ONTOTEXT ANSWER:

There can be a bit of confusion with how Kafka interacts with GraphDB. There is the Kafka Connector, which is used to stream RDF data from GraphDB to the message bus as JSON. Then, there is the Kafka Sink, which does the inverse, producing RDF from JSON messages.

The process for updating data with Kafka is called SPARQL templates. SPARQL templates define a layer that may modify the list of provided triples. You can write a SPARQL template that is triggered upon a Kafka message being processed. The various fields that are part of the JSON message will then be injected inside the SPARQL update.

The Sink connector is the frontend for using the Smart templates. It is an open source project that Ontotext hosts on GitHub. It is used to bridge between Kafka and GraphDB. The service is separate from GraphDB, so you need to only install it if you use Kafka.

On the other hand, the Kafka connector is similar to the Elasticsearch connector. It allows the mapping of RDF data to JSON documents according to a predefined set of rules:

- Fields defined by name, property chain and optional modifiers;
- Nested document support: creates nested JSON documents as values;
- Filtering;
- Type filter;
- Literal languages filter;
- Entity filter;
- Multiple instances of a connector can be defined and coexist at the same time;
- It supports the repair operation that is equivalent to dropping and recreating the connector with the same settings.

Some of our customers combine both methods!

Imagine the following scenario:

1. A third party data source, for example an automated ETL pipeline produces some triples directly to GraphDB. It is not managed by you, so you can’t really modify the triples before they are ingested.

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

2. Some of these triples are passed into Kafka for further enrichment.
3. A client that is a Kafka consumer reads the JSON representation of the triples and performs some business logic on it.
4. The output of the business logic is fed into a Kafka topic that triggers the smart update to enrich the original triples.

Step 3 can be skipped if all processing that has to be done on the incoming data can be carried out with SPARQL. We have given an example of this capability within our recent data linking blog post. Utilizing Kafka as a message bus, you can have a transactional ETL system that is fully automated and resilient.

Help us get better: review GraphDB on Gartner

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com