adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 28-30 SEPTEMBER 2015

Scaling the query with Oak

Davide Giannella, Adobe.

Introduction

Presentation Goals

- Better understanding of Query engine
- Better fine tune queries and indexes

What’s Oak?

- Scalable Content Repository
- Adhere JCR2 Specifications
- Designed for concurrent access (MVCC)
- Pluggable Components (storage, index)
- Combined with Sling power AEM 6.x

Jackrabbit 2 VS Oak (aka JR3)

Jackrabbit 2

- Index everything
- It’s (mostly) synchronous

Oak

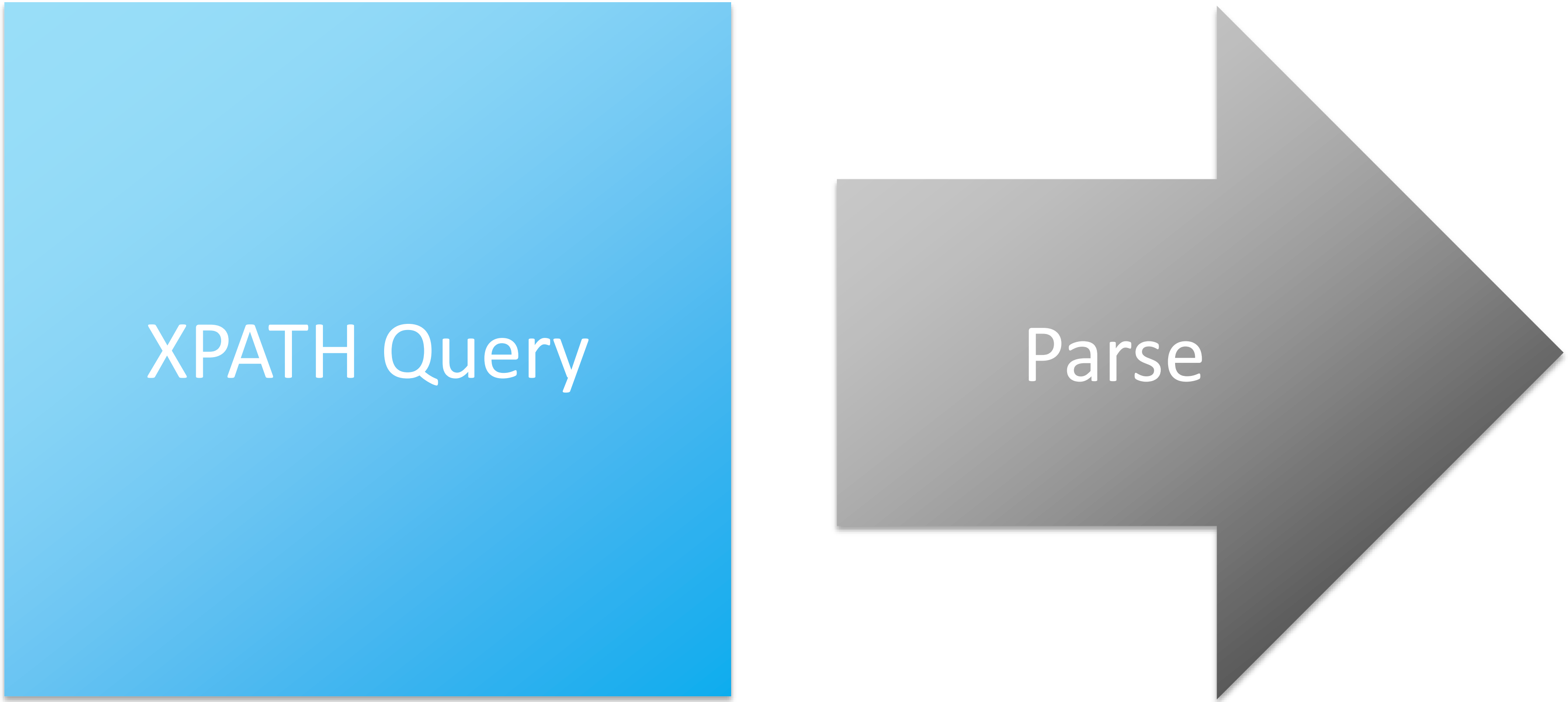
- Index nothing (depending on Initialiser)
- Can be synchronous and asynchronous

A day in a life of a query

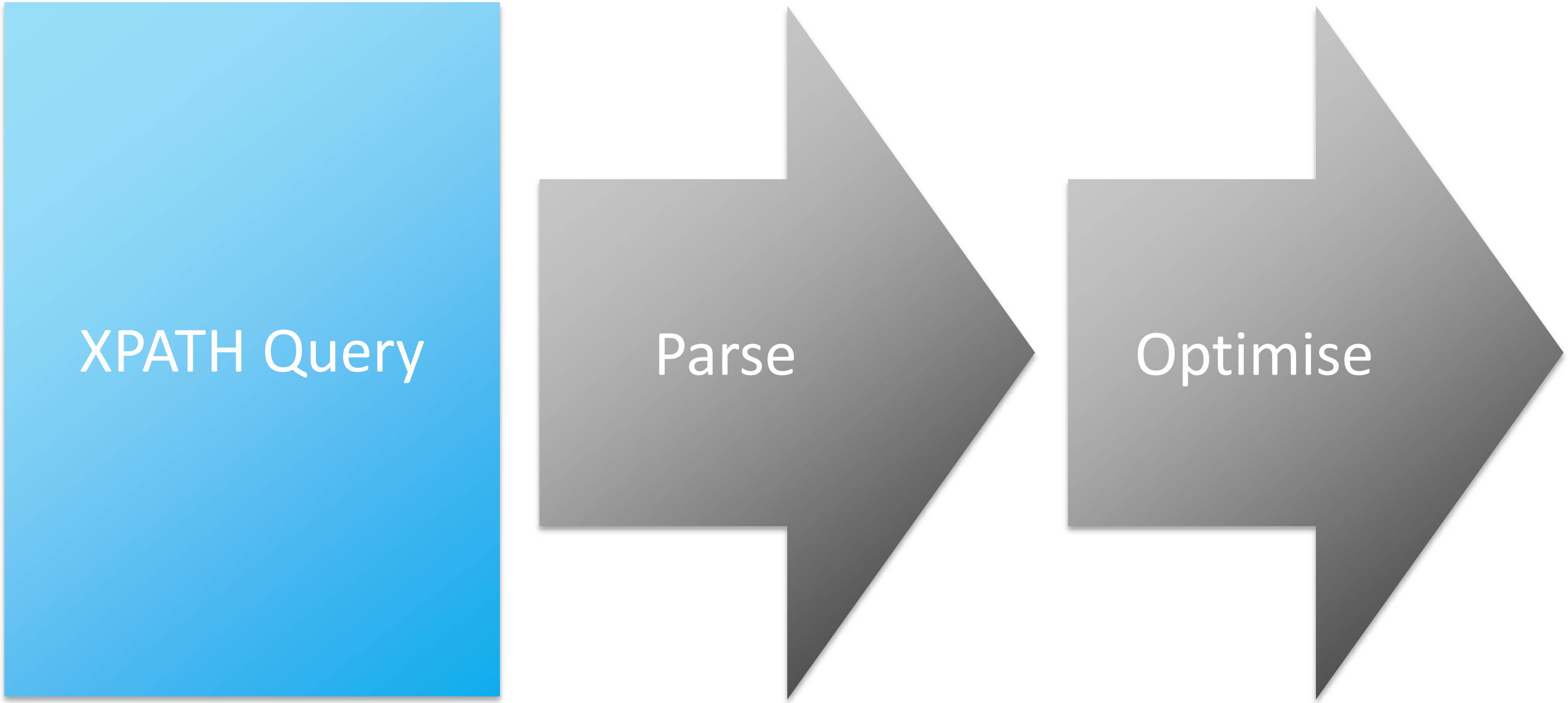
A day in a life of a query

XPATH Query

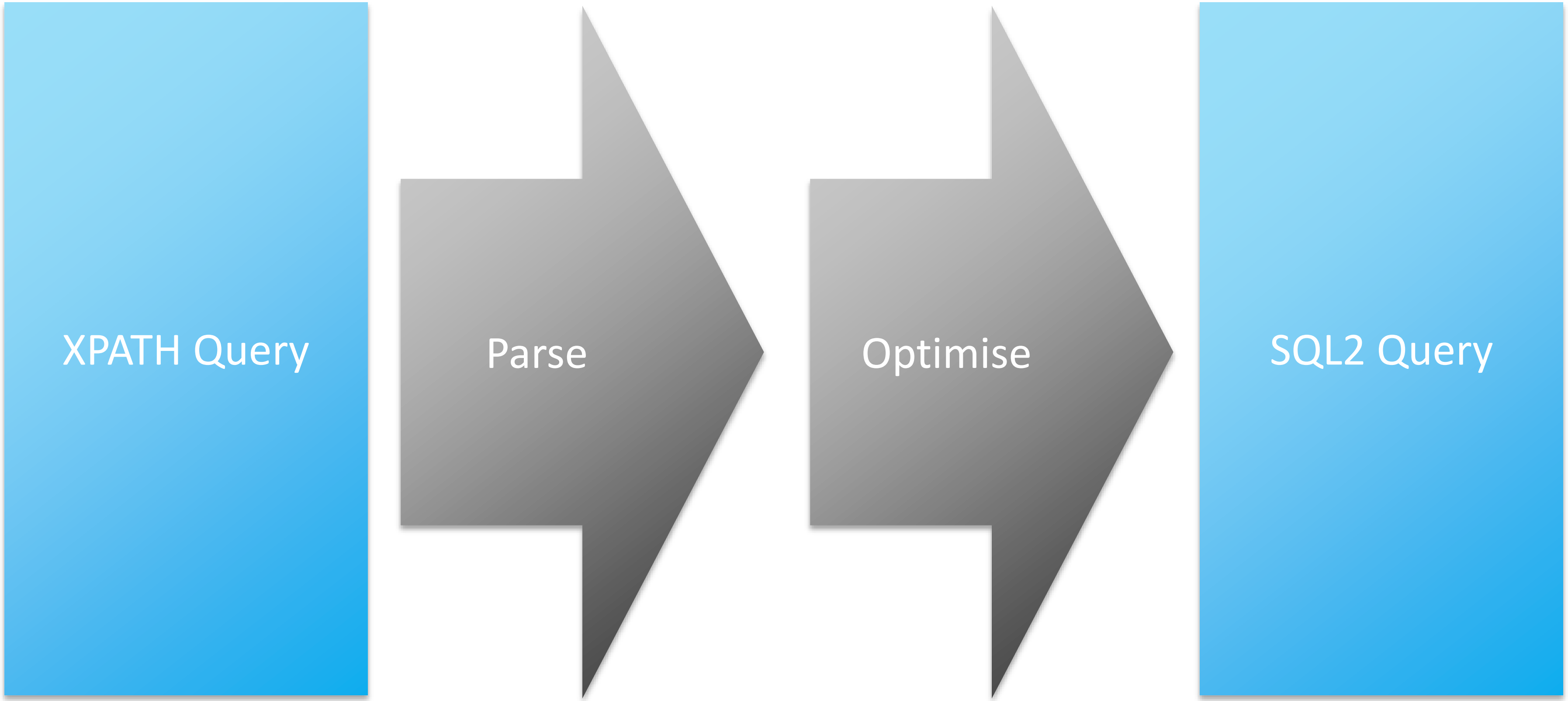
A day in a life of a query



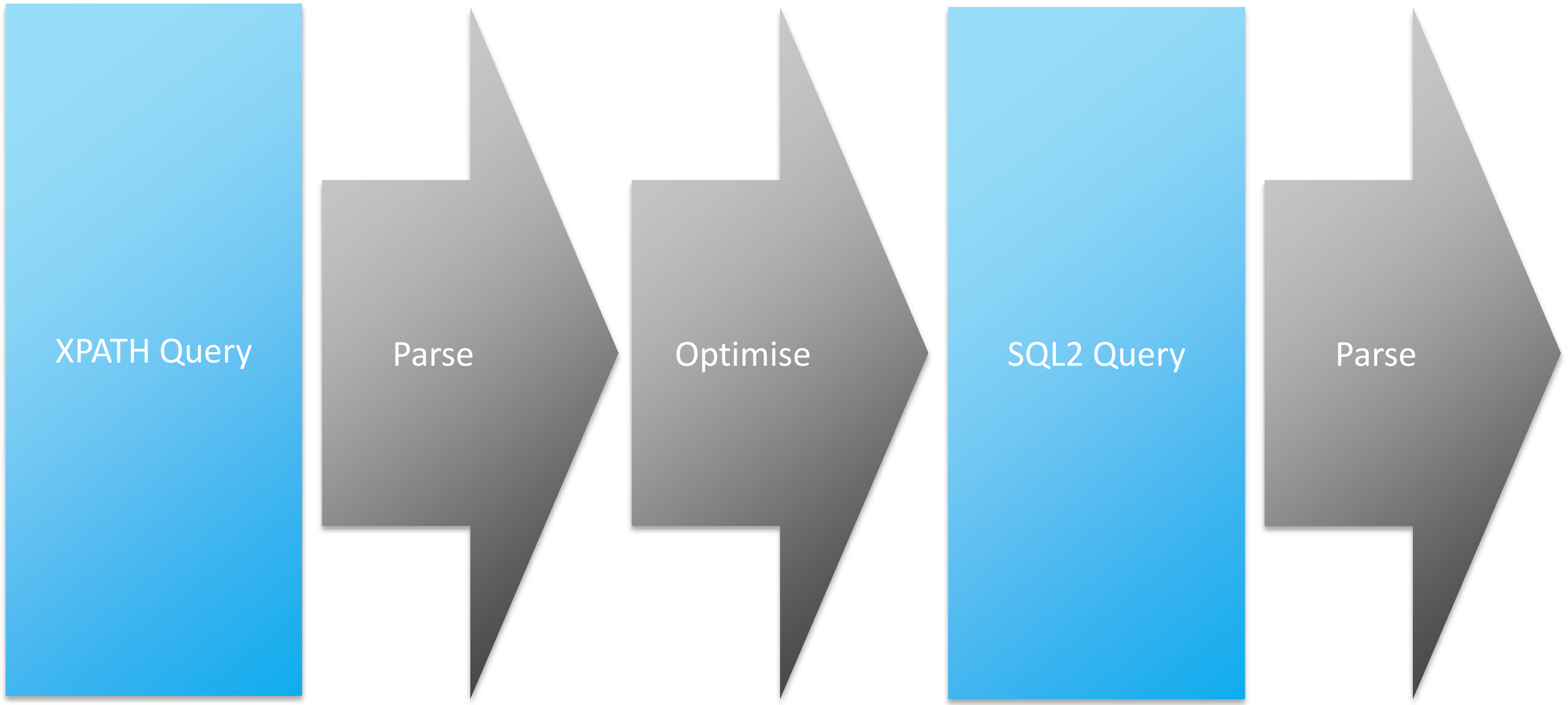
A day in a life of a query



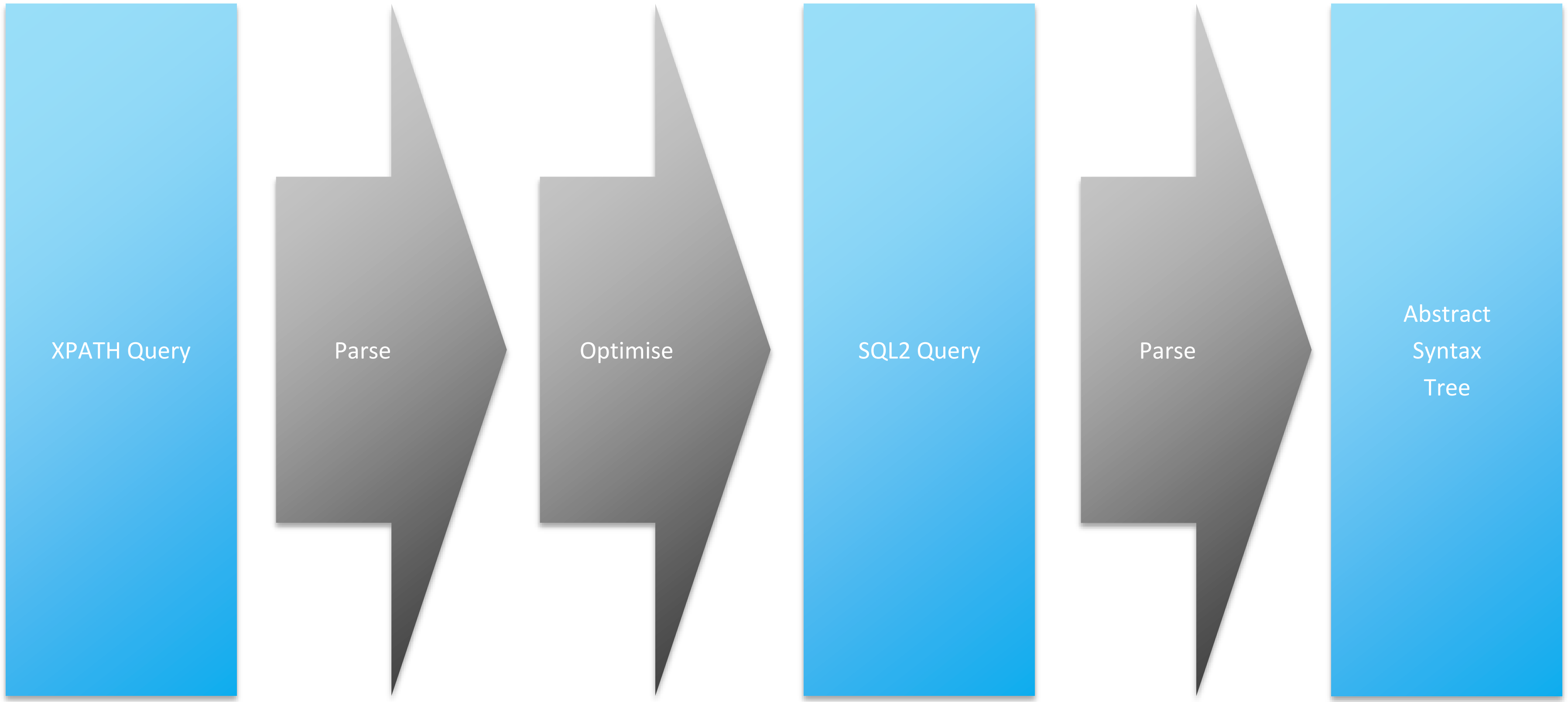
A day in a life of a query



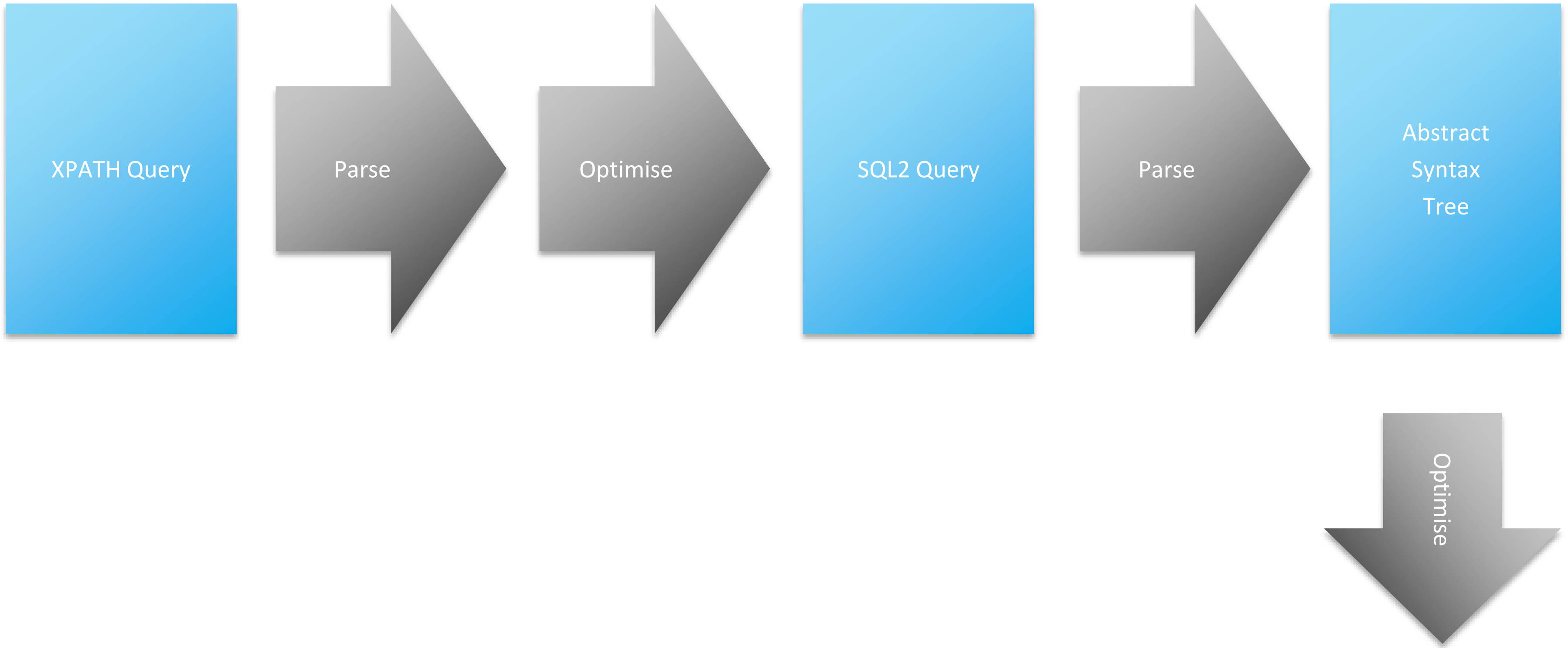
A day in a life of a query



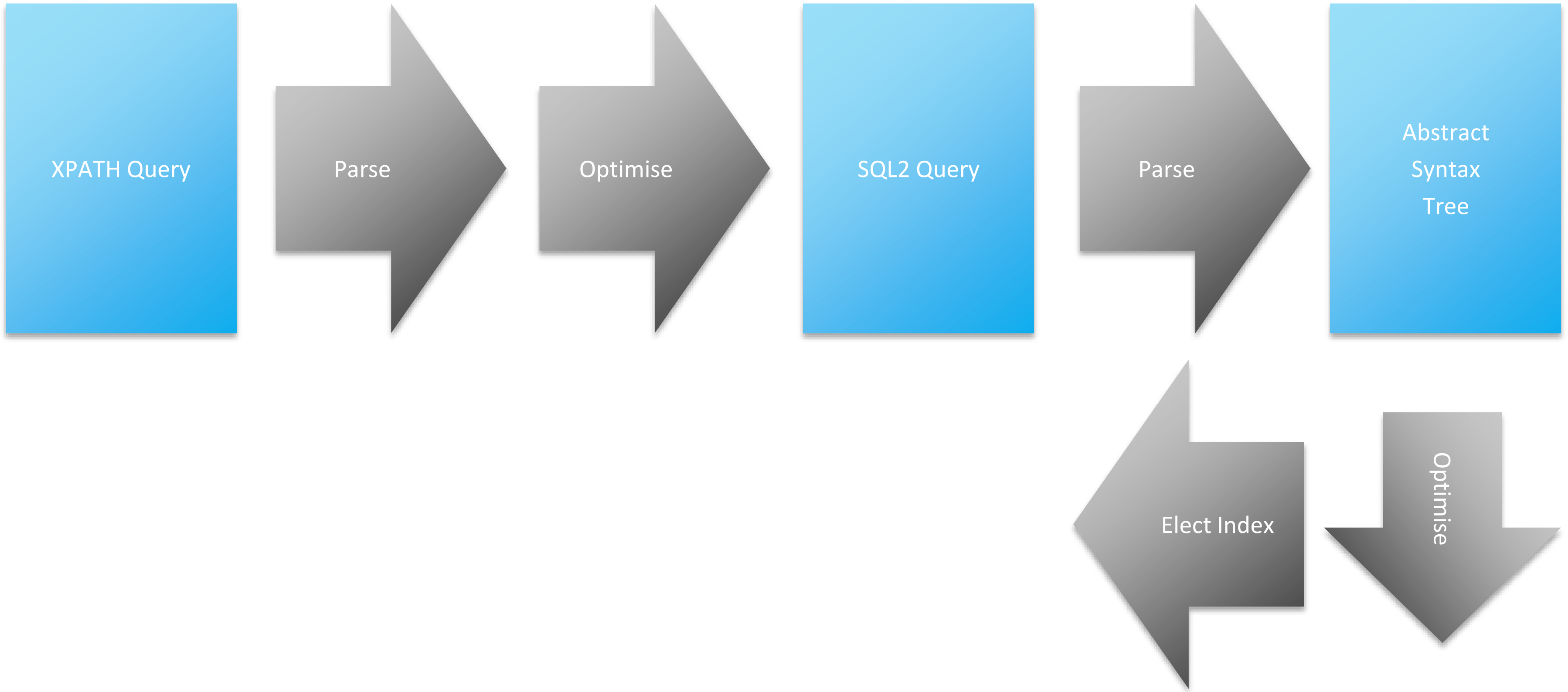
A day in a life of a query



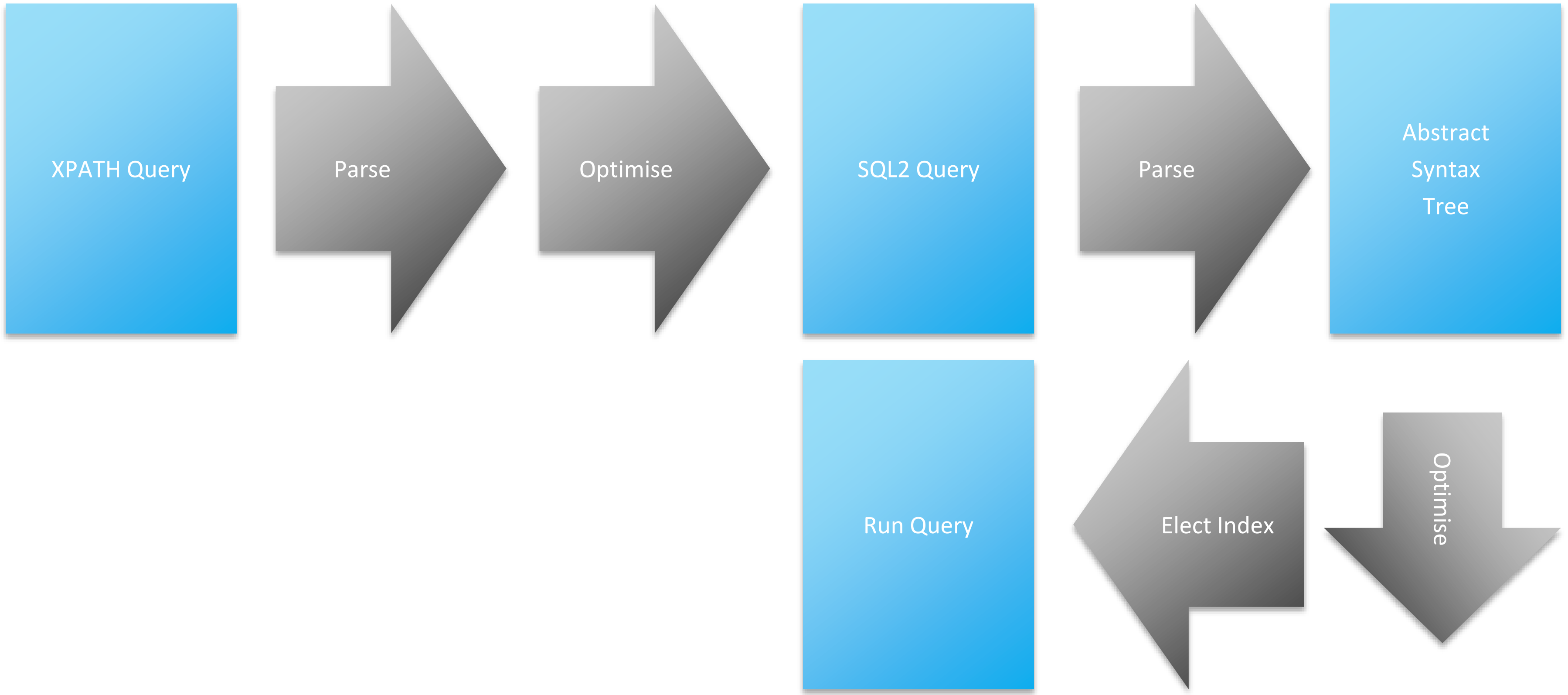
A day in a life of a query



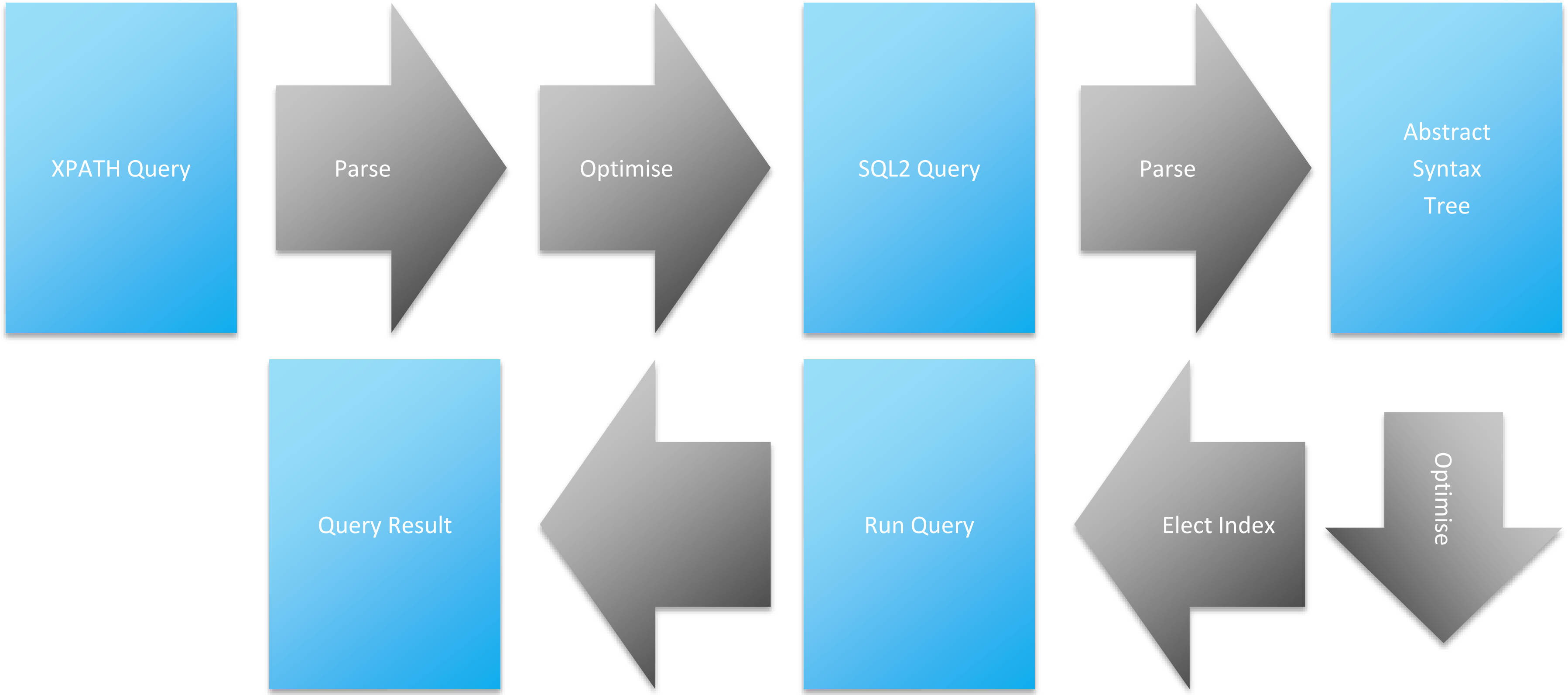
A day in a life of a query



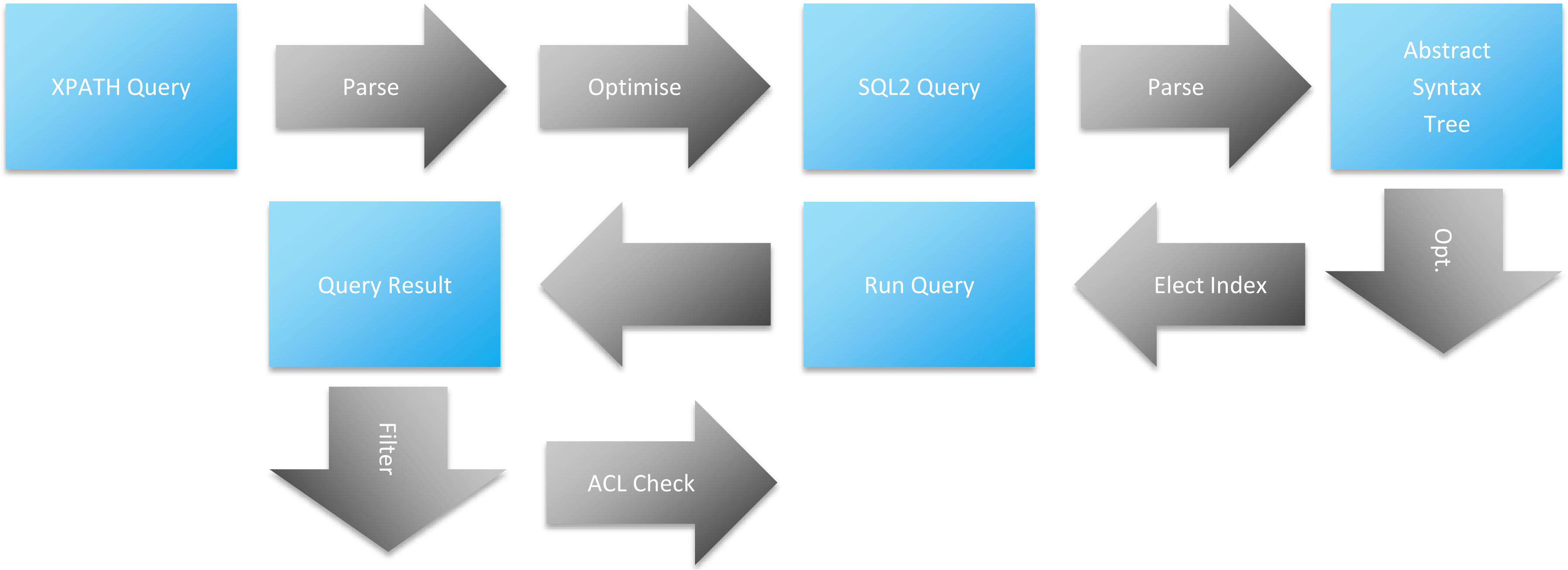
A day in a life of a query



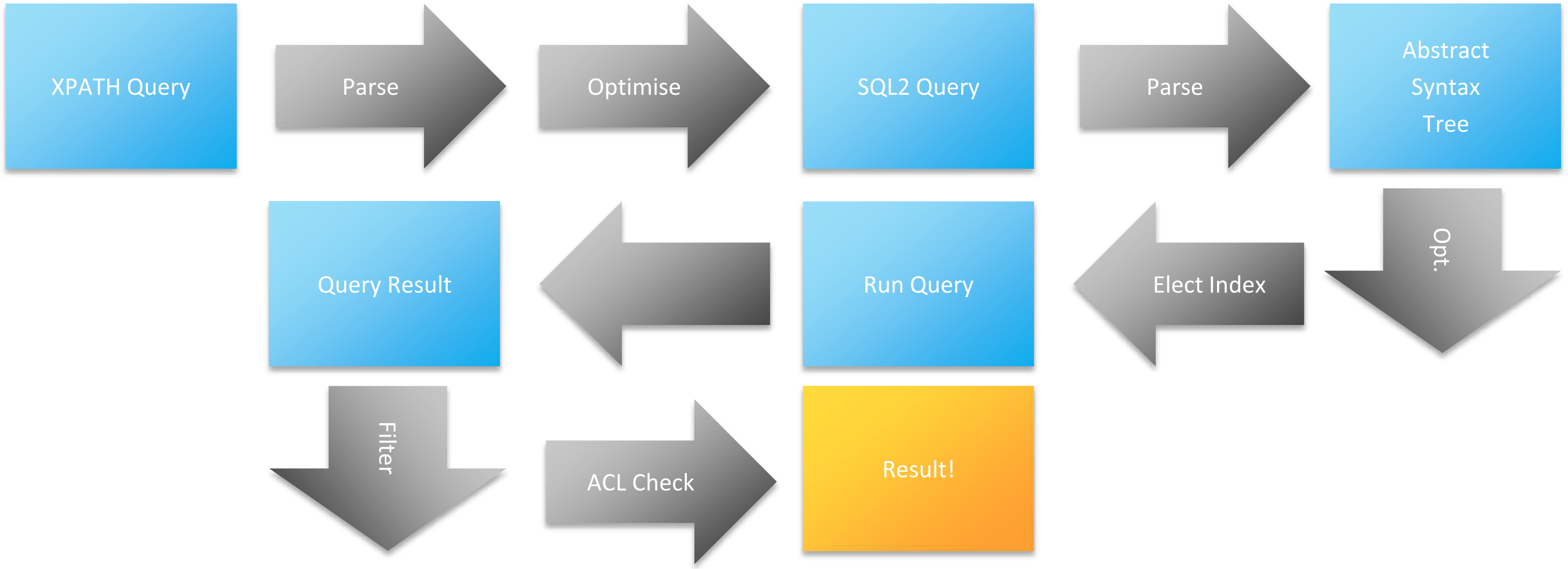
A day in a life of a query



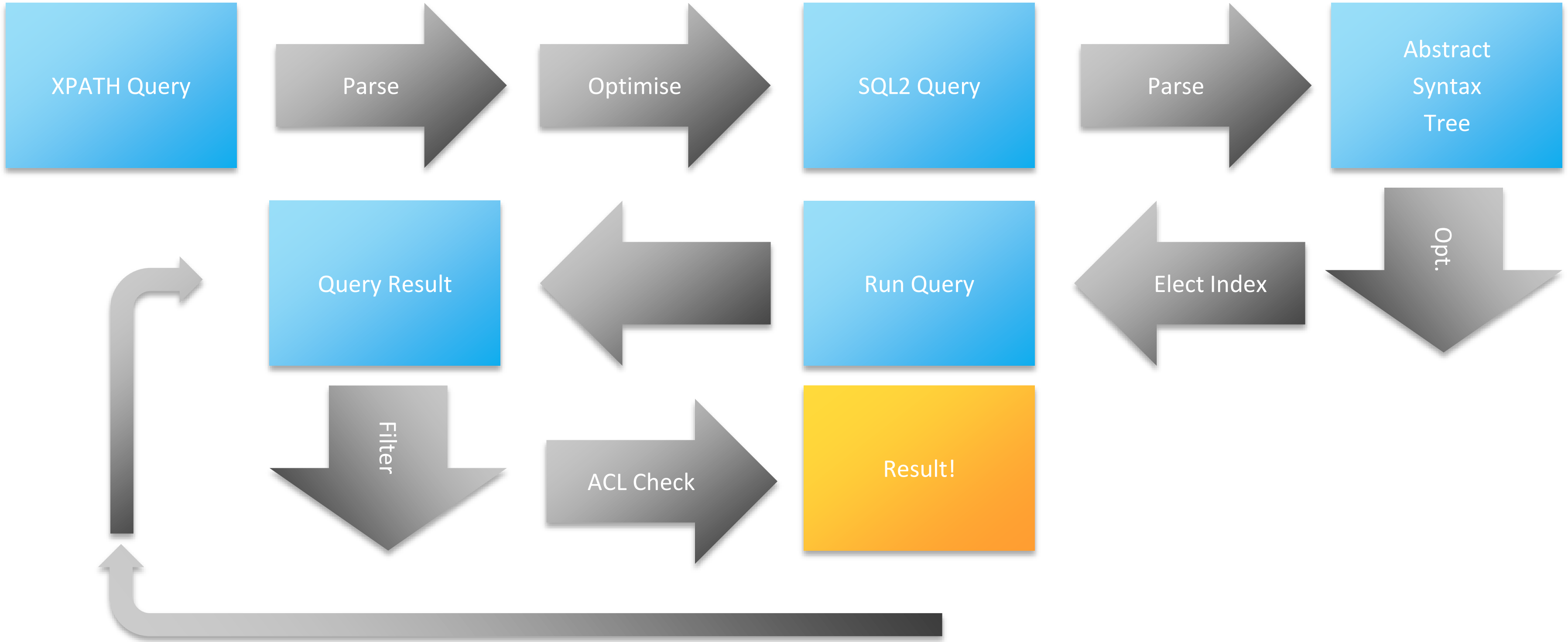
A day in a life of a query



A day in a life of a query



A day in a life of a query



Query Engine features (Overview)

Query engine features

- Pluggable index types
- Different type of indexes
- Cost based optimiser
- Different dialects (Xpath, SQL)

Index types

- Property Index
  - Index nodes based on a specific property
  - Synchronous, Asynchronous
  - Unique, non-unique

Index types

- Traversing index
  - Traverse the repo
  - Synchronous

Index types

- Nodetype index
  - Based on property index
  - Index all node types (nt:unstructured, …)
  - Synchronous

Index types

- Lucene Aggregate
  - Full text
  - Aggregation
  - Asynchronous

Index types

- Lucene Property
  - Full text
  - Sorting
  - Property index
  - Aggregation
  - Asynchronous

Index types

- Solr
  - Full text
  - Property
  - Embedded or remote (cloud)
  - Mostly configured on Solr side
  - Asynchronous

Few (bad) words on Solr

- It’s not 42!
- Opt for lucene in first stance
- Remote == network latency

Few (good) words on Solr

- Geo location
- Full text
- Suggester
- Can scale on the cloud

Few words on lucene

- Can evaluate path restrictions
- Can include certain paths (whitelist)
- Can exclude certain paths (blacklist)
- Suggestion, spellchecking, …

- http://jackrabbit.apache.org/oak/docs/query/lucene.html

Few words on lucene. When?

- Synchronous is **not** a requirement.
- Always my first choice (personal opinion)
- You need
  - Sorting, Aggregation, Full-text, multiple properties filter.

Index Selection

- Works on a cost basis. The cheapest (lowest cost) wins.
- The cost is an estimate of number of nodes in the index
- The broader the index the higher the cost

Index Selection

```
org.apache.jackrabbit.oak.query.QueryImpl cost for aggregate lucene is
Infinity
org.apache.jackrabbit.oak.query.QueryImpl cost for lucene-property is 1001.0
org.apache.jackrabbit.oak.query.QueryImpl cost for reference is Infinity
org.apache.jackrabbit.oak.query.QueryImpl cost for ordered is Infinity
org.apache.jackrabbit.oak.query.QueryImpl cost for nodeType is 10005.0
org.apache.jackrabbit.oak.query.QueryImpl cost for property is Infinity
org.apache.jackrabbit.oak.query.QueryImpl cost for traverse is 1.0E7
```

Index Selection: example

```
SELECT *
FROM [nt:unstructured] AS a
WHERE colour = ‘red’
AND ISDESCENDANTNODE(a, ‘/content’)
ORDER BY size
```

Index Selection: example

Nodetype index

```
SELECT *
FROM [nt:unstructured] AS a
WHERE colour = ‘red’
AND ISDESCENDANTNODE(a, ‘/content’)
ORDER BY size
```

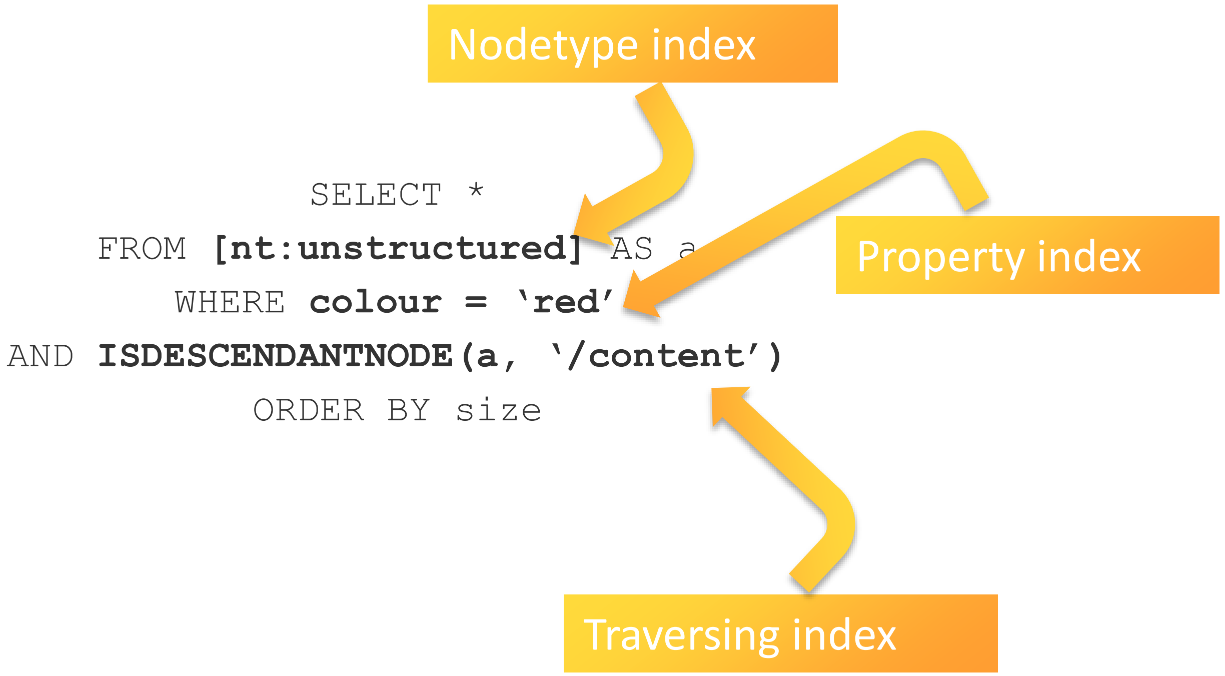
Index Selection: example

Nodetype index

Property index

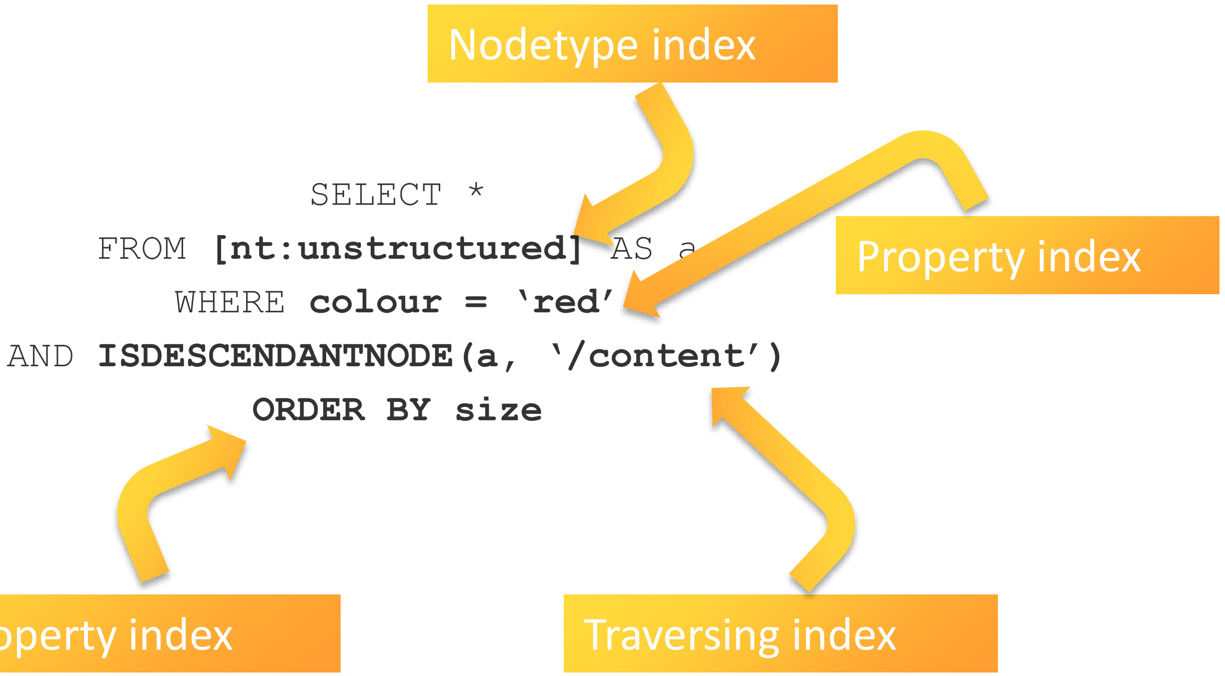
```
SELECT *
FROM [nt:unstructured] AS a
WHERE colour = ‘red’
AND ISDESCENDANTNODE(a, ‘/content’)
ORDER BY size
```

Index Selection: example



```
            SELECT *
   FROM [nt:unstructured] AS a
        WHERE colour = 'red'
AND ISDESCENDANTNODE(a, '/content')
            ORDER BY size
```

Index Selection: example



```
SELECT *
FROM [nt:unstructured] AS a
WHERE colour = ‘red’
AND ISDESCENDANTNODE(a, ‘/content’)
ORDER BY size
```

Xpath VS SQL

XPath VS SQL

- XPATH: XML Path Language
- SQL2: Oak low level

XPath VS SQL

```
/jcr:root//*[@foo='bar']
```

```
SELECT *
FROM [nt:base]
WHERE [foo] = 'bar'
```

XPath VS SQL

```
//jcr:root/content//baz*[@foo='bar'
]
```

```
SELECT b.*
FROM [nt:base] AS a
INNER JOIN [nt:base] AS b ON
ISCHILDNODE(b, a)
WHERE NAME(a) = 'baz'
AND ISDESCENDANTNODE(a, '/content')
AND b.foo = 'bar'
```

XPath VS SQL

```
//*[@foo='bar' OR @foo='baz']
```

```
SELECT [jcr:path], [jcr:score]
FROM [nt:base]
WHERE [foo] in('bar', 'baz')
```

XPath VS SQL

```
//*[@foo='bar' OR @fur='baz']
```

```
SELECT [jcr:path], [jcr:score]
FROM [nt:base]
WHERE [foo] = 'bar'
UNION
SELECT [jcr:path], [jcr:score]
FROM [nt:base]
WHERE [fur] = 'baz'
```

The End

@flyingedivad