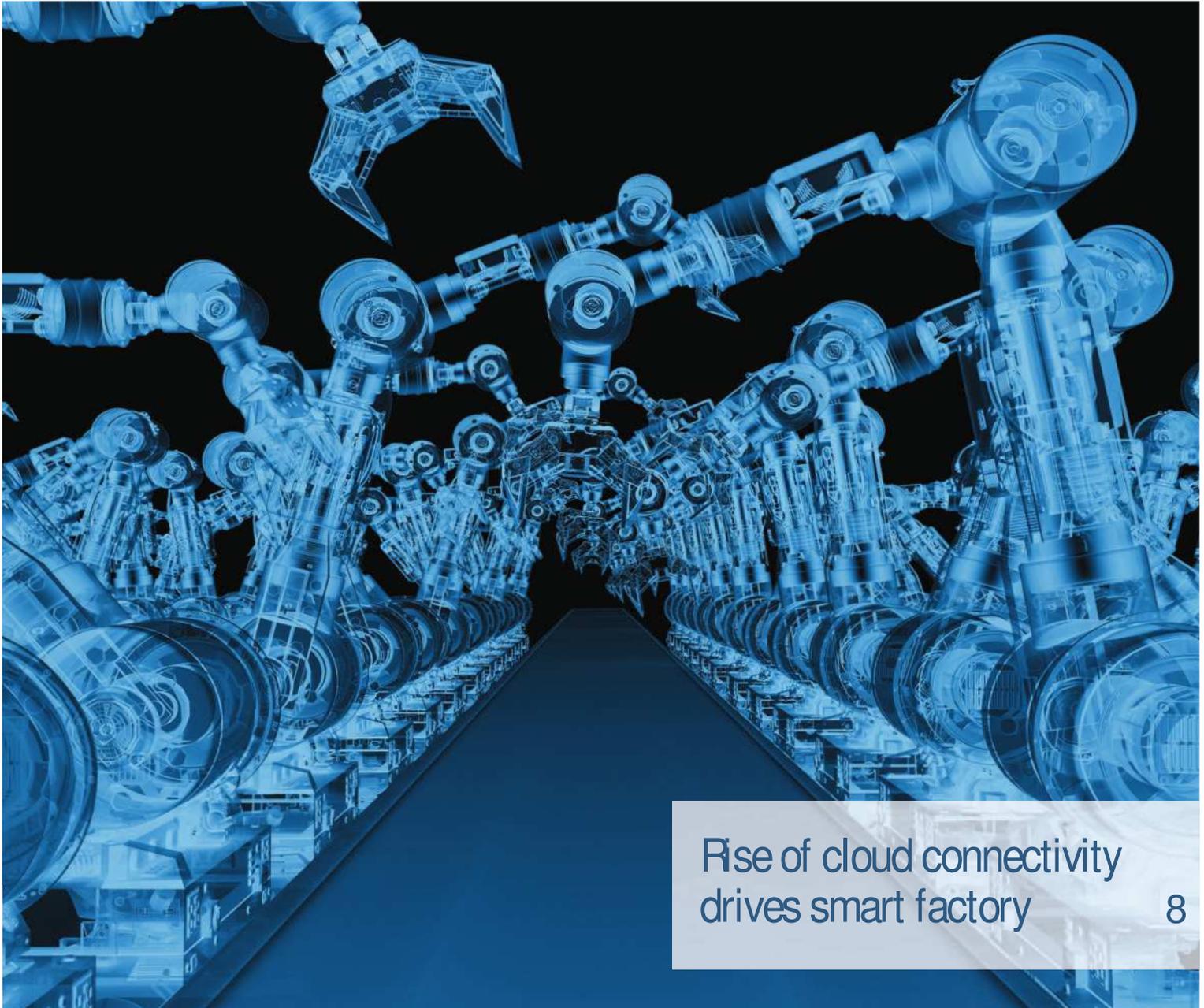


# industrial ethernet book

The Journal of Industrial Networking and IoT



Rise of cloud connectivity drives smart factory 8

Common Industrial Cloud Interface for CIP data 12

Single-pair balanced Ethernet transmission 30

IIoT best practices using MQTT architectures 33

Fast roaming challenge for industrial WiFi 42

# Open architecture enables edge-computing applications

Fragmentation and vendor lock-in are preventing companies from adopting IoT solutions to remotely manage devices at the edge. There is a need for standard-based, flexible and modular device application frameworks to develop IoT and edge-computing applications.

TECHNOLOGICAL INNOVATION, together with a steady decline of the price of sensors and devices, has led to a constant increase in the number of interconnected devices. Billions of devices will reportedly be interconnected in the next years. The fourth industrial revolution, or Industry 4.0, is based on the assumption that devices are becoming smarter, thanks to advanced software applications which allow simple and cost-effective communications among them.

## Edge vs. cloud computing

Traditional IoT cloud architectures are based on physical assets which collect data and send them to cloud data centers for further analysis. However, for some field applications latency is not an option. Data has to be managed at the field level and response must be immediate, and cloud computing might not be the best option. There is a need to bring the data center to the field. In addition, bandwidth and infrastructure costs become an issue when huge amount of data need to be transferred (e.g. via cellular connectivity); local data management is usually limited to some filtering, triggering or averages in order to make "distilled" information.

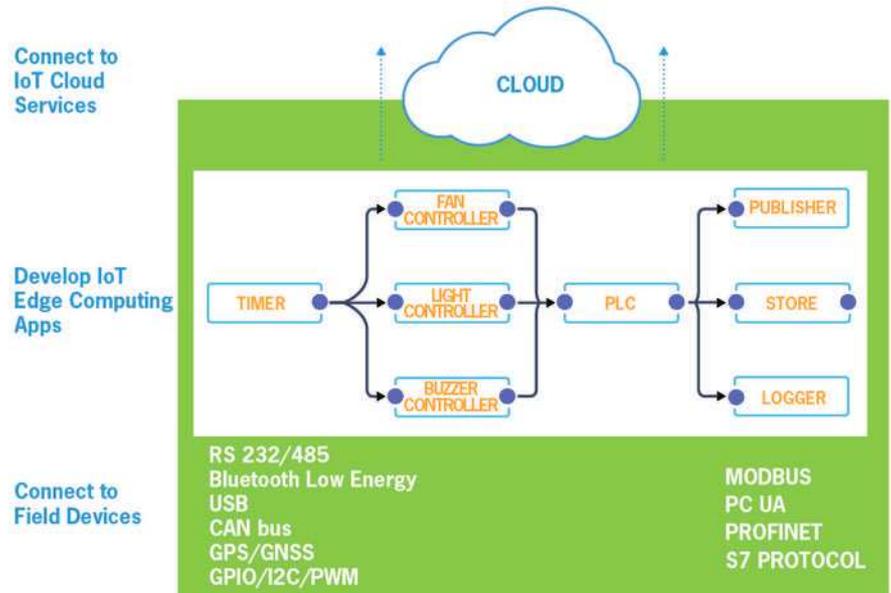
Reducing system architecture complexity is a key success factor in IIoT applications as well as turning field devices into edge computers. They become able to process data and apply intelligence, so that only the data which have already been analyzed are published to the cloud. This simplifies business applications development and leads to increased productivity. Moreover, bringing IoT application development to the field implies that edge computing can reduce the gap between information (IT) and operations technology (OT).

Interoperability between the factory and the office becomes possible thanks to IoT edge devices that securely connect the service platform layer with the factory environment. These powerful edge devices, or next generation gateways, become bridges for IT/OT communication that benefit both layers. IT has a direct and remote access to field processes and devices, whereas OT can interact with data analytics and other advanced applications to optimize production processes.

## Java and OSG for edge computing platforms

Even if some effort has been put in place in order to provide some standards and common protocols there are still some barriers that are responsible for slowing down the process of adopting IoT solutions such as vendor lock-in and fragmentation. At the edge, "things" connect to field machines through sensors, actuators, controllers, agents and gateways. There are numerous cloud platforms for data storage and analysis, and in several cases, each of them has its own communication protocols. Therefore, IoT solutions are usually tied to specific vendors, generating deeper fragmentation between IT and OT.

Java-based device application frameworks in the Open Services



Java/OSG-based building blocks for developing IoT applications.

Gateway initiative (OSG) model can speed application development at the IoT edge-device level, as well as offer advanced remote management capabilities not available through traditional embedded agents. As a matter of fact, still some of the agents are often constrained on specific hardware and mostly hard coded in C/C++, preventing developers to apply changes (they are rarely allowed to access some APIs to set a few parameters). Those agents cannot ensure advanced remote management capabilities. Remote maintenance, over-the-air updates and upgrades are difficult to perform, and developers often have to start from scratch to upgrade the devices.

At the edge-device level, a more robust and open device application framework is desirable to enable advanced device management features. A Java/OSG-based application development framework offers a portable, modular and flexible solution to easily communicate with cloud-based management platform and develop advanced IoT applications for remote monitoring and control.

## Advanced device management

Companies are interested in more advanced remote management capabilities and broad access to a wider range of parameters to control field devices. For example, a multi-national manufacturing company would want to remotely access data coming from its plants distributed worldwide, in order to monitor them and perform predictive maintenance and software updates, thus reducing maintenance costs and optimizing the production line. This kind of advanced device management can be achieved with an edge-computing platform, and Java/OSG is the best way to ensure open standards, modularity, flexibility and a standard interface.

OSGi is a modular platform for Java that implements a complete component model, providing modularization of Java applications and infrastructure allowing components to communicate locally and across a distributed network with a vendor-independent approach. As a result, developers have access to a coherent IoT services architecture based on specifications that are highly scalable for long-term remote management and maintenance. As an OSGi

deployment bundle, Java-based applications can be remotely managed and easily configured. By adding and/or removing new application bundles it is easy get directly into the application management layer. An edge-computing platform deployed this way, by means of its Java/OSGi-based applications architecture, allows the developers to easily manage various parts of the device from an application standpoint.

Developers do not like fragmentation. They prefer to develop and manage their apps in the same way across different edge devices, whether it be a Raspberry Pi or custom-built hardware, it should be managed in the same way. Using a hardware abstraction layer on top of an OSGi-based container and Java Virtual Machine (JVM) simplifies application development and optimizes portability across systems and hardware architectures. Leveraging a proven architecture and software building blocks that provide extensive services, an edge-computing platform will allow shorter device software development. Once this standard software platform is in place, processing valuable data becomes easier than it has ever been before. Moreover, integrating Eclipse Kura (a Java/OSGi based open-source project for IoT applications development) means preventing vendor lock-in and guaranteeing the protection of the software investment.

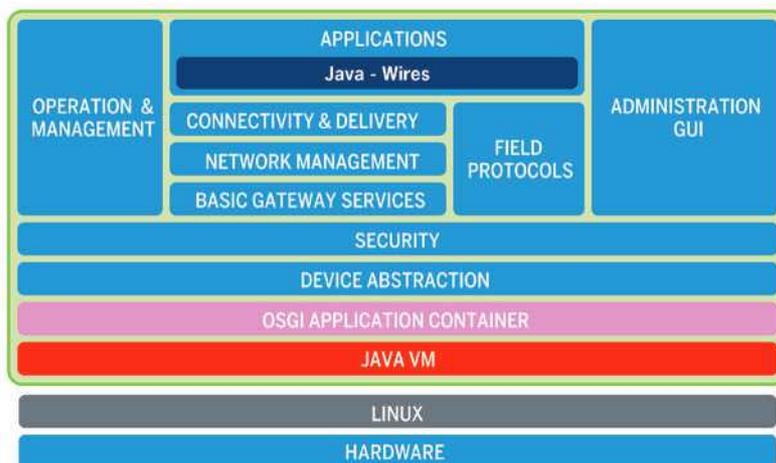
### Simplifying edge-computing application development

After providing the Eclipse Foundation with the Kura programming code, Eurotech enhanced it and realized a commercial version for the development of IoT applications at the edge-device level. Basic requirements for effective and integrated IoT applications are:

- Connectivity to field devices
- Virtualization of field assets
- Connectivity to IoT cloud services

The platform simplifies the communication between the edge-computing device and the devices by employing a single model: it comes with pre-installed field protocols libraries (Modbus and OPC-UA, for instance), so that a common format can be reused across different devices. Thanks to a device abstraction layer, it creates a digital twin of the device, by providing APIs to connect to I/O interfaces of an IoT gateway, such as Serial communication (RS 232/485), Bluetooth 2.1 and 4.0, BLE, USB and CAN Bus.

The virtualization of field devices enables visual development of IoT applications. The Wires programming environment offers a modular data flow programming tool to define data collection and processing pipelines at the edge by simply selecting components from a palette and wiring them together. This way users can, for instance, configure an asset, periodically acquire data from its channels, store them in the edge device, filter or aggregate them using powerful SQL queries, and send the results to the Cloud. Different components are represented as "wired" nodes and added and connected with drag-and-drop, freeing the developer from programming any code.



Visual IoT application architecture for connecting field devices to the cloud.

Virtualized devices are now ready to be connected to the cloud. This is possible thanks to a Cloud Service API which manages data collected by the gateway and publishes them to a remote server via the MQTT protocol. A store-and-forward functionality provides out-of-the-box support for connecting to different IoT cloud providers, allowing the development and deployment of IoT gateway solutions that are not tied to a particular vendor. In addition, the API simplifies the implementation of more complex interaction

flows such as request/response or remote resource management and offers a policy-driven publishing system to abstract the application developer from the complexity of the network layer and the publishing protocol used.

From a security standpoint, the platform offers an extensive security management set of services to provide a secure application execution environment, to reduce risks related to remote management of field devices and to simplify the management of certificates, keystores, application signing and system integrity. Additionally, a VPN client service allows system administrators to access the devices using a secure VPN connection via Ethernet, Wi-Fi or cellular modems.

Technology report by Eurotech.

**While you look ahead**  
... we have an eye for the rest.

**sps ipc drives**  
Nuremberg  
28-30 November 2017  
Hall 9, Stand 231

**360° Network Reliability for Smart Factory Automation**

- 3 industrial protocols with one-click setup
- 2 installation options: DIN-Rail and Rackmount
- 1-page configuration dashboard