

# N3uron

Industrial IoT connectivity solutions

# N3uron Specifications

# Specifications

## Accessibility

The number of clients/developers that have access to the platform is unlimited, and their access can be done from any device, regardless of the operating system used, including iOS and Android.

No additional software installation is required on client/developer's devices to access the platform from the Web, except for the common web browsers.

Updates are instantaneous. Once a change is made to the project, this one will be updated automatically, avoiding costs for maintenance.

## Scalability

The system must be modular and scalable, allowing future amendments to be integrated fastly and efficiently, with a low impact on the running system production.

## Multi-platform

The system must be able to run on the most widespread versions of Windows and Linux and lightweight devices such as Raspberry Pi and ARM.

## Databases

The system can simultaneously connect to as many databases as the project requires. The platform will use MongoDB as a database for storing historical data. The software should also be able to connect to the most popular SQL databases, such as SQL Server, MySQL, MariaDB and Postgre.

## Modular architecture

The software must be modular so that its license covers only the project's real needs, being able to acquire additional modules in case of requiring an increase in functionality in the future.

## Connectivity

The platform will have connectors for the most widespread protocols, including:

- Modbus RTU/TCP, server and client
- DNP3, server and client
- Profinet
- IEC 60870-5-102
- OPC XML client and DA
- OPC UA Client and Server
- MQTT client, continue to subscribe and publish data via MQTT to any Broker, including AWS IoT core, Azure IoT Hub and Google IoT Core, among others and supporting versions 3.1, 3.1.1 or 5.
- MQTT Spark Plug B.

- REST server allows access to third-party services through REST API, HTTP and HTTPS connections. Any data will be exposed by this means, regardless of whether they are real-time or historical values, both for variables and alarms. In addition, other information may also be exchanged and modified, such as that related to system diagnosis, backup and license management, etc.
- REST client, the platform will allow access through REST API to any service that has this possibility, including at least the use of the following methods: POST, PUT and PATCH. The connection will be possible via HTTP and HTTPS.

In addition, the software will have the possibility of developing connectors for any other protocol, either through a specific module for it or with scripting through the use of public and private libraries.

## Import and export files

The platform will allow importing and exporting files in CSV and XML format, locally or through FTP and SFTP servers.

## Licensing

The licensing of the software will not be subject to any limitation, license per server and being from the first moment:

- Unlimited in number of tags.
- Unlimited in clients and users.
- Unlimited in number of devices.
- Unlimited connections.

## Design and development

The design and development tool will be free and won't require any license, and will be accessible from any remote computer with no need to install additional software. It will also facilitate object-oriented development.

With a responsive design, each window will be capable of adapting itself perfectly to the client's device size with no loss in resolution. SVG vector files are recommended for a higher quality resolution.

## Authentication

Different levels of access and user authentications are allowed from the designer and the tool. It's also possible to use external authentication systems, such as Windows Active Directory.

## Integrity

The historian tools must have Store & Forward functionality. This way, historian-data integrity is guaranteed even when there's communication loss with the database.

## Backups

The system must have the option of storing and restoring backup copies of all the information necessary to guarantee the return to production or the migration of the server in less than an hour.

## Event Log

The system must include a registry that allows troubleshooting and debugging any problem or incident.

## Registration of events

The system must include a record that allows diagnosing and debugging any problem or incident.

## Security and registration

System security will be based on users, roles and security zones (variable groups and/or hosts). All communications and access to the system will be encrypted through TLS and HTTPS, using their own certificates or those issued by an authorized entity.

## Redundancy

The SCADA must support redundancy at the level of the server itself, as well as in terms of connections with other servers and/or devices, such as OPC, REST, MQTT brokers, SQL databases, etc.

## Reports

The tool will have a reporting system whose content is dynamic.

## Alarms

The system will have a whole tool for the management, analysis, recognition, storage and notification of alarms and events. Existing at least two different channels for sending notifications: email and SMS.

## Historian

The platform will include an Historian to store historical data, that will use MongoDB as a database, which must support partitioning as well as old data pruning.

The graphic representation of historical data must be quick and simple to implement, being these data visible from the configuration module itself, through a chart or in tabular form, being also possible to export them in a CSV file.

## Scripting

The software provides a module for generating scripts using one of the most common languages and executing them through events or periodic triggers.

## Technical support

The software provider must provide technical support that guarantees a response time of less than 24 hour in case of an incident and less than 48 hours in case of a query. The contractor must include in its proposal the support and maintenance of the platform for a period of not less than three years.

## Technology

The system must use standard technologies familiar to IT departments, such as Node.js, HTML5, CSS, SQL, MQTT, OPC UA, REST API, etc.

## Manage centralized

An application or service will be available to enable centralized access to every node running the software. Being possible to monitor the status of each of the nodes, as well as remote access not only to the node itself in which the software is running but also to computers and systems on the same local network as the node, in order to facilitate the early resolution of problems, all without being necessary to open any inbound ports in the plant firewall.