



TAD

Elasticsearch Query Language

ES|QL

David Pilato - @dadoonet
Developer | Evangelist

Query DSL

```
GET _search
{
  "query": {
    "match": {
      "quote": "I know kung-fu"
    }
  }
}
```

Query DSL

```
GET _search
```

```
{  
  "aggs": {  
    "quotes-by-character": {  
      "terms": {  
        "field": "character"  
      }  
    }  
  }  
}
```





How many query languages use `_search`?



Query String

SQL

EQL

Query DSL

Vega

Timelion

Lucene

KQL



_search



A brief history of Elasticsearch's analytical capabilities





~17 months of development

Commit

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



ES|QL

the language

ES|QL Features

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

ES|QL commands

Source (From, Row)

Filter (Where)

Processing (Eval)

Aggregation (Stats)

TopN (Sort + Limit)

Expansion (Enrich , MV_Expand)

Extraction (Dissect, Grok)

75+ functions:

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



ES|QL

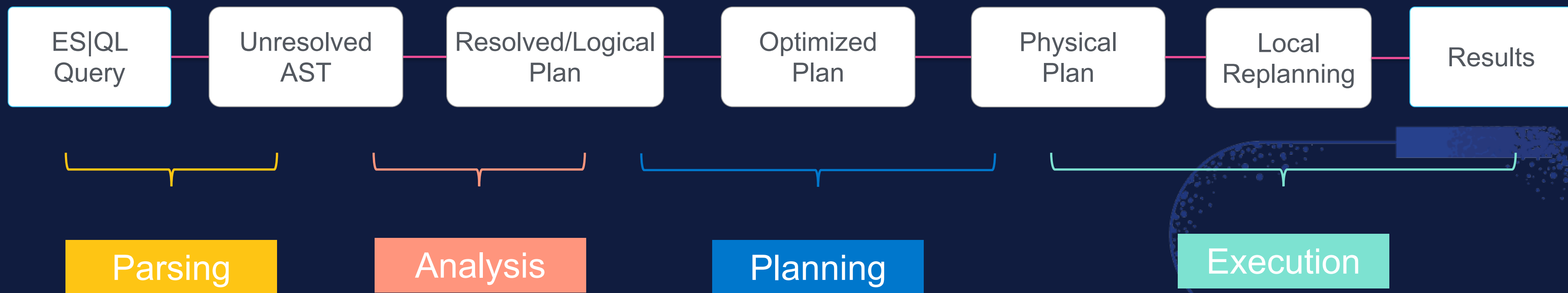
the engine

66

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.

Query planner

- ✓ Flexible distributed execution
- ✓ Allow multiple roundtrips



Compute engine

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

Vectorization

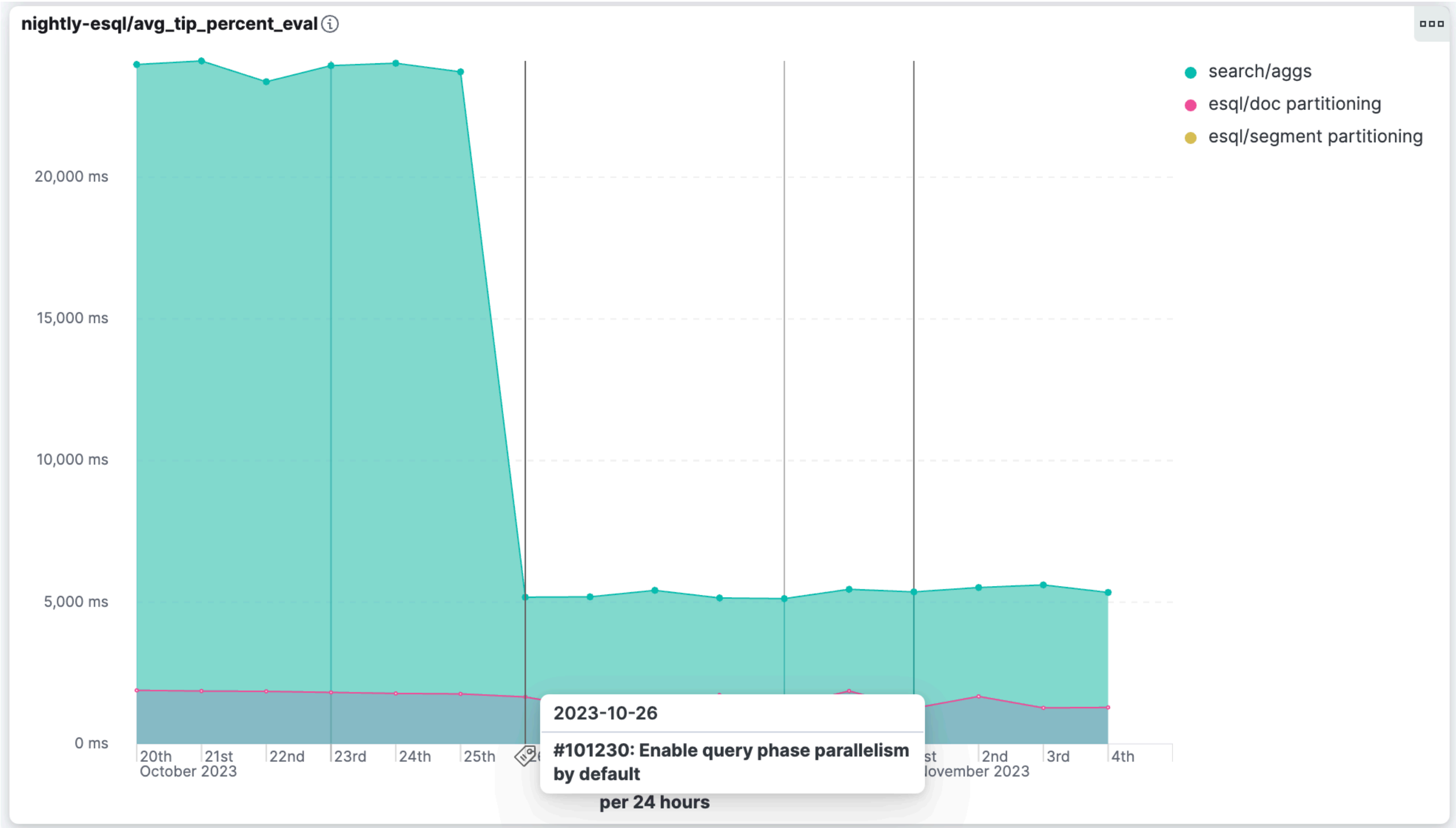
“convert from a scalar implementation, which processes a single pair of operands at a time, to a vector implementation, which processes one operation on multiple pairs of operands at once.”

```
for (i = 0; i < n; i++)  
    c[i] = a[i] + b[i];
```

https://en.wikipedia.org/wiki/Automatic_vectorization

Benchmarks

<https://elasticsearch-benchmarks.elastic.co/#tracks/esql/nightly/default/30d>





ES|QL

the visualization

ES|QL

in action

<https://github.com/dadoonet/esql-demo>



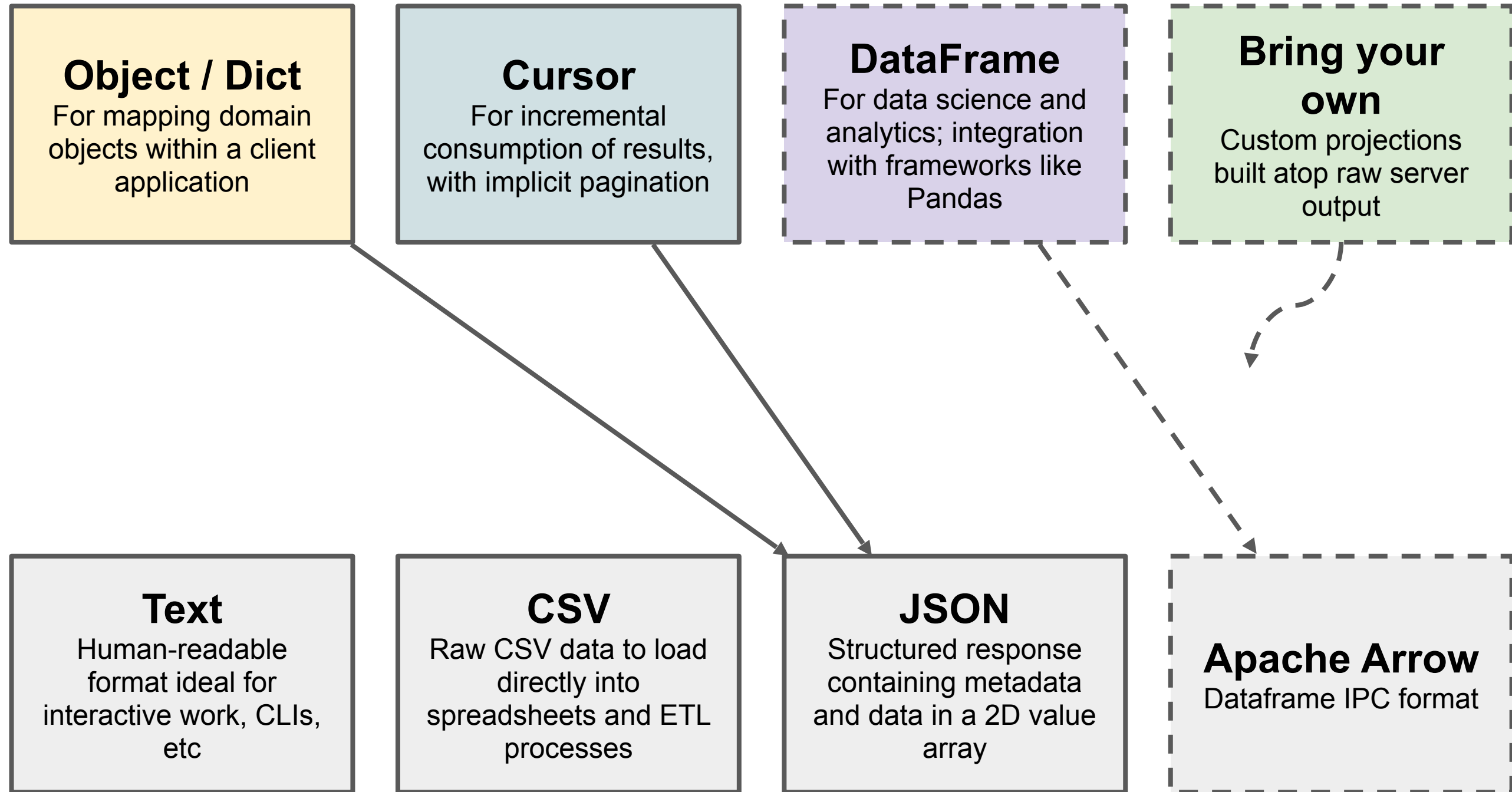
ES|QL



with (Java) client

Ways to consume ES | QL results

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



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

Object API

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

```
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());
}
```


ResultSet JDBC API

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

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

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



ES|QL

- Language
- Engine
- Visualization



Try it!

<https://esql.demo.elastic.co>



TAD

Elasticsearch Query Language

ES|QL

David Pilato - @dadoonet
Developer | Evangelist

