



# At the Edge of Robotic Applications

Conf42 IoT 2024



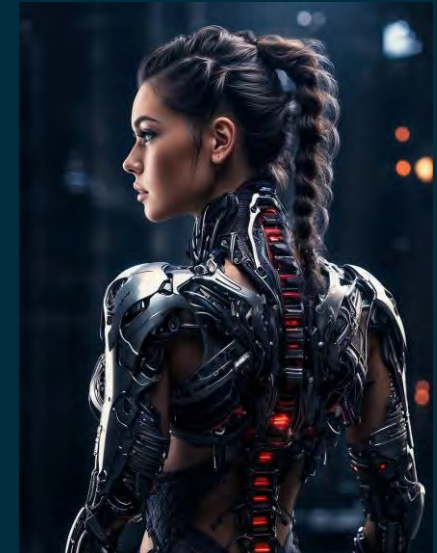
# Imagine a Robot



**Humanoid**



**Droid**



**Exoskeleton**

# The Most Common Robots



**Industrial robots:  
articulated six-axis arm,  
DELTA robot arm**

# A Few Stats About Industrial Robots

- In 2023:
  - Robots installed worldwide: 541,302
  - Robot density: 162 units/10,000 employees
- Where are the robots?
  - Automotive industry installed 25% of the robots
  - Electronics industry accounted for 23%

Source: IFR. (2024). World Robotics 2024 – Industrial Robots

# Robots Rarely See a Cloud





# Why Should Robots Send Data to the Cloud?

- **Condition Monitoring**
  - Proactive Maintenance: Remote Diagnostics & Troubleshooting
  - Accessibility & Availability
- **Data-Driven Insights**
  - Predictive Maintenance
  - Remote Control
  - Efficiencies & Potential Improvements

# Why AREN'T Robots Sending Data to the Cloud?

- **Too many data, too quickly**
  - Speed
  - Network Bandwidth
  - Cost
- **Can we trust the Cloud?**
  - Cybersecurity & Visibility
  - Latency
  - Standardization

# How to Connect Robots to the Cloud

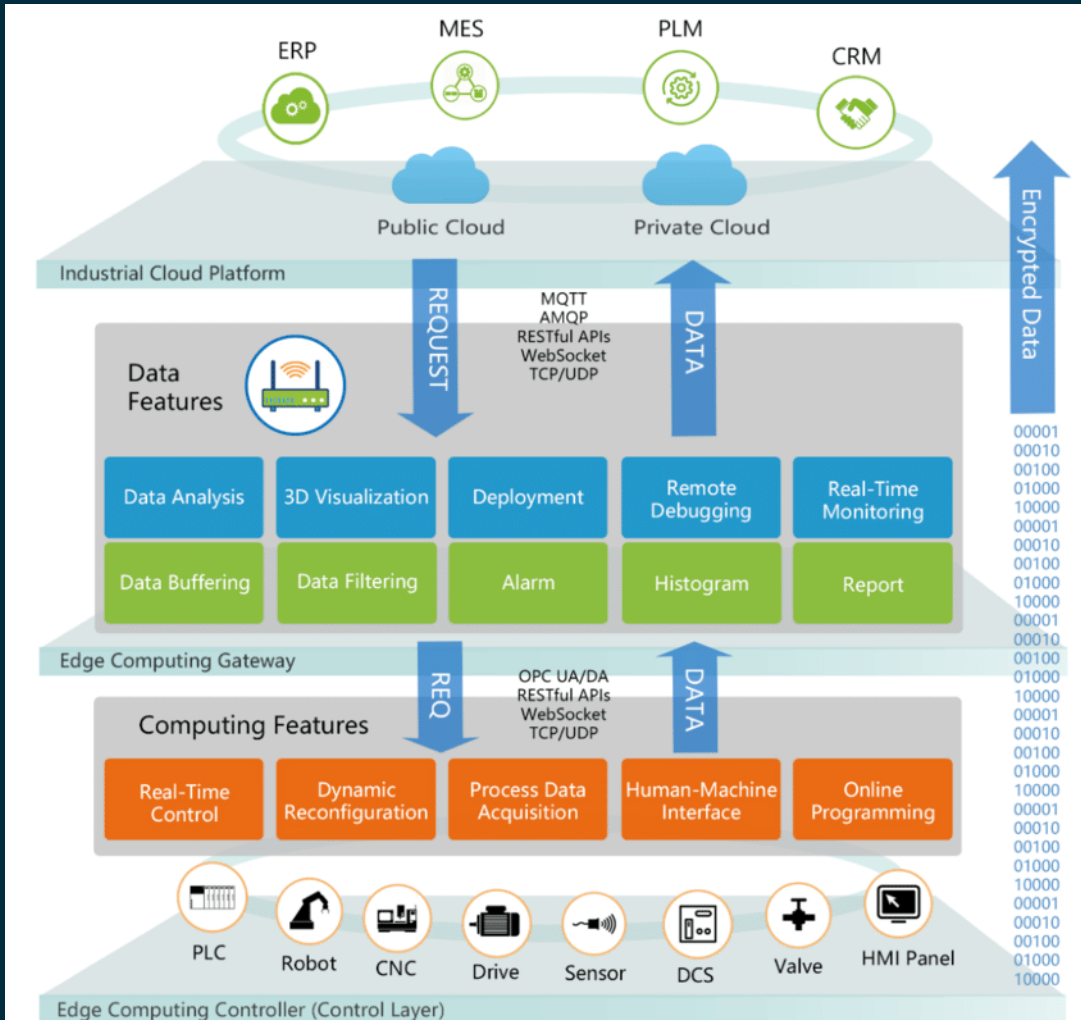
- **Edge Computing**: processing power at the periphery of industrial automation networks
  - Data collection, filtering and pre-processing
  - Speed & timely feedback to shop floor
  - Resilience



# How to Move to the Edge and Beyond?

- All the edge computing elements need to run somewhere, but where?
- **Application Servers for Edge Computing**
- Create own solution or rely on something already available:
  - Expertise
  - Costs
  - Time and Resources
  - Support

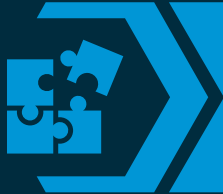
# Edge Computing



Dai, Wenbin & Nishi, Hiroaki & Vyatkin, Valeriy & Huang, Victor & Shi, Yang. (2019). Industrial Edge Computing: Enabling Embedded Intelligence. IEEE Industrial Electronics Magazine. 13. 48-56. 10.1109/MIE.2019.2943283.

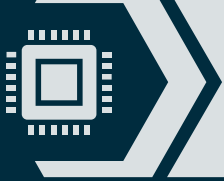
# What is an Application Platform?

Software that provides the runtime to deploy apps smoothly, securely & reliably



## Simplified Application Development

Simplify development by using pre-built infrastructural components and services, speeding up development time, reducing code complexity and allowing developers to focus on application logic



## Efficient Resource Management

Resources like memory, processing power and database connections are managed automatically



## Middleware for Effortless Integration

Different parts of an app communicate and work together seamlessly with minimal developer input



## Automated Scaling & Load Balancing

Workloads are automatically distributed across multiple servers, preventing crashes and ensuring smooth operation under peak loads



## Security & Compliance

Payara offers robust security measures such as authentication mechanisms, access control & encryption, as well as compliance features like audit trails & logging

Application platforms optimise performance, manage computing resources and facilitate seamless collaboration among different app components, meaning **developers can ignore complex infrastructure and concentrate on code**

# Development with a Platform



# How to Choose The Right Framework

- Clear trend towards **open, backward compatible, interoperable vendor-neutral** technologies
  - **For Users:**
    - Benefit from greater flexibility
    - Utilize the (automation) products that are best suited to address their specific requirements and intended applications
    - Create more connected systems that support IIoT
  - **For Vendors:**
    - Deliver compatibility with a broader range of devices (and broader market reach)
    - Establish synergistic collaborations with other industry players

# Why Jakarta EE...

- **Vendor Neutrality**

- As Jakarta EE is vendor-neutral, it prevents developers from being locked into a single vendor and gives them the ability to select the technologies that are most suitable for their particular needs.

- **Open Specifications**

- The platform provides a comprehensive collection of open specifications for the development of contemporary Java applications that are native to the cloud.
- This ensures that the platform will continue to support and innovate over time.

- **Interoperability**

- Jakarta EE APIs are in a position that is unique in their ability to support forward-looking, interoperable platform engineering practices. These practices can range from edge devices to complex cloud-native microservices solutions.



# Why Jakarta EE...

- **Community-Driven Development**

- The Jakarta EE community is actively involved in the maintenance and evolution of the platform, which ensures that it remains current with the requirements of the industry throughout its existence.

- **Stability and Backward Compatibility**

- Jakarta EE offers a stable and tested platform for innovation, allowing organizations to fully capitalize on their existing investments in enterprise Java applications.

- **Multi-Platform Support**

- Jakarta EE applications can run on various platforms, from traditional servers to cloud environments, providing flexibility in deployment.

# Why Jakarta EE...

- **Modular Architecture**

- Jakarta EE's modular design allows applications to scale seamlessly. It supports microservices architecture, enabling individual components to be scaled independently based on demand.

- **Robust APIs**

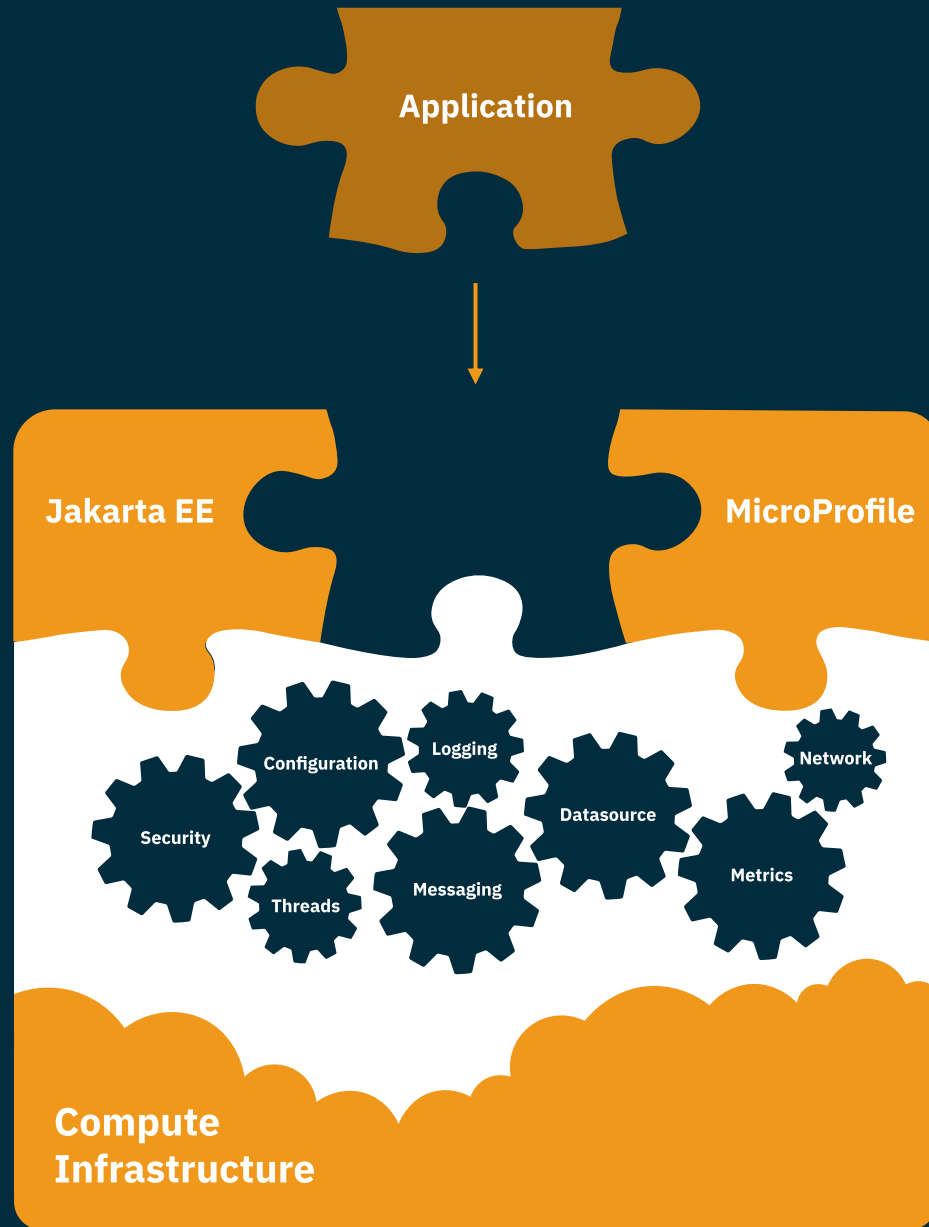
- With a comprehensive suite of APIs, Jakarta EE simplifies the development of scalable applications, managing resources efficiently and optimizing performance.

- **Cloud-Native Readiness**

- Jakarta EE is designed to integrate with cloud environments, supporting auto-scaling and resource management features typical of cloud platforms.

# Benefits of Jakarta EE Model

For Payara Platform Users



# Jakarta EE 11 Specifications

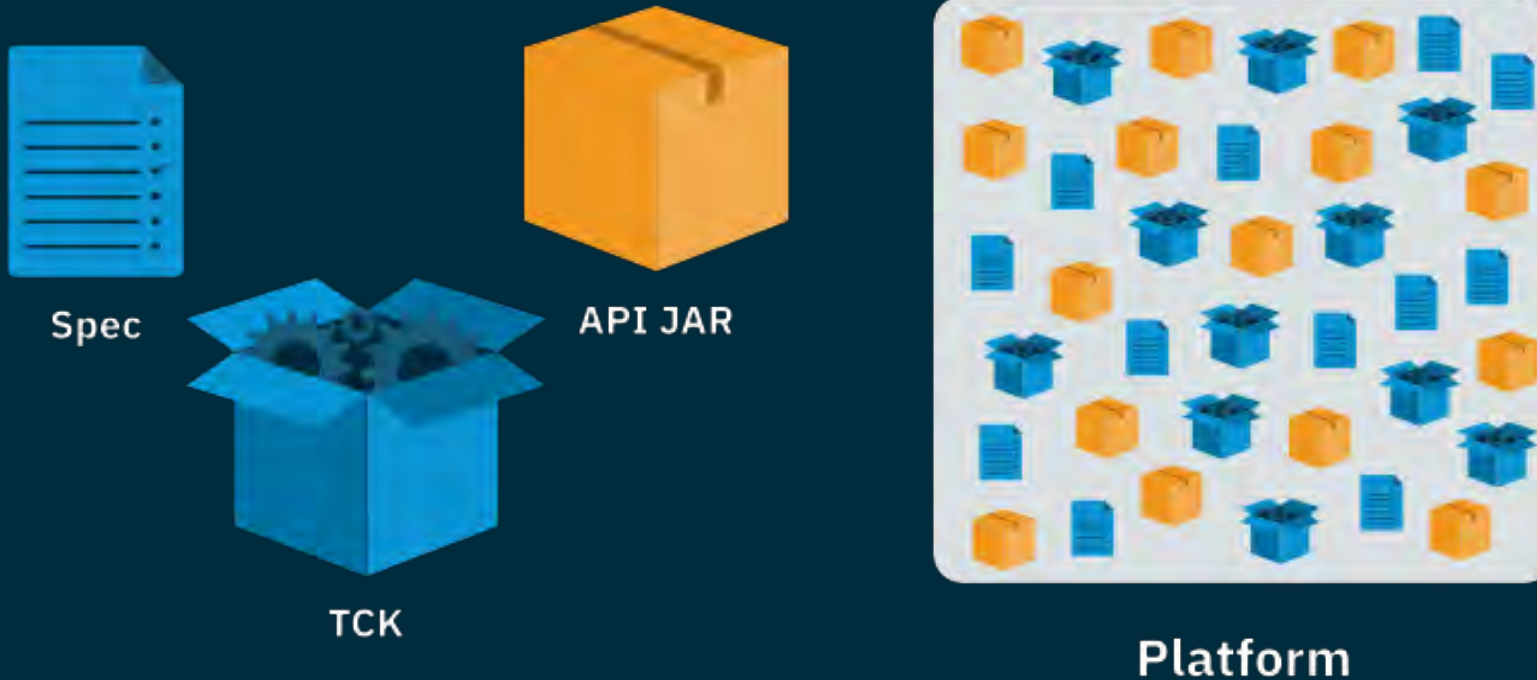
| Platform                  | Web Profile             |                            | Core Profile             |
|---------------------------|-------------------------|----------------------------|--------------------------|
| Authorization 3.0         | Authentication 3.1      | Persistence 3.2            | Annotations 3.0          |
| Activation 2.1            | Concurrency 3.1         | Server Pages 4.0           | CDI Lite 4.1             |
| Batch 2.1                 | CDI 4.1                 | Validation 3.1             | Interceptors 2.2         |
| Connectors 2.1            | <b>Data 1.0</b>         | WebSocket 2.2              | RESTful Web Services 4.0 |
| Enterprise Beans 4.0      | Expression Language 6.0 | Debugging Support 2.0      | Dependency Injection 2.0 |
| Mail 2.1                  | Faces 5.0               | Enterprise Beans Lite 4.0  | JSON Binding 3.0         |
| Messaging 3.1             | Security 4.0            | Standard Tag Libraries 3.0 | JSON Processing 2.1      |
|                           | Servlet 6.1             | Transactions 2.0           |                          |
| <b>Optional Specs</b>     |                         |                            |                          |
| SOAP with Attachments 4.0 | XML Binding 4.0         | XML Web Services 4.0       |                          |

**Updated**

**New**

**Not Updated**

# What is Jakarta EE Specification



# Performance, Monitoring

- Concurrency and synchronization is built in
  - Thread pools
  - ForkJoin threads (Payara)
  - Virtual threads (Jakarta EE 11)
  - Configurable on server or by a program
- MicroProfile Metrics and Health are supported
  - Endpoint for Prometheus (→ Grafana)
- Monitoring Console is built in for simple graphs
- JMX, connectors, REST



# Why the Payara Platform...



**First for  
Jakarta EE  
Developers**

**Committed to  
Jakarta EE**





# Payara Services Helps Shape the Future of the Industry

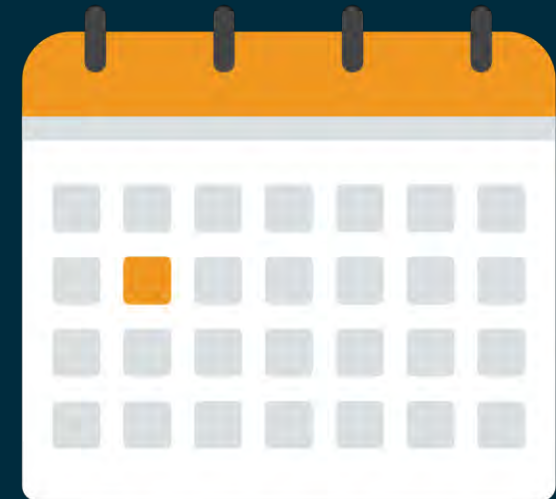
- Contributor Members of the Eclipse Foundation
- Strategic Members of the Jakarta EE Working Group
- Members of the MicroProfile Working Group
- Project Management Committee member of Jakarta EE



# Why the Payara Platform...

## Stable & Supported Software Designed for Mission-Critical Production Environments

- Stability with 10-year software lifecycle
- Security alerts and patches
- Regulatory compliance assurance
- Choice of support: Migration & Project Support, 24x7, or 10x5



# Why the Payara Platform...

Each Payara product fulfils a key developer use case



Payara's core application platform kernel supporting mission-critical Java production systems



Configure, manage & monitor multiple instances of Payara Micro for scale & resilience



Next-generation of cloud-native application runtime. Deploy your Java apps on the cloud in one click, automatically – just like magic

Deployment



IoT / Edge



Containers



Bare Metal



Virtualisation



Kubernetes



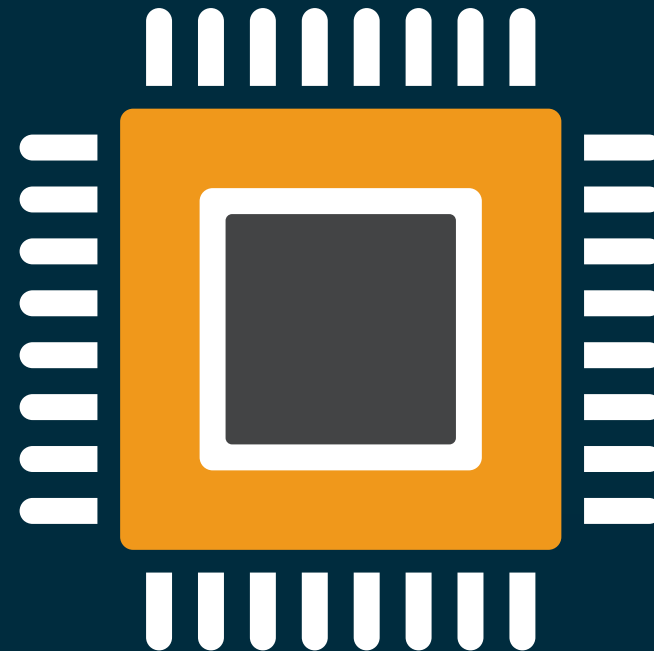
Cloud



# Why the Payara Platform...

**The Payara Platform is an ideal solution for building intelligent Edge Servers.**

- Payara Server is ideal to support the needs of Edge computing.
- For application on the Cloud, Payara Micro is your go to. It is optimized to work in containerized environments with a lightweight and compact footprint.
- IoT devices can send data over MQTT Cloud Connector to the Payara Platform on the Edge, which can then apply business logic, aggregate and analyze data.



# Payara Platform Enterprise

## Payara Server Enterprise

Robust. Reliable. Supported.

The best application platform for production Jakarta EE apps.

## Payara Micro Enterprise

Small. Simple. Serious.

The platform of choice for containerized Jakarta EE microservices deployments.





# Demo



# Conclusions

- Robot-based operations can greatly benefit from the Cloud
- To adopt the Cloud for industrial automation, the Edge is a must
- Suitable technology is needed to create effective Edge computing frameworks
  - Openness
  - Standardization
  - Flexibility & scalability
- Jakarta EE-specific Payara Platform offers an ideal application server for cutting-edge IIoT



# Thank You