

ESUP-DAYS (41)

MIGRATION SIFAC VERS SIFAC+ UNE BASE INTERMÉDIAIRE POUR L'INTEROPÉRABILITÉ

WEBER Stéphane – Chef de projet Sifac+
GONON Emmanuel – Développeur
HENRION Olivier – Responsable exploitation Sifac



**UNIVERSITÉ
DE LORRAINE**

CONTEXTE
LE PROJET DISTRIBUÉ
NOTRE BASE ET SON UTILISATION

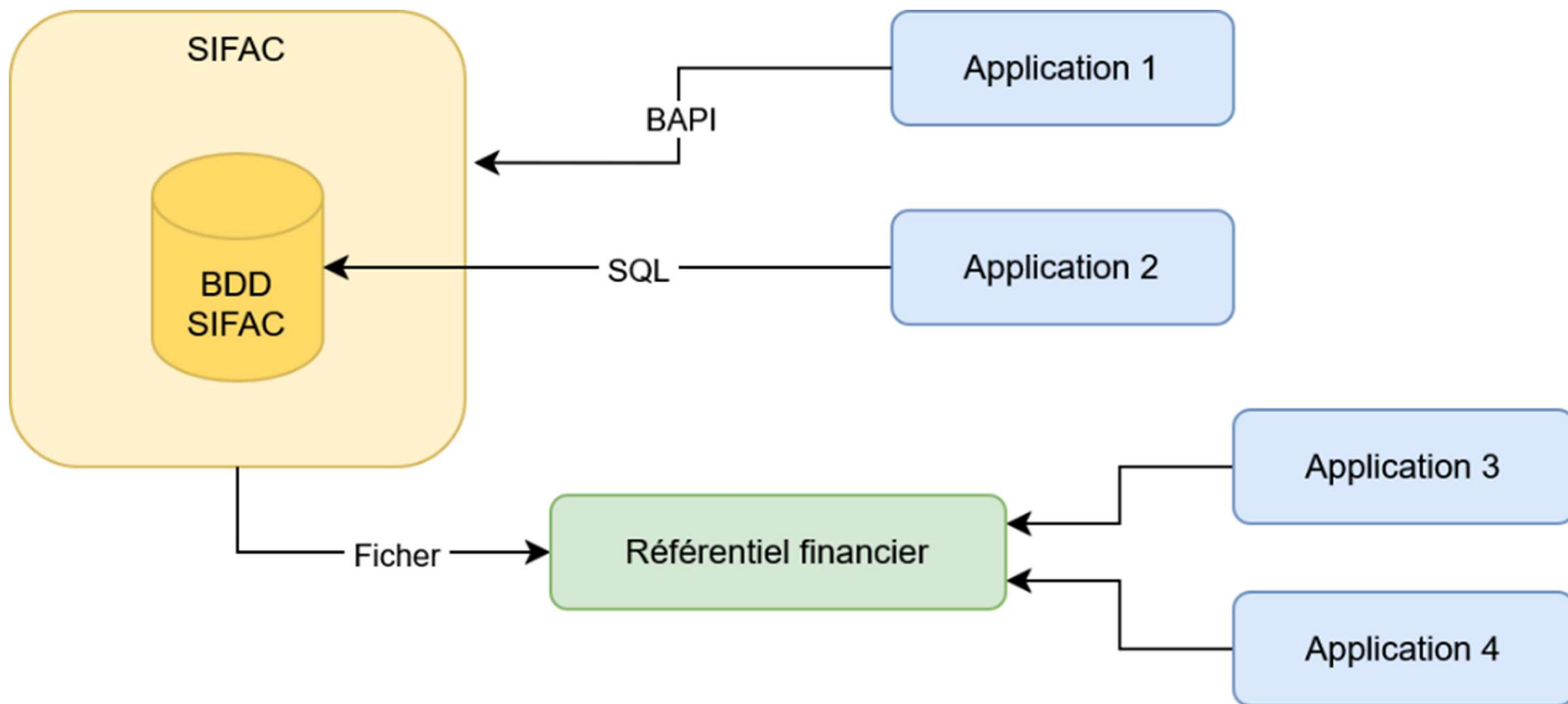
CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

L'INTEROPÉRABILITÉ SUR SIFAC À L'UL

L'INTEROPÉRABILITÉ SUR SIFAC À L'UL



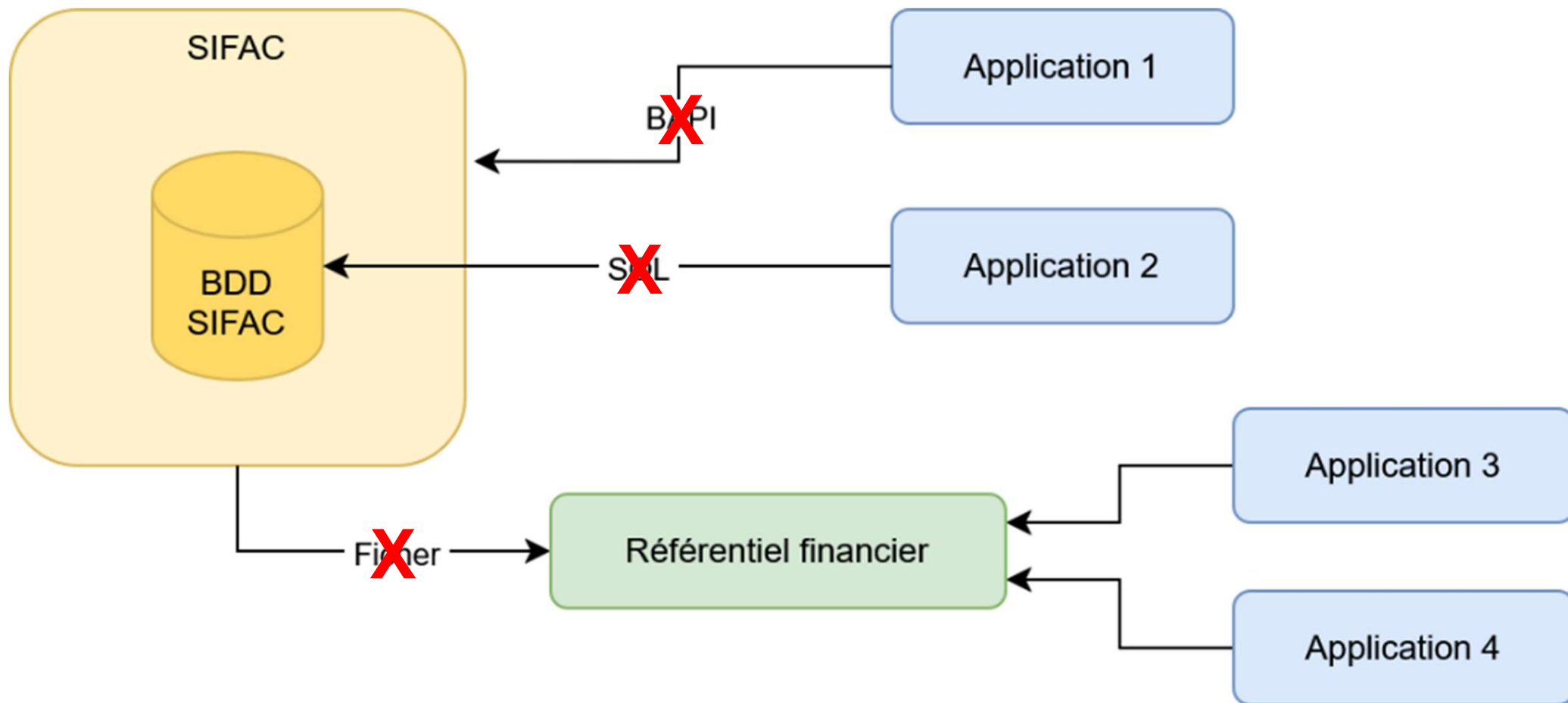
CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

LES LIMITES OFFERTES AVEC SIFAC+ SAAS À L'AMUE

LES LIMITES OFFERTES AVEC SIFAC+: SAAS À L'AMUE



CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

PROPOSITIONS API AMUE

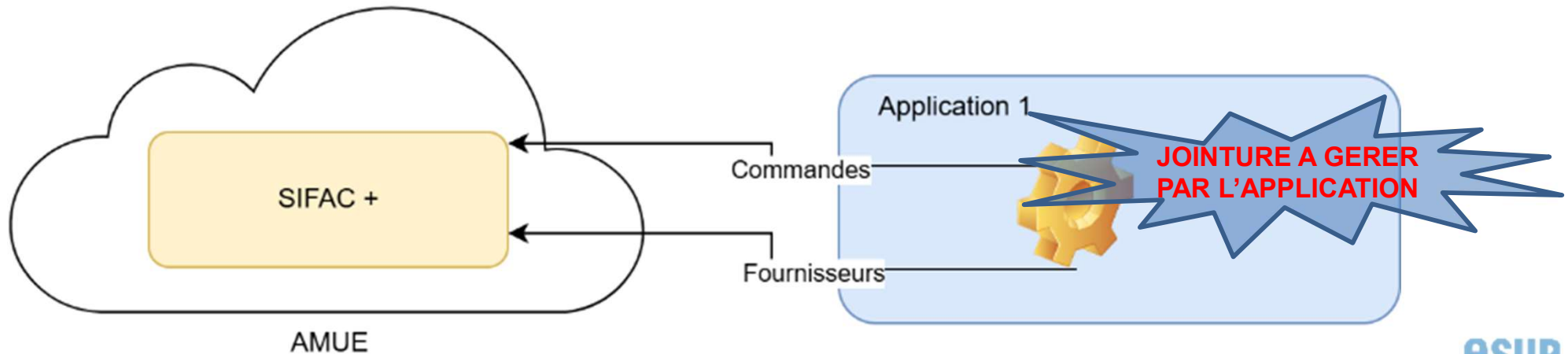
PROPOSITIONS API AMUE

- Des API propres aux applications de l'AMUE (SIHAM, FCA, PayFip...)
- Des API pour les échanges externes (INFINOE, CHORUS PRO, ...)
- Une API pour les pièces jointes
- Des API sur des « tables » pour consulter les données à J-1

PROPOSITIONS API AMUE

Des « tables » à J-1 accessibles en REST.
Les tables sont appelées des CDSVIEWS.
(pas toujours une table et des fois un peu de jointure)

X Mais forte dégradation des performances pour les jointures



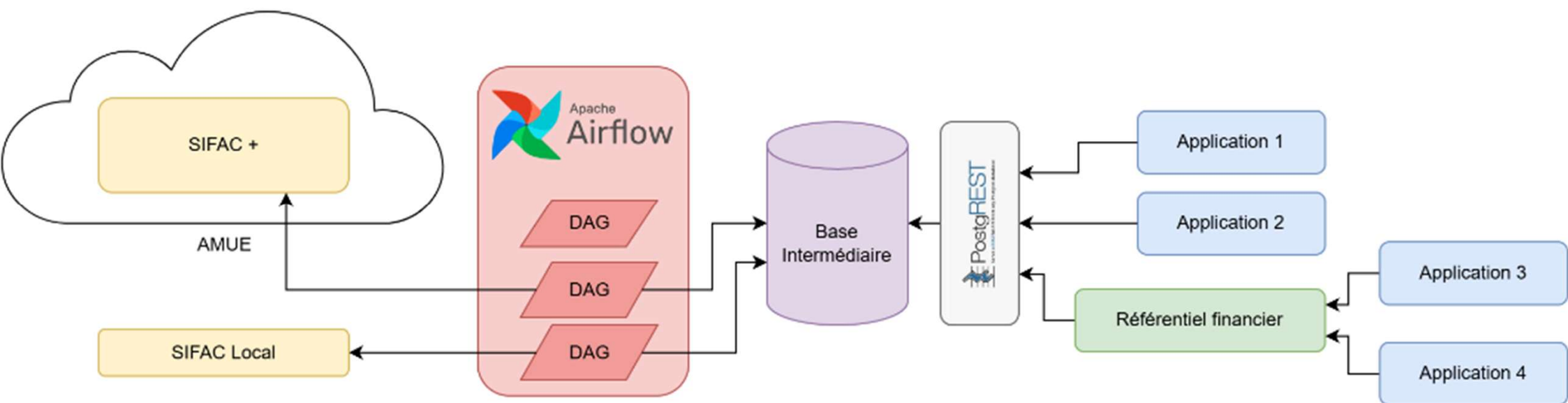
CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

LA SOLUTION: UNE BASE INTERMÉDIAIRE

LA SOLUTION: UNE BASE INTERMÉDIAIRE



CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

AIRFLOW

AIRFLOW

Principe

- Orchestrateur de workflows de données
- Planifie, surveille et relance des DAGs
- Concept central : le DAG (enchaînement de tâches avec dépendances)
- Interface web de monitoring
- Retry automatique, alertes email

Open-source, créé par Airbnb (2014), projet top-level Apache depuis 2019

AIRFLOW

Bienvenue

Statistiques

- Dags échoués: 1
- Dags en cours d'exécution: 0
- Dags actifs: 4

10 premiers Dags favoris

amue_multi_table_import

État de santé

- Base de données Meta: ✓
- Planificateur: ✓
- Déclencheur: ✓
- Analyseur de Dag: ✓

Emplacements de pool: 128

Historique: 24 dernières heures (2026-03-25 14:25:34 - 2026-03-26 14:25:34)

4 Exécutions de Dag

Statut	Nombre	Pourcentage
En file	0	0.00%
En cours	0	0.00%
Succès	3	75.00%
Échoué	1	25.00%

106 Instances de tâche

Statut	Nombre	Pourcentage
Succès	100	94.34%

Événement d'Asset: 0 (Aucun événement d'actif trouvé.)

AIRFLOW

Dag
amue_multi_table_import

Options ▾

amue_multi_table_import

Planification 0.2*** Dernière exécution 2026-03-26 03:00:00 Prochaine exécution 2026-03-27 03:00:00 Propriétaire airflow Tags amue, production Dernière version du Dag v12

Documentation du Dag Retirer le Dag des favoris

Aperçu Runs Tâches Calendrier Rattrapages Journal d'audit Code Détails

24 dernières heures 2026-03-25 14:28:04 - 2026-03-26 14:28:04

0 Tâche échouée 0 Run échoué

Dernières 10 exécutions du Dag

Durée

Durée en file d'attente
Durée d'exécution

11:06:40
05:33:20
0
01:30:47

2026-03-19 03:00:00 2026-03-22 03:00:00 2026-03-25 03:00:00

Exécuté après

- 1 nœud = 1 tâche

- 1 flèche = 1 dépendance

CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

LES PARAMÈTRES

LES PARAMÈTRES

	A-Z table_name ↑	<input checked="" type="checkbox"/> enabled	A-Z primary_key	A-Z delta	A-Z fingerprint_api	A-Z fingerprint_local	updated_at	A-Z ecc_query	A-Z setup_status
1	ADR6	[v]	addrnumber,persnur		aad1a96a3350246c1fb3e	02e6100c87c12a747609d7	02:00:22.882 +0200	[NULL]	ready
2	AGR_1251	[v]	agr_name,counter		dd233adb02c7771fd493	e706a529c9524fa32ab35c7	02:00:22.734 +0200	[NULL]	ready
3	AGR_AGRS	[v]	agr_name,child_agr		1ccf09a31324c15bb9e7c	7d2fd0cbac237c03a09387	02:00:23.298 +0200	SELECT agr_name,	ready
4	AGR_TCODES	[v]	agr_name,type,tcode		e8994ea813c8d8f63dbf4	328efe8f07e4037dc946307	02:00:23.222 +0200	[NULL]	ready

Liste des tables à importer (en base, pas en config)

- Nom de la table
- Tables à traiter ou non
- Colonne delta
- Query SQL pour les données SIFAC

LES PARAMÈTRES

- Variables Airflow :
 - Plannings,
 - Seuils retry,
 - Destinataires mail
 - ...

<code>amue_api_max_retries</code>	3
<code>amue_api_retry_delay_seconds</code>	30
<code>amue_force_import</code>	true
<code>amue_import_batch_size</code>	5000
<code>amue_import_parallel_workers</code>	1
<code>amue_import_schedule</code>	0 2 * * *
<code>amue_max_wait_hours</code>	6
<code>amue_monitor_schedule</code>	0 22 * * *
<code>amue_polling_exponential_backoff</code>	false

LES PARAMÈTRES

<input type="checkbox"/>	ID de connexion	Type de connexion	Description	Hôte	Port	
<input type="checkbox"/>	oauth_api	http		https://sandbox.api.amue.fr		  
<input type="checkbox"/>	oracle_data	odbc	Connexion ecc	[REDACTED]		  
<input type="checkbox"/>	postgres_data	postgres		[REDACTED]		  

Connexions Airflow :

- API
- Base intermédiaire
- Base de données Sifac

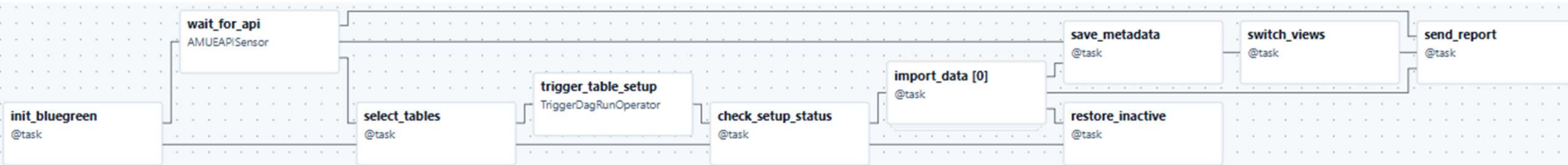
CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

1 – ALIMENTATION DEPUIS SIFAC+

1 – ALIMENTATION DEPUIS SIFAC+

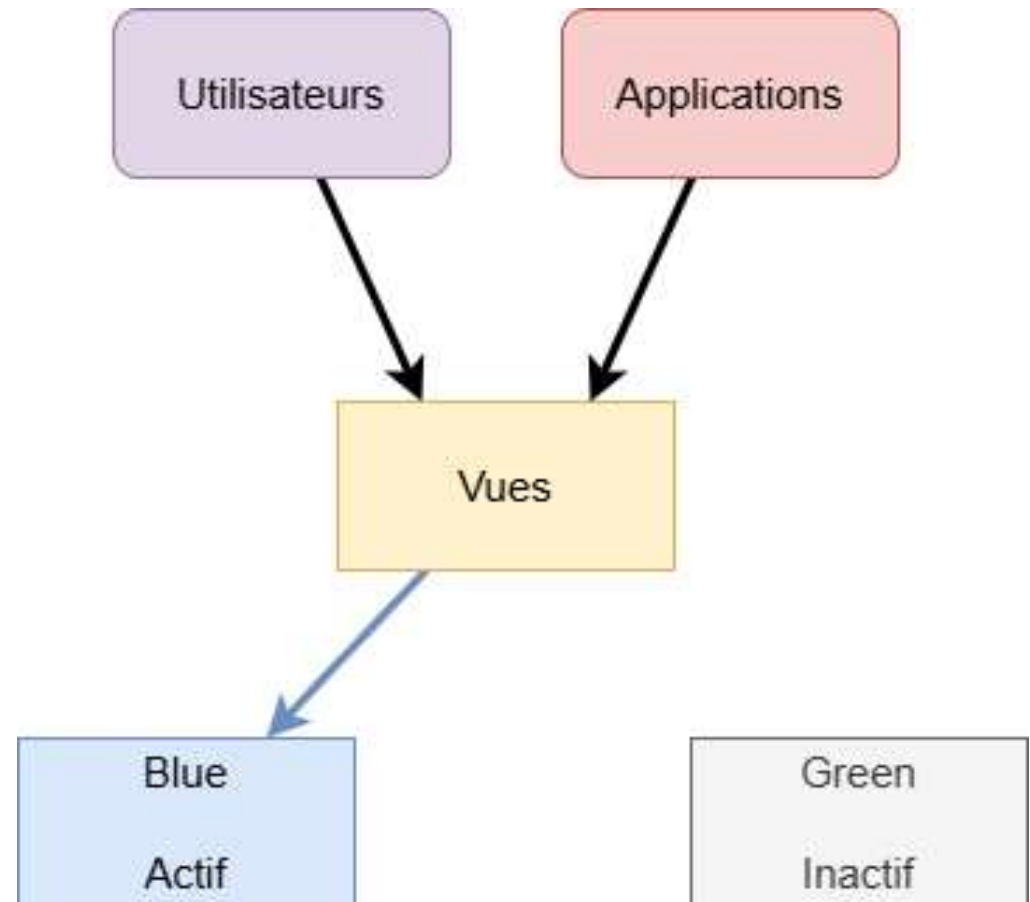


Workflow

1. Initialise le Blue/Green
2. Attend les données à J-1 de l'AMUE
3. Prépare les tables
4. Importe les données
5. Enregistre les métadonnées
6. Switch les vues
7. Envoie un rapport d'exécution

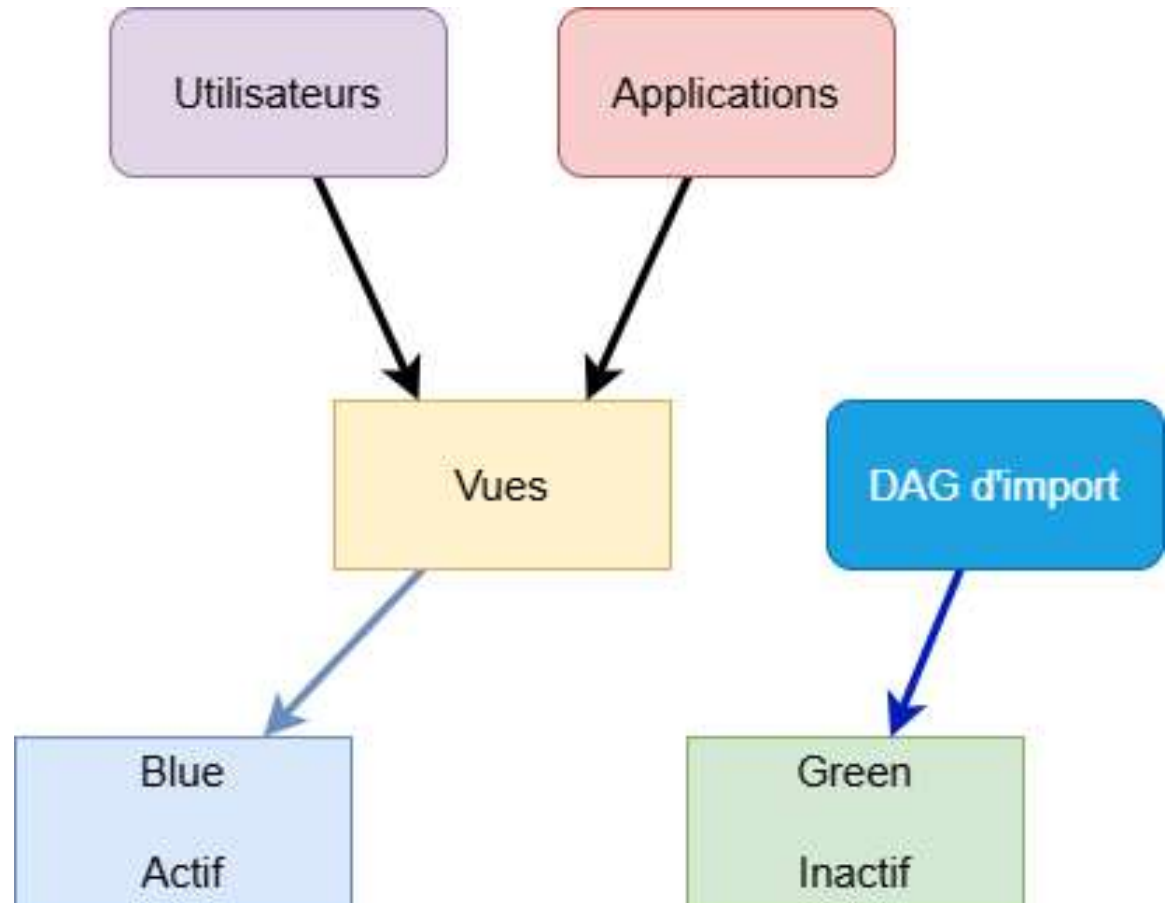
1 – ALIMENTATION DEPUIS SIFAC+

- 2 schémas : Blue (actif) et Green (inactif)
- 1 schéma de vues : point d'entrée des applications
- Le schéma inactif est la cible des écritures du prochain import



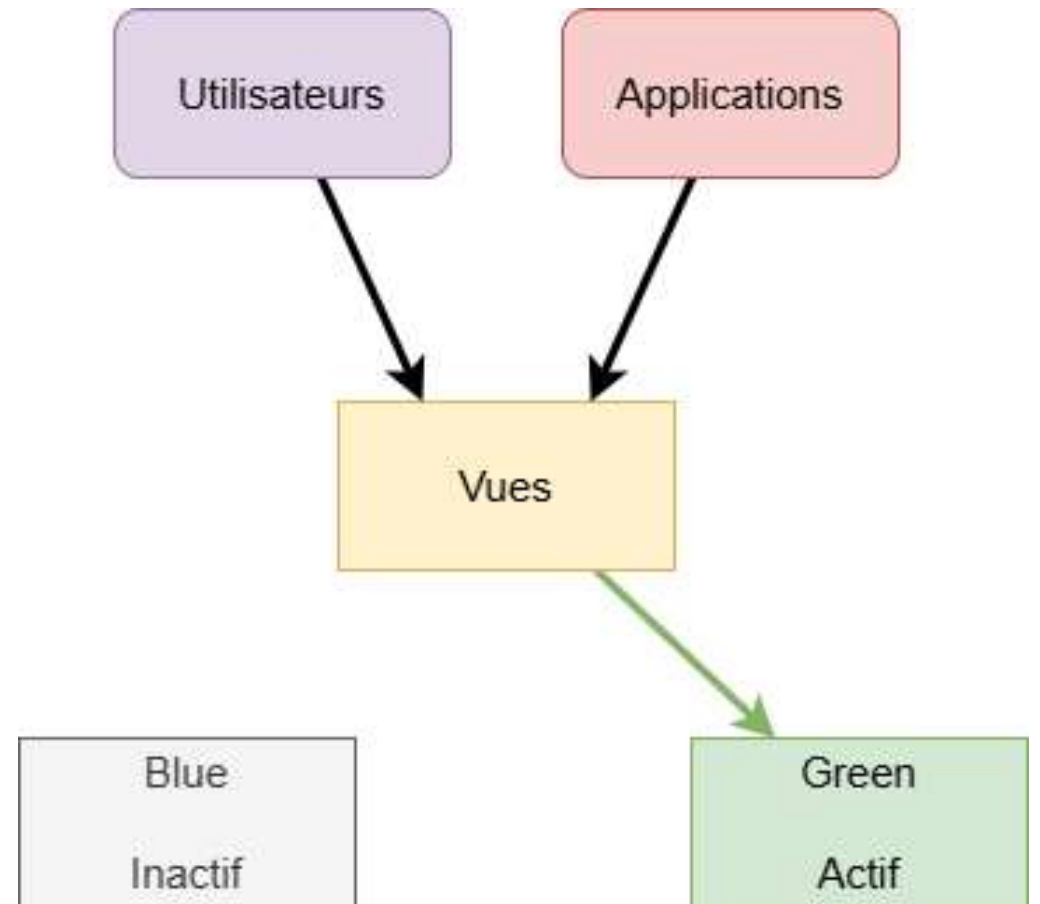
1 – ALIMENTATION DEPUIS SIFAC+

- Écriture dans le schéma inactif (green)
- Lectures applicatives continuent sur l'actif
- Polling côté API + import en parallèle par table
- Si l'import échoue : on n'a rien cassé, on rejoue



1 – ALIMENTATION DEPUIS SIFAC+

- Switch atomique : DROP + CREATE des vues dans une seule transaction
- À la fin du switch : Green devient l'actif, Blue l'inactif
- Bascule instantanée vue côté applis : pas de coupure, pas de demi-état
- Le schéma précédent reste intact



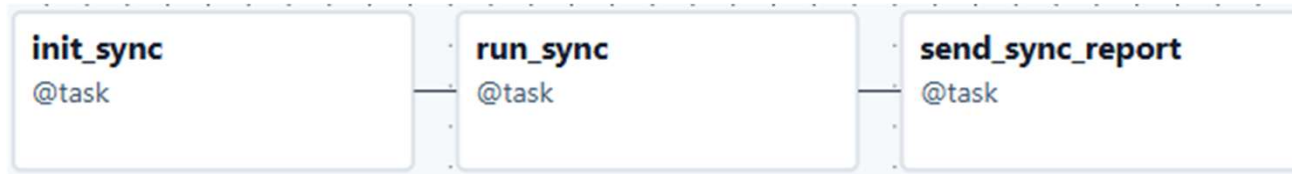
CONTEXTE

LE PROJET DISTRIBUÉ

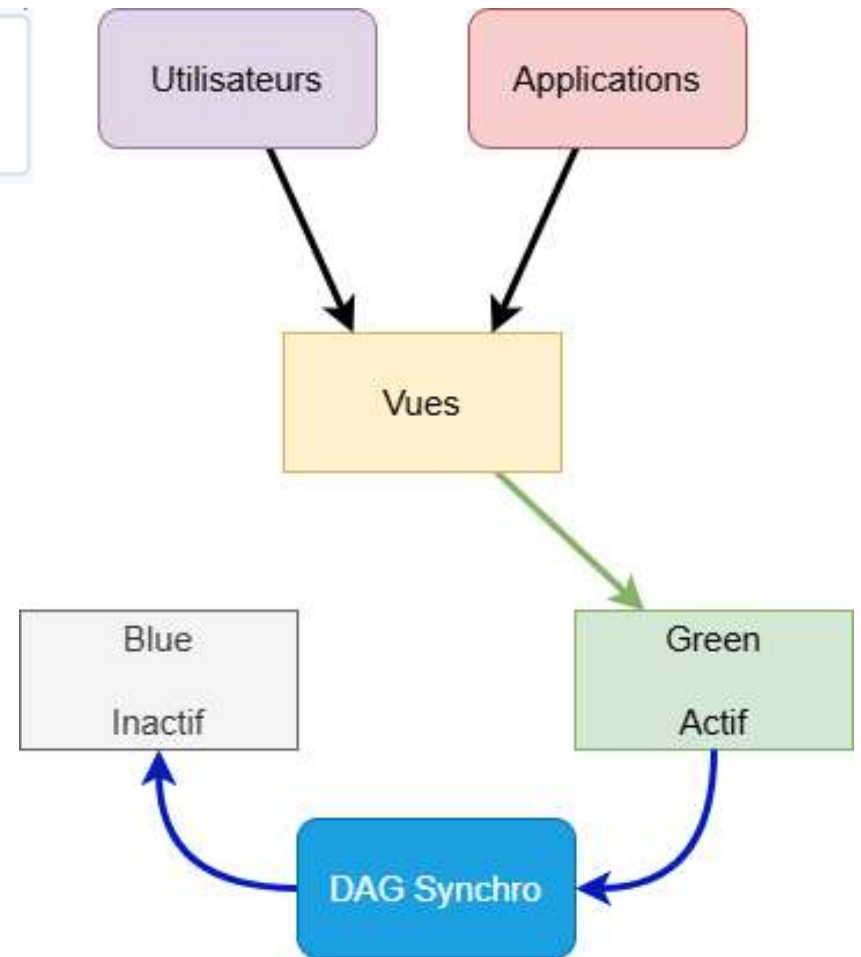
NOTRE BASE ET SON UTILISATION

2 – DAG SYNCHRONISATION BLUE/GREEN

2 – DAG SYNCHRONISATION BLUE/GREEN



- **Source** = schéma **actif**
Cible = schéma **inactif**
- TRUNCATE + INSERT par table, commit par table
- Rapport mail : tables, lignes copiées, erreurs



CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

3 – CRÉATION/VÉRIFICATION DES STRUCTURES DES TABLES

3 – CRÉATION/VÉRIFICATION DES STRUCTURES DES TABLES



Workflow

1. Récupère toutes les tables paramétrées à traiter
2. Vérifie/Compare les structures existantes avec l'API (S+)
 1. Créer un hash avec les structures de l'API (S+)
 2. Compare avec le hash existant, sinon l'insert en base
 3. Créer les tables
3. Agrège les résultats dans un rapport

3 – CRÉATION/VÉRIFICATION DES STRUCTURES DES TABLES

- Détection automatique
- Statut bloqué
- Mail explicite
- Action attendue : revue par un exploitant, validation manuelle

Anomalie Setup AMUE

2026-04-23 08:45

Setup Incomplet — Action Requise

1 table(s) bloquée(s) (changement de structure détecté)

DAG :

amue_table_setup

Statut :

INCOMPLET

Tables Bloquées — Structure Modifiée

Ces tables ne peuvent pas être importées tant que leur structure n'a pas été vérifiée et validée.

Table	FP API	FP Config	Cause probable
ADR6 16 col.	MODIFIÉ	INCHANGÉ	Types ou colonnes côté API uniquement

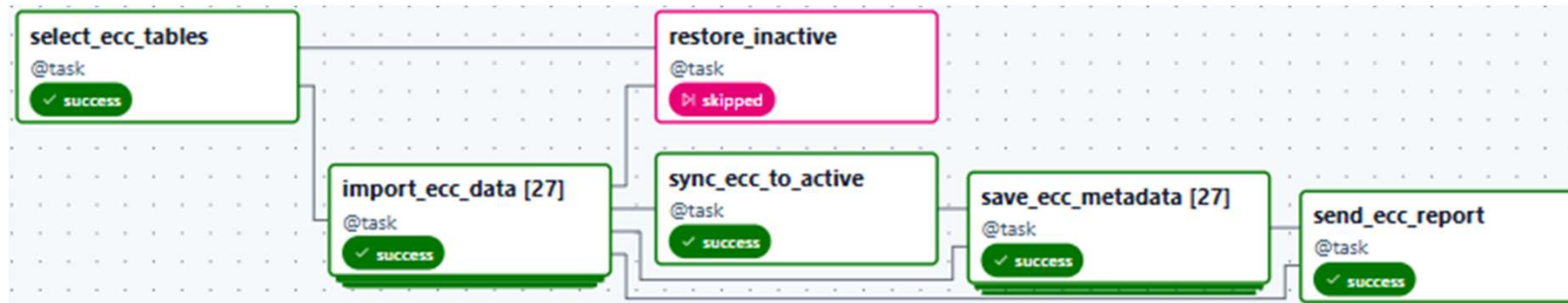
CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

4 – ALIMENTATION DEPUIS ECC (SIFAC ACTUEL: ON PREMISE)

4 – ALIMENTATION DEPUIS ECC (SIFAC ACTUEL: ON PREMISE)



Workflow

1. Récupère toutes les tables ayant une requête ECC configurée
2. Récupère et insert dans le schéma inactif
3. Synchronise avec le schéma actif
4. Agrège les résultats dans un rapport

4 – ALIMENTATION DEPUIS ECC (SIFAC ACTUEL: ON PREMISE)

- Récapitulatif par table
- Statut global
- Destinataires configurables

Import ECC Réussi

2026-04-21 09:10

Import ECC Terminé avec Succès

DAG :
ecc_multi_table_import
Date :
2026-04-21 09:10
Durée :
9s
Tables :
1 table(s)
Lignes source :
3,966 ligne(s) récupérée(s)
Lignes DB :
0 ligne(s) insérée(s)
Mises à jour :
877 ligne(s) mise(s) à jour
Ignorées :
3,089 ligne(s) protégée(s) sifac_plus
Statut :
ECC SUCCÈS

Détail des tables importées

Table	Récupérées	Insérées	MAJ	Ignorées	Type	Statut
AGR_AGRS	3,966	0	877	3,089	full	SUCCESS

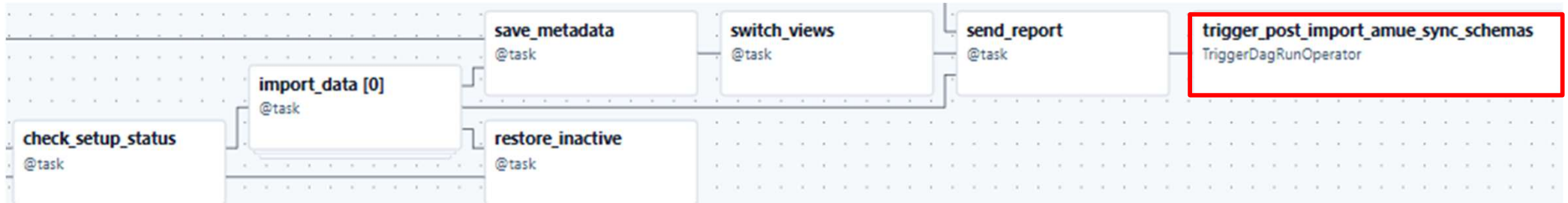
CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

AJOUTER UN DAG

AJOUTER UN DAG



- Deux variables Airflow :
 - amue_pre_import_dags
 - amue_post_import_dags
- Aucune modification de code
- Exécution séquentielle

```
amue_post_import_dags      ["amue_sync_schemas"]  
amue_pre_import_dags      []
```

CONTEXTE
LE PROJET DISTRIBUÉ
NOTRE BASE ET SON UTILISATION

LES CHIFFRES

LES CHIFFRES

- 33 tables identifiées stockant le référentiel financier et 10 années d'exercice comptables et budgétaires (2016-2026)
 - Top tables :
 - Ecritures budgétaires ~50.000.000 enregistrements
 - Ecritures comptables ~7.000.000 enregistrements
 - Données budgétaires ~1.600.000 enregistrements
- 100 Go
- 17 vues applicatives (à terme 34)

CONTEXTE
LE PROJET DISTRIBUÉ
NOTRE BASE ET SON UTILISATION

POSTGREST

POSTGREST

Transformer une base PostgreSQL en API REST

PostgREST en une phrase

Génère automatiquement une API REST à partir d'une base PostgreSQL -> Pas de backend à développer

La base devient le backend

- Tables → endpoints REST
- Vues → endpoints personnalisés

Architecture simple

Client → HTTP → PostgREST → PostgreSQL

Installation et configuration facile

Télécharger `postgrest-v14.1-linux-static-x86-64.tar.xz`, décompresser dans un dossier, configurer `postgrest.conf`, lancer

Postgrest.conf

```
db-uri = "postgres://authenticator:user@localhost:5432/password"  
db-schemas = "splus"  
db-anon-role = "anon"  
server-port = 3000  
db-max-rows = 500
```

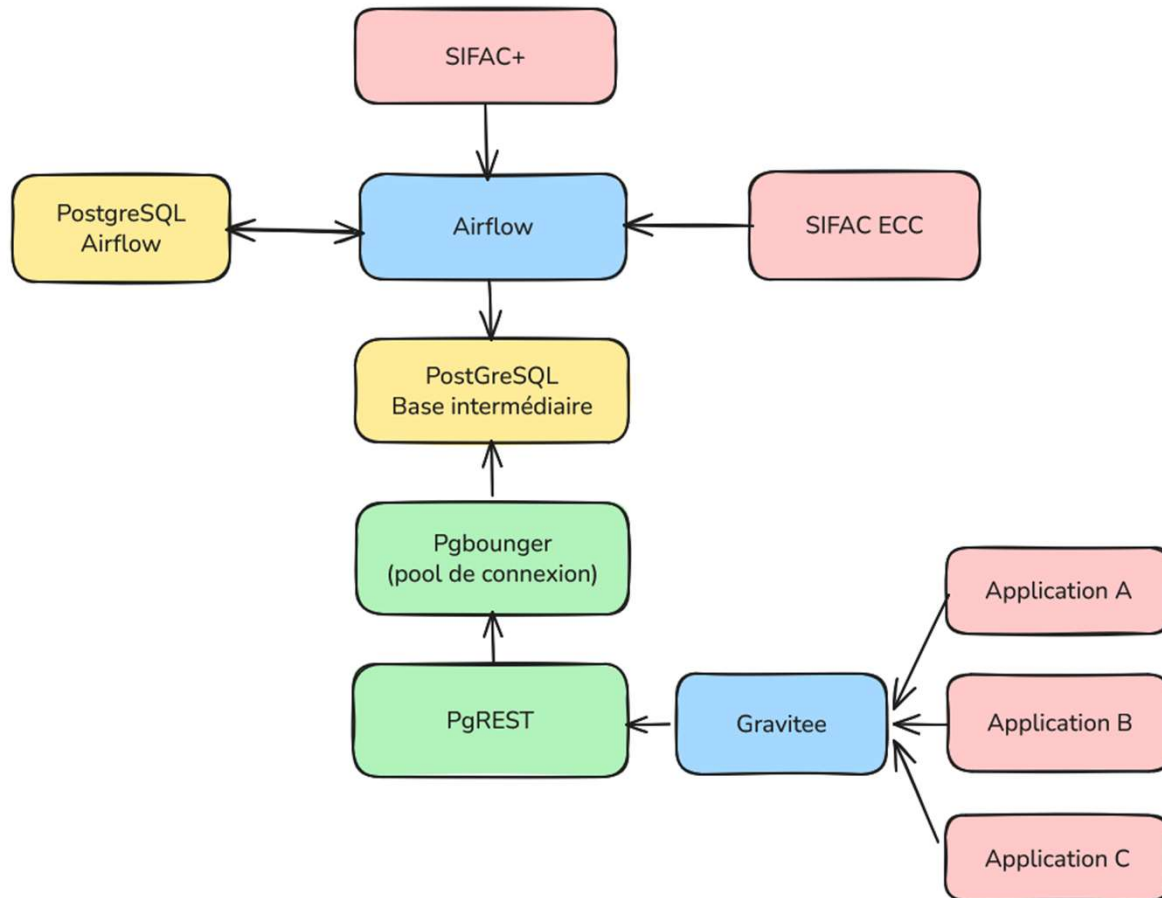
POSTGREST (CRUD)

Exemple: récupérer le(s) client(s) qui s'appelle(nt) EDF

GET /kna1?name1=eq.EDF

```
▼ 0:  
kunnr:      "731 ████████ "  
land1:      "FR "  
name1:      "EDF          "  
name2:      "              "  
ort01:      "CHALETTE SUR LOING CEDEX "  
pstlz:      "45123      "  
regio:      "          "  
sort1:      "EDF SA     "  
stras:      "TSA 50008   "  
telf1:      "              "  
telfx:      "              "  
xcpdk:      0  
adrnr:      "737094     "  
mcod1:      "EDF          "  
mcod2:      "              "  
mcod3:      "CHALETTE SUR LOING CEDEX "  
anpad:      "              "
```

ARCHITECTURE DOCKER UL



CONTEXTE

LE PROJET DISTRIBUÉ

NOTRE BASE ET SON UTILISATION

LA DISTRIBUTION ESUP ET QUELS PILOTES

LA DISTRIBUTION ESUP ET QUELS PILOTES

Distribution:
à la rentrée

<https://www.esup-portail.org/wiki/x/BoDOZ>

Pilotes:

LYON 1 (déjà en production Sifac+) à partir de mai 2026

CAEN (vague 2028) à la fin de l'année

