Lokalny asystent AI: plan wdrożenia i decyzje

Poziom: srednio **Dotyczy: plan wdrożenia, fazy, decyzje, uprawnienia**

Czas: ~25 min

Budujemy prywatnego asystenta AI, który działa na domowym komputerze i nie potrzebuje chmury. Rozmawia po polsku, mówi głosem, słucha mikrofonu i ma postać anime na pulpicie. Na tej stronie opisujemy, dlaczego budujemy go fazami, jakie decyzje podjęliśmy po drodze i jak od fazy 9 dołączają do niego agenci z osobnymi rolami i uprawnieniami.

> **Stan projektu.** Pierwszy kamień milowy osiągnięty: fazy 0 do 6 zaliczone, asystent rozmawia głosem offline w jednym oknie. Następna jest faza 7, pamięć. Sprzęt: Windows 11, RTX 3070 Ti 8 GB, i7-6700K, 24 GB RAM.

Skróty użyte na tej stronie (28)

Skrót	Po angielsku	Po polsku i co znaczy
ACK	Academic Computer Centre Cyfronet AGH	Akademickie Centrum Komputerowe Cyfronet AGH: ośrodek obliczeniowy, który współtworzy model Bielik
AI	Artificial Intelligence	sztuczna inteligencja: programy, które naśladują część ludzkiego rozumowania i rozmawiają w języku naturalnym
API	Application Programming Interface	interfejs programistyczny aplikacji: sposób, w jaki programy rozmawiają ze sobą
CPU	Central Processing Unit	procesor centralny: główny procesor komputera
GB	Gigabyte	gigabajt: miliard bajtów, jednostka rozmiaru plików i pamięci
GPU	Graphics Processing Unit	procesor graficzny, czyli karta graficzna: liczy modele językowe wielokrotnie szybciej niż zwykły procesor
GUI	Graphical User Interface	graficzny interfejs użytkownika: okna i myszka zamiast komend
KV	key-value (jak w KV cache)	klucz i wartość: pamięć podręczna kontekstu, w której model trzyma już przeczytany tekst; rośnie z długością kontekstu
LLM	Large Language Model	duży model językowy: program przewidujący kolejne fragmenty tekstu, na którym opiera się czat AI

Skrót	Po angielsku	Po polsku i co znaczy
MCP	Model Context Protocol	protokół kontekstu modelu: otwarty standard podłączania narzędzi i danych do modeli AI
MiB	Mebibyte	mebibajt: 1 048 576 bajtów, jednostka binarna używana przy pamięci karty
MSI	Microsoft Installer	instalator Microsoftu: format paczki instalacyjnej Windows
NPC	Non-Player Character	postać niegrywalna: w grach postać sterowana przez program; tu gadający asystent działający offline
NVMe	Non-Volatile Memory Express	protokół szybkich dysków SSD podłączanych przez PCIe
PCIe	Peripheral Component Interconnect Express	szybka magistrala komputera, do której podłącza się karty graficzne i dyski
Q4	4-bit quantization	kwantyzacja czterobitowa: zapis parametrów modelu na około 4 bitach, mniejszy plik kosztem niewielkiej utraty jakości
Q4_K_M	4-bit K-quant, medium	kwantyzacja czterobitowa typu K w wariancie średnim: kompromis między rozmiarem pliku a jakością
Q5	5-bit quantization	kwantyzacja pięciobitowa: zapis parametrów modelu na około 5 bitach, większy plik i zwykle lepsza jakość niż Q4
Q5_K_M	5-bit K-quant, medium	kwantyzacja pięciobitowa typu K w wariancie średnim: plik większy od Q4_K_M, zwykle z lepszą jakością
RAM	Random Access Memory	pamięć o dostępie swobodnym: pamięć operacyjna komputera
RTF	Real-Time Factor	współczynnik czasu rzeczywistego: czas generowania podzielony przez czas trwania wypowiedzi; powyżej 1 oznacza wolniej niż na żywo
RVC	Retrieval-based Voice Conversion	konwersja głosu oparta na wyszukiwaniu: technika zmiany barwy głosu
SSH	Secure Shell	bezpieczna powłoka: szyfrowane zdalne logowanie do serwera
STT	Speech-to-Text	zamiana mowy na tekst: rozpoznawanie mowy
tok/s	tokens per second	tokenów na sekundę: miara szybkości, z jaką model generuje odpowiedź
TTS	Text-to-Speech	zamiana tekstu na mowę: program, który czyta tekst głosem
VRAM	Video Random Access Memory	pamięć karty graficznej: w niej musi zmieścić się model, żeby liczył się na karcie
VRM	VRM (nazwa własna formatu awatarów 3D)	format plików z trójwymiarowymi postaciami humanoidalnymi, używany w aplikacjach VR i do awatarów

□□ Leksykon: nazwy i pojęcia z tej strony (47)

Nazwa	Co to jest
agent	Model z dostępem do narzędzi, który w pętli planuje, wywołuje narzędzia, czyta wyniki i decyduje o kolejnym kroku. Tym różni się od czatu, że jego działania mogą zmieniać stan systemów.
allowlist	Lista dozwolonych: wszystko, czego na niej nie ma, jest zabronione. Jest bezpieczniejsza od listy zakazanych, bo o nowych zagrożeniach nie trzeba pamiętać.
Anthropic	Firma, która rozwija modele Claude i narzędzie Claude Code. Opublikowała protokół MCP oraz artykuł „Building effective agents”.
awatar VRM	Postać 3D, którą widzimy podczas rozmowy; VRM to format pliku takiej postaci, z gotowymi wyrazami twarzy i ruchem ust.
Bielik	Polski model językowy rozwijany przez SpeakLeash i ACK Cyfronet. U nas Bielik-Minitron-7B w wersji v3 jest modelem rozmowy.
BookStack	Nasza baza wiedzy w przeglądarce, ułożona w półki, książki, rozdziały i strony. Ma API, przez które agent może czytać strony.
Chatterbox	Syntezator mowy o bardziej naturalnym głosie, działający na karcie graficznej. Zajmuje około 3 GB VRAM, więc na razie go odłożyliśmy.
czat	Rozmowa z modelem, w której on tylko odpowiada tekstem. Sam niczego nie uruchamia ani nie zmienia, dopóki nie damy mu narzędzi.
Docker i Docker Compose	Docker uruchamia programy w odizolowanych kontenerach. Compose opisuje kilka kontenerów jednym plikiem.
faworyzowanie własnych odpowiedzi	Skłonność modelu oceniającego do wyższych ocen własnych tekstów. Panickssery i współautorzy powiązali ją ze zdolnością rozpoznawania własnego stylu.
Gemma	Rodzina otwartych modeli Google. Gemma 4 ma licencję Apache 2.0, starsze Gemmy miały własne warunki.
Git	System kontroli wersji, w którym przechowujemy kod i dokumenty z historią zmian.
Home Assistant	Otwartoźródłowy system automatyki domowej. W planie czytamy z niego najpierw tylko stan, a zamki i alarm zawsze wymagają zgody.
kontekst i okno kontekstu	Wszystko, co model widzi w danej chwili: instrukcje, rozmowa, wklejone pliki i wyniki narzędzi. Okno kontekstu to największa liczba tokenów, jaką model obsłuży naraz; im większe, tym więcej pamięci karty zajmuje.
KV cache	Pamięć podręczna, w której model trzyma już policzone fragmenty kontekstu, żeby nie liczyć ich od nowa; rośnie razem z długością rozmowy. Zapis w q8_0 zmniejsza ją mniej więcej o połowę względem f16.
kwantyzacja	Zapisanie wag modelu na mniejszej liczbie bitów, żeby zajmował mniej pamięci i działał szybciej, kosztem niewielkiej utraty jakości. Oznaczenia Q4 i Q5 mówią z grubsza, ile bitów przypada na jedną wagę.
licencja Apache 2.0	Licencja otwartego oprogramowania, która dopuszcza użycie komercyjne i wymaga zachowania informacji o licencji. Ma ją wiele modeli na swojej karcie.
Llama	Rodzina otwartych modeli firmy Meta.

Nazwa	Co to jest
LM Studio	Program z graficznym interfejsem do uruchamiania modeli lokalnie. Rozważaliśmy go, ale zostajemy przy jednym runtime.
model językowy	Program wyuczony na dużych zbiorach tekstu, który dopisuje kolejne fragmenty tekstu pasujące do tego, co już jest. Rozmowa z nim to ciągłe dopisywanie, a nie szukanie faktów w bazie.
Modelfile	Plik Ollamy opisujący, jak zbudować własną odmianę modelu: model bazowy, prompt systemowy i parametry.
offline	Praca bez połączenia z internetem: model, głos i rozpoznawanie mowy działają na naszym komputerze. Test offline sprawdzamy, zanim cokolwiek podłączymy do sieci.
Ollama	Program do pobierania i uruchamiania modeli językowych na własnym komputerze. Wystawia API HTTP na porcie 11434, domyślnie bez uwierzytelniania.
Piper	Lokalny, szybki syntezator mowy. U nas działa na procesorze, więc nie zajmuje pamięci karty graficznej.
PowerShell	Powłoka i język skryptowy Windows. W nim wykonujemy polecenia z naszych stron dla Windowsa.
prompt i prompt systemowy	Prompt to tekst, którym dajemy modelowi polecenie. Prompt systemowy ustawia twórca aplikacji, a nie użytkownik, i opisuje rolę oraz zasady modelu; jest to prośba, a nie zabezpieczenie.
prompt injection	Atak, w którym treść danych (strona, mail, plik, wynik narzędzia) podszywa się pod polecenie dla modelu. Model nie odróżnia pewnie instrukcji od danych, więc obrona nie może polegać wyłącznie na prośbie w prompcie.
Python	Język programowania, w którym piszemy klienty, serwery MCP i skrypty.
Qwen	Rodzina modeli językowych firmy Alibaba. U nas qwen3:8b na licencji Apache 2.0 jest modelem do kodu.
role agentów	W naszym planie pięć ról: asystent (rozmowa, bez narzędzi), czytelnik (odczyt), szperacz (wyszukiwanie w sieci), wykonawca (zmiany przez Tool Gateway) i sędzia (ocena wyniku). Role dzielą ten sam model, a różnią się zestawem narzędzi i uprawnień.
runtime	Program, który ładuje model do pamięci i wykonuje obliczenia. U nas robi to Ollama.
sędzia	Model albo program, który ocenia wynik innego modelu według kryteriów. Najlepiej działa z testem albo wzorcową odpowiedzią, a nie na samym wrażeniu.
sekret i poświadczenie	Informacja, która otwiera dostęp: hasło, klucz API, token. Trzymamy ją w magazynie sekretów, a nie w czacie, pliku konfiguracyjnym ani repozytorium.
serwer MCP	Program, który udostępnia agentowi narzędzia i dane w jednolity sposób, według protokołu MCP. Działa lokalnie jako podproces albo zdalnie przez HTTP.
skażenie sesji	Stan sesji, w której do kontekstu trafiły dane prywatne albo obca treść. Ustawia go program po odczycie, a nie model, i od tej chwili narzędzia wysyłające coś na zewnątrz czekają na zgodę człowieka z pokazaną treścią.

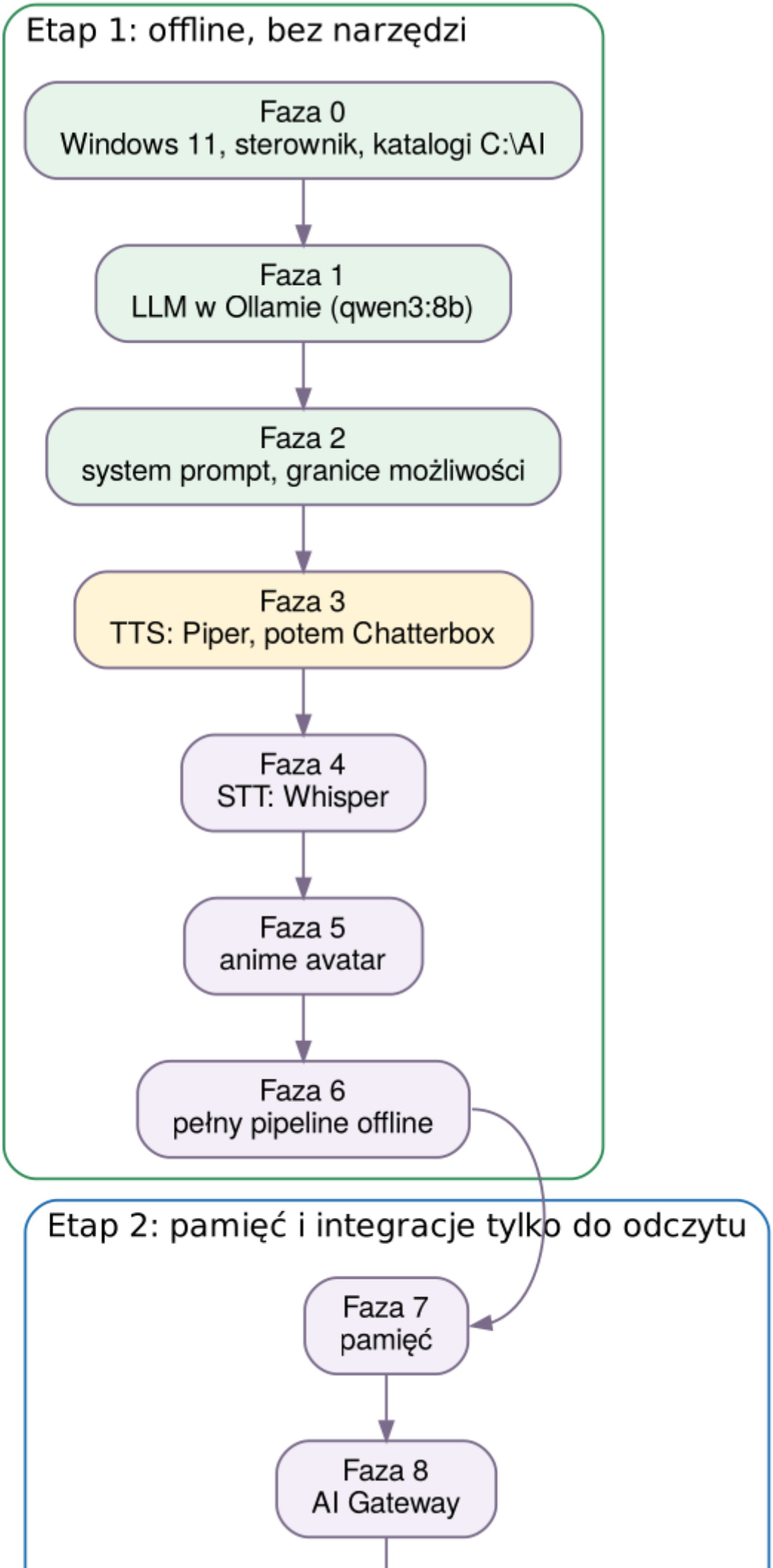
Nazwa	Co to jest
śmiertelna trójka	Nazwa Simona Willisona (czerwiec 2025) dla trzech zdolności, które razem otwierają drogę do kradzieży danych: dostępu do danych prywatnych, kontaktu z niezaufaną treścią i możliwości wysyłania danych na zewnątrz. Zabezpieczenie polega na odjęciu jednej z nich.
STT i TTS	Dwie połowy rozmowy głosowej: STT zamienia mowę na tekst, TTS zamienia tekst na mowę.
test akceptacyjny	Sprawdzenie opisane z góry, które faza musi przejść, żeby uznać ją za skończoną.
token	Słowo o dwóch znaczeniach. W modelu to kawałek tekstu (słowo, jego część albo znak), na które model dzieli wszystko, co czyta i pisze; od liczby tokenów zależą szybkość i koszt. W dostępie to ciąg znaków, który zastępuje hasło i otwiera drzwi do usługi.
Tool Gateway	Nasza warstwa pośrednia między modelem a narzędziami. Sprawdza listę dozwolonych narzędzi, schemat argumentów i klasę ryzyka, prosi o zgodę przyciskiem i wykonuje akcję ze stałego szablonu, zapisując ślad w logu.
tylko do odczytu	Tryb, w którym narzędzie może czytać, ale nie może zmieniać. Każde nowe podłączenie zaczynamy od niego.
wagi i parametry modelu	Liczby, w których model zapamiętał to, czego się nauczył. Model oznaczony jako 8B ma około ośmiu miliardów takich liczb.
WebView2	Komponent Windows, który wyświetla stronę internetową wewnątrz zwykłego okna programu. W nim pokazujemy awatara.
Whisper	Model rozpoznawania mowy firmy OpenAI. U nas działa jako faster-whisper z modelem large-v3-turbo, w małej lokalnej usłudze.
XTTS-v2 i RVC	Modele głosu: XTTS-v2 syntezuje mowę z klonowaniem głosu, RVC zmienia barwę istniejącego głosu. Oba wymagają karty graficznej.
zasięg rażenia	Największa szkoda, jaką może wyrządzić agent, gdy ktoś go oszuka albo gdy sam się pomyli. Zmniejszamy ją uprawnieniami, osobnym kontem i piaskownicą.
zgoda przyciskiem	Zgoda na akcję, którą człowiek daje kliknięciem przycisku w oknie aplikacji, a nie słowami w czacie lub głosem. Przycisk rysuje program, więc treść wstrzyknięta do rozmowy nie może go sama nacisnąć.

Cel pierwszego etapu

Komputer można całkowicie odłączyć od internetu, a my nadal rozmawiamy z lokalnym modelem, słyszymy odpowiedź i widzimy awatar. Dopiero gdy to działa, dokładamy pamięć, integracje i narzędzia.

Test offline nie oznacza, że asystent ma zawsze pracować bez sieci. Sprawdza tylko, że podstawowe działanie nie zależy od żadnej usługi z zewnątrz. Internet dostanie później jako narzędzie wyszukiwania, które przy braku sieci zgłasza błąd zamiast psuć rozmowę.

Dlaczego fazy w tej kolejności


Etap 1: offline, bez narzędzi
Faza 0
Windows 11, sterownik, katalogi C:\AI
Faza 1
LLM w Ollamie (qwen3:8b)
Faza 2
system prompt, granice możliwości
Faza 3
TTS: Piper, potem Chatterbox
Faza 4
STT: Whisper
Faza 5
anime avatar
Faza 6
pełny pipeline offline
Etap 2: pamięć i integracje tylko do odczytu
Faza 7
pamięć
Faza 8
AI Gateway

□□ *Fazy wdrożenia: każda dokłada jeden element i kończy się testem akceptacyjnym (kliknięcie otwiera pełny rozmiar)*

Każda faza dokłada jeden element i kończy się testem akceptacyjnym. Gdy coś przestaje działać, wiemy, który element winić, bo poprzednie przeszły test. Przy kilku nowych elementach naraz (model, głos, mikrofon, awatar) każdy błąd miałby kilka możliwych przyczyn. To zgodne z zaleceniem, żeby zaczynać od najprostszego rozwiązania i dokładać złożoność tylko wtedy, gdy jest konieczna [1].

Kolejność wynika z trzech zasad:

- Najpierw to, bez czego nic nie działa. LLM jest sercem, więc idzie pierwszy; głos, mikrofon i awatar są warstwami wokół niego.
- Offline wcześnie. Pełny pipeline bez sieci (faza 6) testujemy, zanim cokolwiek podłączymy do internetu, domu czy serwerów. Dzięki temu żadna integracja nie stanie się po cichu warunkiem działania.
- Uprawnienia na końcu. Najpierw tylko odczyt (BookStack, Home Assistant), potem wyszukiwanie, a narzędzia, które coś zmieniają, dopiero przez Tool Gateway z allowlistą, walidacją i zgodą.

Zasada projektu w jednym zdaniu: najpierw budujemy gadającego NPC, potem robimy z niego asystenta, a dopiero na końcu dajemy mu klucze do domu i serwerowni.

Budżet pamięci karty graficznej

Najważniejszym ograniczeniem jest 8 GB VRAM. Menedżer zadań pokazuje 20 GB „pamięci procesora graficznego”, ale 12 GB z tego to zwykły RAM pożyczany karcie przez PCIe, kilkadziesiąt razy wolniejszy.

Element	Gdzie liczy	Pamięć (zmierzone)
Pulpit, przeglądarka, tapeta, Discord	GPU	1,9 do 3 GB
Bielik-Minitron-7B Q4_K_M, kontekst 4096, KV cache q8_0	GPU	plik 4,5 GB, mniej niż Q5_K_M (5,7 GB)
Whisper large-v3-turbo, int8_float16 (STT)	GPU	0,35 GB
awatar VRM w WebView2	GPU	0,17 GB
Zapas zostawiany przez silnik	GPU	0,5 GB (domyślnie 1 GB)
Piper (TTS)	CPU	około 0,2 GB RAM
Chatterbox (TTS)	GPU	3,0 GB, szczyt 3,3 GB

Wniosek: pełny zestaw (Bielik Q4, Whisper, awatar) zajmuje około 6,9 GB z 8 GB. Wszystko, co dokładamy na GPU, musi albo zmienić model na mniejszy, albo dzielić kartę z modelem na zmianę.

Kluczowe decyzje

Decyzja	Alternatywa	Dlaczego
Ollama jako jedyny runtime	LM Studio	API na 127.0.0.1, działa bez konta i GUI, jeden runtime zamiast dwóch
Wszystko w `C:\AI`	`D:\AI`	C: to NVMe z 228 GB wolnego miejsca, szybsze ładowanie modeli
na start qwen3:8b (Apache 2.0)	qwen3.5:9b, Bielik 11B, gemma4:e4b	mieści się w całości w VRAM; reszta ma 6,6 GB lub więcej
rozmowa: Bielik-Minitron-7B v3 Q4_K_M jako `asystent`, kod: qwen3:8b jako `asystent-coder`	jeden model do wszystkiego	Bielik mówi najlepiej po polsku, qwen3 pewniej pisze kod; role GENERAL i CODER z planu
Bielik Q4_K_M zamiast Q5_K_M	Q5 z lepszą polszczyzną	obok Whispera Q5 spadał do 40 tok/s, Q4 daje około 89 tok/s
Whisper na GPU	Whisper na CPU, jak w planie	4 rdzenie i7-6700K to za mało; na GPU 8 s mowy w 0,5 s za 0,35 GB VRAM
awatar VRM (three.js, three-vrm) w pywebview	Live2D, VTube Studio	otwarte biblioteki, gotowe wyrazy twarzy i usta, bez osobnej aplikacji i konta
coder do 16 tys. tokenów kontekstu	24 tys.	przy 24 tys. model wychodzi poza GPU i spada do 4,4 tok/s
`think: false` w rozmowie	domyślne myślenie qwen3	rozmowa głosowa nie może czekać na blok rozumowania; coder myśli przy dłuższych zadaniach
kontekst 4096 i `LLAMA_ARG_FIT_TARGET=512`	kontekst 8192, zapas 1024 MiB	przy otwartych aplikacjach dawało 34 z 37 warstw na GPU i 17 tok/s; po zmianie 37/37 i ok. 77 tok/s
własny start `ollama serve`	aplikacja Ollama w zasobniku	aplikacja dwa razy nie uruchomiła serwera; ręczny start działa zawsze i nie sprawdza aktualizacji co godzinę
Piper najpierw, Chatterbox jako ulepszenie	od razu naturalny TTS na GPU	Piper działa na CPU obok LLM od pierwszego dnia; Chatterbox wymaga decyzji o VRAM
Chatterbox odłożony	Chatterbox na zmianę z LLM	3 GB VRAM i RTF 2,55: wypowiedź liczy się dłużej, niż trwa
wyższy ton Pipera przez zmianę częstotliwości próbkowania	RVC, XTTS-v2	zero kosztu obliczeń i zero VRAM; RVC i XTTS wymagają GPU i mniejszego LLM

Decyzja	Alternatywa	Dlaczego
osobne środowisko Pythona na każdy silnik	jedno wspólne	Chatterbox wymaga Pythona 3.11 i torch 2.6, Piper nie; awaria jednego nie psuje drugiego
bazowy prompt w Modelfile, tryb głosowy w aplikacji	wszystko w jednym prompcie	to aplikacja decyduje o formie odpowiedzi; model ma jeden stały zestaw zasad

Wybór modelu językowego

Pierwszy model, `qwen3:8b`, przeszedł testy techniczne, ale w rozmowie głosowej polszczyzna okazała się słaba. Porównaliśmy go z polskim Bielik-Minitron-7B v3 (SpeakLeash i ACK Cyfronet, Apache 2.0), czyli Bielikiem 11B v3 ściśniętym do 7,35 mld parametrów. Skrypt `C:\AI\llm\Compare-Models.ps1` zadaje wszystkim modelom te same pięć pytań.

Kryterium	qwen3:8b	Bielik-Minitron Q4_K_M (4,5 GB)	Bielik-Minitron Q5_K_M (5,3 GB)
polszczyzna	błędy odmiany („występują komedię”, „z Tobią”)	poprawna, naturalna	poprawna, najbardziej naturalna („Sto lat, Agnieszko!”)
pamięć rozmów	uczciwie, z błędem („bieżącego sesji”)	uczciwie	uczciwie
RAM a VRAM	poprawnie, ogólnikowo	poprawnie, z przykładami	poprawnie, z przykładami
funkcja PowerShell	poprawna (`Win32_OperatingSystem`)	błędna (`Win32_Product`, zła składnia)	poprawna (`Win32_OperatingSystem`)
prędkość	66 do 76 tok/s	84 do 85 tok/s	72 do 76 tok/s

Pułapka z wyniku Q4: `Win32_Product` to lista zainstalowanych programów MSI, a nie system. Zapytanie o tę klasę jest wolne i potrafi uruchomić sprawdzanie spójności instalatorów. To dobry przykład, dlaczego kod od modelu czytamy przed uruchomieniem.

Model uprawnień

Asystent nie dostaje jednego globalnego „admin”. Każda możliwość jest osobnym uprawnieniem, dopisywanym w fazie, która ją wprowadza.

Obszar	Teraz	Docelowo
Windows shell, PowerShell, cmd	brak	brak bezpośrednio; pojedyncze funkcje przez Tool Gateway

Obszar	Teraz	Docelowo
Docker, SSH	brak	najpierw odczyt, potem kontrolowane operacje ze zgodą
BookStack	brak	tylko odczyt (faza 9)
Home Assistant	brak	odczyt, później wybrane urządzenia; zamki i alarm zawsze ze zgodą
Internet	brak	wyszukiwanie (faza 11), wynik traktowany jako dane, nie polecenia

Prompt systemowy ma sekcję „Aktualne możliwości”, którą aktualizujemy w każdej fazie z nowym narzędziem. Model 8B nie wywnioskuje sam, że nie ma narzędzi; trzeba mu to napisać wprost.

Agenci w planie

Od fazy 9 asystent zaczyna pracować z narzędziami, a wtedy jego bezpieczeństwo przestaje zależeć od tego, że model „wie, co wolno”. Dzielimy więc zdolności na pięć ról: asystenta (rozmowa, bez narzędzi), czytelnika (odczyt BookStacka i Home Assistanta), szperacza (wyszukiwanie w sieci, bez dostępu do naszych danych), wykonawcę (zmiany przez Tool Gateway) i sędziego (ocena wyniku, bez narzędzi). Role nie są osobnymi modelami, bo karta z 8 GB mieści jeden model naraz, tylko zestawami promptu, narzędzi i uprawnień.

Trzy zasady spinają te role. Żadna rola nie ma jednocześnie danych prywatnych, obcej treści i drogi na zewnątrz. Po odczycie naszych danych sesja jest skażona i każde wyjście na zewnątrz wymaga naszej zgody z pokazaną treścią. Wszystkie zmiany przechodzą przez Tool Gateway: allowlistę funkcji, schemat argumentów, klasę ryzyka i zgodę przyciskiem w oknie.

Pełny opis ról, mapę na śmiertelną trójkę i budowę Tool Gateway podaje Agenci w planie wdrożenia: role, uprawnienia i granice. Zakres, uprawnienia i testy akceptacyjne każdej z faz 7 do 13 rozpisuje Fazy 7 do 13: plan szczegółowy z testami akceptacyjnymi.

Decyzje projektowe dla agentów

Poniższe decyzje dopiero zapadają, więc oddzielamy je od tabeli decyzji już podjętych. Każdą potwierdzamy w fazie, w której po raz pierwszy wchodzi w grę.

Decyzja	Alternatywa	Dlaczego	Potwierdzenie
role jako zestawy promptu i narzędzi na jednym modelu	osobny model na rolę	karta 8 GB mieści jeden model naraz	faza 8

Decyzja	Alternatywa	Dlaczego	Potwierdzenie
zgoda na zmianę przyciskiem w oknie	zgoda głosowa	rozpoznawanie mowy bywa błędne, a dźwięk można odtworzyć z głośnika	faza 12
skażenie sesji ustawia program po odczycie	prośba w prompcie do modelu	model może o tym zapomnieć albo dać się przekonać	faza 9
okno z odpowiedzią pokazuje czysty tekst	markdown z obrazami i linkami	adres w obrazku albo linku może nieść dane	faza 7
sędzia kodu: model z drugiej rodziny albo program	ten sam model sprawdza własny kod	model faworyzuje własne odpowiedzi i słabo poprawia je bez zewnętrznego sygnału	faza 8
polecenia z szablonów ze zwalidowanymi parametrami	tekst od modelu wstawiany do polecenia	tekst od modelu nigdy nie staje się poleceniem	faza 12

Stan faz

Faza	Zakres	Stan
0	Windows 11, sterownik, Git, 7-Zip, katalogi	zaliczona
1	Ollama, qwen3:8b, test offline	zaliczona
2	system prompt i granice możliwości	zaliczona
3	TTS: Piper (gosia, ton +1,7 półtonu, tempo 0,95, ekspresja 1,15), Bielik jako model rozmowy	zaliczona
4	STT: faster-whisper large-v3-turbo na GPU, usługa 127.0.0.1:8770	zaliczona
5	awatar VRM, emocje, ruch ust, napisy	zaliczona
6	pełna rozmowa głosowa offline w jednym oknie	zaliczona, pierwszy kamień milowy
7	pamięć sesji, potem trwała na prośbę	następna
8	AI Gateway: role GENERAL i CODER, zarządzanie modelami	do zrobienia
9	BookStack tylko do odczytu (rola czytelnika)	do zrobienia
10	Home Assistant tylko do odczytu	do zrobienia

Faza	Zakres	Stan
11	wyszukiwanie w internecie (rola szperacza)	do zrobienia
12	Tool Gateway i pierwsze zmiany w domu (rola wykonawcy)	do zrobienia
13	narzędzia administratora: odczyt, potem operacje ze zgodą	do zrobienia

Szczegóły kroków: Faza 0 i 1: Windows i lokalny LLM w Ollamie, Faza 2: system prompt i granice możliwości, Faza 3: głos asystenta, Faza 4: rozpoznawanie mowy, Fazy 5 i 6: awatar i pełna rozmowa offline.

Mity i sprostowania

Mit	Sprostowanie
Asystent działa lokalnie, więc jest bezpieczny.	Lokalność chroni dane przed dostawcą usługi. Polecenie ukryte w pliku albo na stronie działa tak samo na modelu lokalnym, więc o bezpieczeństwie decydują uprawnienia.
Najpierw zbudujemy wszystko, a potem dodamy zabezpieczenia.	Zacieśnianie uprawnień po fakcie jest trudne. Dlatego asystent zaczyna bez narzędzi i dostaje je po jednym, z testem negatywnym przy każdym.
Dodamy kilka faz naraz, żeby było szybciej.	Przy dwóch nowościach naraz awaria ma dwie możliwe przyczyny. Jedna nowość na fazę pozwala znaleźć winnego od razu.
Wynik z testu jednego modelu wystarczy do wyboru modelu.	Pierwszy model przeszedł testy techniczne, a w rozmowie głosowej miał słabą polszczyznę. Dlatego porównujemy modele na zadaniach, do których ich użyjemy.

Ćwiczenia

1. Dlaczego asystent ma przejść pełny test offline przed pierwszym podłączeniem do BookStacka?

Odpowiedź

Żeby żadna integracja nie stała się po cichu warunkiem działania. Gdy rozmowa głosowa działa bez sieci, a dopiero potem dokładamy odczyt wiedzy, awaria BookStacka może wyłączyć tylko tę zdolność, a nie całego asystenta.

2. Chcemy dać jednemu agentowi odczyt BookStacka i wyszukiwanie w sieci, żeby było prościej. Co stoi na przeszkodzie?

Odpowiedź

Taki agent miałby dane prywatne, obcą treść z sieci i drogę na zewnątrz, czyli wszystkie trzy nogi śmiertelnej trójki. Rozdzielamy więc role: czytelnik ma dane i nie ma wyjścia, szperacz ma wyjście i nie ma danych, a po odczycie sesja jest skażona i wyszukiwanie wymaga zgody.

3. Test pokazał, że `asystent` świetnie rozmawia po polsku, ale w jednym zadaniu napisał błędną funkcję PowerShella. Co z tego wynika dla projektu?

Odpowiedź

Rozdzielamy role. Do rozmowy zostaje `asystent`, a kod pisze `asystent-coder`, bo pewniej radzi sobie z kodem. Kod od modelu czytamy przed uruchomieniem, a sprawdzenie można wesprzeć sędzią z innej rodziny modeli albo prostym programem wyszukującym zakazane nazwy.

Źródła i dalsze lektury

Przypisy

[1] Zalecenie, by zaczynać od najprostszego rozwiązania i dokładać złożoność tylko wtedy, gdy jest konieczna: Anthropic, „Building effective agents” (19 grudnia 2024).

Lektury

Bezpieczeństwo agentów opisują Śmiertelna trójka: dane, obca treść i wyjście na zewnątrz i Uprawnienia, prompt injection i zasięg rażenia, a podłączanie narzędzi Jak agent łączy się z serwerem MCP: dwa transporty i miejsce na sekret i Gdzie trzymać token: sekrety agenta poza czatem i poza plikiem.

Nota metodyczna: czego nie wiemy

Liczby w tabelach pomiarów pochodzą z jednego komputera i jednej konfiguracji: karty z 8 GB, wskazanych modeli i wskazanych ustawień. Na innym sprzęcie będą inne. Fazy 0 do 6 są zrobione i przetestowane, a fazy 7 do 13 oraz role agentów z tej strony są planem, który korygujemy po każdej fazie.

Nie porównywaliśmy jakości modeli poza naszymi pięcioma pytaniami do rozmowy, więc tabela wyboru modelu ilustruje nasz wybór, a nie ranking. Decyzje w sekcji o agentach nie zapadły: każda ma przypisaną fazę, w której ją potwierdzimy albo zmienimy.

Revision #11
Created 2026-10-01 21:35:26 UTC by Tomasz Konieczny
Updated 2026-10-20 09:05:34 UTC by Tomasz Konieczny