

**Onderzoeksrapport**

**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, 06-04-2022

**Versie:** 1.0

Inhoudsopgave

[Hoofdstuk 1 - Inleiding 3](#_Toc100752094)

[Hoofdstuk 2 - Onderzoek opzet 4](#_Toc100752095)

[Hoofdstuk 3 - Wat is het SENDLab? 5](#_Toc100752097)

[Wat is het doel van het SENDLab? 5](#_Toc100752098)

[Hoofdstuk 4 - Wat is een energie managementsysteem? 6](#_Toc100752099)

[Welke energie managementsystemen zijn op de markt? (NIET KLAAR – ARSHIYA) 6](#_Toc100752100)

[Hoofdstuk 5 - Wat houdt energieneutraal in? 7](#_Toc100752101)

[Hoofdstuk 6 - Wat voor sensoren/actuatoren zijn er beschikbaar in het SENDLab? 8](#_Toc100752102)

[Hoe werken deze sensoren/actuatoren 8](#_Toc100752103)

[Hoofdstuk 7 - Wat is het Smart Netwerk in het SENDLab? 12](#_Toc100752104)

[Hoe werkt het Smart Netwerk? 12](#_Toc100752105)

[Hoe kunnen apparaten worden gekoppeld met het smart netwerk? 12](#_Toc100752106)

[Hoofdstuk 8 - Wat is OpenEMS? 13](#_Toc100752107)

[Hoe werkt OpenEMS? 13](#_Toc100752108)

[Hoe kunnen er componenten gekoppeld worden met OpenEMS? 17](#_Toc100752109)

[Welke protocollen worden gebruikt voor de communicatie met de apparaten en endpoints (server/OpenEMS)? 18](#_Toc100752110)

[Wat is de Eclipse IDE en wat zijn OSGI bundels? 19](#_Toc100752111)

[Waarom wordt Eclipse IDE en OSGi’s bundles gebruikt voor OpenEMS?​ 19](#_Toc100752112)

[Hoe kan OpenEMS zelfstandig draaien zonder de Eclipse IDE? 20](#_Toc100752113)

[Bibliografie 21](#_Toc100752114)

Inhoudsopgave figuren

[Figuur 6.1 : monitor.py script 9](#_Toc100748366)

[Figuur 6.2: SolarEdge 10](#_Toc100748367)

[Figuur 6.3: ABB B23 11](#_Toc100748368)

[Figuur 7.1: Smart netwerk 12](#_Toc100748369)

[Figuur 8.1: Architectuur structuur 13](#_Toc100748370)

[Figuur 8.2: IPO-model 14](#_Toc100748371)

[Figuur 8.3: Configuratie scherm 14](https://avans.sharepoint.com/sites/STU-Project-Energie-Management-System0020/Shared%20Documents/General/Onderzoeksrapport_V1_TI-EMS_24_03_2022.docx#_Toc100748372)

[Figuur 8.4: Meter types 15](https://avans.sharepoint.com/sites/STU-Project-Energie-Management-System0020/Shared%20Documents/General/Onderzoeksrapport_V1_TI-EMS_24_03_2022.docx#_Toc100748373)

# Inleiding

Dit rapport wordt gebruikt als onderzoek document voor het project TI-EMS. Hierin wordt onderzoek gedaan op de verschillende technieken, software en hardware die voor het project wordt gebruikt. Dit zal de studenten een beter beeld geven van alle onderdelen die tijdens het project gebruikt zullen worden. Onderzoek wordt gedaan per opgestelde deelvraag. Dit rapport is een onderdeel van de tussenoplevering van periode 3.3 en eindoplevering van periode 3.4.

In hoofdstuk 2 wordt het onderzoeksresultaten besproken. Hierin wordt de hoofdvraag en deelvragen onderzocht en antwoord op gegeven. Elke hoofdstuk wordt verdeeld per deelvraag. In hoofdstuk 3 wordt het SENDLab geïntroduceerd en in 3.1 kort uitgelegd wat het doel is. Hoofdstuk 4 bespreekt wat een energie management systeem inhoudt. In 4.1 wordt de verschillende management systemen op de markt opgezocht en besproken. In hoofdstuk 5 wordt onderzocht wat energieneutraal inhoudt. Hoofdstuk 6 bevat het onderzoek over de verschillende sensoren en actuatoren die beschikbaar zijn van het SENDLab. In 6.1 werd ook onderzocht hoe de sensoren en actuatoren werken. In hoofdstuk 7 wordt het Smart Netwerk van het SENDLab besproken. 7.1 wordt uitgelegd hoe het Smart Netwerk werkt. 7.2 bevat hoe de apparaten kunnen gekoppeld met het Smart Netwerk. Hoofdstuk 8 bespreekt hoe OpenEMS werkt. 8.1 bespreekt hoe componenten worden gekoppeld met OpenEMS? 8.1.1 kijkt naar de verschillende protocollen die de componenten gebruiken. 8.1.2 wordt onderzoek gedaan naar hoe de communicatie tussen de componenten werkt. In 8.2 wordt de Eclipse IDE en OSGI bundels onderzocht. In 8.2.1 wordt gekeken waarom de Eclipse IDE en OSGI bundels wordt gebruikt voor OpenEMS. In 8.3 wordt gekeken naar hoe OpenEMS zelfstanding kan laten draaien zonder de Eclipse IDE.

# Onderzoek opzet

Binnen het SENDLab zijn verschillende meters en sensoren die het team TI-EMS niet mee bekend zijn. Om een beter begrip te krijgen van alle onderdelen die wordt gebruikt in het SENDLab moet een onderzoek worden uitgevoerd. In het SENDLab moet een Energie Management Systeem (EMS) worden geïmplementeerd. Om een antwoord te geven op hoe dit kan worden gedaan werd een hoofdvraag geformuleerd. De hoofdvraag is: “Op welke wijze kan in het SENDLab een energie management systeem (EMS) geïmplementeerd worden, zodanig dat bestaande sensoren en actuatoren gebruikt kunnen worden om het SENDLab energie neutraal te maken.” Om tot een conclusie te komen op de hoofdvraag wordt het verdeeld in deelvragen.

De deelvragen zijn:

* D1: Wat is het SENDLab?
  + D1.1: Wat is het doel van het SENDLab?
* D2: Wat is een energie managementsysteem?
  + D2.1: Welke energie management systemen zijn op de markt?
* D3: Wat houdt energieneutraal in?
* D4: Wat voor sensoren/actuatoren zijn er beschikbaar in het SENDLab?
  + D4.1: Hoe werken deze sensoren/actuatoren?
* D5: Wat is het Smart Netwerk in het SENDLab?
  + D5.1: Hoe werkt het Smart Netwerk
  + D5.2: Hoe kunnen apparaten worden gekoppeld met het smart netwerk?
* D6: Wat is OpenEMS?
  + D6.1: Hoe werkt OpenEMS?
  + D6.2: Hoe kunnen er componenten gekoppeld worden met OpenEMS?
    - D6.2.1: Welke protocollen worden gebruikt voor de communicatie met de apparaten en endpoints (server/OpenEMS)?
  + D6.3: Wat is de Eclipse IDE en wat zijn OSGI bundels?
    - D6.3.1: Waarom wordt Eclipse IDE en OSGi’s bundles gebruikt voor OpenEMS?​
  + D6.4: Hoe kan OpenEMS zelfstandig draaien zonder de Eclipse IDE?



# Wat is het SENDLab?

Tijdens dit project wordt er nauw samengewerkt met het SENDlab. Het SENDLAB staat voor Smart Energy Delivery Lab en is een project van het lectoraat Smart Energy op Avans. Het SENDLAB zit gevestigd in gebouw LA lokaal 1.21 van Avans Breda. In het lab wordt onderzoek gedaan naar slimme inzet van duurzame energie. In het SENDLAB staan verschillende opstellingen, sensoren en actuatoren die te maken hebben met het opwekken, opslaan en gebruiken van energie. (Avans Hogeschool, sd)

## Wat is het doel van het SENDLab?

Het doel van het SENDLAB is onderzoek doen naar het slim inzetten van duurzame energie. Tevens moet het lab een volwaardige leeromgeving worden die het motto ‘een leven lang leren’ ondersteunt en die open is voor studenten en onderzoekers van binnen en buiten Avans. (Avans Hogeschool, sd)

Het lectoraat Smart Energy binnen het expertisecentrum Technische Innovatie en de Academie van Engineering en ICT te Breda zijn eigenaar van het SENDLAB. In dit lab wordt onderwijs gegeven en onderzoek uitgevoerd op het gebied van de energietransitie. Een verscheidenheid aan apparatuur staat tot de studenten en docentonderzoekers ter beschikking. Te denken aan opslag, zonnepanelen, inverters, warmtepomp, wind turbines, regelbare consumptie en transmissielijnen. (SENDlab, 2022)

# Wat is een energie managementsysteem?

Voordat onderzoek kan worden gedaan over welke verschillende energie management systemen populair zijn moet eerst gedefinieerd worden wat een EMS is.

Een energie managementsysteem is een systeem dat wordt gebruikt om binnen het stroomnet de prestaties hiervan te bewaken, te regelen en te optimaliseren. Dit kan op grootschalig niveau bij bijvoorbeeld beheerders van elektriciteiten netten of op kleine schaal op een zogenaamd microgrid in bijvoorbeeld het SEND-lab.

## Welke energie managementsystemen zijn op de markt?

Voor dit onderzoek is van belang dat de verschillende systemen die er zijn als randvoorwaarde wel van toepassing kunnen zijn binnen het SEND-lab. Hiermee wordt bedoeld dat niet alle systemen gebruikt kunnen worden binnen het SEND-lab omdat deze daar niet voor zijn bedoeld. In dit onderzoek worden alleen de EMS benoemd die bruikbaar kunnen zijn.

Voor toepasbaarheid van een EMS binnen het SEND-lab zijn er twee varianten:

*open-source* gratis systemen en betaalde *closed-source* systemen. Vele bedrijven zoals Schneider, Etap en Ergon bieden een oplossing aan voor een systeem. Hoewel dit systeem prijzig kan zijn en de software is alleen te wijzigen door het bedrijf zelfs. Hierdoor is als tweede voorwaarde dat dit systeem een open-source oplossing biedt zodat hier binnen het SEND-lab gesleuteld kan worden aan de software door onder andere studenten. Ook biedt het de mogelijkheid om de software op maat in te richten voor het gebruik binnen het SEND-lab.

De grootste aanbieders van een open-source EMS zijn: OpenEMS, OpenRemote en OpenEnergyMonitor. Allemaal bieden deze een open-source variant die zelf gratis toegepast en naar eigen wens bewerkt kan worden.

Echter verschillen de drie wel van elkaar. Hieronder staan de verschillen beschreven:

***OpenEMS***

OpenEMS biedt een volledige open source omgeving waar gewerkt wordt met verschillende losse componenten. Hiernaast is OpenEMS het meest ondersteund met een grote community.  
Binnen het standaard pakket van OpenEMS zitten al ontzettend veel componenten van bestaande hardware. Hierbij denkend bijvoorbeeld een component die praat met een tesla muurbatterij. Deze out-of-the-box features bieden de mogelijkheid om snel te werken met OpenEMS en is een van de sterkste voordelen.

De ontwikkeling wordt verricht door een conbinatie van mensen van verschillende tech bedrijven uit Duitsland die samen OpenEMS hebben gevormd. Een van de leden is BayernWerk. Het overgrote deel van de leden is ook betrokken met smart energy.

Door de constante support en de volledigheid van OpenEMS is het een uiterst geschikte optie voor het SEND-Lab.

***OpenRemote***

Open remote is een nederlands IoT platform vergelijkbaar met OpenEMS. Het enige verschil is dat OpenRemote voornamelijk gefocussed op de front-end en user input. Het verschil met OpenEMS en OpenRemote is hierbij ook dat OpenEMS geen uitgebreide UI biedt voor het regelen van energy management.   
Dit is als reden omdat OpenRemote voornamelijk consument gebaseerd is met persoonlijke en huiselijke apparaaten. Hiernaast biedt OpenRemote automatisatie en een mobiele app.

Voor het koppelen van bestaande hardware en devices zijn er niet veel bestaande hardware gekoppeld. Dit betekent dat de gebruiker dit zelf handmatig moet uitvoeren. Dit is een nadeel vergeleken met OpenEMS waar het inbegrepen zit.

Het gebruik van OpenRemote in plaats van OpenEMS is hierdoor afgeraden omdat het niet is gebouwd voor de apparatuur binnen het SEND-lab. Overigens biedt OpenRemote niet de inbegrepen componenten voor de gebruikte hardware en apparatuur van het SEND-lab.

***OpenEnergyMonitor***

OpenEnergyMonitor is een open-source energy monitoring applicatie waar apparatuur aan verbonden kan worden om het energy gebruik te meten en te visualiseren.

OpenEnergyMonitor is vergeleken met OpenEMS geen geschikte kandidaat voor het SEND-lab. Dit is omdat er niks kan worden beheerd met behoud tot energy management. Met OpenEnergyMonitor kan alleen naar data bekeken worden en hier kunnen berekeningen mee worden gedaan. Er kan geen energy geregeld worden hiermee.

# Wat houdt energieneutraal in?

Ongeveer 55% van alle woningen in Nederland en 75% in België is particulier bezit. Aangezien woningeigenaren zelf beslissen wanneer en wat ze investeren in hun huis hebben ze een grote invloed op het al dan niet behalen van de CO2 emissiereductiedoelen voor 2020. Eigenaren moeten zich herkennen in verduurzamingsopties vooraleer ze investeren, en dit sluit helaas niet naadloos aan op het huidige aanbod van de bouwsector. Het is daarom belangrijk om te bekijken hoe we energieneutraal kunnen gaan worden. Maar wat houdt energieneutraal nou precies in? (Avans, 2022)

Energieneutraal is het balans tussen het energieverbruik en energieproductie gelijk. Dit leidt tot weinig tot geen invloed op het milieu en omgeving dan wat is nodig. Energie neutrale ontwerpen worden veel gebruik in Wireless Sensor Netwerken en IoT producten. De producten moeten zo weinig mogelijk energie gebruiken. Verschillende technieken worden toegepast omdat te realiseren. Een voorbeeld is het compressie van data om alleen de benodigde data naar de hardware te sturen of van de hardware ontvangen.

Energie neutrale ontwerpen worden veel toegepast in gebouwen. Om energieneutraal te worden moet de energie neutrale op jaarbasis netto geen energietoevoer nodig is van energiebronnen buiten de woning. Zo kan een energie neutrale woning wel elektriciteit afnemen van een energieleverancier maar zal deze woning ook zelf elektrische energie opwekken door bijvoorbeeld de installatie van zonnepanelen. Ook op dit gebied moet er sprake zijn van een balans om energieneutraal te zijn (Geertsma, 2018).

# Wat voor sensoren/actuatoren zijn er beschikbaar in het SENDLab?

In het SENDLab staan verschillende smartsensoren. Deze sensoren kunnen uitgelezen worden zodat de data te visualiseren is. De sensoren zijn gekoppeld aan een smart netwerk dat opgesteld is door Maurice. Er wordt verwacht dat deze sensoren gekoppeld gaan worden aan het EMS.

Voorbeelden van sensoren zijn:

* Siemens PAC4200
* Warmtemeter
* SolarEdge
* ABB B23

## Hoe werken deze sensoren/actuatoren

**Siemens PAC4200**

De Siemens PAC4200 is een krachtig compact apparaat die stroom kan meten.

Het apparaat is geschikt voor eenfasige, tweefasige of driefasige fasemeting en kan worden gebruikt in 2-, 3- of 4-draads TN-, TT- en IT-systemen.

Dankzij het grote meetspanningsbereik, is de Siemens PAC4200 handig om te gebruiken in laagspanningssystemen.

Het grafische display wordt gebruikt om alle meetwaarden af ​​te lezen.

De Siemens PAC4200 biedt open communicatie met behulp van de standaard ingebouwd Ethernet Modbus TCP, optionele Modbus RTU-, PROFINET- of PROFIBUS-DP-protocollen voor eenvoudige integratie in elk lokaal of extern bewakingssysteem. De gateway-functionaliteit van dit apparaat vermindert de installatie kosten door andere gateway-apparaten te vervangen en te vereenvoudigen bedrading. Eenvoudige configuratie van de meter kan worden gedaan vanaf het display aan de voorzijde of met behulp van een pc met powerconfig setup software die bij de meter wordt geleverd. (Siemens, 2018)

**Warmtemeter**

De warmtemeter van het SENDLab werkt via MBUS protocol. Daarbij hoort een configuratie, wanneer de configuratie is ingesteld dan wordt MBUS aangezet. De seriële data worden dan uitgelezen en naar de MQTT broker verstuurd. In de MQTT broker komt dan de data te staan.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 6.1: [monitor.py script](https://github.com/AvansETI/SENDLab/blob/main/Warmtemeter/monitor.py)

**SolarEdge**

De SolarEdge, zie Figuur 6.2: SolarEdge, is een systeem wat werkt met zonnepanelen. De zonne-energie gaat vervolgens naar een batterij, vanuit van de batterij kan gebruik gemaakt worden door de andere apparatuur (SolarEdge, 2022).

SolarEdge en het systeem lijkt op wat er gerealiseerd gaat worden binnen het project. Alleen het verschil is dat de SolarEdge maar één component is. In het project willen we meerdere componenten samen gaan koppelen.

Het gene waar SolarEdge zich vooral op focust is de nieuwe technieken die de omvormers hebben. Op dit gebied loopt SolarEdge erg vooruit.

Superieur rendement: 99,5% piekrendement, 98,8% gewogen rendement (SolarEdge, 2022).

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur 6.2: SolarEdge

**ABB B23**

De ABB B23, zie Figuur 6.3: ABB B23, is een smart meter. In het SENDLAB zijn er 5 gevestigd in de meterkast. Via deze meters kan het energie verbruik van het SENDLAB worden uitgelezen. De meters kunnen op dit moment alleen nog worden uitgelezen via een interface van het bedrijf wat de meters heeft geplaatst. Binnen het project is er gevraagd of het mogelijk zou zijn om deze meters zelf uit te kunnen lezen, zonder dat de interface van een extern bedrijf gebruikt moet worden. Uiteindelijk zouden deze meters dan ook kunnen worden opgenomen in het energie management system.

**Afbeelding met tekst, klok, apparaat

Automatisch gegenereerde beschrijving**

Figuur 6.3: ABB B23

Om te kunnen communiceren met de meter kan gebruik worden gemaakt van het Modbus protocol of het M-bus protocol.

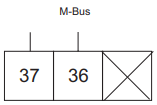
**Modbus**

Modbus is een serieel communicatieprotocol ontwikkeld door Modicon en gepubliceerd in 1979. Het is een methode die wordt gebruikt voor het verzenden van informatie via seriële lijnen tussen elektronische apparaten. Het apparaat dat de informatie opvraagt wordt de Modbus-master genoemd. De apparaten die de informatie leveren worden de Modbus-slaves genoemd. In een standaard Modbus-netwerk is er één Master en maximaal 247 Slaves. Elke Slave heeft een uniek adres van 1 tot 247. De Master kan tevens ook informatie naar de Slaves schrijven.

**M-bus**Het M-Bus of Meter-Bus protocol wordt gebruikt voor het op afstand uitlezen van water-, gas- of elektriciteitsmeters. M-Bus is ook inzetbaar voor andere soorten verbruiksmeters. De M-Bus-interface is gemaakt voor communicatie over twee draden, waardoor het kosteneffectief is.

Het communicatie protocol wat wij willen gaan gebruiken voor deze meter is M-bus. We willen dit protocol gebruiken omdat openEMS al een component heeft van de ABB B23 wat via M-bus gaat. Om eerst te testen hoe de meter werkt hebben we één meter gebruikt voor een test opstelling. Hierbij zit de meter aangesloten op een lamp.

Aansluiting van de meter voor M-bus:

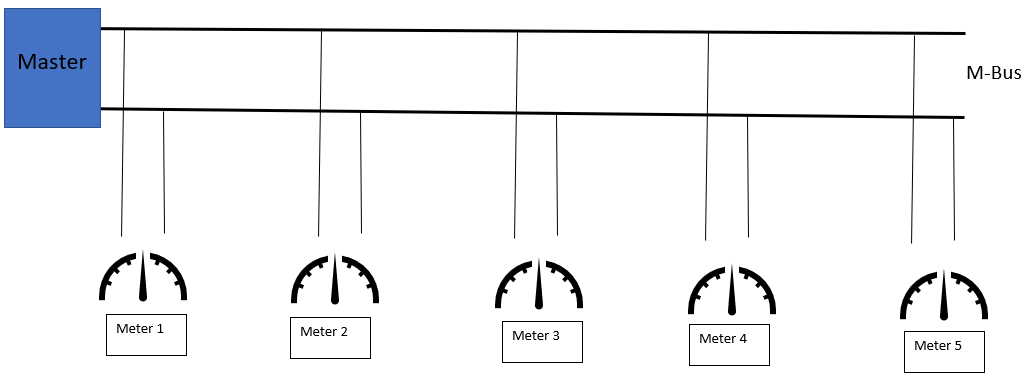


Instelling op de meter voor M-bus:

Afbeelding met tekst, binnen, schermafbeelding

Automatisch gegenereerde beschrijving

**Architecture**

****

# Wat is het Smart Netwerk in het SENDLab?

Het Smart Netwerk is een MQTT Cloud gebaseerde oplossing die data ontvangt en verstuurd via een MQTT broker op de server. Het wordt gebruikt om de data van alle sensoren en apparaten die binnen het SENDLab bevinden te bewerken.

## Hoe werkt het Smart Netwerk?

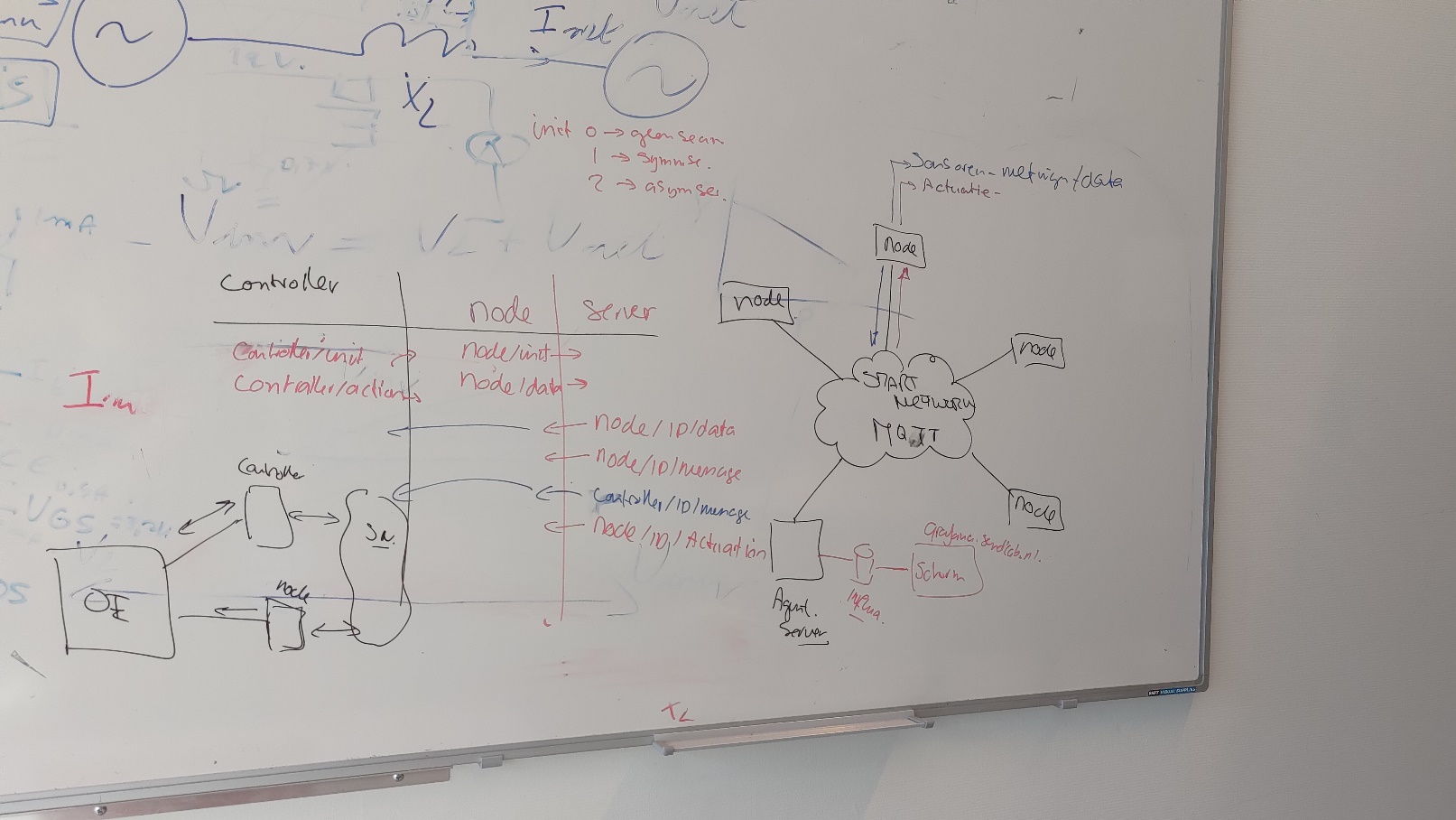
Het Smart Netwerk bestaat uit meerdere Smart Sensoren en een server/agent. Communicatie binnen het netwerk wordt doormiddel van MQTT gedaan. De data die de server ontvangt worden opgeslagen op de database. De data van de database kan vervolgens worden gebruikt door een GUI zoals Grafana voor data weergave.

Alle Smart Sensoren staan met elkaar in verbinding door middel van een Zigbee mesh-netwerk. Eén van de sensoren dient als een coördinator te dienen om de meetgegevens uit het systeem te krijgen. Via deze coördinator kunnen tevens commando’s naar de verschillende sensoren gestuurd worden om bijvoorbeeld een batterij op te laden of de verwarming in te stellen op 20 graden Celsius. Elke sensor die is gekoppeld aan het smart netwerk heeft een ID gekregen met een topic op de MQTT broker. Hieruit kan MQTT clients op de topics abonneren en de data ervan ontvangen.

## Hoe kunnen apparaten worden gekoppeld met het smart netwerk?

Apparaten kunnen worden gekoppeld met de server via MQTT. Elke sensor wordt een node. Hierbij kan informatie doorgestuurd naar de agent/server. Het is ook mogelijk om nodes direct te koppelen aan een controller. Data wordt dan door de controller verstuurd naar server. De controller heeft dan ook controle over de node want elke node kan maar aan 1 controller worden gekoppeld.

Om een node te koppelen aan de smart netwerk moet eerst een init bericht worden verstuurd naar de server. Wanneer de node een ok bericht ontvangt kan het data versturen naar de server, anders wordt het niet opgeslagen. Voor een controller moet dezelfde stappen worden uitgevoerd. Als de controller data wil opvragen van de server moet een actie worden uitgevoerd. De server geeft een topic terug van de node’s ID en de bijbehorende data.



Figuur 7.1: Smart netwerk

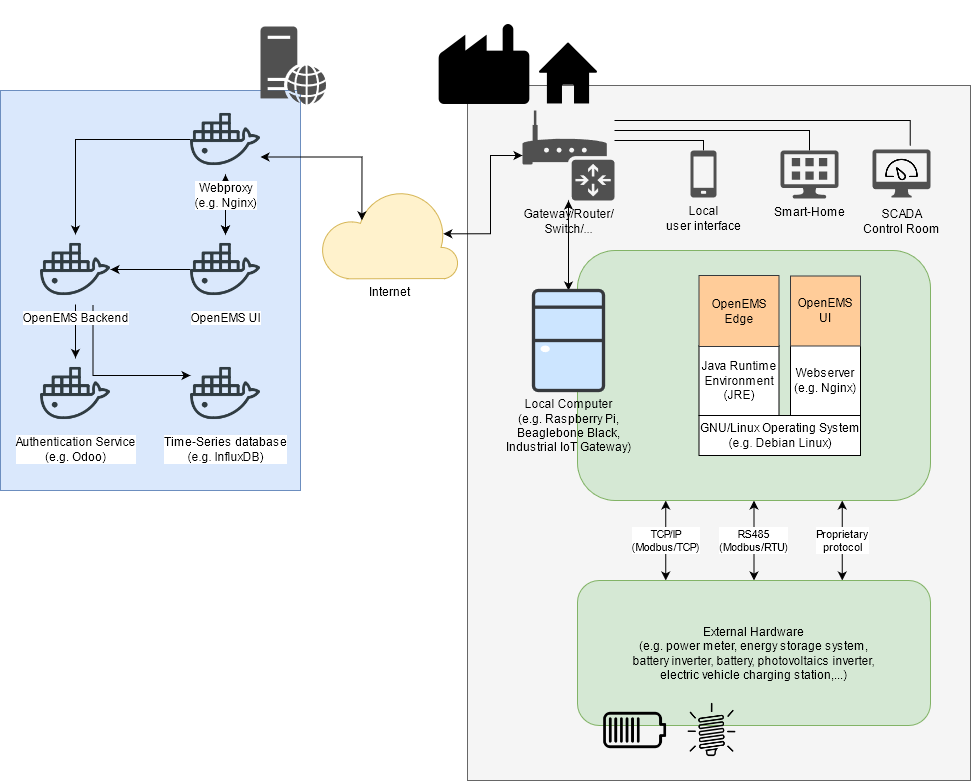
# Wat is OpenEMS?

OpenEMS is een modulair platform voor energiebeheertoepassingen. Het is ontwikkeld rond de vereisten van het bewaken, regelen en integreren van energieopslag samen met hernieuwbare energiebronnen en aanvullende apparaten en diensten zoals oplaadstations voor elektrische voertuigen, warmtepompen, elektrolysers, elektriciteitstarieven op basis van gebruiksduur en meer.

## Hoe werkt OpenEMS?

OpenEMS bestaat uit een IoT Stack:

* OpenEMS Edge draait op locatie, communiceert met apparaten en services, verzamelt gegevens en voert besturingsalgoritmen uit.
* OpenEMS UI is de realtime gebruikersinterface voor webbrowsers en smartphones.
* OpenEMS Backend draait op een (cloud)server, verbindt de decentrale Edge-systemen en zorgt voor aggregatie, monitoring en controle via internet.



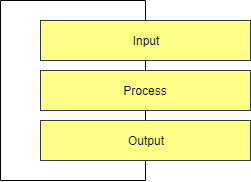
Figuur 8.1: Architectuur structuur

**Edge architectuur:**

De OpenEMS Edge wordt gebruikt voor het SENDLab. De Edge is geïmplementeerd in de programmeertaal Java en vereist een Java Runtime Environment (JRE). De Edge moet worden opgestart via de Eclipse IDE.

*Input-Process-Output*

OpenEMS Edge is gebouwd rond het bekende IPO-model (input-process-output) dat de interne uitvoeringscyclus definieert, zie Figuur 8.2: IPO-model.



Figuur 8.2: IPO-model

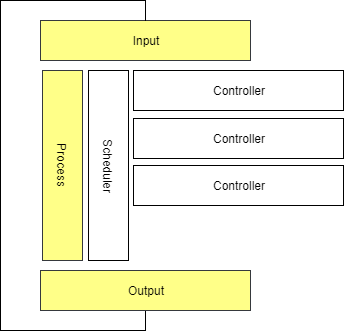
* Invoer: Tijdens de invoerfase wordt alle relevante informatie wordt verzameld en als procesbeeld aangeleverd. Dit procesbeeld verandert gegarandeerd nooit tijdens de cyclus.
* Proces: De procesfase voert algoritmen en taken uit op basis van het procesbeeld.
* Uitgang: De outputfase haalt de resultaten uit de procesfase en past deze toe.

*Controller*

Controllers zijn componenten die kanaalgegevens gebruiken en bedrijfslogica bevatten, b.v. het besturingsalgoritme dat invoergegevens evalueert en setpoints definieert voor de bestuurde hardware. Controllers zijn niet noodzakelijkerwijs beperkt tot besturingsalgoritmen. Verbindingen met backend-systemen worden geïmplementeerd als controllers. Dit zorgt ervoor dat elk instelpuntverzoek door een extern systeem wordt ingebed in het lokale prioriteringssysteem en natuurlijk wordt beperkt door regelaars met een hogere prioriteit.

*Scheduler*

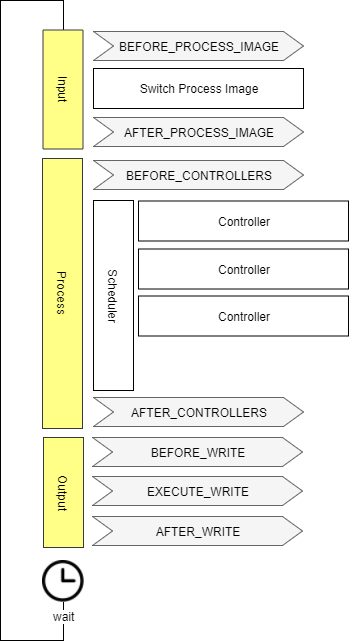
Tijdens de 'proces'-fase kunnen verschillende algoritmen (controllers) proberen toegang te krijgen tot dezelfde bronnen - b.v. twee controllers proberen dezelfde digitale uitgang te schakelen. Het is daarom noodzakelijk om prioriteit te geven aan de uitvoering ervan en de toegang te beperken op basis van prioriteit. OpenEMS Edge gebruikt Scheduler-implementaties om een gesorteerde lijst met controllers te ontvangen. De Controllers worden vervolgens in volgorde uitgevoerd. Later uitgevoerde controllers mogen een eerder geschreven resultaat niet overschrijven.



Figuur 8.3: IPO model met Scheduler en Controllers

*Cyclus*

Het input-process-outputmodel in OpenEMS Edge wordt uitgevoerd in een Cycle - geïmplementeerd door de Cycle-component. Het zorgt voor het instellen van een procesbeeld in de invoerfase en voert de controllers uit in de procesfase. Verder zendt het Cyclusgebeurtenissen uit die in andere Componenten kunnen worden gebruikt om met de Cyclus te synchroniseren.



Figuur 8.4: OpenEMS Edge Cycle

*Process Image*

Vanwege asynchrone communicatie met externe apparaten en services kunnen gegevens op elk moment mogelijk worden bijgewerkt of ongeldig worden gemaakt. Dit kan tot verwarrende situaties leiden, b.v. waarbij een kanaalwaarde verandert tussen twee opeenvolgende controllers die op zijn gegevens reageren. Om deze situaties te voorkomen en de programmeur te ontlasten van allerlei gelijktijdigheidsproblemen, gebruikt OpenEMS een "Process Image", een techniek die zich goed heeft bewezen op het gebied van PLC-programmering. Het idee is om de producenten en consumenten van data los te koppelen en een centrale buffer in te voeren voor alle kanaaldata. Deze buffer - de Process Image - wordt slechts één keer in elke rekencyclus bijgewerkt wanneer het de nieuwste gegevens in elk kanaal activeert.

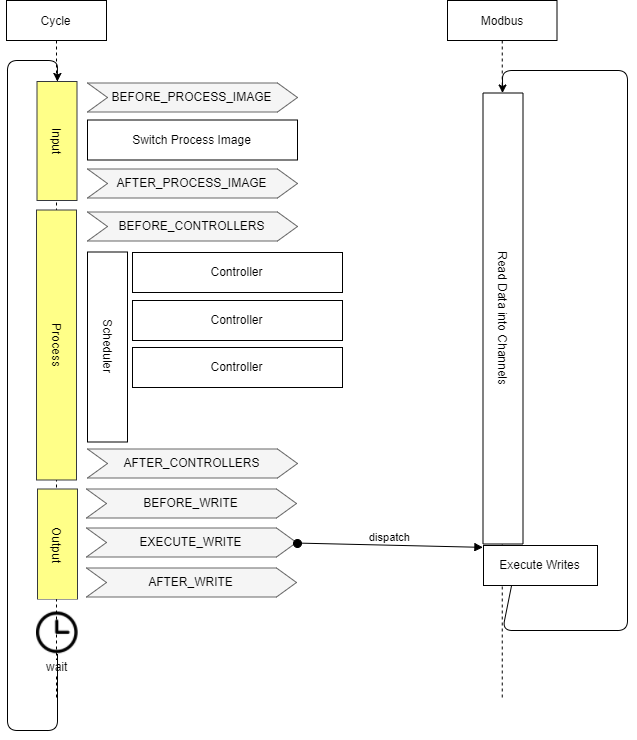
Daarom heeft de implementatie van kanaalobjecten in OpenEMS twee gegevensvariabelen: - Het waardeveld dat de huidige actieve waarde bevat die door consumenten moet worden gebruikt - Het veld nextValue dat de laatste gegevens vertegenwoordigt die zijn ontvangen, b.v. via Modbus-communicatie.

Bij - en alleen bij - 'Switch Process Image' van de Cyclus wordt de nextValue gekopieerd naar het waardeveld. Dit zorgt ervoor dat de gegevens in de Process Image niet veranderen tijdens een rekencyclus.

*Asynchrone threads en cyclussynchronisatie*

Communicatie met externe hardware en services moet worden uitgevoerd in asynchrone threads om het systeem niet te blokkeren. Tegelijkertijd moeten die threads synchroniseren met de Cyclus.

Het volgende voorbeeld laat zien hoe de Modbus-implementatie Cycle Events gebruikt om te synchroniseren met de Cycle:

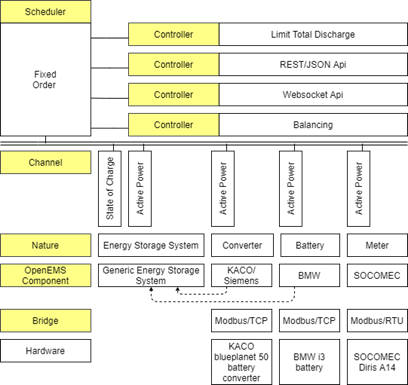


Figuur 8.5: Synchroniseer Cyclus met Modbus lezen/schrijven

*Architectuurschema*

De OpenEMS Edge-softwarearchitectuur is zorgvuldig ontworpen om de communicatie- en besturingsalgoritmen van apparaten te abstraheren op een manier om maximale flexibiliteit, voorspelbaarheid en stabiliteit te bieden, terwijl het implementatieproces van nieuwe componenten wordt vereenvoudigd.

Het volgende schema toont de abstractie van hardware via Channels, Natures en Devices, evenals de uitvoering van besturingsalgoritmen via Scheduler en Controllers.



Figuur 8.6: Architectuurschema

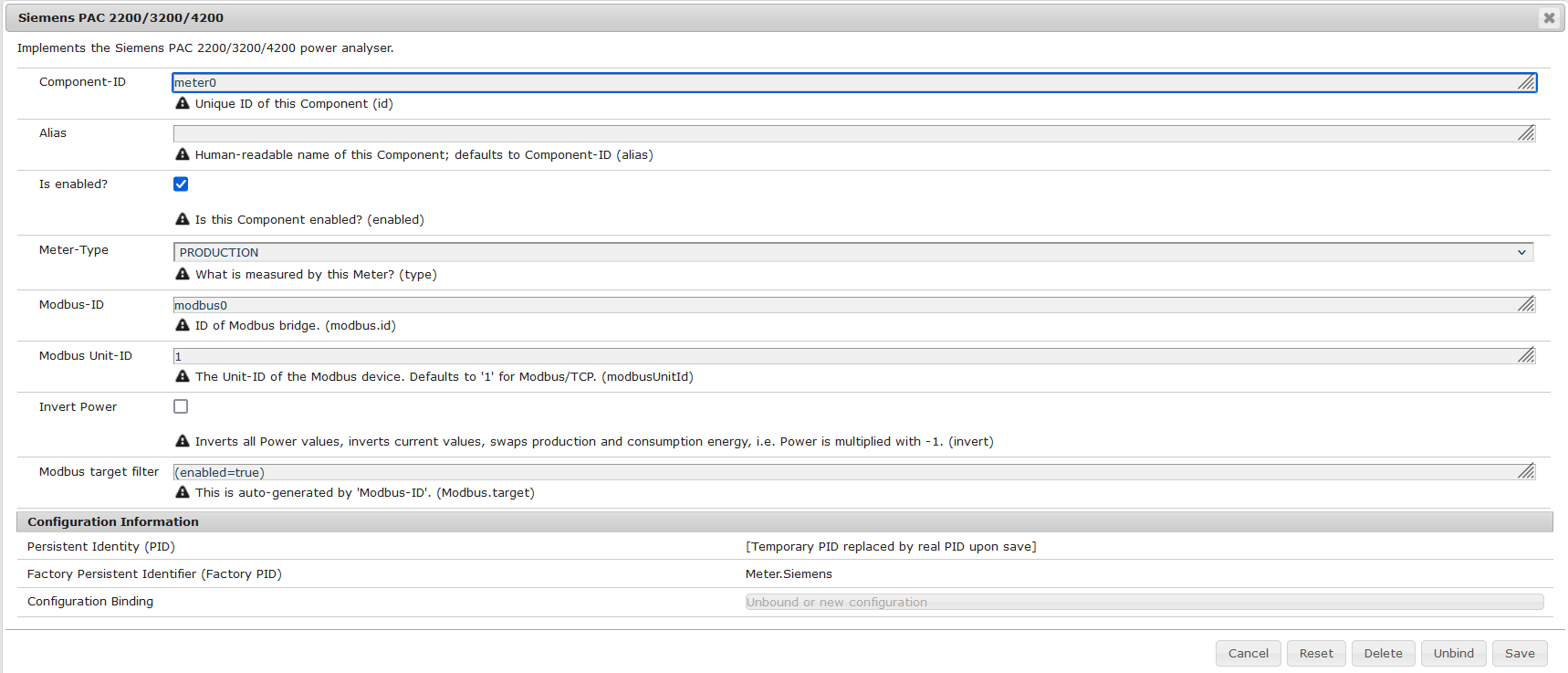
## Hoe kunnen er componenten gekoppeld worden met OpenEMS?

In Figuur 8.3: Configuratie scherm is te zien hoe een component gekoppeld wordt aan OpenEMS. Voordat de component gekoppeld is, moet er bepaalde gegevens ingevuld worden.

In het eerste veld wordt er gevraagd naar de component-id. Dit is een ID die uniek moet zijn, het is niet mogelijk dat er 2 componenten “meter0” heten.

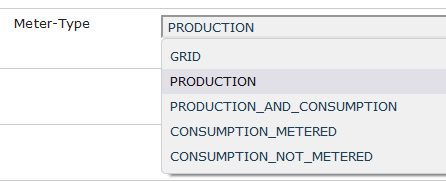
In het volgende veld wordt er een alias gevraagd, dit is dan de naam die we terug kunnen vinden. Dit is optioneel om in te vullen.

Als derde zien we een checkbox waar aangegeven kan worden of de component aan staat of niet.



Figuur 8.7: Configuratie scherm

Het vierde veld gaat over wat voor type de meter is. De verschillende types zijn te zien in Figuur 8.4: Meter types



Figuur 8.8: Meter types

Daarna komen de velden die gaan over het protocol die gebruikt wordt om te communiceren.

### Welke protocollen worden gebruikt voor de communicatie met de apparaten en endpoints (server/OpenEMS)?

Elk apparaat dat gekoppeld is met OpenEMS praat op zijn eigen manier. Sommige apparaten maken gebruik van Modbus, andere van M-bus. De meeste componenten praten op die manier. Maar er zijn ook uitzonderingen, de Tesla powerwall core praat via een api port naar OpenEMS. Ook zijn er verschillende componenten voor databases en controllers. Deze praten meestal via IP of via MQTT.

## Wat is de Eclipse IDE en wat zijn OSGI bundels?

De Eclipse IDE is een Intergrated Development Environment (IDE) die beroemd is voor Java ontwikkeling. Het kan ook worden gebruikt voor C/C++, JavaScript/TypeScript, PHP en meer. De Eclipse Project is onderdeel van de Eclipse Foundation. De Foundation bevat de Eclipse IDE, Jakarta EE en meer dan 350 open source projecten; waaronder runtimes, tools en frameworks voor een breed scala aan technologische domeinen zoals het internet der dingen, automotive, geospatial, systems engineering en vele andere. The Eclipse Project was gemaakt door IBM in nov 2001. De Eclipse Foundation was gemaakt in Jan 2004 als een onafhankelijke non-profitorganisatie om op te treden als rentmeester van de Eclipse-gemeenschap. De onafhankelijke non-profitorganisatie is opgericht om een leveranciersneutrale, open en transparante gemeenschap rond Eclipse tot stand te brengen (N.M.M., n.d.).

OSGi is een modulesysteem dat compatibel is met systemen die op Java zijn gebaseerd en een dynamisch componentmodel implementeert. Enterprise-systemen kunnen OSGi gebruiken om de onderhoudbaarheid van runtime-infrastructuren te verbeteren. Applicaties, in de vorm van bundels, kunnen op afstand worden geïnstalleerd, gestart, gestopt, bijgewerkt en verwijderd zonder dat een herstart nodig is (OSGi overview, 2021).

Het OSGi-applicatieframework biedt een programmeermodel voor het ontwikkelen, samenstellen en implementeren van modulaire applicaties die gebruikmaken van Java EE- en OSGi-technologieën. OSGi-tools voor applicatieontwikkeling bieden een manier om bedrijfsapplicaties te bouwen die profiteren van de modulariteit, dynamiek, versiecontrole en bibliotheekintegratie van derden die wordt geboden door het OSGi-applicatieframework (Learn about OSGi applications, 2021).

Een OSGi-bundel is een Java-archiefbestand dat Java-code, bronnen en een manifest bevat dat de bundel en zijn afhankelijkheden beschrijft. De bundel is de implementatie-eenheid voor een applicatie (OSGi bundles, 2021).

### Waarom wordt Eclipse IDE en OSGi’s bundles gebruikt voor OpenEMS?​

De keuze voor het gebruiken van deze technieken is bewust gemaakt. Om deze vraag goed te kunnen beantwoorden zijn er een aantal vragen gesteld aan een van de voor

## Hoe kan OpenEMS zelfstandig draaien zonder de Eclipse IDE?

Op de server van het SENDLab zijn verschillende scripts die applicaties draaien. Hiervoor moet OpenEMS in een container laten draaien. OpenEMS kan op twee manieren laten draaien, via de Eclipse IDE in debug modus te draaien of de applicatie naar een Jar bestand te exporteren. De jar bestand kan dan via de console of shell laten runnen door “java -jar openems.jar” voor de edge of “java -jar openems-backend.jar” voor de backend in te typen.

Om de applicatie naar een Jar bestand te exporteren op de Eclipse IDE moet de volgende stappen worden doorgelopen (Build OpenEMS Edge, sd) (Build OpenEMS Backend, sd):

*Backend*

1A. Open in Eclipse IDE het project io.openems.backend.application en open het bestand EdgeApp.bndrun.

*Edge*

1B. Open in Eclipse IDE het project io.openems.edge.application en open het bestand EdgeApp.bndrun.

1. Druk op [Exporteren] / [Export] om de Assistent Selectie Wizard Exporteren te starten.
2. Selecteer [Uitvoerbare JAR] / [Executable JAR] en druk op [Volgende >] / [Next >].
3. Selecteer een bestemming voor export naar JAR.
4. Druk op [Voltooien] / [Finish]

# Bibliografie

Avans. (2022, April 13). *Project EnergieNeutraal LEvensloopbestendig Bouwen -ENLEB*. Opgehaald van Project EnergieNeutraal LEvensloopbestendig Bouwen -ENLEB: https://www.avans.nl/onderzoek/projecten/detail/energieneutraal-levensloopbestendig-bouwen-enleb

Avans Hogeschool. (sd). *Project Smart Energy Delivery (SEND) Lab*. Opgehaald van avans.nl: https://www.avans.nl/onderzoek/projecten/detail/smart-energy-delivery-send-lab/introductie

*BEMServer*. (2022). Opgehaald van BEMServer: bemserver.org

*Build OpenEMS Backend*. (sd). Opgeroepen op 03 24, 2022, van OpenEMS: https://openems.github.io/openems.io/openems/latest/backend/build.html

*Build OpenEMS Edge*. (sd). Opgeroepen op 03 24, 2022, van OpenEMS: https://openems.github.io/openems.io/openems/latest/edge/build.html

Geertsma, P. (2018, 1 4). *Wat is energieneutraal?* Opgehaald van Technisch Werken: https://www.technischwerken.nl/kennisbank/duurzaamheid/wat-is-energieneutraal/

*Learn about OSGi applications*. (2021, 08 31). Opgehaald van IBM: https://www.ibm.com/docs/en/wasdtfe?topic=applications-learn-about-osgi

N.M.M. (sd). *About the Eclipse Foundation*. Opgehaald van Eclipse Foundation: https://www.eclipse.org/org/

*Open Remote*. (2022). Opgehaald van Open Remote: https://openremote.io/energy-management/

*OpenEnergyMonitor*. (2022). Opgehaald van OpenEnergyMonitor: https://openenergymonitor.org/

*OSGi bundles*. (2021, 08 31). Opgehaald van IBM: https://www.ibm.com/docs/en/wasdtfe?topic=projects-osgi-bundles

*OSGi overview*. (2021, 08 31). Opgehaald van IBM: https://www.ibm.com/docs/en/wasdtfe?topic=applications-osgi-overview

SENDlab. (2022, April 7). *SENDlab*. Opgehaald van SENDlab: https://sendlab.nl/

Siemens. (2018). *PAC4200 Power Meter.* Opgehaald van PAC4200 Power Meter: https://assets.new.siemens.com/siemens/assets/api/uuid:fe4ef16e-2765-4b58-b352-a8010640c3c0/sie-cm-pac-4200-meter-brochure.pdf

SolarEdge. (2022, April 7). *De toekomst van omvormers als Smart Energy-managers*. Opgehaald van SolarEdge: https://www.solaredge.com/nl/products/pv-inverters#/

SolarEdge. (2022, April 7). *Power Optimizer Module Add-on*. Opgehaald van SolarEdge: https://www.solaredge.com/nl/products/power-optimizers#/