run your own search engine.
today:

Cablecar

Robert Kowalski

@robinson_k
http://github.com/robertkowalski

Search

nobody uses that, right?

Services on the Market

Google
Bing
Yahoo
ask
Wolfram Alpha
Baidu (cn)
Yandex (ru)

Why not use the ones that exist?

What if they - or their countries - don't like our data?

What if we want to see other things than the average customer?

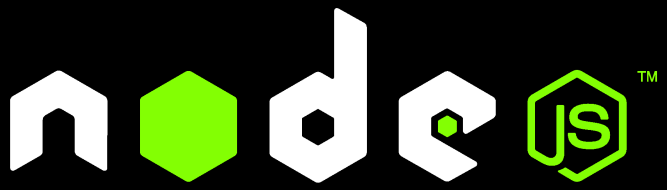
What if we don't want our main search engine to save our queries?

Why not use the ones that exist?

Or what if I want to create my own search engine - just for me, with my data,
at home...

OpenSource here I come!

Building an own system for searches



node.js Service

separate service providing an UI for queries

Frontend-Service

elasticsearch Service

does the heavy lifting (indexing text)

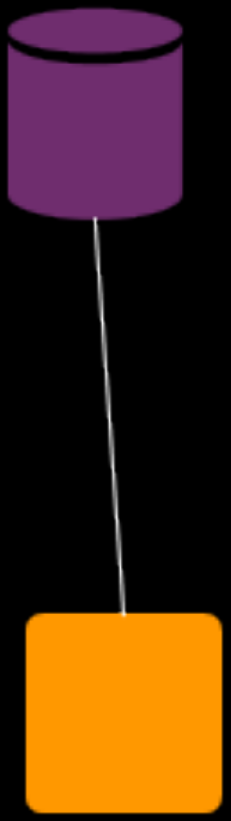
providing one API for different services

Backend-Service

Bash, Perl & curl

Getting data into elasticsearch

Current Status

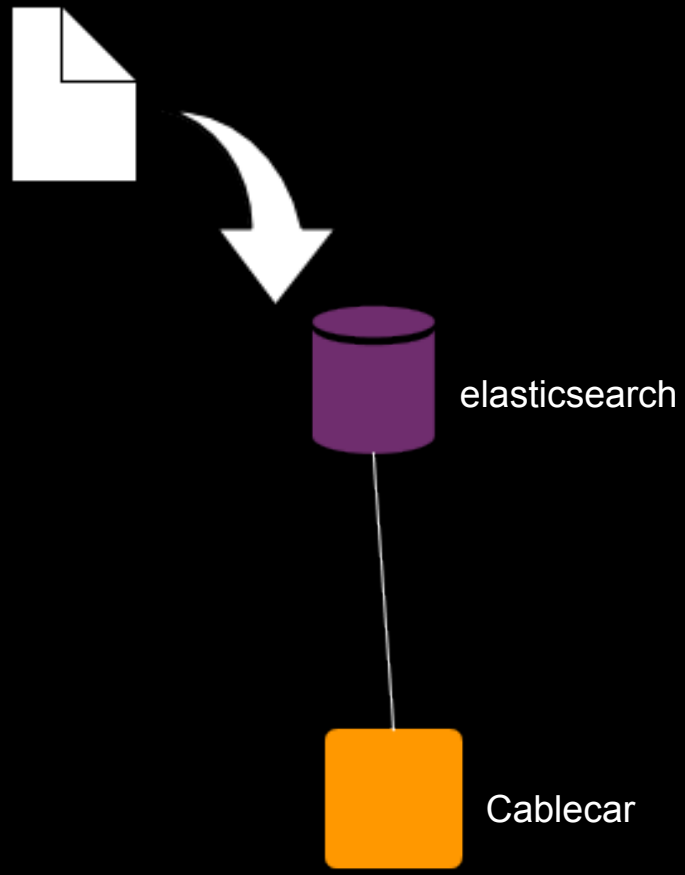


Backend (elasticsearch)

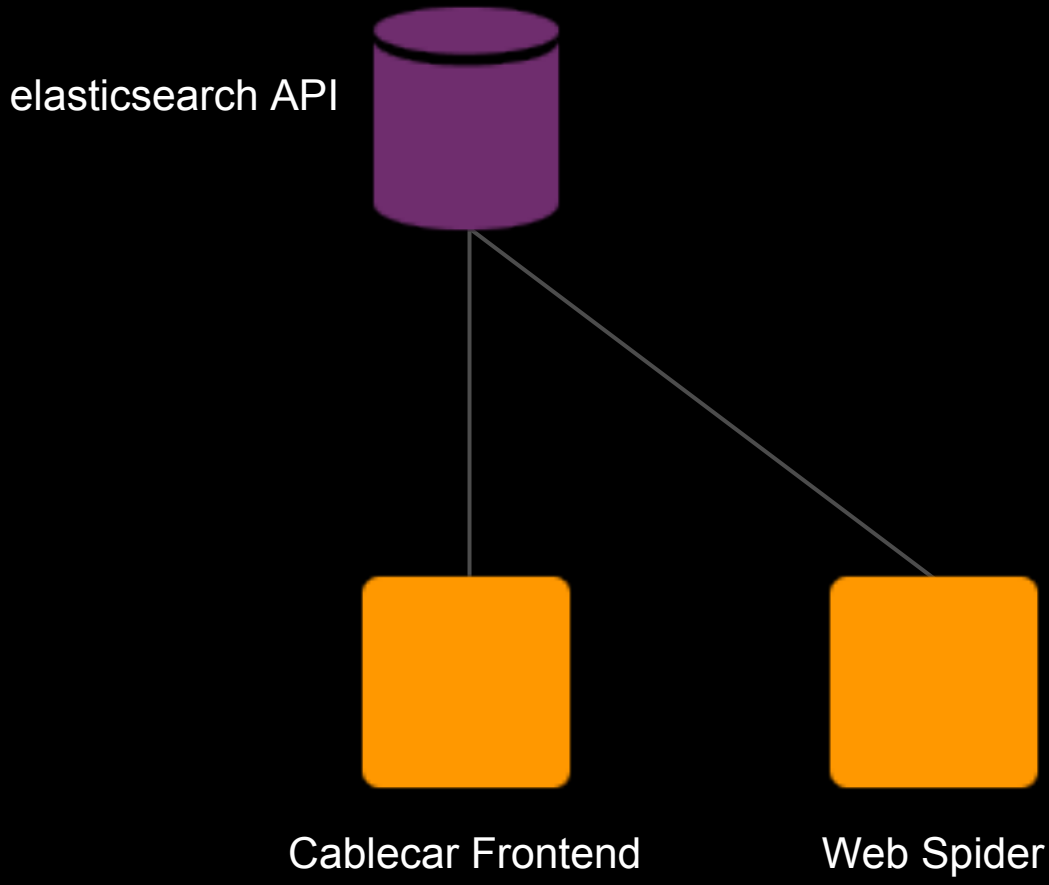
node.js Frontend for searches
(Cablecar)

Current Status

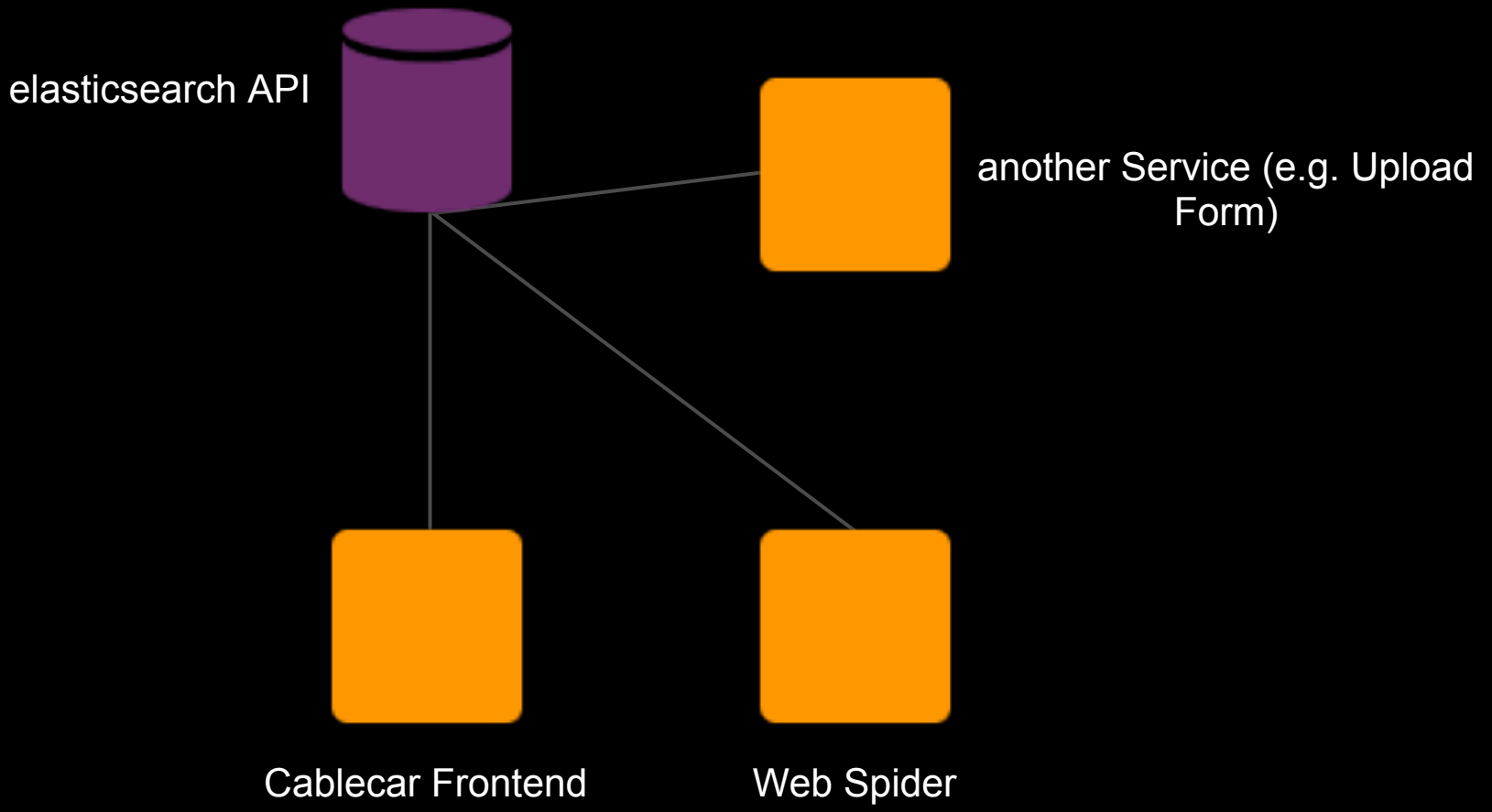
#!/bin/sh



Extensions



More Extensions!



The parts in detail

Cablecar (node.js Frontend-Service)

elasticsearch (Backend-Service)

Shellscript

node.js

JavaScript on the Server

Event driven architecture

Uses V8 Engine

Is fun!

node.js

"C10k problem"

no Threads - a lot of concurrent requests possible

"non-blocking I/O" model

elasticsearch

Search Engine based on Apache Lucene

adds: REST API, easy Clustering...

related Projects: Apache Solr

Can I haz Cluster?

Tutorial Time!

Creating a mapping for the files

```
curl -X PUT "localhost:9200/books/attachment/_mapping" -d '{
  "attachment": {
    "properties": {
      "file": {
        "type": "attachment",
        "fields": {
          "title": {"store": "yes"},
          "download": {"store": "yes"},
          "filename": {"store": "yes"},
          "file": {"term_vector": "with_positions_offsets", "store":"
yes"}
        }
      }
    }
  }
}'
```

Throw a file into the index

"file" -> base64 encode ->
HTTP POST -> elasticsearch

#/bin/sh - Indexing one file

```
#!/bin/sh

file=RedisManual.pdf

encoded=`cat $file | perl -MMIME::Base64 -ne  'print encode_base64($_)'`

json="{\"file\": \"${encoded}\", \"filename\": \"${file}\"}"

echo "$json" > json.file

curl -X POST "localhost:9200/books/attachment"  -d @json.file
```

Searching

REST- API

Query DSL in JSON

Example - Search

```
curl -X POST http://127.0.0.1:9200/_search?pretty=true \
-d '{"fields": ["title", "filename", "download"], "query" { "query_string":
{ "query": "Data" }}, "highlight": {"fields": {"file": {}}}}'
```

Example - Result


```
{ "took" : 112,
  "timed_out" : false,
  "_shards" : {
    "total" : 11,
    "successful" : 11,
    "failed" : 0
  },
  "hits" : {
    "total" : 11,
    "max_score" : 0.062009797,
    "hits" : [ {
      "_index" : "books",
      "_type" : "attachment",
      "_id" : "HbY-6si5QxCA53DqCreZow",
      "_score" : 0.062009797,
      "fields" : {
        "filename" : "RedisManual.pdf",
        "download" : "http://myserver.com/download/pdf/"
      }],
     "highlight" : {
        "file" : [ "Environment for <em>Data</em> Analysis and Graphics\n\nVersion 2.15.1 (2012-06-22)\n\nW. N. Venables, D. M. Smith\nand the R", "6\n1.11 <em>Data</em> permanency and removing objects . . 6\n\n2 Simple", "subsets of a <em>data</em> [...]" ]
     }

   }}}
```

Current Status

Limited - index just the local filesystem

besides this: UP AND RUNNING

The Future of the Project: Extensions

Feed the backend with a webspider

As a separate service

no auto-indexing, index on request

follow robot.txt? Inverse? not at all?

can handle username / password combination

Next Services

Could be:

Python
Ruby
node.js
PHP
Java
...

Github

https://github.com/hamburg-honeybadgers/cablecar