

RAPPORT DE STAGE

RATP



Stagiaire : Pierre FRICHET

Tuteur : Olivier ELISABETH

Du 28 mai au 30 avril

SOMMAIRE

Introduction.....	3
I - L'Entreprise.....	3
1 - RATP / RDS.....	3
A) LA RATP.....	3
B) RDS.....	4
2 – Exploitation.....	4
3 - Unité DOSI.....	5
4 - SAE.....	5
.....	7
1 – Collecte et préparation des données.....	7
a) Extraction des missions théoriques depuis la base SAE-IP.....	7
.....	8
b) Récupération des missions réalisées depuis les archives SAE-TR.....	8
2 – Identification des missions inconnues.....	10
.....	10
a) L'Identification.....	10
b) La synthèse.....	11
3 – Envoi des rapports par mail.....	12
4 – Archivage et automatisation.....	13
5 – Journalisation.....	15
a) Résultats.....	17
b) Difficultés.....	19
Conclusion.....	19
Remerciements.....	20
Annexes.....	21

Introduction

Dans le cadre de ma première année de BTS Services Informatiques aux Organisations (SIO), spécialité SISR (Solutions d'Infrastructure, Systèmes et Réseaux), j'ai effectué un stage de 5 semaines au sein de la RATP (Régie Autonome des Transports Parisiens), acteur majeur du transport public en Île-de-France.

Ce stage, réalisé du 28 avril au 30 mai, avait pour objectif de me familiariser avec le monde professionnel tout en mettant en pratique les compétences acquises en formation. Intégrer une structure aussi vaste et exigeante que la RATP m'a permis de découvrir les réalités techniques et organisationnelles d'un système d'information complexe, au cœur d'enjeux stratégiques comme la disponibilité et la performance des services numériques.

Bien que ma spécialité soit axée sur les infrastructures réseau et systèmes, les missions confiées ont principalement touché au développement de scripts en Bash, dans une optique d'automatisation de tâches. Ce positionnement hybride m'a offert l'opportunité de renforcer mes compétences en administration système, tout en abordant des notions liées au développement, à la logique de script, et à la conception d'application.

I - L'Entreprise

1 - RATP / RDS

A) LA RATP

La RATP (Régie Autonome des Transports Parisiens) est un établissement public à caractère industriel et commercial (EPIC), créé en 1949. Elle est l'un des principaux opérateurs de transports publics en France et dans le monde. Son activité principale consiste à exploiter, entretenir et moderniser les réseaux de transport en commun en Île-de-France, notamment les lignes de métro, de RER, de tramway et de bus. Chaque jour, plus de 12 millions de voyageurs utilisent ses services. Mais au-delà de son rôle de transporteur, la RATP est également une entreprise fortement engagée dans



le développement technologique, l'innovation et la transition numérique. Elle dispose de services informatiques internes performants, qui assurent la gestion, la sécurisation et l'évolution de ses systèmes d'information. Ces infrastructures sont essentielles pour garantir la continuité de service, la supervision des réseaux, la maintenance des équipements, la billetterie, ou encore l'information voyageurs.

B) RDS

Le département RDS (Réseau de Surface), anciennement appelé BUS, est une entité opérationnelle de la RATP chargée de la gestion et de l'exploitation des réseaux de bus en Île-de-France. Il s'agit de l'un des départements les plus étendus de l'entreprise, avec plusieurs milliers (~4000) de véhicules en circulation et un maillage dense sur l'ensemble du territoire francilien (20000 agents). Le RDS regroupe plusieurs centres de bus répartis dans toute la région et assure quotidiennement la mobilité de millions de voyageurs. Son activité ne se limite pas à l'exploitation : elle comprend également la planification des services, l'entretien du matériel roulant, la supervision du trafic en temps réel et la coordination avec les autres modes de transport. Avec la transformation numérique engagée par la RATP, le RDS intègre de plus en plus d'outils technologiques pour améliorer la qualité de service, optimiser les tournées et fiabiliser les données opérationnelles. L'informatique joue un rôle central dans la gestion de ce réseau complexe, qu'il s'agisse de l'automatisation de certaines tâches, de la gestion des données ou de la maintenance prédictive des équipements.

Dans un contexte d'ouverture à la concurrence du réseau de bus en Île-de-France, la continuité de service représente un enjeu stratégique majeur pour le département RDS. Il s'agit de garantir aux usagers une qualité de service constante, indépendamment des changements d'opérateurs ou d'organisation. Cela implique la mise en place de procédures robustes, d'outils de supervision performants et d'une coordination étroite entre les équipes techniques et opérationnelles.

2 – Exploitation

L'exploitation du réseau de bus en Île-de-France est assurée par plusieurs centres bus répartis sur l'ensemble du territoire francilien. Chaque centre bus constitue une entité opérationnelle responsable de la gestion quotidienne des lignes qui lui sont attribuées. Cela comprend l'organisation des



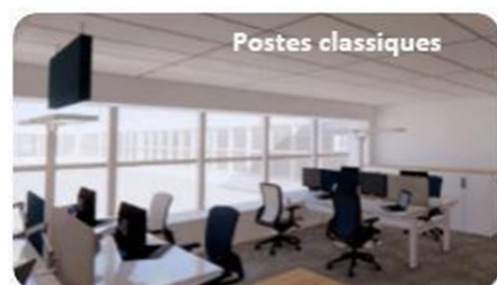
services de conduite (plannings, affectation des agents de conduite), la sortie et le retour des véhicules, ainsi que le suivi du trafic en temps réel. Les centres assurent également la coordination avec les services de maintenance pour garantir la disponibilité des bus en bon état de fonctionnement. Grâce à des outils informatiques et à des systèmes de gestion centralisés, chaque centre peut adapter l'offre de transport aux besoins locaux, réagir rapidement en cas d'incident, et transmettre des informations en temps réel aux voyageurs. Cette organisation décentralisée permet une meilleure réactivité sur le terrain tout en garantissant une exploitation cohérente et homogène à l'échelle du réseau.

3 - Unité DOSI

L'entité DOSI (Direction de l'Organisation et des Systèmes d'Information) du département RDS joue un rôle central dans le bon fonctionnement et la modernisation du réseau de bus. Elle est chargée de piloter l'ensemble des systèmes d'information liés à l'exploitation, à la maintenance et à la supervision du réseau. La DOSI assure la gestion et l'évolution des infrastructures informatiques locales (réseaux, serveurs, postes de travail), le support technique aux utilisateurs, ainsi que la mise en œuvre de projets numériques visant à optimiser les processus métiers. Elle collabore étroitement avec les centres bus pour garantir la disponibilité des outils critiques, comme les systèmes de gestion des tournées, de suivi des véhicules en temps réel, ou encore les applications de planification. Dans un contexte d'ouverture à la concurrence et de transformation numérique, la DOSI joue également un rôle stratégique en accompagnant le déploiement de solutions innovantes (maintenance prédictive, supervision centralisée, cybersécurité, etc.) afin d'assurer la continuité de service et la performance globale du réseau.

4 - SAE

Le SAE (Systèmes d'Aide à l'Exploitation) est une équipe spécialisée au sein de la DOSI du département RDS. Elle est chargée de gérer, maintenir et faire évoluer les systèmes techniques qui permettent le suivi en temps réel de l'exploitation des bus. Ces systèmes sont utilisés par les régulateurs, les centres bus et les agents



sur le terrain pour superviser la position des véhicules, optimiser les temps de parcours et adapter l'offre de transport en fonction des conditions réelles. L'équipe SAE intervient sur l'ensemble de l'infrastructure liée à ces dispositifs, depuis les équipements embarqués dans les bus (comme les systèmes de géolocalisation ou les interfaces conducteur) jusqu'aux serveurs et applications métiers utilisés en centre de supervision. Elle joue un rôle clé dans la fiabilité des données opérationnelles, la continuité de service et l'amélioration constante de la qualité du réseau. En lien étroit avec les équipes métiers et techniques, le SAE contribue activement à la transformation numérique de l'exploitation des bus.

II - La Mission

Durant mon stage au sein de l'équipe SAE (Systèmes d'Aide à l'Exploitation) de la DOSI, j'ai eu pour mission de développer un outil automatisé permettant d'identifier et de traiter les "missions inconnues" dans le réseau de bus de la RATP.

Une mission, dans le cadre de l'exploitation du réseau, correspond à un trajet planifié : elle est définie par une station de départ, une station d'arrivée, un horaire de départ précis et un code d'identification. Ces missions sont préenregistrées dans la base de données du SAE-IP, une application qui initialise et paramètre les données issues de la création de l'offre de transport afin qu'elles puissent être utilisées par le SAE-TR. Le SAE-TR (Système d'Aide à l'Exploitation en Temps Réel) est une application dédiée à l'exploitation quotidienne du réseau, notamment à la régulation en temps réel par les régulateurs présents au centre de supervision (CRIV).

Une mission inconnue désigne quant à elle une course créée en temps réel par un régulateur, mais qui n'est pas présente dans la base du SAE-IP. Cela peut provoquer des erreurs d'affichage pour les voyageurs, des pertes d'information pour les conducteurs, et perturber la régulation de la ligne.

Exemples de conséquence de mission inconnue :

Girouette :



Girouettes sans destination



Tableau de bord sans information

1 – Collecte et préparation des données

Pour identifier les missions inconnues, il était nécessaire de comparer deux types de données :

- **La liste des missions théoriques** : extraite de la base de données PostgreSQL du SAE-IP, et de les enregistrer dans un fichier CSV (prevue.csv).
- **La liste des missions réalisées** : issue des archives quotidiennes du SAE-TR, disponible sous forme de fichiers compressés au format .tar.gz. Ces archives contiennent des fichiers JSON détaillés regroupés par groupes de lignes, qui ont été extraits sur un serveur distant via un protocole sécurisé (SCP avec ProxyJump).
Les données utiles (dépôt, groupe de ligne, code mission, type de mission, etc.) ont été filtrées et extraites à l'aide d'outils en ligne de commande (grep, sed, jq, paste, cut) pour constituer un fichier CSV (reel.csv).

a) Extraction des missions théoriques depuis la base SAE-IP

Les missions dites « théoriques » correspondent aux trajets planifiés et validés lors de la création de l'offre de transport. Elles sont gérées par le SAE-IP, une application qui initialise et paramètre les données pour qu'elles soient utilisables par les outils d'exploitation en temps réel comme le SAE-TR. Ces données sont stockées dans une base de données PostgreSQL.

Fonctionnement du Script (*recuperation_prevue.sh*) :

Chaque jour, un fichier prevue.csv est mis à jour automatiquement. Il contient la liste actualisée des couples « groupe_ligne ; code_mission » présents dans la base SAE-IP extraite via une requête psql. Cette liste est considérée comme la référence pour identifier les missions valides.

```
lib_groupe;miss_code;
101;1;
101;2;
101;3;
101;4;
101;5;
101;8;
101;9;
```

Extrait du fichier prevue.csv

Cependant, il existe un cas particulier : les lignes ayant une génération différée du SAE-IP. Dans ce cas, la mise à jour des

missions pour ces lignes a été programmée mais n'est pas encore appliquée dans l'exploitation. Ainsi, les missions de ces lignes ne sont pas modifiées dans le fichier `prevue.csv` jusqu'à la date effective de la génération différée. Autrement dit, les anciennes missions sont conservées, même si une nouvelle offre a été prévue.

Ce comportement peut entraîner, lors des comparaisons, l'apparition de nombreuses missions inconnues pour ces lignes — alors qu'il ne s'agit pas réellement d'anomalies, mais simplement d'un décalage entre les données théoriques en cours d'application et celles à venir.

Ce phénomène a notamment été observé pendant les phases de test, où certaines lignes affichaient 100 % de missions inconnues à cause d'une génération différée active ayant supprimé ou remplacé toutes leurs missions à venir.

b) Récupération des missions réalisées depuis les archives SAE-TR

Les missions réellement effectuées chaque jour sont enregistrées par le SAE-TR (Système d'Aide à l'Exploitation en Temps Réel), qui collecte les données de circulation en direct et les archive quotidiennement. Ces archives prennent la forme de fichiers compressés au format `.tar.gz`, nommés selon le format `expl3.JJMMYYYY.DSPX.tar.gz`, et contiennent des fichiers JSON classés par groupes de lignes.

Fonctionnement du script (*recuperation_reel.sh*) :

Pour récupérer ces fichiers automatiquement, un script nommé `recuperation_reel.sh` a été développé. Celui-ci établit une connexion sécurisée à un serveur distant contenant les archives du SAE-TR. La commande utilisée est `scp`, qui permet le transfert sécurisé des fichiers via SSH.

Cependant, l'accès à ce serveur nécessite de passer par un serveur intermédiaire (proxy SSH). Pour cela, la connexion a été configurée avec l'option `ProxyJump`, qui permet d'effectuer un "saut" automatique via ce proxy, rendant la procédure entièrement transparente.

Afin de simplifier l'automatisation du transfert et d'éviter la saisie manuelle des mots de passe à chaque exécution, une authentification par clé SSH a été mise en place via la

commande `ssh-copy-id`. Cette configuration permet au script de se connecter sans intervention humaine, ce qui est essentiel pour une exécution quotidienne automatisée.

c) Extraction et filtrage des données pertinentes

Une fois les fichiers `.tar.gz` récupérés localement, le script les décompresse avec la commande `tar`. Le résultat est un ensemble de fichiers JSON organisés par groupe de ligne (par exemple, un fichier X12 regroupe les données pour les lignes 212 et 312). Ces fichiers contiennent un grand volume d'informations sur chaque mission (heure, dépôt, code, véhicule, type, etc.).



```
"version": "1.0",
"groupeLigne": "589",
"ligneNomTmReel": "",
"ligneNomTmAnticipe": "",
"ligneFNoctilien": false,
"lignes": [
  {
    "ligneNomLigneVoyageur": "589",
    "ligneTypeLigne": "BUS",
```

Extrait d'un fichier JSON.

Pour les besoins du projet, seules certaines données étaient nécessaires :

- Le dépôt,
- Le groupe de ligne,
- Le code mission,
- Le mnémonique,
- Le type de mission,
- Et le nombre de fois que cette mission a été utilisée dans la journée.

Un traitement en ligne de commande, à l'aide d'outils comme `jq`, `grep`, `sed`, `cut` et `paste`, a permis d'extraire uniquement ces informations depuis les fichiers JSON.

Le résultat final est stocké dans un fichier temporaire appelé reel.csv, au format CSV (séparateur ;), qui sera utilisé pour la détection des missions inconnues.

```
STM;101;11;STM->JGT;SPECIALE;0;
STM;101;5;JGP->STM;SORTIE;1;
STM;101;3;CA->JGP;VOYAGEUR;32;
STM;101;10;JGT->PO;VOYAGEUR;0;
STM;101;8;CA->CR#;RENTREE;0;
STM;101;9;PO->JGP;VOYAGEUR;0;
STM;101;1;STM->CR;SPECIALE;1;
STM;101;4;JGT->CA;VOYAGEUR;31;
```

Extrait de reel.csv.

2 – Identification des missions inconnues

a) L'Identification

Une fois les fichiers prevue.csv (missions théoriques) et reel.csv (missions réellement réalisées) générés, la prochaine étape consiste à détecter les missions inconnues, c'est-à-dire celles présentes dans les données réelles mais absentes de la base SAE-IP.

Cette détection est assurée par le script `identification_inconnues.sh`.

Fonctionnement du script (*identification.sh*) :

Le principe est simple : pour chaque ligne du fichier reel.csv, le script extrait le couple "groupe_ligne;code_mission" et le compare à la liste des missions théoriques contenues dans le fichier prevue.csv.

Si ce couple n'existe pas dans la liste théorique, alors la mission est considérée comme "inconnue". Dans ce cas, la ligne correspondante du fichier reel.csv (incluant le dépôt, le type de mission, le mnémonique, le nombre d'utilisations, etc.) est copiée dans un nouveau fichier nommé `date_inconnues.csv`, qui regroupe toutes les anomalies pour la journée en question.

L'en-tête du fichier contient la date et l'heure de génération pour en faciliter la traçabilité.

La comparaison est réalisée à l'aide de commandes shell classiques, telles que :

- grep pour rechercher la présence d'un couple dans prevue.csv,
- sed pour nettoyer et formater les données,
- Des structures conditionnelles Bash pour gérer l'inclusion ou non dans la sortie.

Ce processus permet d'obtenir chaque jour une liste fiable de missions réalisées mais non enregistrées dans le SAE-IP, ce qui signale soit une anomalie d'exploitation, soit un retard dans la mise à jour des données théoriques (comme c'est le cas lors de générations différées).

Vérification de la véracité des données générées sur le SAE-IP

The screenshot shows a window titled "Paramétrage des missions" with a "Sélection" tab. It contains a table with the following columns: N° TM, N°TD, Nom, Info. voyageur, Quai, Machinis, Seuil, Logique, and Partielle. The table lists various missions, with the row for mission 92 (GD->VIL) highlighted in red. A dialog box is open in the foreground, titled "Suppression d'une mission créée au TD". It has four buttons: "Mission à supprimer", "Modification d'une mission HLP", "Mission à modifier", and "Mission de référence". A dropdown menu is open next to the "Mission à supprimer" button, showing a list of mission identifiers: GD->VIL, VIL->FAA#, VIL->GD, VIL->LP#, and VIL->CP. The "GD->VIL" option is currently selected.

N° TM	N°TD	Nom	Info. voyageur	Quai	Machinis	Seuil	Logique	Partielle
109	149	ALB->SDP	SANS VOYAGEURS		SDP	600		Non
10	5	ALB->VIL	SANS VOYAGEURS		VIL	600		Non
15	137	CP->AUB	CENTRE BUS		AUB	600		Non
87	135	CP->VIL	PORTE DE LA VILLETTE		VIL	600		Non
107	148	DAP->SDP	SAINT-DENIS - PLEYEL		SDP	600		Oui
89	145	FAV->VIL	PORTE DE LA VILLETTE		VIL	600		Non
99	122	FPV->VIL	PORTE DE LA VILLETTE		VIL	600		Non
92		GD->VIL	PORTE DE LA VILLETTE		VIL	600		Non
21	139	LP->AUB	CENTRE BUS		AUB	600		Non
93	62	LP->VIL	PORTE DE LA VILLETTE					
78	131	RAM->AUB	CENTRE BUS AUBERVILL					
91	120	RAM->VIL	PORTE DE LA VILLETTE					
108	146	SDP->AUB	CENTRE BUS					
105	143	SDP->VIL	PORTE DE LA VILLETTE					
8	11	VIL->AUB	CENTRE BUS					
118	134	VIL->CP	CARREFOUR PLEYEL ME					
141		VIL->FAA#	SANS VOYAGEURS					
115	133	VIL->FP	FRONT POPULAIRE-GAR					
96		VIL->GD	STADE DE FRANCE-ST D					
94	138	VIL->LP	LA PLAINE STADE DE FR					
31		VIL->LP#	SANS VOYAGEURS					

b) La synthèse

Une fois les missions inconnues identifiées, une analyse statistique est générée automatiquement pour fournir une vue d'ensemble de la situation sur l'ensemble du réseau. Cette étape est assurée par le script synthese.sh.

Le but est de produire un fichier `date_synthese.csv` contenant des indicateurs chiffrés utiles aux équipes du SAE pour le suivi qualité, le diagnostic d'anomalies, ou encore la préparation de bilans de performance.

Fonctionnement du script (*synthese.sh*) :

Le script lit les données issues des fichiers `reel.csv` (missions réalisées) et `date_inconnues.csv` (missions inconnues) pour établir plusieurs niveaux de statistiques :

- **Statistiques globales :**
 - Nombre total de missions réalisées dans la journée,
 - Nombre et pourcentage de missions inconnues sur l'ensemble du réseau.
 - Etc...
- **Statistiques par type de mission :**

Certaines missions sont typées (exploitation, service partiel, régulation, formation, etc.). Le script calcule pour chaque type :

 - Le nombre total de missions réalisées,
 - Le nombre de missions inconnues,
 - Leur pourcentage.
 - Etc...
- **Statistiques par ligne et dépôt :**

Le script regroupe les données par combinaison ligne + dépôt, et calcule le ratio missions inconnues / missions totales pour chaque combinaison.

Toutes ces données sont exportées dans un fichier CSV bien structuré (`date_synthese.csv`), utilisable directement par les équipes techniques ou fonctionnelles.

Objectif de cette synthèse :

La génération de statistiques permet d'identifier des lignes à problème ou des erreurs de paramétrage, et de mieux comprendre l'impact réel des missions inconnues sur l'exploitation.

3 – Envoi des rapports par mail

Une fois les fichiers de missions inconnues et de synthèse statistique générés, ils doivent être transmis quotidiennement aux équipes concernées, notamment les

référents SAE, les responsables des centres de bus ou les équipes d'exploitation.

Cette tâche est automatisée par le script `envoi_mail.sh`, qui utilise les outils disponibles nativement sur les systèmes Linux.

Fonctionnement du script `envoi_mail.sh`

L'envoi des rapports s'effectue à l'aide de la commande `sendmail`, un utilitaire de messagerie en ligne de commande. Pour cela, le serveur de messagerie SMTP de la RATP a été configuré directement dans le fichier de configuration de l'application afin de garantir une compatibilité avec le réseau interne.

Le script assemble un message contenant :

- Un objet personnalisé avec la date du jour,
- Un corps de mail expliquant le contenu (missions inconnues détectées, synthèse statistique),
- En pièce jointe, les fichiers `date_inconnues.csv` et `date_synthese.csv`.

Les destinataires sont listés dans un fichier de configuration, ce qui permet de les modifier facilement sans toucher au script principal.

L'objectif est d'assurer une transmission automatique, sans intervention humaine, chaque jour après l'analyse complète des données. Cela permet aux équipes concernées d'avoir à disposition, dès le matin, une vision claire des anomalies à traiter et des éventuelles erreurs dans l'offre de transport.

4 – Archivage et automatisation

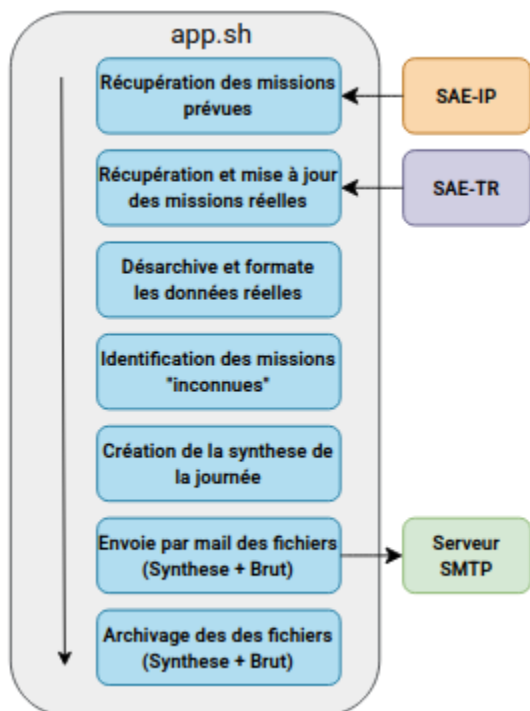
L'ensemble du traitement de la récupération des données à l'envoi des rapports est orchestré par un script principal nommé `app.sh`, conçu pour exécuter dans le bon ordre tous les scripts développés au cours de la mission. Il intègre également la gestion des fichiers temporaires, l'archivage, la planification automatique ainsi que la journalisation des actions.

Fonctionnement du script (`app.sh`) :

Le script `app.sh` agit comme un chef d'orchestre :

- Il exécute successivement les scripts suivants :
 - recuperation_reel.sh (récupération des traces des missions réelles),
 - recuperation_prevue.sh (récupération de la liste des missions théoriques),
 - format_reel.sh (traitement des missions réelles),
 - identification_inconnues.sh (détection des missions inconnues),
 - synthese.sh (génération de statistiques),
 - envoi_mail.sh (envoi des rapports par mail),
 - archivage.sh (archive le rapport / la synthèse avec tar).
- Il supprime les fichiers temporaires (fichiers JSON, CSV intermédiaires) pour libérer l'espace disque.
- Il inclut un mécanisme de nettoyage automatique des anciennes archives, en supprimant celles datant de plus de X jours (paramétrable dans le fichier de configuration) via la commande find.

Schéma de l'application app.sh :



Extrait de `app.sh` :

```

if bash /appli/ident_inc/comp/exe/recuperation_prevue.sh; then
    echo "recuperation_prevue.sh : [V]"
else
    log "ERREUR >> recuperation_prevue.sh >> exit code : $?"
    echo "recuperation_reel.sh : [X]"
fi

if bash /appli/ident_inc/comp/exe/recuperation_reel.sh; then
    echo "recuperation_reel.sh : [V]"
else
    log "ERREUR >> recuperation_reel.sh >> exit code : $?"
    echo "recuperation_reel.sh : [X]"
fi
  
```

Automatisation avec crontab :

Pour que l'ensemble s'exécute automatiquement chaque jour, le script `app.sh` est lancé via le planificateur de tâches `crontab`.

L'exécution est programmée à 4h30 du matin, afin de laisser le temps au SAE-TR de générer ses fichiers d'archives (généralement disponibles vers 4h).

Le script est lancé sous l'utilisateur `ident_inc`, qui possède les droits nécessaires pour accéder aux répertoires, établir les connexions SSH et exécuter les commandes liées à l'application.

5 – Journalisation

Chaque script composant l'application intègre un système de journalisation automatique afin d'assurer une traçabilité complète des traitements réalisés au quotidien. L'ensemble des actions, messages d'état et éventuelles erreurs sont enregistrés dans un fichier de log unique par jour, nommé selon le format `YYYYmmdd.log`.

Cette journalisation est prise en compte dans tous les scripts (`recuperation_reel.sh`, `identification_inconnues.sh`, `synthese.sh`, `envoi_mail.sh`, etc.) via des instructions d'écriture conditionnelles, permettant d'enregistrer :

- Les étapes en cours d'exécution (ex. : début de transfert, début de traitement JSON, génération des fichiers...),
- Les fichiers traités,
- Les codes de sortie des commandes critiques (retour scp, succès ou échec de la décompression, etc.),
- Les anomalies détectées (absence de missions prévues pour une ligne, échec de comparaison, erreurs dans le JSON...),
- Les couples de missions inconnues identifiés.

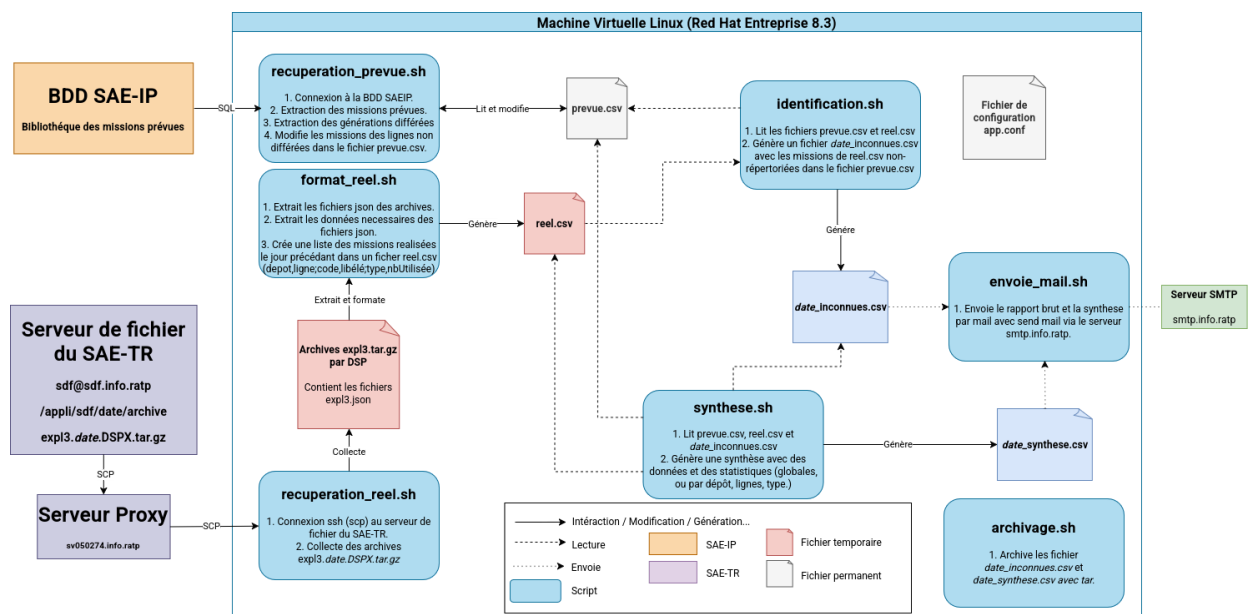
Ce fichier log est généré automatiquement en début de traitement par le script global `app.sh` et est mis à jour tout au long de l'exécution des différents modules. Il constitue un outil indispensable pour le suivi quotidien, le diagnostic en cas de dysfonctionnement, et facilite également les opérations de maintenance ou d'amélioration continue du système.

extrait d'un log :

```
[2025/05/25 04:30:01] Demarage de app.sh
[2025/05/25 04:30:01] Collecte des donnees
[2025/05/25 04:30:01] > Collecte des donnees prevue (SAE-IP BDD)
[2025/05/25 04:30:01] > Creation du fichier temporaire prevue.csv
[2025/05/25 04:30:01] > 7768 couples ligne;codeMission trouve
[2025/05/25 04:30:01] > Collecte des donnees reelles (SAE-TR Serveur de fichier)
[2025/05/25 04:30:03] > 13 DSP trouve
[2025/05/25 04:30:03] Desarchivage et formatage des donnees effectuees
[2025/05/25 04:30:03] > Desarchivage des archives expl3
[2025/05/25 04:30:10] > 270 fichiers JSON extraits
```

7 - Résultats et difficultés

On retrouve une architecture comme ci-dessous :



a) Résultats

L'outil développé au cours de mon stage a avant tout pour vocation de fournir aux équipes du SAE une meilleure visibilité sur les anomalies potentielles liées à l'exploitation des missions de bus. Il ne vise pas à corriger directement les erreurs, mais à mettre à disposition des données structurées et fiables pour faciliter les diagnostics.

Les résultats les plus significatifs sont les suivants :

- **Outil d'aide à l'analyse** : l'application permet de croiser les données théoriques et réelles de circulation afin d'identifier rapidement des incohérences (missions absentes de la base SAE-IP, erreurs de paramétrage, problèmes de synchronisation). Cela aide à orienter les investigations techniques vers le SAE-IP, le SAE-TR ou même vers le processus de création de l'offre de transport.
- **Centralisation des informations** : les rapports quotidiens regroupent en un seul endroit toutes les missions non reconnues, accompagnées de statistiques détaillées, facilitant la détection de tendances anormales (hausse soudaine du taux de missions inconnues, récurrence sur certaines lignes ou dépôts, etc.).
- **Gain de temps et efficacité** : en automatisant entièrement la récupération, l'analyse, la synthèse et la diffusion des données, l'outil libère les équipes de tâches répétitives et leur permet de se concentrer sur des actions correctives ou des audits de qualité plus ciblés.

Données générales de la synthèse :

Couples ligne, codeMission (BDD SAE-IP)	7766
Couples ligne, codeMission (Fichiers expl3)	8740
Couples ligne, codeMission non-repertoriees	1401
% Couples ligne, codeMission non-repertoriees	16
Missions realisees	61482
Missions realisees non-repertoriees	2273
% Missions realisees	3,6
Groupes de lignes traitees (Fichiers expl3)	269
Groupes de lignes traitées (BDD SAE-IP)	286

Données par Type de mission :

Type	Missior	% Miss	Missio	% Mis	Nb de	% Mis	Nb de	% Mis
RACCORDEMENT	27	0,3	16	59,2	21	0	21	100
RENTREE	1325	15,1	198	14,9	454	0,7	149	32,8
SORTIE	1068	12,2	278	26	5343	8,6	390	7,2
SPECIALE	1239	14,1	269	21,7	5360	8,7	264	4,9
VOYAGEUR	5081	58,1	640	12,5	50304	81,8	1449	2,8

Données par Groupe de lignes :

Depot	Ligne	Missior	Missio	% Mis	Missio	% Mis	Nb de	Nb de	% Mi	Nb de	Nb de	% Mis	Date de generation	generation differee	
ASN	175	32	8	25	22	2	9	150	10	6,6	117	3	2,5	28/05/2025 00:17	NON
ASN	235	40	0	0	19	1	5,2	225	0	0	192	0	0	28/05/2025 00:41	NON
ASN	238	40	8	20	18	5	27,7	205	5	2,4	161	3	1,8	07/03/2025 03:57	NON
ASN	340	28	1	3,5	11	1	9	119	2	1,6	92	0	0	27/01/2025 02:51	NON
ASN	363	8	0	0	2	1	50	112	0	0	102	0	0	13/05/2025 01:36	NON

Données par dépôt :

Depot	Missior	Missio	% Mis	Missio	Missio	% Mis	Nb de	Nb de	% Mi	Nb de	Nb de	% Mis
ASN	160	17	10,6	78	7	8,9	893	17	1,9	734	6	0,8
AUB	401	59	14,7	251	26	10,3	2921	140	4,7	2416	109	4,5
BDM	478	86	17,9	266	39	14,6	3466	287	8,2	2821	232	8,2
BEL	286	21	7,3	161	9	5,5	2273	28	1,2	1847	18	0,9
BSM	187	18	9,6	91	8	8,7	1960	14	0,7	1656	9	0,5

b) Difficultés

Ce projet a également été l'occasion de faire face à plusieurs difficultés techniques et organisationnelles, que j'ai dû surmonter au fil du stage :

Prise en main de l'environnement :

L'un des premiers défis a été de comprendre l'environnement complexe dans lequel s'inscrivait l'application : l'architecture des systèmes SAE, les relations entre SAE-IP et SAE-TR, le format des données, ou encore les règles d'accès aux serveurs distants. Une phase de documentation et d'observation a donc été nécessaire pour m'approprier le contexte.

Manipulation des fichiers JSON :

Le traitement des fichiers JSON a représenté une difficulté particulière. En effet, les fichiers étaient très volumineux, parfois partiellement corrompus ou ne respectaient pas la documentation, et contenus dans des archives non homogènes. Il a donc fallu

adapter les scripts pour gérer les exceptions, optimiser les commandes de filtrage (jq, grep, etc.), et garantir la robustesse du traitement.

Conclusion

Ce stage m'a offert une immersion concrète dans le monde professionnel, au sein d'un environnement technique complexe et critique pour le bon fonctionnement du réseau de transport francilien. J'ai pu développer une application complète, utile et utilisée quotidiennement, tout en approfondissant mes compétences en administration système, scripting Bash, manipulation de données et gestion de projet. Les difficultés rencontrées m'ont permis de progresser dans ma capacité à analyser des problèmes, à concevoir des solutions robustes et à m'intégrer efficacement dans une équipe technique.

Remerciements

Je tiens à remercier l'ensemble des personnes qui m'ont accompagné tout au long de ce stage au sein de la RATP, et plus particulièrement au sein de l'équipe SAE du département RDS.

Je remercie tout d'abord mon tuteur de stage, Olivier Elisabeth, pour son accueil, sa disponibilité, et la qualité de son encadrement. Ses conseils techniques, sa pédagogie et sa confiance m'ont permis d'évoluer dans un environnement stimulant et de mener à bien un projet concret et formateur.

Je remercie également Fabien Chantel, de l'équipe SAE, pour m'avoir accueilli dans son service. Même si nos échanges ont été limités au quotidien, j'ai eu l'opportunité de lui présenter en fin de stage l'ensemble du travail accompli, ce qui m'a permis d'avoir un retour professionnel sur le projet et sur les pistes d'amélioration possibles. Je le remercie pour la confiance accordée ainsi que pour l'intérêt porté à mon travail.

Je souhaite également remercier Emmanuel Janand, pour m'avoir donné l'opportunité d'effectuer mon stage au sein de son unité. Son accord m'a permis de vivre cette première expérience professionnelle dans un environnement stimulant, en lien direct avec les systèmes d'information critiques de la RATP.

Enfin, je souhaite remercier l'ensemble de l'équipe pédagogique de mon établissement pour l'accompagnement tout au long de l'année, ainsi que pour la préparation qui m'a permis d'aborder ce stage avec les bonnes bases techniques et professionnelles.

Annexes

Informations missions inconnues :

QU'EST-CE QU'UNE MISSION INCONNUE ?

Définitions :



Une **mission inconnue** est une mission de régulation commandée par le CRIV non connue du SAEIP

Une **mission de régulation** est une mission qui n'est pas prévue dans un TM, qui est créée dans le SAEIP et qui est utilisée pour une manœuvre par un régulateur au CRIV

Une **mission de régulation** peut être de plusieurs types : **commercial**, **remisage**, **HLP**, **semi-direct**

Pourquoi créer une mission de régulation ?



Créer une mission de régulation permet de transmettre les **données d'Informations Voyageurs** en Embarqué (trigramme du terminus arrivée de la mission sur le pupitre machiniste et girouette) et de calculer des **horaires estimés** plus justes en Temps Réel

3



IMPACTS : COMMENT LES RECONNAITRE ?

Impacts Informations Voyageurs :

Sur toutes les coquilles qui utilise la mission inconnue nous aurons les comportements suivants :

Girouette :



Les girouettes latérales et frontales sont noires

Annonceurs visuels embarqué :



Les annonceurs visuels n'indiquent que la date, l'heure et le numéro de la ligne

Annonces sonores :



Les destinations et points d'arrêt ne sont pas annoncés

BIV :



La BIV affiche la destination principale au lieu de la destination partielle

Écran TFT :



L'écran TFT affiche la ligne uniquement

5



COMMENT RÉSOUDRE CE PROBLÈME ?

1. Je remonte le problème à ma cellule technique :

Si vous constatez les impacts vus précédemment sur vos lignes, vous devez fournir les informations nécessaires à votre cellule technique pour l'analyse :

- Origine-destination de la mission (exemple : GD -> VIL)
- Description des impacts (Slides 4 - 5)

2. Confirmer l'absence de cette mission sur le SAEIP :

Sur le SAEIP, il faut vérifier la liste des missions présentes sur la fenêtre destination par missions. Une mission de régulation n'a pas de N°TM.

Ou, vérifier sur la fenêtre gestion des de missions de régulation, la liste des missions à supprimer.

Si la mission recherchée n'existe pas il faudra la créer en suivant le processus de création de mission de régulation vu lors des formations DPUO.

N° TM	N°TD	Nom	Info. voyageur	Qual	Machins	Scall	Logique	Partielle
109	140	ALB->SDP	SANS VOYAGEURS	SDP	600			Non
10	5	ALB->VIL	SANS VOYAGEURS	VIL	600			Non
15	137	CP->ALB	CENTRE BUS	ALB	600			Non
87	120	CP->VIL	PORTE DE LA VILLETTE	VIL	600			Non
107	148	DAP->SDP	SAINT-DENIS - PLEYEL	SDP	600			Oui
89	145	FAV->VIL	PORTE DE LA VILLETTE	VIL	600			Non
99	122	PPV->VIL	PORTE DE LA VILLETTE	VIL	600			Non
92	92	GD->VIL	PORTE DE LA VILLETTE	VIL	600			Non
21	139	LP->ALB	CENTRE BUS	ALB	600			Non
93	62	LP->VIL	PORTE DE LA VILLETTE	VIL	600			Non
78	131	RAM->ALB	CENTRE BUS ALBERVILLE	ALB	600			Non
91	120	RAM->VIL	PORTE DE LA VILLETTE	VIL	600			Non
108	140	SDP->ALB	CENTRE BUS	ALB	600			Non
105	143	SDP->VIL	PORTE DE LA VILLETTE	VIL	600			Non
8	11	VIL->ALB	CENTRE BUS	ALB	600			Non
118	124	VIL->CP	CABRESPOUR - PLEYEL - NE	CP	600			Non
115	133	VIL->FAA#	SANS VOYAGEURS					Non
96	133	VIL->PPP	FRONT POPULAIRE - GAR	PPP	600			Non
94	138	VIL->GD	STADE DE FRANCE - ST D	GD	600			Non
31	119	VIL->LP#	SANS VOYAGEURS					Non
104	119	VIL->RAP	SAINT-DENIS - ANPERSE	ANPERSE	600			Non
106	142	VIL->SDP	SAINT-DENIS - PLEYEL	SDP	600			Non

6

Quelques parties de scripts :
app.sh :

```

#!/bin/bash

config_file=/appli/ident_inc/comp/param/app.conf

log_file=$(grep -oP "(?<=LOG DOSSIER: ')[^']+(?=';)" $config_file)/$(date +%Y%m%d).log
duree_archive=$(grep -oP "(?<=DUREE_ARCHIVE: ')[^']+(?=';)" $config_file)

log () {
    echo "[$(date +%Y/%m/%d ' %H:%M:%S)]" "$@" >> $log_file
}

rm /appli/ident_inc/data/emet/*
rm /appli/ident_inc/tmp/*

log "Demarage de app.sh"
# Collecte des données nécessaire
log "Collecte des donnees"
if bash /appli/ident_inc/comp/exe/recuperation_prevue.sh; then
    echo "recuperation_prevue.sh : [V]"
else
    log "ERREUR >> recuperation_prevue.sh >> exit code : $?"
    echo "recuperation_reel.sh : [X]"
fi

if bash /appli/ident_inc/comp/exe/recuperation_reel.sh; then
    echo "recuperation_reel.sh : [V]"
else
    log "ERREUR >> recuperation_reel.sh >> exit code : $?"
    echo "recuperation_reel.sh : [X]"
fi

# Extraction et formatage du fichier expl3 (reel)
log "Desarchivage et formatage des donnees effectuees"
if bash /appli/ident_inc/comp/exe/format_reel.sh /appli/ident_inc/data/recep/ /appli/ident_inc/tmp/; then
    echo "format_reel.sh : [V]"
else
    log "ERREUR >> format_reel.sh >> exit code : $?"
    echo "format_reel.sh : [X]"
fi

# Identification des missions non-répertoriées
log "Generation de la liste des missions effectuees (*_inconnue.csv)"
if bash /appli/ident_inc/comp/exe/identification.sh /appli/ident_inc/tmp/reel.csv /appli/ident_inc/tmp/prevue.csv; then
    echo "identification.sh : [V]"
else
    log "ERREUR >> identification.sh >> exit code : $?"
    echo "identification.sh : [X]"
fi

if rm /appli/ident_inc/data/recep/*; then
    log "> Fichiers de /data/recep supprimés"
else log "> ERREUR >> suppressions des fichiers de /data/recep >> exit code : $?"
fi

# Envoie des données par mail
log "Envoie des fichier generes par mail"
if bash /appli/ident_inc/comp/exe/envoie_mail.sh; then
    echo "envoie_mail.sh : [V]"
else
    log "ERREUR >> envoie_mail.sh >> exit code : $?"
    echo "envoie_mail.sh : [X]"
fi

# Archivage
log "Archivage"
if bash /appli/ident_inc/comp/exe/archivage.sh; then
    echo "archivage.sh : [V]"
else
    log "ERREUR >> archivage.sh >> exit code : $?"
fi

log "Suppression des archives (+ de ${duree_archive} jours)"
find /appli/ident_inc/data/archive/ -type f -mtime +$duree_archive -exec rm {} \;

log "app.sh termine"

```

Format_reel.sh :

```
#!/bin/bash

missions_reel=$(cat ${1})
missions_prevue=$(cat ${2})

config_file=/appli/ident_inc/comp/param/app.conf
output_file=/appli/ident_inc/data/emet/${date +%Y%m%d}_inconnues.csv

log_file=$(grep -oP "(?<=LOG_DOSSIER: ')[^']+(?=';)" $config_file)/${date +%Y%m%d}.log
log () {
    echo "[$(date +%Y/%m/%d' %H:%M:%S)] > " "$@" >> $log_file
}

log "Creation du fichier $output_file"
echo "$(date +%d/%m/%Y' %H:%M:%S)" > $output_file
log "Generation de la liste des missions reelles non repertoriees dans les missions prevues"
i=0
for reel in $missions_reel; do
    ligne_code=$(echo $reel | grep -m 1 -oP '^[^;]+\K[^;]+;[^;]+')
    est_repertorie=$(echo "$missions_prevue" | grep -m 1 -c "$ligne_code;")
    if [ "$est_repertorie" -eq "0" ]; then
        echo "$reel" >> $output_file
        i=$((i + 1))
    fi
done
log "$i missions non repertoriees trouvees"
```

Synthese.sh :

- données globales

```
nb_prevue=$(echo "$prevue" | wc -l)
echo -e "Couples ligne, codeMission (BDD SAE-IP);$nb_prevue;" >> $output_file

nb_reel=$(echo "$reel" | wc -l)
echo -e "Couples ligne, codeMission (Fichiers expl3);$nb_reel;" >> $output_file

nb_inconnue=$(echo "$inconnue" | wc -l)
echo -e "Couples ligne, codeMission non-repertoriees;$nb_inconnue;" >> $output_file

pct_inconnue_unique=$(echo "$(echo "scale=3; $nb_inconnue / $nb_reel * 100" | bc)" | sed 's/\./,;/; s/00$//; s/^,0,/' )
echo -e "% Couples ligne, codeMission non-repertoriees;$pct_inconnue_unique;" >> $output_file

reel_utilisees=$(echo "$reel" | cut -d ';' -f6 | paste -sd+ - | bc)
echo -e "Missions realisees;$reel_utilisees;" >> $output_file

inconnue_utilisees=$(echo "$inconnue" | cut -d ';' -f6 | paste -sd+ - | bc)
echo -e "Missions realisees non-repertoriees;$inconnue_utilisees;" >> $output_file

pct_inconnue=$(echo "$(echo "scale=3; $inconnue_utilisees / $reel_utilisees * 100" | bc)" | sed 's/\./,;/; s/00$//; s/^,0,/' )
echo "% Missions realisees;$pct_inconnue;" >> $output_file

depots_lignes=$(echo "$reel" | grep -oP '^[^;]+;[^;]+' | sort -u)
echo "Groupes de lignes traitees (Fichiers expl3);$(echo "$depots_lignes" | wc -l);" >> $output_file

nb_ligne_prevue=$(echo "$prevue" | grep -o "^;" | sort -u | wc -l)
echo "Groupes de lignes traitees (BDD SAE-IP);$(echo "$prevue" | grep -oP "^[^;]+;" | sort -u | wc -l);" >> $output_file

# nb missions par type
types_mission=$(echo "$reel" | cut -d ';' -f5 | sort -u)
```

- données par type

```
log "Generation des donnees par type"
echo "" >> $output_file
echo "Type:Missions realisees;\ Missions realisees;Missions realisees non-repertoriees;\ Missions realisees non-repertoriees;Nb de missions effectuees;\ Missions effectuees;Nb de missions effectuees non-repertoriees"
for type_mission in $(types_mission); do
missions_de_type=$(echo "$reel" | grep "$type_mission")
nb_de_type=$(echo "$missions_de_type" | wc -l)
pct_de_type=$(echo "$(echo "scale=3; $nb_de_type / $nb_reel * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')

de_type_inconnue=$(echo "$inconnue" | grep "$type_mission")
nb_de_type_inconnue=$(echo "$de_type_inconnue" | wc -l)
pct_de_type_inconnue=$(echo "$(echo "scale=3; $nb_de_type_inconnue / $nb_de_type * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')

realisees_de_type=$(echo "$missions_de_type" | cut -d ':' -f6 | paste -sd - | bc)
pct_realisees_de_type=$(echo "$(echo "scale=3; $realisees_de_type / $reel_utilisees * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
nb_de_type_realisees_inconnue=$(echo "$de_type_inconnue" | cut -d ':' -f6 | paste -sd - | bc)
pct_de_type_realisees_inconnue=$(echo "$(echo "scale=3; $nb_de_type_realisees_inconnue / $realisees_de_type * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')

echo "type_mission;$nb_de_type;$pct_de_type;$nb_de_type_inconnue;$pct_de_type_inconnue;$realisees_de_type;$pct_realisees_de_type;$nb_de_type_realisees_inconnue;$pct_de_type_realisees_inconnue" >> $output_file
done
```

- données par groupe de lignes

```
log "Generation des donnees par groupe de ligne"
# nb missions par ligne
echo "" >> $output_file
echo "Depot:Lignes uniques realisees;Missions realisees non-repertoriees;\ Missions realisees non-repertoriees;Missions voyageur uniques realisees;Missions voyageur realisees non-repert"
for depot_ligne in $(depots_lignes); do
depot=$(echo "$depot_ligne" | grep -oP '[^:]*')
ligne=$(echo "$depot_ligne" | cut -d ':' -f2)
missions_de_ligne=$(echo "$reel" | grep "$depot_ligne:")
nb_de_ligne=$(echo "$missions_de_ligne" | wc -l)

de_ligne_inconnue=$(echo "$inconnue" | grep "$depot_ligne:")
nb_de_ligne_inconnue=$(echo "$de_ligne_inconnue" | wc -l)
pct_de_ligne_inconnue=$(echo "$(echo "scale=3; $nb_de_ligne_inconnue / $nb_de_ligne * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')

missions_de_ligne_voyageur=$(echo "$missions_de_ligne" | grep ':VOYAGEUR:')
nb_de_ligne_voyageur=$(echo "$missions_de_ligne_voyageur" | wc -l)
de_ligne_voyageur_inconnue=$(echo "$inconnue" | grep -P '[^:]*';$(ligne);[:]*[:]*:VOYAGEUR:')
nb_de_ligne_voyageur_inconnue=$(echo "$de_ligne_voyageur_inconnue" | wc -l)
pct_de_ligne_voyageur_inconnue=$(echo "$(echo "scale=3; $nb_de_ligne_voyageur_inconnue / $nb_de_ligne_voyageur * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
realisees_de_ligne=$(echo "$missions_de_ligne" | cut -d ':' -f6 | paste -sd - | bc)
realisees_de_ligne_voyageur=$(echo "$missions_de_ligne_voyageur" | cut -d ':' -f6 | paste -sd - | bc)
if [ "$nb_de_ligne_inconnue" -eq '0' ]; then
realisees_de_ligne_inconnue='0'
pct_realisees_de_ligne_inconnue='0'
realisees_de_ligne_voyageur_inconnue='0'
pct_realisees_de_ligne_voyageur_inconnue='0'
else
realisees_de_ligne_inconnue=$(echo "$de_ligne_inconnue" | cut -d ':' -f6 | paste -sd - | bc)
pct_realisees_de_ligne_inconnue=$(echo "$(echo "scale=3; $realisees_de_ligne_inconnue / $realisees_de_ligne * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
realisees_de_ligne_voyageur_inconnue=$(echo "$de_ligne_voyageur_inconnue" | cut -d ':' -f6 | paste -sd - | sed 's/^$/g' | bc)
pct_realisees_de_ligne_voyageur_inconnue=$(echo "$(echo "scale=3; $realisees_de_ligne_voyageur_inconnue / $realisees_de_ligne_voyageur * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
fi

date_generation_ligne=$(echo "$date_gen" | grep "$ligne:" | cut -d ':' -f2)
if [ "$date_generation_ligne" == '' ]; then
date_generation_ligne="FAS D'INFO"
fi
echo "$(depot);$(ligne);$nb_de_ligne;$nb_de_ligne_inconnue;$pct_de_ligne_inconnue;$nb_de_ligne_voyageur;$nb_de_ligne_voyageur_inconnue;$pct_de_ligne_voyageur_inconnue;$realisees_de_ligne;$pct"
done
```

- données par dépôt

```
log "Generation des donnees par depot"
echo "" >> $output_file
echo "Depot:Missions uniques realisees;Missions realisees non-repertoriees;\ Missions realisees non-repertoriees;Missions voyageur uniques realisees;Missions voyageur realisees non-repertoriees;"
depots=$(echo "$depots_lignes" | grep -oP '[^:]*' | sort -u)
for depot in $(depots); do
mission_de_depot=$(echo "$reel" | grep -P "$depot:")
nb_de_depot=$(echo "$mission_de_depot" | wc -l)
de_depot_inconnue=$(echo "$inconnue" | grep -P "$depot:")
nb_de_depot_inconnue=$(echo "$de_depot_inconnue" | wc -l)
pct_de_depot_inconnue=$(echo "$(echo "scale=3; $nb_de_depot_inconnue / $nb_de_depot * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
missions_de_depot_voyageur=$(echo "$mission_de_depot" | grep ':VOYAGEUR:')
nb_de_depot_voyageur_inconnue=$(echo "$de_depot_inconnue" | grep ':VOYAGEUR:')
nb_de_depot_voyageur_inconnue=$(echo "$de_depot_voyageur_inconnue" | wc -l)
pct_de_depot_voyageur_inconnue=$(echo "$(echo "scale=3; $nb_de_depot_voyageur_inconnue / $nb_de_depot_voyageur * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
realisees_de_depot=$(echo "$mission_de_depot" | cut -d ':' -f6 | paste -sd - | bc)
realisees_de_depot_voyageur=$(echo "$missions_de_depot_voyageur" | cut -d ':' -f6 | paste -sd - | bc)
if [ "$nb_de_depot_inconnue" -eq '0' ]; then
realisees_de_depot_inconnue='0'
pct_realisees_de_depot_inconnue='0'
realisees_de_depot_voyageur_inconnue='0'
pct_realisees_de_depot_voyageur_inconnue='0'
else
realisees_de_depot_inconnue=$(echo "$de_depot_inconnue" | cut -d ':' -f6 | paste -sd - | bc)
pct_realisees_de_depot_inconnue=$(echo "$(echo "scale=3; $realisees_de_depot_inconnue / $realisees_de_depot * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
realisees_de_depot_voyageur_inconnue=$(echo "$de_depot_voyageur_inconnue" | cut -d ':' -f6 | paste -sd - | sed 's/^$/g' | bc)
pct_realisees_de_depot_voyageur_inconnue=$(echo "$(echo "scale=3; $realisees_de_depot_voyageur_inconnue / $realisees_de_depot_voyageur * 100" | bc)" | sed 's/\./;/; s/00$/;/; s/^,0,/'')
fi

echo "$depot;$nb_de_depot;$nb_de_depot_inconnue;$pct_de_depot_inconnue;$nb_de_depot_voyageur;$nb_de_depot_voyageur_inconnue;$pct_de_depot_voyageur_inconnue;$realisees_de_depot;$realisees_de"
done
```

Extrait de synthese.csv :

- Entête et données globales

Date Heure de generation synthese	20250606	04:32:49
Date des donnees expl3 (SAE-TR)	20250605	
Couples ligne, codeMission (BDD SAE-IP)	7766	
Couples ligne, codeMission (Fichiers expl3)	8740	
Couples ligne, codeMission non-repertoriees	1401	
% Couples ligne, codeMission non-repertoriees	16	
Missions realisees	61482	
Missions realisees non-repertoriees	2273	
% Missions realisees	3,6	
Groupes de lignes traitees (Fichiers expl3)	269	
Groupes de lignes traitées (BDD SAE-IP)	286	

- Données par type de mission

Type	Missions realisees	% Missions realisees	Missions realisees non-repertoriees	% Missions realisees non-repertoriees	Nb de missions effectuees	% Missions effectuees	Nb de missions effectuees non-repertoriees	% Mission effectuees non-repertoriees
RACCORDEMENT	27	0,3	16	59,2	21	0	21	100
RENTREE	1325	15,1	198	14,9	454	0,7	149	32,8
SORTIE	1068	12,2	278	26	5343	8,6	390	7,2
SPECIAL	1239	14,1	269	21,7	5360	8,7	264	4,9
VOYAGEUR	5081	58,1	640	12,5	50304	81,8	1449	2,8

- Données par groupe de lignes

1er partie

Depot	Ligne	Missions uniques realisees	Missions realisees non-repertoriees	% Missions realisees non-repertoriees	Missions voyageur uniques realisees	Missions voyageur realisees non-repertoriees	% Missions voyageur realisees non-repertoriees
BDM	127	48	15	31,2	33	11	33,3
FLA	133	76	14	18,4	51	9	17,6
AUB	45	35	10	28,5	24	6	25
CXN	39	74	26	35,1	60	18	30
VIT	47	39	18	46,1	26	11	42,3

2ème partie

Nb de missions effectuees	Nb de missions effectuees non-repertoriees	% Mission effectuees non-repertoriees	Nb de missions voyageur effectuees	Nb de missions voyageur non-repertoriees effectuees	% Missions voyageur non repertoriees effectuees	Date de generation du SAE-IP	gener ation differ ee
270	208	77	210	195	92,8	03/06/2025 01:02	NON
323	107	33,1	261	98	37,5	26/03/2025 11:08	NON
226	63	27,8	196	60	30,6	06/05/2025 02:49	NON
233	60	25,7	199	53	26,6	15/04/2025 09:56	NON
348	78	22,4	305	75	24,5	01/06/2025 08:40	NON