

Product specification



Version: 18.09.2023

Product description: Smart Sensor Integration Framework „dmc-ssi“

1. Vendor

Address	Valentin-Linhof-Straße 8 81829 Munich Germany	Technologiepark 32 33100 Paderborn Germany
Telephone-Nr.	+49 89 42774 - 0	+49 89 42774 - 150
Telefax-Nr.	+49 89 42774 - 199	
Mail:	info@dmc-smartsystems.dmc-group.com	

2. Product description

The Smart Sensor Integration Framework "dmc-ssi" is designed to collect and forward data. It consists of edge devices on one side and a remote manager on the other side and is based on the Macchina.io IOT framework. The collected data is presented using an app created via the low-code development platform Mendix.

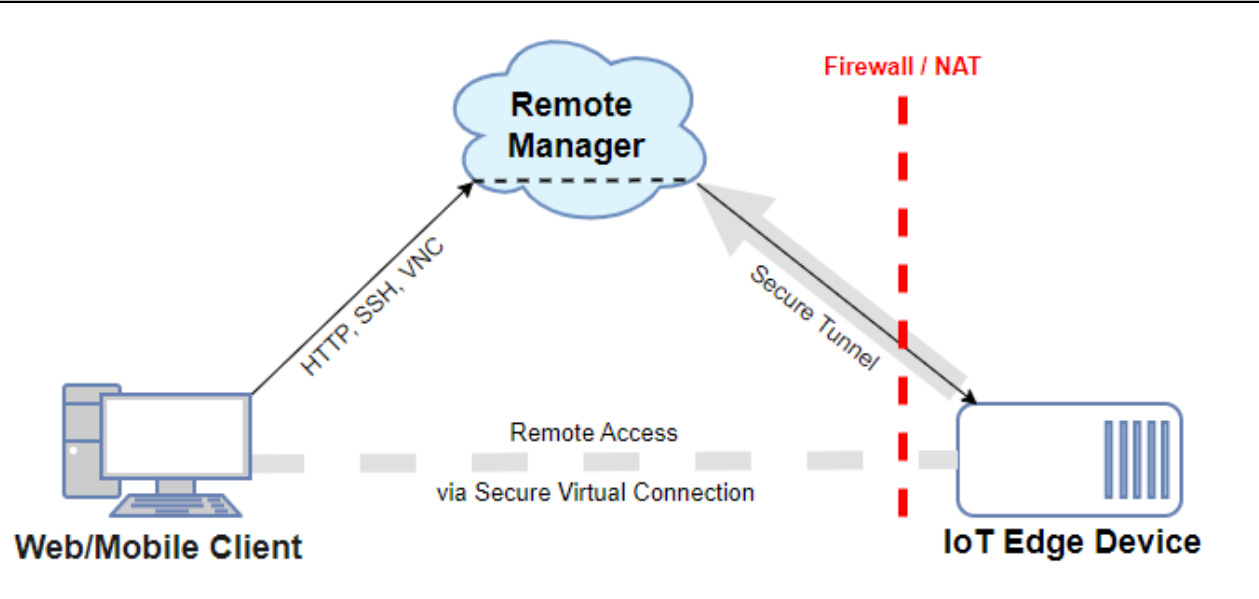
Through the integrated macchina.io Edge SDK, the Edge Devices allow any data functions (collect, sort, filter, aggregate, integrate, ...) to be easily programmed and installed or exchanged at runtime. Especially by providing appropriate basic functionalities for many sensor types and protocols, dmc-ssi is ideally suited to integrate arbitrary sensor technology in a simple and straightforward way. The dmc-ssi edge devices already include a cache mechanism and automated forwarding for sensor data to a dmc-ssi remote manager. In case the connection is disrupted, this functionality protects against data loss.

If the data from the edge device is to be forwarded in a particularly secure manner, for example over the Internet or across firewall boundaries, the connection to the Remote Manager is recommended. The communication is realized via a secured and encrypted web tunnel and allows web (HTTP), shell (SSH), file transfer (SCP, SFTP), remote desktop (VNC, RDP) and other protocols. It should be emphasized that no firewalls with port forwarding, scarce public IPv4 addresses or complex and inflexible VPNs are required.

Our Sensor Integration Framework extends the functionality of Remote Manager by automated reception of sensor data, storage in a time series based database, data access via a WebUI* or RestAPI, as well as an extended device and configuration management compared to Macchina.io.

dmc-ssi is perfectly suited for customized sensor data integration. Separate edge devices can be used to create non-invasive solutions, but the dmc-ssi edge device functionality can also be integrated into different hardware components of your machines, enabling the implementation of a wide range of Industrial IoT applications.

3. Architecture



4. IoT Edge Device

Runtime Environments

- JavaScript (V8)
- C++
- Web User Interface

Supported generic sensors and device APIs

- Accelerometer, Magnetoscope, Gyroscope, Temperature, Humidity, Light/Brightness, Pressure/Power
- Switch, Trigger, Rotary encoder
- GPIO, LED
- GPS/GNSS
- Barcode Scanner

- Digi XBee® Sensoren (Temperature, Humidity, Light)
- Tinkerforge (Temperature, Humidity, Light, Pressure, Movement, Rotary Encoder, DC Engine, GPS/GNSS)
- SimpleLink™ SensorTag (Bluetooth LE, Temperature, Humidity, Light, Pressure, Accelerometer)
- Bosch XDK (Bluetooth LE, Temperature, Humidity, Light, Pressure, Accelerometer, Gyroscope, Magnetometer, Button)
- Bosch CISS (USB, Temperature, Humidity, Light, Pressure, Accelerometer, Gyroscope, Magnetometer)
- Linux GPIO, LED

Supported Protocols

- | | |
|--------------------------------------|---------------------------------------|
| • HTTP, HTTPS (Client und Server) | • XBee® API |
| • WebSocket (Client und Server) | • CAN (basierend auf Linux SocketCAN) |
| • MQTT, MQTTS (Client) | • CANopen |
| • Modbus (RTU, TCP) | • OPC-UA |
| • UDP (6LoWPAN) | • S7 (PLC) |
| • Generic Serial Port (UART, RS-232) | • Bluetooth LE (Client) |
| • LoraWAN (via Cirpstack Gateway) | • |

Extension and Access Points	
<ul style="list-style-type: none"> • Code Generator • TCP/Socket Transport • http Transport • REST Transport • JSON-RPC Transport • JSON-RPC via MQTT Transport • SOAP Transport • WSDL/XSD Code Generator • Sensor Configurator 	
Open Service Platforms	
<ul style="list-style-type: none"> • Core Framework (Extension via Bundles & Services) • User-Authentication Service • Web Application Server • JavaScript Integration • Extensible Command-Line Interface (CLI) Framework • JavaScript Script Scheduler • Mail Delivery • Bundle Signing • Extended database based Authentication Service • LDAP-Integration 	
Standard services	
<ul style="list-style-type: none"> • Device Status • Network Environment • Mobile Connection • Web-Events • Code for measuring units 	
REST APIs	
<ul style="list-style-type: none"> • Basic Device Management • Bundle Management • User Management • Sensors and Devices 	
Databases	
<ul style="list-style-type: none"> • SQLite • Redis 	
Cloud Service Connection	
<ul style="list-style-type: none"> • AWS IoT (via MQTT) • Azure IoT (via MQTT) • Secure Remote-Access and -Management via macchina.io/dmc-ssi REMOTE MANAGER 	
Technical Requirements	
Device Operating System	Linux
Device CPU Architecture	ARM, X86, MIPS
Storage (Minimum)	64MB RAM, 64MB Flash (32MB RAM without JavaScript Engine)
Network	Ethernet, Wi-Fi, Mobil
Compiler Toolchain (C++)	GCC 4.9+, Clang 3.4+
Host Operating System	Linux, MacOS
Software Requirements	GNU Make 3.81+ Linux: GNU C++ 5.0+ / Clang++ 3.4 + (C++14) macOS: Clang++ (Apple LLVM) 10.0+ OpenSSL headers and libraries Python 2.7 (for building V8)

5. REMOTE MANAGER

Standard macchina.io Features

- TLS encrypted Connection to Edge Device and Clients
- Cloud Ready for AWS, Azure or On-Premise (linux)
- Separate Load-Balancer-Setup
- Integration of proprietary Hardware through macchina.io Edge Device Agent
- Device Management
- Rollbased Access control

dmc-ssi Features

- Time-series based Database (InfluxDB)
- Automated Data Reception of SSI Edge Devices
- RestAPI for Data Transfer
- Docker based Installation (ready for different deployments)
- SSI WebUI (Mendix based)
- Evolved Device Management with save and deploy device configuration
- Evolved data Visualisation
- Condition Monitoring and Alerting

Deployment

- On-Premises (Linux Server/VM)
- dmc-smartsystems - Cloud
- AWS Cloud
- Azure Cloud