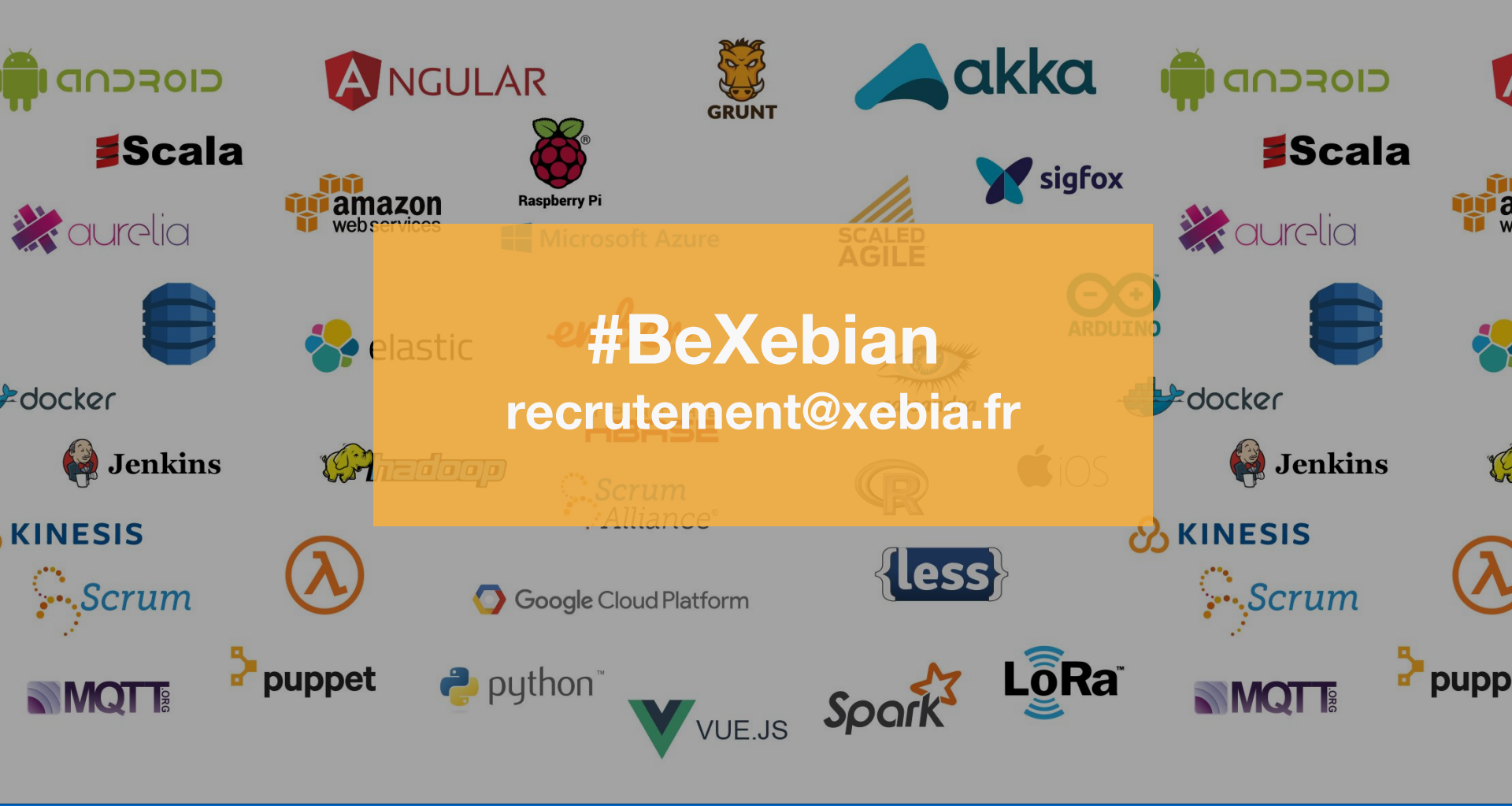

Elasticsearch from zero to hero
Gérôme Egron @GeromeEgron
Ivan Beauvais @ibeauvais

Présentation des speakers


#BeXebian
recrutement@xebia.fr


BE A BETTER
DEVELOPER
Diffusez vos cartes dans la communauté
et devenez un ninja du développement.

Glossaire

- Document
- Index
- Indexation
- Mapping

Différents types de recherche

1. Recherche Full Text
2. Recherche Exacte
3. Aggregation

1 Recherche Full text

Indexation d'un document

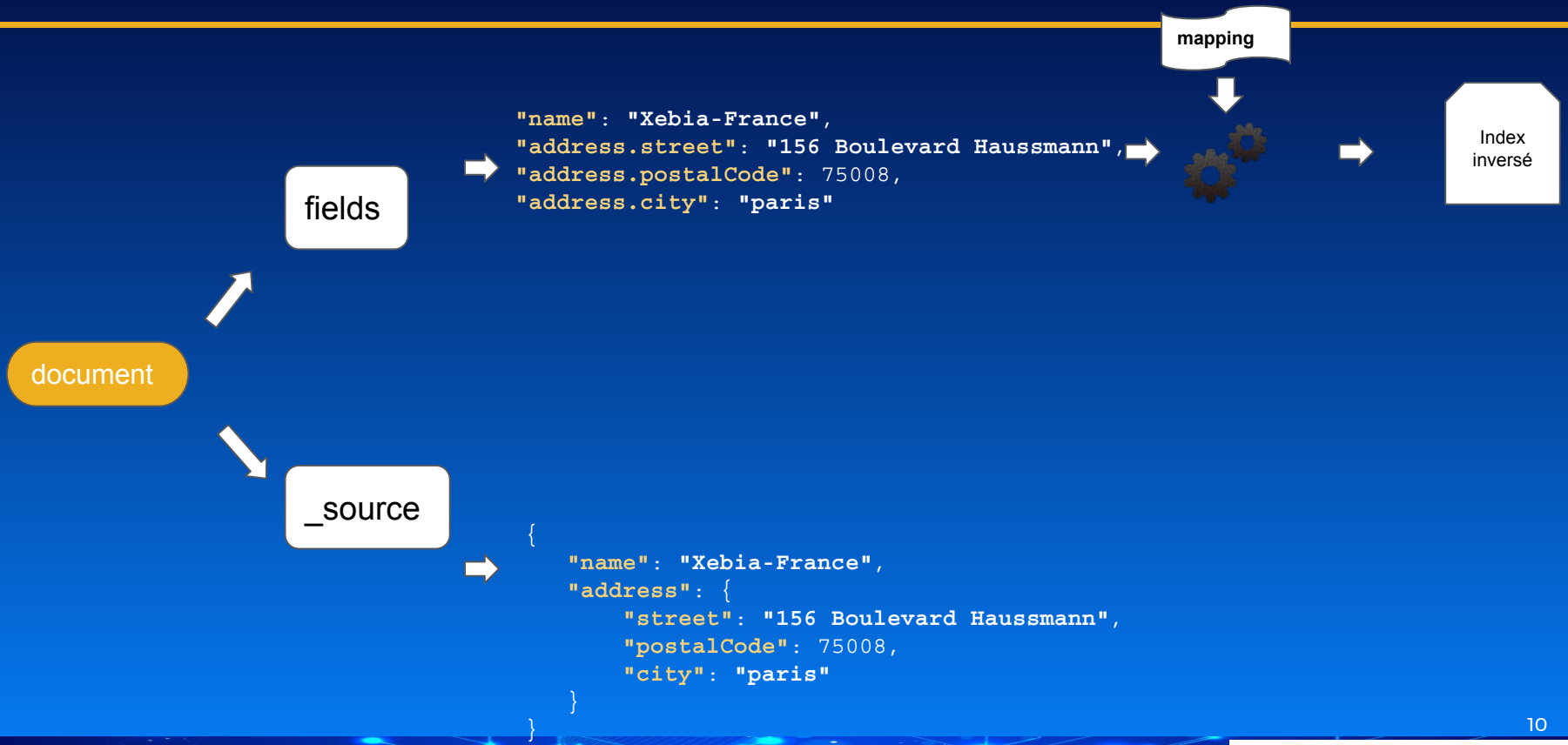
```
POST http://localhost:9200/directory/address
{
     "name": "Xebia-France",
     "address": {
          "street": "156 Boulevard Haussmann",
          "postalCode": 75008,
          "city": "paris"
     }
}
```

Mapping correspondant

```
"name": {
      "type": "text",
      "fields": {
      "keyword": {
            "type": "keyword",
            "ignore_above": 256
      }
},
"address": {
      "properties": {
            "postalCode": {
                  "type": "long"
            },
            "street": {
            "type": "text",

            [...]
```

Traitement d'un document



Index inversé

"name": "Xebia-France"

"name": "Voyages-sncf"

"name": "sncf"

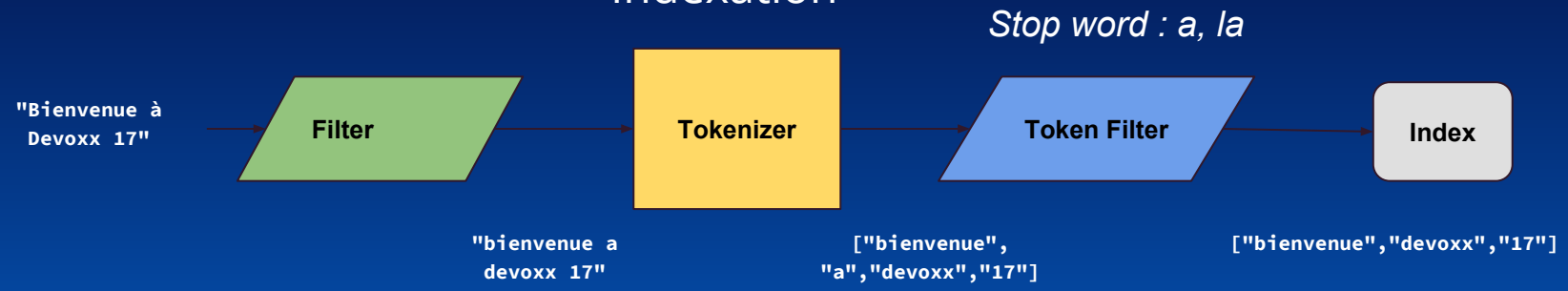
Term	Frequency	ID Document
xebia	1	1
france	1	1
voyages	1	2
sncf	2	2,3

Analyse

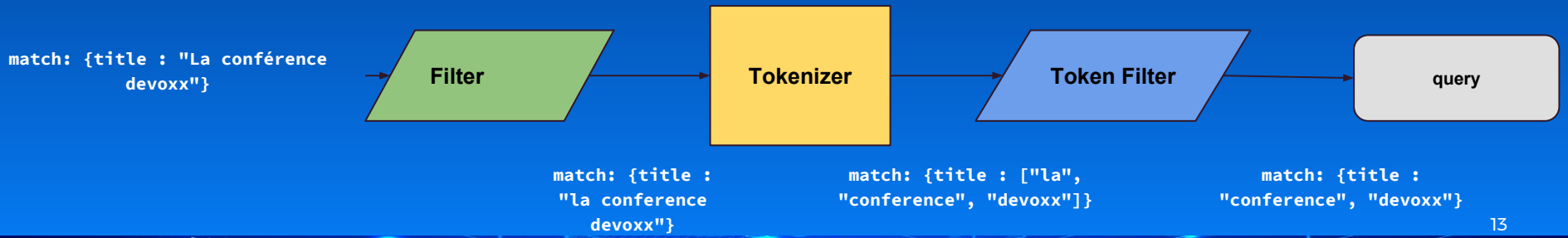
- Dépend du mapping du champ
- Effectuée en 3 phases : Filter, Tokenizer, Token Filter
- Appliquée sur les valeurs des champs avant l'enregistrement dans l'index inversé
- N'impacte pas le document original sauvegardé (_source)

Analyse : exemple

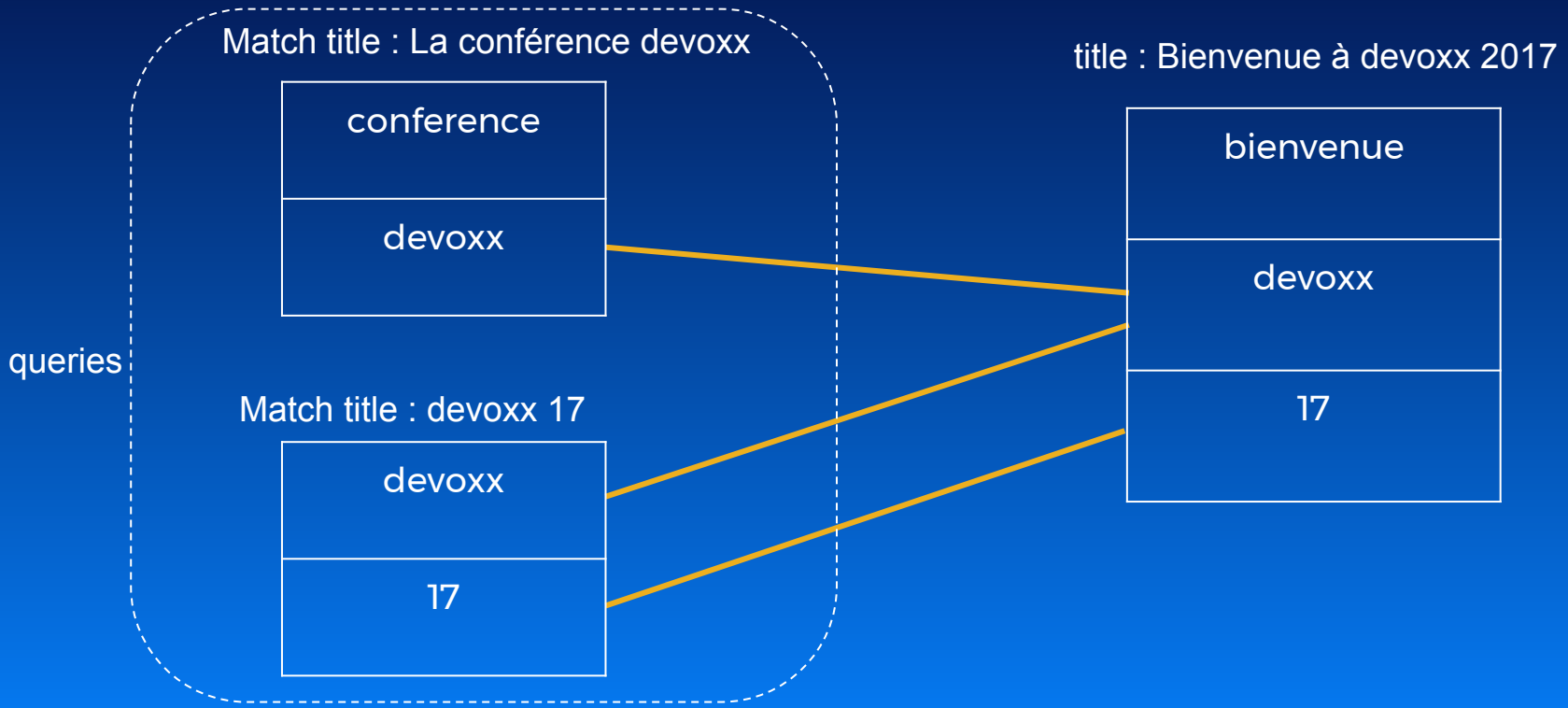
Indexation



Recherche



Le matching


Match title : La conférence devoxx
conference
devoxx
queries
Match title : devoxx 17
devoxx
17
title : Bienvenue à devoxx 2017
bienvenue
devoxx
17

Scoring des documents (BM25)

- **Term frequency** : plus un terme est présent dans un champ plus le score est élevé
- **Inverse document frequency** : plus un terme est présent dans tous les documents de l'index moins le score est élevé
- **Field-length norm** : Plus le champ est court plus le score est élevé

Match query

```
GET my_index/company/_search
{
    "query": {
        "match": {
            "name": "La conférence devoxx"
        }
    }
}
```

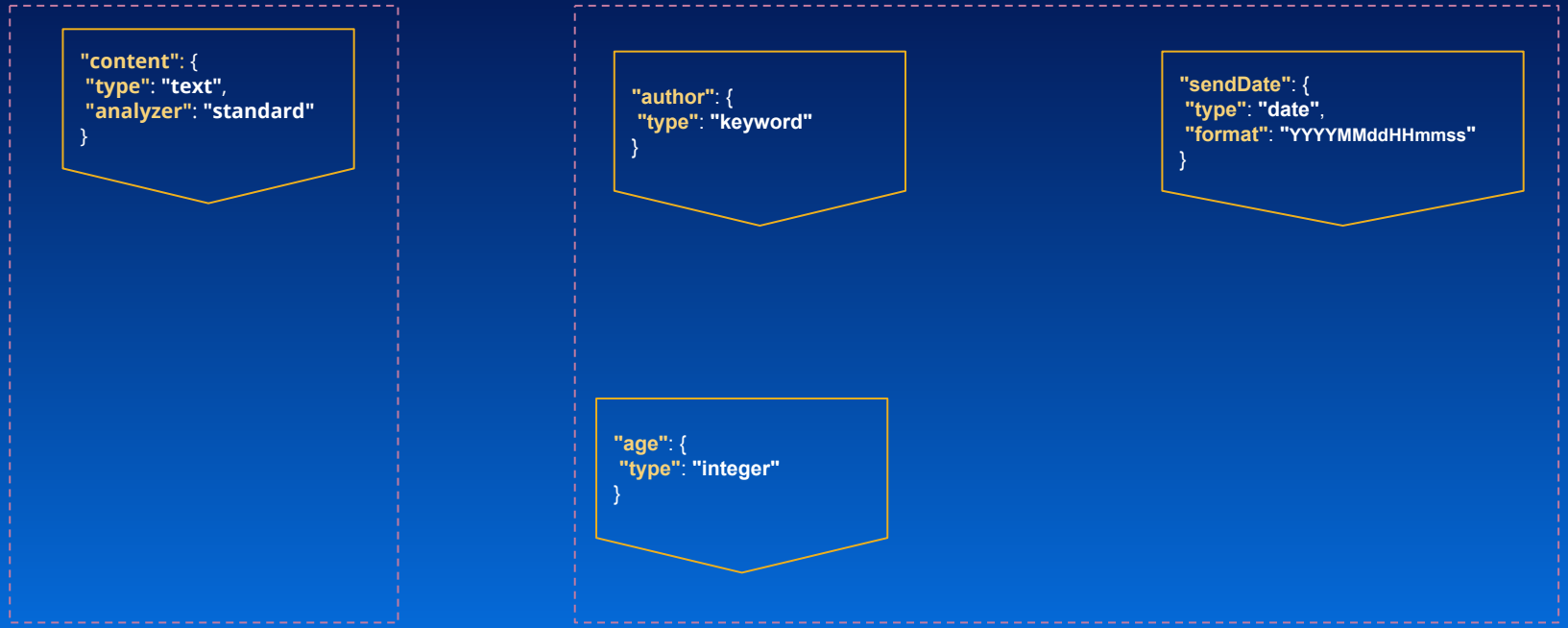
```
Result :
{
    "_index": "my_index",
    "_type": "company",
    "_id": "1",
    "_score": 0.2169777,
    "_source": {
        "name": "Bienvenue à Devoxx 17"
    }
}
```

2 Recherche Exacte

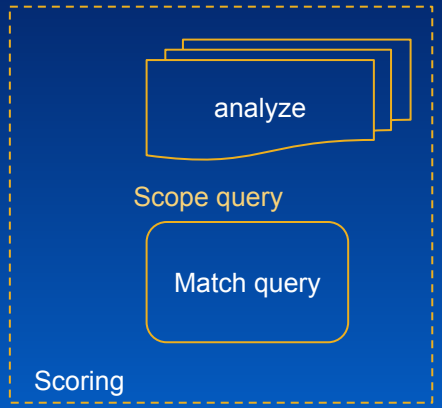
Recherche exacte

- Similaire à une clause "where" en SQL
- Pas de score sur les documents : le document respecte la condition ou ne la respecte pas
- Coupler à la recherche full texte permet de réduire le nombre de document à scorer
- Mis en cache

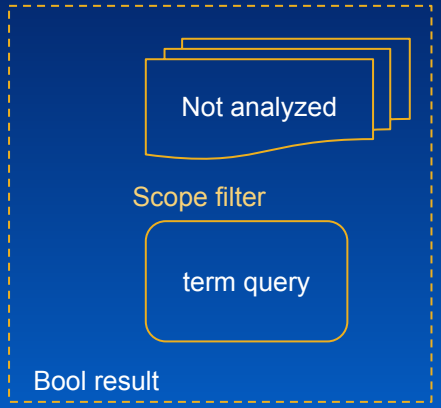
Mapping


"content": {
"type": "text",
"analyzer": "standard"
}
"author": {
"type": "keyword"
}
"sendDate": {
"type": "date",
"format": "YYYYMMddHHmmss"
}
"age": {
"type": "integer"
}
Recherche Full Text
Recherche exacte

Recherche full text vs Recherche exacte



≠



Recherche Full Text

Recherche exacte

Bool query : must

```
GET _search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "content": {
              "value": "Devoxx 2017"
            }
          }
        },
        {
          "match": {
            "content": "Elasticsearch"
          }
        }
      ]
    }
  }
}
```

Bool query : should

```
GET _search
{
  "query": {
    "bool": {
      "should": [
        {
          "match": {
            "content": {
              "value": "Devoxx 17"
            }
          }
        },
        {
          "match": {
            "content": "Breizhcamp"
          }
        }
      ]
    }
  }
}
```

Bool query : must_not

```
GET _search
{
  "query": {
    "bool": {
      "must_not": [
        {
          "match": {
            "content": {
              "value": "2015"
            }
          }
        }
      ]
    }
  }
}
```

Bool query : filter

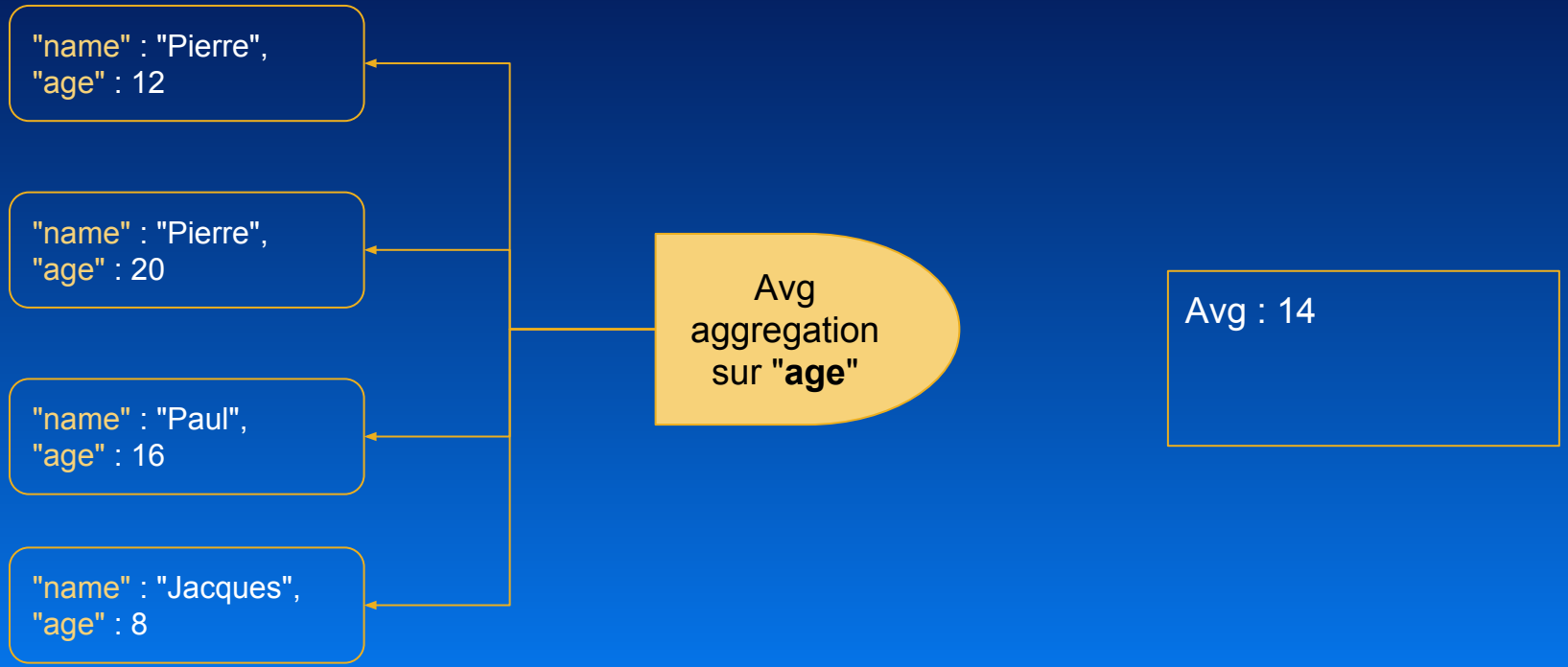
```
GET _search
{
  "query": {
    "bool": {
      "filter": [
        {
          "term": {
            "age": {
              "value": 12
            }
          }
        }
      ]
    }
  }
}
```

3 Aggregation

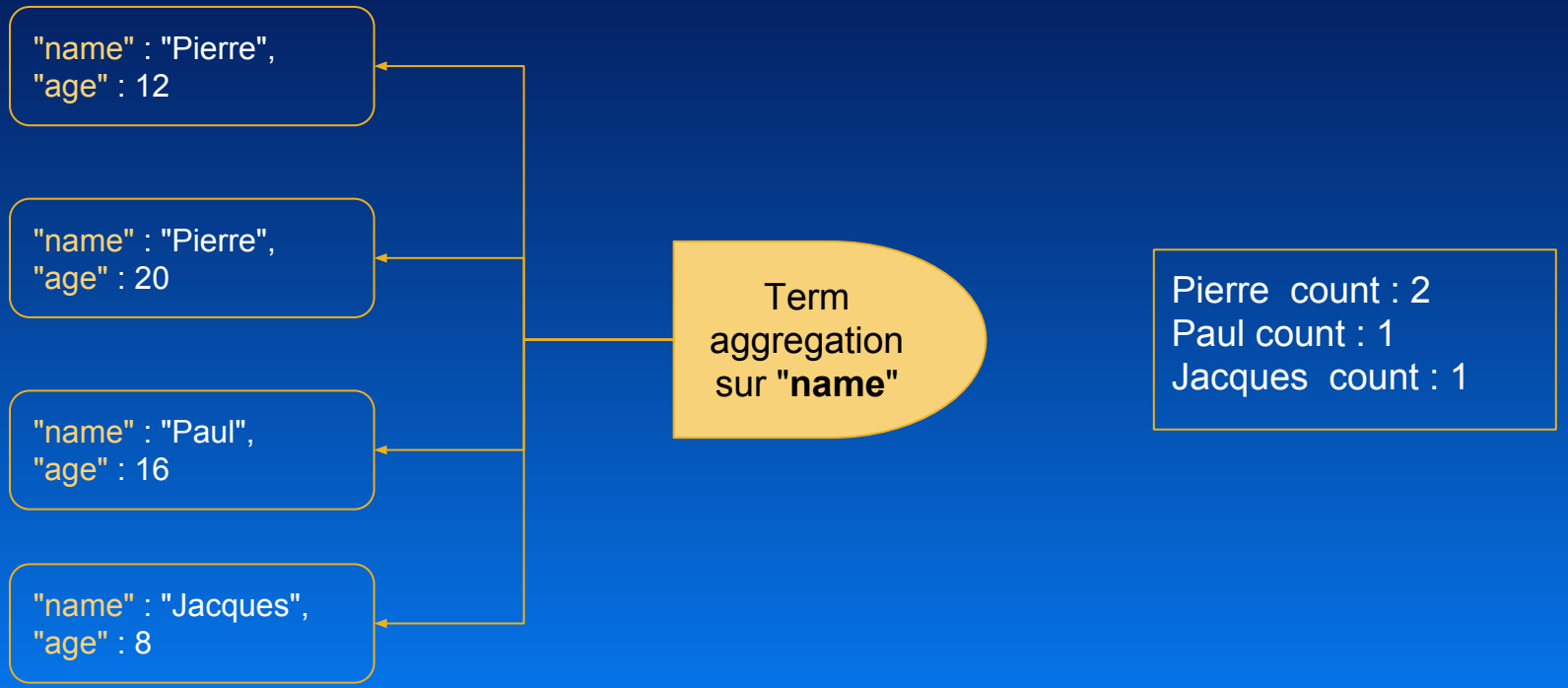
Aggregation

- Equivalent à "count", "groupBy", "Max" ... en SQL
- On ne remonte plus des documents mais on collecte des informations contenues dans les champs des documents
- Utilisés conjointement aux filtres

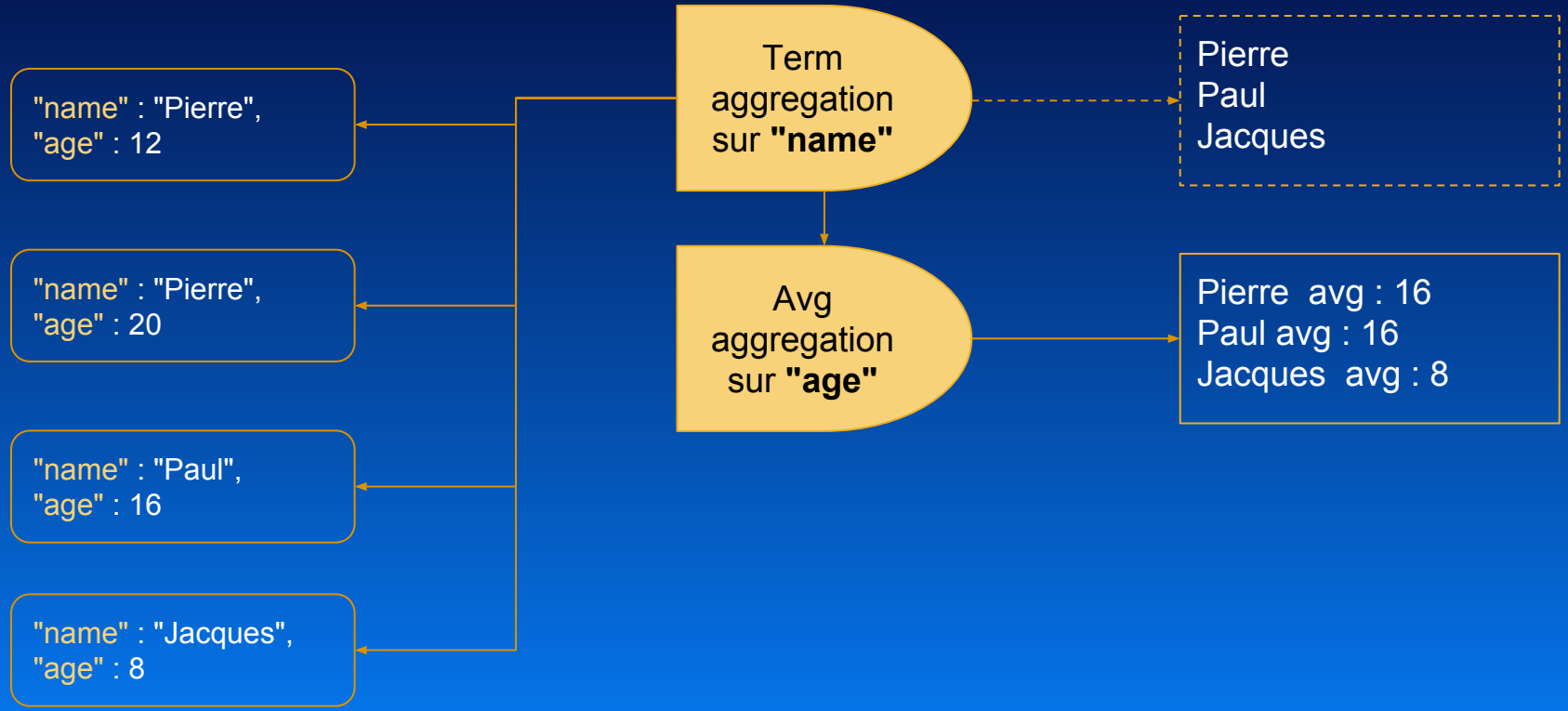
Metrics aggregation



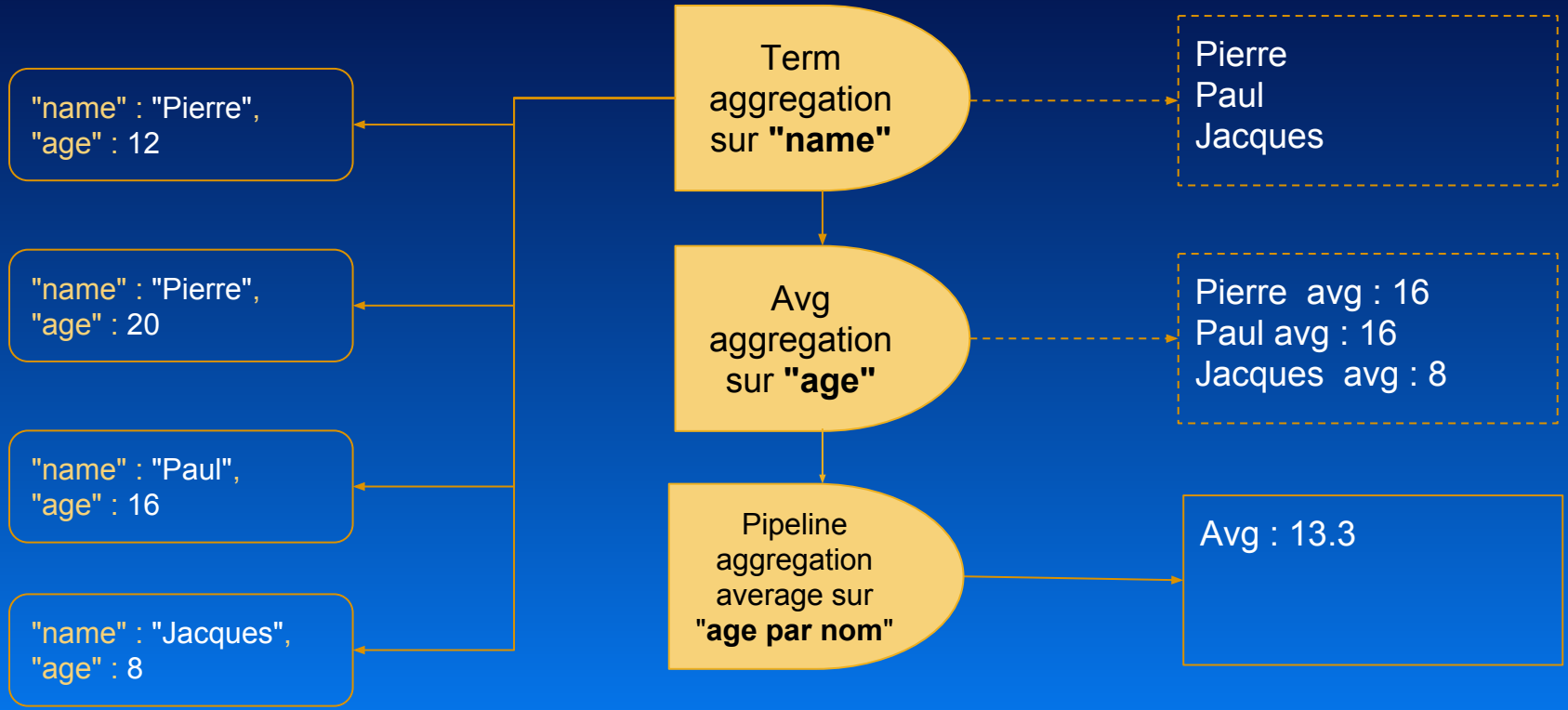
Bucket aggregation



Sub aggregation



Pipeline aggregation



Aggregation en pratique

- Moyenne, minimum, maximum des prix sur un scope de document
- Somme des valeurs d'un champ numérique de tous les documents
- Percentiles des temps de réponse stockés dans l'index
- Toutes les valeurs possibles pour un champ
- Nombre de document compris entre des plages de dates ou des coordonnées GPS

Hand’s on !
https://xebia-france.github.io/es-from-zero-to-hero/

Vous avez vu

- Ajouter/Supprimer/Rechercher des documents
- Utiliser le mapping afin de supprimer le code html du texte
- Utiliser le mapping pour ajouter des synonymes
- Rechercher avec une "Match" query
- Filtrer sur un intervalle de date
- Faire une recherche sur plusieurs champs
- Faire de la suggestion en mode "fuzzy"
- Agréger par term et par term de term
- Faire des requêtes géolocalisées
- Faire des agrégations sur des distances à un point
- Utiliser l'agrégation de type date_histogram
- Filtrer les résultats d'une agrégation grâce à une pipeline agrégation