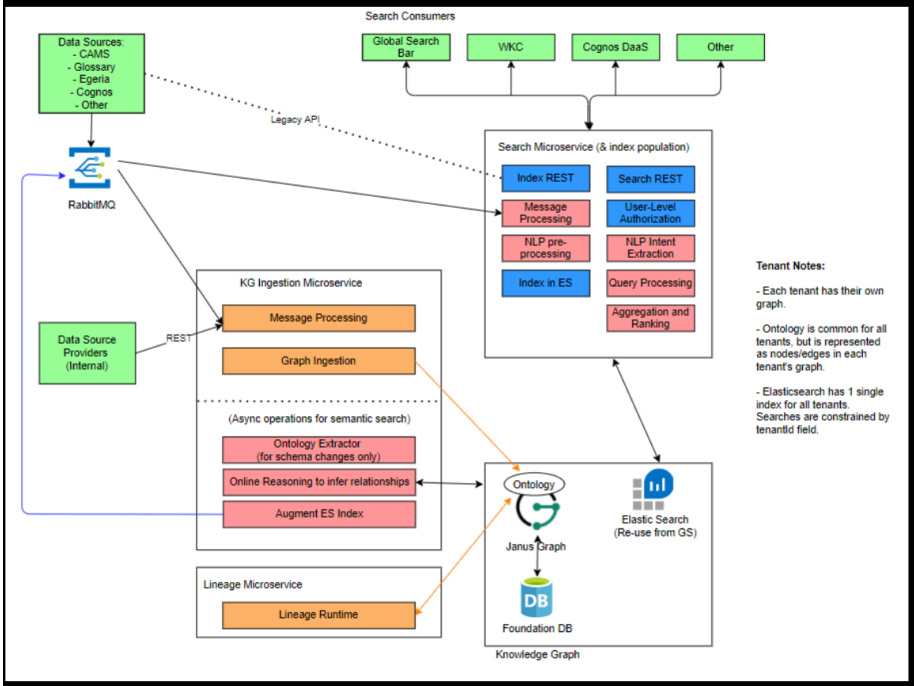
How does semantic search query in knowledge graph

Semantic search architecture

Getting Data into Knowledge Graph

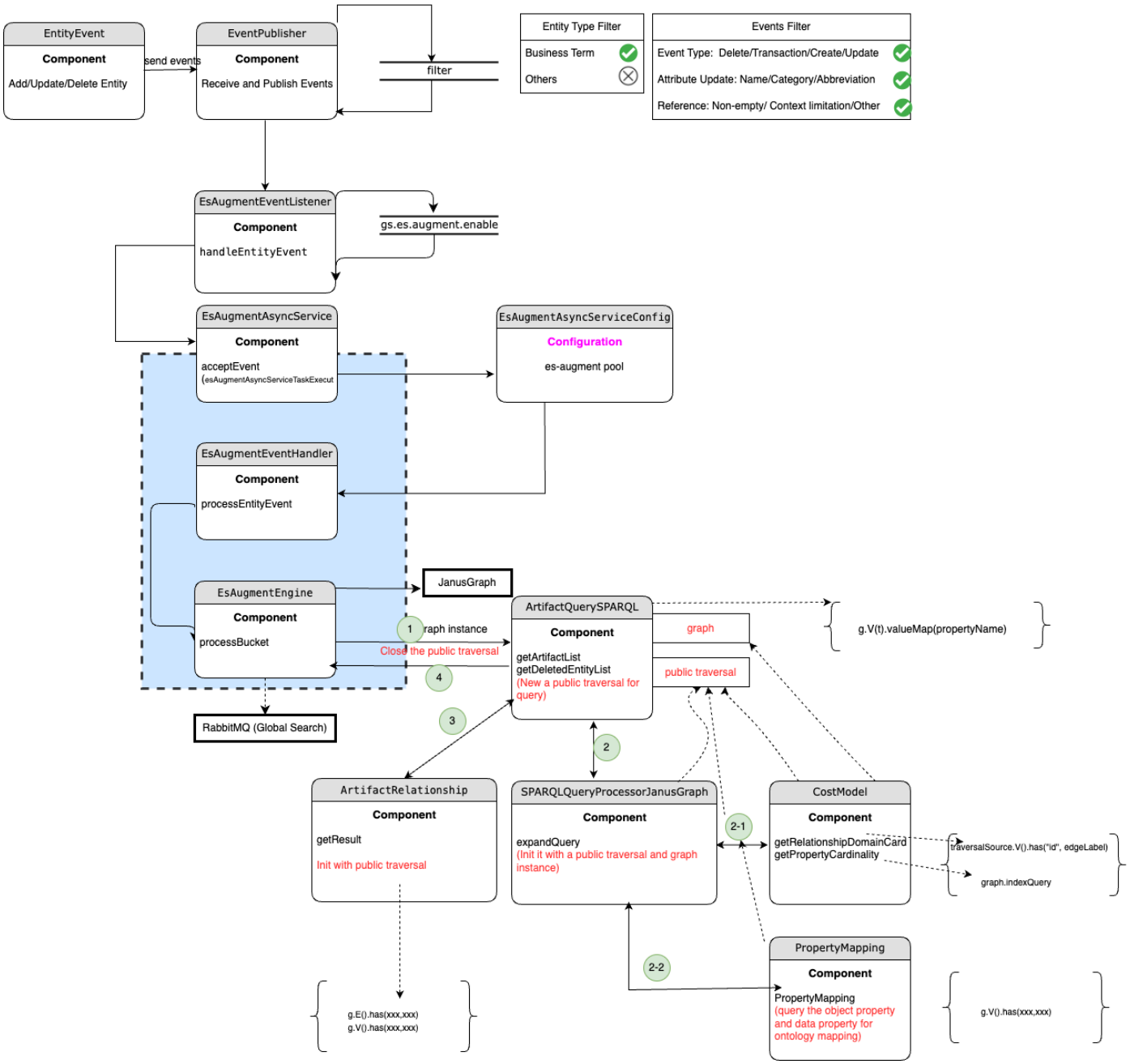


1. Data flows simultaneously into Global Search index and Knowledge Graph
2. Inferencing on Knowledge Graph is used to “augment” Global Search index with extra semantic information
3. Semantic search uses this semantic data at query time

Workflow

Before going to ERGO query, all the following filter conditions should be met at once.

1. Entity type is Business Term
2. Event type is Create_Update or Delete or Transaction
3. Entity info map is non-empty
4. Reference name is non-empty or context limitation or other



ERGO API signature

ERGO reasoning engine initializing

```
/**
 * ERGO reasoning engine initializing
 *
 * @param graph - JanusGraph instance
 * @param graphName - graph name
 * @return a reasoning engine instance
 * @throws ArtifactException
 */
public ArtifactQuerySPARQL(JanusGraph graph, String graphName) throws ArtifactException {}
```

Graph Query API

```
/**
 * Get a list of {@link Artifact} relationship from graph instance based on artifact id
list and type. It covers following scenarios:
 * 1. Create a single business term
 * 2. Create a relationship between two terms or term to data class
 * 3. Update a single business term's property, such as term name, description, etc.
 *
 * @param <T> generic type
 * @param ids list of Artifact id property. If it is null {@link NullPointerException} is
thrown
 * @param type artifact type, only support four types: TERM
 * @param edgeLabel relationship label, use to update the 2 hop artifact
 * @return a list of Artifact object. If there is no result, an empty list is returned.
 * @throws ArtifactException If ids is null or size=0 the exception is thrown
 */
public <T> List<Artifact> getArtifactList(List<T> ids, ArtifactType type, String
edgeLabel) throws ArtifactException {}
```

```
/**
 * Get {@link Artifact} and all relationship of the {@link Artifact} for deleted event. It
convers following scenarios:
 * 1. Delete a single business term
 * 2. Delete a relationship between two terms or term to data class
 *
 * @param a list of vertex id of deleted Artifact
 * @return a list of Artifact object.
 * @throws ArtifactException If ids is null or size=0 the exception is thrown
 */
public List<Artifact> getDeletedEntityList(List<Object> ids) throws ArtifactException {}
```

Reasoning based on edge relationship for business term only

ERGO performs reasoning on the graph using the ontology and uses SPARQL language to obtain a list of target objects based on a business term name and a set of predefined RDF. Subsequently the query result will be converted to an artifact data structure and then persist the data to the Elasticsearch semantic index.

Currently only focus on these edge relationships `is_a_type_of` , `context`, `related` and `synonym` for business term and `context` for category. In addition to 1 hop query in graph, it also queries 2 hops for edge relationship in/out of `is_a_type_of` and `related` in the graph.

these vertix properties `name`, `type`, `id`, `description`, `last_updated`, `abbreviations`, `alternative_names`, `artifact_id`, `last_updated` for business term and `name`, `type`, `producer_property` for category
The vertix name is mandatory element, others are optinal.

Semantic Index

```
{
  "tenant_id": "string",
  "name": "string",
  "aliases": ["string"],
  "artifact_type": "term",
  "equivalent_artifact_names": ["string"],
  "equivalent_artifact_ids": ["string"],
  "parent_artifact_names": ["string"],
  "parent_artifact_ids": ["string"],
  "grandparent_artifact_names": ["string"],
  "grandparent_artifact_ids": ["string"],
  "child_artifact_names": ["string"],
  "child_artifact_ids": ["string"],
  "related_artifact_names": ["string"],
  "related_artifact_ids": ["string"],
  "loosling_related_artifact_names": ["string"],
  "loosling_related_artifact_ids": ["string"]
}
```


Use cases

1. Add new relationship between both business terms
2. Remove relationship from one business term
3. Remove the business term
4. Update business term's name, or description, or abbreviations.