



Teselas vectoriales en Elasticsearch

Jorge Sanz

2022-06-08

<https://ela.st/siglibre2022>



Teselas vectoriales **y más** en Elasticsearch

Jorge Sanz

2022-06-08

<https://ela.st/siglibre2022>

Jorge Sanz

✉ jorge.sanz@elastic.co
🐦 [@xurxosanz](https://twitter.com/xurxosanz)
🐙 [@xurxosanz@mastodon.social](https://mastodon.social/@xurxosanz)
🌐 [jsanz](https://www.linkedin.com/in/jsanz)



Elastic

Elastic ayuda a organizaciones de todo el mundo a poner sus datos a **trabajar** gracias al poder de las **búsquedas**.



Elastic de un vistazo

NYSE: ESTC



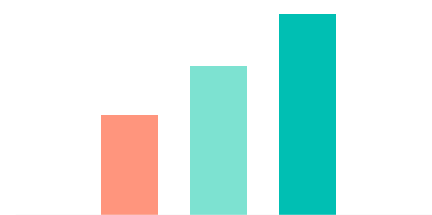
Fundada
en **2012**



3100+
empleados ~**80** en
España



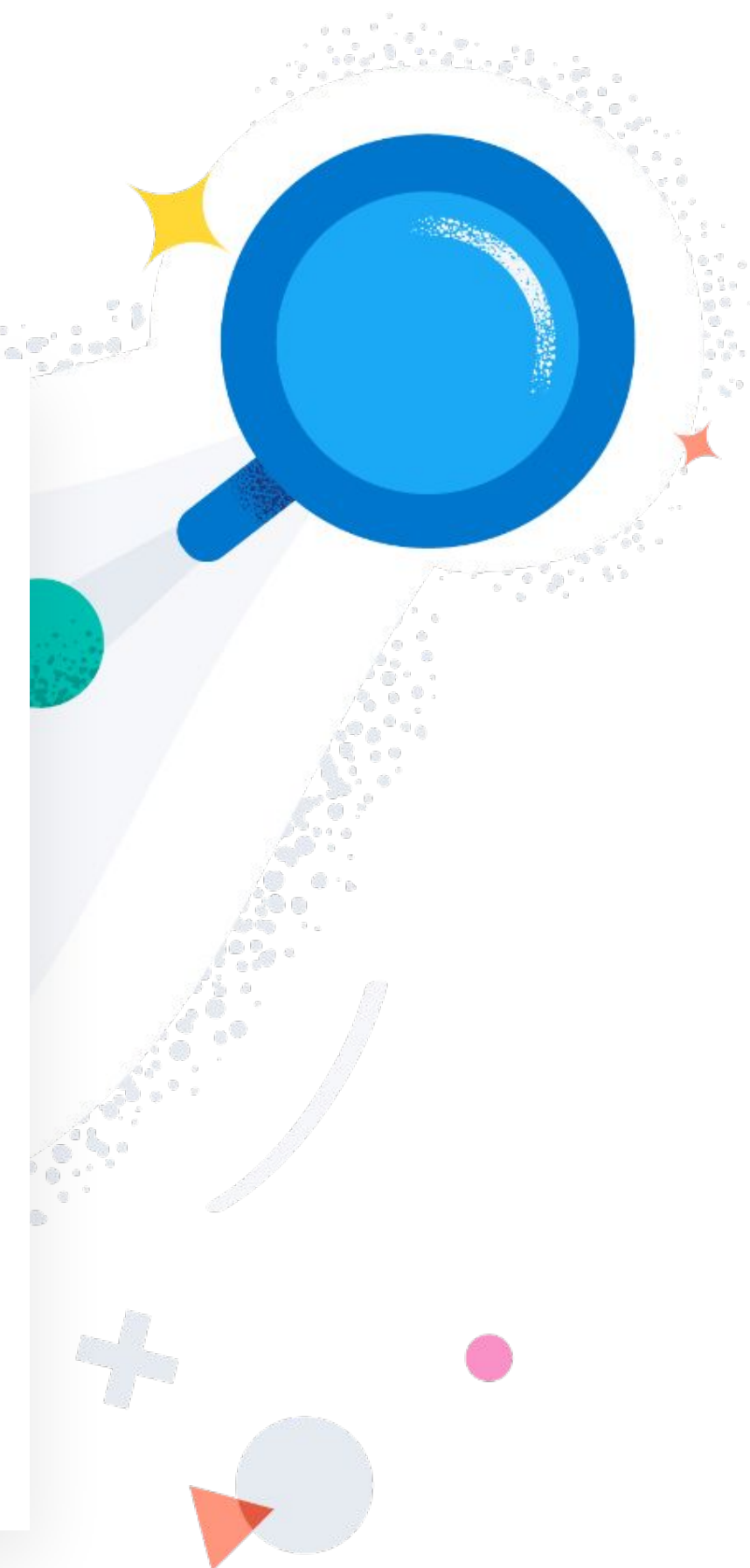
40+
países con
empleados



17,900+
suscripciones

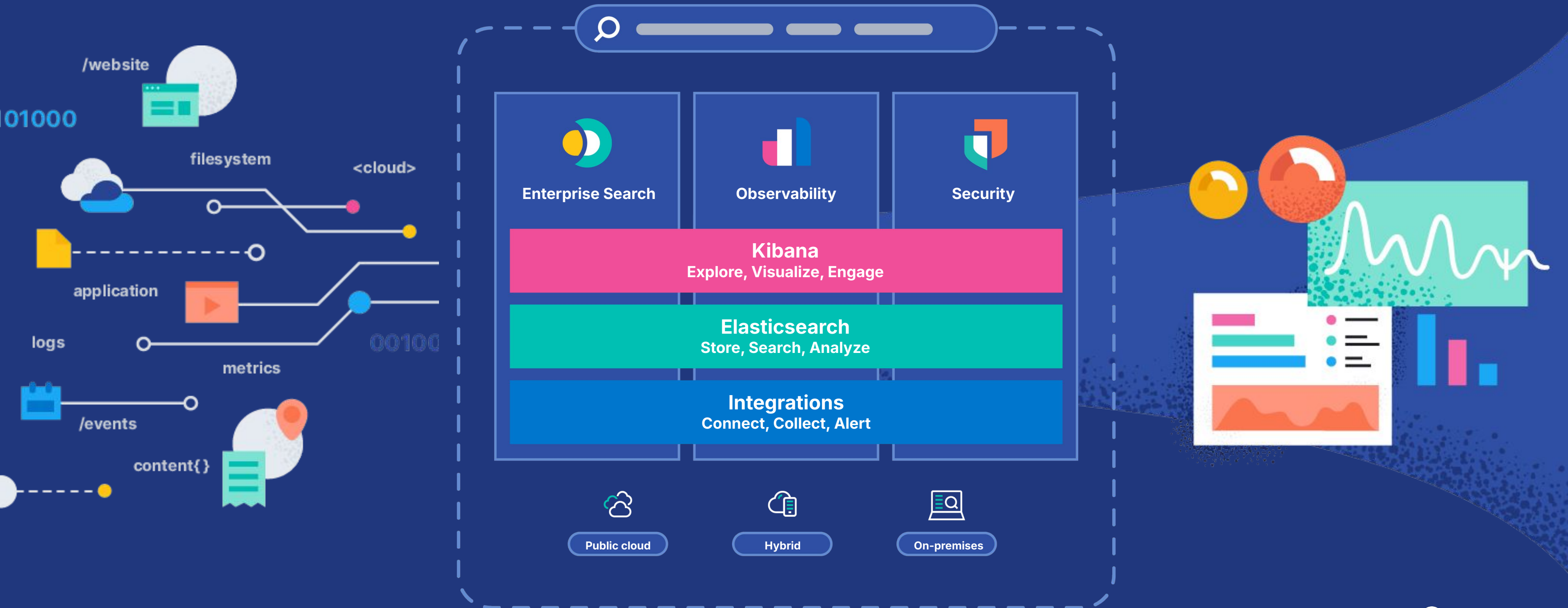


54%
de las empresas del
Fortune 500 confían en
Elastic



Elastic sirve a todo tipo de organizaciones

TECHNOLOGY	FINANCE	TELCO	CONSUMER	HEALTHCARE	PUBLIC SECTOR	AUTOMOTIVE / TRANSPORTATION	RETAIL
							
							
							
							
							
							



Comunidade

<https://github.com/elastic>

<https://ela.st/slack>

<https://discuss.elastic.co>

The screenshot shows the Elastic community forum interface. At the top, there's a navigation bar with the Elastic logo, a search icon, and a user profile icon. Below the navigation bar, there are tabs for 'all categories', 'all tags', 'Categories', 'Latest', 'New (154)', 'Unread (21)', and 'Top'. A '+ New Topic' button is also visible. The main content area is divided into two columns. The left column lists categories: 'Announcements' (1 / week, 1 unread, 1 new), 'Elastic Stack' (322 / week, 19 unread, 129 new), 'Elastic Enterprise Search' (5 / week, 2 new), and 'Elastic Observability' (23 / week, 14 new). Each category has a brief description and a list of sub-categories. The right column shows a list of topics under the 'Latest' tab. Each topic includes a user profile picture, the topic title, the number of replies, the time since the last reply, and the category name. The topics listed are: 'Notes on Using These Forums' (2 replies, Apr 2017), 'Logstash pipeline graceful shutdown: потеря in-memory данных?' (0 replies, 4m), 'Collapse within top hit aggregation results' (0 replies, 5m), 'Drilldown is not working with Visualization' (2 replies, 9m), 'Do not show results on page load' (0 replies, 20m), 'Custom transactions in checkout process' (0 replies, 21m), 'How to view sql queries in APM' (6 replies, 23m), and 'Installation seems to hang' (3 replies).

elastic

all categories all tags Categories Latest New (154) Unread (21) Top + New Topic

Category Topics Latest

Announcements 1 / week
1 unread 1 new
Release and security announcements and other bits about all of our Elastic products that we think will be useful to everyone.
Security Announcements Community Ecosystem 1 unread

Elastic Stack 322 / week
19 unread 129 new
Elasticsearch, Kibana, Beats, and Logstash - also known as the ELK Stack. Reliably and securely take data from any source, in any format, then search, analyze, and visualize it in real time. Please post your topic under the relevant product category - Elasticsearch, Kibana, Beats, Logstash.
Elasticsearch 4 unread 59 new Kibana 14 unread 32 new
Beats 20 new Logstash 1 unread 18 new

Elastic Enterprise Search 5 / week
2 new
Easily implement powerful, modern search experiences for your busy team. Quickly add pre-tuned search to your website, app, or workplace. Search it all, simply.
App Search 2 new Site Search Workplace Search

Elastic Observability 23 / week
14 new
Bring your logs, infrastructure and availability metrics, and APM traces together at scale in a

Notes on Using These Forums 2
Meta Elastic Apr 2017

Logstash pipeline graceful shutdown: потеря in-memory данных? 0
4m
Вопросы на русском языке

Collapse within top hit aggregation results 0
5m
Elasticsearch

Drilldown is not working with Visualization 2
9m
Kibana

Do not show results on page load 0
20m
App Search

Custom transactions in checkout process 0
21m
Elasticsearch

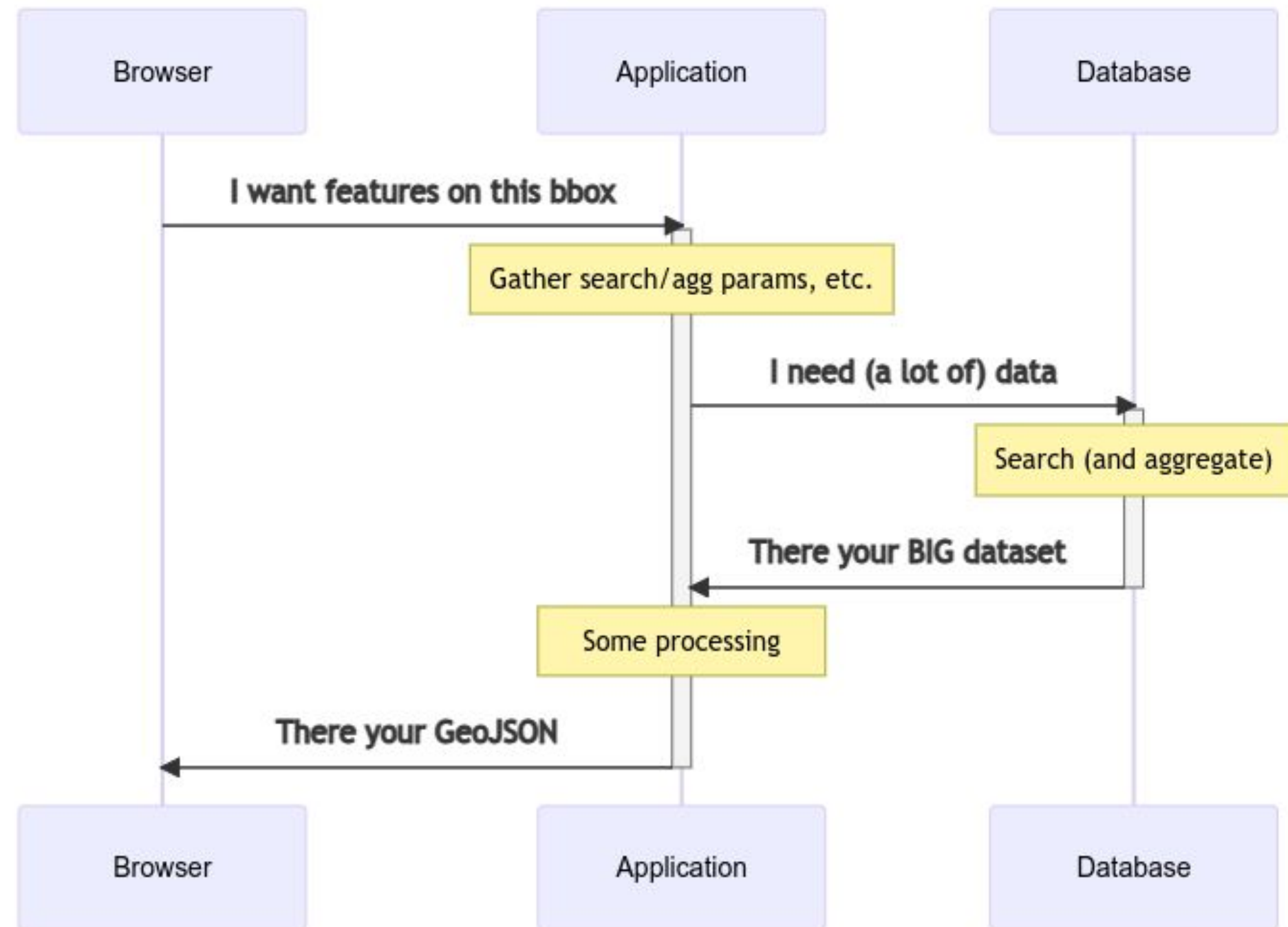
How to view sql queries in APM 6
23m
APM dotnet

Installation seems to hang 3

Why vector tiles in Elasticsearch?

Classic approach

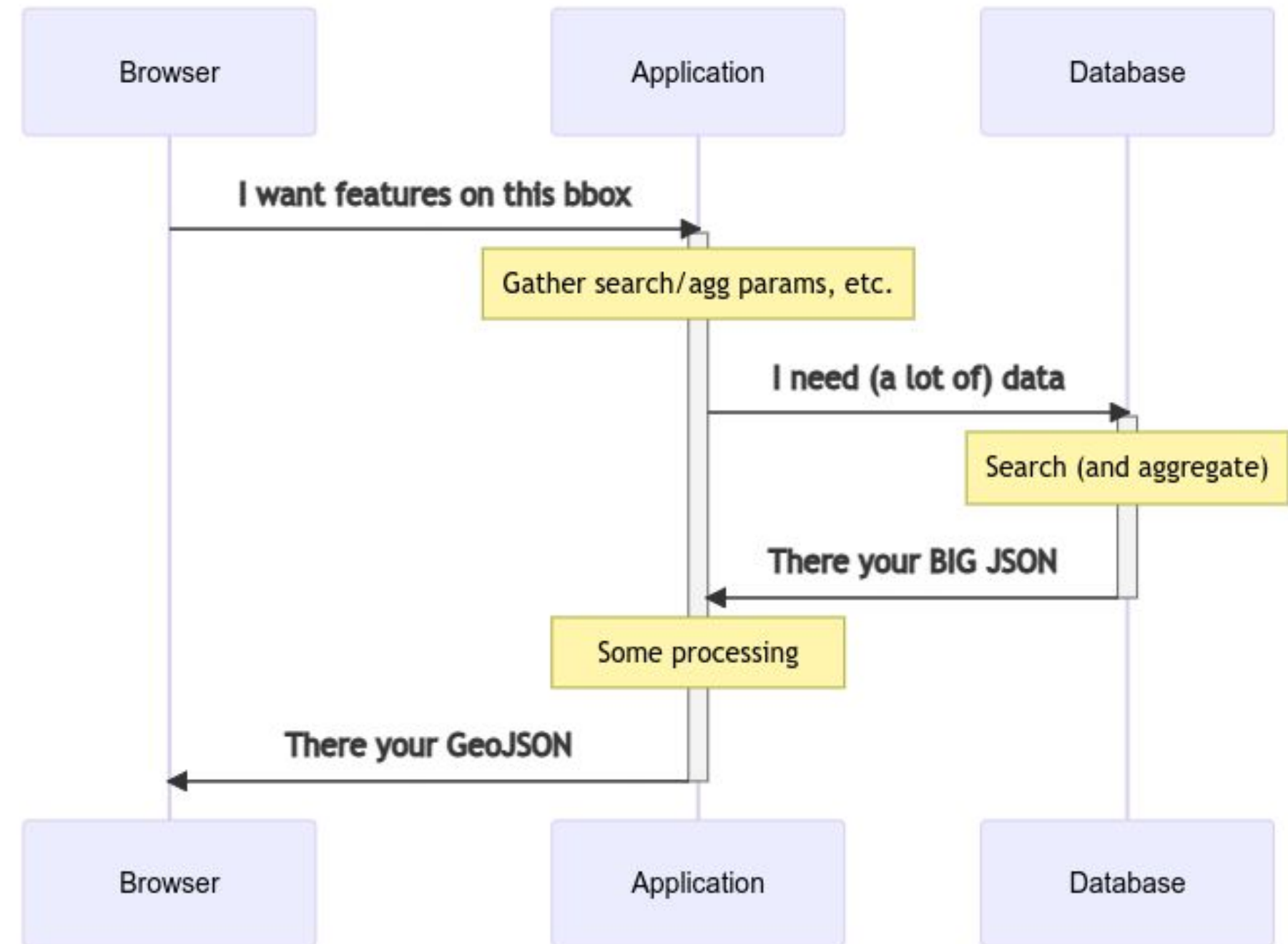
1. Browser requests features by **bounding box**
2. Middleware gathers **query parameters**: aggregation, search filters, etc
3. Database performs the query and returns **raw results**
4. Middleware **process** results and returns **GeoJSON**





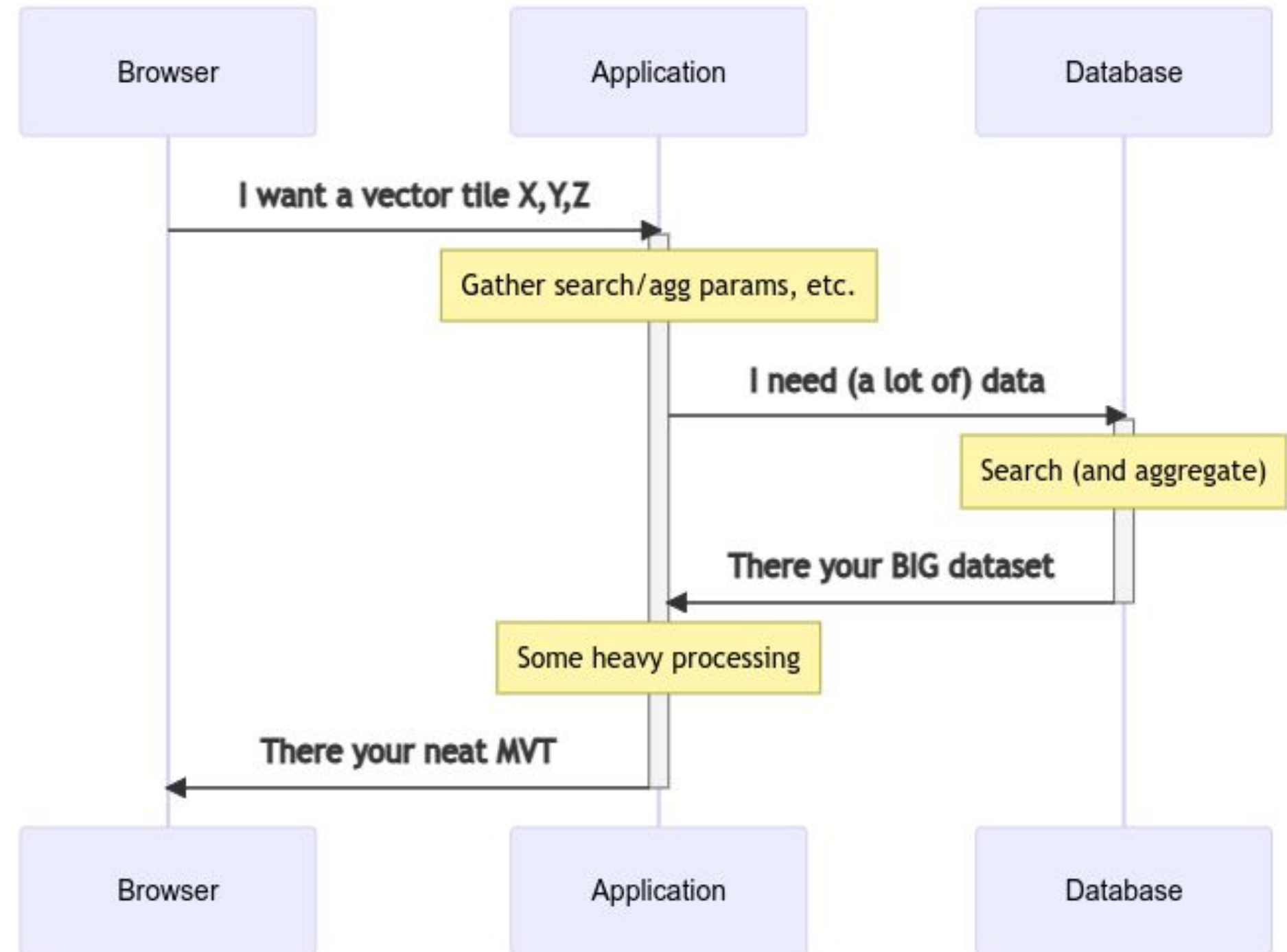
Classic approach

- Elasticsearch outputs **JSON** results (up to 10.000 documents per request)
- Results contain **full geometry** data regardless of rendering zoom level



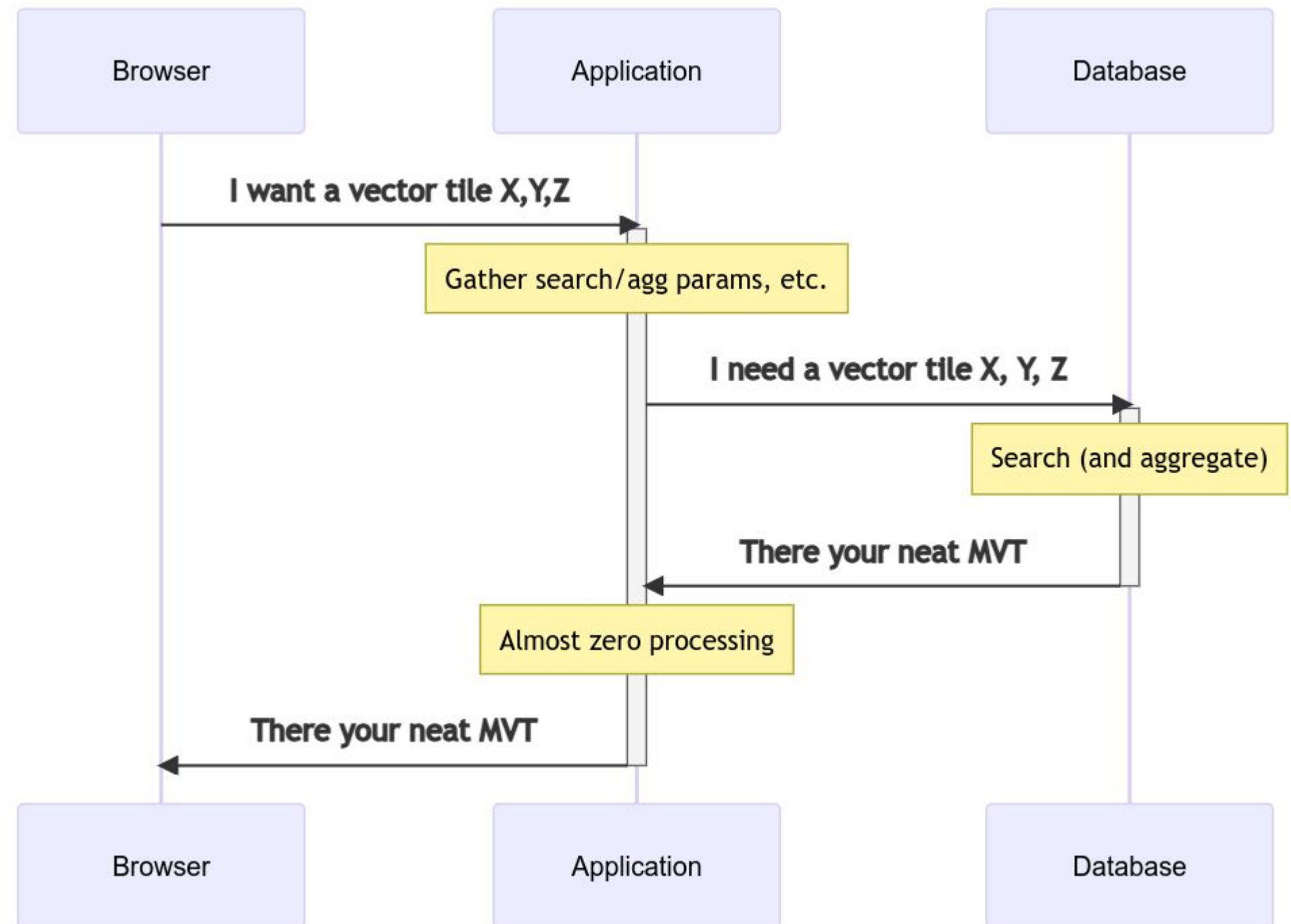
Better approach

1. Browser requests a tile
2. Middleware gathers **query parameters**: aggregation, search filters, etc
3. Database performs the query and returns **raw results**
4. Middleware **process** results into a vector tile
5. **Fast** rendering (ex. maplibre)



Modern approach

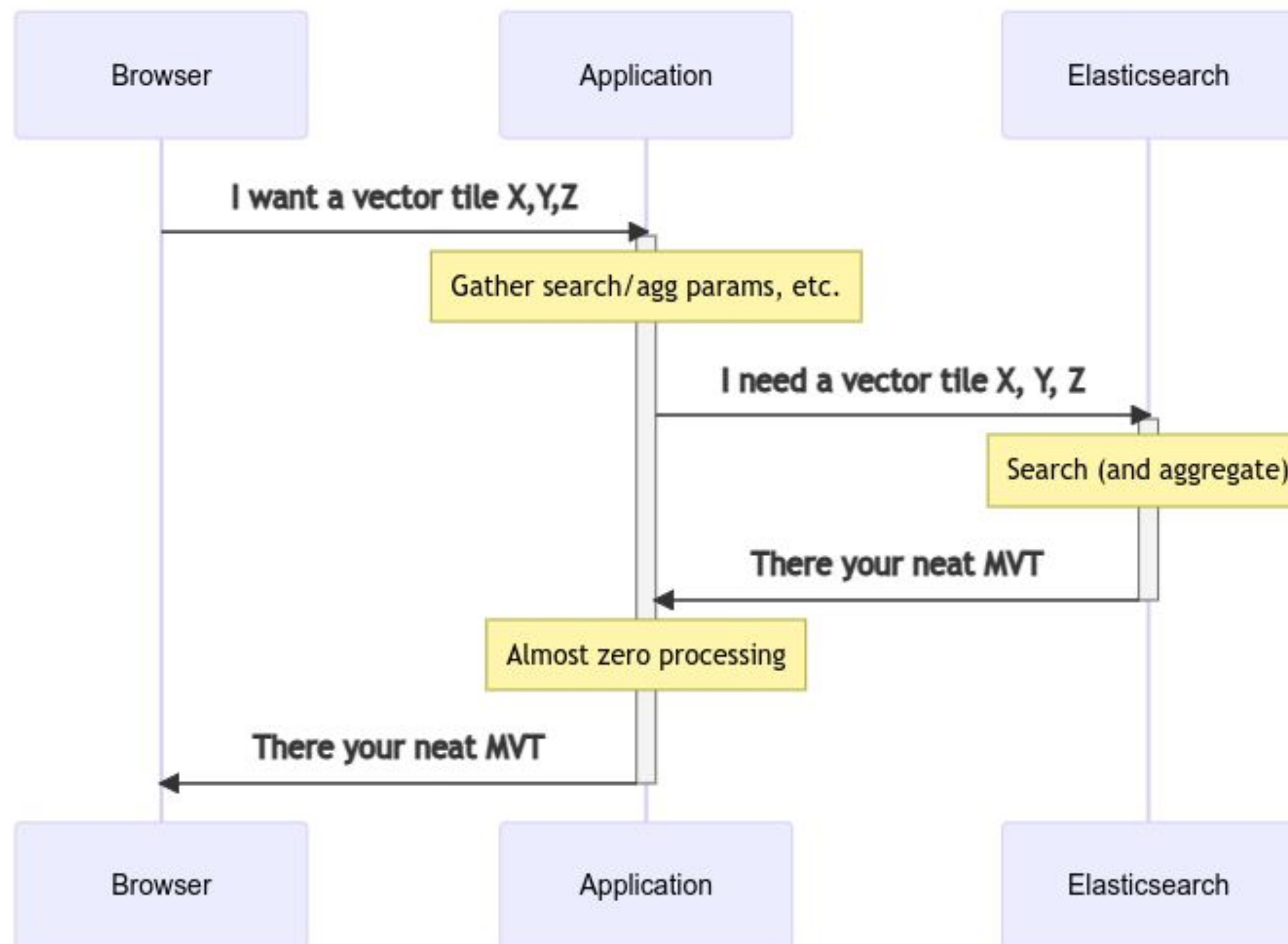
1. Browser requests a tile
2. Middleware gathers **query parameters**: aggregation, search filters, etc
3. Database performs the query **and returns a vector tile**
4. Middleware **forwards** results to client
5. **Fast** rendering





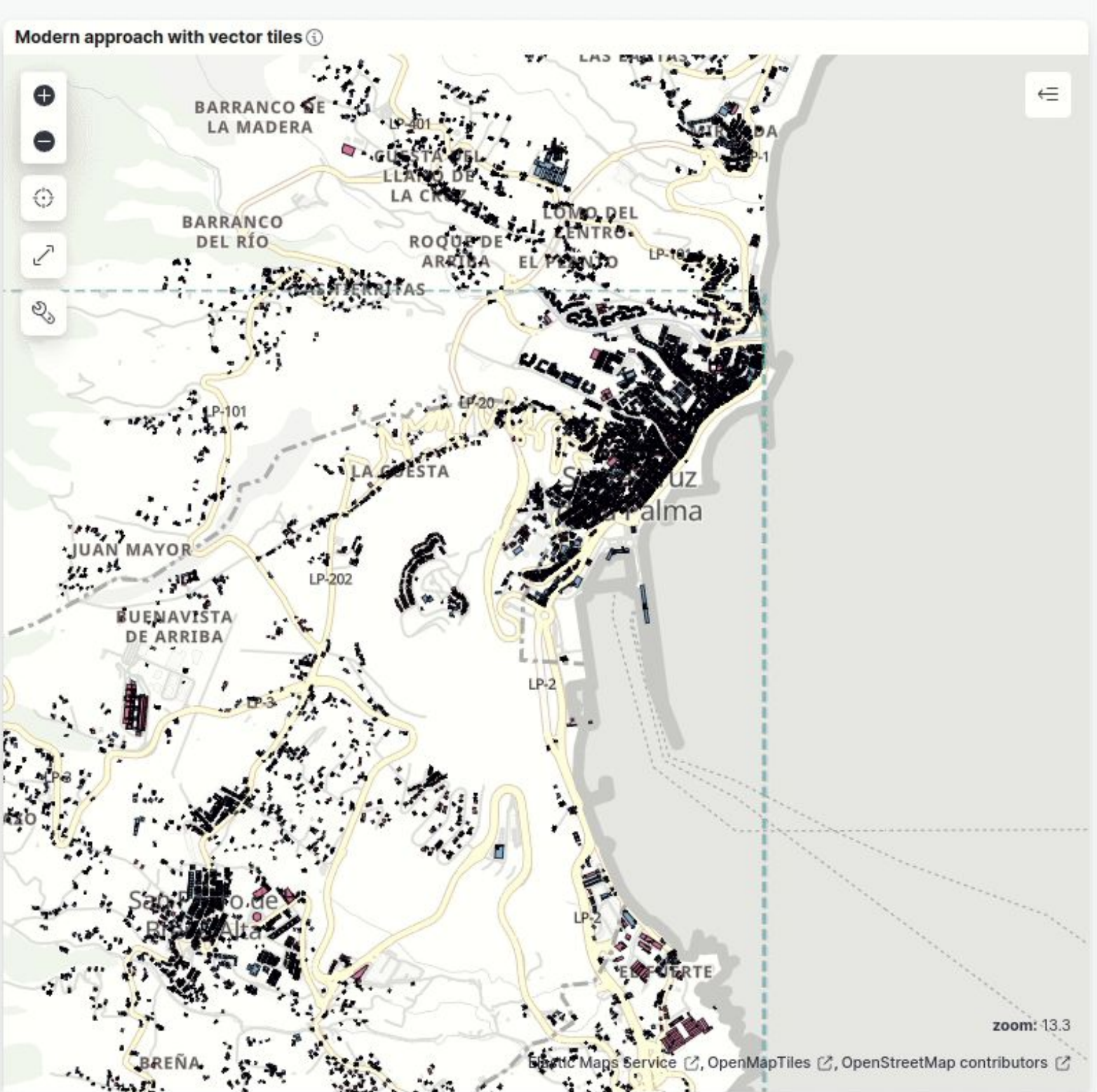
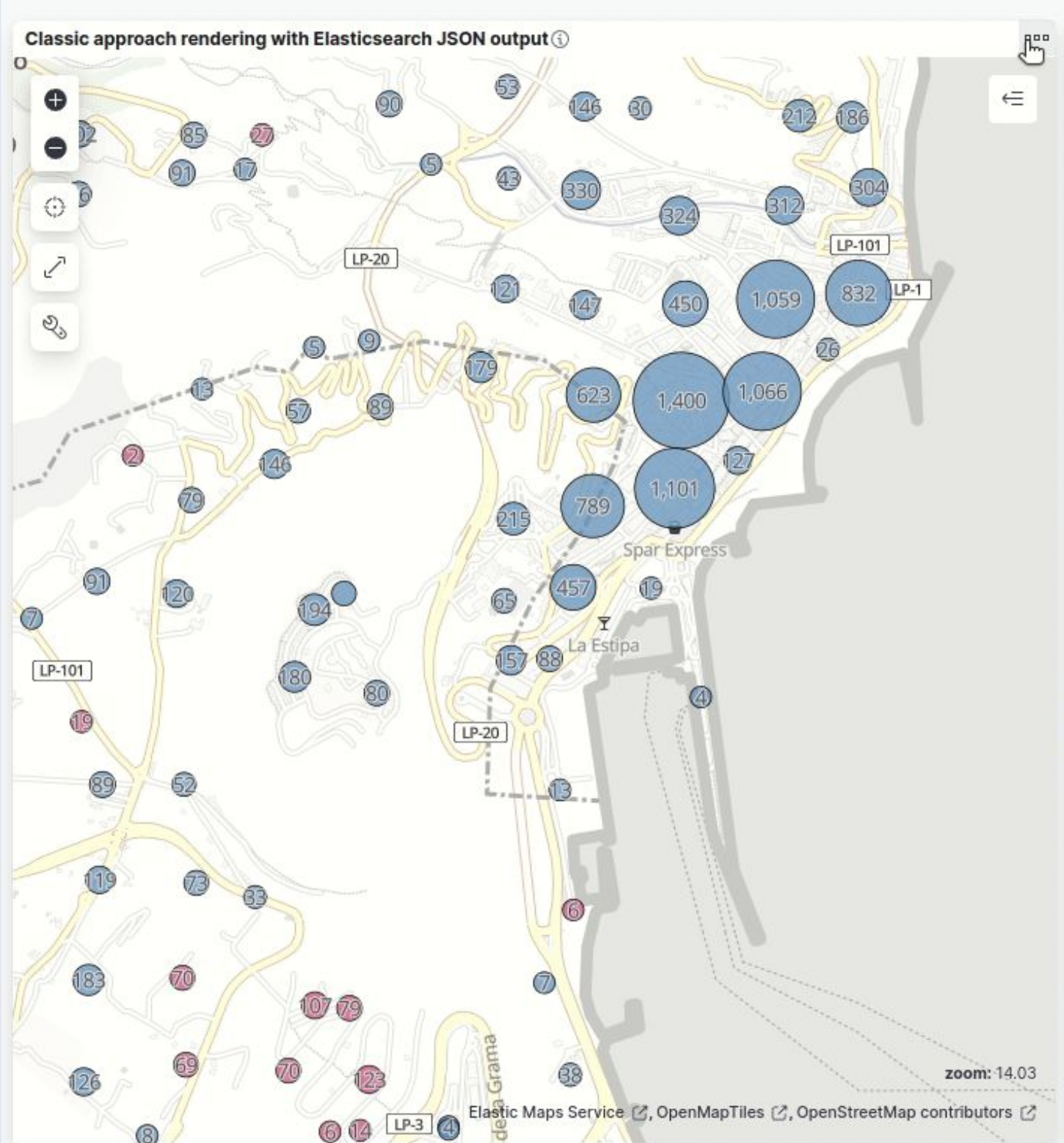
Modern approach

1. New `_mvt` endpoint 
2. Elasticsearch outputs mapbox **vector tiles** in *protobuf* format
3. Can render up to **10.000** documents per tile
4. Geometries are **simplified**
5. `meta` layer with **details**
6. Optional **label** positions



Demo

Comparing both approaches with
Kibana and Elastic Maps



Classic approach

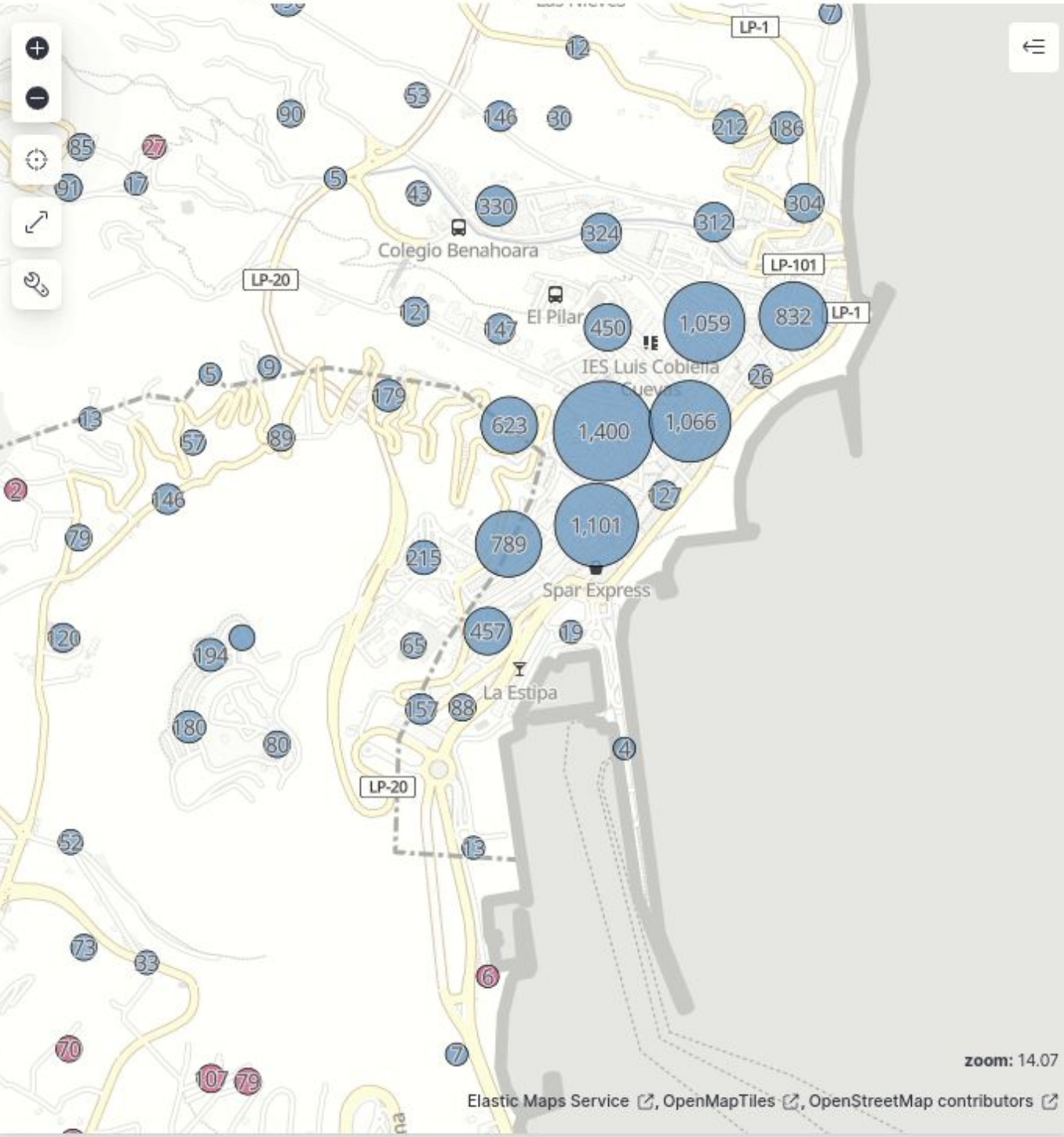
1. Browser requests features by **bounding box**
2. Middleware gathers **query parameters**: aggregation, search filters, etc
3. Database performs the query and returns **raw results**

```
1 GET all_lapalma_buildings/_search
2 {
3   "size": 5,
4   "fields": [
5     "geometry"
6   ],
7   "stored_fields": [
8     "geometry",
9     "name"
10  ],
11  "query": {
12    "bool": {
13      "must": [],
14      "filter": [
15        {
16          "bool": {
17            "must": [
18              {
19                "exists": {
20                  "field": "geometry"
21                }
22              },
23              {
24                "geo_bounding_box": {
25                  "geometry": {
26                    "top_left": [
27                      -17.77588000000003,
28                      28.69059
29                    ],
30                    "bottom_right": [
31                      -17.753910000000002,
32                      28.68095
33                    ]
34                  }
35                }
36              }
37            ]
38          }
39        }
40      ],
41      "should": [],
42      "must_not": [
43        {
44          "match_phrase": {
45            "name": "Edificio. Recinto."
46          }
47        }
48      ]
49    }
50  }
51 }
```

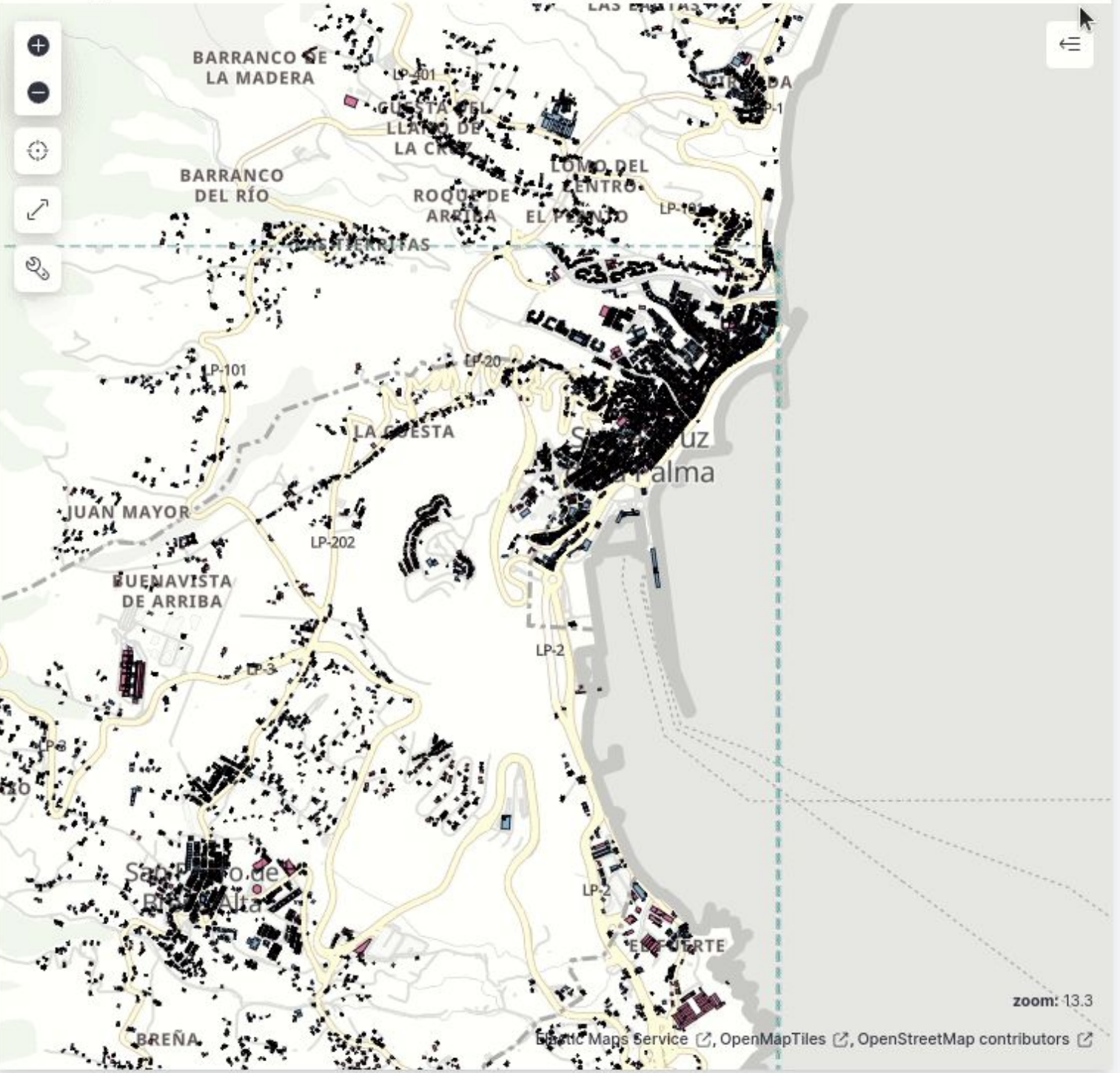
```
1 {
2   "took" : 3,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 9087,
13      "relation" : "eq"
14    },
15    "max_score" : 0.0,
16    "hits" : [
17      {
18        "_index" : "lapalma_buildings",
19        "_id" : "118297",
20        "_score" : 0.0,
21        "fields" : {
22          "geometry" : [
23            {
24              "coordinates" : [
25                [
26                  -17.7704412801182,
27                  28.6894360973088,
28                  92.5650000000023
29                ]
30              ],
31              "type" : "Polygon"
32            }
33          ]
34        }
35      },
36      {
37        "_index" : "lapalma_buildings",
38        "_id" : "118454",
39        "_score" : 0.0,
40        "fields" : {
41          "geometry" : [
42            {
43              "coordinates" : [
44                [
45                  -17.7704230195435,
46                  28.6894489108568,
47                  92.5650000000023
48                ]
49              ],
50              "type" : "Polygon"
51            }
52          ]
53        }
54      },
55      {
56        "_index" : "lapalma_buildings",
57        "_id" : "118455",
58        "_score" : 0.0,
59        "fields" : {
60          "geometry" : [
61            {
62              "coordinates" : [
63                [
64                  -17.7704225273318,
65                  28.6894492547039,
66                  92.5650000000023
67                ]
68              ],
69              "type" : "Polygon"
70            }
71          ]
72        }
73      }
74    ]
75  }
76 }
```

NOT name: Edificio. Recinto. x + Add filter

Classic approach rendering with Elasticsearch JSON output



Modern approach with vector tiles





1. New `_mvt` endpoint 
2. Elasticsearch outputs mapbox **vector tiles** in *protobuf* format
3. Can render up to **10.000** documents per tile
4. Geometries are **simplified**
5. `meta` layer with **details**
6. Optional **label** positions

```
1 POST /lapalma_buildings/_mvt/geometry/19
  /236070/218547
2 {
3   "grid_precision": 0,
4   "exact_bounds": true,
5   "extent": 4096,
6   "query": {
7     "bool": {
8       "filter": [],
9       "must": [],
10      "must_not": [
11        {
12          "match_phrase": {
13            "LNAME": "Edificio. Recinto."
14          }
15        }
16      ],
17      "should": []
18    }
19  },
20  "fields": [
21    "LNAME.keyword"
22  ],
23  "runtime_mappings": {},
24  "track_total_hits": 10001,
25  "with_labels": false
26 }
```

```

1 ▾ [
2 ▾   "meta": [
3 ▾     {
4 ▾       "geometry": {},
31 ▾       "properties": {
32 ▾         "_shards.failed": 0,
33 ▾         "_shards.skipped": 0,
34 ▾         "_shards.successful": 1,
35 ▾         "_shards.total": 1,
36 ▾         "hits.total.relation": "eq",
37 ▾         "hits.total.value": 33,
38 ▾         "timed_out": false,
39 ▾         "took": 9
40 ▾       }
41 ▾     }
42 ▾   ],
43 ▾   "hits": [
44 ▾     {
45 ▾       "geometry": {
46 ▾         "type": "Polygon",
47 ▾         "coordinates": []
75 ▾       },
76 ▾       "properties": {
77 ▾         "_id": "2hjCGoEB_-5IVzyuXS6U",
78 ▾         "_index": "lapalma_buildings",
79 ▾         "LNAME.keyword": "Cubierta plana. Recinto"
80 ▾       }
81 ▾     },
82 ▾     {
83 ▾       "geometry": {
84 ▾         "type": "Polygon",
85 ▾         "coordinates": []
129 ▾       },
130 ▾       "properties": {
131 ▾         "_id": "JhjCGoEB_-5IVzyuXS-U",
132 ▾         "_index": "lapalma_buildings",
133 ▾         "LNAME.keyword": "Cubierta inclinada. Recinto"
134 ▾       }
135 ▾     },
136 ▾     {
137 ▾       "geometry": {
138 ▾         "type": "Polygon"

```

But there's more than search

Tiles can contain aggregation
results



Maps

Create

Map settings

Inspect

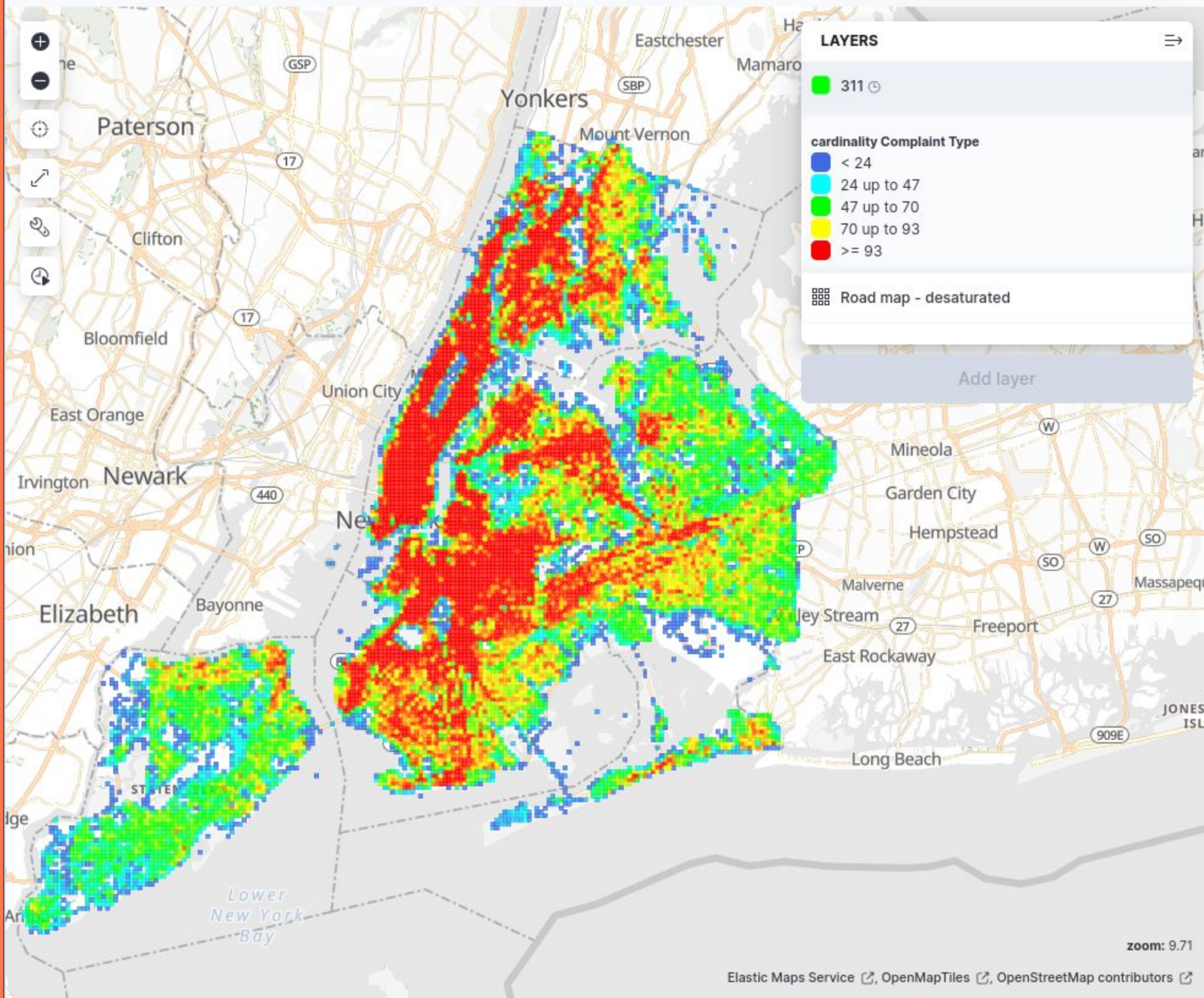
Full screen

 Save

Filter your data using KQL syntax



Last 13 years

 Refresh

311

> Source details

Layer settings

Name

Visibility

Zoom levels

0

→

24

Opacity



75

%

Attribution

[+ Add attribution](#)☒ Include layer in fit to data bounds computation

Metrics

Aggregation

Unique count



Field

Complaint Type



Custom label

[+ Add metric](#)

Clusters

Resolution

low

high

High resolution uses vector tiles.

Cancel

Remove layer

 Save & close



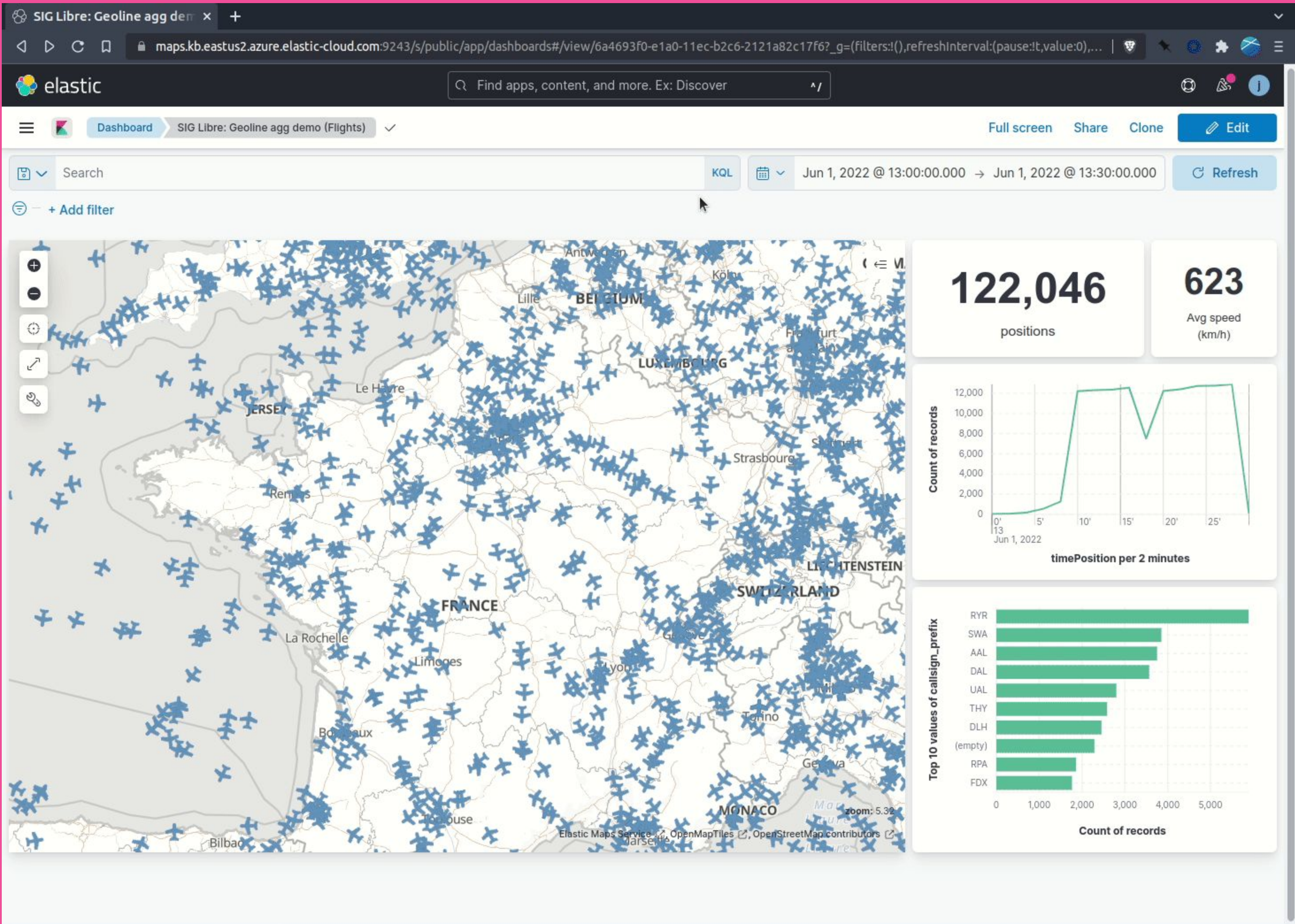
Modern approach

1. New `_mvt` endpoint 
2. Elasticsearch outputs mapbox **vector tiles** in *protobuf* format
3. Can render up to **10.000** documents per tile
4. Geometries are **simplified**
5. `meta` layer with **details**
6. Optional **label** positions

```
1 POST /geonames/_mvt/coordinates/1/0/1
2 {
3   "size": 0,
4   "grid_precision": 8,
5   "exact_bounds": false,
6   "extent": 4096,
7   "query": {
8     "bool": {
9       "filter": [
10        {
11          "bool": {
12            "minimum_should_match": 1,
13            "should": [
14              {
15                "match_phrase": {
16                  "feature_class": "P"
17                }
18              }
19            ]
20          }
21        },
22        {
23          "must": [],
24          "must_not": [],
25          "should": []
26        }
27      ],
28      "grid_agg": "geotile",
29      "grid_type": "centroid",
30      "aggs": {
31        "sum_of_population": {
32          "sum": {
33            "field": "population"
34          }
35        }
36      },
37      "fields": [
38        {
39          "field": "modification_date",
40          "format": "date_time"
41        }
42      ],
43      "runtime_mappings": {},
44      "with_labels": false
45    }
```

```
1 {
2   "meta": [
3     {
4       "geometry": {
5         "type": "Point",
6         "coordinates": [
7           ...
8         ]
9       },
10      "properties": {
11        "_key": "9/185/294",
12        "_count": 2020,
13        "sum_of_population.value": 195198
14      }
15    },
16    {
17      "geometry": {
18        "type": "Point",
19        "coordinates": [
20          ...
21        ]
22      },
23      "properties": {
24        "_key": "9/184/290",
25        "_count": 1544,
26        "sum_of_population.value": 71967
27      }
28    },
29    {
30      "geometry": {
31        "type": "Point",
32        "coordinates": [
33          ...
34        ]
35      },
36      "properties": {
37        "_key": "9/147/270",
38        "_count": 1199,
39        "sum_of_population.value": 188816
40      }
41    },
42    {
43      "geometry": {
44        "type": "Point",
45        "coordinates": [
46          ...
47        ]
48      },
49      "properties": {
50        "_key": "9/189/290",
51        "_count": 961,
52        "sum_of_population.value": 16762883
53      }
54    }
55  ]
56 }
```

New geospatial aggregations





Geo-Line aggregation

1. Typically combined with other aggs (terms or filters)
2. Up to 10.000 documents
3. GeoJSON encoding

```
1 GET flight_tracking_*/_search
2 {
3   "size": 0,
4   "track_total_hits": false,
5   "aggs": {
6     "tracks": {
7       "filters": {
8         "filters": {
9           "KLM1254": {
10            "match_phrase": {
11              "callsign.keyword": "KLM1254"
12            }
13          }
14        }
15      },
16      "aggs": {
17        "path": {
18          "geo_line": {
19            "point": {
20              "field": "location"
21            },
22            "sort": {
23              "field": "timePosition"
24            }
25          }
26        }
27      }
28    }
29  },
30  "query": {
31    "bool": {
32      "must": [],
33      "filter": [
34        {
35          "match_phrase": {
36            "callsign.keyword": "KLM1254"
37          }
38        },
39        {
40          "range": {
41            "timePosition": {
42              "format": "strict_date_optional_time",
43              "gte": "2022-06-01T11:00:00.000Z",
44              "lte": "2022-06-01T11:30:00.000Z"
45            }
46          }
47        }
48      ],
49      "should": [],
50      "must_not": []
51    }
52  }
53 }
```

```
1 {
2   "took" : 600,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 6,
6     "successful" : 6,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "max_score" : null,
12    "hits" : [ ]
13  },
14  "aggregations" : {
15    "tracks" : {
16      "meta" : { },
17      "buckets" : {
18        "KLM1254" : {
19          "doc_count" : 19,
20          "path" : {
21            "type" : "Feature",
22            "geometry" : {
23              "coordinates" : [ [ ] ],
24              "type" : "LineString"
25            },
26            "properties" : {
27              "complete" : true
28            }
29          }
30        }
31      }
32    }
33  }
34 }
```



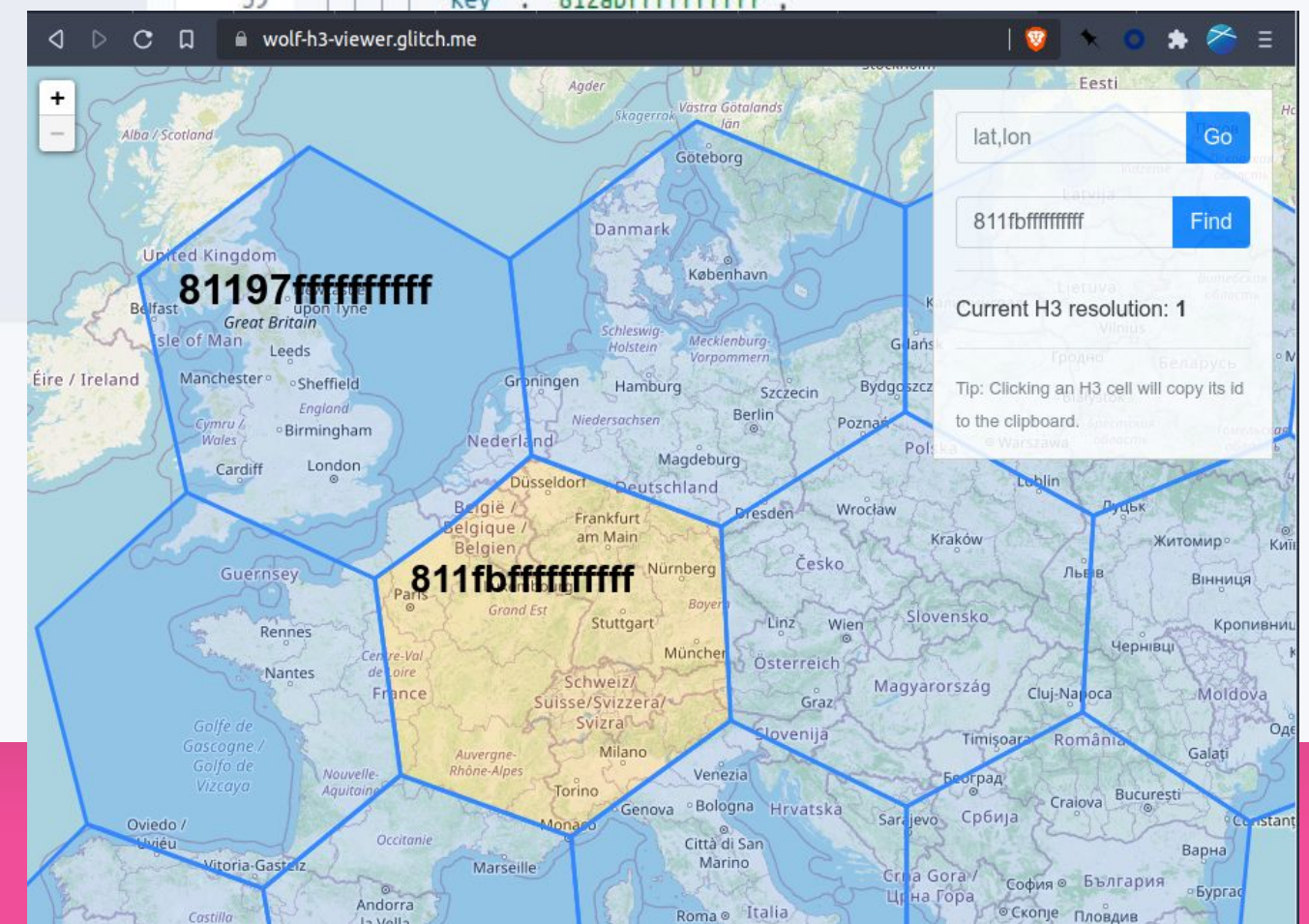


Hexgrid aggregation

1. Port of Uber's H3 algorithm to Java
2. High precision aggs needs combination with bounding box filter
3. Only for geo_point type

```
1 GET flight_tracking*/_search
2 {
3   "size": 0,
4   "query": {
5     "bool": {
6       "filter": [
7         {
8           "query_string": {
9             "default_field": "callsign.keyword",
10            "query": "NOT (\\\"\\\" OR 00000000)"
11          }
12        }
13      ]
14    },
15  },
16  "aggs": {
17    "large-grid": {
18      "geo_hex_grid": {
19        "field": "location",
20        "precision": 1
21      },
22      "aggs": {
23        "avg_speed": {
24          "avg": {
25            "field": "velocity"
26          }
27        },
28        "mostCommonCallsign": {
29          "terms": {
30            "field": "callsign.keyword",
31            "size": 1
32          }
33        }
34      }
35    }
36  }
37 }
```

```
1 {
2   "took": 1892,
3   "timed_out": false,
4   "shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 0,
12    "max_score": null,
13    "hits": []
14  },
15  "aggregations": {
16    "large-grid": {
17      "buckets": [
18        {
19          "key": "811fbfffffffff",
20          "doc_count": 248615,
21          "avg_speed": {
22            "value": 150.88583386167406
23          },
24          "mostCommonCallsign": {
25            "doc_count_error_upper_bound": 369,
26            "sum_other_doc_count": 242944,
27            "buckets": [
28              {
29                "key": "DLZNT",
30                "doc_count": 209
31              }
32            ]
33          }
34        }
35      ]
36    }
37  },
38  "key": "812abfffffffff",
39 }
```



Outcomes



Avoid rendering from Search API

Leverage the Vector Tile API to enable snappy mapping experiences combined with modern web-mapping libraries.



Aggregate

Consider avoiding individual document rendering until the last moment. Aggregate your data into clusters, heatmaps, grids, etc.



Follow Elastic Maps practice

Leverage the “Inspect” tool in Kibana to learn how to craft effective queries that are validated by thousands of users.

Resources

- Relevant documentation
 - [Vector Tiles API](#)
 - [Geo-Line Aggregation](#)
 - [GeoHex Aggregation](#)
- Use cases, trainings, guides, ...
 - elastic.co/geospatial
- FOSS4G 2021 Elastic Stack talk
 - [recording](#)
 - [slides](#)
- Elastic community:
 - Discussions: discuss.elastic.co
 - Slack: ela.st/slack
 - Github: github.com/elastic



¡Gracias!

Jorge Sanz - jorge.sanz@elastic.co

2022-06-08

<https://ela.st/siglibre2022>