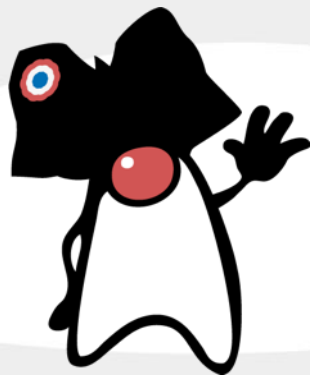




**David Pilato**  
Developer | Evangelist  
@dadoonet

# Advanced search for your legacy application















Feb 7, 2010 – Sep 5, 2016

Contributions to master, excluding merge commits

Contributions: Commits ▾



kimchy

4,737 commits / 1,401,534 ++ / 1,064,899 --

#1



s1monw

2,216 commits / 1,219,296 ++ / 1,217,559 --

#2



dadoonet

1,171 commits / 128,081 ++ / 84,741 --

#3



javanna

1,101 commits / 142,713 ++ / 119,744 --

#4



martijnvg

1,093 commits / 212,940 ++ / 142,674 --

#5



clintongormley

854 commits / 232,442 ++ / 30,976 --

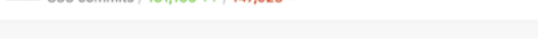
#6



rmuir

833 commits / 161,166 ++ / 147,023 --

#7



jasontedor

772 commits / 62,452 ++ / 37,875 --

#8



jpountz

737 commits / 173,140 ++ / 204,220 --

#9



# Elastic Stack

User Interface



Store, Index,  
& Analyze



Ingest



Security

Alerting

Monitoring

Reporting

Graph

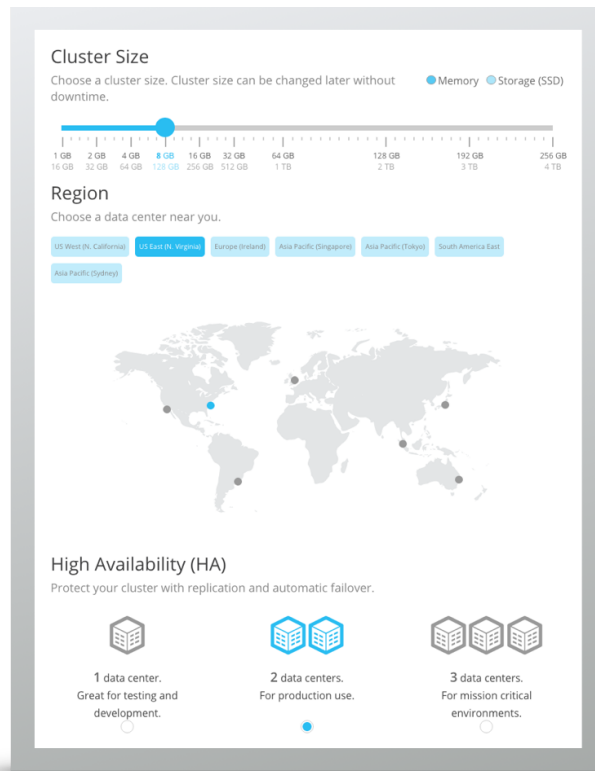






## The only Elasticsearch as a Service offering powered by the creators of the Elastic Stack

- Always runs on the latest software
- One-click to scale/upgrade with no downtime
- Free Kibana and backups every 30 minutes
- Dedicated, SLA-based support
- Easily add X-Pack features: security (Shield), alerting (Watcher), and monitoring (Marvel)
- Pricing starts at \$45 a month



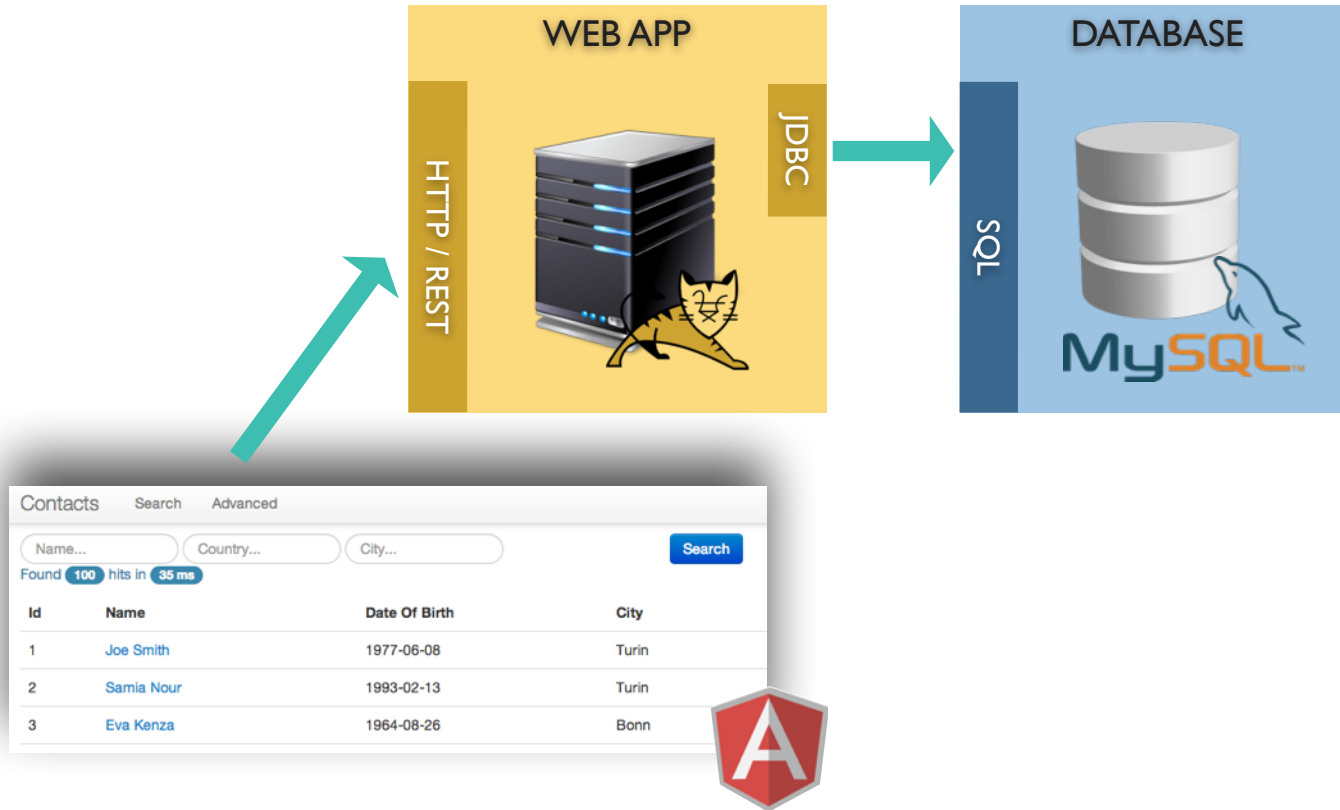
# Elastic Subscription Packages

Elastic Stack	BASIC	GOLD	PLATINUM
Elasticsearch, Kibana, Logstash, Beats	✓	✓	✓
X-Pack			
Marvel (Monitoring & Management)	✓	✓ Multiclust er Support >7 days data retention	✓ Multiclust er Support >7 days data retention
Shield (Security)		✓	✓ Field/document security Custom realms
Watcher (Alerting)		✓	✓
Reporting		✓	✓
Graph Analytics & Visualization			✓
Support			
Support coverage and response times provided by Elastic Support Engineers		Business Hours 4 hour L1 response times	24/7/365 1 hour L1 response times Emergency patches



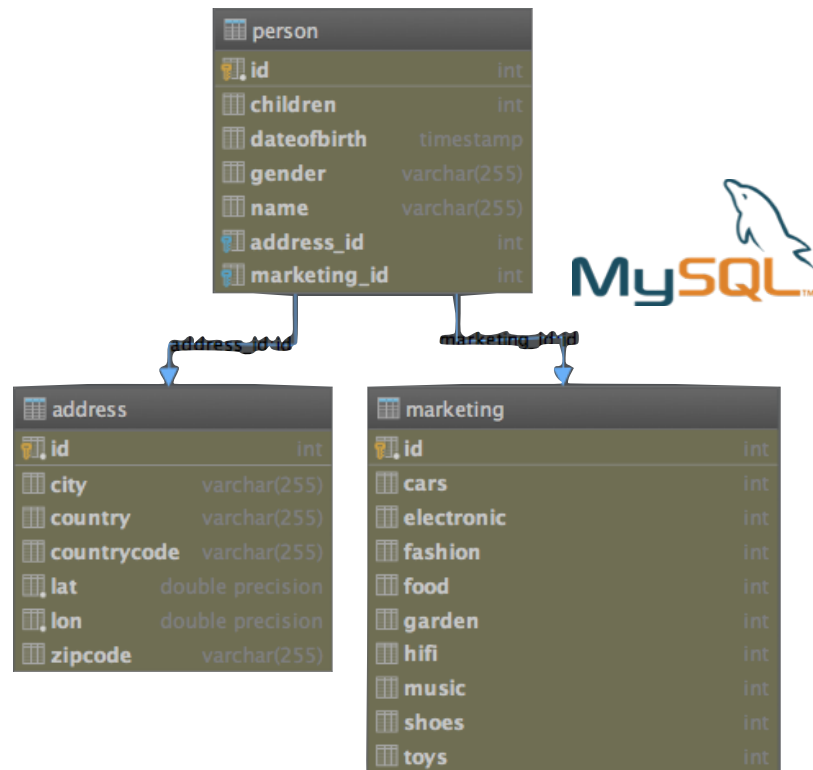
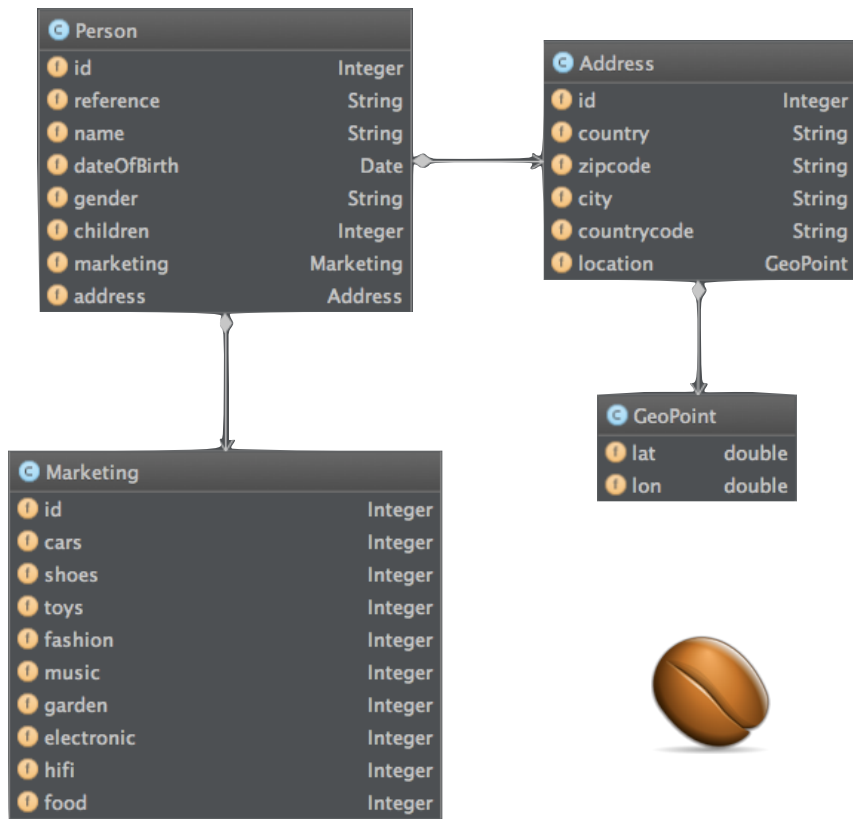


# our legacy platform





# our legacy domain / database



# demo time

our legacy application



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







# Some data

```
CREATE TABLE user
(
  name VARCHAR(100),
  comments VARCHAR(1000)
);
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Docker');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

## Search on term

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');  
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at  
french customs service');  
INSERT INTO user VALUES ('David Gageot', 'Engineer at Docker');  
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name="David";  
Empty set (0,00 sec)
```

# Search like

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Docker');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David%";
```

name	comments
David Pilato	Developer at elastic
David Gageot	Engineer at Docker
David David	Who is that guy?

# Search like

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Docker');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David%Pilato%";
```

name	comments
David Pilato	Developer at elastic



## Search like with inverted terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');  
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at  
french customs service');  
INSERT INTO user VALUES ('David Gageot', 'Engineer at Docker');  
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%Pilato%David%";  
Empty set (0,00 sec)
```

## Search in two fields

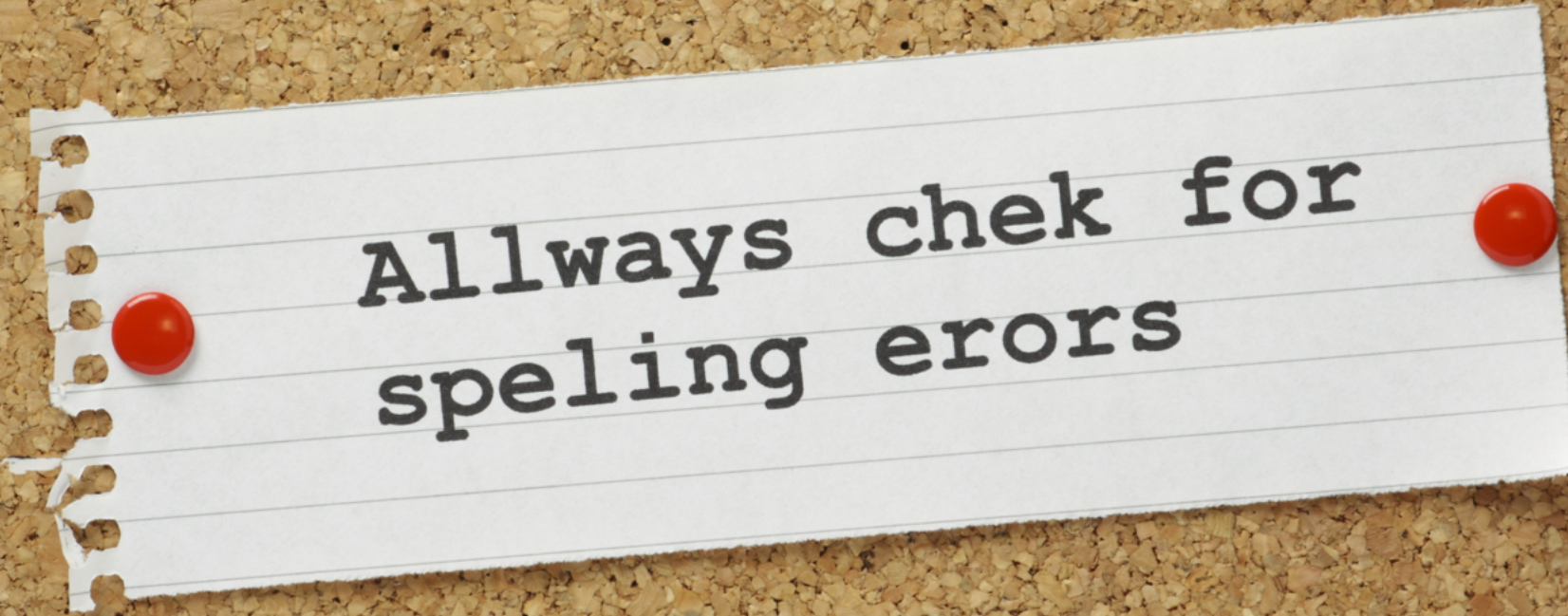
```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Docker');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David%" OR
                        comments LIKE "%David%";
```

name	comments
David Pilato	Developer at elastic
Malloum Laya	Worked with David at french customs service
David Gageot	Engineer at Docker
David David	Who is that guy?







Allways chek for  
speling erors

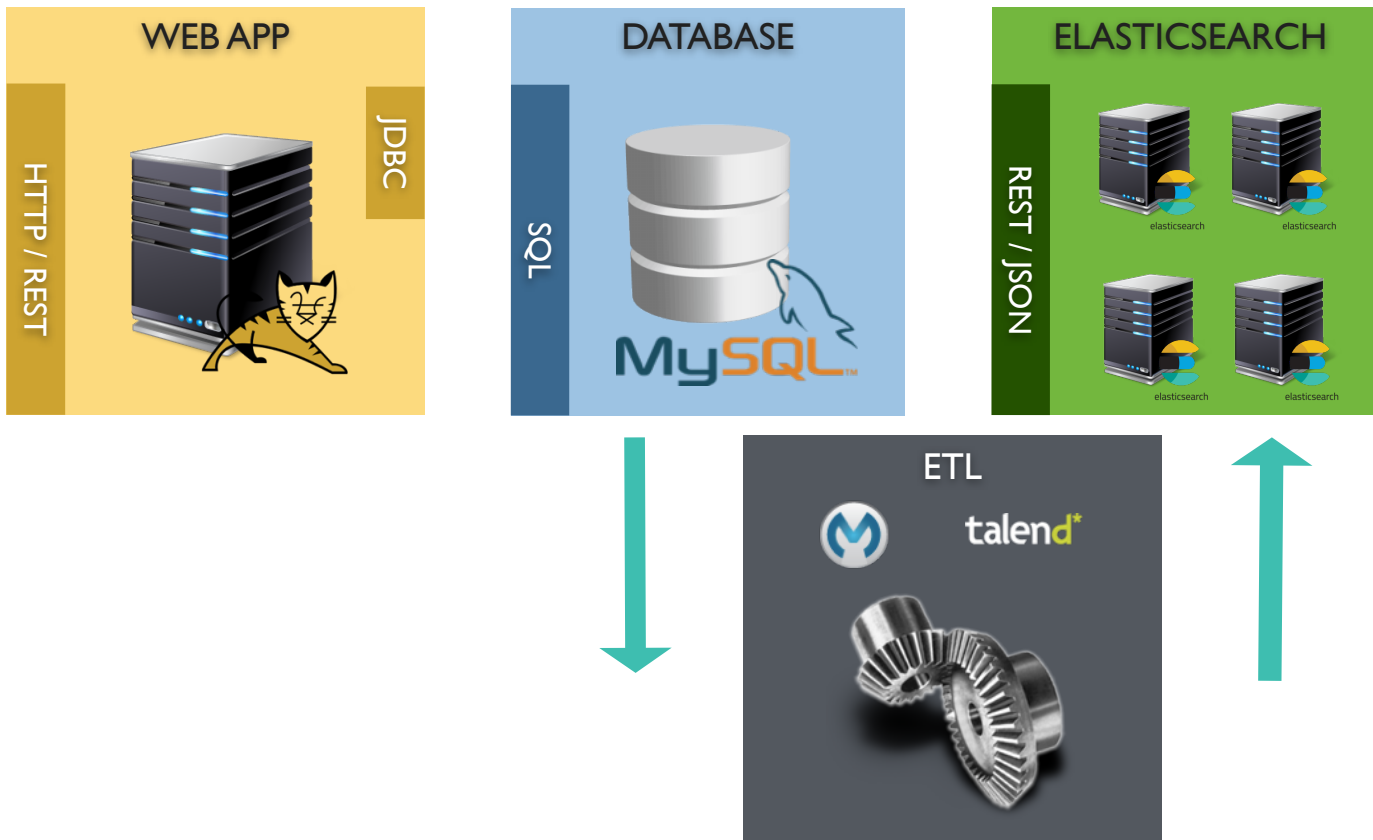




# Connecting with our app Architecture



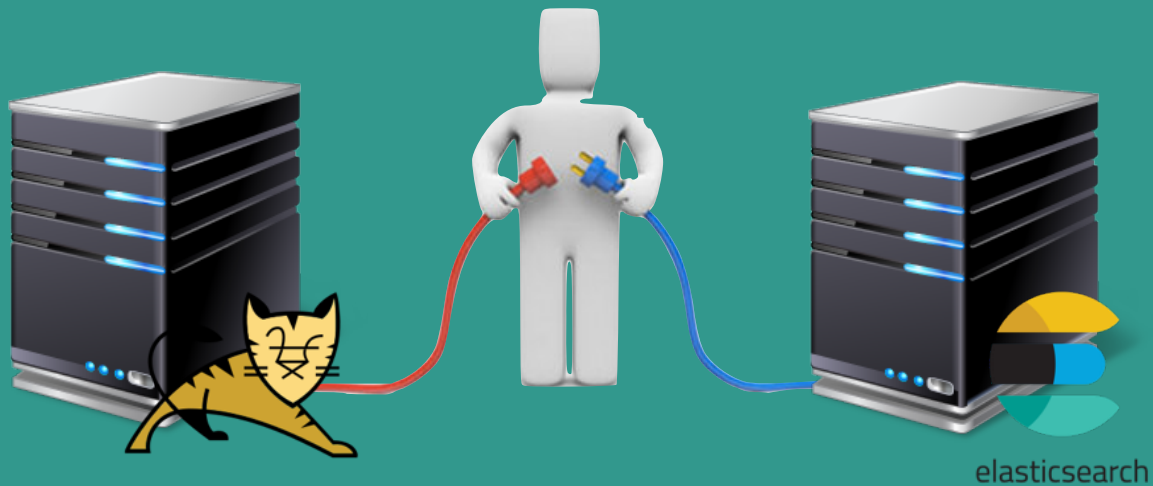
# using a ETL



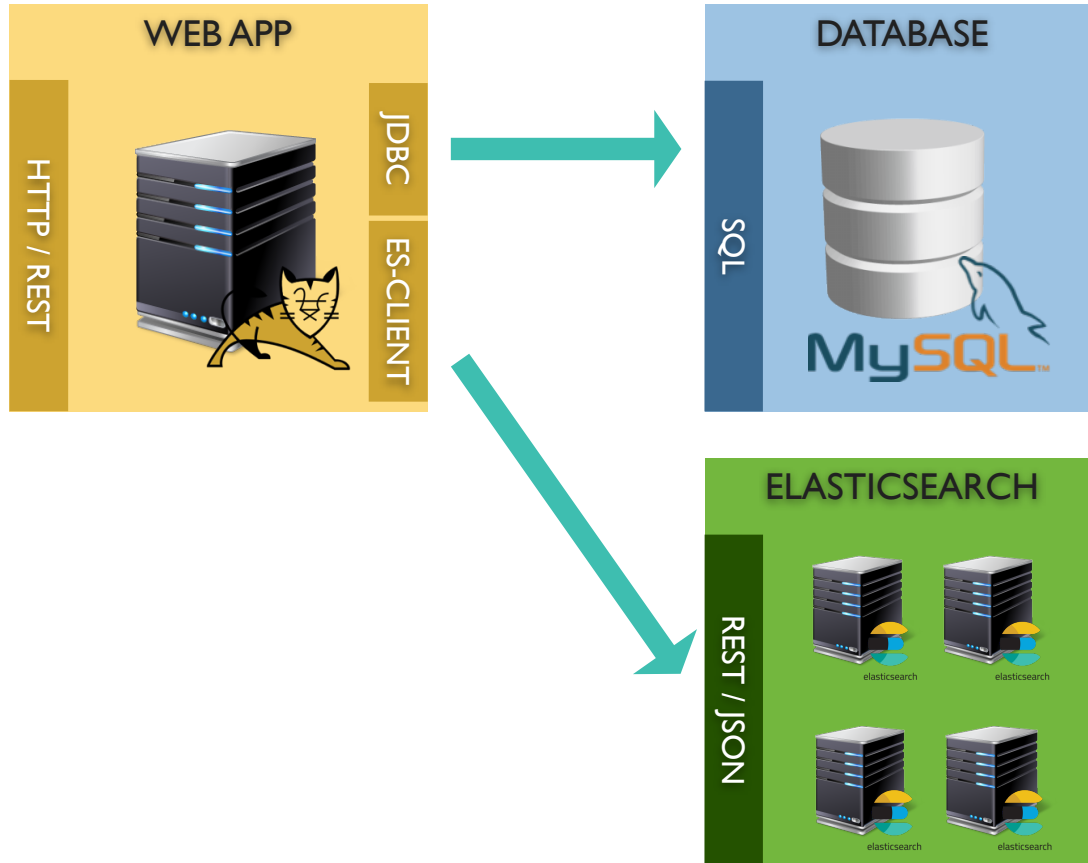




# Do It Yourself Direct Connection



# 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

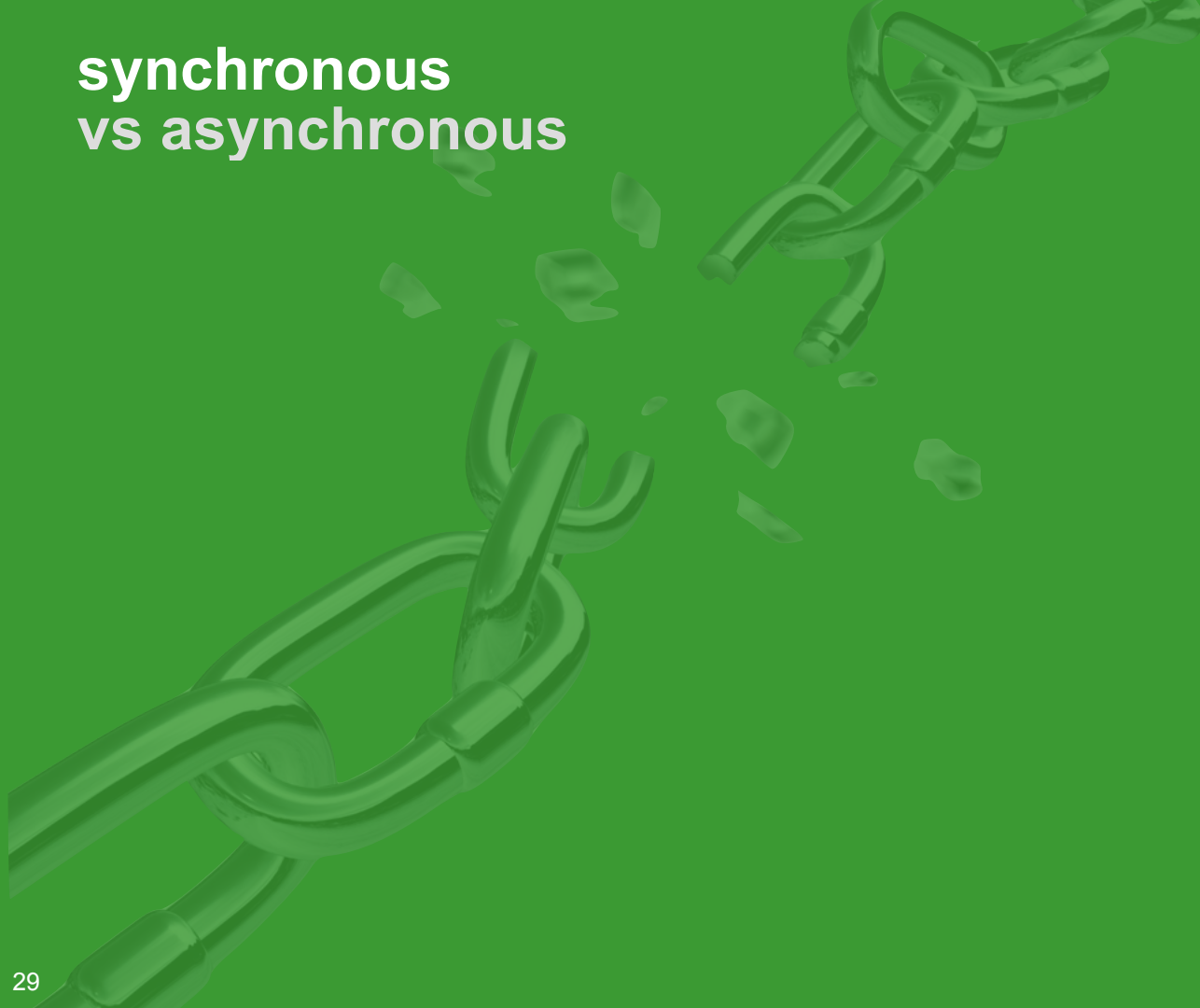
direct connection



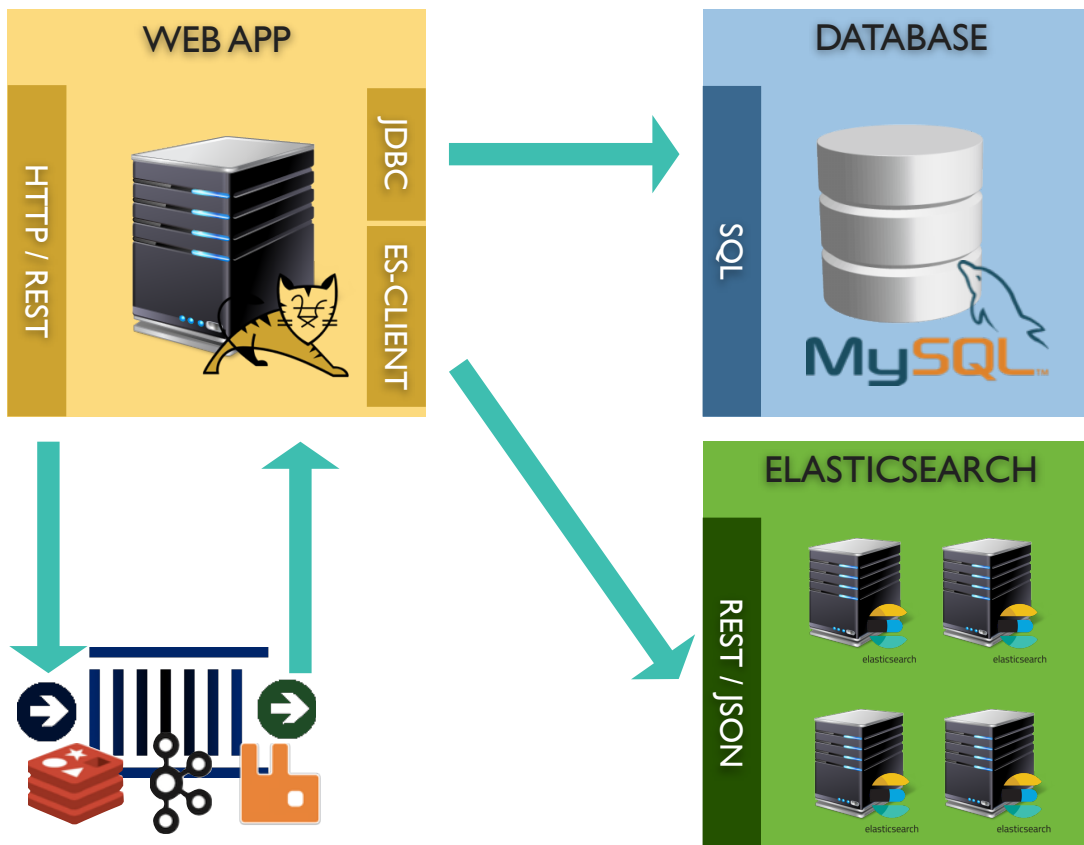
```
$ git clone https://github.com/dadoonet/legacy-search.git
$ git checkout 01-direct
$ git checkout 02-bulk
$ git checkout 03-mapping
$ git checkout 04-aggs
$ git checkout 05-compute
$ mvn clean install jetty:run
$ cat README.markdown
```



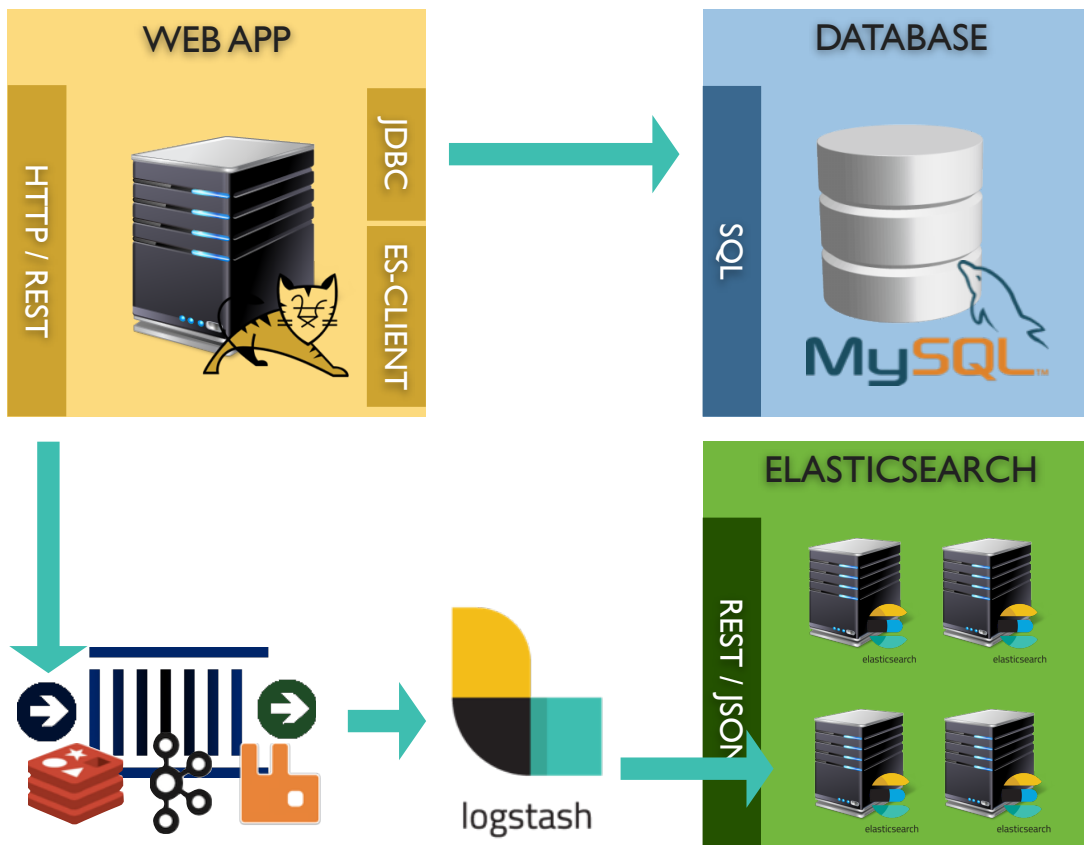
# synchronous vs asynchronous



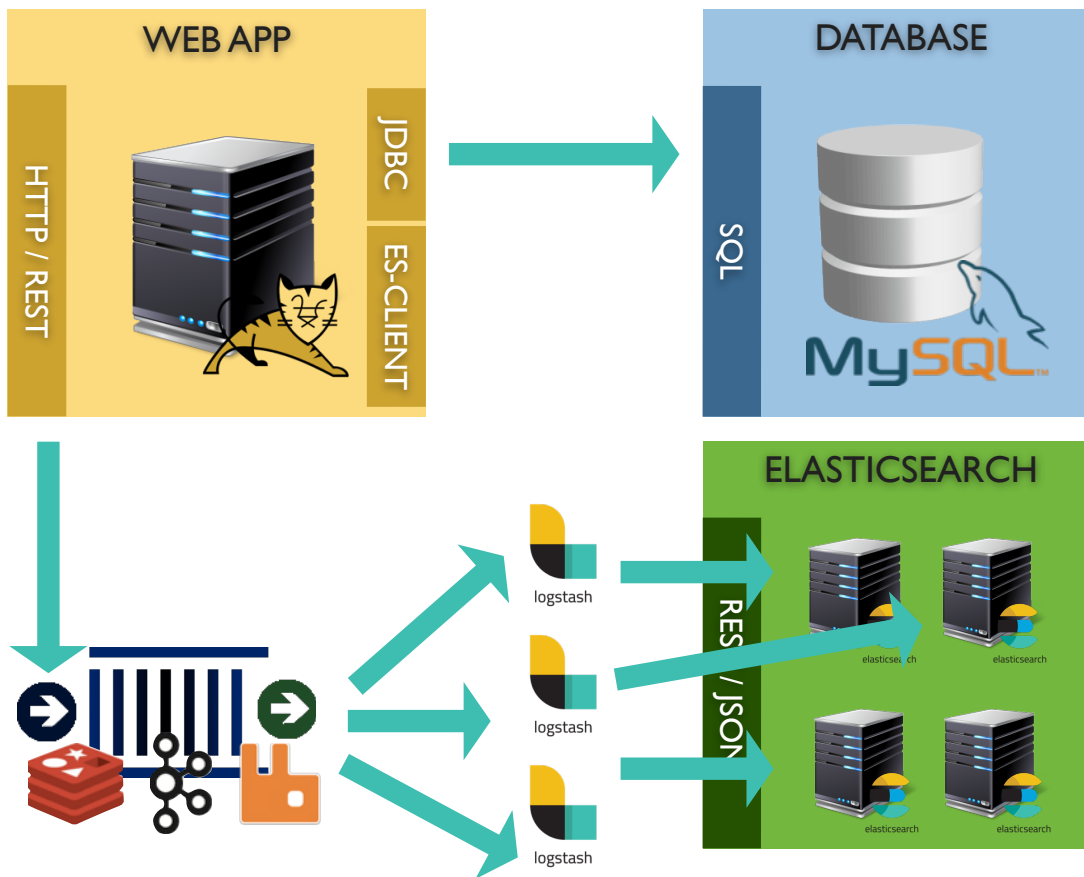
# using brokers



# using brokers



# using brokers



**COME TO THE DARK SIDE**

**WE HAVE COOKIES!**



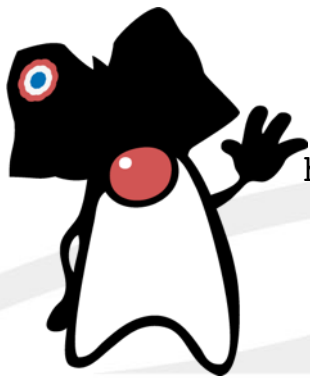




**David Pilato**  
Developer | Evangelist  
@dadoonet

# Thanks!

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



<https://github.com/dadoonet/legacy-search.git>

