Cahier des charges architecture multi-agents

February 4, 2026

1 Database

- Données réelles : issues des données capteurs et de l'utilisateur (simulateur de données)
  Clé Primaire : (heure, jour, mois, année)
  Une ligne par pas de temps qui contient :
  - heure
  - jour
  - mois
  - année
  - t_{frigo}
  - T_{min}
  - T_{max}
  - $T_{out-reel}$
  - $T_{in-reel}$
  - G
  - $\alpha_{presence-reel}$
  - PV_{reel}
  - P_{fixe}
  - $P_{flex-reel}$
  - P_{in}
  - P_{go}
  - E_{bat}
  - S
  - $C_{buy-reel}$
  - $C_{sell-reel}$
- Données prédites : provient des API, modèles de prédiction
  Clé Primaire : (heure, jour, mois, année)
  une ligne par pas de temps qui contient :
  - heure
  - jour
  - mois
  - année
  - T_{out}

- T_{in}
- G
- $\alpha_{presence-predit}$
- PV
- $P_{fixe-predit}$
- $P_{flex-predit}$
- $C_{buy-predit}$
- $C_{sell-predit}$

2 Liste de fonctionnalités

2.1 Optimiseur

Description Répartir la demande utilisateur entre les differentes sources d'energie.
Objectifs

- Minimiser les coûts
- Minimiser les émissions carbone
- Maximiser le confort utilisateur
- Maximiser la charge de la batterie (pour maximiser sa durée de vie)

Entrées

- Horizon de temps: 24h
- Pas de temps: 1h
- (demande) Puissance fixe de l'utilisateur ($P_{fixe}(t)$)
- (demande) Puissance flexible de l'utilisateur (peut être deconnectée) ($P_{flex}(t)$)
- (fixe) Cout de vente de l'energie (au main grid) (C_{grid_sell})
- (fixe) Cout d'achat de l'energie (au main grid) (C_{grid_buy})
- Puissance fournie par PV ($PV(t)$)
- (fixe) Coefficient d'émissions ($C_{emissions\_grid}$)
- (fixe) Paramètres physiques ($C_{bat}, E_{bat}^{max}, P_{go}^{max}, P_{in}^{max}, PV^{max}, P_{ch}^{max} = B^{max}$, $P_{dis}^{max} = A^{max}$)

Sortie Pour chaque pas de temps:

- Puissance achetée au main grid ($P_{\text{in}}(t)$)
- Puissance vendue au main grid ($P_{\text{go}}(t)$)
- Puissance injectee dans les batteries, charge ($P_{\text{ch}}(t)$).
- Puissance utilisée depuis la batterie, décharge ($P_{\text{dis}}(t)$).
- Variable binaire pour couper les charges flexibles ($S(t)$)

Résolution Module d'optimisation, MILP (CPLEX) -¿ cf partie déjà implémentée

2.2 Predicteur

2.2.1 Predicteur tarifs

Predire pour chaque jour suivant les prix d'achat et de vente au main grid

Entrées : Jour j

Sorties Tarif de l'électricité à chaque heure ($C_{\text{sell}}(t)$, $C_{\text{buy}}(t)$)

2 Resolutions possibles :

- Contrat d'élec de la forme : $C_{buy}(t) = C_{max}\mathbb{1}_{H\_pleines}(t)+C_{min}\mathbb{1}_{H\_creuses}(t)$ et C_{sell} déduit par un coef ($C_{sell} = \beta C_{buy}$)
- API pour récupérer modèle de prediction existant de $C_{buy}(t)$ [2] et déduire $C_{\text{sell}}(t)$ via un coef (autre lien API consulté : https://open-dpe.fr/API-prix-electricite/)

2.2.2 Predicteur PV

2.2.3 Predicteur puissance

Predire pour chaque jour suivant la production énergetique PV

Entrées : jour j

Sorties Production PV à chaque heure $PV(t)$

Resolution :

- Récuperer la prévision météo (température, irradiance solaire) via une API et la convertir en PV via :
  - Formule physique reliant l'irradiance et la température reçue sur le panneau photovoltaic et la puissance qu'il produit : [1] :

$$P(t) = P_{\text{STC}} \frac{G(t)}{1000} \left[1 + \beta \left(T_{\text{out}}(t) - 25\right)\right]$$

  - Puis, à partir d'un certain temps (environ quelques mois), la formule précédente n'est plus valable car les performances du panneau photovoltaïque se dégrade -¿ modèle de ML qui apprend la relation $PV(t) = f(T_{out}(t), G(t))$ à partir de nos données capteurs et des de l'historique des données dans notre BDD[3])

2.2.4 Prédicteur demande utilisateur

Description : Prédire la demande utilisateur (fixe et flexible) pour chaque jour suivant.
Définissons les puissances fixes et flexibles :

- P_{fixe} : frigo à température t_{frigo} donnée par utilisateur ou fixée, intervalle de température $[T_{min}, T_{max}]$ donné par l'utilisateur ou fixé (T_{min} pour chauffage en hiver et T_{max} pour clim en été), plaques électriques, lumières, HWE (chauffage eau)
- P_{flex} : température de confort, machine à laver, sèche-linge, lave-vaisselle, appareils électroniques

Entree :

- Historique des données capteurs pour P_{flex} et P_{fixe} (notamment $t, T_{out}(t), T_{in}(t), G(t), \alpha_{presence}(t)$ et les vraies valeurs de $P_{flex}(t)$ et P_{fixe}) + Préférences utilisateur $(t_{frigo}, T_{min}, T_{max})$ pour P_{fixe}

Sortie : Renvoie $P_{\text{fixe}}(t)$ et $P_{\text{flex}}(t)$ à chaque heure (puissance demandée)

Resolution Module de prédiction (Outils statistiques) qui se base sur l'historique de la demande (P_{fixe} et P_{flex}) et qui prend en compte les preferences utilisateur (s'il y'en a)

- P_{fixe} : modèle de ML qui apprend la relation $P_{\text{fixe}}(t) = f(t, T_{out}(t), T_{in}(t), G(t), \alpha_{presence}(t))$
- P_{flex} : modèle de ML qui apprend la relation $P_{\text{flex}}(t) = f(t, T_{out}(t), T_{in}(t), G(t), \alpha_{presence}(t))$

2.3 Data qualité

Description Récupère les données des prédictions et des capteurs et les aggrège dans les databases correspondantes.

Entree : Bases de données, prédictions et données capteurs

Sorties : Bases de données mises à jour

Resolution Traitement de données :

Tâches de l'agent

- **Récupération des données toutes les 24h**
  - Récupérer les nouvelles données **réelles** des capteurs et/ou simulateurs toutes les 24h
  - Récupérer les nouvelles données **prédites** des API et/ou modèles de ML que nous avons fait toutes les 24h
- **Verification et preprocessing de la donnée**
  - S'assurer lors de la récupération des données que ce sont les noms de variables attendues et les bonnes unités (W→kW, Wh→kWh, etc.) et formats (float / int / bool) : mapping source de données (API, générateur de données → schéma interne de notre BDD).
  - Détecter les **doublons** et les **trous** et les flag (OU BIEN on définit des règles : pour les trous de prendre les mêmes valeurs que la journée précédente).
- **Contrôles de plages physiques**
  - Vérifier que les valeurs sont dans les bornes physiques configurées (ex : $T \in [T_{\min}, T_{\max}]$, $G \geq 0$, $\alpha_{presence} \in [0,1]$, $E_{bat} \in [0, E_{bat}^{max}]$).
  - Si hors plage : appliquer une règle (*clamp* à la borne)
- **Contrôles de cohérence interne (règles métier)**
  - Vérifier la cohérence entre variables au même pas de temps (ex : éviter $P_{in}(t) > 0$ et $P_{go}(t) > 0$ simultanément).
  - Vérifier la validité des variables binaires (ex : $S(t) \in \{0,1\}$) et corriger si nécessaire
  - Détecter des incohérences simples (ex : $G(t) = 0$ mais $PV(t)$ élevé) et créer des flags sans correction forcée.
- **Traçabilité et qualité**

  - Ajouter pour chaque variable des métadonnées : source (capteur/API/modèle/simulateur), date d'ingestion, version de règles.
  - Enregistrer les flags qualité (ex : `FORMAT_ERROR`, `OUT_OF_RANGE`, `TIME_GAP`, `IMPUTED`, `INCONSISTENT`).
  - Possibilité d'enrichissement des données, par exemple quand des données ont été bien prédites afin de leur affecter un poids potentiel dans les entrainements futurs

- **Exposition des données prêtes à consommer**
  - Fournir une extraction "clean" sur une fenêtre (ex : 24h) pour l'optimiseur
  - Fournir une extraction historique pour l'analyse statistiques et pour la détection d'anomalies.

2.4 Detection d'anomalies

Description Identifier dysfonctionnements dans les capteurs (valeurs aberrantes). Identifier des comportements utilisateurs irresponsables.

Sorties Alertes et/ou des recommandations sur une planification de maintenance.

Resolution 2 résolutions possibles:

- Etablir des règles "dures" pour toutes les variables provenant des capteurs et des utilisateurs
- Utiliser des méthodes statistiques indiquant si la valeur d'une variable provenant d'un capteur ou d'un utilisateur est anormale : modèle de machine learning non supervisé comme Isolation Forest ou décomposition de la série temporelle X_t selon saison, tendance, résidu puis tests sur les résidus (modèle SARIMA par exemple)

2.5 Analyse de données

A faire plus tard: Représentation des résultats

Description Evaluer le systéme à partir d'indicateurs de performance

Entrées Database

Sortie : Dashboard + chatbot

Resolution : Analyse de données + LLM

References

[1] Simon Meunier. Photovoltaic systems — cours cs 2a renewable energy. https://centralesupelec.edunao.com/pluginfile.php/436931/mod_resource/content/1/Lecture_Photovoltaic_Systems_20240406.pdf, 2024. Consulté en janvier 2026.

[2] RTE France. Prix horaire d'achat d'électricité — données marché eco2mix. https://www.rte-france.com/donnees-publications/eco2mix-donnees-temps-reel/donnees-marche, 2026. Consulté en janvier 2026.

[3] ScienceDirect. Optimal scheduling of smart home energy systems: A user-friendly and adaptive home intelligent agent with self-learning capability. https://www.sciencedirect.com/science/article/pii/S2666792424000209?ref=pdf_download&fr=RR-2&rr=9c18959028cdda89, 2024. Consulté en janvier 2026.