Elasticsearch Query Language
ES|QL

David Pilato - @dadoonet

Developer │ Evangelist

Query DSL

```
GET _search
{
    "query": {
        "match": {
            "quote": "The force is strong"
        }
    }
}
```

Query DSL

```
GET _search
{
    "aggs": {
        "quotes-by-character": {
            "terms": {
                "field": "character"
            }
        }
    }
}
```
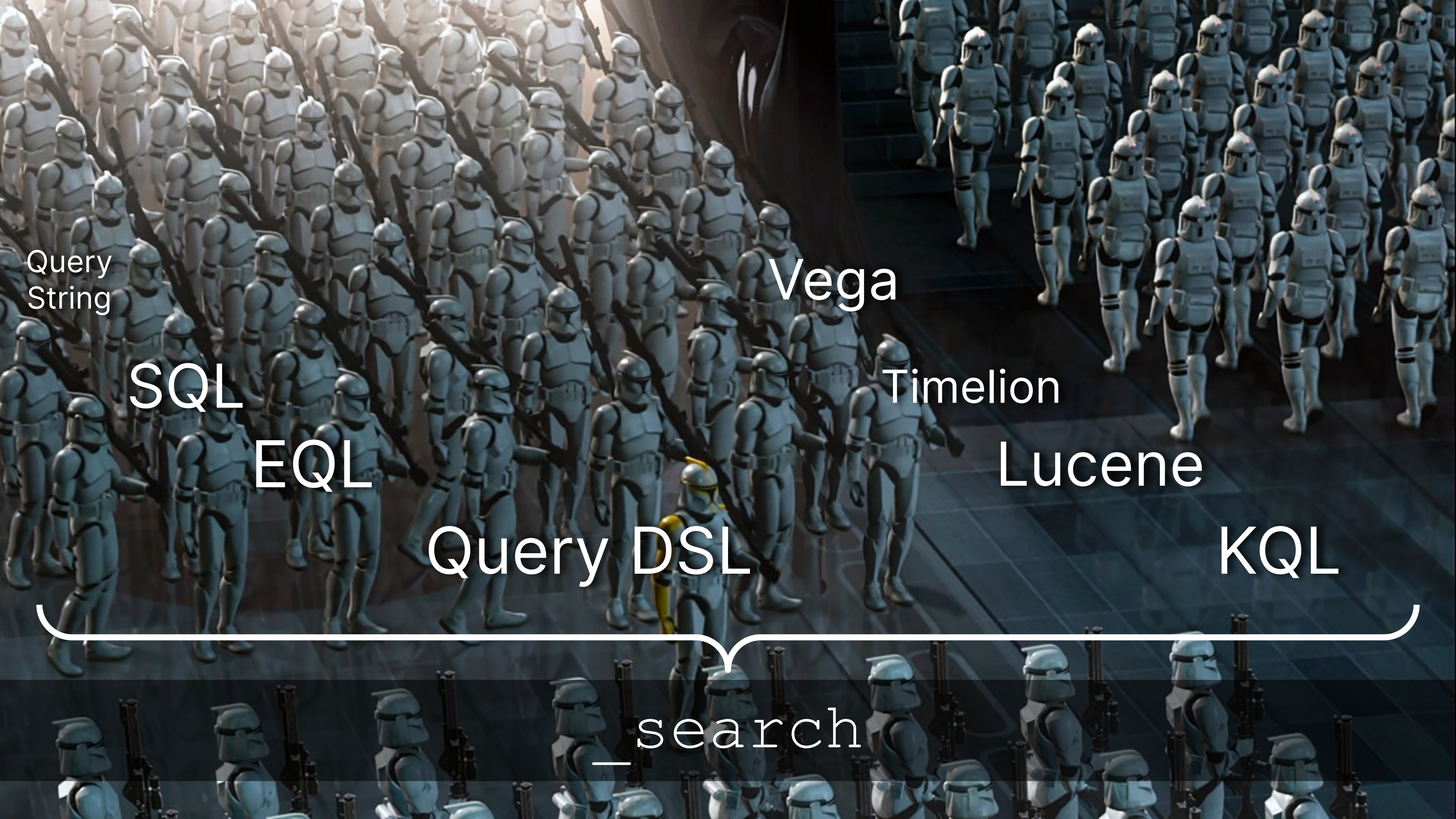
elastic
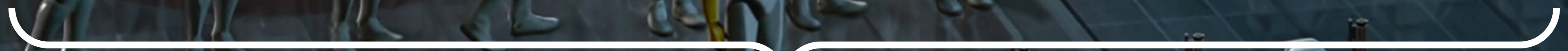
How many query languages use `_search`?

A brief history of Elasticsearch's analytical capabilities

2010 ➡ **2013** ➡ **2014** ➡ **2015** ➡ **2023**

Elasticsearch 0.9

Elasticsearch < 0.90

Elasticsearch 1.0

Elasticsearch 2.0

Elasticsearch 8.11

Facets

Facet terms-stats

Aggregations

Pipeline aggregations

ES|QL

elastic

~17 months of development

ESQL: Setup project (#144)

esql/agg-func-mapping

costin committed on Jun 15, 2022 Verified

Showing **3 changed files** with **58 additions** and **0 deletions**.

https://github.com/elastic/elasticsearch/commit/8a1022e1c75fd2b99fa77a7ec548b0b2979b4662

ES|QL

- Language
- Engine
- Visualization

elastic

ES|QL

the language

elastic

ES|QL Features

- Unstructured and structured data
- Piped query language
- SQL-like filtering and data manipulation
- Lookups

elastic

ES|QL commands

Source (From, Row)

Filter (Where)

Processing (Eval)

Aggregation (Stats)

TopN (Sort + Limit)

Expansion (Enrich , MV_Exand)

Extraction (Dissect, Grok)

75+ functions:

- 10 aggregate
- 20+ math
- 10+ string
- 7 date-time
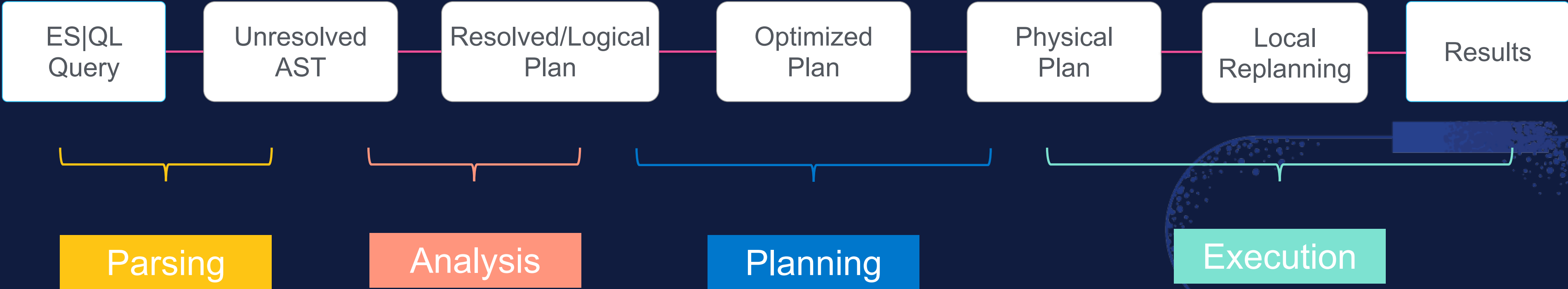- 15 conversion
- 4 conditionals
- 12 multi-value / mv_

elastic

ES|QL

the engine

elastic

> The new ES|QL execution engine was designed with performance in mind — it operates on blocks at a time instead of per row, targets vectorization and cache locality, and embraces specialization and multi-threading. It is a separate component from the existing Elasticsearch aggregation framework with different performance characteristics.

elastic

Query planner

✓ Flexible distributed execution
✓ Allow multiple roundtrips

| ES|QL Query	Unresolved AST	Resolved/Logical Plan	Optimized Plan	Physical Plan	Local Replanning	Results

Parsing **Analysis** **Planning** **Execution**

Compute engine

✓ Tabular data representation
✓ From 1 thread per shard to many
✓ Spilling to disk if needed
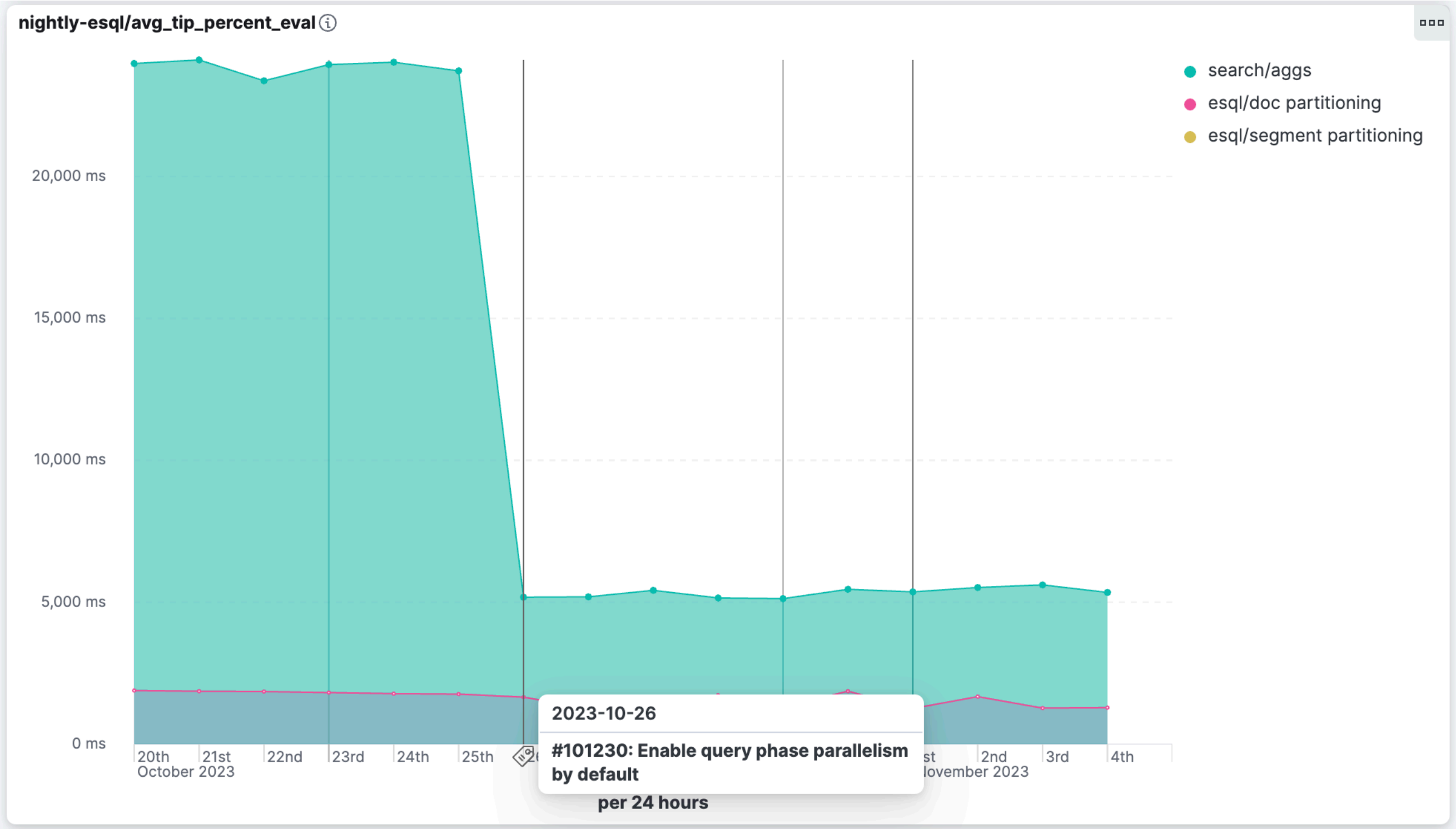✓ Streaming of data across nodes

elastic

Vectorization

*"convert from a scalar implementation,
which processes a single pair of operands
at a time, to a vector implementation,
which processes one operation on multiple
pairs of operands at once. "*

```
for (i = 0; i < n; i++)

    c[i] = a[i] + b[i];
```

elastic
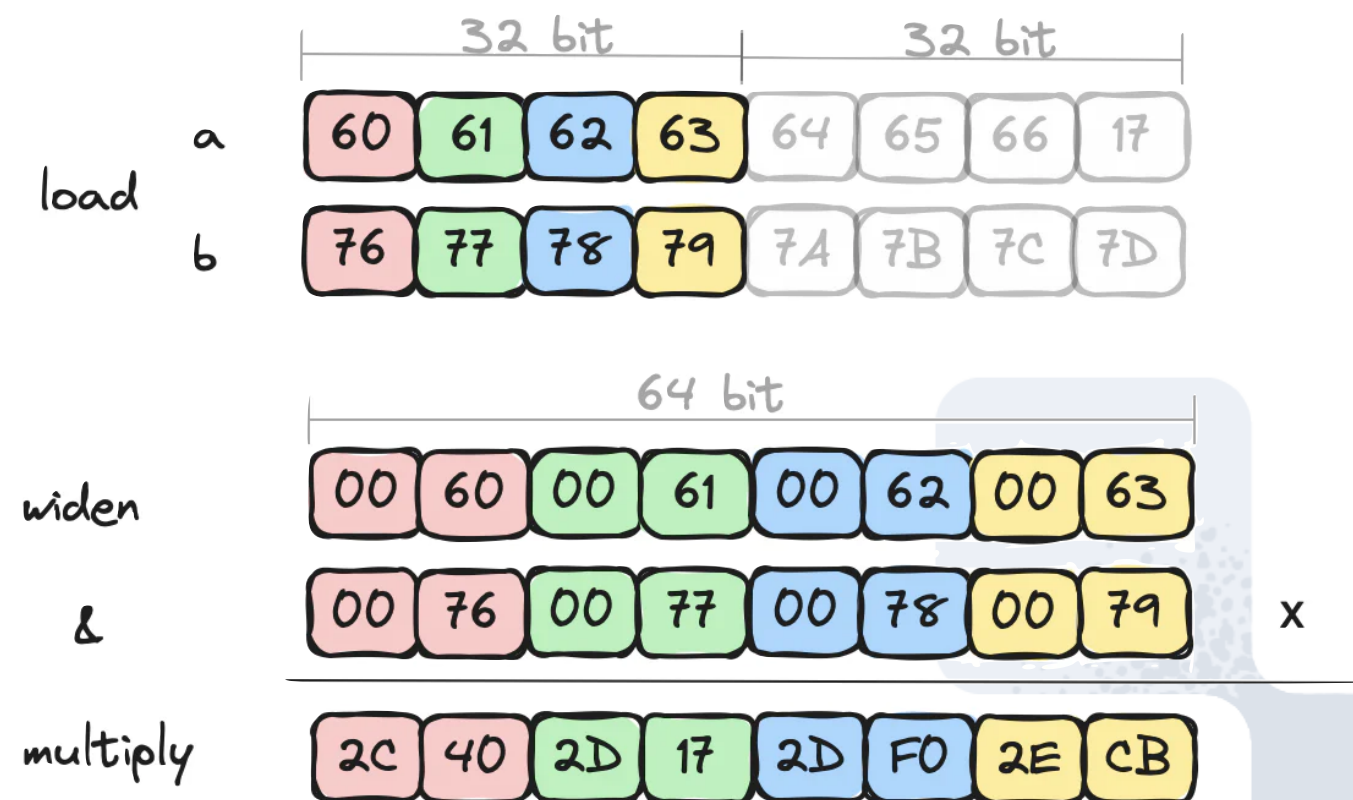
Benchmarks

ES|QL

the visualization

elastic

ES|QL

in action

https://github.com/dadoonet/esql-demo
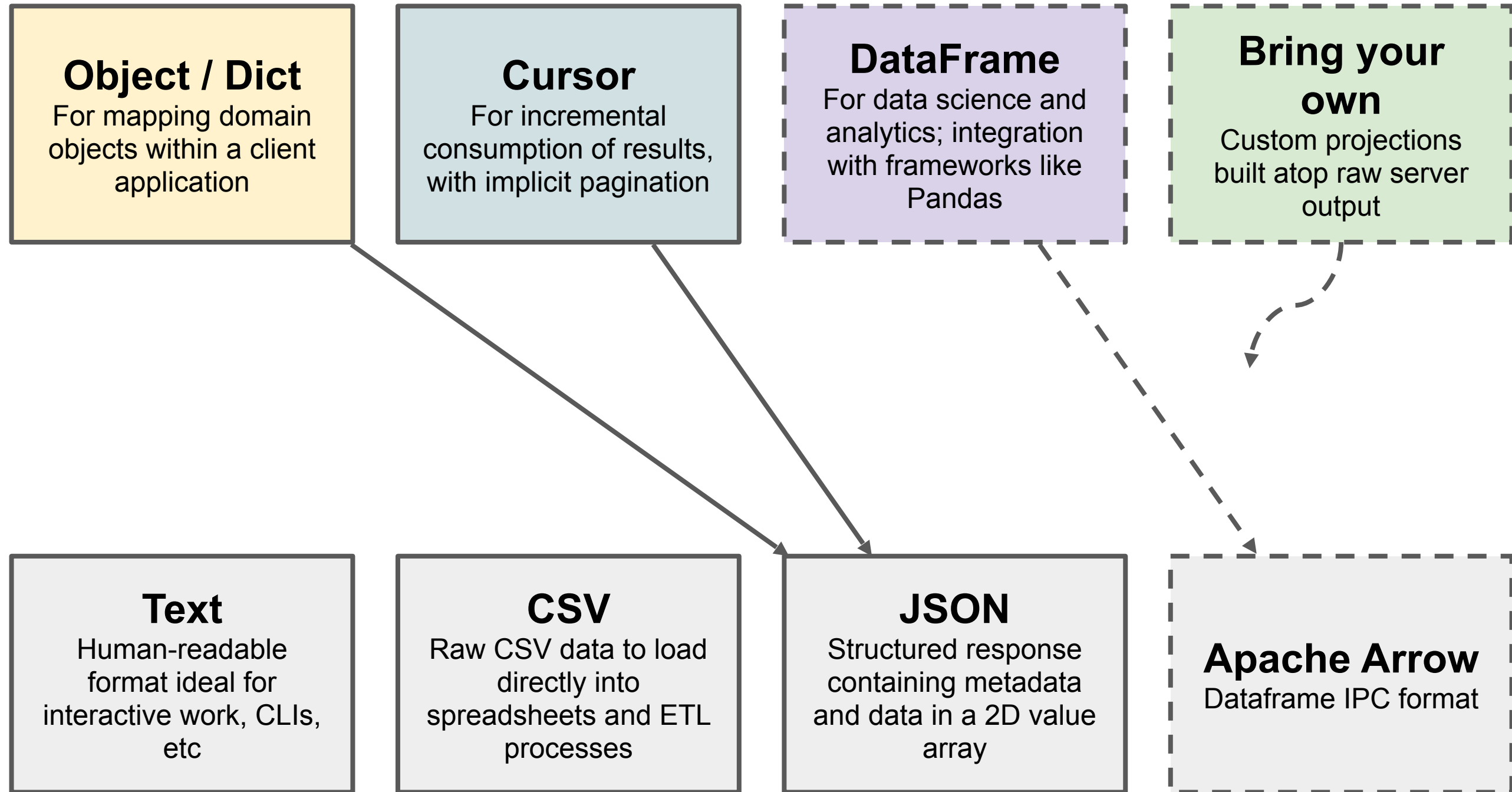
Ways to consume ES|QL results

Each language client will offer a selection of projections relevant to that language ecosystem.

Object / Dict
For mapping domain objects within a client application

Cursor
For incremental consumption of results, with implicit pagination

DataFrame
For data science and analytics; integration with frameworks like Pandas

Bring your own
Custom projections built atop raw server output

Users can consume raw data directly from the server output in one of several formats.

Text
Human-readable format ideal for interactive work, CLIs, etc

CSV
Raw CSV data to load directly into spreadsheets and ETL processes

JSON
Structured response containing metadata and data in a 2D value array

Apache Arrow
Dataframe IPC format

elastic

Object API

https://github.com/dadoonet/elasticsearch-java-client-demo

```java
String query = """
    FROM persons
    | WHERE name == "David"
    | KEEP name
    | LIMIT 1
    """;


Iterable<Person> persons = client.esql()
    .query(ObjectsEsqlAdapter.of(Person.class), query);
for (Person person : persons) {
    assertNull(person.getId());
    assertNotNull(person.getName());
}
```

elastic

ResultSet JDBC API

```java
String query = """
    FROM persons
    | WHERE name == "David"
    | KEEP name
    | LIMIT 1
    """;

try (ResultSet resultSet = client.esql()
    .query(ResultSetEsqlAdapter.INSTANCE, query)) {
        assertTrue(resultSet.next());
        assertEquals("David", resultSet.getString(1));
}
```
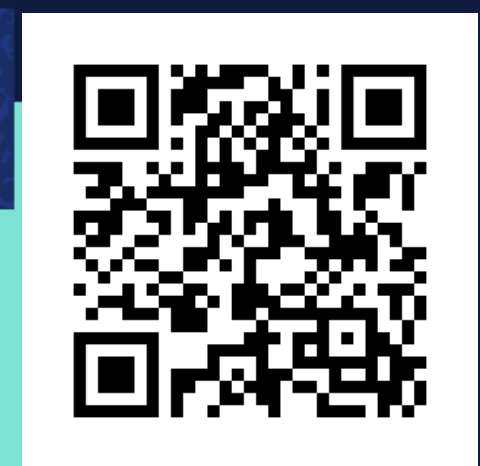
elastic

ES|QL

- Language
- Engine
- Visualization

elastic

Try it!

https://esql.demo.elastic.co

elastic