

# Rapport du Groupe G1

Coordination, Git et Standardisation

*Projet : Détection d'Anomalies dans la Consommation  
Électrique*

## Contexte du Projet

Système de surveillance de la consommation électrique avec détection automatique des comportements anormaux à partir de données réelles.

Année Universitaire 2025-2026  
February 3, 2026

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Contexte général du projet . . . . .	2
1.2	Place du groupe G1 dans le projet . . . . .	2
<b>2</b>	<b>Mission et Responsabilités du Groupe G1</b>	<b>3</b>
2.1	Vue d'ensemble . . . . .	3
2.2	Axes de travail détaillés . . . . .	3
2.2.1	1. Création et gestion du dépôt GitHub . . . . .	3
2.2.2	2. Définition du dictionnaire de données . . . . .	3
2.2.3	3. Définition des contrats d'interface . . . . .	4
2.2.4	4. Garantie de la cohérence globale . . . . .	5
<b>3</b>	<b>Livrables Attendus</b>	<b>6</b>
3.1	Documents de référence . . . . .	6
3.1.1	1. Dictionnaire de données . . . . .	6
3.1.2	2. Contrats d'interface . . . . .	6
3.2	Infrastructure Git . . . . .	6
3.2.1	Structure du dépôt . . . . .	6
3.2.2	Fichiers de configuration . . . . .	7
<b>4</b>	<b>Impact sur les Autres Groupes</b>	<b>8</b>
4.1	Dépendances directes . . . . .	8
4.2	Conséquences d'un travail incomplet . . . . .	8
<b>5</b>	<b>Méthodologie de Travail</b>	<b>9</b>
5.1	Phase 1 : Analyse (Semaines 1-2) . . . . .	9
5.2	Phase 2 : Infrastructure (Semaine 2) . . . . .	9
5.3	Phase 3 : Validation (Semaine 3) . . . . .	9
5.4	Phase 4 : Support (Semaines 4+) . . . . .	9
<b>6</b>	<b>Bonnes Pratiques</b>	<b>10</b>
6.1	Pour le dictionnaire . . . . .	10
6.2	Pour les contrats . . . . .	10
6.3	Pour la gestion Git . . . . .	10
6.4	Pour la coordination . . . . .	10
<b>7</b>	<b>Critères de Réussite</b>	<b>11</b>
7.1	Livrables documentaires . . . . .	11
7.2	Infrastructure technique . . . . .	11
7.3	Coordination . . . . .	11
7.4	Qualité . . . . .	11
<b>8</b>	<b>Conclusion</b>	<b>12</b>
8.1	Importance stratégique . . . . .	12
8.2	Compétences développées . . . . .	12
8.3	Perspective finale . . . . .	12

<b>A Annexe A : Exemple SQL</b>	<b>13</b>
<b>B Annexe B : Format de Commit</b>	<b>13</b>
<b>C Annexe C : Ressources</b>	<b>13</b>

# 1 Introduction

## 1.1 Contexte général du projet

Le projet vise à concevoir un système complet de surveillance de la consommation électrique capable de détecter automatiquement des comportements anormaux à partir de données réelles. Ce système s'inscrit dans une logique de systèmes décisionnels et de monitoring énergétique, similaires à ceux utilisés dans les réseaux électriques intelligents et les environnements industriels.

L'architecture globale repose sur un **pipeline de données** où chaque mesure de consommation électrique suit un cycle structuré :

1. Ingestion des données brutes
2. Stockage centralisé dans une base de données
3. Analyse des comportements normaux
4. Détection automatique des anomalies
5. Visualisation et suivi des résultats

## 1.2 Place du groupe G1 dans le projet

Le projet est organisé en **sept groupes**, chacun ayant une responsabilité spécifique. Le groupe G1 occupe une position centrale et transversale, assurant la cohérence et la coordination de l'ensemble du système.

### Rôle stratégique de G1

Le groupe G1 ne développe pas de module fonctionnel spécifique, mais il fournit l'**infrastructure organisationnelle et technique** qui permet à tous les autres groupes de travailler efficacement ensemble.

## 2 Mission et Responsabilités du Groupe G1

### 2.1 Vue d'ensemble

Le groupe G1 est responsable de trois axes majeurs :

- **Coordination** : Assurer la communication et la synchronisation entre tous les groupes
- **Gestion Git** : Créer et maintenir le dépôt GitHub centralisé
- **Standardisation** : Définir les normes et conventions communes

### 2.2 Axes de travail détaillés

#### 2.2.1 1. Crédit et gestion du dépôt GitHub

Le groupe G1 est chargé de mettre en place l'infrastructure de contrôle de version qui servira de base collaborative pour l'ensemble du projet.

**Responsabilités principales :**

- Crédit du dépôt GitHub principal
- Configuration des droits d'accès pour les différents groupes
- Définition de la structure des branches
- Mise en place des règles de contribution
- Gestion des conflits de fusion
- Maintenance de la documentation

**Bonnes pratiques attendues :**

- Stratégie de branches claire (Git Flow ou GitHub Flow)
- Conventions de nommage des commits
- Processus de revue de code
- Protection des branches principales

#### 2.2.2 2. Définition du dictionnaire de données

Le dictionnaire de données est un document de référence crucial qui définit précisément la structure de la base de données PostgreSQL.

### Contenu du dictionnaire :

- Nom exact de chaque table
- Liste exhaustive des colonnes avec leurs types SQL
- Contraintes (clés primaires, clés étrangères, NOT NULL, UNIQUE)
- Description sémantique de chaque champ
- Format des données (timestamps, encodages, unités)

#### Table power\_consumption

- `id` : SERIAL PRIMARY KEY
- `timestamp` : TIMESTAMP NOT NULL
- `global_active_power` : FLOAT
- `global_reactive_power` : FLOAT
- `voltage` : FLOAT
- `anomaly_score` : FLOAT
- `is_anomaly` : BOOLEAN
- `created_at` : TIMESTAMP DEFAULT NOW()

### Exemple de structure :

#### 2.2.3 3. Définition des contrats d'interface

Les contrats d'interface définissent les règles d'interaction entre les différents modules du système.

### Contenu des contrats :

- **Contrat de lecture** : Quelles données chaque groupe peut lire
- **Contrat d'écriture** : Quelles données chaque groupe peut modifier
- **Règles de synchronisation** : Ordre d'exécution des modules
- **Gestion des erreurs** : Comportement en cas d'échec
- **Format des échanges** : Structure des données partagées

**Contrat G2 vers Base de données****Module source :** G2 (Ingestion)**Action :** Insertion de données**Règles :**

- Insertion uniquement dans `power_consumption`
- Champs obligatoires : `timestamp`, variables de consommation
- Champs interdits : `anomaly_score`, `is_anomaly`
- Format timestamp : ISO 8601

**Exemple de contrat :****2.2.4 4. Garantie de la cohérence globale**

Le groupe G1 joue un rôle de supervision et d'arbitrage.

**Actions de coordination :**

- Réunions de synchronisation inter-groupes
- Résolution des conflits techniques
- Validation de la compatibilité des modules
- Suivi de l'avancement global
- Mise à jour des documents de référence

## 3 Livrables Attendus

### 3.1 Documents de référence

Le groupe G1 doit produire et maintenir deux documents essentiels :

#### 3.1.1 1. Dictionnaire de données

Fichier : docs/data\_dictionary.md

**Structure recommandée :**

1. Introduction générale
2. Vue d'ensemble de la base de données
3. Description détaillée de chaque table
4. Schéma de relations
5. Conventions de nommage
6. Glossaire des termes métier

#### 3.1.2 2. Contrats d'interface

Fichier : docs/interface\_contracts.md

**Structure recommandée :**

1. Principe général des contrats
2. Matrice des interactions
3. Contrat détaillé pour chaque groupe
4. Règles de cohérence temporelle
5. Gestion des dépendances

### 3.2 Infrastructure Git

#### 3.2.1 Structure du dépôt

Le dépôt doit être organisé de manière claire :

```
sdid-energy-anomaly-2025-2026/
  README.md
  docs/
    data_dictionary.md
    interface_contracts.md
  modules/
    g2-ingestion/
```

```
g3-analysis/  
g4-detection/  
g5-dashboard/  
g6-deployment/  
g7-drift/  
scripts/  
database/  
    init.sql  
.github/  
workflows/
```

### 3.2.2 Fichiers de configuration

- `.gitignore` : Exclusion des fichiers sensibles
- `CONTRIBUTING.md` : Guide de contribution
- `.github/PULL_REQUEST_TEMPLATE.md` : Modèle de PR

## 4 Impact sur les Autres Groupes

### 4.1 Dépendances directes

Tous les groupes dépendent du travail de G1 :

- **G2** : Dictionnaire pour la structure de table
- **G3** : Dictionnaire pour lire les données
- **G4** : Contrat pour mettre à jour les scores
- **G5** : Dictionnaire pour l'affichage
- **G6** : Tous les documents pour le déploiement
- **G7** : Dictionnaire et contrat pour l'analyse

### 4.2 Conséquences d'un travail incomplet

Si le groupe G1 ne livre pas ses documents :

- **Blocage total** : Impossibilité de démarrer
- **Incohérences** : Choix incompatibles
- **Refactoring massif** : Réécriture nécessaire
- **Conflits** : Impossibilité d'intégration

#### Point critique

Le groupe G1 est sur le **chemin critique** du projet. Tout retard impacte directement l'ensemble.

## 5 Méthodologie de Travail

### 5.1 Phase 1 : Analyse (Semaines 1-2)

1. Étudier les données sources
2. Analyser les besoins de chaque groupe
3. Concevoir le schéma de base de données
4. Rédiger le dictionnaire v1
5. Définir les contrats

### 5.2 Phase 2 : Infrastructure (Semaine 2)

1. Créer le dépôt GitHub
2. Configurer les accès
3. Définir la stratégie de branches
4. Rédiger CONTRIBUTING.md
5. Créer le script SQL

### 5.3 Phase 3 : Validation (Semaine 3)

1. Présenter les documents
2. Recueillir les retours
3. Ajuster les documents
4. Publier la version validée
5. Former les groupes à Git

### 5.4 Phase 4 : Support (Semaines 4+)

1. Répondre aux questions
2. Mettre à jour les documents
3. Gérer les pull requests
4. Résoudre les conflits
5. Assurer la cohérence

## 6 Bonnes Pratiques

### 6.1 Pour le dictionnaire

- Utiliser des types SQL standards
- Documenter les unités de mesure
- Prévoir des champs de traçabilité
- Anticiper l'évolution
- Ajouter des exemples

### 6.2 Pour les contrats

- Être précis (pas d'ambiguïté)
- Prévoir les cas d'erreur
- Définir qui fait quoi et quand
- Éviter les dépendances circulaires
- Documenter les cas limites

### 6.3 Pour la gestion Git

- Protéger la branche main
- Imposer les pull requests
- Utiliser des commits atomiques
- Maintenir un README à jour
- Utiliser les issues GitHub

### 6.4 Pour la coordination

- Organiser des réunions régulières
- Tenir un journal des décisions
- Être réactif aux questions
- Anticiper les problèmes
- Documenter les changements

## 7 Critères de Réussite

### 7.1 Livrables documentaires

- Dictionnaire complet et à jour
- Contrats validés par tous
- README clair et informatif
- Guide de contribution

### 7.2 Infrastructure technique

- Dépôt GitHub configuré
- Tous les groupes ont accès
- Structure de dossiers cohérente
- Script SQL fonctionnel

### 7.3 Coordination

- Aucun conflit bloquant
- Tous les groupes comprennent leur rôle
- Communication fluide
- Problèmes résolus rapidement

### 7.4 Qualité

- Documentation claire
- Conventions respectées
- Code review systématique
- Historique Git propre

## 8 Conclusion

### 8.1 Importance stratégique

Le groupe G1 joue un rôle **fondamental** dans la réussite du projet. Sans standardisation et coordination, il serait impossible d'assembler les modules en un système cohérent.

### 8.2 Compétences développées

Le groupe G1 développe des compétences essentielles :

- **Gestion de projet** : Coordination d'équipes
- **Architecture système** : Conception d'interfaces
- **DevOps** : Gestion du versioning
- **Documentation** : Rédaction de spécifications
- **Communication** : Animation de la collaboration

### 8.3 Perspective finale

Le succès du projet repose sur la qualité du travail de G1. Un dictionnaire bien conçu, des contrats clairs et une infrastructure Git solide sont les **fondations** du système.

#### Message à G1

Votre travail est moins visible que celui des autres groupes, mais il est **absolument essentiel**. La qualité de l'ensemble dépend de votre rigueur. Prenez le temps de bien faire les choses dès le départ !

## A Annexe A : Exemple SQL

Exemple de script de création de table :

```
CREATE TABLE power_consumption (
    id SERIAL PRIMARY KEY,
    timestamp TIMESTAMP NOT NULL UNIQUE,
    global_active_power FLOAT,
    global_reactive_power FLOAT,
    voltage FLOAT,
    global_intensity FLOAT,
    sub_metering_1 FLOAT,
    sub_metering_2 FLOAT,
    sub_metering_3 FLOAT,
    anomaly_score FLOAT DEFAULT NULL,
    is_anomaly BOOLEAN DEFAULT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_timestamp
    ON power_consumption(timestamp);
CREATE INDEX idx_is_anomaly
    ON power_consumption(is_anomaly);
```

## B Annexe B : Format de Commit

**Bon exemple :**

```
feat(docs): add data dictionary

- Define power_consumption table
- Add constraints and indexes
- Include examples
```

**Mauvais exemple :**

```
update
```

## C Annexe C : Ressources

- Git Flow : [nvie.com/posts/a-successful-git-branching-model/](http://nvie.com/posts/a-successful-git-branching-model/)
- Conventional Commits : [conventionalcommits.org](https://conventionalcommits.org)
- PostgreSQL : [postgresql.org/docs/](https://www.postgresql.org/docs/)
- Markdown Guide : [markdownguide.org](https://markdownguide.org)