Performance evaluation of BaSyx based Asset Administration Shells for Industry 4.0 Applications

Christian Sauer and Holger Eichelberger

{sauer,eichelberger}@sse.uni-hildesheim.de

Software Systems Engineering
University of Hildesheim

www.sse.uni-hildesheim.de
## Content

1. **Motivation** 3
2. Background 4
3. Contribution 5
4. Approach 6
5. Experiment and Results 7
6. Conclusion 13
7. Future Work 14
Motivation

• Interoperability and information exchange in IIoT/Industry 4.0.

• Currently several proprietary information models in different IIoT-platforms exist, increasing complexity and costs for IIoT/Industry 4.0 integration.

• Asset Administration Shell (AAS) is an upcoming information model standard for IIoT/Industry 4.0.

• BaSyx middleware for AAS: Exploring the capabilities and performance of BaSyx-based AAS, consideration for IIP-Ecosphere.
Background

Asset Administration Shell: Purpose and structure

Based on Source: ZVEI
Contribution

• We present an evaluation of the performance scalability of BaSyx-based AAS

• Evaluated BaSyx-based AAS:
  ▪ Local AAS
  ▪ Network-connected AAS
  ▪ Scale between 1 and 100 AAS
  ▪ Scale between 10 to 1000 AAS submodels
Evaluation of the performance scalability of BaSyx-based AAS

Based the initial examples of local and network-connected AAS from BaSyx we created 4 Experiments:

• 2 Experiments to evaluate local AAS:
  ▪ 1 local AAS with 10 to 1000 sensor-submodels
  ▪ 100 local AAS with 10 to 1000 sensor-submodels each

• 2 Experiments to evaluate network-connected AAS:
  ▪ 1 network-connected AAS with 10 to 1000 sensor-submodels
  ▪ 100 network-connected AAS with 10 to 1000 sensor-submodels each
Experiment and Results

• BaSyx state at January 2022

• Standard PC with an Intel(R) Core(TM) i7-8665U CPU 1.90GHz and 32 GB RAM

• OS: Windows 10 professional

• Eclipse IDE for Enterprise Java and Web Developers, version 2022-06 (4.24.0)

• Java Development Kit (JDK) version 15

• Apache-Tomcat Server version 9.0

• Compensated for fluctuations by performing each measurement 10 times
Experiment and Results

Experiments procedure:

• Measure system time to create first 1 then 100 AAS and their subsequent 10 to 1000 sensor-submodels.

• Always 11 iterations, start: 10 sensor-submodels, then 10 iterations from 100 to 1000 sensor-submodels.

• The measurement was as follows:
E1: 1 local AAS with 10 to 1000 sensor-submodels

- Time for generation 1 AAS and subsequently increasing number of sensor-submodels stayed level.
- Scaling the number of sensor-submodels for a single AAS had no significant performance impact.
Experiment and Results

E2: 100 local AAS with 10 to 1000 sensor-submodels each

- Time for generation of 100 AAS and subsequently increasing number of sensor-submodels increased, as expected.
- Scaling the number of sensor-submodels showed an almost linear increase, up to 100000 sensor-submodels
E3: 1 network-connected AAS with 10 to 1000 sensor-submodels

- Significant increase in effort for network-connected AAS
- Increase in effort prominent within the first 400 AAS
- More linear increase in effort past 400 AAS, similar to experiment 2
Experiment and Results

E4: 100 network-connected AAS 10 to 1000 sensor-submodels each

- 100-fold increase in scale: 100 network-connected AAS
- Scaling the number of sensor-submodels in the networked environment showed a very clear linear increase in effort.
Based on our experimental results we conclude that:

• The use of BaSyx-based AAS, passive as well as active AAS, scales well, even in larger numbers

• The application of BaSyx-based AAS is well suited for the anticipated industrial-scale dimensions of using AAS

• The application of BaSyx is very accessible

• Limitation:
  ▪ We used a limited set of artificial experiments
  ▪ We excluded deleting AAS or submodels

• Zenodo Link [https://zenodo.org/deposit/7180716](https://zenodo.org/deposit/7180716)
Future Work

• Explore the scalability of realworld AAS implemented with BaSyx, instead of artificial example AAS

• Explore the performance of BaSyx-based AAS within the IIP-Ecosphere platform

• Further performance evaluations of active AAS features