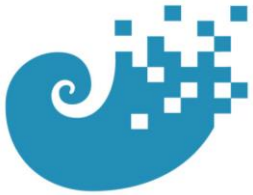
Industry 4.0 Resource Monitoring – Experiences with Micrometer and Asset Administration Shells

Miguel Gomez Casado, Holger Eichelberger

miguel.gomez@alumnos.uva.es
eichelberger@sse.uni-hildesheim.de

Software Systems Engineering
University of Hildesheim

www.sse.uni-hildesheim.de



IIP-Ecosphere

Next Level Ecosphere for
Intelligent Industrial Production

Motivation

- Ecosystem for Intelligent Industrial Production
- One activity: Virtual platform for AI in I4.0
- Focus / promises:
  - Interoperability
  - Vendor neutrality
  - Use of standards
  - Use of open source

IIP-Ecosphere

Next Level Ecosphere for Intelligent Industrial Production

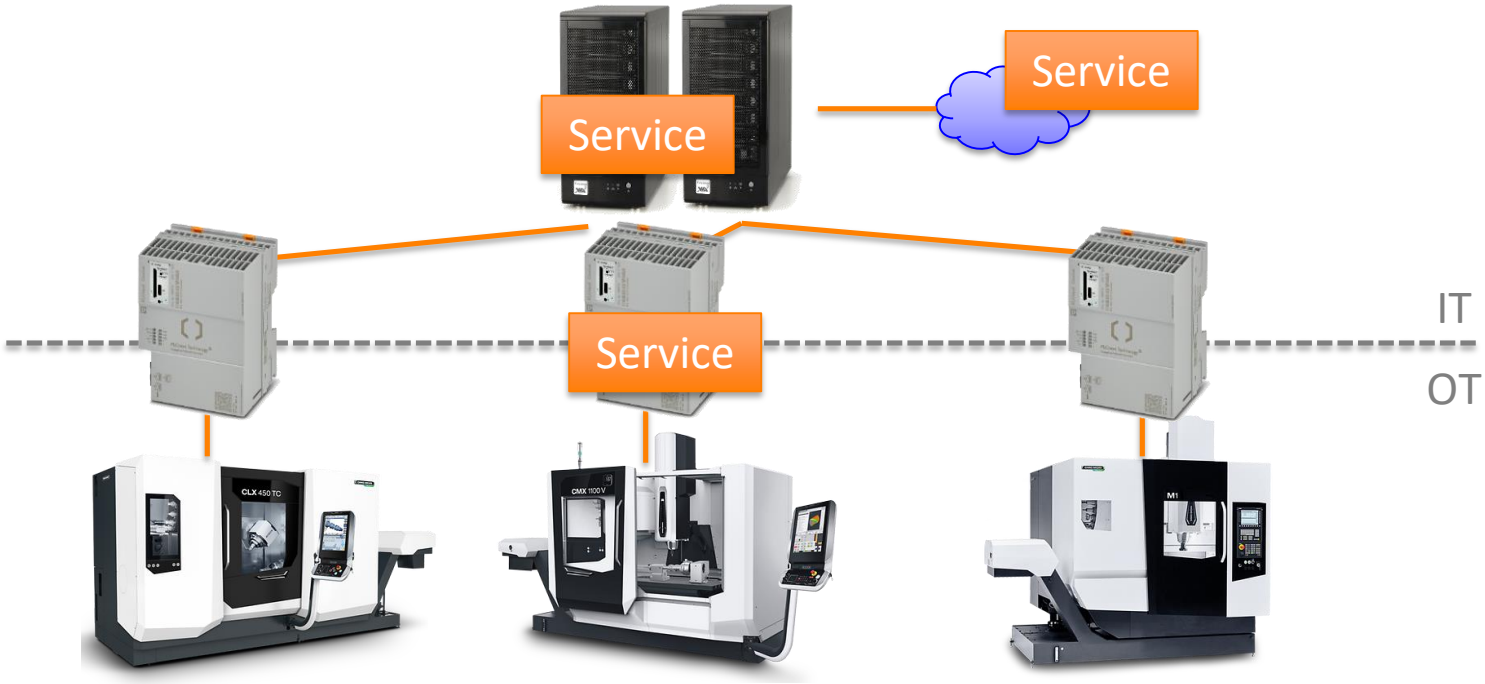
Gefördert durch:

Bundesministerium für Wirtschaft und Energie

AI-Innovation Challenge

Problem

I4.0 resource and service monitoring



How to design and realize efficient resource and service monitoring in an open, interoperable and vendor-neutral manner?

[Pictures: DMG MORI, PHOENIX CONTACT, CC BY]

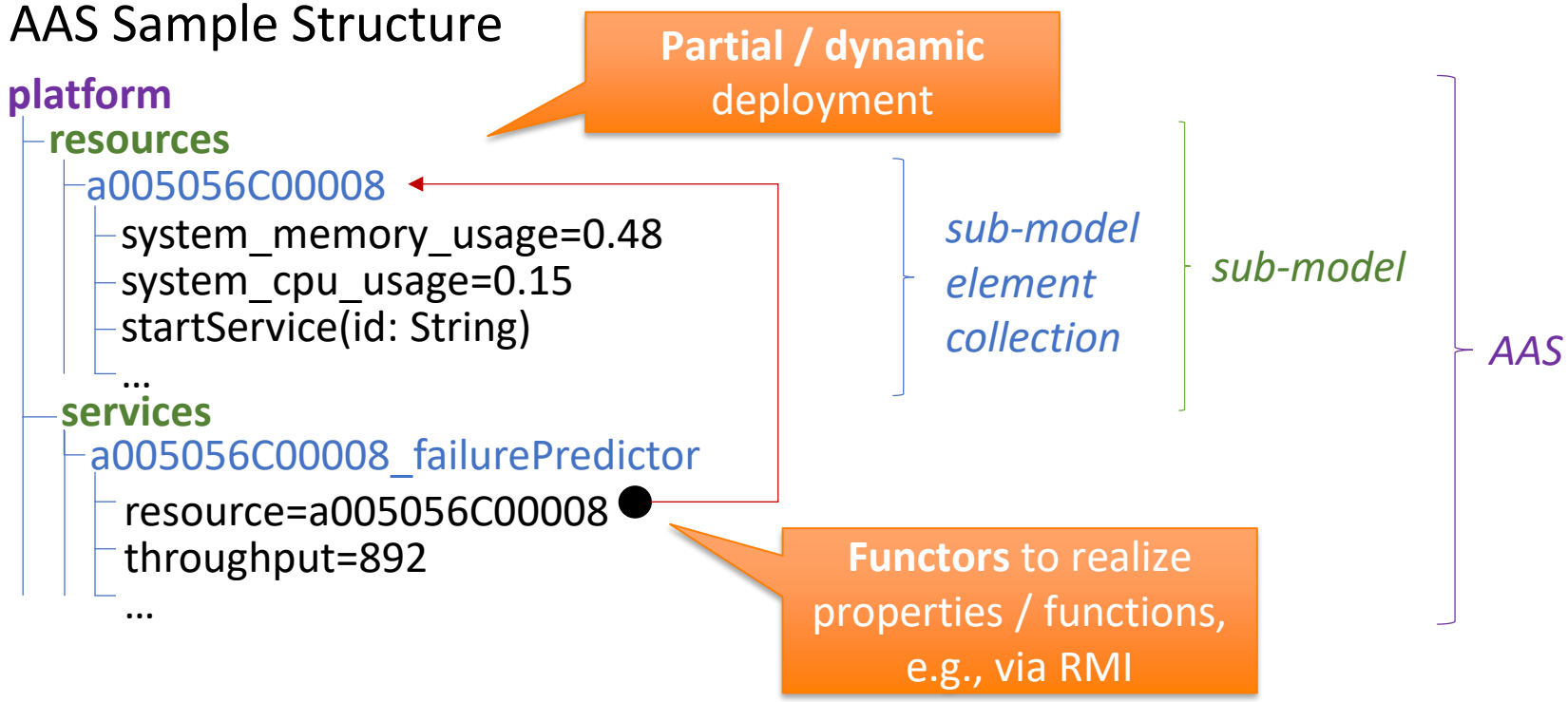
Standards?

- Protocols
  - Field level: PROFIBUS Modbus …
  - Transport level: MQTT AMQP (Advanced Message Queuing Protocol)

- Information models
  - OPC UA, Companion Specs
  - Asset Administration Shell (I4.0 Verwaltungsschale)
    - Asset: Product, Machine, …, digital twin
    - “Semantic” links against catalogues like eclass
    - Specific deployment options

IIP-Ecosphere

- Decisions in IIP-Ecosphere
  - Resource / Component / Service-Interfaces: AAS, Eclipse BaSyx
  - Transport abstraction: MQTT, AMQP, …
  - Monitoring abstraction: Micrometer, with remote proxies
- AAS Sample Structure



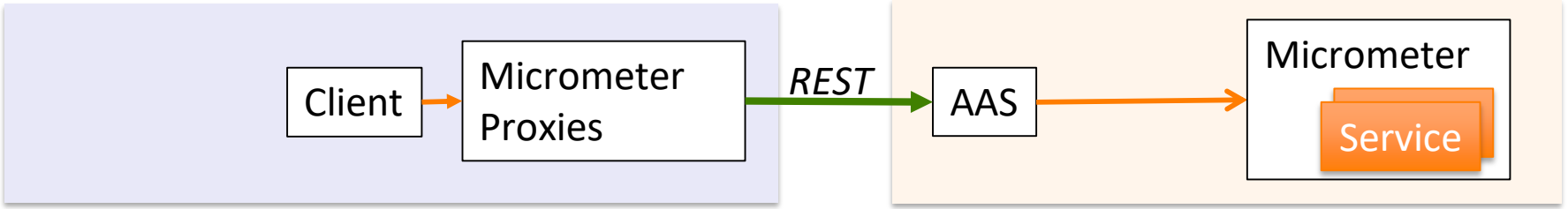
Integration patterns

Legend:
- local call
- remote call
- pub-sub

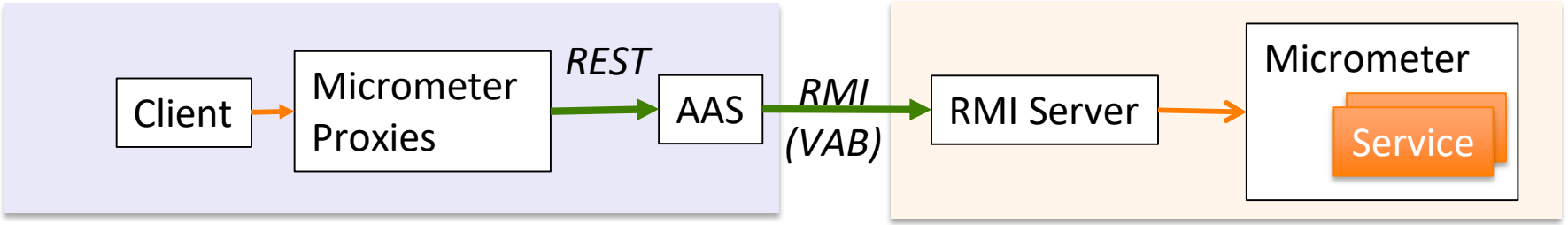
Inquirer, e.g., platform

Devices, e.g., edge

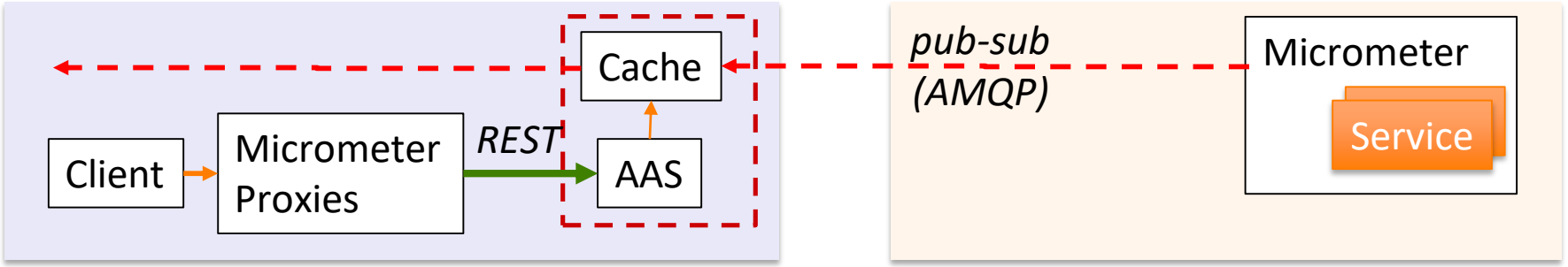
Local AAS



Remote AAS



Remote pub-sub AAS

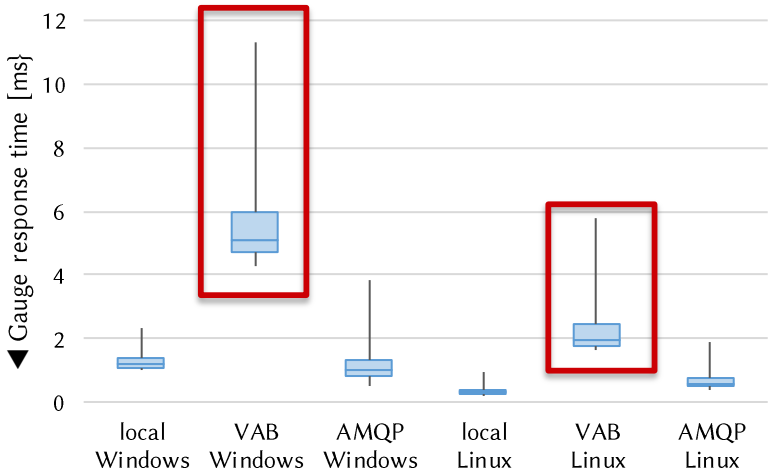


[Pictures: PHOENIX CONTACT, CC BY]

Results

- Setup:
  - Dell 7490 Windows, Ubuntu 20.04 VM on Xeon E5-2640v4
  - Local networking, Spring Cloud Stream services
  - One iteration reads 62 AAS properties, 200 iterations per variant

		Windows		Linux	
		avg [ms]	σ [ms]	avg [ms]	σ [ms]
Gauges	local a)	1,3	0,3	0,4	0,1
	VAB b)	5,5	1,1	2,2	0,7
	AMQP c)	1,2	0,6	0,7	0,3
Timer	local a)	0,8	0,3	0,4	0,2
	VAB b)	5,5	1,2	2,6	1,0
	AMQP c)	1,6	0,8	1,0	0,8
Counter	local a)	0,6	0,3	0,3	0,2
	VAB b)	3,6	0,9	1,7	0,8
	AMQP c)	1,1	0,5	0,6	0,3



- Limitations: BaSyx vs. REST, Hardware

Conclusions & Further Work

- Technical lessons
  - Micrometer Gauges are updated on request. Drop of factor 2-3
  - AAS REST uses temporary network connections. Windows runs out of resources.
- Integration patterns needed: OPC UA, AAS
- Interoperability
  - Field level: very diverse
  - Transport level: Payload
  - Information models: Structure
- Industrial feedback? Case studies?

Model-based integration

IIP-Ecosphere
https://iip-ecosphere.eu
@de_iipecosphere