FRI-2G.302-1-CCT2-08

APPLICATION OF IIOT SOLUTION IN EXISTING INDUSTRIAL SYSTEM

Eng. Svetozar Yolov, PhD Student
Department of Telecommunications,
University of Ruse "Angel Kanchev"
Tel.: +359 89 967 5330
E-mail: syolov@uni-ruse.bg

Prof. Georgi Hristov, PhD
Department of Telecommunications,
University of Ruse "Angel Kanchev"
Tel.: +359 82 888 663
E-mail: ghristov@uni-ruse.bg

Abstract: *The Industrial Internet of Things (IIoT) is transforming the automation industry by enabling seamless connectivity, data exchange, and real-time analytics. IIoT solutions, enhance automation in many fields as data exchange and management, facilitate predictive maintenance, and maximize operational efficiency through devices, smart sensors, and more. The MQTT protocol, which is optimized for efficient and lightweight data transmission across low-bandwidth networks by design, is the foundation of a new paradigm in communication systems. An essential component in IIoT solutions is the MQTT broker (like EMQX), which makes data exchange between publishing/subscribing clients, devices, and cloud platforms fast, extremely scalable, safe, and reliable. By playing a key role in the Industry 4.0 concept, gateways allow legacy systems to seamlessly communicate with contemporary infrastructures and also act as a bridge between current industrial applications and IIoT applications. In addition, gateways provide another level of abstractization, which facilitates data transformation and additive context of the data.*

Keywords: *Industry 4.0, IIoT, MQTT, Broker, Gateway*

INTRODUCTION

MQTT is a publish/subscribe model protocol, which is used for bidirectional message transfer. It is a lightweight, open standard, highly simplified and designed to be easy to use. Its features make it ideal for use in a variety of industrial (and not only!) applications, including more specific ones such as machine-to-machine communication (Machine to Machine, M2M) and Industrial IoT (Industrial Internet of Things, IIoT) (OASIS OPEN, 2015). Its specific architecture and principle of operation make it very suitable for building the so-called Unified Name Space (UNS), allowing receiving and sending data from and to a variety of clients/sources in the industrial ecosystem of an industrial enterprise.

EXPOSITION

MQTT – Brief Description

IBM WebSphere Message Queue as a technology was first conceived in 1993 to solve communication problems of independent and non-competing distributed systems (Lea, P., 2020, p. 633). Derived from the WebSphere Message Queue system, it was created by Andy Stanford-Clark and Arlene Nipper (of IBM) in 1999 to address the particular constraints of connecting remote oil and gas pipelines via satellite link. At that time, a protocol that could be used in communication links with minimal bandwidth was needed to connect oil pipelines via satellite communications (HiveMQ, 2020b, p. 5). "MQ" refers to IBM's MQ series product, developed to support telemetry transport. Here, TT stands for telemetry transport. When Andy Stanford-Clark and Arlene Nipper created their protocol, they named it after the IBM product. In many sources, MQTT is erroneously referred to as Message Queue Telemetry Transport.

As more distinguished features of the MQTT protocol can be pointed (HiveMQ, 2020a):

- Bidirectional protocol, where information can be sent from client to broker and from broker to client;
- Data agnostic (HiveMQ, 2020a, p. 15), i.e. any kind of data can be transmitted in the payload, regardless of type and structure;
- Extremely scalable. From a few tens to tens of millions of customers can connect to the broker;
- Extremely suitable for use in devices with limited computing power and memory;
- Separates publishers from subscribed clients (Lea, P., 2020, p. 394).

The MQTT publish/subscribe model (fig. 1) has four main components: publishing MQTT client, subscribed MQTT client, broker, and topic (EMQX, 2023, p. 11).

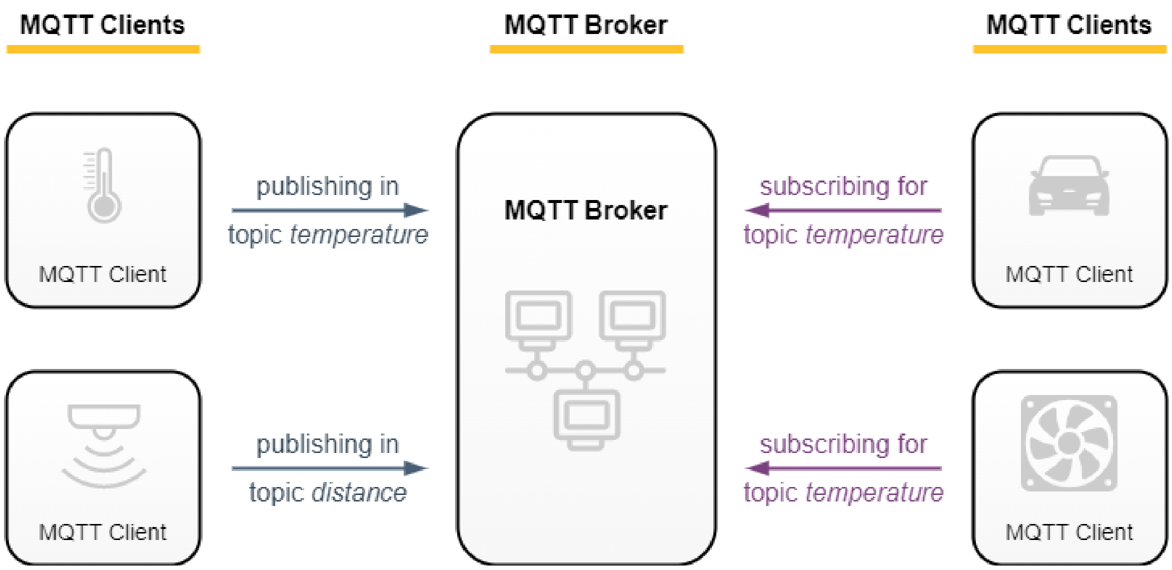

Fig. 1. MQTT model

Data is exchanged by means of messages that are logically grouped into so-called topics. An example of such a topic is the grouping of all climatic data for an established environment (temperature, atmospheric pressure, humidity, etc.) or, for example, operating parameters of a plant. The exchange of messages is managed by a so-called broker (server), which receives messages from publishing clients, filters/allocates them to the appropriate topics and then delivers them to subscribing clients to those topics. The subscriber is a data collector, connected to the broker. The MQTT broker is the central hub that manages data communication between IIoT devices, within an ecosystem (Agnihotri, N., 2023) or industrial network.

EMQX MQTT Broker

The broker available on the market, the one that stands out for its many advantages, is EMQX. It is a large-scale, distributed MQTT messaging platform that offers “unlimited connections, seamless integration, and deployment anywhere” (EMQX, 2024). EMQX is a highly scalable MQTT broker. It is available in the following variants:

- EMQX Cloud Serverless – it is a serverless architecture, particularly suitable for individual developers, small and medium-sized projects, and development or testing environments (EMQX Cloud Team, 2023);

EMQX Cloud Dedicated – a platform providing customers with independently deployed instances of EMQX Cloud;

EMQX Cloud BYOC – Bring Your Own Cloud, it allows customers to deploy an EMQX cluster on their own cloud infrastructure, which at the same time, is still managed by the EMQX team;

EMQX Enterprise – it is the commercial version of EMQX. Thanks to its robust built-in rules engine and modules for integrating data into external databases and platforms, EMQX Enterprise is capable of performing real-time processing, transformation and routing of massive amounts of IIoT data (EMQX, 2024);

EMQX Open Source – it is an open-source platform. EMQX is especially preferred by developers of IIoT and real-time communication applications due to its rich features and stability. EMQX 5 can support up to 100 million concurrent MQTT connections in a single cluster. A single server can transmit and process one million MQTT messages per second with millisecond-level latency. The difference with EMQX Enterprise is that a lot of key features for the Enterprise platform are missing.

In the project, the EMQX Enterprise version was preferred, because of the support of the Enterprise Data Integrations module. This functionality uses Sink and Source components to connect to external data systems (EMQX, 2024a). Flows are used to send messages to external data systems. The sink component can send data to platforms such as MS SQL, MySQL, Kafka, or HTTP services. The source component is used to receive messages from external systems including MQTT, Kafka, or GCP PubSub. EMQX Enterprise can be used for free within 25 connections. To implement the project, a driver (as called Action) is installed and configured in order to write messages to an MS SQL database.

MQTT Gateway

The MQTT gateway plays a key role in the MQTT ecosystem by performing the following more important functions: protocol translation, data aggregation, data hub, security, offline buffering, etc. Gateways can be divided into two main groups in terms of their performance:

Hardware systems, representing complete industrial devices;

Software platforms, which are installed on industrial PCs or IIoT hardware.

At building stage, N3uron was preferred as gateway. It is a complete industrial IIoT EDGE and DataOps platform, enabling seamless integration between the industrial shop floor and third-party applications, whether on-premises or in the cloud (N3uron, 2024). With N3uron one can easily create bi-directional data flows between OT and IT systems. N3uron's user interface is web-based. Thanks to the Web UI module, N3uron can be easily configured, controlled and managed – locally and remotely. Furthermore, it can be installed on Linux platforms without a graphical user interface (e.g. GNOME – GNU Network Object Model Environment).

IIoT Application

The subject of the project is a real production line for weighting components, used in the production of locking solutions in the automotive industry. Essentially, a production line is an Integrated Manufacturing System (IMS), which is a group of machines working together in a coordinated manner, located in a material handling system and interconnected by a control system for the purpose of manufacturing, processing, moving and packaging individual, parts or assemblies (ISO, 2007). In addition to this classification, a production line can also be considered as an industrial robotic cell, consisting of one or more robotic systems, including their associated machinery and equipment, adjacent protective space and safeguards (ISO, 2011, p. 12).

Single Weighting Machine

The single weighting machine works in parts counting mode. The principle of operation is as follows.

Copyrights© 2024 ISBN 978-954-712-951-1 (Print); e-ISBN 978-954-712-952-8

With the help of a loading station, the metal parts fall into the hopper of the machine (Fig. 2). An operator station is used to select the recipe of weighting and counting. The operator, based on the internal unique part number, selects a recipe for operation. The recipe consists of a variety of parameters: for how the vibro-conveyors are to be operated, the number of parts to be contained in, the limits of under- and over-tolerance and panic deviations from the required quantity, etc. The data from the selected recipe is sent to an industrial controller. Before starting the work process, the weighing station is calibrated. The process is then started.

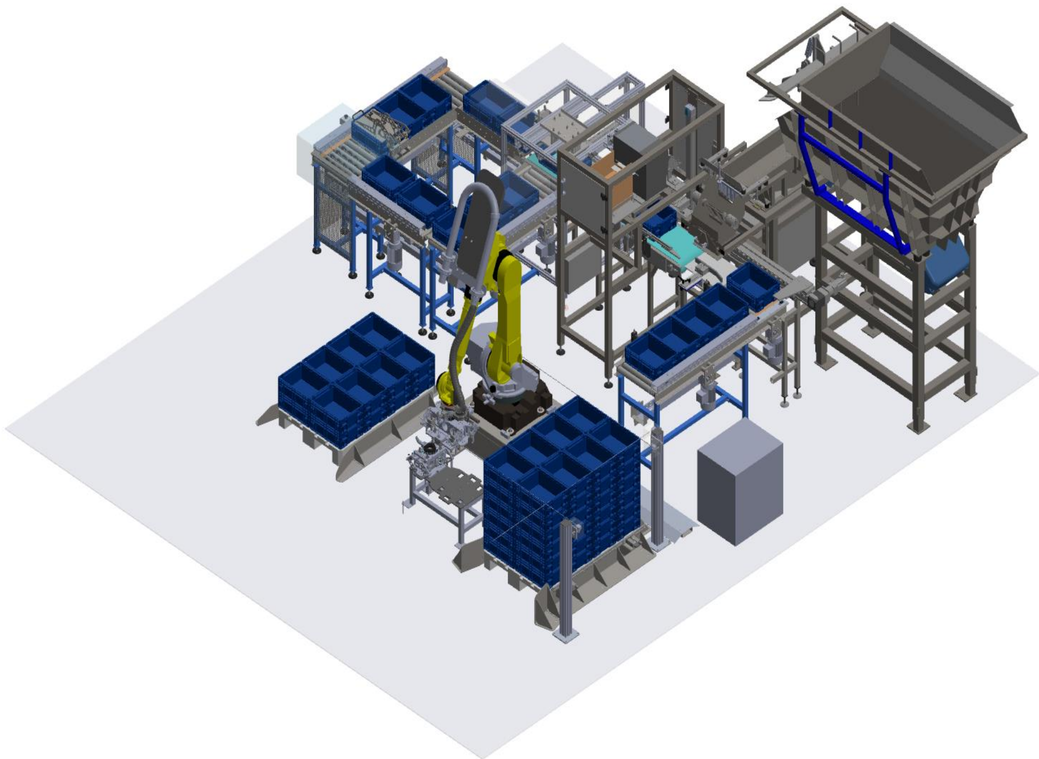

Fig. 2. Details single weighter

The parts are located in a hopper, which is placed on rubber pads. A powerful electromagnet, having its own control module, is mounted to the hopper. The machine controller, via the control module, controls the frequency of vibration of the hopper and hence the speed and quantity of parts leaving the hopper. From the hopper, the workpieces fall onto an input conveyor, which also moves the workpieces to the weighing station.

When a new empty KLT (KLT stands for “Kleinladungsträger” which means small load carrier) needs to be loaded, a servo-driven pusher moves the empty container into the weighing station. The empty container displaces the already filled container by taking its place. The full container moves onto a conveyor belt, which takes it to the labelling station.

The labelling station consists of a levers system, an industrial printer, a vertical transport and stoppers. The vertical transport, by means of a vacuum, takes the aligned printed label and conveys it to the container (KLT). Using a guiding plate, the label is placed in a KLT’s pocket, located on the long side of the container. After the label is placed, the container is released. By means of the conveyor, the full container goes to the attending machine.

Attending machine

Empty containers are stacked on a Euro pallet in several rows. They are placed according to a certain pattern. Depending on its type (KLT 3147, KLT 4147, etc.), the stacking pattern is different. The stacked pallet is placed in its home position. When the automatic operation of the attending system starts, an industrial robot approaches the pallet and starts depalletizing it. The gripper of the robot, with the help of hooks and a pressure plate, grips the next empty container, lifts it and places it on the infeed conveyor. After the robot moves away, the infeed conveyor is driven and moves the container to the stopper. If the second conveyor is empty and the movement of the input conveyor is complete, the stopper moves downwards and the container is moved into the next conveyor. The empty container, one at a time, is introduced into the inlet port of the single weighter.

The filled KLTs, after single weighter are introduced into as called dispatcher. The dispatcher is the station that checks the containers for piling. The check is carried out by means of plates, which are lowered to the container by means of pneumatic cylinders. If they manage to complete their working stroke in a set time, it is considered that KLTs are OK, and there is no piling. Other way, the KLT is considered with piling.

In the absence of piling, the container is directed to conveyors named OK 1 and OK 2. They move the full containers to the pick-up position of the robot.

Not OK containers are routed on conveyors named NOK 1, 2, and 3, outside the safety cell. A manual inspection station is provided on conveyor NOK 2. The operator arranges the workpieces in the container in order to eliminate piling and verifies this by means of the manual station. It consists of a movable plate which has to perform its downward stroke until it completely matches the top edge of the container. If this happens, there is no bunching and a signal lamp indicates that the operation has been successful. If the plate does not reach its full travel, the signal lamp does not give the required feedback signal.

The IIoT Solution Topology

The topological diagram of the IIoT solution is shown in fig. 3. The industrial line has two independent control systems, one for each unit. Each unit is controlled by a Siemens industrial controller, SIMATIC S7 1512SP-1 PN.

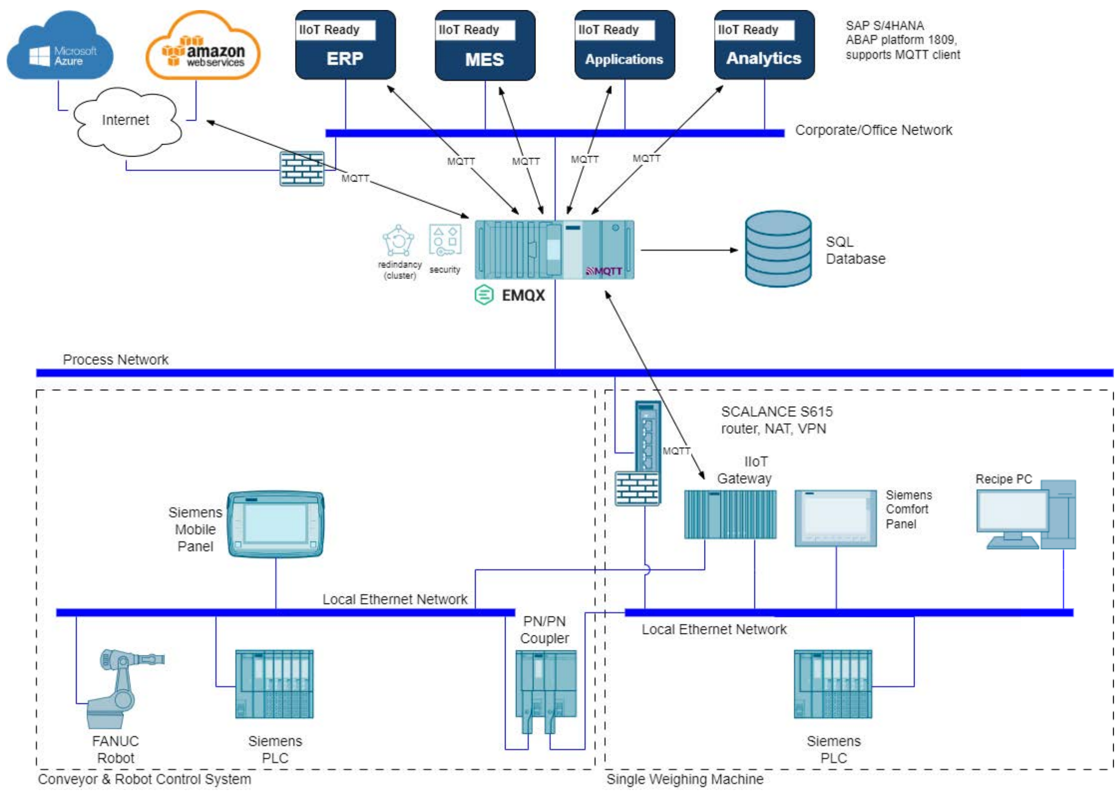

Fig. 3. The industrial system topology

The single weighting machine has a local network. Various devices are connected to it. The most important of which are: the industrial controller, distributed peripherals, the servo drives of the pusher and the labelling mechanism, the weight measurement systems, the remote access router, the operator panel, etc. The controller, on the basis of the installed inputs and outputs, perceives the changes of the input signals and, depending on the program algorithm, controls the individual devices of the machine through its outputs.

A local area network is also established in the attending machine. The industrial controller, distributed peripherals, industrial robot and mobile operator panel are connected to it. The distributed periphery is integrated into the S7 1512SP-1 PN controller via PROFINET IO protocol.

Data from industrial controllers is accessed and collected by an IIoT MQTT gateway.

The software gateway is installed on an industrial type computer. The chosen operating system is Ubuntu, version 22.04 LTS (LTS, Long Term Support). It is installed in a basic version without additional software such as office applications, as there is no need for this.

N3uron has the following modules installed:

Siemens Client. Siemens Client is a module, designed to connect the gateway platform with Siemens PLC S7;

MQTT Client. MQTT Client is a module, designed to connect to any MQTT broker to send and receive messages. Two-way connections are supported;

SQL Client. SQL Client is a module, designed for bidirectional connection with databases such as Microsoft SQL Server, Oracle MySQL, MariaDB, PostgreSQL and others;

Scripting. Scripting is a module, designed to provide the ability to use custom JavaScript scripts. These can be triggered by events such as timers, label changes, conditions, system startup and shutdown;

Web UI. Web UI is a free N3uron module, designed to provide access to a web interface through which users can configure and visualize the state of their node.

The gateway reads data from industrial Siemens controllers. The data is addressed using environment variables called tags. Tags have functionality such as normalization, alarms, trends, etc. Tags can be used in the installed modules. In this particular case, data from the controllers are used in the MQTT module. Changing the value of a tag triggers publishing its value to the topic in the EMQX broker.

The Scripting module allows intermediate processing of the data, before publishing it to the broker. One very important feature of N3uron is the ability to arrange the data in a JavaScript Object Notation (JSON) structure, before publishing it to the MQTT topic. It is most appropriate and optimal to follow the JSON object structure. The use of JSON allows a variety of data to be "packaged" in a so-called payload. In this way, multiple data can be published in one category.

On the other hand, the broker should have the functionality to parse and process data in JSON format. In EMQX, this can either be done directly using the Rule Engine or using the JQ function. JQ is a powerful command line tool and programming language designed primarily for transforming and querying data encoded as JSON.

CONCLUSION

Scalability, security in data collection, processing, and transfer from industrial processes to local (on-premise) platforms or cloud services (such as Google Cloud, HiveMQ Cloud, Siemens Insights Hub) are important features of the proposed platform. We have a state-of-the-art IIoT solution that can be easily extended and adapted for future technology requirements.

Building an IIoT system based on the Siemens SIMATIC S7-1500 PLC and integrated with N3uron as an MQTT gateway is an effective and innovative solution for deploying industrial technologies in the context of Industry 4.0. The N3uron, thanks to its flexible modular platform, allows the expansion of the system functionality depending on future needs. The Scripting module in turn allows flexible intermediate processing of data before sending it to other resources in the local network.

Such an architecture is suitable for deployment in a variety of industries where reliability, flexibility and scalability are key to the success of automated systems.

ACKNOWLEDGMENT

This publication was prepared with the support of Project 2024-FEEA-03 "Development and Evaluation of a Combined Methodology for the Digitization of Objects Using Laser 3D Scanners, Structured-light Scanners and Photogrammetry", financed under the Scientific and Research Fund of the University of Ruse "Angel Kanchev".

REFERENCES

Agnihotri, N. (2023). *The Top MQTT Brokers of 2022*. WTWH Media LLC.

EMQX. (2023). *Mastering MQTT: Your Ultimate Tutorial for MQTT*. EMQX Technologies Inc. https://www.emqx.com/en/resources/your-ultimate-tutorial-for-mqtt.

EMQX. (2024). *EMQX Docs. EMQX Overview*. EMQX Inc.

EMQX. (2024a). *EMQX Docs. Data Integration*. EMQX Inc.

EMQX Cloud Team. (2023). *A Comprehensive Guide to EMQX Cloud Serverless, Dedicated, and BYOC Plans*. EMQ Technologies Inc.

HiveMQ. (2020a). *MQTT Essentials. Features & Characteristics of MQTT | MQTT Essentials. Part 2*. HiveMQ GmbH.

HiveMQ. (2020b). *MQTT & MQTT 5 Essentials. A Comprehensive Overview of MQTT Facts and Features for Beginners and Experts Alike*. HiveMQ GmbH. Landshut.

ISO. (2007). *ISO 11161:2007 Safety of Machinery – Integrated Manufacturing Systems – Basic Requirements*.

ISO. (2011). *ISO 10218-2:2011 Robots and Robotic Devices, Safety Requirements for Industrial Robots. Part 2: Robotic Systems and Integration*.

Lea, P. (2020). *IoT and Edge Computing for Architects*. Second Edition. Birmingham Packt Publishing Ltd.

N3uron. (2024). *N3uron – Introduction – Welcome to N3uron*. N3uron Connectivity Systems.

OASIS OPEN. (2015). *MQTT Version 3.1.1 Plus Errata 01*. OASIS Standard Incorporating. OASIS OPEN.

 Copyrights© 2024 ISBN 978-954-712-951-1 (Print); e-ISBN 978-954-712-952-8