

adelean

EXTRACT TRANSFORM SEARCH



Elasticsearch : What if your database was a search engine?



Benjamin Dauvissat (@bdauvissat)

Software Engineer / Podcaster / Sometimes speaker / Eternal learner

Who are we ?

🔍 Adelean

- 🔍 Experts in **Search** technologies
- 🔍 Integrators of **Elasticsearch**, **OpenSearch** and **Solr**
- 🔍 **Consulting** and **Training** providers
- 🔍 Developers of **a2 - E-commerce** and **Enterprise Search** solution
- 🔍 Developers of **all.site** - your **Collaborative Search Engine**



This talk is an adaptation

- First given in French at ElasticON Global by Lucian Precup



Databases ranking (popularity)

db-engines.com

410 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1247.52	+2.35	-9.31
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	616.50	+1.65	+7.12
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	452.77	-2.42	-35.88
6.	6.	6.	Redis +	Key-value, Multi-model ⓘ	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model ⓘ	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model ⓘ	138.60	-2.56	-23.70
9.	↑ 10.	↑ 10.	SQLite +	Relational	132.67	+1.17	+4.30
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	131.03	-2.33	-0.23
11.	↑ 12.	11.	Cassandra +	Wide column	116.22	-0.09	-7.76
12.	↓ 11.	↑ 15.	Snowflake +	Relational	115.65	-1.60	+32.47
13.	13.	↓ 12.	MariaDB +	Relational, Multi-model ⓘ	96.81	-2.55	-10.30
14.	14.	↓ 13.	Splunk	Search engine	87.08	-1.32	-3.73
15.	15.	↑ 17.	Amazon DynamoDB +	Multi-model ⓘ	79.69	-1.87	-0.67
16.	16.	↓ 14.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ	78.75	-1.62	-6.20
17.	17.	↓ 16.	Hive	Relational	72.12	-2.22	-9.76
18.	18.	18.	Teradata	Relational, Multi-model ⓘ	63.03	-2.40	-5.54
19.	19.		Databricks	Multi-model ⓘ	60.33	-0.49	
20.	20.	20.	Neo4j +	Graph	55.43	-0.41	-2.81

<https://db-engines.com/en/ranking>

Types of databases

Key Value

Document

Relational

Search engine

Graph

Time series

Spatial

Columns

Types of databases

Relational

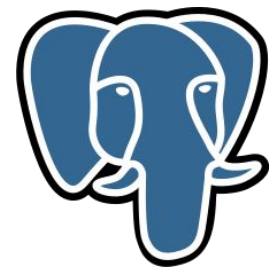
ORACLE



Key Value



Document



Graph



Time series



Spatiale



Column

What type of database for Elasticsearch?

Key Value

Document

Relational



elasticsearch

Graph

Search engine

Time series

Spatial

Columns

What type of database for Elasticsearch?

▸ **Search engine**

▸ Column

▸ Spatial

▸ Time series



▸ Graph

▸ Document

▸ Key Value

▸ Relational

Elasticsearch – Search engine?

Yes

- Based on Apache Lucene
- Used in thousands of search engines (companies, e-shop, data-lakes, ...)
- Thirty to forty search engines conceived, built or enhanced by Adelean every year since 2010
- ...

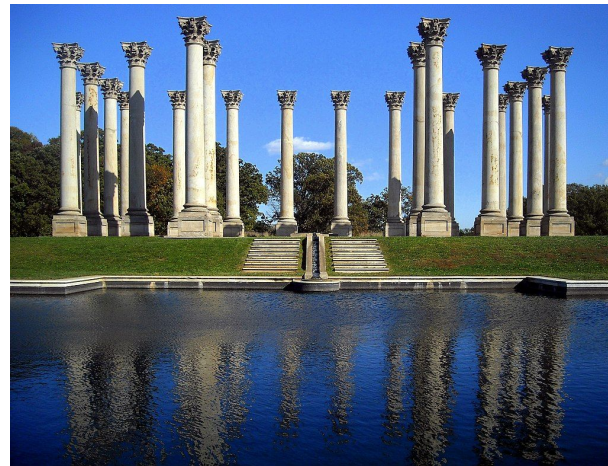
What type of database for Elasticsearch?

▸ Search engine

▸ **Column**

▸ Spatial

▸ Time series



▸ Graph

▸ Document

▸ Key Value

▸ Relational

Elasticsearch – Column oriented database?

Yes

- Apache Lucene – basically document oriented, but
- Since Lucene 4.0 there are DocValues
- Hence very good performances in sorting and aggregating



What type of database for Elasticsearch?

- Search engine
- Column
- **Spatial**
- Time series

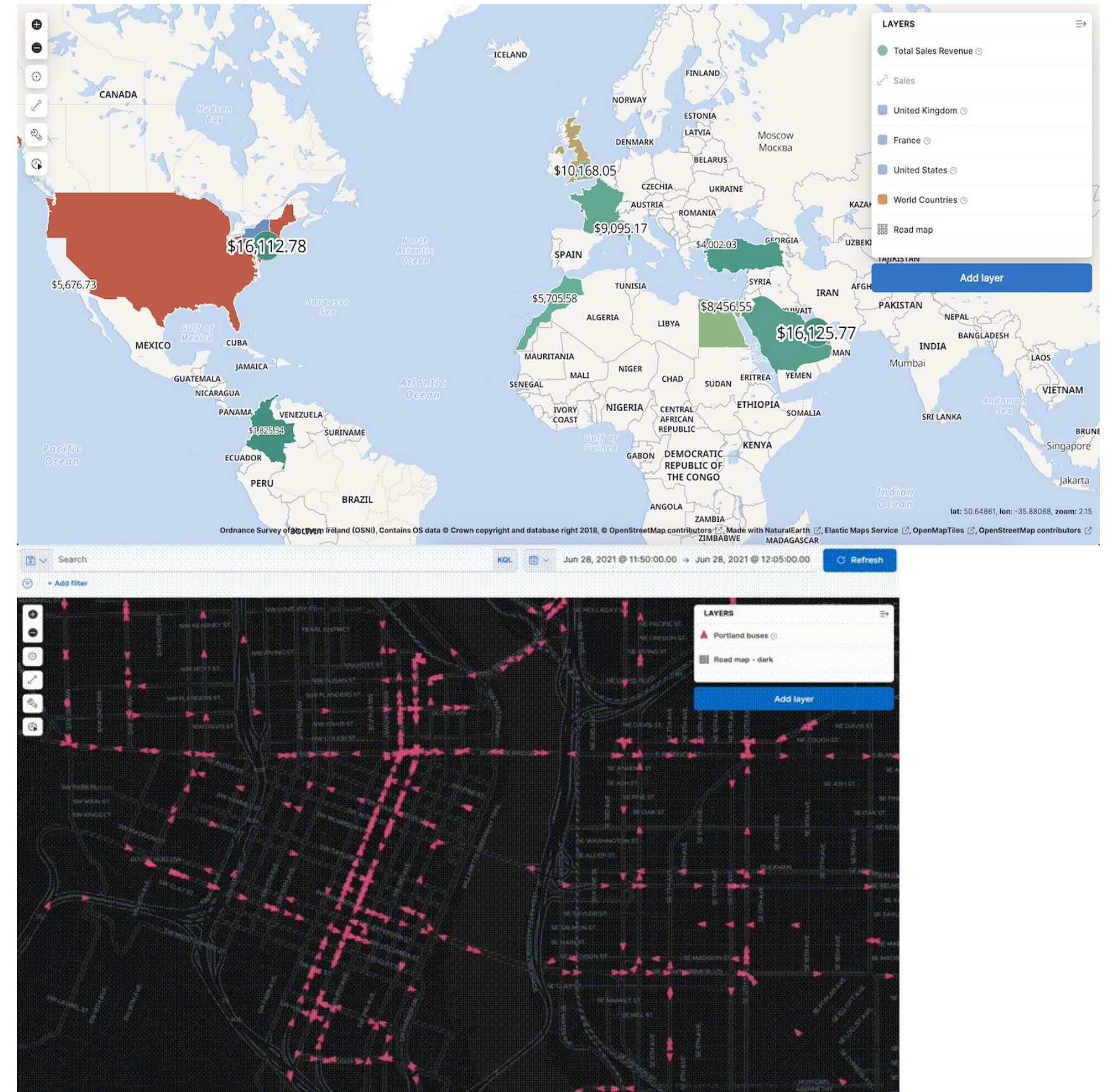


- Graph
- Document
- Key Value
- Relational

Elasticsearch – Spatial database?

Yes

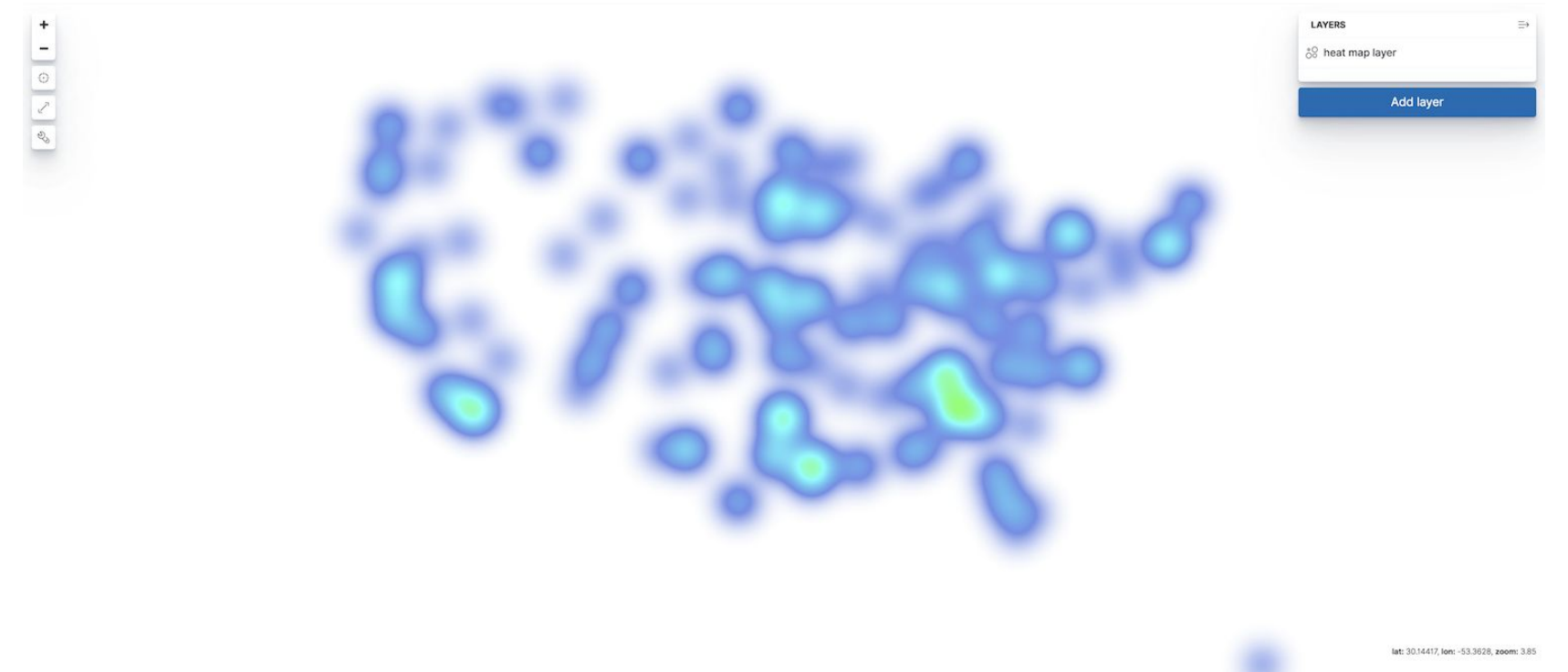
- Geo Point data type
- Geo Shape type
- Excellent support for Geo Point (latitude / longitude objects, latitude / longitude pairs, latitude / longitude arrays, Geo Hash)



Elasticsearch – Spatial database?

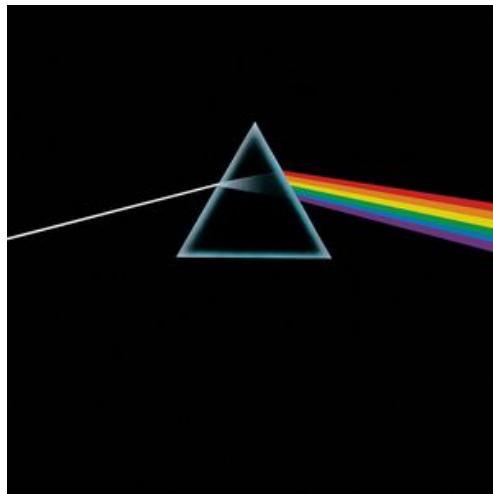
Yes

- Excellent support for Geo Shape (dots, lines, polygons)
- Spatial data requesting (Geo Shape Query, Geo Bounding Box Query, Geo Distance Query, Geo Polygon Query)
- Kibana Maps and the user-friendly support for spatial data in Kibana



What type of database for Elasticsearch?

- Search engine
- Column
- Spatial
- **Time series**

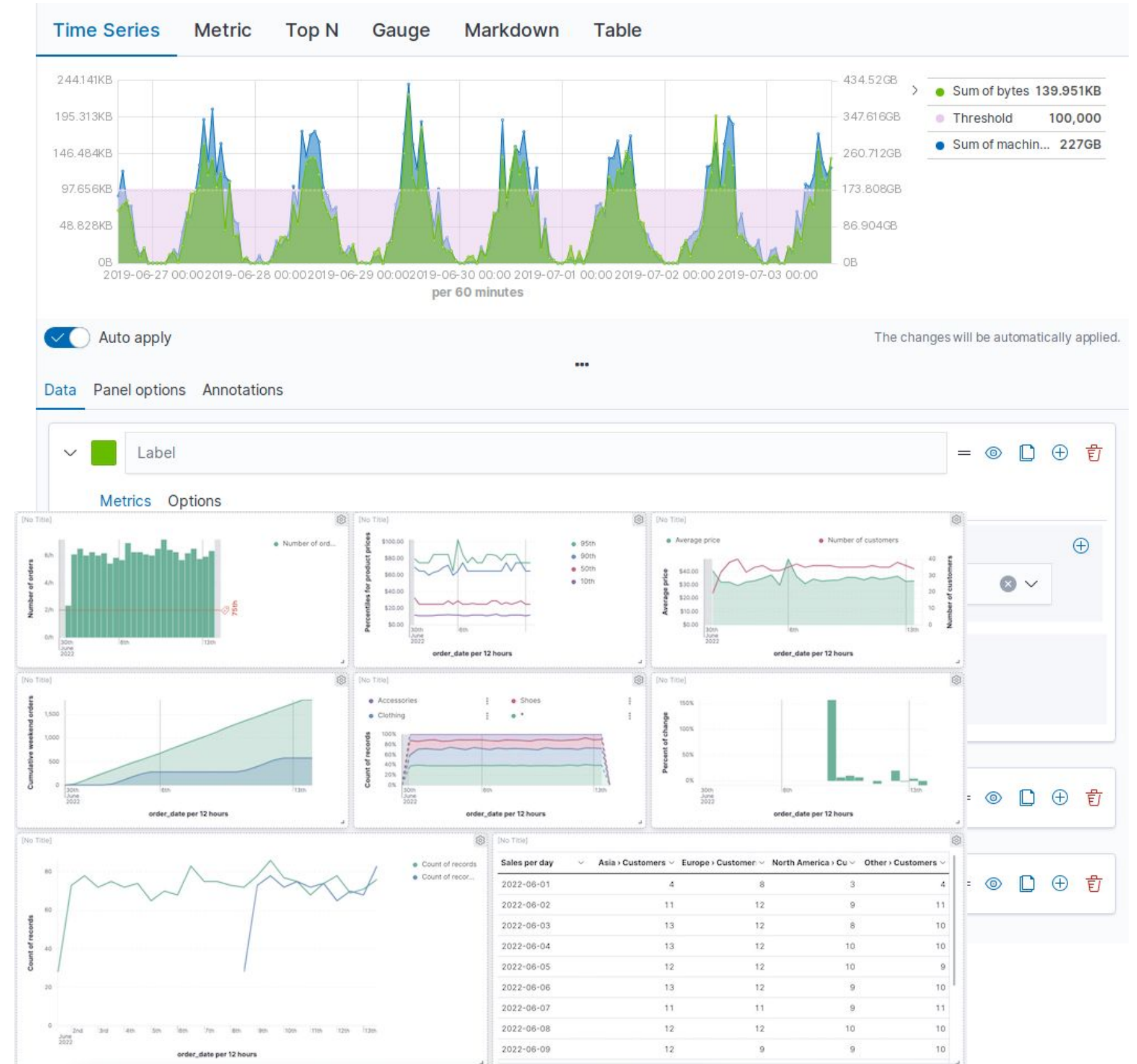


- Graph
- Document
- Key Value
- Relational

Elasticsearch – Time series database?

Yes

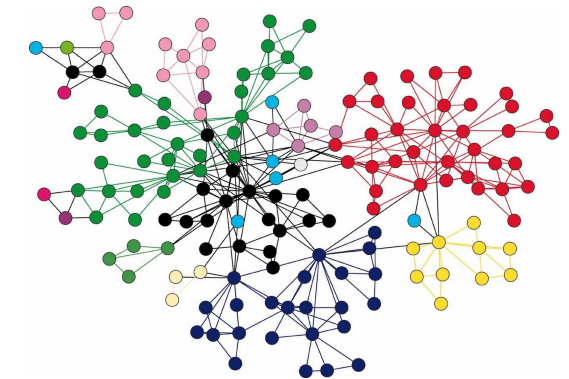
- Data Streams
- ILM and Index Lifecycle Policy
- Cold et Frozen storage
- Runtime fields
- Kibana ability to visualize data
- Time Series Visual Builder (legacy)
- Timelion, Kibana Lens, TSVB, ...



What type of database for Elasticsearch?

- Search engine
- Column
- Spatial
- Time series

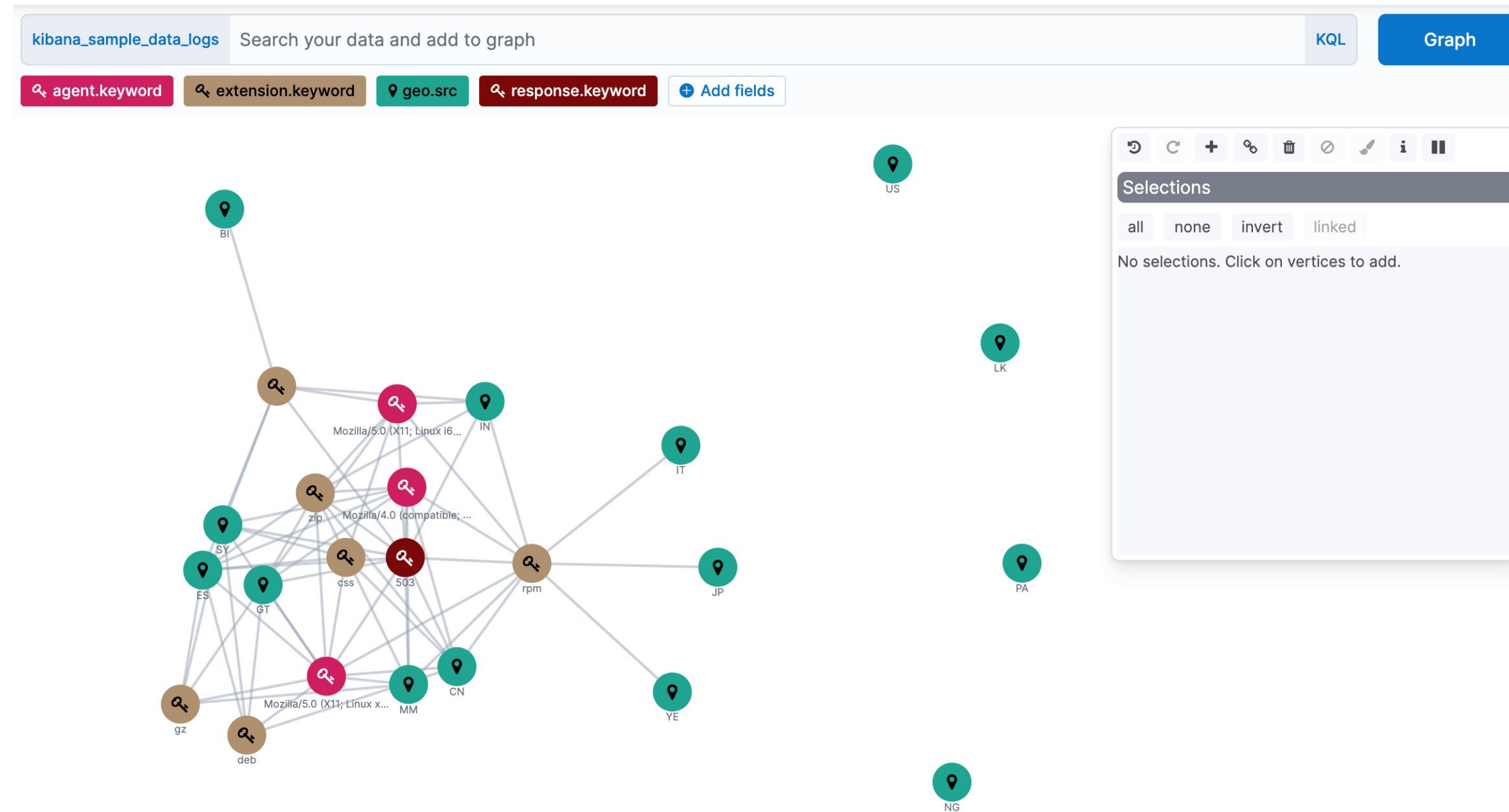
- **Graph**
- Document
- Key Value
- Relational



Elasticsearch – Graph database?

Yes

- Graph Explore API
- Graph UI in Kibana



What type of database for Elasticsearch?

- Search engine
- Column
- Spatial
- Time series
- Graph
- **Document**
- Key Value
- Relational



Elasticsearch – Document database?

Yes

-PUT products/_doc/1

```
{
  "article": "T-Shirt",
  "brand": "Nike",
  "skus": [
    {
      "size": "M",
      "color": "blue"
    },
    {
      "size": "L",
      "color": "red"
    }
  ]
}
```

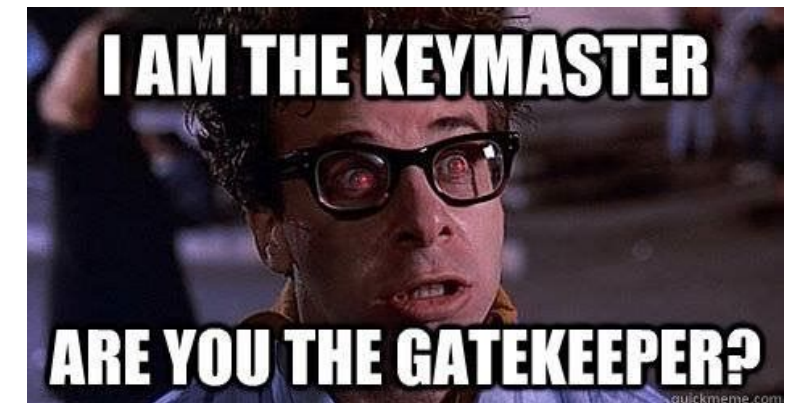
-GET products/_doc/1

-But also ... HEAD, DELETE, Multi get, Bulk API, _source



What type of database for Elasticsearch?

- Search engine
- Column
- Spatial
- Time series
- Graph
- Document
- **Key Value**
- Relational



Elasticsearch – Key Value database?

Yes

- GET products/_doc/1
- _routing, _tier fields
- GET /_search (*real time)

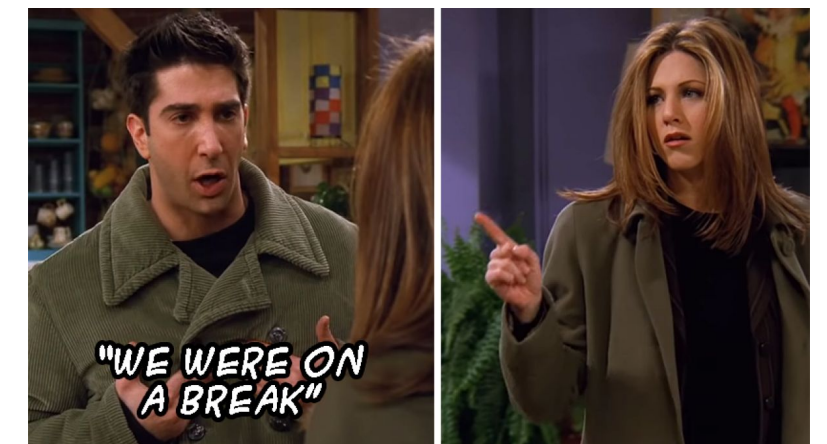
```
{
  "query": {
    "bool": {
      "filter": {
        "term": {
          "user.id": {
            "value": "hash"
          }
        }
      }
    }
  }
}
```

*near real time and unique key managed by the application



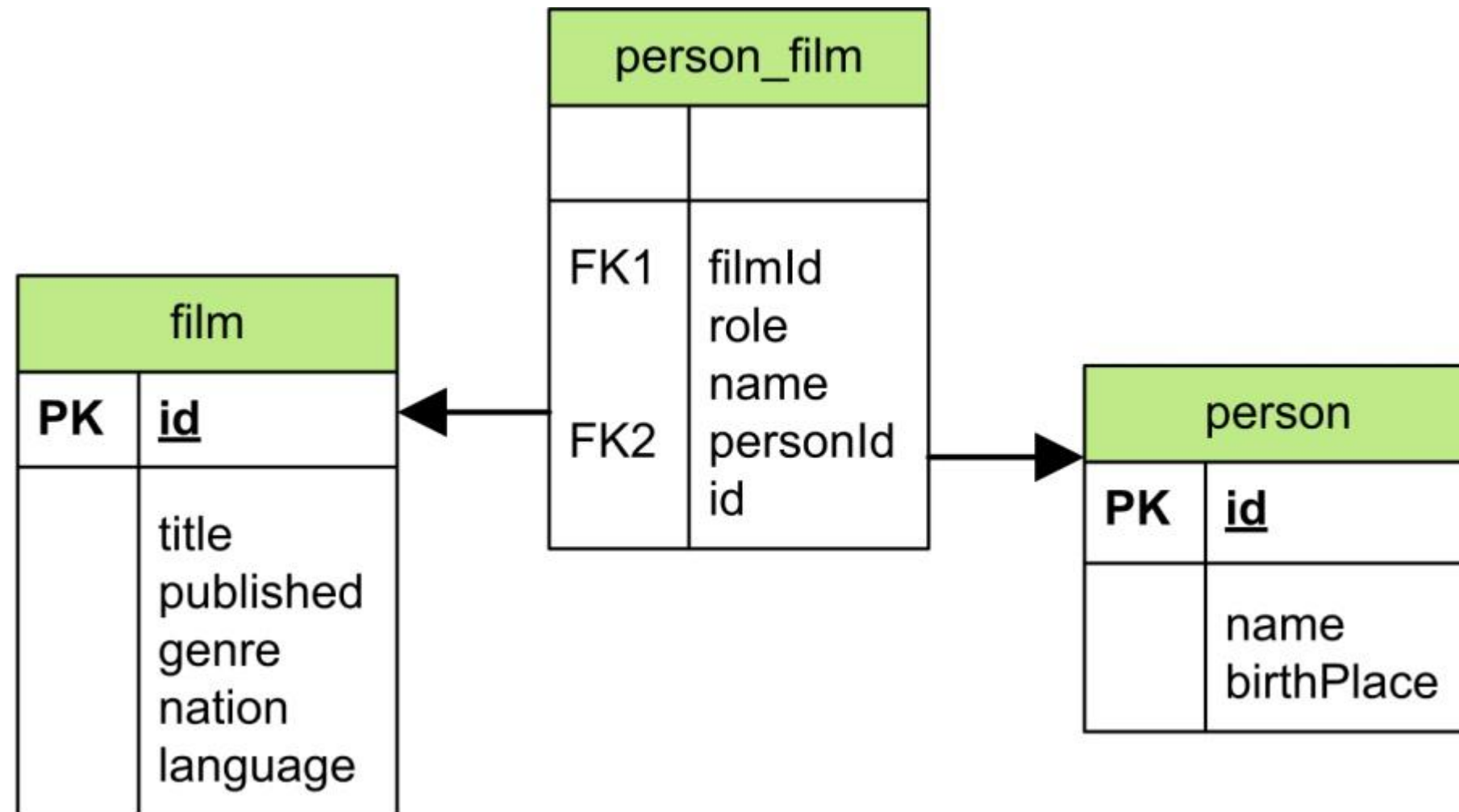
What type of database for Elasticsearch?

- Search engine
- Column
- Spatial
- Time series
- Graph
- Document
- Key Value
- **Relational**



Elasticsearch – Relational database?

Yes



```
{
  "film" : {
    "id" : "183070",
    "title" : "The Artist",
    "published" : "2011-10-12",
    "genre" : ["Romance", "Drama", "Comedy"],
    "language" : ["English", "French"],
    "persons" : [
      {"person" : { "id" : "5079", "name" : "Michel Hazanavicius", "role" : "director" }},
      {"person" : { "id" : "84145", "name" : "Jean Dujardin", "role" : "actor" }},
      {"person" : { "id" : "24485", "name" : "Bérénice Bejo", "role" : "actor" }},
      {"person" : { "id" : "4204", "name" : "John Goodman", "role" : "actor" }}
    ]
  }
}
```


Elasticsearch – Relational database?

Yes

- Elasticsearch SQL !
- Nested type, request and aggregation
- *join* type and *parent-child* aggregations and requests (multi-level)



ACID

Ensure consistency when running a sequence of database operations

ACID transactions:

- Atomicity
- Consistency
- Isolation
- Durability



CAP theorem

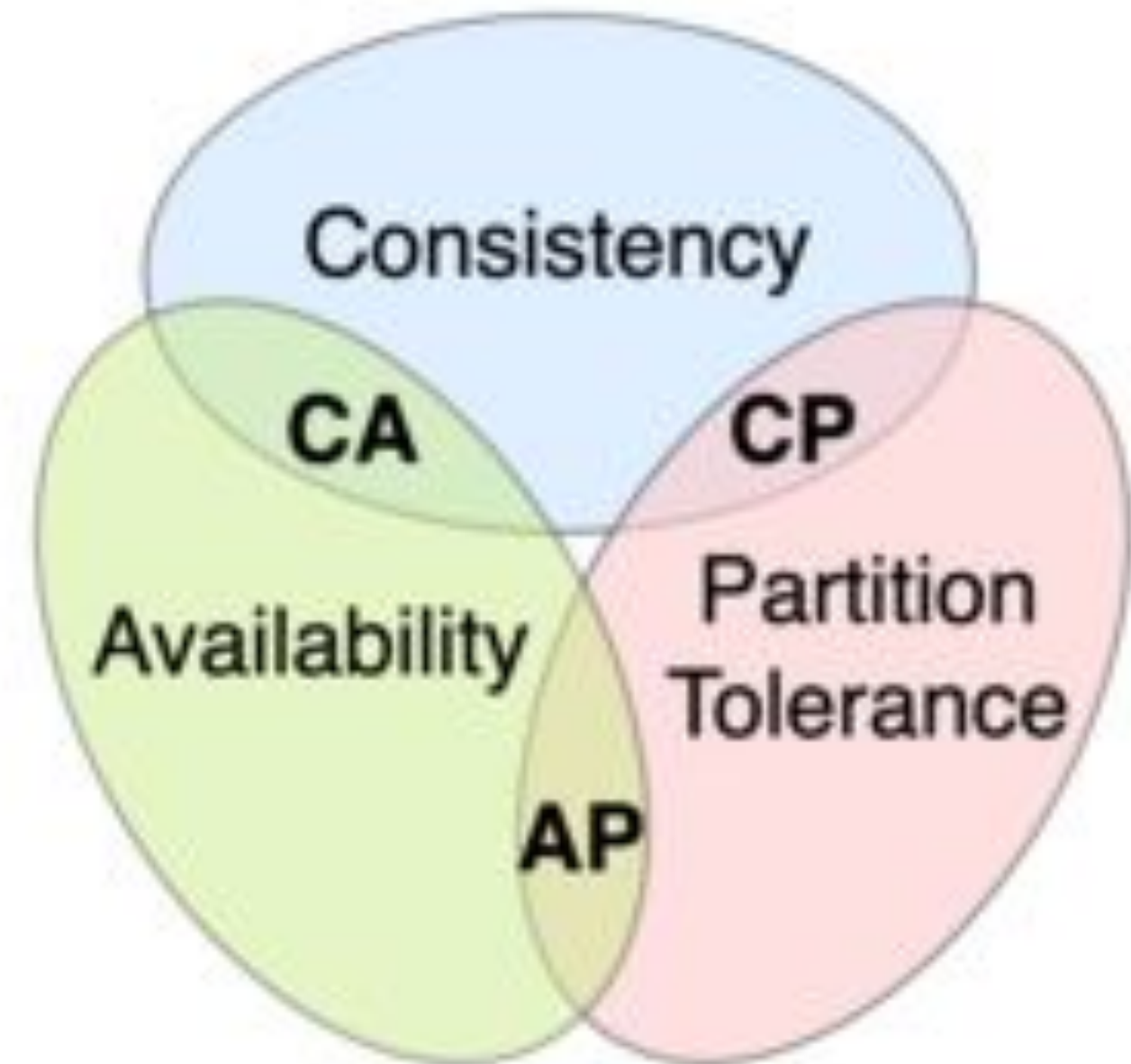
Welcome to distributed systems 😊

CAP

- Consistency
- Availability
- Partition Tolerance

BASE (NoSQL)

- Basically Available
- Soft-state
- Eventually consistent



Elasticsearch – NoSQL database?

Yes ! Even if NewSQL is not there (yet)

C

PUT <index>/_doc/<_id>

GET <index>/_doc/<_id>

POST /<index>/_doc/

R

PUT /<index>/_create/<_id>

DELETE /<index>/_doc/<_id>

U

POST /<index>/_update/<_id>

GET /_mget

POST /_bulk

D

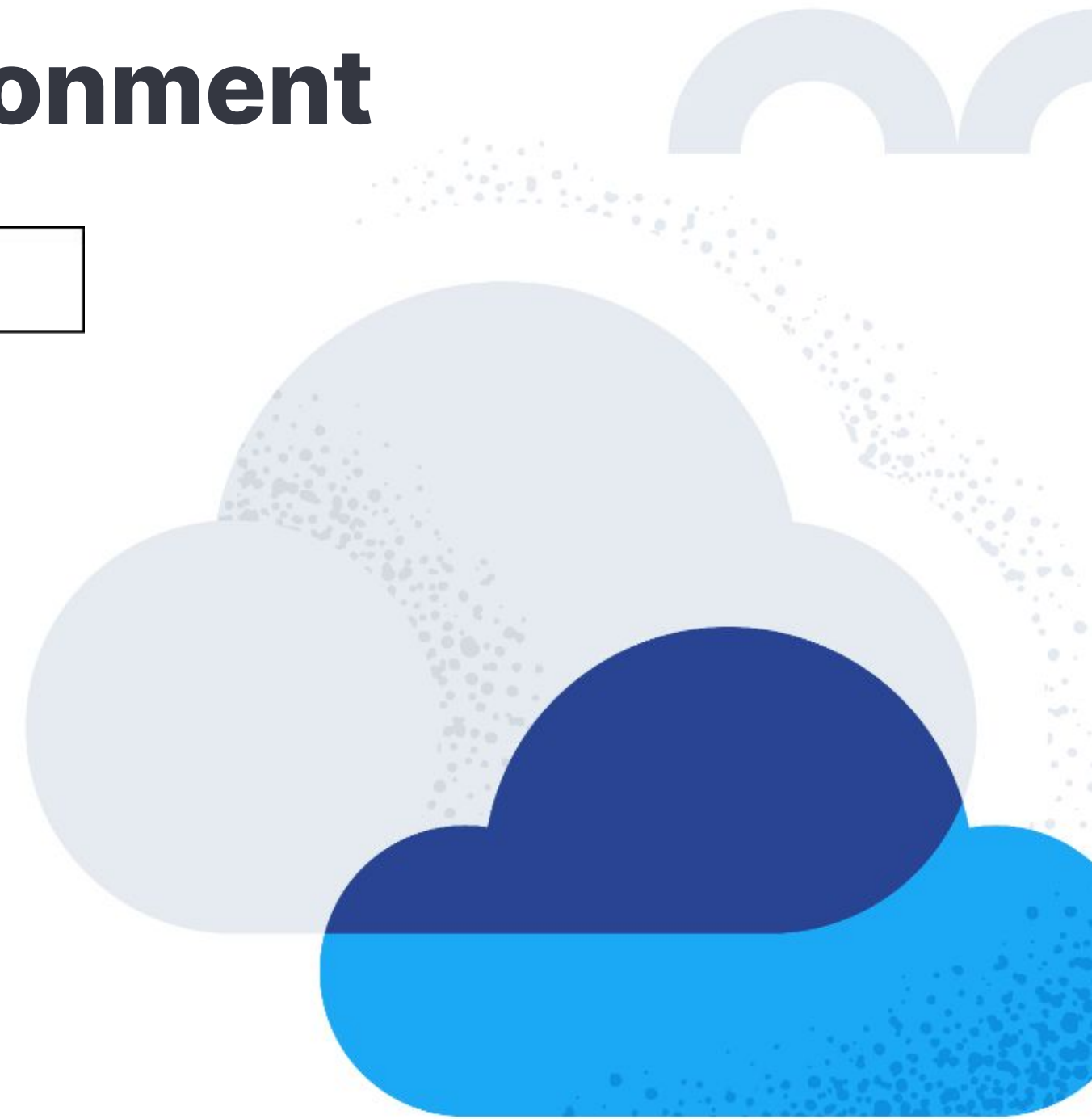
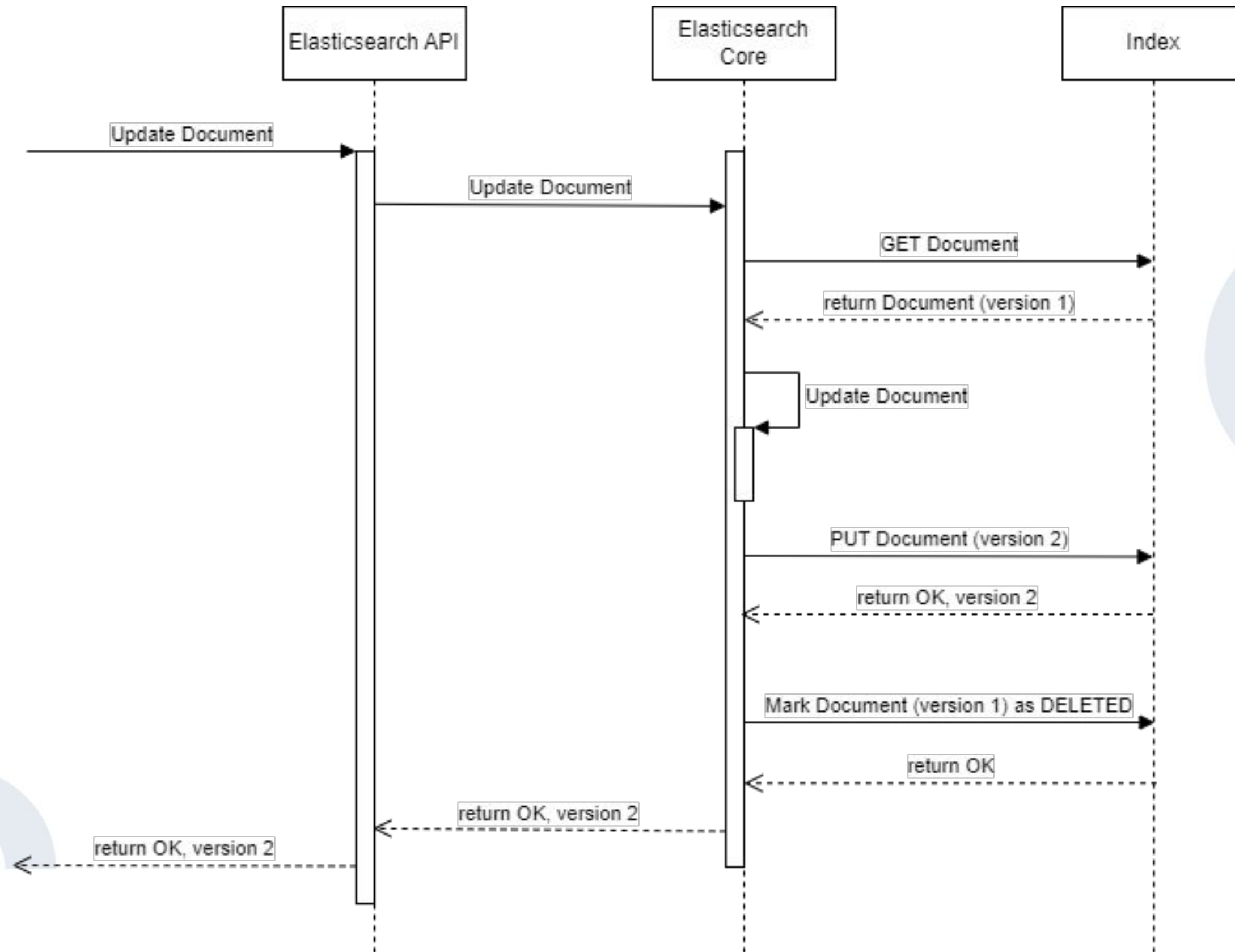
POST /<target>/_delete_by_query

POST /<target>/_update_by_query

POST /_reindex



Update operations in a NoSQL environment



Handling conflicts during concurrent updates

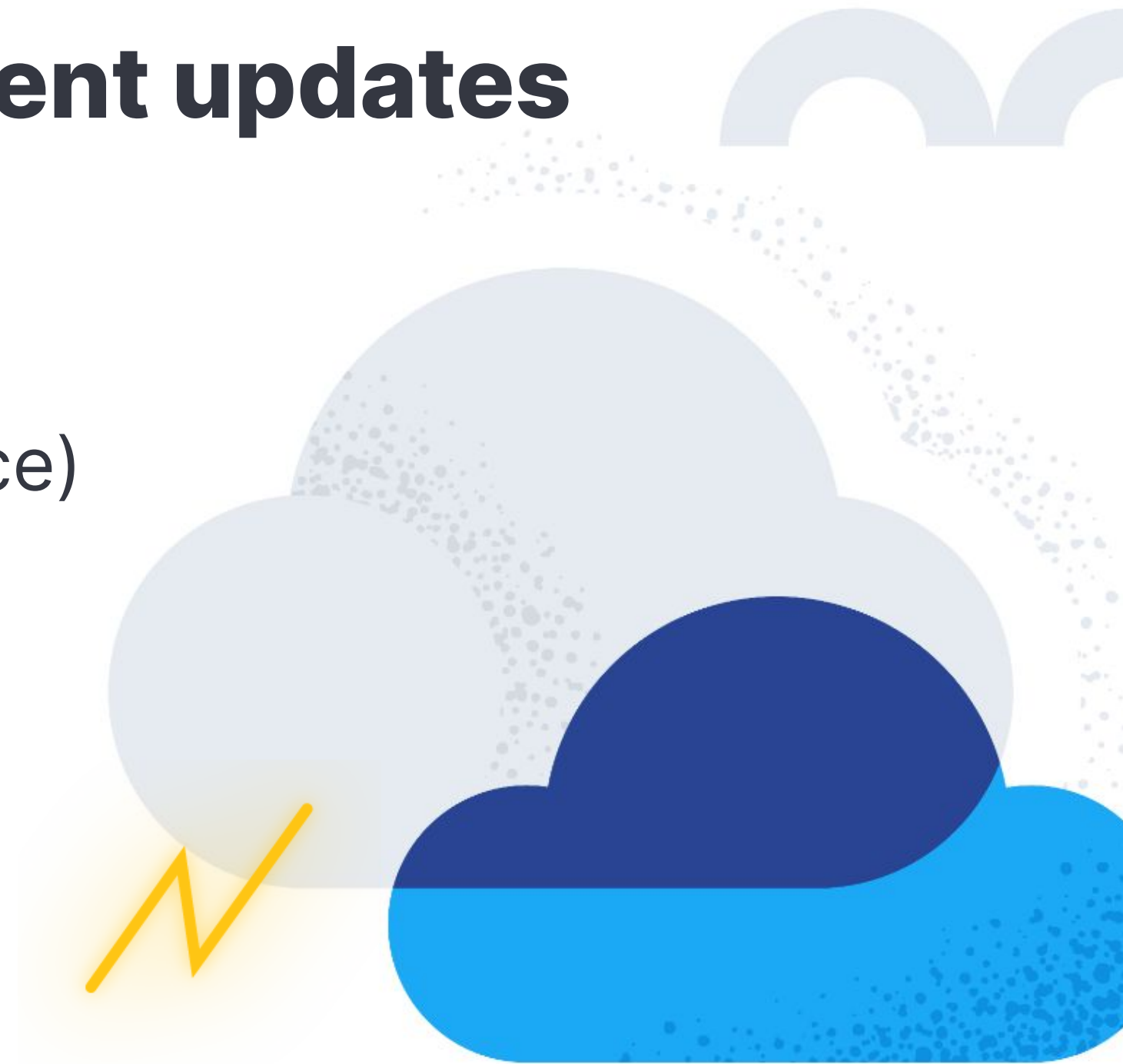
Problem and explanation

Problem :

- Concurrent updates (in bulk mode, for instance) can generate errors of the following kind :
version conflict, current version is higher or equal to the one provided

Explanation :

- Update NoSQL = GET, update and PUT.
- NoSQL systems don't lock the data during update (like relational databases)
- Version number providing « optimistic locking » mechanism



Working with optimistic locking

Solutions and workaround in Elasticsearch

Solutions :

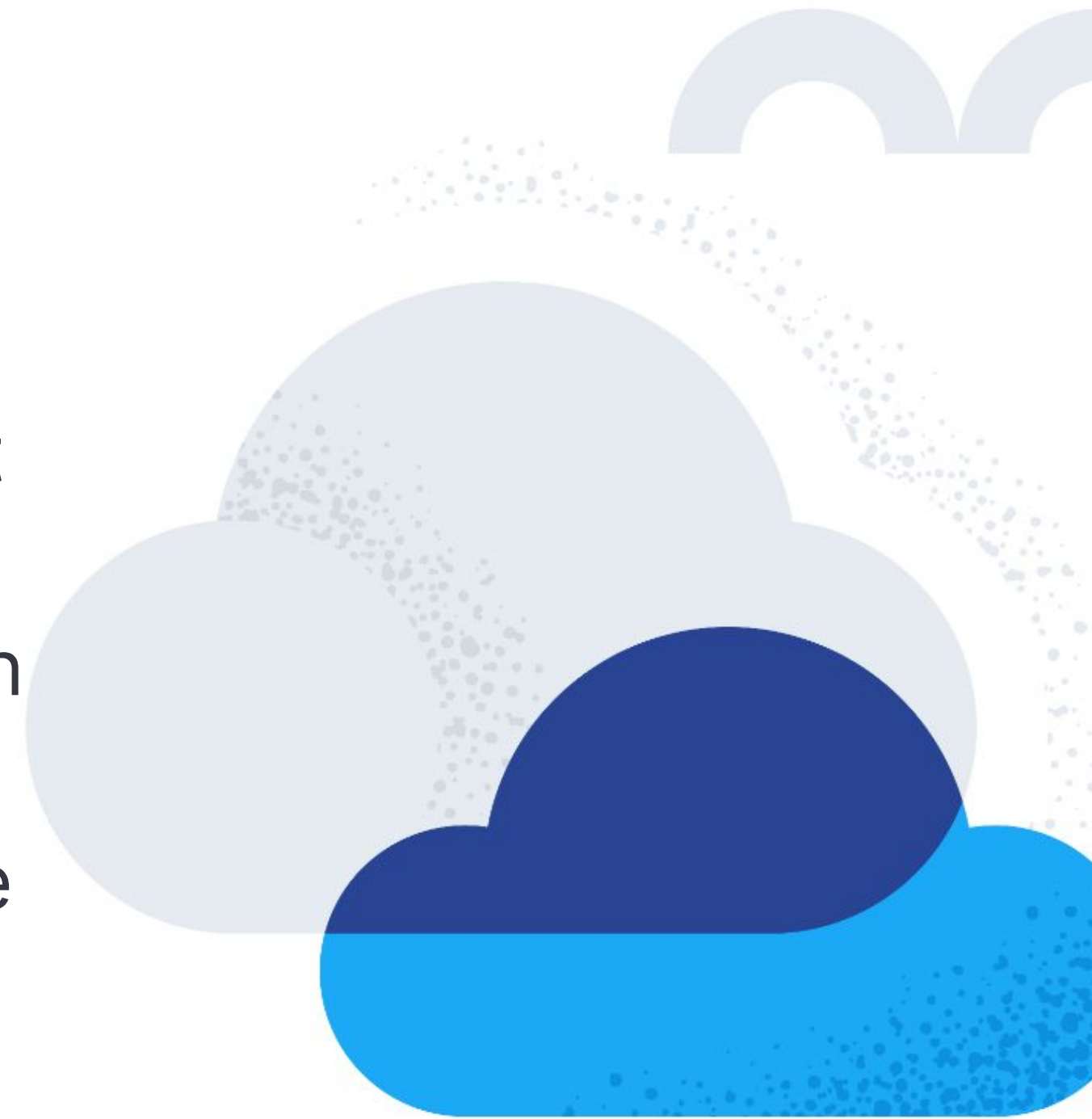
The application manages how it handles the conflict

It can :

- Ignore the previous version et and overwrite with its own version
- Perform the update once again but based on the newest version
- Cancel its modifications

Quick workaround :

- [retry_on_conflict](#) parameter: How many times Elasticsearch retries the update, based on the new



Elasticsearch has some weaknesses

- Not really transactional
- Near real-time
- No joins like SQL
- No transactions nor rollbacks

Elasticsearch has many strengths

- Flexible query API
- Horizontal scalability
- Aggregations
- Wide range of usages
- Designed for many purposes (SIEM, logs, maps, etc.)

What about you? How do you use Elasticsearch?

Key value

Document

Relational



elasticsearch

Graph

Search engine

Time series

Spatial

Columns



adelean

EXTRACT TRANSFORM SEARCH



www.adelean.com



info@adelean.com



[@a2lean](https://twitter.com/a2lean)



[linkedin.com/company/adelean](https://www.linkedin.com/company/adelean)



Thank you!
Merci !

HAYSTACK