SIEMENS

Industrial Communication

Cloud connectivity for the factory floor:

The necessary steps and considerations for implementation

White Paper

Edition 10/2019

siemens.com/cloudconnect

Contents

Introduction

Versatile uses of cloud-based solutions can already be found on factory floors and in automation. The possible uses vary greatly: From the quality control of products to the maintenance of the necessary machines to the manufacture of personalized products that customers themselves have designed. The foundation for all these applications is the provision of data from the field level to cloud-based servers. In practice, the challenge here is to ensure that interactions between IT-oriented cloud services and classic OT-oriented (operational technology) field devices run as smoothly as possible. The white paper discusses the first steps necessary to connect the automation network to the cloud.

First steps with the cloud

"Cloud" is a term that is used very differently and often remains rather abstract without further explanation. You can usually only get a clearer picture by reading into the offers of cloud providers and testing the available services. Surprisingly, these first steps are quite simple. Trial accounts can often be created quickly online that are active for a limited time period or come with a restricted range of functions. This provides you with an overview of all possible services, which can then be ordered for a corresponding fee. Exemplary categories for such services are:

Cloud service category	Description	Example services
Storage	Online storage to store data	Backups, archive storage
Network	Services that are necessary for the operation of a network	DNS server, VPN gateway, web server
Computing power	Outsourcing computing power to virtual machines	Virtual machine environments
Internet of Things (IoT)	Networking and administration of external data suppliers	MQTT broker, streaming analysis

The number of services is growing rapidly and it is easy to lose track. Depending on the industry and/or usage, often only a fraction of these services are actually of interest to the customer.

Ordinarily, each individual service in turn features a variable pricing model, which depends on the configuration of the service itself.

If the specific task is to move data from the automation network to the cloud in order to analyze or visualize the data there, then primarily the services from the IoT category are necessary. The first step is then to create an interface service and to set up access points at this interface that can be used by the field devices.

In most cases, an MQTT broker is instantiated and user devices are created that can publish there (see also **"MQTT overview"**). The broker is then hosted in a (selectable) data center of the cloud provider and is accessible using a personalized URL.

The cost of such a service normally depends on the number of messages that can be sent (by field devices) to the broker in a given time period. For example: 5,000 messages / day for a monthly fee of €4.99.

First steps with the cloud

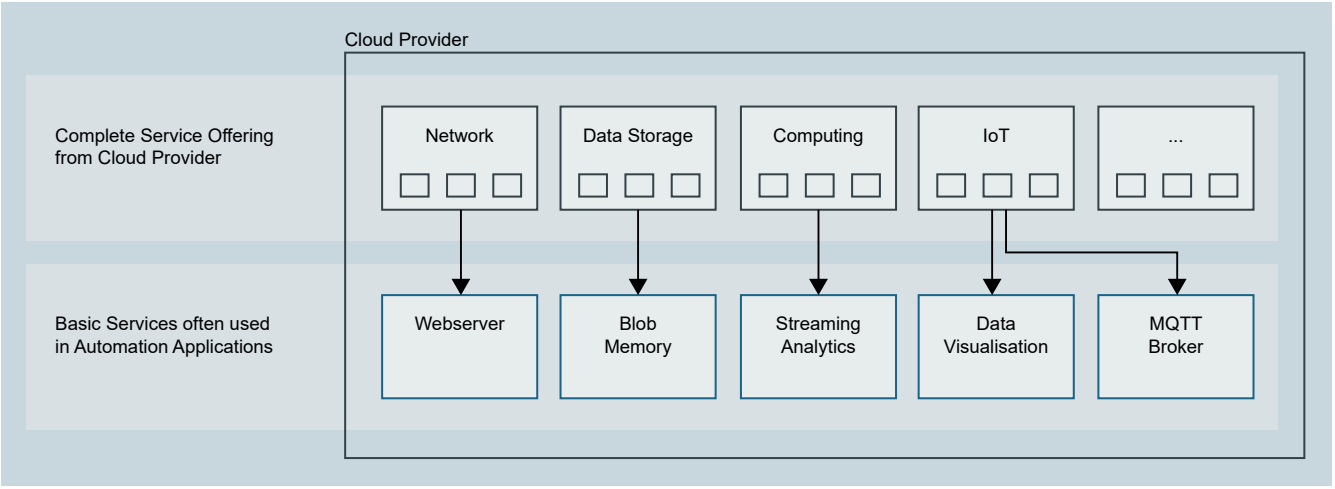
Then it is possible for MQTT-capable terminals to transmit data to or fetch data from this broker.

The implementation of the actual cloud application is just beginning. Further services can now be ordered that analyze, filter, or visualize the data of the broker (and thus of the field devices). For example, a data analysis service could now be employed that examines the data of the broker and filters it according to specific values. This result, in turn, can be fed into a visualization that runs on a web server (hosted by the cloud provider) and is accessible to the customer via smartphone.

Thus, the plant status could be made available to the service technician at any time without having to establish your own infrastructure.

It can also be seen, however, that even this simple case already requires the complex interaction of at least 5 services from a cloud provider. (MQTT broker, data analysis, visualization, storage, web server.)

Since the connection of automation devices to an MQTT broker plays a pivotal role, the implementation of industrial cloud applications requires detailed knowledge of the MQTT protocol.



IIoT-specific services of cloud providers

MQTT overview

MQTT stands for “Message Queuing Telemetry Transport” and is a communication protocol that was originally conceived by IBM employees in 1999 with the aim of exchanging data as efficiently as possible over relatively unstable and low-performance networks. In recent years, however, the protocol has very successfully found its way into IoT applications largely displacing all other protocols. The specification is currently being refined by OASIS (Organization for the Advancement of Structured Information Standards) and is freely available.

The MQTT protocol works according to a publish-subscribe (pub/sub) principle for the exchange of data between any number of participants. There is a central instance, the so-called “broker”, which administers and distributes all circulating data. A participant wanting to share information does so by publishing its data to the broker. A participant interested in data can subscribe to the broker and receives the data it is interested in from the broker. In theory, any number of devices can register with the broker to publish and subscribe. A participant can also be publisher and subscriber at the same time.

The biggest advantage now is that there is no need for point-to-point connections, thus decoupling participants in a positive way and reducing the complexity. This concept offers new freedoms which mainly have the following effects:

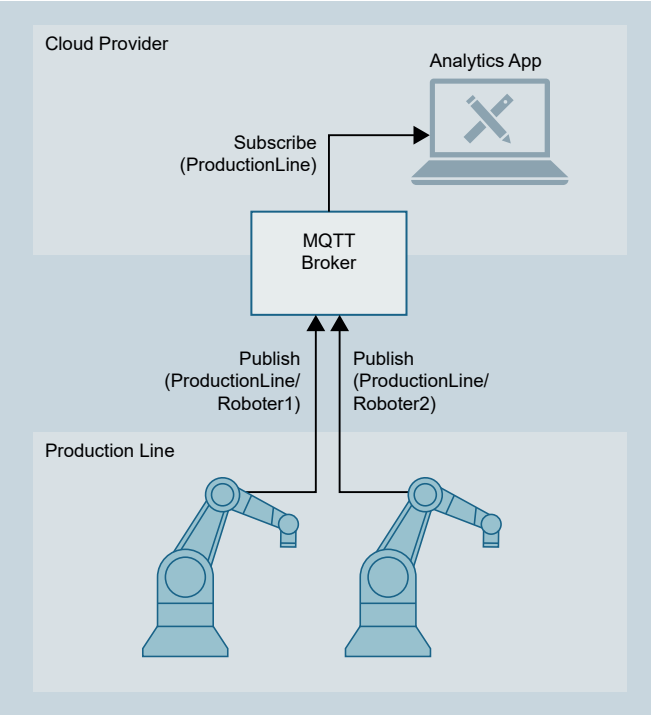
Participant independence:
A sender of data (publisher) has no knowledge of the end recipients (subscribers) and vice versa. Classical participant addresses and their administration are thus not applicable on this level.

Time independence:
Sending and receiving of data can take place at any time and above all at different times. A publisher does not care if the subscribers are interested in the data at the moment or if they are currently switched off and therefore only make use of the data later on.

Thanks to these 2 properties, pub/sub systems gain tremendous scalability. New participants can join the network without much effort or unsubscribe without the others being aware of it. That is exactly one of the main requirements of industrial IoT applications.n.

Another advantage is the minimal footprint of an MQTT stack. With a size of just a few kilobytes, it can be easily integrated into a large number of devices.

So-called “topics” are utilized, so a broker knows which data is of interest to whom. A topic is basically a kind of folder in which specific data is stored. The topic can also be nested to narrow it down further. A very simple example could be a temperature sensor that wants to make its measurement value available. For this it would publish its data in the topic “temperature”. Since an application normally has several such sensors, this is specified more precisely by publishing to the topic “Temperature/Hall_1”. Any participant interested in this value now subscribes to the topic “Temperature/Hall_1” and is then supplied with the new data by the broker every time something is published on this topic.



Publish-subscribe concept of MQTT

In addition to the many advantages that MQTT offers, there is also one major drawback from an industrial application perspective: The user data (payload) of MQTT is an arbitrary string and not subject to any additional rules regarding content or semantics of the data contained (the data is agnostic).

The problem of semantics

When transmitting data, the question always arises fairly quickly as to which data format and content the recipient and sender of information can agree on so that it is compatible. With the introduction of cloud systems, this question has become even more acute: The providers of the data (in the case of automation technology, the factory equipment or machines) are more often than not independent of the applications that make use of this data. This makes it all the more important to actually specify a fixed format upon which everyone can agree. The fact is, however, that precisely such a format does not exist for cloud systems, and it is questionable whether one will ever exist. There are multiple reasons for this. One reason is that cloud providers have a huge number of different customer groups: The cloud solutions are to be used by IT departments, machine builders, financial service providers, marketing departments, and many others. So the platform has to be as flexible as possible. However, standardized data formats are always a constraint, and providers therefore continue to make do with industry-specific detailed solutions regarding data formats.

For automation technology, the problem can be detailed as follows: The information of the plant is in most cases present in the form of program tags in a controller. These tags are images of current process values in specific data types, such as the types Real, Int, or Bool. In the simplest case, in which such a process value is to be cyclically transmitted to a cloud server, at least 3 pieces of information must be encoded in the payload string of MQTT:

1. The current process value of the tag
2. The type of the tag so that the recipient can interpret it
3. A timestamp of when this value was valid so that processing applications can create time series

In addition to these three, additional information such as a QualityCode can optionally be added. To simplify the application, users usually develop their own structures to represent the state of an entire machine. Again, this must also be encoded into the MQTT payload.

Since no uniform standard exists here, in practice it is the case that almost every application has its own format, which then also necessitates that they have their own parsers where the data is to be used.

This fact is a step backwards from the perspective of classic automation technology. In times of OPC UA and related companion specifications, which ensure the interoperability of different devices from different manufacturers, the MQTT interface of a cloud provider appears dysfunctional. Here, unnecessary work has to be put into data processing.

An initial, practical simplification is to use the JSON format, which provides some data representation rules that make reading and parsing the MQTT payload string a bit easier. Though this simplifies parsing, it does not replace it.

In order to come to a satisfactory solution, cloud providers and automation engineers need to come together even closer and jointly shape the future interfaces.

```
{
  "Timestamp":"2019.08.01 15:42:23",
  "DataItems":
  [
    {
      "Variable":"Temperature",
      "Type":"Real",
      "Value":"20°C",
      "QualityCode":"1"
    },
    {
      "Variable":"Overheat",
      "Type":"bool",
      "Value":"true",
      "QualityCode":"1"
    }
  ]

}
```

Example of MQTT payload with 2 process values in JSON format

The limits of cloud solutions

Even if at the moment one new application is emerging after another which was only first made possible through cloud-based approaches, when interacting with the automation world, there are currently still limits, which ideally should be thought about in advance to avoid surprises in the future.

In classical automation technology, the focus in data transmission has always been on a quick, cyclical exchange of relatively small amounts of data. This starts with digital or analog inputs, where very simple information such as “on” or “off” is transmitted in the single-digit microsecond range. This is followed by fieldbuses such as PROFINET or PROFIBUS which help transmit process data with secure transmission cycles in just a few milliseconds. Even the sequence programs in the controllers, which sometimes execute complex algorithms and controls, have cycle times in the millisecond range.

When this data is transmitted from the automation network to the cloud, however, the circumstances change. First, the data must be augmented with additional information such as timestamps and data types (see also **“The problem of semantics”**).

In addition, the communication protocols change towards the IT-typical TCP/IP stacks and publish/subscribe mechanisms, which add further protocol overhead and thus increase the data volume. Since these protocols also do not guarantee secure cycle times or reserved bandwidths, the effective times required to transmit the actual payload from A to B increase.

This results in a kind of a bottleneck in the transition from the automation network to the cloud. This should be considered when discussing possible applications in the cloud.

A classic example sure to test the limits is the idea of analyzing movement paths of milling machines in real-time in the cloud server. The process of milling is far too fast for the accumulating high-frequency data to be transmitted to the cloud quickly enough.

For precisely this reason, the subject of “edge”, i.e., the decentralized data preprocessing very close to the process itself, is highly emphasized by many manufacturers.

Despite these limits, the use cases remain very diverse. For example, an accurate calculation of the service life of the milling machine down to the second is easily feasible and allows the machine builder to promote the new business model so often praised.

Generational conflict: Industrie 4.0 with technology 0.4

Despite the number of cloud providers and thereby the services available through them having made an enormous leap forward in the past few years, in automation technology, both the investment and innovation cycles tend to follow a ten-year cycle. The reasons for this are mundane and obvious: changes are, first of all, associated with large costs and alleged problems that might jeopardize production and thus the entire business.

In many cases, this leads to a concrete conflict: After the company management has decided to connect all plants and machinery to the new cloud system as part of its Industrie 4.0 strategy, the production engineers have to ask themselves how to connect their "technology 0.4" inventory with 10–20 years-old equipment to this cloud system. These devices certainly do not have an MQTT interface. In many cases, the necessary data is also located in PROFIBUS or serial networks, which – from a purely physical point of view – cannot be directly connected to IP networks.

Furthermore, the automation devices have their own complex configuration and programming which cannot be readily changed. These are complexities that have to be borne by the engineers. Extreme cases – in which devices are in use whose configuration was carried out 20 years ago and today cannot be changed any more due to missing files, tools, or know-how – are not as uncommon as one might assume.

Therefore, connecting an installed base to a new cloud system requires expenses that must be taken into account in every Industrie 4.0 strategy so that the added value can be properly calculated.

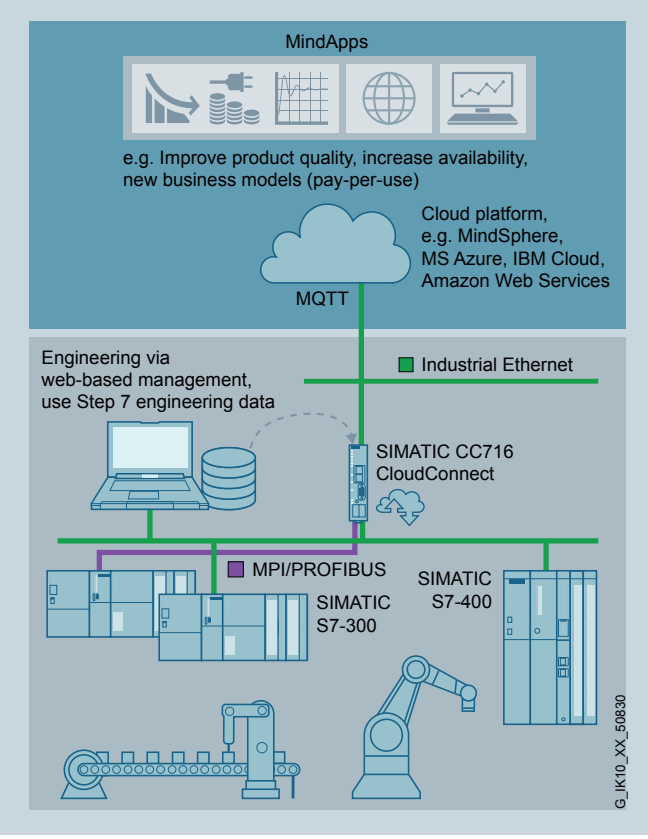
To minimize these costs, many manufacturers offer special devices that simplify the connection to the cloud. In most cases, the use of such Industrial IoT gateways will pay off, but the devices are available with vastly different specifications and with varying functional scopes. Therefore, a technical comparison and testing of such gateways in advance is highly recommended so that the later use goes as smoothly as possible and fits the application.

Selecting the right Industrial IoT gateway

Whenever devices or machines not possessing a native MQTT interface are to be connected to a cloud service, an Industrial IoT gateway is necessary.

The gateway assumes the following basic functions:

1. Collection of data from the automation network
2. Content filtering and conversion of data formats
3. Transmission of data to the cloud interface



System configuration with IoT gateway using Siemens as an example

Industrial IoT gateways are available in many different variations from a wide range of providers. Therefore, there is also a very large variance even for the most basic of functions.

The following considerations should be taken into account when choosing the right Industrial IoT gateway:

1. **Is it a product or a sandbox?**
   First of all, you should clearly distinguish between industrial products and open sandbox systems. An Industrial IoT gateway comes in an enclosure with mounting options designed specifically for control cabinets and the related environmental conditions. Sandbox solutions are usually Raspberry-Pi-like boards housed in a matching enclosure. Such solutions usually come with a development environment that makes it possible to very flexibly develop your own programs or run open source applications. Industrial devices, on the other hand, come with a defined functional scope that is configured via a specially designed configuration option (usually a web server).
   Depending on the application, one or the other solution can be a better fit. If the focus is specifically on implementing the application according to a fixed scheme with as little extra effort as possible, an industrial product is a good option.
2. **Which interfaces to the field level are offered?**
   The type of field devices which can be connected – such as controllers, motors, or sensors – is another main criterion. Here, in the first step, a list of all devices should be created from which data is to be sent to the cloud. Each of these devices supports special protocols such as Modbus, S7, or PROFIBUS. The Industrial IoT gateway should therefore inherently support these protocols so that no changes to the field devices have to be made.

Selecting the right Industrial IoT gateway

3. Is the cloud interface flexible enough?
Once the connectivity options of the field devices have been sorted out, attention should be directed towards the cloud providers to be supported. While most cloud providers support the standard MQTT, there might be some specific restrictions. In some solutions, the topic names and formats (see **«MQTT overview»**) are fixed. The IIoT gateway should be able to master this – so you should either make sure that the cloud systems supported are explicitly mentioned in the product features or that the device is flexible enough for a respective adjustment to be made via the configuration.
Furthermore, an additional requirement is important in many applications: In addition to the new cloud server, the existing MES (Manufacturing Execution System) often continues running in parallel. New process values might be required here as well. Thus, the IIoT gateway should also be able to additionally provide the data to an MES, e.g., via an OPC UA interface.

4. How many devices and data points are actually needed?
The question concerning the configuration limits of the necessary data is probably one of the most frequent points that remains unanswered for too long in the context of creating an Industrie 4.0 strategy. A direct question here would be, for example: «How many process values are needed from a production cell?» This question must be answered roughly in advance, otherwise the implementation may lead to a dead end. For this reason, the information about the configuration limits (how many data points are supported per device) in the technical specifications of the IIoT gateways should also be paid attention to. If there is no information on this on the product, it must be assumed that no performance tests were conducted by the manufacturer. This would represent an increased risk for the implementation, which should be ruled out in advance.

5. Other points:

a. Separate networks
For security reasons, it must be ensured that the gateway can work in 2 completely different subnets (1 x cloud, 1 x automation network) and that a corresponding routing between them is not possible.

b. Support of digital or analog inputs/outputs
In industrial applications, IOs are constantly needed to influence the process when necessary. The simplest example would be a switch for a hard shutdown of data transmission to the cloud because a security risk has been detected.

c. Simple device replacement
A quick and easy device replacement should be possible so as not to unnecessarily delay the production process in case of a replacement. Transferring the device configuration via USB storage into a new device is a reasonable option for this that should be considered.

d. Time synchronization
Since all process values must be provided with a timestamp (see: The problem of semantics), the gateway must be able to synchronize the time via a server. Setting the time on the device manually is useful for commissioning but insufficient for an ongoing process. Automated synchronization (e.g., via NTP) must be offered here to prevent distorted time values should a power failure occur.

Published by
Siemens AG

Digital Industries
Process Automation
Östliche Rheinbrückenstr. 50
76187 Karlsruhe, Germany

PDF
BR 1019 11 En
Produced in Germany

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit **https://www.siemens.com/industrialsecurity.**

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under **https://www.siemens.com/industrialsecurity.**