Conquering large scale visualization challenges with JavaFx and ELK

Working with large-scale automotive data models without a proper visualization support is quite cumbersome. There are a few frameworks which support handling of large dataset out of the box for example, Sirius, Klighd, and JavaFX. Another aspect is lay-outing the scenarios in a properly comprehensive manner. Previously, we had reported our experience of using Sirius framework in large-scale data. In this session, we would like to share our experience of using JavaFX in automotive data models, with focus on the performance and lay-outing of large scale data with Eclipse Lay outing Kernel (ELK).

The attendees can learn about handling of complex visualization scenarios with JavaFx and lay-outing techniques. This talk also explore the other side of JavaFX capabilities than just making fancy UIs. The following aspect shall be covered with examples

- We will discuss how to programmatically integrate JavaFX develop scalable visualization representations.
- We show the different scenarios where we used *custom* JavaFX lay-outing and *inbuilt* JavaFX lay-outing options.
- We reveal bottlenecks in JavaFX during handling huge models and how our engineers solved the issues.
- We reveal our recipe to seamlessly integrate ELK lay outing into JavaFX representations.
- We show how to configure ELK in handling multiple scenarios like border nodes.
- We discuss the possibility to attach model objects to JavaFX nodes.
- We share our secret to dynamic lay-outing using ELK in JavaFX.
- We reveal how ELK lay-outing makes the developer's life easy.

In short, if you are planning any sort of data visualization in your enterprise, this is your one stop for entire A-Z guidance.