F51 LABS · LABORATOIRES F51 INC.

Audit technique indépendant — état réel du système, avec registre des preuves

MIKE
Dossier technique d'audit

Architecture, maturité, risques, incertitudes technologiques
et feuille de route — édition du 2 octobre 2026

Entité	Laboratoires F51 inc. (F51 Labs), Montréal, Québec
Système audité	MIKE — agent commercial (voix et texte) pour PME du Québec, FR/EN, inférence locale
Version du produit	2026.10.02 (canal *preview*, flavor *development*)
Code audité	Dépôt `mike`, branche `review/modelos-2026-09-24`, commit `6f2d858` (tag de coupe `v2026.10.02` sur `3a3972a`)
Date de l'audit	2 octobre 2026 (mesures prises entre 13 h 06 et 14 h 00, heure de l'Est ; rédaction et mise en forme ensuite)
Poste audité	Windows 11 Pro 26200 · Core i7-12700K · 63,7 Go de RAM · RTX 5060 Ti 16 Go + RTX 3060 12 Go
Préparé pour	Marco Barreto, fondateur — usage prévu : évaluation technique externe, dossiers de financement R&D, *due diligence*
Méthode de rédaction	Audit mené avec l'assistance de Claude (Anthropic) : lecture du code, exécution de contrôles non destructifs, extraction de mesures depuis les journaux réels. Aucune affirmation n'est reprise de la documentation sans être marquée comme telle.

> **Verdict d'ingénierie.** Prototype fonctionnel avancé (maturité 3 sur 5) : adapté à une démonstration supervisée et à un pilote encadré ; **non prêt** pour une exploitation sans surveillance chez un client. 67 preuves tracées, 5 641 tests réussis, 177 tours vocaux réels analysés.

Confidentiel. Ce document décrit l'architecture interne d'un logiciel propriétaire. Il ne reproduit aucun secret (clés, jetons, mots de passe, contenu de conversations, numéros de clients). Les valeurs de configuration sensibles ne sont jamais citées. Audit mené avec l'assistance de Claude (Anthropic) ; chaque chiffre est relié à une preuve du registre (Annexe A).

Table des matières

2. Résumé exécutif 5
2.1 Ce qu'est le système 5
2.2 Ce qui est démontré 5
2.3 Ce qui ne l'est pas — constats critiques 5
2.4 Verdict d'ingénierie 6
3. Présentation de F51 Labs et de MIKE 6
4. Objectifs du système 6
5. Méthodologie d'audit 7
5.1 Principe 7
5.2 Contrôles exécutés (tous non destructifs) 7
5.3 Libellés d'état et niveaux de maturité 7
5.4 Limites de l'audit 7
6. Architecture générale 8
6.1 Vue d'ensemble 9
6.2 Principes structurants vérifiés dans le code 10
7. Architecture logicielle 10
7.1 Organisation du code 10
7.2 Cœur et orchestration (axe A) 11
7.3 Traitement d'erreur et repli 12
8. Infrastructure 13
8.1 Matériel mesuré 13
8.2 Pile logicielle 13
8.3 Volumes hors code dans le répertoire du dépôt 13
9. Pipeline IA 14
9.1 Du message du client à la réponse du modèle 14
9.2 Paramètres et limites 14
9.3 Calcul d'argent et exactitude 14
10. Architecture vocale 15
10.1 Preuve par le code : qualification du mode duplex 15
10.2 Parcours d'un appel entrant 16
10.3 Pipeline de voix 17
10.4 Interruptions, silences et reprises 19
10.5 Chaîne de latence mesurée 19
10.6 Appels simultanés 19
10.7 États des composants vocaux 19
11. Données et mémoire 21
11.1 Inventaire des bases 22
11.2 Entités principales et relations 22
11.3 Migrations, index, sauvegardes, réplication, chiffrement 22
11.4 Conséquences d'échelle de SQLite 23
12. RAG et connaissance 23
12.1 Chaîne 23
12.2 État mesuré 23
12.3 Isolation et contamination entre clients 24
13. Outils et MCP 24
13.1 Inventaire 24
13.2 Appel, permissions et traitement d'erreur 24
14. Multi-tenant 26
14.1 Identification du tenant 27
14.2 Ce qui est isolé 27
14.3 Ce qui est partagé — risques de fuite croisée 27
14.4 Contrôle d'accès 27
14.5 Maturité 27
15. Sécurité et garde-fous de conformité 28
15.1 Authentification et sessions 28

15.2 Autorisation et segmentation 28
15.3 Exposition réseau 28
15.4 Gestion des secrets 28
15.5 Chiffrement 28
15.6 Validation des entrées, exécution d'outils, injection 29
15.7 Journaux et données personnelles 29
15.8 Garde-fous de conformité (mécanismes techniques uniquement) 29

16. Auditabilité 31
16.1 Fonctionnement exact 32
16.2 Ce qui a été vérifié (mesure du 2 octobre 2026) 32
16.3 Ce qui n'est pas actif ou pas prouvé 32
16.4 Reconstruction d'un appel 32

17. Intégrations 33

18. Observabilité 34

19. Déploiement 36
19.1 Exécution réelle 37
19.2 Ordre de démarrage 37
19.3 Dépendances à reconstruire sur une machine neuve 37
19.4 Installateur 37
19.5 Constat 37

20. Validation et tests 37
20.1 Suite automatisée 37
20.2 Contrôles d'exécution réalisés pendant l'audit 38
20.3 Preuves d'exploitation antérieures (non refaites ici) 38
20.4 Ce qui n'a pas pu être validé 39

21. Performances mesurées 39
21.1 Cerveau local (Ornith 1.5 35B-A3B Q4_K_M) 39
21.2 Voix (extrait des journaux réels, 78 tours du 29 septembre au 2 octobre) 39
21.3 Ressources 40
21.4 Non mesuré 40

22. Matrice de maturité 40

23. Risques techniques 42
23.1 Écarts critiques 42
23.2 Risques de production 42
23.3 Points uniques de défaillance 43
23.4 Goulets d'étranglement, concurrence et échelle 43
23.5 Persistance 43
23.6 Sécurité propre à l'IA agentique 44
23.7 Observabilité et intégration 44

24. Dette technique 44

25. Incertitudes technologiques 45
25.1 Vue d'ensemble 45
U1 — Dialogue vocal naturel en français québécois sans modèle audio-natif 46
U2 — Latence de bout en bout ≤ 1,5 s (p95) sur matériel grand public 46
U3 — Reconnaissance vocale fiable du français québécois par téléphone 46
U4 — Gestion du tour de parole et du barge-in 47
U5 — Concurrence : N appels simultanés sur deux GPU grand public 47
U6 — Inférence locale MoE : partition CPU/GPU, stabilité et repli sûr 48
U7 — Isolation et équité entre clients pour les ressources partagées 48
U8 — Sécurité d'un agent à outils contre l'injection de prompt 48
U9 — Récupération de mémoire locale, performante et évaluée 49
U10 — Fiabilité des appels d'outils et vérité des actions déclarées 49
U11 — Évaluation objective de la qualité conversationnelle 50

26. Opportunités de R&D 50

27. Feuille de route 50
P0 — Bloqueurs avant toute exploitation chez un client 51
P1 — Importants 51
P2 — Améliorations 52
P3 — Futur 52

28. Conclusion **52**

29. Annexes techniques **54**

Annexe A — Registre des preuves techniques 54
Annexe B — Commandes et scripts exécutés 58
Annexe C — Inventaire de référence 59
Annexe D — Écarts entre la documentation et la réalité observée 59
Annexe E — Composants qui n'ont pas pu être validés 59
Annexe F — Glossaire 60

Lecture : chaque section renvoie à des identifiants de preuve (EV-001) définis dans le registre de l'Annexe A. Ce qui n'a pas pu être mesuré est écrit **NON MESURÉ**. Libellés d'état : IMPLÉMENTÉ ET TESTÉ · IMPLÉMENTÉ MAIS NON VALIDÉ · PARTIEL · PROTOTYPE · PLANIFIÉ / ABSENT.

2. Résumé exécutif

2.1 Ce qu'est le système

MIKE est un agent conversationnel destiné aux PME québécoises à rendez-vous (cliniques, ateliers, services). Il répond au téléphone en français et en anglais, consulte et modifie un agenda, enregistre des demandes de rappel et répond à partir d'une « fiche de l'entreprise » que le propriétaire remplit. Le choix de produit central est **local-first** : le modèle de langage, la reconnaissance vocale, la synthèse vocale et les données tournent sur un ordinateur appartenant à l'entreprise, sans appel à une IA infonuagique EV-020 EV-022. Selon le positionnement du produit, sous la Loi 25 du Québec, une donnée qui ne quitte pas la machine simplifie l'évaluation des facteurs relatifs à la vie privée ; c'est l'argument commercial *(positionnement du dépôt ; aucun avis juridique n'est donné ici)*.

Techniquement, MIKE est un système à **trois processus** : un orchestrateur FastAPI (le « corps », port 8083), un serveur d'inférence `llama.cpp` (le « cerveau », port 8081, modèle Ornith 1.5 35B-A3B en MoE, licence MIT) et un worker de voix isolé dans WSL2 (Whisper, Silero VAD, Smart Turn, Kyutai TTS, port 8998) EV-024 EV-025. Le code compte environ 135 000 lignes Python dans `core/`, 91 000 lignes de tests, 28 000 lignes de JavaScript/HTML/CSS pour les trois surfaces web EV-087.

2.2 Ce qui est démontré

- **La suite de tests automatisés passe intégralement** : 5 641 tests réussis, 4 ignorés, 1 échec attendu documenté, sur le commit audité, en 245 s EV-080.
- **La chaîne d'audit cryptographique de l'époque vivante est intègre** : 10 518 enregistrements, chaînage SHA-256 et signature Ed25519 vérifiés sur une copie en 1,29 s ; une falsification volontaire est détectée au bon enregistrement EV-030 EV-031.
- **Le cerveau local est mesuré** : décodage ≈ 67 jetons/s, préremplissage ≈ 626 jetons/s, premier jeton ≈ 77 ms avec cache de prompt et ≈ 0,92 s sans cache EV-021.
- **Des appels téléphoniques réels ont eu lieu** par Twilio, dont un appel à une personne qui a abouti à une réservation écrite dans l'agenda canonique EV-083.
- **L'isolation par tenant et le contrôle d'accès par route sont mécaniquement vérifiés** : 283 routes déclarées avec un rôle, zéro dette d'exception EV-042.

2.3 Ce qui ne l'est pas — constats critiques

1. **La latence vocale n'atteint pas la cible documentée.** Sur 78 tours réels depuis l'adoption de la synthèse Kyutai (29 septembre), le délai entre la fin de la parole du client et le premier son est **médian 2,9 s, 95e centile 5,8 s**, contre une cible documentée de 1,0 s / 1,5 s. Les tours qui passent par le modèle sont à 3,5 s / 6,2 s EV-009 EV-010.
2. **La voix n'est pas du full-duplex.** C'est une **cascade avec écoute continue** (pseudo-full-duplex) : le micro reste ouvert pendant que MIKE parle et un barge-in est géré, mais la reconnaissance **ne fait pas de streaming** (Whisper transcrit le tour entier), le modèle ne génère rien pendant que le client parle, et MIKE ne produit pas de *backchannels* EV-001 EV-003 EV-014 .
3. **La concurrence est limitée à environ un appel à la fois, et rien ne l'a mesuré.** Un seul créneau LLM (`--parallel 1`), une seule synthèse TTS à la fois (verrou), un seul Whisper résident ; **aucun plafond d'admission d'appels** n'existe dans le code EV-006 EV-008 EV-012. Les appels simultanés sont **NON MESURÉS**.
4. **Détection de répondeur absente du chemin conversationnel, pas de transfert vers un humain, pas de liste nationale de non-appel.** Une détection synchrone du fournisseur existe mais est désactivée par défaut (elle retarde la salutation d'environ 3 s) et n'est lue que par l'IVR de rappel de rendez-vous ; la route de dialogue — celle des appels conversationnels — ne lit jamais le résultat. Un appel sortant conversationnel tombé en boîte vocale est donc traité comme une conversation EV-016 EV-017 EV-061.
5. **Données au repos non chiffrées sur le poste audité, secrets en clair.** Le choix de conception est de ne pas chiffrer dans l'application mais de déléguer au chiffrement de volume (BitLocker), que le code *vérifie* et dont il refuse d'affirmer la conformité sans preuve. Or **BitLocker est désactivé sur le volume C: et sur la clé USB de sauvegarde** (mesuré par la propriété de shell Windows `System.Volume.BitLockerProtection` = 2), aucune attestation d'opérateur n'est enregistrée et le mode est « avertir » (`warn`). Les secrets de session et d'audit sont des fichiers non chiffrés. Par ailleurs, l'ancrage RFC 3161 de la piste d'audit est implémenté mais **désactivé, avec zéro ancre**, et le jeton de l'autorité d'horodatage n'est pas vérifié cryptographiquement par le code EV-046 EV-034.
6. **La mémoire vectorielle est très partielle et non évaluée** : 216 vecteurs pour 64 559 segments du tenant `default` (0,33 %) — dont 96,7 % sont des romans du domaine public indexés en plein-texte seulement, et 10 % du contenu métier est

vectorisé — ; la recherche vectorielle est un balayage linéaire sans index ; le moteur de graphe (LightRAG) est « activé mais non chargé » ; aucun jeu d'évaluation de récupération n'existe EV-071 EV-072 EV-073.

7. **L'époque historique de la piste d'audit (34,6 Mo) est rompue** (HMAC invalide à l'enregistrement 40, chaîne rompue au 27 741) ; une nouvelle époque a été ouverte le 23 septembre EV-032.
8. **L'installateur Windows n'a jamais été installé ni signé** ; le runtime audité est le dépôt, avec les garde-fous de conformité volontairement désactivés pour le développement EV-084 EV-048.

2.4 Verdict d'ingénierie

Le système est un **prototype fonctionnel avancé** (niveau de maturité 3 sur 5 pour l'ensemble), avec plusieurs composants au niveau 4 (agenda canonique, contrôle d'accès, chaîne d'audit, garde de promesses). Il est **adapté à une démonstration supervisée** et à des essais pilotes encadrés. Il n'est **pas prêt pour une exploitation sans surveillance chez un client** : les écarts principaux (latence vocale, concurrence, répondeur, chiffrement, installation propre, observabilité) sont décrits au §23 et ordonnés au §27. Les **incertitudes technologiques** qui relèvent de la recherche appliquée — et non du développement courant — sont isolées au §25.

3. Présentation de F51 Labs et de MIKE

F51 Labs (raison sociale *Laboratoires F51 inc.*, Montréal) est l'éditeur de MIKE ; le logiciel est propriétaire (© 2025-2026 Marco Barreto, fichier `LICENSE`). Le dépôt compte 1 002 commits depuis le 29 juillet 2026 EV-085.

MIKE est distribué comme l'offre **« MIKE Réception »** (identifiant `mike-reception`), décrite par le manifeste canonique `config/product_manifest.json` : marché « PME québécoises à rendez-vous », transversale aux secteurs, langues `fr-CA` et `en-CA`, flux principal *répondre → disponibilités → créer → reprogrammer → annuler* EV-024.

Autour de cette offre, le dépôt contient aussi des capacités plus larges d'un « agent commercial » : 59 compétences déclaratives (`skills/*.yaml`), 25 missions canoniques, 20 serveurs MCP (117 outils Python), une mémoire multi-bancs et un moteur d'autonomie (routines, tâches, gouvernance) EV-086 EV-087. Le présent audit distingue ce qui a été **exercé** de ce qui est seulement **présent** dans le code.

Environnement de démonstration constaté : le tenant `f51-labs` porte la fiche d'une clinique dentaire fictive de démonstration ; le tenant `mike-ventes` (créé le 2 octobre 2026) décrit MIKE lui-même pour des appels de prospection ; 21 autres espaces `qc_*` contiennent des données apparemment synthétiques et ne sont pas classifiés EV-070.

4. Objectifs du système

Les objectifs ci-dessous sont déduits du manifeste, du code et des politiques par défaut — non d'un énoncé marketing.

#	Objectif	Mécanisme constaté	État
O1	Aucune donnée de client hors de la machine sans décision explicite	Un seul backend LLM réel (`llama_server`) ; le backend infonuagique est refusé au démarrage (`assert_llm_backend_local_only`) ; consentement `external_connectors` exigé avant tout flux téléphonique EV-022 EV-044	Implémenté et testé
O2	Ne jamais simuler un succès	Erreurs réelles (chat 500, autonomie 503) ; garde qui bloque toute promesse parlée sans reçu (« rappel », « réservé », « transféré ») EV-013	Implémenté et testé
O3	Répondre au téléphone en FR et EN de façon naturelle	Cascade ouïe → décision → parole avec barge-in EV-001 EV-002	Partiel (latence hors cible EV-009)
O4	Gérer l'agenda de l'entreprise de façon fiable	Agenda canonique SQLite avec révisions, idempotence et outbox EV-074	Implémenté et testé
O5	Prouver ce qui a été dit et fait	Chaîne d'audit SHA-256 + Ed25519 ; transcriptions scellées EV-030	Implémenté et testé (ancrage externe désactivé EV-034)
O6	Isoler les clients entre eux	Contexte de tenant par session signée ; 283 routes à rôle déclaré EV-040 EV-042	Partiel (ressources partagées, EV-041)
O7	Respecter les règles d'appel sortant au Canada	Pré-vol P1–P7 (consentement, liste d'exclusion locale, fenêtre horaire, tentatives, DID, politique de répondeur, attestation) EV-060	Partiel (EV-016 EV-061)

5. Méthodologie d'audit

5.1 Principe

Une affirmation n'est retenue que si elle repose sur au moins l'une des preuves suivantes, par ordre de force croissante : **lecture du code** (L), **test automatisé** qui l'exerce et passe (T), **mesure ou exécution pendant l'audit** (M), **donnée d'exploitation réelle analysée** (D, p. ex. journaux d'appels), **fournisseur externe** (P, p. ex. Twilio). La documentation du dépôt (CLAUDE.md, README, `docs/`) a été traitée comme *hypothèse à vérifier*, jamais comme preuve. Lorsqu'un chiffre provient d'un document non re-mesuré, il est marqué **(source : document, non re-vérifié)**.

5.2 Contrôles exécutés (tous non destructifs)

Contrôle	Résultat
Inventaire du dépôt (`inventory.py`) : lignes de code, routes, tests, MCP, variables (noms), bancs SQLite en lecture seule	`raw/inventory.json` EV-087
Sondes `GET` : `/health`, `/livez`, `/readyz` (MIKE), `/health`, `/v1/models`, `/props` (llama-server)	HTTP 200 ; les endpoints `/v1/health/latency`, `/vram`, `/tts` et `/metrics` répondent 401 (session de profil requise) EV-024
Ligne de commande réelle du `llama-server` (sans secrets)	Paramètres relevés EV-020
Microbenchmark du cerveau local (`bench_llm.py`, 5 répétitions)	Décodage, préremplissage, premier jeton EV-021
Vérification de la piste d'audit sur **copie** (`verify_ledger_copy.py`) + test de falsification	10 518 enregistrements intègres ; falsification détectée EV-030 EV-031
Extraction des latences vocales réelles (`voice_latency_from_logs.py`, `voice_trace_from_logs.py`)	177 tours (`VOICE_LATENCY`) et 193 traces de tour EV-009 EV-011
Suite de tests complète (`pytest -q`)	5 641 réussis EV-080
Requêtes SQL en lecture seule (`mode=ro`) sur les bases de mémoire et d'agenda	Tailles, tables, couverture vectorielle EV-070 EV-071
Sonde du worker de voix côté WSL (processus, versions, GPU, disque)	EV-027
Contrôle des droits NTFS (`Get-Acl`) et du suivi `git` des fichiers sensibles	EV-047 EV-052

5.3 Libellés d'état et niveaux de maturité

Libellés d'état (utilisés dans tout le document) : **IMPLÉMENTÉ ET TESTÉ** · **IMPLÉMENTÉ MAIS NON VALIDÉ** · **PARTIEL** · **PROTOTYPE** · **PLANIFIÉ / ABSENT**.

Niveaux de maturité : **0** absent · **1** concept · **2** prototype · **3** fonctionnel · **4** validé · **5** prêt pour la production. Un niveau 4 exige une preuve d'exécution (test ou mesure) *et* l'absence de défaut critique connu ; un niveau 5 exige en plus une preuve sur l'artefact installé — qui n'existe pour aucun composant de ce système EV-084.

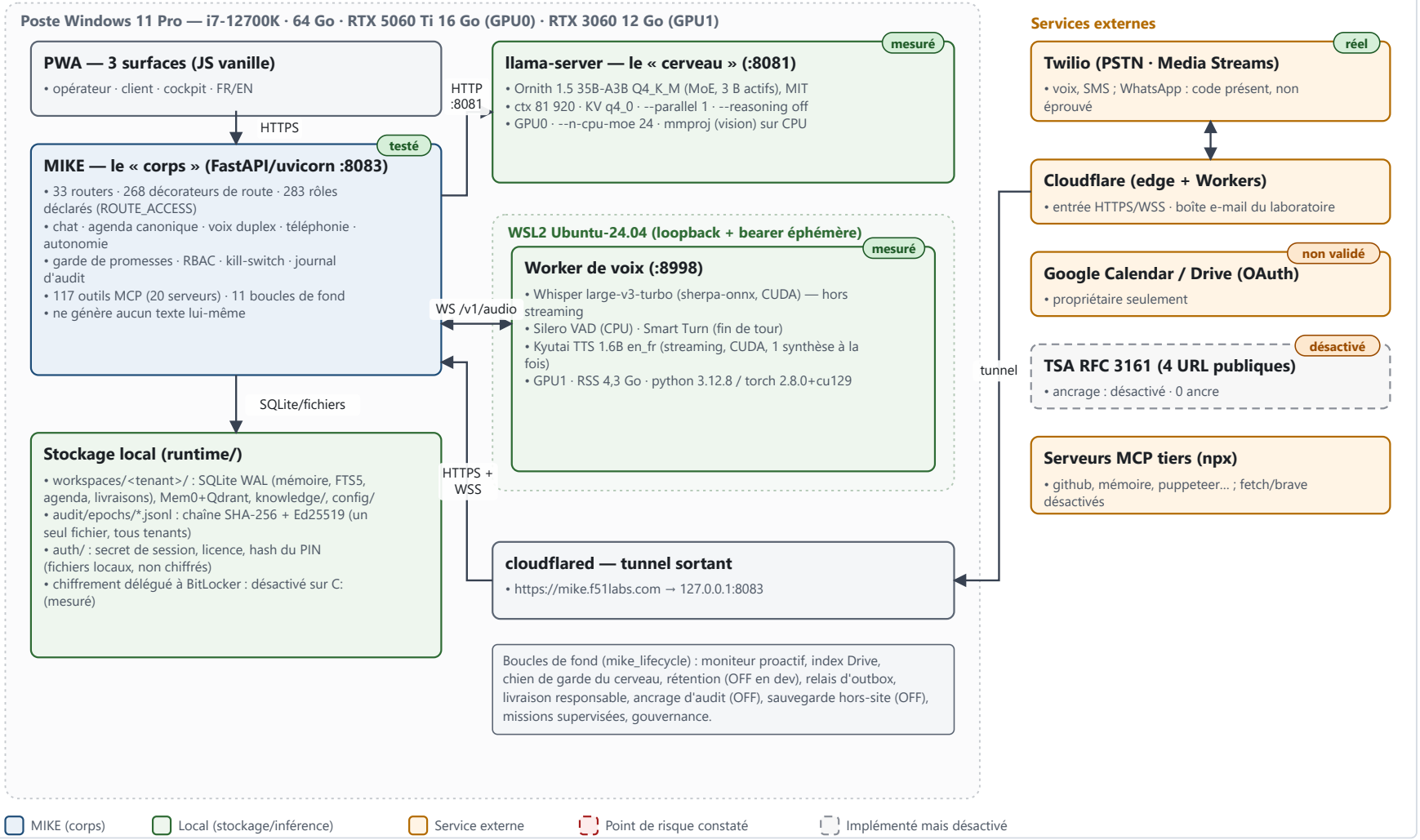
5.4 Limites de l'audit

- **Pas d'appel téléphonique, SMS, WhatsApp ni courriel émis** par cet audit. Les appels réels cités sont antérieurs et proviennent des journaux et des preuves existantes.
- **Pas de test de charge** ni d'appels simultanés (risque de perturber le service en cours et le GPU partagé). La concurrence est donc **NON MESURÉE**.
- **Pas de redémarrage du système**. Le temps de démarrage à froid est **NON MESURÉ**.
- **Endpoints authentifiés** (latence, VRAM, TTS, métriques) non interrogés : aucune session de profil n'a été utilisée ni forgée.
- **Le vérificateur de prontidité du projet** (`check_mike_readiness.py`) exige le PIN de l'opérateur ; il est resté bloqué et a été arrêté. Aucun verdict de ce script n'est revendiqué EV-082.
- Les **sous-agents spécialisés du dépôt** n'ont pas pu être utilisés (erreur 404 sur le modèle configuré dans l'environnement) ; l'audit a été mené directement, sans seconde vérification indépendante.
- **Incident de manipulation** : une commande de relevé de configuration a affiché, dans la sortie d'outil de la session d'audit, la valeur de la clé de signature de la piste d'audit (`MIKE_AUDIT_SIGNING_KEY`). Cette valeur n'a été reproduite dans aucun fichier du dossier. Sa rotation (nouvelle époque d'audit) est recommandée au §27 (P0).

6. Architecture générale

6.1 Vue d'ensemble

Figure 1 — Architecture générale du système MIKE (état constaté le 2026-10-02)



Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

MIKE s'exécute sur **un seul hôte Windows** et se compose de trois processus qui communiquent par des interfaces locales EV-025 :

Processus	Rôle	Interface	Technologie constatée
MIKE (« corps »)	Orchestrateur : chat, agenda, téléphonie, mémoire, outils, autonomie, audit, contrôle d'accès. **Ne génère aucun texte lui-même.**	HTTP/WS `127.0.0.1:8083`	Python 3.12.10, FastAPI/uvicorn, 33 routers enregistrés (0 en échec) EV-024
llama-server (« cerveau »)	Inférence du modèle de langage	HTTP compatible OpenAI `127.0.0.1:8081`	`llama.cpp` build `b10826-73a43d1f6`, Ornith 1.5 35B-A3B Q4_K_M EV-020
Worker de voix	Ouïe (VAD, tour de parole, reconnaissance) et parole (synthèse)	WS authentifié `127.0.0.1:8998` (via `wslrelay`)	WSL2 Ubuntu-24.04, Python 3.12.8, torch 2.8.0+cu129, sherpa-onnx 1.13.7 EV-027

À ces trois processus s'ajoutent un tunnel sortant `cloudflared` (webhooks Twilio vers `https://mike.f51labs.com`) et les surfaces web, servies par MIKE : un **PWA en JavaScript sans framework** (≈ 16 000 lignes JS, 4 200 HTML, 7 900 CSS) décliné en trois surfaces — opérateur, client et cockpit — en FR et EN EV-087.

6.2 Principes structurants vérifiés dans le code

1. **Local-first sans repli silencieux.** La chaîne de repli du modèle ne contient qu'un backend réel (`llama_server`) ; le backend `mock` n'existe que s'il est sélectionné explicitement, et un backend infonuagique est refusé au démarrage par `assert_llm_backend_local_only` (`core/server/mike_lifecycle.py`, test `tests/unit/test_single_brain.py`) EV-022.
2. **Les pannes réelles restent des erreurs.** Sans cerveau : chat en erreur 500, autonomie/moniteur en 503. Sans worker de voix : pas de voix locale, ce qui est dit dans `/v1/health/tts` plutôt que dissimulé EV-024.
3. **Fail-closed.** Le *kill-switch* gèle `run_command`, les mutations de fichiers, le cycle de gouvernance, la réponse automatique et la synthèse vocale (`assert_not_frozen`). Ne pas pouvoir évaluer le kill-switch **interdit** l'action EV-049.
4. **Pas d'affirmation sans reçu.** Une phrase promettant un rappel, une réservation ou un transfert est filtrée si aucun reçu typé du même tenant, appel et tour ne l'appuie (`core/voice/mike_voice_claims.py`) EV-013.
5. **Identité par la session signée.** La frontière de tenant naît de la session HMAC (`extract_profile_session` → `TenantContext.from_session`), jamais d'un en-tête ou d'un paramètre de requête EV-040.

7. Architecture logicielle

7.1 Organisation du code

Paquet (`core/`)	Fichiers Python	Lignes	Rôle
`server/`	157	72 512	FastAPI, authentification, chat, SSE, outils locaux, routes (`mike_routes_*`, `routers/`), cycle de vie
`autonomy/`	34	20 263	Routines, missions (25 canoniques), gouvernance, compétences, tableau de tâches, kill-switch
`voice/`	26	12 111	Moteurs et protocole de voix (le code s'exécute dans le worker WSL)
`mcp/`	33	8 946	Serveurs MCP : agenda, calendrier, Drive, Excel, GA4, Ads, Shopify, prix, scraping…
`memory/`	14	6 859	SQLite, recherche hybride, Mem0, LightRAG, reranking, embeddings
`comms/`	9	5 225	Twilio (voix, SMS), webhooks, boîte courriel, fiche d'entreprise, identité téléphonique
`audit/`	7	2 551	Piste d'audit : chaîne, signatures, Merkle, ancrage RFC 3161
`chat/`	3	1 953	Contexte de conversation
`orchestration/`	4	1 843	TaskMesh, SDK d'agents, essaim
`integrations/`	5	1 691	Client `llama-server`, Google OAuth, recherche web
autres	12	1 003	`security`, `shared`, `docgen`, `installer`
Total	**304**	**134 960**	

Source : `inventory.py` EV-087. La concentration est forte : `core/server` porte 54 % du code et les fichiers les plus gros dépassent 3 000 lignes (`mike_voice_flow.py` 3 809, `mike_tools_local.py` 3 738, `routers/telephony.py` 3 047), ce qui est une dette de maintenabilité (§24).

7.2 Cœur et orchestration (axe A)

Processus principal. Un seul processus uvicorn (PID 33412 au relevé, 811 Mo de RSS) héberge toutes les routes, le moteur d'autonomie et les boucles de fond EV-025. Le point d'entrée est `core/server/mike_server.py` ; le cycle de vie (`mike_lifecycle.py`) construit les services au démarrage et attend le cerveau (`MIKE_BRAIN_BOOT_TIMEOUT`, 300 s selon la documentation) au lieu d'abandonner à la première sonde.

Boucles de fond. Le cycle de vie démarre jusqu'à onze tâches asynchrones : moniteur proactif, indexation Drive, **chien de garde du cerveau** (relance le *même* `start_qwen36_server.ps1`), rétention, relais d'outbox, livraison « responsable », ancrage d'audit, sauvegarde hors-site, une tâche en fil séparé, missions supervisées et gouvernance. Les tâches d'ancrage et de sauvegarde sont **désactivées par défaut** ; la rétention est **désactivée sur ce poste** EV-023 EV-048.

Gestion des appels. Chaque appel est une connexion WebSocket `/v1/voice/stream` qui crée une session de worker, puis deux tâches `asyncio` parallèles — pompe audio de sortie (`_pump_agent_audio`) et pompe d'évènements de l'ouïe (`_pump_worker_events`) — plus des tâches accessoires (salutation différée, veille de silence). Un tour de parole est une tâche annulable (`_run_voice_turn`) EV-001. L'état de conversation est tenu en mémoire par `CallSid` (`voice_flow`), les reçus de lecture par le registre `_HangupDelivery`, et la transcription finale est scellée dans la piste d'audit à la fermeture EV-030.

Machines à états. Trois machines à états explicites ont été relevées : le tour de parole du worker (`TurnState` : *user_speaking → endpoint_pending → transcript_final…*), les rendez-vous canoniques (transitions interdites p. ex. *REMINDER_SENT → DECLINED* refusées et testées) et les ordres durables d'autonomie (`WorkStatus`, contrats seulement ; le magasin durable n'est **pas encore relié** aux moteurs — tâches 1 à 3 sur 12 selon `PRODUCT_READINESS.md`).

Concurrence et parallélisme. La boucle d'événements `asyncio` est protégée contre les E/S bloquantes par `asyncio.to_thread` (poll de l'ouïe, écritures SQLite, ancrage). Le worker interroge l'ouïe toutes les **20 ms** (`POLL_SECONDS = 0.02`) ; la transcription anticipée partage un **pool de 2 threads** pour *toutes* les sessions ; l'accès GPU du Whisper et la gravure de CUDA graph de la Kyutai sont sérialisés par un verrou commun (`core/voice/mike_gpu_capture.py`) EV-006 EV-007. Il n'existe **aucun plafond d'appels simultanés** ni file d'admission au niveau des appels EV-008.

Persistance d'état. SQLite en mode WAL par tenant (mémoire, agenda, livraisons), fichiers JSON de configuration par tenant, et une piste d'audit JSONL unique. Aucun service de base externe n'est utilisé EV-070.

7.3 Traitement d'erreur et repli

Figure 8 — Traitement d'erreur et repli (règle : jamais de repli silencieux)



Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

Le tableau de la Figure 8 résume 12 défaillances et la réaction constatée. Deux points méritent attention : (i) `MIKE_AUDIT_STRICT=false` fait qu'une panne d'écriture d'audit **n'empêche pas** l'action EV-035 ; (ii) les files d'inférence sont bornées (`max_waiters` = 2 par défaut) et lèvent une erreur explicite plutôt que d'attendre sans limite EV-012.

8. Infrastructure

8.1 Matériel mesuré

Ressource	Valeur constatée EV-026
Processeur	Intel Core i7-12700K — 12 cœurs / 20 threads
Mémoire vive	63,7 Go (environ 16 Go libres au moment du relevé)
GPU0	NVIDIA GeForce RTX 5060 Ti 16 Go — cerveau ; **sert aussi l'affichage du bureau**
GPU1	NVIDIA GeForce RTX 3060 12 Go — voix (Whisper + Kyutai)
Pilote	595.97 (les deux cartes)
Disques	C: NTFS 948 Go (192 Go libres) ; pas de redondance
Système	Windows 11 Pro, build 26200 ; WSL2 Ubuntu-24.04 pour la voix

Occupation VRAM relevée : GPU0 11,5 Go / 16,3 Go, GPU1 8,8 Go / 12,3 Go EV-026. La documentation du dépôt explique que `--n-cpu-moe 24` est un choix propre à ce poste : le GPU0 est partagé avec le bureau (base mesurée entre 0,76 et 4,1 Go), et une marge insuffisante fait basculer le pilote vers la mémoire partagée *sans erreur*, avec un effondrement du décodage d'environ 70 % *(source : `docs/arquitectura/hardware.md` §2.0.2, non re-mesuré)*.

8.2 Pile logicielle

Couche	Composant	Version/paramètre constaté
Orchestrateur	Python, FastAPI, uvicorn	CPython 3.12.10 (`.venv`) EV-025
Inférence	llama.cpp (`llama-server`)	build b10826-73a43d1f6 ; `--ctx-size 81920 --flash-attn on --cache-type-k/v q4_0 --parallel 1 --cont-batching --cache-reuse 256 --jinja --reasoning off --n-gpu-layers 99 --n-cpu-moe 24 --threads 12` EV-020
Modèle	Ornith 1.5 35B-A3B Q4_K_M	35 505 251 456 paramètres, MoE 256 experts/8 actifs ; contexte d'entraînement 262 144 ; vocabulaire 248 320 ; licence MIT EV-020
Vision	`mmproj-Ornith-1.5-35B-BF16.gguf` sur CPU	modalités annoncées : vision et vidéo (audio : non) EV-020
ASR	Whisper large-v3-turbo (sherpa-onnx, fp32, CUDA)	un seul moteur de reconnaissance résident (une langue à la fois, ~3,3 Go) EV-003
VAD	Silero VAD	CPU par défaut (`MIKE_VAD_PROVIDER=cpu`)
Fin de tour	Smart Turn v3.2	CUDA (manifeste)
TTS	Kyutai TTS 1.6B en_fr (moshi, torch 2.8.0+cu129) ; Kokoro en repli explicite	24 kHz → µ-law 8 kHz ; `n_q = 32` EV-006
Embeddings	BGE-M3 ONNX 1 024 dimensions, local	≈ 2,3 Go (poids ONNX)
Bases	SQLite WAL (+FTS5), Qdrant local (Mem0), LightRAG	EV-070
Tunnel	cloudflared	hôte public `mike.f51labs.com`
Tests	pytest	5 641 tests collectés et passés EV-080

8.3 Volumes hors code dans le répertoire du dépôt

Le dossier du dépôt héberge des volumes qui n'appartiennent pas au code : `artifacts/` 93,2 Go, `build/` 36,7 Go (dont les staging de l'installateur), `training/` 18,2 Go (pipeline LoRA, données et poids), `llama.cpp/` 4,5 Go, `runtime/` 5,0 Go, `.test-`

`tmp/` 3,6 Go, `llama.cpp.new/` 2,1 Go EV-085. Cela est cohérent avec un poste de développement mais complique la reproductibilité et la sauvegarde ; l'inventaire des zones réellement indispensables à la restauration figure au §19.3.

9. Pipeline IA

9.1 Du message du client à la réponse du modèle

Le chemin de **chat** (surface web) et le chemin de **voix** convergent vers le même serveur d'inférence mais empruntent des routes distinctes :

1. **Identification.** La requête porte une session signée ; le `TenantContext` est lié au contexte d'exécution (`ContextVar`) pour toute la durée de la requête EV-040.
2. **Contexte.** `core/chat/mike_context.py` assemble le prompt système (identité de l'agent, règles), la **fiche de l'entreprise** du tenant actif (`config/business.json`, injectée à la voix *et*, depuis le 1er octobre 2026, au chat), la mémoire récupérée (FTS, vecteurs, Mem0) et des « faits système » (date locale du Québec, jours fériés ; source : document) EV-088.
3. **Décision.** Pour la voix, un **flux déterministe d'agenda** (analyseurs FR/EN à expressions régulières) répond sans appeler le modèle lorsqu'il le peut ; sinon le tour est envoyé au cerveau. Sur 78 tours réels récents, **34 ont été résolus sans modèle et 44 ont requis le modèle** EV-010.
4. **Admission d'inférence.** Le client `llama-server` (`core/integrations/mike_llama_server_client.py`) place la requête dans une file bornée avec trois classes d'urgence — `realtime` (voix), `interactive`, `batch` (routines, indexation) — au plus un appel actif, délai d'attente borné, et un portail d'admission par VRAM qui peut refuser sans enfiler EV-012.
5. **Génération.** `POST /v1/chat/completions` en *streaming* ; les jetons sont découpés en phrases pour la synthèse. L'appel d'outils utilise le *function calling* du gabarit `--jinja` (le gabarit annonce `supports_tool_calls` et `supports_parallel_tool_calls`) EV-020.
6. **Boucle d'outils.** `MIKE_MCP_MAX_STEPS=8` borne les enchaînements d'outils par message.
7. **Garde de sortie.** Avant d'être dites, les phrases passent par le filtre de promesses ; un rappel offert par le modèle est transformé en demande de rappel *enregistrée* avant d'être prononcé, ou remplacé par une formule honnête EV-013.

9.2 Paramètres et limites

Paramètre	Valeur	Remarque
Fenêtre de contexte	81 920 jetons	31 % du contexte d'entraînement (262 144)
Cache KV	`q4_0` (clés et valeurs)	compromis mémoire/qualité **non évalué** ici
Concurrence	1 créneau (`--parallel 1`)	`--cont-batching` actif mais sans second créneau
Raisonnement	désactivé (`--reasoning off`, `MIKE_ENABLE_THINKING=false`)	décision documentée : avec les budgets de jetons de la voix (60 à 4 096), le raisonnement consomme le budget et renvoie un contenu vide *(source : CLAUDE.md, non re-vérifié)*
Échantillonnage par défaut du serveur	température 1, top-k 20, top-p 0,95	valeurs par défaut du serveur ; les appels de MIKE fixent leurs propres paramètres
Modèle de brouillon	aucun (`MIKE_MODEL_DRAFT=none`)	décodage multi-jetons natif du modèle
Délais	file : borné (`queue_timeout_seconds`) ; santé du serveur : 2 s	

9.3 Calcul d'argent et exactitude

Le code contient un *intercepteur déterministe* de TPS/TVQ (`core/server/mike_tax_intercept.py`) parce que le modèle seul est instable sur ces calculs : le dépôt consigne 3 bonnes réponses sur 9 et des résultats différents à entrée identique *(source : `docs/delivery/PRODUCT_READINESS.md`, non re-mesuré)*. Aucun autre calcul monétaire (remises, sous-totaux, rapport fiscal trimestriel promis par une mission) n'a de garde équivalente — c'est un risque d'exactitude consigné au §23.

10. Architecture vocale

> **Verdict technique sur le mode duplex.** MIKE n'est **ni half-duplex strict, ni full-duplex réel**. C'est une **cascade « ouïe → décision → parole » avec écoute continue et barge-in par règles** — un *pseudo-full-duplex*. Le micro reste ouvert pendant que MIKE parle et une interruption est gérée ; mais la reconnaissance ne fait pas de streaming, le modèle ne génère rien tant que le client n'a pas terminé un tour, et MIKE ne produit pas de *backchannels*. Preuves : EV-001 EV-002 EV-003 EV-004 EV-014.

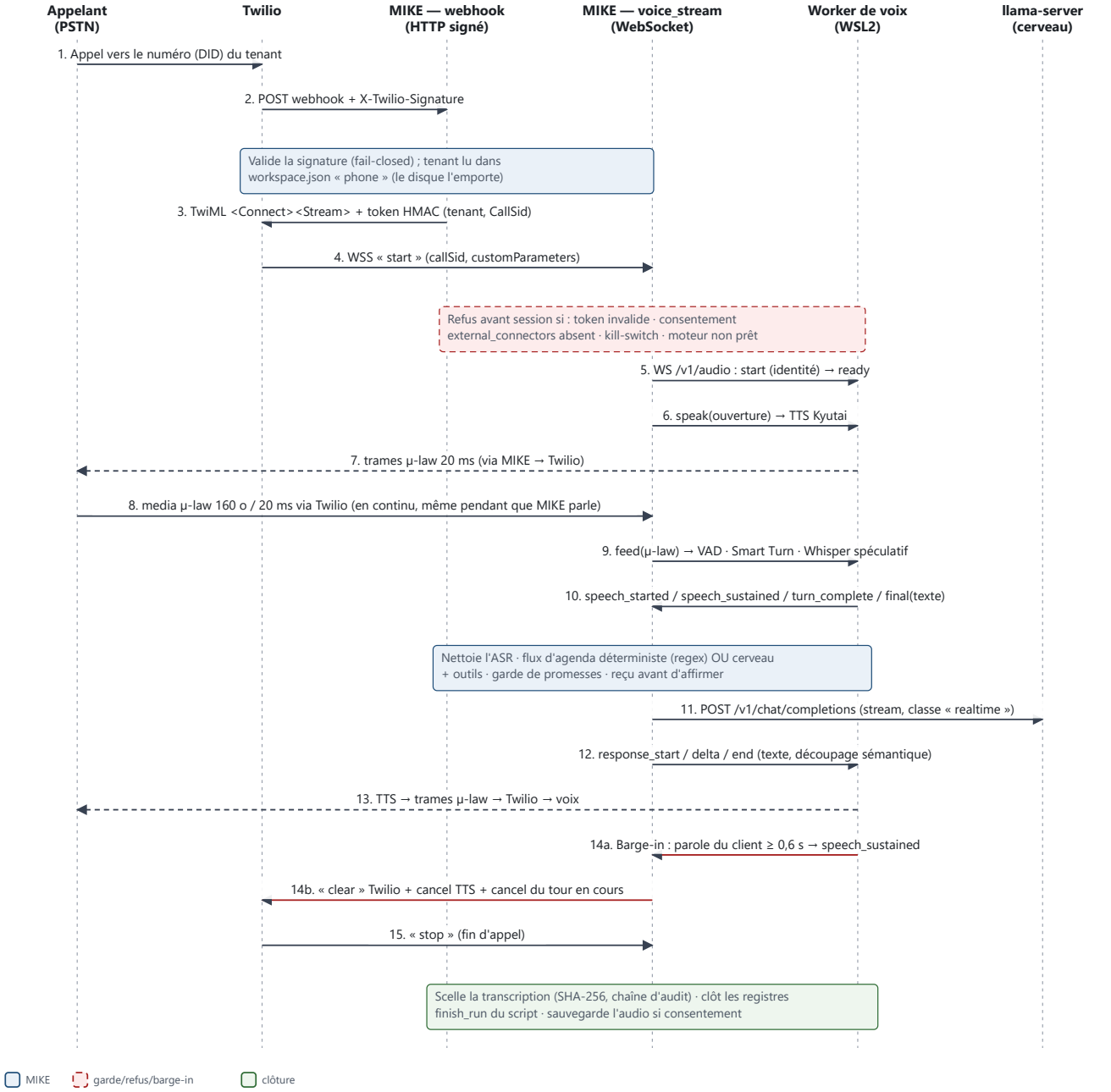
10.1 Preuve par le code : qualification du mode duplex

Critère	Half-duplex	Pseudo-full-duplex	Full-duplex réel	Constat sur MIKE	Preuve
Audio entrant traité pendant que MIKE parle	non	oui	oui	**Oui.** Chaque trame `media` du Twilio est transmise à `session.feed()` sans condition d'état de lecture ; l'ouïe tourne en continu	`voice_stream()` — événement `media` ; `_WorkerConnection.feed` EV-001
Interruption du MIKE par le client	non	oui (règles)	oui (apprise)	**Oui, par règles** : plan d'arrêt — son ≥ 0,6 s (`speech_sustained`) ou texte qui n'est pas un simple acquiescement coupe la lecture (`clear` Twilio + annulation du TTS + annulation du tour)	`_pump_worker_events._barge` , `_WorkerConnection._poll` EV-002
Reconnaissance en continu (hypothèses partielles)	non	partielle	oui	**Non.** `WhisperStream.step()` , `partial()` retournent toujours `""` , `endpoint()` toujours `False` ; le tour entier est transcrit après sa clôture	`WhisperStream` EV-003
Raisonnement/génération pendant que le client parle	non	limitée	oui	**Non.** Aucun appel au modèle avant l'évènement `final` ; seule la *transcription* est anticipée (pause ≥ 0,2 s)	`StreamingEars._start_speculation` ; `_run_voice_turn` EV-004
Backchannels produits par le système (« mm-hm »)	non	parfois	oui	**Non produits.** MIKE *ignore* ceux du client (liste FR/EN) ; sa seule intervention spontanée est « Vous êtes toujours là ? » après 8 s de silence	`voice_flow.is_backchannel` , `_idle_watch` EV-014
Modèle unique audio-à-audio	non	non	oui	**Non.** Chaîne de cinq modèles distincts (VAD → Smart Turn → Whisper → LLM → Kyutai)	EV-014

Conclusion : le dispositif relève de la **cascade à écoute continue**. Il évite les deux défauts du half-duplex (client muet pendant que MIKE parle ; MIKE incapable de s'arrêter), mais il hérite des limites de la cascade : latence cumulée de cinq étapes et absence de prosodie conversationnelle (chevauchements, *backchannels*, complétion de phrase).

10.2 Parcours d'un appel entrant

Figure 2 — Flux d'un appel téléphonique entrant (code : routers/telephony.py, routers/voice_stream.py)

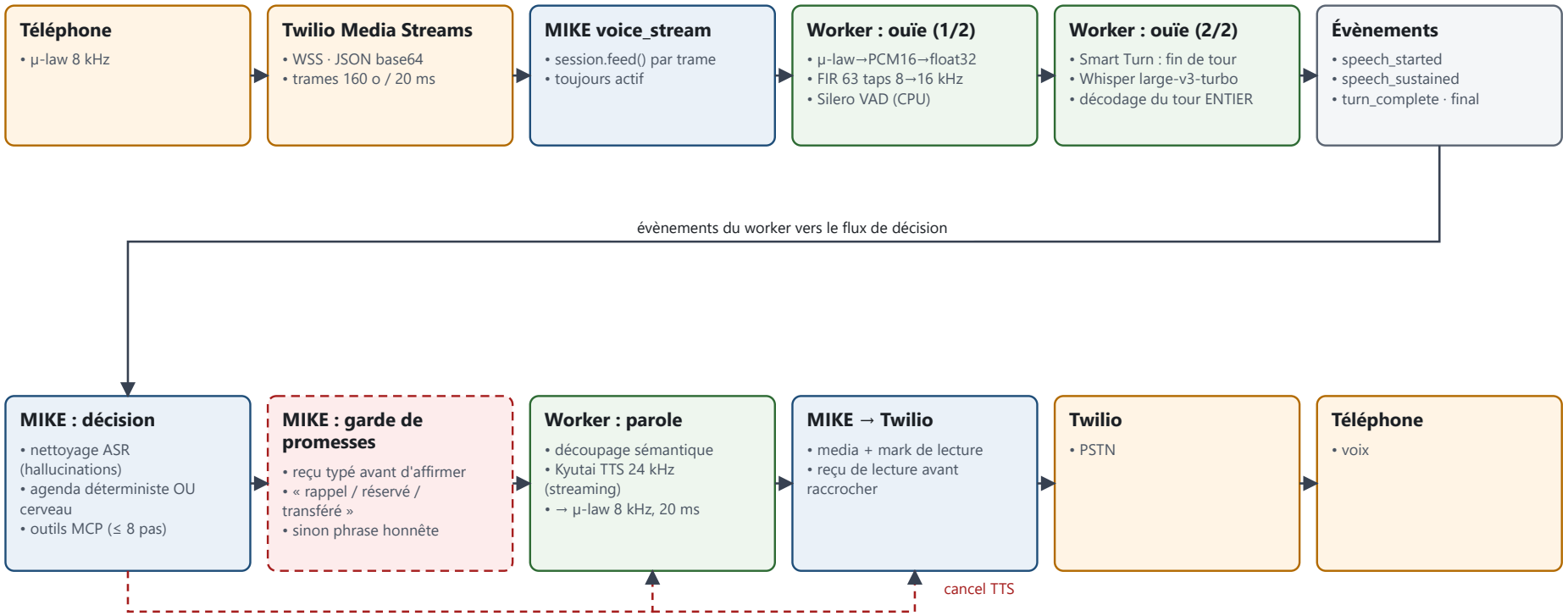


Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

Le webhook Twilio (signé, voir §15) renvoie un TwiML `<Connect><Stream>` portant un **jeton HMAC** lié au tenant et au `CallSid` ; le tenant qui répond est lu dans `workspace.json` (clé `phone`) et non dans une variable d'environnement (qui n'est qu'une réserve et provoque une erreur de configuration si elle diverge) EV-043 EV-090. À l'évènement `start` du WebSocket, la session est **refusée avant d'exister** si le jeton est invalide, si le consentement `external_connectors` du tenant est absent, si le kill-switch est engagé ou si le moteur n'est pas prêt EV-044 EV-089.

10.3 Pipeline de voix

Figure 3 — Pipeline de voix réel (cascade avec écoute continue — pas de full-duplex natif)



Mesuré dans logs/mike.log (78 tours, 20 sessions, 2026-09-29→10-02 ; chaîne composée dérivée des marques) :

fin de tour (endpoint)	ASR (transcription anticipée)	1er texte (tours cerveau)	1er audio TTS Kyutai	fin de parole → 1er son (composé)
p50 504 ms · p95 804 ms	p50 0,1 ms · p95 457 ms	p50 1 711 ms · p95 4 254 ms	p50 940 ms · p95 1 254 ms	p50 2 919 ms · p95 5 827 ms

Cible documentée (mike-voice-bench) : P50 ≤ 1,0 s · P95 ≤ 1,5 s — non atteinte. Délai réseau/PSTN non inclus.

Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

Transport et codecs

- **Twilio Media Streams** : WebSocket sécurisé (route `/v1/voice/stream`, EV-015), messages JSON dont la charge utile est du **µ-law 8 kHz en base64**, par trames de **160 octets (20 ms)** ; événements `connected`, `start`, `media`, `dtmf`, `mark`, `stop`. La validité du base64 est vérifiée (`validate=True`) ; une trame illisible est ignorée sans interrompre l'appel EV-001.
- **Conversion entrante** (`core/voice/mike_ears.py`) : µ-law → PCM16 → float32, puis rééchantillonnage 8 → 16 kHz par un filtre passe-bas **FIR à 63 coefficients** (`_lowpass_kernel`, `resample_pcm16_8k_to_float32_16k`). `audioop` est utilisé s'il est présent, avec une implémentation Python de repli.
- **Conversion sortante** : la Kyutai produit du PCM float à **24 kHz** ; `MulawStreamer` le convertit en µ-law 8 kHz et le découpe en trames de 160 octets (`FRAME_BYTES = 160`) ; `_media_out` les encapsule pour Twilio.
- **Tampons et contre-pression** : files bornées **à perte du plus ancien** (`BoundedDropOldestQueue` côté worker, `_put_drop_oldest` côté MIKE ; capacités `MAX_AUDIO_QUEUE = 500` et `MAX_EVENT_QUEUE = 200`). Une surcharge dégrade donc l'audio plutôt que de bloquer.
- **Reçus de lecture** : avant de raccrocher après une phrase finale, MIKE envoie une marque Twilio (`mark`) et attend son accusé ; sans accusé dans le délai, l'appel est clos en erreur (`_PlaybackReceiptTimeout`). Un compteur d'avancement de lecture (`_playback_advance`) alimente la décision « MIKE est-il audible ? ».

Détection d'activité vocale et fin de tour

- **VAD** : Silero (modèle sherpa-onnx) sur **CPU** par défaut ; **une instance par session** (l'état est interne au détecteur, partagé il mélangerait deux appels). Il déclare une fin de segment après **320 ms** de silence (bande validée 250–400 ms).
- **Smart Turn v3.2** (modèle de fin de tour) tranche à la fermeture d'un segment : *terminé* → le tour se clôt ; *incomplet* → le tour reste ouvert avec un délai de silence **raccourci à 0,8 s** (`MIKE_ASR_SILENCE_AFTER_REJECT_S`) ; si le VAD n'a jamais fermé de segment, le **filet de sécurité est de 2,4 s** (`MIKE_ASR_MAX_SILENCE_S`).
- **« Fin de tour sémantique »** : lorsque Smart Turn dit *incomplet* mais que le texte anticipé est une **réponse fermée courte** (≤ 6 mots, vocabulaire **codé en dur** FR/EN : *oui, non, d'accord, 9 h 30, lundi…*), le tour est clos par le texte. C'est une **heuristique lexicale**, pas un modèle appris EV-005.
- **Observé en exploitation** (78 tours depuis le 29 septembre) : 66 tours clos par Smart Turn (85 %), 12 par silence (15 %), 0 par le critère sémantique ; la détection de fin de tour prend **médiane 504 ms, 95e centile 804 ms** EV-005 EV-009.

Reconnaissance vocale

- **Whisper large-v3-turbo** exporté pour sherpa-onnx (`OfflineRecognizer`, décodage glouton, fp32, CUDA). **Aucun streaming** : l'audio du tour est accumulé puis transcrit en une fois EV-003.
- **Transcription anticipée** (*eager end-of-turn*) : dès 0,2 s de silence, un instantané de l'audio est envoyé à un pool de **2 threads partagé par toutes les sessions** ; si la parole reprend, le résultat est jeté. Le texte final est déjà prêt dans **78 tours sur 78** depuis le 29 septembre, d'où une durée d'ASR au moment de la clôture de **médiane 0,1 ms, 95e centile 457 ms** EV-004.
- **La langue est fixée à la construction du moteur de reconnaissance** (le changement d'option par flux est accepté puis ignoré, mesuré par l'équipe). Un seul moteur (~3,3 Go de VRAM) reste résident ; changer de langue entre appels recharge le modèle (~30 s selon les commentaires du code, **non mesuré ici**) ; le worker préchauffe la langue de la session dès l'ouverture pour éviter qu'un chargement bloque l'ouïe en plein appel EV-003.
- **Garde anti-hallucination** (`core/voice/mike_asr_guard.py`) : rejet des répétitions, des légendes de formation et des textes à rapport de compression > 2,4 (critère de la communauté OpenAI Whisper).
- **Exactitude (source : document, non re-mesuré)** : sur un banc interne de 40 énoncés de locuteurs québécois, Whisper donne 15/40 exacts (propre) et 14/40 (qualité téléphonique) ; 37/40 en anglais ; sur un énoncé sans parole, il a produit un texte dans 2/10 cas (`docs/delivery/PRODUCT_READINESS.md`). L'exactitude de l'ASR en français québécois par téléphone est donc la **première incertitude de recherche** (§25).

Synthèse vocale

- **Kyutai TTS 1.6B en_fr** (moshi/mimi, `n_q = 32`, température 0,6, voix de référence locales à licence ouverte) génère par pas de 80 ms et diffuse les trames en continu ; **une seule synthèse à la fois** (`threading.Lock` autour de `generate`) EV-006. La première trame audible arrive **médiane 940 ms, 95e centile 1 254 ms** après la demande dans l'exploitation récente (contre médiane 226 ms sur l'ensemble de l'historique, qui incluait Kokoro) EV-009.
- **Kokoro** reste disponible comme repli *explicite* (`MIKE_TTS_BACKEND=kokoro`) ; une valeur inconnue refuse le démarrage.
- **Découpage sémantique** du texte entrant en phrases (`SemanticSpeechChunker`) avec minuterie de vidange, normalisation du texte pour la synthèse, et reconstitution du texte *réellement prononcé* (`delivered_text`) pour la transcription scellée.

- **Verrou de capture CUDA** (`mike_gpu_capture`) : la Kyutai enregistre un *CUDA graph* à chaque phrase ; en mode global de PyTorch, un décodage Whisper concurrent déclenchait `cudaErrorStreamCaptureUnsupported` et tuait le TTS jusqu'au redémarrage. Le verrou commun a réduit les échecs à 0/30 en banc d'essai *(source : `docs/arquitectura/hardware.md` §10.3.1, non re-mesuré)*, avec le test unitaire `tests/unit/test_voice_gpu_capture.py` EV-007.

10.4 Interruptions, silences et reprises

- **Barge-in** : sur `speech_started`, si le plan d'arrêt est actif (`MIKE_VOICE_BARGE_VOICE_S` = 0,6 s) et que MIKE est audible, la décision est **différée** : elle attend `speech_sustained` (parole soutenue) ou le texte. Un acquiescement bref (« oui », « mm ») est ignoré ; une courte parole à contenu (« non », « attendez ») coupe immédiatement ; au-delà, l'audio en cours est vidé (`clear`), le TTS annulé et le tour en réponse annulé EV-002.
- **Faux positifs du VAD** : un son sans texte qui interrompt MIKE est journalisé (« barge-in sans parole ») pour mesurer le taux réel ; le chiffre n'a **pas** été extrait lors de cet audit (**NON MESURÉ**).
- **Silence du client** : après `MIKE_VOICE_IDLE_S` (8 s) sans son, MIKE dit « Vous êtes toujours là ? » puis répète la question ouverte, au plus deux fois, avant l'adieu.
- **Tour vide** : si l'ouïe a entendu au moins `REPEAT_MIN_SPEECH_MS` (600 ms) de parole mais n'a pas de texte, MIKE demande de répéter (au plus 2 demandes consécutives).
- **Touches DTMF** : une touche est un tour à part entière et coupe la lecture en cours.

10.5 Chaîne de latence mesurée

Les temps ci-dessous proviennent des **journaux réels** (`logs/mike.log`, traces `voice.turn.trace`), pas d'un banc synthétique. Ils agrègent des appels réels et des sondes de répétition sans téléphone ; la séparation n'est pas enregistrée dans le journal EV-009 EV-011.

Étape (depuis le 29 septembre ; 78 tours / 20 sessions)	Médiane	95e centile	Remarque
Détection de fin de tour (dernière trame parlée → décision)	504 ms	804 ms	Smart Turn ou silence
ASR à la clôture (texte anticipé)	0,1 ms	457 ms	78/78 anticipés
Premier texte — tours avec modèle (n = 44)	1 711 ms	4 254 ms	TTFT du cerveau dans le contexte réel
Premier texte — tours déterministes (n = 34)	8 ms	521 ms	analyseur d'agenda sans modèle
Premier audio TTS (Kyutai)	940 ms	1 254 ms	1re trame audible
Fin de parole → premier son (composé) — tous les tours	**2 919 ms**	**5 827 ms**	endpoint + transcript→socket
— tours déterministes	1 441 ms	2 951 ms	
— tours avec modèle	3 482 ms	6 202 ms	

Calcul : composé = `endpoint_detection_ms` (worker) + `twilio_first_media` (MIKE, mesuré depuis le texte final). Il **n'inclut pas** le délai réseau/PSTN ni la mémoire tampon de gigue du combiné. 23 tours sur 78 dépassent 3 s et 10 dépassent 5 s. La cible de conception documentée (médiane ≤ 1,0 s, 95e centile ≤ 1,5 s) n'est **pas atteinte** EV-009 EV-010.

10.6 Appels simultanés

Trois ressources partagées sérialisent le travail entre appels : (i) **un seul créneau LLM** avec file d'urgence non préemptive ; (ii) **une seule synthèse Kyutai à la fois** ; (iii) un **seul moteur Whisper** résident, décodé sous le verrou GPU, avec un pool de 2 threads de transcription anticipée EV-006 EV-012. Le code n'oppose aucune limite au nombre de sessions (`DuplexEngine.start_session`, route du worker `/v1/audio`) : un troisième appel serait accepté puis dégradé. La capacité réelle est **NON MESURÉE** ; la seule mesure antérieure à deux sessions simultanées est périmée (autre moteur TTS) et n'est pas reprise EV-008.

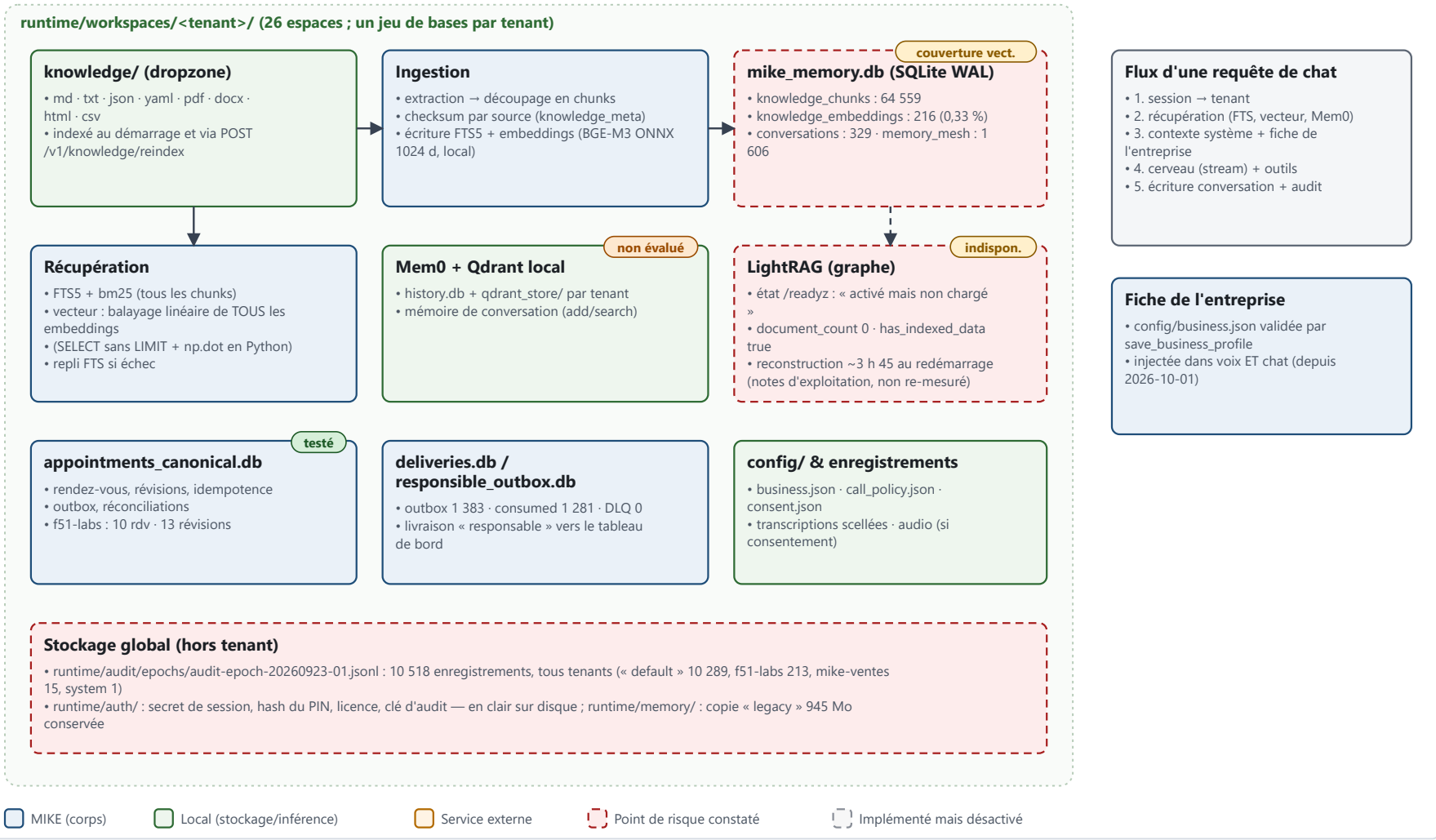
10.7 États des composants vocaux

Composant	État	Preuve
Transport Twilio Media Streams (entrant et sortant)	**IMPLÉMENTÉ ET TESTÉ** (appels réels)	EV-083
VAD, Smart Turn, fin de tour	**IMPLÉMENTÉ ET TESTÉ** ; paramètres validés par mesure interne	EV-005
ASR Whisper	**PARTIEL** : fonctionnel, non-streaming, exactitude québécoise limitée	EV-003

Composant	État	Preuve
Barge-in	**IMPLÉMENTÉ ET TESTÉ** (unitaire) ; **non éprouvé** sur la durée en appel réel	EV-002
TTS Kyutai	**PARTIEL** : qualité retenue par le propriétaire, premier son lent, une synthèse à la fois	EV-006
Garde de promesses parlées	**IMPLÉMENTÉ ET TESTÉ**	EV-013
Détection de répondeur (AMD)	**PARTIEL** : option synchrone du fournisseur (désactivée par défaut), lue seulement par l'IVR de rappel ; **absente de la voie duplex** conversationnelle	EV-016
Transfert vers un humain	**ABSENT**	EV-017
Full-duplex natif	**ABSENT** (cascade)	EV-014
Appels simultanés	**NON MESURÉ**	EV-008

11. Données et mémoire

Figure 4 — Flux de données et mémoire (tenant « default » mesuré : base mémoire 949 Mo, espace 1,6 Go)



Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

11.1 Inventaire des bases

Le relevé (`inventory.py` , ouverture `mode=ro`) trouve **36 fichiers SQLite** dans `runtime/` (hors caches). Les plus significatifs EV-070 :

Fichier (sous `runtime/`)	Taille	Tables	Contenu essentiel
`workspaces/default/memory/mike_memory.db`	**949,3 Mo**	24	64 559 segments de connaissance ; 329 conversations ; 1 606 liens `memory_mesh` ; 364 entrées de cache d'embeddings
`memory/mike_memory.db` (copie *legacy*)	945,1 Mo	24	Copie adoptée au démarrage, conservée sur disque
`workspaces/f51-labs/memory/mike_memory.db`	2,1 Mo	23	174 segments, tous vectorisés
`workspaces/default/memory/deliveries.db`	2,0 Mo	6	*outbox* 1 383, évènements consommés 1 281, livraisons 18, comptage de mesure 11, files de rebut (DLQ) 0
`workspaces/f51-labs/memory/appointments_canonical.db`	0,08 Mo	6	10 rendez-vous, 13 révisions, 13 entrées d'*outbox*, 10 clés d'idempotence
`workspaces/*/memory/responsible_outbox.db`	≈ 0,02 Mo	1	Livraison « responsable » vers le tableau de bord
`workspaces/*/memory/mem0/`	< 1 Mo	2 + Qdrant	Mémoire de conversation Mem0 (historique + stockage Qdrant local)
`memory/mem0/backups/mem0_reset_*`	≈ 10 Mo chacun	—	Deux sauvegardes de réinitialisation après changement de dimension d'embeddings (1 024 ↔ 384)
`audit/epochs/audit-epoch-20260923-01.jsonl`	8,0 Mo	—	Piste d'audit vivante, 10 518 enregistrements
`audit/audit.jsonl`	33,0 Mo	—	Piste d'audit historique, chaîne rompue (§16)

Vingt-six espaces de travail existent : `default` , `f51-labs` , `mike-ventes` , `a` , `clinic_b` et vingt-et-un espaces `qc_*` de contenu apparemment synthétique, non classifiés EV-070. Leurs `workspace.json` portent un champ `pin` en clair que le code de `core/` ne lit pas EV-053.

11.2 Entités principales et relations

- `knowledge_chunks (id, source, title, kind, chunk_index, checksum, content)` ← `knowledge_fts` (FTS5, table externe) et `knowledge_embeddings (chunk_id PK, cache_key, model, dims, vector BLOB)` ; `knowledge_meta (source PK, checksum)` pour la détection de changement.
- `conversations` ← `conversations_fts` , `conversation_embeddings` ; `memory_mesh (source_id, target_id, relation, strength)` relie des éléments « similaires ».
- Agenda canonique : `appointments` , `appointment_revisions` , `origin_links` , `appt_idempotency` , `appt_outbox` , `reconciliations` — **séparé** de la mémoire, avec clés d'idempotence et révisions EV-074.
- Livraison : *outbox* transactionnelle (`outbox` , `consumed_events` , `outbox_dlq` , `consumer_dlq` , `deliveries` , `metering_events`) — schéma de type *transactional outbox* avec file de rebut EV-074.

11.3 Migrations, index, sauvegardes, réplication, chiffrement

Sujet	Constat
Migrations	Aucun mécanisme versionné : `PRAGMA user_version = 0` sur la base principale. Le magasin de mémoire détecte « si une migration est nécessaire » et fait une copie avant (`mike_memory_store.py` , ligne ~442) ; les changements de dimension d'embeddings ont été traités par **réinitialisation + sauvegarde** (deux répertoires `mem0_reset_*`). Le schéma des évènements a un champ `schema_version` avec rebut des versions inconnues EV-070
Index	18 index sur la base principale ; FTS5 pour le plein-texte
Espace libre	**71,2 % des pages de `mike_memory.db` sont libres** (172 914 de 243 009 pages de 4 Ko) : ≈ 675 Mo récupérables par `VACUUM` , ≈ 274 Mo de données utiles ; `auto_vacuum = 0` EV-071
Sauvegarde	La copie locale `runtime/backups/` a été retirée volontairement (même disque). Sauvegarde hors-site chiffrée (AES-256-GCM + signature Ed25519) **implémentée mais désactivée par défaut** (`MIKE_OFFSITE_BACKUP_ENABLED=false`) ; copie manuelle `sync_usb.ps1` vers une clé USB **non chiffrée** EV-046

Sujet	Constat
Réplication	Aucune
Chiffrement au repos	**Pas de chiffrement applicatif**, par décision documentée (`core/server/mike_encryption.py` : une clé stockée à côté des données n'apporte rien contre un voleur de machine, et SQLCipher laisserait en clair LightRAG, Mem0 et le dropzone). La protection retenue est le **chiffrement de volume BitLocker**, que le module *détecte et rapporte* sans jamais le présumer (`unknown` n'est pas « chiffré ») ; mode `MIKE_REQUIRE_DISK_ENCRYPTION=warn` ici. **Mesure du 2 octobre : BitLocker désactivé sur C: et sur la clé USB E:** (propriété de shell `System.Volume.BitLockerProtection` = 2 ; la confirmation par `manage-bde` exige des droits d'administrateur, refusés) ; aucune attestation d'opérateur. Fernet ne protège que les jetons du module de prix ; AES-256-GCM que la sauvegarde hors-site EV-046
Point de restauration	Aucun *point-in-time recovery* ; une base corrompue = perte du tenant, sauf copie manuelle

11.4 Conséquences d'échelle de SQLite

SQLite en mode WAL avec **un fichier par tenant** est un choix cohérent pour des PME : pas de contention inter-clients, déploiement simple. Les limites à l'échelle sont néanmoins concrètes :

1. **Un seul écrivain par base** : acceptable tant qu'un tenant reste à quelques appels simultanés ; non mesuré.
2. **Croissance du fichier** : le tenant `default` est passé à 949 Mo en environ deux mois, dont 71 % de pages libres jamais récupérées — l'espace n'est pas rendu à l'OS (`auto_vacuum` désactivé).
3. **Recherche vectorielle en balayage linéaire** (§12) : coût proportionnel au nombre de vecteurs ; avec 64 559 segments vectorisés en 1 024 dimensions flottantes, une requête lirait ≈ 260 Mo.
4. **Piste d'audit unique** (JSONL) sérialisée par un verrou de fichier : tous les tenants écrivent dans le même fichier ; l'écriture inclut un `fsync` par enregistrement EV-036.
5. **Sauvegarde/restauration** : copier une base WAL ouverte sans checkpoint peut donner une copie incohérente. La copie de données vers la clé USB réalisée le jour de l'audit s'est faite **pendant que MIKE tournait** ; la cohérence n'est pas garantie.

12. RAG et connaissance

12.1 Chaîne

Ingestion (`mike_memory_store.upsert_knowledge_file` , indexation au démarrage et via `POST /v1/knowledge/reindex`) : extraction (md, txt, json, jsonl, yaml, pdf, docx, html, csv) → découpage en segments (≈ 916 caractères en moyenne, max 1 200) → écriture `knowledge_chunks` + FTS5 + vecteur BGE-M3 ONNX local (1 024 dimensions, modèle `BAAI/bge-m3`). La détection de changement utilise une somme de contrôle par source EV-071.

Récupération : `search_knowledge` = FTS5 avec classement `bm25` ; `search_knowledge_vector` = *tous* les embeddings du modèle courant sont lus, désérialisés et comparés au vecteur de requête par produit scalaire en Python, puis triés ; en cas d'échec ou de vecteur dégénéré, repli sur FTS (`MIKE_VECTOR_FALLBACK_TO_FTS`) EV-072. Un *reranker* existe (`mike_reranker.py`) mais est **désactivé par défaut**. Mem0 fournit une mémoire de conversation distincte ; LightRAG fournit une recherche par graphe.

12.2 État mesuré

Indicateur (tenant `default`)	Valeur	Lecture
Segments de connaissance	64 559	
… dont textes de romans du domaine public (`public_domain_fused/books`)	**62 407 (96,7 %)**	corpus de test/culture générale, **non vectorisé**
… hors livres (`drive_docs` 1 316, `harvested` 455, `learnings` 348, `current_docs` 23, autres 10)	2 152	contenu « métier »
Segments vectorisés	**216 (0,33 % du total ; 10,0 % du contenu hors livres)**	tous hors livres
Segments vectorisés du tenant `f51-labs`	174 / 174 (100 %)	
Conversations mémorisées	329	

Indicateur (tenant `default`)	Valeur	Lecture
Statut LightRAG (`/readyz`)	**« activé mais non chargé »**, `document_count 0`, `has_indexed_data true`	capacité `knowledge` : *unavailable*, non bloquante

Interprétation prudente : la faible couverture vectorielle du tenant `default` s'explique en grande partie par la présence de 62 407 segments de romans indexés seulement en plein-texte. **Ce qui est établi** : un tenant client « propre » (`f51-labs`) est entièrement vectorisé ; le balayage linéaire est donc aujourd'hui peu coûteux. **Ce qui ne l'est pas** : la qualité de récupération (rappel, précision), l'effet des romans sur le classement `bm25` du tenant `default`, et la pertinence du moteur de graphe, indisponible. Aucun jeu d'évaluation de récupération n'a été trouvé (**NON MESURÉ**).

12.3 Isolation et contamination entre clients

Chaque tenant possède sa propre base `mike_memory.db`, son propre `mem0/` et son propre dropzone `knowledge/` ; la fabrique de services met en cache un runtime **par tenant** avec un plafond LRU (`MIKE_MAX_TENANT_RUNTIMES`). Les suites de tests contiennent des cas d'isolation (graphe, mémoire, agenda, cache de requêtes d'embeddings : fichiers `test_graph_tenant_isolation.py`, `test_tenant_isolation.py`, `test_query_embed_cache_tenant_scope.py` …). **Risque résiduel** : le repli `current_tenant() or default_tenant()` (51 occurrences) route une requête sans contexte vers le tenant `default` — qui contient 1,6 Go de données réelles — et c'est ce motif qui a produit une fuite mesurée le 1er octobre 2026 selon le commentaire du code EV-041.

13. Outils et MCP

13.1 Inventaire

Le manifeste `config/mcp_servers.json` déclare **20 serveurs** (18 actifs, 2 désactivés : `fetch`, `brave-search`). S'y ajoutent le **serveur d'outils par défaut** `core/mcp/mike_workspace_mcp.py` (13 outils : fichiers et commande) et le pont `mike_mcp_bridge.py` (3 outils) exposé à Claude Code par `.mcp.json`. Le code Python définit **117 outils** (`@mcp.tool`) répartis en 15 fichiers EV-086.

Groupe	Serveurs / fichiers (outils)	Accès	Finalité	Risque principal	État
Fichiers et commande	`mike_workspace_mcp` (13)	propriétaire	lire/écrire/déplacer/supprimer dans les racines autorisées ; `run_command` (délai ≤ 300 s)	exécution de commande, suppression	IMPLÉMENTÉ ET TESTÉ (confinement `resolve_safe_path`, kill-switch) EV-049
Agenda et rendez-vous	`agenda` (4), `appointments` (10), `calendar` (7)	propriétaire	agenda canonique, disponibilités, Google Calendar	mutation d'agenda	IMPLÉMENTÉ ET TESTÉ (canonique) ; Google : NON VALIDÉ
Documents et données	`drive` (4), `excel` (7), `docgen` (2)	propriétaire	Drive, tableurs, génération de documents	lecture de fichiers sensibles	IMPLÉMENTÉ MAIS NON VALIDÉ
Marketing et e-commerce	`mike_ga4` (14), `google-ads` (11), `shopify` (14)	propriétaire	analytique, publicité, boutique	actions sur comptes tiers	IMPLÉMENTÉ MAIS NON VALIDÉ
Veille et web	`price` (14), `scrape` (4), `puppeteer` (npx), `crawlconsole`, `huggingface` (6)	propriétaire / ops	prix de détaillants, scraping, navigateur	contenu non fiable → injection ; respect des conditions de sites	PARTIEL / NON VALIDÉ
Mémoire et raisonnement	`memory-persistent`, `sequential-thinking` (npx)	tous / tous	mémoire inter-sessions, raisonnement pas à pas	duplication de la mémoire native	IMPLÉMENTÉ MAIS NON VALIDÉ
Développement	`github` (npx), `remote-agent` (4)	propriétaire	dépôt, agents distants	exfiltration de code	IMPLÉMENTÉ MAIS NON VALIDÉ

13.2 Appel, permissions et traitement d'erreur

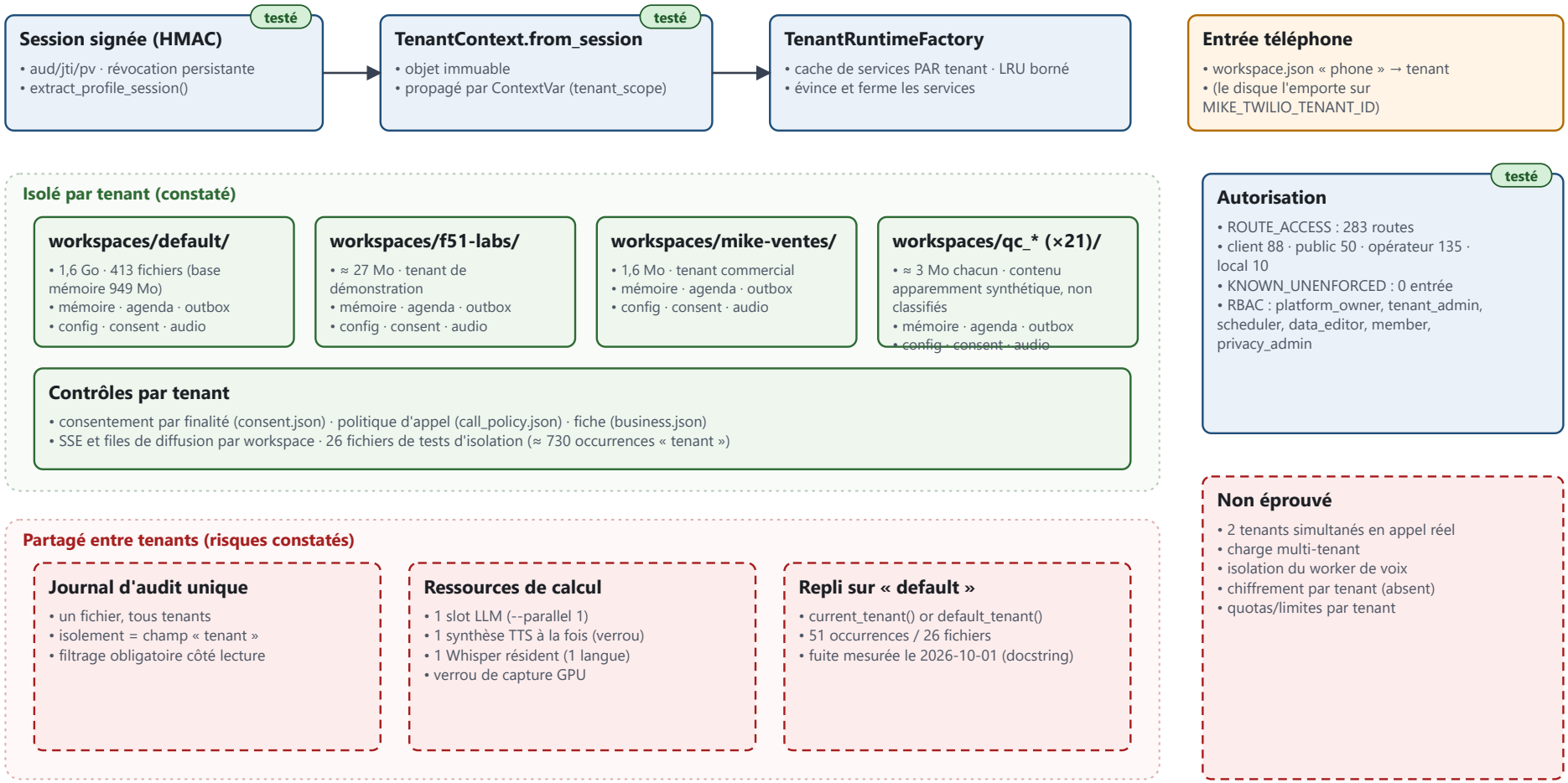
- **Appel** : `MCPClient.call_tool(raw_name, arguments, timeout=90 s)` ; les noms sont préfixés par serveur (`tool_prefix`). La boucle de chat borne les enchaînements à **8 pas** (`MIKE_MCP_MAX_STEPS`).
- **Permissions** : chaque serveur déclare `access` (`any`, `owner`, `ops`) ; la valeur par défaut est `owner`. Seuls `sequential-thinking`, `calendar`, `drive` et `puppeteer` déclarent `tenant_safe: true` — les autres ne portent **pas** d'attestation

d'isolation par tenant EV-086.

- **Consentement** : les outils de serveurs externes exigent le consentement `external_connectors` du tenant. **Sur ce poste, ce contrôle est désactivé** (`MIKE_CONSENT_ENFORCED=0`) EV-048.
- **Erreur** : un serveur indisponible renvoie une erreur réelle ; le kill-switch bloque les mutations. Un format de retour d'erreur homogène entre les 20 serveurs n'a **pas** été vérifié.

14. Multi-tenant

Figure 6 — Architecture multi-tenant : où l'isolation est réelle, où elle est partagée



Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

14.1 Identification du tenant

Canal	Mécanisme	Preuve
Web (chat, tableau de bord)	Session HMAC-SHA-256 (`aud` , `jti` ; `pv` et révocation persistante selon la documentation) → `TenantContext.from_session(session)` ; sans session ou sans tenant : 403	EV-045 EV-040
Téléphone entrant	Numéro appelé → tenant lu dans `workspace.json` (clé `phone`) ; deux espaces déclarant le même numéro = **aucun ne répond** (ambiguïté refusée)	`scripts/ops/tenant_line.py` , test `test_tenant_comercial_mike.py` EV-090
WebSocket de voix	Jeton HMAC dans `customParameters` du `start` , lié au tenant et au `CallSid`	EV-044
Tâches de fond	`tenant_scope(ctx)` explicite par tâche	EV-040

Le contexte est **immuable** et propagé par `contextvars` , donc traverse les `await` . Le mode strict (`MIKE_MULTI_TENANT=1` , **actif par défaut**) lève `NoTenantContext` si un service sensible est lu sans contexte EV-040 .

14.2 Ce qui est isolé

Données (bases SQLite, Mem0/Qdrant, dropzone), configuration (`business.json` , `call_policy.json` , `consent.json`), SSE et files de diffusion (par workspace), mémoire téléphonique, enregistrements audio et transcriptions, et jeu d'outils (`test_tenant_tool_isolation.py`). Vingt-six fichiers de tests portent sur l'isolation (≈ 730 occurrences du mot « tenant »).

14.3 Ce qui est partagé — risques de fuite croisée

Ressource partagée	Risque	Gravité estimée
Piste d'audit unique (tous tenants dans un fichier)	Une lecture non filtrée par tenant exposerait les évènements d'un autre client ; l'isolement repose sur le champ `tenant` et le filtrage côté lecture	Moyenne
Repli sur `default` (51 occurrences dans 26 fichiers)	Une requête sans contexte reçoit la mémoire de l'entreprise `default` (1,6 Go) ; fuite mesurée en 2026-10-01	**Élevée** *(classification des 51 occurrences non faite)*
Créneau LLM, TTS, Whisper	Un appel long d'un tenant retarde les autres (pas d'équité par tenant) ; pas une fuite de données mais une fuite de **performance**	Moyenne
Worker de voix	Un seul process pour tous les appels ; un VAD par session (état séparé) mais Whisper/Kyutai partagés	Moyenne
Cache d'embeddings de requête	Test de portée par tenant existant (`test_query_embed_cache_tenant_scope.py`)	Faible

14.4 Contrôle d'accès

`ROUTE_ACCESS` déclare le **rôle minimal** de 283 routes (client 88 · public 50 · opérateur 135 · local 10). Un test balaie les routes enregistrées et **échoue si une route n'est pas déclarée** ; la liste d'exceptions `KNOWN_UNENFORCED` est **vide**. Par-dessus, la matrice RBAC définit six rôles (`platform_owner` , `tenant_admin` , `scheduler` , `data_editor` , `member` , `privacy_admin`) et des surcharges route × rôle × tenant EV-042 . Les dix « trous » H1–H10 de l'audit du 20 août 2026 ont été fermés pour H1–H4 ; H5–H10 sont listés comme ouverts dans la décision `2026-08-20-fronteira-cliente-S1-S3.md` *(source : document, non re-vérifié)*.

14.5 Maturité

Niveau 3 (fonctionnel). Les mécanismes de frontière sont nombreux, testés et vérifiés mécaniquement ; mais l'isolation n'a **pas été exercée sous charge concurrente multi-tenant** (deux appels réels simultanés sur deux tenants), le repli sur `default` n'est pas éliminé, et il n'y a ni chiffrement par tenant ni quotas par tenant.

15. Sécurité et garde-fous de conformité

15.1 Authentification et sessions

Mécanisme	Constat	Preuve
Sessions	Jetons signés **HMAC-SHA-256** avec `aud` (`mike:profile`, `mike:service`) et `jti` ; `pv` et la révocation persistante selon la documentation	`core/server/mike_auth.py` EV-045
Mots de passe de profil	**PBKDF2-HMAC-SHA-256, 600 000 itérations** par défaut (plancher 50 000), sel aléatoire par utilisateur ; une dérivation « leurre » égalise le temps quand l'identifiant n'existe pas	`PBKDF2_ITERATIONS` ; `_burn_password_work` EV-045
PIN d'opérateur	Hash sur disque (`runtime/auth/operator_pin.hash`) ; vérification à temps constant ; authentification renforcée par opération (`verify_stepup`)	`routers/auth.py`, `mike_operator_pin.py`
Protection contre la force brute	Limitation de débit et verrouillage sur `setup` et `unlock`	`_check_setup_rate_limit`, `_check_unlock_rate_limit`
Mise en place d'opérateur	Jeton unique d'amorçage requis (`scripts/ops/mike_bootstrap_provision.py`)	`docs/arquitectura/identidade-rbac.md` *(source : document)*
Confiance implicite en localhost	**Active sur ce poste** (`MIKE_TRUST_LOCALHOST=true`) ; documentée comme désactivée en production	EV-048

15.2 Autorisation et segmentation

Le contrôle d'accès est **déclaratif et vérifié mécaniquement** : 283 routes avec un rôle minimal, 0 exception, plus une matrice RBAC à six rôles EV-042. Un middleware de passerelle d'opérateur (`OperatorGateMiddleware`) et un middleware de vidange (`DrainMiddleware`, `/readyz` passe à 503 à l'arrêt) complètent la pile ; `SlowAPIMiddleware` assure une limitation de débit globale. Les en-têtes de sécurité sont posés par le serveur : `X-Frame-Options: DENY`, `Content-Security-Policy`, et `Strict-Transport-Security` lorsque la requête arrive en HTTPS EV-051.

15.3 Exposition réseau

Les trois processus n'écoutent que sur **127.0.0.1** (8081, 8083, 8998 via `wslrelay`) EV-025. L'entrée publique est exclusivement le tunnel sortant `cloudflared` ; les webhooks Twilio sont authentifiés par **signature** (`X-Twilio-Signature`) sur 5 points d'entrée, avec un contournement local explicite seulement EV-043. Le WebSocket de voix exige un jeton HMAC dans les paramètres `start` EV-044. La documentation interdit d'exposer le port 8083 sans HTTPS ni tunnel authentifié ; l'accès LAN exige `MIKE_HOST=0.0.0.0`. *Observation hors périmètre* : un processus nommé « Manager » (PID 6040) écoute sur toutes les interfaces au port 8000 ; il n'appartient pas à MIKE et n'a pas été analysé.

15.4 Gestion des secrets

Aspect	Constat
Stockage	Variables d'environnement lues depuis `config/.env.runtime` et `config/.env.runtime.local` ; matériel cryptographique en fichiers dans `runtime/auth/` (`session_secret`, `audit_key`, `credentials.json`, `operator_pin.hash`, licence) — **en clair sur le disque** EV-046
Contrôle d'accès disque	Les ACL NTFS du fichier `.env.runtime.local` et du dossier `runtime/auth` accordent **lecture au groupe local** `CodexSandboxUsers` et **modification à deux SID non résolus**, en plus du propriétaire, des administrateurs et de SYSTEM EV-047
Dépôt Git	**Aucun secret suivi** : la recherche de noms (`.env*`, `profiles.yaml`, `identity.json`, jetons, clés `.pem`) dans `git ls-files` ne renvoie rien ; `.gitignore` couvre `.env.*` et `runtime/audit/` ; un `secret_scan.py` s'exécute à chaque commit EV-052
Rotation	Procédure de clé de signature par **époque d'audit** ; pas de coffre-fort de secrets ni de rotation automatisée
Existence d'un mécanisme	**Oui, minimal** (fichiers + ACL + scan) ; **non** pour un coffre dédié (DPAPI, Credential Manager, HSM)

15.5 Chiffrement

En transit : TLS de bout en bout entre Twilio/Cloudflare et le tunnel ; HTTP clair sur boucle locale entre les processus (acceptable sur un hôte unique, sans protection contre un processus local malveillant). **Au repos** : le code **ne chiffre pas** les bases, la piste d'audit ni `runtime/auth` — choix documenté de s'en remettre au chiffrement de volume, que `mike_encryption.py` détecte et rapporte (mode `warn`) sans jamais le présumer. **Mesure : BitLocker désactivé sur C: et sur la**

clé USB (`System.Volume.BitLockerProtection` = 2). Le chiffrement applicatif n'existe que pour la sauvegarde hors-site (AES-256-GCM, PBKDF2 600 000, signature Ed25519 — désactivée par défaut) et le magasin de jetons de détaillants (Fernet) EV-046. La clé USB de sauvegarde créée le jour de l'audit contient donc ces fichiers **en clair**.

15.6 Validation des entrées, exécution d'outils, injection

- **Validation** : modèles FastAPI/pydantic aux frontières HTTP ; validation stricte et codes d'erreur par champ sur la fiche de l'entreprise ; trames de contrôle du worker bornées (`MAX_CONTROL_BYTES = 16 384` , texte ≤ 4 000 caractères). Une analyse systématique de toutes les routes n'a pas été faite.
- **Exécution d'outils** : `resolve_safe_path` confine toute opération de fichier aux racines autorisées ; `run_command` vérifie le kill-switch, confine le répertoire de travail, valide la commande contre les racines et borne le délai (1 à 300 s) EV-049. **Il n'existe pas de bac à sable au niveau du système** (conteneur, jeton restreint) : le confinement est applicatif.
- **Injection de prompt** : les défenses constatées sont (i) un texte de garde dans le prompt de voix (`_ANTI_INJECTION_GUARD` , FR/EN), (ii) la restriction des outils sensibles au propriétaire, (iii) le filtre de promesses sur la sortie parlée et (iv) le kill-switch. **Aucun marquage de provenance ni assainissement du contenu issu d'outils** (pages web, scraping, courriels) avant qu'il entre dans le contexte du modèle n'a été trouvé EV-050. C'est la principale lacune de sécurité propre à l'IA agentique de ce système.
- **Abus d'outils** : mitigé par le RBAC, le consentement `external_connectors` (désactivé sur ce poste), l'authentification renforcée et le kill-switch ; non éprouvé par des tests adverses.

15.7 Journaux et données personnelles

Les numéros de téléphone sont masqués dans les journaux d'appel (`_redact_phone`) et les lignes de journal examinées du worker ne contiennent que des métadonnées (le texte des conversations en est exclu). La piste d'audit stocke des **pseudonymes** (HMAC à clé séparée) plutôt que des identifiants en clair, et IP et *user-agent* y sont pseudonymisés *(source : `docs/delivery/evidence/2026-09-30-privacy-backup-production.md` , non re-vérifié)*. Les transcriptions d'appels et l'audio sont stockés par tenant, non chiffrés ; l'enregistrement audio exige un interrupteur *et* le consentement `telephony_recording` ainsi qu'un avis dit à l'ouverture.

15.8 Garde-fous de conformité (mécanismes techniques uniquement)

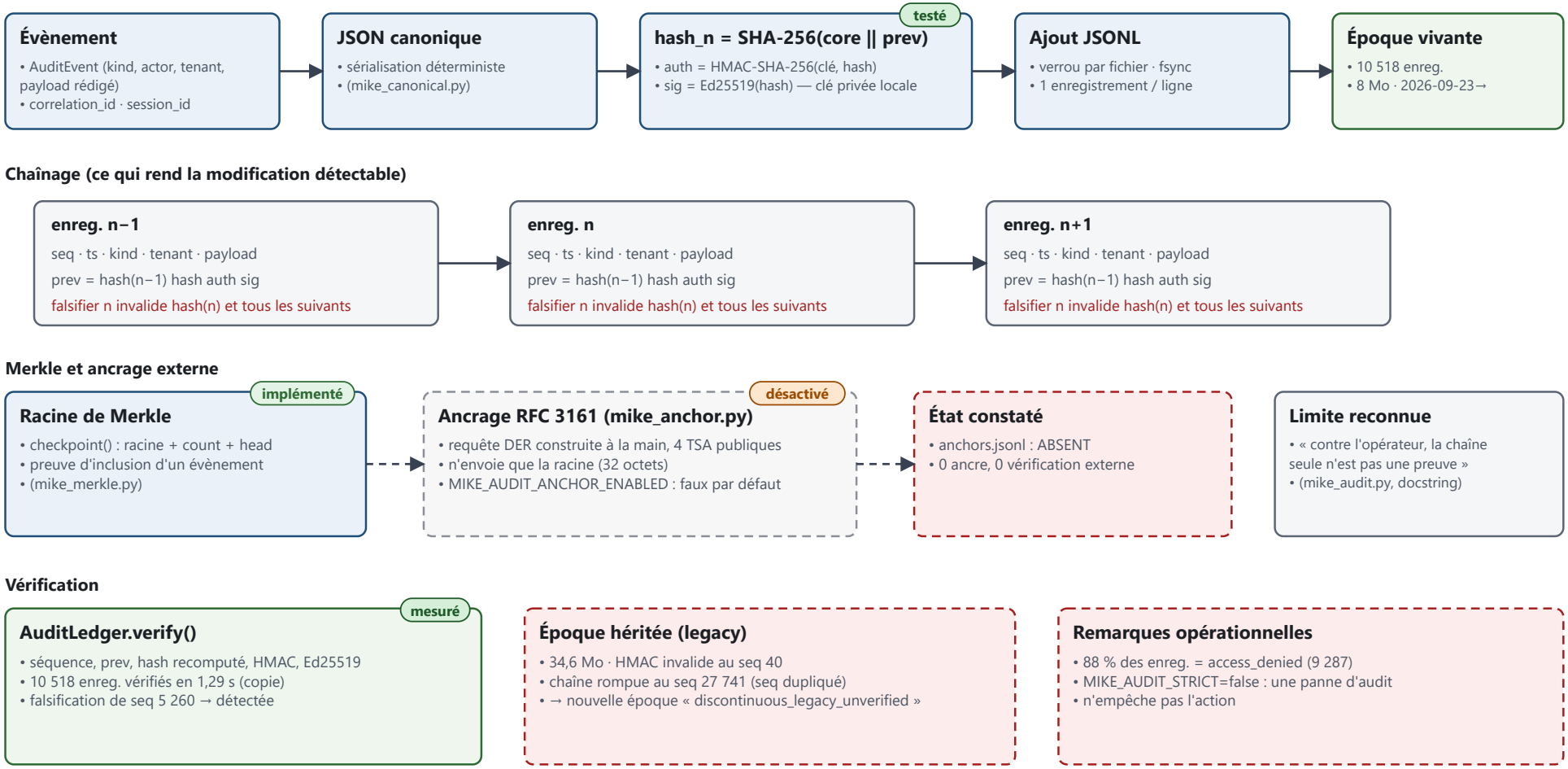
Cette section décrit des mécanismes implémentés ; elle ne constitue **pas** un avis juridique.

Garde-fou	Mécanisme	État
Consentement par finalité	6 finalités : `essential` , `memory_retention` , `rag_indexing` , `external_connectors` , `autonomy_outbound` , `telephony_recording` ; l'absence d'enregistrement vaut refus ; un fichier illisible n'est jamais un « oui » ; la piste survit à l'effacement (`mike_privacy.py`)	Implémenté et testé ; **application désactivée sur ce poste** EV-062 EV-048
Droit à l'effacement et à la rétention	Effacement sélectif (SQLite + `VACUUM` , FTS/vecteurs orphelins, graphe marqué à reconstruire), balayage quotidien de rétention (`mike_erasure.py` , `mike_retention.py`)	Implémenté ; **rétention désactivée ici** EV-048
Avis de décision automatisée (art. 12.1)	Enregistrés *avant* livraison ; ce qui n'est pas arrivé reste `pending` et visible (`mike_decision_notice.py`)	Implémenté ; 7 évènements dans la piste EV-030
Pré-vol d'appel sortant P1–P7	P1 consentement ; P2 liste d'exclusion ; P3 fenêtre horaire CRTC (09:00–21:30, heure locale du destinataire, valeur par défaut lue dans la politique d'exemple) ; P4 tentatives par destinataire (30 jours) ; P5 santé du DID ; P6 politique de répondeur ; P7 attestation STIR/SHAKEN (consultatif)	Implémenté et testé EV-060
Liste de non-appel	**Locale au tenant seulement** (`dncl.json`) ; la Liste nationale du CRTC n'est **pas** consultée	**PARTIEL / ABSENT pour la liste officielle** EV-061
Détection de répondeur automatique	P6 vérifie qu'une **politique** (`voicemail: allowed|denied`) est déclarée. Côté fournisseur, `MIKE_TWILIO_MACHINE_DETECTION=sync` active la détection Twilio (désactivée par défaut : elle bloque l'appel ~2,9 s avant la salutation, mesure du code du 15 septembre) ; seul l'IVR `/twilio/voice/appointment` lit `AnsweredBy` . La **route de dialogue duplex ne le lit jamais**	**PARTIEL** EV-016
Prise en main humaine	Aucun transfert d'appel ; la garde interdit de promettre un transfert (`live_transfer`)	**ABSENT** EV-017
Kill-switch	Fail-closed ; gèle commande, fichiers, gouvernance, réponse automatique, TTS	Implémenté et testé EV-089

Garde-fou	Mécanisme	État
Approbation de contenu	Les messages sortants passent par une *outbox* « responsable » avec nom, Reply-To et signature du tenant ; pas de workflow d'approbation humaine préalable constaté	**PARTIEL**
Restrictions géographiques	Fuseau du destinataire déduit de l'indicatif pour la fenêtre horaire ; pas de blocage par pays/région	**PARTIEL**
Journal de consentement	Évènements de consentement inscrits dans la piste d'audit	Implémenté

16. Auditabilité

Figure 5 — Chaîne d'audit : ce qui est actif, ce qui ne l'est pas



Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

16.1 Fonctionnement exact

Chaque évènement (`AuditEvent` : type, acteur, tenant, charge rédigée, `correlation_id`, `session_id`) est sérialisé en **JSON canonique** déterministe (`mike_canonical.py`), puis :

```
hash_n = SHA-256( canonique(enregistrement_n sans hash/auth/sig) ‖ hash_{n-1} )
auth_n = HMAC-SHA-256( clé_d'autorité, hash_n )
sig_n  = Ed25519( clé_privée, hash_n )
```

L'enregistrement est ajouté en fin de fichier JSONL sous un **verrou par fichier** (partagé entre les deux noms d'import du module, pour éviter des `seq` dupliqués), suivi d'un `fsync`. Le `seq` et le hash précédent sont lus **depuis la dernière ligne** uniquement (selon un commentaire du code, une lecture complète à chaque ajout coûtait 1,6 s à 50 000 enregistrements). Un arbre de **Merkle** permet la preuve d'inclusion d'un évènement et un `checkpoint()` renvoie racine, nombre et tête EV-033.

16.2 Ce qui a été vérifié (mesure du 2 octobre 2026)

Contrôle	Résultat
Vérification de l'époque vivante sur copie : séquence, `prev`, hash recalculé, signature Ed25519 avec la clé **publique** seule	**10 518 enregistrements intègres, 10 518 signés**, 1,29 s EV-030
Falsification d'un champ à l'enregistrement 5 260	**Détectée** à `seq` 5 260 : « contenu altéré : le hash ne correspond pas à l'enregistrement » EV-031
Époque historique (`audit.jsonl`, 34,6 Mo)	**Rompue** : HMAC invalide à `seq` 40 ; chaîne rompue à `seq` 27 741 (« seq hors ordre : attendu 27 742, trouvé 27 741 »). Les causes plausibles sont des écritures concurrentes à deux verrous distincts, corrigées ensuite (commentaire du code) — **non démontré**. Le fichier a été conservé (SHA-256 consigné) et une époque neuve a été ouverte le 23 septembre EV-032
Répartition par type	`access_denied` 9 287 (88 %), `governance_action` 554, `voice_turn` 201, `gate_raised` 150, `request` 100, `response` 84, `voice_call_transcript` 29, `business_action` 22, `telephony_call_preflight` 21, `telephony_call_execution` 16 EV-036
Répartition par tenant	`default` 10 289 · `f51-labs` 213 · `mike-ventes` 15 · `system` 1 EV-036

16.3 Ce qui n'est pas actif ou pas prouvé

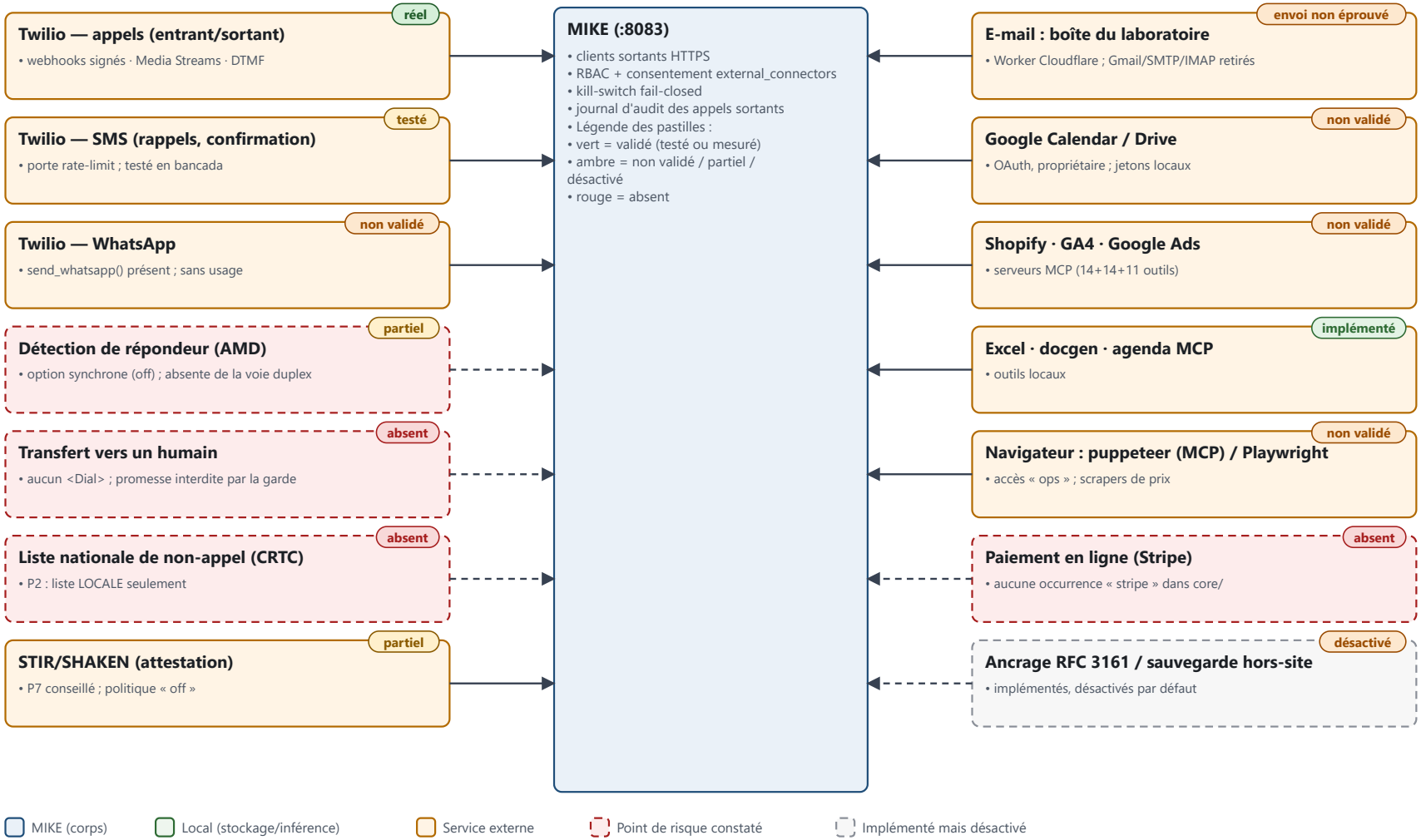
- **RFC 3161 (horodatage par tiers)** : le module construit à la main la requête DER, interroge quatre autorités publiques et ne transmet que la racine (32 octets). Mais (i) il est **désactivé par défaut** et ne l'est pas non plus dans l'environnement audité, (ii) `anchors.jsonl` **n'existe pas** — aucune ancre n'a jamais été obtenue ici, (iii) le code ne contrôle que le **statut** de la réponse (`PKIStatus` 0/1) et déclare lui-même que ce n'est « pas une vérification cryptographique » ; la validation du jeton est déléguée à OpenSSL hors du programme EV-034.
- **Limite reconnue par le code** : contre l'opérateur qui détient la machine et la clé, la chaîne seule n'est pas une preuve ; seule une ancre externe ferme cette faille. Elle n'est pas en service.
- **Horloge** : les horodatages sont ceux de l'horloge système (UTC, ISO 8601) ; aucune source de temps de confiance.
- **WORM / immuabilité du support** : aucune ; le fichier est modifiable par les comptes disposant de droits d'écriture (voir ACL, §15.4).
- `MIKE_AUDIT_STRICT=false` : une panne d'écriture d'audit n'arrête pas l'action EV-035.
- **Clé de signature** : stockée en variable d'environnement dans un fichier local ; sa valeur a été affichée par erreur dans la session d'audit (§5.4) → rotation recommandée.

16.4 Reconstruction d'un appel

Un appel peut être reconstruit à partir de : l'évènement `telephony_call_preflight` (décisions P1–P7), `telephony_call_execution`, la suite de `voice_turn`, la transcription scellée `voice_call_transcript` (SHA-256 du texte *réellement prononcé*, `delivered_text`), l'audio local si consenti, et le journal d'exploitation (`voice.turn.trace`) corrélé par `CallSid`. **Limite connue** : la transcription scellée perd la première moitié d'une phrase scindée en deux énoncés *(source : `PRODUCT_READINESS.md`, non re-mesuré)*.

17. Intégrations

Figure 7 — Intégrations externes et état de chacune



Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

Intégration	Classe	Détails et preuve
Twilio — appels entrants/sortants, DTMF	IMPLÉMENTÉ ET TESTÉ	Appels réels autorisés (dont un avec réservation écrite) ; webhooks signés EV-083 EV-043
Twilio — SMS (rappels, confirmation, réponses)	PARTIEL	Porte de limitation de débit, états de rendez-vous testés unitairement ; envois réels non exercés par cet audit
Twilio — WhatsApp	IMPLÉMENTÉ MAIS NON VALIDÉ	`send_whatsapp()` présent dans `mike_twilio.py` ; aucun usage constaté
Courriel	IMPLÉMENTÉ MAIS NON VALIDÉ	Boîte du laboratoire (Worker Cloudflare, `mike_mailbox_cloudflare.py` : liste, lecture, recherche, envoi) ; Gmail/SMTP/IMAP **retirés** ; lecture réelle éprouvée antérieurement, **envoi réel jamais exercé** *(source : document)*
Cloudflare Tunnel	IMPLÉMENTÉ ET TESTÉ	Processus `cloudflared` actif ; webhooks reçus
Google Calendar / Drive	IMPLÉMENTÉ MAIS NON VALIDÉ	OAuth propriétaire, jetons locaux ; consentement `external_connectors`
Excel, docgen, agenda MCP	IMPLÉMENTÉ (non exercé par l'audit)	Outils locaux (7 + 2 + 4 outils)
Shopify, GA4, Google Ads	IMPLÉMENTÉ MAIS NON VALIDÉ	14 + 14 + 11 outils MCP ; aucune exécution constatée
Navigateur / scraping (puppeteer MCP, Playwright des scrapers de prix)	IMPLÉMENTÉ MAIS NON VALIDÉ	`access: ops` ; contenu non fiable (voir §15.6)
HuggingFace, GitHub, mémoire persistante (MCP npx)	IMPLÉMENTÉ MAIS NON VALIDÉ	`huggingface` accessible à tous (`access: any`)
Ancrage RFC 3161 / sauvegarde hors-site	IMPLÉMENTÉ, DÉSACTIVÉ	§16 et §11
Pipedrive (CRM)	PLANIFIÉ / ABSENT	Variables `MIKE_PIPEDRIVE_*` et point d'extension dans `mike_appointments.py` ; **aucun module** `MikePipedrive`
Paiement en ligne (Stripe)	ABSENT	aucune occurrence dans `core/`
Autres CRM/ERP/prises de rendez-vous tiers	ABSENT	aucune intégration trouvée
Transfert humain, liste nationale de non-appel	ABSENT	§15.8
Détection de répondeur (voie conversationnelle)	PARTIEL	§15.8

18. Observabilité

Sujet	Constat
Journaux	Fichiers sous `logs/` (345 fichiers, ≈ 124 Mo) : `mike.log` cumulatif depuis le 23 juillet (14 Mo), `governance_journal.jsonl` (6 Mo), `heartbeat_history.jsonl`, journaux du cerveau et du worker EV-075
Rotation	**Aucune dans le chemin de lancement par défaut** (aucun `RotatingFileHandler` dans `core/`). Seule l'installation en service Windows (NSSM, `install_mike_service.ps1`) active une rotation à 50 Mo
Métriques	`mike_latency.py` : anneau de **512 échantillons par étape** en mémoire, percentiles p50/p95/p99 ; export Prometheus à `/v1/health/metrics` (**session requise**). **Non persistantes** : perdues au redémarrage. Pas d'OpenTelemetry (choix documenté) EV-076
Traçage	Pas de traçage distribué. Corrélation par `correlation_id` / `session_id` dans l'audit et par `VoiceTurnTrace` (marques de temps par tour) dans le journal : riche pour la voix (décomposition complète par étape) (193 traces analysées)
Instrumentation du cerveau	Le champ `llama={cache_n, prompt_n, prompt_ms, predicted_per_second…}` des traces est `None` **dans les 193 tours** : la part de préremplissage réutilisée du cache de prompt n'est donc pas observable en exploitation EV-009

Sujet	Constat
Tableaux de bord	Console d'opérateur (11 pages) et espace client en PWA ; `/v1/health/latency` classe les étapes par p95
Alertes	Notifications internes (moniteur proactif, heartbeat, `gate_raised`). Aucune alerte externe (page, courriel d'astreinte) n'a été constatée
Volume d'audit	88 % des enregistrements sont des `access_denied` : bruit qui noie les évènements utiles
Diagnostic en production	Bon pour un tour de voix isolé ; limité pour le cerveau (pas de statistiques de cache), pour le TTS (pas de taux d'échec persistant) et pour les tendances (métriques non conservées)

19. Déploiement

Figure 9 — Déploiement réel (une seule machine) et ordre d'arrêt/démarrage

Matériel

- Intel Core i7-12700K · 12 cœurs / 20 threads
- RAM 63,7 Go (≈ 16 Go libres au relevé)
- GPU0 RTX 5060 Ti 16 Go (aussi carte vidéo du bureau)
- GPU1 RTX 3060 12 Go (voix) · pilote 595.97
- C: NTFS 948 Go (192 libres)

Ordre de démarrage (launch_mike_all.ps1 → restart_mike_clean.ps1 → launch_mike.ps1)

0	Vérifie le portproxy hérité ; tue les orphelins du projet
1	start_qwen36_server.ps1 → llama-server :8081 (attend /health, 300 s)
2	start_voice_worker_wsl.ps1 → worker :8998 + sonde /v1/probe authentifiée
3	start_mike.ps1 → uvicorn :8083 (refuse sans jeton éphémère du worker)
4	tunnel cloudflared (obligatoire si la voix Twilio est active)
5	ouvre les 3 surfaces (opérateur, client, cockpit)

Processus et ports constatés (relevé du 2026-10-02 13:06)

llama-server.exe

- PID 26176 · 127.0.0.1:8081
- RSS 22,8 Go · démarré 09:28
- b10826-73a43d1f6 · 12 threads

python (MIKE)

- PID 33412 · 127.0.0.1:8083
- RSS 811 Mo · démarré 09:28
- .venv Python 3.12.10

worker voix (WSL2)

- wslrelay 127.0.0.1:8998
- /opt/mike-voice/.venv (3.12.8)
- RSS 4,3 Go · 3 h 44 de marche

cloudflared

- tunnel sortant
- mike.f51labs.com
- RSS 37 Mo

Planificateur Windows

- install_heartbeat.ps1 prévoit heartbeat 30 min + briefing 7 h
- Get-ScheduledTask : aucune tâche MIKE/F51 visible (sans droits admin)
- heartbeat_history.jsonl écrit à 13:16 : mécanisme non établi

Constats de déploiement

- Un seul hôte, un seul tenant de calcul : pas de redondance, pas de haute disponibilité ; arrêt = indisponibilité totale
- VRAM : GPU0 11,5 / 16,3 Go · GPU1 8,8 / 12,3 Go (relevé) ; le GPU0 sert aussi l'affichage du bureau (--n-cpu-moe 24 choisi pour cela)
- Services Windows (NSSM) possibles via install_mike_service.ps1 : rotation de logs 50 Mo ; le chemin par défaut n'en a pas

Pastilles : vert = validé (testé ou mesuré) · ambre = non validé, partiel ou désactivé · rouge = absent ou risque constaté. Source : constats de l'audit du 2026-10-02 (registre des preuves, Annexe A).

19.1 Exécution réelle

Un **poste de travail unique** héberge tout (§8). Au relevé du 2 octobre 2026 13:06 : `llama-server` (PID 26176, 22,8 Go de RSS, démarré à 09:28), MIKE (PID 33412, 811 Mo, démarré à 09:28), worker de voix dans WSL2 (≈ 4,3 Go de RSS, 3 h 44 de marche), `cloudflared` EV-025 EV-027.

19.2 Ordre de démarrage

1. `launch_mike_all.ps1` (raccourci du bureau) : vérifie le *portproxy* hérité, arrête les processus orphelins du projet.
2. `restart_mike_clean.ps1` : cerveau propre, puis `launch_mike.ps1`.
3. `start_qwen36_server.ps1` (nom hérité) → `llama-server` :8081 ; attend `/health`.
4. `start_voice_worker_wsl.ps1` → worker :8998, **sonde `/v1/probe` authentifiée** (audio → VAD → STT) ; jeton éphémère partagé avec MIKE.
5. `start_mike.ps1` → uvicorn :8083 — **refuse de démarrer seul** (le jeton éphémère du worker l'en empêche : on ne redémarre pas « seulement MIKE »).
6. Tunnel `cloudflared` (obligatoire si la voix Twilio est active : sans tunnel, le démarrage échoue plutôt que de laisser mourir les appels).
7. Ouverture des trois surfaces.

19.3 Dépendances à reconstruire sur une machine neuve

Pas dans la sauvegarde (se régénèrent mais **exigent Internet**) : le modèle GGUF (≈ 20,2 Gio, URL et SHA-256 épinglés dans `.env.runtime`), les poids de voix (≈ 7,6 Go, catalogue épinglé par révision et SHA-256 dans `scripts/ops/fetch_voice_models.py`), l'environnement Python Windows (`config/requirements.txt`, CPython 3.12.10) et celui du WSL (`/opt/mike-voice/.venv`, ≈ 9,3 Go), la compilation de `llama.cpp` CUDA, et le cache de l'embedder BGE-M3 (≈ 2,3 Go). Règle du dépôt : **deux environnements Python seulement** (Windows et worker WSL).

19.4 Installateur

Un kit Windows de développement 2026.10.02 (exécutable de 5,7 Mo + 7 volumes de ≈ 1,9 Go, 11,8 Go au total) a été compilé avec Inno Setup à partir d'une arborescence propre ; les huit SHA-256 ont été recalculés hors du script. **Il n'a jamais été installé**, l'exécutable n'est pas signé (`NotSigned`), le kit ne contient ni le modèle ni les poids de voix (téléchargés après installation, réseau requis) et la voix reste en WSL2, ce qui contredit l'exigence « 100 % Windows » du propriétaire (décision d'architecture ouverte) EV-084.

19.5 Constat

Il n'y a ni redondance, ni haute disponibilité, ni environnement de préproduction. Un arrêt de l'hôte, du GPU0 ou du WSL entraîne l'indisponibilité totale. C'est cohérent avec une démonstration et un pilote supervisé ; ce n'est pas une architecture d'exploitation pour plusieurs clients simultanés.

20. Validation et tests

20.1 Suite automatisée

Commande : `.\.venv\Scripts\python.exe -m pytest -q --basetemp <tmp> -p no:cacheprovider --durations=12` — exécutée le **2 octobre 2026 à 13:12**, commit `6f2d858`, durée 245,5 s EV-080.

> **Résultat : 5 641 réussis · 4 ignorés · 1 échec attendu (xfail) · 0 échec · 4 avertissements.**

Élément	Valeur
Fichiers de test unitaires	376 (le `pytest.ini` ne collecte que `tests/unit/`)
Fonctions de test définies	4 468 (5 641 exécutions après paramétrage)
Tests d'intégration et de bout en bout	3 + 3 fichiers (3 + 10 fonctions) — **non collectés par défaut ; non exécutés** : ils démarrent et arrêtent des services réels

Élément	Valeur
Code de test	≈ 90 800 lignes Python (soit 67 % du volume du code applicatif)
Ignorés (4)	deux tests « script contre serveur vivant » ; un test de vision manuel ; une évaluation aller-retour de la voix qui exige le runtime (`MIKE_RUN_VOICE_WER=1`)
Échec attendu (1)	`test_real_calls_corpus.py::test_defeito_conhecido_continua[CA858ef9-hangup_last-0]` — défaut connu, documenté
Test le plus lent	16,6 s (`test_agenda_and_tasks_flow_local`)

Répartition **approximative** par domaine (affectation par nom de fichier) EV-081 :

Domaine	Fichiers	Fonctions de test
Voix, téléphonie, Twilio	76	1 142
Multi-tenant, accès, sécurité	65	664
Audit et intégrité	23	282
Installateur, manifeste, déploiement	15	240
Autonomie, missions, routines	22	218
MCP et outils	22	218
Tableau de bord et UI	21	201
Agenda et rendez-vous	10	139
Cerveau, chat, prompts	13	126
Mémoire, RAG, connaissance	15	105
Autres	94	1 133

La couverture est **forte en voix, sécurité et audit**, **faible en mémoire/RAG** (105 tests pour 7 000 lignes de code) et **sans évaluation de qualité de récupération** (pas de jeu de rappel/précision). Aucune mesure de **couverture de code** (pourcentage de lignes) n'a été exécutée (**NON MESURÉ**). Deux garde-fous de méthode méritent d'être signalés : un test de **corpus d'appels réels rejoués** (vérifie des conversations passées, texte seulement) et des tests de **contrat** (routes sans rôle, manifeste, interface).

20.2 Contrôles d'exécution réalisés pendant l'audit

Contrôle	Résultat	Date/heure
`GET /health` , `/livez` , `/readyz` (MIKE)	200 ; `ready: true` ; 33 routers enregistrés (0 échec) ; voix STT/TTS « ready » (`backend: linux_wsl` , CUDA) ; **LightRAG « unavailable »**	13:09
`GET /health` , `/v1/models` , `/props` (llama-server)	200 ; modèle, contexte 81 920, un créneau	13:09
Endpoints authentifiés (`/v1/health/latency` , `/vram` , `/tts` , `/metrics`)	401 — non interrogés (aucune session forgée)	13:09
Vérification de l'époque d'audit vivante (copie)	10 518 intègres, falsification détectée	13:13
Microbenchmark du cerveau	voir §21	13:11–13:12
`check_mike_readiness.py`	**Non terminé** — bloqué en attente du PIN d'opérateur ; arrêté ; aucun verdict revendiqué EV-082	13:16
`run_mike_a_to_z.ps1` , `preflight_live_demo.ps1`	**Non exécutés** : ils démarrent/arrêtent des services ou préparent des appels réels	—

20.3 Preuves d'exploitation antérieures (non refaites ici)

Le dépôt consigne des épreuves réelles, citées avec leur source et **non reproduites** par l'audit : appel autorisé à une personne ayant produit 12 tours scellés et une réservation écrite (`CAfd9940…` , 110 s) ; sondes sans téléphone multi-tours avec rappel et commentaire final enregistrés dans le journal et le tableau de bord ; vérification Chrome sans interface de la fiche d'entreprise (édition, erreur par champ, restitution octet pour octet) ; mesure de la course TTS/Whisper (0/30 échec avec verrou) `(docs/delivery/evidence/2026-10-02-*.md, hardware.md §10.3.1)` EV-083 .

20.4 Ce qui n'a pas pu être validé

- **Installation propre du kit** (jamais installé), redémarrage à froid et temps de démarrage.
- **Appels simultanés**, charge, endurance, reprise après panne du worker en plein appel.
- **Isolation de deux tenants sous appels concurrents réels.**
- **Envois réels** (SMS, courriel, WhatsApp) et intégrations tierces (Google, Shopify, GA4, Ads).
- **Anglais au téléphone** sur une chaîne complète mesurée (une partie des mesures documentées est en français).
- **Qualité de récupération** de la mémoire (aucun jeu d'évaluation).
- **Pourcentage de couverture de code.**
- **Endpoints protégés** de métriques (aucune session utilisée).
- **Taux de faux positifs du barge-in** (journalisé mais non extrait).

21. Performances mesurées

Toutes les valeurs ci-dessous sont **mesurées le 2 octobre 2026**, sauf mention « (source : document) ». Le microbenchmark (`bench_llm.py`) a été exécuté sur le `llama-server` en service, sans appel en cours, avec `cache_prompt` désactivé pour le préremplissage et le démarrage à froid, 5 répétitions par test EV-021.

21.1 Cerveau local (Ornith 1.5 35B-A3B Q4_K_M)

Métrique	Valeur	Dispersion
Décodage (200 jetons, température 0)	**67,1 jetons/s**	min 65,7 · max 68,2 · σ 0,9
Préremplissage (prompt de 2 487 jetons, sans cache)	**626 jetons/s**	min 493 · max 663 · σ 67
Premier jeton — prompt de voix de 246 jetons, sans cache	**925 ms** (médiane)	915 à 973 ms
Premier jeton — même prompt, avec cache de prompt	**77 ms** (médiane de 4 mesures)	75 à 78 ms ; la 1re mesure de la série, sans cache, 918 ms
Contexte configuré	81 920 jetons	262 144 à l'entraînement

Divergence signalée. La documentation du dépôt indique un premier jeton de **167 ms** pour un prompt de 250 jetons (mesure du 25 septembre) *(source : `hardware.md` §2.0)*. La mesure sans cache de cet audit est de **925 ms** ; avec cache de prompt, **77 ms**. L'écart avec le chiffre documenté n'est **pas expliqué** (le document ne précise pas l'état du cache). Le décodage mesuré (67 jetons/s) est, lui, **cohérent** avec la plage documentée (63 à 69). Dans les appels réels, le premier texte des tours passant par le modèle est **médian 1 711 ms** EV-010, ce qui est compatible avec des prompts plus longs (faits système, fiche, historique) et un cache de prompt partiel.

21.2 Voix (extrait des journaux réels, 78 tours du 29 septembre au 2 octobre)

Métrique	Médiane	95e centile	Maximum
Détection de fin de tour	504 ms	804 ms	1 191 ms
ASR à la clôture (texte anticipé)	0,1 ms	457 ms	556 ms
Premier audio TTS Kyutai	940 ms	1 254 ms	5 596 ms
Premier texte — tours avec modèle (n = 44)	1 711 ms	4 254 ms	—
Premier texte — tours déterministes (n = 34)	8 ms	521 ms	—
Texte final → première trame vers Twilio	2 500 ms	5 477 ms	6 031 ms
Fin de parole → premier son (composé)	**2 919 ms**	**5 827 ms**	6 708 ms

Historique complet (177 tours `VOICE_LATENCY` du 21 septembre au 2 octobre, avant et après la Kyutai) : texte final → socket médian **953 ms**, 95e centile **5 200 ms** ; première trame TTS médian **231 ms** — donc le passage à la Kyutai a multiplié par environ **4** la médiane du premier audio EV-011.

21.3 Ressources

Ressource	Valeur mesurée
VRAM GPU0	11 482 Mo utilisés sur 16 311 Mo (cerveau + bureau)
VRAM GPU1	8 652 Mo utilisés sur 12 288 Mo (voix + bureau)
RAM — `llama-server`	22,8 Go (RSS) — experts MoE en mémoire centrale
RAM — MIKE (uvicorn)	811 Mo
RAM — worker de voix (WSL)	≈ 4,3 Go
RAM libre sur l'hôte	≈ 15,9 Go sur 63,7 Go
Taille des bases	tenant `default` 949 Mo (275 Mo utiles) ; piste d'audit vivante 8 Mo
Suite de tests	245,5 s pour 5 641 tests

21.4 Non mesuré

Utilisation CPU en charge · RTF de la synthèse (le document indique 0,57–0,59, **non re-mesuré**) · appels simultanés et débit · taux d'erreur par tour · temps de démarrage à froid du système · latence de bout en bout jusqu'au combiné (réseau + PSTN) · qualité subjective de la voix · taux d'abandon d'appel.

22. Matrice de maturité

Niveaux : **0** absent · **1** concept · **2** prototype · **3** fonctionnel · **4** validé · **5** prêt pour la production. Aucun composant n'est coté 5 : il n'existe pas de preuve sur un artefact installé et signé.

Composant	État	Preuve	Niv.	Risque	Prochain pas
Orchestrateur FastAPI (routes, cycle de vie)	Implémenté et testé	5 641 tests ; 33 routers EV-024 EV-080	4	Moyen (monolithe 54 % du code)	Découper les fichiers > 3 000 lignes
Serveur d'inférence local	Implémenté et testé	Mesures EV-021	4	Moyen (1 créneau)	Préemption + mesure de charge
Chaîne de repli du modèle	Implémenté et testé	EV-022	4	Faible	—
Moteur de voix (pipeline)	Partiel	Appels réels ; latence hors cible EV-009	3	Élevé	Budget de latence (§25)
Transport Twilio Media Streams	Implémenté et testé	Appels réels EV-083	4	Moyen	Épreuve multi-appels
VAD + fin de tour	Implémenté et testé	Mesures EV-005	3	Moyen (heuristique lexicale)	Évaluer sur corpus québécois
ASR Whisper (non-streaming)	Partiel	EV-003	3	Élevé (exactitude québécoise)	Étudier un ASR en streaming FR-CA
Barge-in	Implémenté et testé (unitaire)	EV-002	3	Moyen	Mesurer faux positifs en appel
TTS Kyutai	Partiel	EV-006	3	Élevé (1er son 0,94 s, mono-flux)	Réduire TTFA, paralléliser
Garde de promesses	Implémenté et testé	EV-013	4	Faible	Étendre à d'autres canaux
Détection de répondeur	Partiel (option synchrone hors voie duplex)	EV-016	1	Élevé	Détection asynchrone sur la voie duplex
Transfert vers un humain	Absent	EV-017	0	Élevé	Concevoir
Agenda canonique	Implémenté et testé	Tables, idempotence, rdv réel EV-074	4	Faible	Épreuve de concurrence

Composant	État	Preuve	Niv.	Risque	Prochain pas
Fiche d'entreprise (édition, voix + chat)	Implémenté et testé	Navigateur + tests EV-088	4	Faible	—
Mémoire SQLite + FTS5	Implémenté et testé	EV-070	3	Moyen (espace libre 71 %)	VACUUM, versionnement
Recherche vectorielle	Partiel	EV-072	2	Moyen	Index ANN, évaluation
Mem0	Implémenté, non validé	—	2	Moyen	Mesurer l'apport
LightRAG (graphe)	Partiel	« non chargé » EV-073	2	Moyen	Rétablir ou retirer
Outils MCP (20 + 13)	Implémenté, non validé (sauf fichiers/agenda)	EV-086	3	Moyen	Attestations `tenant_safe`, tests adverses
Autonomie (routines, missions)	Partiel	Magasin durable : contrats seulement	2	Moyen	Relier le magasin durable
Multi-tenant (frontière)	Implémenté et testé	EV-040 EV-042	3	Moyen	Classer les 51 replis `default`
Contrôle d'accès par route	Implémenté et testé	283 routes, 0 exception EV-042	4	Faible	—
Authentification et sessions	Implémenté et testé	PBKDF2 600 k, HMAC EV-045	4	Faible	—
Chiffrement au repos	Partiel : délégué à BitLocker et vérifié par le code ; **BitLocker désactivé** sur le poste audité et la clé USB (mesuré)	EV-046	1	Élevé	Activer et prouver BitLocker (ou attestation nommée) ; chiffrer les secrets
Gestion des secrets	Partiel	Fichiers + ACL + scan EV-047 EV-052	2	Élevé	Durcir ACL, magasin dédié
Piste d'audit (SHA-256, Ed25519)	Implémenté et testé	10 518 vérifiés EV-030	4	Moyen (époque héritée)	Rotation de clé, `STRICT=true`
Ancrage RFC 3161	Implémenté, non validé	0 ancre EV-034	2	Moyen	Activer + vérifier le jeton
Pré-vol d'appels (P1–P7)	Implémenté et testé	EV-060	4	Moyen (liste locale)	Liste officielle
Consentement et effacement	Implémenté et testé	Code + tests ; désactivé ici EV-062	3	Moyen	Épreuve en profil « client »
Observabilité	Partiel	EV-075 EV-076	2	Moyen	Persistance, alertes
Tableau de bord PWA (3 surfaces)	Implémenté et testé	Navigateur + tests	3	Faible	Accessibilité (lecteur d'écran)
Installateur Windows	Prototype	Kit compilé, jamais installé EV-084	2	Élevé	Installation propre + signature
Sauvegarde/restauration	Partiel	Hors-site désactivée ; USB manuelle	2	Élevé	Activer + restaurer en essai
Haute disponibilité	Absent	Hôte unique	0	Élevé	Hors périmètre pilote

Lecture d'ensemble : niveau 3 (fonctionnel). Le cœur applicatif et les mécanismes de preuve sont au niveau 4 ; les trois facteurs qui maintiennent l'ensemble à 3 sont la voix temps réel (latence, concurrence), l'absence de chiffrement/gestion de secrets et l'absence d'installation propre.

23. Risques techniques

Échelles : **Probabilité** — *Certaine* (constatée), *Haute*, *Moyenne*, *Faible*. **Sévérité** — *Critique* (bloque la production ou expose des données), *Élevée*, *Moyenne*, *Faible*. Chaque risque renvoie à sa preuve. Les recommandations sont techniques ; elles ne chiffrent pas d'effort.

23.1 Écarts critiques

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-01	**Latence vocale hors cible** : fin de parole → premier son médian 2,9 s, p95 5,8 s (cible : 1,0 / 1,5 s)	EV-009 EV-010	Conversation perçue comme lente, interruptions mutuelles, abandon	Certaine	Élevée	Fixer un budget par étape (§25, U2) ; mesurer en continu ; faire dire d'abord une première phrase courte sans attendre la suite ; réduire le TTFA de la synthèse
R-02	**Détection de répondeur absente de la voie conversationnelle** : sur un appel sortant tombé en boîte vocale, MIKE tient une conversation avec l'annonce (constaté lors d'un essai du 2 octobre 2026, source : exploitation, non reproduit par l'audit)	EV-016	Appels gaspillés (comptés dans la limite de tentatives, P4), message incohérent	Haute	Élevée	Détection asynchrone côté fournisseur (sans retarder la salutation) + réponse « rappelez ce numéro » conforme à la politique du tenant ; une conception existe, non implémentée
R-03	**Pas de transfert vers un humain**	EV-017	Un client hors périmètre reste sans issue ; la seule sortie est « une personne vous rappellera »	Haute	Élevée	Transfert conditionnel avec reçu typé et heures d'ouverture
R-04	**Liste nationale de non-appel du CRTC non consultée** (P2 locale)	EV-061	Non-conformité possible d'une prospection sortante	Moyenne	Élevée	Décision commerciale du propriétaire ; en attendant, ne pas lancer de prospection à froid
R-05	**Installateur jamais installé, non signé, sans modèle ni voix**	EV-084	Impossible de livrer à un client ; avertissement « éditeur inconnu »	Certaine	Élevée	Installation propre + batterie de tests de livraison B ; certificat de signature
R-06	**Garde-fous de conformité désactivés sur le poste audité** (consentement, assurance, confinement, rétention)	EV-048	Les mesures du poste ne représentent pas la configuration client ; comportements « protégés » non exercés	Certaine	Moyenne	Profil d'essai « client » avec les quatre garde-fous actifs ; rejouer la suite et les appels

23.2 Risques de production

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-07	**Données au repos non chiffrées sur le poste audité** : BitLocker est désactivé sur C: et sur la clé USB de sauvegarde (mesuré) ; le code le détecte mais le mode est « avertir » et aucune attestation n'est enregistrée	EV-046	Vol ou copie du disque/clé USB = lecture complète des données de clients	Moyenne	Élevée	Activer BitLocker sur le volume de données et sur les supports amovibles, le **prouver** (sortie `manage-bde` ou attestation nommée) et passer le mode à `block` pour une installation client
R-08	**Secrets en clair et ACL permissives** (lecture pour `CodexSandboxUsers`, modification pour deux SID non résolus)	EV-047	Un processus ou compte local peut lire les clés de session et d'audit	Moyenne	Élevée	Restreindre les ACL au compte de service ; magasin de secrets dédié ; rotation
R-09	**Clé de signature d'audit affichée dans une session d'audit** (et copiée sur une clé USB non chiffrée)	§5.4 ; EV-030	Quiconque la détient peut signer de faux enregistrements	Faible	Moyenne	Rotation immédiate (nouvelle époque) ; garder la clé hors des copies
R-10	**Ancrage RFC 3161 absent, jeton non vérifié par le code**	EV-034	La piste n'est pas opposable à l'opérateur ; réécriture indétectable de l'extérieur	Moyenne	Moyenne	Activer l'ancrage, vérifier la signature du jeton, conserver les ancres hors de l'hôte

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-11	`MIKE_AUDIT_STRICT=false`	EV-035	Une panne d'écriture d'audit n'empêche pas l'action	Moyenne	Moyenne	`STRICT=true` en production avec alerte
R-12	**Anglais non mesuré de bout en bout**	§20.4	Qualité EN inconnue pour un produit bilingue	Moyenne	Moyenne	Corpus d'appels anglais et mêmes mesures qu'en français

23.3 Points uniques de défaillance

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-13	**Hôte unique, GPU unique par fonction, aucune redondance**	EV-025 EV-026	Toute panne = indisponibilité complète ; pas de reprise d'appel	Moyenne	Élevée (hors pilote)	Hors périmètre du pilote ; documenter le RTO/RPO et prévoir un second hôte froid
R-14	**Supervision du worker de voix non établie** : le chien de garde relance le cerveau ; aucun équivalent n'a été constaté pour le worker (le lanceur le démarre et le sonde)	EV-023	Worker mort en cours de journée = voix indisponible sans relance	Moyenne	Élevée	Vérifier/ajouter une relance supervisée avec alerte
R-15	**GPU0 partagé avec l'affichage du bureau** (`--n-cpu-moe 24` choisi pour cela)	EV-026 (source : `hardware.md` §2.0.2)	Une application graphique peut pousser le pilote vers la mémoire partagée *sans erreur* et effondrer le décodage de ~70 %	Moyenne	Moyenne	Poste dédié sans bureau, ou surveillance de la VRAM avec seuil d'alerte

23.4 Goulets d'étranglement, concurrence et échelle

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-16	**Ressources sérialisées** : un créneau LLM, une synthèse Kyutai à la fois, un Whisper résident, pool de 2 threads d'ASR anticipé	EV-006 EV-012	Le 2e appel simultané voit sa latence s'ajouter à celle du 1er	Haute	Élevée	Mesurer N appels ; envisager un second créneau LLM, un pool de TTS, un second GPU voix
R-17	**Aucun plafond d'admission d'appels**	EV-008	Un 3e appel est accepté puis dégradé au lieu d'être refusé proprement	Moyenne	Élevée	Contrôle d'admission avec message d'occupation et enregistrement du refus
R-18	**File d'inférence non préemptive** : une génération `batch` en cours (par exemple 4 096 jetons ≈ 61 s au débit mesuré de 67 jetons/s) retarde un appel `realtime`	EV-012	Silence au téléphone pendant une routine en arrière-plan	Moyenne	Élevée	Annulation/abandon des travaux `batch` à l'arrivée d'un `realtime` ; planifier les routines hors des heures d'appel
R-19	**Recherche vectorielle en balayage linéaire**	EV-072	Coût et latence croissent linéairement avec le nombre de vecteurs ; ≈ 260 Mo de lecture par requête à 64 559 vecteurs	Moyenne (aujourd'hui 216 vecteurs)	Moyenne	Index ANN local (HNSW) ou stockage vectoriel dédié ; limiter la requête
R-20	**Piste d'audit unique, sérialisée, `fsync` par enregistrement ; 88 % de bruit**	EV-036	Goulot et lisibilité dégradée à l'échelle	Moyenne	Moyenne	Ne pas inscrire en piste les refus répétitifs (compteur agrégé) ; une piste par tenant ou par lot

23.5 Persistance

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-21	**71 % de pages libres** dans la base de 949 Mo (`auto_vacuum = 0`)	EV-071	Disque gaspillé, copies/sauvegardes 3,5× plus lourdes que nécessaire	Certaine	Faible	`VACUUM` planifié ; `auto_vacuum=INCREMENTAL`
R-22	**Pas de versionnement de schéma** (`user_version = 0`) ; migrations ad hoc et réinitialisation des embeddings	EV-070	Mise à jour client risquée, pas de retour arrière contrôlé	Moyenne	Moyenne	Table de migrations versionnée ; tests de migration

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-23	**Sauvegarde à chaud non garantie cohérente** ; pas de restauration éprouvée ; hors-site désactivée	§11.3	Perte ou corruption lors d'un incident	Moyenne	Élevée	Checkpoint WAL avant copie ; test de restauration périodique ; activer la sauvegarde chiffrée
R-24	**Époque d'audit héritée rompue** (34,6 Mo)	EV-032	L'historique antérieur au 23 septembre n'est pas probant	Certaine	Faible	Conserver, documenter, ne pas s'y fier

23.6 Sécurité propre à l'IA agentique

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-25	**Injection de prompt sans assainissement ni marquage de provenance** du contenu issu d'outils (web, scraping, courriel, documents)	EV-050	Un texte hostile dans une page ou un document peut orienter le modèle vers des outils du propriétaire	Moyenne	Élevée	Séparer instructions et données ; étiqueter la provenance ; approbation humaine des actions sensibles ; tests adverses
R-26	**Repli sur** `default_tenant()` : 51 occurrences en 26 fichiers, fuite mesurée le 2026-10-01	EV-041	Une requête sans contexte lit la mémoire d'une autre entreprise	Moyenne	**Critique**	Classer chacune ; faire échouer fermé (`NoTenantContext`) hors des chemins explicitement mono-tenant
R-27	**Seuls 4 serveurs MCP sur 20 déclarent** `tenant_safe`	EV-086	Un outil sans attestation peut mélanger les données de tenants	Moyenne	Moyenne	Attestation + test d'isolation par serveur, ou désactivation en multi-tenant
R-28	**Calcul d'argent hors TPS/TVQ sans garde déterministe** (remises, sous-totaux, rapport fiscal)	§9.3	Montants erronés énoncés à un client	Moyenne	Élevée	Étendre l'intercepteur ; interdire l'énoncé de montants non calculés

23.7 Observabilité et intégration

ID	Problème	Preuve	Impact	Prob.	Sév.	Recommandation
R-29	**Logs sans rotation ; métriques non persistantes ; instrumentation du cerveau vide dans les traces**	EV-075 EV-076 EV-009	Diagnostic de tendance impossible ; cache de prompt invisible	Certaine	Moyenne	Rotation, export des métriques, alimenter `llama={…}` dans les traces
R-30	**LightRAG « activé mais non chargé »** ; reconstruction très longue après redémarrage (notes d'exploitation)	EV-073	Capacité de graphe silencieusement indisponible	Certaine	Moyenne	Rétablir et mesurer, ou retirer (7 couches de la politique de suppression)
R-31	**Intégrations tierces non validées** (Google, Shopify, GA4, Ads, courriel sortant, WhatsApp)	EV-086	Promesse produit supérieure à la preuve	Haute	Moyenne	Valider par intégration ou retirer de l'offre
R-32	**Exactitude de l'ASR en français québécois** (15/40 sur banc interne, source document)	§10.3	Mal-compréhension de mots courts (heures, jours, oui/non)	Haute	Élevée	Voir §25, U3 ; boucler sur la confirmation orale des éléments critiques

24. Dette technique

Domaine	Constat	Preuve
Monolithes	`core/server` = 54 % du code ; au moins 12 fichiers dépassent 1 500 lignes (`mike_voice_flow.py` 3 809, `mike_tools_local.py` 3 738, `routers/telephony.py` 3 047, `mike_chat_core.py` 2 776, `routers/voice_stream.py` 2 059…)	EV-087
Import double de modules	Le même module est chargé sous deux noms (`routers.x` et `core.server.routers.x`) ; l'état en RAM existait en deux copies (cause d'un appel muet) ; contournement par des ancres dans `sys.modules` (verrous d'audit, usine de tenants)	`mike_audit.py` , `mike_tenant.py` ; `test_tenant_module_duplication.py`
Code et configuration orphelins	Variables `MIKE_PIPEDRIVE_*` sans module ; chemin `local` (llama-cpp-python en processus) et `mock` conservés dans le cycle de vie ; relay ConversationRelay désactivé ; dossier `llama.cpp.new/` (2,1 Go) ; copie *legacy* `runtime/memory` (945 Mo) ; 24 routes sans client selon le dépôt	EV-085
Surface de configuration	152 variables dans `.env.runtime` , 511 noms `MIKE_*` référencés dans `core/` : comportements difficiles à raisonner et à documenter	EV-087

Domaine	Constat	Preuve
Nommage hérité	Script `start_qwen36_server.ps1` sert le modèle Ornith ; plusieurs commentaires et constantes mentionnent d'anciens moteurs retirés	EV-020
Langues du code	Commentaires et docstrings souvent en portugais, produit en FR/EN : barrière pour une revue externe et une industrialisation	lecture du code
Dérives documentation ↔ réalité	(i) TTFT documenté 167 ms ≠ 925 ms mesuré sans cache ; (ii) la description du sous-agent `mike-skills` annonce 54 compétences, il y en a 59 ; (iii) la compétence `entrega-produto` affirme que `LICENSE` n'existe pas, il existe (2 190 octets) ; (iv) `hardware.md` §3 décrit le cerveau précédent	EV-021 EV-087
Volumes hors code dans le dépôt	93 Go d'artefacts, 37 Go de `build/`, 18 Go de `training/`, 3,6 Go de `.test-tmp/` dans le dossier de travail	EV-085
Tests	4 468 fonctions, mais un seul fichier de corpus d'appels réels (texte) ; peu de tests sur la qualité de récupération ; pas de mesure de couverture	§20
Outillage de développement	62 agents Claude Code décrits dans `.claude/agents/` ; indisponibles lors de cet audit (erreur de modèle configuré)	§5.4

25. Incertitudes technologiques

Cette section isole les problèmes qui **ne se résolvent pas par du développement courant** : leur issue est incertaine et exige expérimentation et mesure. Pour chacun, le dossier donne l'état constaté (renvoyant aux preuves), la raison de la difficulté, une hypothèse, les expériences, les métriques et un résultat attendu. Les **critères de succès chiffrés sont des propositions** à ratifier avec les parties prenantes, pas des constats. Les ressources sont décrites par nature ; aucune durée ni aucun coût n'est estimé faute de base concrète.

25.1 Vue d'ensemble

#	Incertitude	Écart mesuré aujourd'hui	Domaine
U1	Dialogue vocal naturel en français québécois sans modèle audio-natif	Cascade ; pas de *backchannel* ni de génération pendant la parole EV-014	Voix
U2	Latence de bout en bout ≤ 1,5 s (p95) sur matériel grand public	Médiane 2,9 s · p95 5,8 s EV-009	Voix / IA
U3	Reconnaissance vocale fiable du français québécois par téléphone	15/40 exacts sur banc interne (source : document) ; pas de streaming EV-003	Voix
U4	Gestion du tour de parole et du *barge-in* sans coupure ni faux positif	Heuristique lexicale + Smart Turn ; taux de faux positifs non extrait EV-002 EV-005	Voix
U5	Concurrence : N appels simultanés sur deux GPU grand public	Ressources sérialisées ; capacité non mesurée EV-006 EV-008 EV-012	Plateforme
U6	Inférence locale MoE : partition CPU/GPU, stabilité et repli sûr	Effondrement silencieux possible ; aucun repli EV-020 EV-022	IA / Plateforme
U7	Isolation et équité entre clients pour les ressources partagées	Isolation logique, calcul partagé EV-041	Multi-tenant
U8	Sécurité d'un agent à outils contre l'injection de prompt	Défenses textuelles seulement EV-050	Sécurité IA
U9	Récupération de mémoire locale, performante et évaluée	216 vecteurs ; balayage linéaire ; aucune évaluation EV-071 EV-072	Données
U10	Fiabilité des appels d'outils et vérité des actions déclarées	Garde de promesses couvrant 4 catégories EV-013	Agents
U11	Évaluation objective de la qualité conversationnelle	Corpus d'appels réels rejoués en texte (14 appels selon le dépôt, non recompté) ; aucun indice de qualité	Évaluation

U1 — Dialogue vocal naturel en français québécois sans modèle audio-natif

Rubrique	Contenu
État actuel	Cascade VAD → Smart Turn → Whisper → LLM → Kyutai avec écoute continue et *barge-in* par règles. Pas de génération pendant que le client parle, pas de *backchannels* produits, pas de reconnaissance en continu EV-001 EV-003 EV-014. Une étude interne (29 septembre 2026) conclut qu'aucun modèle duplex publié ne parle français, le candidat le plus proche étant anglais/chinois *(source : `docs/produto/estudos/2026-09-29-duplex-*.md`, non re-vérifié).*
Problème technique	Obtenir des chevauchements acceptables, des acquiescements et une complétion de phrase naturels en français québécois avec des modèles locaux tenant sur 12 à 16 Go de VRAM.
Pourquoi ce n'est pas trivial	Aucun modèle de référence ouvert en duplex français ; la cascade cumule cinq latences ; l'entraînement ou l'adaptation exige des conversations québécoises annotées, rares et soumises à la Loi 25.
Hypothèse	Un module appris de gestion de tour (prédiction de fin de tour + de *backchannel*) greffé sur la cascade réduit la latence perçue et améliore la naturalité sans modèle audio-natif.
Expériences	(a) Constituer et annoter un corpus consenti de tours, chevauchements et *backchannels* ; (b) comparer A/B règles actuelles, module appris et duplex ouvert (anglais) ; (c) évaluation par panel d'écoute bilingue.
Métriques	Taux de coupures intempestives ; délai de réponse perçu ; indice de naturalité (MOS) ; taux de complétion de tâche.
Critères de succès (proposés)	Coupures intempestives < 5 % des tours ; naturalité ≥ référence cascade + 0,3 MOS ; aucune régression de complétion de tâche.
Ressources	Corpus et annotateurs bilingues FR-CA/EN ; cadre juridique de consentement ; GPU de test 12–24 Go ; ingénieur(s) voix/ML.
Résultat attendu	Méthode et banc d'évaluation reproductibles de gestion de tour pour le français québécois en pipeline local ; décision d'architecture (cascade améliorée ou duplex) fondée sur des mesures.

U2 — Latence de bout en bout ≤ 1,5 s (p95) sur matériel grand public

Rubrique	Contenu
État actuel	Fin de parole → premier son : médiane 2 919 ms, p95 5 827 ms. Décomposition : fin de tour 504 ms ; premier texte du LLM 1 711 ms (tours avec modèle) ; premier audio Kyutai 940 ms EV-009 EV-010. Le premier jeton mesuré vaut 925 ms sans cache de prompt et 77 ms avec EV-021.
Problème technique	Réduire des étapes **en série** et dépendantes (décision → génération → synthèse autorégressive) sans perdre en exactitude ni en qualité vocale.
Pourquoi ce n'est pas trivial	Le préremplissage du MoE dépend de l'offload CPU ; la synthèse 1,6 B génère par pas de 80 ms ; chaque gain d'une étape (taille de contexte, `n_q`, découpage) modifie la qualité ; le GPU0 est partagé avec l'affichage.
Hypothèse	(i) Un préfixe de prompt stable rend le cache systématiquement chaud (925 → ~80 ms) ; (ii) démarrer la synthèse sur une première proposition courte déterministe masque le temps du modèle ; (iii) régler `n_q` et le découpage de la Kyutai ramène le premier audio sous 400 ms ; (iv) la génération spéculative pendant la pause de fin de tour économise l'attente.
Expériences	Banc de **rejeu des appels réels** avec ablations étape par étape ; mesure du taux de réussite du cache de prompt en exploitation (instrumentation aujourd'hui vide, EV-009) ; `n_q` ∈ {8, 16, 32} ; politique de découpage ; variantes de contexte.
Métriques	p50/p95 composé ; TTFT ; TTFA ; facteur temps réel de la synthèse ; WER et MOS pour détecter les régressions.
Critères de succès (proposés)	p95 ≤ 1,5 s pour les tours déterministes et ≤ 2,5 s pour les tours avec modèle ; dégradation MOS ≤ 0,2.
Ressources	Banc de rejeu et corpus d'appels ; GPU dédié ; ingénieur inférence/voix.
Résultat attendu	Recette documentée de réduction de latence pour une cascade locale FR/EN et budget de latence par étape, publiables comme connaissance technique.

U3 — Reconnaissance vocale fiable du français québécois par téléphone

Rubrique	Contenu
État actuel	Whisper large-v3-turbo sans streaming, fp32 ; transcription anticipée qui masque ≈ 0,45 s ; 15/40 exacts sur un banc interne de locuteurs québécois (14/40 en qualité téléphonique) *(source : document, non re-mesuré)* ; plusieurs moteurs en streaming essayés puis retirés *(source : document)* EV-003 EV-004.
Problème technique	Reconnaître de façon fiable des mots courts décisifs (oui/non, heures, jours, noms) en accent québécois, en bande étroite 8 kHz, avec bruit de ligne, et obtenir des hypothèses partielles utilisables.

Rubrique	Contenu
Pourquoi ce n'est pas trivial	Peu de données publiques FR-CA à 8 kHz ; les erreurs portent sur des énoncés d'une syllabe où le contexte manque ; le compromis latence/exactitude des modèles en streaming ouverts n'a pas été favorable lors des essais internes.
Hypothèse	Un biais lexical contextuel (jours, heures, noms de services de la fiche) et/ou une adaptation légère sur données téléphoniques FR-CA, avec confirmation orale des éléments critiques, porte l'exactitude des entités critiques au niveau requis sans streaming.
Expériences	Banc téléphonique FR-CA consenti (≥ 500 énoncés, annotés) ; comparaison Whisper de base, biais lexical, adaptation ; mesure de WER/CER et d'exactitude d'entités ; test du coût d'une confirmation orale.
Métriques	WER, exactitude d'entités (heure, jour, oui/non), latence de décodage, taux de demandes de répétition.
Critères de succès (proposés)	Exactitude des entités critiques ≥ 95 % *après* confirmation ; WER relatif −30 % ; latence de décodage ≤ valeur actuelle.
Ressources	Données de parole consenties et annotées ; GPU d'entraînement ; expertise ASR ; cadre Loi 25.
Résultat attendu	Recette et banc d'évaluation ASR pour le français québécois téléphonique, avec seuils d'exactitude par type d'entité.

U4 — Gestion du tour de parole et du *barge-in*

Rubrique	Contenu
État actuel	Silence de fin de segment 320 ms ; Smart Turn v3.2 tranche (66 tours sur 78 clos par lui, 12 par silence) ; critère « sémantique » = liste fermée de réponses courtes FR/EN ; plan d'arrêt du *barge-in* à 0,6 s ; faux positifs du VAD journalisés mais non extraits EV-002 EV-005.
Problème technique	Ni couper un client qui réfléchit ou énumère un numéro, ni le faire attendre ; distinguer interruption réelle, acquiescement, écho du combiné et bruit ; fonctionner en FR, EN et alternance de langues.
Pourquoi ce n'est pas trivial	Les indices prosodiques s'érodent à 8 kHz ; l'écho du combiné ressemble à de la parole ; la langue peut changer en cours de tour ; chaque session consomme du calcul GPU.
Hypothèse	Un classificateur léger combinant prosodie, texte partiel et attente de la question (le système attend un nombre, une heure, un oui/non) réduit à la fois coupures et attentes.
Expériences	Corpus annoté de fins de tour et d'interruptions, avec bruit et écho injectés ; comparaison règles / Smart Turn / modèle contextuel ; extraction du taux réel de faux positifs en exploitation.
Métriques	Coupures intempestives ; délai de décision de fin de tour ; faux positifs de *barge-in* par heure ; délai d'arrêt de la lecture (`voice.barge_in_stop`) ; F1 de fin de tour.
Critères de succès (proposés)	Coupures < 3 % ; délai de décision p95 ≤ 600 ms ; faux positifs de *barge-in* < 2 % des tours ; arrêt de lecture < 300 ms.
Ressources	Corpus annoté ; banc d'injection d'écho et de bruit ; ingénieur ML/signal.
Résultat attendu	Détecteur de tour bilingue évalué sur matériel grand public et jeu de paramètres justifié par la mesure.

U5 — Concurrence : N appels simultanés sur deux GPU grand public

Rubrique	Contenu
État actuel	Un créneau LLM (`--parallel 1`) avec file d'urgence non préemptive ; une synthèse Kyutai à la fois (verrou) ; un Whisper résident, verrou GPU commun ; aucun contrôle d'admission ; **capacité non mesurée** EV-006 EV-008 EV-012.
Problème technique	Servir un nombre cible d'appels (p. ex. 3 à 5) dans le budget de latence avec un LLM MoE à offload CPU et une synthèse autorégressive non groupée.
Pourquoi ce n'est pas trivial	Le groupage continu du LLM améliore le débit mais allonge le premier jeton ; la synthèse autorégressive se groupe mal ; Whisper et Kyutai se disputent le GPU1 et le verrou de capture CUDA ; la VRAM est limitée.
Hypothèse	`--parallel` 2–4 avec cache de prompt par créneau, un pool ou un groupage de synthèse, et une admission avec refus gracieux permettent 3 appels simultanés dans une dégradation bornée.
Expériences	Charge progressive N = 1…6 par rejeu d'appels ; variation de `--parallel` , `n_q` , `n-cpu-moe` , nombre de threads d'ASR anticipé ; mesure de la saturation et des rejets.
Métriques	p50/p95 de latence composée par N ; débit de tours ; VRAM ; taux d'erreur et de rejet.
Critères de succès (proposés)	N cible atteint avec p95 ≤ budget U2 + 1 s ; aucun appel muet ; rejet explicite au-delà de N.
Ressources	Banc de charge ; GPU de test ; ingénieur plateforme.
Résultat attendu	Courbe de capacité mesurée et règles d'admission pour une cascade voix locale sur matériel grand public.

U6 — Inférence locale MoE : partition CPU/GPU, stabilité et repli sûr

Rubrique	Contenu
État actuel	Ornith 35B-A3B Q4_K_M, `--n-cpu-moe 24` choisi parce que le GPU0 sert aussi l'affichage ; avec 16 le décodage serait plus rapide (≈ 78 contre 63–69 jetons/s) mais un pic de VRAM du bureau déclenche un effondrement d'environ 70 % *sans erreur (source : `hardware.md` §2.0.2)* ; cache KV `q4_0` non évalué en qualité ; aucun repli : un cerveau en panne est une erreur réelle EV-020 EV-022.
Problème technique	Maximiser le débit sous une VRAM variable sans régression silencieuse, garantir la qualité en cache KV quantifié et assurer une continuité de service dégradée sans retomber dans le nuage.
Pourquoi ce n'est pas trivial	Le comportement du pilote vidéo (mémoire partagée) n'est pas exposé comme une erreur ; l'effet de la quantification du KV sur les tâches FR-CA est inconnu ; un modèle de secours doit rester cohérent avec la mémoire et le ton de l'agent.
Hypothèse	Un contrôleur qui mesure la VRAM libre et ajuste `n-cpu-moe` évite la falaise ; un modèle local léger de secours (CPU ou petite GPU) tient les phrases fixes et l'agenda en mode dégradé annoncé.
Expériences	Balayage `n-cpu-moe` × charge de bureau ; évaluation de qualité KV `q4_0` contre `q8_0` sur un banc FR/EN ; essai d'un modèle de secours sur les tours déterministes.
Métriques	Jetons/s et TTFT en fonction de la VRAM libre ; qualité (exactitude sur banc) ; taux de bascule ; durée de dégradation.
Critères de succès (proposés)	Aucune chute > 20 % du débit sans alerte ; perte de qualité KV ≤ 2 points ; service dégradé disponible en < 5 s après panne.
Ressources	Matériel avec/sans bureau ; banc d'évaluation ; ingénieur inférence.
Résultat attendu	Règles de configuration auto-adaptatives pour l'inférence MoE locale et schéma de repli sans nuage.

U7 — Isolation et équité entre clients pour les ressources partagées

Rubrique	Contenu
État actuel	Données isolées par tenant (bases, Mem0, dropzone) ; calcul (créneau LLM, TTS, ASR) partagé sans équité ; piste d'audit unique ; fuite mesurée le 1er octobre via le repli `default_tenant()` ; cache de prompt à créneau unique EV-040 EV-041.
Problème technique	Garantir l'absence de fuite *et* un partage équitable du calcul quand plusieurs clients utilisent la même machine, y compris via des canaux latéraux (cache de préfixe, temps de réponse).
Pourquoi ce n'est pas trivial	Les caches de préfixe partagent des états entre requêtes ; l'équité exige de la préemption absente aujourd'hui ; l'évaluation d'une fuite par canal latéral demande une méthode de test adverse.
Hypothèse	Une classe de service par tenant avec quotas, des caches de préfixe cloisonnés et un échec fermé sans contexte éliminent les fuites constatées et bornent l'impact d'un client bruyant.
Expériences	Tests adverses de fuite inter-tenant (contexte absent, cache de préfixe, rejeu) ; charge mixte avec un client bruyant ; mesure d'équité de latence.
Métriques	Nombre de chemins sans contexte ; taux de fuite sur banc adverse ; ratio de latence d'un client bien-élevé en présence d'un client bruyant.
Critères de succès (proposés)	0 chemin de repli implicite ; 0 fuite sur banc adverse ; dégradation d'un client ≤ 25 % sous voisin bruyant.
Ressources	Banc adverse ; ingénieur sécurité/plateforme.
Résultat attendu	Modèle d'isolation vérifiable pour un agent local multi-client et méthode de test de fuite.

U8 — Sécurité d'un agent à outils contre l'injection de prompt

Rubrique	Contenu
État actuel	Garde textuelle dans le prompt de voix ; outils sensibles réservés au propriétaire ; filtre de promesses sur la sortie ; kill-switch. Aucun marquage de provenance des contenus d'outils ni assainissement EV-050.
Problème technique	Empêcher un contenu hostile (page web, courriel, document, parole du client) d'orienter un agent doté d'outils puissants (fichiers, commande, courriel, agenda).
Pourquoi ce n'est pas trivial	Aucune séparation parfaite entre instructions et données pour un LLM ; chaque défense réduit l'utilité ; les attaques évoluent.
Hypothèse	Une politique à capacités (l'outil reçoit seulement ce que la tâche autorise), l'étiquetage de provenance et la confirmation humaine des effets de bord réduisent fortement le succès des attaques.

Rubrique	Contenu
Expériences	Banc adverse d'injections directes/indirectes sur les chemins web, courriel et document ; mesure du taux de succès avant/après ; mesure du coût d'utilité (tâches légitimes refusées).
Métriques	Taux de succès d'attaque ; faux refus ; nombre d'actions sensibles sans confirmation.
Critères de succès (proposés)	Réduction ≥ 90 % du taux de succès des attaques ; faux refus ≤ 5 %.
Ressources	Banc adverse ; ingénieur sécurité IA.
Résultat attendu	Architecture de contrôle d'agent à outils pour PME et jeu d'attaques de référence.

U9 — Récupération de mémoire locale, performante et évaluée

Rubrique	Contenu
État actuel	FTS5 + `bm25` sur tous les segments ; vecteurs BGE-M3 pour 216 segments (10 % du contenu hors livres) ; balayage linéaire ; LightRAG « non chargé » ; aucun jeu d'évaluation EV-071 EV-072 EV-073.
Problème technique	Récupérer des passages pertinents pour des corpus de PME (documents, historique de conversations) sous contraintes CPU/VRAM, avec mises à jour incrémentales et bilinguisme.
Pourquoi ce n'est pas trivial	Absence de vérité terrain étiquetée ; segmentation sensible au type de document ; coût d'un graphe de connaissances ; équilibre qualité/latence en voix (budget de quelques centaines de millisecondes).
Hypothèse	Un index ANN local, une fusion hybride FTS+vecteur et un reranker léger améliorent significativement le rappel pour un coût de latence borné.
Expériences	Construire un jeu étiqueté (requêtes, passages pertinents) FR/EN ; comparer FTS seul / vecteur / hybride / + reranker / + graphe ; mesurer la latence.
Métriques	Rappel@5, MRR, latence p95, coût mémoire, temps de réindexation.
Critères de succès (proposés)	Rappel@5 ≥ référence FTS + 15 points ; latence p95 ≤ 150 ms ; réindexation incrémentale sans reconstruction totale.
Ressources	Annotateurs ; ingénieur recherche d'information.
Résultat attendu	Chaîne de récupération locale évaluée et jeu d'évaluation réutilisable.

U10 — Fiabilité des appels d'outils et vérité des actions déclarées

Rubrique	Contenu
État actuel	Quatre catégories de promesses gardées (modification d'agenda, rappel humain, transfert, enregistrement générique) avec reçus typés ; boucle d'outils bornée à 8 pas ; calcul TPS/TVQ corrigé par un intercepteur parce que le modèle seul donne 3 bonnes réponses sur 9 *(source : document)* EV-013.
Problème technique	Garantir que ce que l'agent dit avoir fait correspond à un effet prouvé, et mesurer le taux de réussite des appels d'outils.
Pourquoi ce n'est pas trivial	Les formulations sont infinies (subjonctif, périphrases, bilingue) ; une garde produit des faux positifs/négatifs ; la fiabilité du *function calling* dépend du modèle et du gabarit.
Hypothèse	Un « moteur d'obligations » généralisé (tout engagement exige un reçu avant d'être dit) et des tests métamorphiques réduisent les actions fantômes à un niveau mesurable.
Expériences	Corpus de formulations d'engagement FR/EN ; mesure faux positifs/négatifs de la garde ; taux de réussite des appels d'outils par catégorie ; extension à de nouveaux canaux (SMS, courriel).
Métriques	Taux d'actions déclarées sans reçu ; taux de refus à tort ; taux de réussite des appels d'outils.
Critères de succès (proposés)	0 action déclarée sans reçu sur le banc ; faux refus ≤ 3 %.
Ressources	Corpus annoté ; ingénieur agents.
Résultat attendu	Méthode de « vérité des actions » pour agents conversationnels, avec banc de référence.

U11 — Évaluation objective de la qualité conversationnelle

Rubrique	Contenu
État actuel	Un corpus d'appels réels rejoués en texte (test de non-régression) ; mesures de latence ; aucune métrique de qualité de dialogue, de satisfaction ou de naturalité reproductible.
Problème technique	Comparer objectivement des versions et des modèles sur la qualité d'un dialogue vocal bilingue.
Pourquoi ce n'est pas trivial	L'annotation coûte cher et touche la vie privée ; les sondes synthétiques biaisent ; la qualité est multidimensionnelle (compréhension, tâche, ton, rythme).
Hypothèse	Un indice composite (réussite de tâche, WER d'entités, latence, jugement local calibré) corrèle fortement avec un panel d'écoute humain.
Expériences	Constituer un panel d'écoute ; calibrer un juge automatique local ; mesurer corrélation et stabilité entre versions.
Métriques	Corrélation indice–panel ; variance inter-évaluateurs ; sensibilité aux régressions connues.
Critères de succès (proposés)	Corrélation ≥ 0,8 ; détection de 90 % des régressions injectées.
Ressources	Panel bilingue ; corpus d'appels ; ingénieur évaluation.
Résultat attendu	Indice d'évaluation reproductible pour agents vocaux locaux FR/EN.

26. Opportunités de R&D

Les incertitudes précédentes se regroupent en cinq programmes de recherche appliquée. Ce regroupement est une **proposition de cadrage** ; l'admissibilité de chaque programme à un financement (CCTT / *Mon succès numérique*, PARI-CNRC / *AI Assist*, autres) relève de ses critères propres et doit être validée avec l'organisme.

Programme	Incertitudes	Objectif de recherche	Livrables de preuve	Actifs existants réutilisables
A. Dialogue vocal temps réel local en français québécois	U1, U2, U3, U4	Atteindre une latence perçue et une naturalité acceptables avec une cascade locale FR/EN sur matériel grand public	Banc d'évaluation, corpus consenti, recette de latence, détecteur de tour évalué	Journaux de traces par tour (193 traces), rejeu d'appels, worker de voix, sondes sans téléphone
B. Inférence locale multi-appels sur matériel grand public	U5, U6	Courbe de capacité mesurée, ordonnancement et admission, configuration MoE auto-adaptative, repli sans nuage	Rapport de capacité N × latence, contrôleur de VRAM, évaluation de qualité de quantification	Microbenchmark, `hardware.md` (historique de mesures), file d'inférence à classes d'urgence
C. Sécurité et véracité des agents à outils	U8, U10, U7	Architecture de contrôle (capacités, provenance, confirmation) et « moteur d'obligations »	Banc adverse, taux de succès avant/après, banc de véracité d'action	Garde de promesses, RBAC déclaratif, kill-switch, piste d'audit signée
D. Mémoire et connaissance locales évaluables	U9	Récupération hybride locale mesurée sur corpus de PME FR/EN	Jeu d'évaluation étiqueté, comparaison FTS/vecteur/graphe, index ANN	Pipeline d'ingestion, FTS5, embeddings BGE-M3, Mem0, LightRAG
E. Évaluation objective de la qualité conversationnelle	U11	Indice reproductible de qualité pour agents vocaux	Panel, juge calibré, corrélation	Corpus d'appels réels, traces de latence

Points d'attention pour tout dossier : (i) les chiffres de ce document sont des **états initiaux mesurés** — la valeur de la recherche se démontre par l'écart mesuré avant/après sur un banc fixé à l'avance ; (ii) la collecte de voix de clients relève de la Loi 25 (consentement, finalité, rétention) et doit être planifiée avant l'expérimentation ; (iii) la dette de maintenabilité (§24) et l'absence d'installation propre (§19) ne sont **pas** de la R&D et ne devraient pas être présentées comme telles.

27. Feuille de route

Cette feuille de route est dérivée **uniquement des écarts constatés** (§23, §24, §25). Elle ordonne par criticité, non par effort : aucune durée n'est estimée, faute de base mesurée. « Dép. » = dépendances ; « Risque » = ce qui peut mal tourner dans le travail lui-même.

P0 — Bloqueurs avant toute exploitation chez un client

ID	Problème (réf.)	Travail	Dép.	Risque	Résultat attendu
P0-1	Clé d'audit exposée ; ACL permissives sur les secrets (R-08, R-09)	Générer une nouvelle clé de signature (nouvelle époque d'audit, ancienne conservée), restreindre les ACL de `runtime/auth` et du fichier d'environnement au seul compte de service, retirer les SID non résolus	—	Rupture de continuité de la piste si mal faite	Droits minimaux vérifiés par `Get-Acl`, nouvelle époque vérifiée
P0-2	Données au repos non chiffrées (R-07)	Activer BitLocker sur le volume de données et les supports amovibles ; le prouver (sortie `manage-bde` ou attestation nommée) ; mode `block` pour tout poste client	Droits admin ; politique de clés de récupération	Perte de clé de récupération	Volume chiffré prouvé, signalé par `/v1/governance`
P0-3	Repli sur `default_tenant()` : 51 occurrences (R-26)	Classer chaque occurrence ; faire échouer fermé (`NoTenantContext`) partout sauf aux points explicitement mono-tenant ; test de non-régression qui interdit le motif	—	Régressions sur des chemins non testés	0 repli implicite ; test qui échoue si le motif réapparaît
P0-4	Concurrence non mesurée, aucun plafond d'appels (R-16, R-17)	Banc de charge par rejeu (N = 1 à 6) ; contrôle d'admission avec message d'occupation ; courbe de capacité publiée	Banc de rejeu (§25 U5)	Découvrir que la capacité utile est de 1 appel	Capacité mesurée + règle d'admission testée
P0-5	Latence vocale hors cible (R-01)	Budget par étape ; préfixe de prompt stable pour un cache chaud ; instrumentation du cache dans les traces ; premier audio plus court ; réglage de la synthèse	P0-4 (banc)	Dégradation de qualité vocale	Médiane/p95 composés ramenés à la cible convenue, mesurés avant/après
P0-6	Installateur jamais installé ni signé (R-05)	Installation sur machine propre (batterie B), certificat de signature, clé de publication ; essai de reprise après redémarrage et de retour arrière	Décision du propriétaire sur le certificat ; décision « tout WSL ou tout Windows » (§19.4)	Voix WSL non empaquetée ; installation exigeant Internet	Kit installé, démarre, passe les contrôles, rollback prouvé
P0-7	Garde-fous désactivés sur le poste (R-06)	Profil d'essai « client » (consentement, assurance, confinement, rétention actifs) ; rejouer la suite et un jeu d'appels	—	Révèle des blocages cachés	Mesures reproduites en configuration client
P0-8	Liste nationale de non-appel (R-04)	Décision commerciale (inscription au registre et abonnement à la liste officielle) ; en attendant, ne pas utiliser le tenant de prospection à froid	Décision du propriétaire	—	Conformité documentée avant toute prospection sortante

P1 — Importants

ID	Problème (réf.)	Travail	Dép.	Risque	Résultat attendu
P1-1	Détection de répondeur sur la voie duplex (R-02)	Détection asynchrone (sans retarder la salutation) ; message « rappelez ce numéro » selon la politique du tenant ; reçu de l'issue	Fournisseur de téléphonie	Faux positifs coupant un humain	Appels en boîte vocale traités correctement, mesurés
P1-2	Pas de transfert vers un humain (R-03)	Transfert conditionnel (heures, disponibilité) avec reçu typé ; message honnête sinon	Numéro de destination par tenant	Appel bloqué si destination occupée	Sortie vers un humain, prouvée par un reçu
P1-3	Injection de prompt (R-25)	Marquage de provenance, séparation instructions/données, confirmation humaine des effets de bord, banc adverse	—	Faux refus	Réduction mesurée du succès des attaques
P1-4	Ancrage et audit (R-10, R-11, R-20)	Activer l'ancrage RFC 3161, vérifier la signature du jeton, conserver les ancres hors hôte ; `AUDIT_STRICT=true` ; agréger les `access_denied`	Accès Internet pour l'ancrage (hash seul)	Dépendance à des TSA publiques	Piste opposable ; bruit réduit
P1-5	Persistance (R-21, R-22, R-23)	`VACUUM` + `auto_vacuum`, table de migrations versionnée, checkpoint WAL avant copie, essai de restauration périodique, sauvegarde chiffrée activée	P0-2	Migration sur base de production	Sauvegarde cohérente et restaurée en essai
P1-6	Observabilité (R-29, R-14)	Alimenter les statistiques du cerveau dans les traces, rotation de logs, métriques persistantes, alertes externes, supervision et relance du worker de voix	—	Bruit d'alertes	Diagnostic de tendance possible ; worker relancé automatiquement

ID	Problème (réf.)	Travail	Dép.	Risque	Résultat attendu
P1-7	File d'inférence non préemptive (R-18)	Annulation des travaux `batch` à l'arrivée d'un `realtime` ; planification des routines hors des heures d'appel	—	Travaux interrompus	Silence au téléphone supprimé pendant les routines
P1-8	Anglais non mesuré (R-12)	Corpus d'appels anglais, mêmes mesures qu'en français	Corpus	—	Qualité EN chiffrée

P2 — Améliorations

ID	Problème (réf.)	Travail	Dép.	Risque	Résultat attendu
P2-1	Récupération de mémoire (R-19, §25 U9)	Index ANN, vectorisation du contenu métier, jeu d'évaluation, décision sur LightRAG (R-30)	Annotateurs	Régression de rappel	Rappel@5 mesuré ; capacité de graphe rétablie ou retirée
P2-2	Serveurs MCP sans attestation (R-27)	Attestation `tenant_safe` + test d'isolation par serveur, ou désactivation en multi-tenant	—	Perte de fonctions	Chaque serveur isolé ou désactivé
P2-3	Intégrations non validées (R-31)	Valider (Google, courriel sortant, WhatsApp, Shopify, GA4, Ads) ou retirer de l'offre	Comptes de test	—	Promesse = preuve
P2-4	Calcul d'argent sans garde (R-28)	Étendre l'intercepteur déterministe ; interdire l'énoncé de montants non calculés	—	—	Montants vérifiables
P2-5	ASR en français québécois (R-32, §25 U3)	Banc téléphonique FR-CA, biais lexical, confirmation orale des éléments critiques	Données consenties	Cadre Loi 25	Exactitude des entités critiques mesurée
P2-6	Dette de maintenabilité (§24)	Découper les fichiers > 1 500 lignes, retirer le code orphelin (Pipedrive, `llama.cpp.new`, copie *legacy*), unifier la langue des commentaires, nettoyer le disque	—	Régressions	Surface réduite, revue externe facilitée

P3 — Futur

ID	Problème (réf.)	Travail	Résultat attendu
P3-1	Naturalité du dialogue (§25 U1, U4)	Gestion de tour apprise, *backchannels*, évaluation d'un duplex ouvert	Décision d'architecture fondée sur mesures
P3-2	Haute disponibilité (R-13)	Second hôte froid, reprise documentée (RTO/RPO)	Reprise après sinistre éprouvée
P3-3	Secrets et immuabilité (R-08, R-10)	Magasin de secrets matériel (HSM/TPM), support WORM pour la piste	Preuve opposable à l'opérateur
P3-4	Repli sans nuage (§25 U6)	Modèle de secours local léger pour les tours déterministes	Service dégradé annoncé en cas de panne du cerveau
P3-5	Indice de qualité (§25 U11)	Panel d'écoute, juge local calibré	Comparaison objective entre versions

28. Conclusion

Ce que MIKE est aujourd'hui. Un système local-first cohérent et bien instrumenté (deux mois d'historique Git) : un orchestrateur de 135 000 lignes, un cerveau de 35 milliards de paramètres servi à 67 jetons/s sur une carte grand public, une cascade vocale qui a tenu des appels réels, un agenda canonique idempotent, un contrôle d'accès vérifié mécaniquement sur 283 routes et une piste d'audit signée dont l'intégrité se vérifie en 1,3 s. Les décisions de conception sont argumentées et, fait rare, **les limites sont écrites dans le code** (promesses refusées sans reçu, chiffrement délégué et vérifié plutôt que simulé, ancrage externe reconnu comme condition d'opposabilité).

Ce qui est prouvé. La suite de 5 641 tests passe ; la chaîne d'audit vivante est intègre et détecte la falsification ; le cerveau local est mesuré ; des appels réels ont abouti ; la frontière de tenant et les rôles de routes sont verrouillés par des tests.

Ce qui ne l'est pas. La latence vocale (médiane 2,9 s contre 1,0 s visé), la capacité à plusieurs appels, la qualité de l'anglais et de la reconnaissance du français québécois, l'installation propre du kit, le chiffrement des données au repos, l'isolation sous charge multi-tenant et les intégrations tierces. Le système est un **prototype fonctionnel avancé** — niveau de maturité 3 sur 5 — adapté à une démonstration supervisée et à un pilote encadré, **pas** à une exploitation sans surveillance chez un client.

Ce qu'il faut faire d'abord. Les huit éléments P0 de la feuille de route : rotation des secrets exposés et durcissement des droits, chiffrement de volume prouvé, élimination du repli sur le tenant `default`, mesure de la concurrence avec contrôle

d'admission, réduction mesurée de la latence, installation propre et signée, profil d'essai « client » et décision sur la liste nationale de non-appel.

Ce qui relève de la recherche. Onze incertitudes (§25) — naturalité et latence du dialogue vocal local en français québécois, reconnaissance vocale téléphonique, gestion du tour de parole, concurrence sur matériel grand public, inférence MoE locale stable, isolation et équité entre clients, sécurité des agents à outils, récupération de mémoire évaluée, véracité des actions, évaluation objective de la qualité. Elles justifient un cadrage en cinq programmes (§26) dont la valeur se démontre par des écarts **mesurés avant/après**, à partir des états initiaux consignés dans ce dossier.

Déclaration d'exactitude. Les chiffres de ce document proviennent de contrôles exécutés le 2 octobre 2026 ou d'extractions des journaux réels, chacun relié à une preuve du registre (Annexe A). Tout chiffre issu d'un document du dépôt est marqué *(source : document, non re-vérifié)*. Ce qui n'a pas pu être mesuré est écrit **NON MESURÉ**.

29. Annexes techniques

Annexe A — Registre des preuves techniques

Lecture : chaque identifiant `EV-nnn` est cité dans le corps du document. *Statut* : **Mesuré** = valeur obtenue par exécution pendant l'audit ou par extraction de données d'exploitation ; **Vérifié** = constaté par lecture du code, de la configuration ou d'une réponse de service ; **Implémenté et testé** = comportement exercé par au moins un test automatisé de la suite (5 641 réussis le 2 octobre 2026) ; **Partiel / Absent / Non concluant** = comme indiqué. Les fichiers de sortie sont dans `F51_MIKE_Dossier_Evidence/raw/`, les scripts de mesure dans `F51_MIKE_Dossier_Evidence/build/`.

ID	Affirmation	Fichier	Fonction / classe	Test / commande	Statut
EV-001	L'ouïe est alimentée en continu, y compris pendant que MIKE parle	`core/server/routers/voice_stream.py` ; `core/voice/mike_voice_worker.py`	`voice_stream()` (évènement `media` → `session.feed`) ; `_WorkerConnection.feed`	`test_voice_stream.py`, `test_voice_duplex.py` ; suite complète	Implémenté et testé
EV-002	Barge-in par plan d'arrêt : parole soutenue ≥ 0,6 s ou texte non-acquiescement ; acquiescements ignorés	`voice_stream.py` ; `mike_voice_worker.py` ; `mike_voice_duplex.py`	`_pump_worker_events._barge` ; `_WorkerConnection._poll`, `_speech_sustained` ; `stop_speaking_voice_s()`	`test_voice_stop_speaking_plan.py`	Implémenté et testé (unitaire) ; durée en appel réel non éprouvée
EV-003	Whisper sans streaming : `step()` et `partial()` renvoient `""`, `endpoint()` renvoie `False`	`core/voice/mike_stt_whisper.py`	`WhisperStream`, `WhisperStreamingSTT.transcrever`	Lecture du code	Vérifié
EV-004	Transcription anticipée sur pause ≥ 0,2 s, pool de 2 threads	`core/voice/mike_ears.py`	`StreamingEars._start_speculation`, `_take_speculation`, `_speculation_pool`	`voice_trace_from_logs.py` : `asr_antecipado` vrai pour 78 tours sur 78	Mesuré
EV-005	Fin de tour : silence 320 ms, Smart Turn, délai court 0,8 s, filet 2,4 s, critère lexical fermé	`mike_ears.py`	`StreamingEars.poll`, `closed_answer`, `_silence_limit_seconds`, `endpoint_silence_seconds`	Traces : `closed_by` Smart Turn 66, silence 12 (depuis le 29/09)	Mesuré
EV-006	Une seule synthèse Kyutai à la fois (verrou autour de `generate`)	`core/voice/mike_tts_kyutai.py`	`KyutaiTTSBackend._gerar`, `stream_mulaw_frames`	Lecture ; `test_tts_kyutai.py`	Vérifié
EV-007	Verrou de capture CUDA partagé Whisper/Kyutai	`core/voice/mike_gpu_capture.py`	`install()`, `gpu_section()`	`test_voice_gpu_capture.py` ; 0/30 échec en banc (source : `hardware.md` §10.3.1)	Implémenté et testé
EV-008	Aucun plafond de sessions simultanées	`mike_voice_duplex.py` ; `mike_voice_worker.py`	`DuplexEngine.start_session` ; route `/v1/audio`	Recherche négative (`MAX_SESSIONS`, `max_sessions`, `MAX_CONCURRENT`) dans `core/voice` et `core/server`	Absent ; capacité non mesurée
EV-009	Fin de parole → premier son : médiane 2 919 ms, p95 5 827 ms (78 tours, 20 sessions)	`logs/mike.log`	trace `voice.turn.trace stage=twilio_media`	`voice_trace_from_logs.py` → `raw/voice_trace_logs.json`	Mesuré
EV-010	Tours déterministes (n = 34) 1 441 / 2 951 ms ; tours avec modèle (n = 44) 3 482 / 6 202 ms	idem	champ `llm_required`	Calcul sur les mêmes traces (sortie consignée)	Mesuré
EV-011	177 lignes `VOICE_LATENCY` du 21/09 au 02/10	`logs/mike.log`	ligne `VOICE_LATENCY`	`voice_latency_from_logs.py` → `raw/voice_latency_logs.json`	Mesuré
EV-012	File d'inférence à 3 classes (`realtime`, `interactive`, `batch`), 1 créneau, sans préemption	`core/integrations/mike_llama_server_client.py`	`_acquire`, `slot`, `BUDGET_CLASSES`, `_URGENCY`	`test_inference_admission.py`, `test_budget_class_and_routine_lock.py` ; aucune annulation dans le client	Implémenté et testé

ID	Affirmation	Fichier	Fonction / classe	Test / commande	Statut
EV-013	Garde de promesses parlées (4 catégories, reçus typés)	`core/voice/mike_voice_claims.py`	`SpokenClaimFilter`, `reply_allowed_claims`, `guard_reply_text`	`test_voice_claims.py`, `test_voice_transport_claims.py`	Implémenté et testé
EV-014	Cascade (pas de full-duplex natif) ; MIKE ne génère pas de *backchannel*	`voice_stream.py` ; `core/server/mike_voice_flow.py`	`is_backchannel` (détection seulement), `_idle_watch`	Recherche `backchannel` : détection uniquement	Vérifié
EV-015	WebSockets : `/v1/voice/stream` (actif), `/v1/voice/relay` (désactivé), worker `/v1/audio`	`voice_stream.py:71` ; `routers/voice_relay.py:55` ; `mike_voice_worker.py:1440`	`WS_PATH`, `@app.websocket`	Recherche ; `MIKE_VOICE_RELAY_ENABLED=0`	Vérifié
EV-016	Détection de répondeur : option synchrone désactivée, lue seulement par l'IVR de rappel ; absente de la voie duplex	`core/comms/mike_twilio.py` ; `mike_twilio_webhooks.py` ; `mike_voice_preflight.py`	`machine_detection_mode`, `initiate_call`, lecture de `AnsweredBy` (l.739), `gate_amd_policy`	Lecture ; variable `MIKE_TWILIO_MACHINE_DETECTION` non définie (défaut `off`)	Partiel
EV-017	Aucun transfert d'appel vers un humain	`mike_voice_claims.py`	`LIVE_TRANSFER`, `_TRANSFER_CLAIM_RE`	Recherche `<Dial`, `.dial`, `<Refer`, `conference` dans `core/comms` et `core/voice` : aucune	Absent
EV-020	Configuration du `llama-server` : Ornith 1.5 35B-A3B Q4_K_M, ctx 81 920, KV q4_0, `--parallel 1`, `--n-cpu-moe 24`, build b10826	processus `llama-server.exe`	`/props`, `/v1/models`, ligne de commande	`raw/health_llama_props.txt`, `health_llama_models.txt` (13:09)	Vérifié
EV-021	Décodage 67,1 jetons/s ; préremplissage 626 jetons/s ; premier jeton 925 ms (sans cache) / 77 ms (cache)	`llama-server` :8081	`timings` du serveur	`bench_llm.py` → `raw/bench_llm.json` (13:11–13:12)	Mesuré
EV-022	Chaîne de repli à un seul backend réel ; nuage refusé	`core/server/mike_lifecycle.py`	`assert_llm_backend_local_only`, chaîne de repli	`test_single_brain.py`	Implémenté et testé
EV-023	Jusqu'à 11 tâches de fond ; chien de garde du cerveau	`mike_lifecycle.py`	`_bg_tasks.append(…)` (l.1719–1736), `_brain_watchdog_loop`	Lecture	Vérifié
EV-024	`/health`, `/livez`, `/readyz` : 200, 33 routers, voix prête, LightRAG indisponible	MIKE :8083	endpoints de santé	`raw/health_mike_health.txt`, `health_mike_readyz.txt` (13:09)	Vérifié
EV-025	Ports en boucle locale 8081, 8083, 8998 ; processus et mémoire	—	—	`Get-NetTCPConnection`, `Get-Process` (13:06)	Mesuré
EV-026	Matériel : i7-12700K, 63,7 Go, RTX 5060 Ti 16 Go + RTX 3060 12 Go, pilote 595.97	—	—	`Win32_Processor`, `Win32_ComputerSystem`, `nvidia-smi` (13:06)	Mesuré
EV-027	Worker de voix dans WSL : Python 3.12.8, torch 2.8.0+cu129, sherpa-onnx 1.13.7, RSS ≈ 4,3 Go	`/opt/mike-voice`	—	Sonde `wsl -d Ubuntu-24.04`	Mesuré
EV-030	Époque d'audit vivante : 10 518 enregistrements, chaîne SHA-256 et signatures Ed25519 valides	`core/audit/mike_audit.py`	`AuditLedger.verify`	`verify_ledger_copy.py` → `raw/audit_verify.json` (13:13)	Mesuré
EV-031	Falsification d'un champ à `seq` 5 260 détectée	idem	idem	idem (test de falsification sur copie)	Mesuré
EV-032	Époque héritée rompue (HMAC `seq` 40 ; chaîne `seq` 27 741)	`runtime/audit/epochs/audit-epoch-20260923-01.report.json`	—	Lecture du rapport	Vérifié
EV-033	Arbre de Merkle, preuve d'inclusion, `checkpoint()`	`core/audit/mike_merkle.py`	`merkle_root`, `inclusion_proof`, `verify_inclusion`	`test_audit_anchor_*`	Implémenté et testé

ID	Affirmation	Fichier	Fonction / classe	Test / commande	Statut
EV-034	RFC 3161 : désactivé, 0 ancre, jeton non vérifié cryptographiquement par le code	`core/audit/mike_anchor.py`	`_parse_tsr_status`, `request_timestamp`, `verify_against_anchors`	`runtime/audit/anchors.jsonl` absent ; `MIKE_AUDIT_ANCHOR_ENABLED` non défini	Vérifié
EV-035	`MIKE_AUDIT_STRICT=false`	`config/.env.runtime` ; `core/audit/mike_audit_runtime.py`	`audit_strict()`	Lecture	Vérifié
EV-036	88 % des enregistrements = `access_denied` ; fichier unique pour tous les tenants	`raw/audit_verify.json`	—	idem EV-030	Mesuré
EV-040	Contexte de tenant immuable par `ContextVar` ; fabrique de services LRU ; mode strict	`core/server/mike_tenant.py`	`TenantContext`, `tenant_scope`, `TenantRuntimeFactory`, `NoTenantContext`	26 fichiers de tests « tenant »	Implémenté et testé
EV-041	Repli `default_tenant()` : 51 occurrences dans 26 fichiers	`core/**`	`default_tenant()`	Recherche textuelle	Mesuré
EV-042	283 routes déclarées avec rôle ; `KNOWN_UNENFORCED` vide ; RBAC à 6 rôles	`core/server/mike_route_access.py` ; `mike_rbac.py`	`ROUTE_ACCESS`, `KNOWN_UNENFORCED`	Import Python (client 88 · public 50 · opérateur 135 · local 10) ; `test_route_access_registry.py`	Mesuré + testé
EV-043	Signature Twilio vérifiée sur 5 webhooks	`core/comms/mike_twilio_webhooks.py`	`_validate_twilio_signature` (l.714, 796, 948, 1139, 1258)	`test_twilio_security.py`	Implémenté et testé
EV-044	Jeton HMAC du stream, consentement `external_connectors` exigé avant la session	`voice_stream.py`	`stream_token_for`, `_verify_stream_token`, `_external_connectors_granted`	`test_voice_stream.py`	Implémenté et testé
EV-045	PBKDF2-HMAC-SHA-256 à 600 000 itérations ; sessions HMAC avec `aud` et `jti`	`core/server/mike_auth.py`	`PBKDF2_ITERATIONS`, `_burn_password_work`	Lecture	Vérifié
EV-046	Chiffrement au repos délégué à BitLocker et vérifié par le code ; BitLocker désactivé sur C: et sur la clé USB	`core/server/mike_encryption.py`	`encryption_status`, `enforcement_mode`	`System.Volume.BitLockerProtection` = 2 pour C: et E: ; `MIKE_REQUIRE_DISK_ENCRYPTION=warn`	Mesuré
EV-047	ACL permissives sur les secrets (lecture `CodexSandboxUsers`, modification pour 2 SID)	`config/.env.runtime.local` ; `runtime/auth`	—	`Get-Acl`	Mesuré
EV-048	Garde-fous désactivés sur le poste audité (consentement, assurance, confinement, rétention)	`config/.env.runtime.local`	`MIKE_CONSENT_ENFORCED=0`, `MIKE_ASSURANCE_ENFORCED=0`, `MIKE_CONTAINMENT_ENABLED=0`, `MIKE_RETENTION_ENABLED=0`	Lecture des drapeaux	Vérifié
EV-049	Confinement des chemins, de la commande et du délai des outils de workspace	`core/mcp/mike_workspace_mcp.py`	`resolve_safe_path`, `run_command`	`test_command_confinement.py`	Implémenté et testé
EV-050	Injection de prompt : garde textuelle seulement, pas d'assainissement des sorties d'outils	`voice_stream.py`	`_ANTI_INJECTION_GUARD`	Recherche (`injection`, `untrusted`, `sanitize`)	Partiel
EV-051	En-têtes de sécurité (CSP, X-Frame-Options, HSTS conditionnel), limitation de débit globale	`core/server/mike_server_middleware.py` ; `mike_server.py`	middlewares, `SlowAPIMiddleware`	Lecture	Vérifié
EV-052	Aucun fichier secret suivi par Git ; `.env.*` ignoré ; scan de secrets au commit	`.gitignore` ; `scripts/ops/secret_scan.py`	—	`git ls-files` (aucun), `git check-ignore`	Mesuré

ID	Affirmation	Fichier	Fonction / classe	Test / commande	Statut
EV-053	Champ `pin` dans les `workspace.json` d'exemple, non lu par le code	`runtime/workspaces/qc_*/workspace.json`	—	Recherche `"pin"` dans `core/` : seul le PIN d'opérateur	Constat
EV-060	Pré-vol d'appel P1–P7	`core/voice/mike_voice_preflight.py`	`gate_consent`, `gate_dncl`, `gate_calling_window`, `gate_attempts`, `gate_did`, `gate_amd_policy`, `gate_attestation`	`test_voice_preflight_and_phrases.py`, `test_call_attestation.py`	Implémenté et testé
EV-061	Liste d'exclusion locale ; liste nationale du CRTC non consultée	`mike_voice_preflight.py`	`load_dncl`, `gate_dncl`	Lecture (docstring)	Vérifié
EV-062	Consentement par finalité (6 finalités), effacement et rétention	`core/server/mike_privacy.py`, `mike_erasure.py`, `mike_retention.py`	constantes de finalités ; `is_granted`	Tests de confidentialité ; application désactivée ici (EV-048)	Implémenté et testé
EV-070	36 fichiers SQLite ; 26 espaces de travail ; tailles et tables	`runtime/**`	—	`inventory.py` → `raw/inventory.json` (13:08)	Mesuré
EV-071	64 559 segments (96,7 % romans, 0 vecteur) ; 216 vecteurs hors livres ; 71,2 % de pages libres	`runtime/workspaces/default/memory/mike_memory.db`	—	`db_profile.py` → `raw/db_profile.json`	Mesuré
EV-072	Recherche vectorielle en balayage linéaire (SELECT sans LIMIT + produit scalaire en Python)	`core/memory/mike_memory_store.py`	`search_knowledge_vector`	Lecture	Vérifié
EV-073	LightRAG « activé mais non chargé »	—	—	`/readyz` `capabilities.knowledge`	Vérifié
EV-074	Agenda canonique avec révisions, idempotence, outbox ; livraison transactionnelle	`runtime/workspaces/*/memory/appointments_canonical.db`, `deliveries.db`	tables	Requêtes SQL `mode=ro` ; tests d'agenda	Mesuré + testé
EV-075	Aucune rotation de journaux dans le chemin par défaut	`core/**` ; `scripts/ops/install_mike_service.ps1`	—	Recherche `RotatingFileHandler` (aucune) ; NSSM `AppRotateBytes`	Vérifié
EV-076	Métriques : anneau de 512 échantillons par étape, en mémoire	`core/server/mike_latency.py`	`observe`, `prometheus_text`	Lecture	Vérifié
EV-080	Suite complète : 5 641 réussis, 4 ignorés, 1 xfail, 245,5 s	`tests/unit`	—	`pytest -q …` → `raw/pytest_full.txt` (13:12–13:16, commit 6f2d858)	Mesuré
EV-081	376 fichiers de test, 4 468 fonctions, répartition par domaine	`tests/`	—	`tests_by_domain.py` → `raw/tests_by_domain.json`	Mesuré
EV-082	Vérificateur de prontidité bloqué sur le PIN d'opérateur	`scripts/ops/check_mike_readiness.py`	—	`raw/readiness_check.txt`	Non concluant
EV-083	Appels réels antérieurs, dont un avec réservation écrite (12 tours scellés)	`logs/mike.log` ; `docs/delivery/evidence/2026-10-02-recado-e-ficha-no-runtime.md`	traces `CAfd9940…`	Lecture des journaux et du document	Observé (non reproduit)
EV-084	Kit installateur : compilé, 8 SHA-256 vérifiés, jamais installé, exécutable non signé	`build/installer/output/` ; `docs/delivery/evidence/2026-10-02-entrega-preview.md`	—	`Get-AuthenticodeSignature` = `NotSigned` ; `Get-FileHash`	Mesuré

ID	Affirmation	Fichier	Fonction / classe	Test / commande	Statut
EV-085	1 002 commits ; volumes hors code dans le dossier de travail	dépôt	—	`git rev-list --count HEAD` ; tailles de dossiers	Mesuré
EV-086	20 serveurs MCP (2 désactivés), 117 outils, `tenant_safe` déclaré par 4 serveurs	`config/mcp_servers.json` ; `core/mcp/*.py`	`@mcp.tool`	`inventory.py`	Mesuré
EV-087	Code : 304 modules Python (134 960 lignes) dans `core/` , 90 824 lignes de tests, 268 décorateurs de route	`core/**`	—	`inventory.py` → `raw/inventory.json`	Mesuré
EV-088	La fiche de l'entreprise est injectée dans le contexte de voix et de chat	`core/chat/mike_context.py` ; `core/server/mike_request_helpers.py`	`_business_prompt_block` , `tenant_facts_block` , `_light_system_prompt`	`test_chat_business_context.py` , `test_business_profile_routes.py`	Implémenté et testé
EV-089	Kill-switch fail-closed (commande, fichiers, TTS, stream de voix)	`voice_stream.py` ; `mike_workspace_mcp.py` ; `mike_tts_kyutai.py`	`_kill_switch_block_reason` , `assert_not_frozen`	17 fichiers de tests mentionnant le kill-switch	Implémenté et testé
EV-090	Le tenant qui répond à un numéro est lu dans `workspace.json` (clé `phone`) ; deux déclarants = aucun ne répond	`core/server/routers/telephony.py` ; `scripts/ops/tenant_line.py`	`_telephony_owner_tenant` ; `owners_of` , `move_line`	`test_tenant_comercial_mike.py`	Implémenté et testé

Total : 67 preuves enregistrées.

Annexe B — Commandes et scripts exécutés

#	Commande ou script	Heure (EDT)	Résultat
1	Inventaire matériel et réseau : `Win32_Processor` , `nvidia-smi --query-gpu=…` , `Get-NetTCPConnection` , `Get-Process`	13:06	Matériel, ports, processus
2	`build/inventory.py`	13:08	`raw/inventory.json`
3	`GET` de dix endpoints de santé	13:09	`raw/health_*.txt` (6 × 200 ; 4 × 401)
4	Lecture de la ligne de commande du `llama-server` (sans clés)	13:09	Paramètres consignés
5	`build/bench_llm.py`	13:11–13:12	`raw/bench_llm.json`
6	`pytest -q --basetemp … -p no:cacheprovider --durations=12`	13:12–13:16	5 641 réussis (`raw/pytest_full.txt`)
7	`build/verify_ledger_copy.py` (copie de la piste + test de falsification)	13:13	`raw/audit_verify.json`
8	`scripts/ops/check_mike_readiness.py`	13:16 → arrêt	Bloqué sur le PIN ; non concluant
9	`build/voice_latency_from_logs.py`	13:23	`raw/voice_latency_logs.json`
10	`build/voice_trace_from_logs.py`	13:24	`raw/voice_trace_logs.json`
11	`build/db_profile.py`	13:41	`raw/db_profile.json`
12	`build/tests_by_domain.py`	13:46	`raw/tests_by_domain.json`
13	Sonde WSL (processus, GPU, disque, versions) ; propriété de shell BitLocker pour C: et E: ; `Get-Acl` ; `git ls-files` , `git check-ignore`	entre 13:06 et 14:00 (relevés successifs)	Voir EV-025, EV-027, EV-046, EV-047, EV-052
14	`make_diagrams.py` , `render_png.py`	entre 13:30 et 14:05 (régénérés après corrections)	9 diagrammes SVG/PNG

Annexe C — Inventaire de référence

Code (lignes) EV-087 : `core/` Python 134 960 · `scripts/` Python 17 833 et PowerShell 11 314 · `dashboard/` JS 15 993, HTML 4 153, CSS 7 936 · `tests/` Python 90 824 · `skills/` YAML 2 502 · `docs/` Markdown 35 083.

Routes : 268 décorateurs (GET 152, POST 103, PUT 5, PATCH 2, DELETE 6), 3 WebSockets (dont un désactivé et un côté worker) ; préfixes principaux : `/v1/autonomy` 34, `/v1/client` 15, `/v1/auth` 15, `/v1/operator` 11, `/v1/privacy` 11, `/v1/agenda` 10, `/v1/admin` 10, `/v1/memory` 10, `/v1/onboarding` 10, `/v1/skills` 9, `/v1/missions` 8, `/v1/data` 8, `/v1/health` 8. Fichier le plus chargé : `routers/autonomy.py` (72).

Outils MCP par fichier (117) : GA4 14 · prix 14 · Shopify 14 · workspace 13 · Google Ads 11 · rendez-vous 10 · calendrier 7 · Excel 7 · HuggingFace 6 · agenda 4 · Drive 4 · agent distant 4 · scraping web 4 · pont MCP 3 · docgen 2.

Configuration : 152 variables dans `.env.runtime` (37 dans `.env.runtime.local`, 145 dans `.env.example`) ; 511 noms `MIKE_*` référencés dans `core/` ; noms sensibles présents (valeurs non reproduites).

Compétences et agents : 59 compétences YAML ; 62 agents Claude Code décrits.

Annexe D — Écarts entre la documentation et la réalité observée

D.1 Divergences vérifiées

#	Document	Affirmation	Constat	Preuve
1	`docs/arquitectura/hardware.md` §2.0	Premier jeton 167 ms (prompt de 250 jetons)	925 ms sans cache de prompt, 77 ms avec ; écart non expliqué	EV-021
2	Description de l'agent `mike-skills`	« 54 skills »	59 fichiers YAML dans `skills/`	EV-087
3	Compétence `entrega-produto` (`.claude/skills/`)	« `LICENSE` n'existe pas dans le disque »	`LICENSE` existe (2 190 octets)	contrôle d'existence
4	`CLAUDE.md` (exception de développement)	Les quatre garde-fous sont « actifs » dans une installation client	Documenté : désactivés sur ce poste par `.env.runtime.local` ; ce n'est donc pas un écart, mais les mesures ne représentent pas la configuration client	EV-048

D.2 Affirmations de la documentation confirmées par le disque ou l'exécution

Ports 8081/8083/8998 et liaison à la boucle locale · cerveau Ornith 1.5 35B-A3B Q4_K_M, contexte 81 920, KV `q4_0`, `--n-cpu-moe 24`, `--reasoning off` · ASR Whisper et TTS Kyutai par défaut · VAD Silero sur CPU · worker de voix en WSL2 Ubuntu-24.04 · CPython 3.12.10 pour le `.venv` · 59 compétences · 117 outils MCP et 20 serveurs · 62 agents · 283 routes déclarées par rôle · flavor `development` sans signature de manifeste · ancrage RFC 3161 et sauvegarde hors-site désactivés par défaut · `MIKE_MULTI_TENANT=1` par défaut.

Annexe E — Composants qui n'ont pas pu être validés

1. Installation propre du kit, démarrage à froid, temps de démarrage, retour arrière.
2. Appels simultanés, charge, endurance, reprise après panne du worker en plein appel.
3. Isolation de deux tenants sous appels réels concurrents.
4. Envois réels (SMS, courriel, WhatsApp) ; intégrations Google, Shopify, GA4, Ads.
5. Chaîne complète en anglais.
6. Qualité de récupération (RAG) et de LightRAG.
7. Endpoints authentifiés de latence, VRAM, TTS, métriques (aucune session).
8. Taux de faux positifs du *barge-in* ; RTF de la synthèse ; utilisation CPU en charge.
9. Verdict du vérificateur de prontidité du projet (PIN requis).
10. Pourcentage de couverture de code.
11. Confirmation de l'état BitLocker par `manage-bde` (droits d'administrateur refusés ; indication par propriété de shell seulement).
12. Tests d'intégration et de bout en bout (démarrent des services).

Annexe F — Glossaire

ASR reconnaissance automatique de la parole · **AMD** détection de répondeur · **Barge-in** interruption de MIKE par le client · **Backchannel** acquiescement bref (« mm-hm ») · **Cascade** enchaînement de modèles séparés · **CRTC** Conseil de la radiodiffusion et des télécommunications canadiennes · **DID** numéro de téléphone entrant/sortant · **FTS5** recherche plein-texte de SQLite · **KV (cache)** mémoire d'attention du modèle · **LLM** grand modèle de langage · **MCP** *Model Context Protocol* · **MoE** *mixture of experts* · **PIA** évaluation des facteurs relatifs à la vie privée · **RFC 3161** horodatage par tiers de confiance · **RSS** mémoire résidente d'un processus · **TTFA** délai jusqu'au premier audio · **TTFT** délai jusqu'au premier jeton · **TTS** synthèse vocale · **VAD** détection d'activité vocale · **WAL** journal d'écriture anticipée de SQLite.