Systemspecifikation för Secure IoT Solution (PoC)

1. Arkitektur

Komponenter:

• Sensorenhet (Virtuell Enhet via Docker): En Docker-container som kör ett Python-skript för att simulera sensordata och skicka till backend.

• InfluxDB (Databas): Lagrar den inkommande sensordatan från sensorenheten.

• Grafana (Visualisering): Visualiserar sensordatan i realtid.

• Docker Compose (Infrastruktur): Hanterar installation och sammankoppling av alla komponenter i ett integrerat nätverk.

2. Kommunikationsflöde

Dataflöde:

• Sensorenhet ➔ InfluxDB: Sensorenheten genererar data och skickar via HTTP POST till InfluxDB API.

• InfluxDB ➔ Grafana: Grafana hämtar data från InfluxDB och visar i dashboards.

Protokoll:

• HTTP: För kommunikation mellan enheter och tjänster.

• Intern Docker Network: Säkerställer att kommunikationen sker inom ett slutet nätverk utan exponering mot internet.

3. Säkerhetsåtgärder

• Isolering med Docker: Varje komponent körs i sin egen Docker-container, vilket skyddar mot säkerhetsbrister genom att hålla systemet isolerat.

• Säker Kommunikation (HTTPS eller VPN): Möjlighet att implementera TLS-certifikat för att kryptera dataöverföringar. Detta förhindrar avlyssning och manipulation av data.

• Autentisering och Auktorisering: Grafana och InfluxDB är konfigurerade med användarnamn och lösenord för att säkerställa att endast auktoriserade användare kan komma åt systemen.

4. Cyber Resilience Act (CRA) Krav

Säkerhet-by-Design:

• Isolering och Containerisering: Systemet är byggt med isolering av varje komponent för att förhindra att sårbarheter sprids.

• Krypterad Kommunikation: Möjlighet att lägga till TLS för att säkra dataöverföringar.

Uppdaterbarhet:

• Docker Containers: Varje komponent kan enkelt uppdateras genom att bygga om containern. Detta säkerställer att systemet kan hållas uppdaterat med säkerhetsfixar utan att störa andra delar.

Sårbarhetshantering:

• Snabba Uppdateringar via Docker Compose: Om en sårbarhet upptäcks kan nya versioner av varje komponent snabbt distribueras genom att köra nya versioner av containrar.

• Sårbarhetsskanning: Docker har verktyg för att skanna och upptäcka sårbarheter i containrar, vilket säkerställer att du kan agera snabbt om problem upptäcks.