



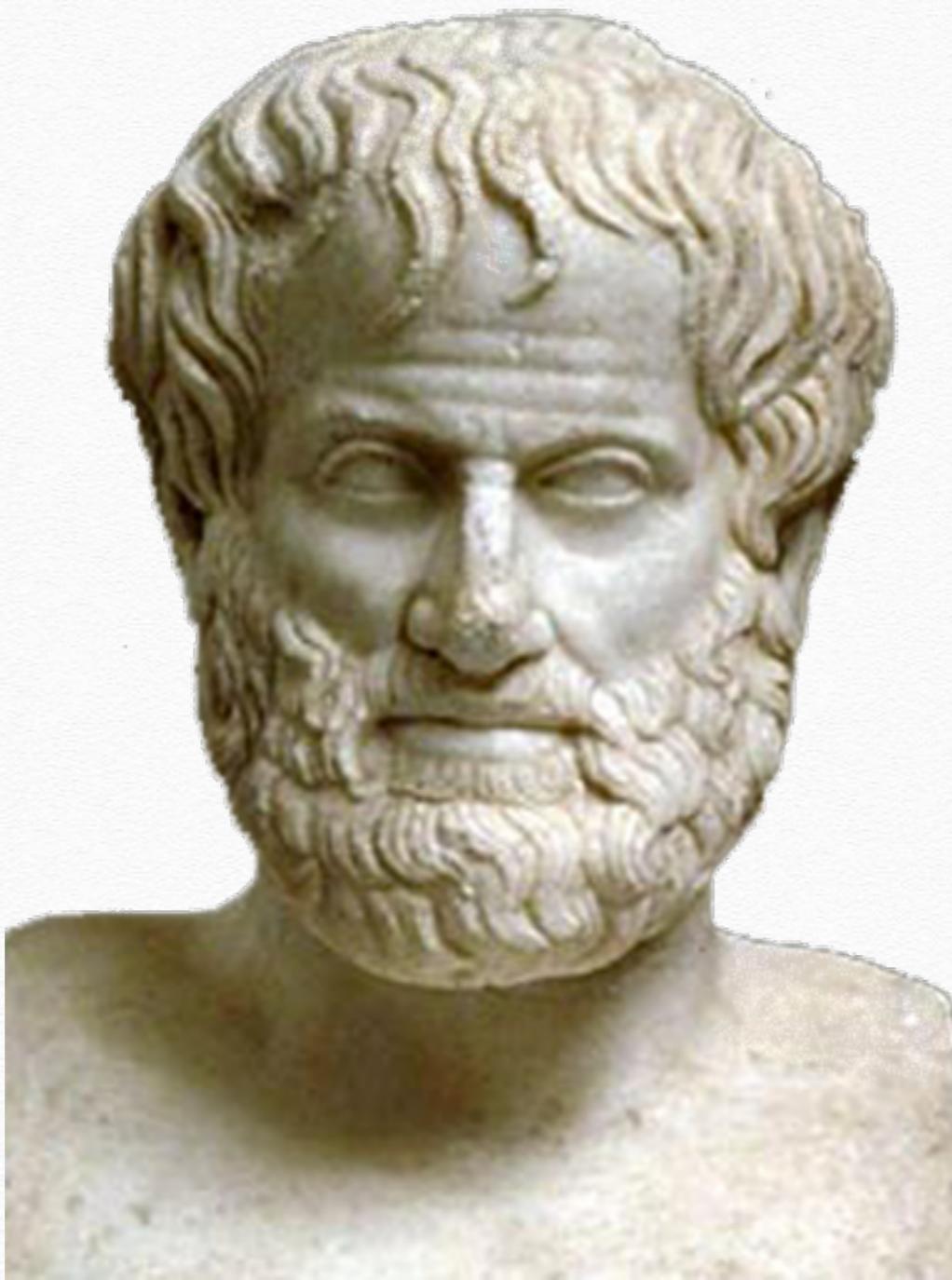
Rails & Elasticsearch

Las Vegas **Ruby** Group

Why add search?

My reasons:

- ❖ Empower users to find what they want quickly and efficiently
- ❖ Keep users on my website instead of sending them to Google, or never coming back



Sorry Aristotle, I don't
have a third reason

Before we get started...

KNOW YOUR DATA!

(easily done by spending a bunch of time in the rails console)

gem 'elasticsearch-rails'

Elasticsearch

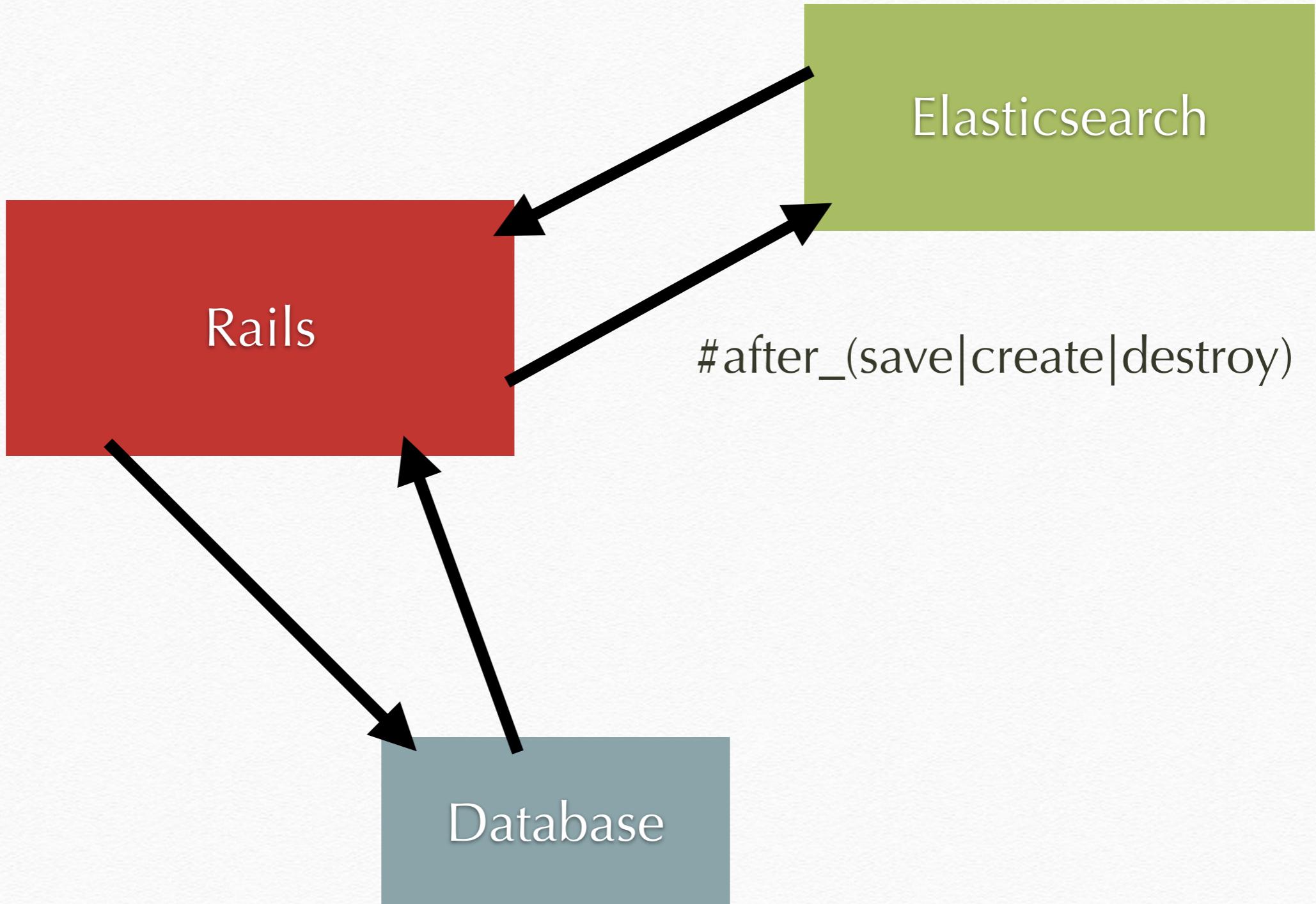
Rails

Database

<http://rubydoc.info/gems/elasticsearch-rails>

Let's look at some code!

```
include Elasticsearch::Model::Callbacks
```



Rabbits play in a field

**KITCHEN ASSISTENT
REQUIRED,**

**GOOD ENGLISH
ESSENCIAL,
APPLY WITHIN**



ENGLISHFUNNY.com

Rabbit

Ribbat

R**i**abbait

2 Steps

Rabbit

Rabbit

1 Step

Will both of these match if fuzziness is set to 1?

Rabb**bit**

R*ia*bb**a**it

completion suggester



- te
- p
- c



In order to understand the format of suggestions, please read the [Suggesters](#) page first.

NOTE

The `completion` suggester is a so-called prefix suggester. It does not do spell correction like the `term` or `phrase` suggesters but allows basic auto-complete functionality.

why another suggester? why not prefix queries?



The first question which comes to mind when reading about a prefix suggestion is, why you should use it all, if you have prefix queries already. The answer is simple: Prefix suggestions are fast.

The data structures are internally backed by Lucenes `AnalyzingSuggester`, which uses FSTs to execute suggestions. Usually these data structures are costly to create, stored in-memory and need to be rebuilt every now and then to reflect changes in your indexed documents. The `completion` suggester circumvents this by storing the FST as part of your index during index time. This allows for really fast loads and executions.



aka Filtering results (by distance)



#degen



N-Grams

Set N-grams to 1

Elephant

“e” “l” “e” “p” “h” “a” “n” “t”

...I have no idea why anyone would use this

Synonyms

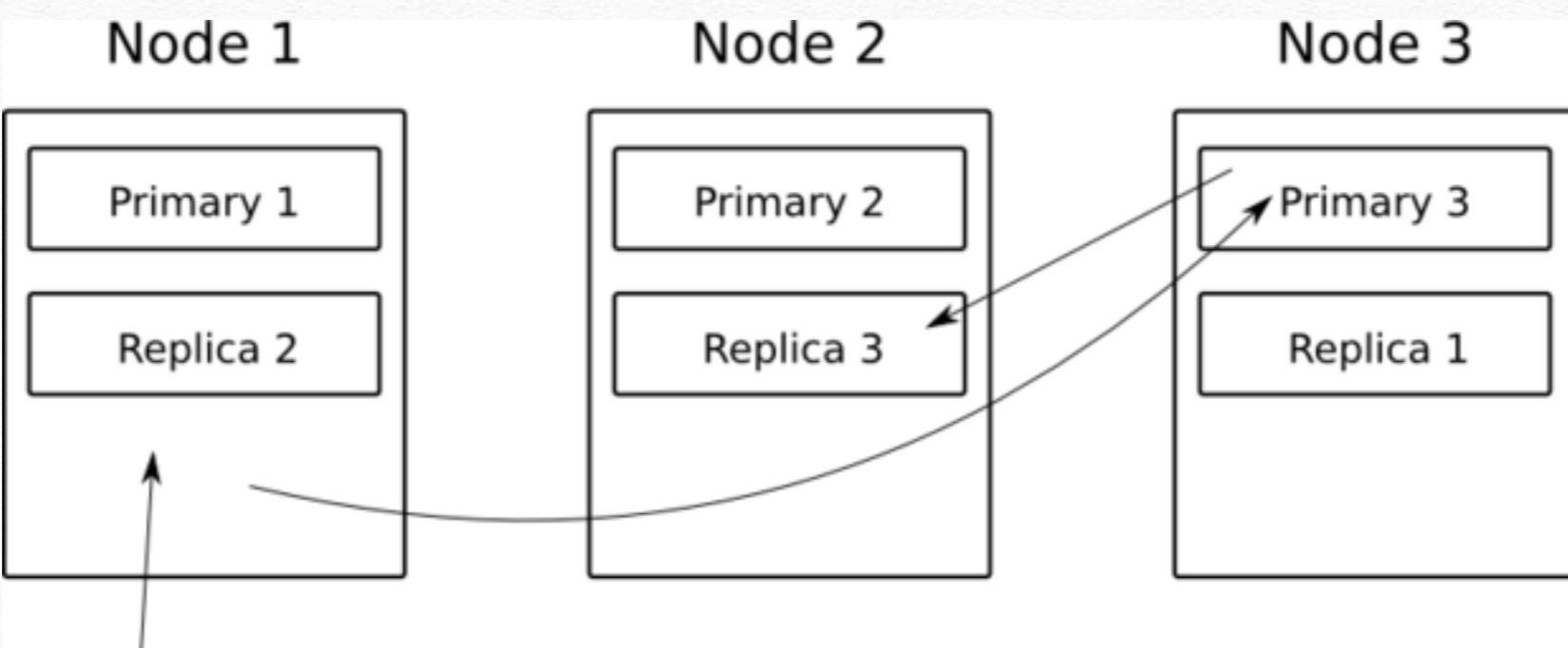


aka [pirate => robot]

Snowball



Production



<http://mobz.github.io/elasticsearch-head/>

ElasticSearch http://192.168.7.8:9200/ Connect Rick cluster health: yellow (6, 18)

Overview Browser Structured Query Any Request Info Status Nodes Stats Cluster Nodes Cluster State Cluster Health

Cluster Overview New Index

	cu_docs	bnil	cu_msg	anvil
Leon 3Wqr1xaCRu-b0uEzDkmrDg inet[/192.168.7.8:9202]	size: 180Gb (540Gb) docs: 995131 (995131) Info Actions	size: 80kb (480kb) docs: 90 (90) Info Actions	size: 313Gb (1.56Tb) docs: 10047450 (10140915) Info Actions	index: close Info Actions
Pris L8qx7ilfSI-kcKq_6bMbWw inet[/192.168.7.8:9204]	0 1	0 1	0	
Rick Vnptra1FNTGirwRfZsZ2RxQ inet[/192.168.7.8:9200]	1 2	0 1	0 1 2 3 4	
Rachel 87KsIv0FTVSkkqwENaja6A inet[/192.168.7.8:9203]	1 2	0 1	0 1 2 3	
Zhora b6NxRTxsR_WUQl5cXPKHbw inet[/192.168.7.8:9205]	0 2	0 1	0 1 2 3 4	
Roy _8RI2wYVT7Svn_v5F97jJA inet[/192.168.7.8:9201]	0 2	0 1		
Unassigned	0 0	0 0	0 1 2 3 4	
	1 1			

A context menu is open over the "Actions" button for the "Rick" node, showing options: Refresh, Flush, Gateway Snapshot, Test Analyser, Close, and Delete... The "Actions" button for the "Rick" node is highlighted in blue.

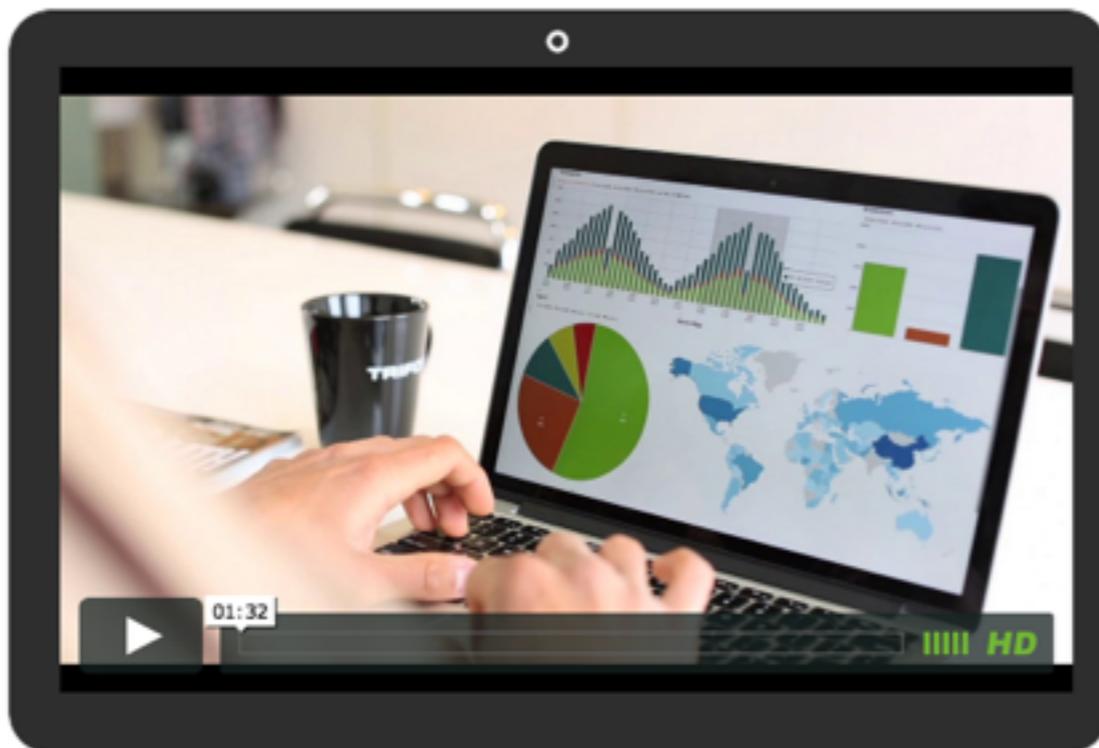
A modal dialog box is open over the "Actions" button for the "Rick" node, displaying the node's configuration:

```
{  
  name: "Leon",  
  transport_address: "inet[/192.168.7.8:9302]",  
  attributes: [ ],  
  http_address: "inet[/192.168.7.8:9202]",  
  os: {  
    refresh_interval: 5000,  
    cpu: {  
      vendor: "Intel",  
      model: "Macmini4,1",  
      mhz: 2400,  
      total_cores: 2,  
      total_sockets: 1,  
      cores_per_socket: 2,  
      cache_size: "3kb",  
      cache_size_in_bytes: 3072  
    }  
  }  
}
```

Kibana

visualize logs and time-stamped data

elasticsearch works seamlessly with kibana to let you see and interact with your data



No code required

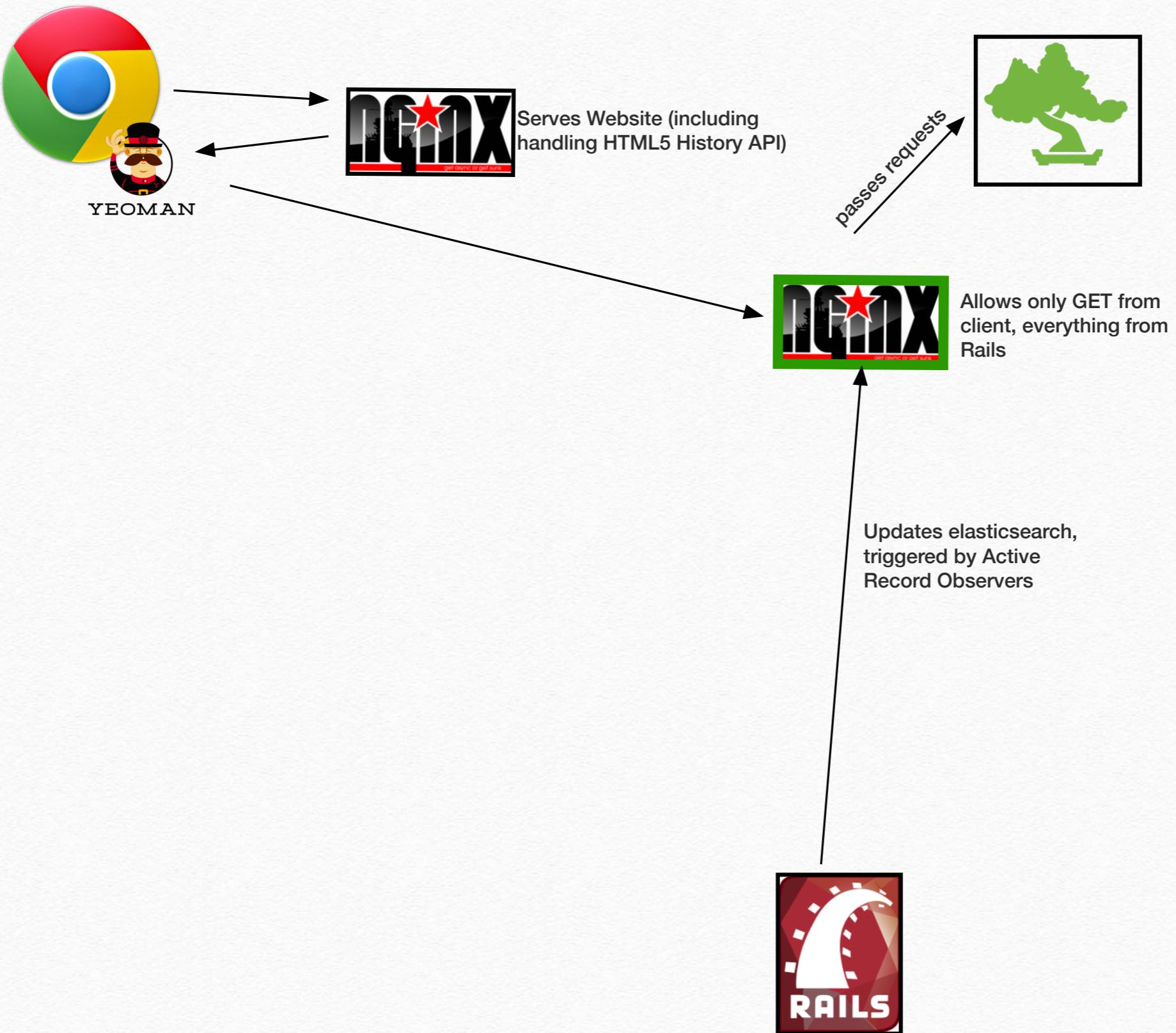
Real-time analysis of streaming data

Highly scalable

Open source, community driven

Seamless integration with [Logstash](#), [Apache Flume](#),
[Fluentd](#) and others

The Future? (I'd love this)



<https://github.com/shicholas/elasticsearch-presentation>

Questions?