

# Eindrapportage MOBEG 'Monitoring en beheer van gebouwen voor een Smart Grid'

---



Monitoring en beheer van gebouwen voor een Smart Grid (MOBEG) is een project in het kader van de TKI SWITCH2SmartGrids gesubsidieerd door RVO. Het is bekend onder nummer TESH113018. De projectperiode van het project is 1 januari 2014 – 1 maart 2016.

#### Betrokken partners en geïnformeerde personen

Partner	Betrokkenen
IPSUM Holding BV (penvoerder)	Peter de Bie, Paul Casteleijns, Vincent Bakker, Albert Molderink
Universiteit Twente	Gerard Smit, Gerwin Hoogsteen, Jonathan Cruz, Gerbrand Bosch, Albert Molderink
Derden	Bijdrage
Bas Energy	Test locaties leveren
Salland Electronics	Ondersteuning ontwikkeling + productie prototype hardware
Centric	Installatie hardware
Cegeka Benelux	Backend
Purive	Project ondersteuning
SubVice	Project ondersteuning

#### Versie informatie

Versie	0.1
Status	Work in progress
Datum	Mei 2016
Opdrachtgever	Consortium Ipsum/UT
Referentie	TESG113018
Auteur	Albert Molderink
Redactie	Vincent Bakker
Verspreidlijst	RVO; Projectleiders partners

Het project is uitgevoerd met subsidie van het Ministerie van Economische Zaken, Subsidieregeling energy en innovatie (SEI), Topsector Energie uitgevoerd door Rijksdienst voor Ondernemend Nederland.

## Inhoudsopgave

Inhoudsopgave.....	3
1. Inleiding en doelstelling.....	4
2. Uitvoering project en resultaten.....	7
2.1 Ontwikkeling aan algoritme en backend.....	7
2.2 Ontwikkeling hardware CPB.....	12
2.3 Prototyping, testen en evalueren.....	16
2.4 Conclusie .....	18
3. Discussie.....	20
4. Conclusie en aanbevelingen .....	21
5. References .....	22
Appendix A: protocol omschrijving .....	23

## 1. Inleiding en doelstelling

In de huidige energiemarkt (in Nederland, maar dit geldt onverminderd in vele landen in de wereld) bestaat een toenemende vraag naar mogelijkheden om een beter beeld te krijgen van het energieverbruik vanuit professionele gebruikers, zoals winkels, kantoren en fabrieken. Deze vraag wordt gestuwd door de wens om het eigen energieverbruik slimmer in te richten en te verminderen. Op korte termijn bestaat de wens om het energieverbruik te reduceren voornamelijk vanuit een bepaalde duurzaamheidsambitie of kostenoverwegingen. Op langere termijn zal een gedetailleerde bepaling van het energieverbruik ook noodzakelijk zijn om de vraag naar energie in balans te brengen met het aanbod. Het is de verwachting dat in de toekomst een grotere rol wordt toegekend aan netbalancing door het beïnvloeden van het momentane energiegebruik als onderdeel van het SmartGrids concept.

Het rationaliseren van het energieverbruik kan gebeuren door apparaten en installaties automatisch te schakelen óf, indien dit niet mogelijk of wenselijk is, door de gebruikers van die apparaten en installaties bewust te maken van de gevolgen van hun handelen. In de praktijk blijkt echter dat met de bestaande technische hulpmiddelen die inzicht geven, zoals slimme meter, minder effectief met energie wordt omgegaan dan was verwacht. Het knelpunt blijkt veelal te liggen bij het feit dat het beschikbaar maken van meer informatie over het globale energieverbruik, zoals bijvoorbeeld elektriciteitsverbruik in een object, niet direct leidt tot daadwerkelijke bewustwording, laat staan een gedragsverandering. Vaak heeft ook alleen de facility manager toegang tot deze informatie.

Onderzoek heeft aangetoond dat inzicht gedetailleerd en inzichtelijk moet zijn, bij voorkeur op apparaat niveau. Daarmee wordt inzicht verkregen (of gegeven) over waar besparingspotentie zit en hoeveel het is. Daarnaast heeft onderzoek naar Demand Side Management (DSM), het afstemmen van de vraag op marktprijzen of aanbod ofwel Smart Grid, uitgewezen dat dit (gedeeltelijk) autonoom moet werken. Gebruikers willen niet constant bezig zijn met hun energieverbruik en aan/uitschakelen van apparaten. Wel willen ze een gedeelte van de controle houden, bijvoorbeeld door een periode aan te kunnen geven waarin een apparaat aan moet of uit mag.

Samengevat is er behoefte aan een betaalbare oplossing voor het monitoren van het energieverbruik op apparaat niveau en deze gegevens ófwel te gebruiken om automatisch apparaten aan- en uit te schakelen, ófwel het bewustzijn van besparingsmogelijkheden te vergroten door een directe terugkoppeling van informatie naar de gebruiker. Dit leidt op korte termijn tot energiebesparing, en biedt op langere termijn de mogelijkheid om elektriciteitsafname te sturen (Demand Side Management) als onderdeel van netbalancing.

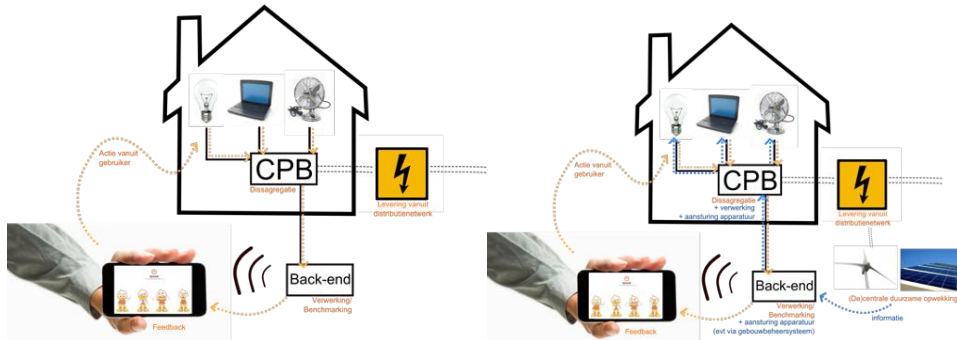


Figure 1 Huidige versie van de CPB (links) en de nieuwe in dit project te ontwikkelen CPB

Ipsum heeft een algoritme ontwikkeld dat op basis van metingen op één of enkele plaatsen in de meterkast inzicht kan geven in het verbruik op apparaat niveau. Het elektriciteitsverbruik wordt in de meterkast gemeten met behulp van de zogenoemde Coded Power Box (CPB) en doorgestuurd naar Ipsum. De CPB meet het stroomverbruik met behulp van meetspoelen om de bedrading. Momenteel is de CPB een of-the-shelf product ontwikkeld door Mpare (onderdeel van Alliander).

De Universiteit Twente heeft veel ervaring met DSM methoden en algoritmen. Het combineren van deze twee in een energie management systeem met de bijbehorende interfaces voorziet in de bovengenoemde behoefte.

In dit project worden deze twee expertises gecombineerd in een nieuwe versie van de CPB en wordt de backend gedeeltelijk opnieuw ontworpen en geïmplementeerd om tweeweg communicatie mogelijk maken. Deze nieuwe oplossing moet het geven van inzicht gemakkelijker/duidelijk maken én moet het mogelijk maken om Demand Side Management toe te passen. Dit is gevisualiseerd in Figure 1.

### Doelstelling: een sterk verbeterd Energie Management Systeem (EMS)

*IPSUM en Universiteit Twente stellen zich in dit project ten doel een EMS te ontwikkelen dat op basis van dezelfde analysetechnologie zoals hierboven beschreven is ook automatisch individuele apparaten kan schakelen. Dit doen zij door de mogelijkheid aan de CPB toe te voegen om apparatuur uit te schakelen ter besparing van energie, maar ook om automatische Demand Side Management mogelijk te maken. De CPB verandert in dit project van een eenrichtings-communicatiesysteem naar een tweerichtingscommunicatiesysteem. Tevens wordt andere functionaliteit van de CPB uitgebreid zodat minder communicatie met het back-end noodzakelijk is. Dit project moet leiden tot een EMS waarmee met beperkte kosten voor de gebruiker een groot resultaat te behalen is.*

Het doel van dit project is dus de bestaande CPB (schematisch weergegeven in Figure 2) van een simpel meetinstrument dat de metingen doorstuurt naar een backend te transformeren naar een Energie Management Systeem dat naast meten ook kan schakelen.



Figure 2 Schematische weergave datastromen huidige CPB

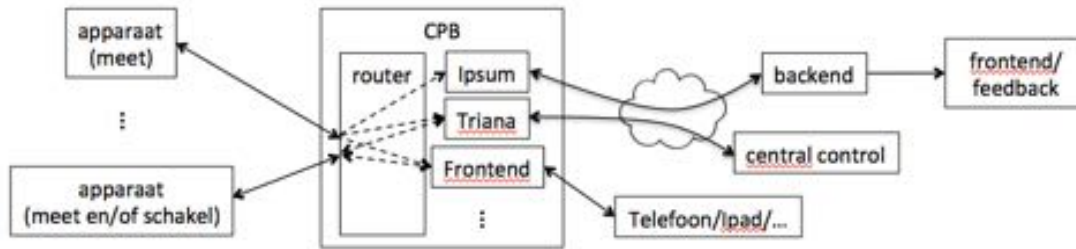


Figure 3 Schematische weergave datastromen te ontwikkelen EMS/CPB

Hiervoor moet de mogelijkheid toegevoegd worden om te communiceren met apparaten (via “pluggen” of direct met “Smart Appliances”). Tweerichtingscommunicatie is nodig om de analyses van Ipsum terug te kunnen sturen naar het EMS zodat er lokaal iets mee gedaan kan worden en doorgegeven kan worden naar de DSM applicaties. Daarnaast kunnen er eventueel direct stuursignalen gegeven worden. Verder moet er een interface komen met (bestaande) Demand Side Management methoden. Het EMS wordt een spil in het verbinden van apparaten, data stromen en services.

De werkzaamheden in dit project, buiten projectmanagement, zijn opgesplitst in drie WPs. Hieronder is per WP weergegeven wat de doelen zijn.

1. Ontwikkeling aan algoritme en backend  
 Het algoritme moet aangepast worden zodat de snelheid wordt verhoogd, voornamelijk de reactiesnelheid op wijzigingen om snelle feedback te kunnen geven. Hiervoor moet gekeken worden naar mogelijkheden een gedeelte van het algoritme op de CPB uit te voeren. Daarnaast moet de backend tweerichtingscommunicatie ondersteunen en ondersteuning bieden voor feedback aan de gebruiker (meer dan alleen verbruiksdata). Bij dit alles moet de schaalbaarheid voor grote aantallen gebruikers in de gaten gehouden worden.
2. Ontwikkeling hardware CPB  
 De huidige CPB moet aangepast worden om naast meten en versturen van data, ook verwerken van data en het schakelen van apparaten te ondersteunen. Dit moet zowel in hardware als in software ondersteund worden.
3. Prototyping, testen en evalueren  
 Bovengenoemde wijzigingen moeten getest en geëvalueerd worden. Hiervoor moet er een prototype gebouwd worden, maar moet er ook met (toekomstige) gebruikers afgestemd worden wat er verwacht wordt en in de praktijk getest worden of het werkt en wat er bij installatie en in praktijk mis kan gaan.

## 2. Uitvoering project en resultaten

In dit hoofdstuk wordt de uitvoering van het project beschreven en de resultaten benoemd. Hierbij komen de werkwijze, de problemen met oplossingen en wijzigingen ten opzichte van het projectplan aan bod. Dit wordt gedaan door in eerste instantie per workpackage de uitvoering en resultaten te bespreken, gevolgd door een samenvatting waarin de samenhang en uiteindelijke resultaten en lessons-learned worden besproken.

### 2.1 Ontwikkeling aan algoritme en backend

Doel van dit workpackage is om de feedback van het algoritme te versnellen en verbeteren, de backend schaalbaar op te zetten en tweerichtingscommunicatie te ondersteunen.

In de beschrijving van dit workpackage is onderscheid gemaakt tussen algoritme en backend; het algoritme is een los onderdeel van de software die op de backend draait. Dit is onderdeel van het trade-secret van Ipsum en wordt daarom niet gedeeld met de andere partijen. Dit gedeelte van de software doet alleen de apparaat herkenning (desaggregatie) en wordt als bibliotheek ingeladen in de backend software. Deze software verzorgt alle andere taken: het ontvangen van de ruwe meet-data, het opslaan van de gedesaggregeerde data (inclusief totalen, verschillen, besparingen, groepstotalen, enz.), analyse en weergave van de data.

Hieronder worden eerste de wijzigingen aan het algoritme besproken en vervolgens de wijzigingen aan de backend (zowel infrastructuur als software). De wijzigingen en verbeteringen zijn iteratief doorgevoerd; wijzigingen zijn getest op het prototype en in een aantal veldtesten (zie WP3) en op basis van observaties en feedback is een volgende iteratie uitgevoerd. Er zijn vele kleine iteraties doorgevoerd (voornamelijk op het prototype) die niet allemaal afzonderlijk besproken zijn, de grote iteraties (veldtesten) worden besproken in sectie 2.3.

Dit workpackage is grotendeels volgens projectplan verlopen, de wijzigingen worden in de gedeelten hieronder benoemd.

#### 2.1.1 Algoritme

De kern van de oplossing die Ipsum aanbied en een van de twee uitgangspunten voor dit project is het desaggregatie algoritme van Ipsum. Dit algoritme kan op basis van metingen in de meterkast van verschillende apparaten achterhalen wanneer ze aan hebben gestaan en hoeveel ze verbruikt hebben. In dit WP is in eerste instantie het algoritme expliciet opgesplitst in verschillende stappen (zie Figure 4). Doel hiervan is om het mogelijk te maken om een gedeelte van het algoritme op de CPB te kunnen draaien. Daarnaast wordt het op deze manier een streaming applicatie die gemakkelijker te schalen is en wat bovendien een “good-practice” is in big-data applicaties. Dit, naast verbeteringen in de implementatie, heeft de snelheid significant verhoogd (zie volgende sectie). Ook zijn de resultaten verbeterd; op basis van de veldtest in Lochem (zie WP 2.3) zijn tekortkomingen naar boven gekomen. In de laatste veldtests is gebleken dat de resultaten sterk verbeterd zijn en goed genoeg voor een eerste versie (zie Figure 5).



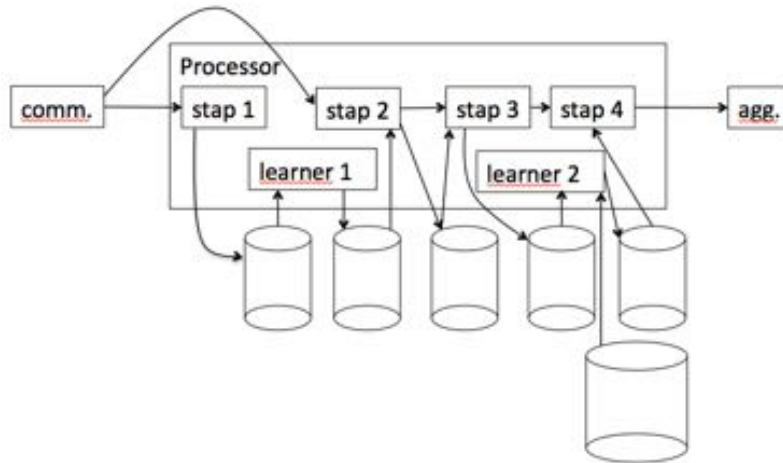


Figure 4 De verschillende stappen in het algoritme

Als eerste stap in het uitvoeren van een gedeelte van het algoritme op de CPB is een dynamische buffering op de CPB geïmplementeerd. Deze kan een willekeurig aantal metingen bufferen; op deze manier kan bijvoorbeeld alleen bij wijzigingen de data opgestuurd worden of wanneer de gebruiker feedback verwacht. Hiermee is een gedeelte van de intelligentie naar de CPB verplaatst. Een wijziging ten opzichte van het projectplan is het toevoegen van ondersteuning voor slimme meter data. In eerste instantie was de scope gericht op bedrijven, hiermee is ook ondersteuning voor consumenten toegevoegd. Bij het gebruik van slimme meter data vervalt de noodzaak voor het aanschaffen en installeren van (dure) meetapparatuur. De slimme meter beschikt over een P1-poort die redelijk simpel uitgelezen kan worden met behulp van een aangepast RS-232 protocol.

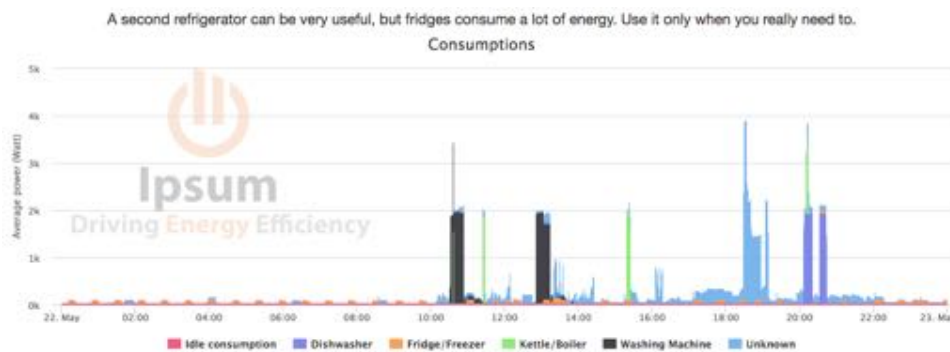


Figure 5 Verbeterde resultaten algoritme



Figure 6 Ook ondersteuning voor de slimme meter is toegevoegd om aanschaf en installatie van dure meetapparatuur te voorkomen.



### 2.2.2 Backend

De backend is structureel aangepast, zowel wat de software betreft als de infrastructuur. Het doel hierbij is schaalbaarheid verhogen het toevoegen van mogelijkheden tot feedback (zoals gewenst gebleken is) en tweerichtingscommunicatie. Hieronder worden de aanpassingen aan zowel software als infrastructuur beschreven.

#### Software

De applicatie is opgesplitst in meerdere kleinere applicaties om de schaalbaarheid te vergroten; op deze manier kan er gericht capaciteit toegevoegd worden. Wanneer de bottleneck ligt bij het ontvangen van de data, wordt er niet een applicatie toegevoegd die alle taken omvat, maar een applicatie specifiek gericht op het ontvangen van data. Dit vergroot de schaalbaarheid en flexibiliteit. De verschillende taken communiceren met elkaar via databases en Hazelcast (zie verderop voor meer details). Daarnaast zijn de protocollen voor het ontvangen van data als voor het opvragen van de resultaten omgezet naar standaard protocollen. In Figure 7 staat een schematisch overzicht van de applicaties en de protocollen. De comm(unicator) verzorgt het ontvangen van de data. Dit gebeurt via google's protobuf berichten over een HTTPS of WSS verbinding. De ruwe gegevens worden opgeslagen in de noSQL database. Vervolgens verzorgt de processor het verwerken van de gegevens, waarbij instellingen vanuit SQL geladen worden en toestandsgegevens opgeslagen. De resultaten worden doorgegeven aan de aggregator die de resultaten (en totalen, gemiddelden, verschillend, etc.) opslaan in noSQL. Via de API kunnen de resultaten opgevraagd worden via een RESTfull API bovenop HTTPS. De zogenoemde assets (klantgegevens, instellingen, etc.) kunnen worden beheerd via de assets applicatie via een RESTfull API bovenop HTTPS. De keuzes voor de database en opbouw worden verder verklaard in onderstaande alinea's. De stippellijn in het plaatje geeft de toevoeging van tweerichtingcommunicatie aan. Naast dat meet-data vanuit CPBs opgestuurd kan worden, kunnen er ook desaggregatie resultaten en stuursignalen teruggestuurd worden. In Appendix A is de specificatie van het protocol te vinden en in [1] is een uitgebreide onderbouwing van de keuze voor dit protocol te vinden.

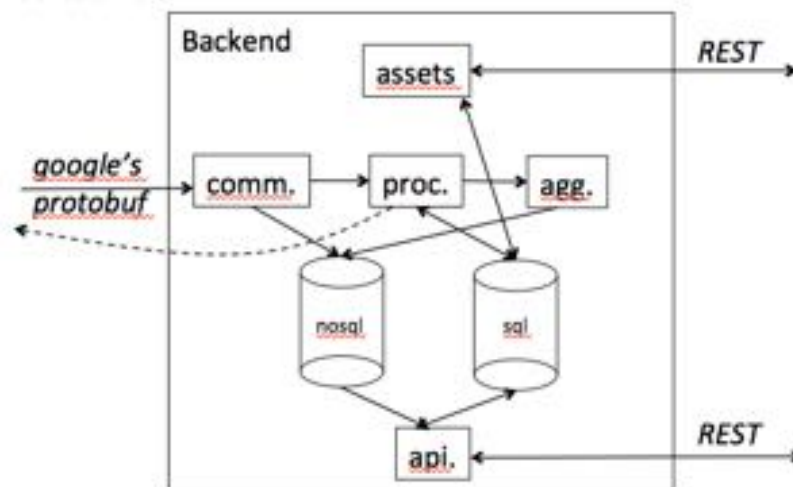


Figure 7 Schematisch overzicht van de applicaties en protocollen voor de backend

Een andere belangrijke uitbreiding van de backend is het geven van (snellere) feedback. Uit onderzoek is gebleken dat naast het geven gedetailleerd inzicht, ook het geven van een analyse en bij voorkeur een vergelijking met anderen belangrijk is [2]. Daarom is de backend uitgebreid met de mogelijkheid om gerichte, persoonlijke tips op te vragen. Hiervoor is een tip-engine gebouwd die op basis van Velocity [3] expliciete tips kan geven. Voorbeelden van een tips zijn: *“Uw idle-verbruik is 100W, dit betekent ongeveer 200 euro per jaar”* en *“Uw verbruik is 10% lager dan vorige week, maar gemiddeld is het verbruik 12% lager!”*.

De tip-engine bestaat uit een lijst met tips; deze tips kunnen gebruik maken van gegevens die uit het algoritme of de database komen. Wanneer de benodigde gegevens niet beschikbaar zijn, wordt automatisch een volgende tip gemaakt. Op deze manier krijgt een gebruiker altijd een persoonlijke op hem van toepassing zijnde tip. Eerste ervaringen met de tip-engine laten zien dat dit het inzicht van consumenten verhoogd; de grafiek en getallen worden in perspectief geplaatst. Op basis van deze ervaringen is geconcludeerd dat deze vorm van feedback zeer belangrijk is en wordt dit dan ook verder ontwikkeld na het project (zie hoofdstuk 4). Hierbij wordt er naast een tip-engine een behavior change module ontwikkeld (met uitgebreidere analyses en vergelijking met historische data en data van anderen) en staat ook het toevoegen van push berichten (mensen expliciet betrekken) op de roadmap. Dit is mede gebaseerd op de theorie in [4].

### **Infrastructuur**

Om de schaalbaarheid van de oplossing te garanderen is in samenwerking met Cegeka een nieuw ontwerp gemaakt van de infrastructuur. Dit ontwerp bestaat uit specifieke nodes voor specifieke functionaliteit, toegespitst op de opsplitsing van de software in verschillende applicaties. Op dit moment zijn er drie typen nodes voorzien: applicatie-nodes, SQL-nodes en noSQL-nodes. Op de applicatie-nodes draaien op dit moment alle applicaties, dit kan later naar behoefte verder uitgesplitst worden. Daarnaast is er gekozen voor twee typen databases: SQL en noSQL. De SQL is een relationele database wat het beheren van assets gemakkelijker en sneller maakt. Dit is een relatief kleine database (weinig data). Voor het opslaan van de energie-data (ruwe data en gedesaggregeerde data) is gekozen voor een noSQL oplossing. Een noSQL database is geen relationele database wat het combineren van data lastiger (of zelfs onmogelijk) maakt, maar het kan wel veel meer data opslaan en is vele malen sneller. Het idee is om de data net zo op te slaan als dat het wordt opgevraagd. Dit is een good-practice in big-data applicaties (zie volgende sectie). Zowel de SQL als de noSQL nodes zijn clusters; ze werken samen bij het opslaan en opleveren van data. De data wordt op meerdere nodes opgeslagen zodat het uitvallen van één node niet tot dataverlies lijdt. Ander voordeel van het scheiden van klantgegevens en data is een beter beveiliging en waarborging van de privacy.

De applicatie nodes zijn, zoals eerder besproken, streaming nodes. De data “stroomt” door de verschillende applicaties heen die ieder een eigen operatie op de data doen en het vervolgens doorgeven. Op dit moment wordt Hazelcast [5] gebruikt voor het bijhouden van het cluster en het distribueren van de taken. Ondertussen zijn de tekortkomingen hiervan naar boven gekomen (voornamelijk stabiliteit en niet in eerste instantie ontwikkeld voor streaming applicaties). In eerste instantie lijkt Apache Storm [6] een betere optie voor dit soort applicaties.

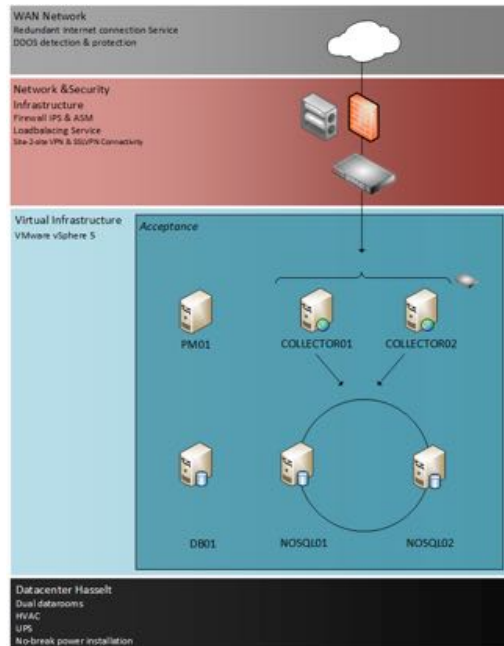


Figure 8 Infrastructuur van de backend

Een groot verschil tussen conventionele nodes en de nodes die hierboven beschreven zijn is dat dit disposable nodes zijn; geen enkele node heeft uniek informatie en wanneer een node wordt bijgeschakeld wordt deze automatisch in het cluster opgenomen. Dit betekent dat een slecht functionerende node niet gerepareerd hoeft te worden, maar deze simpelweg verwijderd kan worden en een nieuwe toegevoegd. Dit is een andere manier van omgaan met (virtuele) nodes. Dit vereist een andere manier van werken bij de ICT-infrastructuur leveranciers, iets wat op dit moment nog vaak verschil van inzicht oplevert.

Op basis van deze veranderingen zijn er een aantal snelheidstesten uitgevoerd. De resultaten hiervan zijn te zien in Figure 9. Uit deze testen blijkt dat er met de huidige infrastructuur 5000 tot 10000 gebruikers per server verzorgd kunnen worden. Aangezien er vooral in het algoritme nog ruimte is voor optimalisatie, is dit een voldoende snelheidswinst.

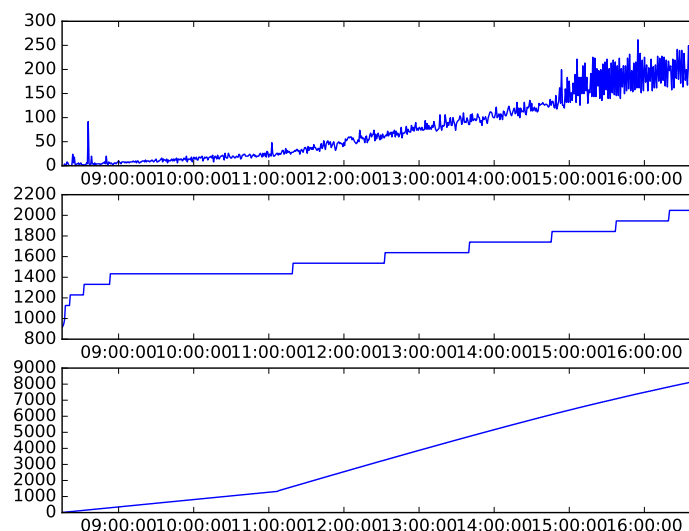


Figure 9 Processor load uitgezet tegen aantal verbindingen

## Big data

De ontwikkelde oplossing zal, wanneer er veel klanten zijn, veel (energie)data ontvangen. Deze data is, per klant, op zich al zeer interessant voor klanten omdat het veel inzicht geeft in energieverbruik en omdat deze data gebruikt kan worden voor DSM toepassingen. Maar wanneer deze informatie wordt gecombineerd ontstaat er nog veel waardevollere informatie. Deze informatie kan gebruikt worden om betere feedback te geven aan consumenten zoals feedback in vergelijking met anderen, aangeven of apparaten veel/weinig verbruiken en competitie. Daarnaast kan deze data gebruikt worden voor DSM toepassingen, om voorspellingen te maken en om resultaten van bepaalde beslissingen te evalueren (lerende algoritmen). Ook voor beslissingen over aanpassingen/uitbreidingen van het elektriciteitsnetwerk en elektriciteitsvoorziening kan deze informatie gebruikt worden.

Bij grote hoeveelheden data is het belangrijk dat de operaties per meting eenvoudig is, bij voorkeur simpele optel en aftrek operaties. Daarom is er een boomstructuur ontwikkeld voor de aggregators die de data interpreteren en direct op de juiste manier opslaan in de noSQL database en daarbij de mogelijkheden van noSQL benut. Deze structuur is redelijk a-specifiek, het zou voor veel meer soorten data gebruikt kunnen worden.

## 2.2 Ontwikkeling hardware CPB

Zoals eerder genoemd werd bij het begin van het project een oplossing van MPARE in combinatie met de door hen ontwikkelde ENDAX software voor het versturen van data [7]. Binnen dit framework is tweerichtingcommunicatie niet mogelijk en is het installeren en configureren van de hardware en verbinding vrij lastig. Dit is de reden om zelf een CPB te ontwikkelen. Wel is het idee achter de ENDAX, de gebruiker die ten allen tijde controle heeft over wie z'n data krijgt, erg interessant. Vooral voor de acceptatie van nieuwe technologieën, zeker bij privacygevoelige technologieën als deze, kan dit erg helpen. Daarom is er, na oplevering van dit project, ook weer meer contact met MPARE om te bepalen in hoeverre de tekortkomingen die tijdens dit project naar boven zijn gekomen meegenomen kunnen worden in hun opvolger van de ENDAX, HelloData [8]. Op deze manier zou de technologie voor meer services beschikbaar komen. Iets wat precies het idee is (zie verderop onder software CPB).

Dit workpackage bestaat, ondanks de titel "hardware CPB" uit twee componenten, de hardware en de software. Immers, bij nieuwe hardware en nieuwe eisen aan de CPB moet ook de software aangepast worden. Hieronder wordt eerste de ontwikkeling aan de hardware beschreven, daarna de ontwikkeling aan de software.

### Hardware CPB

Het ontwikkelen van de nieuwe hardware is begonnen met het opstellen van de functionele eisen. Deze functionele eisen zijn in samenwerking met Salland opgesteld. Het volledige document is op te vragen [11]. De eisen zijn:

- de kwaliteit van de metingen (resolutie, foutmarge, etc.),
- het synchroniseren van de metingen,
- het verwerken en opsturen van de metingen,
- de beschikbare rekenkracht en geheugen voor extra taken.



Figure 10 Prototype ontwikkelt door Salland

Op basis van deze eisen is een prototype gebouwd door Salland. Een foto van het prototype is te zien in Figure 10. Het is gebleken dat het opstellen van eisen hierbij niet zo gemakkelijk is, deze zijn vaak nog op meerdere manieren te interpreteren. Het resultaat is een prototype dat functioneel werkt, maar waar nog een verbeteringsslag aan gedaan moet worden om aan de eisen van de installateurs (veiligheid) en verzekeringsmaatschappijen te voldoen (ISO standaarden).

Een belangrijk onderdeel van de CPB is de rekeneenheid, de embedded processor of microcontroller. Deze moet naast communicatie en buffering, ook voldoende rekenkracht en geheugen hebben om andere functionaliteiten te kunnen ondersteunen. Daarom is hiervoor onderzocht welke mogelijkheden er zijn. Ook hier zijn weer eerst de eisen opgesteld en vervolgens een keuze gemaakt. De eisen hierbij zijn functionele, maar ook eisen met betrekking op leveringszekerheid, veiligheid, ISO normen en omstandigheden (o.a. temperatuurbereik). Een volledige omschrijving van deze eisen is te vinden in [9]. De conclusie is dat een Caramboia 2 of CM-T335 de beste keuze is omdat ze gebruik maken van een veelgebruikte, goedkope processor, ze veel interfaces hebben en het industriële producten met leverzekerheid zijn.

### Software CPB

Voor dit project zijn er naast eisen aan de hardware, ook (meer) eisen aan de software. Naast het meten en versturen van metingen, moet er ook intelligentie op de CPB komen. Daarnaast moet tweerichtingscommunicatie ondersteund worden en moeten apparaten aangestuurd kunnen worden. Als laatste moet er ondersteuning voor DSM toegevoegd worden.

Om al deze eisen te ondersteunen is een soort van “data-router” ontwikkeld voor op de CPB, de Universal Smart House Controller (USHC). Dit is een interface laag tussen de apparaten aan de onderkant en services aan de bovenkant. Op deze manier kan er op een generieke manier gecommuniceerd worden met apparaten, er kan voor ieder apparaat een driver gemaakt worden die deze interface implementeert. Aan de bovenkant zijn er services die data ontvangen van apparaten en stuursignalen sturen naar de apparaten. Op deze manier is het mogelijk om meerdere services gebruik te laten maken van dezelfde meet-data.





Figure 11 Ontwerp van de Universal Home Controller

Tot nu toe waren de meeste oplossingen zo gebouwd dat de meet hardware specifiek data opstuurde naar 1 gebruiker. Dit past ook goed in de filosofie van MPARE en de ENDAX/HelloData (zie boven). Een schematische weergave van deze USHC is te vinden in

Figure 11, meer informatie over eisen en implementatie in [10].

Om deze implementatie te testen en om een volledig prototype te kunnen bouwen is er een prototype gebouwd (zie Figure 14) waarbij een slimme meter en stuurbare apparaten van Eaton gebruikt zijn als slimme apparaten. Voor deze apparaten is een driver gebouwd. Vervolgens is er een front-end gebouwd als service, zie Figure 12. Deze service geeft weer hoeveel energie er is gebruikt en geeft de mogelijkheid om apparaten de schakelen.

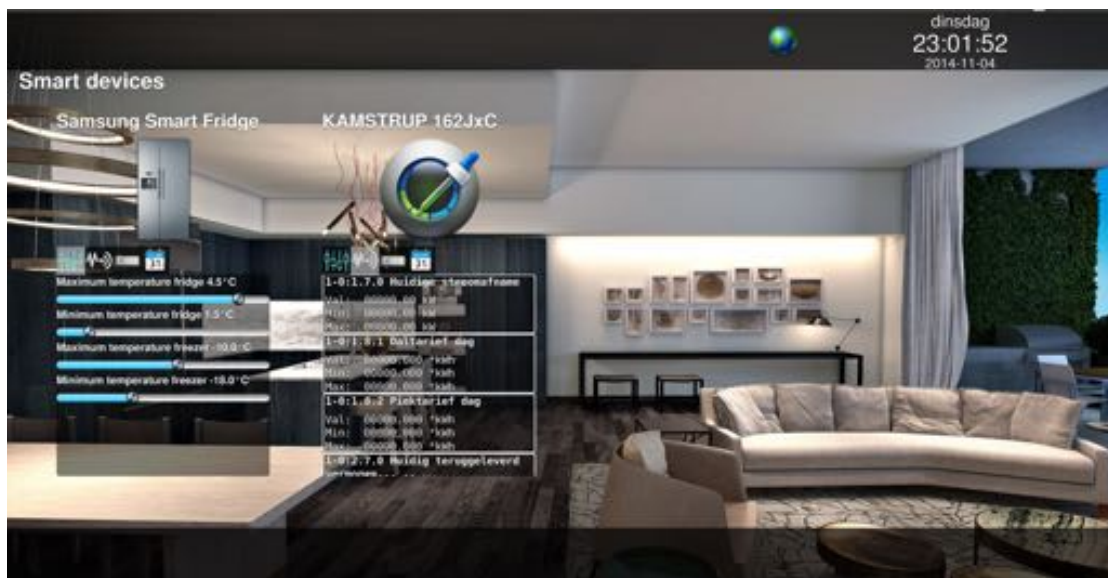


Figure 12 Frontend op de USHC

Deze front-end is ook gebouwd als smartphone app zodat de meet-data ingezien kan worden en apparaten gestuurd vanaf een smartphone.

Daarnaast is ook aan de CPB kant de tweerichtingcommunicatie geïmplementeerd. Er is een Ipsum service geïmplementeerd die geabonneerd is op de data van de P1 poort. Deze data wordt doorgestuurd via het opgestelde protocol en stuursignalen en gedesaggregeerde data worden ontvangen. De stuurdata wordt gebruikt om de apparaten aan de sturen (via USHC). De gedesaggregeerde data wordt aangeleverd voor andere services; dit betekent dat de Ipsum service zich ook bij USHC aanmeldt als (virtual) apparaat. Op deze manier kunnen andere services gebruik maken van de data.

Daarnaast is er FPAI service geïmplementeerd. FPAI is op een framework dat een generieke interface biedt voor DSM methodes. Er zijn momenteel al verschillende DSM methodes die met FPAI werken, bijvoorbeeld de veelgebruikte PowerMatcher [13]. Echter, op het moment zijn er nog weinig apparaten aangesloten op FPAI, er zijn voornamelijk virtuele apparaten geïmplementeerd. Door het koppelen van FPAI aan USHC kan FPAI gebruik maken van de apparaten die aan USHC gekoppeld zijn. Dit is schematisch weergegeven in **Error! Reference source not found.**Figure 13. Ook Triana, de DSM optimalisatie methode ontwikkeld aan de Universiteit Twente, kan gebruik maken van Triana. Dus met het koppelen van FPAI aan USHC is de DSM methode van de Universiteit Twente gekoppeld, maar daarnaast ook andere DSM methodes en is er gebruik gemaakt van een universele interface.

Meer informatie en details over de tweerichtingscommunicatie en de integratie met FPAI is terug te vinden in [1].

Als laatste is er een installatie module gemaakt die het installeren en configureren van de CPB eenvoudiger maakt. Deze installatie module bestaat uit een stappenplan waarin de (draadloze) verbinding geconfigureerd wordt, de verschillende meetpunten van een correcte configuratie een naamgeving worden voorzien en de CPB aan de juiste gebruiker gekoppeld wordt. In [9] zijn alle detail te vinden.



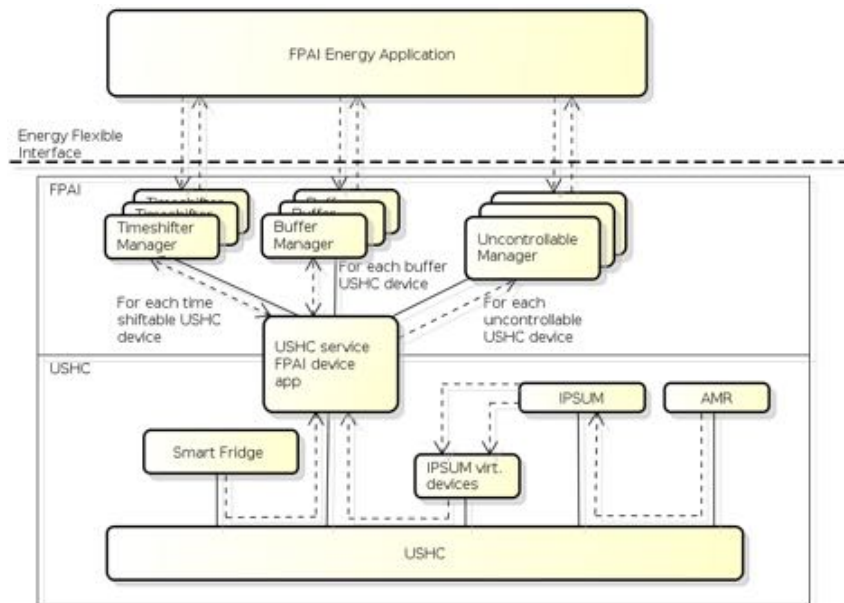


Figure 13 Integratie van FPAI in USHC

### 2.3 Prototyping, testen en evalueren

Op basis van de bovenstaande resultaten is een prototype gebouwd. Een foto van dit prototype is te vinden in Figure 14. Het prototype bestaat uit een mini pc waar de cpb software op geïnstalleerd is, inclusief USHC en koppelingen. De CPB leest de slimme meter uit als meet-data en er is een integratie gemaakt met Eaton X10 sensoren en actuatoren. Daarnaast is een integratie met Plugwise meet/stuur pluggen gemaakt. Met behulp van de eerder beschreven front-end kan het energieverbruik uitgelezen worden en de apparaten aangestuurd. Ook de desaggregatie resultaten worden ontvangen en weergegeven. Een uitgebreide beschrijving van het prototype is te vinden in [10].

Naast de twee bovenstaande use-cases (inzicht/schakelen lokaal via front-end en integratie met resultaten uit de Ipsum-backend) is er ook een use-case uitgewerkt voor een DSM methode. Dit is een relatief simpele use-case om het principe aan te tonen. Als DSM methode is een Triana implementatie gemaakt gebaseerd op biedingsfuncties. Het idee is dat het energieverbruik wordt verminderd wanneer er een piekvraag is in het netwerk. Het apparaat dat hiervoor gekozen is, is de waterkoker. Wanneer de waterkoker aan staat en er een piekvraag in het netwerk is, gaat de waterkoker uit. Het koffiezetapparaat daarentegen wordt niet uitgezet. De informatie over welk apparaat aan staat komt vanuit Ipsum, dit wordt via het USHC en FPAI doorgegeven aan de Triana controller. Deze bepaalt vervolgens aan de hand van deze gegevens en (virtuele) gegevens uit het netwerk of de waterkoker uitgeschakeld gaat worden. Dit gebeurt vervolgens weer met behulp van het FPAI en USHC framework. Deze test is geslaagd en heeft aangetoond dat de hele keten werkt.

Daarnaast zijn er bij verschillende zakelijke en privé personen meetapparaten geïnstalleerd. Er is niet in de praktijk getest met schakelen, voornamelijk vanwege de impact dit dat kan hebben. Onder de proefpersonen in dit project ware er zijn weinig tot geen mensen bereid om hieraan mee te werken. Ondertussen is er in meerdere (TKI) projecten ervaring opgedaan met het schakelen van apparaten. Helaas is dat in dit project niet gelukt. Wel is er een

onderzoek gedaan onder mensen in Lochem (onder deelnemers In4Energy) waar de bereidwilligheid is onderzocht; hieruit bleek dat informatie verstrekking erg belangrijk is [14]. Verder wees het onderzoek uit sturen in verschillende stappen geïntroduceerd moet worden: eerst meten, dat tips geven, aangeven wat geschakeld zou worden, daadwerkelijk schakelen. Schakelen is daarom wel opgenomen in een vervolgproject (zie Hoofdstuk 4). Bij een aantal bedrijven (o.a. JayDee, ABN en de universiteit) zijn meetkastjes geïnstalleerd (zie **Error! Reference source not found.**Figure 15). Bij deze installaties is enerzijds kennis opgedaan

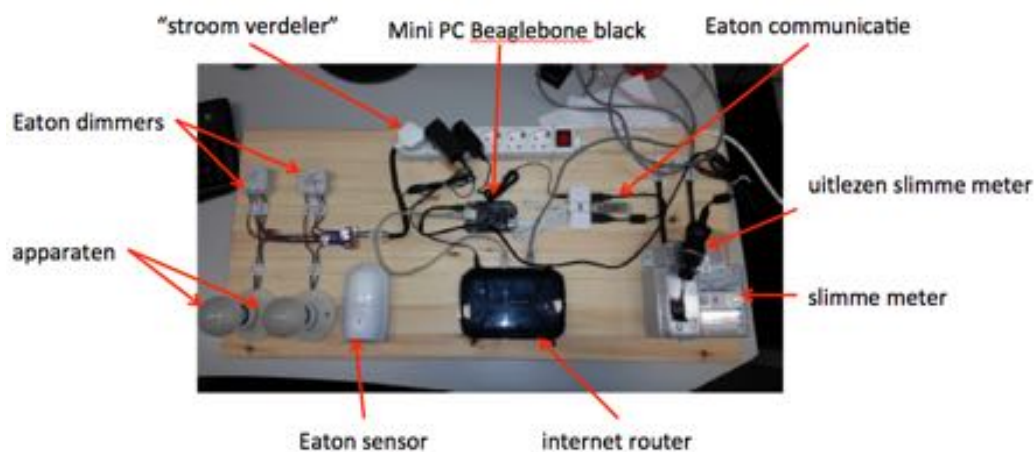


Figure 14 Mobeg prototype

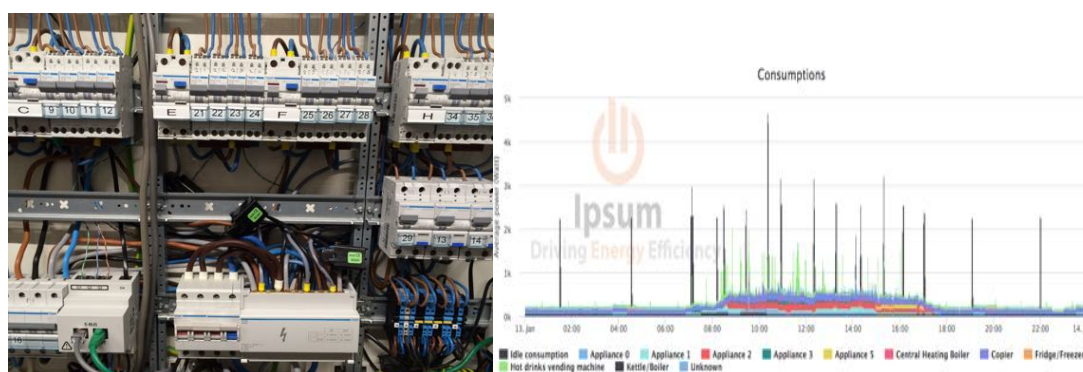


Figure 15 Bedrijfsinstallatie en bijbehorende resultaten

over het installeren van meetapparatuur (wat is daadwerkelijk nodig) en anderzijds met wat bedrijven willen zien voor informatie. Ook dit is input voor vervolgonderzoek en vervolgtrajecten (zie Hoofdstuk 4). De belangrijkste conclusies zijn dat er in de B2B wereld erg weinig mogelijk is, erg weinig inzicht is in energieverbruik en dat simpelweg gedetailleerd inzicht geven voor veel bedrijven al een zeer gewenste oplossing is. Wanneer hier nog analyses en bespaartips aan toegevoegd worden is dit een waardevolle oplossing die de acceptatie van Smart Grid toepassingen verhoogd. Op het gebied van installatie is naar boven gekomen dat er veel verschillende meterkasten zijn en de omstandigheden heel verschillend. Er moeten, voordat de installateur ter plaatse is, foto's gemaakt worden en/of de situatie ter plaatse moet opgenomen worden om de juiste configuratie van meetapparatuur te leveren en eventuele benodigde wijzigingen aan de installatie vooraf af te stemmen.

Daarnaast zijn er bij een groot aantal privé huizen ook meetkastjes opgehangen. In eerste instantie in Lochem (in samenwerking met het project) waar ook een gedeelte van de mensen met de Mobeg oplossing (zonder schakelen) kennis heeft gemaakt. Hier was het belangrijkste doel om te achterhalen wat mensen voor feedback willen hebben. Dit is gedaan met behulp van interviews en vragenlijsten [2]. Hieruit is naar voren gekomen dat gedetailleerd inzicht (op apparaat niveau) belangrijk is om de aandacht vast te houden. Wel is duidelijk geworden dat de resultaten die op dat moment gepresenteerd werden, niet goed genoeg waren. Zowel qua resultaten van de desaggregatie als voor de visualisatie. Daarnaast is er onderzoek gedaan naar welke apparaten mensen bereid zijn te laten schakelen [12]. Hieruit is naar boven gekomen dat mensen voornamelijk witgoed willen laten schakelen, terwijl verwarming (warmtepomp) en de EV op meer verzet stuit. Dit terwijl deze laatste juist meer effect hebben en met minder impact op comfort gebruikt kunnen worden. Hier is dus voorlichting zeer belangrijk.

Als laatste zijn, vlak voor het afsluiten van het project, 150 dongels uitgedeeld aan privé personen in het kader van een veldtest met ABN en met NLE. Een dongel is een zeer eenvoudig stukje hardware dat niets anders doet dan de slimme meter uitlezen en meet-data opstuurt naar de backend. De resultaten kunnen vervolgens bekeken worden op een website. Bij deze veldtest is gebruik gemaakt van de verbeterde versie van de software. Deze resultaten bleken een stuk beter en de reacties dan ook een stuk positiever (zie screenshot in Figure 5). Eerste resultaten van deze veldtest laten zien dat naast gedetailleerd inzicht, ook het helpen interpreteren van de data (analyses) en het geven van tips zeer belangrijk zijn voor het besparen van energie. Deze twee aspecten zijn, samen met het ontwikkelen van een smartphone app, onderdeel van en vervolgproject en onderzoek (zie Hoofdstuk 4).

## 2.4 Conclusie

In dit hoofdstuk zijn de resultaten van de verschillende workpackages gepresenteerd: dit project heeft geresulteerd in een werkend prototype dat voldoet aan de eisen en er is ervaring opgedaan met meerdere veldtesten. Helaas zijn er niet in de praktijk apparaten geschakeld, dit blijkt lastiger te realiseren dan verwacht.

Het algoritme dat de meet-data desaggregeerd is sterk verbeterd en levert resultaten waar gebruikers iets mee kunnen. Wel is gebleken dat naast inzicht, vooral analyses en tips (behavior change) belangrijk zijn. Dit is dan ook opgenomen als future work en ondertussen work in progress. De software en architectuur van de backend is geschikt gemaakt voor grote aantallen gebruikers met een goede responstijd en een tweerichtingscommunicatie om desaggregatie resultaten en stuursignalen te sturen is geïmplementeerd. De backend is verder ingericht naar de huidige standaarden voor big-data analyses om analyses op basis van vergelijking en competitie te ondersteunen. Het implementeren hiervan is ook opgenomen als future work en ondertussen work in progress. Er is een nieuwe Coded Power Box ontworpen die voldoet aan de eisen die zijn gesteld. Dit heeft aanpassingen aan de hardware tot gevolg gehad, maar vooral aanpassingen aan de software. De CPB is een “data-router” geworden die met verschillende slimme apparaten en pluggen kan communiceren en de functionaliteit hiervan op een generieke manier aan services aanbied. De Ipsum

propositie is geïmplementeerd als een van de services, maar ook het FPAI framework is geïntegreerd en daarmee een door meerdere partijen ontwikkelde interface voor DSM methodes.

Er is een prototype gebouwd waarmee een proof-of-concept van het nieuwe EMS is ontwikkeld. Op dit prototype zijn verschillende use-cases getest, daaruit blijkt dat directe feedback aan de gebruiker en het aansturen van apparaten door de gebruiker via een lokale front-end werkt. Daarnaast is er een use-case met DSM getest, waarbij de waterkoker werd uitgeschakeld in geval van congestie.

Er zijn meerdere veldtesten gedaan met meetapparatuur en het geven van feedback, zowel bij consumenten als bij bedrijven. Bij de consumenten zijn de belangrijkste lessen die geleerd zijn het soort feedback dat gegeven moet worden en de kwaliteit. Het blijkt dat analyses/tips en vergelijkingen met anderen erg belangrijk zijn. Bij bedrijven is gebleken dat er momenteel erg weinig inzicht beschikbaar is, dus dat het al snel goed is. Wel is het installeren hier een stuk complexer en vraagt meer voorbereiding.

### 3. Discussie

Energie wordt een steeds belangrijker onderwerp, waarbij de focus enerzijds ligt op het besparen van energie en anderzijds op het balanceren van vraag en aanbod. Waar in het huidige energiesysteem het aanbod zich aanpast aan de vraag, is dit bij niet-stuurbare hernieuwbare bronnen niet mogelijk. Dit kan opgelost worden door het grootschalig toepassen van opslag, maar een gedeelte kan ook opgelost worden door het verschuiven van de vraag, Demand Side Management. In dit project is er gewerkt aan een Energie Management Systeem dat in beide zaken voorziet.

De eerste uitdaging is om mensen betrokken te krijgen bij energie; energie is een commodity dat altijd aanwezig is en waar eenmaal per jaar een afrekening van komt. De gebruikers is er niet mee bezig, maar de gebruiker heeft ook geen inzicht in waar zijn verbruik in zit. De eerste stap is dus om gebruikers betrokken te krijgen bij zijn energieverbruik en ten tweede om hem inzicht te geven waar hij wat mee kan. In dit project is gewerkt aan een gadget-achtige oplossing waarbij een EMS een toevoeging is voor de gebruiker; het is een handige manier om zijn apparaten te bedienen. Daarnaast geeft het inzicht op apparaat niveau, iets wat nodig is. Uit tests is echter gebleken dat dit niet voldoende is om echte energiebesparing teweeg te brengen, de gebruiker heeft hulp nodig bij het nemen van acties. Enerzijds omdat door aan te geven met behulp van analyses waar te besparen valt, anderzijds door zijn aandacht vast te houden met vergelijkingen met anderen/competities en andere “hooked” methoden [4].

In het project is gebleken dat gebruikers weinig gevoel hebben voor energie, bijvoorbeeld voor hoeveel een apparaat verbruikt en welk apparaat impact heeft op het comfort van de gebruiker. Naast inzicht en hulp bij analyse is dus vooral voorlichting ook zeer belangrijk. Verder blijkt ook hier dat wanneer gebruikers eerste inzichten krijgen de wensen voor wat ze zouden willen ook veranderen. Dit is dus een iteratief proces, maar betekent ook dat het precieze beginpunt niet heel precies komt. Het echte inzicht komt pas na het initiële inzicht.

Het meten, opslaan en verwerken van energiedata is iets waar de meeste gebruikers geen problemen mee hebben. Het schakelen van apparaten is een ander ding. Hier moet ook duidelijk gemaakt worden wat de reden is en wat het voordeel voor de gebruiker is. Dit is in een zakelijke omgeving gemakkelijker dan in een consumenten omgeving. Het moet dus voorzichtig geïntroduceerd worden, maar het heeft zeker potentie om de stabiliteit van het energiesysteem te verhogen, zeker in toekomstige scenario's met veel duurzame opwek en elektrisch vervoer. Dit in combinatie met het bovenstaande betekent dat het EMS waar een begin mee gemaakt is, moet worden uitgebouwd in nauwe samenwerking met de eindgebruikers. Dit is wat het onderwerp is in het vervolgproject 'Living lab' waarin een aantal kantoren en consumenten huizen “smart-grid ready” gemaakt gaan worden.

Uniek aan het Mobeg-concept is de combinatie van feedback en sturing van apparaten. Voor het geven van feedback is een enkel meetpunt in de meterkast nodig, een eenvoudige installatie. Op basis hiervan kan gedetailleerd inzicht gegeven worden in het energieverbruik. Van hieruit kan stapsgewijs het systeem uitgebreid worden met het sturen van apparaten. Hierbij hoeven alleen de apparaten die geschakeld gaan worden uitgerust te worden met extra hardware.

## 4. Conclusie en aanbevelingen

In dit project is een eerste versie van een EMS ontwikkeld en op basis daarvan is een prototype ontwikkeld. Dit prototype voldoet aan de eisen die gesteld zijn en de doelstelling zoals verwoord op pagina zes en in het projectplan behaald. Het EMS geeft feedback aan de gebruiker en is in staat om apparaten te meten en te schakelen, zowel op basis van signalen van de gebruiker als op basis van signalen van een DMS systeem. Helaas is dit prototype niet in de praktijk getest. Wel zijn er veldtesten gedaan met het meet- en feedback gedeelte waaruit waardevolle lessen zijn geleerd waarmee de feedback naar gebruikers iteratief is verbeterd en kennis is verkregen over het installatie proces. Belangrijk is dat een dergelijk systeem in samenspraak met de gebruikers gemaakt moet worden en dat de gebruikers goed geïnformeerd moeten worden; ze hebben weinig kennis over deze materie en de gevolgen voor henzelf en het systeem. De combinatie van een stapsgewijze uitbreiding vanuit een zeer simpel systeem met een enkel meetpunt en alleen extra hardware voor de apparaten die geschakeld gaan worden ondersteund dit proces.

Binnen het Mobeg-concept is een belangrijk uitgangspunt dat er een open platform is waarbij slimme apparaten eenvoudig aangesloten kunnen worden op het systeem (door het schrijven van een plugin voor de driver) en dat meerdere services *tegelijktijd* gebruik kunnen maken van de apparaten; bijvoorbeeld de combinatie van een front-end in huis waarbij apparaten rechtstreeks aangestuurd worden met een DSM methode die de apparaten ook aanstuurt (binnen de overgebleven marges).

Daarnaast zijn er een aantal vervolgotrajecten opgestart:

### Project living lab

In dit project wordt verder gewerkt aan het prototype ontworpen in dit project. Dit prototype zal in vier kantoren in Enschede, waarvan twee van de Universiteit Twente, geïnstalleerd worden. Daarnaast zal de dongel uitgedeeld worden aan 700 consumenten. Het doel van dit project is om de kantoren Smart Grid ready te maken (inzicht in verbruik en DSM) en bij de huizen kennis te vergaren hoe er energie bespaard gaat worden. Hiervoor is al een nieuwe website voor zakelijke gebruikers in ontwerp (requirements studie met gebruikers) en is een eerste versie van een smartphone app inclusief analyses en feedback ontwikkeld. De verwachting is dat deze zomer de eerste kantoren met apparatuur uitgerust gaan worden.

### Geautomatiseerde analyses

Zoals gebleken zijn analyses erg belangrijk, er wordt gewerkt aan automatische analyses van het verbruik (op basis van vergelijkingen/big-data analyses). Dit moet op de langere termijn gaan leiden tot een behavior change module.

### Hardware ontwikkeling in Polen

Op basis van de requirements en ervaringen in dit project is er een traject opgestart met de SUT in Polen in samenwerking met een elektronica producent voor de grootschalige (250K) productie van CPBs.



## 5. References

- [1] System integration for Smart Energy Systems, Bosch G. (Bachelor thesis)
- [2] Requirements for feedback to save energy: A comparison between professionals in small to medium sized enterprises and consumers, Vosslander S. (Master thesis)
- [3] <http://velocity.apache.org/>
- [4] Eyal N. and Hoover R., Hooked: How to build Habit-forming products, December 2013, ISBN 9781494277536
- [5] <https://hazelcast.com/>
- [6] <http://storm.apache.org/>
- [7] <https://www.endax.net/spas/1>
- [8] <https://www.hellodata.org/>
- [9] Project: Coded Power Box, Bosch, G. (internship)
- [10] Universal Smart Home Controller, Cruz, J. (Bachelor thesis)
- [11] Requirements document CPB Ipsum/Salland (op te vragen bij Ipsum)
- [12] Work in progress, see <http://caes.ewi.utwente.nl/research/projects/active-projects/i-care/albert-molderink.html> for details
- [13] <https://www.tno.nl/en/focus-area/energy/sustainable-energy/smart-energy-system-solutions/powermatcher/>
- [14] User accepted flexibility model for Demand Side Management, Molderink, A. and Hoogsteen, G. and Vosslander, S. and Hurink, J.L. and Noordzij, M. (work in progress)



## Appendix A: protocol omschrijving

```
message IpsumMessage {
    required uint32 version = 1;
    repeated StatusMessage statusMessage = 2;
    repeated CPBMeasurement cpbMeasurement = 3;
    repeated Reconfiguration reconfiguration = 4;
    repeated MeasurementAck measurementAck = 5;
    optional DeviceMessage deviceMessage = 6;
}

message StatusMessage {
    enum Status {
        OK = 0;
        NOK = 1;
        UNAUTHORIZED = 3;
    }
    required Status status = 1;
    optional string msg = 2;
}

message CPBMeasurement {
    optional int64 timestamp = 1; // CPB-meting voor alle spoelen
    repeated SensorMeasurement sensorMeasurement = 2; // de unixtimestamp (in seconden)
    message SensorMeasurement { // 1 kring/groep/spoel
        optional UUID sensorSN = 1; // Sensor serial number
        optional sint32 activePower = 2; // in watt
        optional sint32 reactivePower = 3; // in var
        optional sint32 voltage = 4; // verschil met 230.0 V(decivolt)
    }
}

message Reconfiguration {
}

message MeasurementAck {
    optional int64 timestamp = 1; // de unix timestamp (in seconden)
    repeated UUID sensorSN = 2; // Sensor serial number
}

message DeviceMessage { // CPB-meting voor alle spoelen
    optional int64 timestamp = 1; // de unixtimestamp (in seconden)
    repeated Device devices = 2;
    message Device {
        required string deviceType = 1;
        optional string deviceDescription = 2;
        required UUID deviceId = 3;
        repeated Usage usages = 4;
        message Usage {
            required int64 timestamp = 1;
            required int32 powerUsage = 2;
        }
    }
}
}
```