

14TH ITU ACADEMIC CONFERENCE

ITU KALEIDOSCOPE
ACCRA 2022

The knowledge graph as interoperability foundation for Augmented Reality

7-9 December 2022
Accra, Ghana





Alexandra Rowland

Kadaster, University of Twente

Session 2 – Augmented reality systems:
design and implementation

Paper S2.3



ITUKALEIDOSCOPE
ACCRA2022



Erwin Folmer

Kadaster, University of Twente

Session 2 – Augmented reality systems:
design and implementation

Paper S2.3

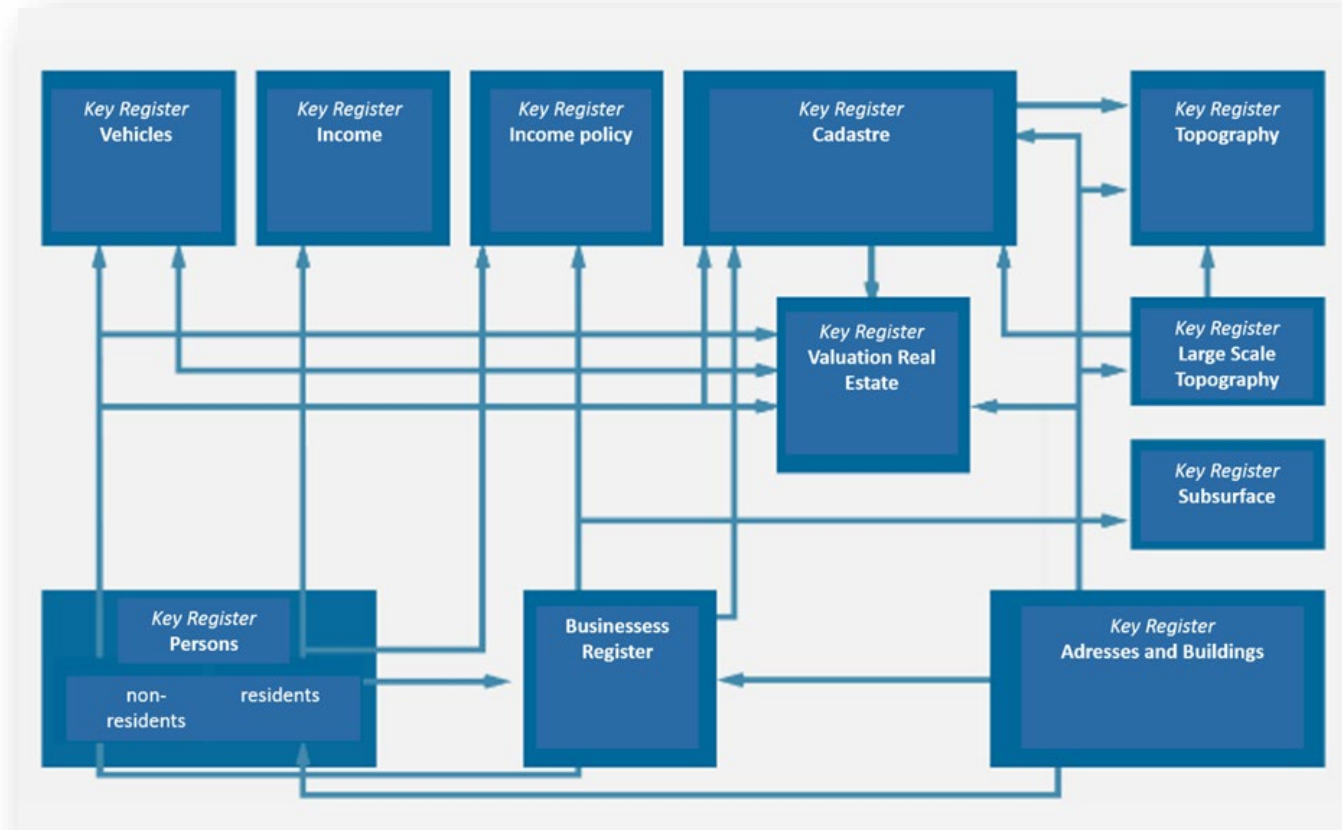


ITUKALEIDOSCOPE
ACCRA2022

Kadaster

Dutch Land Registry and Mapping Agency

- **Role:** National agency tasked with the maintenance and publication of several key registers.
- Spatial Data Infrastructure (SDI) developments in the Netherlands are strongly related to the key registers using in e-government policy
- Organisation is based on the an interrelated (although not automatically connected) system.

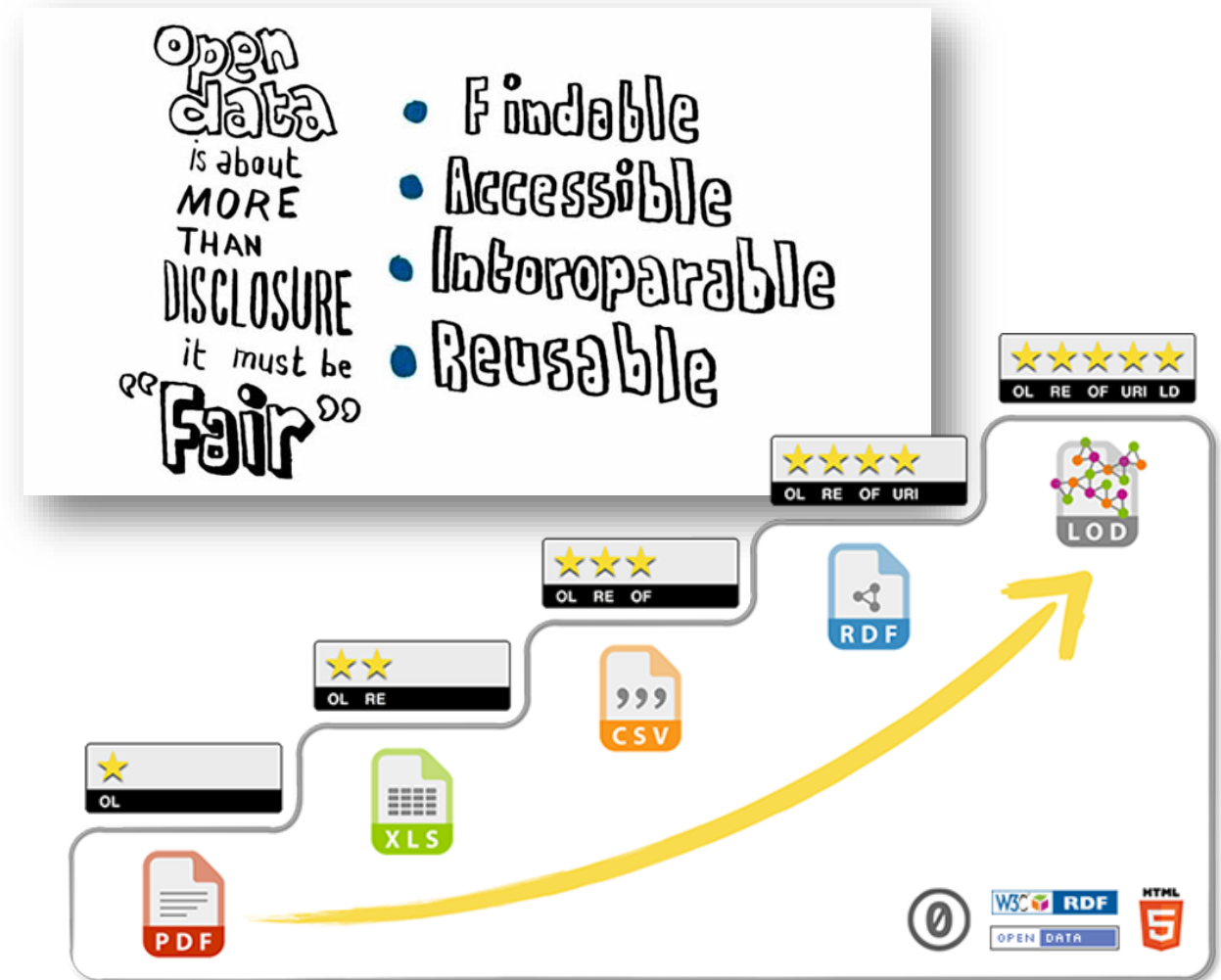


Data per Key Register



From Data Siloes to Linked Data

- Data siloes are a consequence of needing to organize information.
- A need for more certainty and more reuse of data calls for integration and ease of use.
- An integration of data needs to be explicitly modelled (with **semantics**), include **provenance**, published (using **open standards**) and easy to use (provide **metadata**).
- Our data should, therefore, be:
 - Findable, accessible, interoperable and reusable (FAIR)
 - 5-star data





TED2009 • February 2009 | 1.8M views

Like (54K) Share Add

The next web

Read transcript

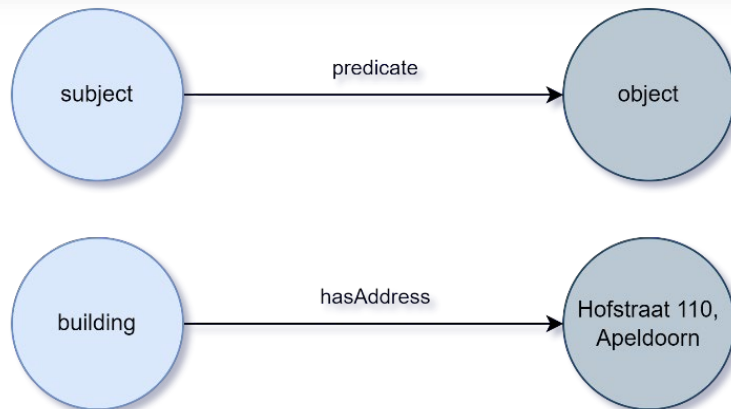
20 years ago, Tim Berners-Lee invented the World Wide Web. For his next project, he's building a web for open, linked data that could do for numbers what the Web did for words, pictures, video: unlock our data and reframe the way we use it together.

This talk was presented at an official TED conference. TED's editors chose to feature it for you.

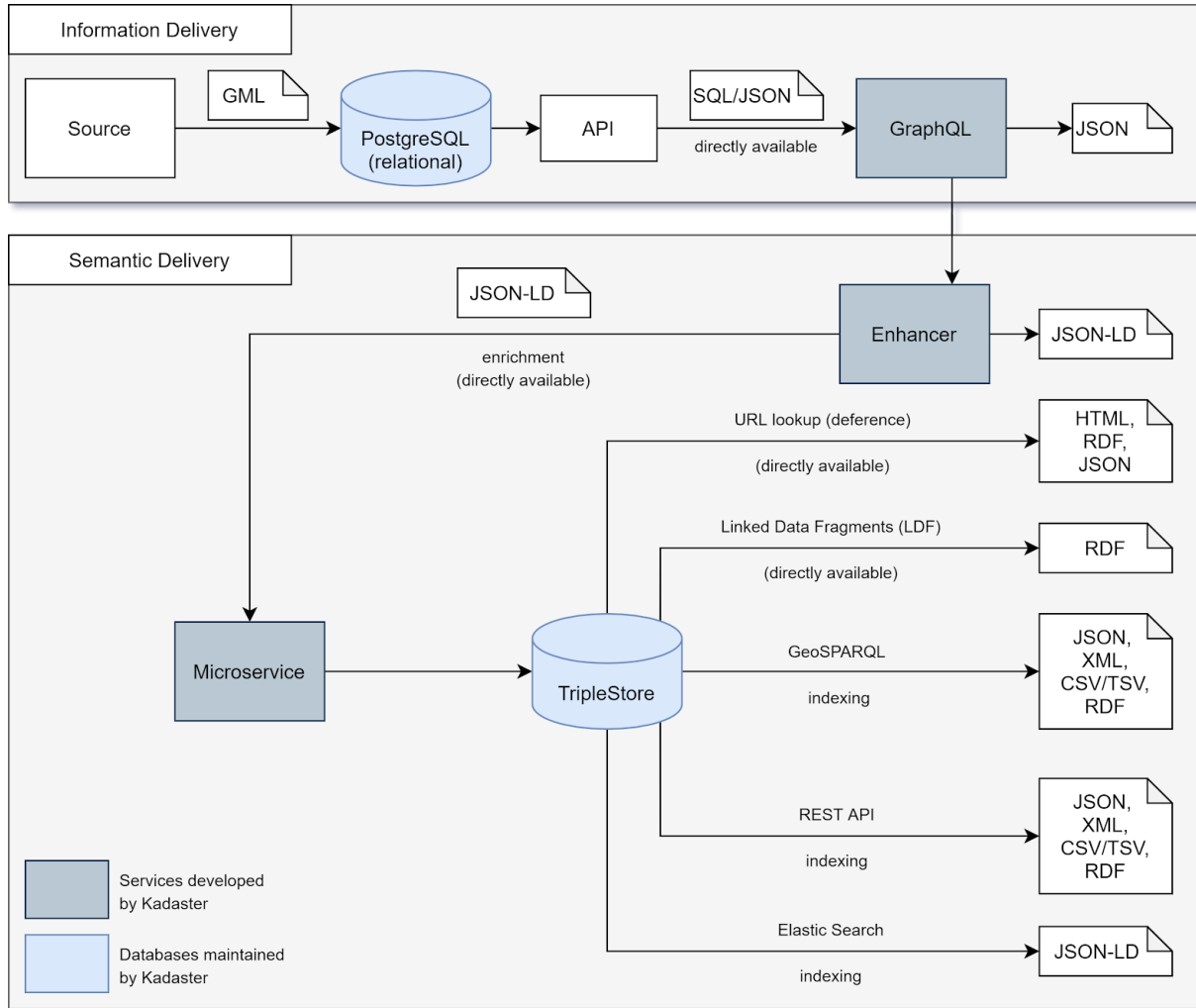
Linked Data?

- A way of publishing data (reusable, web standards, semantics, related to open and big data) on the web which is stored as triples (RDF standard) and can be (in a federated manner) queried with SPARQL.

- Semantics = shared understanding
- **Open standards = increase reuse**
- **Web standards = increase findability**
- Interoperability = increased connectivity between distributed data sources



Building the Kadaster Knowledge Graph (KKG)

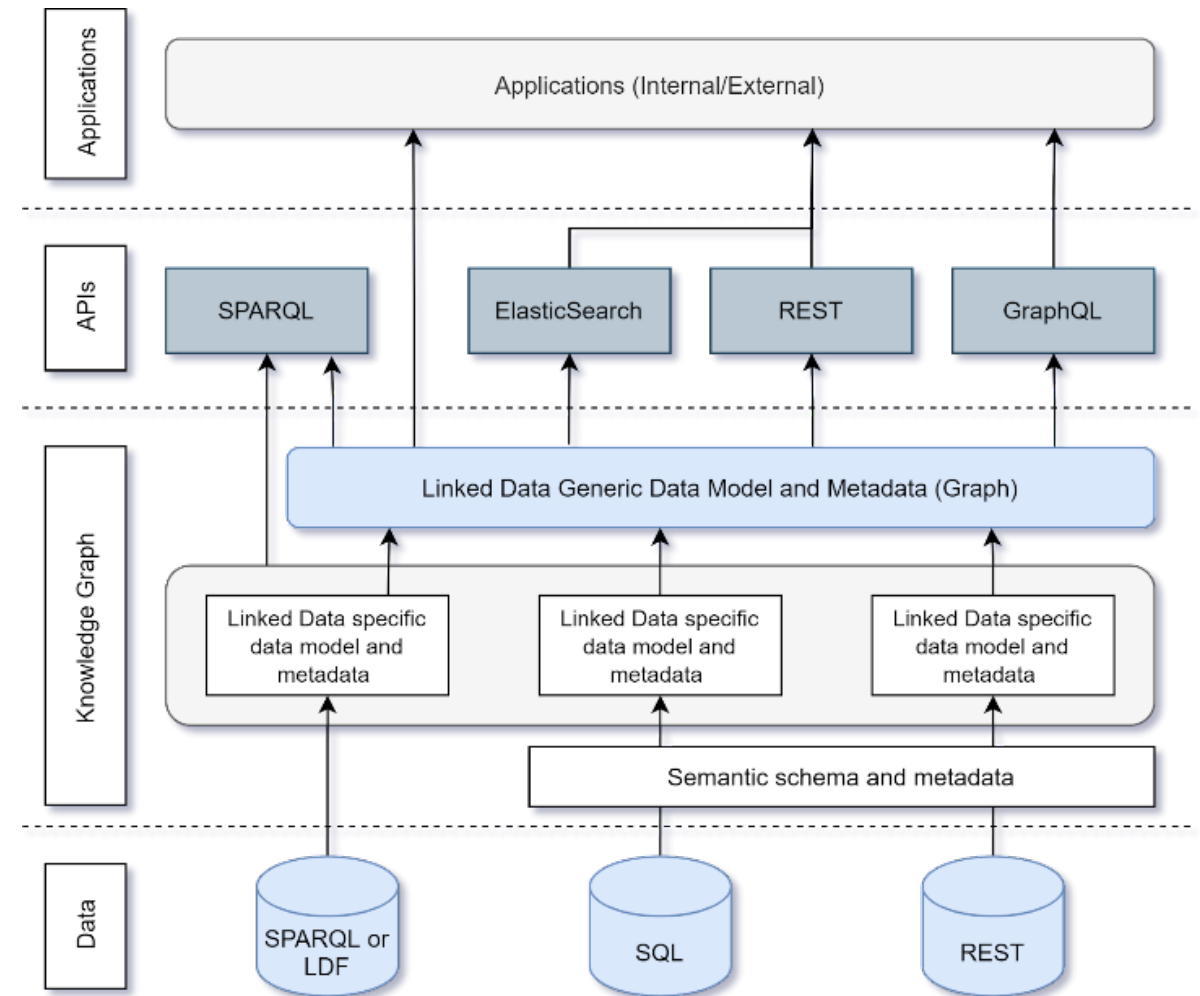


- The KKG contains several key registers, each first made available as a siloed linked data source through an ETL process.
- Extract, Transform, Load (ETL) process:
 - GML indexing step
 - GraphQL endpoint
 - Enhancer microservice to return JSON-LD format
 - SHACL validation step
 - Publication in TriplyDB
 - SPARQL service



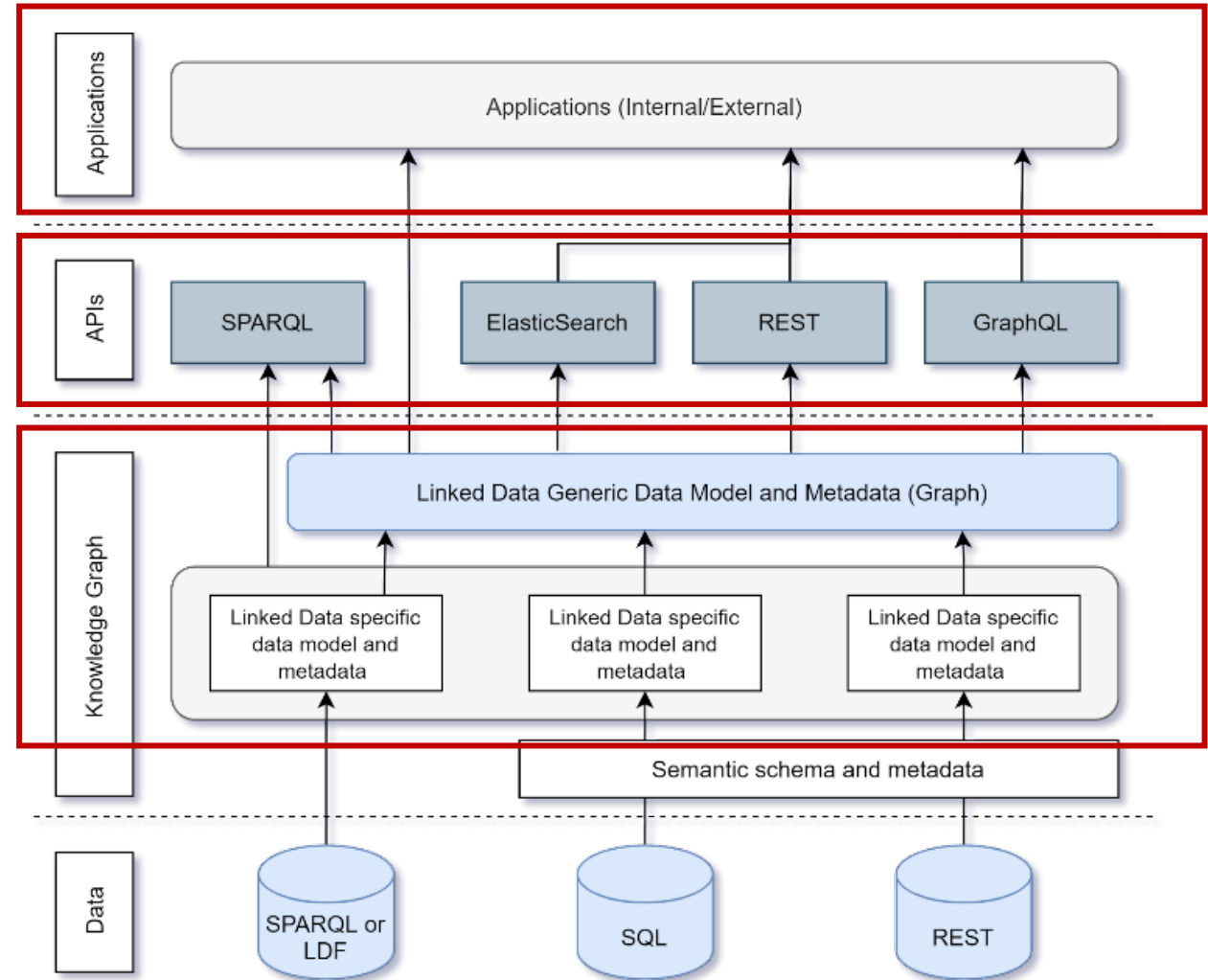
Building the Kadaster Knowledge Graph (KKG)

- Combine key registers with a central data model:
 - Samenhangende Object Registratie (SOR)
- Siloes to Knowledge Graph is achieved by implementing SPARQL Construct queries
- Layered approach:
 - Preservation of provenance
 - Traceability
- KKG contains contains approximately 680 million triples
- Updated on a quarterly basis



Proposed Architectural Approach

- Data remains 'at the source'
- Application-independent data management
- **Open standards-based** interfacing between applications and data sources



The Augmented Reality Application

- **Goal:** Support the interoperability between distributed data sources and end-user applications, allowing users to interact (in a low-threshold manner) with authoritative data published by the organization.
- **Architectural approach:** Separation of application and data source
- **Techniques:** Linked Data, Standardized Interfaces, Augmented Reality (AR)
- **Spatial coverage:** The Netherlands
- **Initial prototype:** Built in 2021/2022 and updated in the summer of 2022
- **Main functionality:** Allow the user to 'scan' a building and see information about that building on their phone screen.



Standardised Interfaces SPARQL APIs

- SPARQL queries used as interfaces between the KKG and the AR application
- 3 SPARQL queries
- The results of each query are made available through a standardized API which can be used directly in the application.
- Available serialisations:
 - JSON-LD
 - Turtle
 - N-Quads

```
1 PREFIX sor: <https://data.kkg.kadaster.nl/sor/model/def/>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 prefix geof: <http://www.opengis.net/def/function/geosparql/>
5 prefix uom: <http://www.opengis.net/def/uom/OGC/1.0/>
6 prefix xsd: <http://www.w3.org/2001/XMLSchema#>
7 prefix vbo: <https://data.kkg.kadaster.nl/nen3610/model/def/>
8 prefix skos: <http://www.w3.org/2004/02/skos/core#>
9
10 SELECT ?gebouw
11   ?gebouwzone
12   a sor:Gebouw
13   sor:geregistreerdMet ?vbo
14   geo:hasGeometry ?vbo
15   filter (isUri(?vbo))
16 }

```

```
1 PREFIX geo: <http://www.opengis.net/def/function/geosparql/>
2 PREFIX sdo0: <http://www.opengis.net/def/sdo/0.0/>
3 PREFIX sor: <https://data.kkg.kadaster.nl/sor/model/def/>
4 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
5 PREFIX time: <http://www.w3.org/2006/time#>
6 PREFIX wbk: <https://data.kkg.kadaster.nl/wbk/model/def/>
7 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
8 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
9
10 SELECT DISTINCT ?gebouw
11   ?bouwjaar ?gebruiksdoel
12   ?vbo
13   a sor:Verblijfsobject
14   sor:gebruiksdoel/sk ?gebruiksdoel
15   sor:maaktDeelUitVar ?vbo
16   sor:hoofdadres ?nummeraanduiding
17   sor:oppervlakte ?vloerOppervlakte
18   ?gebouw
19   a sor:Gebouw;
20   sor:oorspronkelijkBouwjaar ?bouwjaar;
21   sor:bronhouder ?bronhouder .
22
23 ?bronhouder
24   dct:identificatie ?gemeenteCode;
25   skos:prefLabel ?gemeenteNaam .
26
27 optional {
28   ?gebouwzone sor:hoortBij ?vbo
29   optional { ?gebouwzone kad:typeGebouw ?typeGebouw }
30   optional { ?gebouwzone brt_def:soortnaam ?gebouwNaam }
31 }
32
33 optional {
34   ?perceel
35   a <https://data.kkg.kadaster.nl/sor/model/def/Perceel>;
36   sor:hoortBij ?nummeraanduiding;
37   sor:oppervlakte ?perceelOppervlakte .
38 }
39 }

```

API GET: <https://api.labs.kadaster.nl/queries/dst/AR-punt-distance/run?>

Variables +

☰ punt
POINT(4.7892923355103 51.669044494629) ⓘ

View populated query ↗

```
1 PREFIX sor: <https://data.kkg.kadaster.nl/sor/model/def/>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 prefix geof: <http://www.opengis.net/def/function/geosparql/>
5 prefix uom: <http://www.opengis.net/def/uom/OGC/1.0/>
6 prefix xsd: <http://www.w3.org/2001/XMLSchema#>
7 prefix nen3610: <https://data.kkg.kadaster.nl/nen3610/model/def/>
8 prefix geo: <http://www.opengis.net/ont/geosparql#>
9
10 SELECT ?gebouwId ?polygon where {
11   ?gebouwzone
12     a sor:Gebouwzone;
13     sor:geregistreerdMet/nen3610:identificatie ?gebouwId;
14     geo:hasGeometry/geo:asWKT ?polygon.
15     filter (geof:distance (?punt, ?polygon, uom:meter ) < 100 )
16 }
```



Feature 1: Geolocation

- Application returns latitude and longitude of GPS location and transforms to a point.
- Point is used as a parameter for the query
- The query returns all building identifiers and polygons of buildings within a 100m radius.
- UI allows user to select which object based on a house icon.



ITUKALEIDOSCOPE
ACCRA2022

API GET: <https://api.labs.kadaster.nl/queries/dst/ar-demo-verblijfsobjecten-per-gebouwzone/run?>

Variables +

gebouwid
100207638

[View populated query](#)


```
1 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
2 PREFIX nen3610: <https://data.kkg.kadaster.nl/nen3610/mod
3 PREFIX sor: <https://data.kkg.kadaster.nl/sor/model/def/>
4 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6 SELECT * WHERE {
7   ?gebouwzone
8     a sor:Gebouwzone;
9     sor:geregistreerdMet/nen3610:identificatie ?gebouwid;
10    sor:hoortBij ?vbo .
11   ?vbo
12     a sor:Verblijfsobject;
13     sor:geregistreerdMet/nen3610:identificatie ?vboId;
14     sor:hoofdadres ?nummeraanduiding .
15   ?nummeraanduiding
16     a sor:Nummeraanduiding;
17     sor:huisnummer ?huisnummer;
18     sor:postcode ?postcode;
19     sor:ligtAan/skos:prefLabel ?straatnaam .
20 optional {?nummeraanduiding
21   sor:huisletter ?huisletter;
22   sor:huisnummertoevoeging ?toevoeging }
23 }
```




Feature 2: House Number Selection

- Key registers have a distinction between a whole building and parts of the building (e.g. apartments)
- Some attributes are associated with the building object, some with the building part.
- UI displays all building parts based on identifier and allows user to select building part of interest based on a number icon.



API GET: https://api.labs.kadaster.nl/queries/dst/ar-demo-data-voor-plaats/run? 

Variables 

vbold
0003010000129471 


View populated query 

```
1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2 PREFIX dct: <http://purl.org/dc/terms/>
3 PREFIX sdo: <https://schema.org/>
4 PREFIX brt_def: <http://brt.basisregistraties.overheid.nl/def/to
5 PREFIX kad: <https://data.kkg.kadaster.nl/kad/model/def/>
6 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
7 PREFIX sdo0: <http://schema.org/> #error in CBS data
8 PREFIX sor: <https://data.kkg.kadaster.nl/sor/model/def/>
9 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
10 prefix time: <http://www.w3.org/2006/time#>
11 PREFIX wbk: <https://data.labs.kadaster.nl/cbs/wbk/vocab/>
12 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
13 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
14 SELECT DISTINCT ?gebouwNaam ?buurtNaam ?buurtID ?gemeenteNaam ?g
bouwjaar ?gebruiksdoel ?basisschoolAfstand ?treinstationAfstand
15 bind(iri(concat('https://data.kkg.kadaster.nl/id/verblijfsobje
?vbo
16 ?vbo
17 a sor:Verblijfsobject;
18 sor:gebruiksdoel/skos:prefLabel ?gebruiksdoel;
19 sor:maaktDeelUitVan ?gebouw;
20 sor:hoofdadres ?nummeraanduiding;
21 sor:oppervlakte ?vloerOppervlakte0 .
22 ?gebouw
23 a sor:Gebouw;
```





Adres
't Spiker 24
7231JT


woonfunctie


 Bouwjaar: 1976


 Vloer Oppervlakte: 394 m2

 Perceel Oppervlakte: 530 m2

 Gemeente: Zutphen

 Buurt: Dorp Warnsveld

 Aantal basisscholen binnen 3km: 12

 Afstand treinstation: 2.7 km

[Meer informatie](#)

Klopt er iets niet?

Feature 3: Building Information

- Selected building identifier and building part identifier is used as input, all attribute information for both objects are returned in the UI
- Includes:
 - Building year
 - Floorsize
 - Parcel size
 - Municipality name
 - Neighbourhood name
 - Number of schools within a 3km radius
 - Distance to the nearest trainstation



Added Value of Architectural Approach

- **Open standards-based approach:**
 - ✓ SPARQL relatively easy to understand
 - ✓ Open standards-based API is recognisable for all developers
 - ✓ Only requirement is results visualisation, no integration or transformation
 - ✓ Updates are immediately reflected in the application
 - ✓ Copying and self-storage of data is no longer required
 - Small learning curve involved in learning to write performant queries
 - Underlying schema changes require maintenance to the query.
- **Application-Independent Data Publication:**
 - ✓ Data source is easily updated and expanded
 - ✓ Data is immediately digestible by the application
 - ✓ Organization dependent: only a single KG is required, and various applications can consume the data
 - ✓ Supports cross-organization interoperability, federated SPARQL queries.
 - Only appropriate where data (content and structure) is not context-specific

Geoinformation for everyone?



ITUKALEIDOSCOPE
ACCRA2022

Thank you!