



advanced search for your legacy app



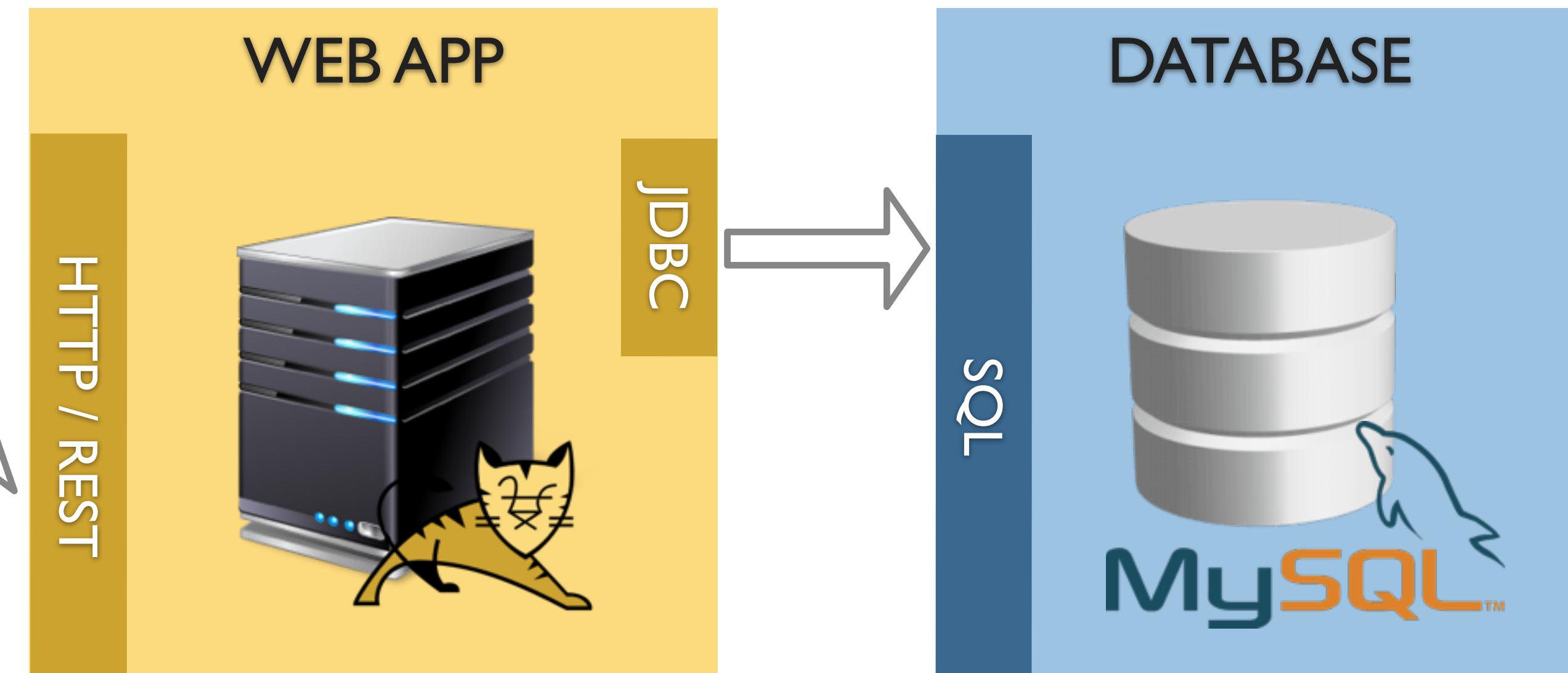
David Pilato
Developer | Evangelist
@dadoonet



our use case



our legacy platform

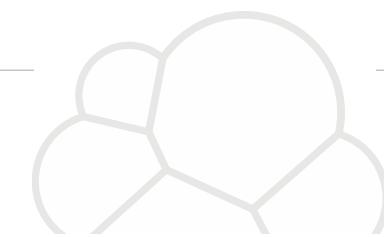
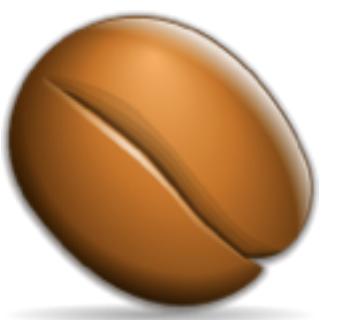
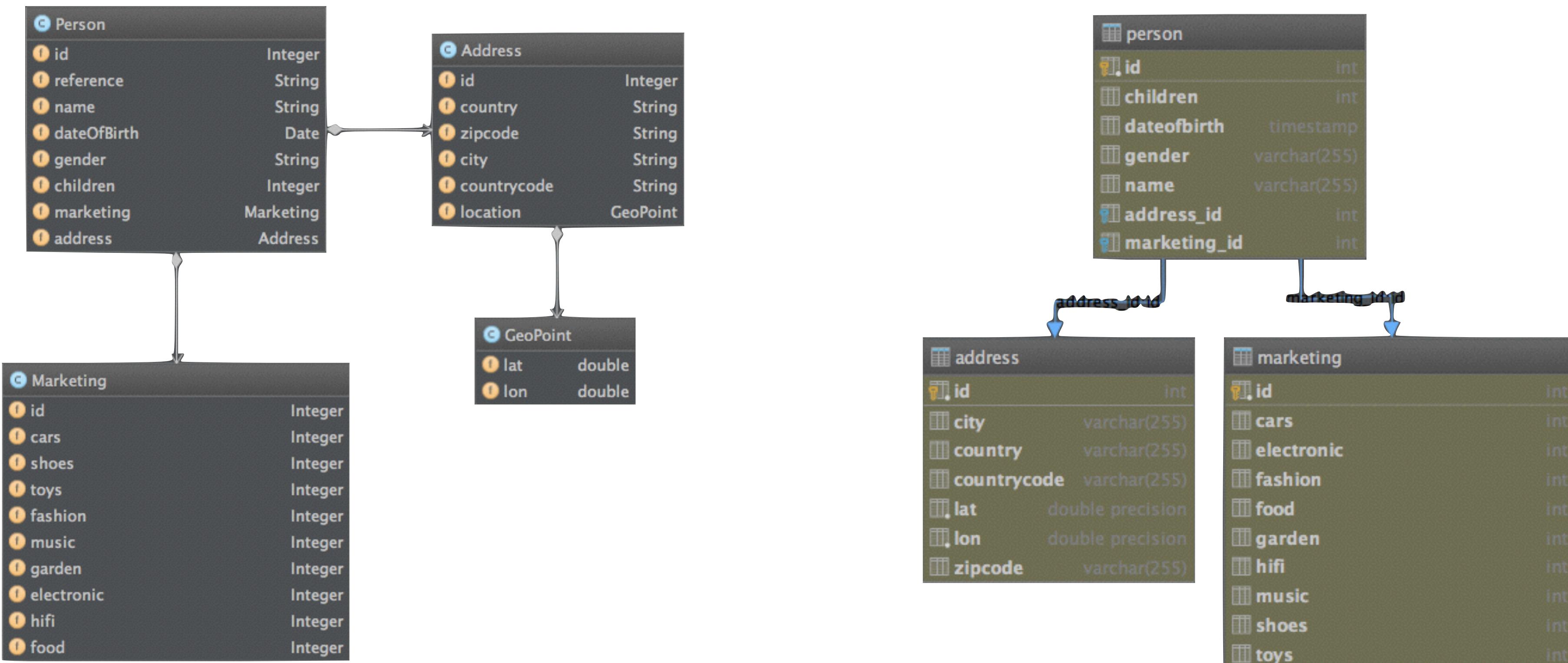


A screenshot of a legacy web application interface. The top navigation bar includes 'Contacts', 'Search', and 'Advanced' buttons. The search bar contains fields for 'Name...', 'Country...', and 'City...', with a 'Search' button. Below the search bar, a message states 'Found 100 hits in 35 ms'. A table displays contact information with columns: 'Id', 'Name', 'Date Of Birth', and 'City'. The data is as follows:

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

A red AngularJS logo is positioned at the bottom of the screenshot.

our legacy domain / database

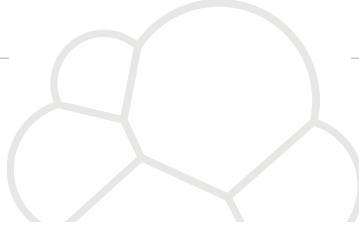


demo time

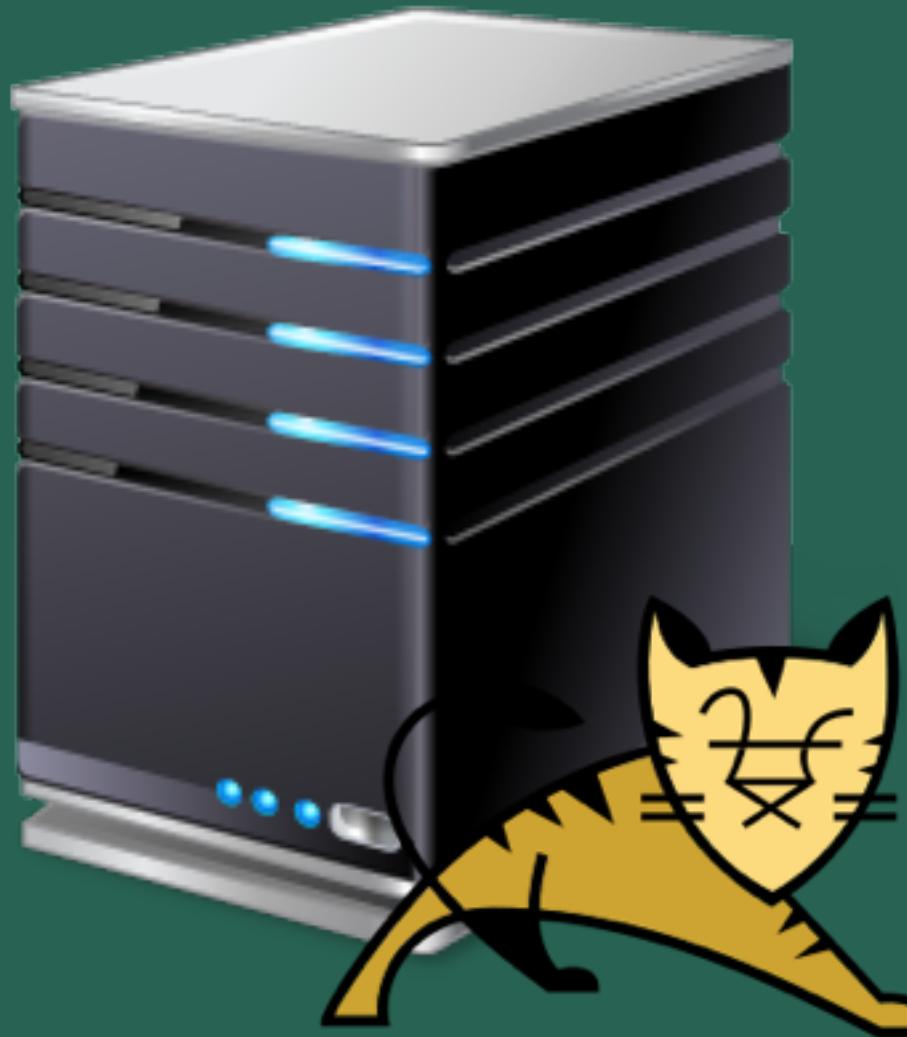
our legacy app



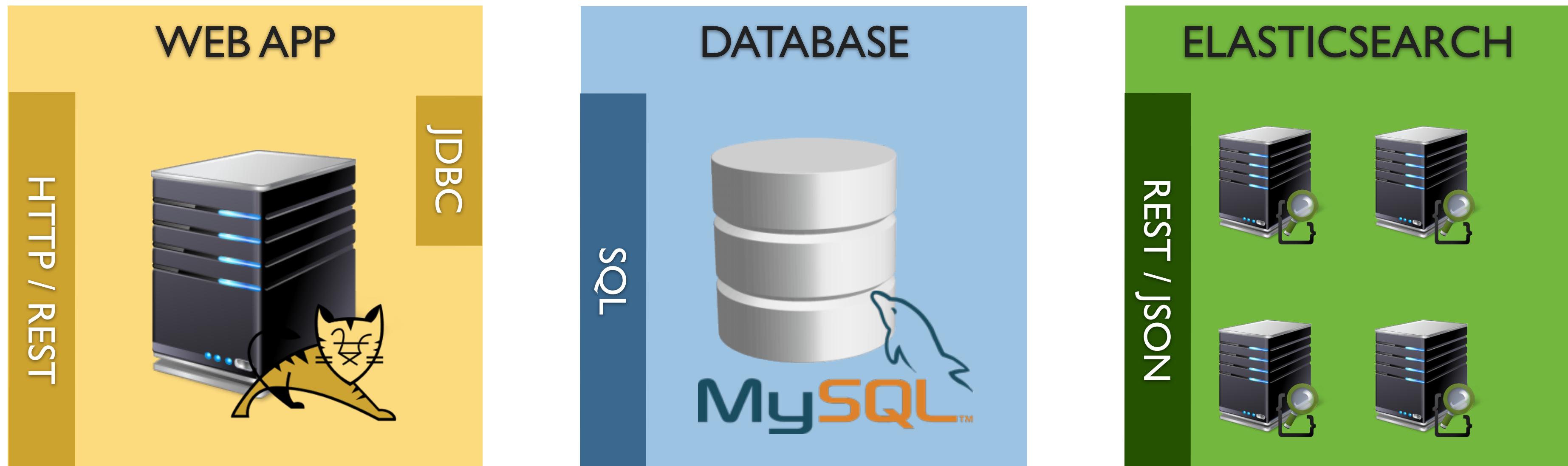
```
$ git clone https://github.com/dadoonet/legacy-search.git  
$ git checkout 00-legacy  
$ mvn clean install jetty:run
```



connecting with our app architecture



using a ETL



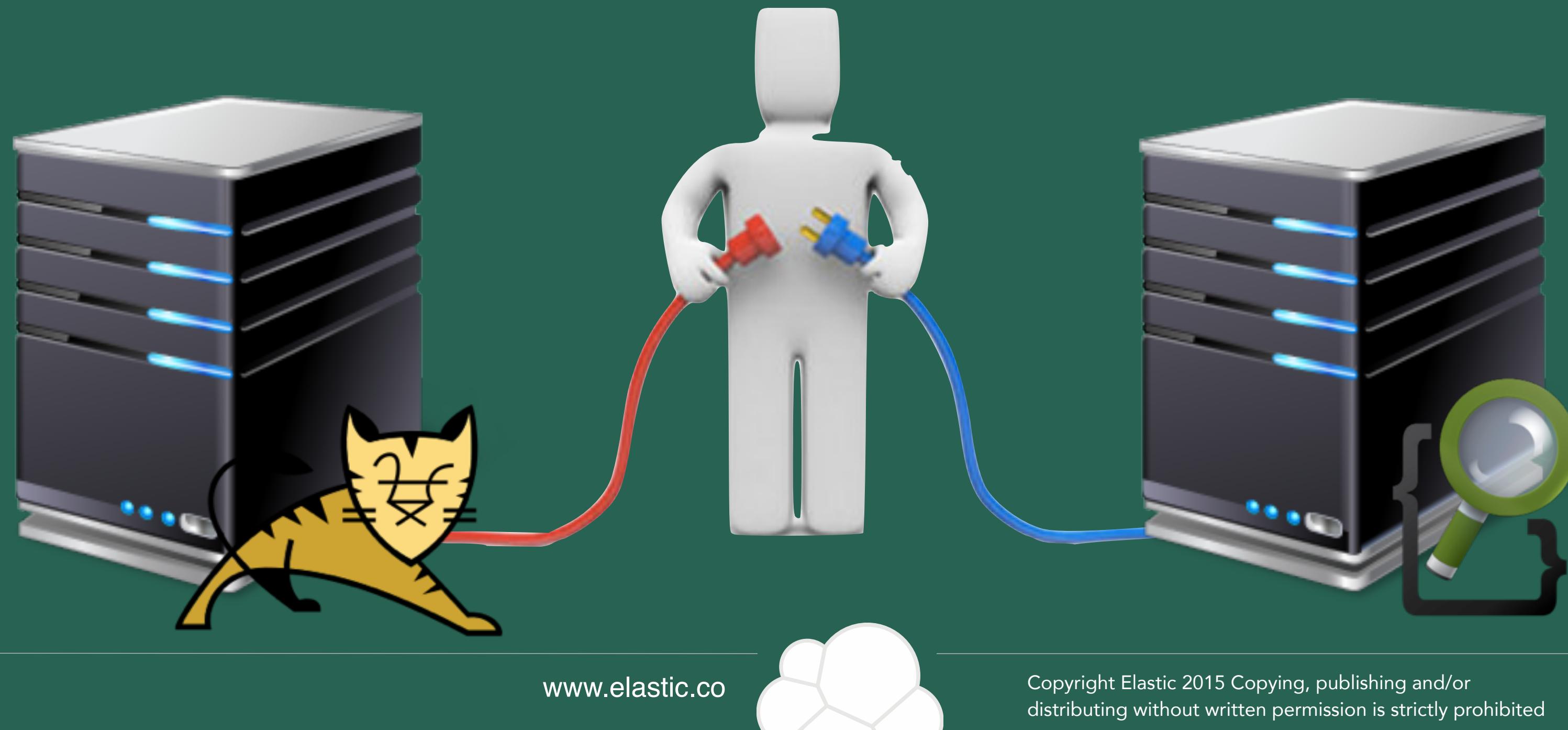
ROAD.

ENDS IN.

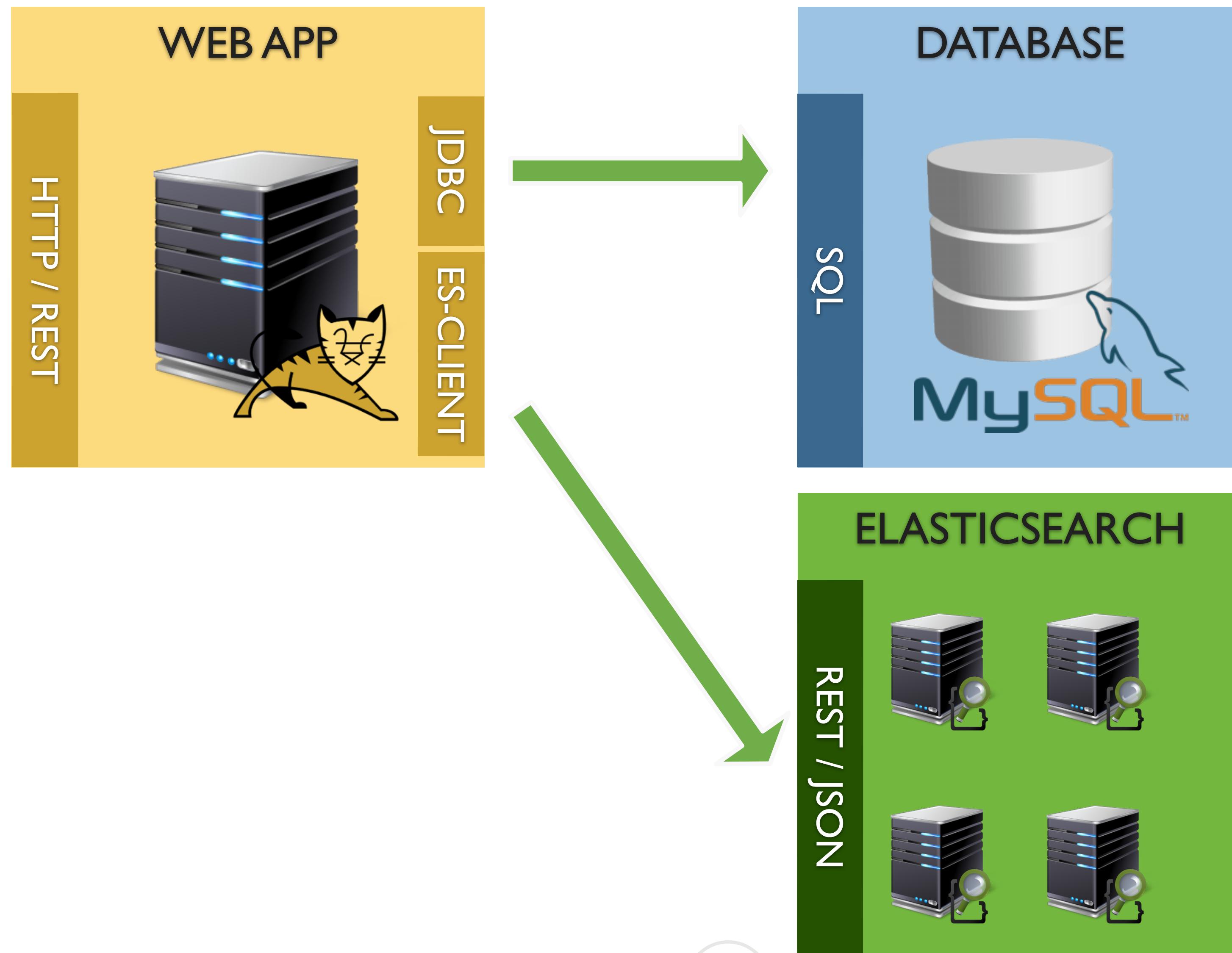
WATER



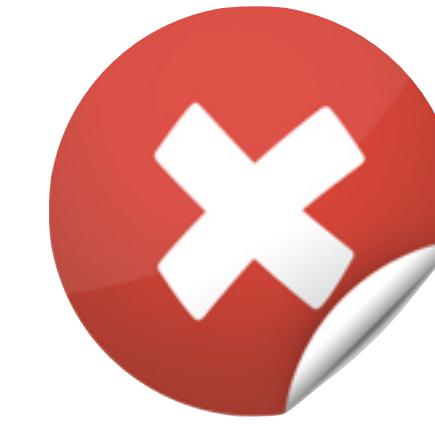
Do It Yourself direct connection



direct connection



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

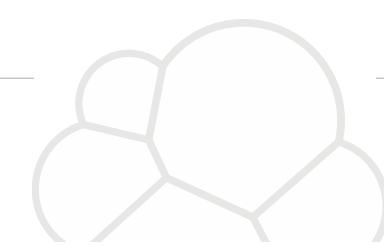
```
{  
  "name": "Joe Pink",  
  "dateOfBirth": "1971-12-26",  
  "address_id": "2",  
  "marketing_id": "3"  
}
```

PUT /person/address/2

```
{  
  "city": "Paris",  
  "country": "France"  
}
```

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  
  }  
}
```

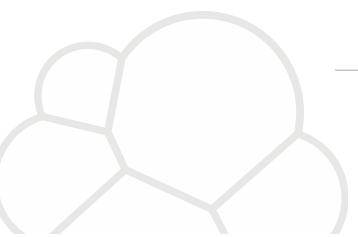
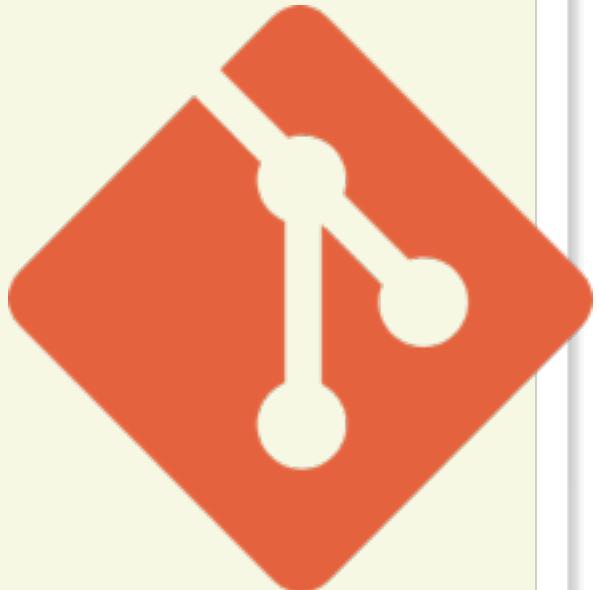


demo time

direct connection



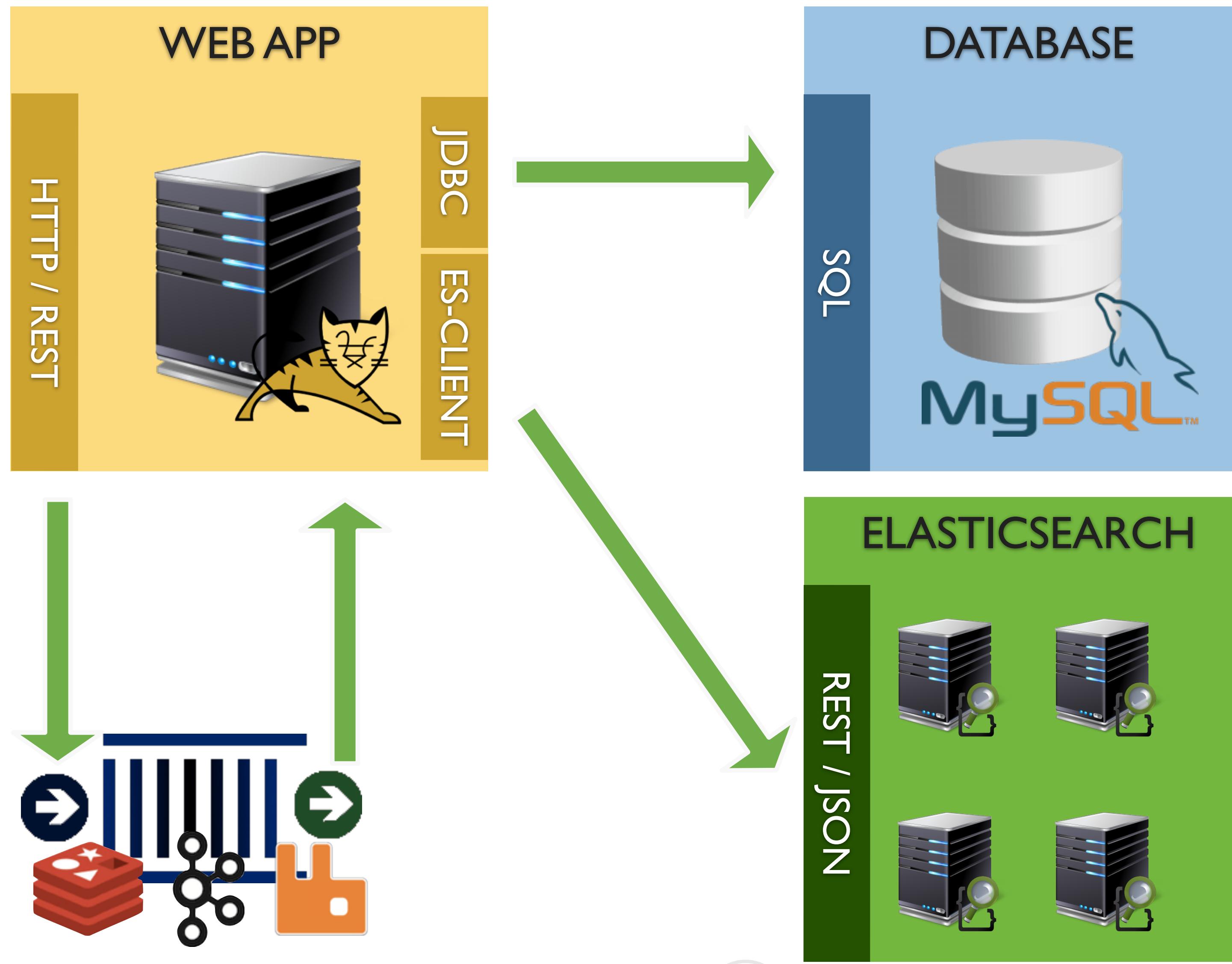
```
$ git checkout 01-direct
$ git checkout 02-bulk
$ git checkout 03-mapping
$ git checkout 04-aggs
$ git checkout 05-compute
$ mvn clean install jetty:run
$ cat README.markdown
```



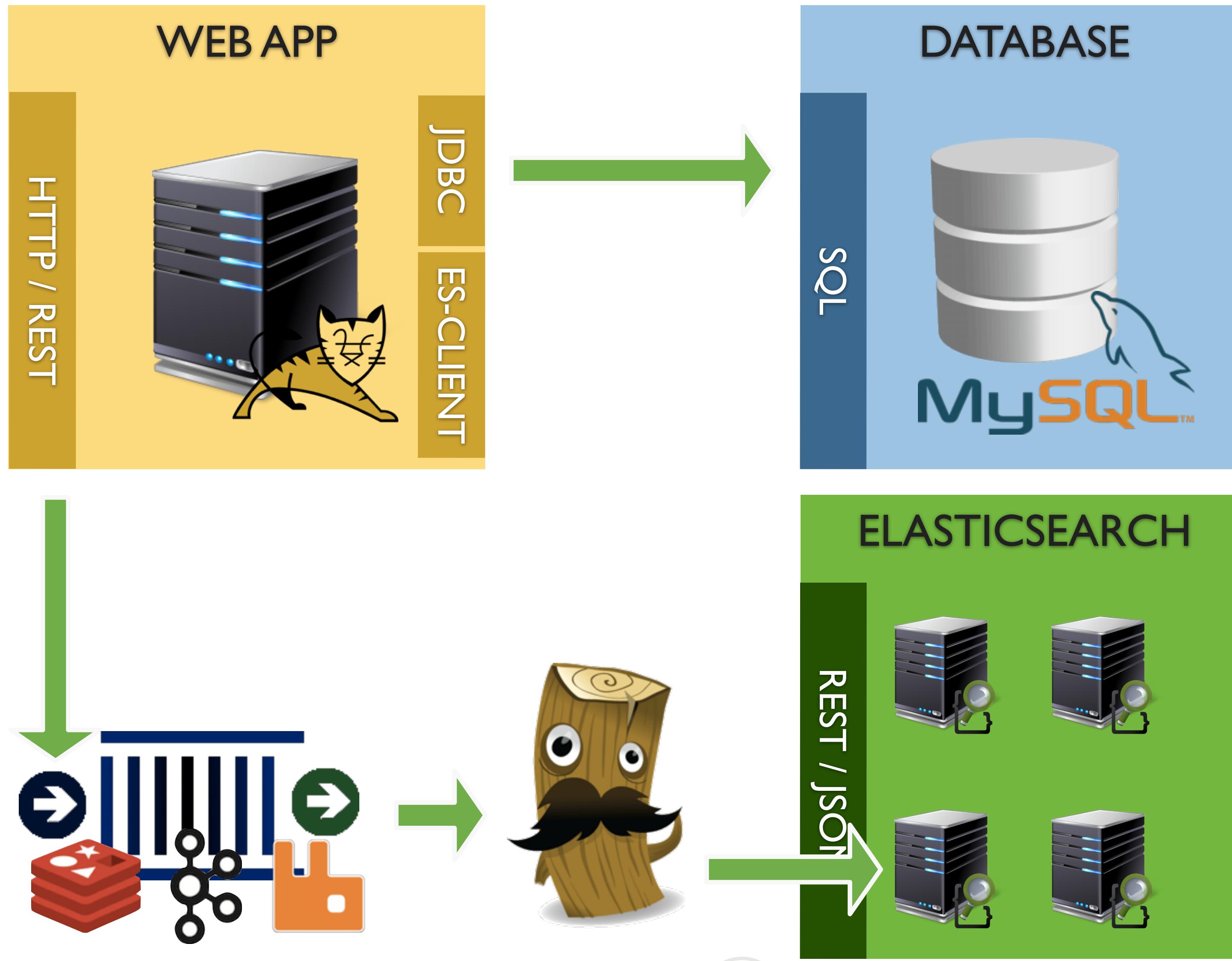
synchronous vs asynchronous



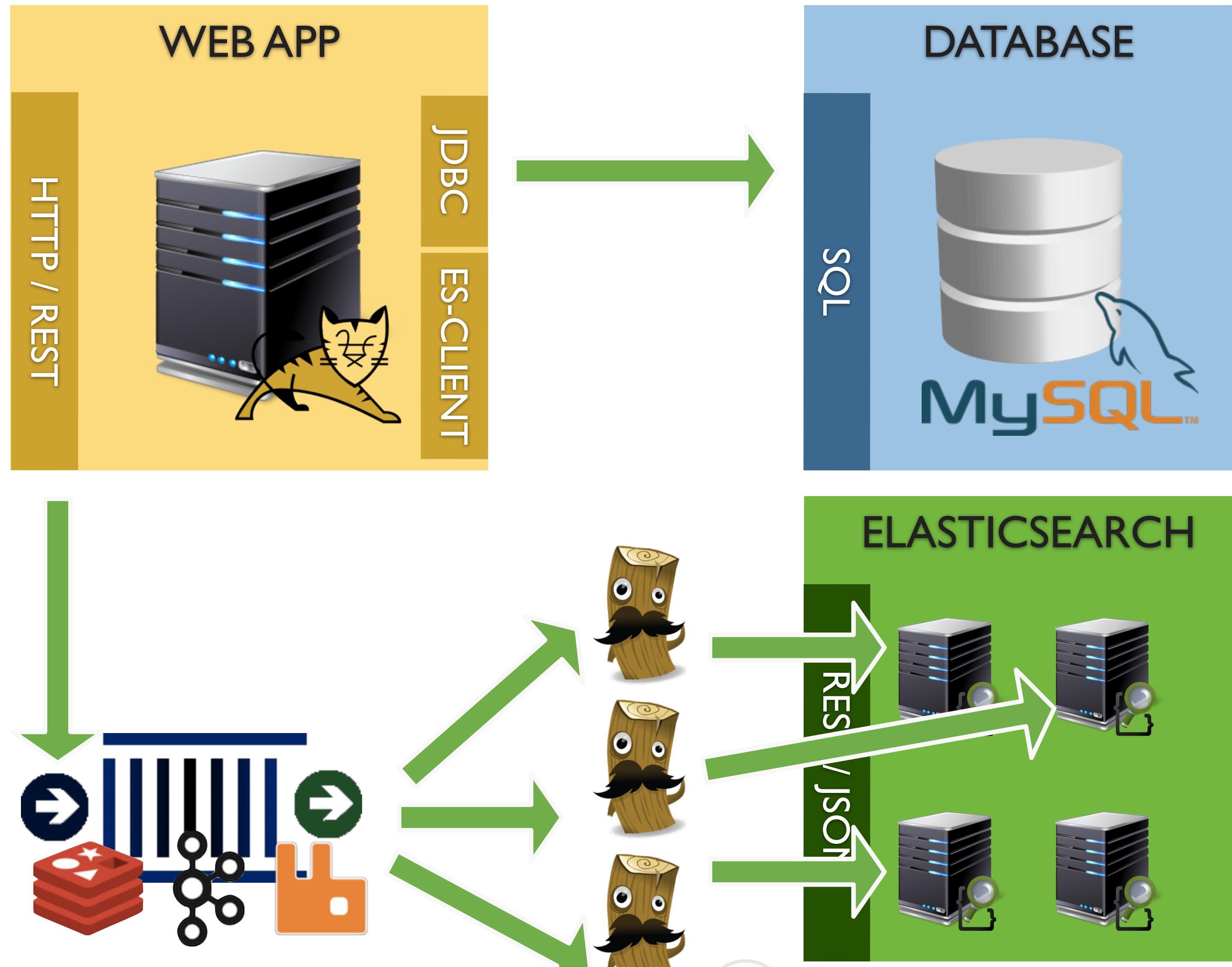
using brokers



using brokers



using brokers



COME TO THE DARK SIDE



WE HAVE COOKIES!



K4

Q

+

H

F

D

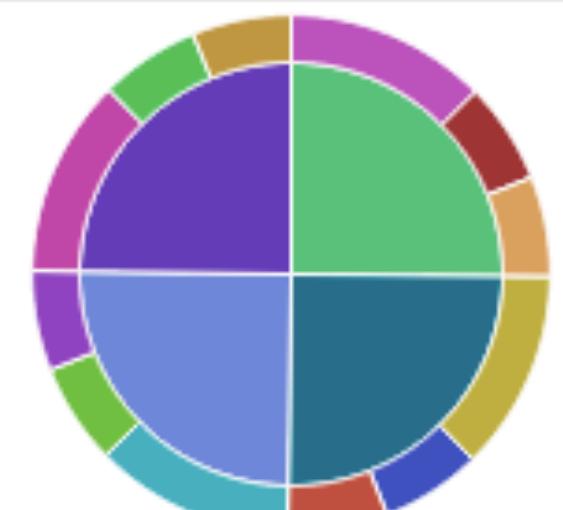
E

Help

Marketing data demo

This demo shows marketing data generated randomly.

Per country and city



Legend ⓘ
Italy
Germany
France
England
Turin
Ischia

Per gender



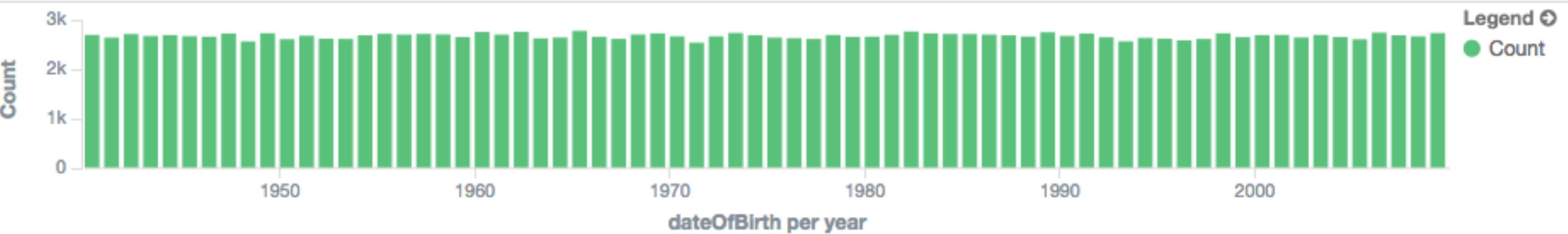
Legend ⓘ
female
male

Children AVG

1.995

Average children

By year



Legend ⓘ
Count

Stats per gender and city

Top 2 gender: female

Top 5 address.city ⓘ Q	Count ⓘ	Average children ⓘ
Turin	11839	1.986
Bonn	11822	2.012
Nantes	11770	1.998
Plymouth	11760	2.001
Ischia	6005	1.986

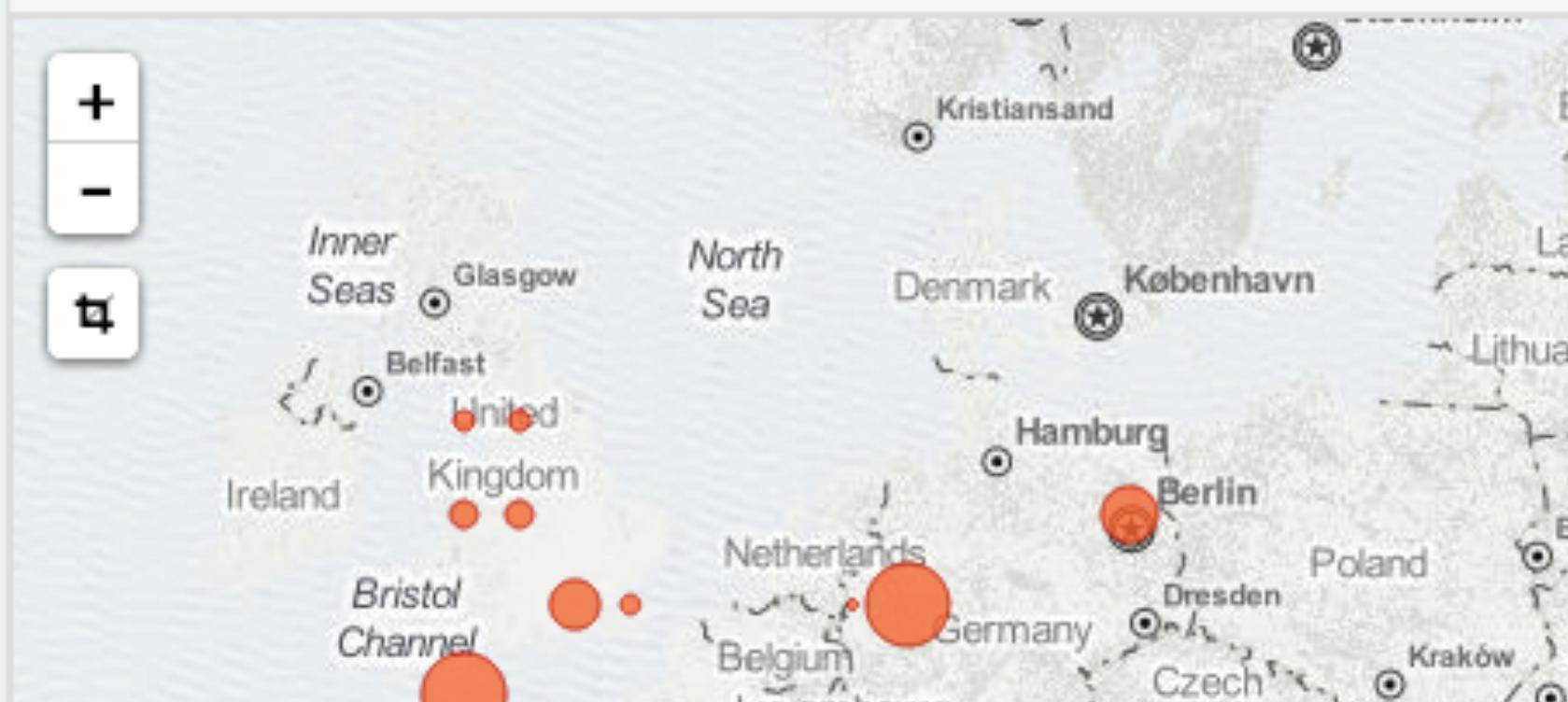
Export: [Raw](#) [Formatted](#)

Per country and gender



Legend ⓘ
England
Germany
France
Italy

Localisation



Top 2 gender: male

Top 5 address.city ⓘ Q	Count ⓘ	Average children ⓘ
Turin	11839	1.986
Bonn	11822	2.012
Nantes	11770	1.998
Plymouth	11760	2.001
Ischia	6005	1.986

David Pilato
Developer | Evangelist
@dadoonet



Thanks!



<https://www.elastic.co/subscriptions>

