Sélène

Une intelligence qui tient dans une machine grand public

Introduction au livre d'or — 11 février au 9 octobre 2026

Le pari

Sélène est une intelligence artificielle vocale qui tourne entièrement sur l'ordinateur de celui qui l'utilise. Pas une ligne ne part vers un serveur. Pas d'abonnement, pas de clé d'API, pas de télémétrie, pas de connexion requise. On parle, elle écoute, elle réfléchit, elle répond avec sa voix — et tout cela se passe à l'intérieur de la machine posée sur le bureau.

Ce n'est pas une préférence technique, c'est le point de départ. L'objectif est que n'importe qui puisse avoir chez lui une IA personnelle qui lui appartient vraiment, sur le matériel qu'il possède déjà. Une machine grand public, pas une ferme de calcul.

Et c'est de là que vient toute la difficulté de ce livre.

Parce qu'une machine grand public, cela veut dire un petit modèle. Les grands modèles que tout le monde connaît pèsent des centaines de milliards de paramètres et vivent dans des centres de données. Ce qui tient dans une carte graphique de salon, c'est quelques milliards. Le pari de Sélène tient en une phrase : **obtenir d'un petit modèle l'autonomie, la liberté, la cohérence et l'intelligence qu'on croit réservées aux grands.** Tout ce qui suit est le récit de cette tentative, avec ses impasses.

Première réponse : encadrer le modèle

La première intuition était la bonne au mauvais endroit. Si le modèle est petit, on compense par l'architecture : on construit autour de lui un cerveau en Python qui décide à sa place, et on ne lui laisse que la parole. Le principe est écrit noir sur blanc dans un document du 24 février 2026 :

> Le modèle linguistique n'est pas l'intelligence. Il est l'organe d'expression.
>
> Le cerveau scriptural décide.

Cette architecture n'était pas un brouillon. Elle était sophistiquée, pensée, mesurée. Un état cognitif de dix-sept variables réécrit à chaque échange. Cinq pulsions internes en homéostasie lente. Un moteur de décision qui choisissait l'acte, un moteur d'intention qui lisait le contexte social, un moteur de saillance qui notait l'importance de chaque message. Des *guardians* qui surveillaient le tout en arrière-plan et corrigeaient les dérives toutes les cinq minutes. Une mémoire vectorielle. Un second modèle, DeepSeek R1, chargé du raisonnement pendant que le premier parlait.

La carte ci-dessous est celle du 7 mars 2026. Elle se lit comme un plan de métro : les points d'entrée en haut, le cerveau cognitif, le noyau régulateur, les modules fonctionnels, les données persistantes.



7 mars 2026 — la carte cognitive au moment où l'architecture Python est encore élégante : points d'entrée, cerveau cognitif, noyau régulateur, modules fonctionnels, données persistantes.

Le point haut, et le mur

Sept semaines plus tard, le 25 avril, la même carte affiche en en-tête les chiffres de ce qu'elle était devenue : **177 fichiers Python, plus de 40 JSON persistants, 7 guardians, 3 modèles de langage intégrés, six mois de travail.**

Et le système tournait. C'est ce qu'il faut comprendre : il n'est pas tombé en panne. Il faisait exactement ce pour quoi il avait été conçu.

Le problème est qu'il le faisait trop bien. À force de décider à la place du modèle, l'architecture ne lui laissait plus rien à décider. Chaque défaut observé appelait une règle de plus, chaque règle réduisait un peu l'espace dans lequel le modèle pouvait encore être lui-même. On a vu apparaître un verrou comportemental : des réponses courtes, évasives, méta — « Mmh », « C'est comme ça » — produites par un modèle parfaitement capable, mais enfermé. Un seuil d'engagement fixé à 0,65 empêchait de basculer sur un sujet nouveau et sérieux ; le système repartait indéfiniment sur l'ancien. Les corrections s'empilaient sur les corrections.

Le constat est posé : six mois d'architecture pensée et testée, qui n'a pas fonctionné. Non pas parce qu'elle était mal faite, mais parce que la vision était court-termiste. Encadrer un modèle, c'est gagner du contrôle tout de suite et perdre la liberté cognitive pour toujours. Un comportement orchestré en Python autour du modèle ne devient jamais un comportement du modèle.



25 avril 2026 — le point haut. L'en-tête annonce 177 fichiers Python, plus de 40 JSON persistants, 7 guardians, 3 modèles de langage, six mois de travail.

Le retournement

La leçon tirée est devenue la règle de conception du projet, et elle tient en une opposition : **ce qu'on installe autour du modèle échoue, ce qu'on installe dans le modèle fonctionne.**

Plutôt que de décrire à Sélène le caractère qu'elle devait avoir, il a fallu le lui apprendre. C'est le passage au fine-tuning, et la bascule a été lente : le premier entraînement sérieux date du 9 mai 2026, et l'architecture Python a mis des semaines à fondre. De 157 modules en mai, on tombe à 49 le 22 juin. Les trois cartes reproduites ici — mars, avril, juin — rendent cette courbe lisible.

Ce qui restait de Python n'était plus un tuteur cognitif mais un système nerveux périphérique : écouter, parler, se souvenir, exécuter. Le jugement, lui, avait déménagé dans les poids du modèle.

La pièce maîtresse de ce déménagement porte un nom modeste : la famille E1 à E8. Huit tensions entre deux valeurs également légitimes — l'ambition contre le réalisme, l'exploration contre la conclusion, la prudence contre l'action, la vérité contre le confort, l'importance contre l'intérêt, l'apparence contre l'essence, la loyauté contre l'intégrité, le court terme contre le long terme. On n'y enseigne pas de bonnes réponses : on y enseigne à arbitrer. C'est de là, selon l'auteur du projet, qu'a émergé le « je » interne du modèle.

WIKI SYSTÈME PLANÉTAIRE V5.0 — ARCHITECTURE ÉPURÉE — 22 JUIN 2026

SÉLÈNE

Architecture cognitive — vue planétaire allégée

49	3	4	33	29
MODULES ACTIFS	MAÎTRES CENTRAUX	ORBITES	CONNEXIONS	MODULES RETIRÉS

✅ ARCHITECTURE ÉPURÉE — 49 modules actifs / 29 retirés depuis la Phase 3 (22 mai → 21 juin 2026) | Production : Qwen 32B Q3_K_M

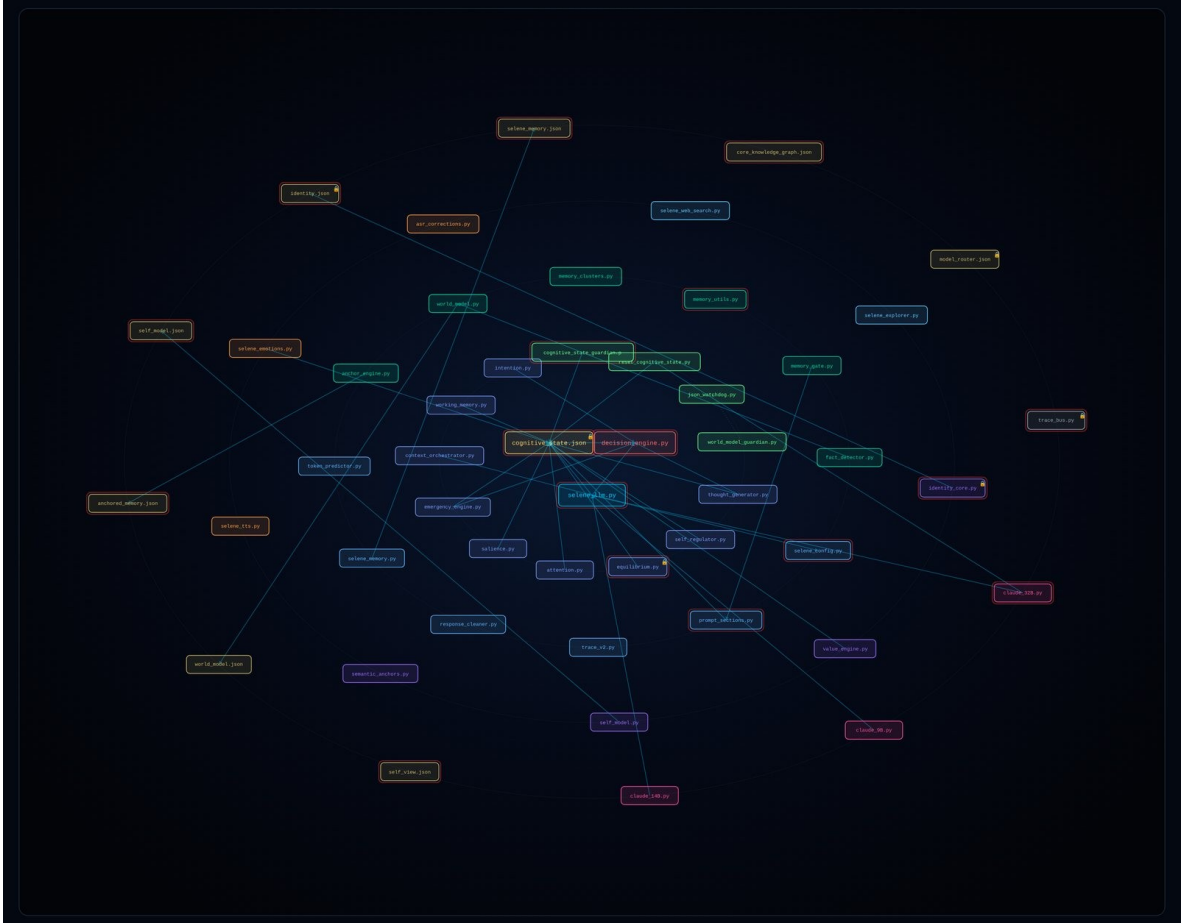
Système planétaire épuré. Au centre les trois maîtres : *cognitive_state.json*, *decision_engine.py*, *selene_llm.py*. Les 49 modules actifs gravitent en 4 orbites — plus proche du centre = plus critique. Les flèches montrent les connexions réelles : cyan = agit, violet tireté = analyse, vert pointillé = observe. 🔒 = ne pas modifier, halo rouge = critique. Cliquer sur un module pour les détails. **Les modules désactivés ne figurent plus dans ce schéma — ils sont listés séparément ci-dessous.**

MAÎTRES CENTRAUX
- cognitive_state — état
- decision_engine — acte
- selene_llm — pipeline

FAMILLES
- Cognition
- Pipeline
- Mémoire
- Guardians
- Identité

FAMILLE (SUITE)
- Interface (TTS/ASR)
- Externe (web/explorer)
- Launchers
- JSON persistants
- Observateurs

CONNEXIONS
- Agit (lit/écrit)
- Analyse
- 🔒 Ne pas modifier
- Critique (halo)



MODULES RETIRÉS / EN SOMMEIL

29 modules — Phase 3 (22 mai → 21 juin 2026)

Ces modules ne font plus partie du pipeline actif de Sélène. Ils ont été désactivés au fur et à mesure que le finetuning (32B / 9B) a démontré porter nativement les capacités qu'ils tentaient de compenser — l'architecture s'est allégée sans perte de comportement.

OBSOLÈTES (PHASE 3 — 22-26 MAI 2026) (8)

behavior_guardian.py *DANGEREUX — delta nul mesuré le 22 mai*	engagement_engine.py *Blacklist méta-réponses être BB — 32B finetuné gère nativement*	core_loop_controller.py *Boucle auto-réflexive jamais branchée*	proto_switch.py *Activation noyau redondante avec decision_engine*
core_nucleus.py *Noyau central — complexité gérée nativement par 32B*	token_estimator.py *Jamais branché au pipeline réel*	introspection_engine.py *Court-circuit LLM — redondant post-finetuning*	music_perception.py *Désactivé — provoquait une boucle XTTS*

REDONDANTS AVEC LA COGNITION FINETUNÉE (LE MODÈLE PORTE CES CAPACITÉS DANS SES POIDS) (10)

planner.py *Plans multi-étapes — redondant avec raisonnement natif 32B*	self_interpreter.py *Redondant avec self_model.py*	reflection_engine.py *Réflexion interne — redondante post-finetuning*	self_evaluation.py *Arbitrage plan/interruption — redondant*
humor_engine.py *Humour scripté — modèle finetuné le produit nativement*	reasoner.py *Raisonnement — redondant avec decision_engine V3.6*	reasoning_engine.py *Moteur avancé — redondant*	selene_self_improve.py *Auto-amélioration — jamais opérationnel*
knowledge_filter.py *Filtre légitimité web — redondant*	expressivity_engine.py *Niveau expression — géré nativement par le modèle*		

SYSTÈME OPINION (ARCHITECTURE 8B/2B COMPENSATOIRE — OBSOLÈTE) (4)

selene_opinion.py *_SOMMEIL=True — post_response() neutralisé*	opinion_manager.py *Client socket + boucle interne — non appelé*	opinion_match_v2.py *Match ES+tags — non appelé*	opinion_server / cognitive_server (2B) *Gemma 2B Q6_K — serveur non lancé*

THINKER 0.8B + CORTEX PARALLÈLE (SYSTÈME PARALLÈLE EN SOMMEIL) (6)

thinker_08.py *Cerveau secondaire 0.8B — non lancé*	cognitive_server.py (thinker) *Serveur socket port 5678 — non lancé*	interceptor.py *Interception messages → thinker — non lancé*	selene_cortex.py *ChromaDB + MiniLM — "Cortex vectoriel non chargé"*
cortex_query.py / cortex_init.py *Interface cortex — système parallèle en sommeil*	flux_input.json / flux_output.json (dual) *Fichiers résiduels vides — non alimentés*		

PERSONAS (COMPENSATION PETITS MODÈLES — OBSOLÈTE SUR 32B) (1)

persona_engine.py *_PERSONAS_ENABLED=False — Kirk/Spok/Chekov/Klingon non utilisés*

22 juin 2026 — après la coupe. De 157 modules à 49. Ce qui reste de Python n'est plus un tuteur cognitif mais un système nerveux périphérique.

Tout a été essayé

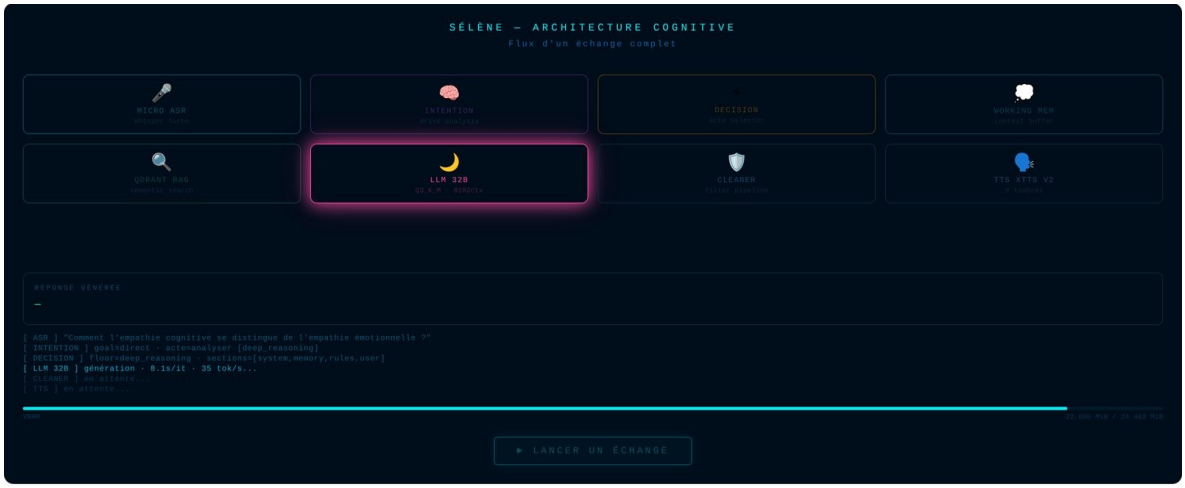
Une question revient toujours : pourquoi tant d'essais ? Parce qu'aucun petit modèle ne se comporte comme un autre une fois fine-tuné, et que le seul moyen de le savoir est de le faire. La base disponible au grand public a été passée en revue, entraînée, mesurée, auditée — pas survolée.

Modèles conversationnels passés au banc, tels que le journal les atteste :

Famille	Déclinaisons essayées	Ce qui en est ressorti
Qwen 2.5	7B, 14B, 32B Instruct, 72B évalué	Le 14B puis le 32B ont tenu la production plusieurs mois ; le 7B reste la cible commerciale
Qwen 3	4B, 8B	Le 4B est conservé pour l’exécution des outils ; le 8B est en cours de développement. Les modes « thinking » ont été écartés.
Qwen 3.5	9B, 27B	Le 9B entraîné puis jeté : architecture SSM, inutilisable après fusion
Qwen 3.6	27B	Essayé, non retenu
Gemma 2	2B, 9B instruct	Socle des premiers fine-tunings sérieux, versions v2 à v11
Gemma 4	12B-it, 26B A4B (MoE raisonnant), 31B	Le 12B est le modèle actuel ; le 26B raisonnant a fonctionné mais n'a jamais su s'arrêter de générer
DeepSeek	R1 Distill Qwen 7B	Le raisonneur de la première architecture, et plus tard le moteur d'opinion

Et la chaîne autour du modèle a été auditée de la même façon : Whisper en *small*, *medium*, *large-v3* et *large-v3-turbo* pour l'écoute ; XTTS v2, AllTalk, Piper et Chatterbox Multilingual v3 pour la voix ; Resemblyzer, ECAPA-TDNN et des empreintes MFCC pour reconnaître qui parle ; E5 multilingual — lui-même fine-tuné sur 3 878 paires propres au projet — MiniLM, Qdrant et ChromaDB pour la mémoire.

Quelques modèles essayés avant février 2026, Phi parmi eux, ne figurent pas dans ce livre : à cette date, rien n'était encore consigné.



2 juin 2026 — le trajet d'un seul échange, étage par étage : l'écoute, l'intention, la décision, la mémoire, le modèle, le nettoyage, la voix. En bas, la jauge de mémoire vidéo : 24 463 Mio, et pas un de plus.

Trois machines, et une contrainte qui ne ment pas

Ce travail n'a pas été fait sur du matériel de laboratoire. Il a commencé sur **8 Go de VRAM**, a continué sur **12 Go**, et se poursuit aujourd'hui sur **24 Go** — une carte de portable. Chaque palier a redéfini ce qui était possible, et c'est précisément ce qui donne sa valeur au résultat : ce qui tourne ici tournera chez les autres.

Il faut distinguer deux moments. **L'entraînement** est loué à l'heure, sur des cartes qu'on ne possède pas — A100 80 Go, H100, H200, RTX PRO 6000 Blackwell 96 Go. On paie quelques dizaines d'euros, on récupère un modèle, on rend la machine. **L'exécution**, elle, tient entièrement sur l'ordinateur de la maison. C'est tout l'enjeu : le coût de fabrication est mutualisé une fois, l'usage est gratuit et privé pour toujours.

La contrainte ne se négocie pas, et une décision du 2 octobre 2026 le montre bien. Entre une quantification Q8 et une Q4_K_M, le Q8 est meilleur — « rien à redire ». Il a pourtant été écarté : 18,4 Go de VRAM chargés contre 13,4. C'est la VRAM qui décide, pas la préférence. Toute l'ingénierie de ce livre se joue dans cet écart de cinq gigaoctets.

Ce que ce livre contient

Le journal couvre la période du 11 février au 9 octobre 2026, dans l'ordre chronologique. Il réunit les relevés de mesure, les spécifications des blocs d'entraînement, les comptes rendus de runs — réussis comme ratés — et les cartographies successives de l'architecture. Les résultats négatifs y figurent au même titre que les autres : une enquête de plusieurs semaines sur une régression de verbosité, un modèle de 26 milliards de paramètres entraîné puis écarté, six mois d'architecture abandonnés.

Le projet est développé par une seule personne, sur du matériel acheté dans le commerce.

La conclusion que ce travail documente est la suivante : une intelligence artificielle locale, cohérente et autonome sur du matériel grand public relève d'un problème d'ingénierie identifiable, dont les principales difficultés sont la distribution du corpus d'entraînement, la mémoire persistante et la contrainte de mémoire vidéo. Les pages qui suivent exposent les étapes, les mesures et les décisions qui ont conduit à l'état actuel du système.