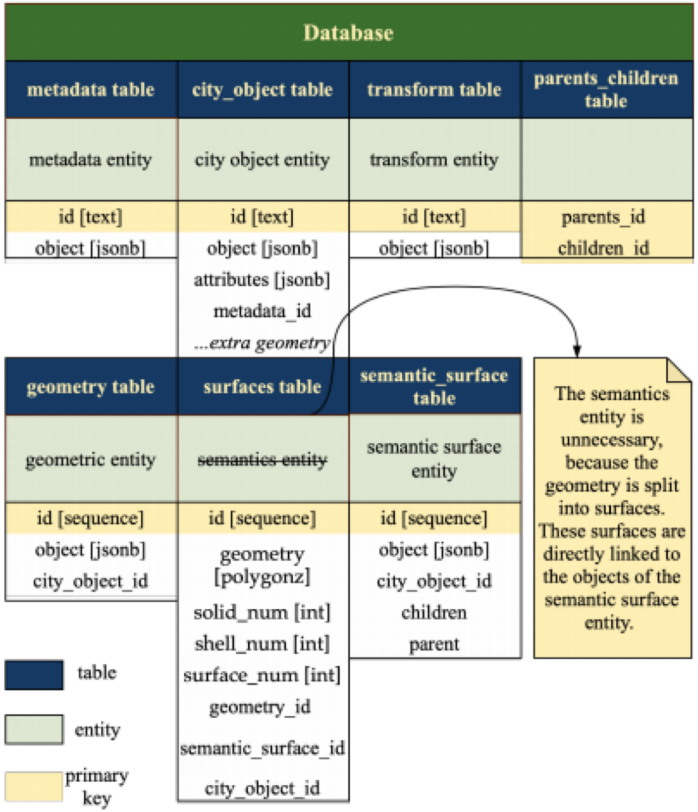
Database



Working with JSONB in

PostgreSQL

Image Source: haselt



An overview of the relational database schema (Staring, 2020)

Streaming

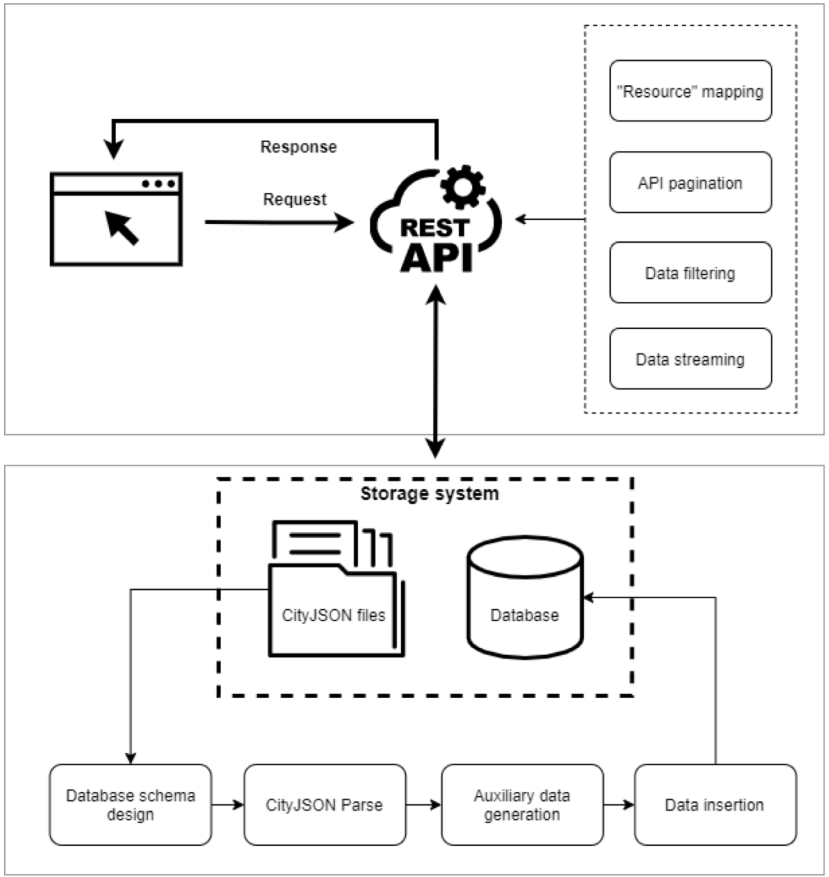
Data format modification to to enable the stream-based delivery of geometry data
Newline-delimited JSON + CityJSON → CityJSONFeature (Ledoux, 2020)

```
{
    "type": "CityJSONFeature",
    "id": "myid",
    "CityObjects": {},
    "vertices": [],
    "appearance": {}
}
```

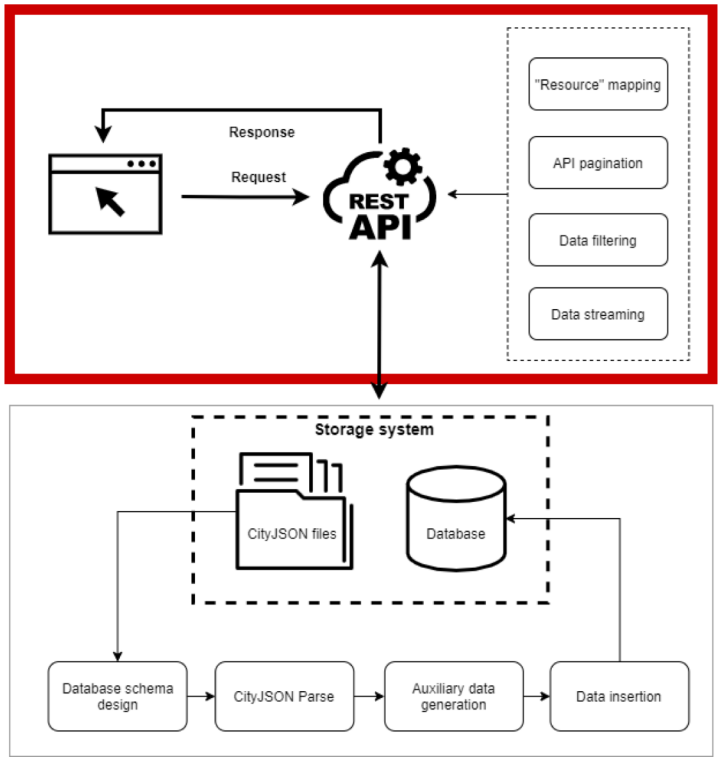
TU Delft

Methodology

Overview


"Resource" mapping
API pagination
Data filtering
Data streaming
Response
Request
REST API
Storage system
CityJSON files
Database
Database schema design
CityJSON Parse
Auxiliary data generation
Data insertion

Developing a RESTful API



Resource mapping

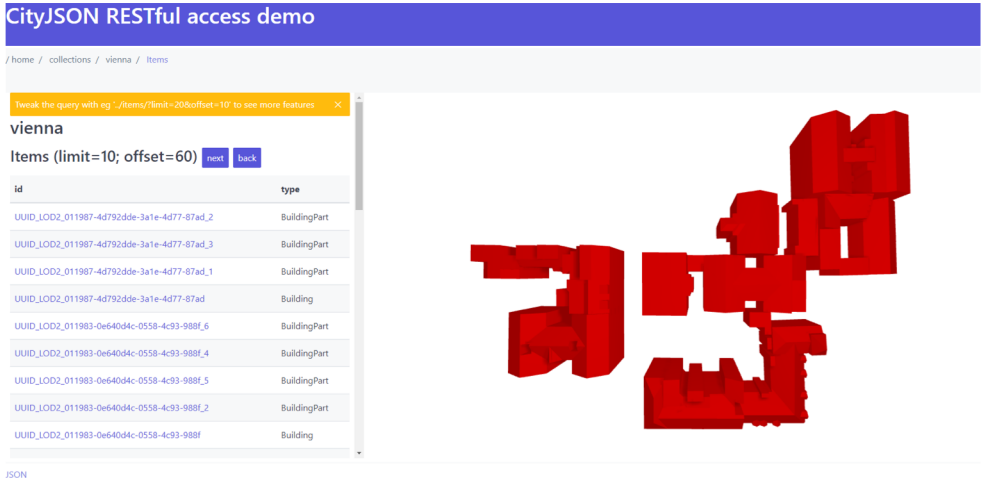
Resource	Path	CityJSON
Feature collections	/collections	Overview of all datasets
Feature collection	/collections/{collectionId}	Overview of a dataset
Features	/collections/{collectionId}/items	A (sub)CityJSON object
Feature	/collections/{collectionId}/items/{featureId}	One CityJSONFeature

Features → large amounts of data → Pagination

Resource	Path	Responce
Features	/collections/{collectionId}/items?limit={10} &offset= {60}	A (sub)CityJSON object

- API Pagination : a key strategy for making sure the API run smoothly and effectively.

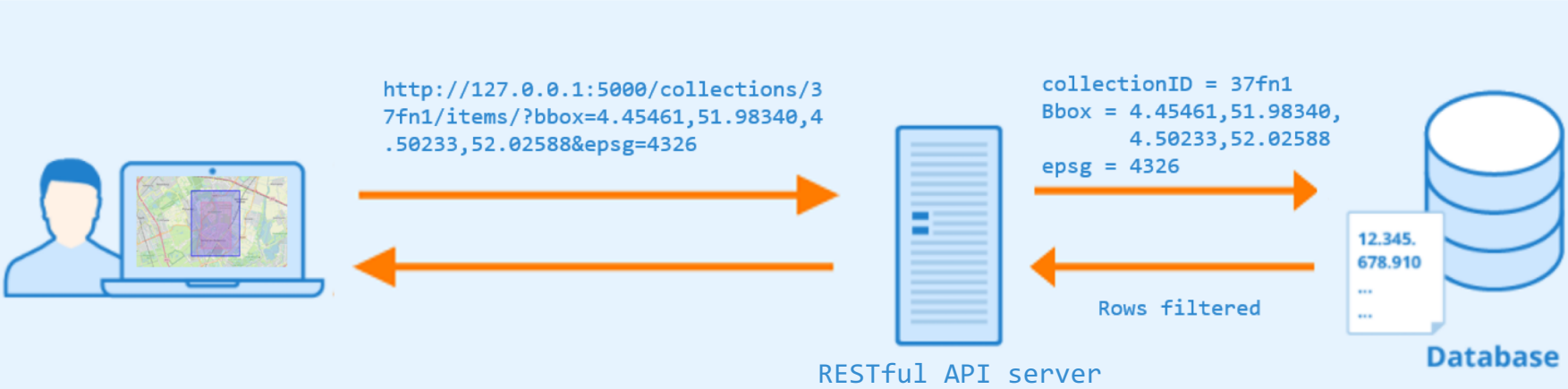
- Offset Pagination
  - simplicity
  - **`limit`** and **`offset`** are included in the **`SQL`** library



TU Delft

Bounding box filtering on one CityJSON dataset

Resource	Request
Features	/collections/{collectionId}/items/?bbox={[minx, miny, maxx, maxy]} & epsg= {n}

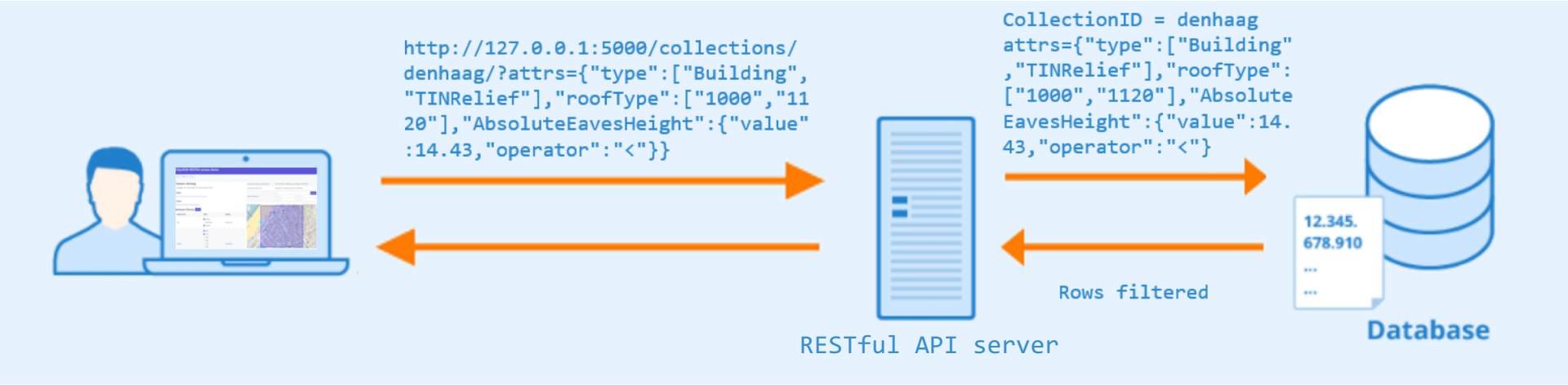


Bounding box filtering on multiple CityJSON datasets

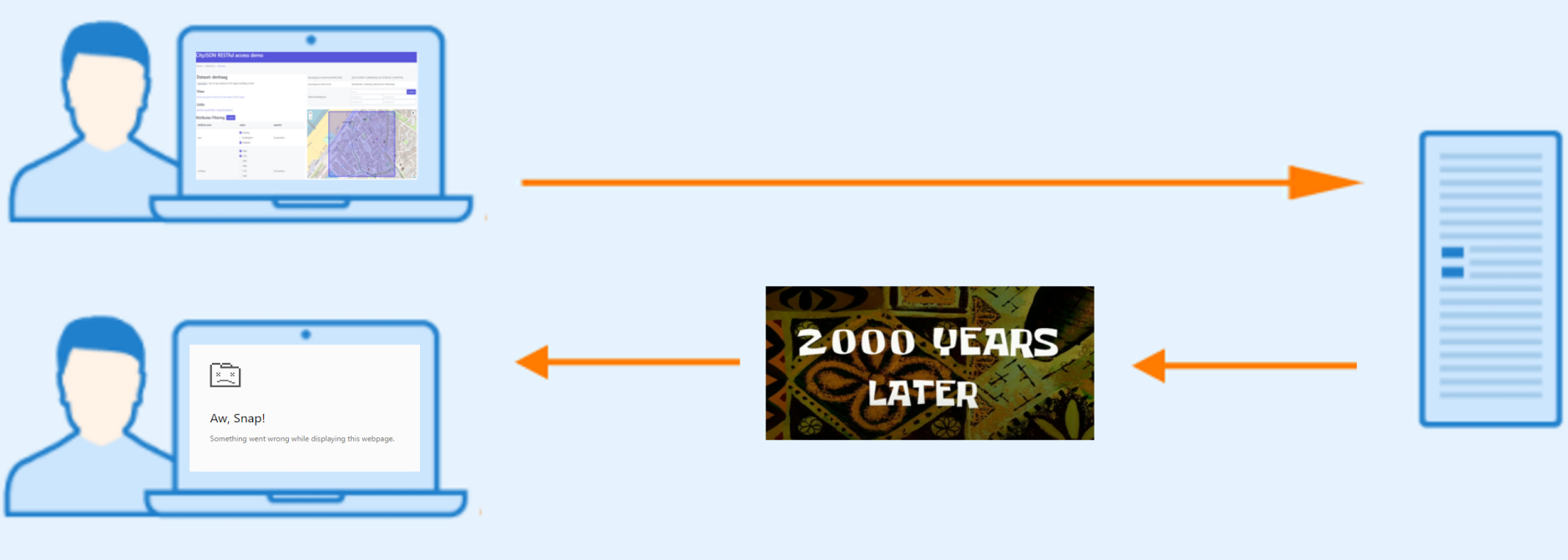
Resource	Request
Collections	/collections/bbox={[minx, miny, maxx, maxy]} & epsg= {n}

Attribute filtering on one CityJSON dataset

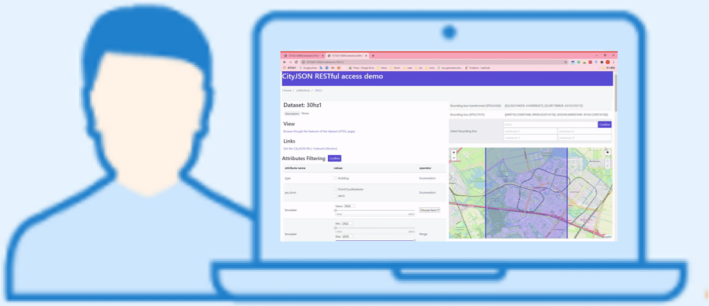
Resource	Request
Features	/collections/{collectionId}/items/?attrs=

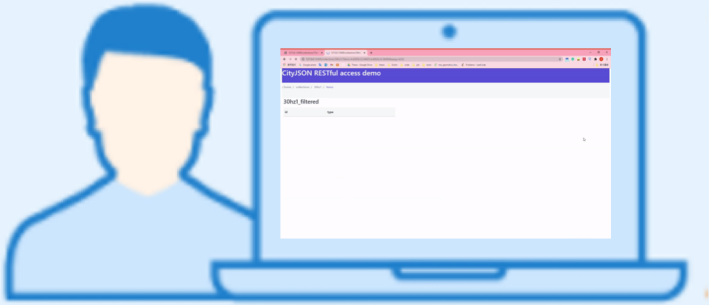


How to send the **large** filtered data to the user?



Streaming



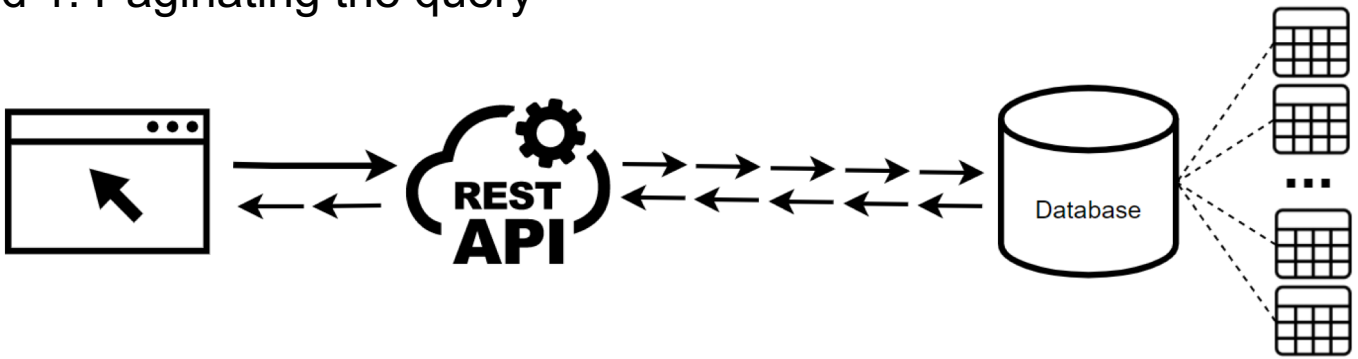


The large CityJSON → CityJSONFeatures

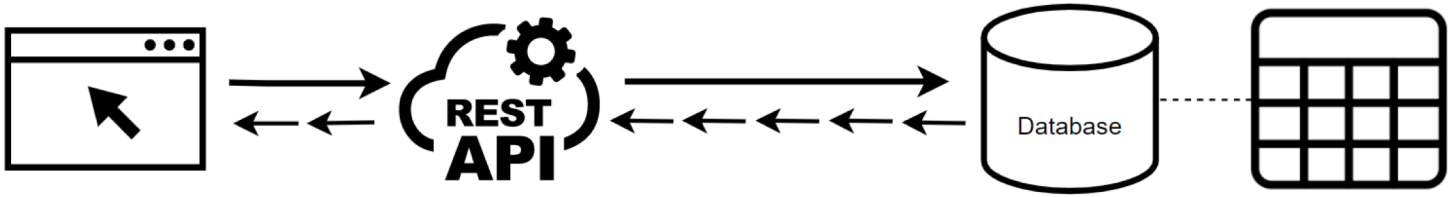
RESTful API server

Streaming data from the database to the RESTful API

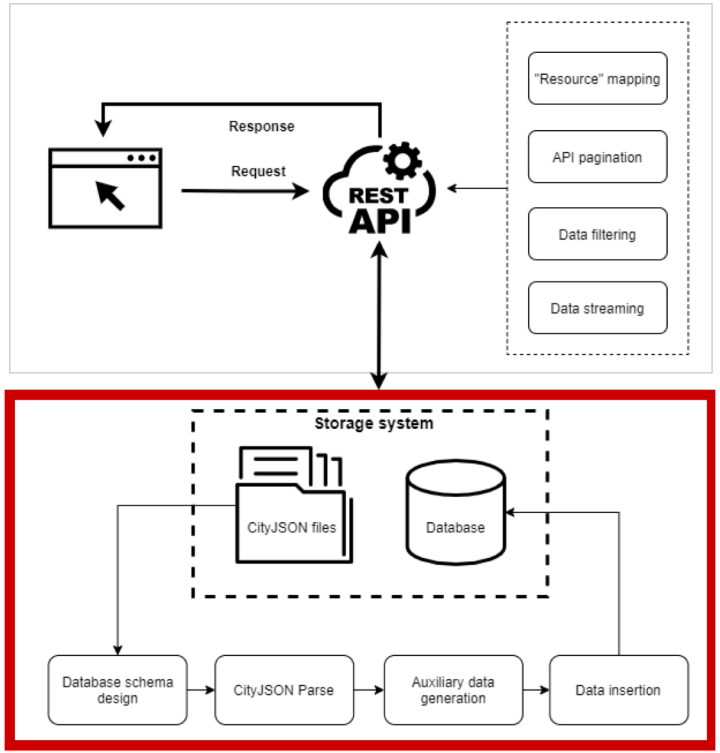
- Method 1: Paginating the query



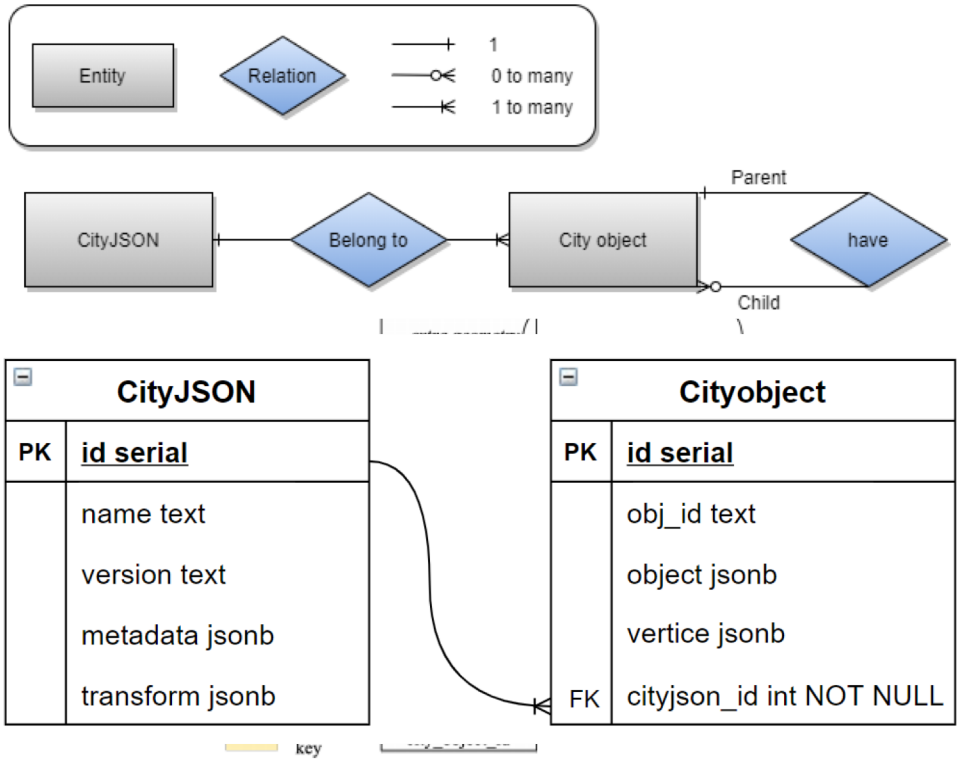
- Method 2: Chunking the query result



Storing CityJSON in a database



Database schema design



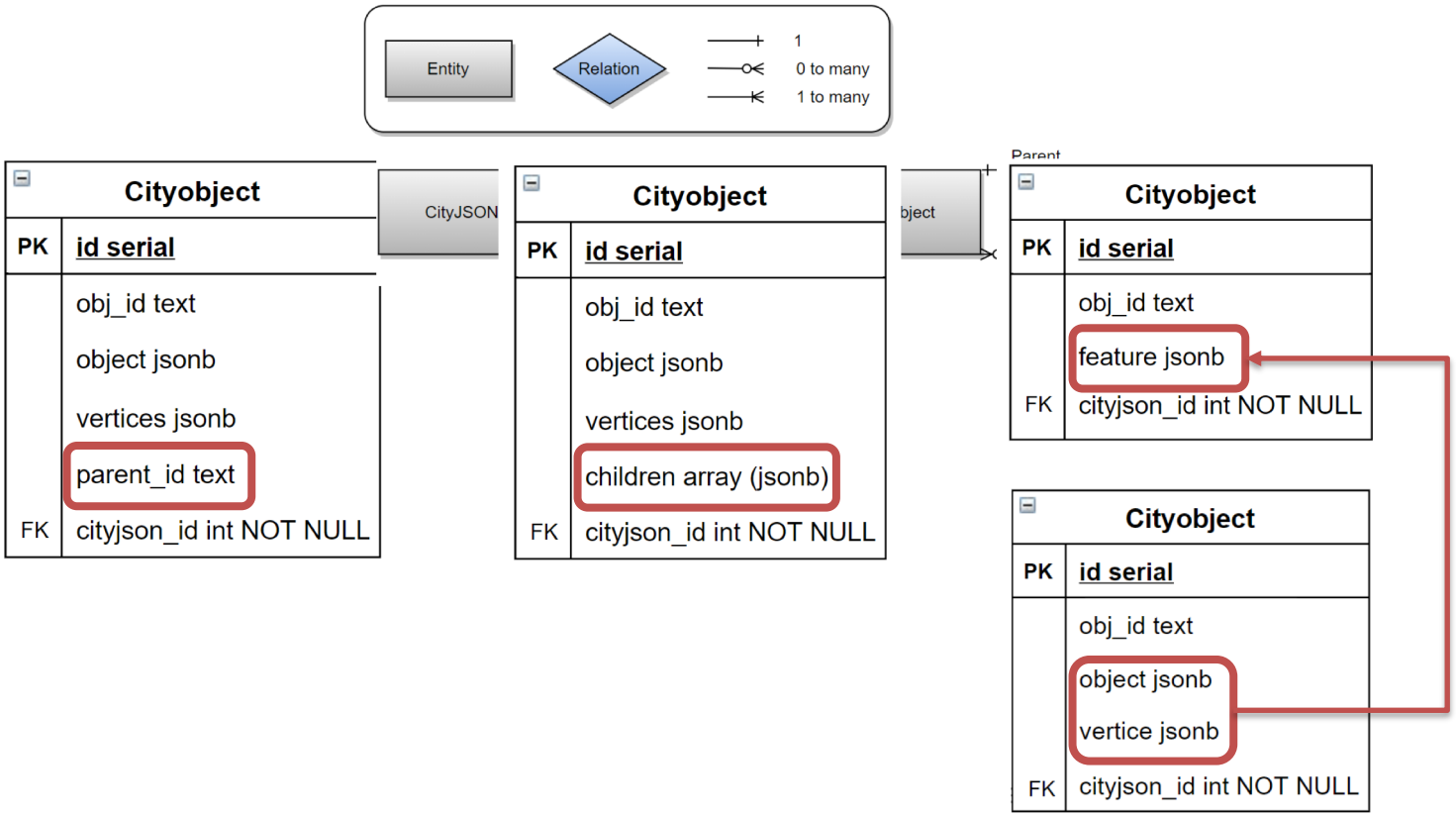
Geometry unit → 3D surface

- Difficulties in reconstructing the original 3D geometries
- The minimum accessible resource in the RESTful API is a city object
- Increased storage size

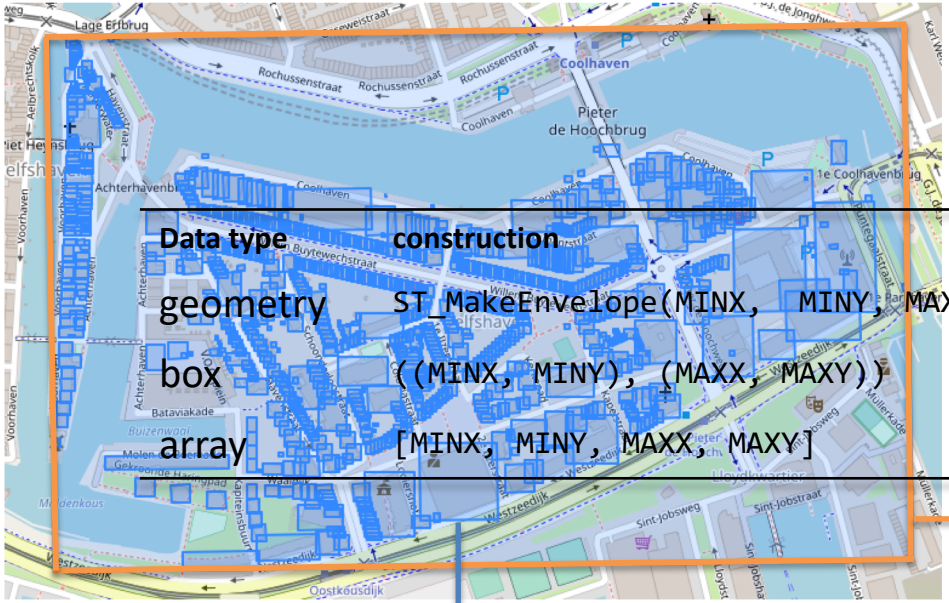
	Size of storage
File system	406 MB
Old schema	1268 MB
New schema	375 MB

TU Delft

Parent-child relation



2D bounding boxes

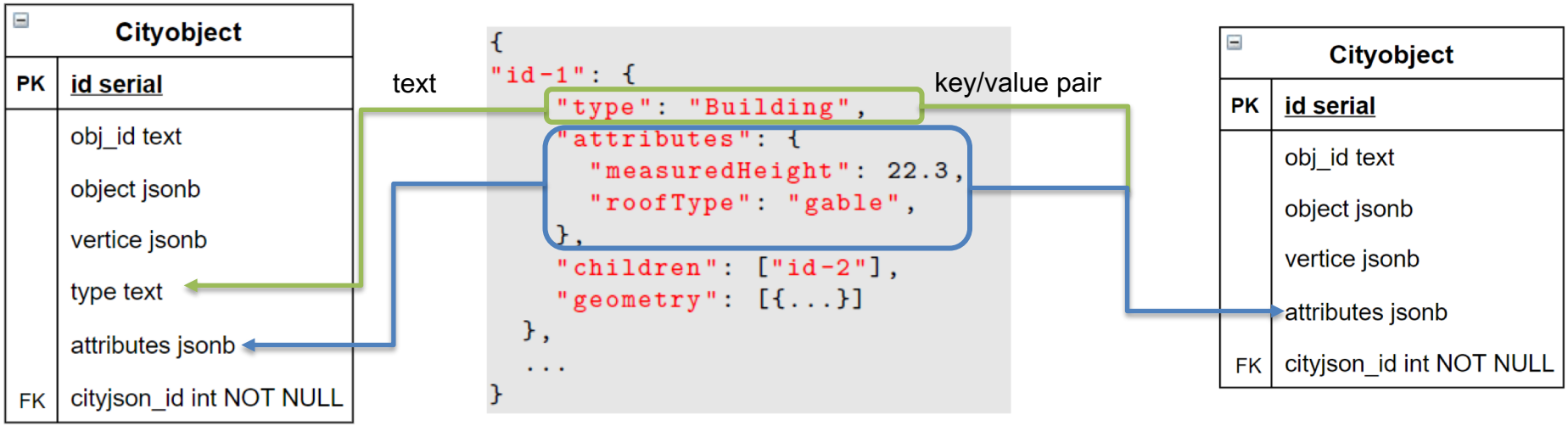


Data type	construction
geometry	ST_MakeEnvelope(MINX, MINY, MAXX, MAXY, srid)
box	((MINX, MINY), (MAXX, MAXY))
array	[MINX, MINY, MAXX, MAXY]

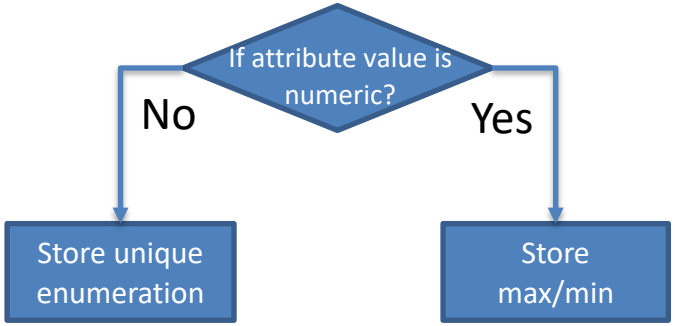
CityJSON			
PK	id serial		
	name text	index	intersection
	version text		
	metadata jsonb	GiST	&&
	transform jsonb	GiST	&&
	bbox ?	B-tree	numeric

Cityobject	
PK	id serial
	obj_id text
	object jsonb
	vertices jsonb
	parent_id text
	bbox ?
FK	cityjson_id int NOT NULL

Semantics (at city object level)



Semantics: Queryables + values



CityJSON	
PK	**id serial**
	name text
	version text
	metadata jsonb
	transform jsonb
	attribute_info jsonb

	id [PK] integer	name text	attribute_info jsonb
1	1	37en2_volledig	{"type": ["Bridge", "WaterBody", "PlantCover", "G
2	2	Zurich_Building...	{"type": ["Building", "BuildingPart"], "class": ["BB
3	3	37en2	{"type": ["Building"], "pw_bron": ["PointCloudKac
4	4	37ez1	{"type": ["Building"], "pw_bron": ["PointCloudKac
5	5	37ez2	{"type": ["Building"], "pw_bron": ["PointCloudKac
6	6	30dz2	{"type": ["Building"], "pw_bron": ["PointCloudKac


```
{
    "type": [
        "Bridge", "WaterBody", "PlantCover",
        "GenericCityObject", "LandUse",
        "Building", "Road"
        ],
    "baseheight": [
        -5.46000003814697,
        5.57000017166138
        ],
    "roofheight": [
        -0.240000009536743,
        149.120010375977
        ],
    "onbegroeidterreindeeloptalud": [
        "false",
        "true"
        ],
    ...
}
```


TU Delft

Implementation
html response

Tools

Jinja template engine
JavaScript/Jinja

Flask server
Python

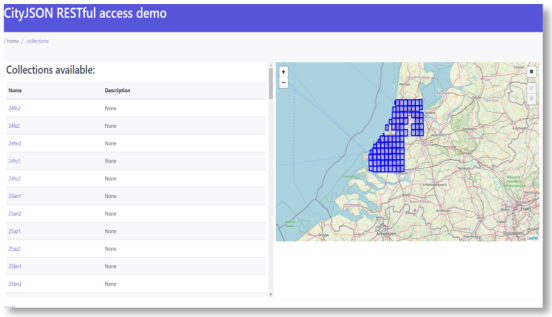
PostgreSQL
SQL

jquery.js
leaflet.js
Three.js

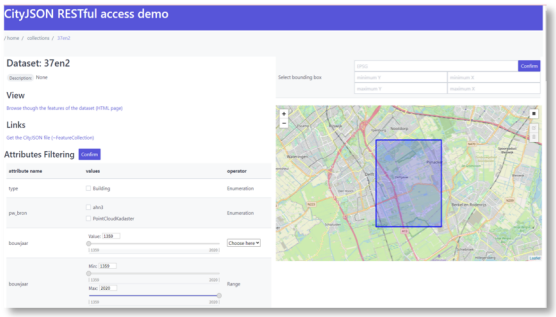
Psycopg2
Cjio
pyproj
flask

PostGIS
extension

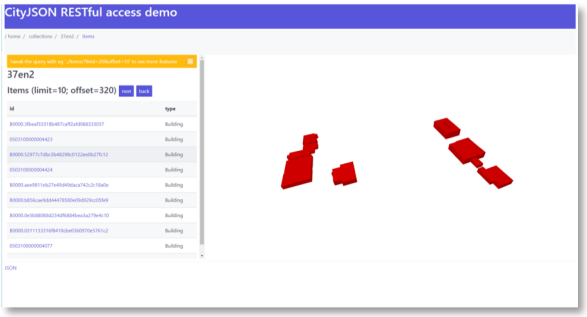
Data access



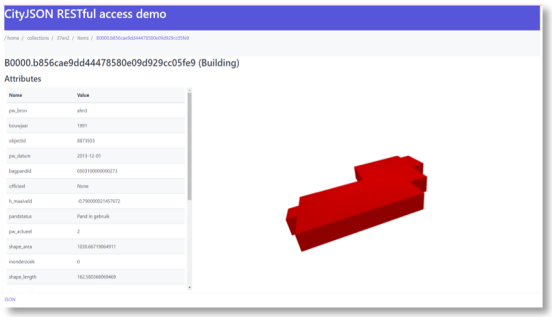
Collections



Collection

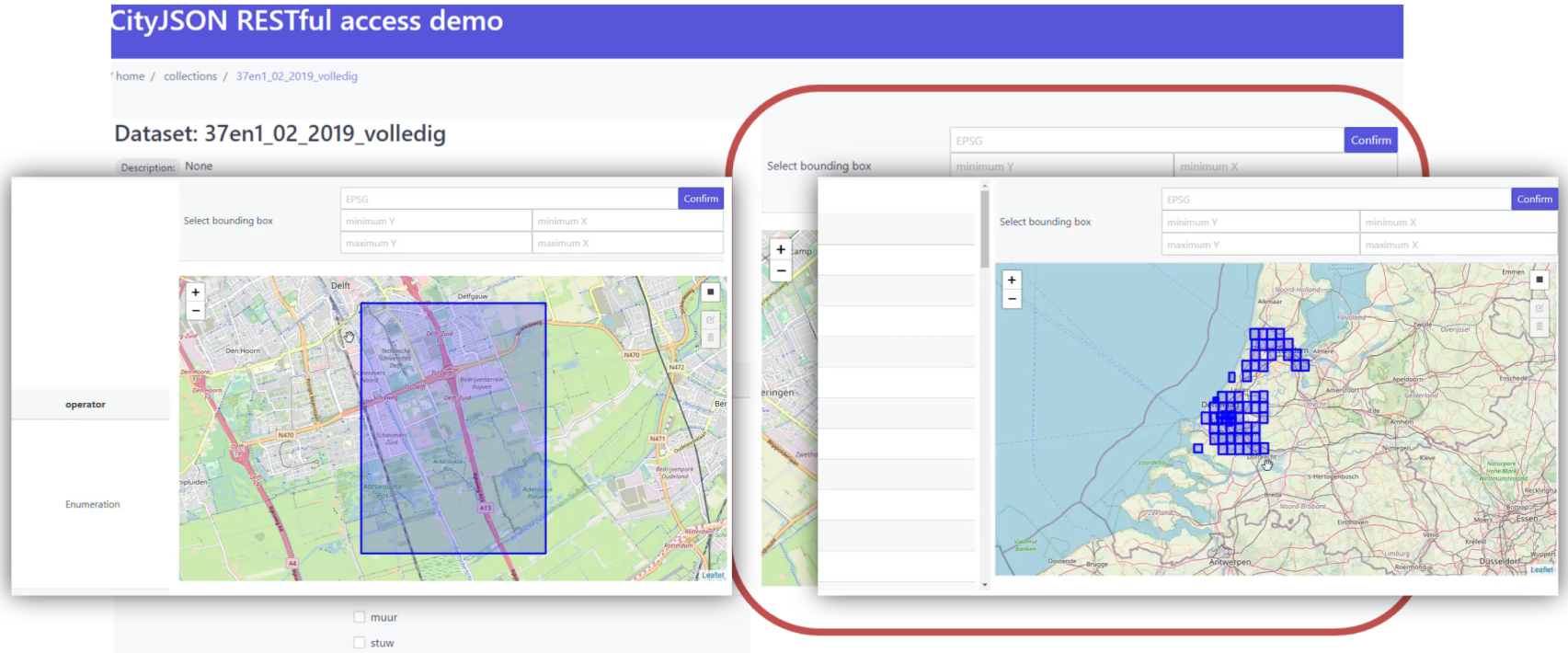


Features



Feature

Bounding box filtering



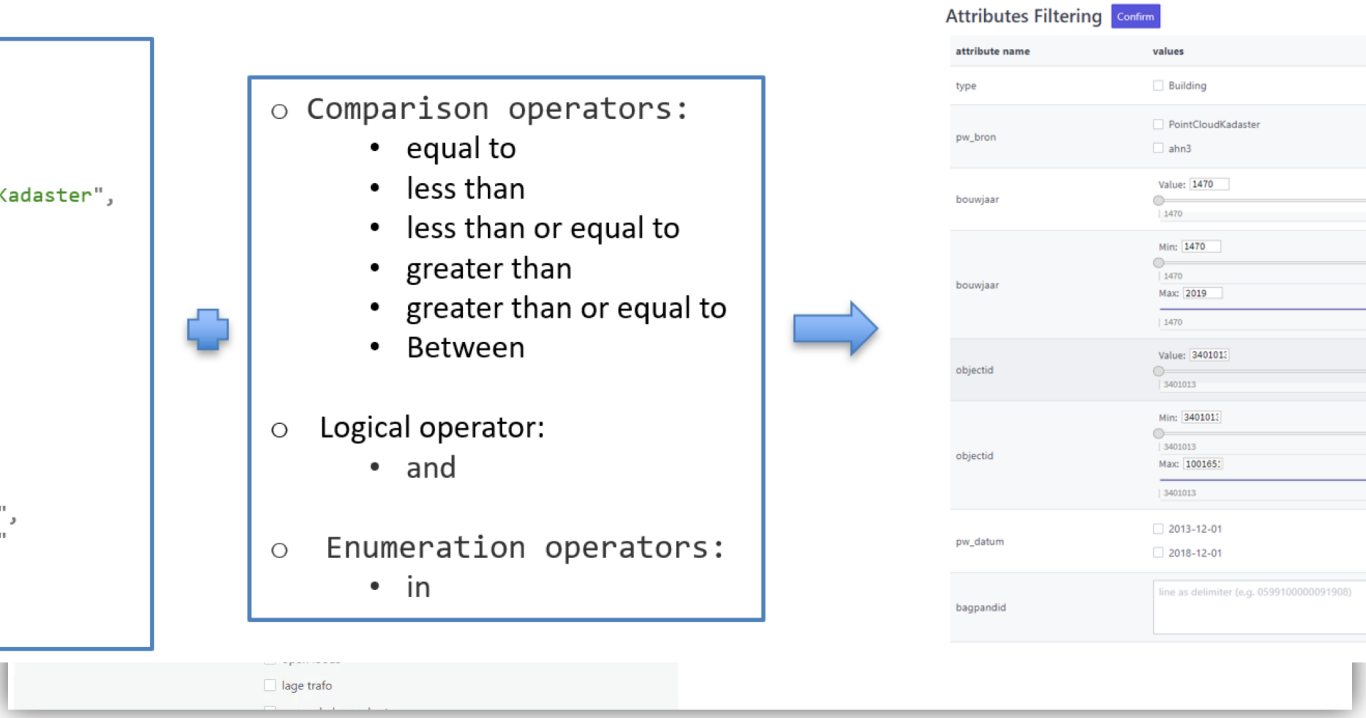
Filtering one CityJSON dataset

Filtering multiple CityJSON datasets

Attribute filtering

```
{
    "type": [
        "Building"
    ],
    "pw_bron": [
        "PointCloudKadaster",
        "ahn3"
    ],
    "bouwjaar": [
        1470,
        2019
    ],
    "objectid": [
        3401013,
        10016518
    ],
    "pw_datum": [
        "2013-12-01",
        "2018-12-01"
    ]
    ...
}
```

- Comparison operators:
  - equal to
  - less than
  - less than or equal to
  - greater than
  - greater than or equal to
  - Between
- Logical operator:
  - and
- Enumeration operators:
  - in

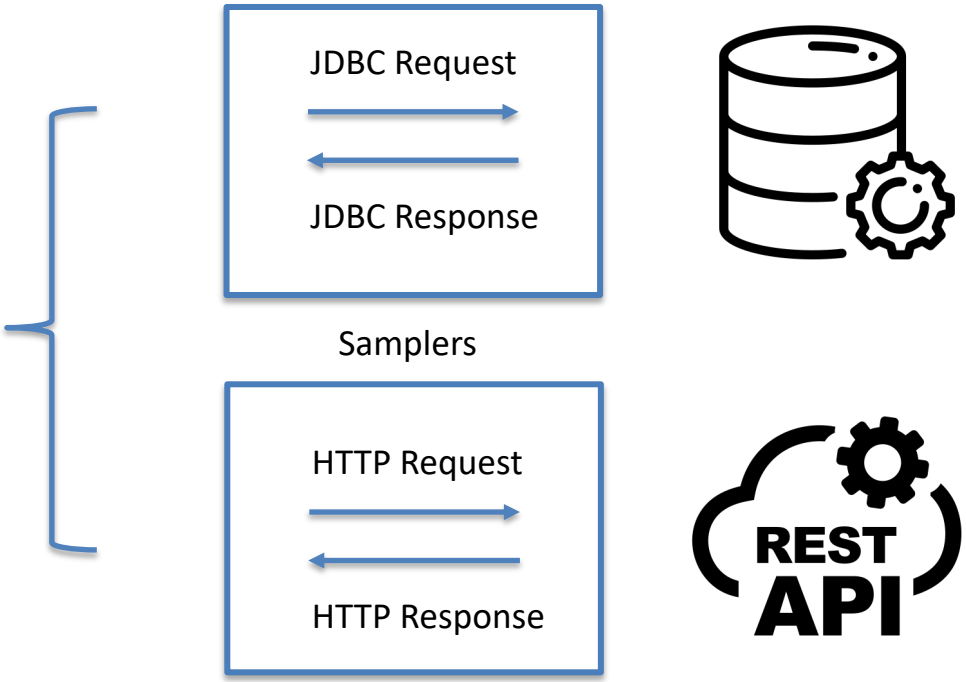


Benchmarking and results

json response

Tools

Create Sampler



- HTTP Cache Manager
- Throttling outgoing bandwidth to simulate the WIFI network speed

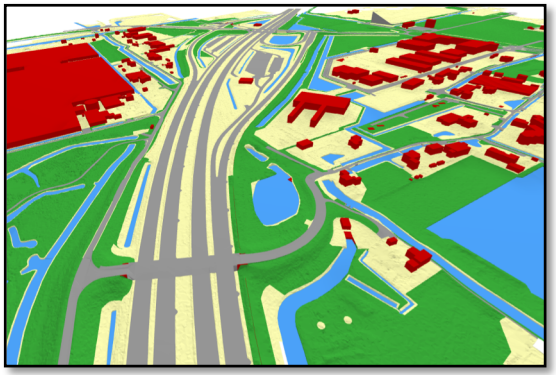
TUDelft

Datasets

File name	File size (MB)	City objects	Characteristics	Source
Zurich_Building_LoD2_V10	286.1	198699	Multi-part buildings	CityJSON [2021]
37en1 ... (57)	20.2 - 133.8	10585 - 63386	Single part buildings	PDOK [2021]
37en1_volledig ... (5)	438.4.5 - 2218.9	19239 - 106417	Various city objects	PDOK [2021]

KAARTBLAD:		
INHOUD	**FORMAAT**	**LINK**
3D Basisbestand Volledig	CityJson (gezipt)	
3D Basisbestand Gebouwen	CityJson (gezipt)	
3D Hoogtestatistieken Gebouwen	GeoPackage 1.2 (gezipt)	

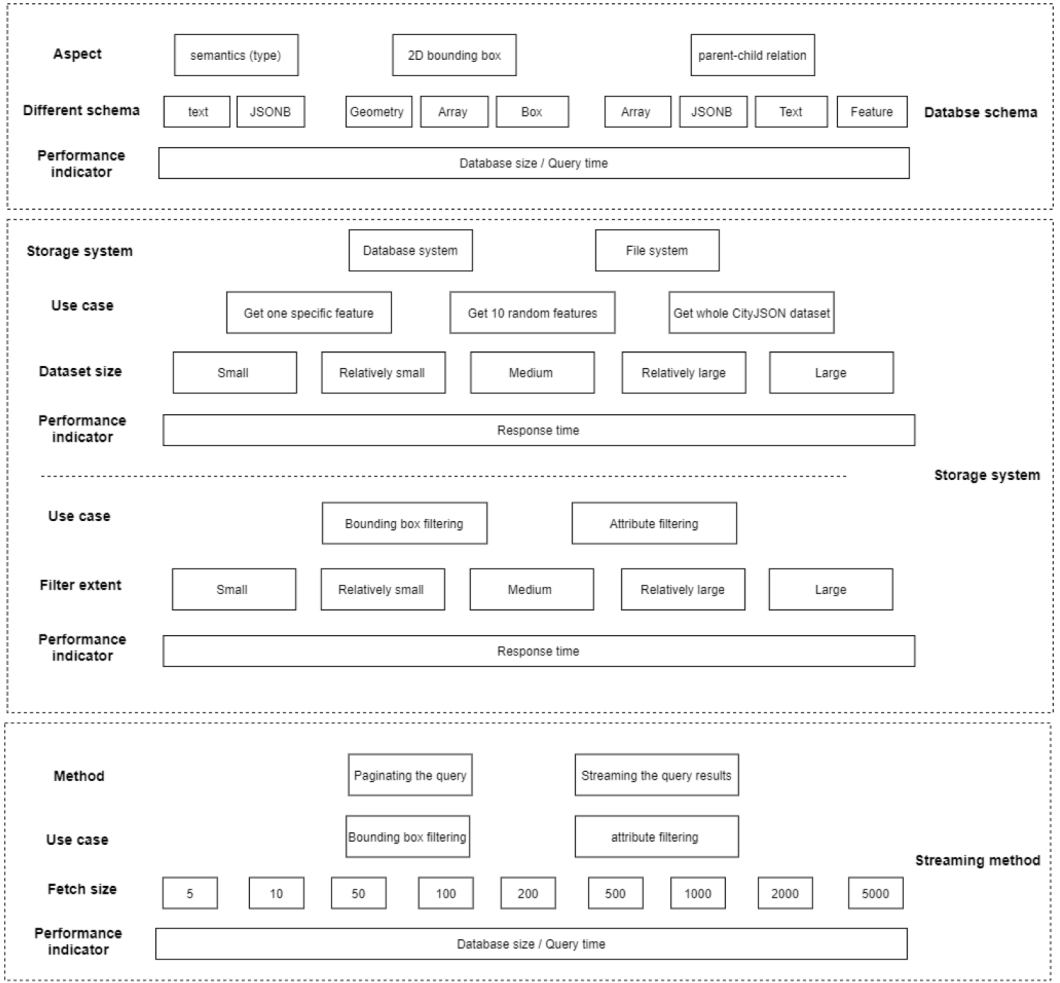
LANDSDEKKEND		
INHOUD	**FORMAAT**	**LINK**
3D Hoogtestatistieken Gebouwen	GeoPackage 1.2 (gezipt)	Download



TU Delft

Methodology

- Database schema
- Storage system
- Streaming method



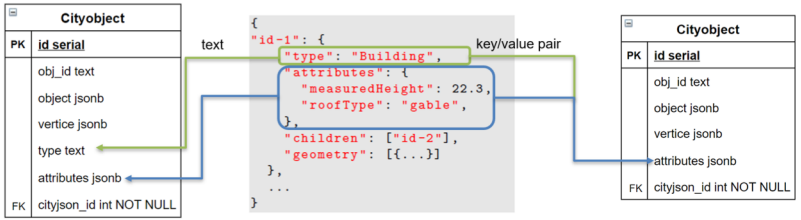
Database schema

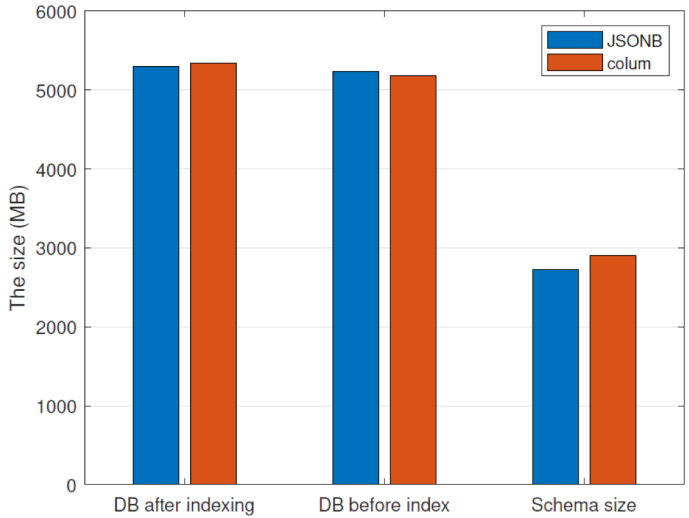
Metric 1: Storage size

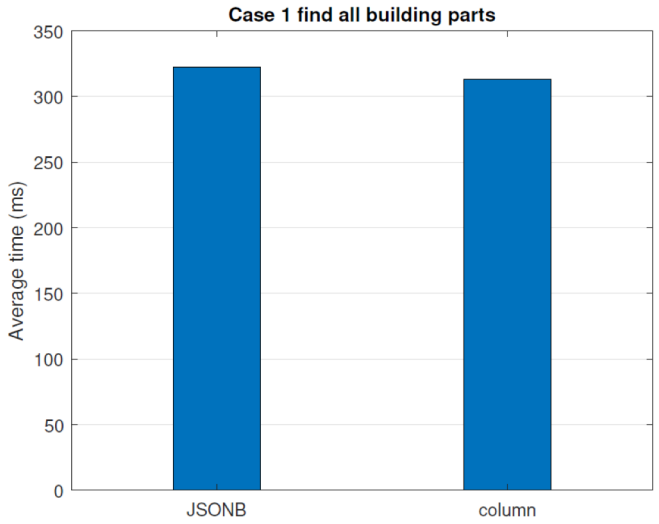
- Schema size → sum of all table sizes
- Database size before extra indexes → including all default indexes, TOAST space, free space map, and visibility map
- Database size after extra indexes

Metric 2: Query time

Semantics: type

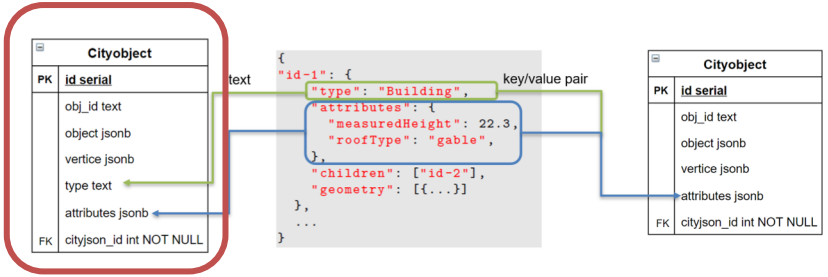


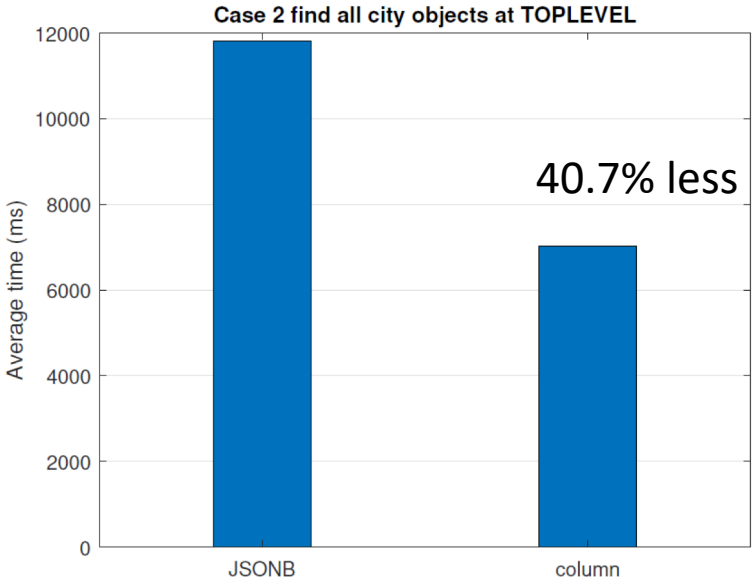




TU Delft

Semantics: type





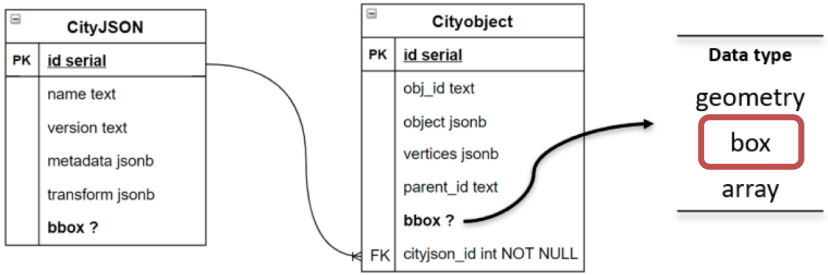
Most city objects are at TOPLEVEL

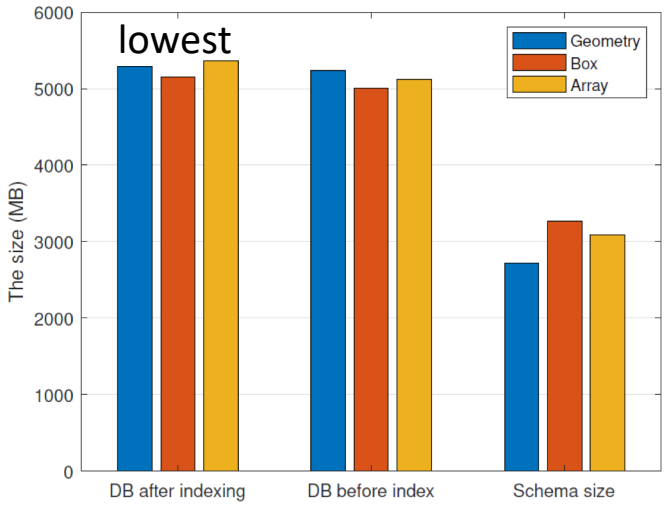
Indexing mechanism → extra seeking time

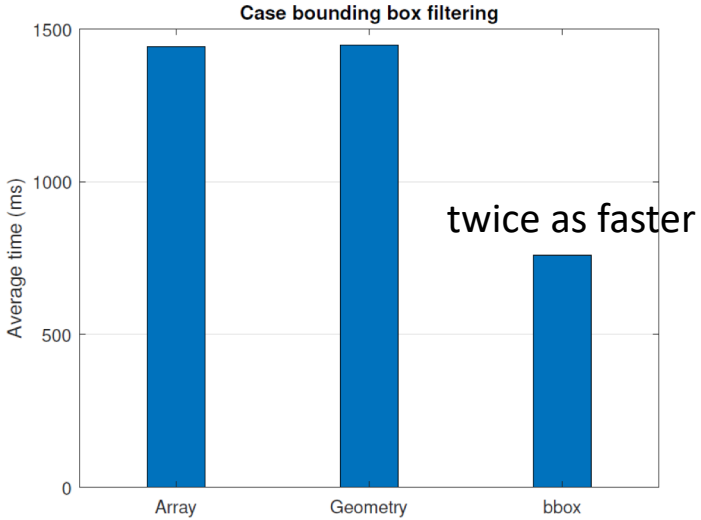
Statistics on traditional types (*text*) → better query planner

TU Delft

2D bounding box





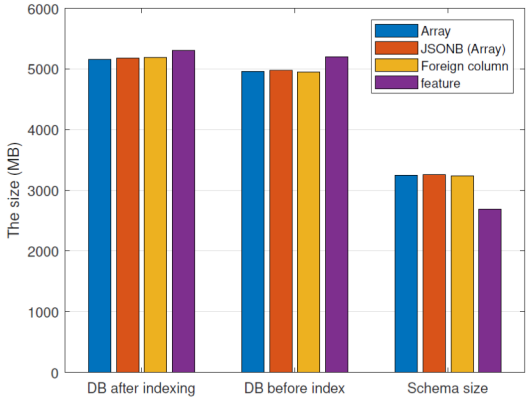


Parent-child relation

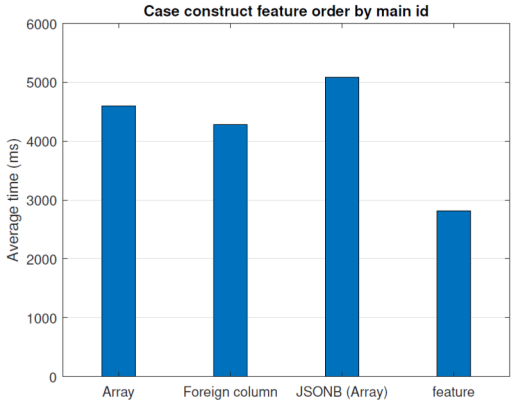
Cityobject	
PK	**id serial**
	obj_id text
	object jsonb
	vertices jsonb
	parent_id text
FK	cityjson_id int NOT NULL

Cityobject	
PK	**id serial**
	obj_id text
	object jsonb
	vertices jsonb
	children array (jsonb)
FK	cityjson_id int NOT NULL

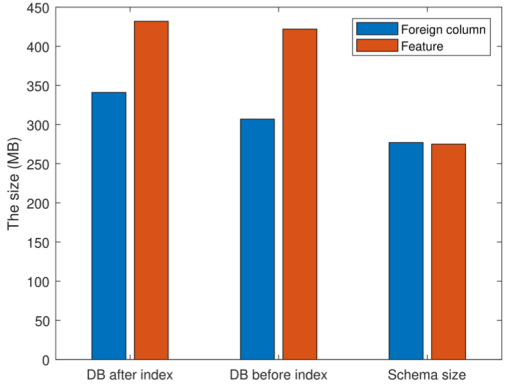
Cityobject	
PK	**id serial**
	obj_id text
	feature jsonb
FK	cityjson_id int NOT NULL



63 CityJOSN datasets



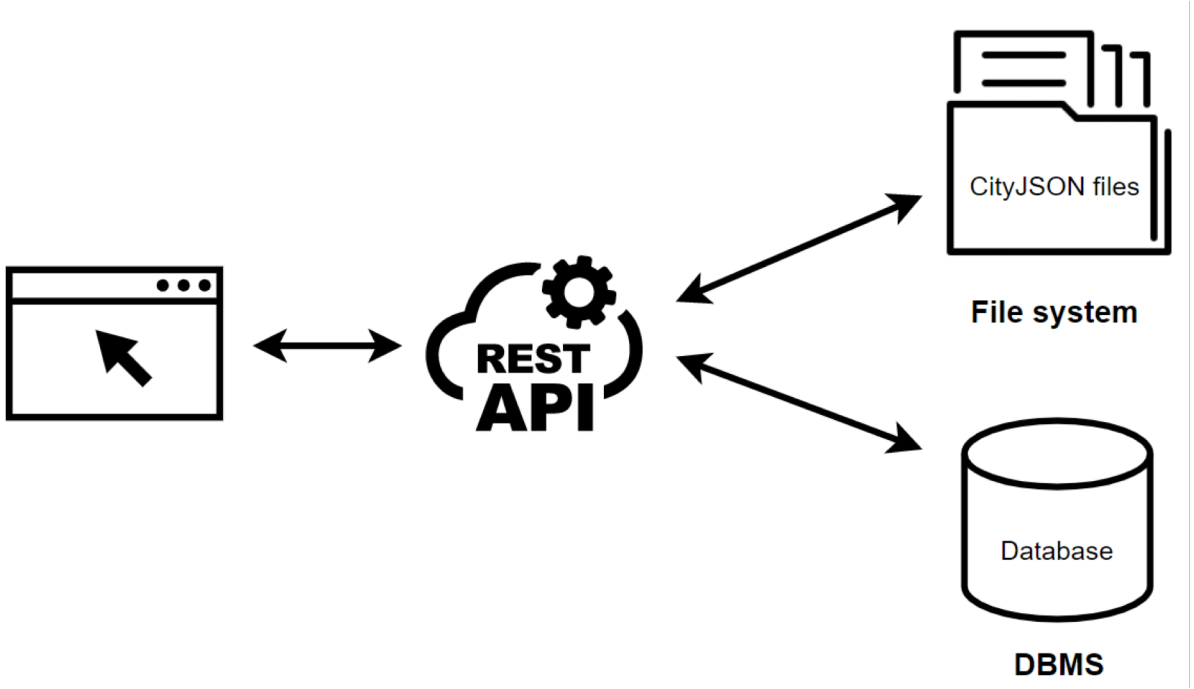
increases by 37%



Only *Zurich* dataset in DBMS

Storage system

Metric: The response time

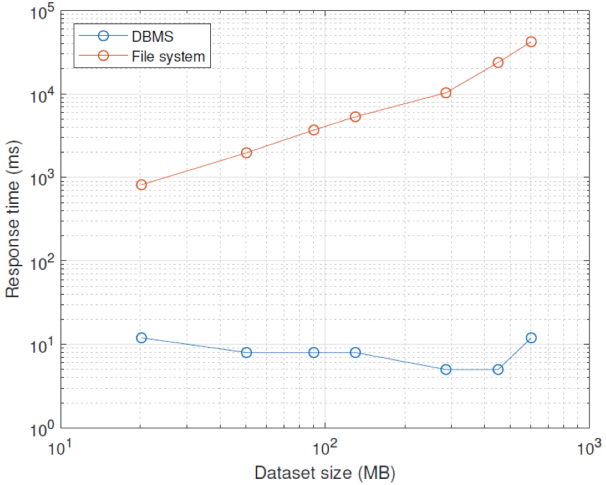


Data access without filtering

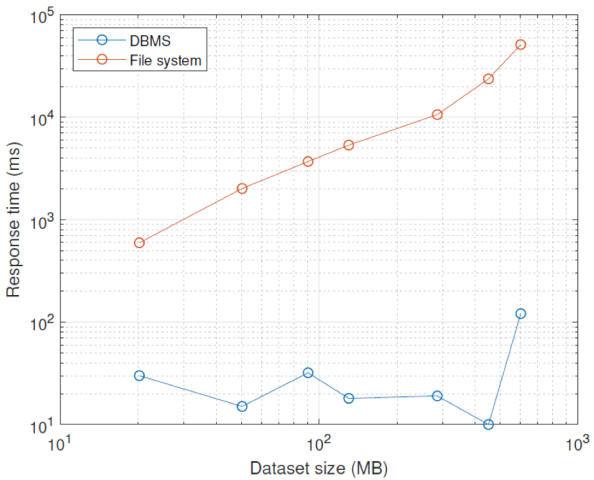
7 (63 total) CityJSON datasets

File name	File size (MB)
25an2	20,2
38cn1	50,4
37hn1	90,6
30dz2	130,1
Zurich_Building_LoD2_V10	286,1
37en1_01_2019_volledig	451,1
30gz2_02_2019_volledig	601,2

Case 1: access to one specific city object



Case 2: access to 10 random city objects



TU Delft

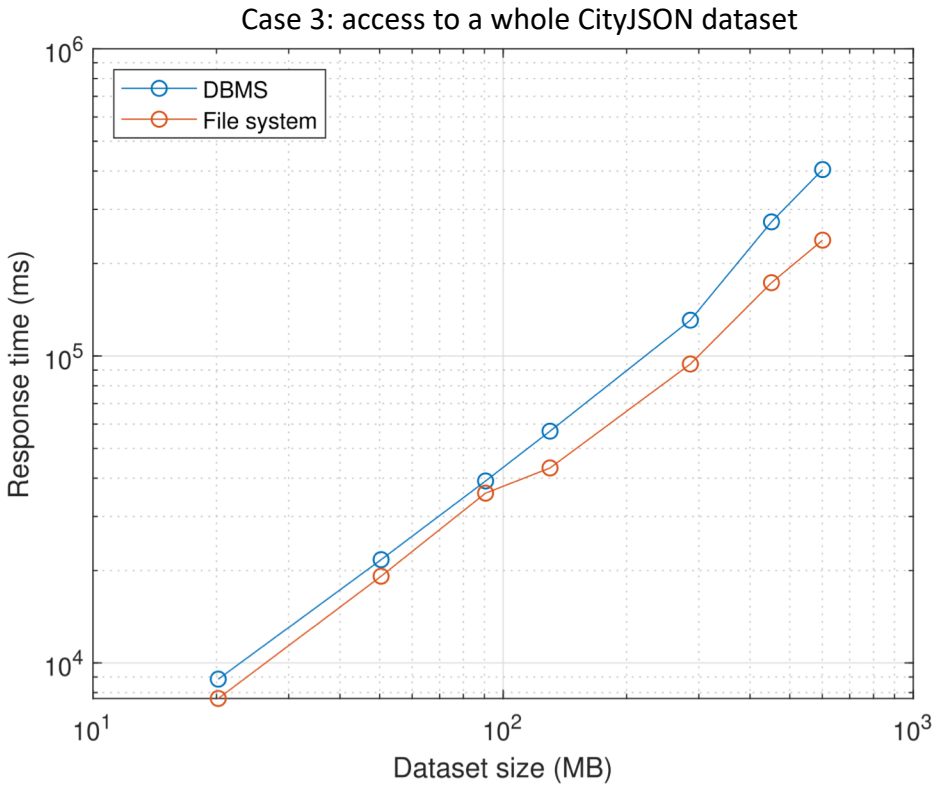
Data access without filtering

- Extra reconstruction work

  - city objects to CityJSON

- Database connection and query

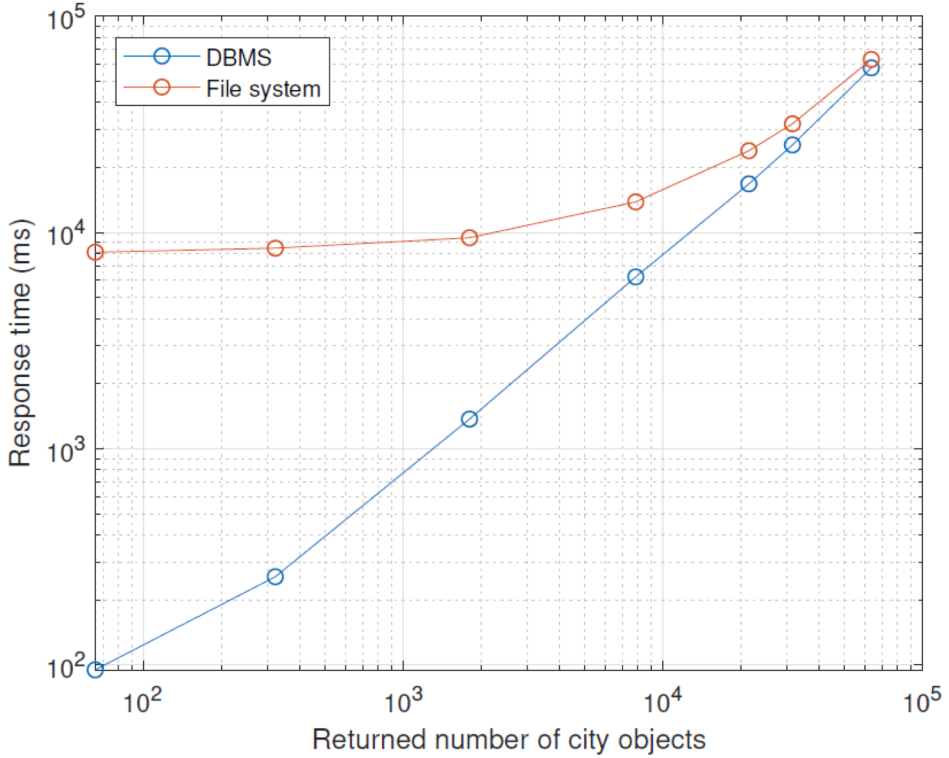
  - resource intensive



TU Delft

Bounding box filtering with *30dz2* dataset

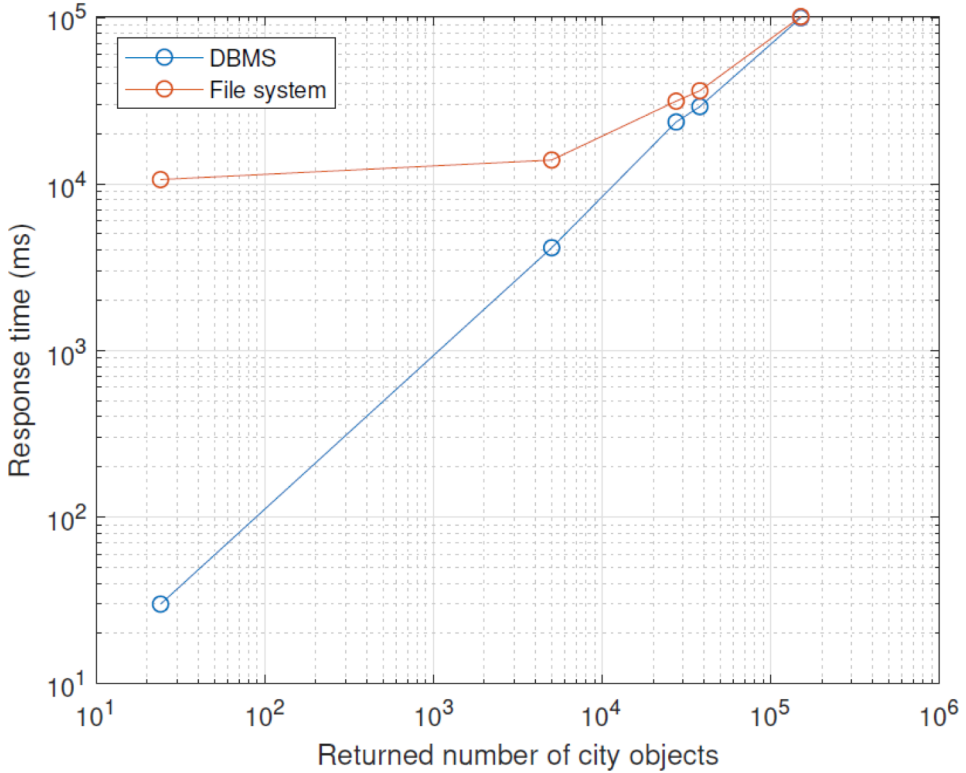
Case	Number of filtered city objects (63386)	Filter ranges %
1	65	0,1
2	321	0,5
3	1798	2,8
4	7848	12,4
5	21400	33,8
6	31496	49,7
7	63386	100



TU Delft

Attribute filtering with *Zurich* dataset and *class* attribute

Case	Number of filtered city objects (198699)	Filter ranges %
1	24	0,01
2	5013	2,5
3	27417	2,8
4	37894	13,8
5	150421	75,7



TU Delft

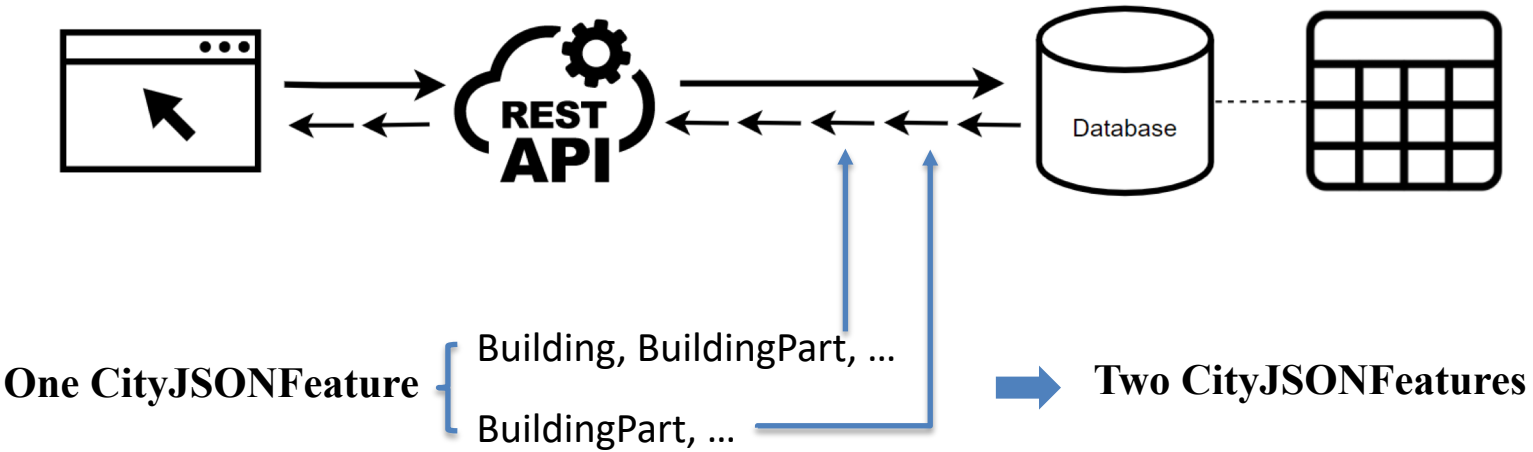
Streaming method

Metric 1: The first response time

Metric 2: The total response time

Metric 3: The correctness rate of the returned *CityJSONFeatrues*

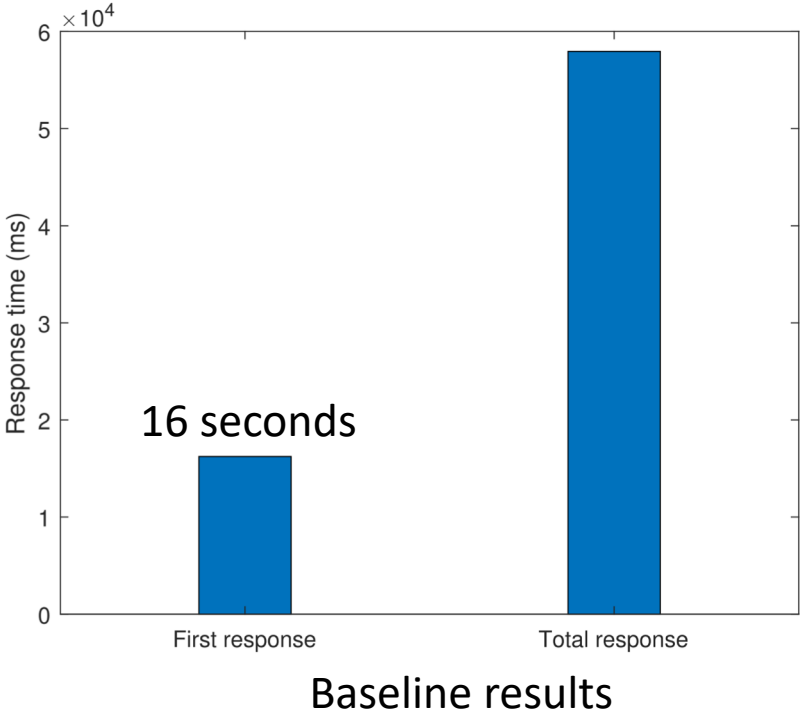
Why error?


REST
API
Database
One CityJSONFeature
Building, BuildingPart, …
BuildingPart, …
Two CityJSONFeatures

Use case : bounding box filtering on *Zurich* dataset

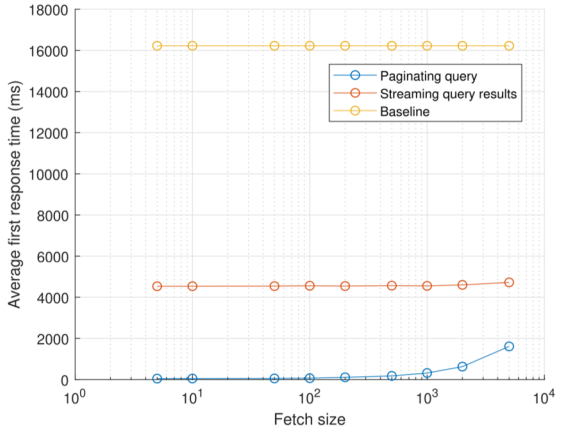
2D bbox of Zurich dataset

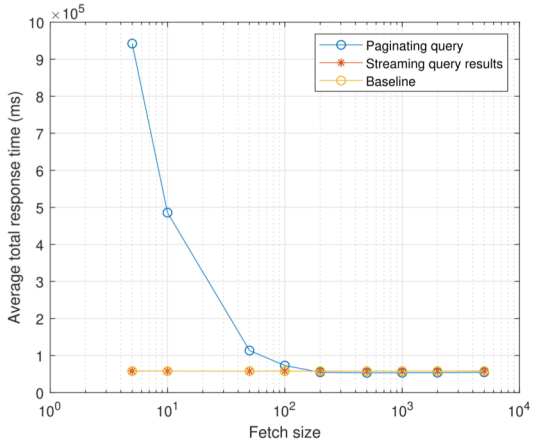
2D bbox of filtering range

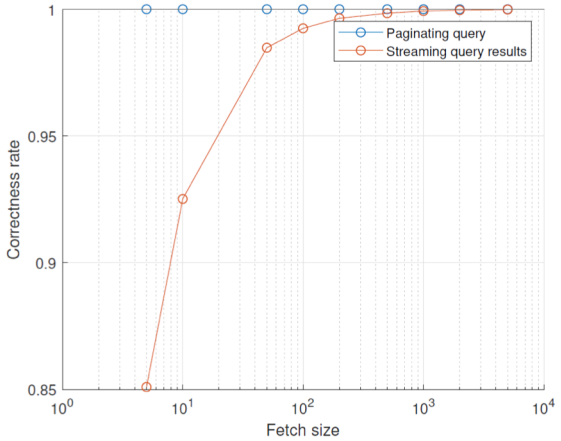


Baseline results

Use case : bounding box filtering on *Zurich* dataset

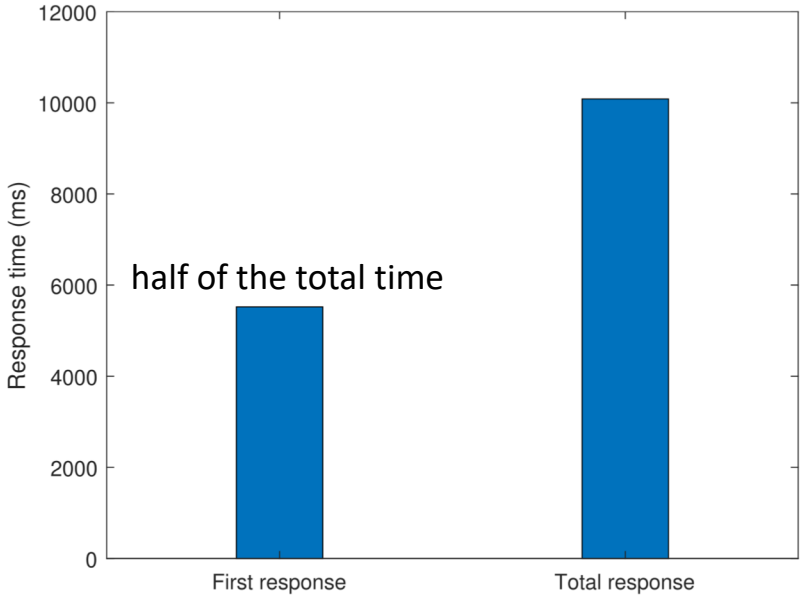






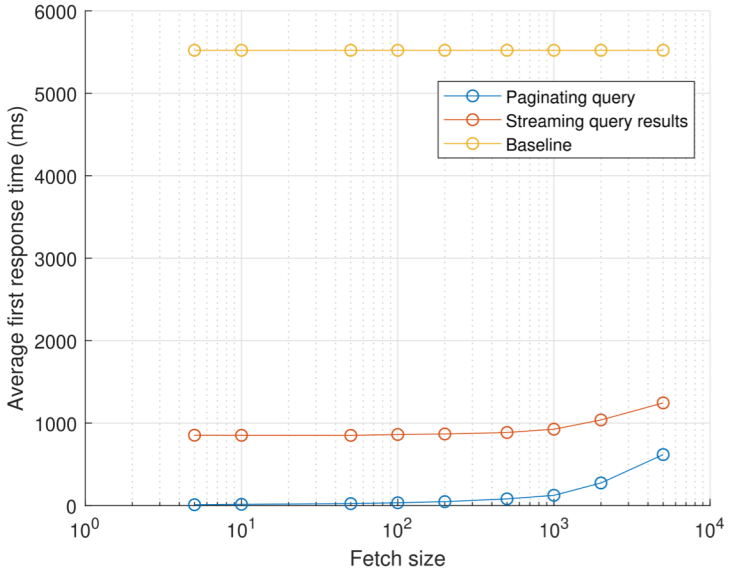
Use case : attribute filtering on *37en2_volledig* dataset

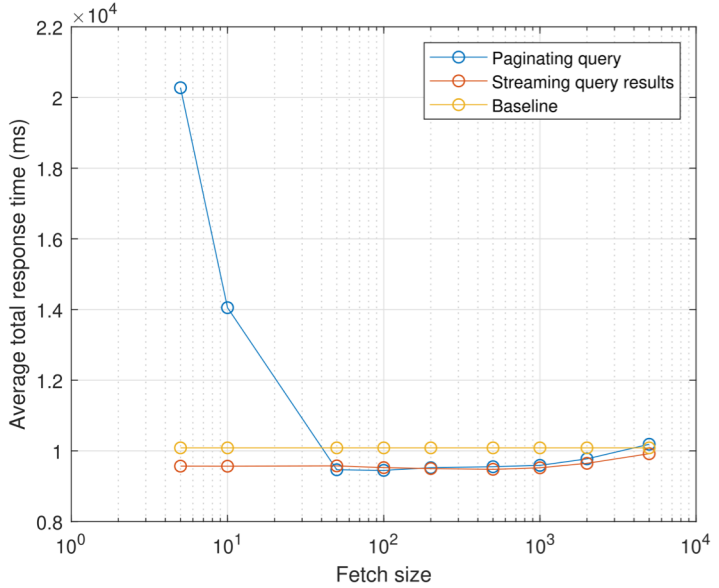
Type	Number	Percentage
LandUse	45822	43.1%
Road	20190	19.0%
Building	20053	18.8%
PlantCover	12527	11.8%
WaterBody	4126	3.9%
GenericCityObject	2337	2.2%
Bridge	1362	1.3%



Baseline results

Use case : attribute filtering on *37en2_volledig* dataset





TU Delft

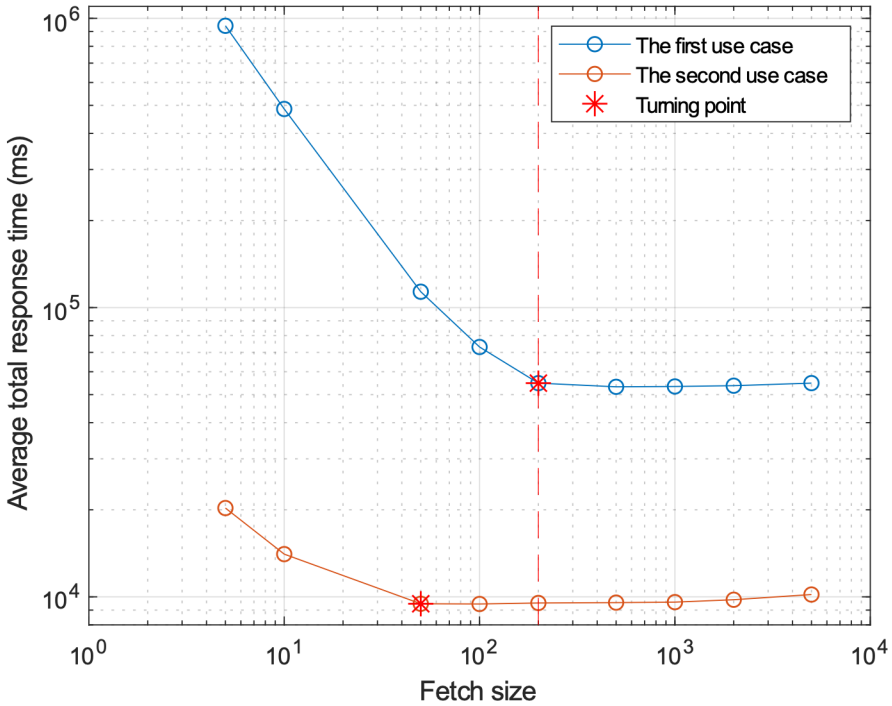
Reasonable fetch size for the first method

- The first response → neglectable

fetch size: 5000	First response time (s)	Number of filtered city objects	Result size (MB)
The 1st case	1,61	136,418	199,4
The 2nd case	0,62	20,053	35,7

- The MAX(turning point1, turning point2 …)
- The fixed fetch size → 200

fetch size: 200	First response time (s)	Total response time (s)
The 1st case	0,112	54,801
The 2nd case	0,047	9,524



TU Delft

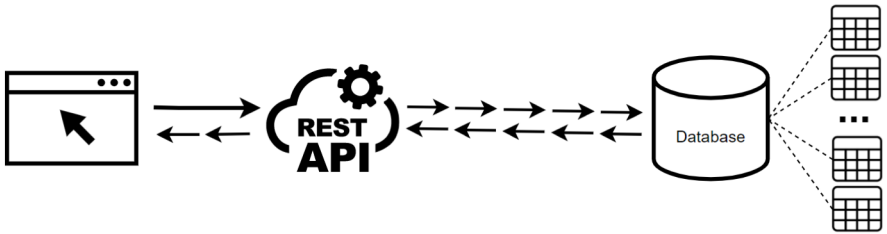
Conclusion and future work

Conclusion

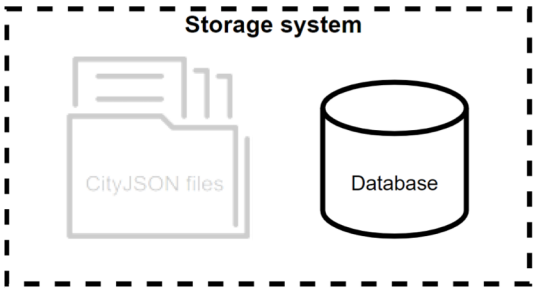
To enable city objects to be directly accessed on the web, a RESTful API for CityJSON with fast **data access**, efficient **data filtering** and **streaming** is implemented.

Conclusion

- The first streaming method with a fixed fetch size of 200
  - fast first response
  - acceptable total response
  - entirely correct results

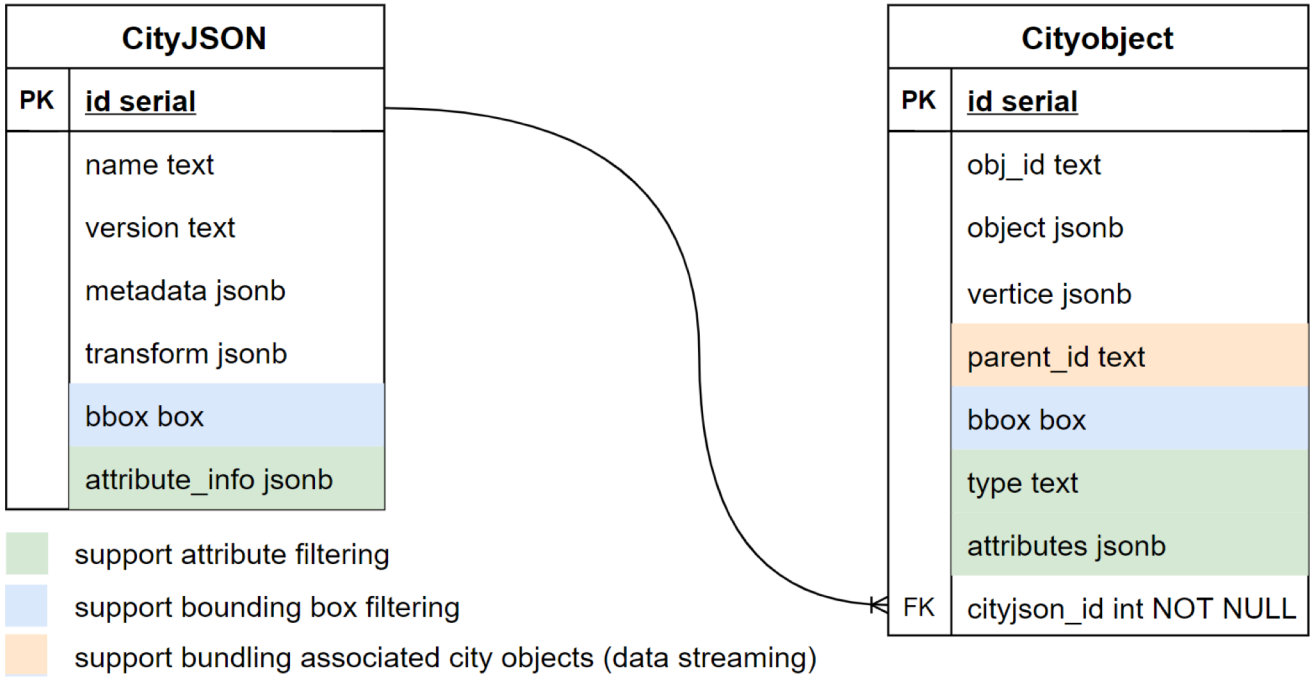


- DBMS can better support the RESTful API
  - built-in query
  - index mechanism
  - extra construction work



TU Delft

Conclusion


CityJSON
PK
id serial
name text
version text
metadata jsonb
transform jsonb
bbox box
attribute_info jsonb
Cityobject
PK
id serial
obj_id text
object jsonb
vertice jsonb
parent_id text
bbox box
type text
attributes jsonb
FK
cityjson_id int NOT NULL
support attribute filtering
support bounding box filtering
support bundling associated city objects (data streaming)

Future work

- Data compression in database systems
- Eliminating the split of CityJSONFeature in the second streaming method
- Dynamically determining optimal fetch size in the first streaming method
- Extending 2D bounding box filtering to 3D
- Fixing the issue of having too large city objects

… …

TU Delft

Thank you!

References

F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889, dec 2015. ISSN 2220-9964. doi: 10.3390/ijgi4042842. URL http://www.mdpi.com/2220-9964/4/4/2842.

CityJSON. CityJSON Specifications 1.0.1, 2020. URL https://www.cityjson.org/specs/1.0.1/{#}city-object.

B. Dukai, R. Peters, T. Wu, T. Commandeur, H. Ledoux, T. Baving, M. Post, V. Van Altena, W. Van Hinsbergh, and J. Stoter. GENERATING, STORING, UPDATING and DISSEMINATING A COUNTRYWIDE 3D MODEL. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 44(4/W1):27–32, 2020. ISSN 16821750. doi: 10.5194/isprs-archives-XLIV-4-W1-2020-27-2020.

G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele. OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0. *OGC Document No. 12-019*, page 344, 2012. URL https://portal.opengeospatial.org/files/?artifact{_}id=47842.

H. Ledoux. Compression factor for a few open CityGML datasets, 2019a. URL https://github.com/cityjson/specs/wiki/Compression-factor-for-a-few-open-CityGML-datasets.

H. Ledoux. CityJSON Specifications 1.0.1, 2019b. URL https://www.cityjson.org/specs/1.0.1/.

H. Ledoux. cityjson_ogcapi, 2020. URL https://github.com/hugoledoux/cityjson{_}ogcapi/blob/master/README.md.

Open Geospatial Consortium. An Introduction to OGC API - Features — OGC e-Learning 2.0.0 documentation, 2020. URL http://opengeospatial.github.io/e-learning/ogcapi-features/text/basic-main.html.

Stackoverflow. postgresql - Streaming data from Postgres into Python - Stack Overflow. URL https://stackoverflow.com/questions/21997978/streaming-data-from-postgres-into-python.

K. Staring. Combination of CityJSON with PostgreSQL, MongoDB and GraphQL. Technical report, 2020. URL https://repository.tudelft.nl/islandora/object/uuid{%}3A7f9209f7-248f-4c93-9ba3-5866655e6040.