Référentiel de compétences, adossé aux preuves

Chaque compétence de cette fiche est mesurée, pas déclarée : nombre de tickets, de commits, de fichiers, de tests. C'est ce qui la rend défendable en entretien technique.

Complète `22-portfolio-cv-realisations.md`, qui décrit les réalisations. Ici, ce sont les compétences et leur niveau.

Sources et méthode

Source	Périmètre analysé
Dossiers de demandes	138 dossiers, 224 fichiers SQL, 130 fichiers de suivi
Suivi de demandes (outil de ticketing)	181 demandes attribuées
Historique de contrôle de version	411 contributions sur 8 dépôts
Code des projets	219 fichiers Python, 44 100 lignes applicatives
Outillage d'assistant	33 procédures, 12 crochets, 3 serveurs d'outils
Wiki d'entreprise	16 pages créées, 18 en contribution, sur 17 mois

Trois précautions d'honnêteté, tenues dans toute la fiche :

1. Les compétences observées sur un seul chantier sont signalées comme telles (prouvées en profondeur, pas en largeur).
2. Ce qui n'apparaît pas dans le corpus est dit explicitement, plutôt que supposé.
3. La distinction entre ce que j'ai créé et ce que j'ai repris et fait évoluer est maintenue partout.

Volumétrie d'ensemble

Mesure	Valeur
Demandes traitées	181
Taux de livraison effective	81 % (129 en production, 18 closes)
Durée médiane de traitement	8 jours (moyenne 44, deux régimes coexistants)
Contributions au code	411
Code applicatif produit	44 100 lignes
Code de test produit	12 550 lignes (28 % du total)
Fonctions et méthodes	1 925
Fonctions de test	982
Documentation technique dans le code	167 fichiers, 18 diagrammes
Pages de wiki d'entreprise	16 créées, 18 en contribution
Pipelines d'intégration continue	4
Chaînes d'ordonnancement	19

Compétences techniques

SQL et PostgreSQL

Compétence	Preuve	Niveau
Architecture de vues en couches (source, transformation, exposition fédérée)	45 demandes, 40 fichiers avec table distante fédérée	maîtrise

Jointures complexes et agrégations	72 fichiers avec jointures, 38 avec regroupements	maîtrise
Procédures et fonctions procédurales avec gestion d'erreur	28 fichiers, 17 avec levée d'exception explicite	autonome à maîtrise
Fonctions fenêtrées (numérotation, première valeur, décalage)	11 fichiers, usages : déduplication, dernière valeur connue, détection de trous	autonome à maîtrise
Requêtes imbriquées nommées	9 fichiers, dont contrôles de cohérence	autonome
Vues matérialisées et leur rafraîchissement	13 fichiers, dont identification d'un risque de verrou exclusif	autonome
Introspection du catalogue système	23 fichiers Python interrogeant les tables système	maîtrise
Fédération inter-bases et ses limites	conception d'architecture, diagnostic d'un désalignement de type provoquant une erreur de lecture distante	maîtrise
Sécurité SQL (validation d'identifiants, échappement, paramètres liés)	20 tests dédiés à une faille d'injection identifiée et corrigée	maîtrise
Indexation	17 fichiers de création d'index, sans index partiel ni fonctionnel observé	autonome

Ce qui n'apparaît pas dans le corpus : déclencheurs (triggers), requêtes récursives, partitionnement. À ne pas revendiquer.

Oracle

Compétence	Preuve	Niveau
Migration vers PostgreSQL	4 demandes structurantes	autonome
Lecture de requêtes existantes (syntaxe, variables de liaison)	analyse de rapports à reprendre	notions à autonome
Diagnostic de vestiges de migration	identification d'un traitement dupliqué non nettoyé après bascule	autonome

Formulation honnête : lecture et migration d'un existant Oracle, pas conception Oracle native.

ETL et ordonnancement

Compétence	Preuve	Niveau
Modification de traitements d'intégration	25 demandes : changement de source, de connexion, correction de composant, mise aux normes	maîtrise
Diagnostic par lecture du code généré	remontée d'une erreur d'exécution jusqu'à la ligne générée, cause racine identifiée (point d'entrée d'API supprimé par l'éditeur)	maîtrise
Analyse de robustesse de chaîne	diagnostic d'un traitement à 14 étapes en cascade stricte, sans tolérance de panne	autonome à maîtrise
Audit de parc par script	recherche exhaustive sur plusieurs centaines de traitements, distinction usage direct / par configuration	autonome
Diagnostic de dépendances temporelles	cas de référence : déclenchement avant fin du chargement de la source, corrigé par replanification plus garde-fou applicatif	maîtrise
Ordonnancement d'entreprise	usage des interfaces existantes, peu de configuration native	notions à autonome

Restitution BI

Compétence	Preuve	Niveau
Mise en production de rapports	14 demandes dédiées	maîtrise
Administration par API	7 scripts d'audit, pagination, dégradation à 3 niveaux selon les droits	autonome à maîtrise (un seul chantier)
Sécurité au niveau ligne	mise en œuvre avec formule de dérivation d'identité	autonome
Diagnostic du langage de requête	traçage de l'origine réelle d'une donnée par lecture du code de transformation	maîtrise
Optimisation par rafraîchissement incrémental	méthode outillée (accès au point d'entrée d'administration, inspection de partitions), critères d'acceptation explicites	autonome
Langage de calcul (expressions de mesure)	usage ponctuel démontré, pas de production massive	autonome

À ne pas surestimer : pas de mesures avancées ni d'intelligence temporelle dans le corpus.

Python et ingénierie logicielle

Mesuré directement dans le code des projets :

Compétence	Preuve	Niveau
Structuration d'un outil (noyau pur, points d'entrée multiples)	3 interfaces sur un noyau unique, aucune logique dupliquée	maîtrise
Tests sans infrastructure	982 fonctions de test, 472 doublures, 25 % du code	maîtrise
Typage statique	140 fichiers annotés, vérification stricte sur le noyau métier	maîtrise
Robustesse d'appel réseau	112 mécanismes de reprise, 42 limiteurs de débit, 24 disjoncteurs	maîtrise
Programmation asynchrone	101 fonctions asynchrones	autonome à maîtrise
Journalisation structurée	796 appels, format exploitable par un agrégateur	maîtrise
Gestion d'erreur spécifique	384 captures typées, aucune capture nue	maîtrise
Masquage de secrets	33 mécanismes, testés	maîtrise
Écriture atomique	14 usages (fichier temporaire puis renommage)	autonome
Analyse de code (SQL déclaratif et procédural)	approche mixte assumée et documentée, avec limites explicites	autonome à maîtrise
Algorithmique de graphe	propagation depuis les feuilles, analyse d'impact, sur 9 000 nœuds	autonome à maîtrise
Interfaces de données	état multi-étapes, tâches de fond avec verrou	autonome

Conteneurisation, déploiement, exploitation

Compétence	Preuve	Niveau
Conteneurisation durcie	5 descriptions d'image : construction multi-étapes, image épinglée par empreinte, utilisateur non privilégié, capacités système retirées, système de fichiers en lecture seule	autonome à maîtrise

Intégration et déploiement continus	4 pipelines, branchement sur gabarits partagés, validation manuelle en production	autonome
Automatisation de déploiement	3 playbooks, 10 gabarits de configuration, attente active du contrôle de santé puis test de fumée	autonome
Gestion de secrets par coffre-fort	authentification de rôle, secret éphémère non persistant, rotation sans redéploiement	autonome
Observabilité	conception de métriques avec discipline de cardinalité, 3 sondes distinctes, règles d'alerte	autonome à maîtrise
Découverte de services	enregistrement avec contrôle de santé	notions à autonome
Exploitation et astreinte	rotation hebdomadaire, triage, diagnostic jusqu'à la cause racine	maîtrise

Nuance à assumer : l'historique de contrôle de version montre un apprentissage du déploiement continu par tâtonnement (série de contributions exploratoires) avant stabilisation. C'est une progression réelle, pas une expertise initiale.

Gouvernance de données

Compétence	Preuve	Niveau
Déploiement d'un catalogue	plateforme complète : serveur, métadonnées, recherche, ingestion ordonnancée	maîtrise
Développement de connecteurs	4 familles de sources non couvertes nativement	maîtrise
Lineage dont colonne à colonne	extraction par analyse statique, 5 types de relations	maîtrise
Conduite d'audit de décommissionnement	croisement de 6 sources indépendantes, correction publique de 2 hypothèses invalidées	maîtrise
Mesure d'usage réel d'un patrimoine	algorithme de propagation, 9 000 objets, 3 états dont un « incertain » assumé	maîtrise
Conformité (données personnelles)	traitement récurrent, signalement proactif d'un écart de politique de comptes	autonome
Patching d'un projet amont	12 correctifs documentés, 9 identifiés comme contribuables	autonome à maîtrise

Outillage et ingénierie d'agents

Trois niveaux, du plus répandu au moins répandu.

Niveau 1 : usage avancé d'un assistant (reproductible par un profil sérieux)

Compétence	Preuve
Procédures encodant une méthode	33 procédures (17 métier, 9 génériques), 11 800 lignes
Crochets de cycle de vie	12 crochets : sécurité, qualité, préservation de contexte, notification
Critères de vérification chiffrés	seuils de couverture, de confiance, conditions d'arrêt explicites

Ce qui distingue une procédure d'un simple prompt : conditions d'activation, ordre imposé, seuils numériques, et conditions d'arrêt. Exemple mesurable : après 3 tentatives infructueuses, retour aux hypothèses ; après 5, remise en cause de l'architecture.

Niveau 2 : intégration d'outils à un agent

Compétence	Preuve
Serveurs d'outils exposant une logique métier	2 : générateur de DDL (6 outils), proxy de catalogue

Sécurisation par défaut	mode simulation actif par défaut, revalidation côté serveur, réponse structurée
Discipline d'élagage des réponses	réduction mesurée de 86 % du volume retourné sur un cas réel
Outils en ligne de commande pilotables	7 outils, 2 540 lignes, aucune dépendance externe

L'élagage n'est pas une optimisation cosmétique : une réponse non restreinte mesurée à 70 000 caractères pour 13 éléments sature la fenêtre de contexte en trois appels.

Niveau 3 : ingénierie de systèmes agentiques (rare)

Compétence	Preuve
Vérification adverse par juge indépendant	modèle distinct de l'agent de travail, procédure contradictoire bornée, sur-ingénierie traitée comme motif d'échec
Deux barrières de sécurité indépendantes	liste blanche déclarative plus crochet impératif, fermeture par défaut, 43 cas de test dédiés
Robustesse opérationnelle sans supervision	verrou avec détection de péremption, rapport de plantage garanti, budget recalculé sur signal de quota, 3 états d'échec distincts
Durcissement itératif documenté	5 failles de sécurité fermées après audit, avec historique daté

Détail du mécanisme de juge : le modèle du juge est différent de celui de l'agent de travail, pour casser le biais de confirmation ; l'agent peut contester un reproche avec un argument vérifiable, le juge restant arbitre ; et un plafond atteint sans convergence produit un état distinct d'un échec technique.

Distinction à tenir : le niveau 1 relève d'une pratique de configuration, accessible à qui prend le temps de documenter. Les niveaux 2 et 3 relèvent d'une conception de système, concevoir un dispositif de défiance méthodique envers ses propres agents plutôt que de leur faire confiance.

Documentation et transmission de savoir

Mesuré sur le wiki d'entreprise : 16 pages créées, 18 en contribution, sur 17 mois, avec une concentration sur les derniers mois.

Nature	Nombre
Documentation d'outil ou de composant	6
Procédure opératoire (installation, mise en production)	5
Note de cadrage ou d'analyse d'incident	1
Documentation destinée aux utilisateurs métier	1
Support de formation et de vulgarisation	1
Suivi de projet	1
Reprise d'une page existante d'un tiers	1

Ce qui est réellement remarquable ici

Le même projet documenté à trois registres différents, pour trois publics :

Public	Page	Registre
Utilisateur métier non technique	page d'accueil de l'outil	trois questions organisatrices, captures commentées, questions fréquentes, canal de support balisé, aucun jargon
Pair technique	documentation de composant	tableaux de métriques typées, points d'entrée, configuration, exemple de règle d'alerte
Futur mainteneur	architecture et décisions	schémas, décisions justifiées par les alternatives écartées, forme de registre de décisions d'architecture

C'est la preuve la plus directe d'une capacité d'adaptation au lecteur, et elle est vérifiable.

Autres preuves de transmission

Preuve	Ce qu'elle démontre
Guides d'installation autosuffisants, avec tableau de dépannage anticipant les erreurs courantes	documentation pensée pour quelqu'un qui n'a pas le contexte
Documentation d'un prototype abandonné, avec la correspondance ancien vers nouveau composant	capitalisation entre deux générations d'outillage
Documentation explicite des limites d'un outil (analyse dynamique non couverte, déclencheurs non traités)	honnêteté sur le périmètre de non-couverture
Procédure de reprise sur incident, avec avertissement sur une information définitivement perdue si relance prématurée	connaissance opérationnelle critique transmise
Conventions de nommage formalisées en tableau exhaustif	formalisation d'un usage jusque-là oral
Support de formation déconnecté du contexte interne	pensé comme réutilisable
Page d'accueil avec questions fréquentes anticipant les questions récurrentes	réduction de la charge de support

Pratique répétée observée : documenter à la fois *comment utiliser* (procédure), *comment ça fonctionne* (architecture) et *pourquoi c'est construit ainsi* (décisions).

Compétences humaines, avec leurs preuves

Chaque compétence est adossée à un type de trace observable, pas à une affirmation.

Rigueur et méthode

Preuve observée	Ce qu'elle démontre
Non-régression chiffrée systématique (comparaison bidirectionnelle, résultat attendu et reporté)	le chiffre prouve, l'impression ne prouve pas
Vérification par deux méthodes indépendantes avant conclusion	défiance envers une source unique
Correction explicite d'hypothèses invalidées, documentée comme telle	honnêteté intellectuelle plutôt que réécriture silencieuse
Limites méthodologiques reconnues dans les livrables	un résultat partiel annoncé vaut mieux qu'un résultat faux présenté comme complet
83 % des contributions au code avec un corps de message substantiel, chiffres de validation inclus	traçabilité au-dessus de la moyenne

Collaboration et coordination

Preuve observée	Ce qu'elle démontre
Reprise de deux projets conçus par un collègue, devenu mainteneur de fait (toutes les fusions finales)	capacité à faire évoluer du code d'autrui sans le casser
Répartition écrite des responsabilités avec une équipe tierce, deux scénarios posés, décision tracée	coordination explicite plutôt qu'implicite
Escalade argumentée après épuisement du périmètre propre, avec questions précises	escalade utile, pas remontée de blocage
Identification proactive du bon interlocuteur par indices convergents	efficacité relationnelle

Préparation active d'une revue de sécurité avant mise en production	anticipation des contraintes d'autrui
Section « contributeurs » d'une page projet attribuant nommément les apports de chacun	reconnaissance du travail d'autrui plutôt qu'appropriation
Renvois systématiques vers d'autres pages plutôt que duplication de contenu	souci de cohérence documentaire collective

Relation avec les demandeurs

Preuve observée	Ce qu'elle démontre
Transformation d'une demande informelle en spécification exploitable	traduction besoin vers solution
Sollicitation de clarification avant de figer une implémentation	refus de deviner
Validation croisée avant mise en production	le métier est juge de la justesse fonctionnelle
Arbitrages documentés avec leurs options écartées	décision défendable six mois plus tard
Taux d'annulation de 13 % concentré sur les demandes ponctuelles, pas sur les développements structurants	l'arbitrage amont fonctionne, les chantiers de fond aboutissent

Autonomie et initiative

Preuve observée	Ce qu'elle démontre
Signalement de dette non demandé, consigné sans être traité d'office	discipline de périmètre
Correction d'un outil de diagnostic jugé insuffisant en cours de mission	remise en cause de ses propres moyens
Conception d'outils internes issus d'un constat chiffré, pas d'une envie technique	industrialisation justifiée
Refus documenté de sur-ingénierie	jugement, pas application mécanique

Communication écrite

Preuve observée	Ce qu'elle démontre
Contraste assumé entre notes internes denses et messages externes courts	adaptation au lecteur
Messages différenciés par destinataire, préparés avant envoi	soin de la réception
Emploi de la référence connue du demandeur plutôt que de la clé interne	adaptation au canal
167 documents techniques, 18 diagrammes, 16 pages de wiki	documentation comme réflexe, pas comme corvée
Même sujet écrit à trois registres selon le public	adaptation démontrée, pas déclarée

Gestion de plusieurs sujets

Preuve observée	Ce qu'elle démontre
Journal daté tenu au fil de l'eau, reprise possible après plusieurs semaines	gestion de l'interruption
Mise en réserve assumée et datée d'un volet, plutôt qu'avancement désordonné	priorisation explicite
Quatre projets menés simultanément sur un même mois (159 contributions)	capacité de charge réelle
Médiane de traitement à 8 jours, moyenne à 44	deux régimes coexistants : flux réactif et chantiers de fond
53 demandes (29 %) tracées depuis un portail tiers	gestion de la demande métier avec traçabilité formelle

Trajectoire mesurée sur trois ans

Deux sources indépendantes donnent la même bascule, ce qui la rend solide.

Vue par le suivi de demandes

Période	Demandes	Nature dominante
Année 1 (dernier trimestre)	7	démarrage
Année 2	40	exécution : migrations, exports, reporting. 90 % de récits
Année 3	69	densification : apparition du type tâche (21), travail plus granulaire et réactif
Année 4 (partielle)	65	bascule : 35 décisionnel plus 30 gouvernance et infrastructure

Lecture du découpage : ce sont des années calendaires, pas des années d'expérience. Trois ans d'alternance démarrés en cours d'année et terminés en cours d'année couvrent quatre années calendaires, dont la première et la dernière sont partielles. D'où sept demandes sur la première et un total qui reste bien celui de trois ans.

La quatrième année voit apparaître un type de travail absent des trois premières : épopée structurée en 12 sous-tâches, anomalies sur des connecteurs, montées de version avec correctifs de sécurité, arbitrages entre correctif officiel et développement spécifique.

Le découpage en sous-tâches apparaît uniquement la dernière année (13 sur 13) : signe d'un passage à des travaux de conception longs nécessitant une structuration.

Vue par l'historique de contrôle de version

	Début	Fin
Nature	script applicatif solo, reprise de déploiement par tâtonnement	intégrateur mainteneur, incidents de production, outillage transverse
Sécurité	absente	audit d'injection, typage strict, durcissement de conteneur, préparation de revue
Documentation	fichier de démarrage	procédures de reprise, plans de consolidation, schémas d'architecture
Outillage	aucun	agent de fond, outils d'équipe, procédures partagées
Rapport à l'existant	réécrit un script	reprend du code tiers sans le casser, devient le point de fusion final

Convergence des deux sources : le mois de pic de contributions (159, quatre projets simultanés) correspond à l'année d'apparition du second projet dans le suivi de demandes. La bascule est datée et corroborée.

Formulations prêtes pour un CV

Titre

Selon le poste visé :

Cible	Formulation
Poste data analyste ou BI	*Analyste data / BI, orienté industrialisation et gouvernance*
Poste ingénieur data	*Ingénieur data, PostgreSQL et gouvernance, avec pratique du déploiement*
Poste hybride	*Analyste data devenu ingénieur, du traitement de demande à la mise en production*

Éviter « architecte » : la période ne couvre pas de schéma directeur.

Puces d'expérience, avec chiffres

- Traitement de 181 demandes sur trois ans (taux de livraison 81 %, médiane 8 jours) sur un entrepôt de données et un parc de plusieurs centaines de rapports : création et modification d'objets SQL, exports, mises en production, diagnostics d'incident.

- Conception d'un générateur de DDL dérivant l'intégralité du code depuis l'introspection de la base, exposé en trois interfaces sur un noyau unique, 203 tests : temps de publication d'un objet divisé par dix.
- Reprise et industrialisation d'un service de supervision d'une plateforme BI (65 contributions sur 97) : coffre-fort de secrets, conteneurisation durcie, pipeline de déploiement, découverte de services. Détection des pannes ramenée de plusieurs heures à quelques minutes sur 470 modèles.
- Co-maintien puis maintien d'un catalogue de données d'entreprise (156 contributions sur 233) : développement de connecteurs pour quatre familles de sources non couvertes, lineage colonne à colonne, 12 correctifs sur un connecteur amont dont 9 contribuables au projet open source.
- Conception d'un algorithme de mesure d'usage d'un patrimoine de 9 000 objets par propagation depuis les feuilles du graphe de dépendances, avec état « incertain » explicite pour ne jamais classer inutilisé ce qui n'a pas pu être mesuré.
- Conception d'un environnement de développement assisté par IA : 33 procédures méthodologiques, 12 crochets automatisés, 3 serveurs d'outils, et un agent autonome borné par deux barrières de sécurité indépendantes avec vérification adverse par un juge sur modèle distinct.
- Exploitation : rotation d'astreinte hebdomadaire, diagnostic d'incident jusqu'à la cause racine, comptes rendus systématiques.
- Documentation et transmission : 16 pages de wiki d'entreprise, dont un même projet documenté à trois registres distincts (utilisateur métier, pair technique, futur mainteneur), procédures d'installation autosuffisantes et procédures de reprise sur incident.

Bloc compétences

Données : PostgreSQL (architecture en couches, fédération, procédures, fonctions fenêtrées, introspection, sécurité SQL), Oracle (lecture et migration), modélisation en couches, non-régression chiffrée.

Développement : Python (outils multi-interfaces, tests sans infrastructure, typage strict, asynchrone, robustesse d'appel réseau), algorithmique de graphe, analyse de code SQL.

Plateformes : conteneurisation durcie, intégration et déploiement continus, automatisation de déploiement, coffre-fort de secrets, observabilité, ETL et ordonnancement, plateforme BI, catalogue de données.

IA appliquée : conception de procédures et de crochets, serveurs d'outils pour agent, ingénierie d'agents autonomes avec garde-fous et vérification adverse.

Méthode : traitement de demande de bout en bout, industrialisation depuis un constat chiffré, documentation as code, coordination inter-équipes.