

CityJSON in combination with MongoDB, PostgreSQL and GraphQL

Karin Staring - 4952510

P5 presentation

Stelios Vitalis

Linda van den Brink

Balázs Dukai

Willem Korthals Altes

```

▼<gml:boundedBy>
  ▼<gml:Envelope srsName="urn:ogc:def:crs:EPSG::28992" srsDimension="3">
    <gml:lowerCorner>90409.32 435440.44 0.0</gml:lowerCorner>
    <gml:upperCorner>91453.879 436089.946 41.158</gml:upperCorner>
  </gml:Envelope>
</gml:boundedBy>
▼<cityObjectMember>
  ▼<bldg:Building gml:id="{3E04FD3E-CDF4-4EC0-88BA-645257958409}">
    ▼<gen:doubleAttribute name="TerrainHeight">
      <gen:value>1.97</gen:value>
    </gen:doubleAttribute>
    ▼<gen:stringAttribute name="bron_tex">
      <gen:value>UltraCAM-X 10cm juni 2008</gen:value>
    </gen:stringAttribute>
    ▼<gen:stringAttribute name="voll_tex">
      <gen:value>complete</gen:value>
    </gen:stringAttribute>
    ▼<gen:stringAttribute name="bron_geo">
      <gen:value>Lidar 15-30 punten - nov. 2008</gen:value>
    </gen:stringAttribute>
    ▼<gen:stringAttribute name="status">
      <gen:value>1</gen:value>
    </gen:stringAttribute>
    ▼<bldg:boundedBy>
      ▼<bldg:RoofSurface gml:id="UUID_77d73b95-6244-45EE-836A-1DB9FEF06E04">
        ▼<bldg:lod2MultiSurface>
          ▼<gml:MultiSurface>
            ▼<gml:surfaceMember>
              ▼<gml:Polygon gml:id="UUID_89ca3b34-039a-4566-45EE-836A-1DB9FEF06E04">
                ▼<gml:exterior>
                  ▼<gml:LinearRing>
                    ▼<gml:posList srsDimension="3">
                      91081.975 435795.152 11.776 91081.385000000000
                    </gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
            ▼<gml:surfaceMember>
              ▼<gml:Polygon gml:id="UUID_7f0b8bfb-1aec-4c8c-adc5-089">
                ▼<gml:exterior>
                  ▼<gml:LinearRing>
                    ▼<gml:posList srsDimension="3">
                      91081.975 435795.152 11.776 91081.385000000000
                    </gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
          </gml:MultiSurface>
        </bldg:lod2MultiSurface>
      </bldg:RoofSurface>
    </bldg:boundedBy>
  </bldg:Building>

```

CityGML (XML)

CityJSON (JSON)

```

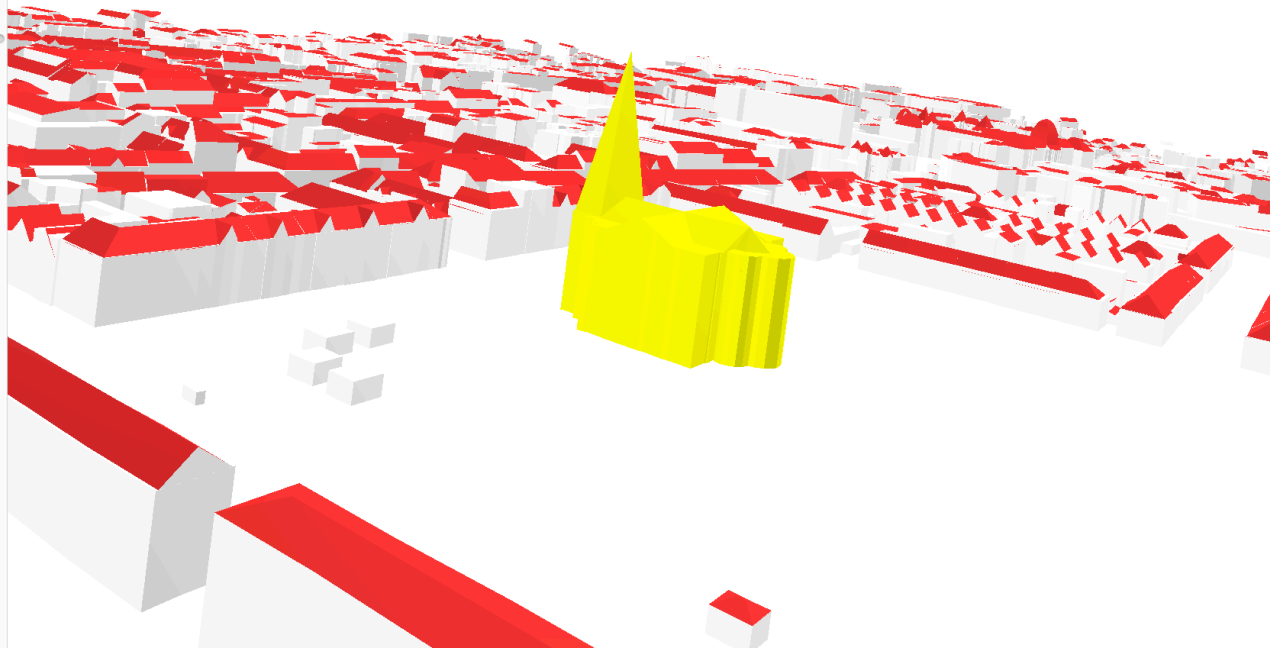
{
  "type": "CityJSON",
  "version": "1.0",
  "metadata": {
    "referenceSystem": "urn:ogc:def:crs:EPSG::28992",
    "geographicalExtent": [
      90409.32,
      435440.44,
      0,
      91453.879,
      436089.946,
      41.158
    ]
  },
  "objects": {
    "3E04FD3E-CDF4-4EC0-88BA-645257958409": {
      "Building": {
        "TerrainHeight": 1.97,
        "bron_tex": "UltraCAM-X 10cm juni 2008",
        "voll_tex": "complete",
        "bron_geo": "Lidar 15-30 punten - nov. 2008",
        "status": "1"
      },
      "geometry": [
        {
          "type": "MultiSurface",
          "boundaries": [
            {
              "type": "RoofSurface",
              "lod2MultiSurface": {
                "MultiSurface": {
                  "surfaceMembers": [
                    {
                      "type": "Polygon",
                      "id": "UUID_89ca3b34-039a-4566-45EE-836A-1DB9FEF06E04",
                      "exterior": [
                        {
                          "type": "LinearRing",
                          "posList": [
                            {
                              "x": 91081.975,
                              "y": 435795.152,
                              "z": 11.776
                            },
                            {
                              "x": 91081.385,
                              "y": 435000.0,
                              "z": 0.0
                            }
                          ]
                        }
                      ]
                    }
                  ]
                }
              }
            }
          ]
        }
      ]
    }
  }
}

```

Search

- Building (UUID_8e26c0bb-68b9-4f52-a5cb-cc054871207f)
- Building (UUID_0d9c0284-a859-4193-9528-a832f575fd63)
- Building (UUID_a41d83a6-c179-4bf3-af3b-6228e6df6692)
- Building (UUID_a2cdbc2b-37ff-4113-bb44-eadcc6b883c2)
- Building (UUID_9d9b51cd-fdd1-40f9-ba4a-dfeec40d6e44)
- Building (UUID_cc578dcc-449d-48eb-8a2f-8f01469a5ac5)
- Building (UUID_69a2aef9-a6e6-4755-90b2-0bdb3bdb7006)
- Building (UUID_141905c5-1692-4ca4-ae3e-4348633a0d0d)
- Building (UUID_0f7a2b6b-32e5-4730-bad4-0bb470727f7c)
- Building (UUID_93fad353-113a-4e7c-b9a9-132e77c46b89)
- Building (UUID_87adc5ec-5a07-471a-9bf9-53bad975ea56)
- Building (UUID_56682be2-c7bc-4471-900c-9cbbc59b0236)
- Building (UUID_b26f11d2-b79d-439f-9f04-b96e430fff77)
- Building (UUID_16ce0bfa-8c47-4164-a058-d1945e69f54b)
- Building (UUID_5d55beb8-c19d-4160-b265-fcc70f656659)
- Building (UUID_0ec53f1c-89bb-458e-8d4b-3e7d2c646958)
- Building (UUID_f8ddebdf-7d35-4a00-8fca-2dfaf31d3b3d)
- Building (UUID_ab5ad8f8-1fdc-4bcd-97ab-dc947c4b8819)
- Building (UUID_53dacc14-19f7-4a77-baae-622541c4ca7e)
- Building (UUID_4999bcb3-67cd-4564-b6bd-f190faa2e1f1)
- Building (UUID_1a624476-c817-4f26-ad50-04350826582b)
- Building (UUID_8f6c21ca-f52e-4ce8-ac1f-5eda30832c34)
- Building (UUID_d889fad3-c551-4814-b63c-e9c26b878868)
- Building (UUID_d956563-f6bb-47bf-9d5b-70766d407a5b)
- Building (UUID_59eb9491-33b2-4a73-8ac5-9f774ab6ae59)
- Building (UUID_f07a8b86-4ab8-45fe-bfa2-b3f6e0b0a2ae)
- Building (UUID_a116e149-85bc-4126-82a0-0568c588cf65)
- Building (UUID_b834c976-f989-45b0-a4b2-266ee6dfbdf2)
- Building (UUID_749d3067-7303-4341-a969-66dadef02f2)

Attribute	Value
OS	1141
TILE	3367218.391959628 5806639.061027913
DFIHO	0
DFORM	0
FOKHO	31320
FOKHW	5807393170
FOKHW	3368003260



MongoDB = NoSQL/document database

PostgreSQL = Relational database

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with navigation options like '5 DBS', '7 COLLECTIONS', and 'CityJSON'. The main area displays a list of documents under the 'CityJSON.CityObjects' collection. Each document is a JSON object with fields like '_id', 'type', 'attributes', 'geometry', and 'metadata_id'. The 'type' field is consistently 'Building'.

```
{ "_id": "00138BE2-F4EB-4B9E-A6F0-77EECB4E9D56", "type": "Building", "attributes": Object, "geometry": Array, "metadata_id": "metadata_delfshaven" }
```

```
{ "_id": "0015085B-C2E8-43AE-98DF-6A0E807A9DAA", "type": "Building", "attributes": Object, "geometry": Array, "metadata_id": "metadata_delfshaven" }
```

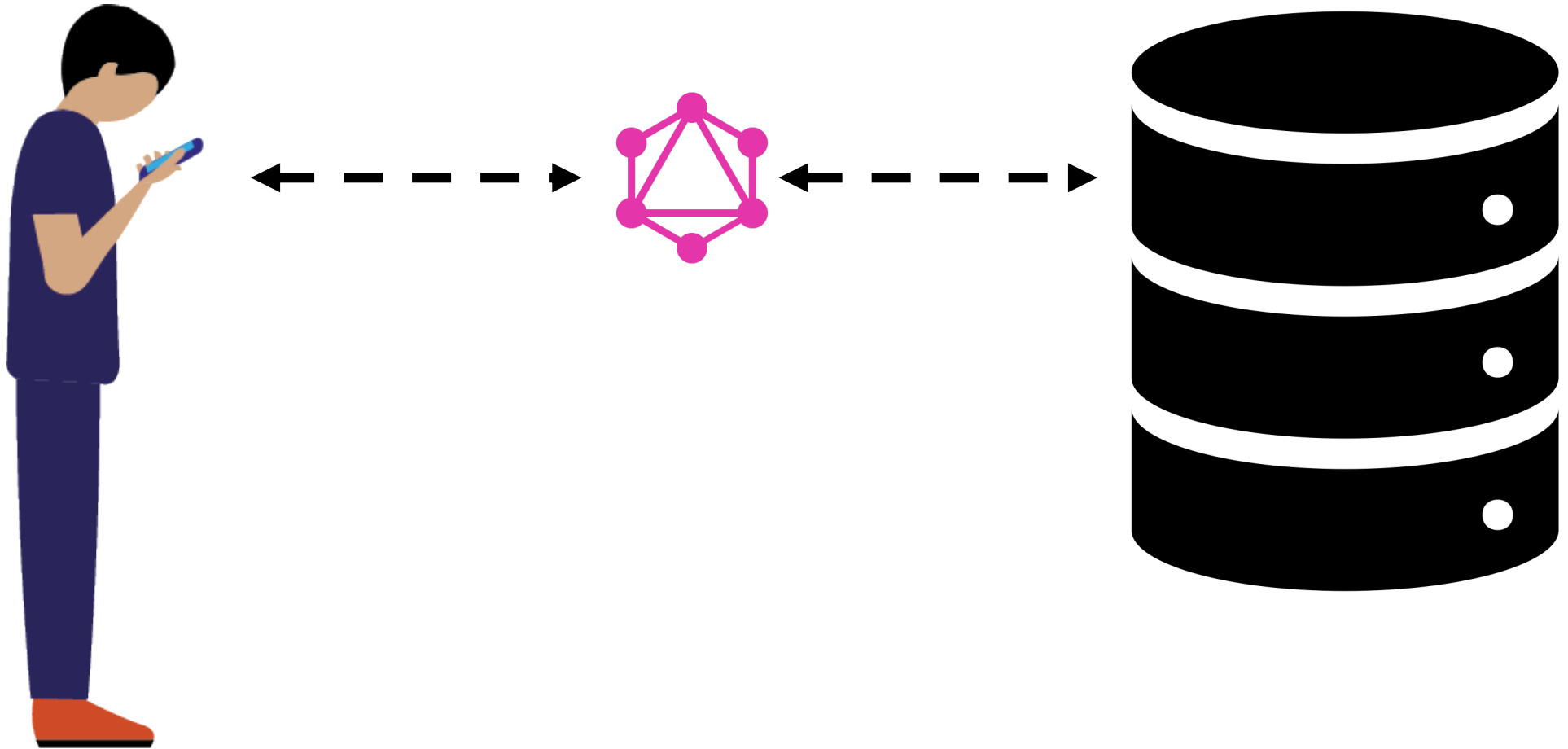
```
{ "_id": "0031458C-0F2F-40B4-B170-90D6305F9F78", "type": "Building", "attributes": Object, "geometry": Array, "metadata_id": "metadata_delfshaven" }
```

```
{ "_id": "008F0D4C-BFDE-4419-B7B4-D4C489234FA4", "type": "Building", "attributes": Object, "geometry": Array, "metadata_id": "metadata_delfshaven" }
```

The screenshot shows the pgAdmin interface. A SQL query is entered in the Query Editor: `select * from cityjsondb.city_object;`. The results are displayed in a table with columns 'id' and 'object jsonb'. The table contains 41 rows of data, each representing a building record with a unique ID and a JSON object.

id	object jsonb
1	{ "type": "BuildingPart", "parents": ["UUID_c9884c4e-1cac-4336-9ff8-6679ebfdfe3"] }
2	{ "type": "Building", "children": ["UUID_93fc5bae-4446-4336-9ff8-6679ebfdfe3"] }
3	{ "type": "BuildingPart", "parents": ["UUID_ba0bb815-5276-4e35-b4c1-878cbf6ba934"] }
4	{ "type": "Building", "children": ["UUID_a4a09780-153f-4385-ad19-3a92a6c4eec4"] }
5	{ "type": "BuildingPart", "parents": ["UUID_bb1835bc-7437-453f-ac08-885de0503aaa"] }
6	{ "type": "Building", "children": ["UUID_053775db-7203-44df-9ae9-a5e4803c0696"] }
7	{ "type": "BuildingPart", "parents": ["UUID_3fd9ef25-2e40-4a1d-bcbe-1c63d481ebd0"] }
8	{ "type": "Building", "children": ["UUID_a9a94ab7-add6-41ad-874b-90675f224317"] }
9	{ "type": "BuildingPart", "parents": ["UUID_7a813bc5-8c88-4852-9eda-956504857976"] }
10	{ "type": "BuildingPart", "parents": ["UUID_48475dba-fa80-4ef2-be06-ffdb431dad58"] }
11	{ "type": "Building", "children": ["UUID_ac25c46d-8200-4821-a2c8-0568ae28147a"] }
12	{ "type": "BuildingPart", "parents": ["UUID_e9b3c89c-6805-40da-91b6-8d76a593d77a"] }
13	{ "type": "BuildingPart", "parents": ["UUID_900fe536-8e9f-4307-ac30-6b9a964df9a8"] }
14	{ "type": "BuildingPart", "parents": ["UUID_eed9bf7a-2343-4f81-bc9a-2e8e46b2aa9a"] }
15	{ "type": "Building", "children": ["UUID_336558af-95a0-409d-b5e4-47f8daa6109d"] }
16	{ "type": "BuildingPart", "parents": ["UUID_d122008b-4665-47c5-ba95-b0a38214ff20"] }
17	{ "type": "Building", "children": ["UUID_9b6ca973-950d-4418-bb77-d298b6bfc0ca"] }
18	{ "type": "BuildingPart", "parents": ["UUID_764c723d-e5b7-4cd5-832f-e0da1f2050bc"] }
19	{ "type": "BuildingPart", "parents": ["UUID_a30fec5f-0718-40d2-b4b4-ec722e912780"] }
20	{ "type": "BuildingPart", "parents": ["UUID_70398cf8-2ea3-4d93-8a81-ff65af8a00f6"] }
21	{ "type": "Building", "children": ["UUID_b605899f-a402-4e9e-8698-6c7e54e2596a"] }
22	{ "type": "BuildingPart", "parents": ["UUID_4e6de9fd-d3f9-4178-b9a3-3d5b2fed4739"] }
23	{ "type": "Building", "children": ["UUID_5708524e-57ce-4cff-af31-4dfa5fa7d471"] }
24	{ "type": "BuildingPart", "parents": ["UUID_2aa1c7fc-8e2d-4f1a-81a9-9e7c46e93273"] }
25	{ "type": "Building", "children": ["UUID_f1ee2f23-94ba-46fe-9d10-9739de1688db"] }
26	{ "type": "Building", "children": ["UUID_3bd986d-bf70-447d-9850-cf6eb86d0f5"] }
27	{ "type": "BuildingPart", "parents": ["UUID_0106d15e-1625-4ebf-86a4-4e3ed90d65b0"] }
28	{ "type": "Building", "children": ["UUID_c510f7c0-62f1-4e91-8b63-59daadf14449"] }
29	{ "type": "BuildingPart", "parents": ["UUID_eb7cb018-ab1b-49c4-8e73-e3033ca4495e"] }
30	{ "type": "Building", "children": ["UUID_7bd33b0e-bd9a-47e1-9d97-69c95baf1a0b"] }
31	{ "type": "BuildingPart", "parents": ["UUID_3e2ffc71-a595-4192-b1d9-67a59974dc85"] }
32	{ "type": "Building", "children": ["UUID_97e7cddf-a812-49d5-bf44-2f7718ed8165"] }
33	{ "type": "BuildingPart", "parents": ["UUID_f2a9063f-f7e8-4158-9102-1e2889a49b31"] }
34	{ "type": "Building", "children": ["UUID_c8e746d4-ee7b-44bf-a571-f3929219fa78"] }
35	{ "type": "BuildingPart", "parents": ["UUID_daaca1fd-82f0-461b-94ef-8dda17c39d9b"] }
36	{ "type": "Building", "children": ["UUID_0a9bf9e3-4540-4583-a46e-6d63244c7801"] }
37	{ "type": "BuildingPart", "parents": ["UUID_5b4a1acc-c9b3-4497-9d2f-76d4f426f7e4"] }
38	{ "type": "Building", "children": ["UUID_3be91a76-bf44-4dd7-85a6-c8bd497f4f8d3"] }
39	{ "type": "Building", "children": ["UUID_a464f5a2-ef6a-44b4-a519-975da4a2bfff"] }
40	{ "type": "BuildingPart", "parents": ["UUID_0b47b43b-509c-47b3-9064-1c5044ae7e9c0"] }
41	{ "type": "Building", "children": ["UUID_1e31c690-b11e-430c-afcc-1ae5022e7fa0"] }

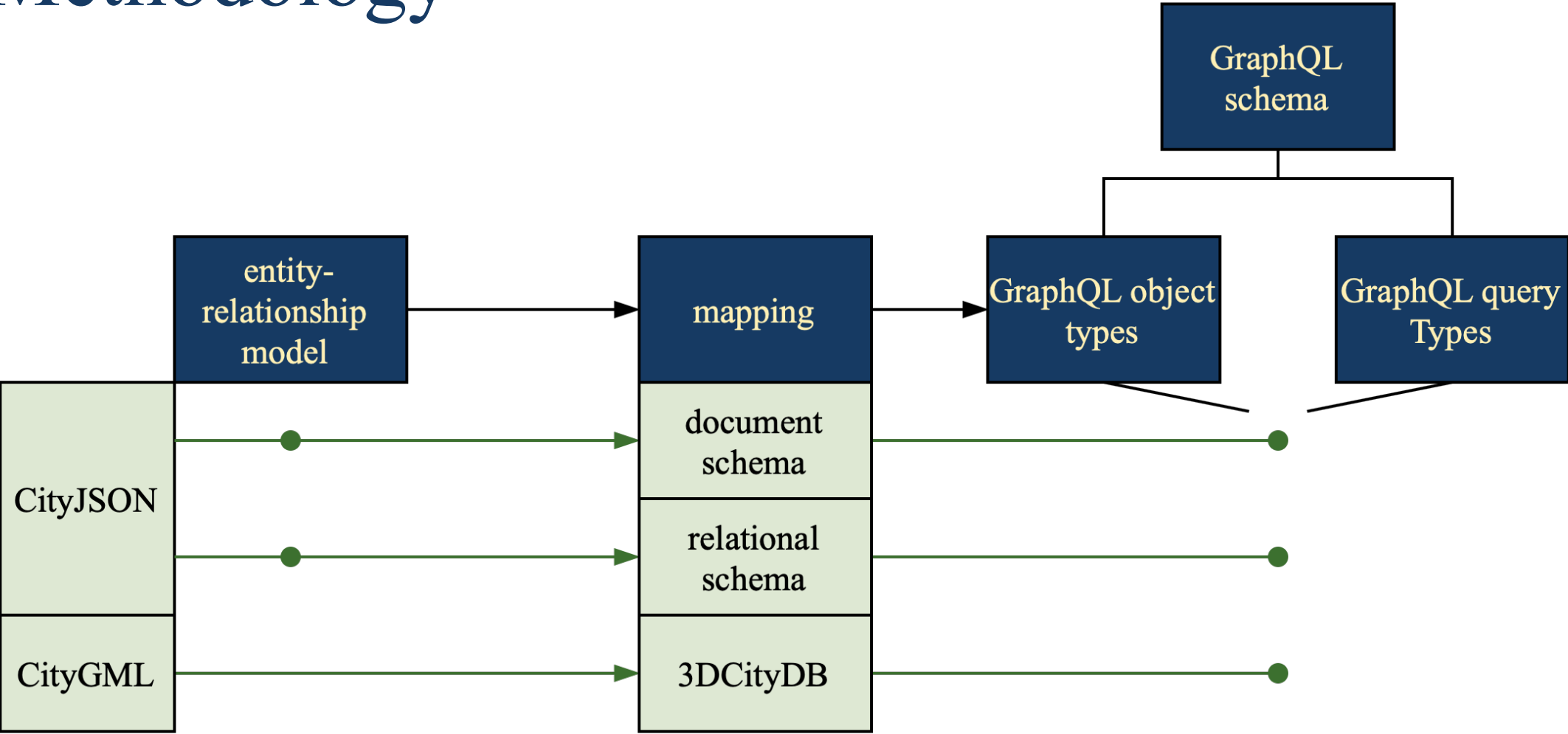
Use case



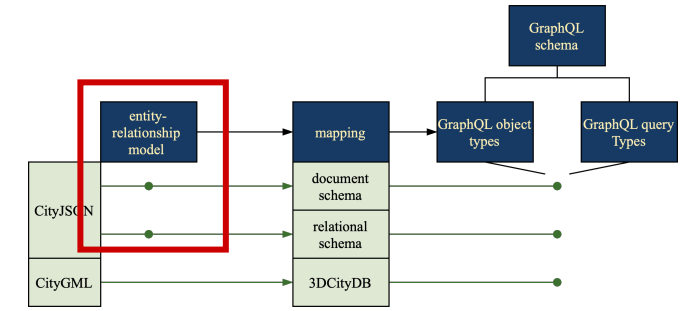
Research

How suitable are MongoDB and PostgreSQL for the storage and querying of CityJSON using GraphQL?

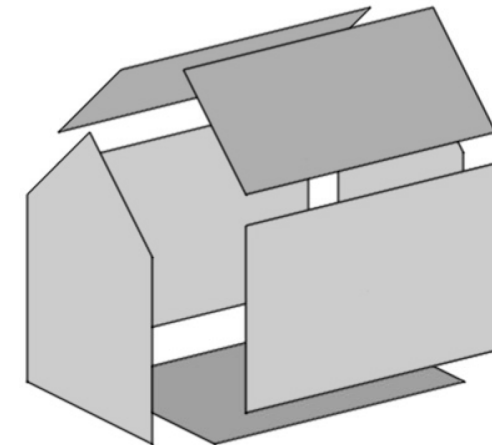
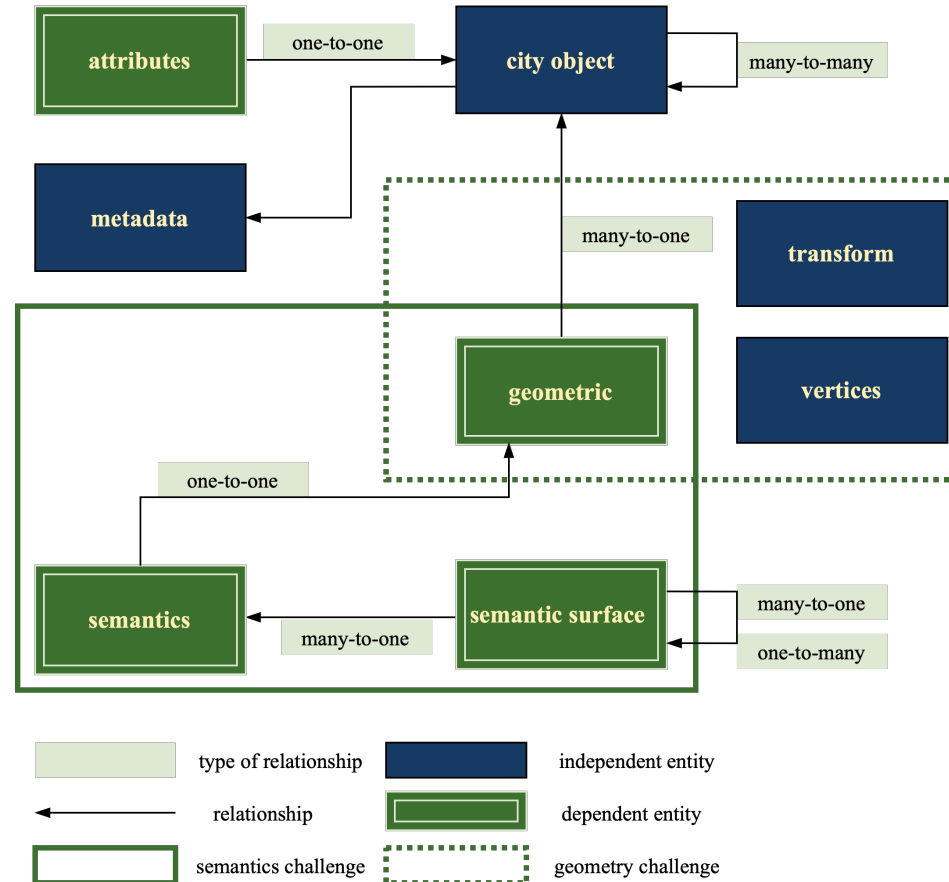
Methodology



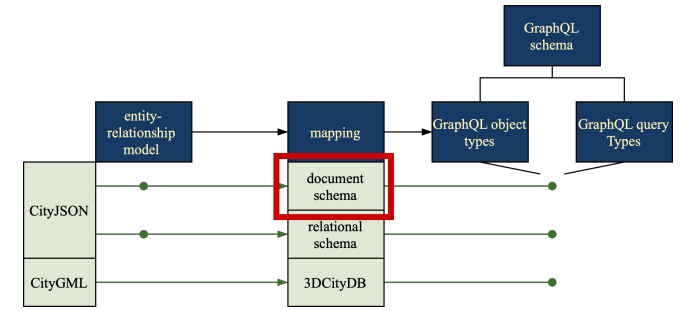
Entity-relationship model



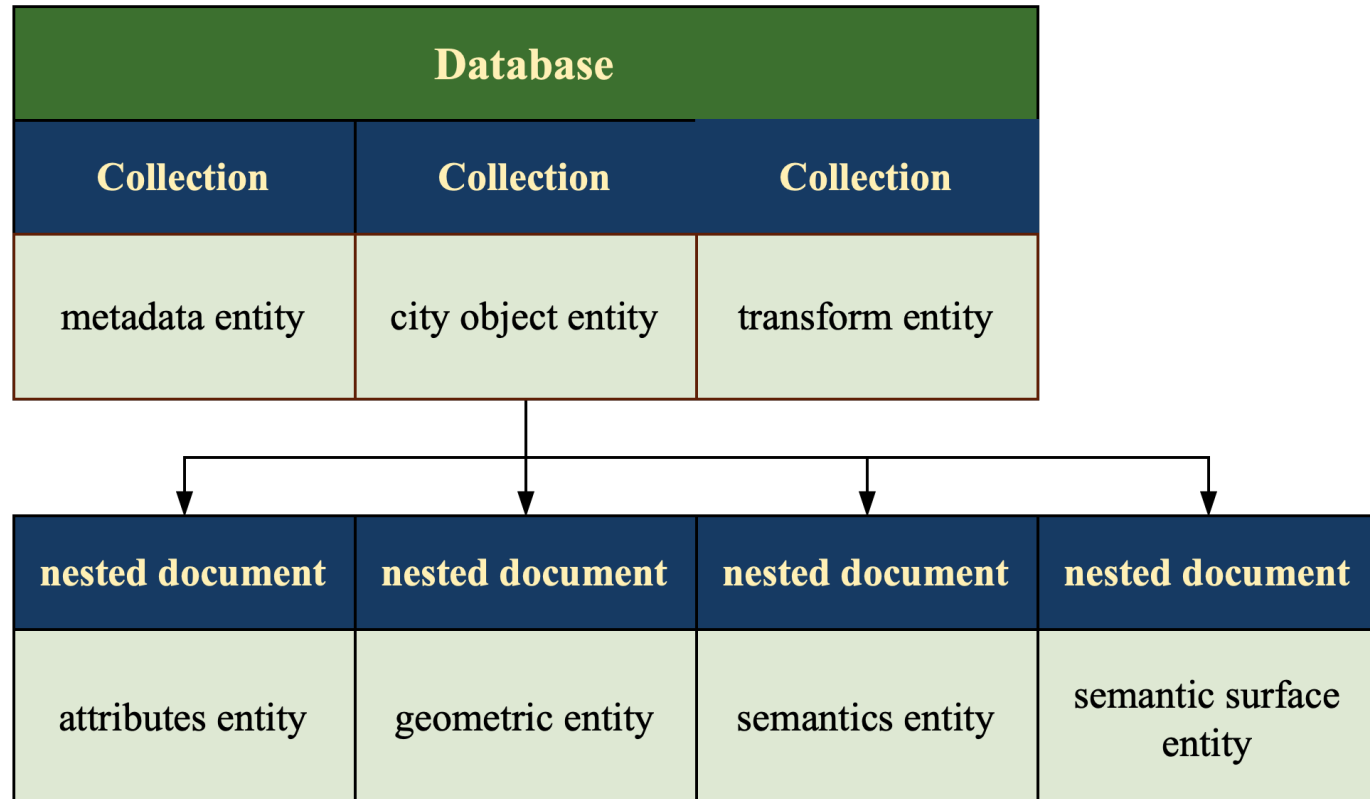
- Dependent/ independent entities
- Many-to-many relationship
- Geometry challenge
- Semantics challenge
- JSON



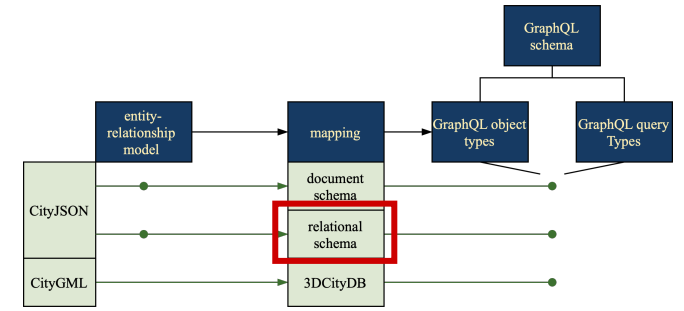
Document schema



- Dependent/ independent entities
- Many-to-many relationship
- Geometry challenge
- Semantics challenge



Relational schema



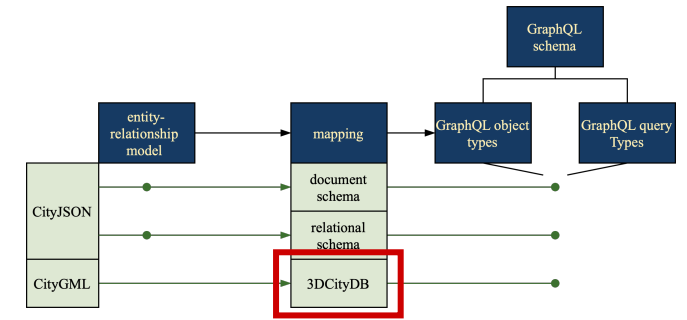
- Dependent/ independent entities
- Many-to-many relationship
- Geometry challenge
- Semantics challenge

Database			
metadata table	city_object table	transform table	parents_children table
metadata entity	city object entity	transform entity	
id [text]	id [text]	id [text]	parents_id
object [jsonb]	object [jsonb] attributes [jsonb] metadata_id ...extra geometry	object [jsonb]	children_id
geometry table	surfaces table	semantic_surface table	
geometric entity	semantics entity	semantic surface entity	
id [sequence]	id [sequence]	id [sequence]	
object [jsonb] city_object_id	geometry [polygonz] solid_num [int] shell_num [int] surface_num [int] geometry_id semantic_surface_id city_object_id	object [jsonb] city_object_id children parent	

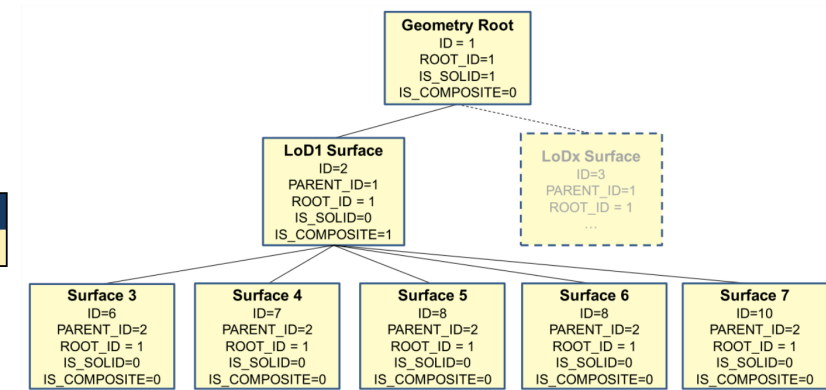
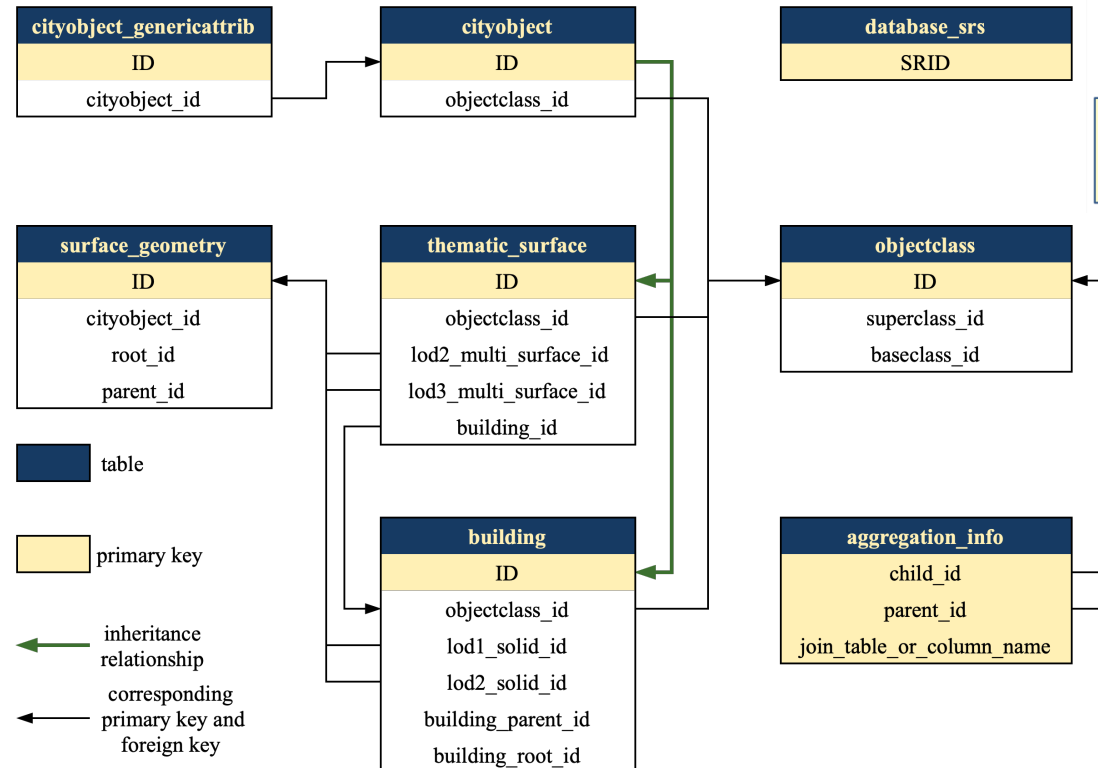
The semantics entity is unnecessary, because the geometry is split into surfaces. These surfaces are directly linked to the objects of the semantic surface entity.

table
 entity
 primary key

3DCityDB

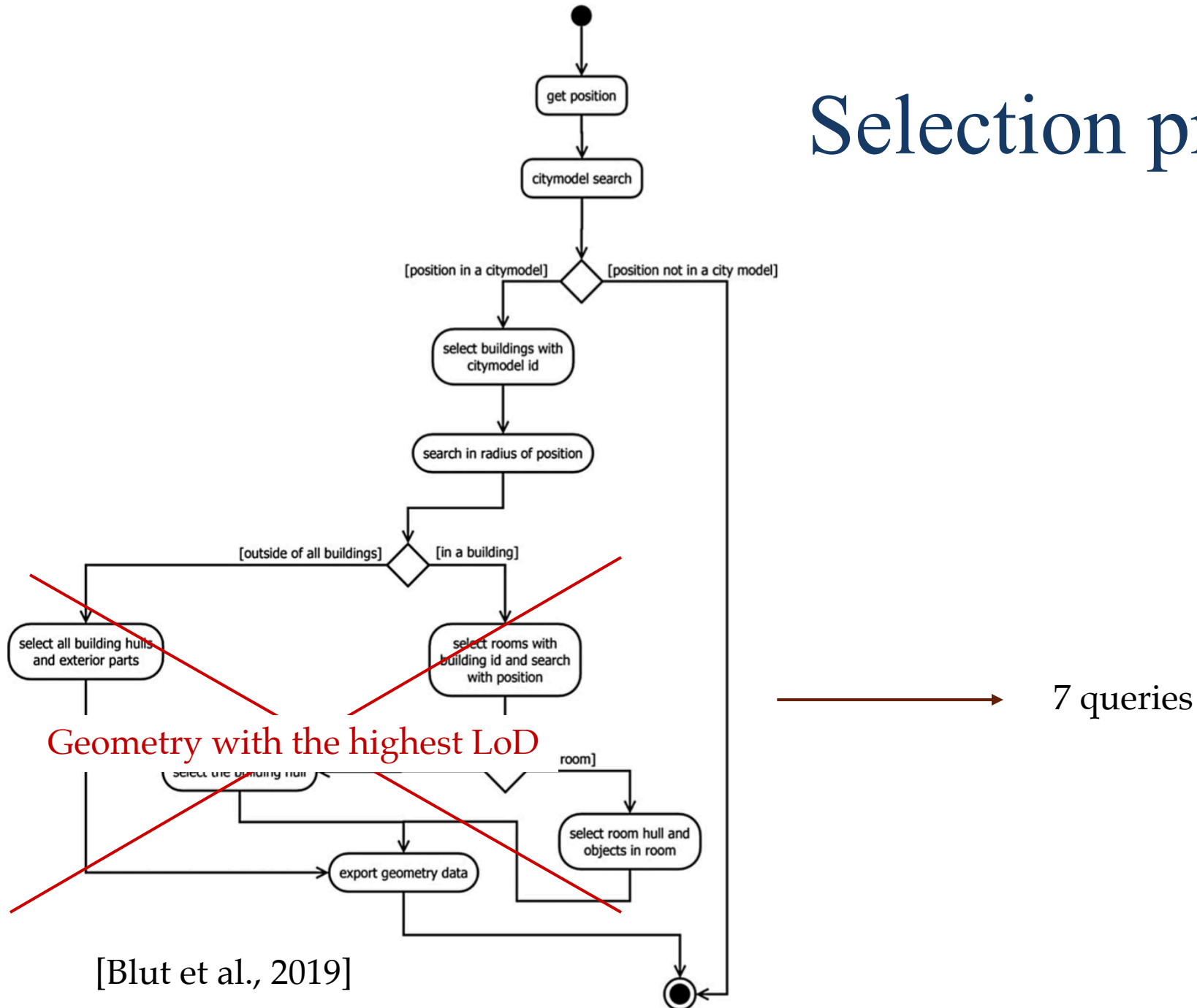


- Inheritance relationship
- Generic attributes
- Geometries



[Kolbe, 2019]

Selection process

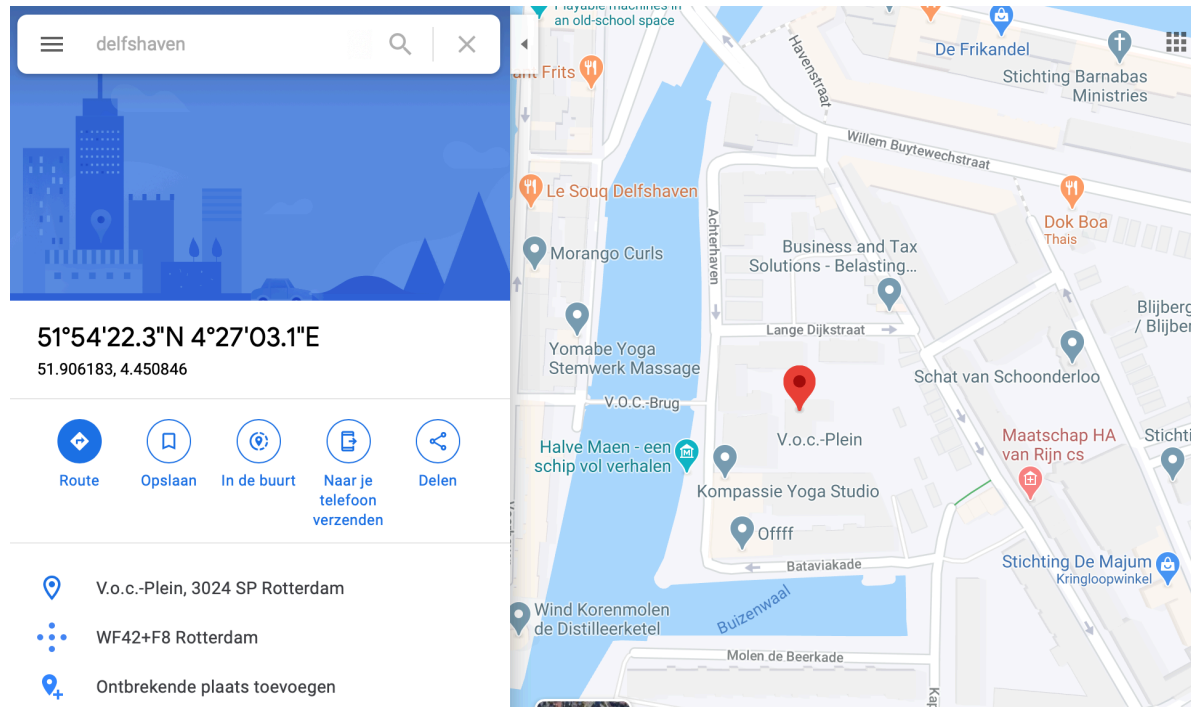


The 7 queries

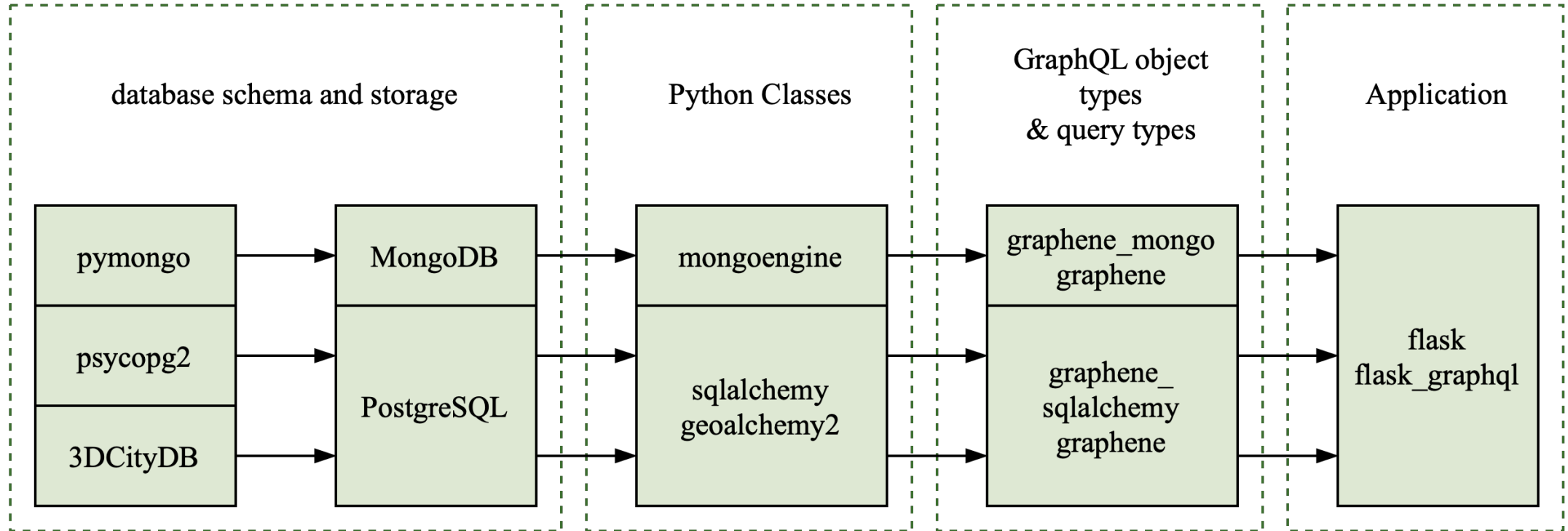
query number and name	argument	operations	filter	returned data
1: location	position with the fields lat, long and alt			location with the fields latitude, longitude and altitude
2: citymodel	position with the fields lat, long and alt	intersection		city model ID
3: radius100	position with the fields lat, long and alt	within 100 meter	city object type = Building	city object IDs
4: inside	position with the fields lat, long and alt	intersection	city object type = Building	city object ID
5: maxlod	city object ID			ID of the geometry with the highest LoD
6a: cityobjects	city object ID		ID-based	return the ID of the city object
6b: cityobjects	city object ID		ID-based	return the ID and attributes of the city object
6c: cityobjects	city object ID		ID-based	return the ID and geometries of the city object
7: surfaces	surface ID or other		ID-based or other	return the surface with the related semantic surface object

Data

Denhaag	2.6 MB	EPSG:4715	parent-child relationship	
Delfshaven	1.4 MB	EPSG: 28992		
Potsdam	15.7 MB	EPSG: 25833	multiple LoDs	presentLoDs



Software

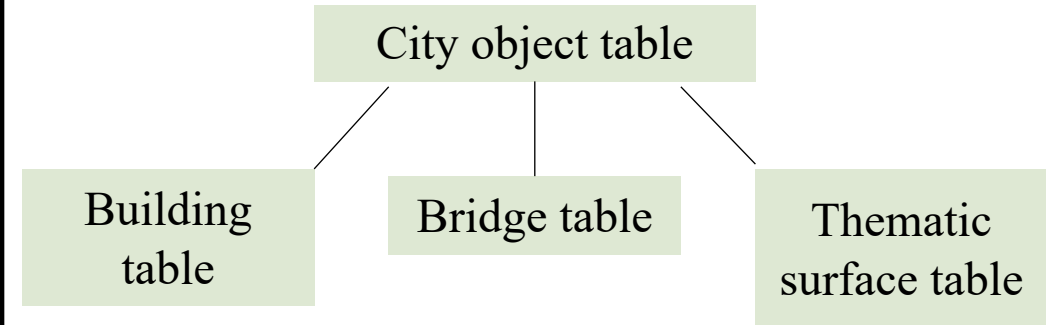


CityJSON in PostgreSQL

	id [PK] text	object jsonb	attributes jsonb
1	UUID_93f...	{"type": "Building..."}	{"Geomtype": 1, "creation..."}
2	UUID_c98...	{"type": "Building", "p..."}	{"class": "BB01", "Region"...
3	UUID_a4a...	{"type": "Building..."}	{"Geomtype": 1, "creation..."}
4	UUID_ba0...	{"type": "Building", "p..."}	{"class": "BB07", "Region"...

```
{ cityobjects {
  id
}}
```

CityGML in PostgreSQL with 3DCityDB



```
{ cityobjects {
  building {
    id
  }}}
```


CityJSON in PostgreSQL

CityJSON in MongoDB

	<code>"presentLoDs": {"2.0": 145865}</code>
one city object – multiple tables	one city object – one document
reference transformations	no reference transformations
Maximum field size = 1GB	Maximum document size = 16MB

MultiSurface :

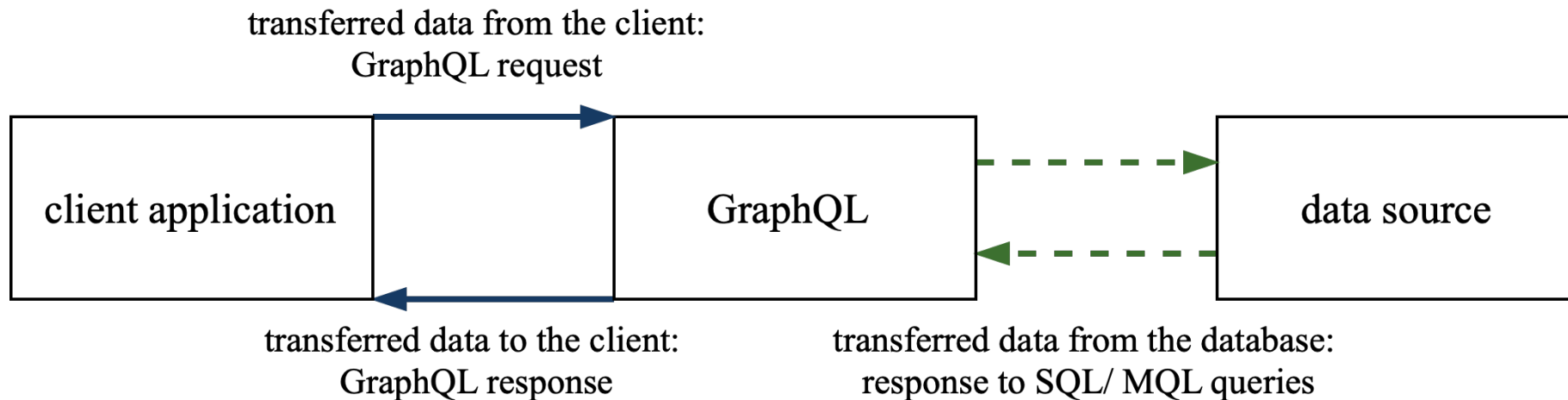
```
boundaries = ListField ( ListField ( ListField ( IntField ( ))) ) )
```

MultiSolid :

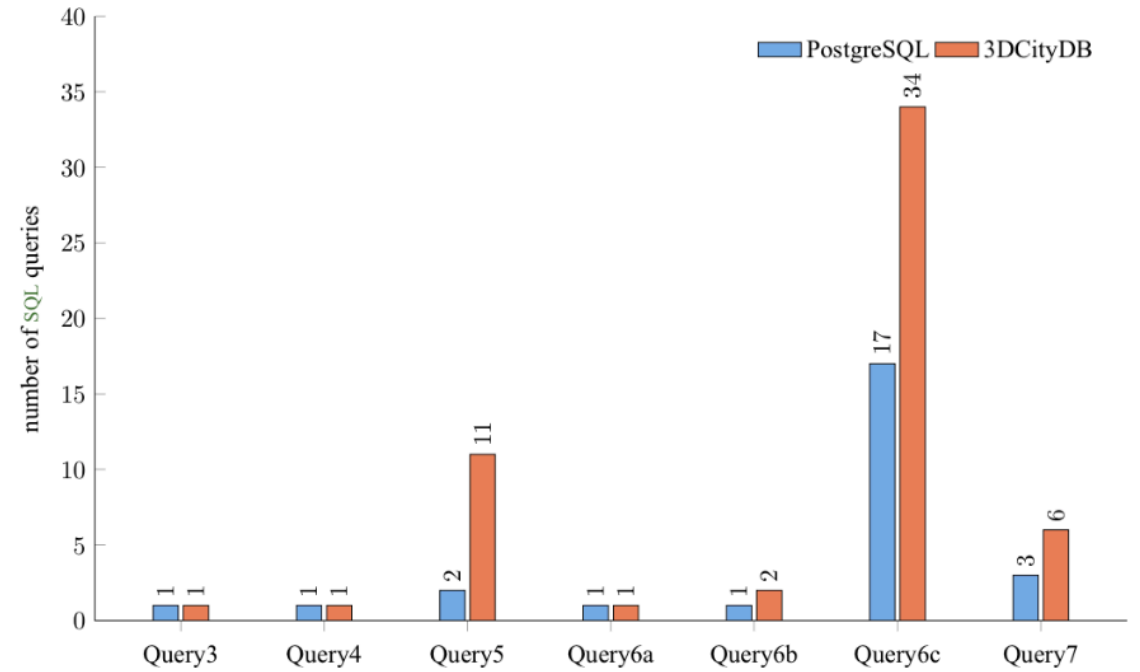
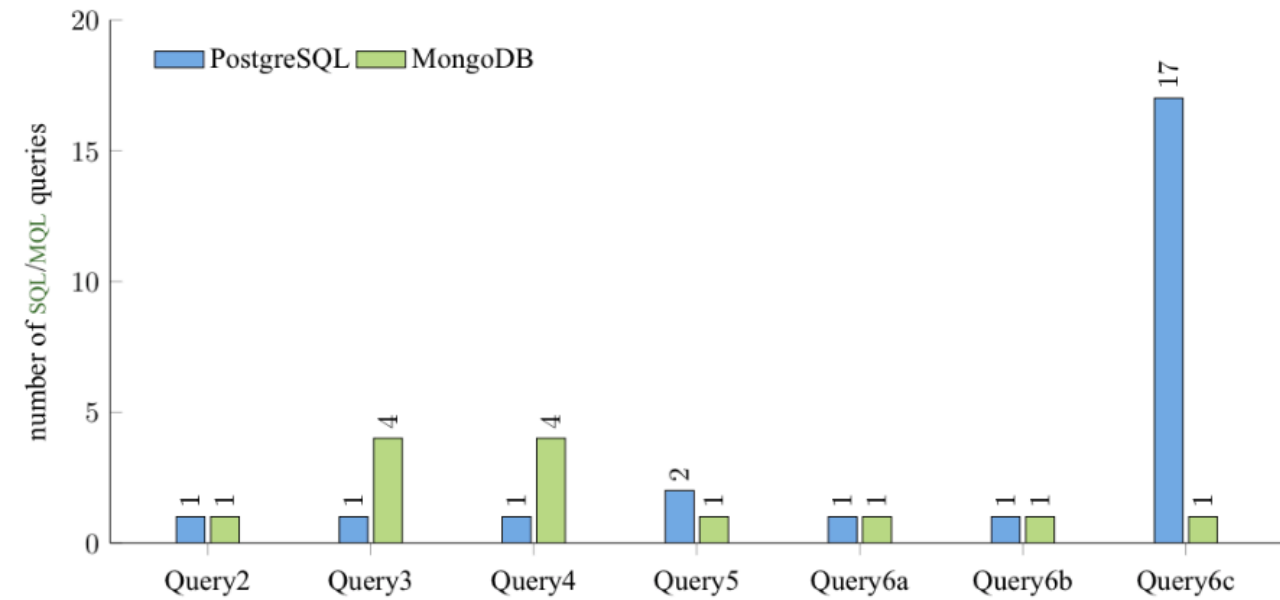
```
boundaries = ListField ( ListField ( ListField ( ListField ( ListField ( IntField ( ))) ) ) ) )
```

Experiments

- Retrieval times
- Request and response sizes
- Over-fetching



Results



Results

- More queries lead to higher retrieval times
 - Possible due to ORM/ODM → not using joins
- More overfetching for MongoDB and 3DCityDB than PostgreSQL
 - Possible due to ORM/ODM → not selecting fields

How suitable are MongoDB and PostgreSQL for the storage and querying of CityJSON using GraphQL?

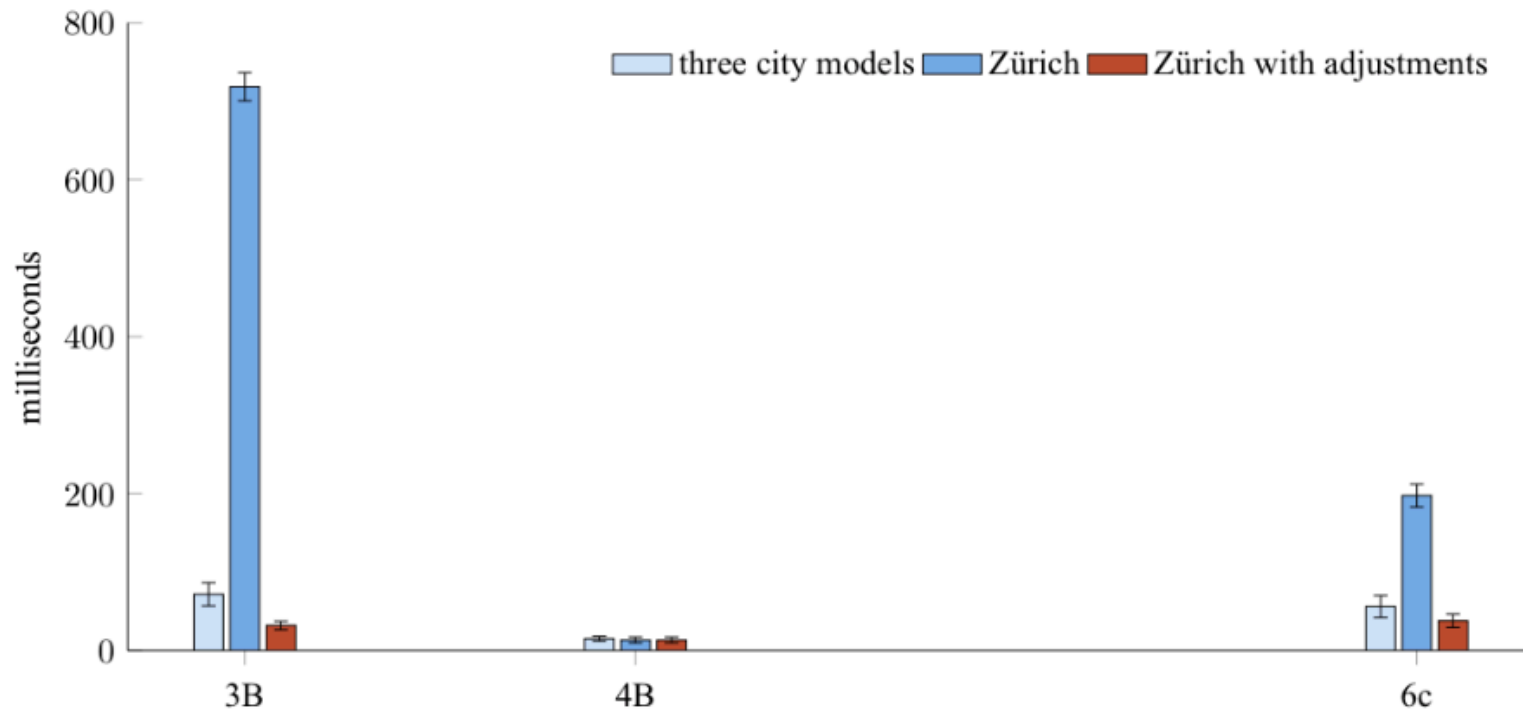
- No real signs yet that MongoDB and PostgreSQL are not suitable
 - Vertices list
 - Attribute presentLoDs
 - reference transformations
- Integration of JSON
 - Attributes are mapped more flexibly → less tables → less queries/joins → lower retrieval times
 - Support for fields with varying data types
 - + support to map the hierarchy of arrays
 - - GraphQL is less flexible
 - Querying a JSON attribute -→ higher retrieval times = not investigated

Recommendations

- Implement the SQL/MQL queries as efficient as possible without being connected to GraphQL first
 - + better understanding database performance
 - + better understanding abilities GraphQL
- A more general understanding of the suitability for all use cases could be provided with a framework that tests more types of queries.
 - Aggregate queries
 - Queries on GeoJSON objects in MongoDB
 - Queries on JSON attributes in PostgreSQL and MongoDB

Recommendations

	Three datasets together	Zürich
Size (MB)	19.7	292.8
number of objects	6258	198699



Questions?

Spatial queries per implementation

Use one city model instead of multiple

Store convex hull as GeoJSON object

PostgreSQL		MongoDB
supports many reference systems geometrical spatial operations on 2D and 3D geometries geographical spatial operations on 2D geometries		supports GeoJSON objects in 2D with WGS84 as global reference system
CityGML in PostgreSQL with 3DCityDB	CityJSON in PostgreSQL	CityJSON in MongoDB
envelope transformed in the query (in the database)	convex hull transformed in the query (in the database)	global convex hull transformed before query
index on envelope	additional indexes: index on convex hull index on global convex hull	hierarchy of arrays transformed in query (outside the database)