

Observability of the SCADA Systems Using Elastic APM, Reactive Streams and Asynchronous Communication

Igor Khokhriakov, DESY
Olga Merkulova, DESY
Victoria Mazalova, CFEL
Alexander Nozik, JetBrains Research

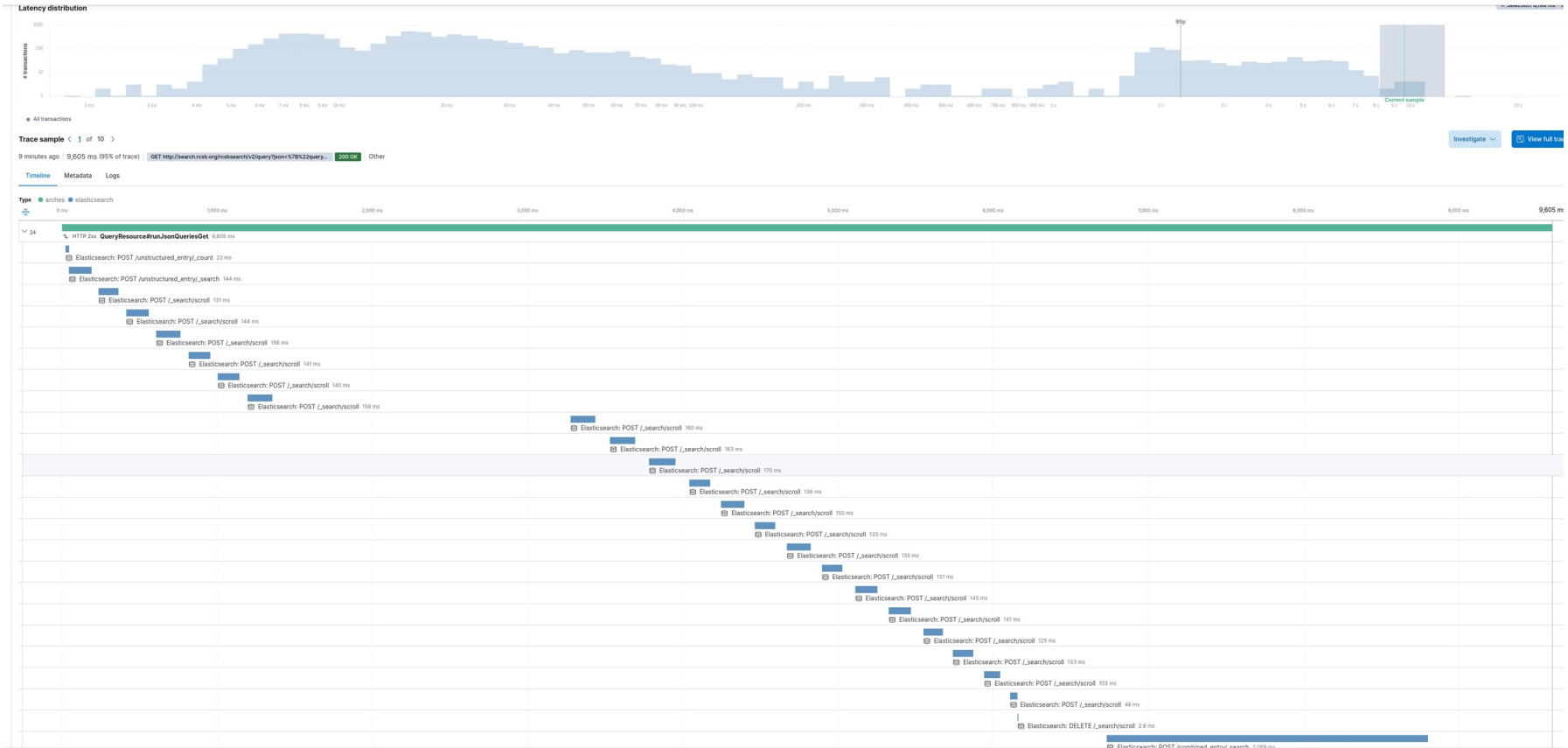
Who am I and why you should care?

- 15+ yrs in Software Development
- 10+ yrs in Scientific Software Development
- 3+ yrs in Tango-Controls Kernel development
- 1+ yrs in writing Tango RFC
- 500+ citations
- Event-driven systems architect
- Reactive programming advocate

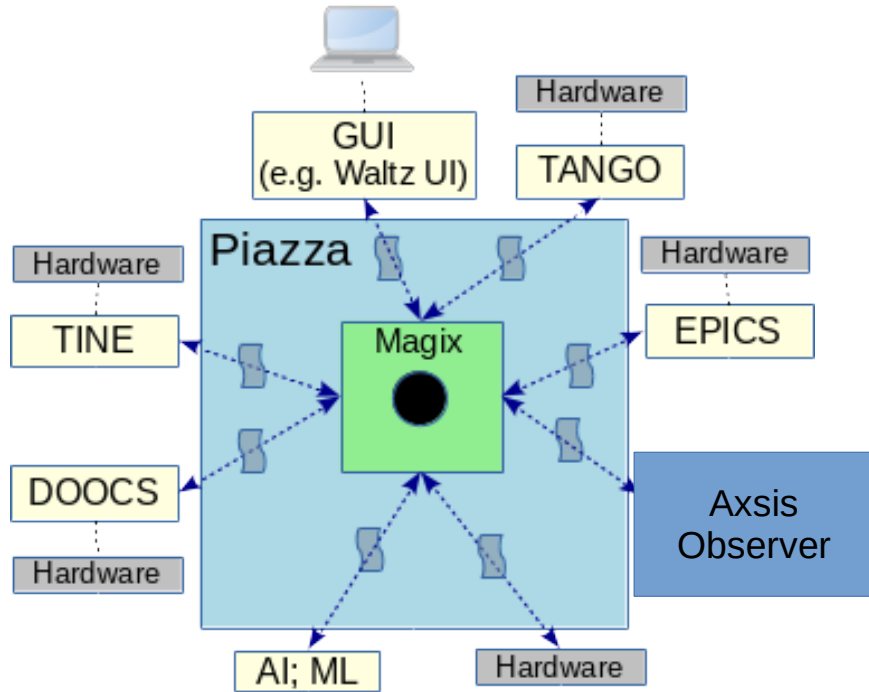
11 Oct 2023



Goal

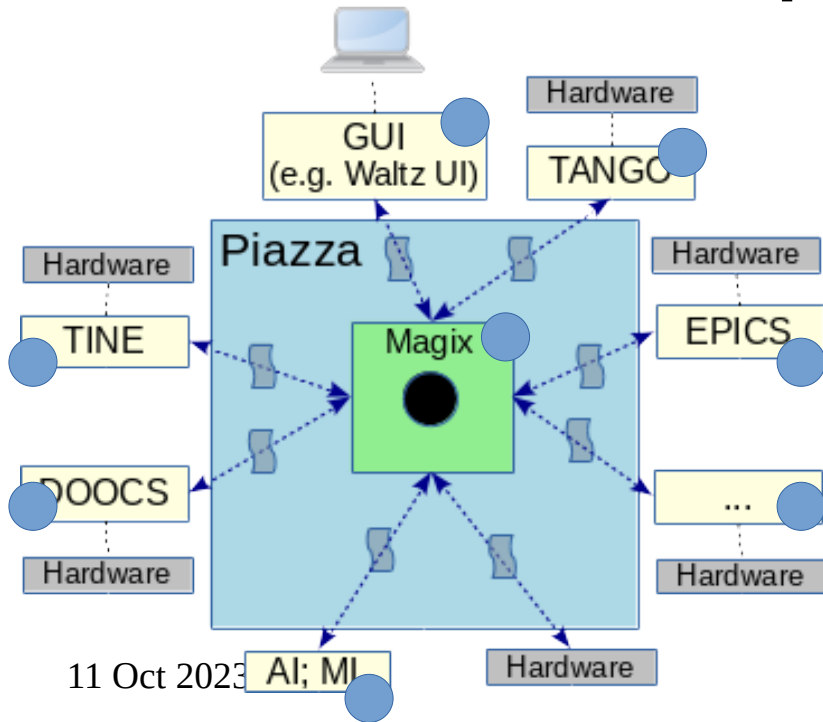


First approach – a dedicated micro-service



- **Language & Design:** Developed as a Java-based microservice.
- **Purpose:** Enhances the observability of SCADA systems.
- **Operation:**
 - Listens to various events and messages.
 - Operates on a designated channel.
 - Processes events and messages asynchronously.
- **Integration with Elastic APM:**
 - Monitors and traces system components
 - Provides real-time performance insights.

Second approach – injected APM agents into applications



● -- APM agents

An APM (Application Performance Management) agent is a software component designed to monitor and collect data on the performance of applications.

It tracks various metrics such as response times, error rates, and transaction volumes.

Java

- Straightforward using Java JVM agent:

```
java -javaagent:/path/to/elastic-apm-agent.jar \  
-Delastic.apm.service_name=my-application \  
-Delastic.apm.application_packages=org.example \  
-Delastic.apm.server_url=http://localhost:8200 \  
-jar your-application.jar
```



Python

```
import elasticapm
```

```
kApmServerUrl = os.getenv('APM_SERVER_HOST', 'http://localhost:8200')
```

```
kEnvironment = os.getenv('MODE', default='simulation')
```

```
kApmClient = elasticapm.Client(service_name='axis-magix', environment=kEnvironment,  
server_url=kApmServerUrl)
```

```
elasticapm.instrument()
```

NodeJs and JS client

- NodeJs:

```
import { init as initApm } from '@elastic/apm-rum'
```

```
const apm = initApm({
```

```
  // Set required service name (allowed characters: a-z, A-Z, 0-9, -, _, and space)
  serviceName: '',
```

```
  // Set custom APM Server URL (default: http://localhost:8200)
  serverUrl: 'http://localhost:8200',
```

```
  // Set service version (required for sourcemap feature)
  serviceVersion: ''
```

```
})
```

- JS:

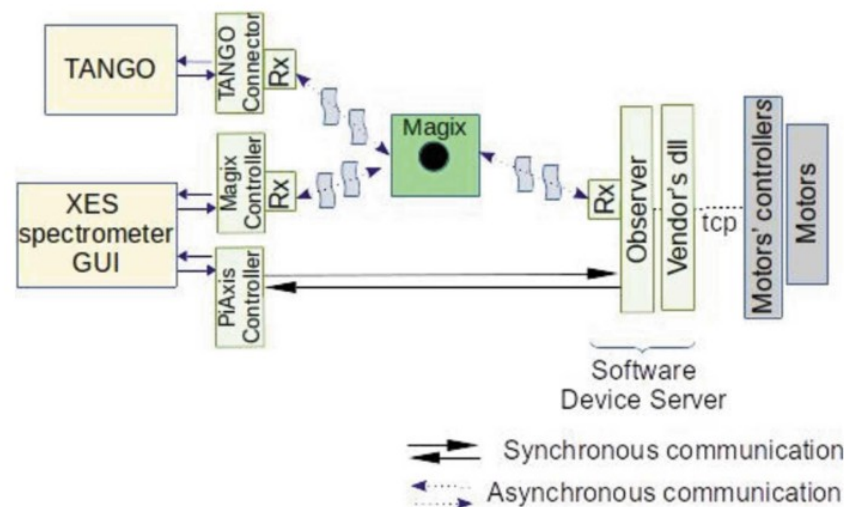
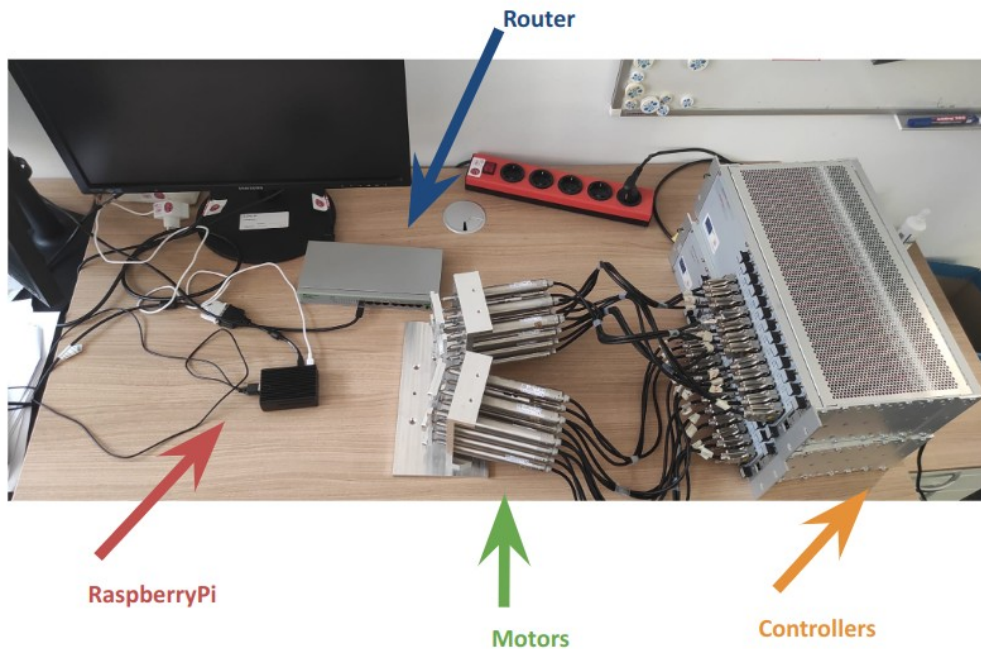
```
<script src="https://<your-cdn-host>.com/path/to/elastic-apm-rum.umd.min-<version>.js"
                                                crossorigin></script>
```

```
<script>
  elasticApm.init({
    serviceName: '<instrumented-app>',
    serverUrl: '<apm-server-url>',
```


Both have their pros and cons

- Dedicated micro-service easier to maintain
 - Not very suited for complex systems
- Embedded gives a better picture
 - Requires code ownership for non-JVM languages

Context



Khokhriakov, Merkulova, Mazalova, Nozik.

“A novel solution for controlling hardware components of accelerators and beamlines”

Volume 29 | Part 3 | May 2022 | | 10.1107/S1600577522002685

Summary

- Observability and Application Performance Monitoring are essential for complex long running application
- Easy to achieve using plugins and 3rd party libraries
- Infrastructure requires competences and maintenance
- Tango-Controls moves towards integration with OpenTelemetry out of the box



Acknowledgments



elastic



MicroK8s

**This work was supported by the consortium DAPHNE4NFDI in the context of the work of the NFDI e.V. The consortium is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project number 460248799*



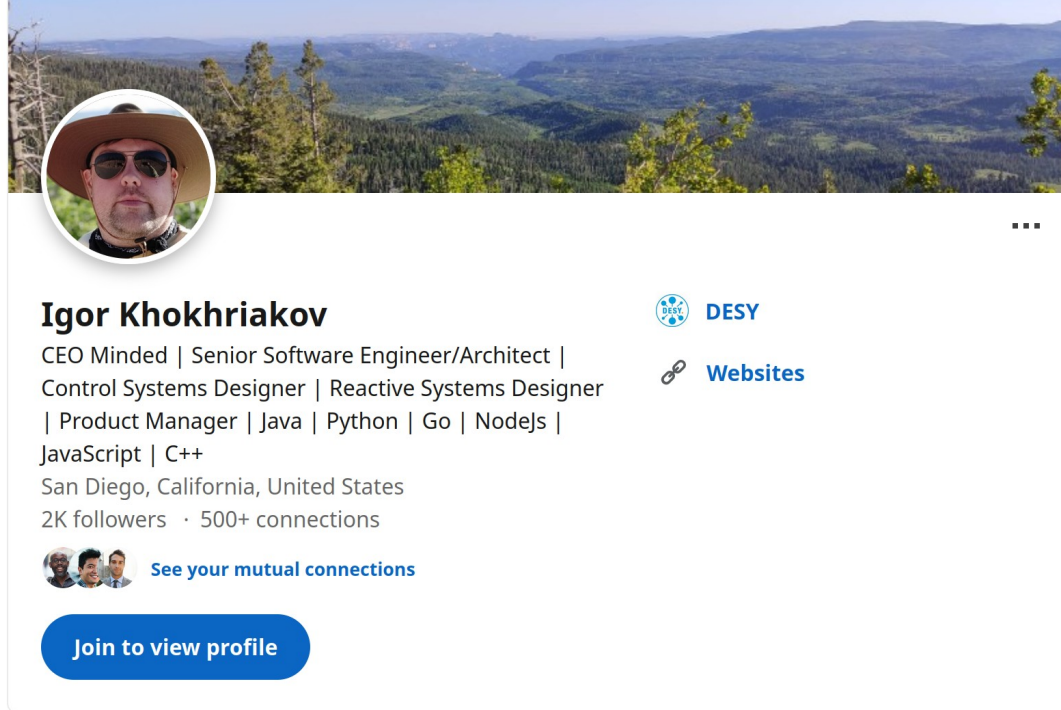


“Harnessing the Power of Ubuntu and MicroK8s:
Case Studies from San Diego Supercomputer Center and
Center for Free Electron Lasers”

11 Oct 2023

by Igor Khokhriakov 13

Join 2,5K+ followers on LinkedIn



The image shows a LinkedIn profile card for Igor Khokhriakov. At the top is a large landscape photo of a valley with mountains. Below it is a circular profile picture of a man wearing a hat and sunglasses. To the right of the profile picture is a three-dot menu icon. Below the profile picture, the name "Igor Khokhriakov" is displayed in bold. Underneath the name is a list of job titles and skills: "CEO Minded | Senior Software Engineer/Architect | Control Systems Designer | Reactive Systems Designer | Product Manager | Java | Python | Go | NodeJs | JavaScript | C++". Below this list is the location "San Diego, California, United States" and the follower/connection counts "2K followers · 500+ connections". To the right of the name and job titles, there are two icons: a globe icon labeled "DESY" and a key icon labeled "Websites". Below the job titles and location, there is a small circular icon with three people and the text "See your mutual connections". At the bottom of the profile card is a blue button with the text "Join to view profile".

Igor Khokhriakov
CEO Minded | Senior Software Engineer/Architect |
Control Systems Designer | Reactive Systems Designer
| Product Manager | Java | Python | Go | NodeJs |
JavaScript | C++
San Diego, California, United States
2K followers · 500+ connections

[See your mutual connections](#)

[Join to view profile](#)

About

Expert in multiple areas of software development, from web development to low-level C++, with an ability to see the big picture when it comes to software architecture and design.

Versatile and well-rounded software developer who is capable of delivering high-quality, end-to-end solutions. Can bring a holistic understanding of the software development process to

Thank you!

Questions?

igor.khokhriakov@desy.de

ingvord.mail@gmail.com