Tom Maucotel

Data & Insight Analyst | Data Architect

Toulouse · tom.maucotel@gmail.com · https://www.linkedin.com/in/maucoteltom/ · https://github.com/0-M-T

Profil

Trois ans en équipe décisionnelle sur un entrepôt PostgreSQL multi-instances et un parc de plusieurs centaines de rapports. 181 demandes traitées, taux de livraison de 81 %, médiane de 8 jours.

Trajectoire assumée : deux premières années d'exécution (migrations, exports, reporting), puis bascule vers la conception d'outils internes, la gouvernance de données et la mise en production de services. La dernière année combine les deux axes.

Deux principes de travail, vérifiables dans mes livrables : un travail n'est terminé que quand un chiffre le prouve, et mieux vaut aucune valeur qu'une valeur fausse présentée comme fiable.

Réalisations principales

Générateur de DDL pour une couche de restitution

Problème. Publier un objet transformé vers la couche de restitution demandait de créer manuellement deux objets, avec des conventions de nommage à respecter de mémoire, des colonnes et types à recopier, des droits à redéfinir. Dix à quinze minutes par objet, plusieurs fois par semaine.

Action. Constat chiffré d'abord (mesure du temps réel et vérification que toute l'information était déjà interrogeable). Développement d'un outil dérivant l'intégralité du DDL de l'introspection. Trois interfaces (graphique, ligne de commande, serveur d'outils pour agent) sur un noyau unique sans logique dupliquée. Sécurité SQL à trois niveaux. Mode simulation fidèle au run réel. Traçabilité cumulative apposée sur chaque objet. Conteneurisation durcie et pipeline de déploiement.

Résultat. Moins d'une minute par objet, conventions garanties mécaniquement, droits validés avant écriture. 203 tests sans base de données, dont des tests de non-régression sur une faille d'injection et sur l'annulation de transaction.

Catalogue de données et lineage de bout en bout

Problème. Aucune visibilité sur le cycle de vie de la donnée : personne ne pouvait répondre à « d'où vient cette donnée, qui la consomme, que casse-t-on en la modifiant ».

Action. Reprise d'un projet initié par un collègue, puis maintien (156 contributions sur 233, toutes les fusions finales). Développement de connecteurs pour quatre familles de sources non couvertes nativement. 12 correctifs documentés sur un connecteur open source, dont 9 identifiés comme contribuables en amont. Authentification déléguée à l'annuaire d'entreprise, avec procédures de reprise sur incident.

Résultat. Analyse d'impact amont et aval sur toute la chaîne, de la table source au rapport, avec lineage colonne à colonne sur le périmètre relationnel.

Supervision d'une plateforme BI, mise en production

Problème. Aucune alerte native sur échec de rafraîchissement. Les pannes étaient découvertes par un utilisateur métier plusieurs jours après.

Action. Reprise d'un service démarré par un collègue (65 contributions sur 97) et industrialisation complète : coffre-fort de secrets avec secret éphémère non persistant, client HTTP robuste (limiteur de débit, disjoncteur, reprise sur erreurs transitoires uniquement), métriques avec discipline de cardinalité, trois sondes de supervision, conteneurisation durcie avec proxy paramétrable, pipeline branché sur les gabarits d'entreprise, déploiement en validation manuelle.

Décision d'architecture assumée. Exposer des métriques plutôt qu'écrire les statuts dans l'entrepôt : une supervision qui dépend du système surveillé ne signale pas la panne de ce système.

Résultat. Surveillance de 470 modèles, taux de succès rendu mesurable donc pilotable, détection ramenée de plusieurs heures à quelques minutes.

Mesure d'usage d'un patrimoine de données

Problème. Répondre de façon défendable à « lesquels de ces milliers d'objets sont réellement utilisés ». Les compteurs natifs sont pollués par transitivité : un accès élevé sur une table source peut venir entièrement d'un rafraîchissement automatique.

Action. Conception d'un algorithme de propagation depuis les feuilles du graphe de dépendances, seul endroit où une lecture signifie un usage réel. Deux parcours en largeur inversés, avec un troisième état « incertain » pour ne jamais classer inutilisé ce qui n'a pas pu être mesuré. Abstraction encapsulant l'hétérogénéité des sources de statistiques.

Résultat. Sur 9 000 objets et 12 700 relations : un quart confirmé utilisé, deux tiers candidats au nettoyage, couverture statistique de 85 % du patrimoine. Un repli de mesure a été abandonné après mesure réelle (couverture nulle constatée) plutôt que maintenu avec une fiabilité inconnue.

Environnement de développement assisté par IA

Problème. Un assistant utilisé au coup par coup produit un travail inégal, ignore les conventions et peut toucher des systèmes sensibles.

Action. 33 procédures méthodologiques encodées (conditions d'activation, étapes ordonnées, seuils chiffrés, conditions d'arrêt) plutôt que de simples consignes. 12 crochets automatisés de sécurité et de qualité, dont un crochet de protection des secrets développé comme composant de sécurité à part entière, avec sa suite de tests. 3 serveurs d'outils exposant la logique métier à un agent, avec mode simulation par défaut et revalidation côté serveur. Agent autonome borné par deux barrières indépendantes (liste blanche déclarative et crochet impératif fermé par défaut), avec vérification adverse par un juge sur modèle distinct contrôlant factuellement les affirmations produites.

Résultat. Un environnement où les règles coûteuses sont garanties par des crochets plutôt qu'espérées, et où les tâches de fond avancent sans supervision avec un contrôle anti-affabulation. Configuration packagée et partagée à l'équipe.

Avant l'alternance : automatisation d'un calcul de rejets

atmosphériques Contexte. Stage d'un mois dans le secteur du transport de gaz.

Action. Automatisation du calcul des volumes de gaz relâchés lors des opérations de mise à l'atmosphère, dans le but de quantifier les rejets. Tableur et langage de script associé.

Ce que ça montre. Le réflexe d'automatiser un calcul métier répété est antérieur à l'alternance. Le sujet est resté le même sur trois ans, seule l'échelle a changé : remplacer une opération manuelle par un calcul reproductible et vérifiable.

Compétences, avec niveaux

Données

Compétence	Niveau
PostgreSQL : architecture en couches, fédération inter-bases	maîtrise
PostgreSQL : jointures complexes, agrégations, introspection du catalogue	maîtrise
PL/pgSQL : procédures avec gestion d'erreur et idempotence	autonome à maîtrise
Fonctions fenêtrées (déduplication, dernière valeur connue)	autonome à maîtrise
Sécurité SQL : validation d'identifiants, échappement, paramètres liés	maîtrise
Diagnostic de données et non-régression chiffrée	maîtrise
Oracle : lecture et migration vers PostgreSQL	autonome

Développement

Compétence	Niveau
Python : structuration d'outil, noyau pur et interfaces multiples	maîtrise
Tests sans infrastructure (doublures, fonctions pures)	maîtrise
Robustesse d'appel réseau (reprise, limiteur, disjoncteur)	maîtrise
Typage statique, journalisation structurée, gestion d'erreur spécifique	maîtrise
Programmation asynchrone	autonome à maîtrise
Algorithmique de graphe (propagation, analyse d'impact)	autonome à maîtrise
Analyse de code SQL déclaratif et procédural	autonome à maîtrise

Plateformes et exploitation

Compétence	Niveau

ETL : modification de traitements, diagnostic dans le code généré	maîtrise
Plateforme BI : mise en production, administration par API, sécurité au niveau ligne	maîtrise
Catalogue de données : déploiement, connecteurs, lineage	maîtrise
Conteneurisation durcie	autonome à maîtrise
Observabilité : conception de métriques, cardinalité, alerting	autonome à maîtrise
Intégration et déploiement continus, automatisation de déploiement	autonome
Coffre-fort de secrets, découverte de services	autonome
Exploitation et astreinte	maîtrise

IA appliquée

Compétence	Niveau
Conception de procédures et de crochets pour assistant	maîtrise
Serveurs d'outils sûrs par défaut pour agent	autonome à maîtrise
Ingénierie d'agents autonomes avec garde-fous et vérification adverse	autonome à maîtrise

Non revendiqué, faute de preuve dans mon expérience : déclencheurs et requêtes récursives SQL, partitionnement, mesures avancées de langage de calcul BI, conception Oracle native, orchestration de conteneurs à l'échelle.

Environnement technique traversé

Bases PostgreSQL multi-instances avec fédération, Oracle sur périmètre historique Intégration atelier ETL graphique, ordonnanceur d'entreprise, transferts de fichiers managés Restitution plateforme BI cloud, passerelle de données, rapports paginés Gouvernance catalogue de données open source, ingestion automatisée multi-sources Infrastructure conteneurs sur machines virtuelles, registre privé, coffre-fort de secrets, annuaire de services, proxy sortant Observabilité collecteur de métriques, visualisation, gestionnaire d'alertes, console d'astreinte Développement forge logicielle avec gabarits partagés, automatisation par playbooks

Formation

- Master Informatique, parcours Data Architecte · Université de La Rochelle (en alternance) · 2024-2026
- BUT Science des Données · IUT de Pau (en alternance) · 2020-2024
- Baccalauréat Scientifique (spécialité SVT, option ISN) · 2020

Langues

Français (langue maternelle) · Anglais (B2 technique)