

ConnectorIO

Appliances by Market



INTELLIGENT SOFTWARE HUB FOR CONNECTING HARDWARE DEVICES

Table of contents

[Table of contents](#)

[New \(smart\) devices = New challenges](#)

[Hardware & Software interface compatibility](#)

[Hardware interface solution have their life-cycle](#)

[Hardware-level vs. software-level changes](#)

[A good practice is to keep most changes on software level since replacing the hardware is usually more expensive and requires large amounts of manual work plus servicing.](#)

[What Connectorio is all about?](#)

[Advantages of a Software HUB](#)

[Internet of Things Ready](#)

[There's a Future in Software Gateways](#)

[Advantages of software over hardware for IoT](#)

[Mobile apps for IoT - Why it's not a good idea?](#)

[Disadvantages of using mobile for IoT](#)

[Connectorio's Platform](#)

[Platform concept \(diagram\)](#)

[Diagram 1. Architecture \(simplified\) - Connectorio's architecture.](#)

[Integrations](#)

[Platform components](#)

[IoT Gateway Runtime](#)

[Plug & Play for new hardware](#)

[Other Runtime functions](#)

[Gateway Core](#)

[Drivers](#)

[Possible Applications](#)

[Smart-Home](#)

[Smart-Office](#)

[Smart-City](#)

[Industry 4.0](#)

[Automotive industry](#)

[Cloud offerings](#)

[OEM offerings](#)

[Country-wide monopolies](#)

[Large hardware manufacturers](#)

[INTEGRATIONS](#)

[Integration offerings](#)

[About Connectorio](#)

[Why we use Open-Source based solutions?](#)

[Connectorio's expertise](#)

[Our advantages](#)

New (smart) devices = New challenges

The amount of hardware which is around is growing faster than ever. With this growth, we see increasing amounts of interesting **data** which could be used in many analytical scenarios.

At the same time, it gets **harder to gather all this data** as each and every device is different and quite often uses a different standard to communicate.



The Software Industry has had this challenge for many years - trying to “glue” multiple components together. Every piece of software runs on some computer but the amount of hardware configurations and its variations is so significant that we can’t even guess all possible setups.

Hardware & Software interface compatibility

For this reason, hardware components interact with each other using well-defined interfaces and operating systems. The operating systems (OS) are built on top, allowing to plug any device as long as it uses a **standard interface** and/or is shipped with OS **compatible drivers**.

Primary interfaces in this area which consisted of a solid base for today’s computers evolved slowly over time and are still used until today in industrial solutions.

Hardware interface solution have their life-cycle

What took years or decades in hardware, took months in software. For example, serial interfaces (serial bus) are still widely used in hardware while the service bus (used for software integrations) became obsolete creating **compatibility issues**.

Hardware-level vs. software-level changes

Even in today's global economy which allows fast manufacturing of new hardware devices, software offers much easier handling of multiple versions of hardware simultaneously.

A good practice is to keep most **changes on software level** since replacing the hardware is usually more expensive and requires large amounts of manual work plus servicing.

What Connectorio is all about?

Connectorio's idea is quite basic in its form. We offer an **intelligent software hub solution** which is like an universal software bus allowing different devices using incompatible communication standards to "talk to each other".

It can be summarized as something like a **Google Translate tool for hardware devices**.

Advantages of a Software HUB

The advantages of using a software HUB over hardware solutions:

- Software can easily be embedded on **various** hardware configurations.
- Provides a **unified** way of handling 3rd party devices.
- Constitutes a communication bridge with the external world and **cloud infrastructure**.

Because sensors are usually quite simple, and they should stay like this for a reason, a software gateway or hub is something that becomes a perfect platform to coordinate **information exchange** and even more.

Internet of Things Ready

The paragraph above is a short characterization of the **Internet of Things** (IoT) gateways and a foundation for building dedicated products for different markets.

There's a Future in Software Gateways

We believe that "gateways" are going to be widely used in the long run as they provide a few key functionalities such as:

- Decentralization of solution.
- Customization of configuration.
- Local security handling.
- Adapters for simplified devices.

Advantages of software over hardware for IoT

The Internet of Things is a wide term which can be used in relation to hardware sensors, industrial controllers or even a cloud suite which handles the load from multiple devices.

Our belief is that IoT will not happen without improved hardware manufacturing processes, but we also believe that there is no way to pack all the functionalities into the hardware devices.

There is no way to pack all the functionalities into hardware devices alone. Some things are **easier** to be done **in software** and that's our focus.

Mobile apps for IoT - Why it's not a good idea?

We see that a lot of push is made on mobile devices to become the central points of management for IoT devices, as we carry them all the time with ourselves.

However, we do not see it as an improvement. Instead of 5 remotes, we get 5 different apps on the screen which we need to constantly swipe in order to be able to manage our daily tasks.

Disadvantages of using mobile for IoT

From our point of view, a **mobile device** can only be a **tool**, but NOT the main point of the system.

- It is yet another input to the platform, but not a full blown platform itself.
- It can't be used for many appliances because of the hardware pluggability limitations.

Thus focusing only on mobile might **cut off** all existing standards except **Bluetooth**.

Connectorio's Platform

In order to host properly most of the existing interfaces, a computer with pluggable inputs/outputs is required.

While most external communications can be made over the network, almost all inputs require local connectivity or short distance.

This will allow devices and software to cover a wide spectrum of standards - from existing **wired installations** such as [KNX/EIB](#), [Modbus](#), [BACnet](#), to **wireless** such as [ZWave](#), [ZigBee](#) as well as long-range [LoRa](#) (low power).

Platform concept (diagram)

A unified, yet simplified, view of the platform looks like this:

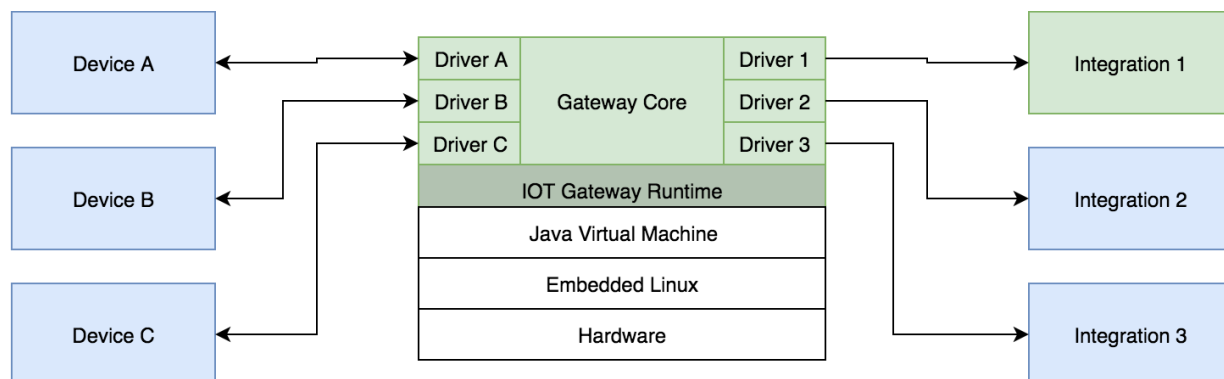


Diagram 1. Architecture (simplified) - Connectorio's architecture.

How to read this diagram?

This document contains several diagrams. For reader's convenience, there are a few basic rules which are followed when it comes to these diagrams.

1. All external components, which are not the main point of interest, are marked with **blue color**. This includes 3rd party devices, external software, and other integrations.
2. Internal components, which are a point of interest, and which are going to be developed and/or maintained, are marked with **green color**.

The image above shows a few important elements - multiple devices and multiple integrations.

The devices are **blue** because they come from 3rd party vendors. We do not plan to ship our own hardware for sensors, switches or dimmers. What we are focused on are **software-based controllers and/or receivers**.

Most of the integrations are also marked with **blue** color because the platform is able to send or receive data from anywhere, not only to or from its own vendor's equipment.

Integrations

This means that there might be integrations with **cloud offerings** from big vendors such [Azure IoT Hub](#) or [Google Cloud IoT](#) to store data as well as trigger channels hosted via [IFTTT](#) or voice assistant.

The **green** placeholder left for **Integration 1** is left purposely for vendor-specific cloud addons which will allow delivering **additional integrations** with other services, **control authentication** and **authorization** via remote channels and more.

The amount of possible scenarios which can be implemented with this architecture is quite wide and limited solely by the connectivity and hardware configuration.

Thanks to the use of **Java Virtual Machine** as a software basis, the entire solution will scale along with hardware.

Platform components

There is a holy grail in the software industry which is called “**platform**”. We are no different, we want to achieve this holy grail too. The paragraphs below briefly describe the role of elements mentioned in the diagram ([Image 1](#)).

IoT Gateway Runtime

While Runtime doesn't participate in business functions (most of the time), it is an ultimate foundation for making the whole thing work. There are important aspects, which are required to be present - which is for example **pluggability**.

This element alone would be sufficient to write a book, but for the sake of this document we will just assume that pluggability of Runtime should allow to install and uninstall new modules and handle plugs at different levels.

Plug & Play for new hardware



The user, who brings new hardware home, can take advantage of **pluggability** to connect this hardware to the system with **just a few clicks** on screen or a few taps on a mobile device.

Other Runtime functions

The Runtime is also a place which hosts important low level elements related to configuration management, configuration updates, auditing and more. Our selection of Runtime is based on own experience and observations of global market. The same Runtime is used by multiple vendors in smart city, software defined network and enterprise software area.

Gateway Core

The heart of the system is hosted above the runtime layer. This is place containing all abstractions which allows to create drivers and let devices interact between each other, without knowing each other. Core contains representation of "things" which might reflect physical devices as well as other software components hosted abroad.

Thanks to detaching from all standards and definition of lowest common denominator new functionalities can be built. This includes a rule system, a trigger framework and complete authentication.

Because many of hardware elements which are hosted locally do not implement any security or its security is fairly limited, core can provide a centralized way to control and audit access to certain aspects of deployed systems.

Gateway Core is based on open source components which are already used in commercial products and proved to work in wide deployments at (smart) home level.

Drivers

Because the **System Core** is free of any major influences from any standards, drivers must provide adapter logic from its own system to the Gateway Core.

Drivers are likely to know internal details of integrated protocol and will use this knowledge to deliver more detailed information to system about connected devices. For example a device adapter can implement discovery mechanism specific to standard. A specific persistence adaptor will store state updates for later to be displayed in user interface.

Drivers are small bridges who work on the edges - they connect specific field to or from core. Thanks to them core can work independently of any device. Given that amount of abstractions in core is more than just device, there are different kinds of drivers which are not always related to physical items. They might be very well a remote triggers for certain actions to happen.

There are many drivers already implemented and available under open source licence and there's a wide community of people who are testing them for free.

Possible Applications

A **unified platform idea** allows customizing solutions for different markets. Some integrations and drivers will be common between them and will allow the reduction of costs across deployments or sectors.

At this moment no major customizations in software are expected. Primary differences are integration components deployed in Runtime among hardware variations which might differ in order to support edge cases.

The list below contains all possible application of Connectorio's software by market or business type and specifies the benefits in each area.

SMART HOME

Smart-Home

Access to this market is open for exploration and has already been started by multiple entities.

Gateway Core with its runtime is available as open-source and widely deployed worldwide by early-adopters and computer-aware people.

The amount of commercially deployed installations based on customized hardware in the European economic zone remains undisclosed. As it is backed by big telecom companies, and assuming the scale of their businesses, it can be assumed that it goes in tens of thousands.

A free cloud offering for open source version and early-adopters alone has **20000 registered users**.

This service is fairly simple as it does not include any data storage option. It is used only as a reverse proxy and integration hub (eg. with [Amazon Alexa](#)). It is also used as a sandbox for startups based on it.

SMART OFFICE

Smart-Office

The same solution is used for smart-home and is known to be deployed in a few commercial facilities to integrate several industrial grade protocols such as [DALI](#), [BACnet](#) and [Modbus](#).

Users in this area are quite often left without integration option and look for tools which will allow them to control the entire building in an effective way.

One of the issues in this area is, for example, centralised steering of industrial HVAC units which consume huge amounts of energy. It is entirely possible to group devices and manage them together via an adapter developed by [Code-House](#) (an earlier incarnation of [Connectorio](#)) back in summer 2016.

Even though the interest in this sector remains low, it is worth exploring as the number of components to manage in commercial spaces is huge and quite dated. Thanks to software approach old standards can start working with modern technology at low cost.

SMART CITY

Smart-City

Similar concept of IoT gateway is used in at least one commercial offering which is focused on low power sensors and **narrowband IoT (nbIoT)**.

City area requires additional set of hardware sensors due to the nature of deployment which disallows static wiring and promotes wireless and low energy profile usage.

Gateway Core is currently not tested against this scenario, however its abstractions should work properly with most of cases. Given that most sensors deployed in the field will be basic (due to area to cover in big metropolis) they should be easy to integrate.

This creates a unique possibility to partner with hardware vendors to provide own solution or let them do such under own brand.

Smart city will require its own **dedicated cloud solution** to aggregate traffic from multiple gateways and connected sensors which will definitely have peaks in activity.

On the other hand due to structured nature of events inside of platform core it will become a perfect playground for **artificial intelligence** and algorithmically advanced solutions.

INDUSTRY 4.0

Industry 4.0

Plugging **production lines to the cloud** is currently a big trend promoted by machinery producers as well as big software firms who urge to grab this piece of data.

Many of the industrial equipment producers have industry 4.0 in their marketing materials and cloud related offerings, but all of them are based on their **proprietary solutions**. While it's not bad, it might lead to difficult situations if two vendors are found in one or two facilities.

A hardware agnostic approach offered by us might allow producers to get a better position in negotiations and use their own strategy for powering this revolution.

Our gateway after deploying dedicated UI might be used to provide local **adapter between incompatible standards** and protocols at **low cost**.

AUTOMOTIVE

Automotive industry

The same kind of IoT Gateway Runtime is used in cars to connect multiple systems and sensors. This means that the lowest level of software developed by a company will have another market which is not competitive to any of the above.

Since most of the car manufacturers already solved their integration troubles this offering would be interesting only for **arising markets** which are developing their own technology and might not be able to pay high costs of getting **expensive technology**.

CLOUD

Cloud offerings

Each and every above scenario leaves a big opportunity for server side software which is free of hardware limitations and can be scaled elastically.

Most basic cloud offering includes **storage of collected data** while more advanced scenarios might include configuration management and Runtime administration.

Some of the services must be developed from scratch for own use - for example remote administration of runtimes which can be interesting for Automotive and Industry 4.0.



OEM offerings

For markets which have **high entry barriers** due to licensing or legal costs, or places where we would like to **speed up market adoption** we can enter into an agreement with an existing company which has a dominant position.

Country-wide monopolies

OEM offering might be interesting for country wide monopolies such electricity and gas providers who plan to deploy **smart energy meters** (not yet in Poland).

Thanks to the IoT gateway their offer could become more attractive for end consumers.

Large hardware manufacturers

Another set of OEM offerings can be aimed to big hardware manufacturers who did not develop their **IoT strategy** or do not want or simply cannot invest in **software competences**.

Thanks to handling this task by us, they will be able to **bridge the gap** between their products and users who are willing to connect their hardware with other equipment solutions available locally.

≡ INTEGRATIONS

Integration offerings

The last option for keeping some revenue stream is related to **open source components** which are based on a mix of products mentioned above.

Given that system core remains open we can use this to provide official (or semi-official) **support for extensions** which are dedicated to certain hardware vendors.

Thanks to that these vendors will be able to access wide market of smart homes powered by our infrastructure while keeping it at a fairly low cost.

After custom development a regular maintenance fee would be charged for such a solution.

About Connectorio

Why we use Open-Source based solutions?

Contemporary software is composed of multiple components and many of them are based on open-source software (**OSS**). Thanks to the use of the open-source code, companies are able to deliver their complete software solutions faster by basing parts of the solution on OSS.

A significant number of open-source projects are useful as utilities but quite often they provide little value as stand-alone solutions and have to be combined together with other components to show real power.

Connectorio's expertise

Connectorio's experience is entirely based on Open-Source and middleware¹. Over the past **11 years**, our primary focus was this. We have spent the last 8 years working with customers who used open-source boosted by our knowledge to build bespoke products.

¹ Software responsible for mediation, routing and other activities which enables more complicated business processes to be not only drawn, but implemented and run in organizations

Our expertise let telecoms, insurance, banks, and other businesses grow faster and deliver a better experience for their customers.

Our advantages

- **agility and flexibility** which is usually smaller in large organisations,
- perfect **knowledge** of the Runtime,
- ability to implement a **security framework** to the Gateway Core components.