

Direct Assurance

Rapport d'activité

3ème année de licence
MIASHS parcours MIAGE

Tuteur Pédagogique : KORNYSHOVA Elena

Tuteur Entreprise : MIAN Balal

TABLE DES MATIÈRES

Remerciements	
Introduction	
1. Présentation de l'entreprise	
1.1 Direct Assurance, filiale d'AXA	
1.1.1 Historique et positionnement sur le marché de l'assurance	
1.1.2 Chiffres clés et enjeux stratégiques	
1.2 L'écosystème AXA et ses filiales	
1.2.1 Les missions d'AXA dans le secteur de la data et de l'innovation	
1.2.2 Focus sur Direct Assurance et son département data	
1.3 Mon service d'affectation	
1.3.1 Organigramme du Pôle Data	
1.3.2 Rôle du service	
1.3.3 Outils utilisés (Power BI, Azure DevOps, GIT, SHINE)	
2. Missions et objectifs de l'alternance	
2.1 Contexte des missions data	
2.1.1 Enjeux du suivi des pipelines de données chez Direct Assurance	
2.1.2 Objectifs du rapport visuel Power BI	
2.2 Cadre de l'alternance	
2.2.1 Durée et rythme	
2.2.2 Compétences visées en Licence MIAGE	
3. Réalisations techniques	
3.1 Apprentissage des outils et compréhension des données	
- Formation : Power BI, DAX (mesures dynamiques fonctions avancées), Git/Azure DevOps (gestion de version)	
- Analyse des pipelines : Cartographie des flux de données	

TABLE DES MATIÈRES

3.2 Développement du rapport Power BI	
3.2.1 Mission 1 : Création 1 ^{er} KPI	
3.2.2 Mission 2 : Ajout de nouvelle Pipeline	
3.2.3 Mission 3 : Ajout du rapport (GIT, AzureDevOps)	
3.2.4 Mission 4 : V1 de l'outils	
3.3 Collaboration et bonnes pratiques	
- Git : Gestion des <i>branches</i> pour les évolutions du rapport	
- Azure DevOps : Suivi des tâches et des bugs via les <i>boards</i>	
4. Bilan et perspectives	
4.1 Compétences acquises	
Techniques :	
• Power BI/DAX	
• Gestion de projet agile	
Transversales : Autonomie, communication technique	
4.2 Résultats	
Gains opérationnels : Réduction du temps de vérification des pipelines	
Améliorations prévues :	
- Ajout de l'envoi d'un mail quotidien	
- Intégration création de graphique pour analyser certaine métrique (temps moyen mois/année, ...etc.)	
Conclusion	

REMERCIEMENTS

Avant d'entamer le développement de ce rapport, je tiens à adresser mes plus sincères remerciements à M. Balal MIAN et Mme Sonia ASSAM ZINEB pour m'avoir recrutée en tant qu'alternant au sein de Direct Assurance, et pour m'avoir accompagné avec pédagogie et bienveillance durant mes débuts. Leur soutien, leur patience et leurs explications ont grandement facilité mon intégration et ma montée en compétences.

Je remercie également chaleureusement l'ensemble de mon équipe et tous les collaborateurs de Direct Assurance pour leur accueil et leur disponibilité au quotidien.

J'adresse également ma gratitude à Madame Elena KORNYSHOVA, ma tutrice pédagogique, pour son suivi régulier tout au long de cette alternance.

Enfin, je souhaite remercier l'équipe pédagogique de l'Université Paris 1 Panthéon-Sorbonne ainsi que le CFA Numia pour leur accompagnement dans la recherche d'une alternance. Une mention particulière à Madame Gaëlle DE COUESSIN, pour son rôle clé dans la mise en relation avec les entreprises et son suivi attentif des apprentis.

MERCI !

INTRODUCTION

Dans le cadre de la validation de ma troisième année de licence MIASSH – parcours MIAGE à l’Université Paris 1 Panthéon-Sorbonne, j’ai effectué une année d’alternance au sein de l’entreprise Direct Assurance.

Cette première immersion dans le monde professionnel m’a permis de découvrir concrètement le secteur de l’assurance, mais aussi de me familiariser avec les enjeux liés à la gestion et à la valorisation des données dans une grande entreprise. Ce fut également l’occasion de mettre en pratique les connaissances théoriques acquises au cours de ma formation universitaire.

Durant cette alternance, j’ai intégré l’équipe Data engineering, chargée de la gestion du Data Lake de Direct Assurance. Cette équipe manipule des données provenant de diverses sources, internes comme externes (notamment les partenaires), afin de les rendre accessibles et exploitables par les différents métiers de l’entreprise. Elle intervient notamment dans des projets liés au Big Data, à l’enrichissement du Data Lake, ou encore à l’évolution de la plateforme data (Azure, ADF, Databricks).

C’est dans ce contexte que j’ai intégré l’équipe data lakehouse, rattachée au Pôle Data, au sein de la Direction Informatique et de l’innovation (D2I).

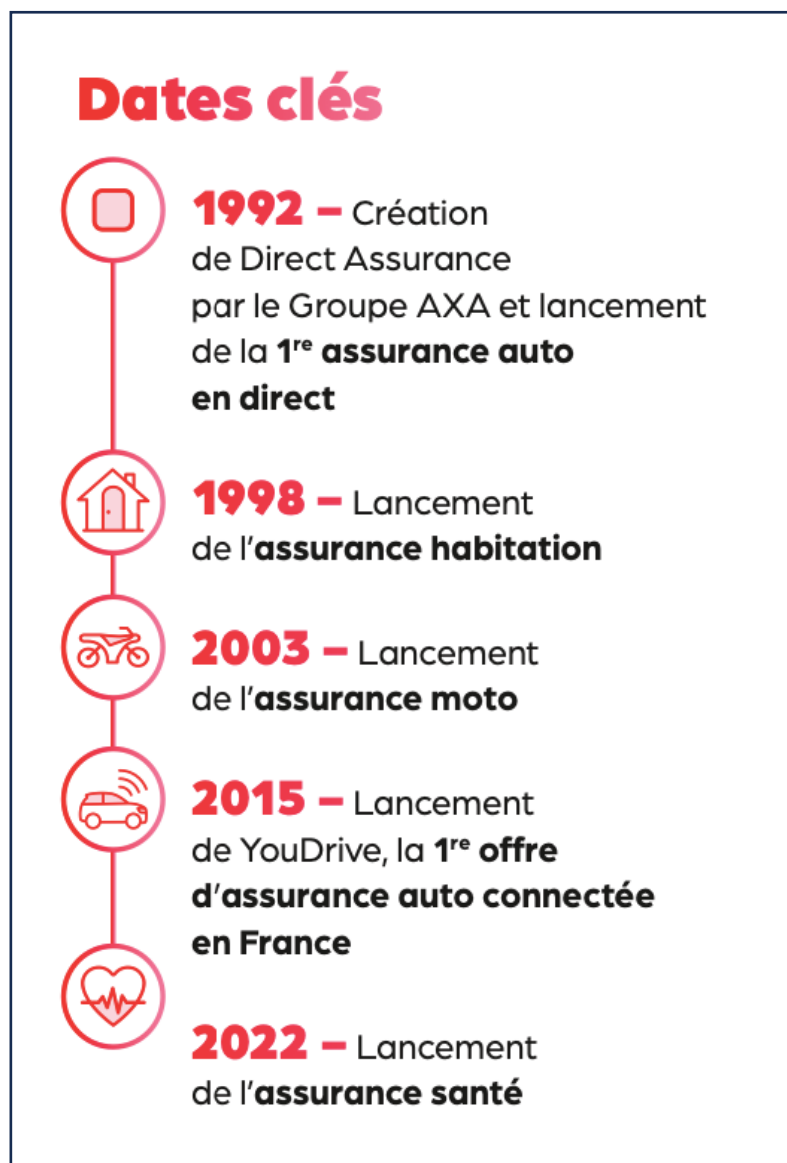
Ma mission principale consistait en la création d’un outil de suivi permettant de visualiser, le bon déroulement des pipelines de données utilisés dans les traitements automatisés.

Ce rapport présente dans un premier temps l’entreprise et son écosystème, puis dans un second temps mes missions, les outils utilisés, les résultats obtenus, ainsi qu’un bilan des compétences développées tout au long de cette année.

1.1 Direct Assurance, filiale d'AXA

1.1.1 Historique et positionnement sur le marché de l'assurance

Direct Assurance a été créée en 1992 avec l'ambition d'innover dans le secteur de l'assurance grâce au digital. Dès ses débuts, l'entreprise a misé sur la simplicité d'usage et la rapidité de souscription en direct, sans passé par un agent, révolutionnant ainsi l'expérience client traditionnelle. Cette démarche disruptive a permis à Direct Assurance de capter une clientèle jeune, au fil du temps connectée et à la recherche d'une gestion autonome de leurs contrats.



En s'appuyant sur l'expertise et la solidité du groupe AXA, Direct Assurance a pu élargir son offre, notamment en assurance auto, habitation, santé et prévoyance. Aujourd'hui, Direct Assurance se positionne comme un acteur incontournable du marché français, reconnu pour ses tarifs compétitifs, sa transparence et sa qualité de service.

Le positionnement de Direct Assurance repose sur trois axes principaux :

- **Digitalisation complète** : Du devis à la gestion des contrats, toutes les démarches s'effectuent en ligne, avec un accès facile via ordinateur ou mobile.
- **Tarification compétitive** : Grâce à l'optimisation des coûts opérationnels et la sophistication des modèles tarifaires, Direct Assurance propose des tarifs attractifs adaptés aux profils des assurés.
- **Orientation client** : Des outils intuitifs, un service client réactif et la mise en place d'offres personnalisées renforcent la satisfaction et la fidélisation.

Face à un marché en pleine mutation, marqué par l'arrivée de nombreuses insurtechs et l'évolution rapide des usages numériques, Direct Assurance mise également sur l'innovation, notamment via l'exploitation de la data et des nouvelles technologies d'intelligence artificielle, pour simplifier le parcours de souscription et réduire les risques de fraudes.

1.1.2 Chiffres clés et enjeux stratégiques

Direct Assurance occupe une place importante sur le marché français de l'assurance en ligne, avec une croissance continue grâce à sa stratégie digitale et son positionnement tarifaire compétitif. Voici quelques données illustrant cette performance :

Chiffres clés

- **N° 1** de l'assurance auto en ligne
- Plus de **350 000** nouveaux clients en 2023
- CA de **735 M€** en 2023
- **1 600** collaborateurs
- **1,3 million** de clients
- Classée « **Assurance la moins chère** »
pour la 2^e année consécutive par Auto-Moto Magazine
- **1,7 million** de contrats pour ses assurances auto, MRH, moto, santé et protection juridique
- **310 €** d'économie globale sur l'assurance automobile et l'assurance habitation constatée par les nouveaux clients (Ipsos 2022-2023)
- Un score de **98/100** à l'index de l'égalité professionnelle femmes-hommes mis en place en 2019 par le ministère du Travail

La stratégie de Direct Assurance repose sur quatre grands piliers :

1. Digitalisation totale du parcours client

Direct Assurance est pionnière dans l'assurance 100 % digitale. Elle a automatisé la quasi-totalité du parcours client : de la souscription à la déclaration de sinistres. Cette approche réduit les coûts, améliore l'expérience client, et permet une adaptation rapide aux évolutions technologiques.

2. Excellence opérationnelle et automatisation

Grâce à des outils comme la data visualisation, l'intelligence artificielle et l'automatisation des process (chatbots, traitement automatique des sinistres, suivi en

temps réel), Direct Assurance optimise la gestion des dossiers et réduit les délais de traitement.

3. Innovation continue et utilisation de la data

L'entreprise investit massivement dans la donnée : collecte, traitement et valorisation des flux de données internes pour :

- Détecter les fraudes
- Anticiper les comportements clients
- Améliorer les offres tarifaires
- Personnaliser les services

4. Expérience client simplifiée

L'objectif est de rendre l'assurance « simple, rapide, et transparente ». Cela passe par des interfaces conviviales (site, application mobile), une assistance réactive, et une communication fluide avec les assurés.

1.2 L'écosystème AXA et ses filiales

1.2.1 Les missions d'AXA dans le secteur de la data et de l'innovation

AXA, en tant que groupe international d'assurance et de gestion d'actifs, place la donnée et l'innovation au cœur de sa stratégie de transformation numérique. Cette vision repose sur plusieurs axes majeurs :

Accélérer la transformation digitale du groupe.

AXA a lancé plusieurs programmes d'envergure mondiale pour moderniser ses systèmes d'information, automatiser les processus internes, et améliorer l'expérience client. Cela inclut :

- Le développement d'un écosystème cloud unifié (via Microsoft Azure),
- L'intégration d'une architecture data globale,
- L'investissement dans des startups insurtech via AXA Venture Partners.

Exploiter la data pour mieux anticiper les risques.

AXA utilise des données massives (big data) pour affiner ses modèles de risques, personnaliser les offres et prévenir les sinistres. Les données sont notamment utilisées pour :

- Analyser les comportements de conduite (télématique auto),
- Anticiper les événements climatiques (catastrophes naturelles),
- Personnaliser le tarif via l'IA.

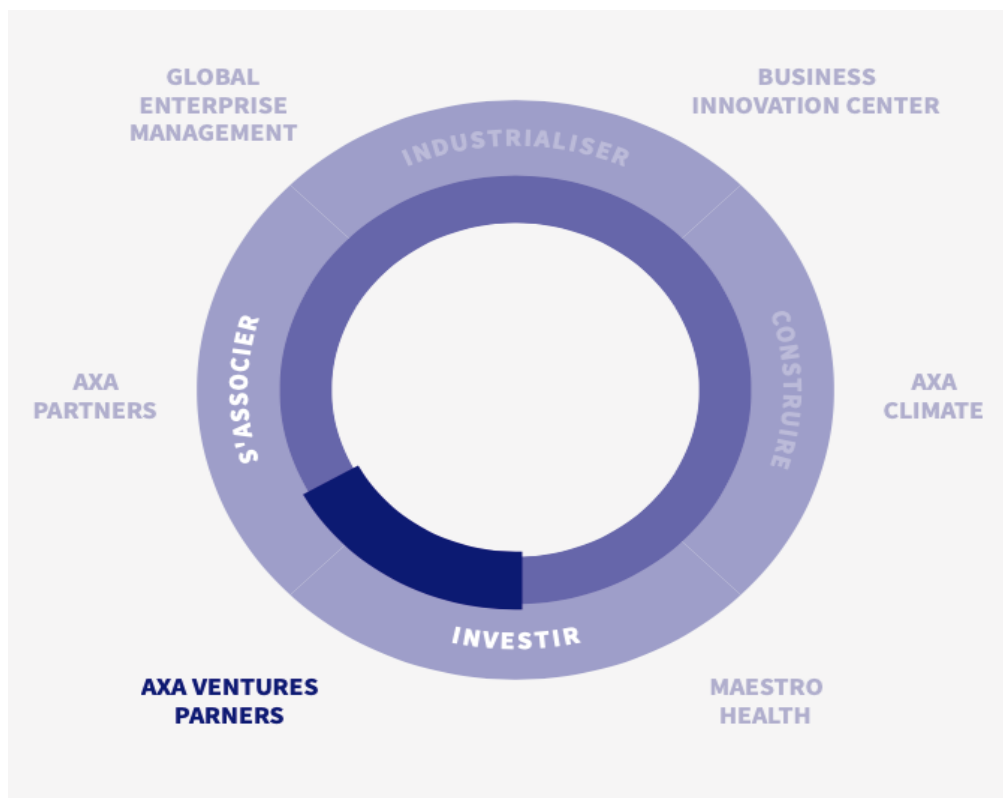
Renforcer la cybersécurité et la gouvernance des données.

Avec la multiplication des données sensibles (santé, mobilité, finance), AXA a renforcé ses dispositifs de protection des données via :

- Une politique stricte de gouvernance des données (RGPD, audibilité),
- Des équipes Data Protection Officers (DPO) par entité,
- Des audits réguliers et une traçabilité complète des flux de données.

Promouvoir l'innovation responsable.

AXA veille à intégrer des principes éthiques dans ses projets technologiques : transparence des algorithmes, impact environnemental du numérique, inclusion des publics fragiles.



AXA Next – un écosystème composé de 6 unités pour accélérer l'innovation au sein du Groupe

1.2.2 Focus sur Direct Assurance et son département data

Au sein de cet écosystème, Direct Assurance joue un rôle clé dans la mise en œuvre opérationnelle des stratégies data du groupe, en particulier grâce à son département data. Celui-ci est structuré autour de plusieurs missions :

- **Collecte et traitement des données** : Les équipes ingèrent de grandes quantités de données issues de multiples sources. Ces données concernent les contrats, les sinistres, les clients, les interactions web, etc.
- **Surveillance et qualité des pipelines de données** : Le département assure le suivi rigoureux des flux de données (pipelines), détecte et corrige les anomalies pour garantir la fiabilité des analyses.
- **Analyse et exploitation** : Les data analysts et data scientists produisent des rapports, des indicateurs clés (KPIs), et des modèles prédictifs qui alimentent les décisions stratégiques et opérationnelles.

Les missions du département data de Direct Assurance permettent de mieux comprendre l'environnement dans lequel s'inscrit mon alternance.

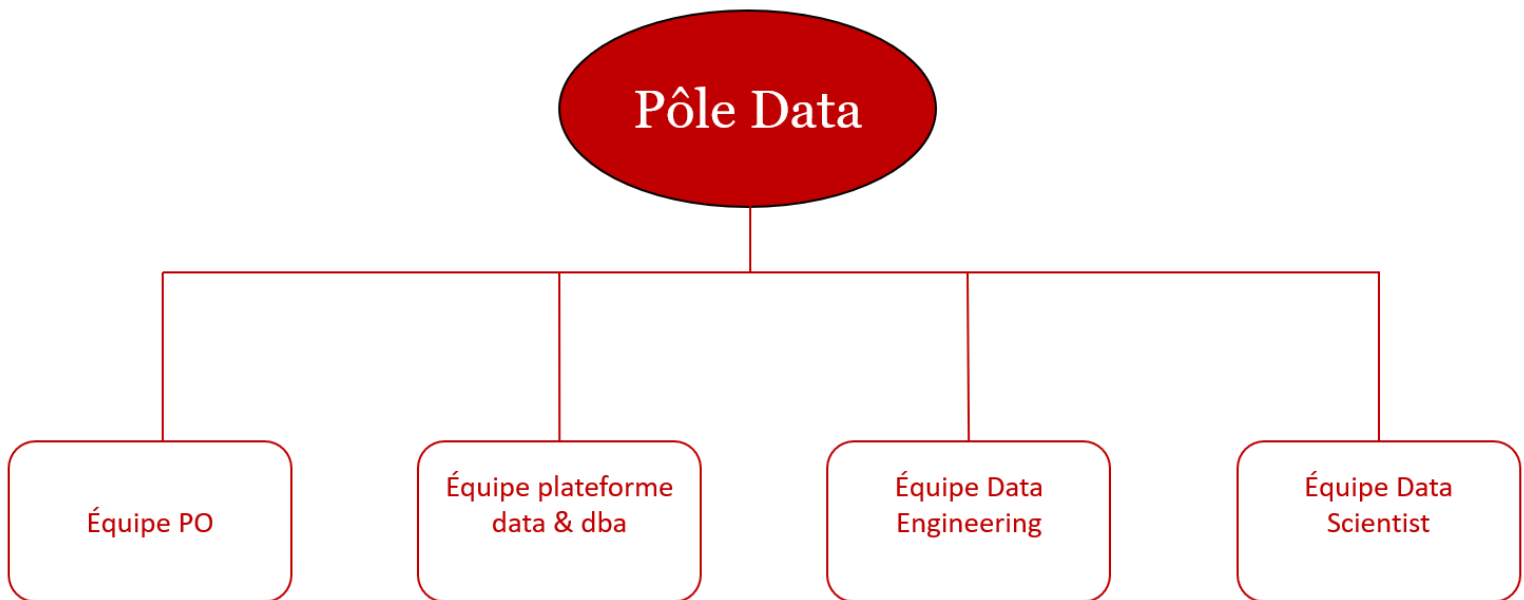
La section suivante présente plus en détail le pôle data, mon service d'affectation, ses missions spécifiques, sa place dans l'organigramme data, ainsi que les outils mobilisés dans le cadre de mes projets.

1.3 Mon service d'affectation

1.3.1 Organigramme du Pôle Data

Le Pôle Data de Direct Assurance est structuré autour de plusieurs équipes spécialisées, chacune jouant un rôle clé dans la gestion du cycle de vie des données. Ce pôle est rattaché à la Direction Informatique et de l'innovation (D2I).

Voici une représentation simplifiée de son organisation fonctionnelle :



Je suis rattaché à l'équipe Data Engineering, composée de plusieurs ingénieurs spécialisés dans la gestion des flux de données, la surveillance des pipelines, et la mise en production des traitements automatisés.

1.3.2 Rôle du service

L'équipe Data Engineering est chargée de la collecte, transformation et mise à disposition des données. Elle assure également le bon fonctionnement des pipelines de données. Elle agit en amont des équipes BI et data science, en assurant la disponibilité, la fiabilité et la performance des flux.

Missions principales

- Ingestion et transformation des données issues de diverses sources (bases de données relationnelles (SQL Server et Oracle), fichiers plats, API partenaires).
- Construction de pipelines de traitement (flux automatisés) pour alimenter le Data Lake.
- Surveillance et détection des anomalies
- Optimisation de la performance des flux pour réduire les temps de traitement et les erreurs.

- Documentation technique et suivi des évolutions, en lien avec la gouvernance data et les métiers.

Cette équipe est en interaction constante avec les équipes métiers, BI et Data Science. Elle est sollicitée dès qu'il y a :

- Un nouveau besoin en données pour un projet ou un rapport,
- Une anomalie détectée dans un pipeline ou un chargement,
- Une évolution nécessaire dans la structure des flux ou leur fréquence.

1.3.3 Outils utilisés

L'environnement technique de l'équipe Data Engineering repose sur un ensemble d'outils robustes et collaboratifs :

Outils	Usages
Azure DevOps	Suivi des tâches, documentation des anomalies, gestion des sprints.
Git	Contrôle de version du code (scripts de transformation, configuration des flux).
SIMON / TWS	Supervision des pipelines : alertes, logs, erreurs, suivi journalier.
Power BI	Visualisation de l'état des pipelines (via les rapports).
Databricks	Environnement de traitement de données à grande échelle : développement de pipelines,

	transformations distribuées, exécution de jobs.
Bases de données relationnelles (SQL Server et Oracle)	Requêtage des bases de données sources pour extraction ou contrôle qualité.

2. Direct Assurance, filiale d'AXA

J'ai rejoint l'équipe Data Engineering pour concevoir un outil de monitoring des pipelines. Celui-ci a pour objectifs de :

- Connaître le statut des pipelines et de disposer des informations nécessaires pour débayer en cas de plantage.
- Anticiper les problèmes de performances et donc d'agir de manière proactive.
- Communiquer auprès des utilisateurs sur la disponibilité des données.

2.1 Direct Assurance, filiale d'AXA

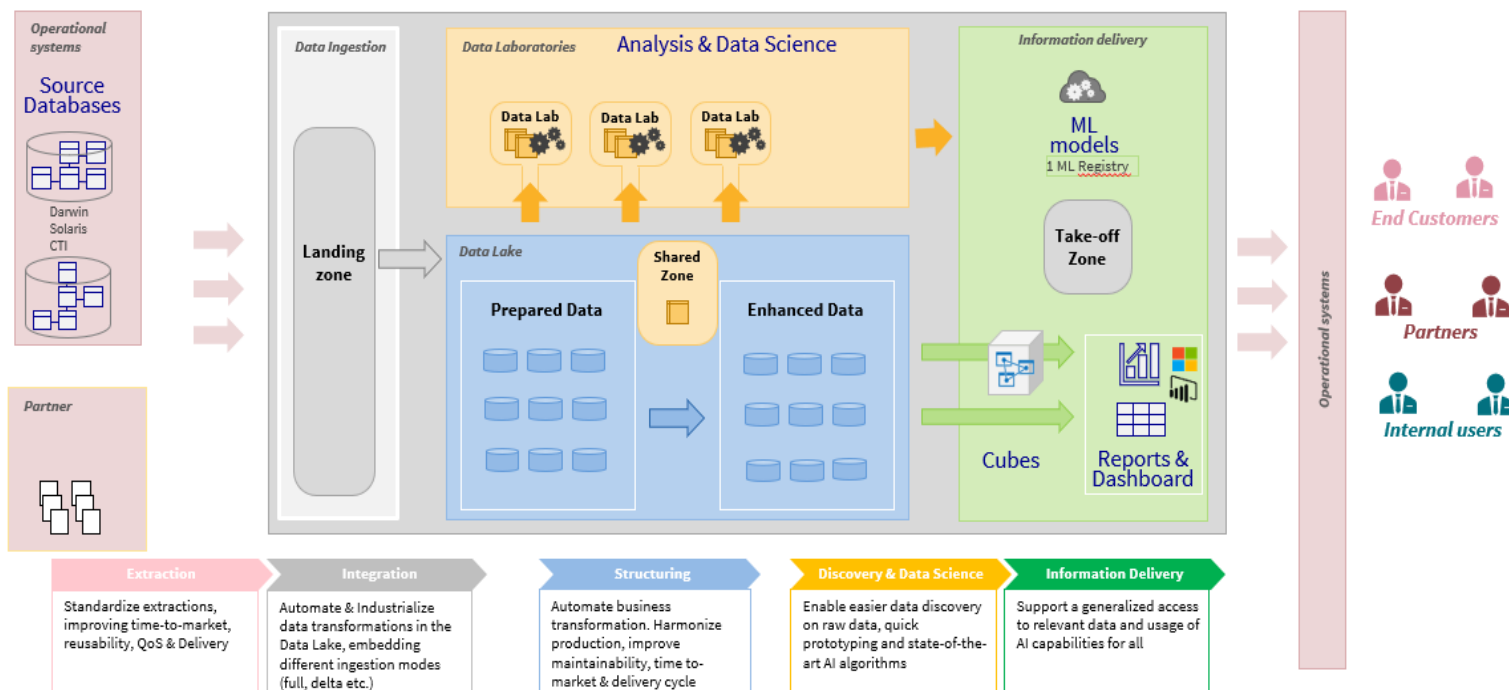
2.1.1 Enjeux du suivi des pipelines de données chez Direct Assurance

Dans un environnement digitalisé comme celui de Direct Assurance, la donnée constitue un levier stratégique pour le pilotage de l'activité, la gestion des risques et l'innovation produit.

Les pipelines de données sont des processus automatisés qui permettent :

- D'ingérer des données issues de multiples sources (bases de données internes, fichiers plats, API partenaires),
- De les transformer selon des règles définies (nettoyage, normalisation, enrichissement),

- Puis de les charger dans des bases prêtes à l'analyse ou dans le Data Lake.



Ces pipelines sont à la base :

- Des rapports Power BI utilisés quotidiennement par les directions métier (marketing, actuariat, pilotage client),
- Des modèles développés par les data scientifiques.
- Des outils de supervision.

Enjeu	Conséquences si pipeline défaillant
Mise à jour des rapports décisionnels	Données obsolètes → erreurs de pilotage stratégique
Conformité réglementaire (Solvabilité 2)	Risques de non-conformité, sanctions potentielles

Détection de fraude ou d'incidents	Retards dans l'identification d'événements critiques
Réputation interne du service data	Perte de confiance des utilisateurs métiers dans les rapports livrés

2.1.2 Objectifs du rapport visuel Power BI

Afin de renforcer la transparence et la supervision des pipelines, l'équipe Data Engineering m'a confié la création d'un rapport Power BI de suivi.

Objectifs principaux du rapport :

- Visualiser l'état d'exécution des pipelines (succès, erreurs, retards),
- Filtrer les données par date, plage horaire, famille de flux, ou statut,
- Mettre en évidence les anomalies récurrentes ou critiques (flux en échec plusieurs jours consécutifs),
- Accroître la réactivité de l'équipe grâce à une meilleure priorisation des incidents,

Contenu du rapport :

- Liste des pipelines par ordre d'exécution.
- Date / heure de début et de fin de chaque traitement.
- Le statut (OK/WARNING/KO), avec la couleur selon le statut.
- La durée totale entre le premier traitement et le dernier (Start / End).

Voici un exemple :

OK	
Start	End
20:35:40	20:43:53
00:08:13	
TimeDuration	

2.2 Cadre de l'alternance

2.2.1 Durée et rythme

L'alternance s'est déroulée sur une période de 12 mois, du 16 septembre 2024 au 3 septembre 2025, dans le cadre du contrat d'apprentissage en Licence MIASHS parcours MIAAGE.

- Structure académique : Université Paris 1 Panthéon-Sorbonne
- Rythme d'alternance :
 - 2 semaines : Cours à l'université
 - 2 semaines : Présence en entreprise

Ce rythme m'a permis de progresser régulièrement sur les projets confiés tout en approfondissant mes compétences théoriques et pratiques.

Objectifs pédagogiques définis :

- Appliquer les acquis universitaires dans un contexte professionnel,
- Participer à des projets concrets de traitement de données,
- Renforcer l'autonomie et la rigueur professionnelle,
- Se former aux outils et méthodes utilisés en entreprise.

2.2.2 Compétences visées en Licence MIAGE

L'alternance est l'occasion de développer des compétences alignées avec les attendus pédagogiques de la formation MIAGE (Méthodes Informatiques Appliquées à la Gestion des Entreprises).

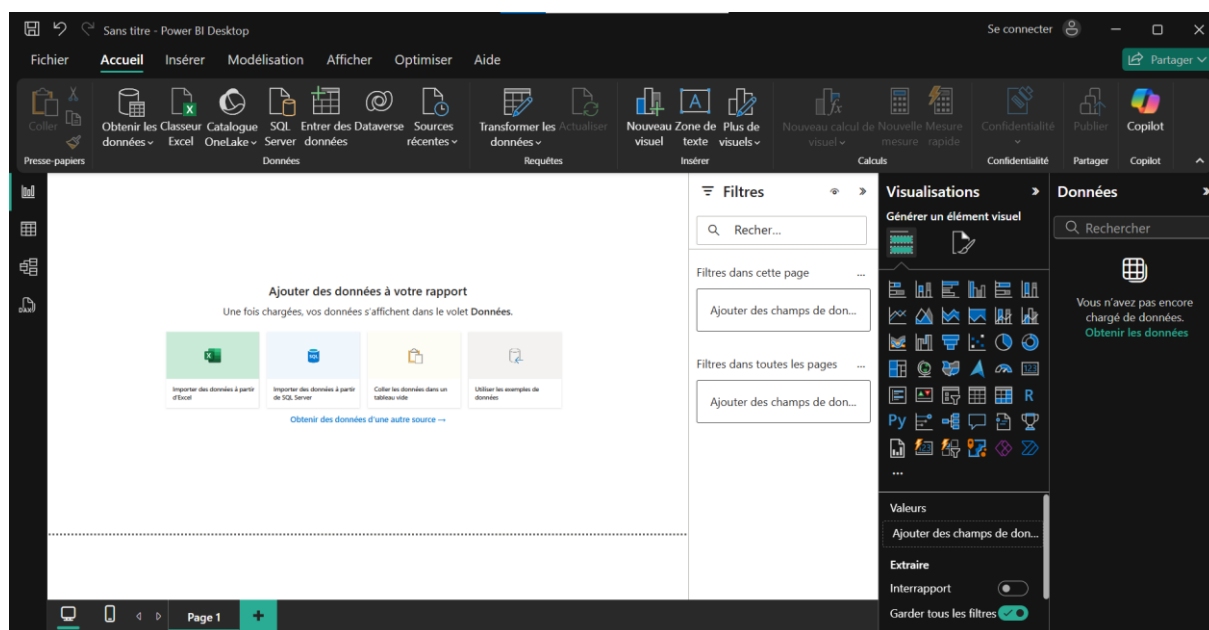
Capacités/Compétences	Blocs de compétences
1.1 Identifier les usages numériques et les impacts de leur évolution sur le ou les domaines concernés par la mention	Utiliser les outils numériques de référence (Bloc 1)
1.2 Utiliser les outils numériques de référence et les règles de sécurité informatique pour acquérir, traiter, produire et diffuser de l'information ainsi que pour collaborer en interne et en externe.	
2.1 Identifier, sélectionner et analyser avec esprit critique diverses ressources dans son domaine de spécialité pour documenter un sujet.	Exploiter les données à des fins d'analyse (Bloc 02)
2.2 Analyser et synthétiser des données en vue de leur exploitation.	
2.3 Développer une argumentation avec esprit critique.	
3.1 Se servir aisément des différents registres d'expression écrite et orale de la langue française.	S'exprimer et communiquer à l'oral, à l'écrit, et dans au moins une langue étrangère (Bloc 03)
3.2 Communiquer par oral et par écrit, de façon claire et non-ambiguë, dans au moins une langue étrangère.	

3. Réalisations Techniques

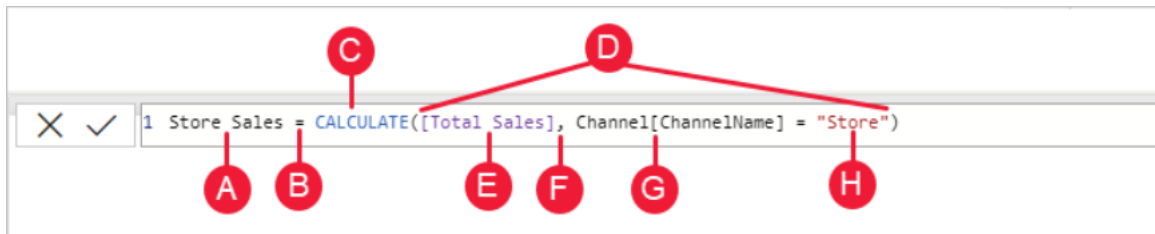
3.1 Apprentissage des outils et compréhension des données

Dès les premières semaines de mon alternance, une phase de montée en compétences a été nécessaire pour comprendre à la fois l'environnement technique de l'équipe Data Engineering et la logique des pipelines en place. Cette phase a été progressive et structurée, mêlant observation, autoformation, accompagnement par les ingénieurs du pôle, et mise en pratique sur des jeux de données réels.

L'un des premiers outils que j'ai dû appréhender a été Power BI, bien que n'étant pas l'outil principal de l'équipe Data Engineering. Je l'ai utilisé pour concevoir l'interface de supervision du rapport, en lien avec les données issues des pipelines. J'ai commencé par m'initier à l'interface de Power BI Desktop, à la logique des modèles de données, puis à la syntaxe DAX utilisée pour créer des indicateurs et des mesures dynamiques. Cette phase m'a permis de comprendre comment transformer des données brutes en visualisations claires et interactives, capables de répondre à des besoins de pilotage opérationnel.



Images – rapport Power BI Desktop vide



Images – Mesure en DAX (Data Analysis Expressions)

Cette formule comprend les éléments de syntaxe suivants :

A. Le nom de la mesure, Total des ventes.

B. L'opérateur de signe égal (=), qui indique le début de la formule. Après le calcul, un résultat est retourné.

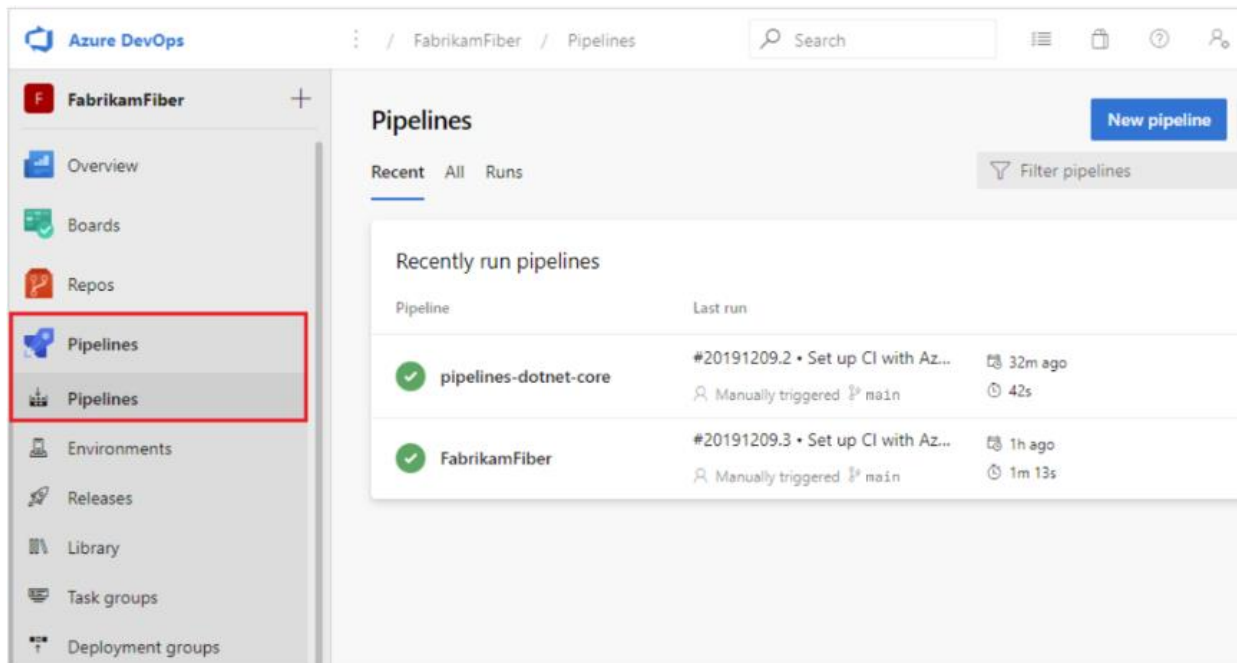
C. La fonction DAX SUM, qui additionne tous les nombres dans la colonne Sales[SalesAmount]. Vous en apprendrez plus sur les fonctions ultérieurement.

D. Parenthèses (), qui entourent une expression qui contient un ou plusieurs arguments. La plupart des fonctions nécessitent au moins un argument. Un argument passe une valeur à une fonction.

E. La table référencée, Ventes.

F. Colonne référencée, [SalesAmount], dans la table Sales. Avec cet argument, la fonction SUM sait sur quelle colonne agréger une somme.

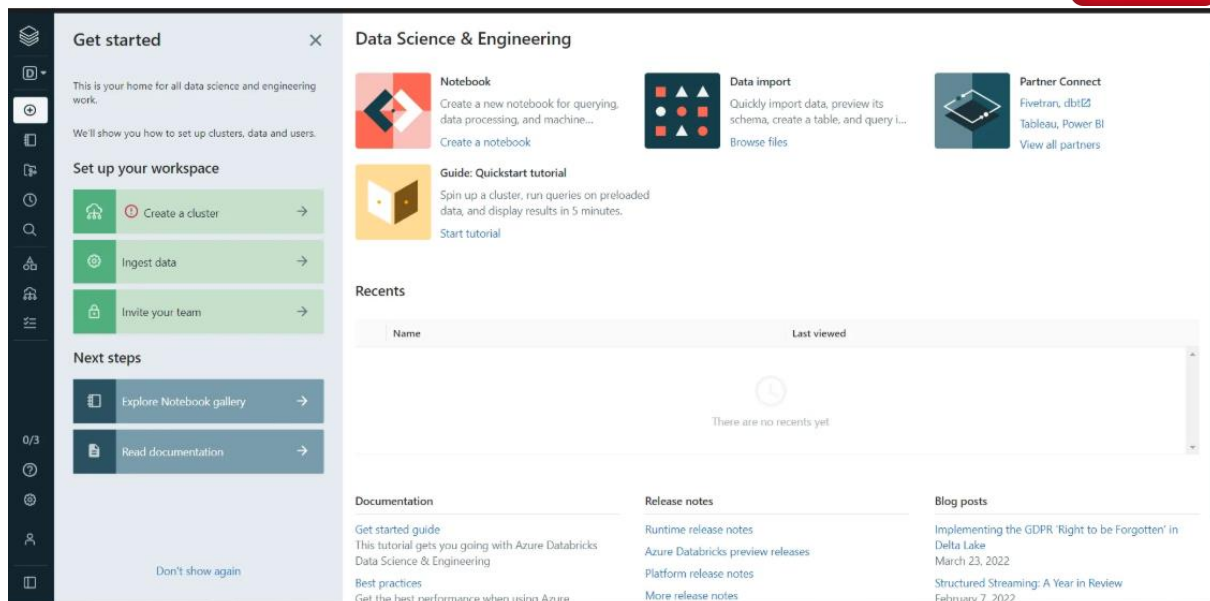
En parallèle, j'ai découvert l'environnement Azure DevOps, qui structure le travail de l'équipe selon une logique agile. Très vite, j'ai été amené à suivre les tâches de développement, à mettre à jour des tickets d'évolution, et à contribuer à la documentation du projet Cockpit dans le wiki interne. Azure DevOps m'a également permis de collaborer efficacement avec mes collègues grâce à son intégration avec Git, outil fondamental dans notre gestion de versions. J'ai appris à créer mes propres branches, à versionner les fichiers (notamment les fichiers.pbix du rapport Power BI), à faire des pull requests, et à suivre des cycles de revue de code avant chaque mise en production.



Images – interface Azure DevOps

Pour mieux comprendre les flux de données en jeu dans les pipelines, j'ai travaillé au quotidien avec l'outil interne de monitoring SIMON. Cet outil permet de visualiser l'ensemble des pipelines exécutés, avec leurs statuts (succès, erreur, en attente), leur durée, et les éventuels messages d'erreur. Grâce à SIMON, j'ai acquis une vision globale du fonctionnement des traitements automatisés de données chez Direct Assurance. Cette compréhension a été essentielle pour définir la structure du tableau de bord que j'ai développé.

Une autre partie importante de mon apprentissage a été la maîtrise de Databricks, une plateforme de traitement distribué fondée sur Apache Spark. Bien que principalement utilisée par les data engineers, j'ai pu accéder à plusieurs notebooks et comprendre comment les transformations complexes sont mises en œuvre à grande échelle. J'ai réalisé des manipulations simples en Python dans Databricks, notamment pour filtrer, agréger et enrichir des fichiers plats ou des tables issues du Data Lake. Cette expérience m'a sensibilisé aux problématiques de performance, de volume, et à la logique des traitements par batch.



Images – Databricks

Cette première phase d'immersion technique a été très formatrice. Elle m'a permis de mieux comprendre l'architecture data de Direct Assurance, tout en développant ma capacité à naviguer entre plusieurs outils et à faire le lien entre des traitements techniques complexes et les enjeux métiers qui en dépendent.

3.2 Développement du rapport Power BI

3.2.1 Mission 1 : Création du premier KPI

Ma première mission concrète a été centrée sur la création d'un indicateur de supervision du traitement Téléphonie, un flux métier exécuté chaque nuit. Ce traitement permet d'alimenter des données exploitées par les équipes opérationnelles, il est donc impératif de vérifier quotidiennement son bon déroulement.

Le KPI à produire devait refléter trois statuts possibles :

1. OK : si le traitement s'était exécuté entre 00h00 et 01h30 à la date du jour sans erreur,

OK	
Start	End
TimeDuration	

2. WARNING : si le traitement avait démarré après 01h30, tout en se terminant sans échec,

WARNING	
Start	End
TimeDuration	

3. KO : si le traitement ne s'était pas déclenché du tout ou s'était terminé avec une erreur.

KO	
Start	End
TimeDuration	

Pour construire ce KPI, j'ai d'abord analysé les données disponibles dans les logs SHINE, à la recherche des éléments exploitables : heure d'exécution, statut final, date de fin.

StartTime	Status	EndTime	Duration	Error_msg
08/04/2025 06:43:45	SUCCESS	08/04/2025 06:43:50	5	
08/04/2025 06:43:45	SUCCESS	08/04/2025 06:43:50	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:40	SUCCESS	08/04/2025 06:43:45	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	
08/04/2025 06:43:35	SUCCESS	08/04/2025 06:43:40	5	

J'ai ensuite utilisé Power BI pour créer des mesures DAX intermédiaires permettant de capturer la dernière exécution, de vérifier qu'elle correspondait à la date du jour, puis d'évaluer si des erreurs étaient présentes. Ces mesures ont été combinées dans une mesure finale restituant dynamiquement le statut (OK, KO ou WARNING) selon la logique définie.

Création d'une mesure permettant de récupérer la date/heure du traitement de la téléphonie :

```
LastDate = MAXX(FILTER(ADF_logs, ADF_logs[PipelineName]=""), ADF_logs[GenerationTime]) + MAXX(FILTER(ADF_logs, ADF_logs[GenerationTime].[Date]), ADF_logs[GenerationTime])
```

Après on utilise cette mesure, dans une autre pour créer de nouvelles variables à partir de cette donnée :


```

LastTreatment =
VAR LastTreatmentYear = YEAR(Logs_analytics[LastDate])
VAR LastTreatmentMonth = MONTH(Logs_analytics[LastDate])
VAR LastTreatmentDay = DAY(Logs_analytics[LastDate])
VAR LastTreatmentDate = DATE(LastTreatmentYear, LastTreatmentMonth, LastTreatmentDay)

VAR LastTreatmentHour = HOUR(Logs_analytics[LastDate])
VAR LastTreatmentMinute = MINUTE(Logs_analytics[LastDate])
VAR LastTreatmentSecond = SECOND(Logs_analytics[LastDate])
VAR LastTreatmentTime = TIME(LastTreatmentHour, LastTreatmentMinute, LastTreatmentSecond)
VAR TodayDate = Logs_analytics[Today]

```

Ici on crée des variables pour récupérer à partir des méthodes adéquates l'année, le mois, le jour pour reconstituer la date sans l'heure.

Après même chose, mais cette fois si pour l'heure.

Et pour terminer on fait les différents tests pour savoir si tout est OK :

```

RETURN
IF(ISBLANK(SELECTEDVALUE(ADF_logs[PipelineName])),
    "NULL",
    IF(SELECTEDVALUE(ADF_logs[PipelineName]) = "telephonie_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
        "telephonie_ods_cti_export_incremental",
        IF(LastTreatmentDate = TodayDate,
            IF(LastTreatmentTime > TIME(1,30,0),
                "KO",
                Logs_analytics[LastStatus]
            )
        ),
    ),
)

```

Là on teste pour voir si le traitement s'était exécuté entre 00h00 et 01h30 à la date du jour. Si oui alors on passe à la mesure suivante qui fera les autres tests, sinon c'est « KO ».

```

LastStatus =
VAR LastTreatmentStatus = CONVERT(MAXX(FILTER(ADF_logs, ADF_logs[PipelineName]=""), ADF_logs[Status]), DATETIME)
VAR FailTreatmentStatus = CONVERT(CALCULATE(COUNTROWS(FILTER(ADF_logs, ADF_logs[PipelineName]="")), ADF_logs
[Status] = "FAILURE", ADF_logs[GenerationTime] = LastTreatmentStatus), DATETIME)

```

La prochaine mesure LastStatus, permet de récupérer pour chaque statut de chaque traitement (SUCCESS ou FAILED). Si un traitement a échoué (FAILED) alors il sera automatiquement KO.

```
RETURN
IF(SELECTEDVALUE(ADF_logs[PipelineName]) = "telephonie",
    IF(FailTreatmentStatus > 0,
        "KO",
        IF( FailTreatmentStatus = 0,
            "OK",
            "Warning"
        )
    ),
),
```

Cette mission m'a servi d'introduction concrète à Power BI. J'ai appris à structurer un modèle de données, à écrire mes premières expressions DAX, à exploiter des filtres contextuels, et à construire une logique métier avancée à partir d'une base de logs. Ce premier exercice m'a permis de rentrer dans le sujet du monitoring tout en posant les fondations de l'interface de supervision que j'allais étoffer dans les missions suivantes.

3.2.2 Mission 2 : Intégration de nouveaux pipelines

Une fois les indicateurs Téléphonie stabilisés, j'ai été chargé de les intégrer dans le rapport principal, en les adaptant au modèle plus global. Ce transfert technique a nécessité un travail de nettoyage et d'harmonisation du code DAX, ainsi que l'intégration des visuels dans une nouvelle page du rapport, dédiée aux traitements critiques.

En parallèle, j'ai dû développer un système d'indicateurs pour deux autres flux métiers appelés Solaris, composés de deux sous-traitements distincts : LKWH et SLKH. La logique métier était plus complexe : les deux traitements doivent s'exécuter l'un après l'autre. Plusieurs cas de figure pouvaient se présenter :

1. Si les deux s'exécutent correctement dans le bon ordre, le statut est OK.
2. Si l'un d'eux échoue ou que l'ordre est inversé, alors c'est un WARNING.
3. Si aucun des deux ne s'exécute ou s'ils tombent en erreur, alors le statut passe en KO.

Pour modéliser ce scénario, j'ai utilisé une combinaison de mesures DAX conditionnelles, prenant en compte les statuts, les ordres d'exécution et les dates. Ce travail m'a permis de renforcer ma logique algorithmique, en apprenant à traduire une règle métier complexe en expressions analytiques exploitables dans Power BI.

La seule condition qui change par rapport à la téléphonie, c'est l'ajout d'une mesure pour voir si les deux traitements distincts LKWH et SLKH s'exécute dans le bon ordre :

```
IF(SELECTEDVALUE(ADF_logs[PipelineName]) = "solaris_clean_sources" || SELECTEDVALUE(ADF_logs
[PipelineName]) = "solaris_copy_lds_table_from_lz" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"solaris_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "solaris_runners",
    IF(FailTreatmentStatusSolarisCleanSources > 0 && FailTreatmentStatusSolarisMaster > 0 &&
FailTreatmentStatusSolarisCopy > 0 && FailTreatmentStatusSolarisRunners > 0,
        "KO",
        IF(FailTreatmentStatusSolarisCleanSources > FailTreatmentStatusSolarisMaster &&
FailTreatmentStatusSolarisCopy && FailTreatmentStatusSolarisRunners,
            "Warning",
            IF(FailTreatmentStatusSolarisCleanSources > 0 || FailTreatmentStatusSolarisMaster > 0 ||
FailTreatmentStatusSolarisCopy > 0 || FailTreatmentStatusSolarisRunners > 0,
                "Warning",
                "OK"
            )
        )
    ),
```

De plus, l'une des autres problématiques rencontrées est le fait de voir le statut global de tous les pipelines de Sitecom et non de chaque entité une à une. J'ai donc créé une mesure à part permettant de voir le statut global de toute la partie sitecom :

```
VAR TodayDate = TODAY()
VAR Hour = TIME(HOUR(NOW()), MINUTE(NOW()), SECOND(NOW()))

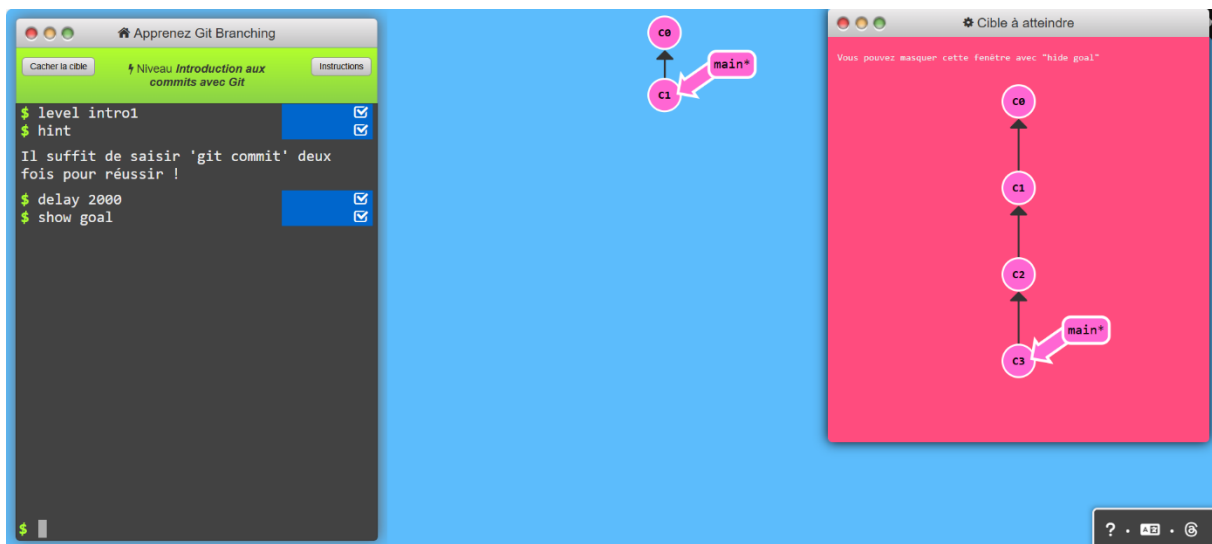
RETURN
    IF(FailTreatmentStatusSolarisCleanSources > 0 && FailTreatmentStatusSolarisMaster > 0 && FailTreatmentStatusSolarisCopy > 0 &&
FailTreatmentStatusSolarisRunners > 0,
        "KO",
        IF(FailTreatmentStatusSolarisCleanSources > FailTreatmentStatusSolarisMaster && FailTreatmentStatusSolarisCopy &&
FailTreatmentStatusSolarisRunners,
            "Warning",
            IF(FailTreatmentStatusSolarisCleanSources > 0 || FailTreatmentStatusSolarisMaster > 0 || FailTreatmentStatusSolarisCopy > 0 ||
FailTreatmentStatusSolarisRunners > 0,
                "Warning",
                "OK"
            )
        )
    )
```

Enfin, cette mission incluait également l'intégration du projet dans Azure DevOps, avec le dépôt du rapport.pbix sous Git dans le repository de l'équipe. Cela m'a demandé de me former aux bases de Git : initialiser un dépôt, créer une branche, effectuer des commits, et synchroniser mon travail avec Visual Studio. Cette étape a marqué mon passage d'un développement local à un cycle de gestion de version collaboratif, rigoureux et traçable.

3.2.3 Mission 3 : Publication, versioning et gestion collaborative du rapport

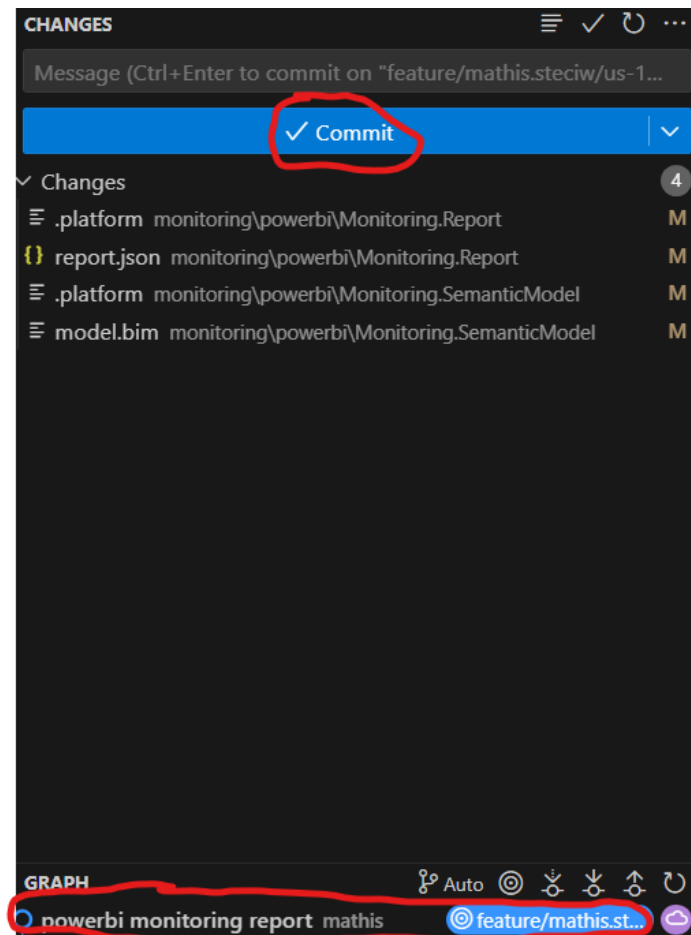
Dans la continuité, j'ai poursuivi l'apprentissage du versioning Git avec pour objectif d'intégrer formellement mon rapport Power BI dans le cycle DevOps de l'équipe. Cette mission m'a amené à comprendre comment gérer un fichier.pbix versionné via Visual Studio et Azure DevOps.

J'ai commencé par explorer des ressources pédagogiques pour bien comprendre le fonctionnement de Git : son modèle de branches, les commandes de base, le processus de merge, et les bonnes pratiques de commit.



Images – site [learngitbranching](https://www.learnitbranching.com/)

J'ai ensuite mis en application ces principes en créant ma propre branche pour développer en autonomie.



Une fois une évolution validée (comme une modification de visuel ou l'ajout d'un indicateur), je réalisais un commit descriptif, suivi d'un pull request vers la branche de recette.

Ce processus m'a permis de me conformer aux pratiques d'équipe tout en garantissant la qualité et la stabilité du rapport publié.

3.2.4 Mission 4 : Livraison d'une première version aboutie du rapport (V1)

La dernière mission, et sans doute la plus exigeante, a concerné l'intégration des traitements Sitecom dans le rapport de monitoring. Sitecom est l'un des ensembles de traitements les plus volumineux de l'écosystème data de Direct Assurance. Il regroupe plusieurs flux interdépendants, exécutés quotidiennement. Leur supervision est essentielle, car une défaillance a un impact direct sur les restitutions aux métiers.

J'ai d'abord développé une version test du rapport dédié à Sitecom, afin de pouvoir expérimenter sans impacter le rapport principal. J'ai créé des mesures DAX

similaires à celles déjà développées pour Téléphonie et Solaris, mais adaptées à la structure particulière des flux Sitecom. La logique de validation reposait sur deux étapes : d'abord vérifier que tous les traitements avaient tourné à la date du jour, puis évaluer s'ils s'étaient tous terminés en succès (OK), ou si des erreurs ou des retards étaient survenus (KO ou WARNING).

Cette première mesure permet, comme pour la téléphonie et Solaris, de vérifier la date à laquelle le dernier traitement s'est terminé :

```
IF(SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_accounting_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_accumulator_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_accumulator_purge_master" || SELECTEDVALUE
(ADF_logs[PipelineName]) = "sitecom_claims_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_common_master" ||
SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_ctr_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_external_elynx_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_external_lesfurets_master" || SELECTEDVALUE
(ADF_logs[PipelineName]) = "sitecom_external_novagile_agent_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_external_novagile_competence_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "SitecomHealthClaimsSecret" ||
SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_external_novagile_mail_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_extract_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_health_master"
|| SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_home_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_init_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_motor_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_score_master" || SELECTEDVALUE(ADF_logs
[PipelineName]) = "accounting_cube_monthly_master" ,
    IF(LastTreatmentDate == TodayDate || LastTreatmentDate <> TodayDate,
        Logs_analytics[LastStatus],
        "KO"
    )
)
```

La suivante, qui se trouve de même dans la mesure LastStatus, qui est la suite des vérifications, où l'on peut retrouver la condition pour la partie Sitecom qui vérifie également s'il y a eu des traitements entre le premier et le dernier (inclus) qui ont eu pour résultat FAILED :

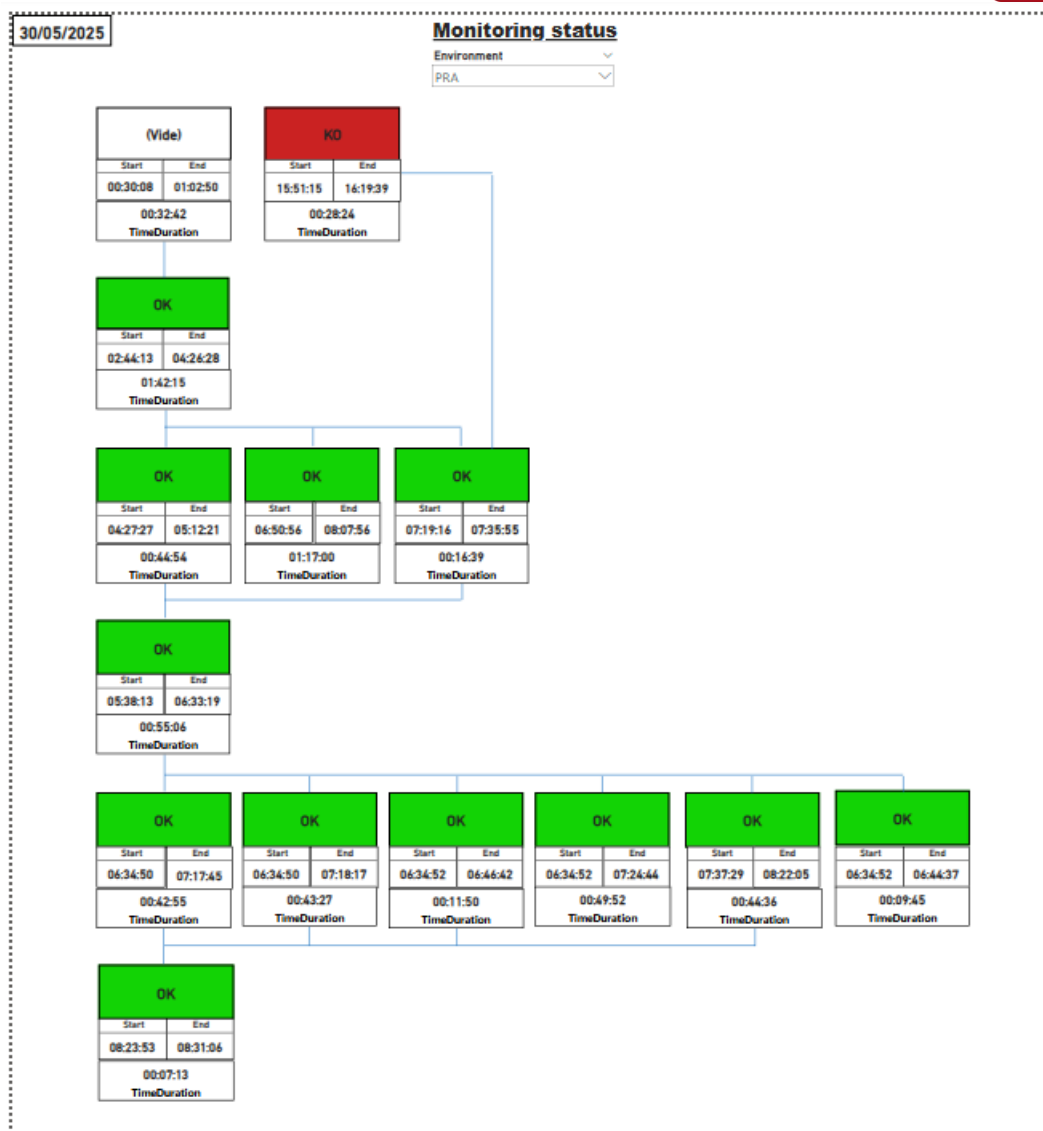
```
IF(SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_accounting_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_accumulator_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_accumulator_purge_master" || SELECTEDVALUE(ADF_logs
[PipelineName]) = "sitecom_claims_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_common_master" || SELECTEDVALUE
(ADF_logs[PipelineName]) = "sitecom_ctr_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_external_elynx_master" ||
SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_external_lesfurets_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_external_novagile_agent_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_external_novagile_competence_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_external_novagile_mail_master" ||
SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_extract_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_health_master"
|| SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_home_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_init_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_master" || SELECTEDVALUE(ADF_logs[PipelineName]) =
"sitecom_motor_master" || SELECTEDVALUE(ADF_logs[PipelineName]) = "sitecom_score_master" || SELECTEDVALUE(ADF_logs
[PipelineName]) = "accounting_cube_monthly_master",
    IF(FailTreatmentStatus > 0,
        "KO",
        IF( FailTreatmentStatus = 0,
            "OK",
            "Warning"
        )
    )
)
```

La dernière mesure qui est Sitecom Statut permet, comme pour la partie Solaris, de voir le statu global de Sitecom, est non le statut de chaque pipeline une à une :

```
RETURN
IF(FailTreatmentStatusSitecomAcc > 0 && FailTreatmentStatusSitecomAccu > 0 && FailTreatmentStatusSitecomAccuPu > 0 &&
FailTreatmentStatusSitecomCommon > 0 && FailTreatmentStatusSitecomClaim > 0 && FailTreatmentStatusSitecomCtr > 0 &&
FailTreatmentStatusSitecomNovaMail > 0 && FailTreatmentStatusSitecomExtract > 0 && FailTreatmentStatusSitecomLeLyx > 0 &&
FailTreatmentStatusSitecomHealth > 0 && FailTreatmentStatusSitecomLesFurets > 0 && FailTreatmentStatusSitecomHome > 0 &&
FailTreatmentStatusSitecomInit > 0 && FailTreatmentStatusSitecomNovaCpt > 0 && FailTreatmentStatusSitecomNovaAgt > 0 &&
FailTreatmentStatusSitecomMaster > 0 && FailTreatmentStatusSitecomMotor > 0 && FailTreatmentStatusSitecomScore > 0,
    "KO",
    IF(LastTreatmentTimeSitecomAcc > LastTreatmentTimeSitecomAccu && LastTreatmentTimeSitecomAccuPu && LastTreatmentTimeSitecomClaim &&
LastTreatmentTimeSitecomCommon && LastTreatmentTimeSitecomCtr && LastTreatmentTimeSitecomExtract &&
LastTreatmentTimeSitecomHealth && LastTreatmentTimeSitecomHome && LastTreatmentTimeSitecomInit && LastTreatmentTimeSitecomLeLyx &&
LastTreatmentTimeSitecomLesFurets && LastTreatmentTimeSitecomMaster && LastTreatmentTimeSitecomMotor &&
LastTreatmentTimeSitecomNovaAgt && LastTreatmentTimeSitecomNovaCpt && LastTreatmentTimeSitecomNovaMail &&
LastTreatmentTimeSitecomScore,
        "Warning",
        IF(FailTreatmentStatusSitecomAcc > 0 || FailTreatmentStatusSitecomAccu > 0 || FailTreatmentStatusSitecomAccuPu > 0 ||
FailTreatmentStatusSitecomCommon > 0 || FailTreatmentStatusSitecomClaim > 0 || FailTreatmentStatusSitecomCtr > 0 ||
FailTreatmentStatusSitecomNovaMail > 0 || FailTreatmentStatusSitecomExtract > 0 || FailTreatmentStatusSitecomLeLyx > 0 ||
FailTreatmentStatusSitecomHealth > 0 || FailTreatmentStatusSitecomLesFurets > 0 || FailTreatmentStatusSitecomHome > 0 ||
FailTreatmentStatusSitecomInit > 0 || FailTreatmentStatusSitecomNovaCpt > 0 || FailTreatmentStatusSitecomNovaAgt > 0 ||
FailTreatmentStatusSitecomMaster > 0 || FailTreatmentStatusSitecomMotor > 0 || FailTreatmentStatusSitecomScore > 0,
            "Warning",
            "OK"
        )
    )
)
```

Une fois le rapport test stabilisé, j'ai sollicité un feedback, notamment auprès de Sonia, afin de valider la pertinence de la logique et des visuels. Les retours ont permis d'affiner certaines règles d'évaluation et de simplifier l'interface. J'ai ensuite intégré Sitecom dans le rapport officiel, en suivant les processus de mise à jour, documentation, et commit Git comme précédemment.

Pour conclure cette mission, j'ai adapté le visuel global du rapport, en privilégiant une interface plus sobre, plus claire, avec des couleurs standards pour les statuts (vert = OK, orange = WARNING, rouge = KO) et des cartes lisibles. J'ai ainsi finalisé une V1 complète et exploitable du tableau de bord, regroupant Téléphonie, Solaris et Sitecom, et prête à être enrichie avec d'autres flux ultérieurement.



Images – rapport Power BI V1

3.3 Collaboration et bonnes pratiques

Tout d'abord, le travail au sein de l'équipe a été organisé selon une méthodologie agile, structurée autour d'Azure DevOps. Chaque tâche, qu'il s'agisse d'une évolution du rapport, d'un bug à corriger ou d'une demande métier, était formalisée sous forme de ticket. Ce tableau était mis à jour quotidiennement lors des points d'équipe. Cette organisation m'a permis de visualiser clairement les priorités, de suivre l'avancement du projet en temps réel, et d'apprendre à découper les développements en livrables progressifs et testables.

Chaque tâche était également reliée à un sprint, d'une durée de 3 semaines. À la fin de chaque sprint, nous réalisions une revue pour présenter les développements réalisés, faire remonter les difficultés rencontrées, et préparer les prochaines étapes. Ces rituels agiles m'ont donné un cadre structurant pour gérer mon travail, mais aussi un espace pour valoriser mes livrables et bénéficier des retours de l'équipe.

En parallèle, j'ai appris à collaborer de manière rigoureuse et sécurisée via Git. Toute modification sur le fichier Power BI faisait l'objet d'un versioning contrôlé. Je créais une branche Git spécifique à chaque fonctionnalité, poussais mon travail dessus, puis soumettais un pull request pour relecture.

De plus, j'ai également documenté l'ensemble du rapport Power BI que j'ai développé, en expliquant les sources de données utilisées, la signification des mesures DAX, la structure des pages du rapport, et les règles de gestion appliquées.

Enfin, un autre aspect fondamental de cette mission a été la mise en œuvre de bonnes pratiques de test et de validation. Avant toute mise en production, je réalisais des tests manuels : vérification de la cohérence des données affichées, de la fiabilité des indicateurs, et du bon fonctionnement des filtres interactifs.

Cette approche collaborative, outillée et rigoureuse m'a permis de gagner en maturité professionnelle. Elle m'a préparé à travailler efficacement dans des environnements techniques complexes, avec des contraintes de production, de performance et de sécurité des données.

4. Bilan et perspectives

Cette année d'alternance chez Direct Assurance m'a offert une expérience technique et professionnelle riche, structurée autour de missions concrètes, alignées avec les enjeux de l'entreprise et les objectifs de ma formation universitaire. En rejoignant l'équipe Data Engineering, j'ai pu m'impliquer dans un projet à fort impact opérationnel : la mise en place d'un outil de supervision des pipelines de données, indispensable pour garantir la fiabilité des traitements et la disponibilité des informations métier.

4.1 Compétences acquises

Sur le plan technique, cette alternance m'a permis de progresser significativement sur l'ensemble des outils et technologies utilisés par les équipes data.

J'ai d'abord acquis une maîtrise opérationnelle de Power BI, en particulier sur la modélisation de données, l'écriture de mesures DAX complexes, et la conception de tableaux de bord interactifs adaptés à des usages métiers et techniques. J'ai appris à structurer un modèle de données robuste, à créer des indicateurs fiables, et à optimiser les performances d'un rapport connecté à des volumes importants.

En parallèle, j'ai développé mes compétences en gestion de pipelines de données, en utilisant des outils tels que SIMON pour le suivi des traitements et Databricks pour l'exécution distribuée de transformations plus complexes. J'ai compris le fonctionnement d'un flux automatisé, les points de vigilance en matière de supervision, et les méthodes pour identifier et corriger les anomalies dans une logique d'exploitation quotidienne.

Enfin, j'ai été formé à l'utilisation d'Azure DevOps et de Git, en intégrant pleinement mon travail dans un cycle agile avec gestion de tickets, versioning des fichiers.pbix, pull requests, et documentation dans un wiki partagé. Cela m'a permis d'adopter les bonnes pratiques de développement collaboratif, souvent peu abordées dans un cadre académique mais essentielles en entreprise.

Au-delà de l'aspect technique, cette alternance m'a permis de développer des compétences transversales qui me seront utiles tout au long de ma carrière. J'ai appris à être autonome dans la gestion de mes tâches, à organiser mon travail autour de livrables clairs et progressifs, et à solliciter mes collègues aux bons moments lorsque je rencontrais une difficulté.

J'ai également progressé en communication technique, en étant capable d'expliquer mes choix de conception, de documenter mon code ou mes rapports de manière compréhensible, et de présenter mes livrables en réunion d'équipe. Cette capacité à vulgariser des concepts techniques pour un public non-spécialiste est aujourd'hui une compétence clé dans le domaine de la data.

4.2 Résultats

Mon principal livrable est un rapport Power BI complet de supervision, regroupant plusieurs traitements critiques (Téléphonie, Solaris, Sitecom), chacun modélisé avec des statuts (OK / WARNING / KO) générés dynamiquement selon des règles métiers précises.

Ce rapport a permis :

- Une réduction du temps de contrôle quotidien des pipelines,
- Une meilleure réactivité en cas d'incident,
- Une centralisation des informations clés dans une seule interface,

Ce rapport n'est pour le moment pas encore utilisé quotidiennement par les membres de l'équipe, et constitue un socle sur lequel de futures évolutions pourront s'appuyer.

Le projet étant conçu comme un outil évolutif, plusieurs pistes d'amélioration sont envisagées à court et moyen terme :

- Ajout de nouveaux traitements à superviser.
- Automatisation des notifications, avec un système de mails ou d'alertes Teams déclenchés automatiquement selon des seuils ou types d'erreurs.

Ajout d'indicateurs de performance agrégés, comme :

- Taux de succès par semaine ou par mois,
- Évolution de la durée moyenne d'exécution des flux,
- Visualisation des tendances d'incidents par flux.
- Amélioration de l'ergonomie du rapport (affichage, filtres dynamiques simplifiés, accessibilité métier).

Ces pistes permettraient de faire évoluer le tableau de bord actuel d'un outil de surveillance technique vers une plateforme complète de pilotage de la qualité des données, accessible à des profils non techniques.

CONCLUSION

Cette année d'alternance chez Direct Assurance a marqué une étape décisive dans mon parcours de formation en MIAE. Elle m'a permis de passer d'une approche théorique de la donnée à une pratique concrète, ancrée dans les besoins réels d'une entreprise leader dans son secteur.

En intégrant l'équipe Data Engineering, j'ai non seulement appris à maîtriser des outils et des méthodes de travail professionnels (Power BI, Git, Azure DevOps, SHINE, Databricks), mais j'ai surtout compris l'importance de la donnée dans le pilotage quotidien des activités. À travers le développement du rapport de monitoring des pipelines, j'ai pu concevoir un livrable utile, qui sera adopté par l'équipe, et évolutif.

Cette expérience m'a permis de développer des compétences techniques solides, mais aussi des savoir-faire transversaux : communication, autonomie, rigueur, gestion de projet. Elle m'a conforté dans mon intérêt pour les métiers de la donnée, en particulier dans les domaines de la supervision, de l'architecture data et de l'ingénierie des flux.

Je tiens à remercier chaleureusement toute l'équipe Data de Direct Assurance, et plus particulièrement mon tuteur, pour leur accompagnement, leur confiance, et les conditions de travail qui m'ont été offertes tout au long de cette année. Cette alternance a pleinement rempli son objectif : me préparer concrètement à la suite de mon parcours professionnel, et m'inscrire durablement dans le monde de la data.