Elasticsearch

Charles Severance



History

- Emerged from a desire to make an "open source" search engine
  - Scalable – firehose, data size, parallel search
  - Inverted index – full text
  - Ranking / relevance
  - Recommendation engine
- Built on top of Apache Lucene
  - A "Google" of your own
- Has evolved into NoSQL Applications



License – Open Core

- The essential parts are free under an Apache license
- The Elastic" company supports open source and sells hosting / consulting / extras but you can use this without paying Elastic

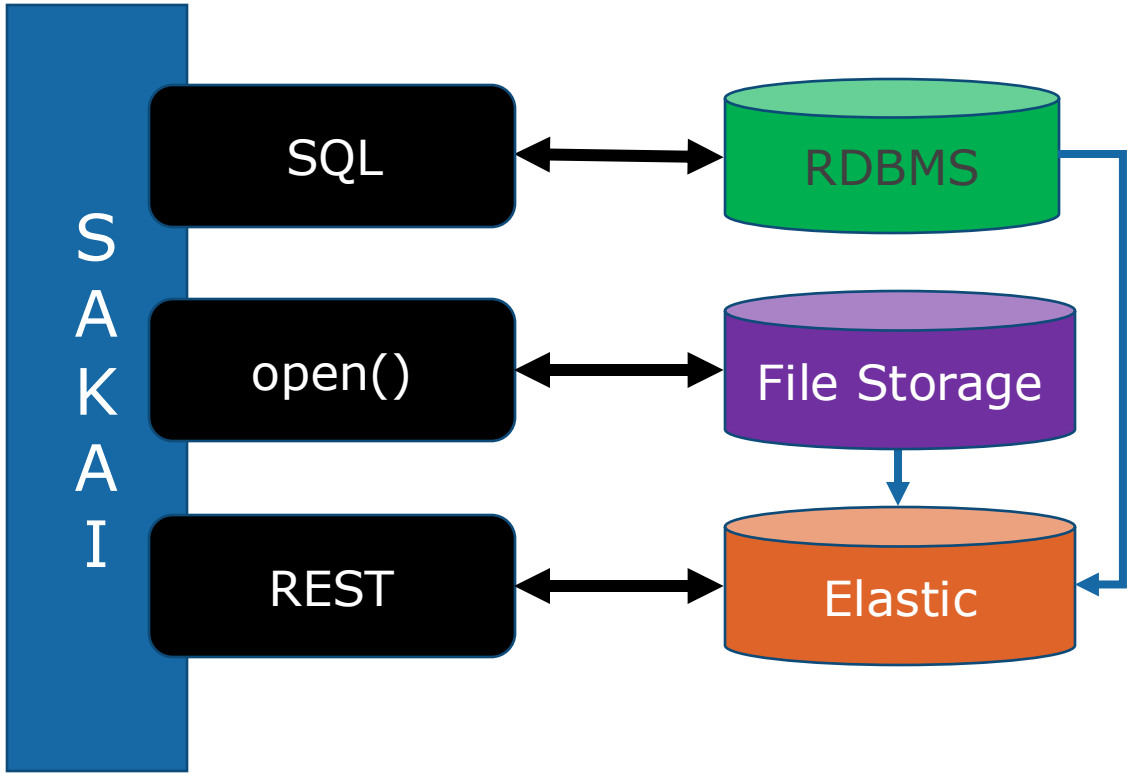
https://github.com/elastic/elasticsearch/blob/master/LICENSE.txt
https://www.elastic.co/products/elasticsearch
https://en.wikipedia.org/wiki/Open-core_model



Application: Sakai

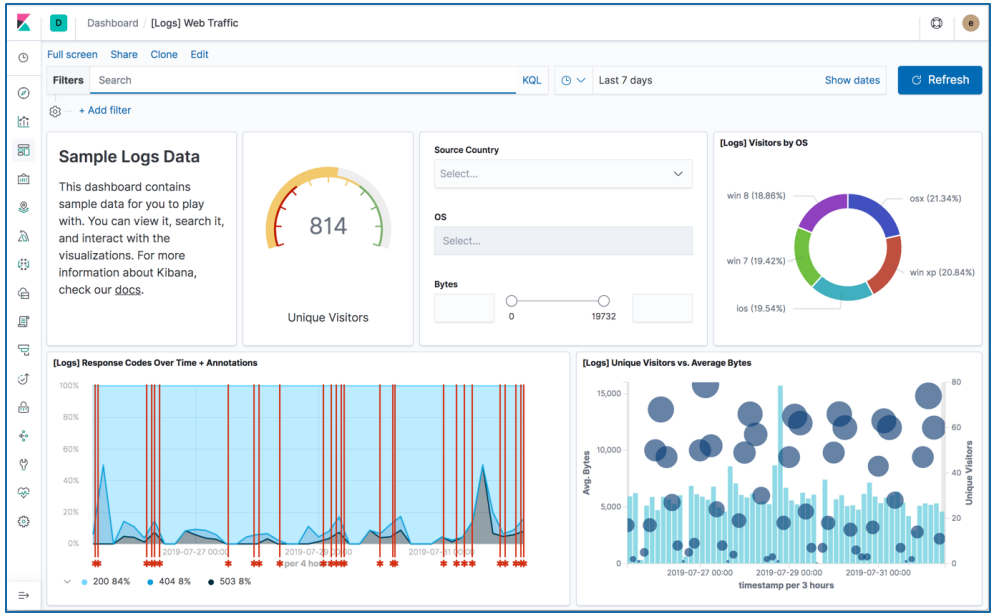


- Open Source Learning Management System



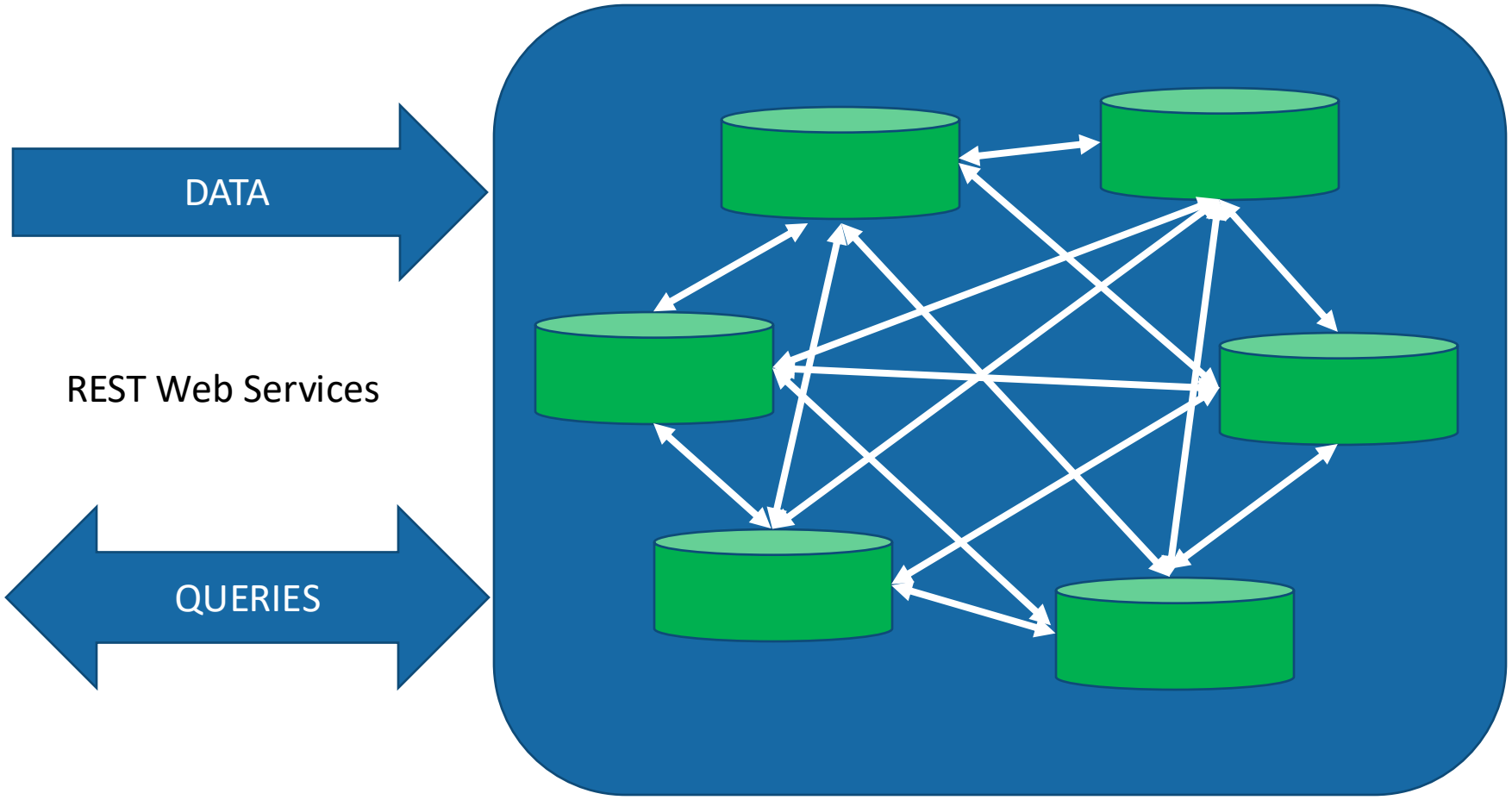
Application: ELK Stack

- Elasticsearch – distributed NoSQL database
- Logstash – ingests streams of activity data
- Kibana – Visualization / Dashboards

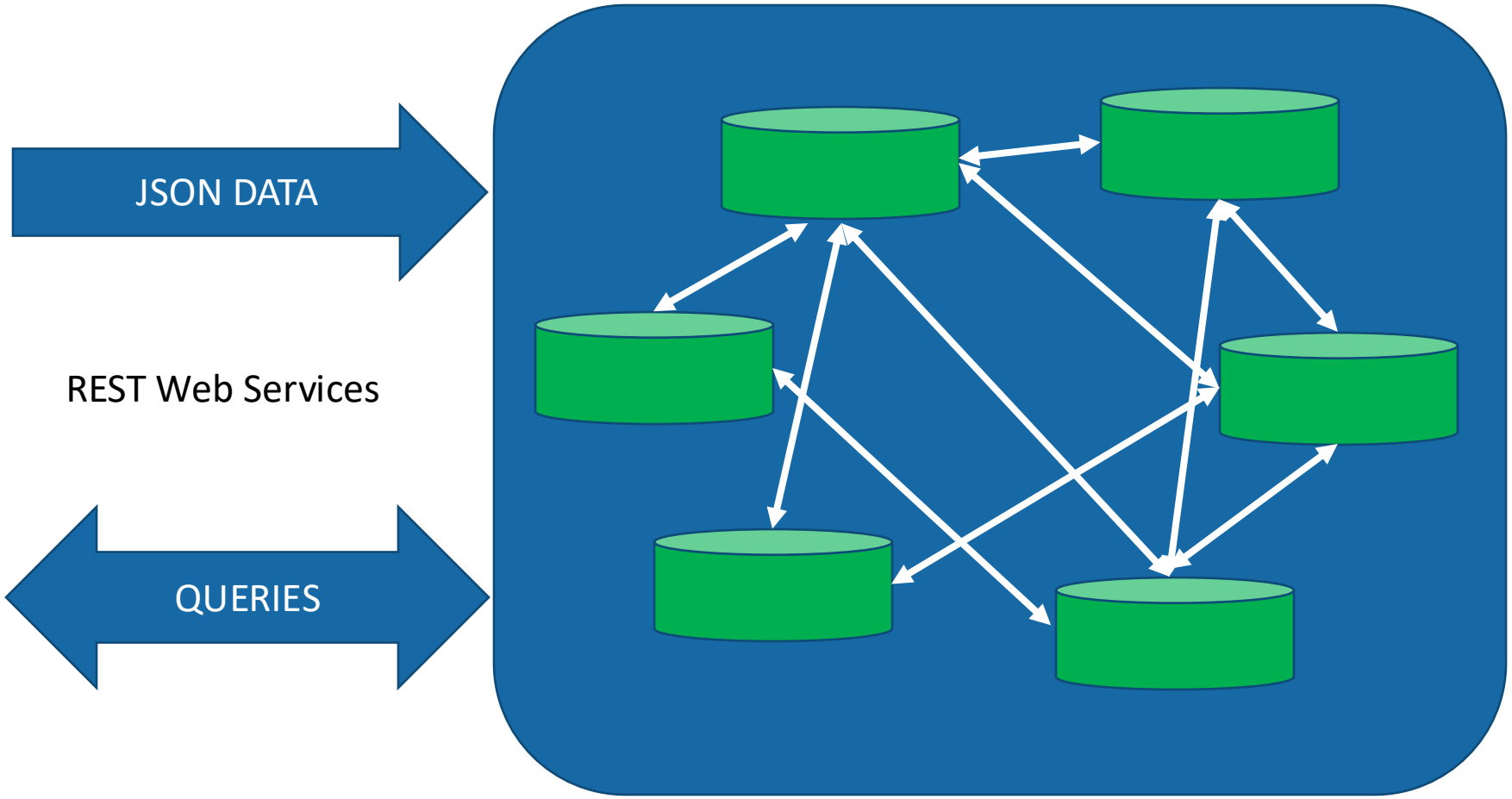


https://www.elastic.co/guide/en/kibana/current/dashboard.html

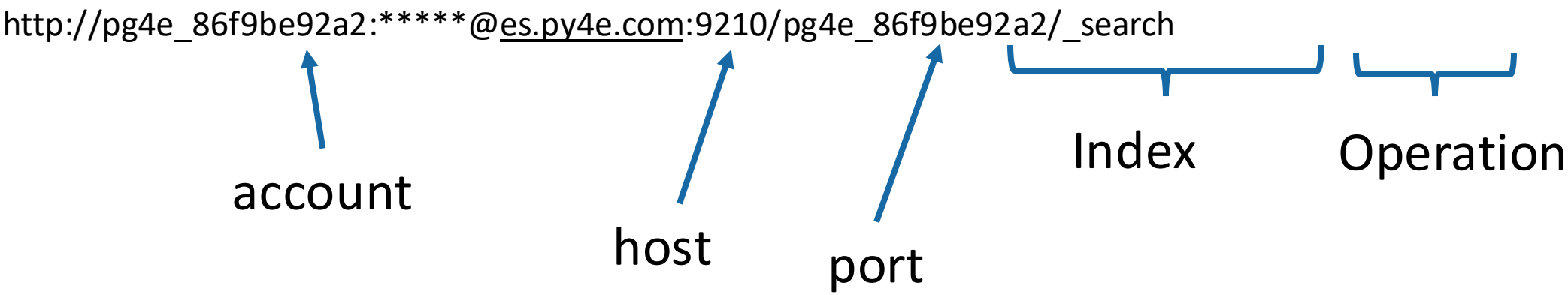
Architecture



Architecture – Eventual Consistency



Document Storage URLs



Programming Elasticsearch



Reading docs - REST APIs

Match all query

The most simple query, which matches all documents, giving them all a _score of 1.0.

```
GET /_search
{
    "query": {
        "match_all": {}
    }
}
```

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-all-query.html



In Python...

```
queryurl = http://pg4e_86f9:*@es.py4e.com:9210/prefx/testindex/_search?pretty'

body=json.dumps( {"query": {"match_all": {}}} )
hdict = {'Content-type': 'application/json; charset=UTF-8'}

response = requests.post(queryurl, headers=hdict, data=body)

text = response.text
status = response.status_code
js = json.loads(text)
```

https://www.pg4e.com/code/elastictool.py



Python Elasticsearch Library

```
# pip3 install elasticsearch

from elasticsearch import Elasticsearch

es = Elasticsearch(
 [secrets['host']],port=secrets['port'],url_prefix=secrets['prefix'],
 http_auth=(secrets['user'],secrets['pass']),scheme="http",
 )

res = es.search(index="testindex", body={"query": {"match_all": {}}})
print(res)
```

https://elasticsearch-py.readthedocs.io/en/master/
https://www.pg4e.com/code/elastictweet.py



Summary

- Elasticsearch gives us Google-like features
  - Scalable ingest / data size / search performance
  - Accessible through a "REST API"
- Can be used as a full-text "search engine"
- Can be used as a scalable NoSQL database



Acknowledgements / Contributions

These slides are Copyright 2019- Charles R. Severance (www.dr-chuck.com) as part of www.pg4e.com and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles R. Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here

