

# ConnectorIO

## Cloud BMS & IIoT Connectivity Solutions for CRE & Industry 4.0

---

connectorio 

[Connectorio.com](https://connectorio.com)

26th November, 2019

- **ConnectorIO Whitepapers**

---

# Table of contents

<b>Table of contents</b>	<b>1</b>
<b>Brief introduction</b>	<b>3</b>
<b>What is ConnectorIO?</b>	<b>4</b>
ConnectorIO's goal	4
What can be automated with ConnectorIO?	4
<b>How is ConnectorIO different from a traditional BAS/BMS?</b>	<b>5</b>
<b>ConnectorIO vs. Traditional BAS/BMS comparison</b>	<b>5</b>
<b>How ConnectorIO is built?</b>	<b>7</b>
ConnectorIO's infrastructure map	7
Central Unit's internal architecture and key components	8
<b>Hardware specifications</b>	<b>8</b>
Operating system	9
System's security	9
Minimal/Optimal C.U./gateway hardware requirements	10
<b>ConnectorIO's User Interface overview</b>	<b>11</b>
Data visualization options	11
Extensibility - Adding new devices and Software plugins	12
Device auto-discovery	12
Plug & Play for new devices	12
Device Control Panel	13
Device Channels	14
Device management	14
Device Automations	15
The rule engine	15
<b>ConnectorIO's implementation cost</b>	<b>16</b>
<b>Additional Resources</b>	<b>17</b>



## Brief introduction

ConnectorIO is a cloud-based software solution for commercial and industrial properties.

Unlike traditional systems, we enable the integration of devices from different suppliers, thanks to the open system architecture.

We facilitate the modernization of technical installations where there is no budget for it, offering a SaaS model (integration as a service).

ConnectorIO measures the performance and energy efficiency of the building's technical systems in real-time and generates smart reports for the management.

Thanks to the possibility of a free grouping of digital twins, the user can precisely determine the media costs for tenants or estimate the unit production cost for specific areas in an industrial facility or commercial building.



ConnectorIO provides end-user comfort and reduces the technical operation risks in real-estate.

## What is ConnectorIO?

[ConnectorIO](#) is a **Building Automation Software** that allows you to integrate HVAC systems, media meters, or smart sensors in a **single control system**, optimize it with **smart automation** and generate **system efficiency and building's sustainability reports** for the property managers or owners.

### ConnectorIO's goal

Our mission is to **reduce the building's environmental footprint by efficiently managing the systems within.**

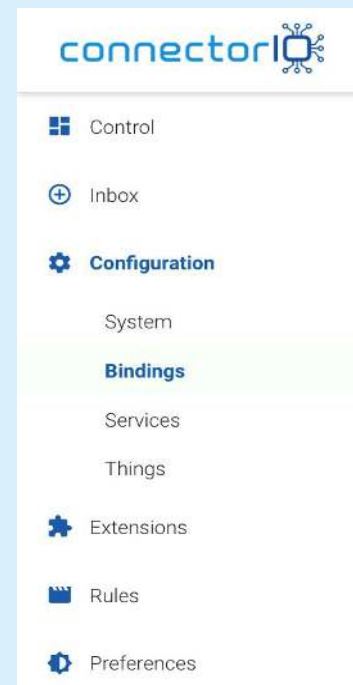
We serve  **commercial** (eg. offices, hotels, etc.) and  **industrial** properties (factories, logistic halls).

### What can be automated with ConnectorIO?

We combine and integrate the following systems:

- Heating.
- Ventilation.
- Air-conditioners.
- Lighting.
- Media consumption meters, such as water, gas, electricity, heat, and cooling.
- Cable installations and actuators.
- Motion and presence sensors, access control.
- Weather information.

ConnectorIO enables automation through **intelligent rules** and **dependencies** between systems. This allows for **efficient use of energy** by these devices and systems.

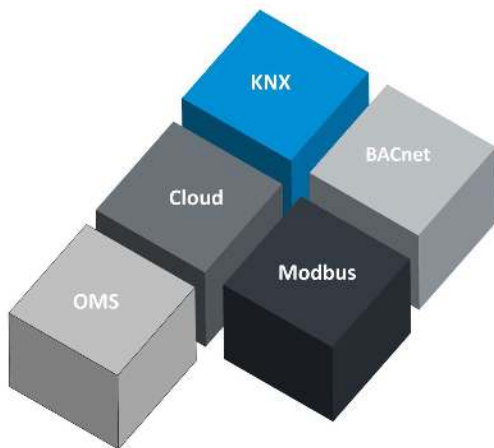


We provide the ability to **monitor** and **visualize** data from these devices, **data cloud-backup** and an intelligent device **optimization** based on operation patterns analyzed in **real-time on big-data**.

## How is ConnectorIO different from a traditional BAS/BMS?

ConnectorIO's approach is different in a way, we are offering an **open system** not tied to a specific **connection** type or device **communication protocol** (BACnet, ModBus, KNX etc.).

**Technically speaking,** Connectorio's idea is quite basic in its form. We offer an intelligent software hub solution that can be embedded in various hardware configurations, provides a unified way of handling 3rd party devices, plus constitutes a communication bridge with the external world and cloud infrastructure.



## ConnectorIO vs. Traditional BAS/BMS comparison

ConnectorIO and traditional Building Management System comparison table.

Feature	Traditional BMS	ConnectorIO
<b>Controller</b>	Closed - dedicated to 1 or 2 defined standards	Open - not tied to a specific communication protocol. Limited only by the physical interfaces. Device auto-discovery possible in the case of some communication standards.
<b>Data streams (input/output)</b>	Limited	Configurable, all available channels can be used if necessary
<b>Gateways - basic (system couplers)</b>	- \$\$\$ expensive (hundreds EU), - combine only 2 standards,	- changes in the configuration of the edge device are moderately simple,

	<ul style="list-style-type: none"> <li>- standards coupling needs to be programmed before operation using industry-specific design tools + firmware update</li> <li>- gateways operate in a distributed way - passive element, limited communication between gateways.</li> </ul>	<ul style="list-style-type: none"> <li>- no direct physical interaction with the edge device is required</li> <li>- changes can be applied through the network</li> </ul>
<b>Gateways - with a webservice</b>	<ul style="list-style-type: none"> <li>- \$\$\$\$ more expensive than the basic solution,</li> <li>- still lots of programming done outside the gateway</li> </ul>	
<b>Operation</b>	Only local - on-premises. Cloud connection possible but limited to functions programmed in device-specific controllers.	Local or Cloud - on or off-premises
<b>Cloud solution</b>	Limited, not flexible: <ul style="list-style-type: none"> <li>- limited to device model and firmware update compatibility.</li> <li>- usually only mainstream providers available or vendor-specific cloud systems.</li> </ul>	Not limited, flexible: <ul style="list-style-type: none"> <li>- works with different mainstream and non-mainstream cloud providers.</li> </ul>
<b>System device updates</b>	Costly and Complicated - demands controller's firmware reprogramming or hardware update and intervention of multiple vendors (integrators, BMS operators) on-premises.	Relatively easily, no need for physical intervention.
<b>System operator competences</b>	Highly skilled professionals needed.	IT intervention not required.
<b>Data storage capabilities</b>	Data is usually stored in an SQL database on a physical storage device located on-premises.	Data can be stored locally or in the cloud.
<b>Stored data range</b>	Limited <ul style="list-style-type: none"> <li>- usually limited to "trend logs" available from the controller's registers and then copied to the SQL database</li> </ul>	Not limited, can be re-programmed, re-defined.
<b>Data retention time</b>	5-60 mins, 1 day, 1 week, 1 month <ul style="list-style-type: none"> <li>- limited by the capacity of local BMS data carrier,</li> <li>- data is being averaged,</li> <li>- data subject to "blurring" of the extreme values,</li> <li>- in case of hardware changes, the storage should be replaced as well.</li> </ul>	Depends only on the resolution of data probing by device and the storage capacity. <ul style="list-style-type: none"> <li>- could be milliseconds and up to months/years of stored data in case of cloud storage.</li> </ul>

## How ConnectorIO is built?

ConnectorIO requires one **Central Unit** to be installed on-premises and/or **ConnectorIO bridges** or “gateways” to connect the different devices. Our system’s infrastructure is presented below.

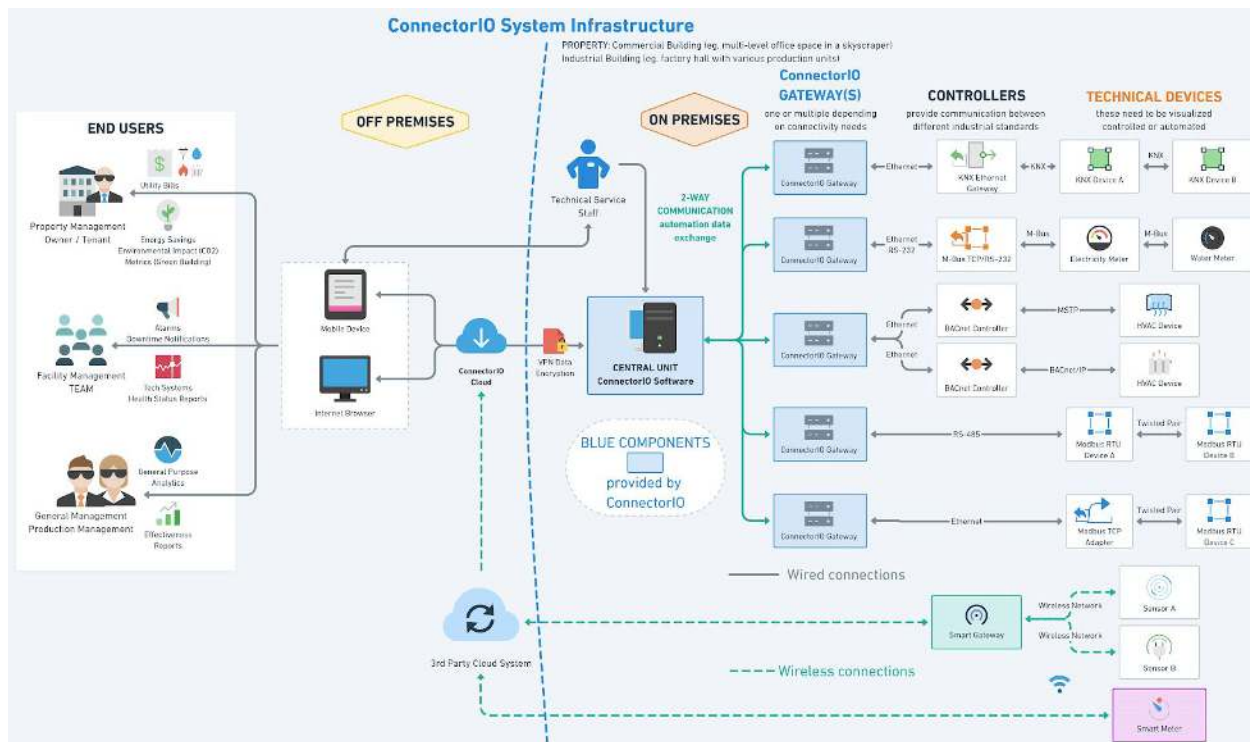
**The Central Unit (C.U.)** - is required only if the system has to be accessed locally and needs to provide open I/O ports. The ConnectorIO Gateway may replace the C.U.

**The ConnectorIO Gateway(s)** - Provides connection with physical systems, allows to group devices in the system by creating **digital twins** and tracks the devices availability.

The Gateways stores basic automation rules definitions to operate the devices in case of unavailable cloud connection.

Multiple gateways are needed in case of a very dispersed end-point devices placement on premises.

## ConnectorIO’s infrastructure map



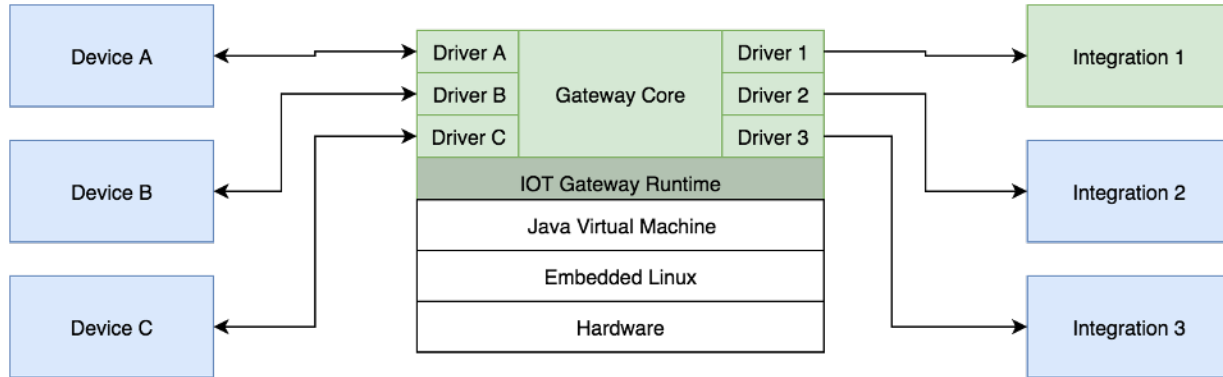
How are different communication standards and devices integrated into ConnectorIO. The **blue items** are provided by ConnectorIO.

For bigger image - [click here](#) (external link, internet connection required)

## Central Unit's internal architecture and key components

The most efficient component of the software is a **smart program bus** that allows connecting devices from various manufacturers, working in communication standards such as:

- KNX, BACnet, Modbus, OMS and wireless standards.



ConnectorIO's software internal architecture and system's logic.

## Hardware specifications

The Central Unit/Bridge of the ConnectorIO's infrastructure is an industrial computer that provides the necessary set of inputs and outputs to connect with industrial system controllers.

ConnectorIO needs at least one **central unit** where the software's engine is installed to perform automation tasks.



ConnectorIO Central Unit/Gateway hardware (eg. Boxer 6405u)

The Central Unit supports the ConnectorIO system in two configurations:



- local, works on-premises
- through the cloud (the central unit is only a device concentrator and data server)

The central unit ports set is versatile enough to cover a wide spectrum of communication standards - from existing KNX / EIB, Modbus, BACnet cable connections to radio standards such as ZWave, ZigBee and long range LoRa (low power).

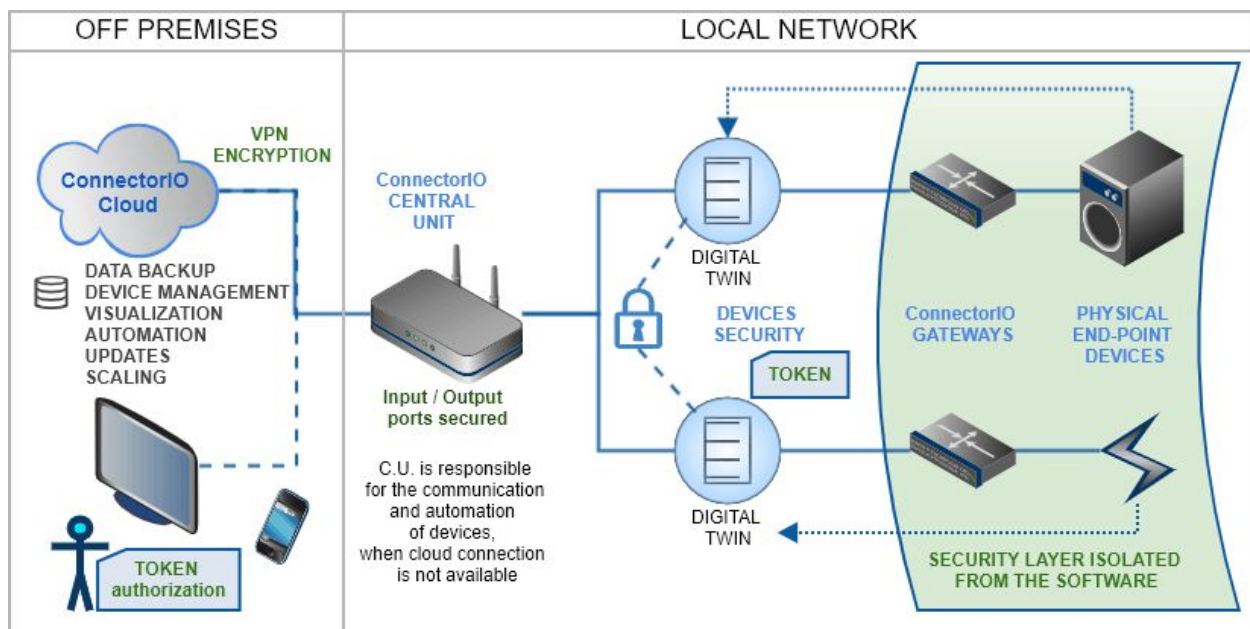
## Operating system

This industrial computer runs a **specialized and secure Linux operating system** and can be produced by any manufacturer, provided that its firmware can be accessed and adapted by ConnectorIO and that it meets the requirements stated below.

## System's security

When connected to ConnectorIO, the system performs a scan and creates a **digital twin** of each connected device. It is a **software layer (firewall)** that **protects the device** against attacks and access to the hardware driver.

The system **never refers directly** to the equipment, but only to the copy - the so-called digital-twin.



To learn more on the system's security measures implemented in ConnectorIO, please refer to:  
 - <https://connectorio.com/bms-cloud-security/>

## Minimal/Optimal C.U./gateway hardware requirements

Hardware requirements for running the ConnectorIO's Software

Feature	Central Unit computer hardware requirements
<b>Processor</b>	2-4 core processor, with 2 threads. recommended: 64-Bit architecture processor with support of floating-point numbers (hard-float). A more efficient processor is required when the frequency of data readings needs to be more granular (eg. a reading every millisecond).
<b>RAM (operating memory)</b>	minimum 2GB, recommended 4GB
<b>Disk drive</b>	minimum 4 GB, recommended 8 GB type: eMMC or SSD used for data buffering tasks in case of network connection issues <b>SD cards are not recommended</b> due to data integrity problems during power failures and lower lifetime
<b>Communication ports</b>	Depends on the systems that need to be integrated. Serial ports: eg. RS-232 and/or RS-485. Ports' voltage: to be specified. Wired connections: 2-3 cords/line (3 for ground).
<b>Network communication</b>	Twisted-pair wiring - traditional Ethernet network cable. Number: depends on the number of devices to be controlled. More ethernet ports needed if more addresses, masks, and subnets are required.
<b>Operating temperatures</b>	Industrial grade computer - operating temperatures range <b>-20...+50°C</b>

## Connetorio's User Interface overview

### Data visualization options



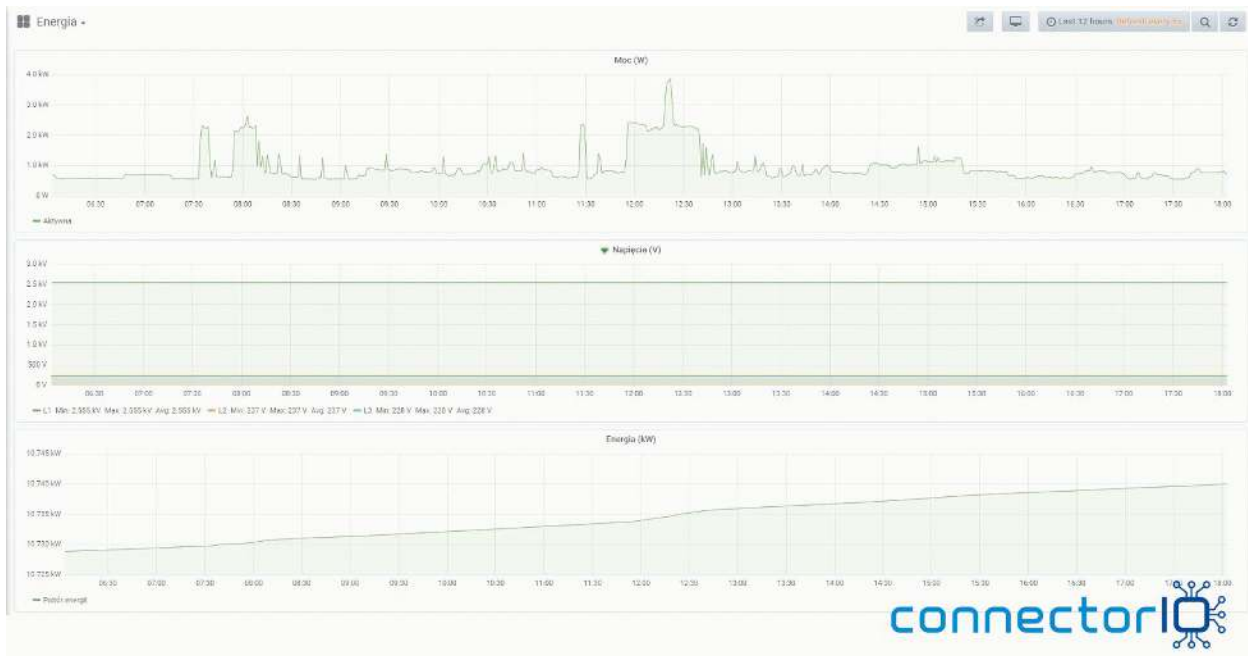
Our system enables you to visualize these types of graphs:

- Electricity graphs ([link-1](#), [link-2](#), [link-3](#), [link-4](#)),
  - Electric energy meters ([link](#)), and electric boards ([link](#)).
  - Gas consumption ([link-1](#), [link-2](#)).
  - Temperatures ([link-1](#), [link-2](#), [link-3](#)).
  - Water valves and filters.
  - Air flows and HVAC unit sections ([link](#)).
  - Floor plans with superposed data ([link](#)).
- ... and more **custom-made** graphs and visualizations designed for you.

Other system features related to data visualization:

- Filtering by: building, floor, pre-defined zones, and filtering by device types.
- Data export to .json/.csv format ([link](#)).

Here's a sample display from electrical units with Power (W), Voltage (V) and Energy (kW/s).



## Extensibility - Adding new devices and Software plugins

The ConnectorIO's user interface is **easily customizable** and extensible through **additional plugins**.

With these extensions you can add new features, like new **communication protocols**: wired or wireless for **recently added devices**.

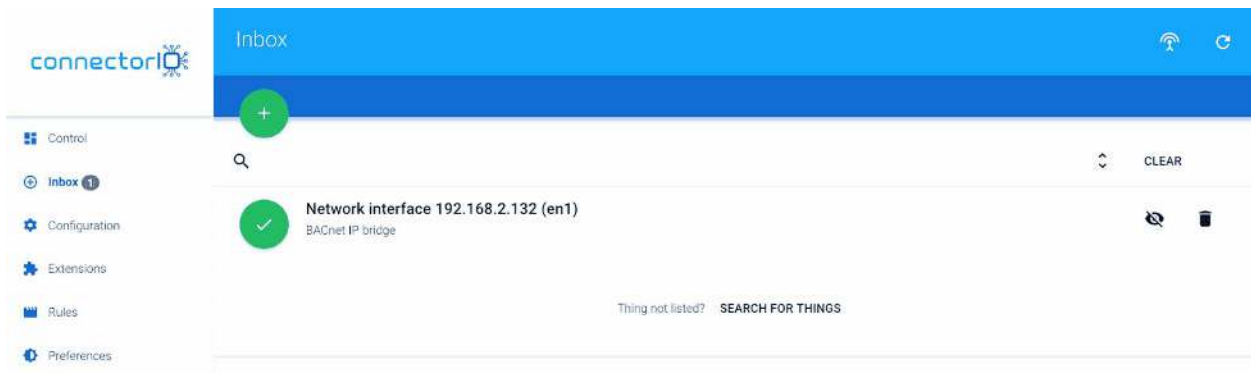
### Device auto-discovery

New devices are **automatically scanned** and **auto-discovered** in the system through the Inbox functionality (provided that the communication protocol of the device allows it).

### Plug & Play for new devices



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.



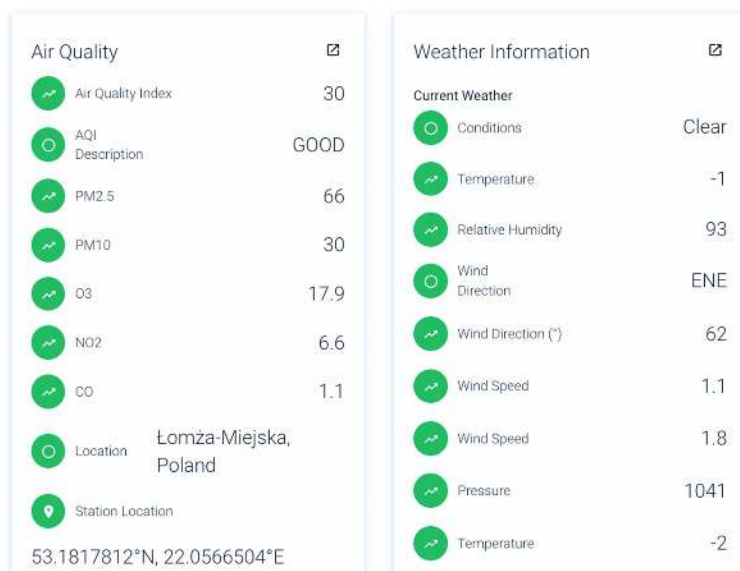
Device auto-discovery feature and scanning of data channels ([link to large image](#)).

- Standard devices which can communicate back are recognized and listed in "Inbox".
- The process is fully automatic, the end user is responsible for accepting of elements.
- The scanning process can be started and repeated multiple times.

## Device Control Panel

In the main control panel users can do the following:

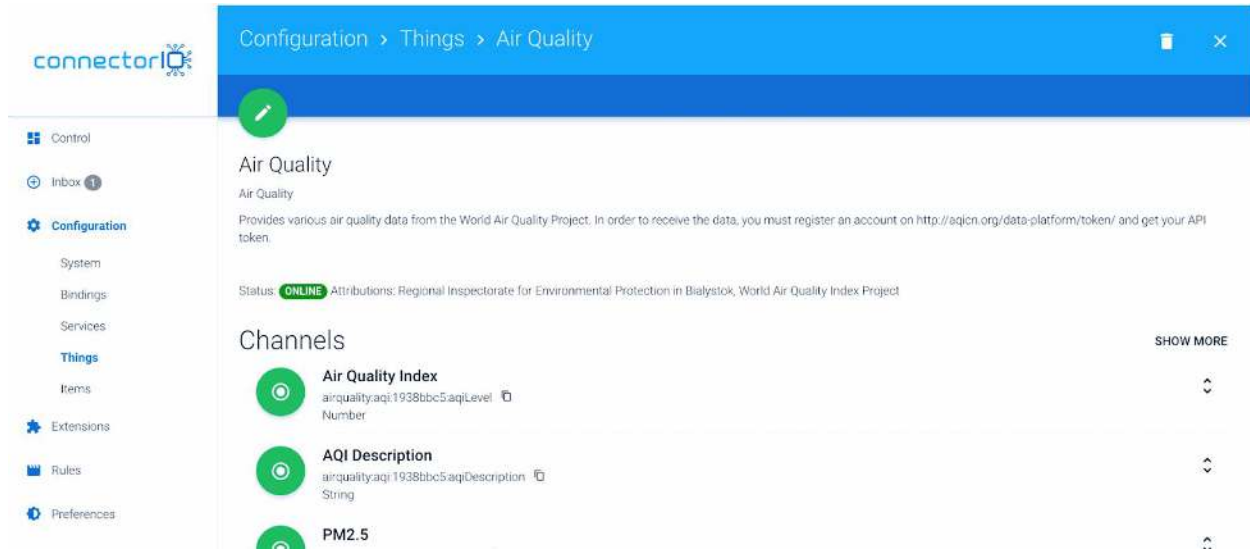
- Users can see the state of inputs and outputs grouped by device.
- In the case of writeable inputs, it's possible to switch them manually.



([link](#))

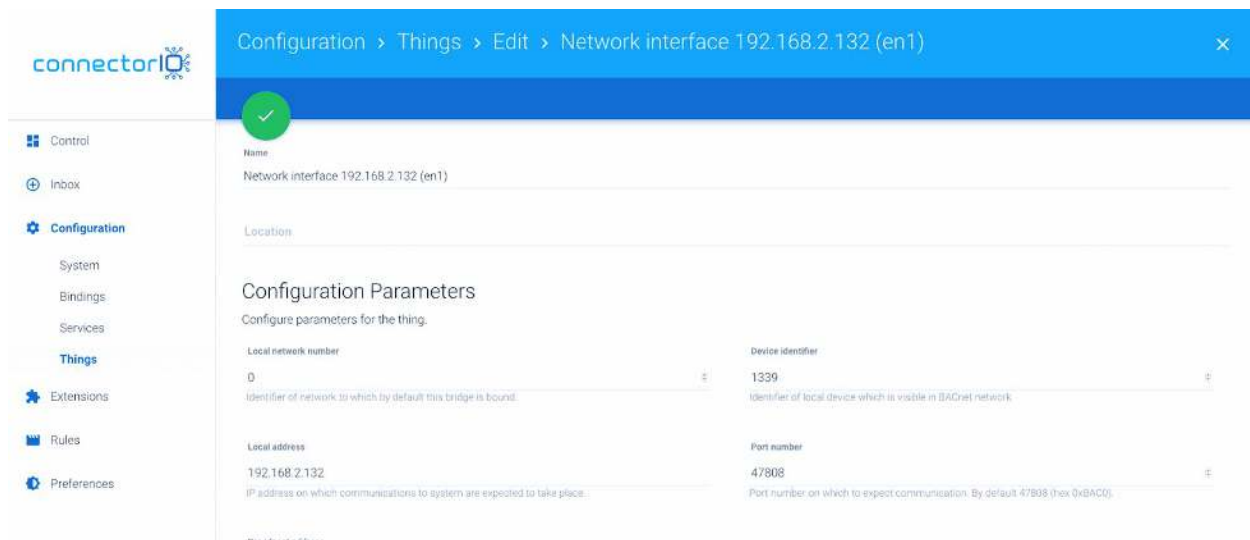
## Device Channels

- Each device has multiple channels which can be linked or not.
- If the appliance needs only a few there is no need to track all of them.



## Device management

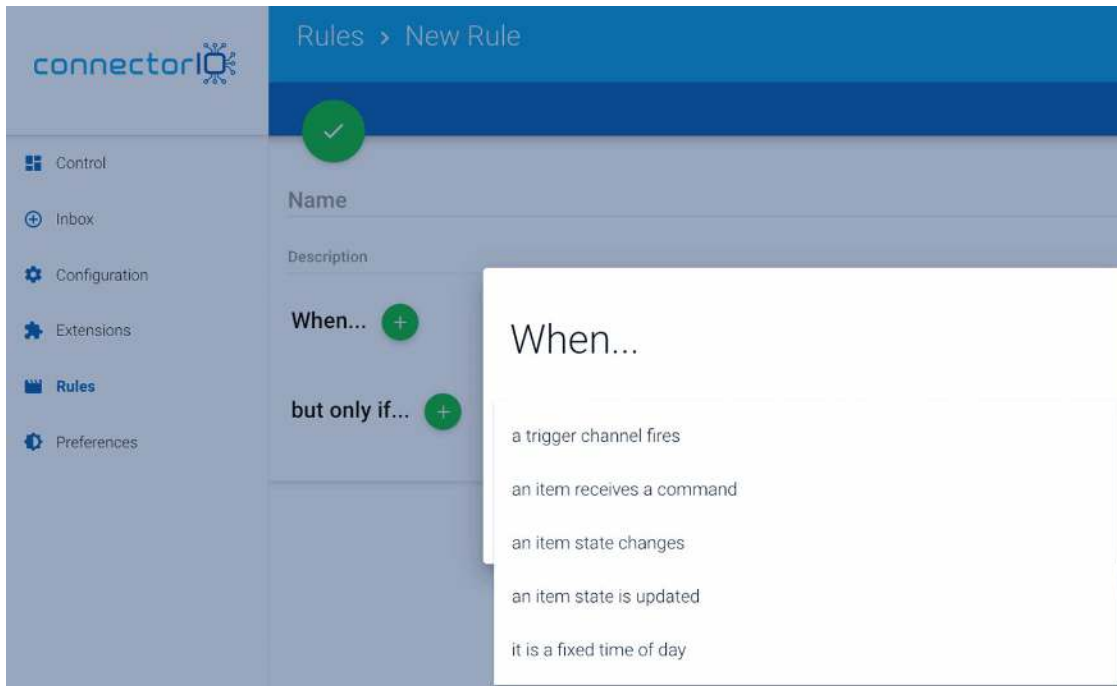
- In case of changes every device and its settings can be edited to match new configuration.
- There is no need to reconfigure anything but eg. IP address.



Device configuration and management menu ([link to large image](#)).

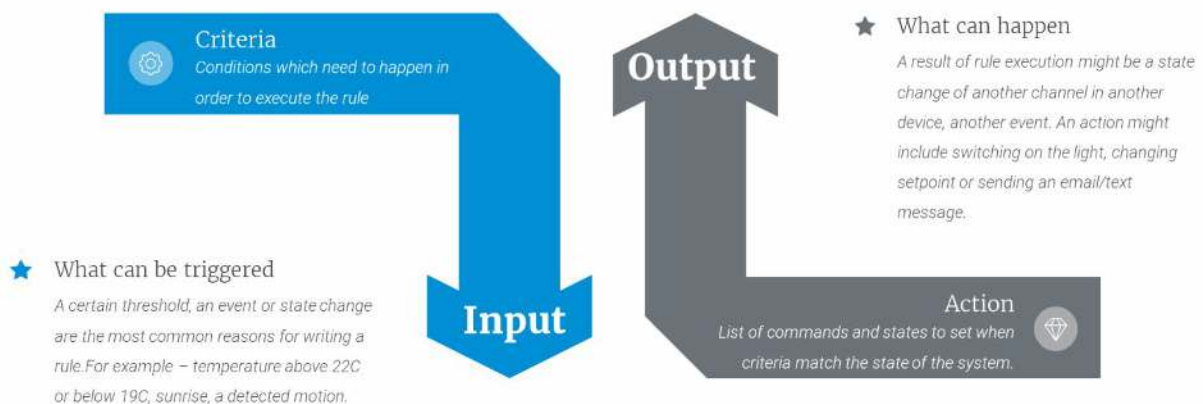
## Device Automations

The system automation is applied by defining a set of rules and interaction between the devices.



Possible rules ([link to large image](#)).

### The rule engine



The rule engine - automation definition ([link to large image](#)).

**Criteria** - specify the events that trigger a rule execution.

**Condition** - add filters for rule execution. Actions of the rule will be executed only if event data satisfies all conditions.

**Action** - perform actual operations in ConnectorIO. If more than one action is specified in a rule they will be executed sequentially.

## ConnectorIO's implementation cost

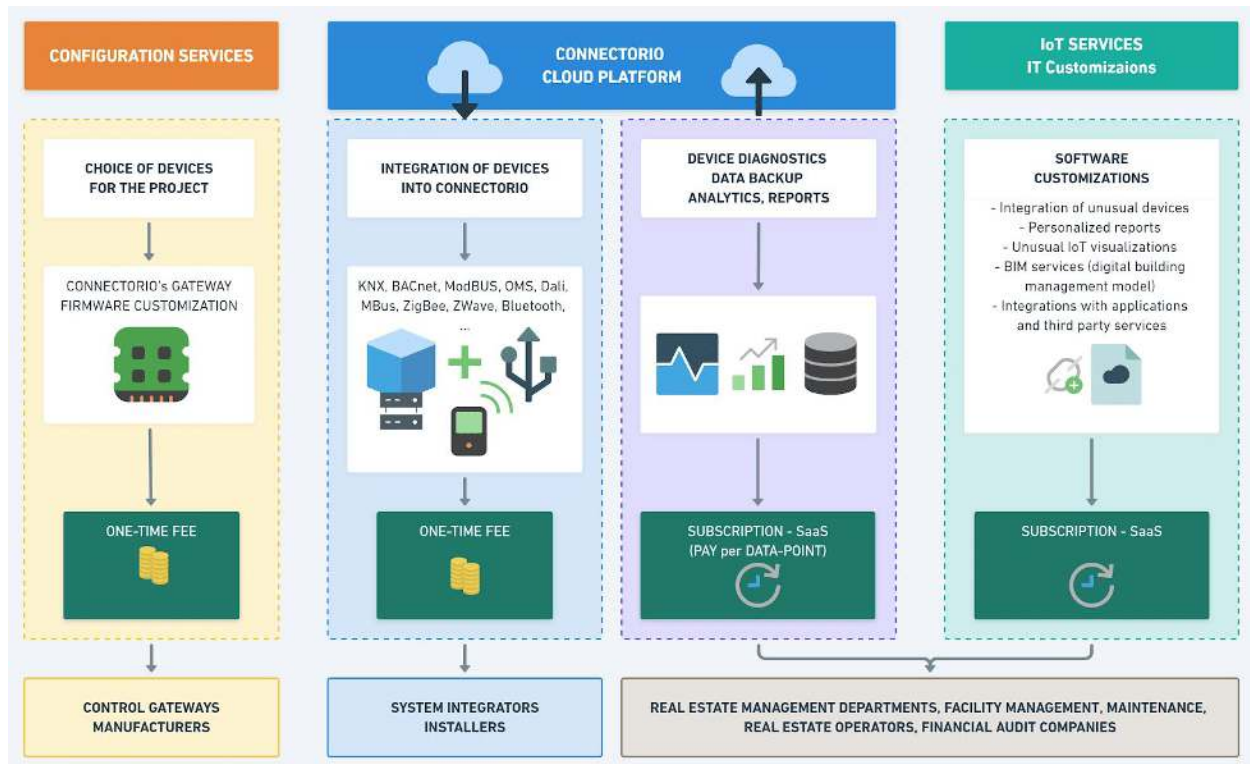
### Minimum installation cost:

- a control device including a license.

### Service cost:

- a subscription fee based on the amount of integrated data points and the sampling frequency of the data.

The payment model is described on this illustration



ConnectorIO's payment model ([link to larger image](#)).





## Additional Resources

- a. ConnectorIO Cloud solutions page:
  - <https://connectorio.com/cloud-solutions-data-export/>
- b. Description of common industrial communication standards that work with ConnectorIO:
  - <https://connectorio.com/common-industrial-standards/>
- c. ConnectorIO's hardware requirements and infrastructure map
  - <https://connectorio.com/hardware-infrastructure/>
- d. More about ConnectorIO's software architecture:
  - <https://connectorio.com/connectorio-software-for-beginners/>
- e. ConnectorIO device management interface
  - <https://connectorio.com/connectorio-user-interface/>
- f. System security measures implemented in ConnectorIO
  - <https://connectorio.com/bms-cloud-security/>



END OF DOCUMENT