

Efficient Presentation of GeoSPARQL-Results

Bachelor's Thesis

Denis Veil

Albert-Ludwigs-Universität Freiburg

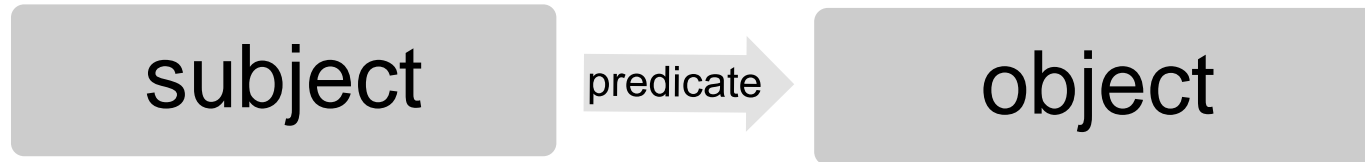
Examiner: Prof. Dr. Hannah Bast

Adviser: Patrick Brosi



**UNI
FREIBURG**

- SPARQL engine for combined search^[1]
- Knowledge base as database
 - contains statements as triples



- Spatial extension to SPARQL
 - for geographic information [2]
- Extending the knowledge base by spatial data [3]
- Attribute “hasGeometry”
 - well-known text literal [4]

Well-known Text Literals

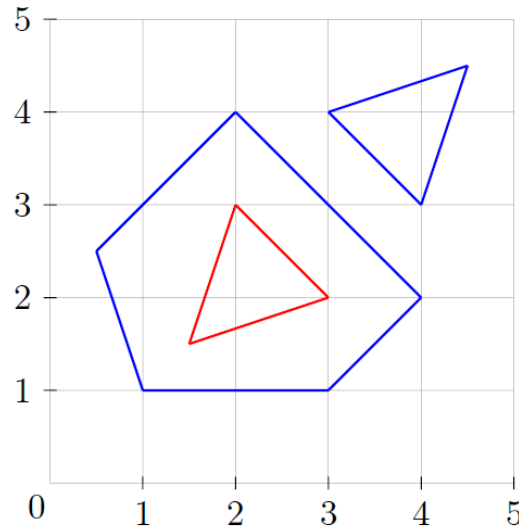


- **Point:**
 - POINT(x0 y0)
- **Line:**
 - LINESTRING(x0 y0, x1 y1, ..., xn yn)
- **Simple 2-dimensional shape:**
 - POLYGON(((x0 y0, x1 y1, ..., xn yn), (a0 b0, ...)))
- **Multiple 2-dimensional shapes:**
 - MULTIPOLYGON((((x0 y0, ..., xn yn), (a0 b0, ...)), ((...)))

Well-known Text Example



- MULTIPOLYGON(((1 1, 3 1, 4 2, 2 4, 0.5 2.5), (1.5 1.5, 3 2, 2 3)), ((4 3, 4.5 4.5, 3 4)))



SPARQL Query



```
PREFIX osmkey: <https://www.openstreetmap.org/wiki/Key:>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX osm: <https://www.openstreetmap.org/>
SELECT ?osm_id ?geometry WHERE {
  ?osm_id osmkey:building "university" .
  ?osm_id geo:hasGeometry ?geometry .
  ?osm_id osm:envelope ?envelope .
  FILTER contained(?envelope, "LINESTRING(7.662006 47.903578, 7.930844
    48.071058)")
}
```

- QLever delivers GeoSPARQL results
 - „all residential highways“ → 57 million objects
 - 100.000 objects ~ 35.7 MB
- Loading all objects at once
 - long request response
- Render all objects at once
 - laggy user experience



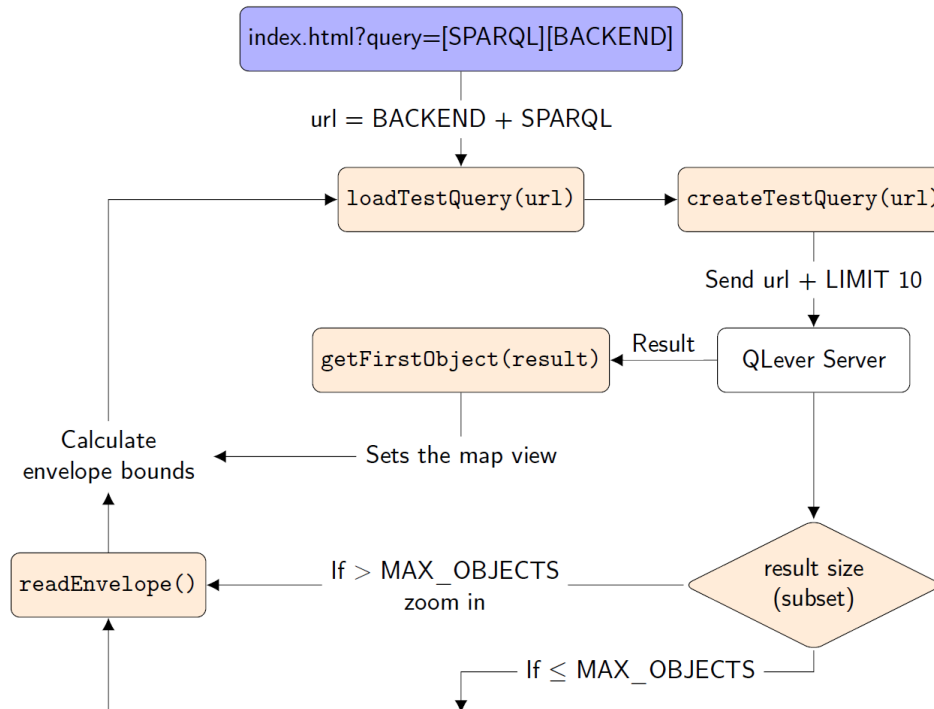
Questions?

Approach

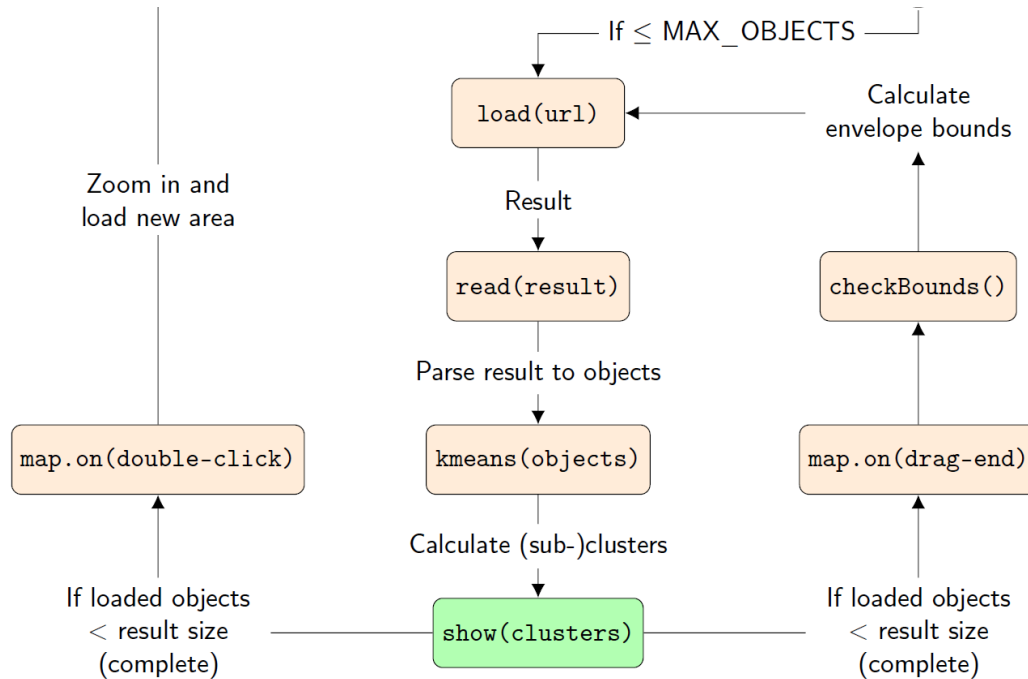


- Create a test query from the original
 - to get < 12.000 objects
- Cluster the result
 - optional: create sub clusters
- Render the result with *Leaflet*^[5]

Flow Chart 1/2

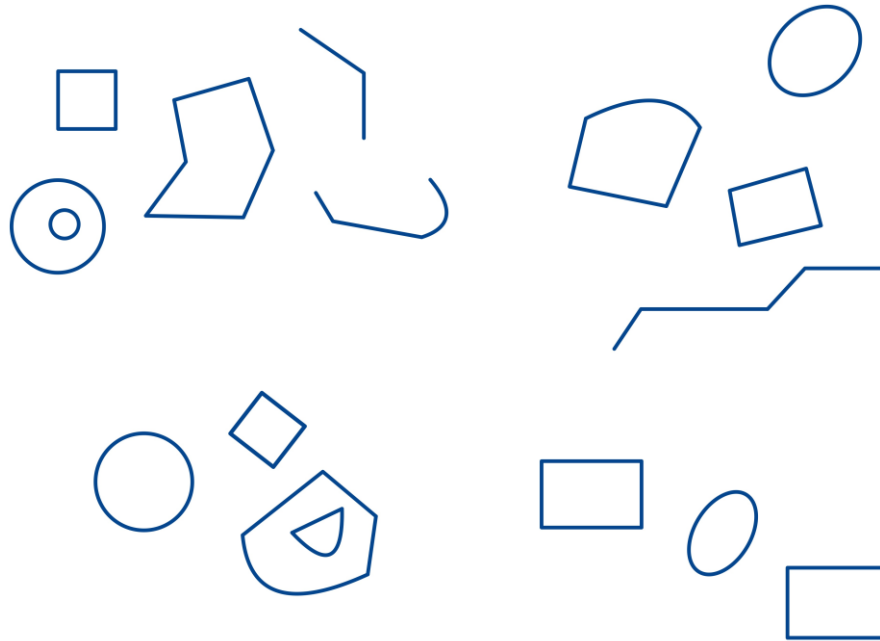


Flow Chart 2/2

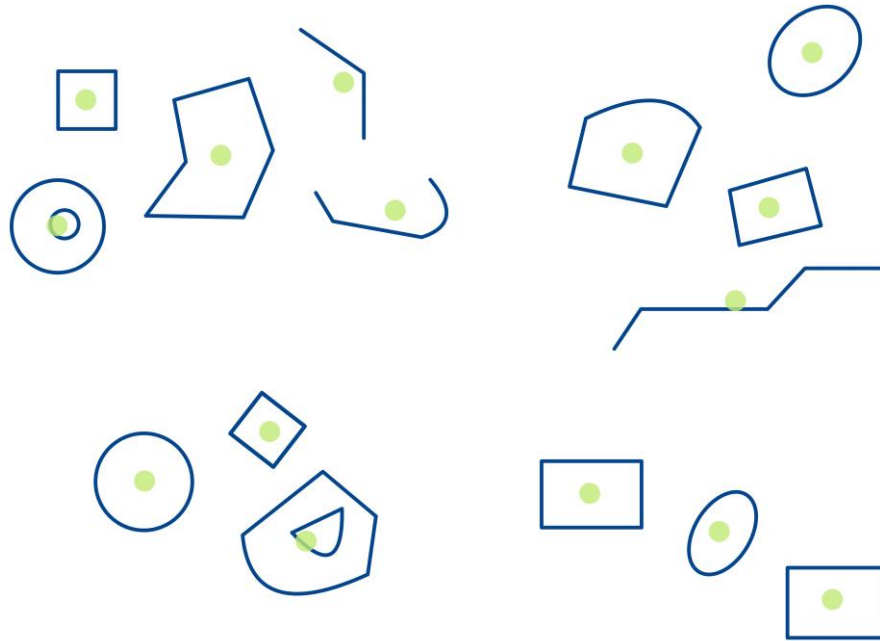


- K-means clustering
 - $k = \lfloor \log_2(\text{length of array}) \rfloor$
- Centroid initialization
 - custom “Naïve Sharding”^[6]
- Create sub-clusters if a cluster has too many objects

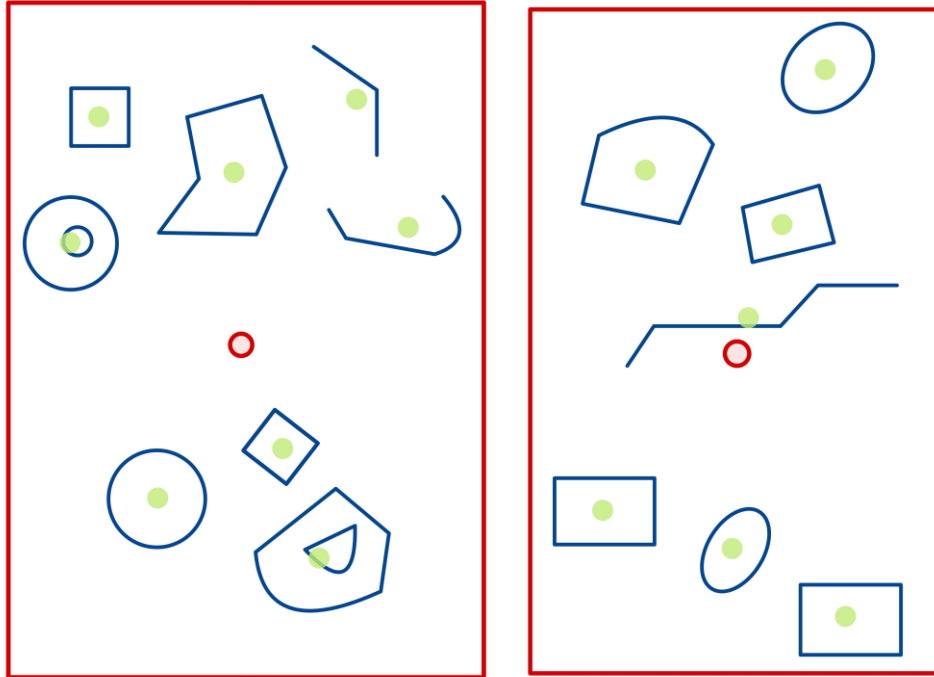
Clustering Example 1/4



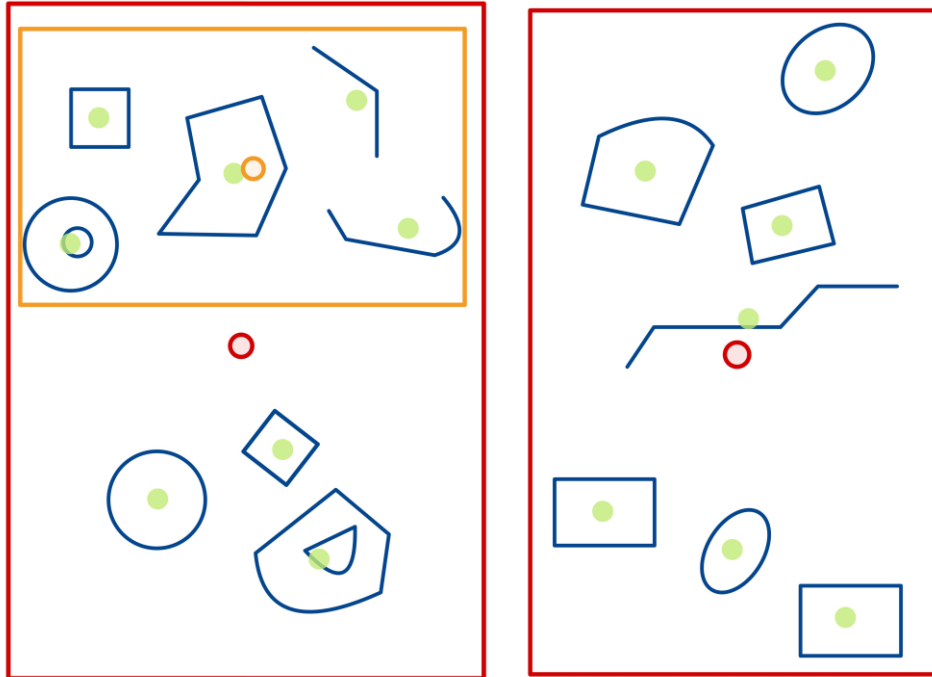
Clustering Example 2/4



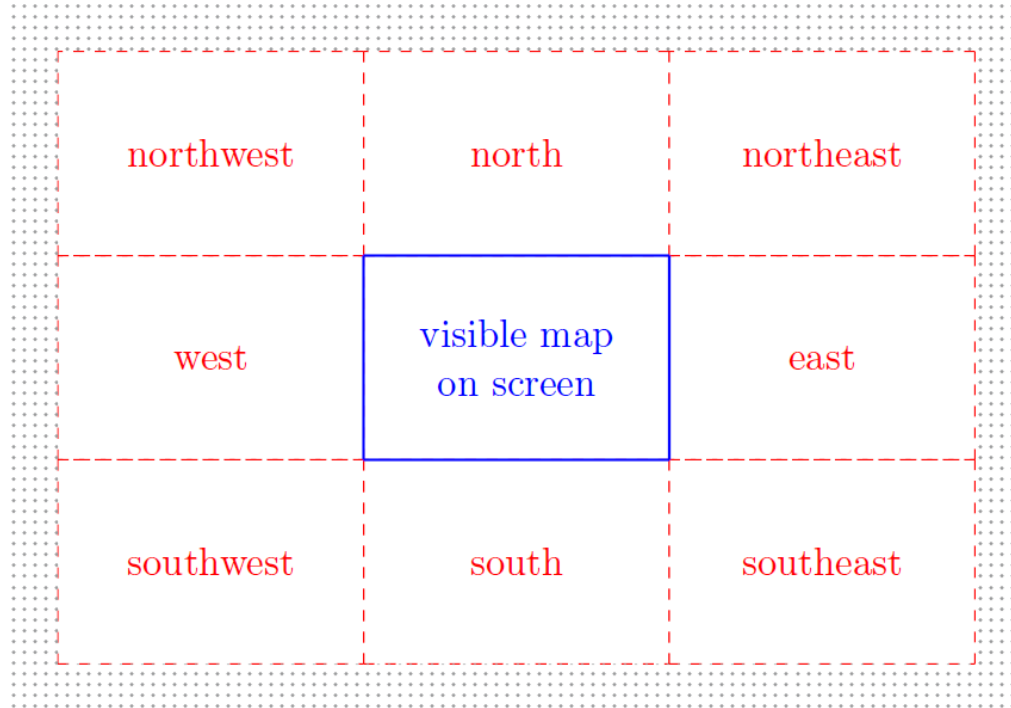
Clustering Example 3/4



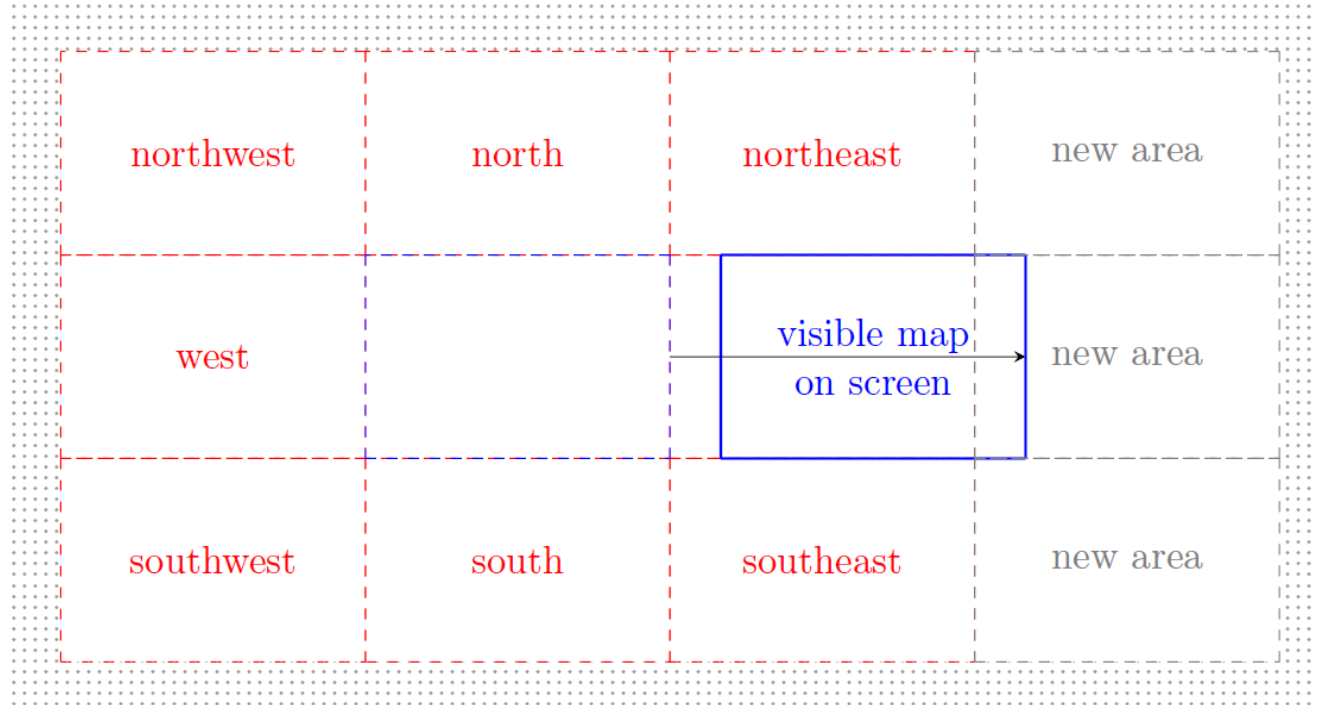
Clustering Example 4/4



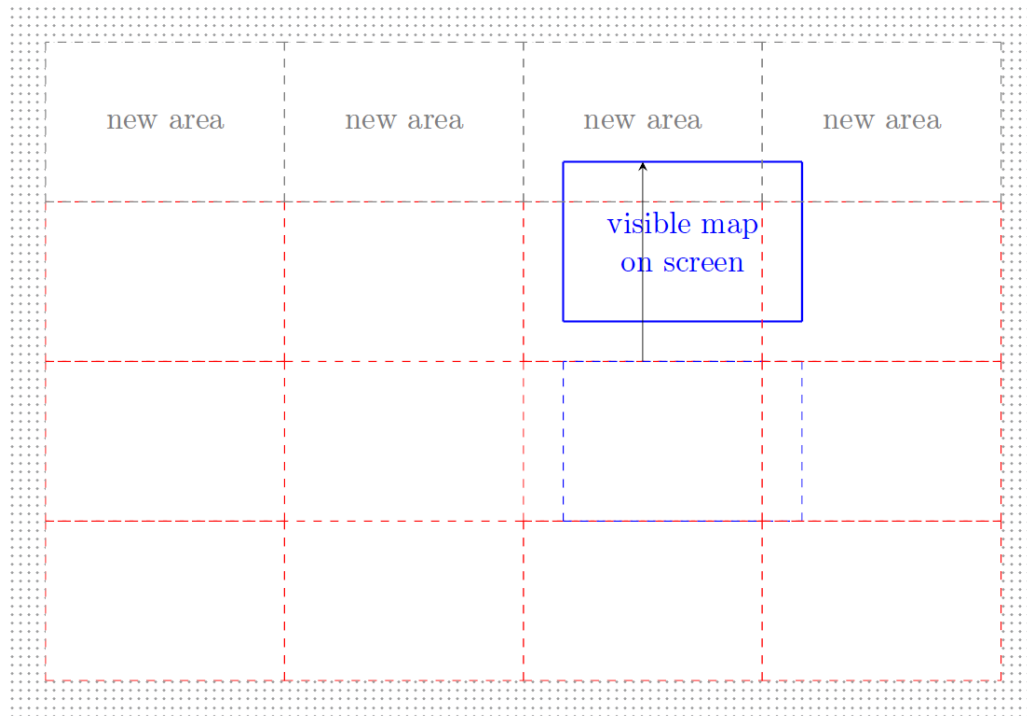
Loading New Area 1/3



Loading New Area 2/3



Loading New Area 3/3





Questions?

Evaluation



	memory (MB)	response time (s) (50Mbit DL)	render time (s)	time total (s)	#objects
Baseline	19,7	2,9	4,7	7,6	50.000
MarkerCluster			7,6	10,5	
Cluster only			3,6	6,5	
Baseline	35,7	5,4	8,1	11	100.000
MarkerCluster			26,2	29,1	
Cluster only			7	9,9	
Filter + Cluster	0,3 – 2,4	0,2 - 0,8	0,2 - 0,6	0,4 - 1,4	950 - 7983


Conclusion



- Average loading time is $< 1\text{sec}$
- Average rendering time is $< 1\text{sec}$
- Loading new area almost instantly



- Extending spatial data on QLever
 - center of the object
 - tile and zoom level of the center
- Pre-compute clusters on QLever
 - send only clusters on lower zoom levels
 - on click send subclusters and/or objects
- Improve the cluster shape



Thank you for your attention



- [1] H. Bast and B. Buchhold, “Qlever: A query engine for efficient sparql+text search,” pp. 647–656, Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 11 2017.
- [2] [GeoSPARQL - A Geographic Query Language for RDF Data](#)
- [3] A. Lehmann, “Creating a rdf knowledgebase from openstreetmap data,” Master’s thesis, Albert Ludwig University of Freiburg, 5 2021



- [4] R. Lott, “Geographic information - well-known text representation of coordinate reference systems.” <http://docs.opengeospatial.org/is/12-063r5/12-063r5.html>, 2015. [Online; accessed 31-August-2021].
- [5] V. Agafonkin, “Leaflet.” <https://github.com/Leaflet/Leaflet>, 2020. [Online; accessed 31-August-2021].
- [6] M. Mayo, “Toward increased k-means clustering efficiency with the naïve sharding centroid initialization method.” <https://www.kdnuggets.com/2017/03/naive-sharding-centroid-initialization-method.html>, 2017. [Online; accessed 31-August-2021].

Appendix: Evaluation



	response time (ms)	render time (ms)	memory	#objects
Baseline (simple loading)	1461.4	2132	9.5 MB	25000
	2920.0	4727.9	19.7 MB	50000
	4121.2	6426.6	28.4 MB	75000
	5401.8	8107.3	35.7 MB	100000
Leaflet.MarkerCluster	1432.7	2798.7	9.5 MB	25000
	2872.5	7654.0	19.7 MB	50000
	4134.2	14511.3	28.4 MB	75000
	5465.2	26200.3	35.7 MB	100000
Cluster only (own implementation)	1480.3	1816.5	9.5 MB	25000
	2884.1	3600.2	19.7 MB	50000
	4103.5	5170.1	28.4 MB	75000
	5417.5	6971.3	35.7 MB	100000
Filter + Cluster (own implementation)	196.8	206.1	302 kB	950
	289.1	293.0	861 kB	2786
	743.6	601.3	2.4 MB	7983
	692.4	414.7	2.4 MB	7983
	706.0	589.9	2.4 MB	7983