

Portabolt Research

Tom Bergisch, Maikel van Es

10 september 2023

Contents

1 Inleiding

In deze sectie wordt een inleiding gegeven tot het project. Hier kun je uitleg geven over de achtergrond, doelstellingen, en algemene context van het project.

Het project beoogt een gedetailleerde beschrijving en ontwikkeling van verschillende werkpakketten die bijdragen aan de succesvolle uitvoering van het MIT AIP Grant Project.

2 Werkpakket 1: Scoping en modelling

In dit werkpakket zal eerst verder onderzocht worden hoe het verdienmodel er exact uitziet. In de keten zit een afnemer, Portabolt met de hardware en AIP met de connectie naar de beurs. Er zal grondig onderzoek nodig zijn om te achterhalen hoe elke schakel in de keten voldoende waarde krijgt. Er zal ook van tevoren moeten worden nagedacht over hoe elke laag met elkaar communiceert en hierover moeten afspraken gemaakt worden. Er zal in kaart gebracht worden hoe een prototype er precies uitziet en hoe we kunnen bewijzen dat alle stappen die gezet moeten worden haalbaar zijn.

2.1 Deliverables

1. Uitgewerkte verdienmodellen per doelgroep en partner
2. Functioneel ontwerp contractuele randvoorwaarden van dienstverlening
3. Interfaces documentatie (tussen partners)
4. Functioneel ontwerp prototype

2.2 Uitgavenplanning begroting

Uren begroot: 740

2.3 Uitgewerkte verdienmodellen per doelgroep en partner

2.3.1 Identificeer doelgroepen en partners

Begin met het identificeren van alle belangrijke doelgroepen en partners die betrokken zijn bij het project. Dit kan onder meer de eindgebruikers (doelgroepen), leveranciers, tussenpersonen en andere relevante partijen omvatten. In dit specifieke geval zijn de belangrijkste partijen mogelijk de afnemers, Portabolt en AIP.

2.3.2 Analyseer toegevoegde waarde

Voor elke geïdentificeerde partij, analyseer en begrijp de specifieke waarde die zij toevoegen aan het product of de dienst. Wat maakt hun bijdrage uniek en waardevol? Hoe dragen ze bij aan het uiteindelijke succes van het project?

2.3.3 Verdienmodel ontwerpen

Portabolt Power is een verhuurbedrijf van modulaire mobiele batterijsystemen. In de verhuur draait alles om de bezettingsgraad. Er zijn voor Portabolt Power 3 verschillende businesscases.

De inzet van assets die niet in de verhuur staan De bezettingsgraad van Portabolt Power ligt momenteel tussen de 50 en 60 procent. Voor deze analyse gaan we uit van een standaard asset van 15 kVA met een 40 kWh batterij. De verhuuropbrengsten van deze asset zijn 450 euro per week. Uitwerken verdienmodel in spreadsheet.

De inzet van assets bij de klant Zodra ze bij de klant draaien, kunnen ze ook ingezet worden om netcongestie op te lossen. Dit kan nuttig zijn bij specifieke toepassingen waarbij de peakload hoog is en de netaansluiting niet toereikend is.

De volledige inzet van assets, stop van de verhuur Dit is naar verwachting de meest rendabele businesscase. Er is een omslagpunt vanaf wanneer de continue omzet van assets op de onbalansmarkt lucratiever wordt dan de losse verhuur van batterijen. Dit omslagpunt wordt nauwlettend in de gaten gehouden.

2.3.4 Rekening houden met belangen

Zorg ervoor dat het ontwikkelde verdienmodel recht doet aan de belangen en investeringen van elke partij. Het moet een win-winsituatie creëren waarbij alle betrokken partijen profiteren van de samenwerking.

2.3.5 Documentatie en communicatie

Documenteer het uitgewerkte verdienmodel duidelijk en communiceer het aan alle betrokken partijen. Zorg ervoor dat er een gemeenschappelijk begrip is van hoe elke partij zal profiteren en wat hun verantwoordelijkheden zijn.

2.4 Functioneel ontwerp contractuele randvoorwaarden van dienstverlening

2.4.1 Definieer contractuele verplichtingen

Begin met het identificeren van alle relevante contractuele verplichtingen die moeten worden vastgelegd tussen de verschillende partners. Dit omvat zaken als serviceniveaus, prijsstructuren, aansprakelijkheid, vertrouwelijkheid en andere voorwaarden die van invloed zijn op de dienstverlening.

2.4.2 Bepaal rechten en verantwoordelijkheden

Specificeer duidelijk de rechten en verantwoordelijkheden van elke partij binnen de overeenkomst. Dit omvat wat elke partij moet leveren, wat ze kunnen verwachten, en welke acties moeten worden ondernomen in verschillende scenario's.

2.4.3 Juridische compliance en risicobeheer

Zorg ervoor dat de contractuele randvoorwaarden voldoen aan alle relevante juridische vereisten en standaarden. Dit omvat het identificeren en beheren van eventuele juridische risico's die kunnen ontstaan uit de dienstverlening.

2.4.4 Prijsstructuren en betalingsvoorwaarden

Definieer duidelijk de prijsstructuren voor de diensten die worden geleverd, inclusief eventuele variabele kosten, abonnementskosten of andere tarieven. Specificeer ook de betalingsvoorwaarden, facturatiecycli en eventuele boetes of vergoedingen voor te late betalingen.

2.4.5 Service Level Agreements (SLA's)

Ontwikkel gedetailleerde Service Level Agreements (SLA's) die de kwaliteit, beschikbaarheid, prestaties en ondersteuningsniveaus van de dienstverlening definiëren. Deze SLA's moeten meetbaar en haalbaar zijn en moeten de verwachtingen van alle betrokken partijen weerspiegelen.

2.4.6 Vertrouwelijkheid en gegevensbescherming

Beheer vertrouwelijkheidsclausules en bepalingen met betrekking tot gegevensbescherming en privacy. Zorg ervoor dat gevoelige informatie adequaat wordt beschermd en dat er passende maatregelen zijn genomen om de privacy van gebruikers te waarborgen.

2.4.7 Onderhandeling en goedkeuring

Faciliteer onderhandelingen tussen de betrokken partijen om tot overeenstemming te komen over de contractuele randvoorwaarden. Zorg ervoor dat alle partijen zich gehoord voelen en dat eventuele geschillen of zorgen worden aangepakt voordat de overeenkomst definitief wordt goedgekeurd.

2.4.8 Documentatie en ondertekening

Documenteer de contractuele randvoorwaarden nauwkeurig en zorg ervoor dat alle betrokken partijen de overeenkomst begrijpen en akkoord gaan met de voorwaarden voordat ze worden ondertekend.

2.5 Interfaces documentatie (tussen partners)

2.5.1 Identificeer de vereiste interfaces

Identificeer alle interfaces die nodig zijn voor de communicatie tussen de partners. In dit geval is het essentieel om de API-specificaties te definiëren die Portabolt Power levert voor de aansturing van het batterijsysteem.

2.5.2 Documenteer de API-specificaties

Documenteer gedetailleerd de specificaties van de API die Portabolt Power levert. Dit omvat de beschikbare endpoints, ondersteunde HTTP-methoden (zoals GET, POST, PUT, DELETE), vereiste parameters, geaccepteerde invoerformaten (bijv. JSON), en de verwachte uitvoerformaten en responscodes.

2.5.3 Authenticatie en autorisatie

Specificeer de authenticatiemethoden en autorisatievereisten die nodig zijn om toegang te krijgen tot de API van Portabolt Power. Dit kan onder meer het gebruik van API-tokens, OAuth 2.0-authenticatie of andere mechanismen omvatten.

2.5.4 Gegevensuitwisseling en protocollen

Beschrijf hoe gegevens worden uitgewisseld tussen de partners via de API. Dit kan onder meer het gebruik van bepaalde gegevensindelingen, zoals JSON of XML, omvatten, evenals de communicatieprotocollen, zoals HTTPS, die worden gebruikt voor de beveiligde overdracht van gegevens.

2.5.5 Foutafhandeling en logging

Definieer hoe fouten worden afgehandeld en gelogd binnen de API. Dit omvat het vastleggen van foutmeldingen, het definiëren van standaardfoutcodes en het bieden van richtlijnen voor foutafhandeling aan de partners die de API gebruiken.

2.5.6 Beveiliging en gegevensbescherming

Adres beveiligingsmaatregelen zoals gegevensversleuteling, beveiligde overdracht van gegevens en toegangscontrolemechanismen om ervoor te zorgen dat de API van Portabolt Power voldoet aan de hoogste normen voor gegevensbescherming en privacy.

2.5.7 Testen en validatie

Ontwikkel testcases en validatiemethoden om ervoor te zorgen dat de API van Portabolt Power correct functioneert en voldoet aan de verwachtingen van de partners. Dit omvat het testen van verschillende scenario's en randgevallen om de robuustheid en betrouwbaarheid van de API te garanderen.

2.5.8 Documentatie en ondersteuning

Lever uitgebreide documentatie voor de API van Portabolt Power, inclusief handleidingen, voorbeelden en API-referenties, om partners te helpen bij het effectief integreren en gebruikmaken van de API.

3 Werkpakket 2: Prototyping

3.1 Prototyping V0.1 (4 maanden)

Het doel van het prototype is om zo snel mogelijk alle afzonderlijke stappen te testen en bewijzen dat het werkt. Het zal in een virtuele omgeving draaien waarbij makkelijk geïtereerd kan worden. Portabolt zal een interface ontwikkelen waarmee extern de batterijen aangestuurd kunnen worden. Er zal een communicatielaag worden gebouwd waarmee AIP de batterijen kan aansturen. Er wordt een testprotocol ontwikkeld waarmee elke afzonderlijke stap getest en gevalideerd kan worden.

3.2 Deliverables

1. Portabolt sturingsinterface
2. Koppeling tussen All in power, Portabolt, Edge, Envitron
3. Werkend prototype in gesimuleerde omgeving

3.3 Uren begroot

Uren begroot: 1720

Opbouwen van de sandbox. We gaan eerst alles in dit prototype volledig digitaal ontwikkelen om te kijken of het werkt en waar we tegenaan lopen. Daarvoor gaan we een aantal modules ontwikkelen. Deze modules zijn de Price Generation Module, de Energy Profile Simulation, en de Optimization Module. We gaan in deze fase nog geen daadwerkelijke data van het internet scrapen, maar alleen bewijzen dat alles werkt.

3.4 1. Price Generation Module

Objective: Simuleer dynamische elektriciteitsprijzen die elke dag om 12:00 uur gegenereerd worden voor de volgende dag.

Input: Random variabelen die de prijzen beïnvloeden (bijv. vraag, aanbod, weersomstandigheden).

Output: Een array met 24 prijzen, één voor elk uur van de volgende dag.

Deze genereert elke keer dat je de code runt een array met prijzen. Er is een trend in gehardcode waarmee we een normaal dagelijkse prijsschommeling simuleren.

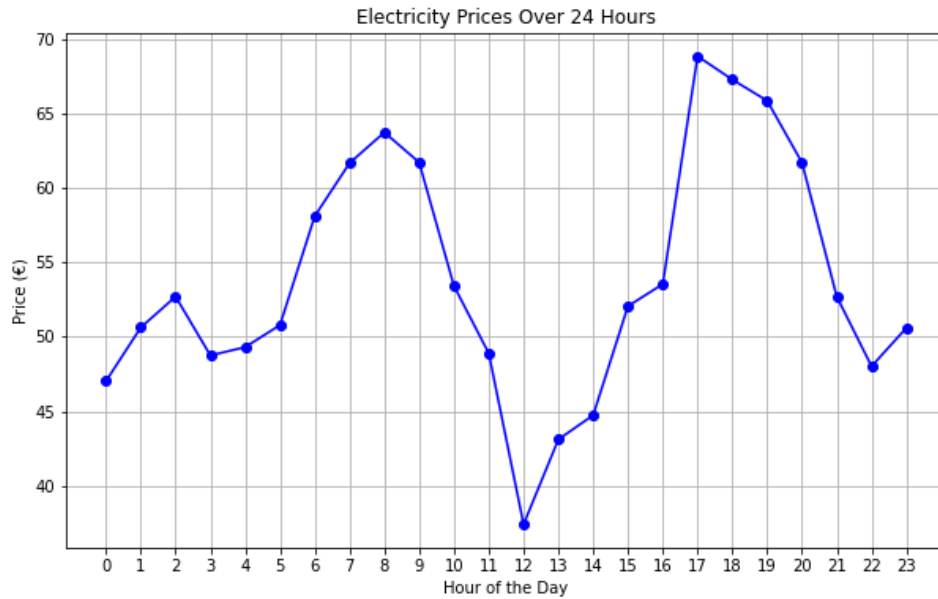


Figure 1: Electricity Prices Over 24 Hours

3.5 2. Energy Profile Simulation

Objective: Creëer een energieverbruiksprofiel voor een gemiddeld bedrijf, dat per uur varieert.

Input: Kenmerken van het bedrijf (bijv. grootte, type industrie).

Output: Een array met 24 energieverbruikswaarden, één voor elk uur van de dag.

Hiermee simuleren we een willekeurig energieverbruik van een bedrijf.

3.5.1 Doel van de simulatie

Het doel is om een representatief energieverbruikspatroon van een gemiddeld bedrijf te simuleren. Dit kan bijvoorbeeld per uur, dag, week of maand worden gegenereerd. Het profiel kan variëren afhankelijk van het type bedrijf, werkuren, seizoensinvloeden en andere factoren.

3.5.2 Belangrijke variabelen

- **Bedrijfsprofiel:** Werkuren, industrietype, locatie, enz.
- **Baseload:** Het minimum energieverbruik van het bedrijf gedurende een dag (bv. verlichting, servers).
- **Piekbelasting:** Het maximale energieverbruik tijdens piekuren.
- **Dagelijks variërend verbruik:** Gebaseerd op het type bedrijf en de werkuren.
- **Seizoensgebonden variatie:** Verwarmings- of koelbehoeften.
- **Weekends en feestdagen:** Mogelijk lager verbruik.

3.5.3 Simulatiemethoden

- **Random generation:** Gebruik een stochastisch model om het energieverbruik te genereren met een bepaalde spreiding rond gemiddelde waarden.
- **Tijdreeksanalyse:** Op basis van historische data (indien beschikbaar) kun je een tijdreekspatroon simuleren.

3.5.4 Python Code Structuur

De code kan worden gestructureerd met de volgende componenten:

Main Simulation Script (main.py) Dit bestand start de simulatie, genereert het energieprofiel, en visualiseert het resultaat.

- Initialiseer de simulatie.
- Roep de functies aan om energieverbruiksprofielen te genereren.
- Plot de resultaten.

Company Profile (company_profile.py) Dit script bevat functies om een bedrijfsprofiel te definiëren (bv. industrieel, kantoor, winkel). Elke categorie kan specifieke energiebehoeften en patronen hebben.

```
def define_company_profile():  
    # Return een standaard bedrijfsprofiel op basis van categorie.
```

Energy Profile Generator (energy_profile.py) Bevat de logica om het energieverbruik-sprofiel te genereren.

```
def generate_daily_profile(company_profile):  
    # Genereert het dagelijkse energieverbruik op basis van bedrijfsprofiel.  
  
def apply_seasonal_variation(profile):  
    # Past seizoensgebonden variaties toe op het energieprofiel.  
  
def simulate_weekends(profile):  
    # Voegt weekend- en feestdagvariaties toe.
```

Utilities (utils.py) Handige functies voor bijvoorbeeld het genereren van random verbruikswaarden, plotten van de data, etc.

3.5.5 Simulatieverloop

- Genereer eerst een basis energieprofiel voor een gemiddelde dag.
- Pas vervolgens variaties toe voor piekbelasting, weekends, en seizoensgebonden schommelingen.
- Resultaten visualiseren met grafieken om een duidelijk overzicht van het verbruik te geven.

3.5.6 GitHub-structuur

Maak de volgende mappen en bestanden in de `energyprofile` folder:

```
energyprofile/  
|-- main.py  
|-- company_profile.py  
|-- energy_profile.py  
|-- utils.py  
|-- data/  
    |-- (eventuele historische data of configuratiebestanden)
```

3.5.7 Visualisatie

Gebruik bijvoorbeeld `matplotlib` om het energieverbruik door de tijd te plotten.

3.6 3. Optimization Module (Trading Algorithm)

Objective: Beslis wanneer elektriciteit moet worden gekocht of verkocht, gebaseerd op prijsfluctuaties en het energieverbruik van het bedrijf.

Input: Prijsarray en energieverbruiksarray.

Output: Een optimalisatiestrategie die maximale winst oplevert door elektriciteit op het laagste punt te kopen en op het hoogste punt te verkopen.

4 Werkpakket 3: Titel van Werkpakket 3

Hier komt de beschrijving van werkpakket 3.

zijn er nog andere zaken die ik wil testen?

5 Werkpakket 4: Titel van Werkpakket 4

Hier komt de beschrijving van werkpakket 4.

6 Werkpakket 5: Titel van Werkpakket 5

Hier komt de beschrijving van werkpakket 5.

7 Werkpakket 6: Titel van Werkpakket 6

Hier komt de beschrijving van werkpakket 6.

8 **Werkpakket 7: Titel van Werkpakket 7**

Hier komt de beschrijving van werkpakket 7.