
1 Werkpakket 2: Prototyping

1.1 Prototyping V0.1 (4 maanden)

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

1.2 Deliverables

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

1.3 Urenbegroting

Uren begroot: 1720

De sandbox wordt opgebouwd. Alles wordt in dit prototype volledig digitaal ontwikkeld om te testen of het werkt en om te zien waar we tegenaan lopen. Hiervoor worden drie modules ontwikkeld: de Price Generation Module, de Energy Profile Simulation, en de Optimization Module. In deze fase wordt nog geen daadwerkelijke data van het internet gescraped, maar alleen de werking van de modules bewezen.

1.4 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 (zoals vraag, aanbod, weersomstandigheden).

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

Bij elke run van de code wordt een array met prijzen gegenereerd. Er is een gehardcodeerde trend waarmee een normale dagelijkse prijsschommeling gesimuleerd wordt.

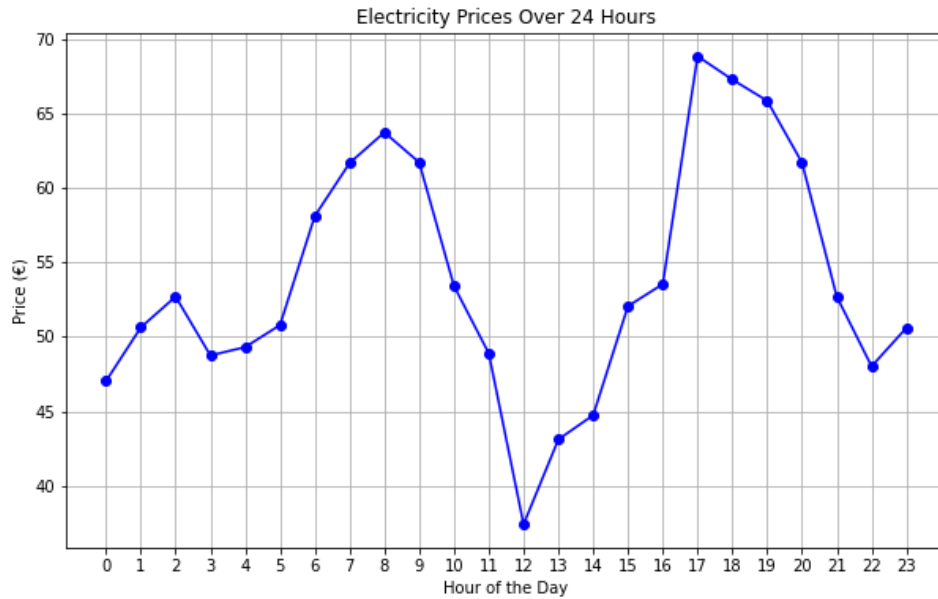


Figure 1: Electricity Prices Over 24 Hours

1.5 Energy Profile Simulation

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

Input: Kenmerken van het bedrijf (zoals grootte en type industrie).

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

Dit module simuleert het willekeurige energieverbruik van een bedrijf.

1.5.1 Doel van de simulatie

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

1.5.2 Belangrijke variabelen

- **Bedrijfsprofiel:** Werkuren, industrietype, locatie, enz.
- **Baseload:** Het minimum energieverbruik van het bedrijf gedurende de dag (bijv. 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.

1.5.3 Simulatiemethoden

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

1.5.4 Python Code Structuur

De code voor de simulatie wordt gestructureerd in verschillende componenten:

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

- Initialiseer de simulatie.
- Roep de functies aan om energieverbruiksprofielen te genereren.
- Visualiseer de resultaten door de data te plotten.

Company Profile (company_profile.py) Dit script bevat functies om een bedrijfsprofiel te definiëren, bijvoorbeeld voor een industrieel bedrijf, kantoor of winkel.

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

Energy Profile Generator (energy_profile.py) Dit script bevat de logica voor het genereren van energieverbruiksprofielen.

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

Utilities (utils.py) Dit script bevat handige functies, zoals voor het genereren van random verbruikswaarden, het plotten van de data, enz.

1.5.5 Simulatieverloop

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

1.5.6 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)
```

1.5.7 Visualisatie

Gebruik `matplotlib` om het energieverbruik over tijd te visualiseren.

1.6 Optimization Module (Trading Algorithm)

Objective: Ontwikkel een algoritme dat beslist wanneer elektriciteit gekocht of verkocht moet worden, gebaseerd op prijsfluctuaties en het energieverbruik van het bedrijf.

Input: Een prijsarray en een energieverbruiksarray.

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