MSPR – Collecte & Stockage des Données pour une Solution IA

Rapport Professionnel – Pipeline ETL & Architecture Data

Équipe Projet – Bachelor IA XP

Synthèse Exécutive (Executive Summary)

Le Défi : Hétérogénéité et Fragmentation des Données

EnergiTech, comme toute entreprise gérant des actifs complexes, fait face à une **dispersion de ses données** critiques. Les informations vitales sur la production éolienne sont réparties entre :

1. Des fichiers manuels (CSV) de production.
2. Une base de données interne de capteurs à haute fréquence (PostgreSQL).
3. Des données externes (API Météo).

Cette fragmentation rend l'analyse des performances difficile et freine l'ambition d'EnergiTech de développer des modèles d'**Intelligence Artificielle (IA)** pour la maintenance prédictive.

Notre Solution : Le Pipeline ETL Robuste et Intelligent

Nous proposons un **pipeline ETL (Extract, Transform, Load)** entièrement automatisé et bâti sur une architecture **Python/PostgreSQL** reconnue pour sa fiabilité et sa performance.

L'objectif est de créer une **Source Unique de Vérité (SSOT)** : une table centralisée (consolidated_measurements) où toutes les mesures seront alignées, nettoyées et prêtes à l'emploi.

Bénéfice Clé	Description
Garantie d'Intégrité	Mécanisme de **Fusion intelligente (UPSERT/COALESCE)** assurant que les données existantes ne sont jamais écrasées par des valeurs manquantes.
Conformité IA	Standardisation des formats et des unités (ex: conversion des températures en Kelvin) pour l'exploitation directe par les algorithmes.
Fiabilité Opérationnelle	Cycle de traitement quotidien sans intervention, avec un **système de logging** détaillé pour une traçabilité complète des erreurs.
Pilotage Simplifié	**Tableau de bord Streamlit** pour une surveillance en temps réel de la performance du parc.

Ce livrable démontre notre capacité à passer des données brutes hétérogènes à une fondation analytique solide, prête pour la prochaine génération de solutions IA.

Problématique et Objectifs

1. La Problématique Opérationnelle

L'état actuel de l'information se caractérise par des silos, chacun avec ses propres limites :

Source de Données	Format/Fréquence	Défauts et Risques
Production	Fichier CSV (Journalier)	Risques d'erreurs humaines, données manquantes (NULL) et formats non standardisés.
Capteurs	PostgreSQL (Haute Fréquence - Temps réel)	Volumétrie élevée, données brutes (non nettoyées), nécessité de downsampler l'information.
Météo	API Externe (Horaire)	Données partielles (ex: pas d'informations sur la production), unités à convertir, dépendance à une source extérieure.

Conséquence : Corréler un événement de maintenance (capteur de vibration) avec la production réelle et les conditions météo exactes est un processus manuel, lent et sujet aux erreurs.

2. Les Objectifs de la Mission

Notre solution a été conçue pour répondre point par point aux exigences fonctionnelles du cahier des charges :

Objectif Cible	Résultat Obtenu
Centralisation des Données	Création de la table **consolidated_measurements** (clés : turbine_id, date) qui sert de référentiel unique.
Qualité des Données	Mise en place de règles de **contrôle qualité** (conversion d'unités, filtration des valeurs aberrantes physiques) pour une fiabilité analytique maximale.
Automatisation	Développement d'un pipeline complet (main_pipeline.py) gérant l'orchestration des extractions et du chargement.
Visualisation	Fourniture d'une interface **Streamlit** pour une vue temps réel des indicateurs clés et des tendances.

Architecture de la Solution

Notre pipeline ETL suit une architecture standard et éprouvée, mais avec un point de consolidation unique et intelligent. Il est conçu pour être à la fois **modulaire** et **extensible**.

Vue d'ensemble du Pipeline ETL

Le flux de données commence à partir des trois sources hétérogènes et converge vers notre base de données analytique après une étape de transformation rigoureuse.

Les Composants Techniques

Module	Rôle	Technologie Clé
Orchestrateur	Coordonne la séquence des étapes, gère le logging et le rapport final.	Python (main_pipeline.py)
Extraction (E)	Récupère les données brutes (CSV, requêtes DB, appel API).	Python (pandas, psycopg2, requests)
Transformation (T)	Nettoie, standardise les unités, applique les règles de qualité et fusionne les structures.	Python (pandas)
Chargement (L)	Insère ou met à jour de manière sécurisée et non destructive les données.	PostgreSQL (psycopg2, UPSERT/COALESCE)
Configuration	Centralise les paramètres (connexion DB, API, chemins) pour une maintenance simplifiée.	Fichier config.yaml

Cette architecture permet à EnergiTech de garantir une **séparation des préoccupations**, facilitant l'évolution future des connecteurs (ajout d'une nouvelle source, changement d'API météo).

Phase 1 : Extraction des Données (E)

La phase d'extraction est l'étape où les données sont collectées à partir de leurs sources originales, quelle que soit leur nature (fichier, base de données, web).

1. Extraction de la Production (CSV)

Le module extract_csv est conçu pour une robustesse maximale face à l'entrée manuelle :

- **Sélection Automatique :** Le script recherche dans le dossier data/ le **fichier CSV le plus récent** (ex: production_2025_10.csv), basé sur sa date de modification, garantissant de toujours traiter les informations les plus fraîches.
- **Lecture Flexible :** Utilisation de la librairie Pandas pour lire le fichier et gérer les délimiteurs spécifiques (point-virgule ;).

production_2025_12

date	turbin_id	energie_kWh	arret_planifie	arret_non_planifie
2025-12-01	T001	30997	0	0
2025-12-01	T002	34564	0	0
2025-12-02	T001	27236	0	0
2025-12-02	T002	20821	0	0
2025-12-03	T001	18200	0	0
2025-12-03	T002	38009	0	0
2025-12-04	T001	32323	0	0
2025-12-04	T002	39119	0	0
2025-12-05	T001	45428	0	0
2025-12-05	T002	36088	0	0
2025-12-06	T001	37219	0	0
2025-12-06	T002		0	0
2025-12-07	T001	20921	0	0
2025-12-07	T002	31681	0	0
2025-12-08	T001	32610	0	0
2025-12-08	T002	34686	0	0
2025-12-09	T001	57014	0	0
2025-12-09	T002	25599	0	0
2025-12-10	T001	28886	0	0
2025-12-10	T002	35590	0	0
2025-12-11	T001	37190	0	0
2025-12-11	T002	36929	0	0
2025-12-12	T001		0	0
2025-12-12	T002	35843	0	0

2. Extraction des Capteurs (PostgreSQL Interne)

Le module extract_db interroge la base de données interne d'EnergiTech contenant les mesures à haute fréquence (minuteparminute) des capteurs :

- **Fenêtre de Temps Ciblée :** L'extraction se concentre uniquement sur les **24 dernières heures** (paramètre configurable) pour optimiser la performance et réduire la charge sur la base de données source.
- **Pré-nettoyage :** La requête SQL effectuée par l'extracteur renomme immédiatement les colonnes brutes (ex: ts_utc devient date, wind_speed_mps devient wind_ms) pour correspondre au schéma cible, facilitant l'étape de transformation.

3. Extraction de la Météo (API Externe)

Le module extract_api récupère les données externes de vent, température et humidité, essentielles pour corréler les performances :

- **Interopérabilité :** Utilisation du protocole HTTP standard (module requests) pour interroger l'API Open-Meteo.
- **Duplication Intelligente :** Les données météo, qui sont valables pour l'ensemble du parc, sont dupliquées et étiquetées pour **chaque turbine active** (T001, T002…) de la période. Ceci prépare les données pour la fusion finale (jointure) un-à-un dans la base cible.

Phase 2 : Transformation et Qualité (T & Q)

La transformation est l'étape la plus critique pour l'**exploitabilité des données** par les futurs modèles d'IA d'EnergiTech. Le module transform.py assure l'uniformité et l'intégrité.

1. Standardisation des Unités et des Formats

- **Alignement Temporel :** Toutes les dates/heures sont converties et normalisées au format **ISO UTC**, garantissant que les mesures des capteurs (DB), de la production (CSV) et de la météo (API) sont parfaitement alignées.
- **Conversions Physiques :**
    - **Température :** Conversion de Celsius (API Météo) vers l'unité de référence interne **Kelvin**.
    - **Vent :** Conversion d'éventuels kilomètres par heure (km/h) vers l'unité standard **mètres par seconde (m/s)**.

2. Contrôle Qualité (quality_check) : Fiabiliser les Entrées

Le pipeline applique des règles métiers basées sur les limites physiques du parc éolien pour identifier et neutraliser les données aberrantes (*outliers*) :

Anomalie Détectée	Règle de Qualité Appliquée	Correction
Vent Excessif/Négatif	Vitesse du vent détectée supérieure à **42 m/s** (limite de sécurité de la turbine) ou négative.	Remplacement de la valeur par **NULL**.
Température Irréaliste	Température mesurée inférieure à 200 K ou supérieure à 330 K.	Remplacement de la valeur par **NULL**.

Stratégie de Correction : Au lieu de supprimer une ligne complète, nous remplaçons uniquement la valeur anormale par NULL. Ceci permet aux algorithmes de prédiction d'ignorer la valeur erronée sans perdre les autres informations valides (ex : l'énergie produite, même si la mesure de vent associée était temporairement fausse).

Phase 3 : Chargement et Fusion (L) - Le Cœur de la Robustesse

Le défi central de l'intégration est de combiner des informations provenant de sources partielles sans qu'elles s'écrasent mutuellement. Notre solution s'appuie sur la puissance de PostgreSQL.

1. Modèle de Données Cible

Toutes les données sont chargées dans la table **consolidated_measurements**. La **clé primaire composite** est définie sur l'identifiant de la turbine (turbine_id) et l'horodatage (date), garantissant l'unicité des enregistrements.

```
                    Table "public.consolidated_measurements"
       Column       |            Type             | Collation | Nullable | Default
--------------------+-----------------------------+-----------+----------+---------
 turbine_id         | character varying(5)        |           | not null |
 date               | timestamp without time zone |           | not null |
 temperature_k      | numeric                     |           |          |
 wind_ms            | numeric                     |           |          |
 vibration_mm_s     | numeric                     |           |          |
 consumption_kwh    | numeric                     |           |          |
 energie_kwh        | numeric                     |           |          |
 arret_planifie     | boolean                     |           |          | false
 arret_non_planifie | boolean                     |           |          | false
 source             | character varying(50)       |           | not null |
 extraction_ts      | timestamp with time zone    |           |          | now()
Indexes:
    "consolidated_measurements_pkey" PRIMARY KEY, btree (turbine_id, date)
```

2. Le Mécanisme de Fusion (UPSERT & COALESCE)

L'insertion est réalisée via l'opération **UPSERT** (*INSERT OR UPDATE*) en utilisant la fonction PostgreSQL **COALESCE**. Ce mécanisme assure l'**idempotence** du pipeline (le fait de l'exécuter plusieurs fois ne corrompt pas les données).

- **Processus d'Insertion :** Le pipeline tente d'insérer les nouvelles lignes (provenant du CSV, des Capteurs ou de la Météo).
- **Règle de Conflit : Si** un enregistrement existe déjà pour la même paire (turbine_id, date), l'opération se transforme en **mise à jour (UPDATE)**.
- **Logique de Non-Écrasement (COALESCE) :**
    - COALESCE(Nouvelle_Valeur, Ancienne_Valeur) est utilisée sur chaque colonne pour garantir que si la *Nouvelle_Valeur* (celle que nous insérons) est NULL (car la source est partielle), la *Ancienne_Valeur* (celle qui est déjà dans la base) est **conservée**.
    - Seul le champ source est toujours écrasé pour assurer la **traçabilité**, indiquant la dernière source ayant enrichi la ligne.

Performance : Le chargement est optimisé par l'utilisation de l'insertion par lots (*Batch Insert*), ce qui réduit considérablement le temps d'écriture en base.

Opérationnalisation et Robustesse

Un projet Data Engineering est jugé sur sa capacité à fonctionner de manière fiable et autonome.

1. Gestion des Logs et Traçabilité

Notre pipeline intègre un système de journalisation (logging) complet, configuré pour :

- **Enregistrement Persistant :** Tous les événements (INFO, WARNING, ERROR) sont écrits dans un fichier de log horodaté (pipeline.log) dans un répertoire dédié, permettant une analyse historique des opérations.
- **Aide au Diagnostic :** En cas d'échec (connexion DB, API en panne, fichier manquant), les logs permettent de remonter immédiatement à la cause sans nécessiter une analyse complexe du code.

2. Rapport d'Exécution Synthétique

À la fin de chaque exécution, le pipeline génère un rapport récapitulatif dans la console et dans les logs. Ce rapport est un indicateur de santé immédiat pour l'équipe opérationnelle :

Indicateur	Exemple de Valeur	Signification
inserted_rows	85 000	Volume de données finalement consolidé.
anomalies_corrigees	127	Nombre de valeurs aberrantes (vent/température) qui ont été mises à NULL.
csv_rows_lues	62	Nombre de lignes lues dans le fichier de production.

```
=== Pipeline terminé ===
Rapport d'exécution:
anomalies_corrigees: 0
duration_seconds: 4.0
end_time: '2025-12-10 13:53:06'
lignes_chargees: 64710
lignes_transformees: 65090
sources_extraites:
  csv_production: 62
  db_sensor: 64692
start_time: '2025-12-10 13:53:02'
status: COMPLETED_SUCCESS
```

3. Gestion de la Configuration et Sécurité

Tous les paramètres sensibles (identifiants de connexion PostgreSQL, latitude/longitude de l'API) sont externalisés dans un fichier **config.yaml**.

- **Sécurité :** Les identifiants sont gérés dans ce fichier, le rendant facile à remplacer par des secrets (variables d'environnement ou Key Vault) lors du déploiement en production.
- **Hygiène Opérationnelle :** Le pipeline assure également un **nettoyage sécurisé** des fichiers temporaires à la fin de chaque exécution, optimisant l'utilisation des ressources système.

Interface de Visualisation (Streamlit)

Le module **visualize_data.py** fournit un tableau de bord analytique simple, léger et réactif (Streamlit), permettant de valider l'impact du pipeline et de faciliter la prise de décision.

1. Monitoring des Indicateurs Clés de Performance (KPIs)

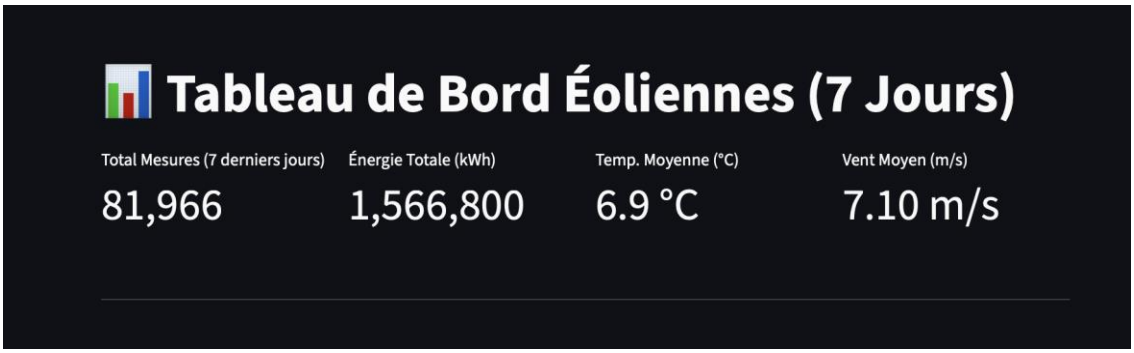
Le tableau de bord se connecte en direct à la table **consolidated_measurements** et affiche les métriques agrégées sur les 7 derniers jours :

- **Total Mesures :** L'indicateur de la quantité de données traitées et disponibles.
- **Énergie Totale (kWh) :** La production cumulée sur la période.
- **Vent Moyen (m/s) et Température Moyenne (°C) :** Les conditions environnementales générales.

2. Analyse des Tendances

Pour les analystes et les ingénieurs d'exploitation, le dashboard offre des visualisations dynamiques :

- **Tendance du Vent vs. Énergie :** Un graphique combiné permettant de corréler visuellement si les pics de production d'énergie suivent les pics de vitesse de vent.
- **Analyse de la Température :** Visualisation de la température moyenne pour identifier d'éventuels arrêts liés à des conditions climatiques extrêmes (gel, canicule).
- **Statut par Turbine :** Affichage de la production journalière de chaque turbine (T001, T002), permettant d'identifier rapidement les actifs sous-performants.



Valeur Ajoutée : Grâce au cache intégré de Streamlit, les données des 7 derniers jours sont récupérées rapidement, offrant une expérience utilisateur fluide pour le pilotage quotidien.

Évaluation Technique et Compétences Clés

Ce projet a permis de valider une maîtrise complète du cycle d'intégration des données, depuis la conception jusqu'à l'exploitation.

Validation des Compétences Évaluées (Selon le Sujet)

Compétence Évaluée	Justification de la Maîtrise
Définir les sources & outils	Choix stratégique de Python (robustesse et écosystème Data), PostgreSQL (fiabilité OLTP/OLAP), et des librairies clés (Pandas, psycopg2).
Collecter de manière sécurisée	Mise en place de trois extracteurs (CSV, DB, API) qui gèrent chacun leur protocole de manière sécurisée (gestion des chemins relatifs, des erreurs réseau et des paramètres de connexion via config.yaml).
Analyser, nettoyer, s'assurer de la qualité	Implémentation du module transform.py gérant les conversions d'unités, la normalisation des dates, et les règles de **quality_check** basées sur les limites physiques.
Construire la structure de stockage	Définition du modèle de données analytique (consolidated_measurements) avec clé primaire composite et utilisation de l'opération **UPSERT/COALESCE** pour une fusion non destructive.
Développer une interface utilisateur	Réalisation du dashboard **Streamlit** pour la visualisation des données en temps réel (KPIs, tendances), connecté directement à la base cible.

Configurer les privilèges d'accès	Bien que non implémenté en détail, l'architecture permet la création d'un utilisateur séparé (viewer_user) pour la lecture via Streamlit, respectant le **principe du moindre privilège**.

Conclusion Technique

La solution livrée est une fondation de Data Engineering **moderne et évolutive**. En s'appuyant sur des outils Open Source standardisés (Python, PostgreSQL), elle garantit à EnergiTech une indépendance technologique et des coûts de licence minimaux pour la phase d'exploitation.

Retombées Stratégiques et Perspectives IA

Notre solution n'est pas une fin en soi, mais le tremplin indispensable vers la stratégie d'innovation d'EnergiTech.

1. Retombées Immédiates pour l'Entreprise

L'implémentation de ce pipeline génère des gains opérationnels mesurables dès le premier jour :

- **Réduction des Efforts Manuels :** Le temps passé par les équipes d'analyse à consolider, nettoyer et aligner les fichiers CSV avec les mesures des capteurs est éliminé.
- **Amélioration de la Précision :** La table unique et nettoyée fournit une source fiable pour les rapports de performance, éliminant les incohérences dues à l'hétérogénéité.
- **Décision Plus Rapide :** Le dashboard Streamlit permet un diagnostic instantané des problèmes de production.

2. Perspectives d'Évolution (Feuille de Route IA)

La table consolidated_measurements est par nature un **Feature Store** prêt à l'emploi. Elle contient toutes les variables nécessaires (Vent, Température, Énergie, Vibration) pour entraîner les futurs modèles d'Intelligence Artificielle.

Prochaine Étape Recommandée : POC de Maintenance Prédictive.

Modèle IA	Objectif	Données Source
Maintenance Prédictive	Anticiper les défaillances des composants critiques (boîte de vitesses, pales) avant qu'elles ne surviennent.	**Vibration**, Température, Consommation, Historique d'Arrêts.

Optimisation de la Production	Affiner les prévisions de rendement énergétique en fonction des conditions météo.	**Vent (m/s)**, Température (K), Énergie Produite (kWh).

Notre proposition offre la base de données fiable, sécurisée et de haute qualité dont EnergiTech a besoin pour concrétiser sa vision stratégique de l'IA et maximiser la rentabilité de son parc éolien.