

# What does the system integration toolbox anno 2019 look like?

Program - TechEvent, Copenhagen, October 24th  
Venue - Kosmopol, Fiolstræde 44 (close to Nørreport station)  
Registration - <https://techevent.nemtilmeld.dk/>

08:30 Registration and breakfast

09:00 Welcome

09:15 ***Cloud-native integration with Apache Camel on Kubernetes***

## **Abstract**

Cloud-native applications of the future will consist of hybrid workloads: stateful applications, batch jobs, microservices, and functions, wrapped as Linux containers and deployed via Kubernetes on any cloud.

In this session, we will explore key challenges with function interactions and coordination, addressing these problems using Enterprise Integration Patterns (EIP) and modern approaches with the latest innovations from the Apache Camel community:

- Apache Camel 3
- Camel K
- Camel Quarkus

Apache Camel is the swizz army knife of integration, and the most powerful integration framework. In this session you will hear about the latest features in the brand new 3rd generation.

Camel K, is a lightweight integration platform that enables Enterprise Integration Patterns to be used natively on any Kubernetes cluster.

When used in combination with Knative, a framework that adds serverless building blocks to Kubernetes, and the subatomic execution environment of Quarkus, Camel K can mix serverless features such as auto-scaling, scaling to zero, and event-based communication with the outstanding integration capabilities of Apache Camel.

We will show how Camel K works. We'll also use examples to demonstrate how Camel K makes it easier to connect to cloud services or enterprise applications using some of the 300 components that Camel provides.

**Speaker (Author Bio)**

Claus Ibsen, a Red Hat senior principal software engineer, has been working with open source integration software for a decade. He works full time on the popular Apache Camel project and on Red Hat® Fuse. Claus is a frequent speaker at conferences about Apache Camel, integration, and recently about microservices and Kubernetes. He recently finished work on the 2nd edition of the Camel in Action book. Claus is a Java Champion and ASF member.

10:45 Break

11:00 ***Asynkron kommunikation/EDA***

What is an Event Driven Architecture and what can it do for your architecture?

In this talk we will see what events are and how they can solve many of the coupling and integration problems that you will experience in a distributed (micro)service landscape. We will get into the different usages for events and showcase them based on real world projects.

After this talk you will know what events are, how they can be used and some of the requirements they put on your design and the way you thinking about software design.

**Speaker**

Jeppe Cramon - <https://www.linkedin.com/in/jeppeccramon/>

12:00 Lunch

12:45 ***How-To Architect A Cloud Native, Event-Driven and Reactive Application Environment***

**Abstract**

Developing applications that are reactive, event-driven, and polyglot are all requirements for today's application architecture. While cloud infrastructure and container Kubernetes solutions, such as Red Hat OpenStack Platform and Red Hat OpenShift, provide a robust infrastructure foundation for distributed environments, similar seamless application services require building applications that take full advantage of such an infrastructure.

In this session, we will have a live demo on how Red Hat's distributed application environment architecture is laid out and provide a blueprint for creating applications of the future at scale and delivers all the elements necessary for success. Attendees will be able to participate in the demo using their mobile phones or browsers and help in saving lives of 100s needing rescue.

This demo is based on Quarkus, Vertx, Microprofile, Spring, Node, Kafka via AMQ Streams, Process Automation Manager (PAM), Decision Manager (DM), Red Hat Data Grid (RDG), Single Sign On (SSO), Prometheus, Grafana, and much more.

**Speaker**

Syed M. Shaaf – Senior Principal Product Marketing Manager and Technical Evangelist with Red Hat

13:45 Break

14:00	Track 1	Track 2
14:00 – 14:30	<p><b>Use Case 1: Scania's experience with using Kafka</b></p> <p>How Kafka can be a key player in an Industry 4.0 journey. We will look at the different roles Kafka can play to enable a vehicle manufacturers Industry 4.0 journey in a multitude of scenarios ranging from replacing outdated integration solutions, offering the backbone for IoT EDA driven solutions as well as enabling analytics and linked data solutions.</p> <p><b>Speaker</b> Benneth Christiansson – Architect and advisor API Management - Redpill Linpro</p>	<p><b>Workshop v. Syed M. Shaaf</b> <b>From zero to (micro-) hero with Vert.x and OpenShift</b></p> <p>Reactive systems allows multiple individual applications to work as a single unit, reacting to its environment and events, while remaining aware of each other—the distributed nature helps to scale up/down, load balancing, etc.</p> <p>It's possible to write a single application in a reactive style (i.e. using reactive programming); however, that's merely one piece of the puzzle.</p> <p>When coming to microservices, event-driven, asynchronous and reactive are quickly mentioned to implement them right. It avoids building distributed monolith. In addition, in order to keep everything on track, you need a way to package and manage them. OpenShift is a container platform, based on Kubernetes, able to build, deploy, manage and update your microservices.</p>
14:30 – 15:00	<p><b>Use Case 2: Kafka and Event-Driven at BEC - matching an alien concept to well-known content</b></p> <p>At BEC, we have recently embarked on a journey towards event-driven architecture with Kafka as the platform. But what does that mean exactly?</p> <p>Most are aware that Kafka is a different beast in infrastructure platform terms - but how do you figure out what to put into Kafka?</p> <p>Hear about BEC's experiences on the journey so far</p> <p><b>Speaker</b> Brian Skovgren Klausen, Principal System Architect · Architecture and Platforms – Strategy and Referrals</p>	<p>Eclipse Vert.x is a toolkit to create reactive distributed applications running on the top of the Java Virtual Machine. Vert.x exhibits very good performances, and a very simple and small API based on the asynchronous, non-blocking development model. With vert.x, you can developed microservices in Java, but also in JavaScript, Groovy, Ruby and Ceylon. Vert.x also lets you interact with Node.JS, .NET or C applications. Vert.x is a container-native runtime taking care of the efficient usage of your CPU and memory granted to your container.</p>
15:00	Break	
15:30 – 16:00	<p><b>Use Case 3: Tryg's experience in using Kafka in an alternative way</b></p> <p>Since May 2018 I have been working on an enterprise data hub project at Tryg, one of Denmark's largest insurance companies. The goal of the project was to use change data capture (CDC) to capture</p>	<p>This lab is an introduction to microservice development using Vert.x and OpenShift. The application is a fake trading application, and maybe you are going to become (virtually) rich! The applications is a federation of interacting microservices running on OpenShift.</p> <p><b>Remember to bring your own PC with browser installed</b></p>

changes (events) in Tryg's many internal applications and store them in an external system (Kafka), so that Tryg could keep a realtime, full history of changes to important business objects.

With a live stream of changes (events) Tryg would be able to write applications that reacts to the events as they happen. With a full history of changes to important business objects Tryg could analyze what has happened in their organization, without having to get these features added to the internal applications. For instance, changes to customers in the CRM system tells them about how Tryg's customer portfolio has changed over time. Changes to employees and departments in the HR system tells Tryg how their employees and departments have changed over time. And changes to calls in their dialer systems tell which communication there has been between Tryg employees and customers. Changes to insurance claims in the claims system tell Tryg about what claims have been made, and which customers who have made them.

Since many of Tryg's internal applications where not designed to emit such change events, the project is using two mechanisms to capture these changes in these systems:

- 1) Using Attunity we listen for changes directly on the database commit log for the given application. This means, that we get all changes in the database tables as separate changes (events).

- 2) Using REST calls to SaaS applications to try to capture what has changed since last REST call.

The project is actually live and running today. However, the project was not without its challenges. In this talk I will dive into the architecture and solution in more detail, and explain what went well, and what challenges we ran into along the way. These challenges fall into the following categories.

- 1) Denormalization of business objects
- 2) Determining events
- 3) Avoiding duplicate events
- 4) Replay of events vs. the outside world
- 5) Tools

**Speaker**

Jakob Jenkov from Jenkov APS. Jenkov is a

tech company engaged in a variety of different projects in the tech and media space

16:00 – 16:30 ***Use Case 4: Speaker will be announced shortly***

16:30 – 17:00 Summary of the day, quiz and evaluation.

17:00 Thank you!