Aalto University
School of Science

CS-E4610
Modern Database Systems
05.01.2018-05.04.2018

Tutorial 2

Unstructured text. IR systems, document retrieval and ranking.

FREDERICK AYALA-GÓMEZ
PHD STUDENT IN COMPUTER SCIENCE, ELTE UNIVERSITY
VISITING RESEARCHER, AALTO UNIVERSITY

Agenda

Why is it important?

45 Mins

What is ElasticSearch?

10 Mins

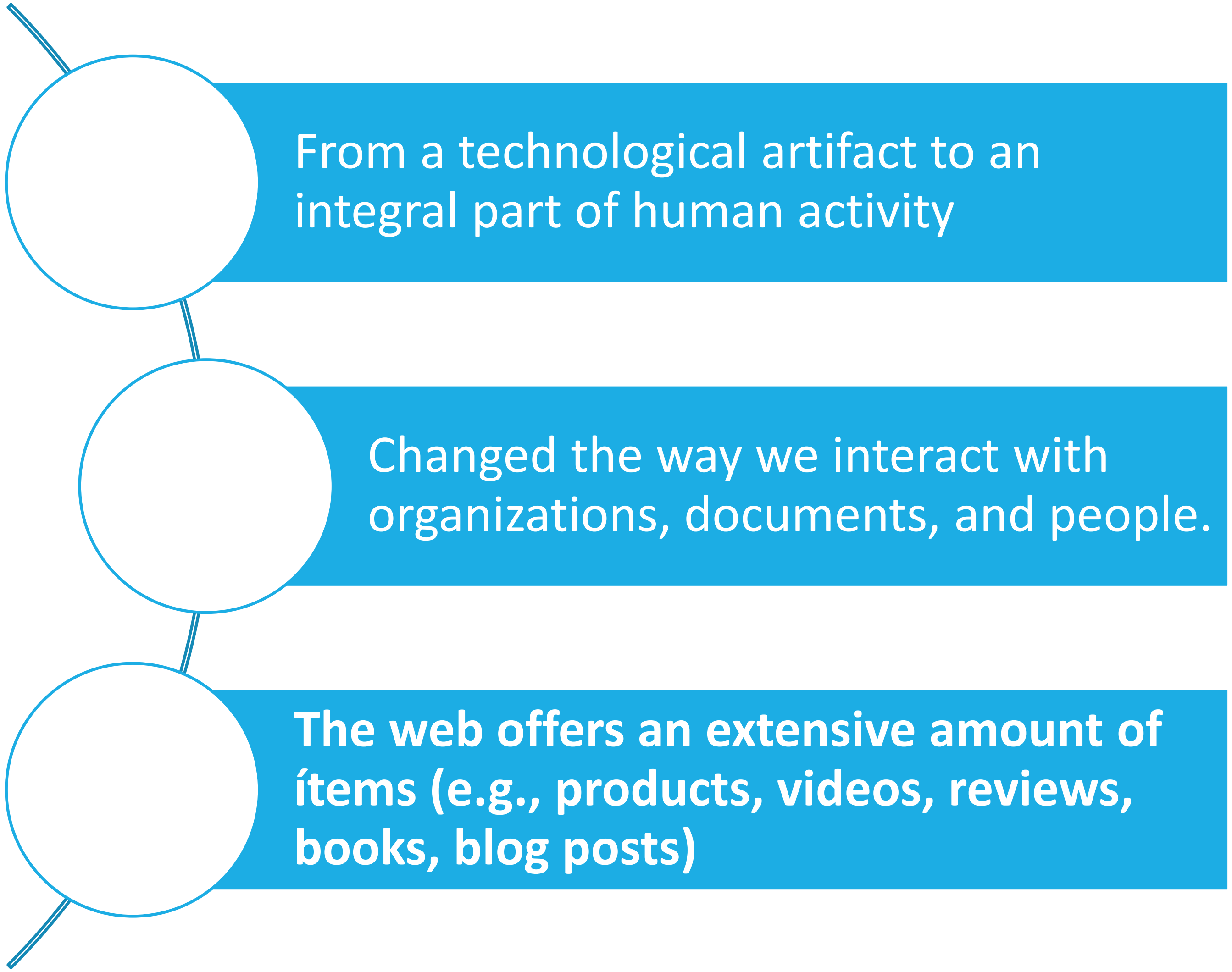
- break -

45 Mins

How to use it? (Tutorial)

The World Wide Web

Hall, W., & Tiropanis, T. (2012). Web evolution and Web science. *Computer Networks*, *56*(18), 3859-3865.

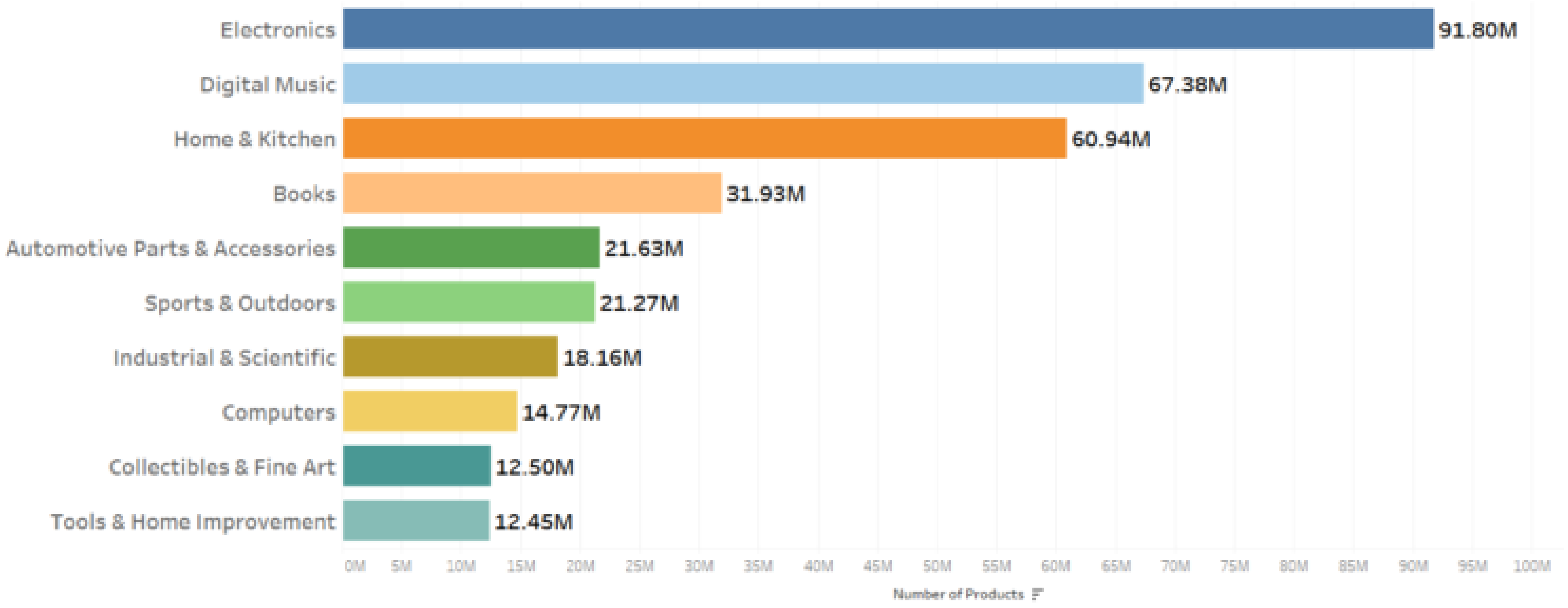


From a technological artifact to an integral part of human activity

Changed the way we interact with organizations, documents, and people.

The web offers an extensive amount of ítems (e.g., products, videos, reviews, books, blog posts)

Amazon.com has almost 400M Products



https://www.scrapehero.com/how-many-products-are-sold-on-amazon-com-january-2017-report/

Online Encyclopedias

~5.5M Articles

~4M Articles

https://en.wikipedia.org/wiki/Wikipedia:Size_comparisons

Social Media

You name it

HELSINGIN SANOMAT

K Market

ILTALEHTI

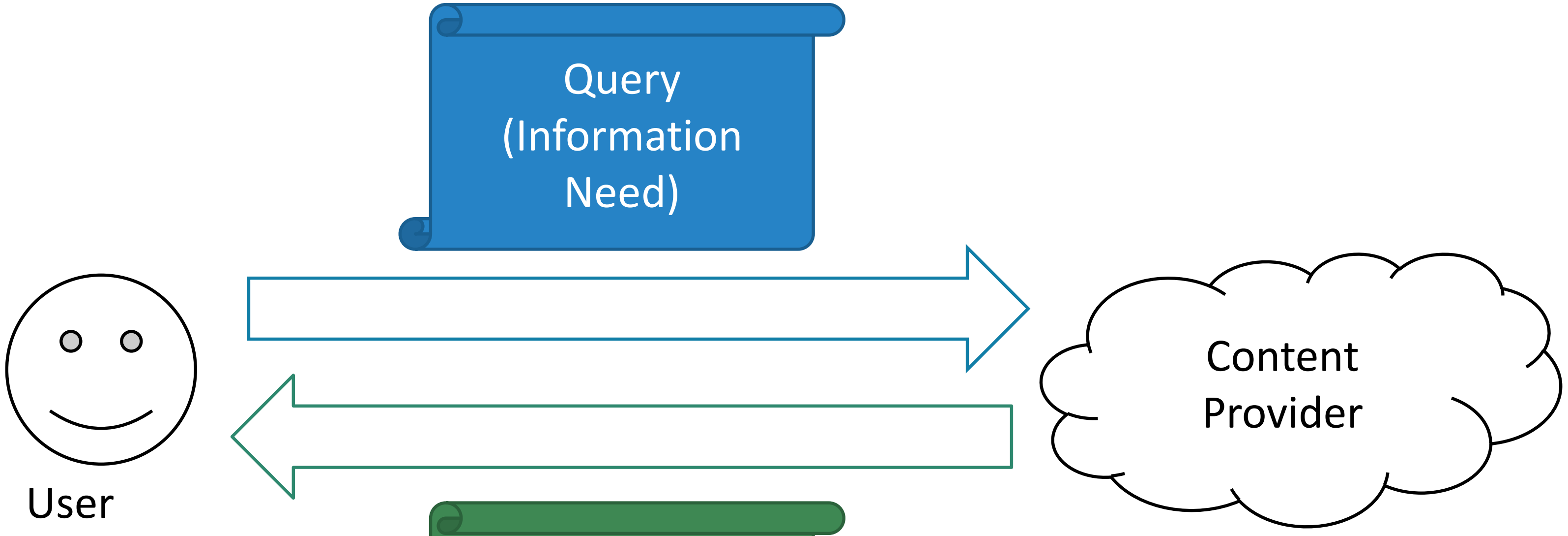
A?

SUOMI 24

S GROUP

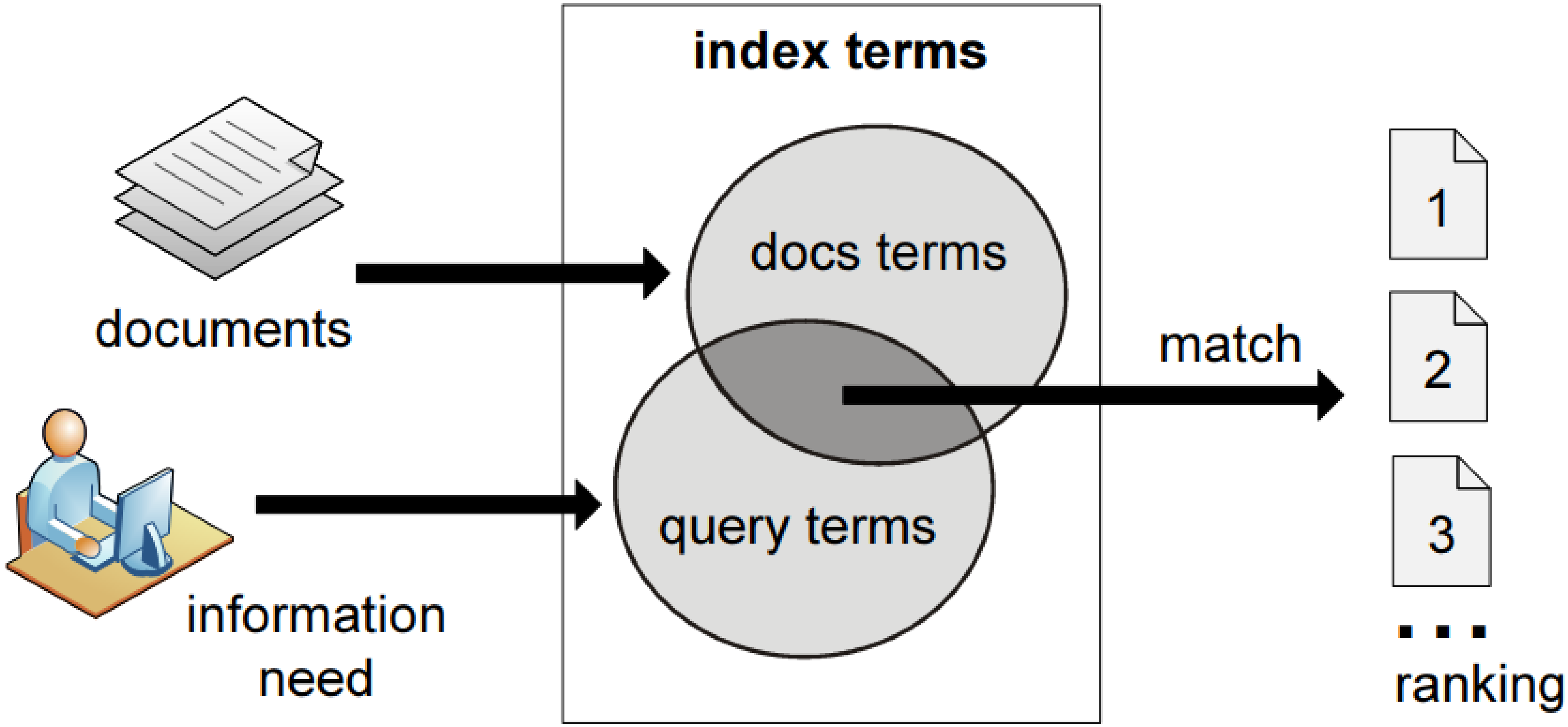
VAUVA

Web users at a glance (IR)



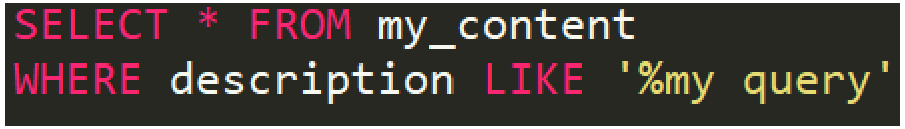
Express the information need as a ***query*** and expects an answer with relevant documents ***fast***

Illustration of an IR model

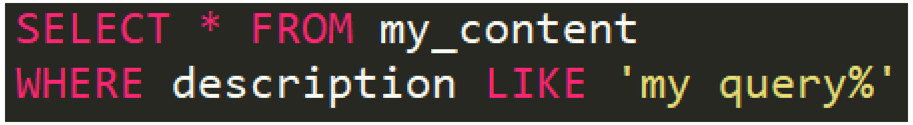


Expected response time in milliseconds

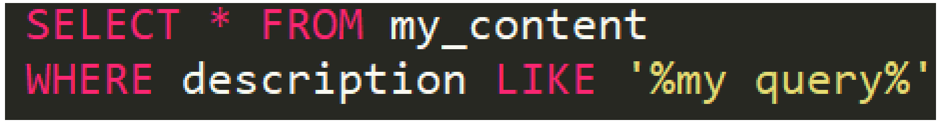
RDBMS approach (B-tree)



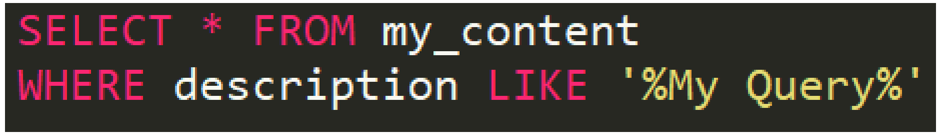
```
SELECT * FROM my_content
WHERE description LIKE '%my query'
```



```
SELECT * FROM my_content
WHERE description LIKE 'my query%'
```

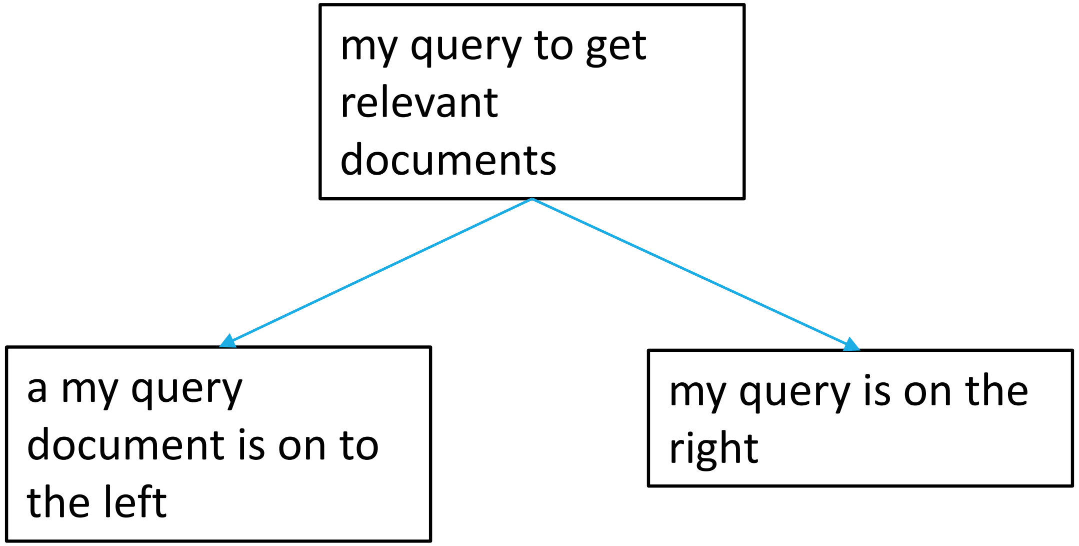


```
SELECT * FROM my_content
WHERE description LIKE '%my query%'
```



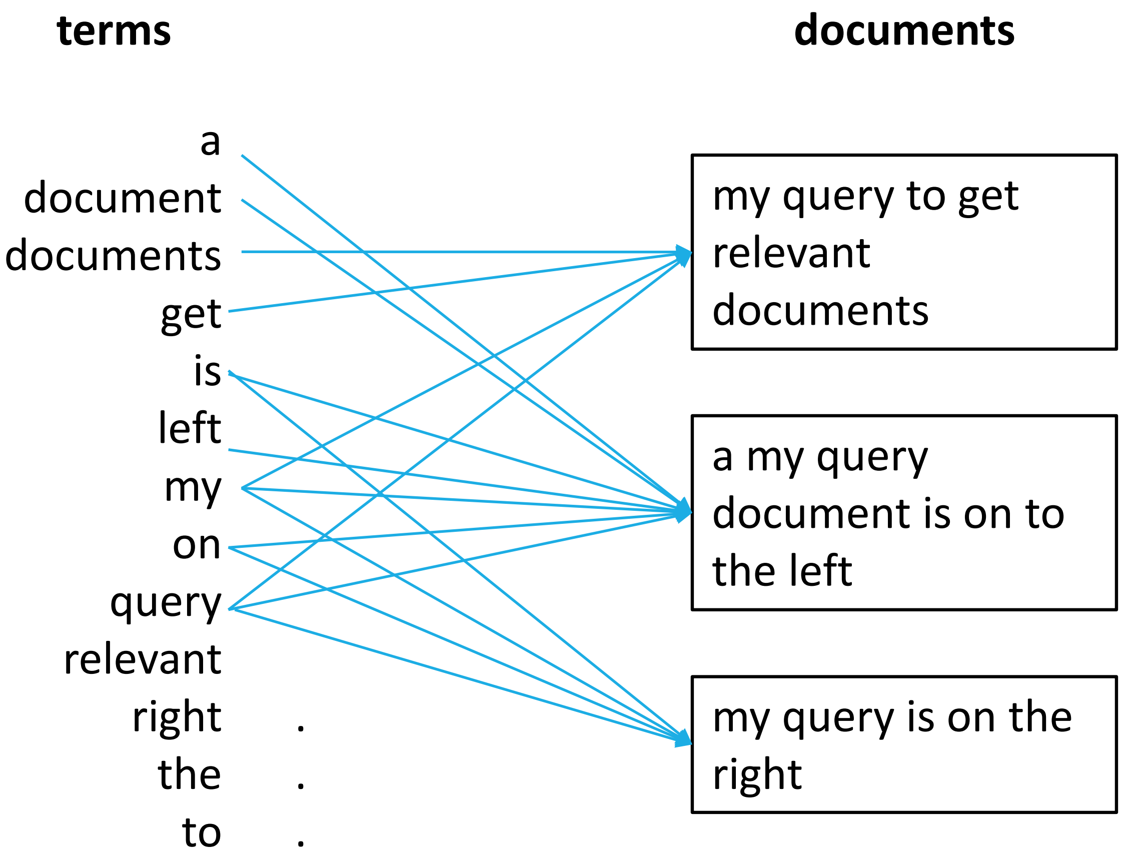
```
SELECT * FROM my_content
WHERE description LIKE '%My Query%'
```

A B-tree on my_content.description



my query to get relevant documents
a my query document is on to the left
my query is on the right

Inverted Index (Lucene approach)



terms

a
document
documents
get
is
left
my
on
query
relevant
right .
the .
to .

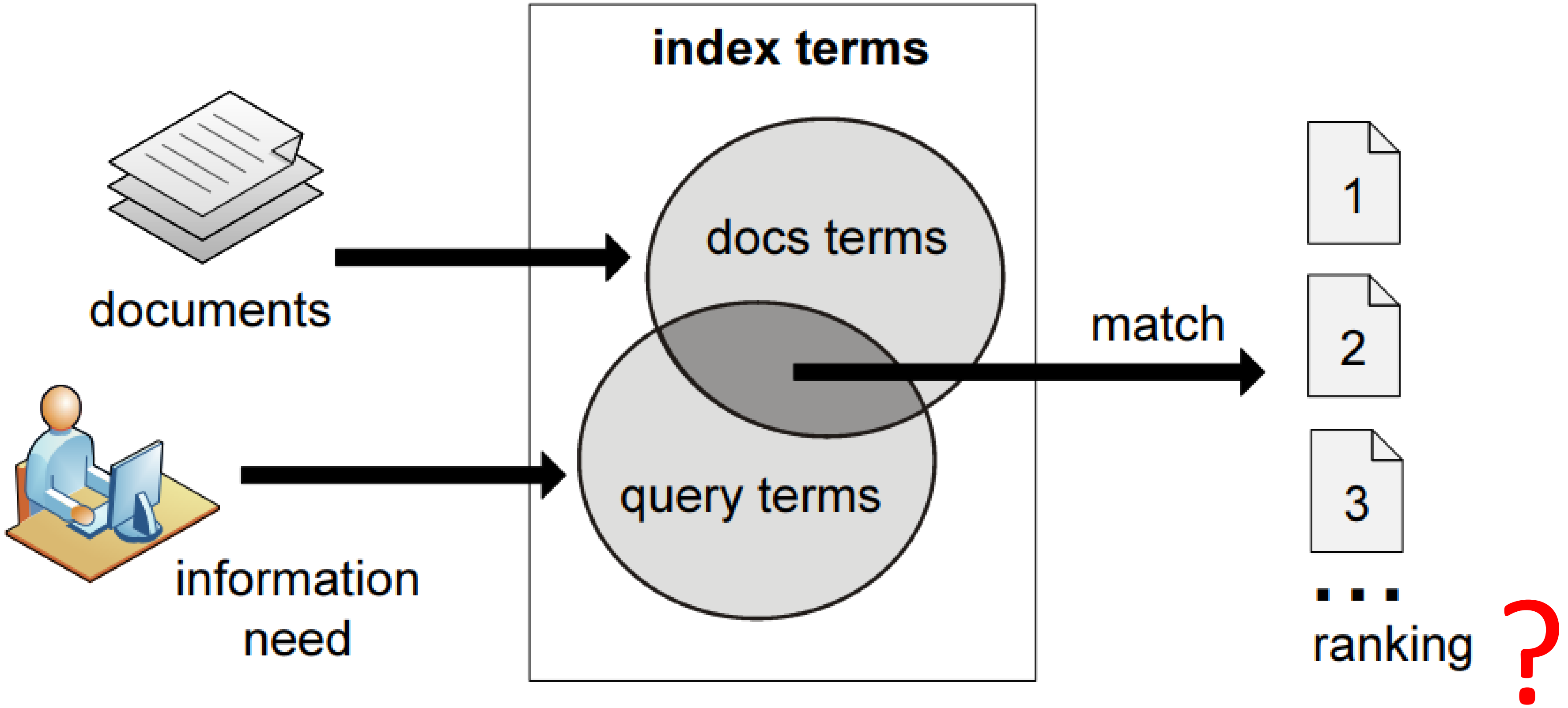
documents

my query to get relevant documents

a my query document is on to the left

my query is on the right

That solves just the matching part...



Expected response time in milliseconds

RDBMS leaves many gaps on the IR side…

How to split text into terms?

Aren’t *documents* and *document* the same?

document
documents

Some words are not that important (a, the, an)

What if the text is in multiple languages?

In what position is the term in each document?

What about the relevance? (tf-idf, BM25)

RDBMS vs IR Software

RDBMS

- Focused on data relations
- Transactions
- ACID

IR Software

- Oriented to efficiently reply search queries
- Support for tokenization
- Support for stop words
- Support for stemming
- Support for advanced querying
- Support for relevance
- Other goodies…

elasticsearch

You know, for search…

elastic.co

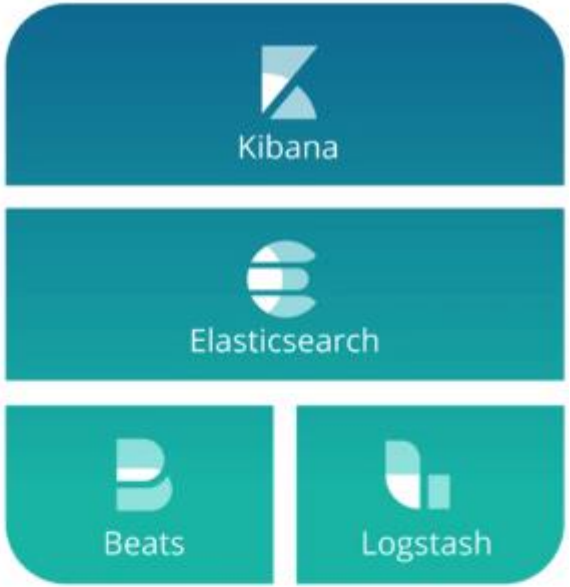
100,000+ Community Members

130M+ Product Downloads

3.7K customers around the world

Elastic Stack

100% Open Source



Explore
Visualize
Analyze

You know,
for search…

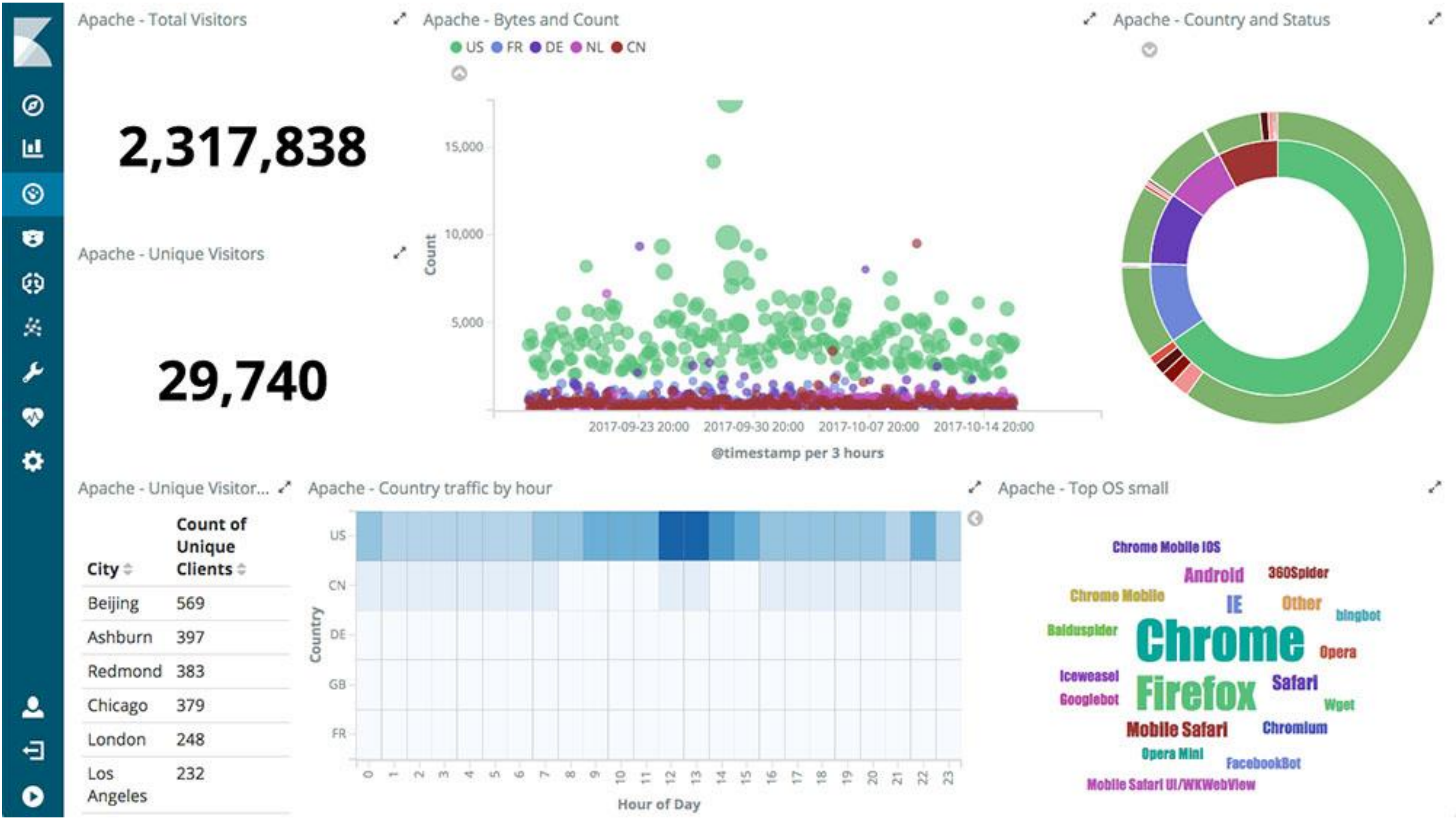
Logs Processing

Enrich data

Elasticsearch is a **distributed, scalable, real-time search and analytics engine.**

You might want to check Kibana

Kibana lets you **visualize your Elasticsearch data** and **navigate** the Elastic Stack with *zero coding.*



Apache - Total Visitors

2,317,838

Apache - Unique Visitors

29,740

Apache - Unique Visitor...

City	Count of Unique Clients
Beijing	569
Ashburn	397
Redmond	383
Chicago	379
London	248
Los Angeles	232

Apache - Bytes and Count

Apache - Country and Status

Apache - Country traffic by hour

Apache - Top OS small

Elasticsearch use cases

Search text

Search text and structure data

Aggregate data (stats, metric, etc)

GeoSearch – Querying over geography

Distributed JSON Document DB (like MongoDB)

Who uses Elasticsearch?

Wikipedia:

- Full-text search
- Search snippets
- *Search-as-you-type*
- *did-you-mean…*

The Guardian:

- Combine web traffic events and social media
- Real-time feedback to the editors about the public’s response to new articles.

Stack Overflow:

- Full-text search
- Geolocation queries
- *more-like-this* to find related questions and answers.

GitHub:

- Searching over 130 billion lines of code.

Elasticsearch - *You know... for search*

Built on top of Apache Lucene

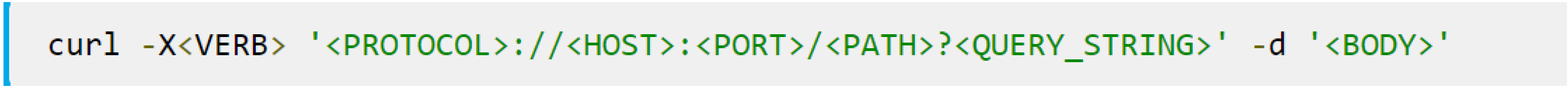
- Advanced IR system
- High-performance
- Fully featured search engine
- Java Library

Elasticsearch:

- A distributed real-time document store
- *Every field* is indexed and searchable
- A distributed search engine with real-time analytics
- Horizontal Scaling

Elasticsearch is accessible via a RESTful API

Elasticsearch RESTful API



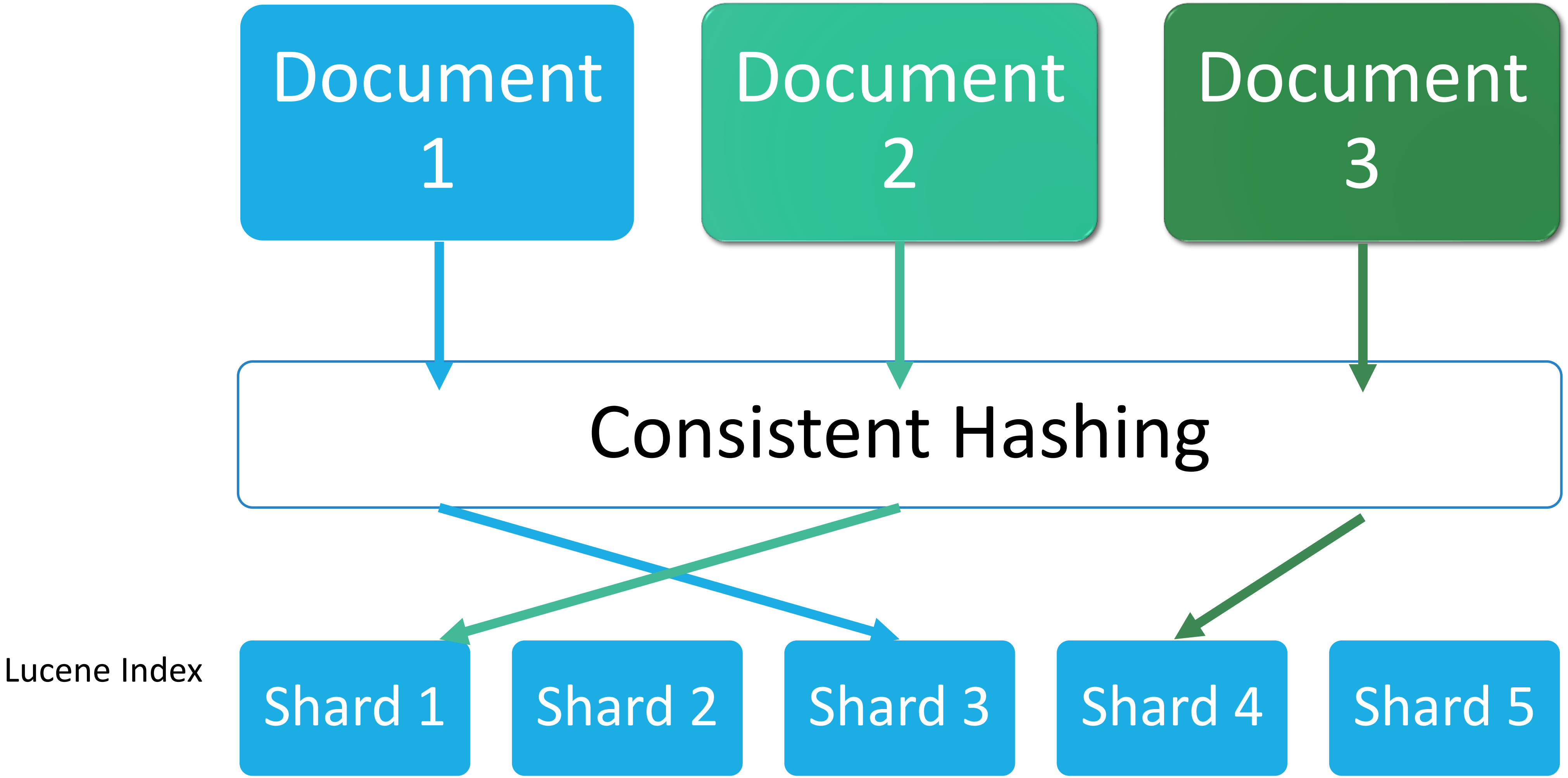
```
curl -X<VERB> '<PROTOCOL>://<HOST>:<PORT>/<PATH>?<QUERY_STRING>' -d '<BODY>'
```

VERB	The appropriate HTTP *method* or *verb*: GET, POST, PUT, HEAD, or DELETE.
PROTOCOL	Either http or https (if you have an https proxy in front of Elasticsearch.)
HOST	The hostname of any node in your Elasticsearch cluster, or localhost for a node on your local machine.
PORT	The port running the Elasticsearch HTTP service, which defaults to 9200.
PATH	API Endpoint (for example _count will return the number of documents in the cluster). Path may contain multiple components, such as _cluster/stats or _nodes/stats/jvm
QUERY_STRING	Any optional query-string parameters (for example ?pretty will *pretty-print* the JSON response to make it easier to read.)
BODY	A JSON-encoded request body (if the request needs one.)

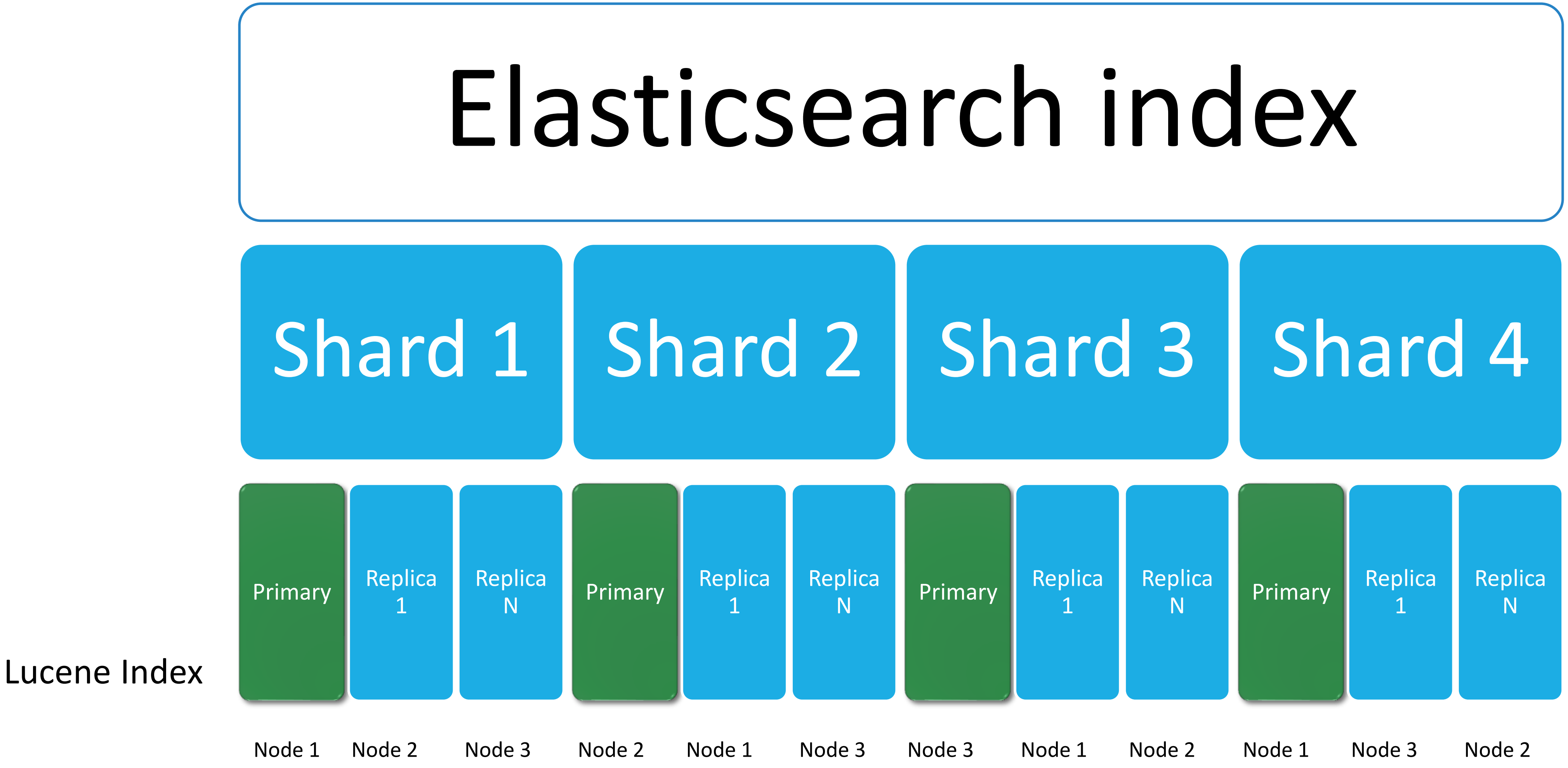
Elasticseach Glossary

Document	Top-level object that is serialized into JSON and stored in Elasticsearch under a unique ID.
Index	Logical *namespace* that points to one or more physical shards.
Shard	Low-level worker unit that holds just a slice of all the data in the index
	A single instance of Lucene
Primary shard	Each document in your index belongs to a single primary shard
Replica shard	A copy of a primary shard.
	Provide redundant copies of data
	Protect against hardware failure, and to serve read requests like searching or retrieving a document.

Document Storage



Elasticsearch Index



Elasticsearch index

Lucene Index

	Primary	Replica 1	Replica N
Shard 1	Node 1	Node 2	Node 3
Shard 2	Node 2	Node 1	Node 3
Shard 3	Node 3	Node 1	Node 2
Shard 4	Node 1	Node 3	Node 2

Writes on Primary Shard and propagates to Replicas

Reads on any Shard

Specifying Shards

The primary shards is defined when the index is created.

The number of replicas can change.

If you want more primary shards, then you will need to re-index the data.

```
PUT /my_index
{       …
        “settings”: {
                "index" : {
                    "number_of_shards" : 5,
                    "number_of_replicas" : 2
                  }
        }
}
```

Mappings

Mappings define how a document, and the fields it contains, are stored and indexed.

Elasticsearch creates a default mapping for the documents.

We can control the mappings by specifying:

Field types:

- Text, short, integer, GeoPoint, date…

```
“properties”:{
   “counter”:{
      “type”:”long”
   }
}
```

Field Index

- Specifies if a field should be analyzed

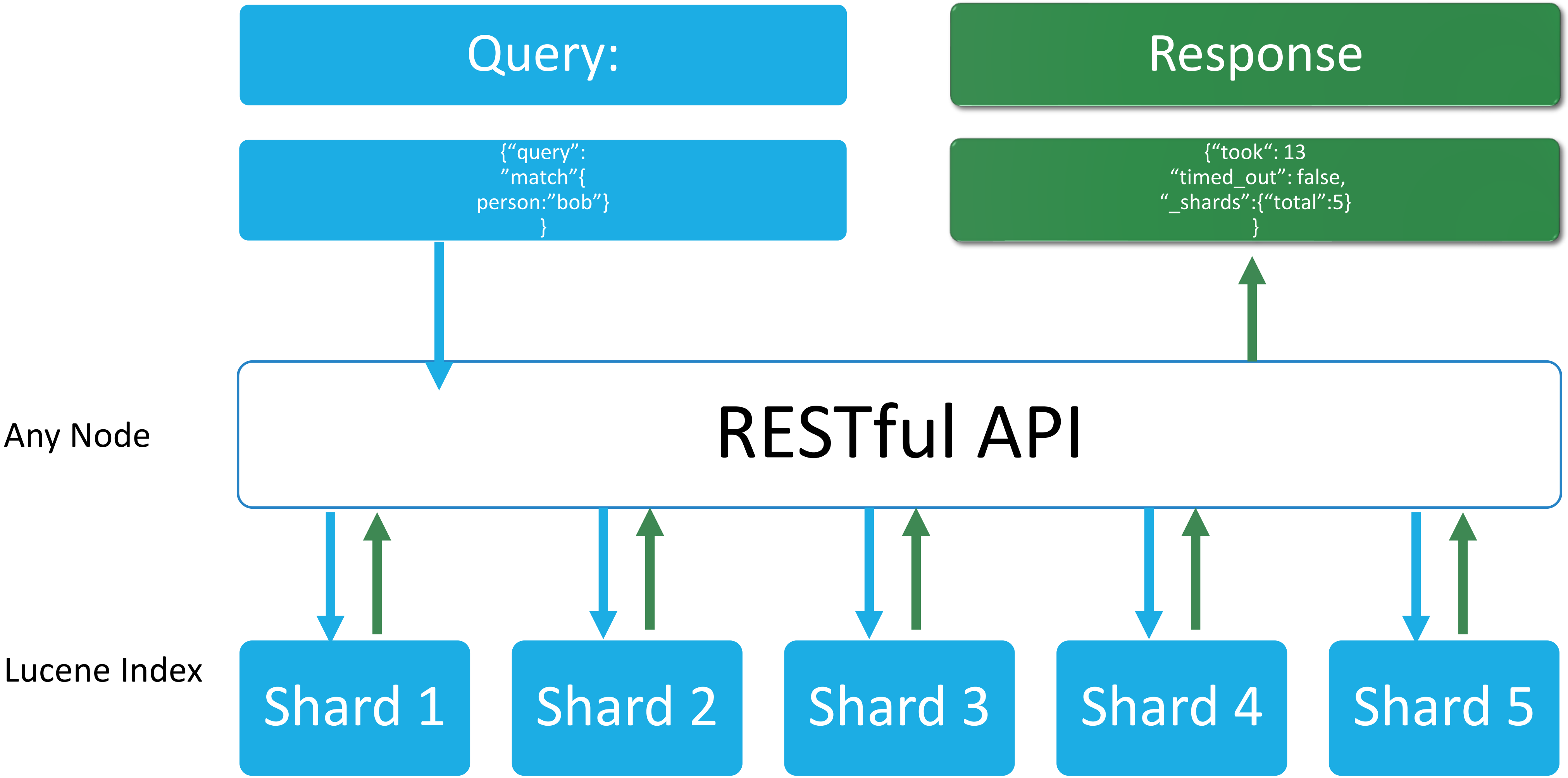
```
“properties”:{
   “country”:{
      “index”:”not_analyzed”
   }
}
```

Field Analyzer

- Defines the analyzer (stop-words, stemming, tokenization)

```
“properties”:{
   “description”:{
      “analyzer”:”english”
   }
}
```

Document Storage



```
{“query”:
”match”{
person:”bob”}
}
```

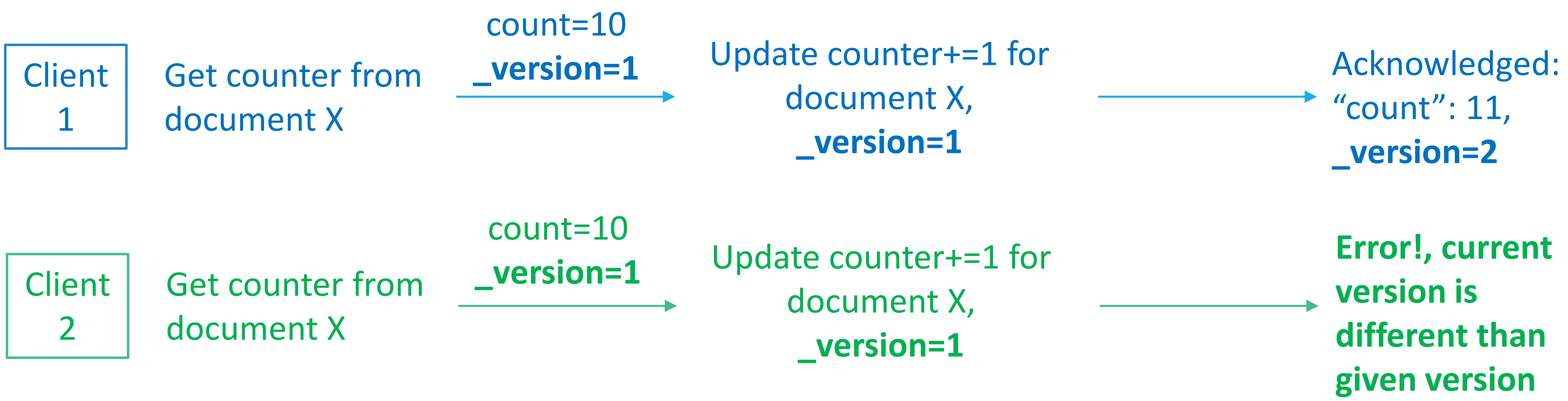
```
{“took“: 13
“timed_out”: false,
“_shards”:{“total”:5}
}
```

Concurrency and Version Control

When a document is updated, a new version is assigned to it.

The version appears in the "**_version**" field of the document.

Update actions can be associated to a specific "_version", this helps handling concurrency.



Client 1

Get counter from document X

count=10
_version=1

Update counter+=1 for document X,
_version=1

Acknowledged:
"count": 11,
_version=2

Client 2

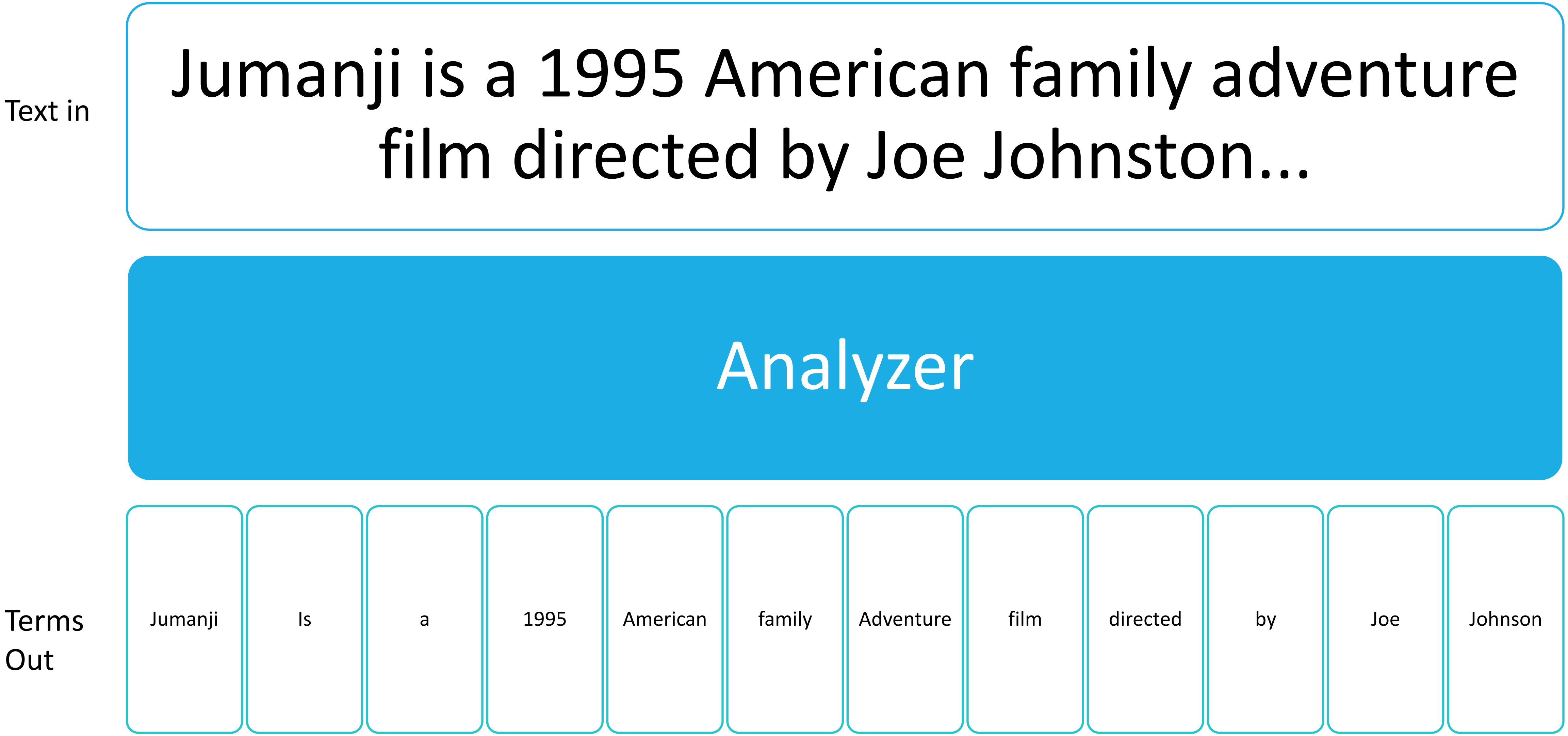
Get counter from document X

count=10
_version=1

Update counter+=1 for document X,
_version=1

Error!, current version is different than given version

Analyzers



Text in

Jumanji is a 1995 American family adventure film directed by Joe Johnston...

Analyzer

Terms Out

Jumanji | Is | a | 1995 | American | family | Adventure | film | directed | by | Joe | Johnson

Scoring and Relevance

Given a query… return most relevant documents

Find all documents that match a query

Score docs using a relevance function (e.g., TF/IDF, BM25)

Query Types:

- Structured Search
- Full-Text Search
- Proximity Matching
- GeoQueries
- *More Like This Queries*
- Autocomplete Queries

As a summary

Good things about Elasticsearch

- Open Source, Apache License 2.0
- Cross-platform
- Scaling is easy, just add more nodes
- High availability and Fault Tolerance
- Fast full text search
- Advanced text analyzers
- Relevance scoring
- Fast aggregations
- Advanced queries:
  - Geographical Queries
  - Did you mean?
  - More Like This
- Elasticstash

Elasticsearch is not for:

- Transactions / ACID
- Primary database

elasticsearch

Short break and then:
Hands-on: A content-based recommender system for movies!

That’s all for now!

Aalto University
School of Science

Thanks!

Questions?

Frederick Ayala-Gómez
frederick.ayala@aalto.fi

Credits and suggested material

An Elasticsearch Crash Course
https://www.elastic.co/videos/an-elasticsearch-crash-course

Elasticsearch: Getting Started
https://www.elastic.co/webinars/getting-started-elasticsearch

Elasticsearch: The Definitive Guide
https://www.elastic.co/guide/en/elasticsearch/guide/2.x/index.html

Elasticsearch 6 and Elastic Stack - In Depth and Hands On!
https://www.udemy.com/elasticsearch-6-and-elastic-stack-in-depth-and-hands-on/?altsc=1251578