

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-
ICT

Masterthesis

IoT-gebaseerde monitoring van professionele audio-installaties

Dries Martin

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

Prof. dr. Maarten WJNANTS

PROMOTOR :

ing. Thomas NAGELS

Gezamenlijke opleiding UHasselt en KU Leuven



Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt



2022
2023

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-
ICT

Masterthesis

IoT-gebaseerde monitoring van professionele audio-installaties

Dries Martin

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

Prof. dr. Maarten WIJNANTS

PROMOTOR :

ing. Thomas NAGELS



KU LEUVEN

Woord vooraf

Deze masterthesis gaat over het IoT-gebaseerd monitoren van professionele audio-installaties. Het vindt plaats binnen de masterproef die in de master in de industriële wetenschappen met als afstudeerrichting elektronica-ICT voltooid moet worden. Dit is een gezamenlijke opleiding van UHasselt en KU Leuven. De masterproef werd uitgevoerd tijdens het academiejaar 2022-2023.

Ik heb gekozen voor dit onderwerp omdat ik een zekere interesse in de muziekindustrie heb. Daarnaast wilde ik graag mijn kennis over de onderlinge communicatie tussen audiovisuele apparaten vergroten. Dit onderwerp was de ideale opportuniteit om hierover bij te leren.

Deze masterproef heeft me geleerd hoe een groot project, dat over een langere termijn gespreid is, succesvol te voltooien. Dit is voornamelijk gelukt dankzij een goede planning met een aparte deadline per onderdeel. Natuurlijk heeft het me ook veel geleerd over de communicatie tussen audiovisuele apparaten, IoT en Odoo.

Ik wil mijn interne en externe promotoren, prof. dr. Maarten Wijnants en ing. Thomas Nagels, bedanken voor hun begeleiding bij de implementatie van de masterproef. Dit zorgde ervoor dat dit vlot verliep en ik de juiste keuzes maakte. Daarnaast wil ik ook Mathijs Willems bedanken om voortdurend klaar te staan om mijn vragen te beantwoorden. Ook wil ik Amptec bedanken voor het voorzien van het nodige materiaal en een werkplek. Tot slot wil ik mijn familie en vrienden bedanken voor hun steun tijdens de masterproef.

Ik wens u veel leesplezier toe.

Dries Martin

Inhoudsopgave

Woord vooraf	1
Lijst met figuren	5
Verklarende woordenlijst	7
Abstract	9
Abstract in English	11
1 Inleiding	13
1.1 Situering	13
1.2 Probleemstelling	13
1.3 Doelstellingen	14
1.4 Methode	15
1.5 Vooruitblik	16
2 Literatuurstudie	17
2.1 Inleiding	17
2.2 Industrie 3.0 naar 4.0 binnen de productie-industrie	17
2.3 Industrie 3.0 naar 4.0 binnen de AV-industrie	19
2.4 Toepassing op de onderzoeksvraag	20
2.5 Conclusie	21
3 POC	23
3.1 Inleiding	23
3.2 d&b R90 PLC	23
3.3 Mosquitto MQTT-broker	25
3.4 Python-server	26
3.5 InfluxDB-database	27
3.6 Grafana-dashboard	27
3.7 Odoohelpdesk	28
4 Resultaten en discussie	31
4.1 Resultaten	31
4.2 Discussie	31
5 Besluit en toekomstig werk	33

5.1 Besluit	33
5.2 Toekomstig werk	33

Literatuurlijst	37
------------------------	-----------

Lijst van figuren

1.1	De R90 PLC	14
2.1	De vergelijking tussen industrie 3.0 (links) en industrie 4.0 (rechts)	18
2.2	Een voorbeeld van een ideaal industrie 4.0-interface	19
3.1	Een schema van de gerealiseerde POC	23
3.2	Een schema van de werking van de d&b R90	24
3.3	Een deel van het JSON-datamodel	24
3.4	Een schema van de werking van de Python-server	26
3.5	Een screenshot van de structuur van de InfluxDB-database	27
3.6	Een screenshot van vier grafieken in het Grafana-dashboard	28
3.7	Een screenshot van vier tabellen in het Grafana-dashboard	28
3.8	Een screenshot van een Odoohelpdeskticket	29

Verklarende woordenlijst

API	Application Programming Interface
AV-industrie	AudioVisuele industrie
IoT	Internet of Things
IP	Internet Protocol
IPC	Industrial Personal Computer
JSON	JavaScript Object Notation
MQTT(S)	Message Queuing Telemetry Transport (Secured)
PLC	Programmable Logic Controller
POC	Proof Of Concept
QoS	Quality of Service
TCP	Transmission Control Protocol
TLS	Transport Layer Security
WAN	Wide Area Network

Abstract

Als distributeur en integrator van professionele audio-installaties biedt Amptec zijn klanten technische ondersteuning aan. Momenteel is de servicedienst pas op de hoogte van problemen bij een klant na contact en is het moeilijk om de oorzaak meteen te constateren met de beperkte informatie die door de klant doorgegeven wordt. Er is dus nood aan een systeem om configuratie- en gebruiksdata van versterkers via de d&b R90 PLC binnen maximaal één minuut, voor realtime monitoring, over een beveiligde verbinding naar deze servicedienst te versturen.

De literatuurstudie resulteerde in een POC waarbij een JSON-datamodel over een MQTTS-verbinding naar een Python-server bij Amptec wordt gestuurd. Deze server verwerkt dit datamodel, slaat de data op in een InfluxDB-database en genereert een ticket in Odoo bij een foutmelding. Een Grafana-dashboard zorgt voor de visualisatie van de data. Ten slotte werd deze POC getest door een fout te genereren op de aangesloten d&b 30D-versterker en vervolgens het Grafana-dashboard en aangemaakte Odoo-ticket te controleren.

Deze test resulteerde in een maximale verwerkingstijd van een fout in een versterker van 11 s. Daarnaast bevat het JSON-datamodel voor 15 verbonden versterkers, de limiet van de R90 PLC, maximaal ongeveer 15 MB. Dit is minder dan de limiet van 256 MB per transmissie bij MQTT. Ook is er bij de met TLS beveiligde Mosquitto MQTT-broker in de cloud geen herconfiguratie van de firewalls nodig. De gerealiseerde POC is dus een geschikte oplossing voor het probleem.

Abstract in English

As a distributor and integrator of professional audio installations, Amptec offers its customers technical support. Currently, the service department is only aware of any problems at a customer after contact and it is difficult to immediately find the cause with the limited information which is given by the customer. There is thus a need for a system to send configuration and usage data of amplifiers via the d&b R90 PLC within a maximum of one minute, for real-time monitoring, over a secure connection to this service department.

The literature study resulted in a POC in which a JSON data model is sent over a MQTTS connection to a Python server at Amptec. This server processes this data model, stores the data in an InfluxDB database and generates a ticket in Odoo in case of an error message. A Grafana dashboard provides the visualisation of the data. Finally, this POC was tested by generating an error on the connected d&b 30D amplifier and then checking the Grafana dashboard and created Odoo ticket.

This test resulted in a maximum processing time of an error in an amplifier of 11 s. In addition, the JSON data model for 15 connected amplifiers, the limit of the R90 PLC, contains a maximum of about 15 MB. This is less than the limit of 256 MB per transmission of MQTT. Also, the TLS-protected Mosquitto MQTT broker in the cloud does not require any reconfiguration of the firewalls. The realised POC is thus a suitable solution for the problem.

Hoofdstuk 1

Inleiding

1.1 Situering

Deze masterproef beoogt om een IoT-gebaseerd monitoringsysteem van professionele audio-installaties te realiseren in samenwerking met Amptec.

Amptec is een distributeur en integrator van professionele audio-installaties met een hoofdsite in Bilzen. Ze werken met merken zoals Avid, d&b audiotechniek, DiGiCo, DPA, TC Electronic en Yamaha. Verder bieden ze ook bekabeling, installatie en ondersteuning aan. Ook hebben ze een aparte MI-afdeling voor de verdeling van muziekinstrumenten en gerelateerde producten [1].

De bekabelings- en installatieafdeling van Amptec levert turnkey pro audio-, video-, studio- en live-installaties voor broadcast en theater en is inmiddels uitgegroeid tot één van de grootste in de Benelux. Daarnaast verdeelt Amptec met zijn fbprotx-merk optische kabels om in te spelen op de vraag naar steeds grotere bandbreedtes in netwerksystemen. Ook het polijsten en lassen van deze kabels wordt onder dit merk aangeboden [1].

Ook is Amptec koploper met zijn ntwrx-merk. Dit merk biedt oplossingen voor netwerksystemen, audio en video over IP, systeembeheer, ondersteuning op afstand, topologie en ontwikkeling en streaming. Het is binnen dit merk dat de masterproef plaatsvindt. Dit merk heeft samen met d&b de R90 ontwikkeld. Dit is een PLC die instaat voor de centrale aansturing van versterkers in grote installaties. Een foto hiervan wordt getoond in figuur 1.1. De bedoeling van de masterproef is om deze PLC te koppelen aan het centrale beheersysteem van de servicedienst. Zo kunnen de servicemedewerkers automatisch op de hoogte gebracht worden als er een probleem zich voordoet in één van de installaties. Dit is het eerste project van Amptec in de wereld van IoT en cloud [1].

1.2 Probleemstelling

De R90 staat in voor de centrale aansturing van versterkers in grote installaties. Indien er een fout optreedt in deze apparatuur, zal er een foutmelding naar de PLC gestuurd worden. Het probleem hierbij is dat de klant regelmatig de PLC moet bekijken om te controleren of alles nog naar behoren werkt. De servicedienst van Amptec is dus pas op de hoogte van eventuele problemen wanneer de klant de PLC controleert en hen vervolgens contacteert. Daarnaast is het voor deze servicedienst moeilijk om de oorzaak meteen te constateren met de beperkte informatie



Figuur 1.1: De R90 PLC
[2]

die door de klant doorgegeven wordt.

Amptec heeft al geëxperimenteerd met het automatisch laten versturen van een sms door de PLC wanneer er een fout optreedt in de apparatuur. Dit bleek een groot succes te zijn. De servicedienst wist hierdoor niet alleen onmiddellijk of er een fout is opgetreden, maar ook welke fout. Echter kan er slechts een beperkte hoeveelheid aan informatie in een sms verstuurd worden en zijn sms'en kwetsbaar voor onderschepping [3]. Daarnaast werd de sms slechts naar één persoon gestuurd en was er geen integratie met het centrale beheersysteem van de servicedienst. Hierdoor is dit niet de meest geschikte oplossing.

Er is dus nood aan een systeem dat onmiddellijk een foutmelding inclusief alle bijbehorende en voorgaande data kan doorsturen naar de servicedienst van Amptec. Dit moet uiteraard ook veilig gebeuren. De hoofdonderzoeksvraag die deze masterproef moet beantwoorden is dus: Wat is de meest geschikte manier om configuratie- en gebruiksdata via de R90 PLC binnen maximaal één minuut, voor realtime monitoring, over een beveiligde verbinding naar de servicedienst van Amptec te versturen? Realtime monitoring is voornamelijk belangrijk bij telefonische ondersteuning.

1.3 Doelstellingen

De hoofddoelstelling van deze masterproef is dat er een werkende POC wordt gerealiseerd die het mogelijk maakt om de installaties van de klanten centraal op te volgen. Deze hoofddoelstelling kan onderverdeeld worden in de volgende deeldoelstellingen.

De eerste deeldoelstelling is het onderzoeken van de meest geschikte manier om deze POC te implementeren. Deze implementatie moet onmiddellijk bij een foutmelding zoveel mogelijk data vanuit de PLC op een veilige manier kunnen versturen naar de servicedienst. De keuze voor een bepaald communicatieprotocol zal de benodigde tijd, maximale hoeveelheid data per transmissie en beveiliging bepalen. De voorkeur gaat dus uit naar een communicatieprotocol dat alle data, maximaal enkele megabytes, in maximaal één minuut en één transmissie kan versturen over een beveiligde verbinding zoals TLS.

De tweede deeldoelstelling is de implementatie van een systeem waarbij de PLC alle nodige data uit de verschillende toestellen kan halen en omzetten in één gemeenschappelijk datamodel. Deze

data omvatten informatie over de instellingen en de meest recent uitgevoerde taken om zo de aanleiding en oorzaak van een probleem te ontdekken. De implementatie die hiervoor gekozen wordt, moet degene zijn die het meest geschikt blijkt te zijn in het voorafgaande onderzoek. De keuze moet hierbij vooral uitgaan voor een lightweight dataformaat dat maximaal enkele megabytes in beslag neemt.

De derde deeldoelstelling is de implementatie van het meest geschikte protocol voor een veilige communicatie met de servers van Amptec. Hierbij is het vooral van belang dat er geen herconfiguratie van de firewalls nodig is. Daarnaast moet het communicatieprotocol ook beveiligd zijn door bijvoorbeeld TLS.

De vierde deeldoelstelling is de implementatie van de opslag en verwerking van de data die door Amptec worden ontvangen. Hiervoor moeten services gebruikt worden die het meest geschikt blijken te zijn volgens het voorafgaande onderzoek. Ook moeten er regels opgezet en onderhouden worden om ervoor te zorgen dat er een ticket aangemaakt wordt bij de servicedienst in het geval van een fout in één van de versterkers.

De vijfde en laatste deeldoelstelling is de uitwerking van de weergave van de informatie. De automatisch doorgestuurde informatie moet hierbij nuttig en begrijpbaar getoond worden aan de servicedienst voor een zo optimaal mogelijke dienstverlening.

1.4 Methode

Om de masterproef tot een goed einde te brengen, werd er in verschillende fasen gewerkt. Hieronder worden de doorlopen fasen besproken.

De eerste fase was het voeren van een literatuurstudie om zo de meest geschikte oplossing voor het probleem te vinden. Hiervoor zijn vooral wetenschappelijke artikels en technische fiches gebruikt. Ook is er informatie gevraagd over de al bestaande infrastructuur om ervoor te zorgen dat er geen herconfiguratie van de firewall bij de klanten nodig is.

De tweede fase bestond uit het implementeren van een systeem om alle nodige data uit de aan de PLC aangesloten apparaten te halen en om te zetten in één gemeenschappelijk JSON-datamodel. Hiervoor is de TwinCAT 3-software gebruikt om de bestaande code uit te breiden. Vervolgens is de werking van de implementatie uit deze fase getest aan de hand van de R90 PLC en de d&b 30D-versterker die met hetzelfde netwerk zijn verbonden.

De derde fase omvatte het implementeren van een protocol om een veilige communicatie tussen de klant en Amptec mogelijk te maken. Hierbij is er gebruik gemaakt van een Mosquitto MQTT-broker in een Docker-container. Deze container werd lokaal gehost voor de POC, maar kan ook in de cloud gehost worden. Daarnaast werd de MQTT-broker beveiligd met TLS aan de hand van een *self-signed* certificaat dat werd aangemaakt met OpenSSL. Vervolgens werd de werking van de MQTT-broker alleen en in combinatie met de implementatie uit de tweede fase getest aan de hand van de applicatie MQTT Explorer.

De vierde fase was het implementeren van de verwerking en opslag van de ontvangen data bij Amptec. Voor de verwerking werd er een Python-script ontwikkeld dat lokaal in een Docker-container werd gehost. Voor de opslag werd er een InfluxDB-database lokaal gehost in een Docker-container. De werking van de implementatie uit deze fase werd vervolgens getest in

combinatie met de implementatie uit de vorige fasen.

De vijfde fase bestond uit het implementeren van een overzichtelijke weergave van de data. Zo moet er geen analyse van de binnenkomende data meer gebeuren. Hiervoor werd er een Grafana-instantie lokaal gehost in een Docker-container. Hiermee kan er een dashboard gemaakt worden voor iedere R90 PLC. De Grafana API werd ook gebruikt om de automatische creatie van nieuwe dashboards te implementeren in het Python-script uit de vierde fase. Daarnaast werd de automatische creatie van een ticket in Odoo aan de hand van de Odoo API in dit Python-script geïmplementeerd binnen deze fase.

De zesde en laatste fase was het testen van de volledige POC. Hierbij werden alle apparaten en toepassingen uit de vorige fasen opnieuw gebruikt om zo de werking van de gehele implementatie te testen. Dit werd gedaan door een fout te laten genereren op de aangesloten d&b 30D-versterker en vervolgens het Grafana-dashboard en aangemaakte Odoo-ticket te controleren.

1.5 Vooruitblik

Hoofdstuk 2 omvat de literatuurstudie, waarin de meest geschikte oplossing voor het probleem wordt gezocht. In hoofdstuk 3 wordt de werking van de geïmplementeerde POC besproken. Hoofdstuk 4 omvat de resultaten en de bespreking hiervan. Tot slot omvat hoofdstuk 5 het besluit en toekomstig werk.

Hoofdstuk 2

Literatuurstudie

2.1 Inleiding

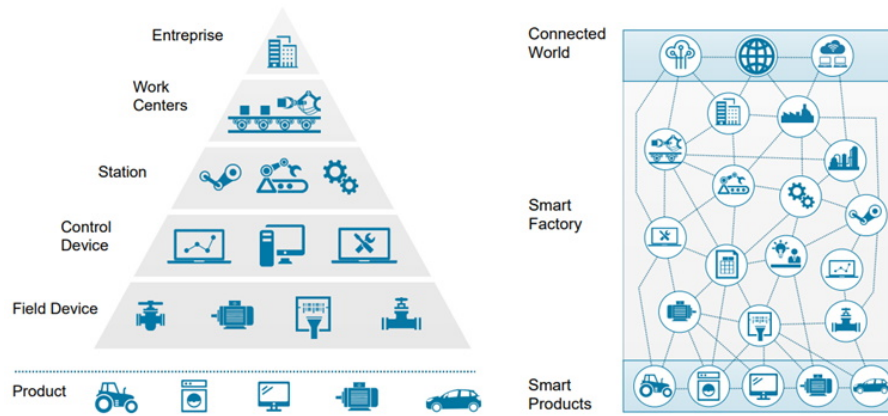
Deze literatuurstudie bestudeert eerst de evolutie van industrie 3.0 naar industrie 4.0 binnen de productie-industrie. Vervolgens wordt er onderzocht hoe deze evolutie ook van toepassing is op de AV-industrie. Daarna wordt er onderzocht hoe deze evolutie binnen de AV-industrie kan toegepast worden om de onderzoeksvraag van deze masterproef op te lossen. Ten slotte wordt er een conclusie getrokken uit deze literatuurstudie.

2.2 Industrie 3.0 naar 4.0 binnen de productie-industrie

De derde industriële revolutie heeft gezorgd voor een automatisatie van productieprocessen in vrijwel alle industrieën wereldwijd. Hierbij wordt een eindproduct gemaakt door een specifieke machine, die aangestuurd wordt door een specifiek controlesysteem. Dit controlesysteem is vervolgens deel van een bepaald station. Elk station heeft zijn eigen netwerk dat meestal gescheiden is van het internet en firewalls heeft om de controlesystemen van de machines te beschermen. Deze stations zijn vervolgens deel van een bepaalde fabriek die deel is van het bedrijf. Er is dus sprake van een communicatie die gebaseerd is op een hiërarchie. Elk eindproduct wordt geproduceerd door zijn eigen machines met specifieke functies die vastgelegd zijn in de hardware [4], [5].

Momenteel is er een nieuwe industriële revolutie aan de gang, namelijk de vierde industriële revolutie. Hierbij wordt de hele fabriek omgevormd tot een *smart factory* bestaande uit één groot netwerk in plaats van verschillende netwerken per station. Als gevolg kunnen de machines en systemen veel flexibeler werken door de functies van iedere machine te verdelen over dit netwerk. Dit zorgt er ook voor dat er communicatie is tussen alle machines en systemen, in plaats van enkel tussen de hiërarchische niveaus zoals bij industrie 3.0 het geval is. Aangezien de eindproducten een deel zijn van het hele netwerk, zijn de eindproducten ook slim. Dit netwerk kan vervolgens verbonden worden met de cloud om monitoring en bediening van eender welk apparaat in de fabriek overal ter wereld mogelijk te maken. Daarnaast wordt er bij industrie 4.0 op basis van *security by design* gewerkt om het hele productieproces te beveiligen. Dit betekent dat er bij de ontwikkeling van industrie 4.0-toepassingen beveiliging vanaf het begin wordt ingebouwd in plaats van achteraf pas toegevoegd aan de hand van andere producten [6]. De vergelijking tussen

industrie 3.0 en industrie 4.0 wordt voorgesteld in figuur 2.1 [4], [5], [7].



Figuur 2.1: De vergelijking tussen industrie 3.0 (links) en industrie 4.0 (rechts) [4]

Het is belangrijk dat het management de productie op de voet kan volgen om enerzijds nauwkeurige financiële voorspellingen te maken. Anderzijds is het ook belangrijk dat de productieprocessen nauwkeurig in de gaten gehouden kunnen worden door de fabrieksarbeiders om pannes van de productielijnen te vermijden. Er is dus nood aan een centraal platform om alle productieprocessen zo efficiënt mogelijk te monitoren. Dit wordt mogelijk gemaakt door de implementatie van industrie 4.0. De data die hierbij getoond worden, moeten afhankelijk zijn van de gebruikersrol. Voor de fabrieksarbeiders is het vooral van belang dat de problemen duidelijk getoond worden om zo zo snel mogelijk de oorzaken hiervan te bepalen en deze problemen op te lossen. Dit is een uitdaging door de steeds groeiende complexiteit van de data. Uit vorig onderzoek blijkt dat de data het beste op een hiërarchische manier getoond worden om zo enkel de benodigde informatie te tonen en verwarring te voorkomen. Daarnaast wordt er best de mogelijkheid gegeven om de interface aan te passen op basis van het eigen mentale model. Dit kan door middel van een *multi-window* interface. Dit interface kan uitgerust worden met *multi-touch* interactie om het snel en gemakkelijk aan te passen. Ook is het belangrijk om zelfverklarende diagrammen te gebruiken in plaats van ruwe data om zo de cognitieve belasting te verlagen bij het monitoren. Een voorbeeld van een interface dat aan de voorgaande voorwaarden voldoet, wordt getoond in figuur 2.2 [8], [9], [10].

De controlesystemen die in de industrie 3.0 gebruikt worden, zijn programmeerbaar door middel van de door de fabrikant meegeleverde software. Dit maakt ze veelzijdig en dus inzetbaar voor andere toepassingen dan het aansturen van machines in een fabriek. Een voorbeeld hiervan is het gebruik van een PLC voor *home automation*. Hierbij kunnen de verlichting, verwarming, poorten,... bediend worden aan de hand van een PLC die centraal in het huis geplaatst wordt. Een PLC kan ook gebruikt worden voor het creëren van slimme verkeerslichten. Op basis van de input van sensoren die het verkeer meten, kan de PLC de prioriteit van de verschillende verkeerslichten bepalen om zo de doorstroming te bevorderen. Net zoals bij een fabriek worden alle verbonden sensoren en actuatoren uit bovenstaande toepassingen gezien als deel van een netwerk dat enkel lokaal bediend kan worden [11], [12].

Ook deze toepassingen kunnen omgevormd worden tot industrie 4.0-toepassingen door de PLC te verbinden met de cloud en dus cloud monitoring mogelijk te maken. Bij home automation kan er bijvoorbeeld een portaal gemaakt worden dat bediend kan worden via WAN. Dit maakt



Figuur 2.2: Een voorbeeld van een ideaal industrie 4.0-interface [8]

de monitoring en bediening van alle apparaten vanaf eender welk apparaat over de hele wereld mogelijk [11].

2.3 Industrie 3.0 naar 4.0 binnen de AV-industrie

Apparaten in de AV-industrie waren eerst voornamelijk analoog en configureerbaar aan de hand van de knoppen die aanwezig waren op het apparaat zelf. Echter is er ook een digitalisatie ontstaan in deze industrie, waardoor de meeste apparaten een eigen microcontroller aan boord gekregen hebben om ze te configureren. In combinatie met een netwerkaansluiting, heeft dit controle en monitoring van op afstand mogelijk gemaakt. Aangezien een PLC gebruikt kan worden voor de controle en monitoring van meerdere toepassingen en het ook een netwerkaansluiting bevat, is dit apparaat dus ook geschikt voor de controle en monitoring van apparaten in de AV-industrie indien er ondersteuning is voor AES70 [13].

Door het gebruik van een PLC in de AV-industrie, kan er hierbij ook gesproken worden over industrie 3.0. Ook hier zijn alle verbonden apparaten, zoals bijvoorbeeld de versterkers, deel van één lokaal netwerk. Door de PLC te verbinden met de cloud, wordt de monitoring en controle van al deze apparaten op afstand mogelijk. Dan kan er gesproken worden over industrie 4.0 binnen de AV-industrie. Dit is voornamelijk handig voor de opvolging van de installatie van een klant door de installateur voor een betere dienstverlening [4], [13].

Analoge apparaten in de AV-industrie kunnen ook deel uitmaken van de industrie 4.0-aanpak door het toevoegen van een IoT-apparaat. Dit apparaat kan bijvoorbeeld een ESP32-microcontroller zijn. Eerder onderzoek heeft uitgewezen dat het volume van versterkers dynamisch kan aangepast worden door de creatie van een IoT-netwerk. Hierbij is elk IoT-apparaat verbonden met een microfoon om het geluidsniveau te meten. De waarde van dit geluidsniveau wordt via het MQTT-protocol gecommuniceerd over een intranet. Dit kan over wifi of door middel van een verbonden ethernetcontroller gebeuren. Vervolgens kunnen de andere IoT-apparaten het ontvangen geluidsniveau vergelijken met hun eigen gemeten geluidsniveau om zo het volume van de

verbonden versterker te verhogen of verlagen. Zo wordt er eenzelfde geluidsniveau bekomen over de hele oppervlakte [14], [15], [16].

Net zoals in een fabriek, is het bij monitoring van apparaten in de AV-industrie ook belangrijk dat de installateur een duidelijk dashboard ter beschikking heeft om de klanten efficiënt te kunnen helpen. Het dashboard heeft bij deze toepassing dus dezelfde vereisten als eerder besproken [8], [9], [10].

2.4 Toepassing op de onderzoeksvraag

De d&b R90 is eigenlijk de in de productie-industrie veelgebruikte Beckhoff CP6606 IPC. Hierop draait een TwinCAT 3-programma dat de functionaliteit bepaalt. TwinCAT 3 ondersteunt meer protocollen dan gangbaar in normale PLC's, waardoor deze IPC ook bruikbaar is als PLC voor doeleinden buiten de productie-industrie [2], [17].

De meeste apparaten in de AV-industrie, zoals versterkers, zijn uitgerust met een ethernet-aansluiting. Aangezien de Beckhoff CP6606 IPC ook over een ethernet-aansluiting beschikt, is het mogelijk om een lokaal netwerk op te zetten. Hierdoor kan er een TwinCAT 3-programma geschreven worden om de versterkers via de IPC te bedienen en te monitoren [17]. Het protocol dat hiervoor gebruikt wordt is AES70. Dit is een kosteloze open standaard die het mogelijk maakt om AV-apparatuur van eender welk merk te bedienen en te monitoren. Daarnaast is AES70 gebaseerd op het TCP/IP-protocol, waardoor het feilloos op standaard TCP/IP-netwerken kan functioneren [18]. Ook biedt TwinCAT 3 ondersteuning voor AES70 middels een betalende library, waardoor deze functionaliteit makkelijk geïmplementeerd kan worden in het programma [19]. Dit heeft de realisatie van de d&b R90 mogelijk gemaakt, waarbij er net zoals bij industrie 3.0 een lokaal netwerk ter bediening en monitoring van apparaten gecreëerd wordt.

Om dit systeem uit te breiden naar een systeem zoals bij industrie 4.0, moet de ontvangen configuratie- en gebruiksdata van de versterkers in een lightweight datastructuur gestoken worden. JSON is hiervoor een geschikte kandidaat omdat het lightweight is en TwinCAT 3 er ondersteuning voor biedt [20], [21].

Na het vormen van deze datastructuur, moet deze op een veilige en lightweight manier naar Amptec verzonden worden. Ook moet er rekening gehouden worden met de eis om de configuratie van de firewalls onveranderd te laten. Net zoals in [14] is MQTT een geschikte oplossing hiervoor. MQTT is lightweight, kan beveiligd worden met TLS en werkt aan de hand van een broker. Aangezien deze broker in de cloud gehost kan worden en MQTT een gangbaar protocol is, moeten de firewalls niet geconfigureerd worden. Hierdoor wordt dit protocol veel gebruikt voor IoT-toepassingen [22]. Ook biedt TwinCAT 3 ondersteuning voor MQTT, waardoor het makkelijk te implementeren is [23]. Natuurlijk moet het lokale netwerk ook eerst veilig met het internet verbonden worden om een beveiligde connectie met de MQTT-broker in de cloud mogelijk te maken.

Wanneer Amptec de verzonden data ontvangt, moeten deze data verwerkt worden en nadien opgeslagen worden in een database. Een geschikte database hiervoor is InfluxDB omdat deze de data opslaat in functie van de tijd [24]. Als gevolg kan Telegraf gebruikt worden voor de verwerking van de ontvangen data. Hoewel dit simpel te implementeren is, is het beter om een custom Python-script te ontwikkelen. Dit vergroot de uitbreidbaarheid zonder het vergroten van

de complexiteit wegens de beschikbaarheid van tal van libraries [25]. Voor de visualisatie van de data is de creatie van een dashboard in Grafana de beste oplossing. Dit is omdat het makkelijk is om een dashboard aan te maken en te integreren met InfluxDB [26]. Ten slotte kan het custom Python-script aangepast worden om automatisch een ticket aan te maken aan de hand van de Odo API [27]. Het gevolg is dat er een systeem gecreëerd wordt zoals bij industrie 4.0 waarbij monitoring van op afstand mogelijk is.

2.5 Conclusie

De evolutie van industrie 3.0 naar industrie 4.0 heeft niet enkel plaatsgevonden binnen de productie-industrie, maar binnen eigenlijk alle verschillende industrieën. Dit heeft de bediening en monitoring van apparaten binnen een bepaald netwerk van op afstand mogelijk gemaakt, terwijl men bij industrie 3.0 altijd fysiek de PLC moest controleren. Door de vele data die bij industrie 4.0 ter beschikking zijn, is er ook een nood aan een overzichtelijk dashboard om de cognitieve belasting bij het monitoren te verlagen. Dit is ook belangrijk om zo snel mogelijk de juiste beslissing te kunnen nemen.

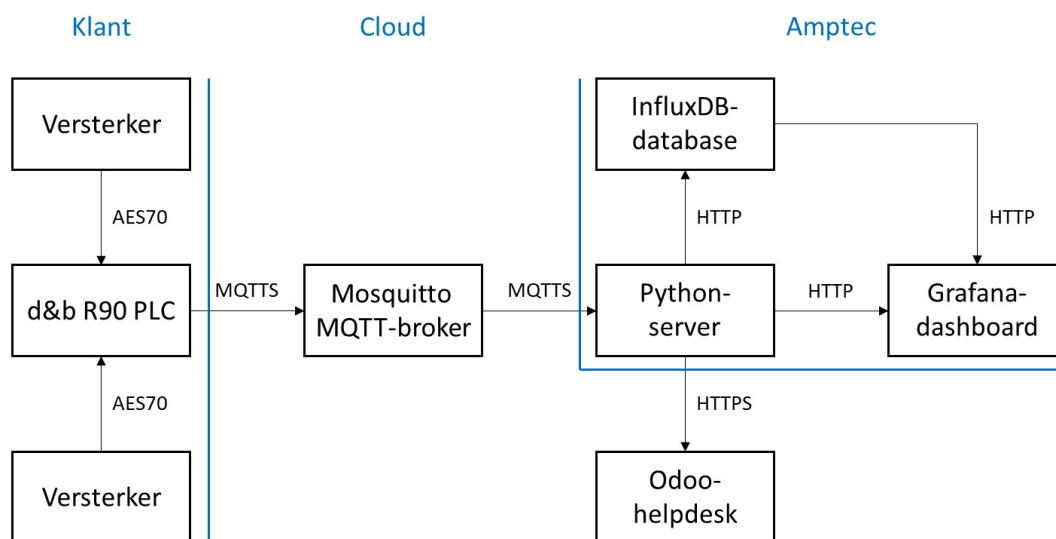
Een PLC kan gebruikt worden om door middel van AES70 te communiceren met versterkers. Op deze manier kan er een lokaal netwerk gecreëerd worden zoals bij industrie 3.0. Dit netwerk kan vervolgens verbonden worden met het internet om monitoring van op afstand mogelijk te maken. Dit kan door de ontvangen data van de versterkers in een JSON-datastructuur te steken en te verzenden via een met TLS beveiligde MQTT-broker. Vervolgens kan een custom Python-script de ontvangen data verwerken, opslaan in een InfluxDB-database en automatisch een ticket laten aanmaken in Odo in het geval van een foutmelding in de ontvangen data. De ontvangen data kunnen ten slotte visueel voorgesteld worden in een Grafana-dashboard. Op deze manier wordt monitoring van de R90 van op afstand mogelijk gemaakt.

Hoofdstuk 3

POC

3.1 Inleiding

Na het voltooien van de literatuurstudie is er een duidelijk beeld gevormd van de te implementeren POC. Deze POC kon vervolgens geïmplementeerd worden. Een schema van de gerealiseerde POC wordt getoond in figuur 3.1. De volgende secties in dit hoofdstuk gaan dieper in op de werking van elk onderdeel van de gerealiseerde POC.

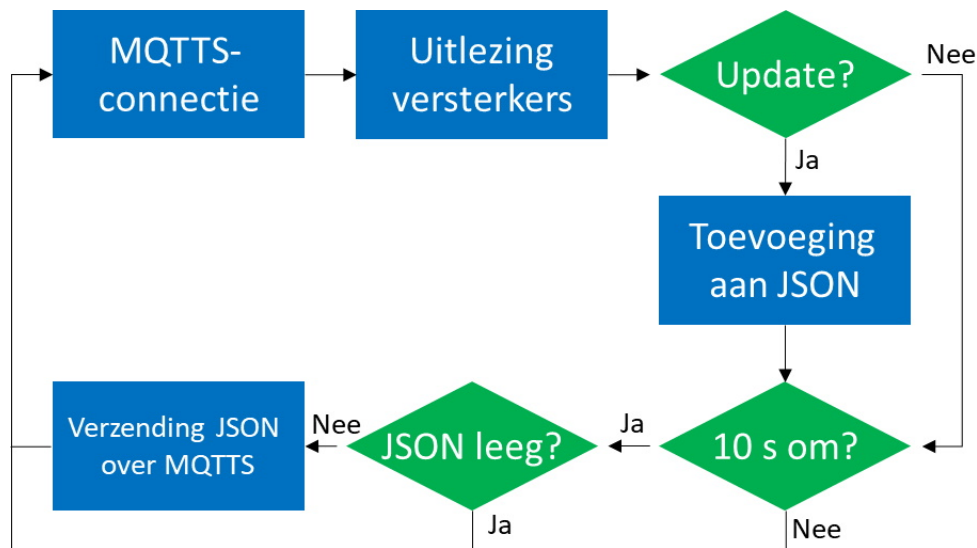


Figuur 3.1: Een schema van de gerealiseerde POC

3.2 d&b R90 PLC

De d&b R90 PLC is geprogrammeerd aan de hand van de TwinCAT 3-software van Beckhoff. De bestaande broncode laat toe om tot 15 versterkers te monitoren en bedienen door middel van het AES70-protocol. Deze broncode werd aangepast om configuratie- en gebruiksdata van de versterkers over MQTTS te verzenden. De werking van de d&b R90 wordt getoond in figuur 3.2.

De d&b R90 werkt aan de hand van een loop die elke 10 ms wordt herhaald. De code start met het verbinden met de MQTT-broker. Bij een succesvolle connectie worden de huidige data van de versterkers uitgelezen. Deze data worden vervolgens vergeleken met de opgehaalde data



Figuur 3.2: Een schema van de werking van de d&b R90

tijdens de vorige iteratie. Indien er een verandering is, worden de huidige data toegevoegd aan een JSON-datamodel. Zo worden enkel de belangrijke data doorgestuurd. Een deel van dit JSON-datamodel wordt getoond in figuur 3.3.

```

{
  "Data": [
    {
      "Amplifier id": "1",
      "Timestamp": "DT#2023-03-01-15:45:21",
      "Active": "TRUE",
      "Device name": "Amp30D"
    }
  ],
  "Serial number": "00001340"
}

```

Figuur 3.3: Een deel van het JSON-datamodel

Het JSON-datamodel bestaat enerzijds uit een array die de data van alle versterkers omvat en anderzijds het serienummer van de d&b R90 PLC. De data-array bevat een item voor elke keer dat er een verandering is gedetecteerd voor een bepaalde versterker. Dit item bevat gegevens zoals het nummer van de versterker, het tijdstip van de gedetecteerde verandering, de activiteitsstatus van de versterker en de naam van de versterker. Het volledige JSON-datamodel bevat meer gegevens per item dan getoond wordt. Er kan ook opgemerkt worden dat er een string gebruikt is voor alle gegevens. De reden hiervoor is dat het gebruik van andere datatypes in een leeg JSON-datamodel resulteerde bij het gebruik van de TwinCAT 3 JSON library.

De code bevat ook een timer van 10 seconden. Indien de timer de 10 seconden nog niet bereikt heeft, zal de code verder gaan met een nieuwe iteratie om mogelijk extra informatie toe te voegen. In het geval dat de 10 seconden wel voorbij zijn, zal het JSON-datamodel gecontroleerd worden op data. Wanneer dit model data bevat, wordt het JSON-datamodel naar de MQTT-broker gestuurd om daarna terug te starten met een nieuwe iteratie en een geresette timer. Ook als dit model geen data bevat, wordt er gestart met een nieuwe iteratie en een geresette timer.

3.3 Mosquitto MQTT-broker

Voor de communicatie tussen de klant en Amptec is er gebruik gemaakt van een MQTTS-verbinding. MQTT is een lightweight communicatieprotocol ontworpen voor IoT-toepassingen. Het werkt aan de hand van een broker of server waarmee users kunnen verbinden. Deze broker bestaat uit verschillende afzonderlijke topics waarop deze users zich kunnen abonneren. Indien een user iets publiceert op een bepaald topic, dan zullen alle geabonneerde users de gepubliceerde data ontvangen. MQTT is bidirectioneel waardoor alle users zowel zich kunnen abonneren als data publiceren op een bepaald topic.

Als MQTT-broker is er gebruik gemaakt van een Mosquitto MQTT-broker die gehost wordt in een Docker-container. Aangezien de POC volledig geïmplementeerd werd binnen één lokaal netwerk, werd deze broker niet in de cloud gehost. Voor een productieomgeving waarbij de d&b R90 PLC en de Python-server op een verschillend netwerk zitten, is dit wel een vereiste. Er is ook de mogelijkheid om te kiezen voor een andere MQTT-broker, zoals bijvoorbeeld HiveMQ of EMQX, die in de cloud gehost wordt. De Mosquitto MQTT-broker is beveiligd met TLS door middel van een self-signed certificaat. Dit certificaat is gegenereerd aan de hand van OpenSSL.

Voor de communicatie over de MQTTS-verbinding moest er ook een QoS-niveau gekozen worden. Dit bepaalt de garantie dat een MQTT-bericht aankomt bij de ontvanger. Er kan gekozen worden uit drie niveaus: 0, 1 of 2. Bij een QoS-niveau van 0 wordt het MQTT-bericht maximaal één keer door de verzender verzonden en wordt er niet gecontroleerd of de ontvanger dit bericht heeft ontvangen. Bij een QoS-niveau van 1 wordt het MQTT-bericht door de verzender verzonden en wordt er vervolgens op een MQTT-bericht van de ontvanger gewacht dat de aankomst van het bericht bevestigt. Indien er geen bevestiging is ontvangen binnen een bepaalde tijdsperiode, dan wordt het bericht opnieuw verzonden door de verzender. Dit kan ervoor zorgen dat de ontvanger het bericht meerdere keren ontvangt wanneer de verzender bijvoorbeeld de bevestiging ontvangt vlak nadat het bericht voor een tweede keer is verzonden. Bij een QoS-niveau van 2 gebeurt hetzelfde als bij een QoS-niveau van 1, maar om te voorkomen dat de ontvanger een bericht tweemaal kan ontvangen, wordt er hierna een releasebericht door de verzender verzonden. Dit releasebericht zorgt ervoor dat de ontvanger het eerdere ontvangen bericht slechts éénmaal doorgeeft. Hierop reageert de ontvanger met een bericht ter bevestiging van de ontvangst naar de verzender.

Er is gekozen voor een QoS-niveau van 0 omdat de communicatie over MQTTS bij de POC volledig over een stabiele en bedrade connectie gebeurt. Een hoger QoS-niveau leidt tot een zekerheid van aankomst van een transmissie bij onstabiele netwerken, maar zorgt hier alleen voor een extra overhead per transmissie. Daarnaast zorgt een hoger QoS-niveau voor een toename van de maximale verwerkingstijd wegens de toename van het aantal requests en responses die nodig zijn voor een succesvolle transmissie van een bericht over de MQTTS-verbinding.

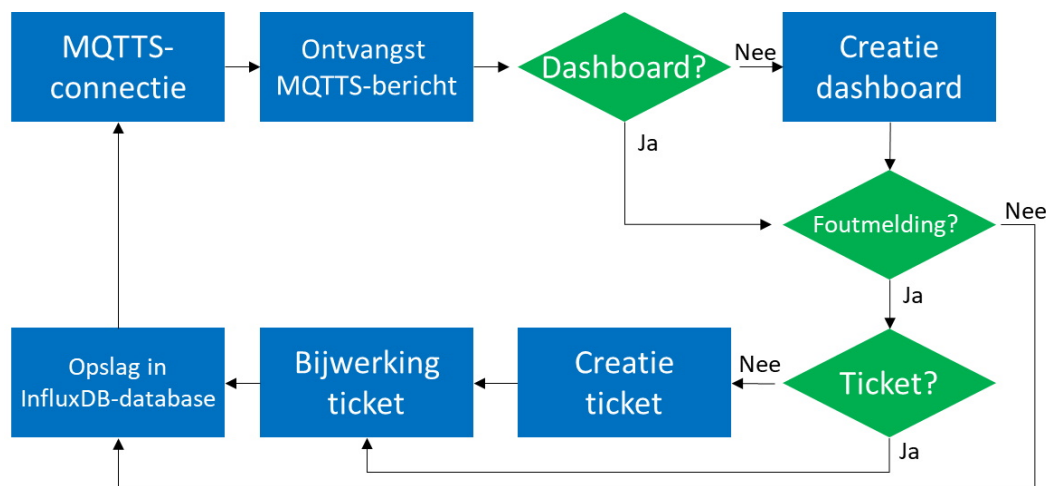
Verder is er gebruik gemaakt van slechts één user waarop zowel de d&b R90 PLC als de Python-server ingelogd zijn. Echter kan er voor een productieomgeving gebruik gemaakt worden van een user per d&b R90 PLC en een aparte user voor de Python-server. Dit verhoogt de beveiliging voor in het geval dat de inloggegevens van een user gelekt worden.

Daarnaast is er voor de POC gebruik gemaakt van slechts één topic. Een andere mogelijkheid is om een topic aan te maken voor elk onderdeel van het JSON-datamodel dat wordt doorgestuurd.

Om de users en topics van de Mosquitto MQTT-broker te beheren, kan er gebruik gemaakt worden van het Mosquitto Management Center. Dit is een dashboard dat verbonden kan worden met de Mosquitto MQTT-broker om zo makkelijk configuratiewijzigingen door te voeren. Vooral voor een productieomgeving met vele users is dit een handige tool.

3.4 Python-server

De verwerking van de door Amptec ontvangen data gebeurt aan de hand van een custom Python-script dat lokaal gehost wordt in een Docker-container. De werking van dit Python-script wordt getoond in figuur 3.4.



Figuur 3.4: Een schema van de werking van de Python-server

Ook de Python-server werkt aan de hand van een loop die voortdurend herhaald wordt. De loop start met het verbinden met de MQTT-broker. In het geval dat er geen verbinding tot stand gebracht kan worden, zal de code niet verder gaan tot er een verbinding tot stand is gebracht. Zodra de Python-server verbonden is met de MQTT-broker, zal er gewacht worden tot er een bericht via de MQTTS-verbinding wordt ontvangen.

Wanneer er een bericht wordt ontvangen via de MQTTS-verbinding, wordt er gecontroleerd of er al een Grafana-dashboard bestaat voor de d&b R90 PLC waar de ontvangen data vandaan komen. Indien er al een dashboard voor deze PLC bestaat, wordt de url hiervan opgehaald. Indien er nog geen dashboard voor deze PLC bestaat, wordt er een nieuw dashboard voor deze PLC aangemaakt aan de hand van de Grafana API. Hierna wordt ook de url van dit nieuwe dashboard opgehaald.

Vervolgens wordt er gecontroleerd of de ontvangen data een foutmelding bevatten. Indien dit niet zo is, worden de ontvangen data al meteen opgeslagen in de InfluxDB-database aan de hand van de InfluxDB Python client library. Indien dit wel zo is, wordt er via de Odoo API gecontroleerd of er al een actief ticket is in de Odoo-helpdesk voor deze PLC. Als er geen ticket voor deze PLC actief is, wordt er een nieuw ticket aangemaakt dat de url van het dashboard en een prioriteit op basis van het type fout bevat. Daarna wordt dit nieuwe ticket bijgewerkt met details over de fout en het nummer van de versterker in een logitem.

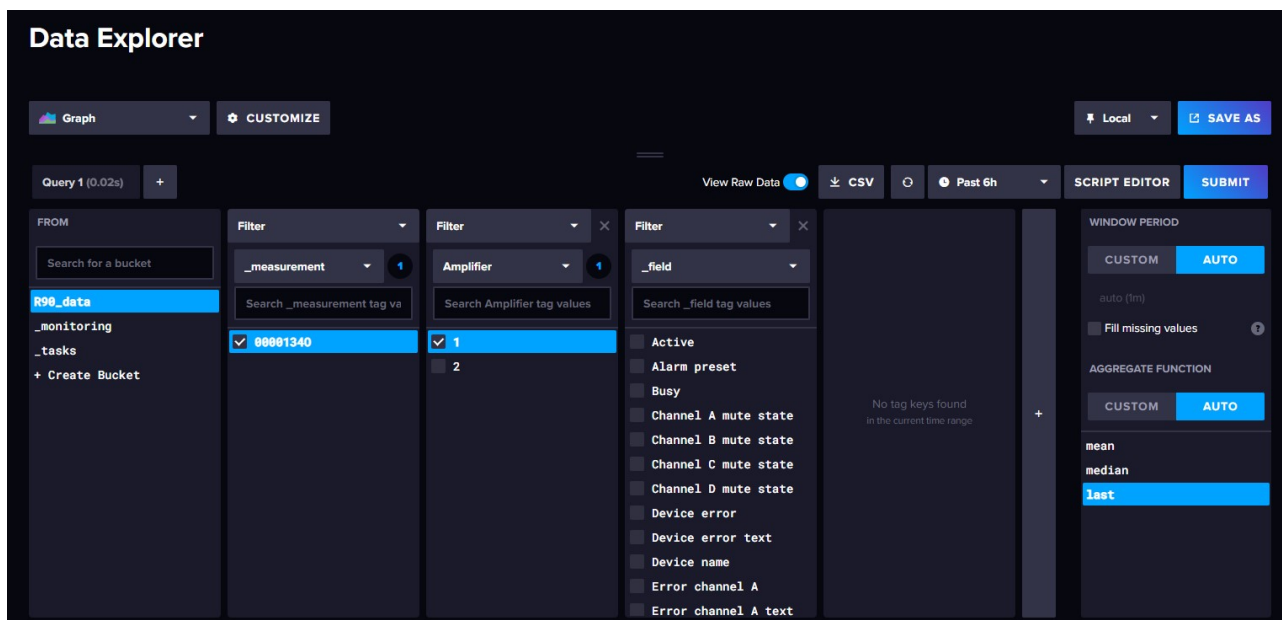
Indien er al een ticket actief is voor deze PLC, wordt het actieve ticket hiermee bijgewerkt. Aangezien er hierbij gewerkt wordt met de logs van een ticket, treden er geen schaalbaarheids-

problemen op bij een voortdurende herhaling van dezelfde fout. Ieder logitem bestaat namelijk op zichzelf en wordt gelinkt aan het bijhorende ticket. Daarnaast worden de logs van een ticket chronologisch en samen met het bijhorende tijdstip getoond, waardoor het overzicht behouden blijft. In het geval van een al bestaand ticket, wordt er ook gecontroleerd welke prioriteit dit ticket heeft. Wanneer deze prioriteit lager is dan de prioriteit van de nieuwe fout, dan wordt de prioriteit van het bestaande ticket verhoogd.

Hieropvolgend worden de ontvangen data ook in de InfluxDB-database opgeslagen aan de hand van de InfluxDB Python client library. Vervolgens start de loop opnieuw vanaf het begin met het controleren van de MQTTS-connectie.

3.5 InfluxDB-database

Voor de opslag van de ontvangen data is er gebruik gemaakt van een InfluxDB-database. Dit is een tijdreeks database en dus geschikt voor de opslag van loggegevens in functie van de tijd. Deze InfluxDB-database wordt ook lokaal gehost in een Docker-container. Figuur 3.5 toont een screenshot van de structuur van de InfluxDB-database.



Figuur 3.5: Een screenshot van de structuur van de InfluxDB-database

Er werd gekozen voor één bucket waarin alle ontvangen data opgeslagen worden. Als measurement werd er gekozen voor het serienummer van de d&b R90 PLC. Daarnaast zijn de volgnummers van de aangesloten versterkers toegevoegd als tags. Verder zijn de data van iedere versterker toegevoegd als fields.

3.6 Grafana-dashboard

Voor de visualisatie van de ontvangen data is er gebruik gemaakt van een Grafana-dashboard dat ook lokaal in een Docker-container gehost wordt.

Dit dashboard bevat een onderdeel voor elk data-item dat wordt doorgestuurd. Indien dit data-item een nummer of boolean is, wordt de waarde van dit data-item getoond in een grafiek in

functie van de tijd. Een screenshot hiervan wordt getoond in figuur 3.6. In het geval dat dit data-item tekst is zoals bijvoorbeeld bij een foutmelding, dan wordt de waarde van dit data-item getoond in een tabel samen met het bijhorende tijdstip. Een screenshot hiervan wordt getoond in figuur 3.7.



Figuur 3.6: Een screenshot van vier grafieken in het Grafana-dashboard

The figure shows four tables in a 2x2 grid, each representing an error log for a different channel. Each table has a 'Time' column and a 'Text' column. The data is as follows:

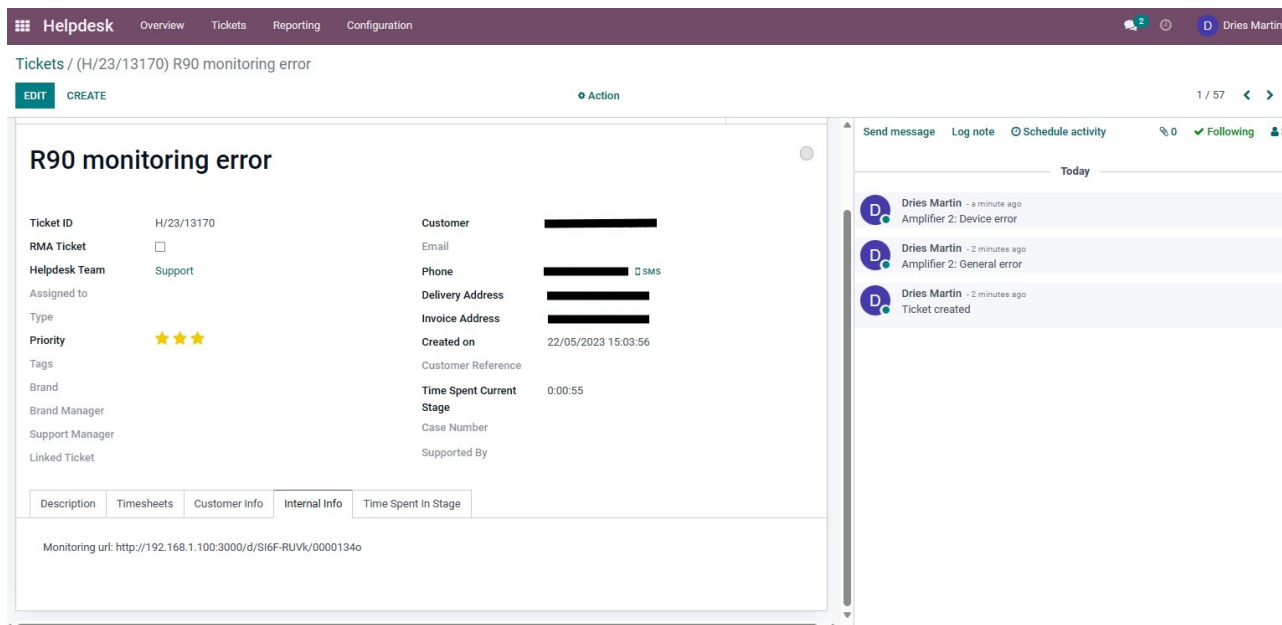
Channel	Time	Text
Error channel A text	2023-05-08 14:30:00	Error channel A text 1
	2023-05-08 14:45:00	
	2023-05-08 14:50:00	
	2023-05-08 15:15:00	Speaker imp. fault
	2023-05-08 16:10:00	
Error channel B text	2023-05-08 14:30:00	Error channel B text 1
	2023-05-08 14:45:00	
	2023-05-08 14:50:00	
	2023-05-08 15:15:00	
	2023-05-08 16:10:00	
Error channel C text	2023-05-08 15:00:00	Error channel C text 2
Error channel D text	2023-05-08 15:00:00	Error channel D text 2

Figuur 3.7: Een screenshot van vier tabellen in het Grafana-dashboard

Wanneer er data van meerdere versterkers, die allen verbonden zijn met dezelfde PLC, zijn ontvangen, worden er meerdere lijnen in iedere grafiek getoond. Hierbij is elke lijn een bepaalde versterker. Verder kan er een selectie gemaakt worden om enkel de grafieken van bepaalde versterkers te tonen. Ook worden er in dit geval meerdere tabellen getoond, met een tabel voor elke versterker.

3.7 Odoo-helpdesk

Voor het bijhouden van tickets maakt Amptec gebruik van de Odoo-helpdesk die in de cloud gehost wordt. Deze software bevat een API die toelaat om tickets aan te maken vanuit een andere software. Een screenshot van een ticket dat gegenereerd werd aan de hand van deze API, wordt getoond in figuur 3.8.



Figuur 3.8: Een screenshot van een Odoo-helpdeskticket

Wanneer een Odoo-ticket wordt aangemaakt door de Python-server bij een fout, dan is de naam van het ticket altijd “R90 monitoring error”. Daarnaast wordt de url van het Grafana-dashboard toegevoegd onder internal info om zo makkelijk de details van het ticket te kunnen bekijken. Er is gekozen om het veld internal info te gebruiken omdat er geen specifiek veld voor een url beschikbaar was. Dit kan wel toegevoegd worden in de uiteindelijke implementatie. Verder wordt de prioriteit van het ticket ingesteld op basis van het type fout. Ook de details van de fout en bij welke versterker deze fout heeft plaatsgevonden worden toegevoegd aan de logs van het ticket.

Odoo staat ook toe om een inventaris van producten bij te houden. Hierin kunnen de serienummers van de verkochte d&b R90 PLC's toegevoegd worden. Deze serienummers kunnen daarnaast binnen Odoo ook verbonden worden met een bepaalde klant. Op deze manier is het mogelijk om meteen de betrokken klant toe te voegen aan een automatisch gegenereerd ticket. Indien de serienummer en/of de connectie met de klant in Odoo ontbreken, wordt er geen klant toegevoegd aan een automatisch gegenereerd ticket.

Bij een foutmelding op een PLC wordt er altijd eerst gecontroleerd of deze PLC reeds een actief ticket in Odoo bezit. Dit wordt gecontroleerd op basis van het veld internal info bij ieder actief ticket. Bij een al bestaand actief ticket voor een bepaalde PLC, wordt het actieve ticket voor deze PLC bijgewerkt. Daarnaast wordt de prioriteit verhoogd indien de fout een hogere prioriteit heeft. Ook worden de details van de fout samen met het nummer van de versterker toegevoegd aan de logs van het actieve ticket.

Hoofdstuk 4

Resultaten en discussie

4.1 Resultaten

De JSON-datastructuur, die door de d&b R90 PLC wordt doorgestuurd, bestaat voor één versterker uit ongeveer één kilobyte. Per versterker die wordt toegevoegd, wordt er ongeveer één kilobyte aan deze datastructuur toegevoegd. Voor 15 versterkers bevat deze datastructuur dus ongeveer 15 kilobytes. Zoals besproken in sectie 3.2, werkt de d&b R90 PLC aan de hand van een loop die elke 10 ms herhaald wordt. Per iteratie van deze loop kunnen er dus ongeveer 15 kilobytes toegevoegd worden aan de JSON-datastructuur. Aangezien er ook een timer van 10 seconden aanwezig is voordat deze datastructuur verzonden wordt, treden er dus 1000 iteraties op voordat de datastructuur verzonden wordt. In het geval dat er iedere iteratie ongeveer 15 kilobytes worden toegevoegd aan de datastructuur, resulteert dit in een maximale grootte van ongeveer 15 megabytes. Dit is ook de maximale hoeveelheid data die per transmissie over de MQTTS-connectie wordt gestuurd, vermits de overhead van deze connectie beperkt is tot tientallen bytes voor een QoS-niveau van 0.

De benodigde tijd om de volledige POC te doorlopen, werd bekomen door het verschil te berekenen tussen het tijdstip van de verzending van de JSON-datastructuur door de d&b R90 PLC en het tijdstip van de creatie van een ticket in Odoo. Dit werd drie keer herhaald en resulteerde iedere keer in een tijd van één seconde voor een QoS-niveau van 0. Indien een fout zich voordoet bij de start van de timer van 10 seconden, zoals beschreven in sectie 3.2, duurt het dus 11 seconden voordat er een ticket wordt aangemaakt in Odoo. Dit is ook de maximale verwerkingstijd van een fout voor een QoS-niveau van 0.

4.2 Discussie

De Mosquitto MQTT-broker ondersteunt maximaal 256 megabytes per transmissie. Aangezien de JSON-datastructuur voor 15 versterkers tot maximaal ongeveer 15 megabytes per transmissie kan bevatten, voldoet deze MQTT-broker voor deze toepassing. Er is dus ook nog genoeg ruimte om het TwinCAT 3-programma van de d&b R90 PLC aan te passen om zo meer versterkers en meer informatie per versterker per transmissie te ondersteunen.

Aangezien de maximale verwerkingstijd van een fout 11 seconden is, voldoet de POC ook aan de eis van een maximale verwerkingstijd van één minuut. Deze verwerkingstijd neemt toe indien er

meer data per transmissie worden verzonden. Indien de Python-server zwaar belast wordt door meerdere d&b R90 PLC's die tegelijkertijd een grote hoeveelheid data doorsturen, neemt deze verwerkingstijd ook toe. Echter kunnen er meerdere instanties van de Python-server parallel opgezet worden, die elk bijvoorbeeld de data van maximaal vijf d&b R90 PLC's verwerken. Dit kan gedaan worden aan de hand van een apart MQTT-topic per instantie van de Python-server. Zo kan er nog altijd voldaan worden aan de maximale verwerkingstijd van één minuut voor een groot aantal d&b R90 PLC's. Daarnaast zorgt het verlengen van de timer van 10 seconden ook voor een toename van de maximale verwerkingstijd, maar ook voor minder overhead bij de transmissie van de data.

De POC voldoet ook aan de eis van een beveiligde verbinding. De MQTT-broker is namelijk beveiligd met TLS door middel van een self-signed certificaat.

De d&b R90 PLC kan ook vervangen worden door een andere PLC, zolang de JSON-datastructuur die via de MQTT-broker wordt doorgestuurd dezelfde blijft. Daarnaast kan de MQTT-broker veranderd worden zonder aanpassing van de rest van de POC, mits de logingegevens en topics identiek blijven. Verder kunnen ook de InfluxDB-database, het Grafana-dashboard en het Odooticketingsysteem vervangen worden. Dit heeft enkel invloed op de Python-server. De vervangende software dient wel te beschikken over een API, zodat deze geïmplementeerd kan worden in de code van de Python-server. De vervangbaarheid van de onderdelen bewijst de aanpasbaarheid, uitbreidbaarheid en modulaire inzetbaarheid van de gerealiseerde POC.

Hoofdstuk 5

Besluit en toekomstig werk

5.1 Besluit

Uit deze masterproef blijkt dat het mogelijk is om een POC te realiseren om configuratie- en gebruiksdata van versterkers via de d&b R90 PLC naar de servicedienst van Amptec te sturen. Hierbij is er voor de datacommunicatie gebruik gemaakt van een MQTT-broker die beveiligd is met TLS, waardoor er aan de beveiligingseisen voldaan wordt. Deze MQTT-broker moet in de cloud gehost worden om een herconfiguratie van de firewalls te vermijden. Daarnaast zorgt de Python-server voor de verwerking van de ontvangen data, de InfluxDB-database voor de opslag en het Grafana-dashboard voor de visualisatie. Een foutmelding in de ontvangen data zorgt automatisch voor de creatie van een ticket in Odoo samen met details van de fout en een link naar het Grafana-dashboard.

Voor 15 verbonden versterkers worden er maximaal ongeveer 15 megabytes per transmissie verstuurd. De limiet van de Mosquitto MQTT-broker van 256 megabytes per transmissie is dus voldoende. Daarnaast bedraagt de maximale verwerkingstijd 11 seconden, waardoor er voldaan wordt aan de eis van een maximale verwerkingstijd van één minuut. De geïmplementeerde POC voldoet dus aan alle vooropgestelde eisen.

5.2 Toekomstig werk

De POC kan verder uitgebreid worden met meer informatie over de versterkers. Zo kunnen bijvoorbeeld de voltages van iedere versterker mee doorgestuurd worden. Deze extra informatie kan het zoeken naar een oorzaak vergemakkelijken.

De aanwezige modulariteit van de geïmplementeerde POC kan ook gebruikt worden om de POC toe te passen op andere PLC's. Dit maakt het mogelijk om de POC te gebruiken voor monitoringstoepassingen buiten de AV-industrie.

Literatuurlijst

- [1] Ampotec, “Over ons.” [Online]. Available: <https://ampotec.be/nl/about>. [Geopend op 11 oktober 2022].
- [2] d&b audiotechnik, “R90 Touchscreen remote control.” [Online]. Available: <https://dbaudio.com/global/en/products/processing-and-matrix/r90/>. [Geopend op 16 mei 2023].
- [3] P. Pistek and M. Hudec, “Using SMS for Communication with IoT Devices,” *Mobile Networks and Applications*, vol. 25, no. 3, p. 896–903, 2020.
- [4] K. Schweichhart, “Reference architectural model industrie 4.0 (rami 4.0).” [Online]. Available: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf. [Geopend op 13 februari 2023].
- [5] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, “The industrial internet of things (IIoT): An analysis framework,” *Computers in Industry*, vol. 101, pp. 1–12, 2018.
- [6] Joinup European Commission, “Security by Design.” [Online]. Available: <https://joinup.ec.europa.eu/collection/common-assessment-method-standards-and-specifications-camss/solution/elap/security-design>. [Geopend op 25 februari 2023].
- [7] R. Langmann and L. F. Rojas-Peña, “A PLC as an Industry 4.0 component,” in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pp. 10–15, 2016.
- [8] M. Di Gregorio, G. Nota, M. Romano, M. Sebillio, and G. Vitiello, “Designing usable interfaces for the industry 4.0,” in *Proceedings of the International Conference on Advanced Visual Interfaces*, pp. 1–9, 2020.
- [9] C. Gröger, M. Hillmann, F. Hahn, B. Mitschang, and E. Westkämper, “The Operational Process Dashboard for Manufacturing,” *Procedia CIRP*, vol. 7, pp. 205–210, 2013.
- [10] A. Vázquez-Ingelmo, F. García-Peñalvo, and R. Therón, “Tailored information dashboards: A systematic mapping of the literature,” in *Proceedings of the XX international conference on human computer interaction*, pp. 1–8, 2019.
- [11] T. Hung, *Creating a User Interface For Home Automation [eindwerk]*. Valkeakoski: Häme University of Applied Sciences, 2017.

- [12] J. I. Tamboli, A. Shinde Shailesh, and S. B. Pharne, “Multipurpose Use of Programmable Logic Controller (PLC),” *International Journal of Innovative Research in Technology & Science (IJIRTS)*, vol. 4, pp. 7–10, 2016.
- [13] K. B. Pathak and K. K. Acharya, “Automation in entertainment industry,” in *2011 Nirma University International Conference on Engineering*, pp. 1–6, 2011.
- [14] N. Al-Oudat, “Iot-based sound-level control for audio amplifiers: mosques as a case study,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 859–867, 2022.
- [15] M. H. Schraven, K. Droste, C. G. C. Carducci, D. Müller, and A. Monti, “Open-source internet of things gateways for building automation applications,” *Journal of Sensor and Actuator Networks*, vol. 11, no. 4, p. 74, 2022.
- [16] N. Sharadha, S. Anitha, and J. Pushpanjali, “Developing an ethernet interface using a general purpose microcontroller and ethernet controller for power amplifier,” in *2018 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1073–1076, 2018.
- [17] Beckhoff, “CP6600, CP6606 - Economy built-in Panel PC.” [Online]. Available: <https://beckhoff.com/nl-be/products/ipc/panel-pcs/cp6xxx-cp7xxx-single-touch-panel-pcs/cp6600-cp6606.html>. [Geopend op 19 mei 2023].
- [18] OCA Alliance, “What is AES70?.” [Online]. Available: <https://ocaalliance.com/what-is-aes70/>. [Geopend op 19 mei 2023].
- [19] Beckhoff, “TF8810 - TwinCAT 3 AES70 (OCA).” [Online]. Available: <https://beckhoff.com/nl-be/products/automation/twincat/tfxxxx-twincat-3-functions/tf8xxx-industry-specific/tf8810.html>. [Geopend op 19 mei 2023].
- [20] JSON, “Introducing JSON.” [Online]. Available: <https://json.org/json-en.html>. [Geopend op 19 mei 2023].
- [21] Beckhoff, “TF6020 - TwinCAT 3 JSON Data Interface.” [Online]. Available: <https://beckhoff.com/en-en/products/automation/twincat/tfxxxx-twincat-3-functions/tf6xxx-connectivity/tf6020.html>. [Geopend op 19 mei 2023].
- [22] N. Nikolov, “Research of the Communication Protocols between the IoT Embedded System and the Cloud Structure,” in *2018 IEEE XXVII International Scientific Conference Electronics - ET*, pp. 1–4, 2018.
- [23] Beckhoff, “TF6701 - TwinCAT 3 IoT Communication (MQTT).” [Online]. Available: <https://beckhoff.com/en-en/products/automation/twincat/tfxxxx-twincat-3-functions/tf6xxx-connectivity/tf6701.html>. [Geopend op 19 mei 2023].
- [24] InfluxData, “InfluxDB Time Series Data Platform.” [Online]. Available: <https://influxdata.com/>. [Geopend op 19 mei 2023].
- [25] InfluxData, “Getting Started with Python and InfluxDB.” [Online]. Available: <https://influxdata.com/blog/getting-started-python-influxdb/>. [Geopend op 19 mei 2023].

- [26] InfluxData, “Getting Started with InfluxDB and Grafana.” [Online]. Available: <https://influxdata.com/blog/getting-started-influxdb-grafana/>. [Geopend op 19 mei 2023].
- [27] Odoo, “External API.” [Online]. Available: https://odoo.com/documentation/16.0/developer/reference/external_api.html. [Geopend op 19 mei 2023].