plus10
IT Documentation

Software as a Service & On-Premise Deployment

IT specifications, security and compliance overview

Version 1.1 • July 2026

Table of contents

1 Introduction

This document gives an overview of the IT requirements for the plus10 software. It describes our recommended deployment options; most options can be adjusted to individual requirements. It covers both the cloud-based Software as a Service (SaaS) offering and on-premise / customer-managed deployments, together with the security and data-protection measures relevant to a deployment.

The plus10 software consists of several services, mainly:

- DataCollector — collects data from different sources, pre-processes it and sends it to the central services.
- API / Analysis service — processes and analyzes data and provides the results.
- Authentication — authenticates users and clients for the plus10 services.
- Database — stores collected and processed data.

The services other than the DataCollector form the plus10 backend and are referred to as the central services. The deployment location of the central services is flexible and depends on customer requirements; the following options are generally possible:

- Software as a Service, managed by plus10 (standard).
- Virtual machine / dedicated host (local or any cloud), managed by the customer (on-premise).

2 Topology

The plus10 solution consists of two parts. The central services (the backend) provide the API and analysis service, the authentication service and the database; they run either as a plus10-managed SaaS or on a customer-managed host (see chapter 3). One or more DataCollector instances run in the factory, close to the machines: they acquire data from the data sources (PLCs and other systems), pre-process it and send it to the central services. A typical installation combines one backend with one or several DataCollector instances, depending on the number and location of the machines to be connected.

The DataCollector, which collects data from machines in the factory, must be deployed as near to the data source as possible in terms of network topology. This is because the timestamp of the collected data is set in the DataCollector — our experience is that the clocks of PLCs are often not synchronized and/or are drifting. Consequently, a near-deterministic network latency between the data source and the DataCollector is required.

The standard solution is to deploy the DataCollector on a dedicated computer connected to the machine network (best case: connected to the switch to which the PLCs are connected). We recommend using an Industrial PC (IPC). Deploying the DataCollector on a virtual machine is also a common option, provided the network connection to the data sources is reliable and low-latency. Depending on the number and location of the machines to be connected, multiple instances can be optimal. Please consider an uninterruptible power supply (UPS) for IPCs to ensure proper shutdowns when machine main switches are turned off.

The diagrams below give an overview of the standard IT service topology, including the required ports. Figure 1 shows the factory part (identical for SaaS and on-premise). Figure 2 shows the on-premise central part, which is not required for standard cloud-based deployments.

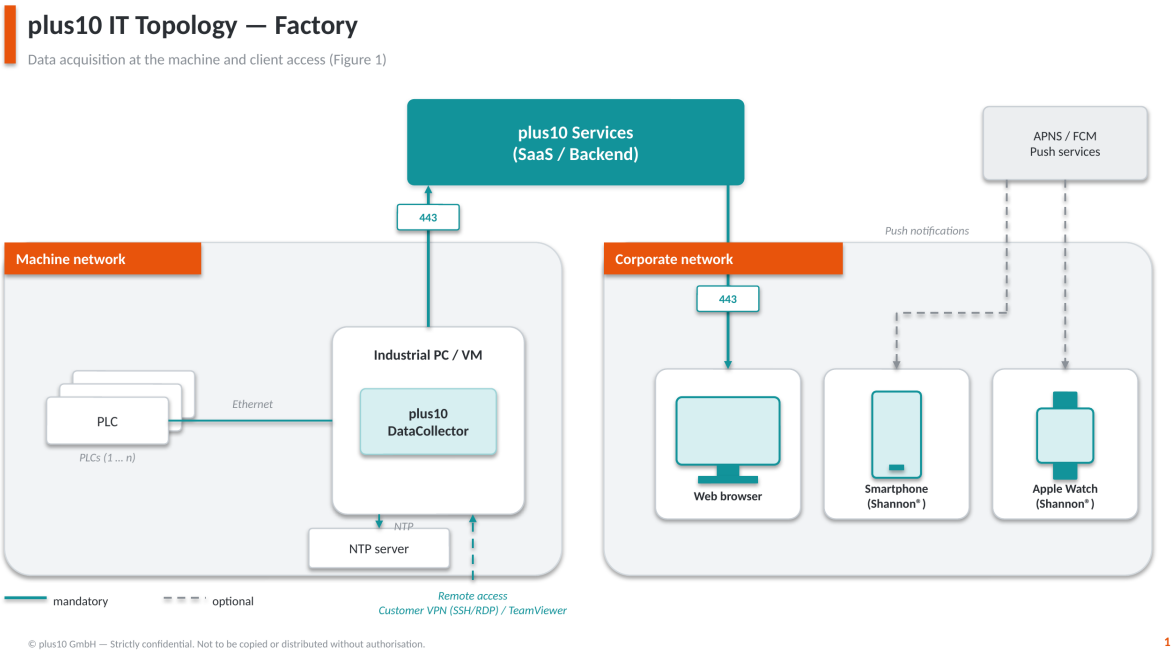

Figure 1 — IT topology (factory part)

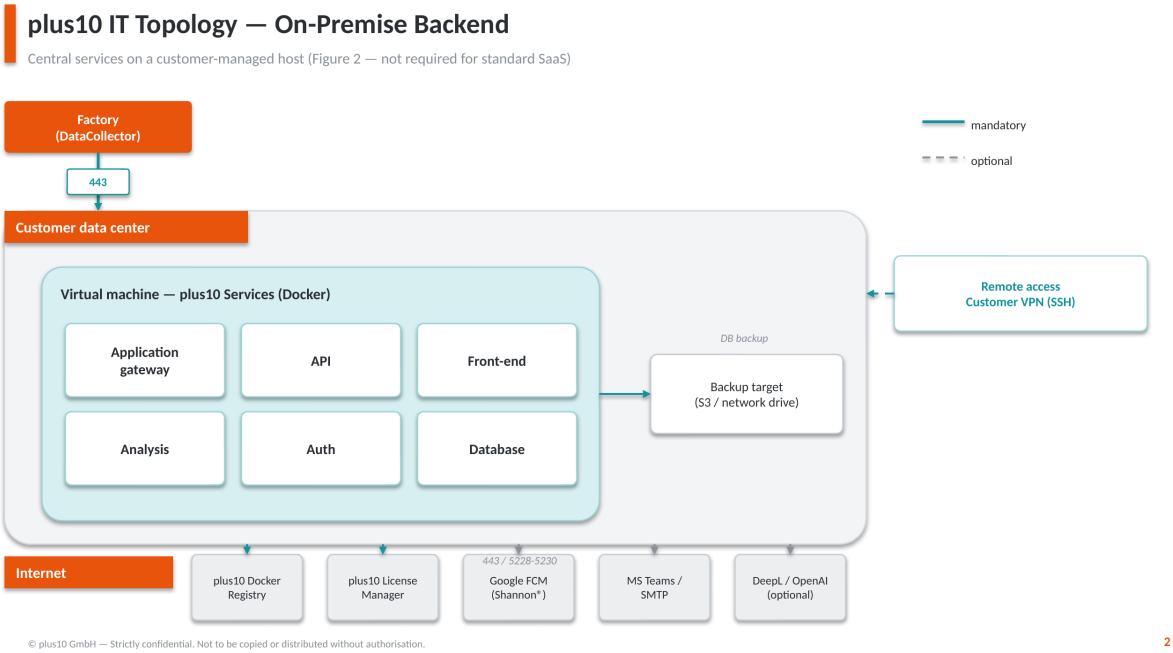

Figure 2 — On-premise IT topology (central part; not required for standard cloud deployments)

3 Hosting and connection

The plus10 backend (central services, see chapter 2) can be hosted either as a plus10-managed SaaS or on a customer-managed host. This chapter describes access, encryption, data storage and retention for both options.

3.1 Software as a Service (standard)

With the plus10 SaaS, the application services run in a shared (multi-tenant) environment, while each customer's data is stored in a dedicated, single-tenant database — so your data is isolated from other customers at the database level. We use Microsoft Azure as the cloud provider; all SaaS environments are hosted in the Azure “Germany West Central” region (Frankfurt, Germany). All access points require authentication and all connections are encrypted.

3.1.1 Access

Your instance is reachable via our plus10-webservices.com domain. You will receive a user with administrator rights from us, with which you can create users with specific rights.

3.1.2 Connection

All connections to the cloud environment are encrypted with TLS, and we limit the number of inbound and outbound ports.

3.1.3 Data storage

Your data is stored in a dedicated, single-tenant database. The database is not shared with other customers, so your data is isolated by design.

3.1.4 Data retention

Calculated data (e.g. downtime data) is stored for two years. Raw data (PLC signal measurements and feature measurements) is stored for three months. This configuration can be adjusted to your requirements.

3.1.5 Backup

A data backup is performed every night. By default, we keep the latest two backups. (See section 9.3 for the full backup and disaster-recovery policy.)

3.2 On-premise (customer-managed)

Alternatively, the central services can be deployed on a customer-managed virtual machine or dedicated host (local or in any cloud). In this case the customer provides and maintains the host; the system requirements are described in section 4.2, network configuration in chapter 5, and the installation procedure in chapter 10. Encryption, retention and access principles are equivalent to the SaaS option; backups are configured against a customer-provided target (see section 4.2).

The standard on-premise deployment uses Docker Compose. Deployment on a Kubernetes cluster is possible on request.

4 IT system requirements

In general, the required computing capacities depend on several relative factors. Mainly, the following variables determine the required computing capacities:

- Quantity of data (commonly PLC signals) to be processed per second.
- Period of data retention.

4.1 Data Collector

General system configuration:

- Standard host: dedicated host in the factory network — an Industrial PC or a virtual machine.
- Operating system: Windows 11 or Linux are both standard; on Linux the DataCollector is deployed as a Docker container.

- Software to be installed: plus10 DataCollector (native installation) or Docker (for the containerized deployment).
- Required user rights: install the specified software; start the plus10 DataCollector; restart the system.
- Network ports: at least one port for the PLC network (more if dedicated connections to each PLC are possible/required); one separate port to reach the network where the central services (VM or cloud) are located.
- NTP: the system clock must be synced with an NTP server so that the system timestamp matches the actual time. You can use an NTP server hosted on plus10 central services or your own NTP server.
- Computing capacity: as a best practice, the base load of the computer (including data collection) should not exceed 50%, leaving enough capacity for other services such as anti-virus software.

If you use an Industrial PC provided by plus10, Windows 11 IoT is installed. You can use your own Windows image and/or include the PC in your domain. If you have specific (IT-security) requirements, you can apply them on the IPC. In general, you are responsible for maintaining the host used for the DataCollector (OS updates etc.). If plus10 should fully maintain the host, this can be defined in a separate support contract.

Network port configuration

Service	Protocol	Outgoing port	Incoming port	Description	Optional
API	HTTPS	443	–	–	No
Machine data streaming	TCP	4222	–	Standard (high-performance streaming)	Yes (alt.: WebSocket)
Machine data streaming	WebSocket	443	–	Fallback if 4222 not possible	Yes
NTP	UDP	123			Yes (alt.: internal NTP server)
Remote access (Windows)	RDP		3389		Yes (alt.: TeamViewer)
Remote access (Linux)	SSH		22		Yes (alt.: TeamViewer)
Remote access	TeamViewer	5938/443	–		Yes (alt.: RDP / SSH)

Backend connectivity (DataCollector → central services)

The DataCollector must be able to reach the plus10 backend (central services). The target depends on the deployment:

Deployment	Target (URL / host)	Port	Description
SaaS	https://plus10-webservices.com	443 (4222, 123)	DataCollector connects to the plus10 cloud backend
On-premise	FQDN / IP of the customer-hosted backend	443 (4222, 123)	DataCollector connects to the central services

Deployment	Target (URL / host)	Port	Description
			on the customer host

Example capacities

- 1 machine with one PLC, collecting 1,000 signals at 100 ms sampling rate — CPU 2 cores, 4 GB RAM, 64 GB disk.
- 1 manufacturing line with 5 PLCs, collecting 10,000 signals at 100 ms — CPU 4 cores, 8 GB RAM, 128 GB disk.
- 5 manufacturing lines with 5 PLCs each, collecting 50,000 signals at 100 ms — 5 PCs, each CPU 4 cores, 8 GB RAM, 128 GB disk.

4.2 Central services (on-premise deployments only)

General system configuration:

- Standard host: virtual machine.
- Operating system: x86-64 Linux preferred (good compatibility with Docker).
- Software to be installed: Docker, Docker Compose, plus10 services as Docker images.
- Required user rights: install the specified software; run Docker containers; restart the system.

Mounts. By default, deployment and media data are stored under /opt/plus10-services/ and /opt/plus10-data/. Database data is stored under /var/local/postgres/. It is recommended to create dedicated mounts for /var/local/postgres (database data) and /var/lib/docker (Docker images and container data).

Example capacities (6-month data retention)

- 1 machine with one PLC, 1,000 signals at 100 ms — CPU 4 cores, 16 GB RAM, 265 GB disk.
- 1 manufacturing line with 5 PLCs, 10,000 signals at 100 ms — CPU 4 cores, 32 GB RAM, 768 GB disk.
- 5 manufacturing lines with 5 PLCs, 50,000 signals at 100 ms — CPU 8 cores, 64 GB RAM, 3 TB disk.

Network port configuration

Service	Protocol	Outgoing port	Incoming port	Description	Optional
API, Front-end	HTTPS	–	443	–	No
Machine data streaming	TCP	–	4222	Standard (high-performance streaming)	Yes (alt.: WebSocket)
Machine data streaming	WebSocket	–	443	Fallback if 4222 not possible	Yes
Google Firebase Cloud Messaging	HTTPS/TCP	443, 5228-5230	–	Sending push notifications to clients	Only for Shannon®
Remote access	SSH	–	22	–	Yes

Service	Protocol	Outgoing port	Incoming port	Description	Optional
Healthcheck notifications	HTTPS	443	–	Notification of system failures (HTTPS or SMTP)	Yes
Healthcheck notifications	SMTP	587	–	Notification of system failures (HTTPS or SMTP)	Yes
Docker images	HTTPS	443	–	Download of Docker images	No
License management	HTTPS	443	–	plus10 license validity check	No
NTP	UDP		123	Internal NTP server	Yes

Outbound internet URLs

Service	URL	Port	Description
plus10 Docker Registry	https://plus10.azurecr.io	443	Download plus10 Docker images
License management	https://plus10-webservices.com	443	plus10 license validity check
MS Teams webhook / SMTP	https://outlook.office.com	443 / 5938	plus10 service health checks
Google Firebase Cloud Messaging	Google IP range or FQDN whitelisting	443, 5228-5230	Push notifications (Shannon®)
DeepL	https://api-free.deepl.com	443	Auto-translation (Shannon®)
OpenAI (optional)	https://api.openai.com	443	AI assistant (optional)

Backup

- Target: network folder, S3 bucket, Azure Blob Storage or similar — used to back up the database.
- Example disk space (6-month retention): 1,000 signals → 265 GB; 10,000 signals → 768 GB; 50,000 signals → 3 TB.

5 Data traffic

From PLC(s) / data sources to DataCollector

- Data: raw PLC signals requested by the DataCollector.
- Required bandwidth is determined by the number of signals and the sampling rate.

Example calculation (estimation):

Number of signals	Sampling rate	KB/s
1,000	100 ms	~13.0
10,000	100 ms	~129.0

Number of signals	Sampling rate	KB/s
100,000	100 ms	~1288.0

Assumption: 70% boolean (1 bit), 30% real data (32 bit) types, 78 bit package overhead.

From DataCollector to central services

- Data: raw PLC signals and calculated features (constructed from signals).
- Only changed data values are sent (~50% less data).
- Required bandwidth is determined by the number of signals, the number of features and the sampling rate.

Example calculation (estimation):

Number of signals	Number of features	Sampling rate	KB/s
1,000	600	100 ms	8.2
10,000	4,000	100 ms	58.8
100,000	40,000	100 ms	588.6

Assumption: 70% boolean (1 bit), 30% real data (32 bit) types, 78 bit package overhead; signals change value on average every 5th sample.

6 Authentication

We provide a dedicated authentication service (OAuth 2-based). With it, you can register users and grant them role-based access to our services. Each user must log in via this authentication service to access the plus10 services. Single sign-on (SSO) with Microsoft 365 is also supported, as is connecting to an LDAP system. The detailed identity and access-management capabilities are described in chapter 9.

7 Client devices

The outbound URLs in this chapter apply to the SaaS offering. For on-premise deployments, clients connect to the customer-specific backend FQDN instead of plus10-webservices.com; the push-notification and app-store URLs remain the same.

7.1 Web browser

Supported browsers:

- Chrome (> v79)
- Firefox (> v70)
- Safari (> v10)
- Microsoft Edge (> v79)

Network port configuration

Service	Protocol	Outgoing port	Incoming port	Description	Optional
API, Front-end	HTTPS	443	–	–	No

Outbound internet URLs

Service	URL	Port	Description
plus10 Services	https://plus10-webservices.com (SaaS) or custom FQDN (On-Premise)	443	

7.2 Smartphone (only for Shannon®)

Supported devices / OS: iOS (minimum iOS 12) and Android (minimum Android 8).

7.2.1 iOS

Network port configuration

Service	Protocol	Outgoing port	Incoming port	Description	Optional
API, Front-end	HTTPS	443	–	–	No
Google Firebase Cloud Messaging	HTTPS/TCP	443, 5228-5230	–	Register for push notifications	Only for Shannon®
Apple Push Notification Service	HTTPS/TCP	443, 5223	–	Receive push notifications	Only for Shannon®

Outbound internet URLs

Service	URL	Port	Description
plus10 Services	https://plus10-webservices.com (SaaS) or custom FQDN (On-Premise)	443	
Google Firebase Cloud Messaging	Google IP range or FQDN whitelisting	443, 5228-5230	Register for push notifications (Shannon®)
Apple Push Notification Service	17.0.0.0/8	443, 5223	Receive push notifications (Shannon®)

Availability: the app is available via the Apple App Store.

7.2.2 Android

Supported devices: minimum Android 9.

Network port configuration

Service	Protocol	Outgoing port	Incoming port	Description	Optional
API, Front-end	HTTPS	443	–	–	No
Google Firebase Cloud Messaging	HTTPS/TCP	443, 5228-5230	–	Sending push notifications to clients	Only for Shannon®

Outbound internet URLs

Service	URL	Port	Description
plus10 Services	https://plus10-webservices.com (SaaS) or custom FQDN (On-Premise)	443	
Google Firebase Cloud Messaging	Google IP range or FQDN whitelisting	443, 5228-5230	Register for push

Service	URL	Port	Description
			notifications (Shannon®)

Availability: the app is available via the Google Play Store (Business). It is only visible after plus10 has added your company's Google Play Company ID to its configuration.

7.3 Apple Watch (only for Shannon®)

Supported devices: watchOS (minimum watchOS 7).

Network port configuration

Service	Protocol	Outgoing port	Incoming port	Description	Optional
API, Front-end	HTTPS	443	–	–	No
Google Firebase Cloud Messaging	HTTPS/TCP	443, 5228-5230	–	Register for push notifications	Only for Shannon®
Apple Push Notification Service	HTTPS/TCP	443, 5223	–	Receive push notifications	Only for Shannon®

Outbound internet URLs

Service	URL	Port	Description
plus10 Services	https://plus10-webservices.com (SaaS) or custom FQDN (On-Premise)	443	
Google Firebase Cloud Messaging	Google IP range or FQDN whitelisting	443, 5228-5230	Register for push notifications (Shannon®)
Apple Push Notification Service	17.0.0.0/8	443, 5223	Receive push notifications (Shannon®)

7.3.1 Installation, configuration and updates

To install the Shannon® app on an Apple Watch, the Shannon® iPhone app must be installed on a paired iPhone. Open the “Watch” app on the iPhone, select the watch and scroll to the list of available apps; tap “Install” next to the Shannon® app. The watch must be connected to the paired iPhone via Bluetooth (enabled on both devices). Installation can take several minutes and may occasionally fail — if so, tap “Install” again. If the app does not appear, restart the Apple Watch.

Configuration: connect the Apple Watch to a Wi-Fi network. To keep the Shannon® app active, in the “Watch” app tap “General” → “Wake Screen” and, under “Return to clock”, choose “After Crown Press” so the app stays in the foreground until the crown is pressed.

First steps: on first launch the app asks for the Server URL (plus10 Shannon® web service) and the plus10 authentication service URL. You receive these from plus10 support. Often both share the same domain/FQDN, in which case entering one URL and tapping “connect” lets the app guess the other. A green border indicates a successful connection; a red border indicates failure. After a successful connect the URLs are stored and can only be changed by reinstalling the app.

Updates: install the update on the iPhone first; the watch app updates automatically once connected to the paired iPhone via Bluetooth. To reset the stored service URLs, uninstall and reinstall the app.

8 Maintenance

8.1 Updates

We continuously improve our software by integrating new features and performing technical updates. The standard procedure for updating the deployed services (DataCollector and, for on-premise, the plus10 server services) is that plus10 support updates them via remote access. Arrangements regarding the update policy — including update authorization, update frequency and time windows — are defined per company.

8.2 Host (OS) updates

Updates of the OS and non-plus10 software on the hosts used are generally performed by the customer's IT. If this is not preferred or possible, a specific agreement for these updates can be defined.

8.3 Health checks (on-premise)

We are committed to deploying only tested and stable software. Nevertheless, several factors could cause IT errors. To respond quickly, we use health checks that continuously check the state of our software services. If an error occurs, a notification is sent to the plus10 support team (optional). As the standard solution, notifications are sent to a Microsoft Teams incoming webhook on HTTPS port 443; alternatively, notifications can be sent by email (SMTP). A notification contains the service name, the time of the event and the status of the service.

8.4 Remote access

To maintain our deployed software with minimum effort, remote access is generally provided by the customer. The following options are acceptable to us:

- Access via customer VPN (SSH/RDP).
- Access via TeamViewer.

8.5 Technical support of plus10

The plus10 technical support portal is available for questions regarding all of your regularly licensed products from Monday to Friday, 09:00–17:00 (Central European Time, UTC+1/+2), excluding public holidays in Augsburg, Germany. You can reach us at https://plus10.de/support/. If a specific service-level agreement is needed, please contact us.

9 Security and compliance

This chapter summarizes the security and data-protection measures of the plus10 platform that are relevant to a deployment. It covers identity and access management for end users, records management (retention, auditing, backups, deletion), incident management and the technical and organizational measures (TOMs) for data protection.

9.1 Identity and access management (end users)

9.1.1 Authorization

We use role-based access control (RBAC). Standard roles are defined with an increasing level of privileges, from a standard user role with limited privileges to an administrator role with all privileges. A privilege defines the right to use a certain functionality and can be detailed with

read, write and delete rights. For standard roles the privileges are fixed; you can also define custom roles with fine-grained privileges. RBAC lets you follow the principle of least privilege — it is up to you, the customer, to decide which user is assigned which role.

Tenant (customer) data is stored in a single-tenant database to ensure encapsulation. Our software needs configurations to work with customer data (data-source connections, machine topologies, machine alarms, etc.). This initial configuration is generally done by plus10 (and is part of the offer/contract); for this, plus10 support has a dedicated user to access the tenant environment, with the same authorization/authentication mechanism as other users. Software or database updates do not require access to tenant data.

9.1.2 Identity provisioning

Our service is invite-only — a user cannot self-register. Initially, an admin user is created for a new tenant and the credentials are shared with the customer; the customer then creates and shares user accounts. New users must change their password at first login. Users can also register via social-login providers (e.g. Microsoft 365), and a tenant admin can then invite those users into their environment. An administrator can disable a user (preventing login) and subsequently delete the user; when a user is deleted, all associations are removed and linked to a ghost user. A password reset is currently performed by an administrator from within the application; users cannot reset their own password.

9.1.3 Authentication

Users can set and change their own passwords; for the Operator role, password change can optionally be restricted to administrators. Instead of a composition policy, we enforce strong passwords using a password-strength algorithm. We enforce the following limits: maximum login failures 10; minimum time between login attempts 1 second; minimum lock-out of 1 minute if attempts are quicker than the minimum time between attempts. Based on recent research indicating that frequent password changes are likely to increase security risk, we do not enforce password changes at a defined interval (a dedicated instance with enforced password change is available on request). Multi-factor authentication is currently supported indirectly via social login (if the provider supports it); a native MFA option is actively being developed. We use OpenID Connect (an OAuth 2 extension) for authentication, and all communication is performed over HTTPS (TLS v1.3).

9.1.4 Identity and access-management integration

Individual IP whitelisting and federated/delegated authentication mechanisms (e.g. SAML, WS-Federation) are not available on the standard shared platform but can be provided on a dedicated instance of our software stack; federation is actively being developed. Social login is supported with Google, Microsoft and OpenShift as third-party identity providers; contact us to evaluate additional providers.

9.2 Records management

9.2.1 Retention policies

Calculated data (e.g. downtime data) is stored for two years and raw data (PLC signal and feature measurements) for three months; static definition data has no retention policy. Retention policies can be configured for special requirements.

9.2.2 Auditing

For data added or altered by a user, we generally store who created the record, who last updated it, and the corresponding timestamps. For frequently altered data (e.g. instructions, data-source configuration, events) we also store the history of changes (versioning).

9.2.3 Backups and disaster recovery

We regularly back up each tenant's database, keeping daily backups for the last five days and weekly backups for the last three weeks. We maintain a documented disaster-recovery process for restoring a backup, which we test regularly.

9.2.4 Deletion of data

Besides data affected by the retention policies, a two-step deletion process is in place: records are first archived, and archived records can then be deleted; only specific user roles can delete records. If a user is deleted, associated data is linked to a ghost user so that data integrity is preserved. There is currently no functionality to apply a legal hold to suspend deletion. If you stop using the plus10 SaaS and your license expires, we delete all your data and backups after 90 days (or earlier on explicit request).

9.3 Business continuity and incident management

An incident is an event that disrupts or reduces the quality of a service and requires an emergency response. A security incident is any attempted or actual unauthorized access, use, disclosure, modification or destruction of information within our organization, including specifically our cloud services holding customer data. An incident is resolved when the affected service resumes its intended functional state.

We operate a monitoring solution for our cloud infrastructure and applications; abnormal behavior (unhealthy application state, abnormally high load, logged exceptions) raises an alarm and notifies the support team. Tools also automatically analyze source code for bugs and vulnerabilities, and employees are trained to be cautious of and report abnormal IT-system behavior. When an incident is detected, an incident ticket is created and, based on on-call schedules and an escalation policy, assigned to a person for initial assessment; a severity is assigned, each step is documented, and customers are informed of steps taken where relevant. The impact severity of a security incident is evaluated as follows:

Category	Definition
None	No information was exfiltrated, changed, deleted or otherwise compromised.
Information breach	Sensitive or proprietary information of customers or employees was accessed or exfiltrated.
Integrity loss	Sensitive or proprietary information of customers or employees was changed or deleted.

If an incident impacts the functionality of our SaaS solution, or if a security incident with an impact severity occurs, we inform the affected customers.

9.4 Data privacy — technical and organizational measures (TOMs)

The following technical and organizational measures are implemented by plus10 to ensure the protection and privacy of personal data processed as part of our SaaS offering, in accordance with the General Data Protection Regulation (GDPR).

9.4.1 Access control

- Role-based access with least privilege.
- Strong password policies and 2FA/MFA.
- Logging and monitoring of access.

9.4.2 Data encryption

- Data encrypted in transit using TLS 1.2 or higher.
- Data encrypted at rest (AES-256).
- Secure key-management practices.

9.4.3 Availability and resilience

- Regular backups and tested disaster-recovery procedures.
- Uptime monitoring and incident response.

9.4.4 Data integrity, minimization and purpose limitation

- Audit logs for data changes.
- Only data necessary for the service is processed.
- Separate environments for development, testing and production.

9.4.5 Organizational measures

- Mandatory GDPR and security training for all staff (annually).
- Internal policies on confidentiality and data handling.
- Incident-response plan in place.
- Data Processing Agreements (DPAs) with all sub-processors, with due diligence and regular reviews.

9.4.6 Physical security

- Cloud infrastructure hosted by a certified provider (Microsoft Azure).
- Restricted physical access to data centers.

9.5 Support and service levels

When you request support via our website, email or phone, a support ticket is created in our management software and assigned to a person who performs an initial assessment, assigns a severity and adds information as necessary. Based on this, further steps are taken (contacting the customer to resolve the issue, assigning a technical support person, or scheduling a development ticket). After each action, we inform the customer who opened the ticket. If a specific service-level agreement is needed, please contact us.

10 Installation of central services (on-premise)

10.1 Prerequisites

The plus10 backend services (“central services”) are dockerized and can in general run in any Docker environment. Currently we support full out-of-the-box functionality using Docker Compose on a full Linux OS (e.g. not Portainer), as bash scripts must be executed. Deployment on a Kubernetes cluster is also supported — please contact us if needed.

Before running the software, make sure Docker and Docker Compose are installed. Please use the official Docker installation guide and avoid non-official Docker packages. It is recommended to allow running Docker commands without sudo rights. By default, deployment and media data are stored under /opt/plus10-services/ and /opt/plus10-data/, and database data under /var/local/postgres/ (a dedicated mount for /var/local/postgres is recommended). Create a valid TLS certificate for the plus10 backend beforehand (see section 10.2).

10.2 Certificate

The plus10 backend needs a TLS certificate, which should be generated based on your company root certificate. Set the validity period to a maximum of 397 days. You can use the certificate-signing-request script provided at https://p10publicdownload.blob.core.windows.net/public/generate_csr.sh, setting the HOST_DOMAIN variable to the FQDN of the server. Place the private key and the public key under /opt/plus10-services/certificate/. To update the certificate, replace these files and run: docker compose up -d traefik --force-recreate.

10.3 Installation of plus10 services

On the Linux machine where the backend will be deployed, download or copy the deployment archive from https://p10publicdownload.blob.core.windows.net/public/plus10-services.tgz (e.g. with wget). Unpack it to /opt/ (tar -xzf plus10-services.tgz -C /opt/); another location is possible but /opt/ is recommended. Depending on your configuration you may need sudo rights.

Go to /opt/plus10-services and edit the env_template file: search for CHANGE_ME_BEFORE_RUNNING_SCRIPT values and set proper values. Add the TLS certificate files if not already done (see section 10.2). Log in to the plus10 Docker registry with the credentials provided by plus10 (docker login plus10.azurecr.io). Then run the initialization script (bash init.sh); it performs pre-checks, downloads the required Docker images, sets system passwords, initializes the authentication services and the database, and verifies that everything works.

After successful initialization, start all services with docker compose up -d. To confirm everything is running, use docker container ls and check that no container is restarting. It can take up to two minutes for all containers to fully start; after that, no container should show an up-time of only a few seconds (which would indicate a recent restart).