The ultimate guide for Elasticsearch plugins

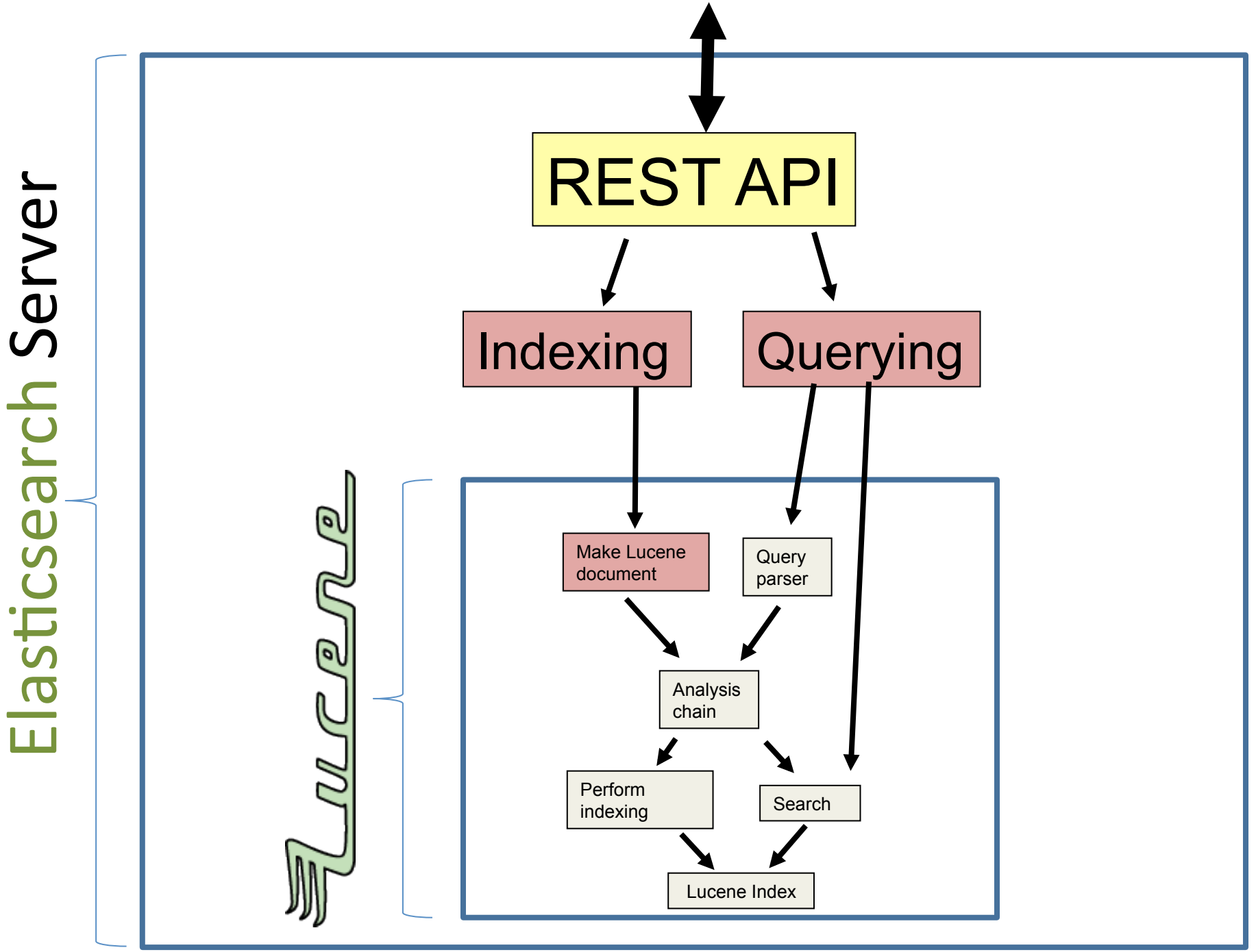
Itamar Syn-Hershko

http://code972.com

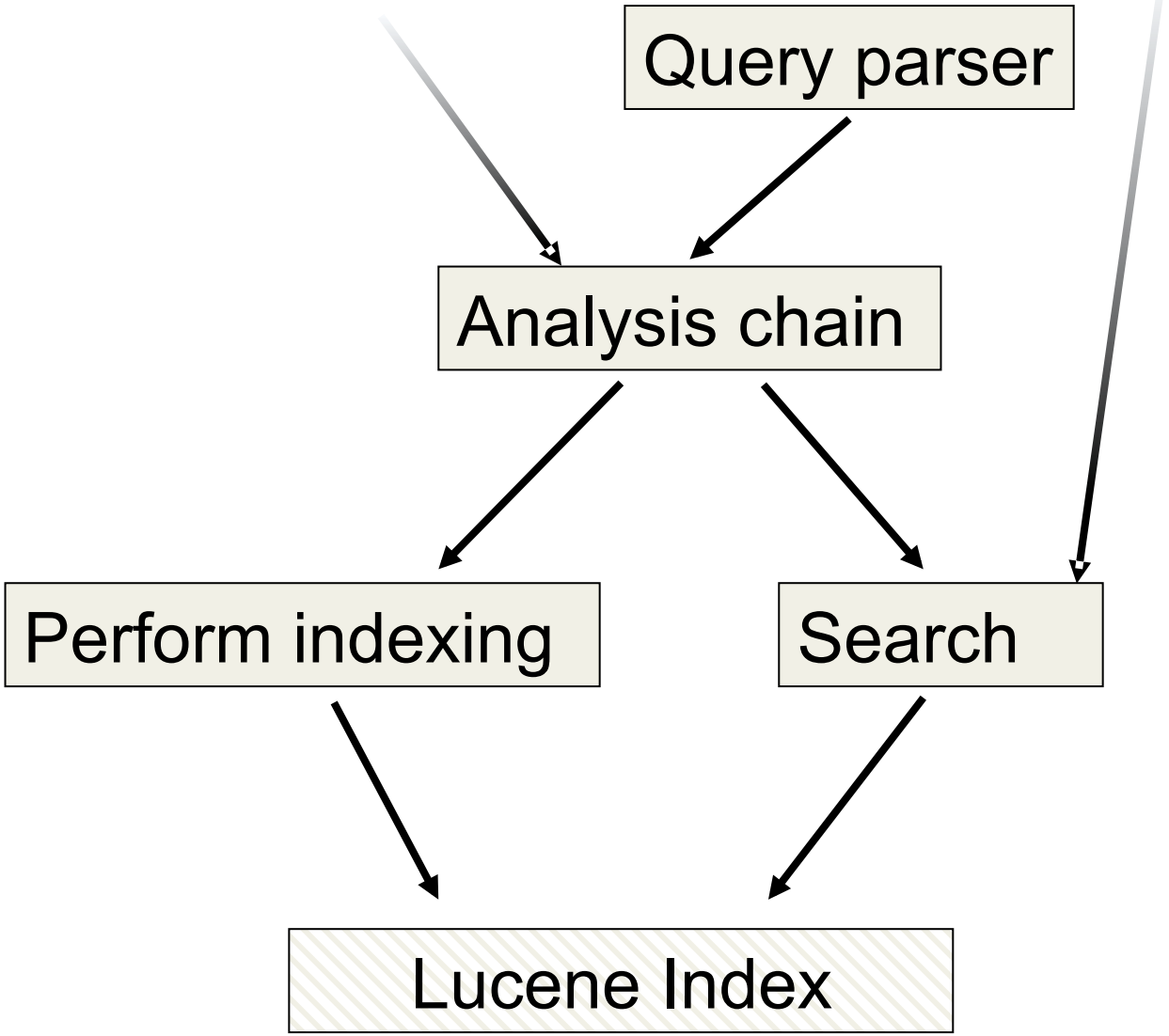
@synhershko

Agenda

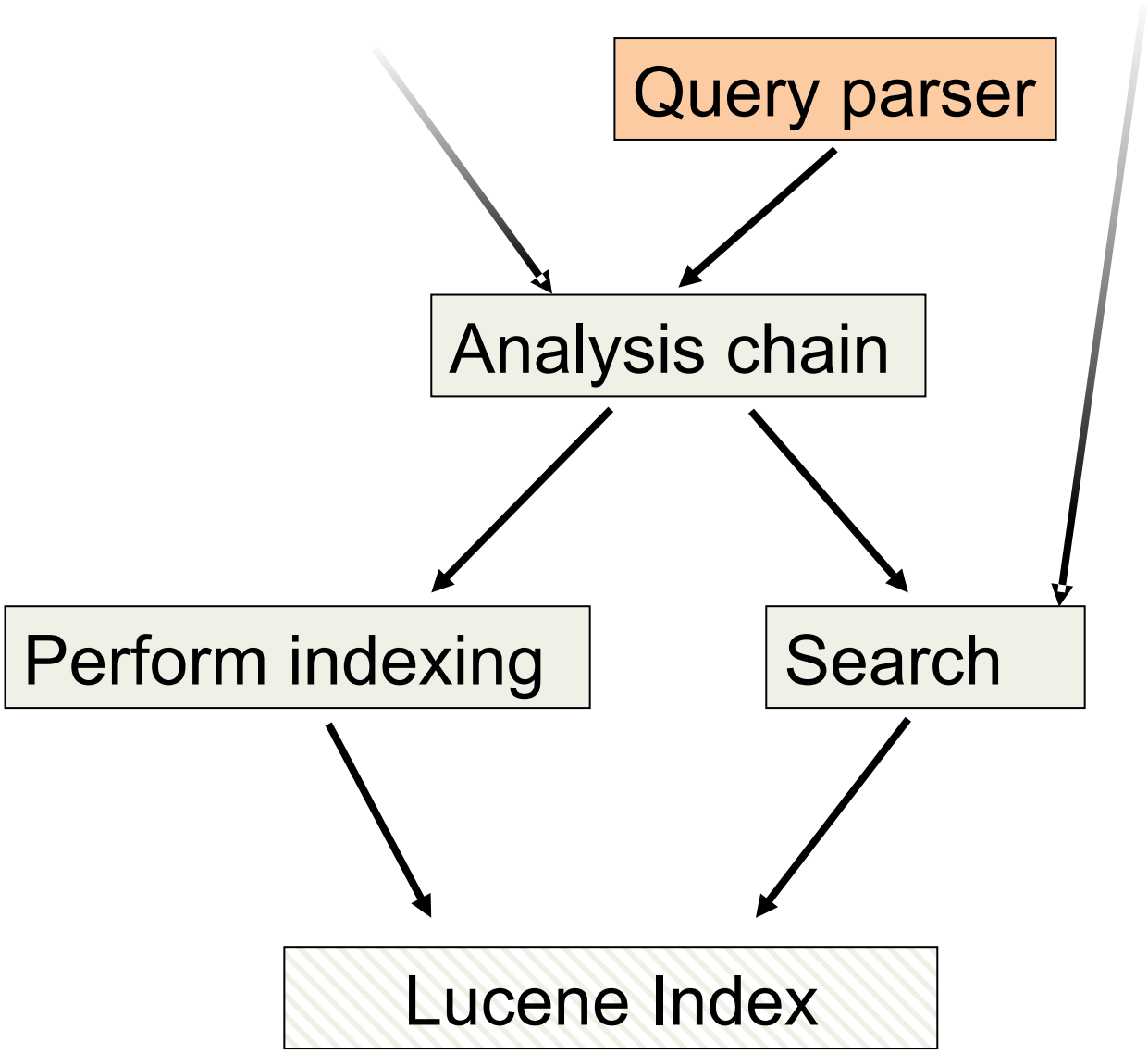
- Integration points & plugin types
- Showcases
- Gotchas
- When-to, How-to
- Q & A


Elasticsearch Server
REST API
Indexing
Querying
Lucene
Make Lucene document
Query parser
Analysis chain
Perform indexing
Search
Lucene Index

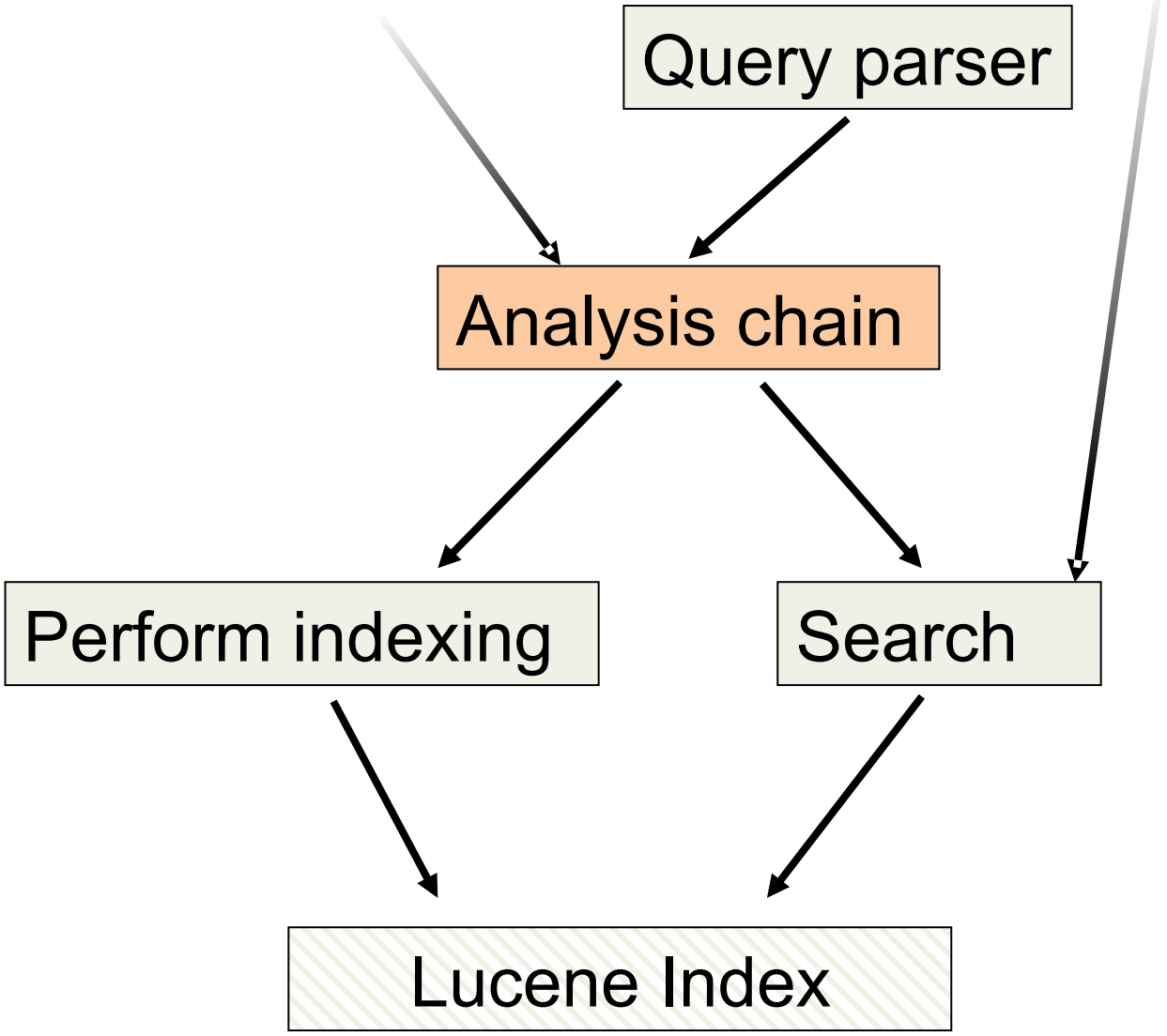
Lucene extension points

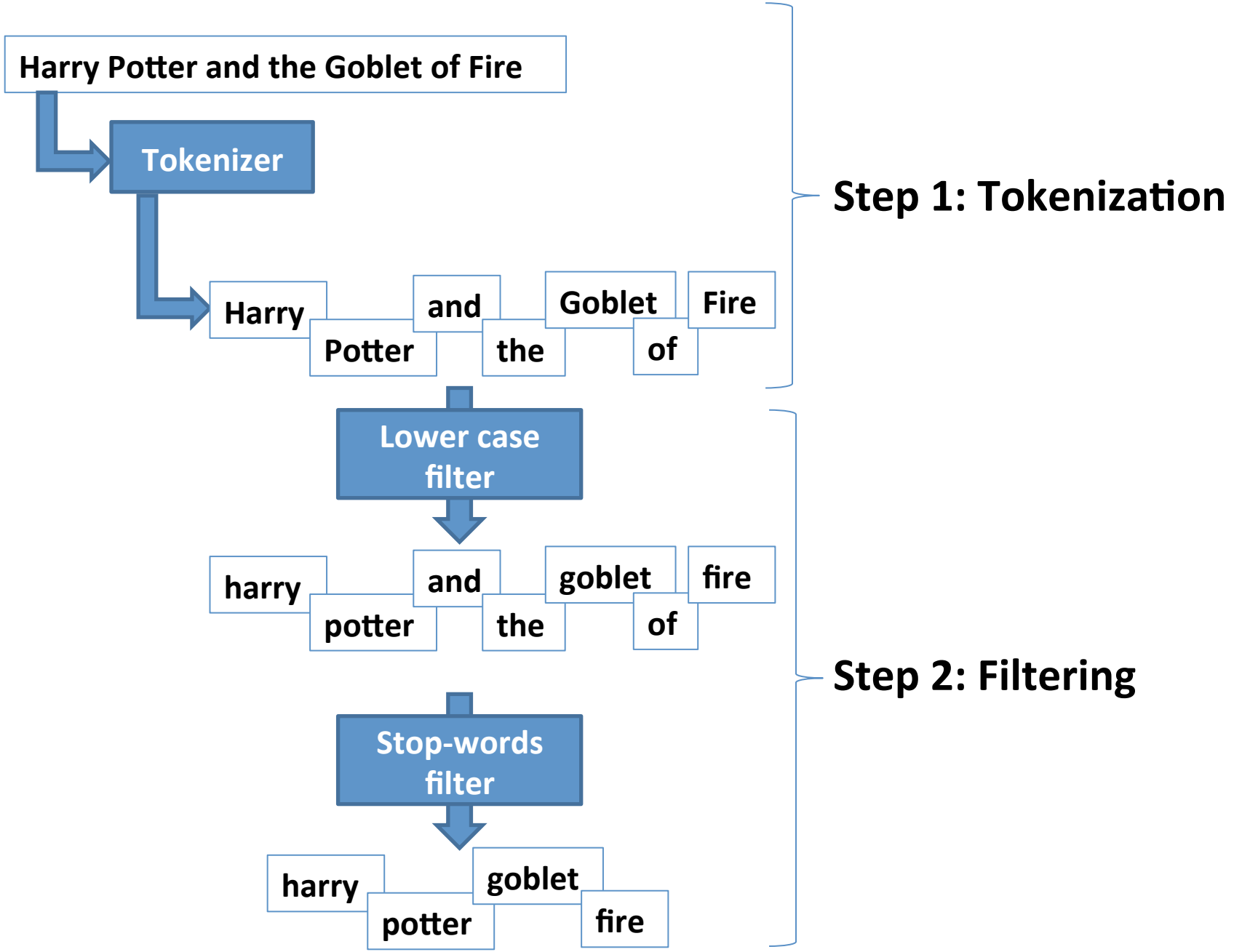


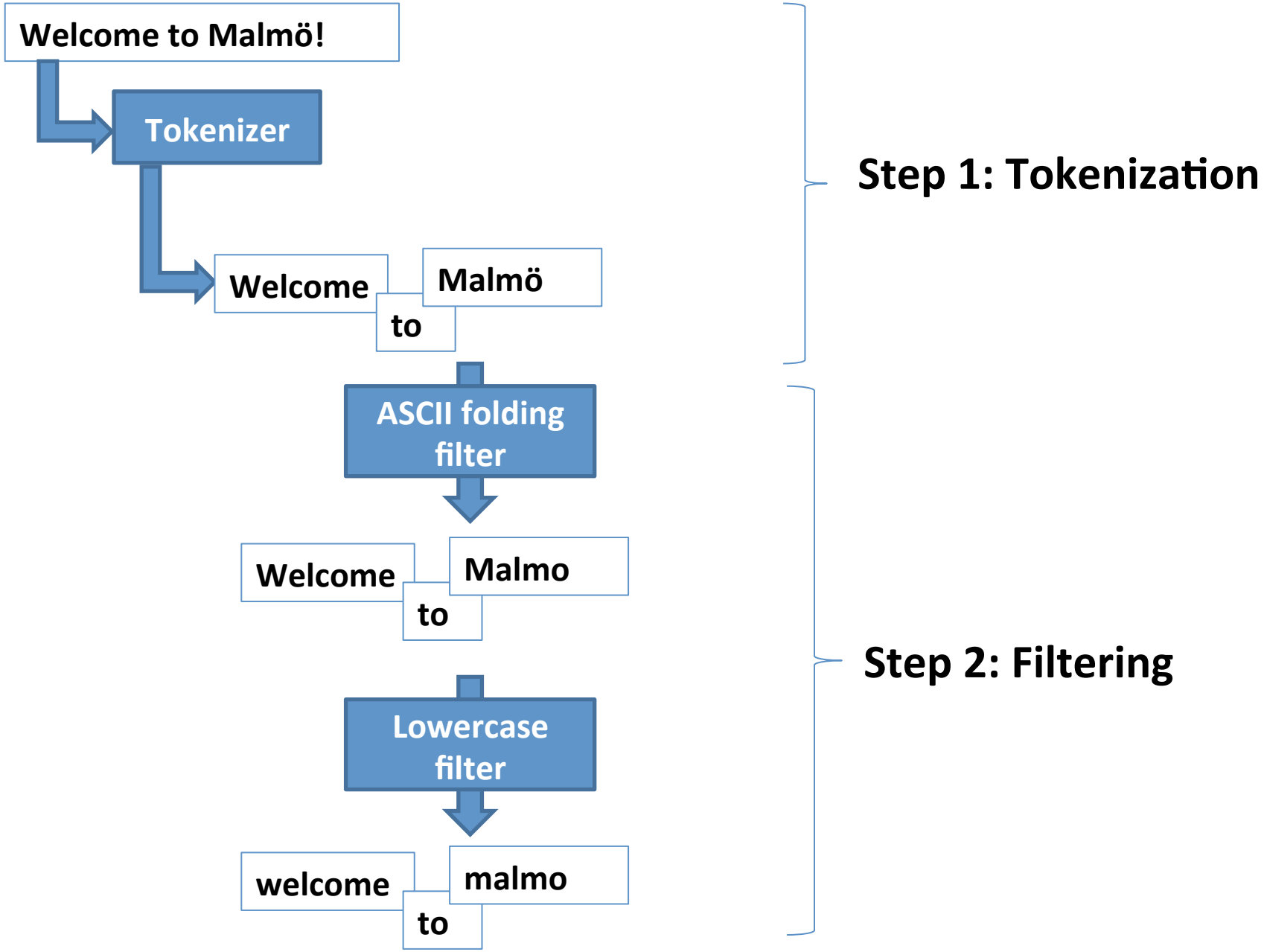
Lucene extension points

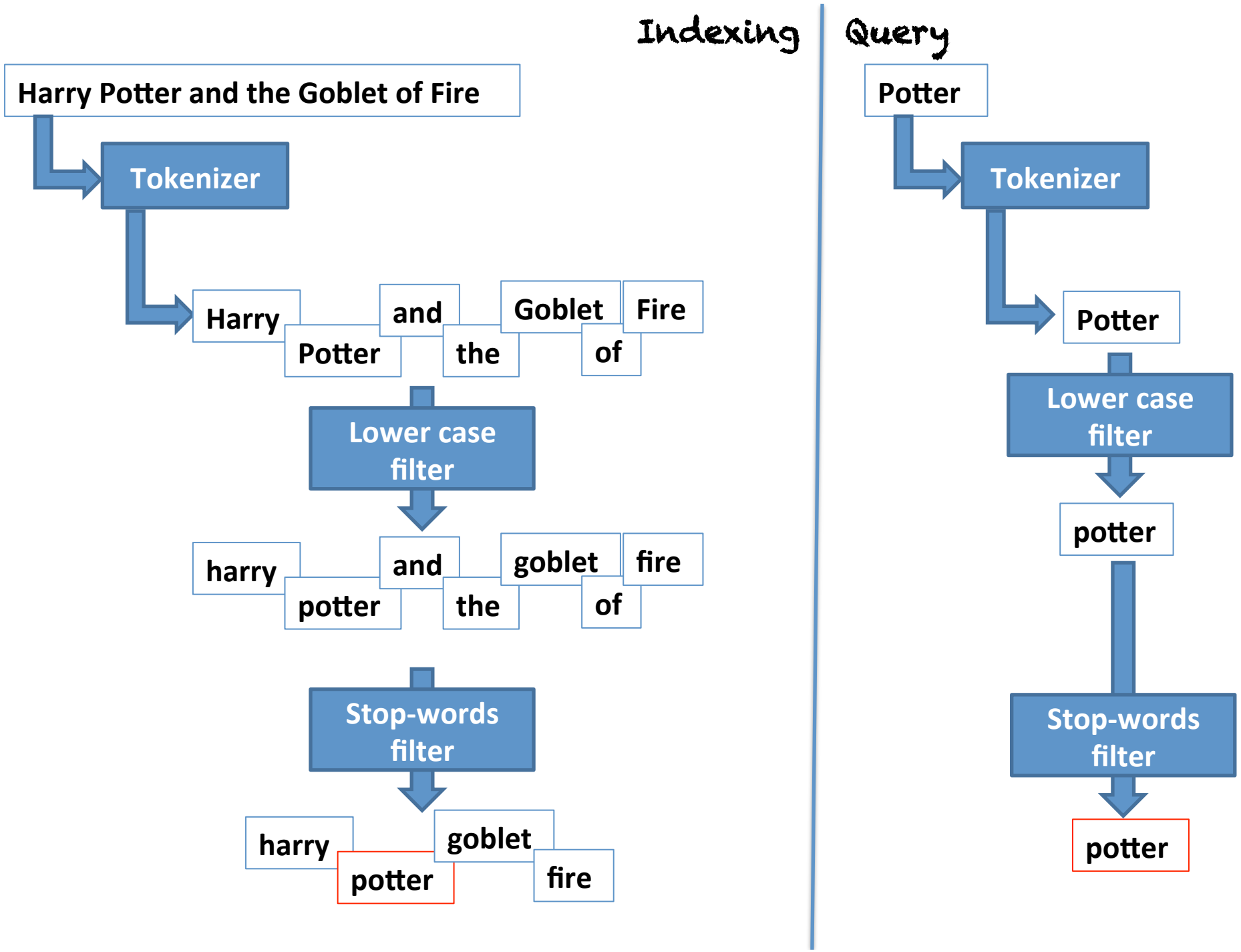


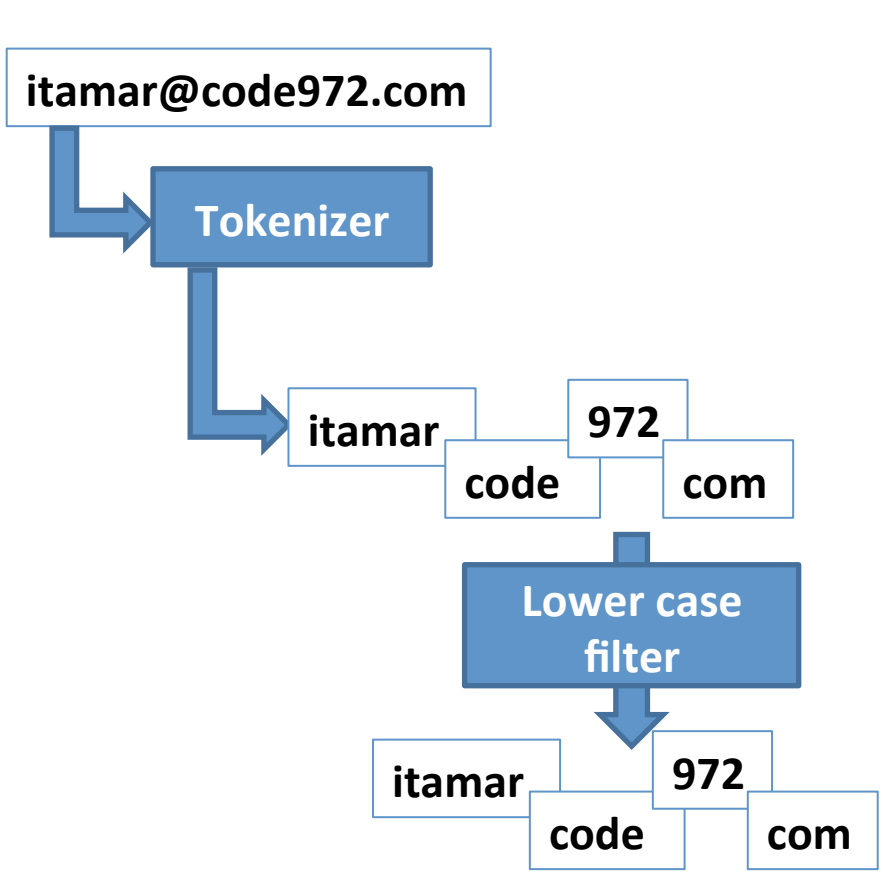
Lucene extension points




Harry Potter and the Goblet of Fire
Tokenizer
Harry
Potter
and
the
Goblet
of
Fire
Step 1: Tokenization
Lower case filter
harry
potter
and
the
goblet
of
fire
Stop-words filter
harry
potter
goblet
fire
Step 2: Filtering


Welcome to Malmö!
Tokenizer
Welcome
to
Malmö
ASCII folding filter
Welcome
to
Malmo
Lowercase filter
welcome
to
malmo
Step 1: Tokenization
Step 2: Filtering

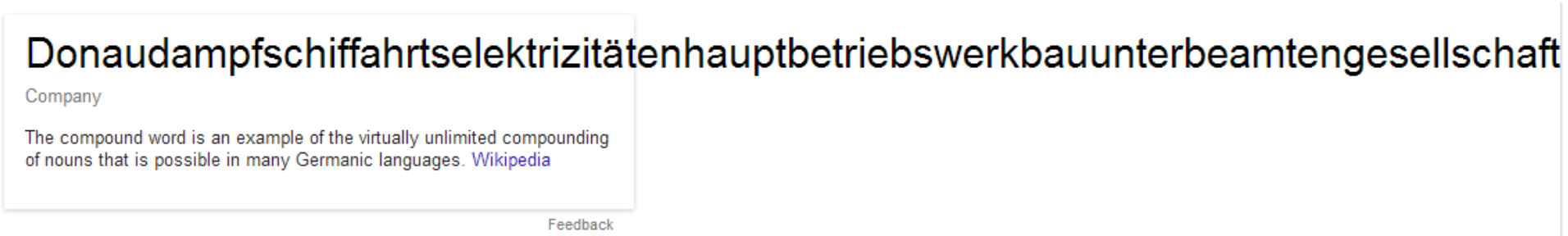

Indexing
Query
Harry Potter and the Goblet of Fire
Tokenizer
Harry
Potter
and
the
Goblet
of
Fire
Lower case filter
harry
potter
and
the
goblet
of
fire
Stop-words filter
harry
potter
goblet
fire
Potter
Tokenizer
Potter
Lower case filter
potter
Stop-words filter
potter

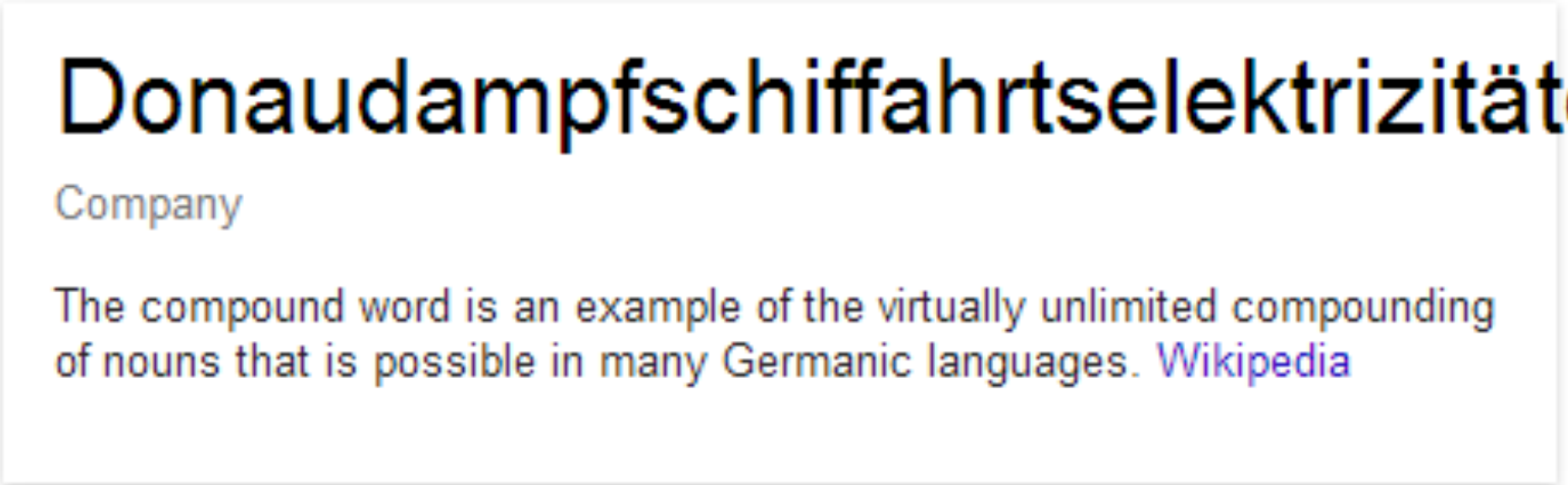


Step 1: Tokenization

Step 2: Filtering

Try searching on German compound words…





Analyzers

The quick brown fox jumped over the lazy dog, bob@hotmail.com 123432.

StandardAnalyzer:

[quick] [brown] [fox] [jumped] [over] [lazy] [dog] [bob@hotmail.com] [123432]

StopAnalyzer:

[quick] [brown] [fox] [jumped] [over] [lazy] [dog] [bob] [hotmail] [com]

SimpleAnalyzer:

[the] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dog] [bob] [hotmail] [com]

WhitespaceAnalyzer:

[The] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dog,] [bob@hotmail.com] [123432.]

KeywordAnalyzer:

[The quick brown fox jumped over the lazy dog, bob@hotmail.com 123432.]

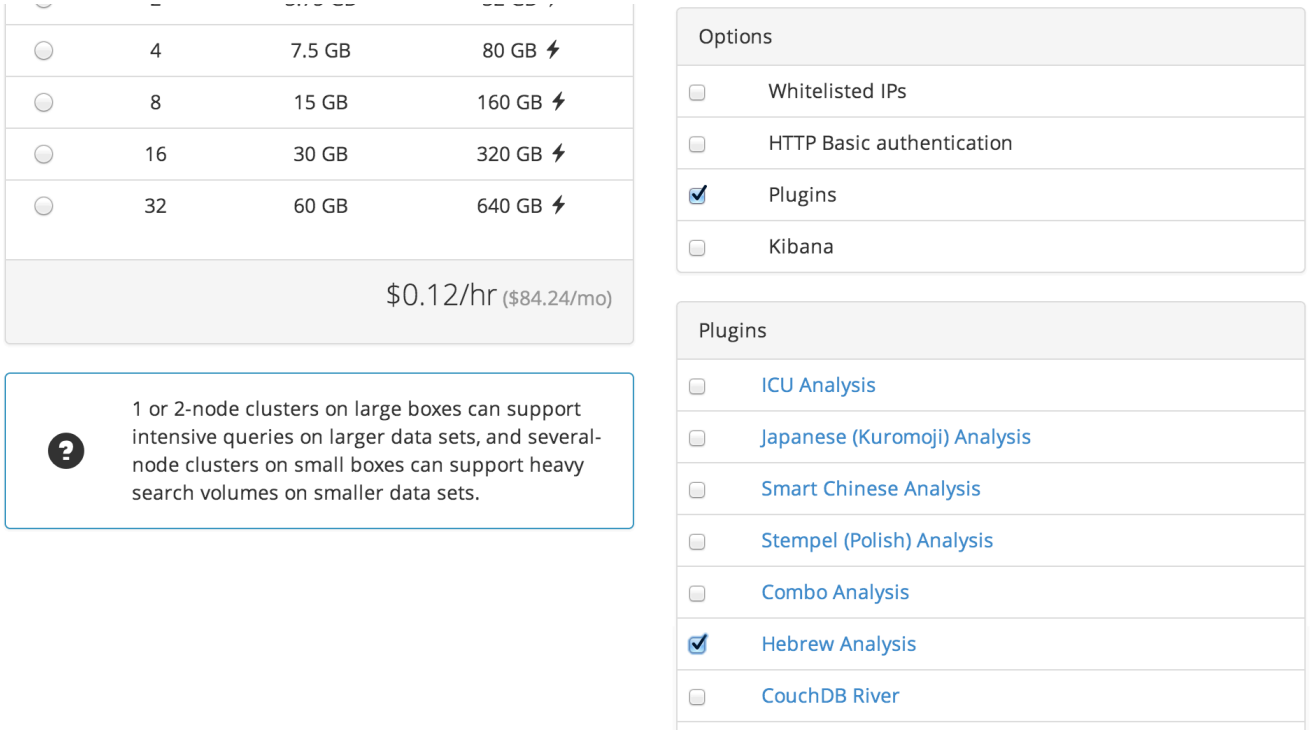
Custom analyzers from code

```
"index" : {
    "analysis" : {
        "analyzer" : {
            "default" : {
                "tokenizer" : "standard",
                "filter" : ["standard", "my_ascii_folding"]
            }
        },
        "filter" : {
            "my_ascii_folding" : {
                "type" : "asciifolding",
                "preserve_original" : true
            }
        }
    }
}
```

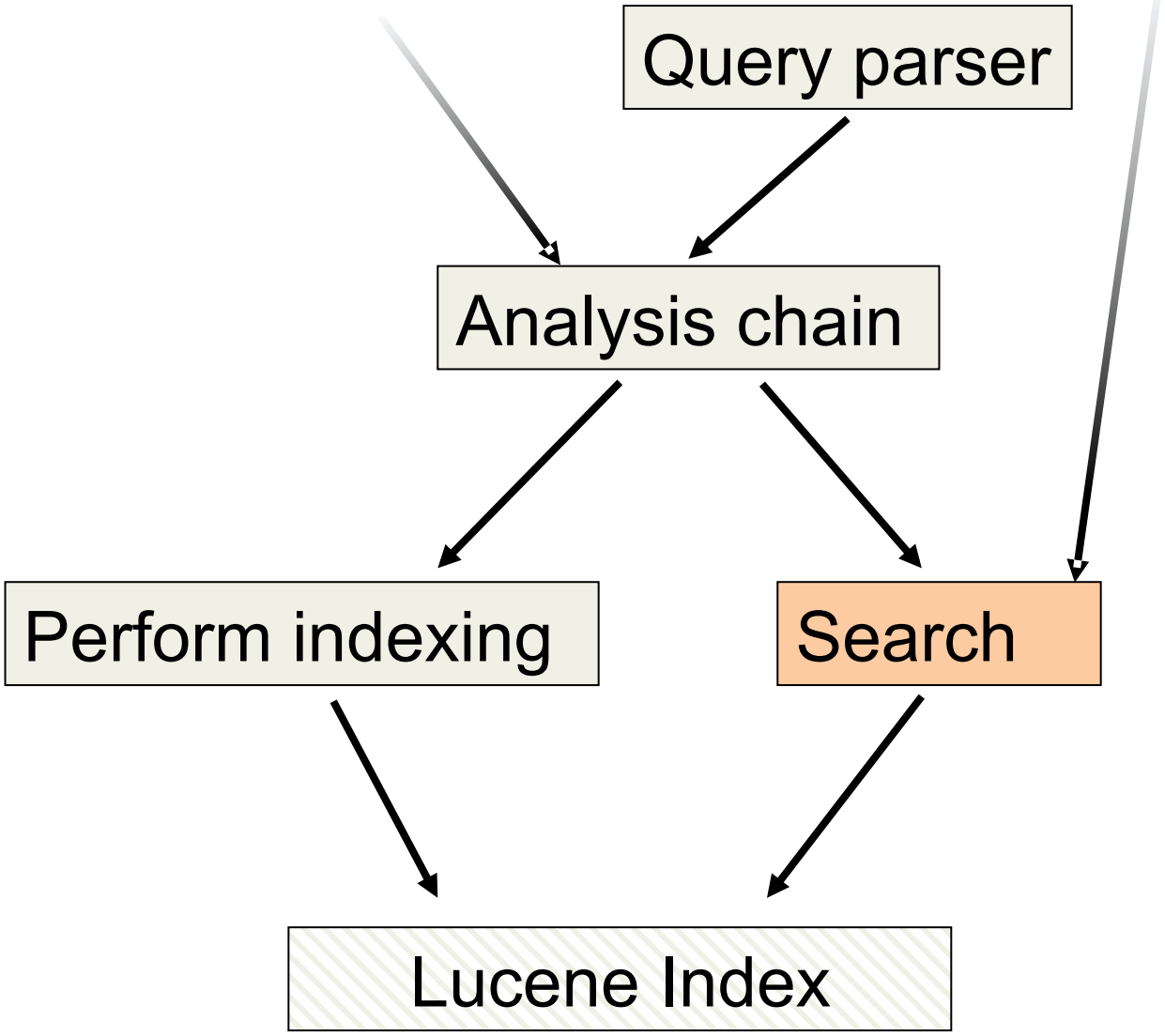
New in Elasticsearch v1.1.0

Showcase: Custom Analyzer - Hebrew analysis plugin for Elasticsearch

- https://github.com/synhershko/elasticsearch-analysis-hebrew
- Available on QBox.io



Lucene extension points



Scripting

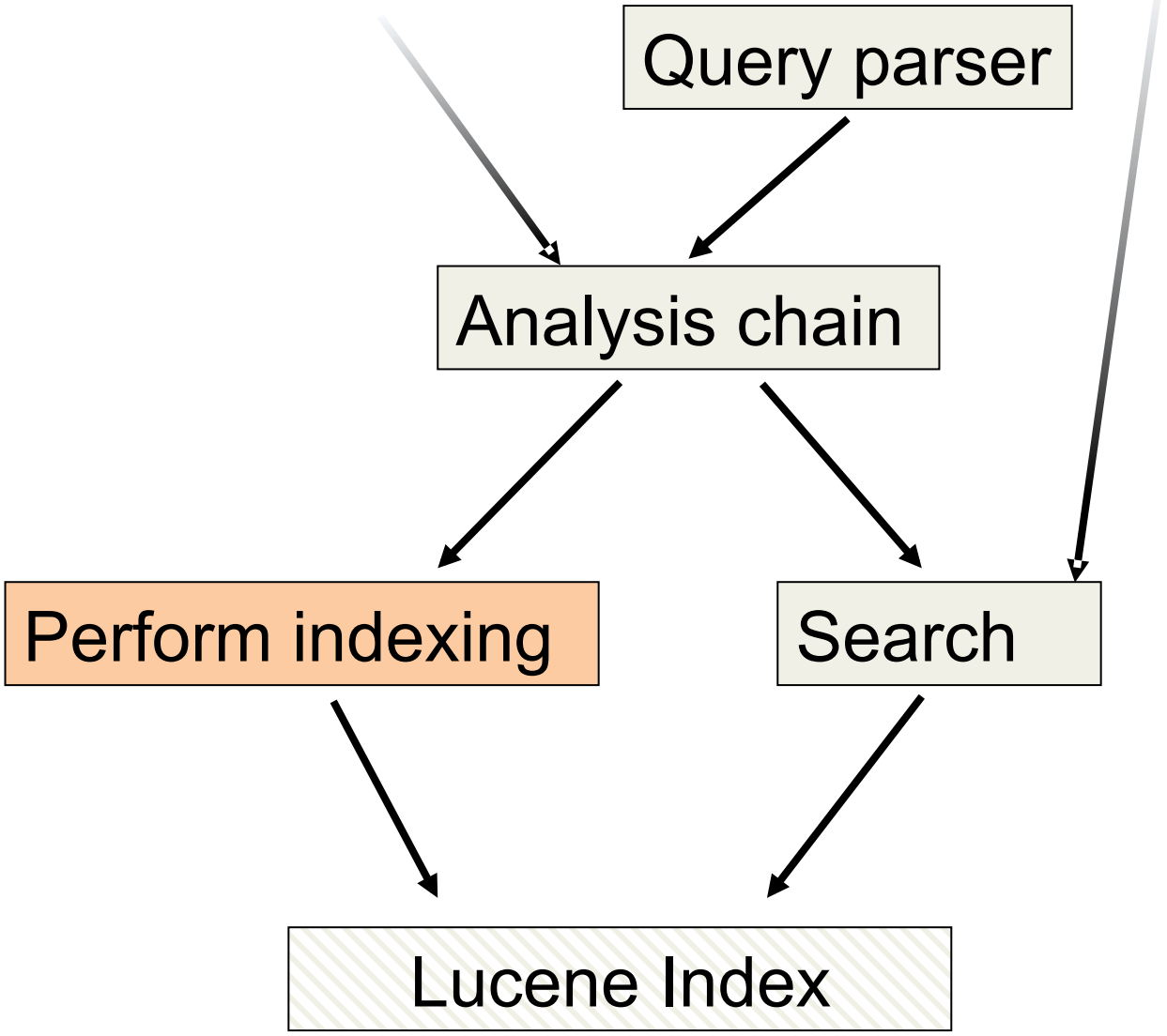
- Sorting, filters, facets, script fields, custom scoring, aggregations, document updates
- MVEL, but others are supported
- Generally speaking: SLOOOOOOOW
- Mostly useful as quick mocks / PoC
- Native scripts using Java by implementing AbstractExecutableScript & AbstractSearchScript

Custom scoring & similarity

- Function score query
  - Previously known as Custom Score Query
- Similarity

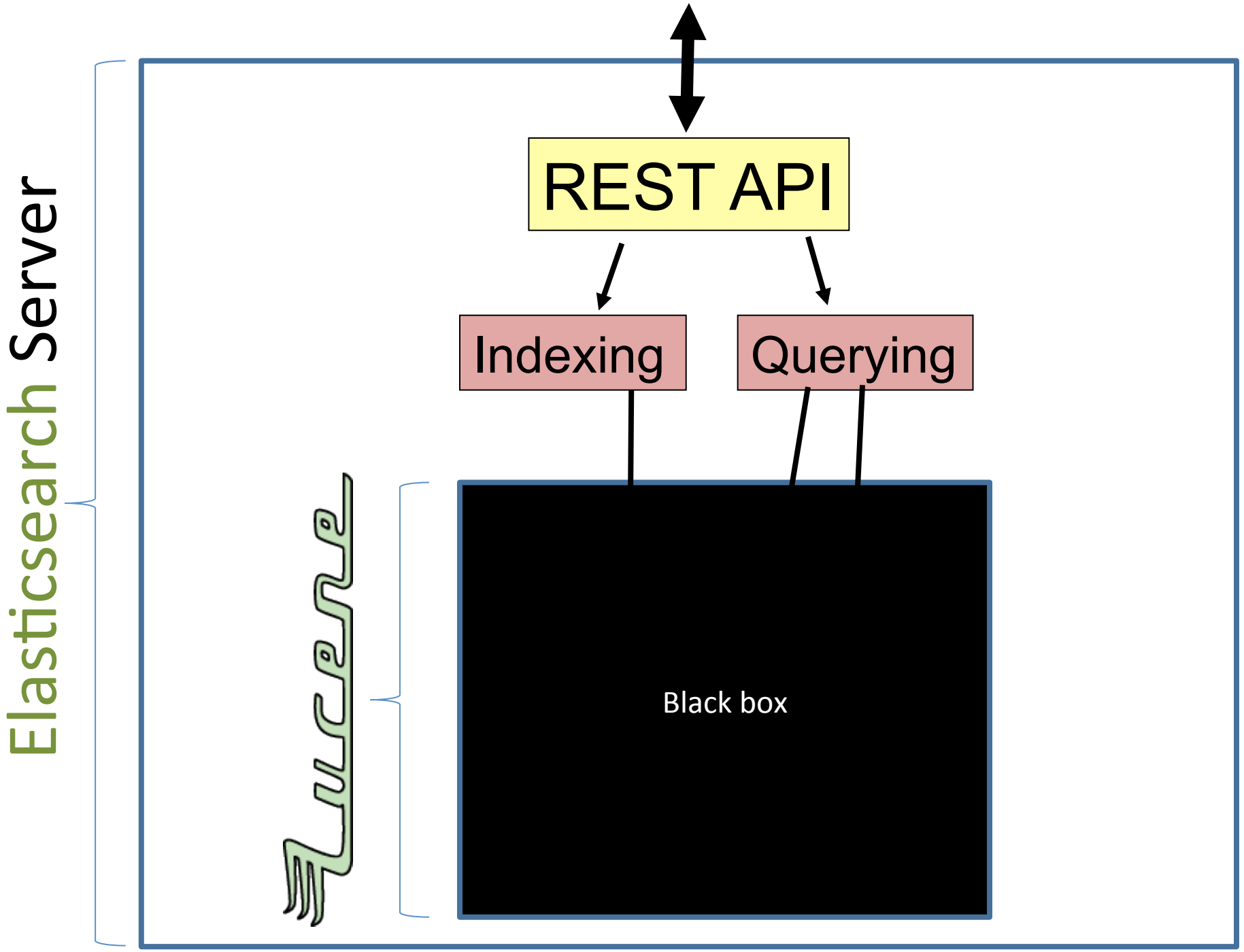
```
"similarity" : {
  "my_similarity" : {
    "type" : "DFR",
    "basic_model" : "g",
    "after_effect" : "l",
    "normalization" : "h2",
    "normalization.h2.c" : "3.0"
  }
}
```

Lucene extension points



Codecs

```
curl -XPUT 'http://localhost:9200/twitter/' -d '{
    "settings" : {
        "index" : {
            "codec" : {
          "postings_format" : {
             "my_format" : {
                "type" : "pulsing",
                "freq_cut_off" : "5"
             }
          }
       }
        }
    }
}'
```


Elasticsearch Server
REST API
Indexing
Querying
Black box

Controlling shard allocation

- Filtering built in
  - By tags, groups, racks, IPs
  - Black list / white list
- Total shards per node
- Disk based
- EXPERT: Roll your own by implementing AllocationDecider

Custom REST endpoints

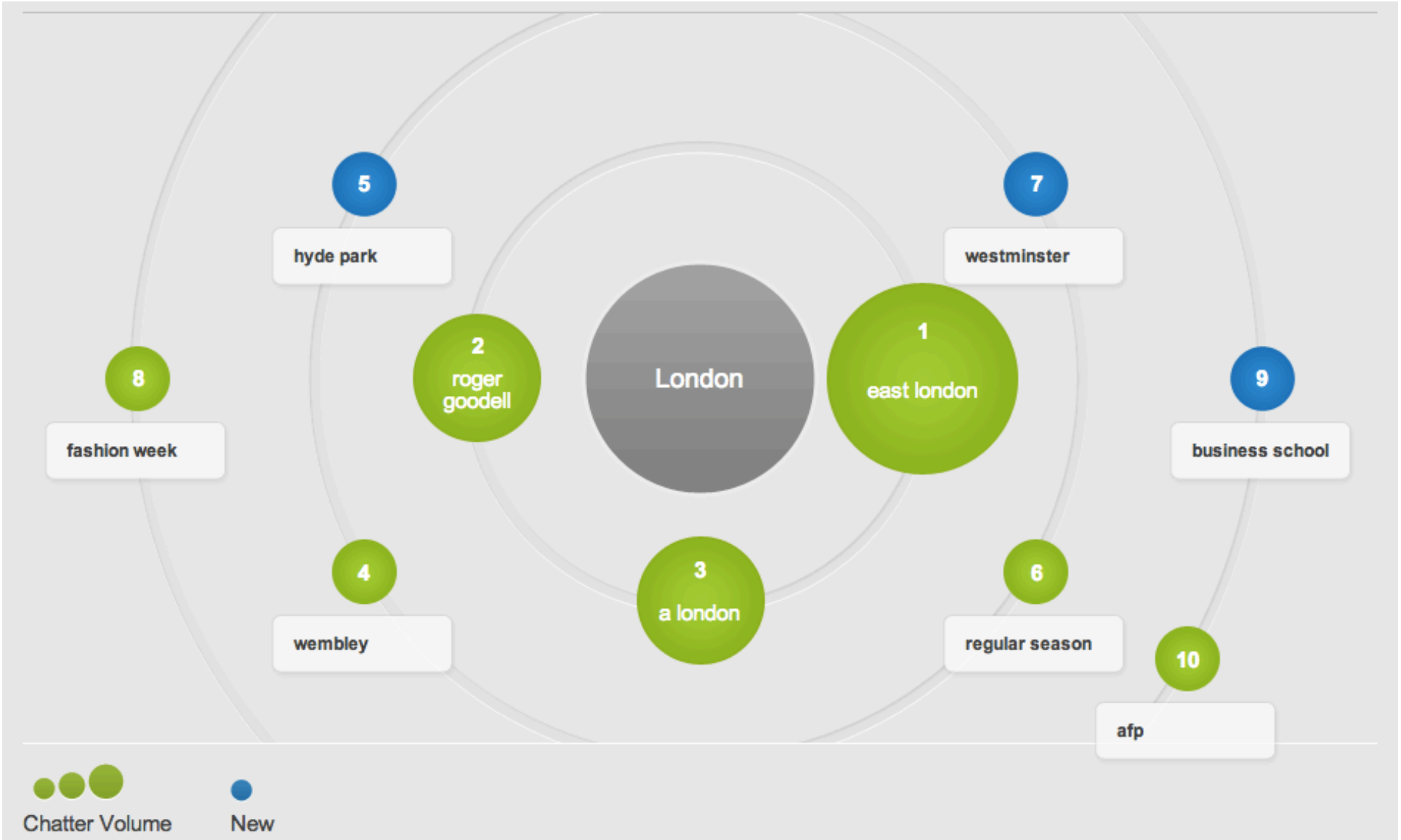
```
public class HelloRestHandler implements RestHandler {
    @Inject
    public HelloRestHandler(RestController restController) {
        restController.registerHandler(GET, "/_hello", this);
    }

    @Override
    public void handleRequest(final RestRequest request, final RestChannel channel) {
        String who = request.param("who");
        String whoSafe = (who!=null) ? who : "world";
        channel.sendResponse(new StringRestResponse(OK, "Hello, " + whoSafe + "!"));
    }
}
```

Transports

- Exposes the Elasticsearch RESTful API over protocols other than HTTP
  - Apache Thrift
  - Memcached
  - Servlet
  - Redis
  - ZeroMq

Showcase: Custom percolator

Showcase: The bubble plugin



Site plugins

- Monitoring
  - BigDesk, ElasticHQ, Paramedic, …
- Hammer (GUI for REST interface)
- Inquisitor (debugging queries)
- SegmentSpy
- WhatsOn

Discovery

- Default is Zen discovery
  - Unicast: I know who my nodes are
  - Multicast: Auto discovery for nodes
- Multicast discovery support for cloud environments
  - AWS
  - Azure
  - Google Compute
- ProTip: Unicast in production unless you know what you're doing
- ZooKeeper plugin

Snapshot / restore repositories

- File system
- AWS S3
- HDFS
- Azure
- Roll your own (e.g. Glacier)

River plugins

- Obsolete
- Use the “shoveller” approach
- logstash, stream2es

Summary: Plugin types

- Lucene components
  - Analysis
  - Similarity
  - Scoring
- REST endpoints
- Scripting
- ES infrastructure (Discovery, Transport, Snapshot/restore)
- Site plugins
- ~~River plugins~~

Installing plugins

- Manual under /plugins
- Official / GitHub / Maven installation:

```
plugin --install <org>/<user/component>/<version>
```

- From zip:

```
bin/plugin --url file:///path/to/plugin --install plugin-name
```

- Plugin management:

```
curl -XGET 'http://localhost:9200/_nodes/plugins'
```

```
plugin --remove <pluginname>
```

When to write a plugin?

Writing your own plugin: Gotchas

- **Maintenance** – the deeper you go in the API the harder it is to keep it up to date
- **Versioning and installation** on (large) clusters
  - Though can be solved using puppet, docker et al
- **Auxiliary data** (like dictionaries etc)
- **Testing & Debugging**

Code: Writing your own plugin

- JAR file with bootstrap code:

```
public class ExamplePlugin extends AbstractPlugin {
    @Override public String name() {
        return "example-plugin";
    }

    @Override public String description() {
        return "Example Plugin Description";
    }

    @Override
    public Collection<Class<? extends Module>> modules() {
        Collection<Class<? extends Module>> modules = Lists.newArrayList();
        modules.add(ExampleRestModule.class);
        return modules;
    }
}
```

- Embed this as es-plugin.properties:

```
plugin=org.elasticsearch.plugin.example.ExamplePlugin
```

Thank you.
Questions?

Itamar Syn-Hershko
http://code972.com
@synhershko