

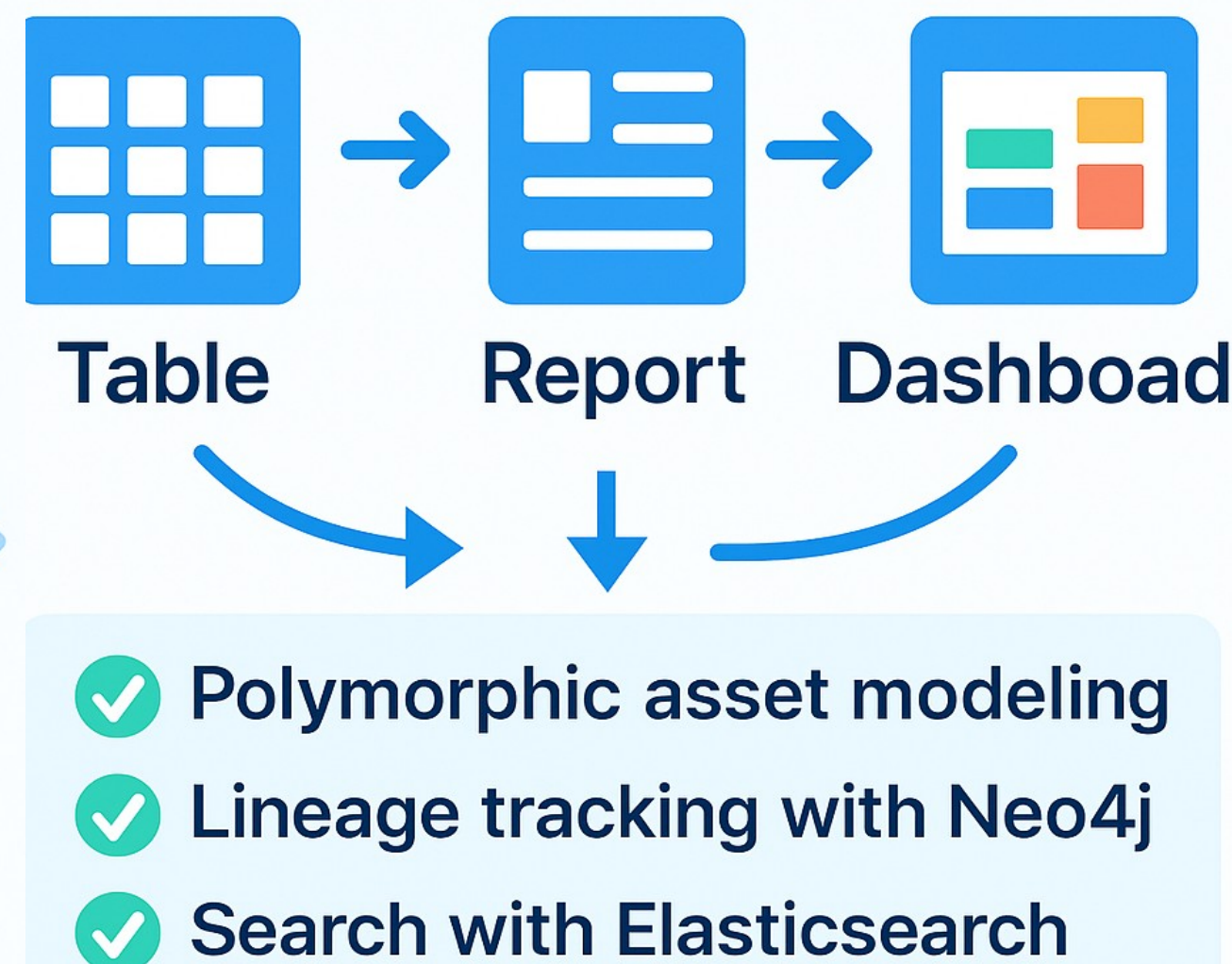
Project Description



The Problem

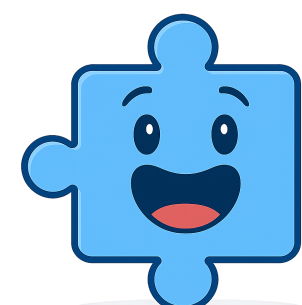


Our Solution



- **Problem:** Overwhelming amounts of **unstructured metadata** create **challenges** in organization and efficient data retrieval
- **Goal:** Streamline metadata ingestion and cataloging to **enhance discoverability, trust, and management** of data assets via a data lifecycle

Requirements & Testing



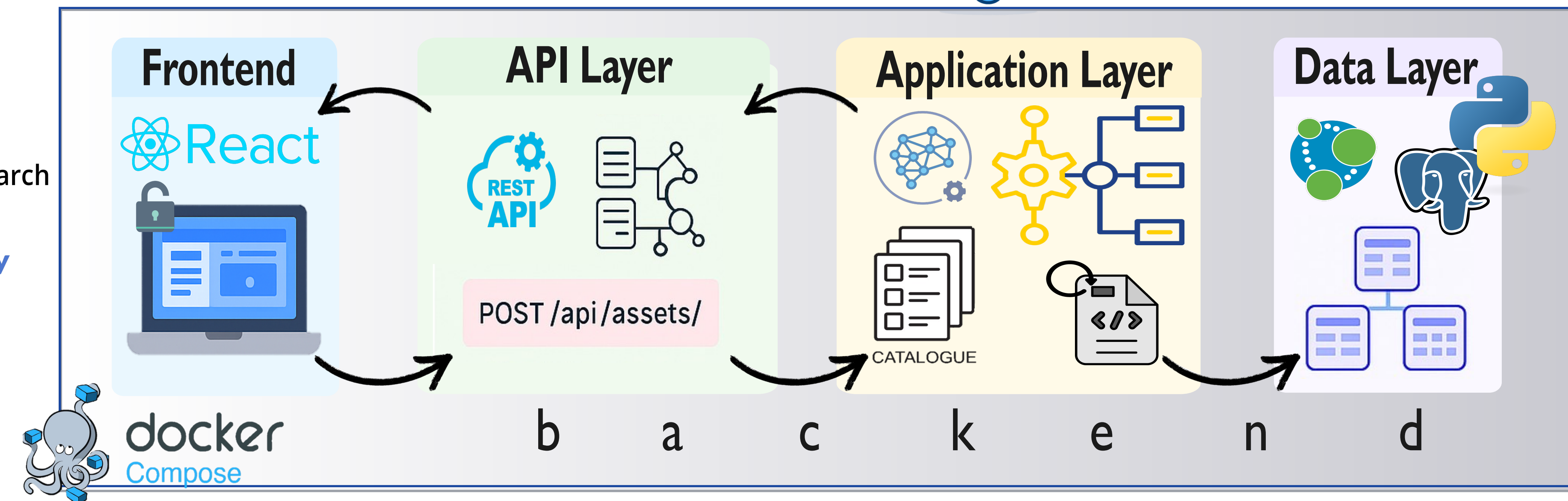
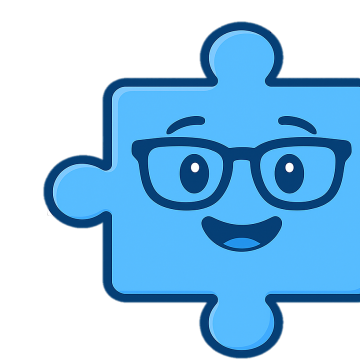
92%
92% Line Coverage

55%
31/56 System Tests Passing

100%
300/300 Unit Tests Passing

- ✓ **UC1:** User Management
- 📁 **UC2:** Metadata Ingestion
- 🧬 **UC3:** Blueprint Creation
- 🧠 **UC4:** Suggestion Engine
- 🔧 **UC5:** Manual Product Creation
- ♻️ **UC6:** Versioning
- 📁 **UC7:** CRUD Support
- 🤝 **UC8:** Shared Ownership
- 🔧 **UC9:** Algorithm Configuration

Design



- ★ neo4j
- ★ elasticsearch
- ★ NumPy
- ★ pandas

Ingestion & Assets



Goal: Ingest millions of metadata & generate assets in less than a minute

- Polymorphic asset creation
- Background metric processing

```
asset.get_real_instance()
    .calculate_metrics()
```



returns scoring values:

```
Recency: logarithmic
Data Completeness: Metadata
Data Quality: Asset Specific
```

- Relationship Tracking
- Bulk creation & Parallel processing

Data Products



Goal: Enable versioned, reusable packages of data assets—defined by blueprints—to solve real business problems with trust and governance.

- Publicly Available
- Shared Ownership
- Versioning



Data Lifecycle

Ranking Algorithm



Goal: develop a configurable algorithm that provides asset suggestions and ranks them according to relevance

- Elasticsearch for associations ★

```
query = {
  "bool" : {
    "must not": [{"terms": {"_id": existing_asset_ids }}],
    "should" : [
      "match": [{"name": {"query": term, "boost": 3.0}},
      "match": [{"description": {"query": term, "boost": 2.5}},
      "match": [{"metadata.algorithm": {"query": term, "boost": 1.8}},
      "match": [{"metadata.serviceType": {"query": term, "boost": 1.8}}
    ],
    "minimum_should_match": 1,
  },
}
```

- User defined configurations
Recency | Completeness | Data Quality + Flexibility | Lineage
- Asset Ranking Engine ★

```
score = np.dot(values, weights)
```

returns weighted average of configurations and weight assigned to each

- Caching Configurations