

**Design Document**

**TI-EMS Energie Management System**

**Studenten:**

Arshiya Sharifi 2139515 [a.sharifi@student.avans.nl](mailto:a.sharifi@student.avans.nl)

Nicholas Le Roux 2156763 [nkw.leroux@student.avans.nl](mailto:nkw.leroux@student.avans.nl)

Robin Koedood 2151018 [jj.koedood1@student.avans.nl](mailto:jj.koedood1@student.avans.nl)

Timo Jenkins 2153042 [tjr.jenkins@student.avans.nl](mailto:tjr.jenkins@student.avans.nl)

**Organisatie:** Avans Hogeschool

**Plaats, datum:** Breda, 9-4-2022

**Versie:** 1.0

Inhoudsopgave

[Hoofdstuk 1 – Inleiding 3](#_Toc106889474)

[Hoofdstuk 2 – Architectuur 4](#_Toc106889475)

[Hoofdstuk 3 – Bi-directionele laadpaal koppeling 5](#_Toc106889476)

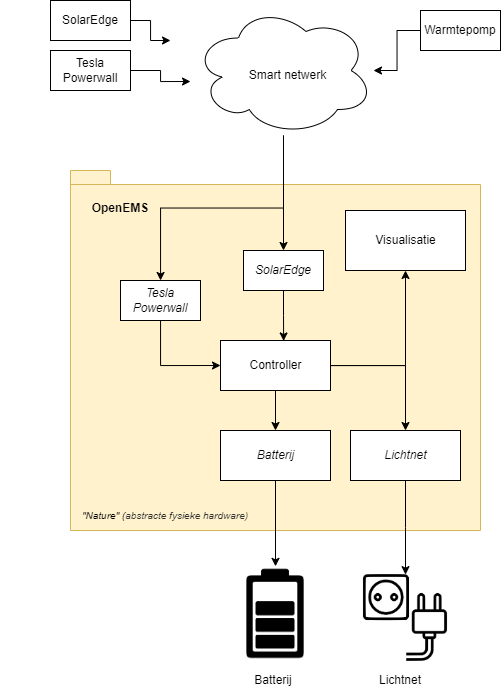
# Hoofdstuk 1 – Inleiding

Tijdens het onderzoek in de analyse fase van dit project is er onderzocht wat de achterliggende vraag is bij het project. Daarnaast zijn er een aantal deelvragen opgesteld i.c.m. een hoofdvraag. De hoofdvraag is: “Op welke wijze kan in het SENDLab een energie managementsysteem (EMS) geïmplementeerd worden, zodanig dat bestaande sensoren en actuatoren gebruikt kunnen worden om het SENDLab energie neutraal te maken.”

In dit document wordt beschreven hoe het ontwerp in elkaar zit voor dat het wordt gerealiseerd. Dit is verdeeld in verschillende onderdelen:

# Hoofdstuk 2 – Architectuur

Er is een globaal ontwerp gesteld voor de architectuur van het netwerk van apparaten verbonden met OpenEMS. Hier onder staat een diagram waarin het de architectuur wordt verduidelijkt.



Figuur 2.1: Globale overzicht

Centraal staat het smart netwerk waar verschillende Nodes aan verbonden staan. Hierbij denkend zijn dit een warmtepomp of het SolarEdge bijvoorbeeld. Dit zijn de echte fysieke hardware die data uitwisselen met het Smart netwerk

Het Smart Netwerk staat vervolgens verbonden met de Natures binnen OpenEMS. Deze natures zijn de abstracte fysieke hardware in de vorm van een OSGi API bundel. Deze bundels zijn vervolgens verbonden met de controller. De controller is vervolgens verantwoordelijk voor de logica waar de apparaten bestuurd mee worden. De controller krijgt invoerdata en hiermee bestuurt het de hardware. Een voorbeeld hiervan is het behouden van een minimumniveau aan stroom bij de batterij bijvoorbeeld. Er is ook een deel visualisatie aan bod. Dit is in feite het UI gedeelte waar er instellingen kunnen worden gedaan via een dashboard naar de controller toe.

# Hoofdstuk 3 – Bi-directionele laadpaal koppeling

Een van de apparaten die met het SENDLAB gekoppeld moeten worden is een bi-directionele laadpaal buiten het gebouw van Avans. Deze koppeling is al deels gerealiseerd door Diederich Kroeske. Binnen de laadpaal is een raspberry pi ingebouwd die staat gekoppeld aan het SENDLAB wifi netwerk doormiddel van SSH.

Het probleem hierbij is dat de raspberry pi nog niet gekoppeld staat met de kaart van de laadpaal. Hierdoor kan er nog geen data uitgewisseld worden tussen het SENDLAB binnen het smart netwerk. Om dit te realiseren moet er eerst een DHCP-server om de raspberry pi gezet worden die de een ip adress geeft aan de kaart van de laadpaal. Er staat al wel een ethernet verbinding alleen is dit niet genoeg. Zo kunnen er requests worden gestuurd naar de laadpaal vanuit het sendlab via een modbus-tcp verbinding. A picture containing table

Description automatically generated

Via bijvoorbeeld PUTTY kan er verbonden worden met de Raspberry PI. Hier kan dan een DHCP op worden geinstaleerd.Text

Description automatically generated

Dit moet eerst worden voltooid voordat er communicatie kan worden gelegd met OpenEMS. Als de laadpaal verbinding eenmaal tot stand staat met de raspberry PI, kan er vanuit OpenEMS een modbus-tcp verbinding worden gelegd om data vanuit de laadpaal uit te lezen en te weergeven binnen OpenEMS.

Om deze verbinding te kunnen realiseren moet er ook een custom module worden gebouwd die communiceert met de laadpaal. Hiervoor zou er volgens de OpenEMS documentatie een OSGI bundel worden aangemaakt waar de connectie gegevens ingevoerd moeten worden zodat OpenEMS kan communiceren met de laadpaal.

De laadpaal levert documentatie over de registers die nodig zijn voor het communiceren over de modbus-tcp verbinding. Deze registers zijn nodig voor het maken van de zelfgemaakte osgi bundel omdat OpenEMS geen standaard component heeft voor de laadpaal. Hiervoor moet zelf een component gebouwd worden met behulp van de modbus-tcp registers.

Zo kan de laadpaal worden gekoppeld met OpenEMS binnen het SENDLAB. Deze stappen zijn nodig om tot dit resultaat te komen.