

advanced search for your legacy app



David Pilato
Technical advocate

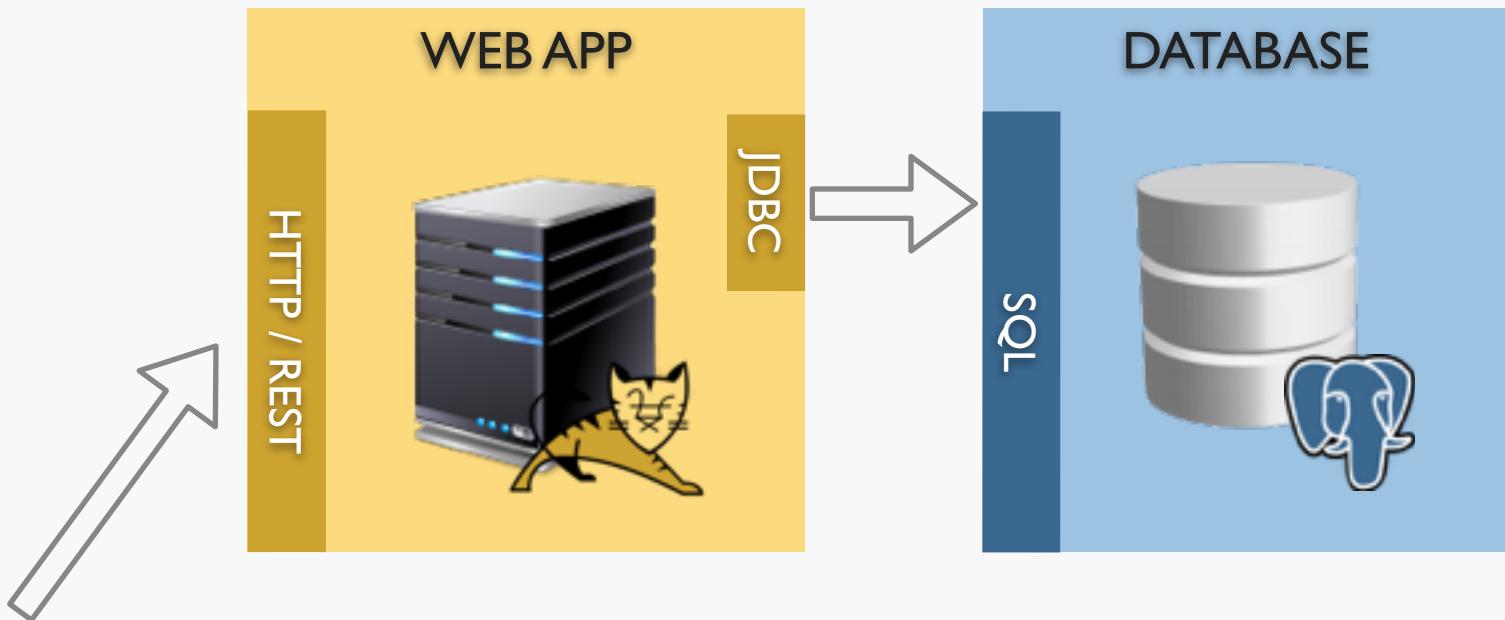
A small green bonsai tree logo.

elasticsearch.
@dadoonet

our use case



our legacy platform



Contacts			
Search Advanced			
Name...		Country...	City...
Found 100 hits in 35 ms			
Id	Name	Date Of Birth	City
1	Joe Smith	1977-06-08	Turin
2	Samia Nour	1993-02-13	Turin
3	Eva Kenza	1964-08-26	Bonn



elasticsearch.

our legacy application



WEB APP

WEBSERVICE LAYER

SERVICE LAYER

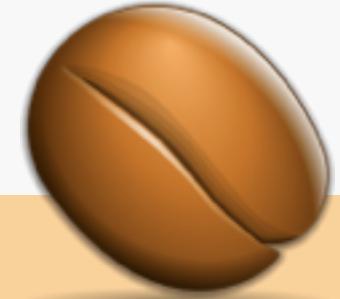
DAO LAYER

DOMAIN LAYER

HTTP / REST

JDBC

our legacy domain



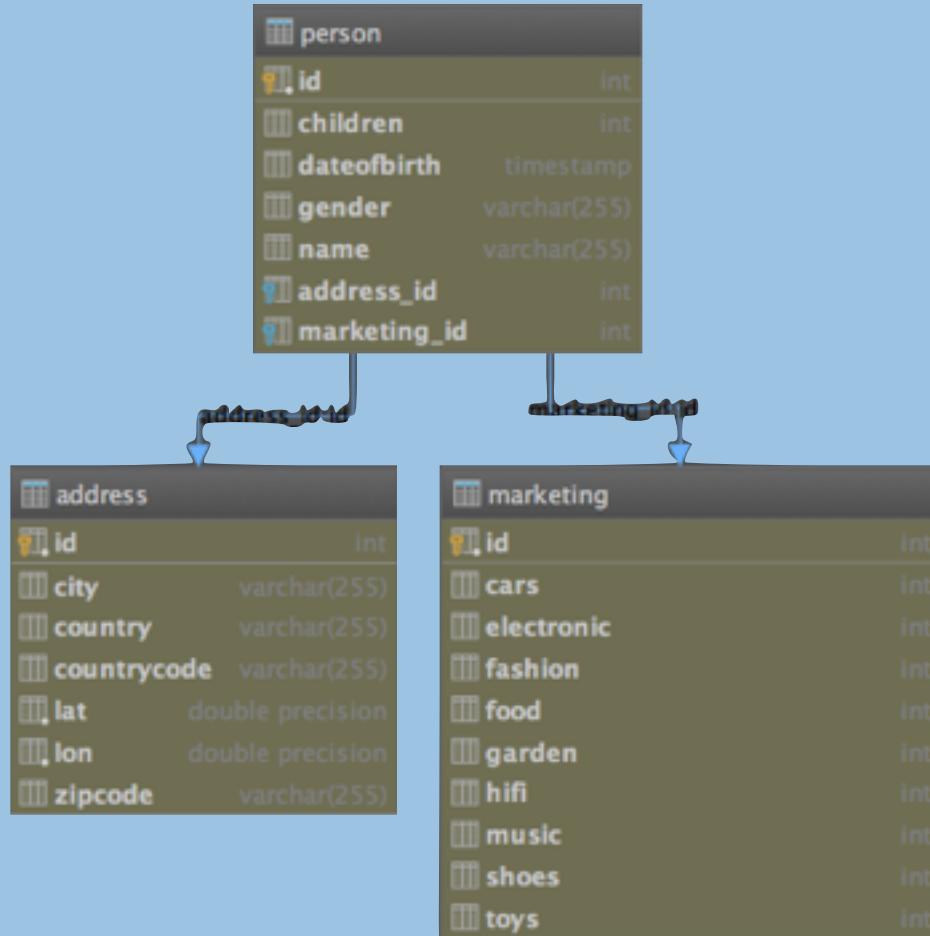
JAVA BEANS



our legacy database



DATABASE



SQL

elasticsearch.

demo time

our legacy app

```
$ git clone https://github.com/dadoonet/legacy-search.git  
$ git checkout 00-legacy  
$ mvn clean install jetty:run
```



connecting with our app architecture



elasticsearch.

JSON document design



person	
id	int
children	int
dateofbirth	timestamp
gender	varchar(255)
name	varchar(255)
address_id	int
marketing_id	int

```
PUT /person/person/1
{
  "name": "Joe Pink",
  "dateOfBirth": "1971-12-26",
  "address_id": "2",
  "marketing_id": "3"
}
```

address	
id	int
city	varchar(255)
country	varchar(255)
countrycode	varchar(255)
lat	double precision
lon	double precision
zipcode	varchar(255)

```
PUT /person/address/2
{
  "city": "Paris",
  "country": "France"
}
```

marketing	
id	int
cars	int
electronic	int
fashion	int
food	int
garden	int
hifi	int
music	int
shoes	int
toys	int

```
PUT /person/marketing/3
{
  "cars": 1000,
  "food": 1500
}
```

JSON document design



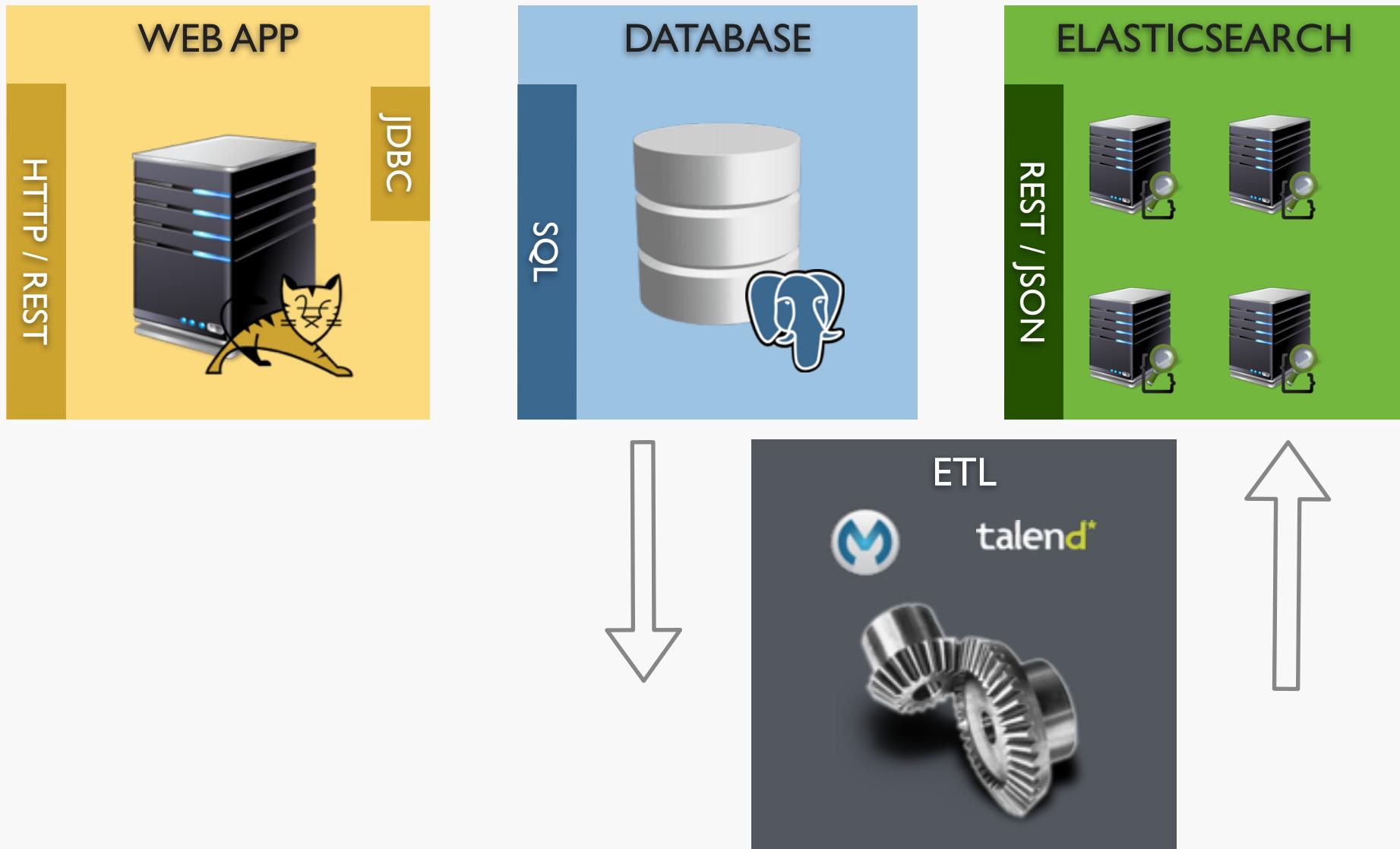
person	
id	int
children	int
dateofbirth	timestamp
gender	varchar(255)
name	varchar(255)
address_id	int
marketing_id	int

address	
id	int
city	varchar(255)
country	varchar(255)
countrycode	varchar(255)
lat	double precision
lon	double precision
zipcode	varchar(255)

marketing	
id	int
cars	int
electronic	int
fashion	int
food	int
garden	int
hifi	int
music	int
shoes	int
toys	int

```
PUT /person/person/1
{
  "name": "Joe Pink",
  "dateOfBirth": "1971-12-26",
  "address": {
    "city": "Paris",
    "country": "France"
  },
  "marketing": {
    "cars": 1000,
    "food": 1500
  }
}
```

using a ETL

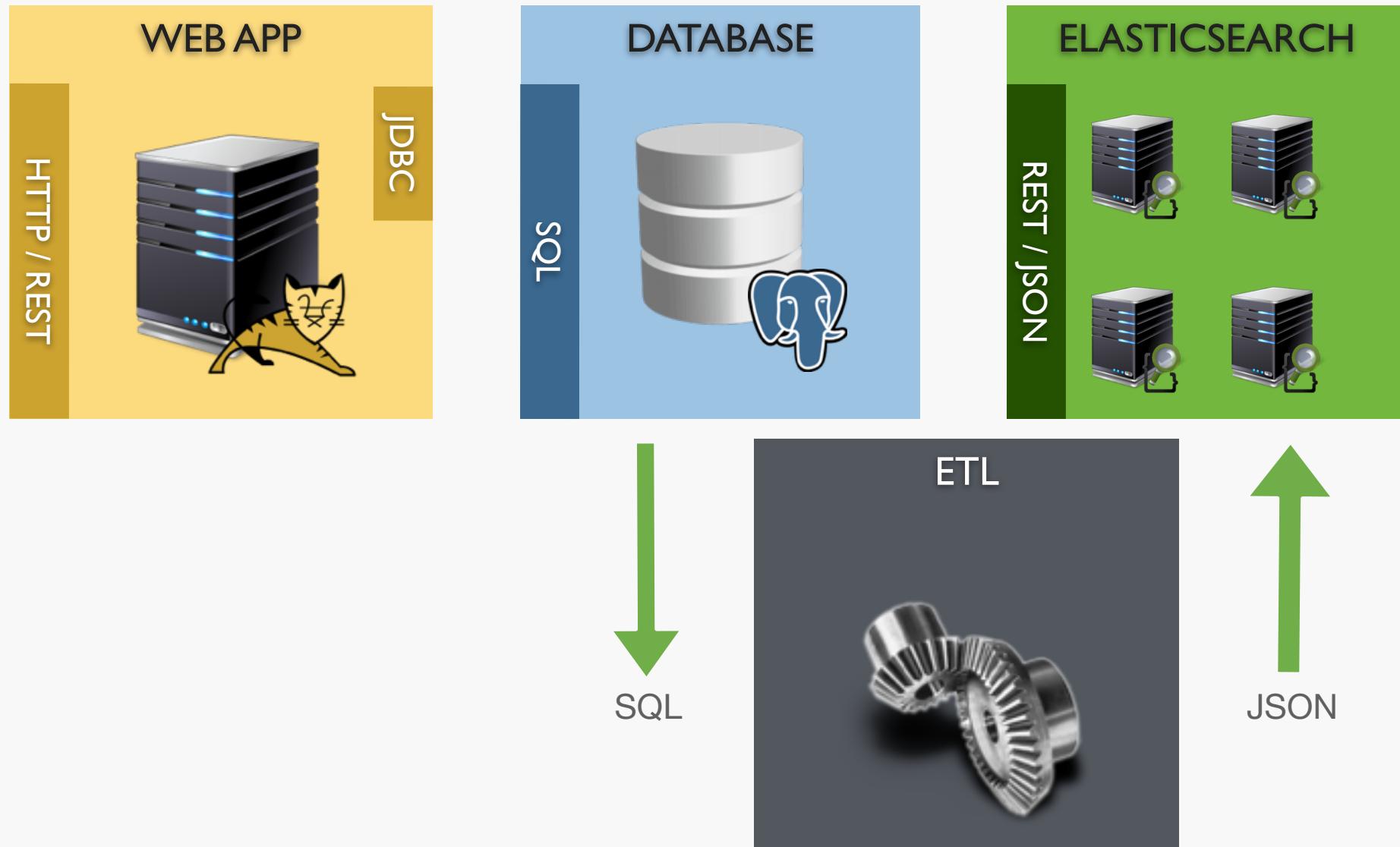


elasticsearch rivers

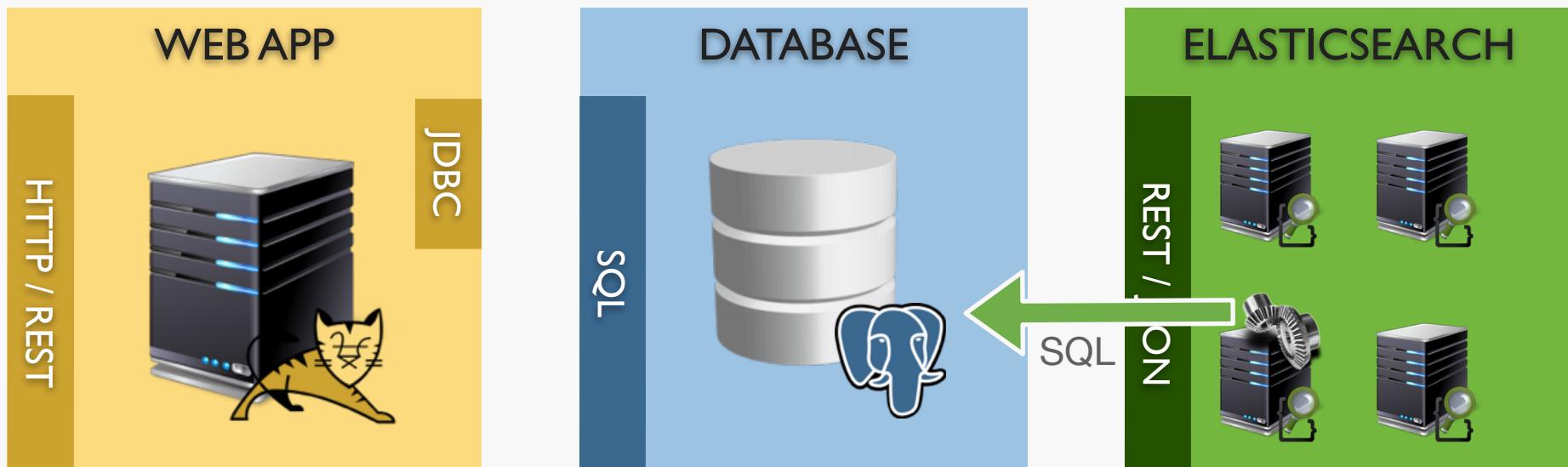
JDBC River



JDBC river



JDBC river

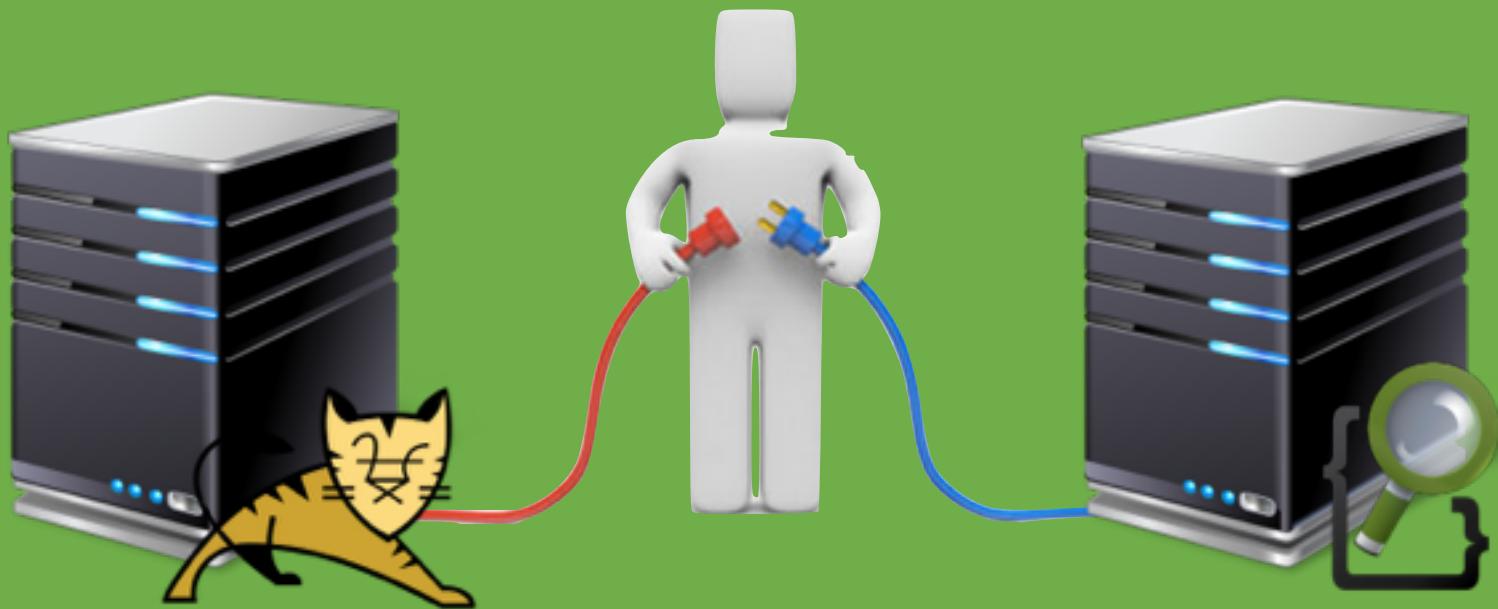




ROAD
ENDS IN
WATER

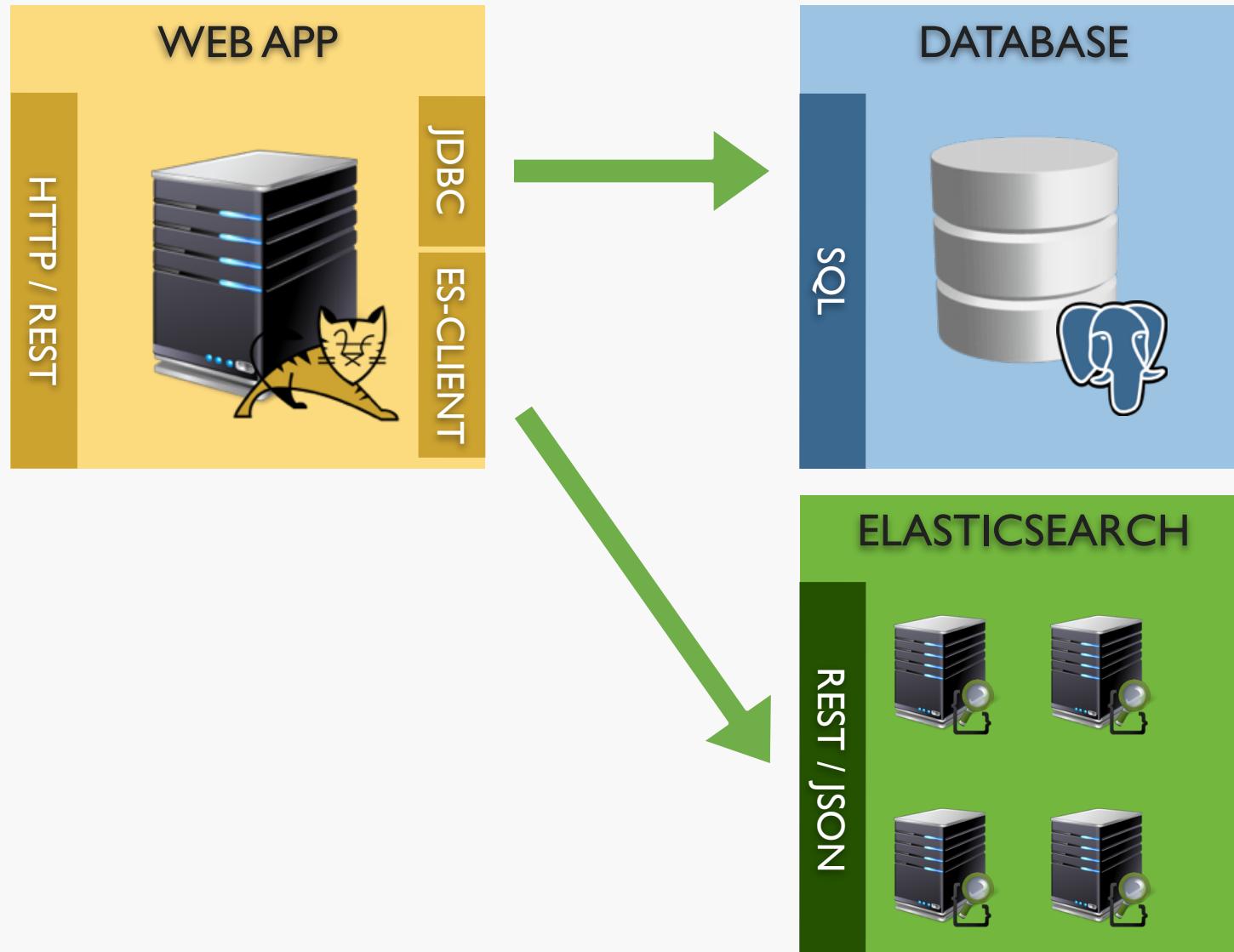


Do It Yourself direct connection



elasticsearch.

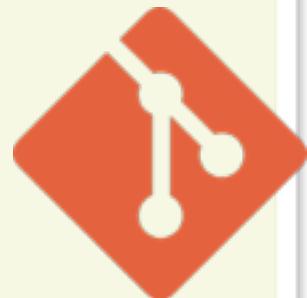
direct connection



demo time

direct connection

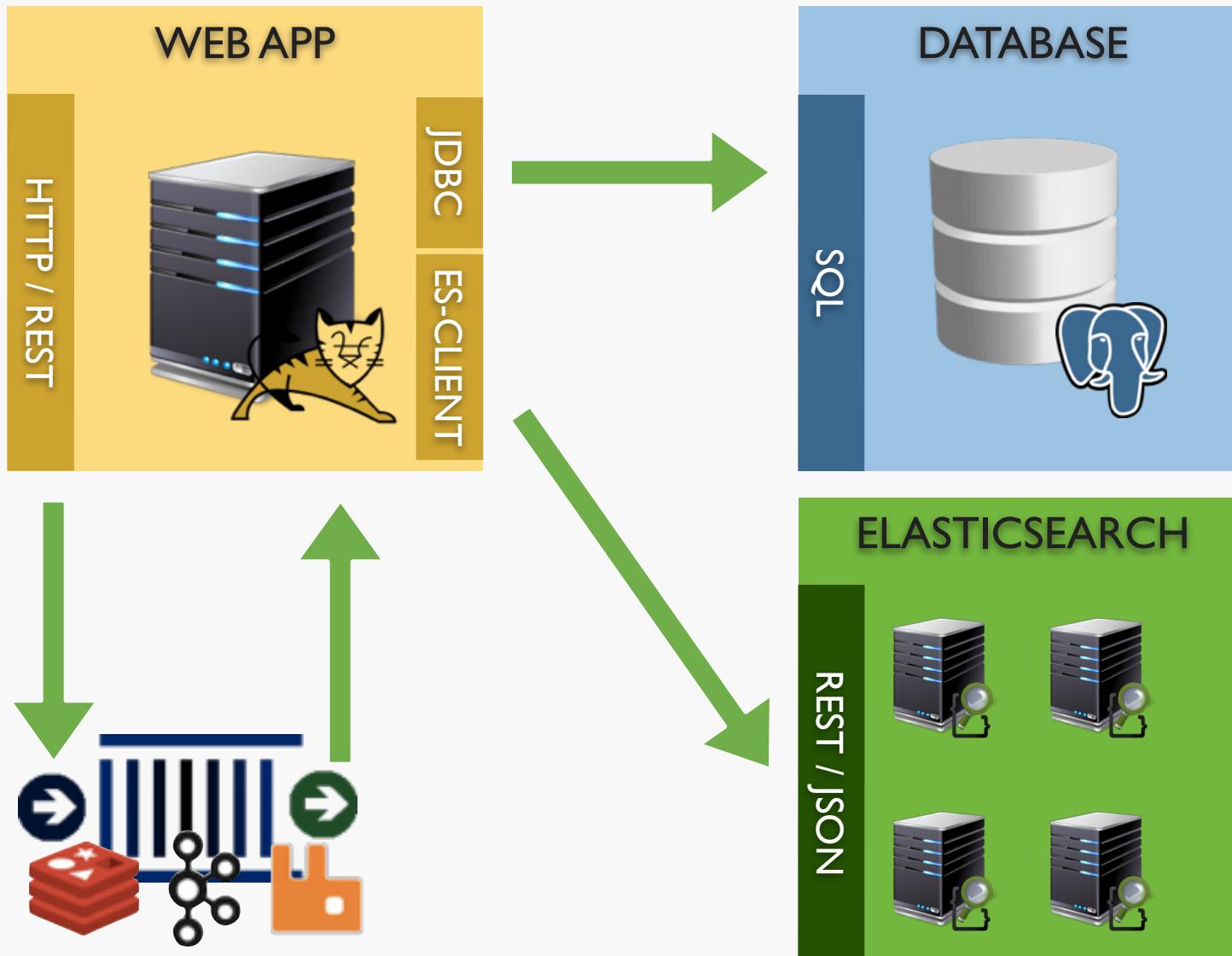
```
$ git checkout 01-direct
$ git checkout 02-bulk
$ git checkout 03-mapping
$ git checkout 04-aggs
$ git checkout 05-compute
$ mvn clean install jetty:run
$ cat README.markdown
```



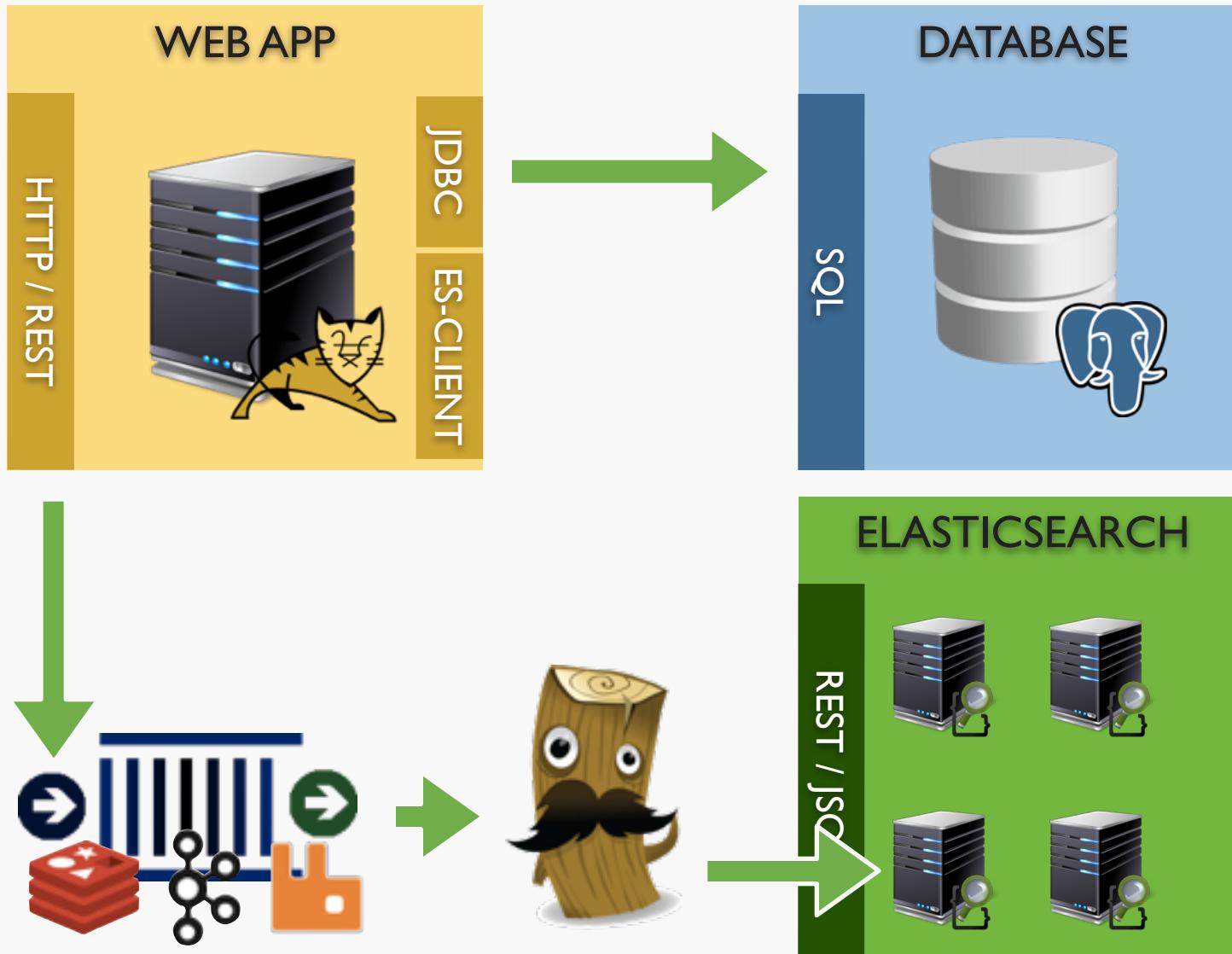
synchronous vs asynchronous



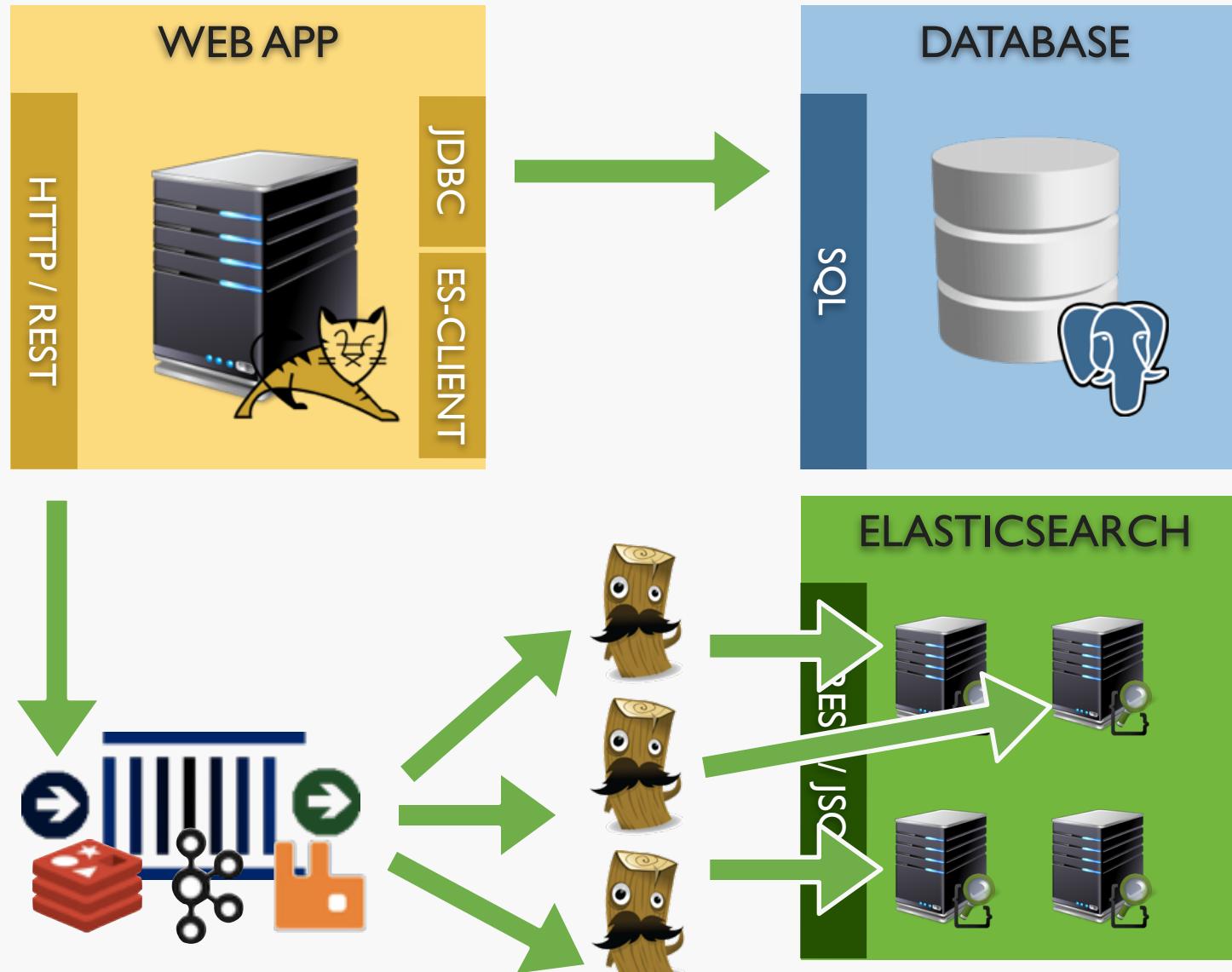
using brokers



using brokers



using brokers



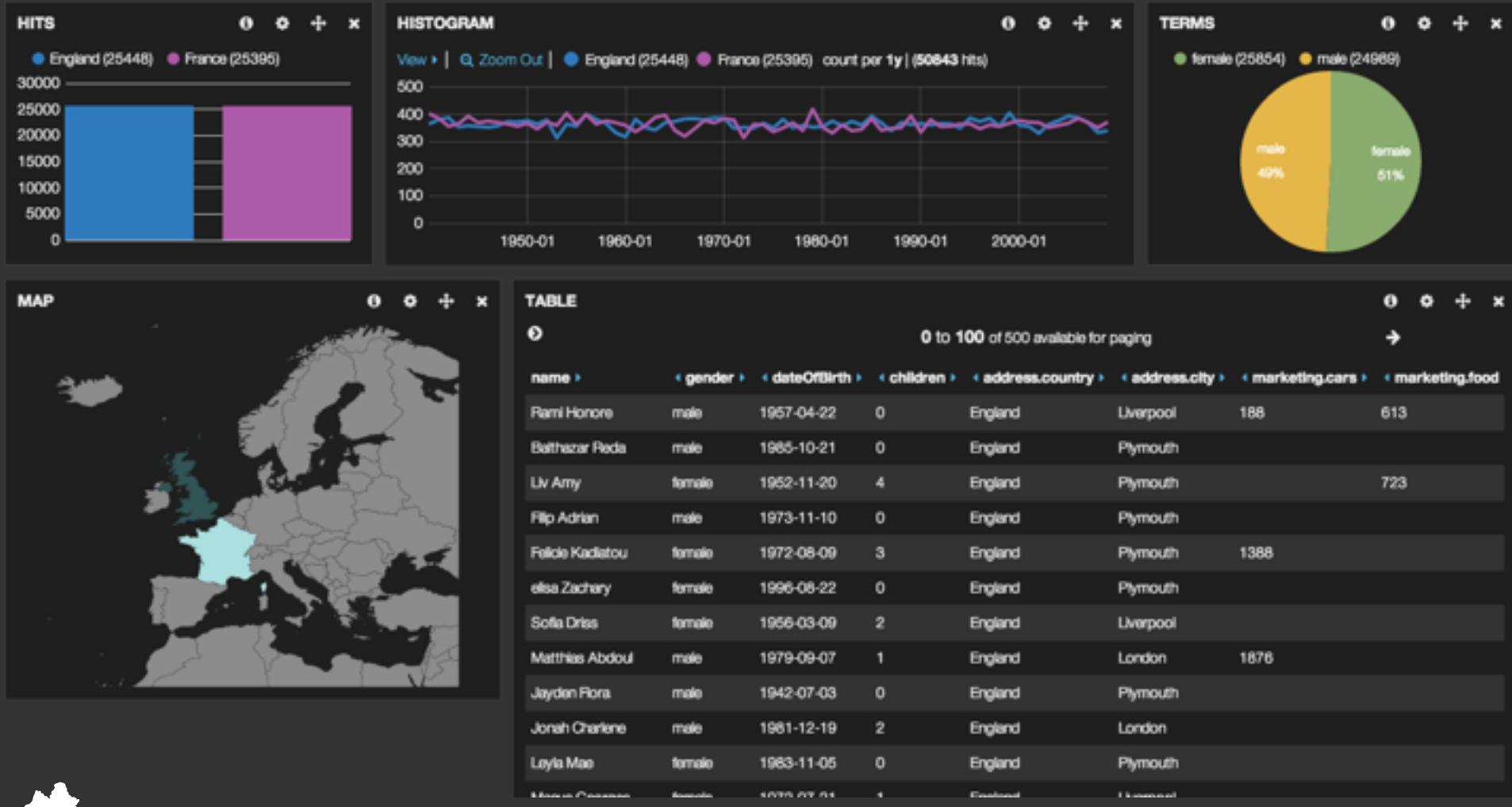
COME TO THE DARK SIDE



WE HAVE COOKIES!



kibana



David Pilato
Technical advocate

elasticsearch.
@dadoonet



<http://elasticsearch.com/support/>

We are hiring!