Exploring Semantic coupling in an event driven architecture

Data Engineering Internship

Ezechiël Syx

05/10/2022

INTRODUCTION

Event driven architecture allows us to build loosely coupled applications. In our case a cloud platform that communicates bi-directionaly with IOT devices. The idea is that each of the software components communicate asynchronously by emitting and reacting to events.

Our current message schema provides the following information:

- What is the producer of the message
- When was the message created
- A random unique identifier for each message
- Payload data
- The type of the payload, describing its schema.

The key difference between our message schema and the cloud native computing foundation's cloudevents spec is that cloud event spec adds an optional uri referencing a formal schema definition for the data payload.

Semantic web technology is suited to define that formal schema definition and allow the possibility of collecting and making these definitions.

Even when using a micro service design, as the data events pass through different components and concepts are reused, these services become semantically coupled. We only have around 50 of these event types and only a handful of dataflows, but this is expected to grow and change over the years, we currently lack a good way to reason about this coupling.

OUR PROPOSAL

1. Define schemas for the existing events. Automatic schema registration at software build time can be explored, but static analysis of types without a matching schema definition is sufficient.

2. Reuse existing semantic ontologies or model a new one to describe core platform concepts, customers, assets, events, software components. This requires a good level of abstraction, for example software components should not be modeled as their own specifics but merely grouped together as either consumers or producers of events. The goal is being able to express dataflows.

3. This data can then be materialized to a graph database like neo4j where we can run queries and visualize results.

Who are we looking for

Someone that is passionate about micro services, semantic web technologies and is looking for a challenge beyond a simple CRUD application. Our proposal doesn't include a set use case to solve and thus has open ended evaluation criteria. An example use case would be: List event types that will become deprecated (has no consumers) and thus can be considered for deletion in a future update. We would expect you to work out at least one use case. We are looking for someone who can work mostly independently.

Experience required

- A proficient understanding of a scripting language with good libraries for RDF, either Python or JavaScript
- Some familiarity with writing web applications.
- Experience with querying a graph database or SPARQL

What we offer

An introduction to modern devops practices, we use Flux for gitops. Our platform runs entirely on Kubernetes and we give workshops on how to leverage kubernetes during software development time. Training material is publicly available at [Skaffold helm tutorial.]

The internship can be done remotely or from our office in Brussels or a hybrid combination.

The internship work itself will offer a lot of freedom in making tech choices.

Recommended reading list

Some links of people doing similar things, that may serve as inspiration for possible use cases

https://www.airpair.com/logging/posts/understanding-our-app-with-neo4j

https://aiven.io/blog/metadata-parser

We love to hear from you!

Like this mission? And think we're looking for you? Drop us a line and your CV at jobs@kapernikov.com.