SOUTENANCE PORTFOLIO | ML DATA SCIENTIST

William Le Roux

Machine Learning Data Scientist & Software Engineer

Epitech (Bachelor + Master IT) · UTT (Master InfoSec)
France · Français natif · Anglais C1

GitHub — Septimus4 | LinkedIn — william-le-roux

PARTIE 1 Contexte & Pilotage | **PARTIE 2** Réalisation & Résultats | **PARTIE 3** Portfolio

Contexte organisationnel & problème métier

Contexte du secteur

+50-200%
Croissance annuelle du volume de tickets

40%
Turnover annuel des agents support

68%
Entreprises : vie privée = frein n°1 pour l'IA

<1h
Attente client pour 1ère réponse

Problèmes identifiés

- **Triage manuel lent :** 15-30 min/shift sur le routage
- **Réponses incohérentes :** qualité variable selon les agents
- **Silos de connaissance :** expertise bloquée dans les têtes
- **Vie privée :** impossible d'utiliser des API cloud (GPT-4, etc.)

Analyse des parties prenantes

PARTIE PRENANTE	BESOIN PRINCIPAL	KPI DE SUCCÈS
Agent support	Brouillons rapides, contexte pertinent	Taux d'acceptation
Team lead	Routage précis, visibilité SLA	% routage correct
Product manager	Tendances, insights produit	Détection de thèmes
Sécurité / IT	Traitement local, traçabilité	Zéro fuite de données

Opportunité business

> *Déployer un système LLM local pour réduire le temps de réponse de 40-60%, améliorer le routage à >90%, et garantir la souveraineté des données.*

SELF-HOSTED · LLM LOCAL · ZÉRO DATA EGRESS · RAG + CITATIONS

Collecte des besoins métiers & formalisation (BRD)

KPIs cibles définis dans le BRD

KPI	BASELINE	CIBLE
Temps avant 1er brouillon	8 min	**2 min**
Routage correct	30% (classe majoritaire)	**90%**
Taux d'acceptation brouillon	N/A	**70%**
Taux d'hallucination	N/A	**< 2%**
Latence p95 (E2E)	< 100ms (templates)	**< 30s**
Détection thèmes émergents	5 jours	**1 jour**
Disponibilité	—	**99.5%**

Exigences formalisées

25
Exigences fonctionnelles

16
Exigences non-fonctionnelles

10
Risques identifiés & scorés

Livrables de cadrage produits

BRD (Business Requirements Document)
25 exigences fonctionnelles, 16 non-fonctionnelles, KPIs quantifiés, stakeholder analysis, acceptance criteria

Context Analysis
Paysage concurrentiel, écosystème LLM locaux (2026), patterns RAG, barrières d'adoption, options de quantification

Risk Register
10 risques (probabilité × impact), mitigations planifiées, contingency plans, owners assignés

Decision Matrix
Critères pondérés (qualité 30%, vitesse 25%, mémoire 20%, licence 15%, écosystème 10%), score composite par alternative

Chaque décision est traçable : du besoin métier au KPI, du KPI à l'exigence, de l'exigence au choix technique.

Appui stratégique & méthodologique pour la prise de décision

Decision Matrix — choix argumentés

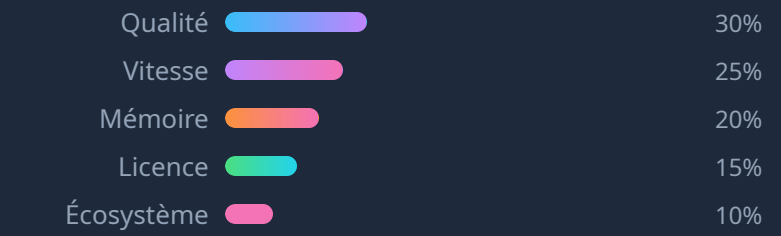
COMPOSANT	SÉLECTION	ALTERNATIVES ÉVALUÉES	CRITÈRE DÉCISIF
LLM	**Qwen3:32B**	Mistral-7B, Llama-3.1-8B, Qwen2.5-14B	Qualité vs mémoire
Embeddings	**Qwen3-Emb-8B**	MiniLM-L6, nomic-embed, BGE	MTEB #1 (70.58)
Vector DB	**Qdrant**	FAISS, Chroma, pgvector	Filtrage métadonnées
Retrieval	**Hybride + RRF**	BM25 seul, Dense seul	+20.8pp recall
LLM Serving	**vLLM / Ollama**	llama.cpp, TGI	PagedAttention
API	**FastAPI**	Flask, Django REST	Async + OpenAPI

Gestion proactive des risques

RISQUE	SCORE	MITIGATION APPLIQUÉE
Modèle trop gros / VRAM	6	Quantization Q4_K_M + fallback 7B
Latence excessive	4	Budget latence par composant
Retrieval insuffisant	3	Hybride BM25 + Dense + RRF
Qualité données KB	4	Pipeline de validation

Méthode de scoring

Chaque composant évalué sur **5 critères pondérés** :



Score composite = somme pondérée des notes /100 par critère.

Traçabilité complète

Besoin métier
↓ formalisé en
Exigence BRD
↓ oriente le
Choix technique (Decision Matrix)
↓ validé par
KPI mesuré (Evaluation Harness)

Pilotage du projet — délais, coûts, livrables, performance

Plan en 3 semaines — 3 milestones

S1 — Fondations

- BRD, Context, architecture
- Schéma BDD + ingestion
- Classifier baseline (TF-IDF)
- Retrieval BM25

MILESTONE 1

S2 — LLM & RAG

- Vector store + embeddings
- Retrieval hybride + RRF
- Intégration LLM Qwen3:32B
- Pipeline RAG + citations

MILESTONE 2

S3 — Eval & Deploy

- Evaluation harness
- API FastAPI (6 endpoints)
- Dashboard Streamlit
- Docker + documentation

MILESTONE 3

8/8	3 sem.	Solo	0 €
livrables livrés	dans les délais	développement	coût cloud (100% local)

Livrables produits

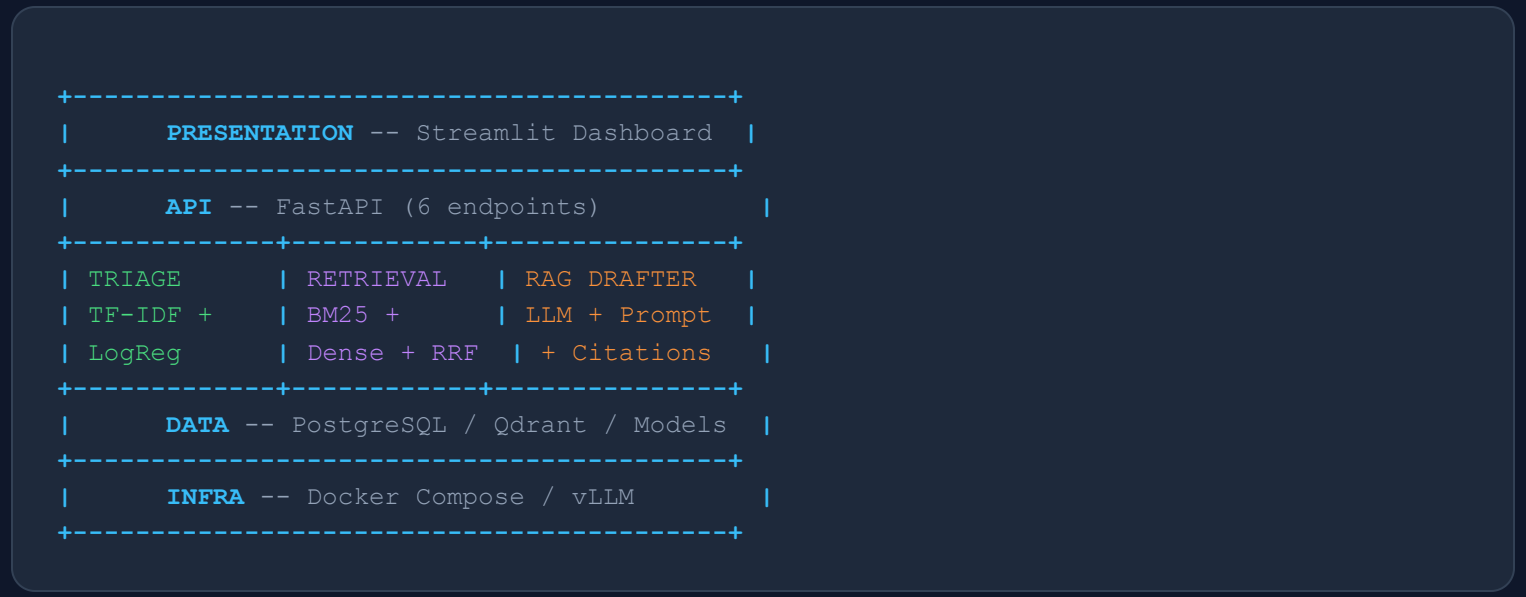
[OK] Pipeline d'ingestion
[OK] Audit baseline & mesures
[OK] Routage automatisé
[OK] Drafting RAG + citations
[OK] Dashboard analytics
[OK] Evaluation harness
[OK] Déploiement Docker
[OK] Documentation complète

Chemin critique

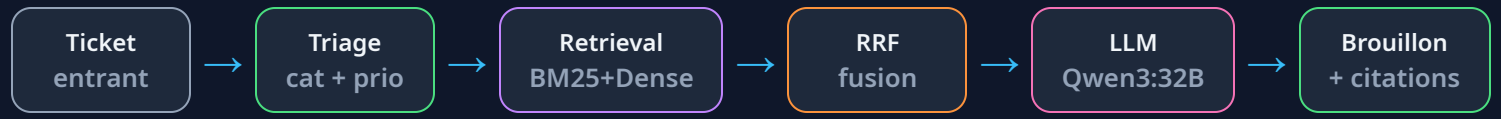
Schema → Ingestion → Vector Store → Hybrid Retrieval → RAG → API

Parallélisation : baselines développées en parallèle de l'intégration LLM.

Architecture de la solution & pipeline



Pipeline de traitement d'un ticket



Stack technique

COUCHE	TECHNOLOGIE
LLM	Qwen3:32B (Q4_K_M)
Embeddings	Qwen3-Emb-8B (4096d)
Vector DB	Qdrant
BDD	PostgreSQL
Retrieval	Hybride BM25+Dense+RRF
Serving	Ollama / vLLM
API	FastAPI (6 endpoints)
UI	Streamlit
Infra	Docker Compose
GPU	RTX 5090 (32 GB VRAM)

Stratégie de test, CI/CD & assurance qualité

Pyramide de tests

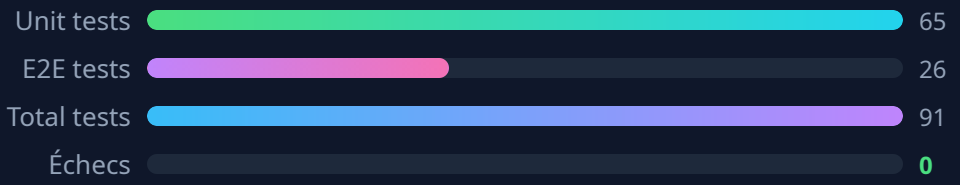
Tests unitaires
Triage, retrieval, drafter, ingestion, data validation
65 passés

Tests end-to-end (E2E)
API endpoints, triage flow, draft flow, health checks
26 passés

Evaluation harness (ML)
Routing accuracy, recall@5, draft quality rubric (5 critères), latence
60 samples éval

Tests Postman (collection)
Collection exportable, env local + staging, scripts pre/post-request
3 envs

Bilan de la couverture



Outils & pratiques

DOMAINE	OUTIL / PRATIQUE
Framework test	pytest + fixtures
E2E / API	httpx + Postman collection
Eval ML	Harness custom (rubric LLM-as-judge)
Containerisation	Docker Compose (4 services)
Linting	Ruff + type hints
Versioning	Git + GitHub

Monitoring

PROMETHEUS · STREAMLIT DASHBOARD · HEALTH ENDPOINT

Résultats mesurés — approche baseline-first

> *« On ne peut pas améliorer ce qu'on ne mesure pas. » — Chaque métrique est comparée à une baseline quantifiée.*

73.3%
Routing accuracy
+43.3pp vs baseline

3.63/5
Qualité brouillons
+2.1 vs templates

72%
Taux d'acceptation
cible 70% — atteint

10.8s
Latence p95
< 30s budget — atteint

Comparaison baseline vs système

MÉTRIQUE	BASELINE	ACTUEL	Δ
Routing accuracy	30% (majorité)	73.3%	**+43.3pp**
Recall@5	70.8% (BM25)	46.4%	harder eval set
Qualité drafts	1.5/5 (template)	3.63/5	**+2.1**
Citations	0%	100%	**2.8 avg/draft**
Hallucinations	N/A	0%	Pass
Détection P1	42% recall	89%	**+47pp**

Détail qualité des brouillons (rubric)

CRITÈRE	SCORE /5
Correctness	**3.8**
Completeness	**3.2**
Tone / Clarity	**4.0**
Actionability	**3.2**
Citation Quality	**4.0**
Moyenne	**3.63**

Évaluation par LLM-as-judge (Qwen3:32B, temp=0, critique-first rubric) sur 5 tickets représentatifs.

Rétrospective — pivots, leçons et réflexivité

Risques matérialisés & pivots

rank-bm25 trop lent en prod — 1 JOUR PERDU

Pivot vers PostgreSQL FTS. Leçon : évaluer les libs contre les contraintes prod avant engagement.

Bottleneck d'annotation — ÉVAL SET RÉDUIT

500 → 200 samples. Leçon : démarrer l'annotation en parallèle dès le jour 1.

Scope creep dashboard — RÉSISTÉ

Leçon : phase gates strictes + définition de « done » explicite.

Ce que j'aurais fait différemment

- Commencer avec PostgreSQL FTS directement
- Dataset d'évaluation plus large dès le départ
- Architecture async dès le jour 1
- Caching de requêtes dans le MVP
- Demos stakeholders hebdomadaires

Évolution méthodologique

AVANT	APRÈS
« Build first, evaluate later »	**Measure first, build with evidence**
Sauter au modèle directement	Baseline → gap analysis → solution
Évaluation informelle	Rubrics formelles (RAGAS, F1)
Documentation en afterthought	BRD, archi, decision matrix, rétro
Fini = entraîné	Déploiement = DÉBUT du cycle

> *Le Data Scientist n'est pas un modélisateur mais un **résolveur de problèmes data full-stack** : gouvernance, communication stakeholders, engineering de production, éthique, apprentissage continu.*

Le portfolio — construction & contenu

Démarche de construction

1 Inventaire
Recensement de 41 repositories GitHub (personnels, formation, OSS)

2 Sélection par pertinence
Filtrage sur compétences ML/DS : classification, RAG, MLOps, deep learning, NLP

3 Structuration en 4 sections
Compétences & projets, capacité réflexive, soft skills, mind map

4 Validation croisée
Chaque compétence est reliée à un projet et à des métriques mesurables

Structure du portfolio

§1 Compétences & Projets
12 projets détaillés avec stack, métriques, liens GitHub

§2 Capacité réflexive
Erreurs, leçons, évolution du regard sur le métier

§3 Soft skills
8 compétences illustrées par des exemples concrets

§4 Mind map
Vue synthétique de l'ensemble du profil et des connexions

Compétences démontrées

COMPÉTENCE	PREUVE
ML supervisé	HR Analytics (+269% F1), LOCALTRIAGE (73.3%)
RAG & LLM	LOCALTRIAGE (3.63/5, 100% citations)
MLOps	Pipeline (FastAPI + Evidently + CI/CD)
Deep learning	Semi-Supervised MRI (ResNet-18)
Gestion de projet	LOCALTRIAGE (3 sem, 8/8 livrables, solo)

> *Principe directeur : chaque affirmation est adossée à un livrable vérifiable.*

DÉMONSTRATION

Ouverture du Portfolio

Le portfolio HTML présente l'ensemble des projets, compétences et réflexions. Je vous propose de le parcourir ensemble.

Portfolio HTML en ligne

github.com/Septimus4/LOCALTRIAGE

Merci

Questions & échanges