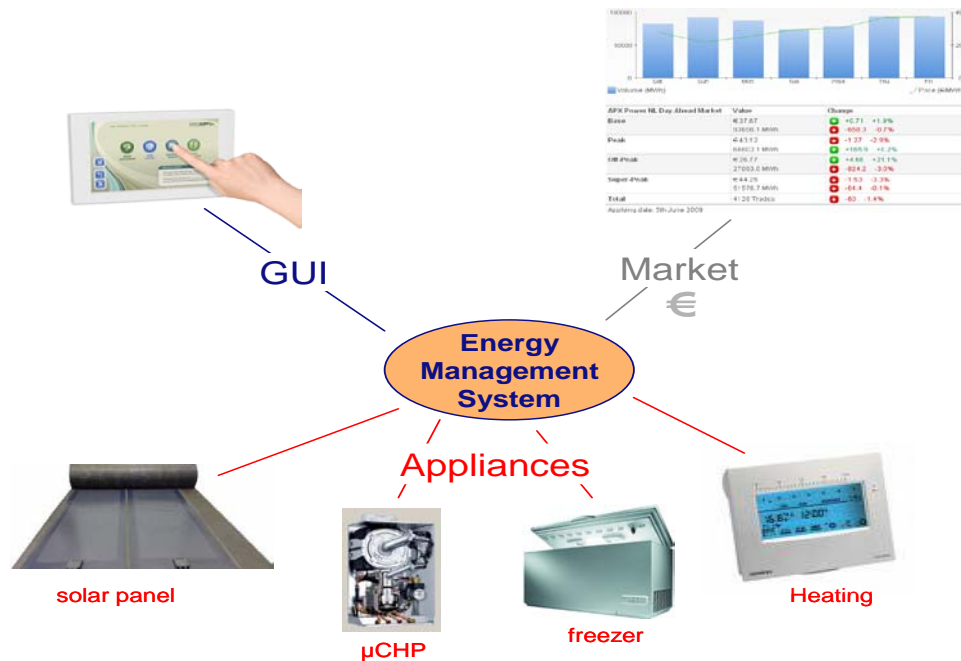


Flexines architectuur

Introductie

Het doel van het Flexines project is om een Energie Management Systeem (EMS) te ontwikkelen voor huishoudens. Figuur 1 geeft de scope van het EMS aan.



Figuur 1: Scope Flexines EMS

Er zijn 3 hoofdtaken die het EMS moet uitvoeren:

1. **Aansturen van de apparatuur.** De apparaten in een huishouden zijn in combinatie met de thermische eigenschappen van het huis en/of van de apparaten, de bron voor flexibiliteit (bijvoorbeeld door het schuiven van een wasprogramma). Naast het benutten van deze flexibiliteit moet het EMS ook rekening houden met de comfort eisen die door een gebruiker aan het apparaat worden gesteld (bv. temperatuur van de verwarming die tussen 19 en 21 graden moet liggen). Ten slotte kunnen er ook nog randvoorwaarden vanuit het apparaat zijn die gerespecteerd moeten worden. Zo zijn er veel apparaten waarbij het kort achter elkaar in en uit schakelen een negatief effect heeft op de levensduur. Het EMS zal ook deze randvoorwaarden moeten bewaken.
2. **Communicatie met de energie markt.** De potentiële flexibiliteit in een huishouden is geld waard voor andere spelers op de energiemarkt. Denk aan programma verantwoordelijken die zo beter in staat zijn om aan hun energie programma's te voldoen. Een andere partij die belang heeft bij flexibiliteit is de netbeheerder. De netbeheerder kan bijvoorbeeld de aangesloten huishoudens stimuleren om te schuiven met consumptie of productie teneinde congestie op zijn netwerk te voorkomen. Het is zelfs denkbaar dat huishoudens onderling energie uitwisselen als de voorwaarden hiervoor gunstiger zijn dan afname bij de gebruikelijke energieleveranciers. Het EMS moet de verschillende aanbieders en afnemers

op de energiemarkten in de gaten houden en continu monitoren waar de beste deals gesloten kunnen worden.

3. **Terugkoppeling aan gebruikers.** Gebruikers zullen niet zondermeer hun energiehuishouding aan een EMS overdragen. Daarvoor is het van belang dat de gebruiker het EMS kan vertrouwen. Gebruikers moeten controle houden over de randvoorwaarden waarbinnen het EMS opereert en moeten ook een duidelijke terugkoppeling krijgen over de acties van het EMS met de daaraan verbonden voordelen die de acties opleveren.

De hierboven genoemde hoofdtaken van het EMS moeten gelijktijdig worden uitgevoerd. Hiervoor moet veel informatie worden uitgewisseld op basis waarvan het EMS continu actie moet kunnen ondernemen. Flexines is daardoor technische gezien meer een IT - als een energie project.

Om te zorgen dat deze soms complexe taken op een goede manier door het EMS worden uitgevoerd is een transparante IT architectuur onontbeerlijk. De architectuur ordent de modules die nodig zijn om het EMS te laten functioneren en de informatie die tussen de modules moet worden uitgewisseld.

Voordat verder wordt ingegaan op de architectuur zelf zal er eerst worden stil gestaan bij de belangrijkste uitgangspunten die aan de IT architectuur worden gesteld.

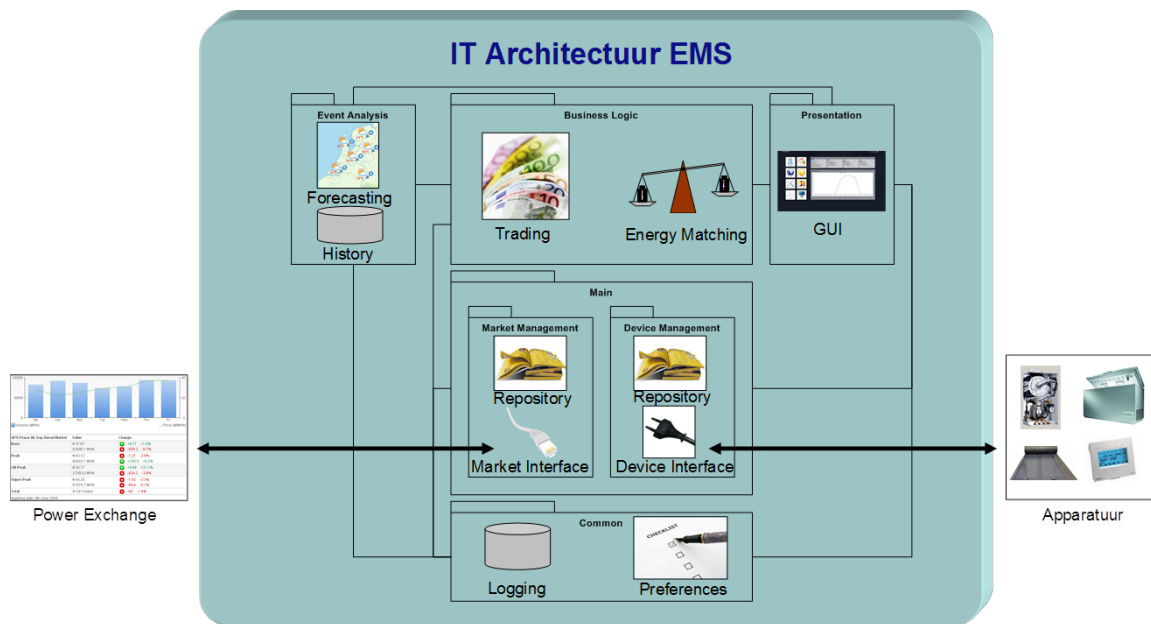
Uitgangspunten

In de beginfase van het project heeft een inventarisatie van requirements plaatsgevonden. In deze sectie worden de belangrijkste de belangrijkste architectuur uitgangspunten die tijdens de requirements fase naar voren zijn gekomen, belicht.

- *Het EMS moet de privacy en daarmee samenhangende belangen van alle betrokken partijen beschermen.* Dit uitgangspunt komt er op neer dat de huishoudens autonoom hun beslissingen kunnen nemen zonder dat andere partijen daar invloed op kunnen uitoefenen.
- *Elk onderdeel van het EMS blijft autonoom functioneren bij communicatie-uitval.* De subsystemen van het EMS en de aangesloten apparatuur moeten zich op een veilige en van tevoren vastgestelde manier gedragen ui uitval en herstel van communicatie met het EMS en/of de buitenwereld.
- *Het EMS moet op elk niveau leverancier onafhankelijk zijn.* Het EMS moet niet afhankelijk zijn van één leverancier. Een eindgebruiker moet bijvoorbeeld zonder problemen apparatuur van het ene merk kunnen vervangen door een ander merk.
- *Het EMS moet kunnen handelen op een marktplaats.* Het moet mogelijk zijn om op basis van (verwachte_ consumptie, productie en buffercapaciteit te kunnen handelen op een marktplaats. Daarbij moeten verschillende marktmodellen kunnen worden ondersteund.
- *Er moet een generiek protocol zijn voor de communicatie met apparaten.* De flexibiliteit die door een apparaat geboden wordt moet op een generieke manier worden beschreven. Hierdoor kan het EMS met een breed scala aan apparaten omgaan.

Architectuur overzicht

In Figuur 2 wordt een schematisch overzicht gegeven van de IT architectuur van het EMS. De belangrijkste modules zijn: Device Management, Energy Matching, Trading en Market Management. De onderdelen van de architectuur worden hieronder verder toegelicht.



Figuur 2: Overzicht Flexines EMS architectuur

Device Management

Deze module is verantwoordelijk voor de juiste aansturing van de apparaten. De fysieke communicatie met apparatuur gebeurt via de device interface (ook wel device driver), hierin worden de instructies vanuit device management omgezet naar de juiste protocollen voor het apparaat (bv. ZigBee, PlugWise, PLC). Ook geeft de device interface essentiële parameters door naar device management (bv. temperatuur van een koelkast). De Device Manager kent de karakteristieken van een apparaat (bv. opwarm curve) en de gewenste instelling door de eindgebruiker (bv. temperatuur tussen 2 en 5 graden celsius). Met behulp van al deze informatie wordt bepaald wat de energetische flexibiliteit van het apparaat is, dit wordt via een 'regelruimte' bericht doorgegeven naar de Energy Matching module. Er worden 4 soorten energetische flexibiliteit onderscheiden:

- **Timeshifters.** Dit zijn apparaten die hun consumptie of opwek over een bepaalde periode kunnen verschuiven. Denk aan een wasmachine met uitgestelde start mogelijkheid. Parameters in de 'regelruimte' van een timeshifter zijn: een energie profiel en de periode waarbinnen het start moment van dat profiel moet vallen.
- **Buffers.** Kenmerk van een bufferend apparaat is dat deze tijdelijk meer energie kunnen opnemen (of opwekken zoals een WKK), waardoor ze later met minder toe kunnen of andersom. Het gaat hierbij vaak om thermische buffers zoals huisverwarming en koelkasten. Voorbeelden van parameters voor de buffer 'regelruimte' zijn totale buffercapaciteit, vulgraad, oplaadcurve en ontladcurve.
- **Storage.** Een storage apparaat lijkt in veel opzichten op een buffer, maar kan zowel energie opnemen als terugleveren. Naast parameters die de buffer beschrijven heeft een storage 'regelruimte' ook parameters voor bijvoorbeeld omzetverliezen.
- **Uncontrolled load/generation.** Zoals de naam al aangeeft is het energie gedrag van deze categorie van apparaten niet aan te sturen (PV panelen, televisies, computers, etc.). Voor dergelijke apparaten wordt dan ook alleen een voorspelling van de te verwachten opwek/load doorgestuurd, zodat de rest van het platform hier rekening mee kan houden.

Device management bewaakt ook de randvoorwaarden op het gebied van comfort en de operationele grenzen waarbinnen een apparaat moet functioneren. Op het moment dat device management om wat voor reden dan ook geen instructies ontvangt over het aansturen van de apparatuur wordt er op een default gedrag teruggevallen waarbij deze randvoorwaarden gewaarborgd worden. Ook is het zo dat een instructie die ertoe zou leiden dat de randvoorwaarden gecompromitteerd worden niet wordt uitgevoerd.

Energy Matching

Deze component verzamelt informatie over de energetische mogelijkheden van de aangesloten apparaten middels de 'regelruimtes' en ontvangt ook informatie van andere marktpartijen (bv. leveranciers, netbeheerder, marktplaatsen) via de trading module. Energy Matching is verantwoordelijk voor de optimale afstemming van opwek, verbruik en storage binnen de doelstellingen van de eindgebruiker. Deze doelstellingen kunnen zeer uiteenlopen. Hieronder enkele voorbeelden:

- **Financiën.** Optimaliseren naar laagste kosten of juist streven naar hoogste opbrengsten (maximaal leveren bij hoge prijs).
- **Infrastructuur.** Optimaliseren naar de laagste net belasting.
- **Bedrijfsvoering.** Optimaliseren naar laagste uitwisseling met het net (zoveel mogelijk zelfvoorzienend zijn).

Binnen Flexines is de keuze gemaakt om te optimaliseren op basis van financiën. Het optimalisatie algoritme zelf kan in principe vrij gekozen worden, zolang het maar om kan gaan met de in- en output berichten van de Energy Matching module. Binnen het project zijn er twee verschillende 'matchings' algoritmen geïmplementeerd.

De beslissingen die door de Energy Matching module zijn genomen worden via een 'allocatie' bericht naar de device managers gestuurd.

Trading

De trading module kijkt welke deals er te maken zijn op de energiemarkt. Hiervoor ontvangt de trading module input vanuit energy matching. Die geeft aan of er een overschot verkocht moet worden dan wel compensatie voor een tekort moet worden ingekocht.

Market Management

Verschillende spelers op de energiemarkt hanteren verschillende protocollen om informatie uit te wisselen. Zo zal de informatie die met een netbeheerder wordt uitgewisseld een heel andere vorm hebben dan het verhandelen van volumes op een energieveiling. De market management module zet de informatie vanuit het EMS om naar de benodigde protocollen via de market interface.

Presentation

De presentation module bestaat uit de GUI. Deze staat in contact met alle hiervoor beschreven modules. Via de GUI wordt allerlei informatie vanuit de modules voor de gebruiker ontsloten, ook kan de gebruiker met behulp van de GUI parameters in de modules instellen.

De GUI is een web gebaseerde interface. Deze keuze is gemaakt zodat de GUI op een breed scala aan devices kan worden getoond.

Communicatie tussen modules

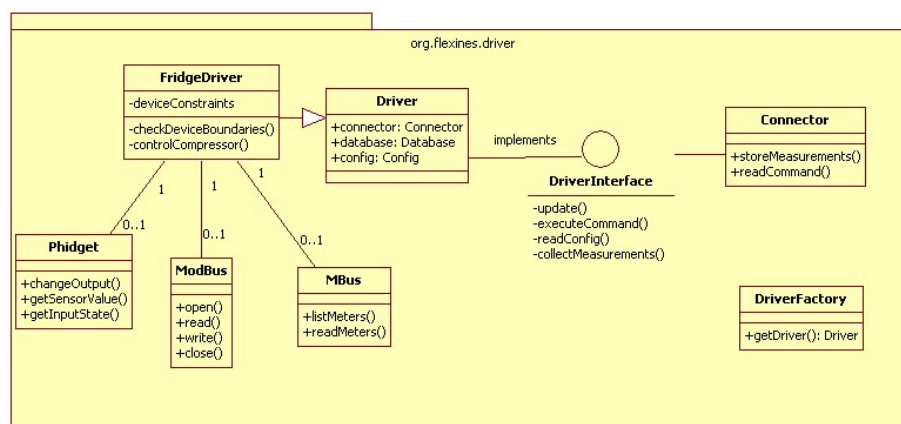
Informatie tussen de modules wordt uitgewisseld met behulp van vooraf gespecificeerde berichten. Voorbeelden hiervan zijn de 'regelruimte' - en de 'allocatie' berichten. Deze berichten worden via een bus structuur uitgewisseld. De bus structuur zorgt voor een robuuste communicatie tussen de modules.

Detailontwerpen modules

In deze sectie wordt verder ingezoomd op de individuele modules. Merk hierbij op dat de naamgeving op sommige plaatsen is aangepast ten opzichte van het overzicht plaatje.

Device driver

De device driver (device interface in het overzichtsplaatje) is verantwoordelijk voor de koppeling met een fysiek apparaat. Hieronder wordt in Figuur 3 verder ingezoomd op de device driver laag.

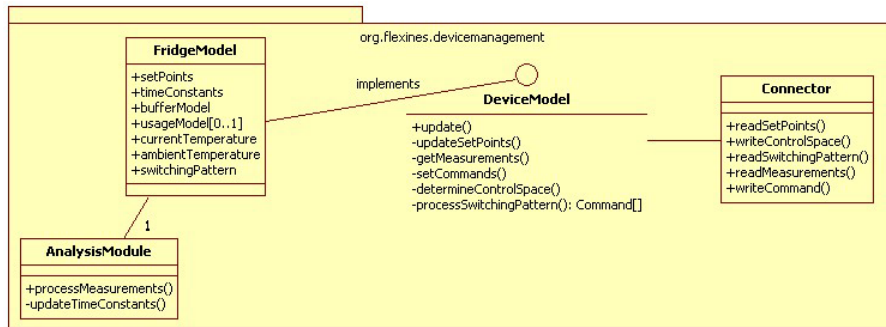


Figuur 3: Device driver laag

Elke driver moet voldoen aan de `DriverInterface`. Deze interface kent een aantal methoden om te communiceren met de device management laag. Voor elk apparaat dient een specifieke driver geschreven te worden, deze neemt eigenschappen over van de `Driver` parent class. Als voorbeeld wordt in de figuur een driver voor een koelkast getoond: de `FridgeDriver`. Deze kent specifieke methoden die alleen van toepassing zijn op koelkasten, zoals de "controlCompressor" methode. Deze `FridgeDriver` kan van meerdere protocollen gebruik maken om met de fysieke koelkast te communiceren. Deze worden gerepresenteerd door de `Phidget`, `ModBus` en `MBus` klassen.

Device Management

Deze laag gebruikt de informatie van de Device Driver laag om vast te stellen wat de energetische flexibiliteit is van een apparaat. Figuur 4 laat zien hoe de Device Management laag in hoofdlijnen is opgebouwd.

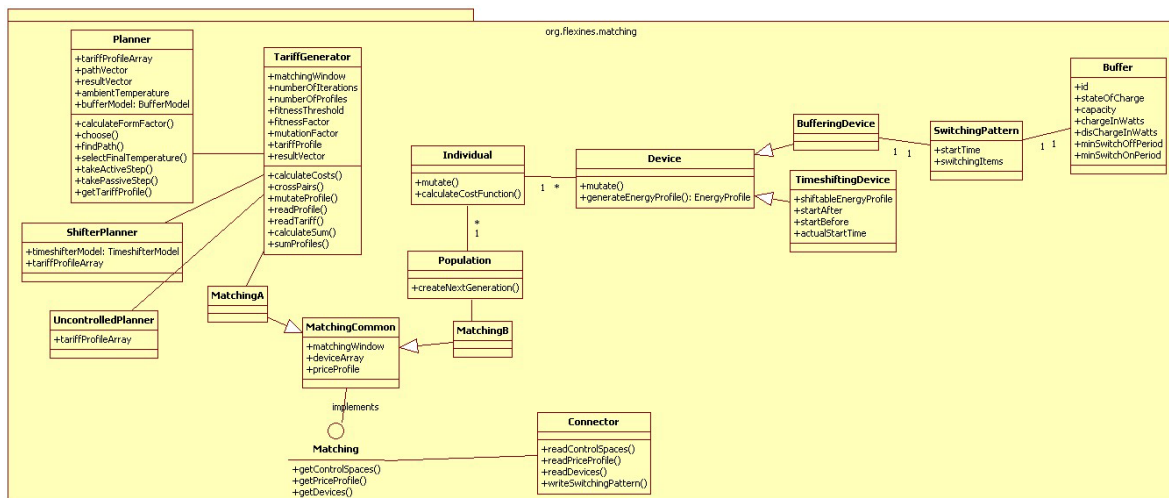


Figuur 4: Device Management laag

De energetische flexibiliteit wordt uitgedrukt in een ControlSpace, deze wordt doorgegeven naar de Matching laag. Om deze ControlSpace te kunnen bepalen wordt er gebruik gemaakt van een DeviceModel. Dit model is gekoppeld aan een specifiek apparaat; in de figuur is als voorbeeld de FridgeModel klasse opgenomen.

Matching

De functie van de Matching laag is om de opwekkers en verbruikers in een huishouden zo optimaal mogelijk op elkaar af te stemmen. Figuur 5 geeft een overzicht van de belangrijkste klassen.



Figuur 5: Matching laag

Er is één Matching interface. Via deze interface komen de ControlSpaces binnen, maar ook prijsprofielen (PriceProfile). Er wordt zodanig met de ControlSpaces (die de energetische flexibiliteit van de apparaten beschrijven) geschoven dat de kosten (gebaseerd op het prijsprofiel) zo laag mogelijk worden. Binnen het Flexines project zijn twee verschillende matching mechanismen ontwikkeld: een decentraal – (MatchingA) en een centraal mechanisme (MatchingB).

De decentrale matching werkt met een intern prijsprofiel dat gegenereerd wordt met de TariffGenerator klasse. Vervolgens wordt er per apparaat een planning gemaakt met behulp van de volgende klassen: Planner, ShifterPlanner en UncontrolledPlanner.

De centrale matching werkt volgens een genetisch algoritme. Voor elk apparaat (BufferingDevice en TimeshiftingDevice klassen) worden energie profielen gegenereerd. Van elk profiel worden de kosten bepaald. De beste oplossingen worden geselecteerd en worden verder ontwikkeld.

Uiteindelijk geldt voor beide matching mechanismen dat ze tot een oplossing komen waar voor elk apparaat het beste energieprofiel is vastgesteld. Dit profiel wordt vervolgens doorgestuurd naar de device management laag.

Market

De market laag (trading laag in het overzichtsplaatje) is verantwoordelijk voor de communicatie met de energie markten. De concepten over het handelen op energie markten zijn binnen het Flexines project niet uitgebreid uitgewerkt. Hiervan is dan ook geen detail ontwerp beschikbaar. De enige functie die de market laag uitvoert is het doorgeven van een prijsprofiel naar de matching laag.

Toekomstig werk

Een deel van het architectuur werk van Flexines wordt binnen TNO verder ontwikkeld. Het gaat hierbij met name om de aansturing van de apparatuur (met timeshifters, buffers, storage en uncontrolled load/generation) zoals dat in de Device Management module plaatsvindt. Deze aansturing is opgenomen in het Open Energy Management Framework (OpenEMF). OpenEMF wordt een operationeel platform (in eerste instantie voor eindgebruikers) waar verschillende demand side management aanpakken (zoals PowerMatcher, BEMI, etc.) op moet kunnen draaien.

In de huidige situatie is er voor elke aanpak een apart kastje dat bij de eindgebruiker geïnstalleerd moet worden. Hierdoor wordt de eindgebruiker belemmerd bij het switchen tussen aanbieders. Met het OpenEMF platform moet dat tot het verleden gaan behoren.