KIS.ME

Keep it simple. Manage Everything.

# INTEGRATION GUIDE

Version 7, 06/2024

# 1 Network infrastructure

The devices (e.g. KIS.BOX and KIS.LIGHT) are using a local WiFi infrastructure to establish a connection to the cloud service (KIS.MANAGER). A 2.4 GHz network according to IEEE 802.11 b/g/n is required. The devices support WPA-PSK or WPA-2-PSK (CCMP). Only status information and metadata of the devices will be send to the cloud service. In addition, the devices can be controlled by the cloud service. Access to the cloud service is possible with a personal user account using a browser. The KIS.API can be used for accessing data via third-party applications. Figure 1 shows the overall infrastructure of KIS.ME.

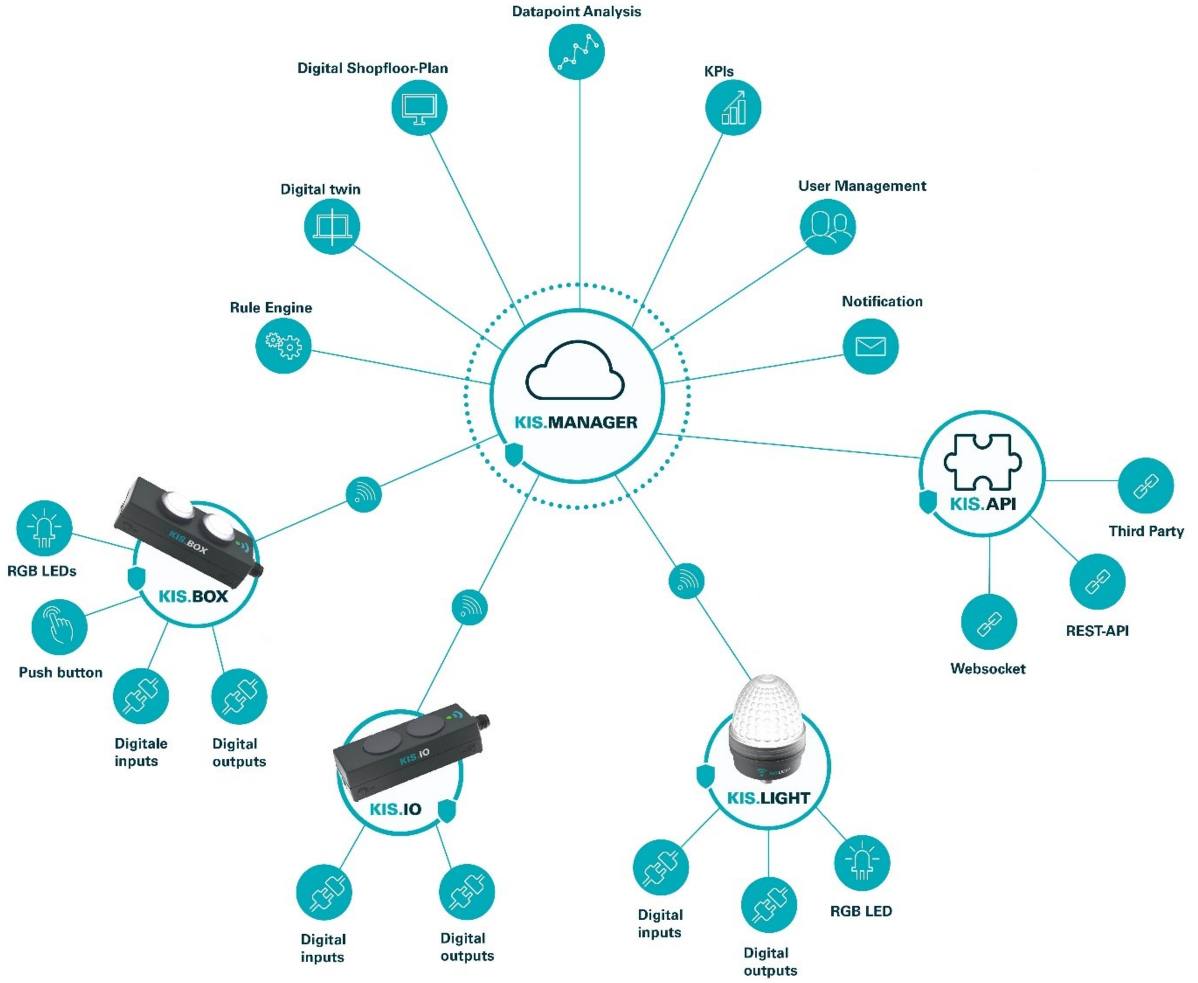

*Figure 1: KIS.ME Infrastruktur*

A Wifi access point is required to use the devices within your network. It is recommended to use a separated network for this (e.g. a guest network). This means that the devices are logically separated from your internal infrastructure. Therefore no access to the internal network is possible. Abbildung 2 shows a common network infrastructure. The logical network infrastructure can be seen in Abbildung 3. A so-called VLAN (Virtual-LAN) or a DMZ (Demilitarized Zone) can be used to limit the logical access.

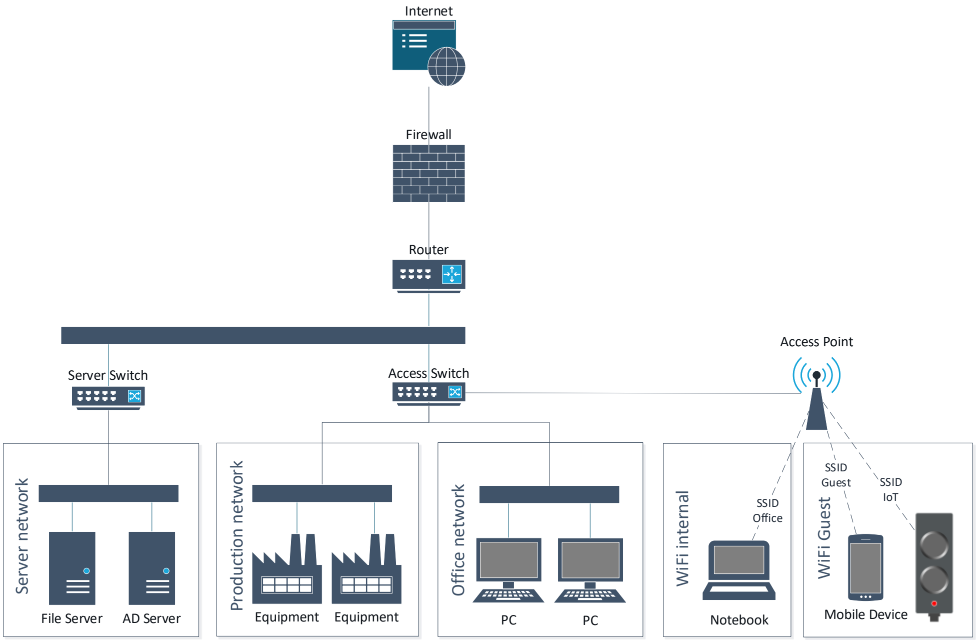

*Abbildung 2: Physikalische Netzwerk Infrastruktur*

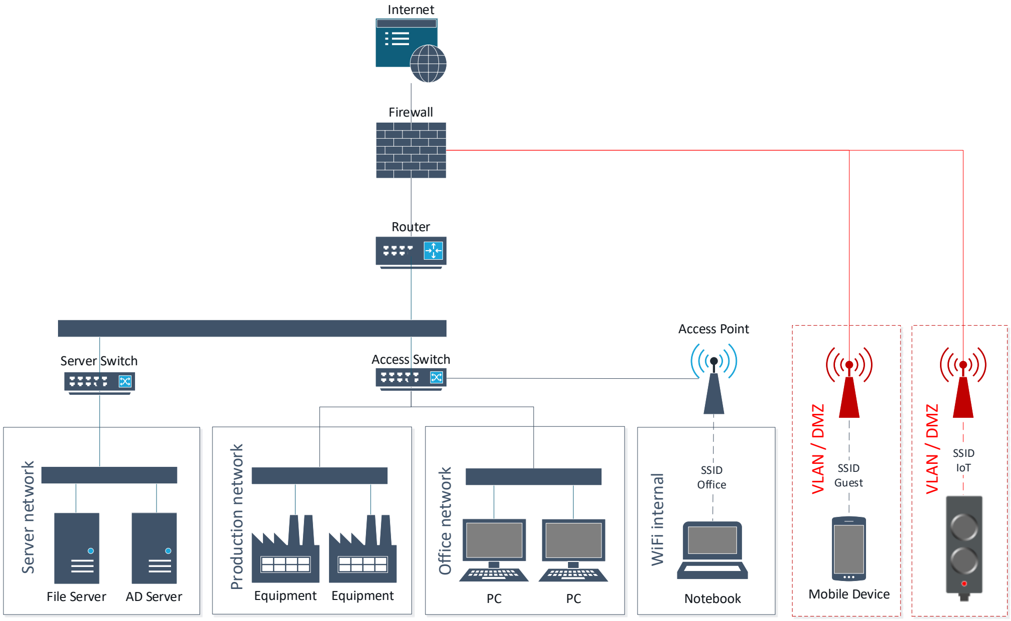

*Abbildung 3: Logische Netzwerk Infrastruktur*

# 2 Firewall settings

## 2.1 KIS.ME device requirements

For the KIS.ME devices to work properly, the usage of a few ports is required. So some settings in the firewall are necessary according to the table below:

*Tabelle 1: Port Übersicht*

| Port | Protokoll | Richtung | Funktion |
|---|---|---|---|
| **-** | ICMP | Incoming | Ping |
| **53** | UDP + TCP | Outgoing | DNS |
| **67** | UDP | Outgoing | DHCP |
| **68** | UDP | Incoming | DHCP |
| **123** | UDP | Outgoing | NTP |
| **443** | TCP | Outgoing | HTTPS |
| **8883** | TCP | Outgoing | MQTT over TLS |

**ICMP**

ICMP ping is not actively used by the devices, but it is allowed in both directions by the device firewall in order to be able to respond to pings (e.g. for network diagnostics).

**DNS: Port 53 (UDP oder TCP, at least one of them)**

Probably only for internal network, depending on the configuration.

**DHCP: Port 67 und 68**

Only required for the local network so that the devices can receive an IP address.

**NTP: Port 123**

Random NTP servers are selected using these DNS names:

- 0.europe.pool.ntp.org
- 1.europe.pool.ntp.org
- 2.europe.pool.ntp.org
- 3.europe.pool.ntp.org

There is no restriction in the firmware. To change the NTP server address, these DNS entries have to be adjusted internally and redirected to the local NTP server if necessary.

**HTTPS: Port 443 (Cloud-API + Firmware Update)**

- iotrafi.centersightcloud.com (KIS.ME Cloud Service)
- bootstrap.centersightcloud.com
- connect.centersightcloud.com
- *.blob.core.windows.net (Firmware Update blob storage)

The Cloud-API is used by the devices to transmit the telemetry data. The data is encrypted by using MQTT over TLS. A certificate-based authentication is used for the Cloud connection.

**MQTT over TLS: Port 8883**

- iotrafi-prod-iot-hub.azure-devices.net (Azure IoT Hub)

The devices send their telemetry data to the cloud server by using MQTT over TLS

## 2.2 KIS.MANAGER Anforderungen

For the KIS.MANAGER to work properly, the usage of a few domains are required. So some settings in the firewall are necessary according to the table below:

*Table 2: Domain Overview*

| **Domain** | **Description** |
|---|---|
| kismanager.kisme.com | KIS.MANAGER Domain |
| kismanager.rafi.de | KIS.MANAGER Domain |
| fonts.googleapis.com | Google fonts |
| api.locize.io | Locize translation service |
| *.Sectigo.com | Intermediate certificate authority |
| *.Comodoca.com | Intermediate certificate authority |

## 2.3 KIS.API Anforderungen

The KIS.API can be used to communicate with the cloud platform. It includes a REST API to extract and change data and states, as well as a web socket interface for real-time communication. The documentation for the KIS.API can be viewed at https://docs.kisme.com/. It contains a description of the functionalities and guides, for example on the use of web sockets.

To be able to use the KIS.API in your own application, it must be able to reach the domain api.kisme.com externally. Communication takes place via HTTPS: Port 443.

## 2.4 FAQs

### 2.4.1 Devices and local network

**Is it possible to intercept data via the local network or through physical access?**
Security mechanisms such as the deactivation of unused network ports and device interfaces, memory encryption and a local device firewall prevent unauthorised access to the device and its data.

**Where can I check whether the Wifi signal strength is sufficient?**
The datapoint *wifiSignalStrength* can be viewed in the KIS.MANAGER via the datapoint overview. We recommend a signal strength of -30dBm to -80dBm.

**Is it possible to change the NTP server?**
The servers mentioned in section 2.1 cannot be changed on the device. It is recommended to allow access to the specified servers.

### 2.4.2 Cloud and security

**Which data are transmitted to the Cloud?**
Only telemetry data of the KIS.ME devices will be transmitted to the Cloud
(e.g. Button_pressed, Set_Color_green etc ...).

**Who has access to the data and for what purpose?**

Only the tenant admin has access to the data. The admin has the possibility to add additional users with defined rights if this is necessary.

**Is it ensured that the data will be processed in Europe (storage)?**

The data are stored in the server infrastructure of the Microsoft Azure Cloud based in Ireland.

**How and by whom is the Cloud-API used and how is the interface secured?**

The Cloud-API is used by the devices to transmit their telemetry data. The data is encrypted by using MQTT over TLS. A certificate-based authentication is used for the cloud connection.

### 2.4.3 Multi-Tenancy

**How does the multi-tenancy separation take place?**

There are different types of a tenancy separation. Since it is a distributed system that uses different persistence services, the type of separation depends on the respective storage backend. An example is the use of separate database schemas for different tenants in SQL databases.

**Is all data stored in encrypted form?**

Yes, all data is saved in encrypted form. What the encryption looks like in practice depends on the respective storage backend.

**What effects does it have on other tenants if a client's admin access has been compromised?**

Should an attacker manage to obtain admin rights within a tenant, this would not affect other tenants.

### 2.4.4 KIS.API

**Is it ensured that the KIS.API can only be accessed by authorised persons?**

To access the API, credentials are created via the KIS.MANAGER. These can only be viewed by KIS.MANAGER admins. Unauthorized access of credentials in your own application should be prevented, for example via secrets management.

**How does the Websocket interface work??**

A web socket interface can be created via the REST API, in which the desired updates (rules or data points) are then sent. These can be accessed via a STOMP client. An example implementation can be found in the documentation.