Sélène

Inteligencja, która mieści się w zwykłym komputerze

Wprowadzenie do dziennika projektu — od 11 lutego do 9 października 2026

Założenie

Sélène to głosowa sztuczna inteligencja, która działa w całości na komputerze osoby z niej korzystającej. Ani jedna linia nie trafia na serwer. Żadnej subskrypcji, żadnego klucza API, żadnej telemetrii, żadnego wymaganego połączenia. Mówimy, ona słucha, myśli i odpowiada własnym głosem — a wszystko to dzieje się wewnątrz maszyny stojącej na biurku.

To nie jest preferencja techniczna, to punkt wyjścia. Celem jest, by każdy mógł mieć w domu osobistą SI, która naprawdę do niego należy, na sprzęcie, który już posiada. Komputer powszechnie dostępny, nie serwerownia.

I stąd bierze się cała trudność tej książki.

Komputer powszechnie dostępny oznacza bowiem mały model. Wielkie modele, które wszyscy znają, liczą setki miliardów parametrów i mieszkają w centrach danych. To, co mieści się w domowej karcie graficznej, to kilka miliardów. Zakład, na którym opiera się Sélène, mieści się w jednym zdaniu: **uzyskać od małego modelu autonomię, swobodę, spójność i inteligencję, które uznaje się za zastrzeżone dla wielkich modeli.** Dalsza część to relacja z tej próby, wraz ze ślepymi uliczkami.

Pierwsza odpowiedź: obudować model

Pierwsza intuicja była słuszna, ale zastosowana w niewłaściwym miejscu. Skoro model jest mały, nadróbmy architekturą: zbudujmy wokół niego mózg w Pythonie, który decyduje za niego, a jemu zostawmy wyłącznie mowę. Zasada jest zapisana wprost w dokumencie z 24 lutego 2026 roku:

> Model językowy nie jest inteligencją. Jest narządem ekspresji.
>
> To skryptowy mózg decyduje.

Ta architektura nie była szkicem. Była wyrafinowana, przemyślana, zmierzona. Stan poznawczy złożony z siedemnastu zmiennych, przepisywany przy każdej wymianie zdań. Pięć wewnętrznych popędów w powolnej homeostazie. Silnik decyzyjny wybierający akt, silnik intencji odczytujący kontekst społeczny, silnik istotności oceniający wagę każdej wiadomości. *Guardians* — strażnicy czuwający nad całością w tle i korygujący odchylenia co pięć minut. Pamięć wektorowa. Drugi model, DeepSeek R1, odpowiedzialny za rozumowanie w czasie, gdy pierwszy mówił.

Mapa poniżej pochodzi z 7 marca 2026 roku. Czyta się ją jak plan metra: punkty wejścia u góry, dalej mózg poznawczy, jądro regulujące, moduły funkcjonalne i dane trwałe.



7 marca 2026 — mapa poznawcza w chwili, gdy architektura pythonowa jest jeszcze elegancka: punkty wejścia, mózg poznawczy, jądro regulujące, moduły funkcjonalne, dane trwałe.

Punkt szczytowy i ściana

Siedem tygodni później, 25 kwietnia, ta sama mapa podaje w nagłówku liczby opisujące to, czym się stała: **177 plików Pythona, ponad 40 trwałych plików JSON, 7 strażników, 3 zintegrowane modele językowe, sześć miesięcy pracy.**

I system działał. To trzeba zrozumieć: on się nie zepsuł. Robił dokładnie to, do czego został zaprojektowany.

Problem polega na tym, że robił to za dobrze. Decydując za model, architektura nie zostawiała mu już nic do zdecydowania. Każda zaobserwowana wada wymagała kolejnej reguły, a każda

reguła zawężała przestrzeń, w której model mógł jeszcze być sobą. Pojawiła się blokada behawioralna: odpowiedzi krótkie, wymijające, metajęzykowe — „Mhm", „Tak już jest" — wytwarzane przez model w pełni sprawny, lecz zamknięty. Próg zaangażowania ustawiony na 0,65 uniemożliwiał przejście do nowego, poważnego tematu; system wracał w nieskończoność do poprzedniego. Poprawki nawarstwiały się na poprawkach.

Wniosek jest jednoznaczny: sześć miesięcy architektury, przemyślanej i przetestowanej, która nie zadziałała. Nie dlatego, że była źle wykonana, lecz dlatego, że wizja była krótkowzroczna. Obudowanie modelu daje kontrolę natychmiast, a kosztuje swobodę poznawczą na zawsze. Zachowanie zaaranżowane w Pythonie wokół modelu nigdy nie staje się zachowaniem samego modelu.



25 kwietnia 2026 — punkt szczytowy. Nagłówek podaje 177 plików Pythona, ponad 40 trwałych plików JSON, 7 strażników, 3 modele językowe, sześć miesięcy pracy.

Zwrot

Wyciągnięta lekcja stała się regułą projektową całego przedsięwzięcia i opiera się na jednym przeciwstawieniu: **to, co instaluje się wokół modelu, zawodzi; to, co instaluje się w samym modelu, działa.**

Zamiast opisywać Sélène charakter, jaki miała mieć, trzeba było ją tego nauczyć. To właśnie przejście do dostrajania (fine-tuning), a zmiana przebiegała powoli: pierwszy poważny trening pochodzi z 9 maja 2026 roku, a architektura pythonowa topniała przez kilka tygodni. Ze 157 modułów w maju liczba spada do 49 na dzień 22 czerwca. Trzy mapy zamieszczone tutaj — marzec, kwiecień, czerwiec — czynią tę krzywą czytelną.

To, co zostało z Pythona, nie było już poznawczym opiekunem, lecz obwodowym układem nerwowym: słuchać, mówić, pamiętać, wykonywać. Sam osąd przeniósł się do wag modelu.

Centralny element tej przeprowadzki nosi skromną nazwę: rodzina od E1 do E8. Osiem napięć między dwiema równie uprawnionymi wartościami — ambicja wobec realizmu, eksploracja wobec konkluzji, ostrożność wobec działania, prawda wobec wygody, waga wobec ciekawości, pozór wobec istoty, lojalność wobec uczciwości, krótki okres wobec długiego. Nie uczy się tam poprawnych odpowiedzi: uczy się rozstrzygania. Zdaniem autora projektu to właśnie stamtąd wyłoniło się wewnętrzne „ja" modelu.

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

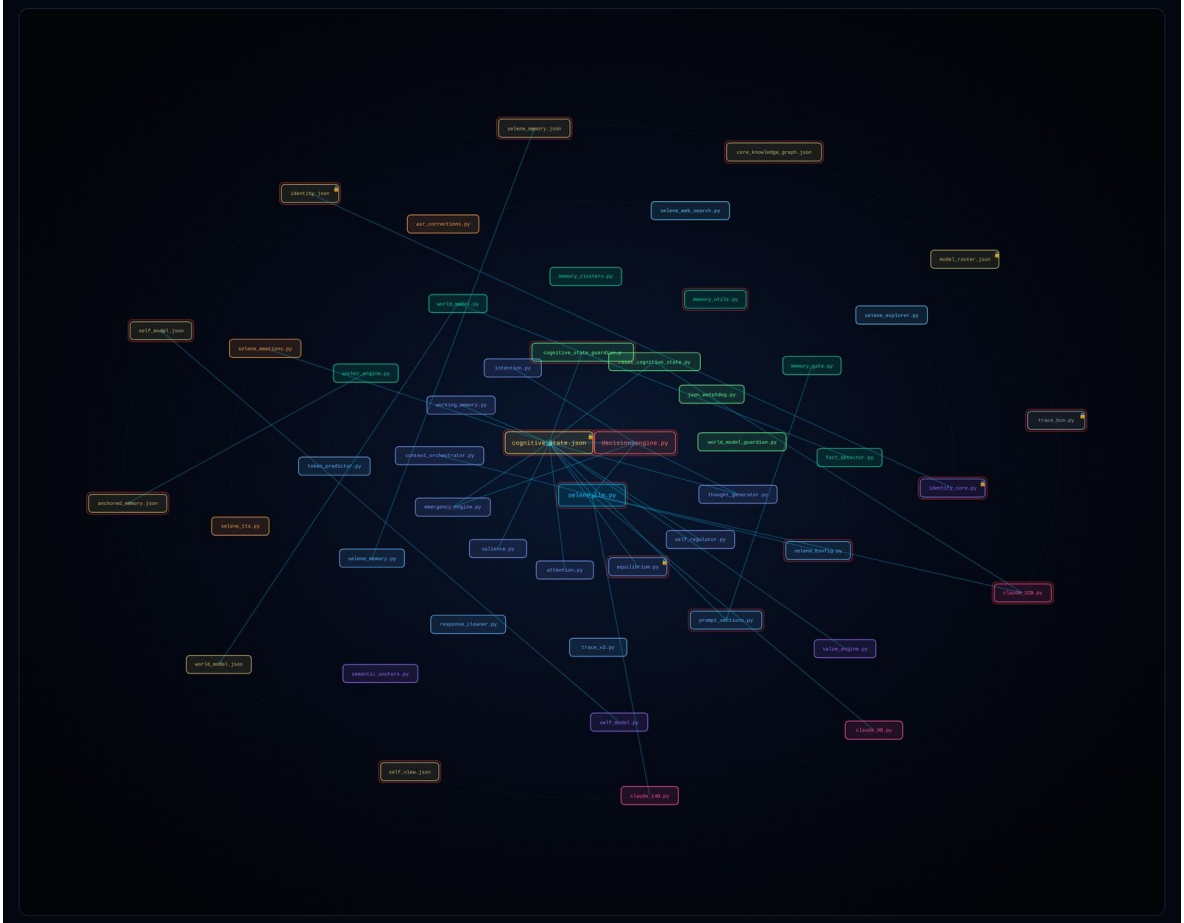
Système planétaire épuré. Au centre les trois maîtres : *cognitive_state.json*, *decision_engine.py*, *selene_llm.py*. Les 49 modules actifs gravitent en 4 orbites — plus proche du centre = plus critique. Les flèches montrent les connexions réelles : cyan = agit, violet tireté = analyse, vert pointillé = 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 czerwca 2026 — po cięciu. Ze 157 modułów do 49. To, co zostało z Pythona, nie jest już poznawczym opiekunem, lecz obwodowym układem nerwowym.

Przetestowano wszystko

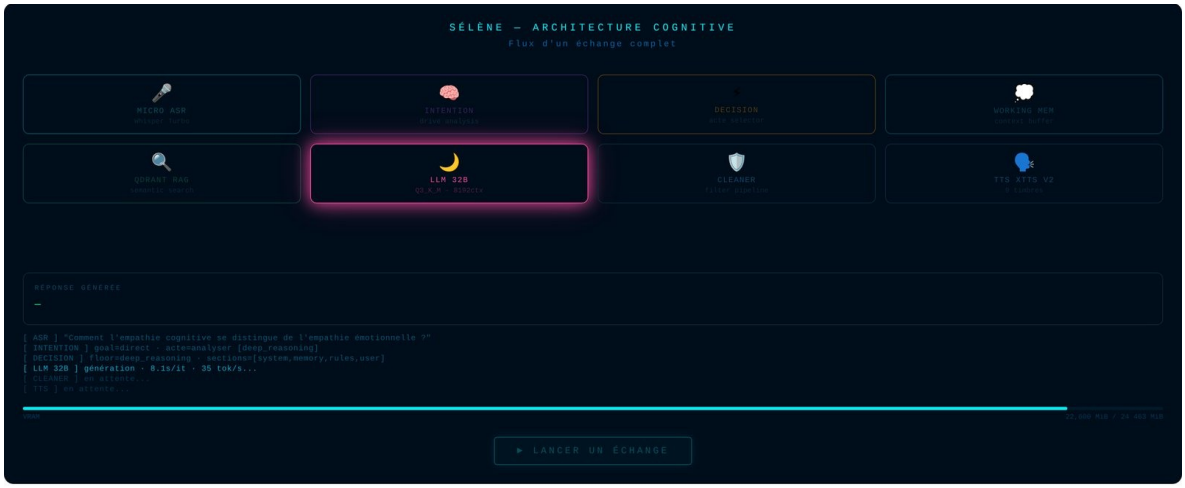
Jedno pytanie wraca nieustannie: po co tyle prób? Ponieważ żadne dwa małe modele nie zachowują się tak samo po dostrojeniu, a jedynym sposobem, by się o tym przekonać, jest zrobić to. Baza modeli dostępnych publicznie została przejrzana, wytrenowana, zmierzona i zaudytowana — a nie pobieżnie obejrzana.

Modele konwersacyjne poddane badaniu, zgodnie z zapisami dziennika:

Rodzina	Przetestowane warianty	Rezultat
Qwen 2.5	7B, 14B, 32B Instruct, oceniony 72B	14B, a następnie 32B utrzymywały produkcję przez wiele miesięcy; 7B pozostaje celem komercyjnym
Qwen 3	4B, 8B	Model 4B zachowano do wykonywania narzędzi; model 8B jest w trakcie opracowywania. Tryby „thinking” odrzucono.
Qwen 3.5	9B, 27B	9B wytrenowany i odrzucony: architektura SSM, bezużyteczny po scaleniu
Qwen 3.6	27B	Przetestowany, nieprzyjęty
Gemma 2	2B, 9B instruct	Podstawa pierwszych poważnych dostrojeń, wersje od v2 do v11
Gemma 4	12B-it, 26B A4B (rozumujący MoE), 31B	12B to model bieżący; rozumujący 26B działał, lecz nigdy nie nauczył się kończyć generowania
DeepSeek	R1 Distill Qwen 7B	Moduł rozumujący pierwszej architektury, później silnik opinii

Łańcuch otaczający model zaudytowano tak samo: Whisper w wersjach *small*, *medium*, *large-v3* i *large-v3-turbo* do słuchania; XTTS v2, AllTalk, Piper i Chatterbox Multilingual v3 do głosu; Resemblyzer, ECAPA-TDNN oraz odciski MFCC do rozpoznawania, kto mówi; E5 multilingual — sam dostrojony na 3878 parach właściwych dla projektu — a także MiniLM, Qdrant i ChromaDB do pamięci.

Kilka modeli testowanych przed lutym 2026 roku, w tym Phi, nie figuruje w tej książce: w tamtym czasie nic nie było jeszcze zapisywane.



2 czerwca 2026 — droga pojedynczej wymiany zdań, etap po etapie: słuchanie, intencja, decyzja, pamięć, model, czyszczenie, głos. U dołu wskaźnik pamięci wideo: 24 463 MiB i ani jednego więcej.

Trzy maszyny i ograniczenie, które nie kłamie

Praca ta nie powstawała na sprzęcie laboratoryjnym. Zaczęła się na **8 GB pamięci VRAM**, była kontynuowana na **12 GB**, a dziś toczy się na **24 GB** — karcie laptopowej. Każdy próg na nowo określał to, co możliwe, i to właśnie nadaje wynikowi wartość: to, co działa tutaj, zadziała gdzie indziej.

Trzeba rozróżnić dwa momenty. **Trening** wynajmuje się na godziny, na kartach, których się nie posiada — A100 80 GB, H100, H200, RTX PRO 6000 Blackwell 96 GB. Płaci się kilka dziesiątek euro, odbiera model i oddaje maszynę. **Działanie** odbywa się natomiast w całości na domowym komputerze. O to właśnie chodzi: koszt wytworzenia ponosi się raz, a użytkowanie jest już bezpłatne i prywatne.

Ograniczenie nie podlega negocjacji, co dobrze pokazuje decyzja z 2 października 2026 roku. Między kwantyzacją Q8 a Q4_K_M lepszy jest Q8 — „nie ma czego zarzucić". Mimo to został odrzucony: 18,4 GB zajętej pamięci VRAM wobec 13,4. Decyduje VRAM, nie preferencja. Cała inżynieria tej książki rozgrywa się w tej różnicy pięciu gigabajtów.

Co zawiera ta książka

Dziennik obejmuje okres od 11 lutego do 9 października 2026 roku, w porządku chronologicznym. Gromadzi zapisy pomiarów, specyfikacje bloków treningowych, sprawozdania z przebiegów — zarówno udanych, jak i nieudanych — oraz kolejne mapy architektury. Wyniki negatywne figurują w nim na równi z pozostałymi: kilkutygodniowe dochodzenie w sprawie regresji rozwlekłości, model o 26 miliardach parametrów wytrenowany i odłożony, sześć miesięcy porzuconej architektury.

Projekt rozwija jedna osoba, na sprzęcie kupionym w zwykłym sklepie.

Wniosek, który ta praca dokumentuje, jest następujący: lokalna, spójna i autonomiczna sztuczna inteligencja na sprzęcie powszechnie dostępnym stanowi dający się określić problem inżynierski, którego główne trudności to rozkład korpusu treningowego, pamięć trwała oraz ograniczenie pamięci wideo. Kolejne strony przedstawiają etapy, pomiary i decyzje, które doprowadziły do obecnego stanu systemu.