



Loonar Compass

Instrukcja użytkownika

wersja 0.8.2 · październik 2026

Loonar Mariusz Głowacki
ul. Gombrowicza 12, 73-110 Stargard
loonar.eu · mariusz@loonar.eu

Spis treści

1. [Czym jest Loonar Compass](#)
2. [Wymagania](#)
3. [Instalacja](#)
4. [Okno aplikacji i nawigacja](#)
5. [Ustawienia i język](#)
6. [Rozmowa z asystentem](#)
7. [Praca z modelem Enterprise Architect](#)
8. [Przykłady użycia](#)
9. [Współpraca z otwartym oknem EA](#)
10. [Compass w Enterprise Architect](#)
11. [Edytor diagramów](#)
12. [Wysyłanie diagramów do Miro](#)
13. [Dokumenty: Word, Excel, obrazki PNG, Confluence](#)
14. [Szablony dokumentów](#)
15. [Pamięć asystenta](#)
16. [Zatwierdzanie zmian i dziennik zmian](#)
17. [Rozmowy: historia, wyszukiwanie, praca w tle](#)
18. [Dane, bezpieczeństwo i prywatność](#)
19. [Rozwiązywanie problemów](#)
20. [Zgłaszanie błędów i nowe wydania](#)
21. [Aktualizacja i odinstalowanie](#)
22. [Compass Hub \(usługa zespołowa\)](#)
23. [Tryb sprawdzania w potokach CI](#)
24. [Licencja, znaki towarowe i kontakt](#)

1. Czym jest Loonar Compass

Loonar Compass to samodzielna aplikacja okienkowa dla Windows: asystent AI do pracy z modelem Enterprise Architect (Sparx Systems). Pisziesz do niego zwykłym językiem, a on:

- **czyta model** (pakiety, elementy, diagramy, połączenia, atrybuty i operacje) i opisuje, co w nim jest,
- **analizuje model**: wpływ zmiany elementu na resztę modelu, miejsca użycia operacji, elementy bez połączeń,
- **zmienia model**: tworzy pakiety, diagramy, elementy i połączenia, dodaje atrybuty i operacje, ustawia wartości tagów i słowa kluczowe, rozmieszcza elementy na diagramach, przenosi je, zdejmuję z diagramu, usuwa, koloruje, dodaje legendy, kopiuje pakiety i robi ich baseline,
- **prowadzi słownik projektu EA** (pojęcia z definicją i typem) oraz **scenariusze elementów** (kroki aktora i systemu),
- **rysuje** diagramy z opisu, ze zdjęcia, ze szkicu albo z załączonego pliku,
- **pokazuje diagramy** jako obrazki na czacie, tak jak rysuje je Enterprise Architect,
- **przygotowuje dokumenty** z treści modelu: Word, Excel, obrazki PNG i strony Confluence, także według szablonów,
- **pamięta ustalenia** między rozmowami (konwencje modelowania, preferencje, poprawki).

Aplikacja korzysta przez Internet z modelu językowego AI, z Twoim kluczem API: domyślnie z Claude Haiku 5.5 (Anthropic), a w ustawieniach możesz wybrać inny model Claude albo model OpenAI (rozdział 5). Z modelem EA łączy się na dwa sposoby: czyta go wprost (z bazy PostgreSQL albo z pliku .qea/.qeax), a zapisuje przez Enterprise Architect, na koncie użytkownika modelu podanym w ustawieniach, jeśli model ma takie konto. Gdy plik modelu masz otwarty w EA, zmiany idą przez Twoje okno EA; w pozostałych przypadkach EA działa w tle. Zmiany w modelu możesz przed zapisem zatwierdzić albo odrzucić (rozdział 16).

Loonar Compass działa jako osobne okno, uruchamiane z menu Start, albo jako okno zadokowane w samym Enterprise Architect, które pracuje na projekcie otwartym w tym oknie EA (rozdział 10).

Interfejs i odpowiedzi asystenta są dostępne w czterech językach: polskim, angielskim, niemieckim i francuskim (rozdział 5). Aplikacja jest bezpłatna, także do użytku komercyjnego (rozdział 24).

Bez okna Compass sprawdza też model w potokach CI (rozdział 23), a osobny program Compass Hub udostępnia modele zespołu ludziom i narzędziom (rozdział 22).

2. Wymagania

Warunki konieczne. Obsługiwana konfiguracja wymaga spełnienia następujących warunków:

1. **Model jest plikiem projektu EA .qea lub .qeax albo jest w bazie danych PostgreSQL.** Starsze formaty plików (.eap, .eapx, .feap) oraz inne bazy nie są obsługiwane; plik .eap/.eapx Enterprise Architect przekonwertuje do .qea transferem projektu (Settings → Model → Transfer → Full Project Transfer).
2. **Model w bazie ma włączone zabezpieczenia użytkowników** (w Enterprise Architect: Settings → Security → Enable Security), w trybie „Require User Lock to Edit”. W pliku modelu zabezpieczenia są opcjonalne: bez nich asystent nie potrzebuje konta EA. Aplikacja loguje się do modelu na konto asystenta i na czas zapisu zakłada blokady. Gdy zabezpieczenia są wyłączone, blokady są pomijane, a taka konfiguracja nie jest obsługiwana. W aplikacji zaznaczasz to w karcie **Model** ustawień opcjami „Model ma włączone zabezpieczenia (konta użytkowników EA)” i „Model wymaga blokady do edycji (Require User Lock to Edit)” (rozdział 5).
3. **W modelu z włączonymi zabezpieczeniami istnieje konto użytkownika dla asystenta** (Settings → Security → Users), z hasłem i z uprawnieniami do edycji. Nazwa konta jest dowolna, na przykład `c1aude` albo `asystent`. Login i hasło tego konta podajesz w karcie **Model** ustawień, razem z plikiem albo bazą modelu.

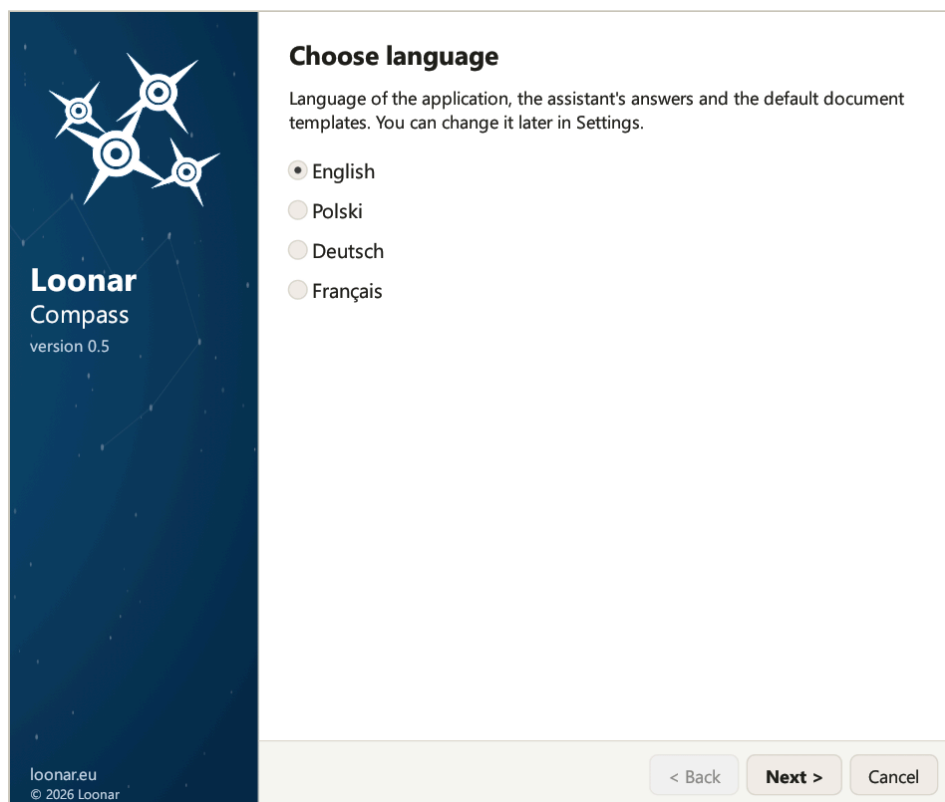
Co	Szczegóły
System	Windows 10 lub Windows 11 (64-bitowy)
Enterprise Architect	Zainstalowany na tym samym komputerze, z ważną licencją. Aplikacja uruchamia go w tle; w tym czasie zajmuje on jedno miejsce w licencji. Compass w EA (rozdział 10) pracuje w Twoim oknie EA.
Model EA	Plik projektu EA .qea lub .qeax (wystarczy jego ścieżka) albo repozytorium w bazie PostgreSQL z włączonymi zabezpieczeniami użytkowników (potrzebne są: serwer, port, nazwa bazy oraz konto i hasło do bazy, hasło bez średnika; tym kontem asystent czyta model, a Enterprise Architect zapisuje zmiany). Plik .qea jest przeznaczony dla jednej osoby, a .qeax dla małego zespołu, który współdzieli plik. Do użytku firmowego daj Compassowi rolę PostgreSQL z samym prawem SELECT na tabelach EA (<code>t_*</code>) i ustaw szyfrowanie połączenia na „Wymagane, z weryfikacją certyfikatu” (rozdział 5). Enterprise Architect zapisuje tym samym kontem, więc model, który asystent ma też zmieniać, wymaga na tych tabelach także INSERT, UPDATE i DELETE, nigdy uprawnień superużytkownika.
Konto użytkownika modelu	Dla modelu w bazie albo pliku modelu z włączonymi zabezpieczeniami: login i hasło konta asystenta w modelu EA. Najlepiej osobne konto tylko dla asystenta, żeby jego zmiany i blokady było widać osobno i żeby nie kolidowały z Twoimi.
Klucz API	Klucz API Claude z konsoli Anthropic (Anthropic Console) albo, gdy wybierzesz dostawcę OpenAI, klucz API OpenAI z platformy OpenAI (rozdział 5).
Internet	Połączenie HTTPS z <code>api.anthropic.com</code> (albo z <code>api.openai.com</code> , gdy dostawcą jest OpenAI), także przez proxy z ustawień internetowych Windows (rozdział 18).
Uprawnienia administratora	Niepotrzebne przy instalacji „Tylko dla mnie”. Wymagane tylko przy instalacji dla wszystkich użytkowników komputera (rozdział 3).
Confluence (opcjonalnie)	Cloud albo Data Center, tylko do publikowania stron przez asystenta: adres, token i klucz przestrzeni (rozdział 13). Bez tego strony też powstają, a kod wklejasz sam.
Miro (opcjonalnie)	Tylko do wysyłania diagramów na tablicę Miro: token dostępu aplikacji Miro (rozdział 12).
Node.js (opcjonalnie)	Tylko dla serwerów MCP uruchamianych przez <code>npm</code> , na przykład gotowych serwerów Azure DevOps i Files (rozdział 7).
Program pocztowy (opcjonalnie)	Program pocztowy ustawiony w Windows jako domyślny, na przykład Outlook: do zgłaszania błędów i zapisu na informacje o nowych wydaniach.

Koszty. Sama aplikacja jest bezpłatna. Za korzystanie z modelu AI płacisz wybranemu dostawcy (Anthropic albo OpenAI) według zużycia, na koncie, do którego należy klucz API.

3. Instalacja

Uruchom plik `LoonarCompass-Setup-0.8.2.exe`. Kreator prowadzi przez kolejne strony przyciskami **Dalej** i **Wstecz**. Uprawnień administratora nie potrzebujesz, chyba że instalujesz aplikację dla wszystkich użytkowników komputera.

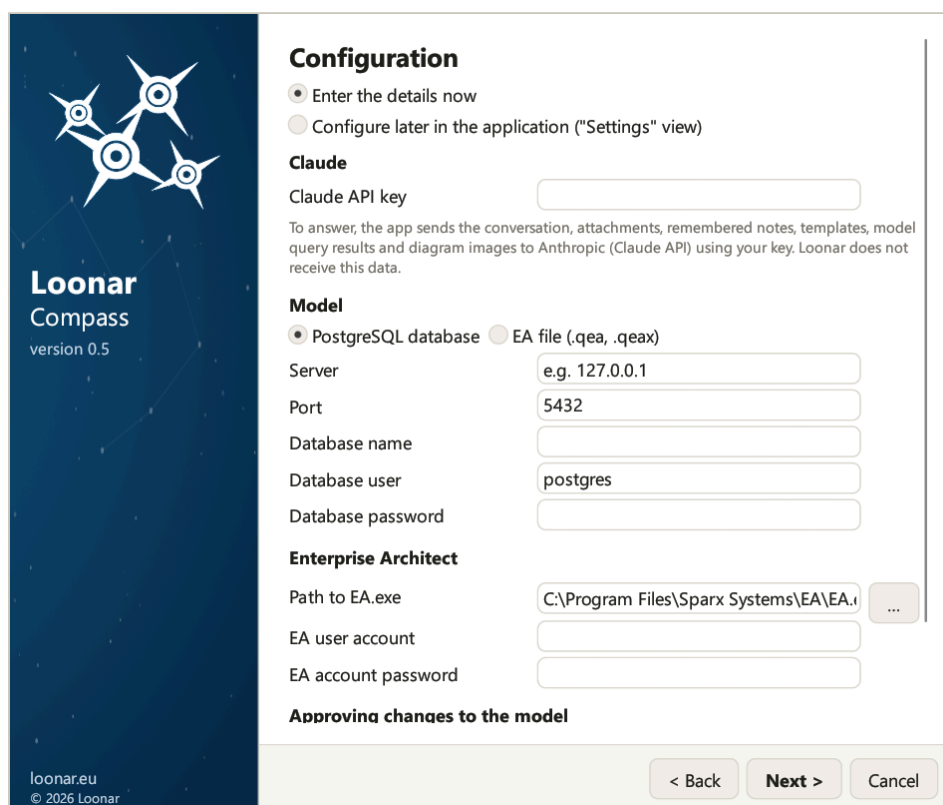
Ostrzeżenie Windows SmartScreen. Instalator nie jest jeszcze podpisany cyfrowo, dlatego przy pierwszym uruchomieniu Windows może pokazać komunikat „System Windows ochronił ten komputer”. Kliknij **Więcej informacji**, a potem **Uruchom mimo to**. Gdy w systemie jest włączona Inteligentna kontrola aplikacji (Smart App Control), Windows blokuje niepodpisane aplikacje bez możliwości ich uruchomienia: poczekaj wtedy na wersję podpisaną. Instalator pobieraj tylko z loonar.eu albo z wydań projektu na GitHubie.



Pierwsza strona kreatora: wybór języka

1. **Język.** Polski, angielski, niemiecki albo francuski. Przy nowej instalacji zaznaczony jest angielski, a przy aktualizacji język zapisany w Twoich ustawieniach. Wybrany język dotyczy kreatora, aplikacji, odpowiedzi asystenta i domyślnych szablonów dokumentów; później zmienisz go w ustawieniach (rozdział 5).
2. **Powitanie.** Warunki konieczne i lista rzeczy, które warto przygotować przed instalacją.
3. **Co nowego.** Przy aktualizacji: zmiany od zainstalowanej wersji. Przy nowej instalacji: najważniejsze zmiany w bieżącej wersji.
4. **Umowa licencyjna.** Zaznacz „Akceptuję warunki umowy licencyjnej”.

5. **Folder docelowy.** Rodzaj instalacji (opisany niżej), folder programu i skróty: w menu Start (aplikacja, instrukcja użytkownika, odinstalowanie) i na pulpicie. Opcja „**Dodaj Loonar Compass do Enterprise Architect**” instaluje Compass w EA (rozdział 10); po instalacji uruchom EA ponownie.
6. **Konfiguracja.** Wybierz „Podam dane teraz” albo „Skonfiguruję później w aplikacji (widok »Ustawienia«)”. Od razu możesz podać klucz API Claude (pod polem jest informacja, co trafia do Anthropic; dostawcę OpenAI wybierzesz później w ustawieniach), model (bazę PostgreSQL: serwer, port, nazwa bazy, konto i hasło do bazy, albo plik EA: ścieżka do pliku .qea/.qeax, którą wybierzesz przyciskiem ...), ścieżkę do EA.exe (przycisk ... otwiera wybór pliku), konto i hasło użytkownika EA oraz tryb zatwierdzania zmian. Puste pole hasła lub klucza zostawia wartość zapisaną wcześniej.
7. **Gotowy do instalacji.** Podsumowanie wyborów. Kliknij **Instaluj**; pasek postępu pokazuje procent i bieżący krok.
8. **Zakończenie.** Zaznaczona opcja „Uruchom Loonar Compass” otwiera aplikację po kliknięciu **Zakończ**.



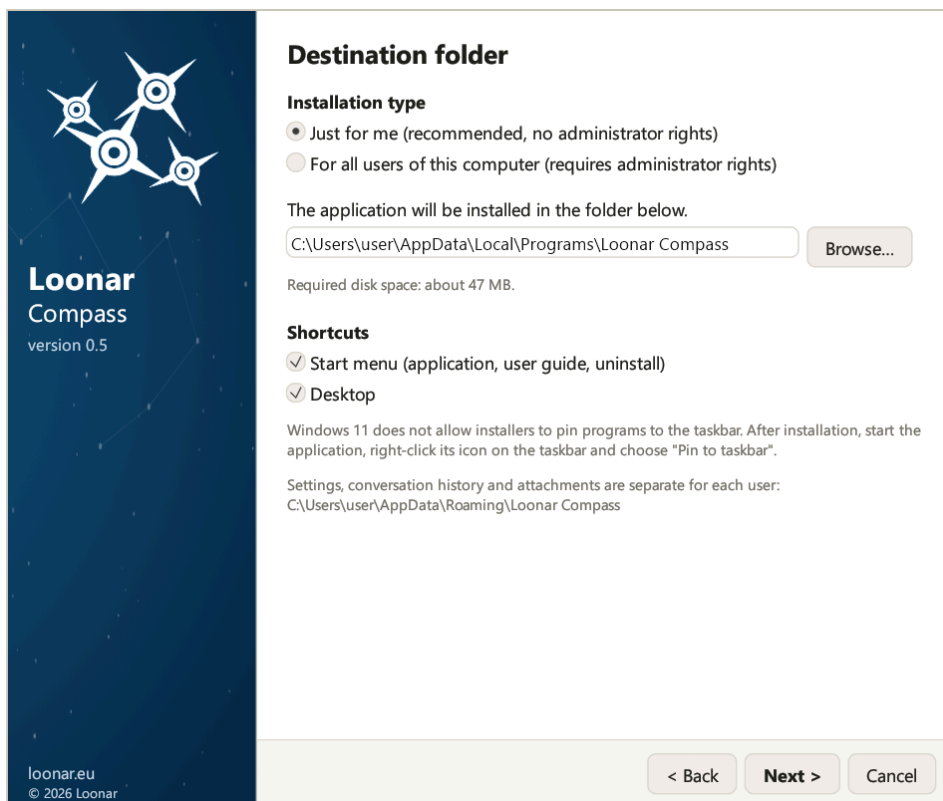
Konfiguracja w kreatorze jest opcjonalna: te same dane podasz później w widoku „Ustawienia”

Rodzaj instalacji

Na stronie „Folder docelowy” wybierasz, dla kogo instalujesz aplikację:

- **„Tylko dla mnie (zalecane, bez uprawnień administratora)”**: program trafia do folderu `%LOCALAPPDATA%\Programs\Loonar Compass`. Windows nie prosi o uprawnienia administratora.
- **„Dla wszystkich użytkowników tego komputera (wymaga uprawnień administratora)”**: program trafia do folderu `%ProgramFiles%\Loonar Compass`, w którym pliki mogą zmieniać tylko administratorzy. Windows prosi o uprawnienia administratora dopiero po kliknięciu **Instaluj**. Jeśli ich nie przyznasz, nic nie zostanie zainstalowane.

Ustawienia i hasła zawsze trafiają do profilu użytkownika, który uruchomił kreator, także wtedy, gdy uprawnienia administratora potwierdza inna osoba. Przy instalacji dla wszystkich użytkowników każdy z nich ma własne ustawienia, rozmowy i hasła.



Rodzaj instalacji, folder programu i skróty

Pasek zadań. Windows 11 nie pozwala instalatorom przypinać programów do paska zadań. Uruchom aplikację, kliknij jej ikonę na pasku prawym przyciskiem myszy i wybierz „Przypnij do paska zadań”.

Aktualizacja z wcześniejszej wersji

Wcześniejsze wersje aplikacji nosiły nazwę Loonar EA Assistant: Loonar Compass automatycznie przejmuje ich ustawienia, rozmowy, pamięć asystenta, szablony i hasła.

- Wcześniejsza wersja zainstalowana tylko dla Ciebie jest usuwana podczas instalacji (pliki programu, skróty i wpis w ustawieniach Windows); Twoje dane zostają.
- Wcześniejszą wersję zainstalowaną dla wszystkich użytkowników kreator pokazuje na stronie „Gotowy do instalacji”. Usuniesz ją przyciskiem **Usuń starą wersję (uprawnienia administratora)** albo później w ustawieniach Windows (Aplikacje). Twoje ustawienia i rozmowy zostają.
- Na pierwszej stronie kreator zaznacza język zapisany w Twoich ustawieniach (ustawienia bez zapisanego języka oznaczają polski), a gdy ustawień nie ma, angielski. Sprawdź ten wybór, zanim klikniesz **Dalej**.
- Zmiana języka w kreatorze nie tłumaczy pamięci asystenta ani szablonów. Jeśli chcesz je przenieść na nowy język, zostaw w kreatorze dotychczasowy język, a po instalacji zmień go w karcie „Język i zatwierdzanie zmian” ustawień (rozdział 5).
- Hasło do bazy, które starsze wersje trzymały w pliku ustawień, trafia do Menedżera poświadczeń Windows, a pliki z polskimi nazwami w folderze danych dostają nazwy angielskie (na przykład `audit_log.json1` i `templates`).

4. Okno aplikacji i nawigacja

Po lewej stronie okna jest granatowy **pasek nawigacji** z ikonami. Po najechaniu myszą na ikonę pojawia się jej nazwa. Kliknięcie przełącza widok. Na dole okna jest pasek z nazwą i wersją aplikacji oraz danymi właściciela.

Ikona	Co zawiera
Czat	Lista rozmów i rozmowa z asystentem (rozdziały 6 i 17)
Ustawienia	Model AI, model EA, Enterprise Architect, język i tryb zatwierdzania, nowe wydania, Confluence, Miro (rozdział 5)
Pamięć asystenta	Ustalenia, które asystent zapamiętał między rozmowami (rozdział 15)
Szablony dokumentów	Szablony, które asystent wypełnia treścią z modelu (rozdział 14)
Dziennik zmian w modelu	Zapisy asystenta do modelu, z eksportem do Excela (rozdział 16)
Changelog	Lista zmian i znanych ograniczeń każdej wersji
Zgłoś błąd	Formularz zgłoszenia błędu ze zdjęciami (rozdział 20)
Wesprzyj projekt (ikona serca, na dole paska, nad ikoną Informacje)	Dobrowolne wsparcie rozwoju aplikacji (opis niżej)
Informacje (na dole paska)	Wersja, właściciel aplikacji, linki (polityka prywatności, kod źródłowy na GitHubie, loonar.eu), informacja o znakach towarowych, licencja i komponenty open source (rozdział 24)

Pomarańczowy znak „!” na ikonach Ustawienia, Pamięć asystenta i Szablony dokumentów oznacza, że po zmianie języka tłumaczenie jeszcze się nie zakończyło (rozdział 5).

Wesprzyj projekt. Loonar Compass jest bezpłatny, także do użytku komercyjnego, i taki pozostanie. Przycisk **Wesprzyj na suppi.pl** otwiera stronę <https://suppi.pl/loonar>, na której możesz dobrowolnie wesprzeć rozwój aplikacji; wpłaty pokrywają między innymi certyfikat do podpisywania kodu, dzięki któremu Windows rozpozna aplikację jako pochodzącą od zweryfikowanego wydawcy. W części „Inne sposoby wsparcia” są linki do kodu źródłowego na GitHubie i do loonar.eu: możesz zgłaszać błędy i pomysły, polecić aplikację innym użytkownikom Enterprise Architect albo dać projektowi gwiazdkę na GitHubie.

Widok czatu

- **Lista rozmów** po lewej jest pogrupowana na „Dzisiaj”, „Wczoraj”, „Ostatnie 7 dni” i „Wcześniej”. Nową rozmowę zaczynasz przyciskiem **+**. Wybrana rozmowa jest podświetlona.
- **Nagłówek rozmowy** pokazuje jej tytuł, przycisk **Model: ...** z modelem tej rozmowy (rozdział 17), a w trakcie pracy asystenta znacznik „Asystent pracuje”.
- **Wiadomości:** Twoje są w beżowym dymku, odpowiedzi asystenta mają ikonę aplikacji. Przy każdej wiadomości jest godzina (przy starszych także data).

- **Ekran nowej rozmowy** pokazuje logo, odpowiedź, co możesz zrobić, oraz model EA, z którym rozmowa będzie pracować, z przyciskami **Wybierz inny model...** i **Zatwierdź model i rozpocznij** (rozdział 17).
- **Karta wiadomości** na dole: pole tekstu, przycisk wklejania, model i tryb zatwierdzania (na przykład „claude-haiku-5-5 · tryb ask”) oraz okrągłe przyciski wysyłania i przerywania (rozdział 6).
- **Przypomnienie** pod kartą wiadomości: „Odpowiedzi i zmiany przygotowuje model AI i mogą zawierać błędy. Sprawdzaj je przed zatwierdzeniem.”

5. Ustawienia i język

Widok **Ustawienia** ma jedenaście kart: **Model AI** i **Język i zatwierdzanie zmian** w lewej kolumnie, **Model** i **Enterprise Architect** w prawej, a pod nimi **Nowe wydania (newsletter)**, **Confluence** i **Miro**. W Compass otwartym w Enterprise Architect nie ma kart Model i Enterprise Architect, bo modelem jest projekt otwarty w tym oknie EA (rozdział 10). Każdą kartę zapisujesz osobno przyciskiem pod nią, a aplikacja od razu sprawdza, czy dane działają. Wynik pojawia się pod przyciskiem: zielony z opisem albo czerwony z przyczyną błędu.

Settings

Each group is saved separately; after saving, the application immediately checks that the details work. Leave a password or key field empty to keep the current value.

1. Claude

Model: claude-haiku-5-5

Max tokens: 10000

Workspace ID: optional, e.g. wrkspc_...

Claude API key: saved

To answer, the app sends the conversation, attachments, remembered notes, templates, model query results and diagram images to Anthropic (Claude API) using your key. Loonar does not receive this data.

Save and test key

2. Model

☒ PostgreSQL database ☐ EA file (.qea, .qeax)

Server: 127.0.0.1

Port: 5432

Database name: loonar-ea-assistant

Database user: postgres

Database password: saved

Connection encryption: Preferred

TLS when the server offers it, otherwise an unencrypted connection. Administrators can enforce TLS for all clients with hostssl entries in pg_hba.conf.

Save and test connection

4. Language and change approval

Language: English

Approval mode: ask

ask - asks before every save

Save

3. Enterprise Architect

Path to EA.exe: C:\Program Files\Sparx Systems\...

EA user account: admin

EA account password: saved

Save and test EA login

Loonar Compass 0.4 · Loonar Mariusz Glowacki, ul. Gombrowicza 12, 73-110 Stargard · loonar.eu · mariusz@loonar.eu

Ustawienia: każda karta z własnym zapisem i testem (zrzut z wcześniejszej wersji: karty miały numery, karta „Model AI” nazywała się „Claude”, a konto EA było w karcie Enterprise Architect)

Karta	Pola	Co sprawdza test
Model AI	Dostawca, Model, Poziom wysiłku, Max tokenów, Workspace ID (tylko Claude) i klucz API wybranego dostawcy: Klucz API Claude albo Klucz API OpenAI (opis niżej). Pod polami jest informacja, co aplikacja wysyła dostawcy (rozdział 18).	„Zapisz i sprawdź klucz” sprawdza klucz krótkim zapytaniem do wybranego modelu
Model	Wybór: Baza PostgreSQL (Serwer, Port, Nazwa bazy, Konto bazy, Hasło do bazy zapisywane w Menedżerze poświadczeń Windows, Szyfrowanie połączenia) albo Plik EA (.qea, .qeax) (Plik modelu; przycisk ... otwiera wybór pliku). Pod spodem konto EA modelu: pole „ Model ma włączone zabezpieczenia (konta użytkowników EA) ”, Konto użytkownika EA i Hasło konta EA (bez zaznaczenia tego pola oba są nieaktywne, a EA otwiera model bez logowania) oraz opcja „ Model wymaga blokady do edycji (Require User Lock to Edit) ”: gdy jest włączona, asystent blokuje zmieniane pakiety i elementy, a potem zwalnia blokady; bez niej niczego nie blokuje.	Baza: „Zapisz i sprawdź połączenie” łączy się z bazą, liczy pakiety, elementy i diagramy, mówi, czy połączenie jest szyfrowane, i ostrzega przed kontem superużytkownika. Plik: „Zapisz i sprawdź plik” otwiera go tylko do odczytu i liczy pakiety, elementy i diagramy. Test zapisuje też w modelu pakiet próbny, sprawdza go i od razu usuwa
Enterprise Architect	Ścieżka do EA.exe (przycisk ... otwiera wybór pliku). Konto EA ustawiasz w karcie Model, bo należy ono do pliku albo bazy modelu.	„Zapisz i sprawdź dostęp do EA” otwiera model w EA (plik modelu, który masz otwarty: w Twoim oknie EA), loguje się na konto z karty Model, zapisuje w modelu pakiet próbny, sprawdza go i od razu usuwa (kilka do kilkunastu sekund)
Język i zatwierdzanie zmian	Język, Tryb (ask, plan albo auto; rozdział 16)	„Zapisz” bez testu; zmiana języka uruchamia przeniesienie treści na nowy język
Nowe wydania (newsletter)	Adres e-mail i zgoda „Chcę dostawać informacje o nowych wydaniach”, z informacją o wykorzystaniu adresu i linkiem do polityki prywatności (rozdział 20)	„Zapisz” przygotowuje wiadomość w programie pocztowym
Confluence	Adres Confluence, E-mail (tylko Cloud), Token, Przestrzeń (klucz), Strona nadrzędna (ID) (rozdział 13)	„Zapisz i sprawdź połączenie z Confluence” łączy się z Confluence i sprawdza przestrzeń

Miro	Token dostępu aplikacji Miro z uprawnieniami boards:read i boards:write (rozdział 12)	Łączy się z Miro i liczy tablice, które widzi token
Jira	Adres Jira, E-mail (tylko Cloud), Token, Domyślny klucz projektu (rozdział 7). Adres musi zaczynać się od <code>https://</code> .	„Zapisz i sprawdź połączenie z Jira” łączy się z Jira i sprawdza projekt
Zespół (Compass Hub)	Adres huba, Token, Ten model w hubie (opcjonalnie) (rozdział 22)	„Zapisz i sprawdź połączenie z hubem” łączy się z hubem i pokazuje Twoje konto, rolę i politykę zespołu
Serwery MCP	Lista własnych serwerów MCP, zapisana w pliku <code>mcp_servers.json</code> obok ustawień, lista Dodaj gotowy serwer i pole Token serwera MCP (rozdział 7)	„Zapisz i sprawdź serwery MCP” łączy się z każdym serwerem i podaje liczbę jego narzędzi
Agenci zdarzeń (Compass w EA)	Reguły w pliku <code>event_agents.json</code> (rozdział 10)	„Zapisz agentów zdarzeń” sprawdza reguły i podaje liczbę agentów

Puste pole hasła, klucza lub tokena oznacza „bez zmian”. Napis **zapisano** obok pola mówi, że wartość jest już zapamiętana w Menedżerze poświadczeń, a **BRAK**, że jeszcze jej nie ma. Hasło do bazy nie może zawierać średnika (;), bo Enterprise Architect nie przekaże go w swoim ciągu połączenia. Adres Confluence musi zaczynać się od `https://`; adres `http://` jest dopuszczalny tylko dla serwera na tym samym komputerze.

Wartości domyślne. Gdy nie ma jeszcze ustawień, pola serwera, nazwy bazy, konta użytkownika EA i haseł są puste (przy serwerze szara podpowiedź „np. 127.0.0.1”), port to 5432, konto bazy to `postgres`, ścieżka do EA to `C:\Program Files\Sparx Systems\EA\EA.exe`, tryb zatwierdzania to **ask**, dostawca to Anthropic (Claude), model to `claude-haiku-5-5`, poziom wysiłku to „domyślny dla modelu”, a Max tokenów to 10000. Jeśli brakuje ustawień albo któregoś z haseł lub klucza, aplikacja otwiera się w widoku Ustawienia i podaje, co trzeba uzupełnić.

Max tokenów. Jeśli asystent przerywa pracę komunikatem o uciętej odpowiedzi, zwiększ tę wartość (na przykład do 20000). Sam limit nic nie kosztuje: płacisz za faktycznie wygenerowany tekst.

Blokowanie pakietów i elementów przy zapisie zależy od opcji „Model wymaga blokady do edycji (Require User Lock to Edit)” w karcie Model: zaznacz ją, gdy model ma w EA włączone „Require User Lock to Edit”.

Model AI: dostawca, model i poziom wysiłku

Karta **Model AI** decyduje, który model językowy odpowiada w rozmowach:

Pole	Znaczenie
Dostawca	Anthropic (Claude) (domyślnie) albo OpenAI .
Model	Pole tekstowe z listą ▼. Claude: <code>claude-haiku-5-5</code> (domyślny), <code>claude-sonnet-5-5</code> , <code>claude-opus-5-5</code> , <code>claude-fable-5-1</code> . OpenAI: <code>gpt-5.1</code> , <code>gpt-5</code> , <code>gpt-5-mini</code> . Możesz też wpisać identyfikator innego modelu wybranego dostawcy.
Poziom wysiłku	domyślny dla modelu (domyślnie), low , medium , high , xhigh albo max . Wyższy poziom daje dokładniejsze odpowiedzi i pracę na modelu, ale zużywa więcej tokenów i czasu, więc więcej kosztuje; niższy działa szybciej i taniej. Przy OpenAI poziom przekłada się na reasoning effort modelu.
Max tokenów	Limit długości jednej odpowiedzi modelu (opis niżej).
Workspace ID	Tylko Claude, opcjonalnie: identyfikator obszaru roboczego (workspace) z konsoli Anthropic, na przykład <code>wrkspc_...</code> .
Endpoint	Tylko OpenAI. Puste = OpenAI. Możesz wpisać adres innego serwera zgodnego z API OpenAI: Azure OpenAI (<code>https://<zasób>.openai.azure.com/openai/v1</code> ; klucz idzie nagłówkiem <code>api-key</code>), Ollama (<code>http://localhost:11434/v1</code>), LM Studio (<code>http://localhost:1234/v1</code>) albo firmową bramkę. Serwery lokalne działają bez klucza, a <code>http://</code> jest dopuszczalny tylko dla localhost. W polu Model wpisz wtedy nazwę modelu (w Azure: nazwę wdrożenia, deployment).
Klucz API Claude / Klucz API OpenAI	Klucz wybranego dostawcy. Klucz Claude tworzysz w konsoli Anthropic (Anthropic Console). Klucz OpenAI tworzysz na platformie OpenAI: <code>platform.openai.com</code> → API keys → Create new secret key . Klucze są zapisywane w Menedżerze poświadczeń Windows, każdy osobno.

Zapisz i sprawdź klucz sprawdza klucz krótkim zapytaniem do wybranego modelu. Dane potrzebne do odpowiedzi trafiają wyłącznie do wybranego dostawcy: do `api.anthropic.com` albo do `api.openai.com` (rozdział 18). Jeśli niczego w tej karcie nie zmienisz, aplikacja działa tak jak wcześniej: z Claude Haiku 5.5 i poziomem wysiłku domyślnym dla modelu. Większe modele i wyższy poziom wysiłku kosztują więcej; ceny podają Anthropic i OpenAI.

Zmiana języka

Aplikacja działa w czterech językach: **polskim, angielskim, niemieckim i francuskim**. Język wybierasz w instalatorze, a zmieniasz w karcie „**Język i zatwierdzanie zmian**”:

1. Wybierz język z listy **Język**.
2. Kliknij **Zapisz**. Interfejs od razu przełącza się na nowy język, a asystent odpowiada w nim od kolejnej wiadomości (jeśli napiszesz do niego w innym języku, odpowie w Twoim).
3. Aplikacja przenosi Twoje treści na nowy język: niezmodyfikowane szablony domyślne od razu zastępuje wersjami w nowym języku, a **pamięć asystenta oraz własne i zmodyfikowane szablony tłumaczy model AI**.

Tłumaczone są tylko pamięć asystenta i szablony. **Historia rozmów nigdy nie jest tłumaczona**: rozmowy zostają w języku, w którym je prowadziłeś.

Brak połączenia z modelem AI. Tłumaczenie wymaga działającego klucza API. Jeśli połączenia nie ma, na ikonach **Ustawienia**, **Pamięć asystenta** i **Szablony dokumentów** pojawia się pomarańczowy znak „!”, a odpowiedź po najechnięciu myszą wyjaśnia, że treści nie są jeszcze przetłumaczone. Tłumaczenie wznowi się przy następnym uruchomieniu aplikacji albo od razu po kliknięciu **Przetłumacz teraz** w karcie „Język i zatwierdzanie zmian”. Zapisywanie innych kart go nie wznowia.

Po zapisaniu języka Compass przełącza też korektę pisowni samego EA na ten język, jeśli EA ma dla niego słownik. EA ma tylko słowniki angielskie, więc dla pozostałych języków teksty sprawdza Compass (rozdział 7, „Pisownia w języku Compassa”).

Szyfrowanie połączenia z bazą

Pole **Szyfrowanie połączenia** w karcie „Model” (opcja „Baza PostgreSQL”) decyduje, czy odczyt modelu przez asystenta idzie przez TLS:

Ustawienie	Działanie
Preferowane (domyślnie)	TLS, gdy serwer go oferuje; w przeciwnym razie połączenie bez szyfrowania. Działa z każdym serwerem.
Wymagane	TLS jest wymagany: bez niego aplikacja się nie połączy. Certyfikat serwera nie jest sprawdzany.
Wymagane, z weryfikacją certyfikatu	TLS jest wymagany, a certyfikat serwera jest sprawdzany w magazynie certyfikatów Windows (także certyfikatów firmowych), razem z nazwą serwera.

- Test połączenia mówi, czy połączenie jest szyfrowane: „Połączenie jest szyfrowane (TLS).” albo „Połączenie NIE jest szyfrowane: serwer nie oferuje TLS.”.
- Przy weryfikacji certyfikatu nazwa w polu **Serwer** musi się zgadzać z nazwą w certyfikacie serwera.
- Administrator bazy może wymusić TLS dla wszystkich klientów wpisami `hostssl` w pliku `pg_hba.conf`.
- Ustawienie dotyczy odczytu modelu przez asystenta. Połączenie samego Enterprise Architect z bazą (przy zapisie) zależy od ustawień EA.

6. Rozmowa z asystentem

Conversations

Today

In a new package "Web s..."

In a new package "Online sho..."

In a new package "Web store", draw a BPMN diagram

Save as...

Open full size

Order_process.png

15:06

I created the package **Web store** in the model root and drew the BPMN diagram "**Order process**" in it. The diagram is below.

The process runs left to right:

- Order placed by customer** (start event).
- Payment** is an embedded subprocess, with its steps drawn inside it: **Choose payment method** (user task), **Authorise payment** and **Confirm payment** (service tasks), bracketed by the start event **Payment started** and the end event **Payment confirmed**.
- Ship order** (service task).
- Notify customer** (send task).
- Order completed** (end event).

All elements are connected with sequence flows. The layout is clean. The text in "Choose payment method" runs slightly past the edge of its box.

15:06

Write a message, drag in files or paste an image (Ctrl+V)...

Drag files into the window to attach them

claude-haiku-5-5 · mode ask

Answers and changes are prepared by an AI model (Claude by Anthropic) and may contain errors. Review them before you approve.

Ready.

Loonar Compass 0.3 · Loonar Mariusz Glowacki, ul. Gombrowicza 12, 73-110 Stargard · loonar.eu · mariusz@loonar.eu

Okno rozmowy: lista rozmów po lewej, czat z obrazkiem diagramu, karta wiadomości na dole

Pisanie

Wiadomość wpisujesz w **karcie wiadomości** na dole okna. Karta zawiera pole tekstu, przycisk wklejania, nazwę modelu i bieżący tryb zatwierdzania oraz okrągłe przyciski: **Wyślij**, a w trakcie pracy asystenta **Przerwij**. Pod kartą jest przypomnienie, że odpowiedzi i zmiany przygotowuje model AI i trzeba je sprawdzać przed zatwierdzeniem.

Klawisz	Działanie
Enter	Wysyła wiadomość
Shift + Enter	Nowa linia w wiadomości
↑ w pustym polu	Przywołuje Twoją ostatnią wiadomość z tej rozmowy; kolejne naciśnięcia cofają się do wcześniejszych, ↓ wraca
Ctrl + V	Wkleja tekst, a gdy w schowku jest obraz albo pliki, dodaje je jako załączniki

Pole wiadomości rośnie razem z tekstem. Błędy pisowni są podkreślane na czerwono słownikiem systemu Windows, w języku z ustawień systemu. Kliknij podkreślone słowo prawym przyciskiem myszy i wybierz **Dodaj „...” do słownika**: słowo trafia do słownika Windows, a polskie wyrażenie we wszystkich formach

odmiany (w liczbie pojedynczej i mnogiej, od mianownika do wołacza). W tym samym menu **Dodaj „...”** do słownika projektu EA prosi asystenta o dodanie pojęcia do słownika projektu (rozdział 7).

Załączniki

Do wiadomości możesz dołączyć do 5 plików, każdy do 10 MB. Sposoby: przeciągnij pliki do okna (z Eksploratora albo z przeglądarki), wklej je ze schowka (**Ctrl** + **V**) albo użyj przycisku wklejania w karcie wiadomości.

Rodzaj pliku	Co widzi asystent
Obrazy (PNG, JPG, GIF, WebP)	Obraz: zdjęcie tablicy, szkic, zrzut ekranu. Na czacie PNG i JPG widać jako obrazek, a GIF i WebP jako kartę pliku.
PDF	Cały dokument
DOCX	Tekst dokumentu
Pliki tekstowe w kodowaniu UTF-8 (BPMN, XML, CSV, JSON, kod...)	Treść pliku
ZIP	Pliki z jego wnętrza, każdy według swojego rodzaju: najwyżej 20 plików i 30 MB łącznie (każdy do 10 MB). ZIP wewnątrz ZIP-a nie jest rozpakowywany.
Inne pliki binarne (XLSX, PPTX, DOC...)	Tylko informację o dołączonym pliku: aplikacja nie odczytuje ich treści

W trakcie pracy asystenta

- Pod wiadomościami widać **kartę kroków**: co asystent właśnie robi, ile to trwa i jaki był wynik każdego kroku. W trakcie pracy karta jest rozwinięta, a po zakończeniu zwiija się do jednej linii; kliknięcie nagłówka karty rozwija ją ponownie.
- W polskiej wersji interfejsu wynik kroku to krótkie podsumowanie, na przykład „wiersze: 12”, „zapisano w modelu EA” albo „plik gotowy”. Pełną treść wyniku (po angielsku) lista pokazuje tylko w angielskiej wersji interfejsu.
- Możesz **pisać dalej**: kolejna wiadomość trafia do kolejki i asystent uwzględni ją przy najbliższym kroku, bez przerywania pracy.
- Przycisk **Przerwij** zatrzymuje pracę po bieżącym kroku. Krok, który już trwa (na przykład zapis do EA), dobiega końca.
- Jedna odpowiedź może mieć najwyżej 24 kroki. Jeśli asystent nie zdąży w tym limicie, przerywa pracę z komunikatem, że wykonał maksymalną liczbę kroków; poproś go wtedy w nowej wiadomości, żeby kontynuował.

Conversations

+ Search conversations...

No saved conversations

In a new package "Online shop", draw a BPMN diagram
Claude is working

In a new package "Online shop", draw a BPMN diagram of the order process: the customer places an order, payment as an embedded subprocess (choose method, authorise, confirm), then shipping and a customer notification. Show me the diagram at the end.

☰ Claude is working · step 9
collapse

```

14:49:14 ✓ Claude is analyzing and planning (step 1) (3 s) → next: db_query, 365 response tokens
14:49:17 ✓ Reading the model database: SELECT package_id, name, parent_id, ea_guid FROM t_package WHERE name ILIKE '%online%' OR name ILIKE '%shop%' OR parent_id IS NULL ORDER BY package_id (0 s) → package_id | name | parent_id | ea_guid 80 | Online shop | 1 | {295F53C2-78B5-4a31-A4DB-172CB0837B05}
14:49:17 ✓ Claude is analyzing and planning (step 2) (2 s) → next: db_query, 219 response tokens
14:49:19 ✗ Reading the model database: SELECT o.object_id, o.name, o.object_type, o.stereotype, o.package_id FROM t_object o WHERE o.package_id IN (SELECT package_id FROM t_package WHERE package_id=8... (0 s) → one query at a time (no semicolons inside)
14:49:19 ✓ Claude is analyzing and planning (step 3) (1 s) → next: db_query, db_query, 201 response tokens
14:49:20 ✓ Reading the model database: SELECT o.object_id, o.name, o.object_type, o.stereotype, o.package_id FROM t_object o WHERE o.package_id = 80 ORDER BY 1 (0 s) → object_id | name | object_type | stereotype | package_id 652 | Order placed by customer | Event | StartEvent | 80 653 | Payment | Activity | Activity | 80 654 | Payment started | Event | StartEvent | ...
14:49:20 ✓ Reading the model database: SELECT d.diagram_id, d.name, d.diagram_type, d.package_id FROM t_diagram d WHERE d.package_id = 80 (0 s) → diagram_id | name | diagram_type | package_id 71 | Order process | Analysis | 80
14:49:20 ✓ Claude is analyzing and planning (step 4) (1 s) → next: ea_show_diagram, 134 response tokens
14:49:22 🖼️ Getting diagram 71 from EA as an image (2 s)
```

Add a message: Claude will take it into account while working...

📎 Drag files into the window to attach them

claude-haiku-5-5 · mode ask

⏹️ ➡️

Answers and changes are prepared by an AI model (Claude by Anthropic) and may contain errors. Review them before you approve.

🔊 Claude is getting a diagram from EA...

Loonar Compass 0.3 · Loonar Mariusz Glowacki, ul. Gombrowicza 12, 73-110 Stargard · loonareu · mariusz@loonareu

Co potrafi asystent? Asystent zna funkcje aplikacji z changelogu. Zapytaj go: „Co potrafisz w tej aplikacji?”.

7. Praca z modelem Enterprise Architect

Pisz tak, jak do współpracownika, który zna EA. Wskazuj pakiety i diagramy po nazwie. Asystent wskazuje obiekty modelu nazwą, pakietem i GUID. Poniżej przykłady poleceń; przykłady uporządkowane według celu są w rozdziale 8.

Odczyt

Jakie pakiety są w modelu i ile diagramów jest w każdym z nich?

Opisz, co przedstawia diagram „Proces zamówienia”, i wskaż elementy bez połączeń.

Pokaż diagram „Przypadki użycia sklepu”.

Wypisz atrybuty i operacje klasy „Zamówienie” razem z parametrami.

Pokazany diagram pojawia się na czacie jako obrazek. Pod nim są przyciski **Zapisz jako...** i **Otwórz w pełnym rozmiarze**.

Analiza wpływu zmiany

Asystent idzie od wskazanego elementu po relacjach w głąb modelu (do 6 poziomów, w jedną albo obie strony, opcjonalnie tylko po wybranych typach relacji). Dla każdego dotkniętego elementu podaje relację, przez którą dochodzi wpływ, ścieżkę od elementu startowego i diagramy, na których element leży. Wynik możesz dostać jako opis, obrazek, dokument Word albo arkusz Excel.

Co się zmieni, jeśli zmienimy komponent „Obsługa płatności”? Sprawdź 3 poziomy w obie strony i daj wynik w Excelu.

Gdzie użyte są operacje

Dla klasy, komponentu albo interfejsu asystent znajduje, gdzie są wywoływane operacje (usługi) tego elementu: na diagramach sekwencji (komunikaty do elementu albo jego linii życia) i aktywności (akcje wywołujące operację), z pakietem i wywołującym. Podaje też operacje nieużywane na żadnym diagramie.

Gdzie są używane operacje interfejsu „API zamówień”? Które nie są użyte nigdzie?

Rysowanie

Utwórz pakiet „Sklep” pod pakietem „Model” i narysuj w nim diagram przypadków użycia: klient przegląda katalog, dodaje produkt do koszyka, składa zamówienie i płaci. Dodaj aktorów i krótkie opisy przypadków.

Przerysuj proces z załączonego zdjęcia jako diagram BPMN w pakiecie „Procesy”.

Z załączonego dokumentu wypisz wymagania, utwórz je w pakiecie „Wymagania” i pokaż na diagramie wymagań.

Podprocesy BPMN

- **Osadzony (embedded):** kroki podprocesu są jego elementami podrzędnymi i są narysowane w jego prostokącie na tym samym diagramie.
- **Zwinięty (collapsed):** kroki są na osobnym diagramie, a dwukrotne kliknięcie podprocesu otwiera ten diagram. Asystent może też powiązać podproces (albo inny element) z istniejącym diagramem jako diagramem złożonym.
- Do istniejącego podprocesu asystent dodaje nowe kroki.

W procesie „Obsługa reklamacji” zamień zadanie „Weryfikacja” na podproces osadzony z trzema krokami: sprawdź dokumenty, sprawdź gwarancję, podejmij decyzję.

Zmiany w istniejącym modelu

Przenieś wymagania FR01–FR05 do pakietu „Katalog i koszyk”.

Na diagramie „Proces zamówienia” pokoloruj zadania wykonywane przez system na zielono, a ręczne na żółto.

Dodaj do klasy „Zamówienie” atrybut „status” typu String i operację „anuluj” z parametrem „powód”.

Zdejmij z diagramu „Proces zamówienia” element „Faks”, ale zostaw go w modelu.

Usuń z diagramu „Start” komórkę prowadzącą do diagramu „Archiwum”.

Dodaj diagram „Start” z kafelkami-linkami (navigation cells) do wszystkich diagramów przypadków użycia, po cztery w wierszu.

Zmień nazwę diagramu „Szkic” na „Proces zwrotu” i dodaj mu krótki opis.

Powiększ pulę „Magazyn” na diagramie „Proces wysyłki”, żeby zmieściły się w niej wszystkie zadania.

Asystent zmienia też położenie i rozmiar elementu na diagramie bez zdejmowania go (połączenia zostają), wartości tagów elementów oraz nazwę, opis i stereotyp istniejącego diagramu.

Słowa kluczowe i legendy

Asystent ustawia **słowa kluczowe** elementów (pole Keywords w EA) i tworzy **legendę diagramu z automatycznym kolorowaniem**, jak w EA: elementy (według słowa kluczowego, stereotypu, statusu albo typu) i połączenia (na przykład według stereotypu) dostają kolor z legendy, a legenda pokazuje wartość, kolor i podpis.

Oznacz wymagania słowem kluczowym „MVP” albo „później” i dodaj na diagramie legendę, która koloruje je według tego słowa.

Linie połączeń

- **Styl linii** (jak Line Style w EA): prosta, automatyczna, łamana, prostokątna, prostokątna zaokrąglona, drzewo i boczna; dla wybranych połączeń albo wszystkich na diagramie.

- **Trasa połączenia:** asystent prowadzi linię przez wskazane punkty załamania i miejsca styku z bokami elementów, na przykład według szkicu na rzucie ekranu.
- **Ukrywanie połączeń** na jednym diagramie (i ponowne pokazywanie): relacja zostaje w modelu i na innych diagramach, więc gęsty diagram odciążysz bez usuwania relacji.

Na diagramie „Architektura” ustaw wszystkie połączenia jako prostokątne i ukryj zależności do komponentu „Logowanie”.

Kopia pakietu i baseline

- **Kopia pakietu** tworzy duplikat pakietu z całą zawartością.
- **Baseline** to zapisana wersja pakietu w EA. Asystent ma polecenie robić baseline przed usuwaniem albo większą przebudową istniejącego pakietu, ale aplikacja tego nie wymusza. Przy ważnych pakietach poproś o baseline wprost i pracuj w trybie **ask**.
- **Przywracanie** wykonujesz w Enterprise Architect: kliknij pakiet prawym przyciskiem myszy, wybierz **Manage Baselines**, zaznacz baseline i użyj **Compare/Restore**. Asystent nie przywraca baseline'ów sam, bo przywracanie przez automatyzację zawiesza EA.

Słownik projektu EA

Asystent dodaje pojęcia do słownika projektu EA (Project Glossary): pojęcie z definicją i typem, domyślnie typ **Słownik**. Polskie pojęcie trafia do słownika we wszystkich formach odmiany (w liczbie pojedynczej i mnogiej, od mianownika do wołacza), więc EA podświetla w notatkach każdą jego formę. Pojęcie dodasz prośbą w rozmowie albo z pola wiadomości: kliknij słowo prawym przyciskiem myszy i wybierz **Dodaj „...” do słownika projektu EA**. Jeśli asystent nie zna definicji albo typu pojęcia, zapyta Cię o nie.

Dodaj do słownika projektu pojęcie „zamówienie”: zlecenie klienta na dostawę wybranych produktów.

Scenariusze elementów

Asystent tworzy, zmienia i usuwa scenariusze elementów, na przykład przypadków użycia. Są to scenariusze strukturalne, widoczne w EA w zakładce **Scenarios** elementu, z krokami wykonywanymi przez aktora albo przez system.

Dodaj do przypadku użycia „Złóż zamówienie” scenariusz główny: klient potwierdza koszyk, system liczy wartość zamówienia, klient wybiera płatność, system zapisuje zamówienie.

Kontrola jakości modelu

Jednym poleceniem asystent sprawdza pakiet razem z podpakietami. Sprawdzenie wykonuje program, a nie model AI, więc nic nie jest zgadywane. Wbudowane kontrole:

- domyślne i puste nazwy (na przykład Class1, Activity 2),
- elementy bez opisu,
- osierocone elementy (bez relacji i na żadnym diagramie),
- powtórzone nazwy w pakiecie i powtórzone relacje,
- relacje elementu do samego siebie oraz do nieistniejących elementów,
- relacje niedozwolone w ArchiMate 3.2 (opis niżej),
- puste pakiety i puste diagramy.

Każde ustalenie ma wagę: **błąd**, **ostrzeżenie** albo **informacja**. Poprawki asystent wprowadza dopiero po Twojej zgodzie.

Sprawdź jakość pakietu Zamówienia.

Reguły organizacji. Własne reguły wpisujesz w pliku `governance.json`, który leży obok `app.properties` (pliku ustawień w folderze danych użytkownika). Gdy takiego pliku nie ma, a Compass jest połączony z hubem zespołu, reguły pochodzą z polityki zespołu (rozdział 22). Przykład:

```
{
  "naming": [{
    "applies_to": "Component",
    "pattern": "*Service",
    "message": "Nazwa komponentu kończy się na Service"
  }],
  "required_tags": [{
    "applies_to": «ArchiMate_ApplicationComponent»,
    "tag": "Owner"
  }],
  "allowed_relations": [{
    "connector_type": "Realisation",
    "source": "Component",
    "target": "Interface"
  }]
}
```

- **naming:** nazwy elementów danego typu muszą pasować do wzorca (`pattern`); `message` to treść ustalenia.
- **required_tags:** element danego typu musi mieć wskazany tagged value.
- **allowed_relations:** typ relacji wpisany na tej liście jest dozwolony tylko między wymienionymi końcami; typy relacji, których na liście nie ma, nie są ograniczane.

W `applies_to` podajesz typ elementu albo «stereotyp»; działają symbole wieloznaczne `*` i `?`.

Relacje ArchiMate 3.2. Gdy relacja i oba jej końce mają stereotypy ArchiMate (na przykład «ArchiMate_ApplicationComponent», «ArchiMate_DataObject»), kontrola stosuje też reguły relacji ArchiMate 3.2, bez żadnej konfiguracji. Zgłasza tylko pewne naruszenia, jako błędy; połączeń, które mogłyby uzasadnić relacja wprowadzona, nie zgłasza. Na przykład:

- relacja access musi kończyć się na elemencie struktury pasywnej (obiekt biznesowy, obiekt danych, artefakt...) i zaczynać od elementu zachowania lub struktury aktywnej,
- relacja influence musi kończyć się na elemencie motywacji,
- relacje triggering, flow i serving nie łączą elementów struktury pasywnej ani elementów motywacji,
- assignment prowadzi od struktury aktywnej do zachowania (albo od węzła do artefaktu),
- specialization łączy dwa elementy tego samego typu,
- element motywacji realizuje (realization) tylko inny element motywacji.

Węzły połączeń (junction) i grupowania (grouping) nie są sprawdzane. Kontrola na bieżąco w Compass w EA (rozdział 10) stosuje te same reguły. Reguły ArchiMate wyłączysz wpisem `"archimate": false` w `governance.json`.

Sprawdź w pakiecie Krajobraz aplikacji, które relacje są niedozwolone w ArchiMate 3.2.

Analiza luk as-is / to-be

Asystent porównuje dwa pakiety, na przykład stan obecny i docelowy, i podaje, co jest **nowe**, co jest **do wycofania** i co się **zmieniło** (stereotyp, opis, relacje). Wynik możesz zapisać do Excela.

Porównaj pakiet „Sklep as-is” z pakietem „Sklep to-be” i zapisz luki do Excela.

Łańcuch wymagań: model, kod, testy, zgłoszenia

Gdy zestawiasz wymagania z modelem (podrozdział 8.5) i podasz też folder z kodem, asystent (narzędzie `trace_requirements`) prowadzi każde wymaganie dalej: wymaganie → element modelu → kod → test → klucze zgłoszeń. Folder czyta tylko do odczytu.

- **Kod:** pliki, które deklarują typ nazwany tak jak powiązany element modelu, oraz pliki źródłowe, które wymieniają identyfikator wymagania.
- **Testy:** pliki testów (w folderze testów albo o nazwach w rodzaju `OrderTest.java`, `order.spec.ts`, `test_order.py`), które wymieniają identyfikator wymagania albo dotyczą typu z tego kodu.
- **Zgłoszenia:** klucze w rodzaju `ARCH-123` w treści wymagania i w powiązanych elementach (nazwa, alias, notatki, tagged values).

Każde wymaganie dostaje status: **kompletny** albo przerwany: **brak w modelu**, **brak kodu** lub **brak testów**.

Prześledź wymagania z C:\Dokumenty\BRD.docx przez model do kodu i testów w C:\repo\sklep: gdzie łańcuch się przerywa?

CMDB a model

Asystent porównuje pozycje konfiguracji (CI) z bazy CMDB z modelem (narzędzie `cmdb_compare`). Tylko czyta: niczego nie zapisuje ani w modelu, ani w CMDB.

- **Dane:** eksport CSV albo JSON, na przykład eksport listy z ServiceNow albo JSON z jego Table API (`{ "result": [...] }`), podany jako ścieżka albo załącznik, albo dane, które w rozmowie zwrócił serwer MCP bazy CMDB. Kolumny takie jak nazwa (name), sys_id, klasa (class), właściciel (owned_by) i status (install_status) są rozpoznawane po nagłówkach.
- **Dopasowanie:** po identyfikatorze CMDB zapisanym w tagged value elementu (`sys_id` albo `CMDB id`), a gdy go nie ma, po nazwie.
- **Elementy:** domyślnie komponenty, węzły, urządzenia, środowiska wykonawcze oraz komponenty aplikacji, węzły, oprogramowanie systemowe i usługi ArchiMate. Możesz wskazać inne typy albo stereotypy i zawęzić porównanie do pakietu.
- **Raport:** tylko w CMDB, tylko w modelu oraz inny właściciel (tagged value Owner) albo status.

Porównaj eksport CMDB C:\Eksport\cmdb_ci_appl.csv z komponentami modelu.

Pobierz z ServiceNow aplikacje (CI) i porównaj je z modelem.

Drugie polecenie wymaga serwera MCP ServiceNow w karcie **Serwery MCP** (zobacz „Własne serwery MCP” niżej). Na życzenie asystent dopisuje do modelu brakujące elementy, właścicieli albo identyfikatory (po Twojej zgodzie) albo zapisuje raport do Excela. W potoku CI to samo porównanie robi opcja `--cmdb` (rozdział 23).

Zgłoszenia Jira

Po uzupełnieniu karty **Jira** (rozdział 5) asystent czyta zgłoszenia po kluczu albo wyszukiwaniem JQL i używa ich jako kontekstu: rysuje diagramy na podstawie opisu zgłoszenia (story) albo sprawdza, czy model pokrywa jego wymagania. **Zawsze po Twojej zgodzie w panelu zatwierdzania** tworzy zgłoszenia, dodaje komentarze i linki do elementów modelu.

Pole karty „Jira”	Co wpisać
Adres Jira	Cloud: <code>https://firma.atlassian.net</code> ; Data Center: adres serwera firmy. Musi zaczynać się od <code>https://</code> .
E-mail (tylko Cloud)	Cloud: e-mail konta Atlassian. Przy osobistym tokenie dostępu (Data Center) zostaw puste.
Token	Cloud: token API konta (tworzysz go tak jak dla Confluence, rozdział 13). Data Center: osobisty token dostępu. Token jest zapisywany w Menedżerze poświadczeń Windows.
Domyślny klucz projektu	Na przykład <code>ARCH</code> : projekt, w którym powstają nowe zgłoszenia, gdy nie wskażesz innego.

Narysuj diagram sekwencji na podstawie zgłoszenia ARCH-123.

Dodaj do zgłoszenia ARCH-123 link do elementu „Koszyk”.

Sprawdzenie zapisu

Po każdym zapisie Compass czyta model jeszcze raz i porównuje go z prośbą: czy utworzone obiekty istnieją, czy nowe relacje mają wskazane końce, czy nowe elementy są we wskazanym pakiecie, czy zmienione nazwy, opisy, stereotypy i tagged values mają nowe wartości oraz czy usunięte obiekty zniknęły. Różnice trafiają do asystenta, który ma o nich powiedzieć, a nie napisać „gotowe”.

Zapis jest **wszystko albo nic**: jeśli nie uda się w połowie, obiekty, które już powstały, są usuwane. Usunięcie elementu, który ma relacje, zawsze wymaga Twojej zgody (także w trybie **auto**), a lista tych relacji jest pokazana w panelu zgody.

Zawsze z pytaniem i pełnymi danymi. Zapis plików (eksport XML, dokumenty z szablonów dokumentów EA) i import XML zawsze wymagają Twojej zgody, także w trybie **auto**, a panel pokazuje pełną ścieżkę. Przyjmowane są tylko pełne ścieżki na dysku lokalnym: udziały sieciowe oraz foldery systemowe, programów i danych aplikacji są odrzucane. Identyfikator musi być liczbą. Zapis ustalenia w pamięci asystenta, zapis szablonu dokumentu oraz zapis i uruchomienie skryptu EA też zawsze wymagają zgody, a panel pokazuje cały tekst albo kod (teksty dłuższe niż 20 000 znaków są odrzucane); skrypt działa dokładnie tak, jak go pokazano, razem z dołączeniami `!INC` . Gdy asystent chce przeczytać plik, który nie jest załącznikiem rozmowy (dokument wymagań, folder z kodem), najpierw pyta.

Własne serwery MCP

Własne serwery MCP dają asystentowi dodatkowe narzędzia, na przykład system zgłoszeń, repozytorium albo firmową bazę wiedzy. Serwery wpisujesz w pliku `mcp_servers.json` obok pliku ustawień, a sprawdzasz w karcie **Serwery MCP**. Serwer może być programem lokalnym (`command` i `args` , komunikacja przez stdio) albo usługą HTTP (`url` i `headers` , Streamable HTTP):

```
{
  "servers": [
    {
      "name": "hub",
      "url": "http://127.0.0.1:8730/mcp",
      "headers": {
        "Authorization": "Bearer ${secret:hub_token}"
      }
    },
    {
      "name": "files",
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "C:/docs"
      ]
    }
  ]
}
```

- `${secret:nazwa}` to wpis w Menedżerze poświadczeń Windows, a `${env:NAZWA}` to zmienna środowiska; dzięki temu tokeny nie leżą w pliku.

- Każde wywołanie narzędzia pyta o Twoją zgodę, chyba że serwer ma `"trusted": true`. Ustawiaj to tylko dla serwerów, którym ufasz: to, że serwer sam deklaruje narzędzie jako tylko do odczytu, nie wystarcza.
- Własnych kluczy i haseł Compassa (klucze dostawców AI, hasła i tokeny bazy, EA, Confluence, Jiry i Miro) nie można użyć w `${secret:...}`. Jedyny wyjątek to `${secret:hub_token}` w nagłówkach własnego punktu MCP huba zespołu (adres huba z Ustawień ze ścieżką `/mcp`), jak w przykładzie powyżej. Tokeny innych serwerów zapisz pod ich własnymi nazwami w Menedżerze poświadczeń Windows.
- Program lokalny (`command`) dostaje tylko podstawowe zmienne środowiska Windows (PATH, TEMP, foldery profilu użytkownika, ustawienia proxy...) i zmienne wymienione w jego `"env"`.
- `"command": "npx"` działa: Compass uruchamia `npx.cmd` (tak samo npm, pnpm i yarn). Serwery uruchamiane przez npx wymagają Node.js na komputerze.
- Przycisk **Zapisz i sprawdź serwery MCP** łączy się z każdym serwerem i pokazuje liczbę jego narzędzi.

Gotowe serwery. W karcie **Serwery MCP** wybierz serwer z listy **Dodaj gotowy serwer** i kliknij **Dodaj**: jego wpis trafia do tekstu `mcp_servers.json`, gdzie go uzupełniasz i zapisujesz kartę. Nazwa, która już jest użyta, dostaje numer.

Serwer	Co uzupełnić
Azure DevOps	Serwer lokalny uruchamiany przez <code>npx</code> : w miejsce <code>YOUR-ORGANIZATION</code> wpisz swoją organizację. Logujesz się kontem Microsoft.
GitHub	Serwer zdalny <code>https://api.githubcopilot.com/mcp/</code> : zapisz osobisty token dostępu GitHub (personal access token) jako <code>mcp_github</code> .
ServiceNow / CMDB	W miejsce <code>YOUR-SERVICENOW-MCP-SERVER</code> wpisz adres serwera MCP zatwierdzonego w Twojej firmie, a jego token zapisz jako <code>mcp_servicenow</code> .
Files (a local folder)	Serwer lokalny uruchamiany przez <code>npx</code> , który czyta folder: zmień <code>C:/docs</code> na swój folder.
Compass Hub	Na liście tylko wtedy, gdy adres huba jest wpisany w karcie Zespół (Compass Hub) : serwer MCP huba z Twoim tokenem huba, oznaczony <code>"trusted": true</code> .

Token serwera MCP. Wpisz nazwę zaczynającą się od `mcp_` (litera, cyfry i `_`, na przykład `mcp_github`) i token, a potem kliknij **Zapisz i sprawdź serwery MCP**. Token trafia do Menedżera poświadczeń Windows; w `mcp_servers.json` używasz go jako `${secret:mcp_github}`.

Skrypty EA

Asystent pokazuje grupy skryptów i skrypty modelu, czyta skrypt, zapisuje skrypt (JScript, VBScript albo JavaScript; brakująca grupa powstaje sama) i go uruchamia. **Zapis i uruchomienie skryptu wymagają zgody za każdym razem**, także w trybie **auto**, bo skrypt może zmienić w modelu wszystko; panel zgody pokazuje cały kod. Skrypt działa w otwartym oknie EA albo, gdy go nie ma, w osobnej sesji EA działającej w tle. `Repository` i `Session.Output` działają jak w EA, a dołączenia `!INC Grupa.Skrypt` są rozwiązywane z modelu. Jeśli w EA jest otwarty inny model, uruchomienie kończy się komunikatem.

Pokaż skrypty w modelu, a potem uruchom skrypt „Raport.Brakujące opisy”.

Dalsze operacje na modelu

- **Automatyczny układ diagramu:** kierunek w dół, w prawo, w górę albo w lewo i odstęp.
- **XMI:** eksport pakietu do pliku XML i import pakietu z pliku (import zastępuje identyfikatory GUID, chyba że poprosisz o ich zachowanie).
- **Dokumenty powiązane** elementów (linked documents), tworzone z tekstu.
- **Pasy (swimlanes)** diagramu.
- **Odwrócenie relacji** (zamiana końców).
- **Edycja pakietu:** nazwa, opis, wersja, status i stereotyp.
- **Dokument z szablonu dokumentów EA** (DOCX, PDF albo RTF), na przykład szablon „Model Report”, jak w EA: Publish → Report Builder.

Ułóż automatycznie diagram „Architektura” w kierunku w prawo i wygeneruj z szablonu „Model Report” dokument DOCX dla pakietu Sklep.

Pisownia w języku Compassa

Narzędzie asystenta `spell_check` sprawdza notatki elementów i diagramów, kroki scenariuszy oraz słownik projektu słownikiem pisowni Windows w języku Compassa (korekta pisowni samego EA ma tylko słowniki angielskie). Nazwy elementów i pojęcia ze słownika projektu są uznawane za znane słowa. Asystent podaje podpowiedzi, a poprawia dopiero po Twojej zgodzie. Jeśli Windows nie ma słownika dla tego języka, dodaj język w Ustawieniach Windows → Czas i język → Język. W Compass w EA notatki zmienionego elementu są sprawdzane na bieżąco (linie „🔍 Pisownia: ...” w System Output, rozdział 10).

Sprawdź pisownię w notatkach pakietu Sklep.

O czym warto wiedzieć

- **Pierwsze wywołanie EA trwa od kilku do kilkunastu sekund**, bo aplikacja uruchamia Enterprise Architect w tle i otwiera model. Kolejne zapisy i pobrania diagramów trwają już krócej niż sekundę. Po 10 minutach bez wywołań EA w tle zamyka się i zwalnia licencję. Zamknięcie aplikacji też zamyka jej sesję EA. Compass w EA (rozdział 10) nie uruchamia EA w tle: zapisuje przez Twoje okno EA.
- **Blokady.** Gdy w karcie Model jest włączona opcja „Model wymaga blokady do edycji”, asystent blokuje obiekt na swoje konto na czas zapisu i po zapisie blokadę zdejmuję. Obiektu zablokowanego przez innego użytkownika nie zmieni; komunikat podaje, kto blokuje element (konto lub grupa, od kiedy), jego pakiet i diagram.
- **Usuwanie jest trwałe.** Usunięcie pakietu usuwa całą jego zawartość. W trybach **ask** i **plan** usuwanie zawsze wymaga osobnej zgody, a panel zgody podaje nazwy usuwanych obiektów (rozdział 16). Przy ważnych modelach używaj trybu **ask**, proś o baseline i rób kopie zapasowe bazy.
- **Zapis „wszystko albo nic”.** Gdy jedna z operacji zapisu się nie uda, obiekty utworzone wcześniej w tym zapisie są usuwane, więc w modelu nie zostaje nic niedokończonego; komunikat podaje przyczynę.
- **Sprawdzaj wynik.** Asystent może się pomylić: źle odczytać zdjęcie, wybrać inny typ elementu albo ułożyć diagram nieczytelnie. Poproś o pokazanie diagramu po zmianach i o poprawki.
- **Duże diagramy** asystent zapisuje w kilku krokach. To normalne.

8. Przykłady użycia

Ten rozdział zbiera przykładowe polecenia według celu. Wypróbowano je na modelu przykładowym z pakietem „Shop domain”, procesem „Order process” i diagramem „Domain model”; w swoim modelu wstaw nazwy własnych pakietów, diagramów i elementów. Asystent nie wymaga dokładnych sformułowań: ważne są nazwy obiektów i to, czego oczekujesz. Szczegóły funkcji są w rozdziale 7, a zasady zatwierdzania zmian w rozdziale 16.

8.1 Pytanie o model

Odczyt modelu niczego nie zmienia i nie wymaga zgody. Compass czyta model przez bazę albo plik EA, a asystent odpowiada na tej podstawie i wskazuje obiekty nazwą, pakietem i GUID. Analiza wpływu idzie po relacjach w głąb modelu i podaje też diagramy, na których leżą dotknięte elementy.

Jakie klasy są w pakiecie Shop domain i jakie mają relacje?

Co zostanie dotknięte, jeśli usuniemy klasę Product? Podaj elementy i diagramy.

Co zobaczysz: kartę kroków z odczytami modelu, a potem odpowiedź na czacie. Przy pytaniu o wpływ każdy element ma relację, przez którą dochodzi wpływ, ścieżkę od elementu startowego i listę diagramów. Okno zgody się nie pojawia.

Wskazówka: dopisz „wynik daj w Excelu”, jeśli lista jest długa: dostaniesz arkusz zamiast tekstu na czacie.

8.2 Tworzenie i zmiana modelu

Asystent tworzy pakiety, diagramy, elementy i relacje oraz zmienia istniejące: nazwy, opisy, tagged values, położenie na diagramie. Zmiany zapisuje do EA dopiero po Twojej zgodzie (w trybie **ask** przed każdym zapisem, w trybie **plan** przed pierwszym zapisem odpowiedzi; rozdział 16).

Utwórz pakiet Benchmark, a w nim diagram klas Katalog z klasami Category i Item; Category agreguje Item.

W procesie Order process dodaj aktywność Check fraud między Payment a Ship order.

Dodaj wszystkim klasom w pakiecie Shop domain tagged value Owner o wartości Team A.

Co zobaczysz: nad polem wiadomości panel „**Asystent chce zapisać zmiany w modelu**” z opisem zmian i częścią „**Przed** → **po:**”: co powstaje, co zmienia się z jakiej wartości na jaką (pole: stara → nowa wartość) i co jest usuwane. Przy trzecim poleceniu zobaczysz tam zmianę dla każdej klasy. Przycisk **Zatwierdź** zapisuje zmiany, a **Odrzuć** niczego nie zmienia i pozwala opisać asystentowi, co ma zrobić inaczej.

- **Sprawdzenie po każdym zapisie.** Compass czyta model jeszcze raz i porównuje go z prośbą (czy obiekty istnieją, czy relacje mają wskazane końce, czy nowe wartości są zapisane). Asystent mówi tylko to, co model potwierdził, a różnice zgłasza zamiast pisać „gotowe”.
- **Wszystko albo nic.** Jeśli zapis nie uda się w połowie, obiekty, które już powstały, są usuwane.
- **Usuwanie elementu z relacjami** zawsze pyta o zgodę, także w trybie **auto**, a panel wymienia te relacje.

Wskazówka: przy zmianie masowej, jak w trzecim poleceniu, przeczytaj „Przed → po” przed zatwierdzeniem: tam widać każdą zmianę.

8.3 Cofanie zmian

Każdy zapis asystenta trafia do dziennika cofania („wehikuł czasu”) razem z operacjami odwrotnymi. Cofnięcie usuwa utworzone obiekty, a zmienionym polom, tagged values, położeniom i przeniesieniom przywraca poprzednie wartości. Dotyczy to zapisów asystenta w tym modelu; zmian wykonanych ręcznie w EA nie obejmuje.

Cofnij ostatnią zmianę.

Pokaż historię zmian asystenta w tym modelu.

Co zobaczysz: przy historii listę zapisów od najnowszego, z informacją, co da się cofnąć. Przy cofaniu panel zgody „Cofnąć ostatnie zapisy asystenta (...)” z listą zapisów (pojawia się w każdym trybie); po zatwierdzeniu Compass cofa zapisy i sprawdza wynik w modelu. Gdy nic nie ma do cofnięcia, asystent mówi, że w tym modelu nie ma nic do cofnięcia.

Usunięcia, importy i automatyczny układ diagramu nie dają się cofnąć operacjami. Przed takim zapisem Compass robi w EA baseline pakietu, którego dotyczy zmiana. Przywracasz go w Enterprise Architect: kliknij pakiet prawym przyciskiem myszy, wybierz **Manage Baselines**, zaznacz baseline i użyj **Compare/Restore**.

Wskazówka: możesz cofnąć kilka zapisów naraz („Cofnij trzy ostatnie zmiany”); najpierw poproś o historię, żeby zobaczyć, co wejdzie w cofnięcie.

8.4 Jakość modelu i reguły zespołu

Kontrolę jakości wykonuje program, nie model AI, więc nic nie jest zgadywane: domyślne i puste nazwy, elementy bez opisu, osierocone elementy, powtórzenia, relacje do nieistniejących elementów, relacje niedozwolone w ArchiMate 3.2, puste pakiety i diagramy oraz reguły z pliku `governance.json` (albo z polityki zespołu w hubie). Pisownię notatek sprawdza słownik Windows w języku Compassa.

Sprawdź jakość pakietu Shop domain i wypisz najważniejsze problemy.

Sprawdź pisownię notatek w całym modelu.

Przykład krótkiego pliku `governance.json` (leży obok `app.properties`), który wymaga tagged value Owner na każdej klasie:

```
{"required_tags":[{"applies_to":"Class","tag":"Owner"}]}
```

Co zobaczysz: podsumowanie sprawdzenia i ustalenia z wagą: **błąd**, **ostrzeżenie** albo **informacja**. Przy pisowni: liczbę sprawdzonych tekstów i słów oraz listę możliwych błędów z odpowiedziami. Okno zgody pojawia się dopiero, gdy poprosisz o poprawki. Jeśli Windows nie ma słownika dla języka Compassa, dodaj język w Ustawieniach Windows → Czas i język → Język.

Wskazówka: po sprawdzeniu poproś o poprawienie jednej grupy ustaleń naraz, na przykład „Uzupełnij brakujące opisy klas w pakiecie Shop domain”. Zapis przejdzie przez zwykłe zatwierdzanie i sprawdzenie po zapisie.

8.5 Dokument wymagań a model

Asystent czyta wymagania z dokumentu (Word `.docx`, `.txt`, `.md`, załącznik rozmowy, strona Confluence po ID albo wklejony tekst) i zestawia je z modelem. Wymaganie to wiersz albo wiersz tabeli zaczynający się od identyfikatora, na przykład W1, R.1.2, REQ-12, FR-01. Dopasowanie robi program: szuka elementów z tym identyfikatorem (w nazwie, aliasie, notatkach, tagged values), relacji śledzenia i podobnych słów. Kandydatów ocenia asystent.

Porównaj wymagania z dokumentu C:\Dokumenty\BRD.docx z modelem: co jest pokryte, a czego brakuje?

Porównaj wymagania ze strony Confluence 123456 z modelem i zapisz macierz śledzenia do Excela.

Utwórz relacje śledzenia dla pokrytych wymagań.

Co zobaczysz: przy pliku, który nie jest załącznikiem rozmowy, najpierw pytanie, czy asystent może go przeczytać; potem podsumowanie „Śledzenie wymagań: ... wymagań – pokryte ..., tylko kandydaci ..., NIEPOKRYTE ..., zgodne ID przy innej treści ...” i listę wymagań z jednym z czterech statusów: **pokryte**, **kandydaci**, **NIEPOKRYTE** albo **ZGODNE ID, INNA TREŚĆ**. Osobna lista to „Elementy modelu, do których nie odnosi się żadne wymaganie”. Asystent wskazuje też wymagania sprzeczne z modelem. Macierz w Excelu dostajesz jako kartę pliku z przyciskami **Otwórz** i **Zapisz jako...**. Relacje śledzenia (Realisation albo Dependency «trace») powstają dopiero po Twojej zgodzie, w zwykłym panelu zatwierdzania.

Wskazówka: stronę Confluence można czytać po uzupełnieniu karty **Confluence** w Ustawieniach (rozdział 5). Dopasowanie jest najpewniejsze, gdy identyfikator wymagania wpisano w nazwę, alias, notatkę albo tagged value elementu; bez niego zostają tylko kandydaci. Zakres możesz zawęzić do pakietu: „...z pakietem Shop domain”. Dodaj folder z kodem, żeby prześledzić wymagania dalej, do kodu, testów i zgłoszeń (rozdział 7, „łańcuch wymagań”).

8.6 Model a kod

Asystent czyta folder z kodem (na przykład lokalny klon repozytorium) tylko do odczytu i porównuje go z modelem: klasy, interfejsy, enumy i rekordy z polami (Java, Kotlin, C#, TypeScript, Rust, Python), tabele i kolumny z SQL (`CREATE TABLE`) oraz operacje API (`operationId` z OpenAPI) z klasami, atrybutami, tabelami i operacjami w modelu.

Porównaj model z kodem w folderze C:\repo\sklep – czego brakuje w modelu, a czego w kodzie?

Co zobaczysz: najpierw pytanie, czy asystent może przeczytać folder, potem podsumowanie „Model ↔ kod: plików źródłowych ...” i sekcje: **W kodzie, brak w modelu**, **W modelu, brak w kodzie**, **Różne pola (kod vs atrybuty modelu)**, **Tabele bazy danych** oraz **Operacje API (OpenAPI operationId) bez odpowiednika w modelu**. Zgoda na zapis nie jest potrzebna, bo nic nie jest zapisywane: ani w kodzie, ani w modelu.

Wskazówka: ogranicz porównanie do jednego pakietu („...z pakietem Shop domain”), jeśli folder zawiera wiele usług, i poproś o raport w Excelu. Uzupełnienie modelu o brakujące klasy to osobna prośba, która przejdzie przez zwykłe zatwierdzanie.

8.7 Wycena zmiany

Z wpływu zmiany w modelu asystent buduje wstępną wycenę: elementy do zmiany, powiązane elementy do dostosowania, dalsze do sprawdzenia i diagramy do aktualizacji dostają wiersze WBS w prostym języku

funkcjonalnym, z godzinami zależnymi od typu elementu. Do tego dochodzą testy, analiza i dokumentacja oraz bufor ryzyka.

Wycień zmianę klasy Product: co trzeba zmienić i ile to zajmie?

Zapisz tę wycenę do Excela.

Co zobaczysz: tabelę WBS z wierszami w rodzaju „Zmiana: Product”, „Dostosowanie: ...”, „Sprawdzenie wpływu na: ...”, „Aktualizacja diagramu: ...”, „Testy zmiany”, „Analiza i dokumentacja” i „Bufor ryzyka”, podsumowanie w godzinach i osobodniach, „Godziny według typu” oraz „Ryzyka” (na przykład element z wieloma relacjami, bez opisu albo zmiana dotycząca wielu pakietów). Excel to karta pliku z przyciskami

Otwórz i Zapisz jako....

Wskazówka: to punkt wyjścia z modelu, a nie oferta: skoryguj godziny wiedzą zespołu. Normy godzinowe są w pliku `estimate.json` obok pliku ustawień; gdy go nie ma, działają normy domyślne.

8.8 Dokumenty i eksporty

Asystent przygotowuje dokumenty Word z treści modelu i diagramów (rozdział 13). Enterprise Architect generuje też raport z własnego szablonu dokumentów EA (DOCX, PDF albo RTF) i eksportuje pakiety do XML. Raport i eksport XML zawsze wymagają Twojej zgody, także w trybie **auto**.

Przygotuj dokument Word z opisem pakietu Shop domain z diagramem Domain model.

Wygeneruj raport z szablonu EA Model Report dla pakietu Shop domain do pliku C:\Raporty\sklep.docx.

Wyeksportuj pakiet Shop domain do pliku XML C:\Eksport\shop.xml.

Co zobaczysz: dla dokumentu Word kartę pliku z przyciskami **Otwórz i Zapisz jako....** Dla raportu i XML panel zgody z opisem („dokumenty z szablonów EA”, „pakiety wyeksportowane do XML”) i pełną ścieżką pliku, a po zapisie komunikat „zapisano raport: ...” albo „zapisano XML: ...” ze ścieżką; Compass sprawdza, czy plik powstał.

Wskazówka: raport musi mieć rozszerzenie `.docx`, `.pdf` albo `.rtf`, a plik XML `.xml` albo `.xmi`. Podaj pełną ścieżkę na dysku lokalnym: udziały sieciowe oraz foldery systemowe, programów i danych aplikacji są odrzucane. Szablon o podanej nazwie musi istnieć w EA; gdy go nie ma, EA nie utworzy raportu i asystent to zgłosi.

8.9 Diagramy

Poza rysowaniem nowych diagramów asystent poprawia istniejące: układa je automatycznie (w dół, w prawo, w górę albo w lewo, z podanym odstępem), dodaje tory (swimlanes) i tworzy dokumenty powiązane (linked documents) elementów z podanego tekstu.

Ułóż automatycznie diagram Domain model od lewej do prawej.

Dodaj na diagramie Order process tory Sprzedaż i Magazyn.

Dodaj do klasy Order dokument powiązany z treścią: Zamówienie klienta – nagłówek i pozycje.

Co zobaczysz: zwykły panel zgody z opisem zmiany, a po zapisie sprawdzenie modelu. Jeśli masz otwarty ten sam model w EA, zmieniony diagram odświeża się w Twoim oknie sam (rozdział 9).

Wskazówka: po zmianie poproś „Pokaż diagram Domain model”: zobaczysz go jako obrazek na czacie i ocenisz, czy układ jest czytelny. Automatycznego układu nie da się cofnąć operacjami, więc Compass robi wcześniej baseline pakietu (podrozdział 8.3).

8.10 Skrypty EA

Asystent pisze skrypty EA (JScript, VBScript albo JavaScript) i uruchamia je w modelu: wypisują dane, sprawdzają coś albo zmieniają wiele obiektów naraz. Brakująca grupa skryptów powstaje sama. Skrypt działa w otwartym oknie EA, a gdy go nie ma, w osobnej sesji EA w tle.

Utwórz i uruchom skrypt EA w JScript, który wypisze nazwy wszystkich pakietów modelu.

Co zobaczysz: najpierw panel zgody na zapis skryptu (grupa, nazwa, język i cały kod), potem osobne pytanie o zgodę na uruchomienie: oba pojawiają się za każdym razem, także w trybie **auto**, bo skrypt może zmienić w modelu wszystko. Skrypt działa dokładnie tak, jak go pokazano, razem z dołączeniami `!INC`. Wynik (to, co skrypt wypisał przez `Session.Output`) trafia na czat jako „Wynik skryptu ...”. Jeśli w EA jest otwarty inny model, uruchomienie kończy się komunikatem.

Wskazówka: zanim zatwierdzisz uruchomienie, przeczytaj kod skryptu w oknie zgody, zwłaszcza gdy skrypt ma coś zmieniać. Na początek pisz skrypty tylko do odczytu.

8.11 Jira i Confluence

Po uzupełnieniu kart **Jira** i **Confluence** w Ustawieniach (rozdział 5) asystent czyta zgłoszenia i strony jako kontekst: sprawdza, których elementów modelu dotyczy zgłoszenie, rysuje diagramy z jego opisu albo zestawia wymagania ze strony z modelem (podrozdział 8.5).

Przeczytaj zgłoszenie ARCH-123 i powiedz, których elementów modelu dotyczy.

Utwórz w Jirze zadanie: zmienić nazwę elementu Class1 w pakiecie Shop domain.

Co zobaczysz: przy czytaniu odczyt zgłoszenia w karcie kroków i odpowiedź z listą elementów. Tworzenie zgłoszeń, komentarzy i linków do elementów modelu **zawsze** wymaga Twojej zgody w panelu zatwierdzania, w każdym trybie. Publikacja strony w Confluence też zawsze pyta i mówi wprost, gdy zastąpi istniejącą stronę (rozdział 13).

Wskazówka: podawaj klucz zgłoszenia i nazwę pakietu. Gdy nie wskażesz projektu, nowe zgłoszenie powstaje w projekcie domyślnym z karty Jira.

8.12 Praca zespołowa (Compass Hub)

Z hubem zespołu (rozdział 22; karta **Zespół (Compass Hub)** w Ustawieniach) zmianę w modelu możesz najpierw **zapropnować** zespołowi: asystent niczego nie zapisuje w modelu, tylko wysyła do huba zestaw zmian z opisem i podglądem „przed → po”. Zespół komentuje go w widoku webowym, a architekt albo administrator zatwierdza lub odrzuca. Nikt, nawet administrator, nie może zatwierdzić własnego zestawu zmian. Zatwierdzony zestaw wprowadzasz w Compassie.

Zaproponuj zespołowi zmianę do przeglądu: dodaj klasę Warehouse do pakietu Shop domain.

Pokaż zatwierdzone zestawy zmian i wprowadź CH-...

Co zobaczysz: po propozycji komunikat „Zestaw zmian dla zespołu: ...” z numerem (CH-...), a w widoku webowym huba nowy wiersz w zakładce **Change sets**. Przy wprowadzaniu: zwykły panel zatwierdzania i sprawdzenie po zapisie; hub oznacza zestaw jako wprowadzony. Wprowadzić można tylko zestaw zatwierdzony i dotyczący tego modelu, i tylko raz: Compass najpierw rezerwuje go w hubie, a po zapisie zgłasza wynik.

Widok webowy. Osoby bez Enterprise Architect logują się w nim nazwą użytkownika i hasłem od administratora huba (adres `https://<hub>/`; `http://` tylko na tym samym komputerze). Zakładki mają angielskie nazwy: **Ask** (pytania do modelu), **Search** (wyszukiwanie elementów), **Model check** (kontrola modelu według reguł zespołu), **Gaps** (luki as-is / to-be), **Audit trail** (zespołowy dziennik zmian), **Team notes** (wspólne notatki) i **Change sets** (propozycje zmian z komentarzami i decyzją). W zakładce **Ask** wpisujesz pytanie własnymi słowami, na przykład „Które systemy korzystają z bramki płatności?”. Asystent w tym miejscu tylko czyta model: nie ma dostępu do plików, Jiry, Confluence ani serwerów MCP i nie może niczego zaproponować ani zapisać. Odpowiada na jedno pytanie naraz (pytanie ma najwyżej 2000 znaków), a pytania trafiają do dziennika huba.

Wskazówka: nadaj propozycji tytuł, z którego recenzent pozna zakres zmiany bez otwierania szczegółów. Zestaw zmian jest przypisany do modelu: wprowadzisz go tylko w Compassie połączonym z tym samym modelem huba.

8.13 Dodatkowe narzędzia z serwerów MCP

Własne serwery MCP dają asystentowi dodatkowe narzędzia: system zgłoszeń, repozytorium, bazę wiedzy albo serwer MCP samego huba. Serwery wpisujesz w pliku `mcp_servers.json` (rozdział 7) albo dodajesz gotowe (Azure DevOps, GitHub, ServiceNow / CMDB, Files, Compass Hub) i sprawdzasz w Ustawieniach, w karcie **Serwery MCP**, przyciskiem **Zapisz i sprawdź serwery MCP**.

Przez serwer MCP hub wypisz źródła modeli zespołu.

Co zobaczysz: w karcie kroków wywołanie narzędzia serwera, a potem jego wynik w odpowiedzi. Każde wywołanie narzędzia pyta o Twoją zgodę, chyba że serwer ma `"trusted": true`; ustawiaj to tylko dla serwerów, którym ufasz.

Wskazówka: w poleceniu podaj nazwę serwera z pliku `mcp_servers.json` („przez serwer MCP hub...”), żeby asystent wiedział, którego użyć. Tokeny trzymaj w Menedżerze poświadczeń Windows pod ich własnymi nazwami (`${secret:nazwa}`), nie w pliku; własnych kluczy i haseł Compassa nie można tam użyć (rozdział 7).

8.14 Compass w Enterprise Architect

Gdy Compass jest zadokowany w EA (rozdział 10), pilnuje modelu w trakcie Twojej pracy i nie wymaga żadnego polecenia. Wyniki trafiają do zakładki **Loonar Compass** okna **System Output**:

- **Kontrola na bieżąco:** po narysowaniu relacji i po edycji elementu (nazwy, stereotypu, tagged values) pojawia się „△ Kontrola modelu: ...”, według tych samych reguł co w podrozdziale 8.4.
- **Pisownia:** notatki zmienionego elementu są sprawdzane w języku Compassa i dają „✎ Pisownia: ...”.
- **Agenci zdarzeń:** własne reguły w pliku `event_agents.json` (karta **Agenci zdarzeń (Compass w EA)**) uruchamiają asystenta po zdarzeniu w EA. Jego uwagi zaczynają się od .

Przykładowa reguła, która po utworzeniu komponentu sprawdza właściciela i opis:

```
{
  "agents": [
    {
      "name": "Component owner",
      "event": "element_created",
      "applies_to": "Component",
      "prompt": "Check that the new component has an owner and a description; suggest what to fill in."
    }
  ]
}
```

Co zobaczysz: w System Output linię w rodzaju „ Component owner: ...” z uwagami asystenta. Agent **tylko czyta model**: nie ma dostępu do plików, Jiry, Confluence ani serwerów MCP i nie może niczego zaproponować ani zapisać. Ten sam agent reaguje na ten sam obiekt najwyżej raz na minutę. Nowy element jest sprawdzany dopiero po pierwszej edycji, bo EA nadaje mu najpierw nazwę Class1.

Wskazówka: pisz krótkie, konkretne polecenia agentów i zawężaj je przez `applies_to` (typ, «stereotyp» albo `*`). Po zmianie pliku kliknij **Zapisz agentów zdarzeń**.

8.15 Sprawdzanie modelu w potoku CI

Tryb sprawdzania wykonuje kontrolę modelu, porównanie z kodem i z CMDB oraz łańcuch wymagań bez okna i bez modelu AI, a kończy się kodem wyjścia, który rozumie potok (rozdział 23).

```
LoonarCompass.exe --check --model model\shop.qea --rules governance.json --code src --requirements docs\wymagania.md --junit junit.xml --fail-on error --lang pl | Out-Host
```

Co zobaczysz: raport Markdown na standardowym wyjściu, który zaczyna się od wyniku, PASSED albo FAILED, a dalej ma sekcje kontroli modelu, modelu a kodu i wymagań. W potoku plik JUnit pokazuje każdą grupę ustaleń jako test, nieudany, gdy jej waga jest na progu `--fail-on` albo wyżej. Kod wyjścia to 0, gdy sprawdzenie przeszło, 1, gdy są takie ustalenia, i 2 przy błędnych argumentach albo modelu, którego nie da się odczytać.

Wskazówka: zacznij od `--fail-on error` i trzymaj ostrzeżenia w raporcie; przejdź na `warning`, gdy model będzie czysty. Trzymaj `governance.json` w repozytorium i podawaj go opcją `--rules`, żeby potok i ludzie stosowali te same reguły.

8.16 CMDB a model

Asystent porównuje pozycje konfiguracji z CMDB (eksport albo dane z serwera MCP) z komponentami modelu i pokazuje, czego brakuje po każdej stronie oraz gdzie różni się właściciel albo status (rozdział 7, „CMDB a model”).

Porównaj eksport CMDB C:\Eksport\cmdb_ci_appl.csv z komponentami modelu.

Pobierz z ServiceNow aplikacje (CI) i porównaj je z modelem.

Co zobaczysz: przy pliku, który nie jest załącznikiem rozmowy, najpierw pytanie, czy asystent może go przeczytać; przy serwerze MCP ServiceNow wywołanie jego narzędzia (z pytaniem o zgodę, chyba że serwer jest zaufany). Potem podsumowanie „CMDB ↔ model: pozycje CMDB: ..., elementy modelu: ..., dopasowane: ...” i sekcje **W CMDB, brak w modelu**, **W modelu, brak w CMDB** oraz **Inny właściciel lub status**. Bez Twojej zgody nic nie jest zapisywane.

Wskazówka: trzymaj identyfikator CMDB (`sys_id`) w tagged value elementów: wtedy pozycje dopasują się także po zmianie nazwy. Zawęż porównanie do pakietu albo wskaż typy elementów, gdy model zawiera więcej niż CMDB.

8.17 Zgłoszenia Jira i strony Confluence z huba

Gdy administrator ustawi `integrations` w `hub.json` (rozdział 22), architekci i administratorzy tworzą zgłoszenia Jira i strony Confluence wprost z widoku webowego huba, bez Compassa i bez EA. Hub używa

własnego konta technicznego.

- **Search** → kliknij element → **Create Jira issue**: wpisz podsumowanie; zgłoszenie dostaje szczegóły elementu.
- **Change sets** → otwórz zestaw zmian → **Create Jira issue for this change set**: jedno zgłoszenie na zestaw zmian.
- Wybierz pakiet w drzewie → **Model check** → **Publish this package to Confluence**: wpisz tytuł (pusty: nazwa pakietu i modelu).

Co zobaczysz: „Created ARCH-...” z linkiem do zgłoszenia; w zestawie zmian wiersz „Jira: ARCH-...”; przy Confluence „Created:” albo „Updated:” z linkiem do strony. Każda akcja trafia do zakładki **Audit trail** z Twoją nazwą.

Wskazówka: strona Confluence to tekstowe podsumowanie pakietu (elementy z opisami i relacje, bez diagramów), zastępowane przy każdej publikacji: zmieniasz model, nie stronę. Stronę z diagramami publikuj z Compassa (rozdział 13).

Wskazówki do dobrych poleceń

- **Nazywaj pakiety i diagramy dokładnie**, tak jak w modelu (najlepiej w cudzysłowie). Przy dwóch obiektach o tej samej nazwie dodaj pakiet.
- **Powiedz, czy zmieniać, czy tylko zaproponować.** „Zaproponuj, ale jeszcze nie zapisuj” daje opis zmian bez zapisu.
- **Przy większej zmianie poproś najpierw o podgląd** i pracuj w trybie **plan**: panel zgody pokaże całość „Przed → po” przed pierwszym zapisem.
- **Poproś o zapis wyniku do Excela albo Worda**, gdy ma być poza czatem: listy, macierze i wyceny.
- **Przy ryzykownych zmianach użyj trybu ask** (Ustawienia → „Język i zatwierdzanie zmian”), poproś o baseline pakietu i sprawdź diagram po zmianach.

9. Współpraca z otwartym oknem EA

Jeśli masz otwarty w Enterprise Architect **ten sam model**, asystent współpracuje z Twoim oknem EA:

- **wie, co masz otwarte i zaznaczone:** jaki diagram jest otwarty i które elementy są zaznaczone na diagramie albo w przeglądarce projektu. Możesz więc pisać „ten element” albo „zaznaczone zadania”,
- **otwiera diagramy** w Twoim oknie EA i **wskazuje elementy** w przeglądarce projektu,
- **po zapisie sam odświeża** zmienione diagramy i pakiety w Twoim oknie, bez naciskania **F5**.

Opisz element, który mam zaznaczony, i pokaż, na jakich diagramach jeszcze występuje.

Otwórz w moim EA diagram „Proces zamówienia” i wskaż w przeglądarce pakiet „Wymagania”.

Klawisz **F5** jest potrzebny tylko wtedy, gdy okno EA nie jest otwarte albo pokazuje inny model: asystent działa wtedy jak zwykle, a zmiany zobaczysz po odświeżeniu pakietu w EA. Jeśli w oknie EA jest otwarte okno dialogowe, EA nie odpowiada asystentowi; zamknij je i ponów prośbę.

Opisana tu współpraca dotyczy osobnej aplikacji. Compass uruchomiony w samym Enterprise Architect zawsze pracuje na projekcie otwartym w swoim oknie EA (rozdział 10).

Odczyty pliku EA, który jest otwarty w oknie EA, idą przez to okno, bo plik na dysku może jeszcze nie mieć Twoich ostatnich zmian.

10. Compass w Enterprise Architect

Loonar Compass działa też wewnątrz Enterprise Architect, jako okno zadokowane obok przeglądarki projektu i diagramów. Pracujesz w nim tak samo jak w osobnej aplikacji: rozmowa, załączniki, zatwierdzanie zmian, pamięć, szablony i dokumenty działają bez zmian. Osobna aplikacja, uruchamiana z menu Start, też działa jak dotąd.

Instalacja i uruchomienie

1. W instalatorze, na stronie „Folder docelowy”, zaznacz **„Dodaj Loonar Compass do Enterprise Architect”**. Działa to z każdą instalacją Enterprise Architect, także z wersją próbną. Po instalacji uruchom EA ponownie.
2. Otwórz projekt w Enterprise Architect: plik albo bazę danych.
3. Na wstążce **Start**, w grupie **Add-Ins**, kliknij **Loonar Compass** (przycisk znajdziesz też w **Specialize**). Przycisk działa tylko przy otwartym projekcie.
4. Compass otwiera się jako okno zadokowane w EA, w tym samym miejscu co okno Properties. Możesz je przeciągnąć w inne miejsce, odpiąć albo dołączyć do grupy zakładki; EA zapamiętuje, gdzie je umieściłeś.

Projekt i rozmowy

- W EA Compass zawsze pracuje na projekcie otwartym w tym oknie EA: pliku albo dowolnej bazie danych. Czyta model przez sam Enterprise Architect, więc nie potrzebuje ustawień modelu, haseł ani wyboru modelu rozmowy. Otwarcie w EA innego projektu przełącza Compass na ten projekt; bez otwartego projektu Compass nie działa.
- Rozmowy wybierasz z listy rozwijanej u góry okna. Lista pokazuje tylko rozmowy z tym projektem. Przycisk + zaczyna nową rozmowę, a ołówek zmienia nazwę wybranej.
- Ikony widoków są na dole okna, a w wąskim oknie strony przewijają się w poziomie i w pionie.
- **Ctrl** + **V** wkleja obrazy, pliki i tekst, tak jak w osobnej aplikacji.
- Ustawienia w EA nie mają kart Model i Enterprise Architect. Karty Model AI, Język i zatwierdzanie zmian, Confluence, Miro, Jira i Zespół (Compass Hub) działają tak samo jak w osobnej aplikacji (rozdział 5).

Zmiany, diagramy i odwołania do obiektów

- Zapisy idą przez to samo okno EA, więc zmiany widać od razu w przeglądarce projektu i na diagramach. Zatwierdzanie zmian działa jak w osobnej aplikacji (rozdział 16).
- **Pokaż diagram** i **Edytuj diagram** otwierają diagram w samym EA, zamiast pokazywać obrazek na czacie.
- Kliknij element, pakiet albo diagram prawym przyciskiem myszy (w przeglądarce projektu albo na diagramie) i wybierz **Specialize** → **Dodaj do rozmowy w Compass**. Do wiadomości trafia odwołanie do obiektu, na przykład «Pakiet: Sklep (id 5)», a asystent wie, o który obiekt chodzi.

Opisz «Pakiet: Sklep (id 5)» i zaproponuj brakujące wymagania.

Kontrola na bieżąco

Gdy Compass jest zadokowany w EA, sprawdza zmiany w trakcie Twojej pracy według tych samych reguł co kontrola modelu (wbudowanych i z `governance.json` albo z huba). Kontrola rusza po narysowaniu relacji i po edycji elementu (nazwy, stereotypu, tagged values). Ustalenia pojawiają się w zakładce **Loonar Compass** okna System Output, każde tylko raz, jako „ Kontrola modelu: ...”. Nowy element jest sprawdzany dopiero po pierwszej edycji, bo EA nadaje mu najpierw nazwę Class1.

Agenci zdarzeń

Karta **Agenci zdarzeń (Compass w EA)** zawiera reguły, według których asystent reaguje na to, co dzieje się w EA. Reguły są w pliku `event_agents.json`:

```
{ "agents": [ { "name": "Component owner", "event": "element_created", "applies_to": "Component",  
  "prompt": "Check that the new component has an owner and a description; suggest what to fill in." } ] }
```

- Zdarzenia (event): `element_created`, `element_changed`, `connector_created`, `diagram_created`. W `applies_to` podajesz typ elementu, «stereotyp» albo `*` (działają symbole `*` i `?`).
- Agenci działają, gdy Compass jest otwarty w swojej zakładce w EA. Asystent czyta model i odpowiada w zakładce **Loonar Compass** okna System Output (linie zaczynają się od ). **Tylko czyta model:** nie ma dostępu do plików, Jiry, Confluence ani serwerów MCP i nie może niczego zaproponować ani zapisać.
- Ten sam agent reaguje na ten sam obiekt najwyżej raz na minutę.
- Przycisk **Zapisz agentów zdarzeń** zapisuje reguły i podaje liczbę agentów.

Pisownia na bieżąco. Notatki zmienionego elementu są sprawdzane w języku Compassa; uwagi pojawiają się w System Output jako „ Pisownia: ...”.

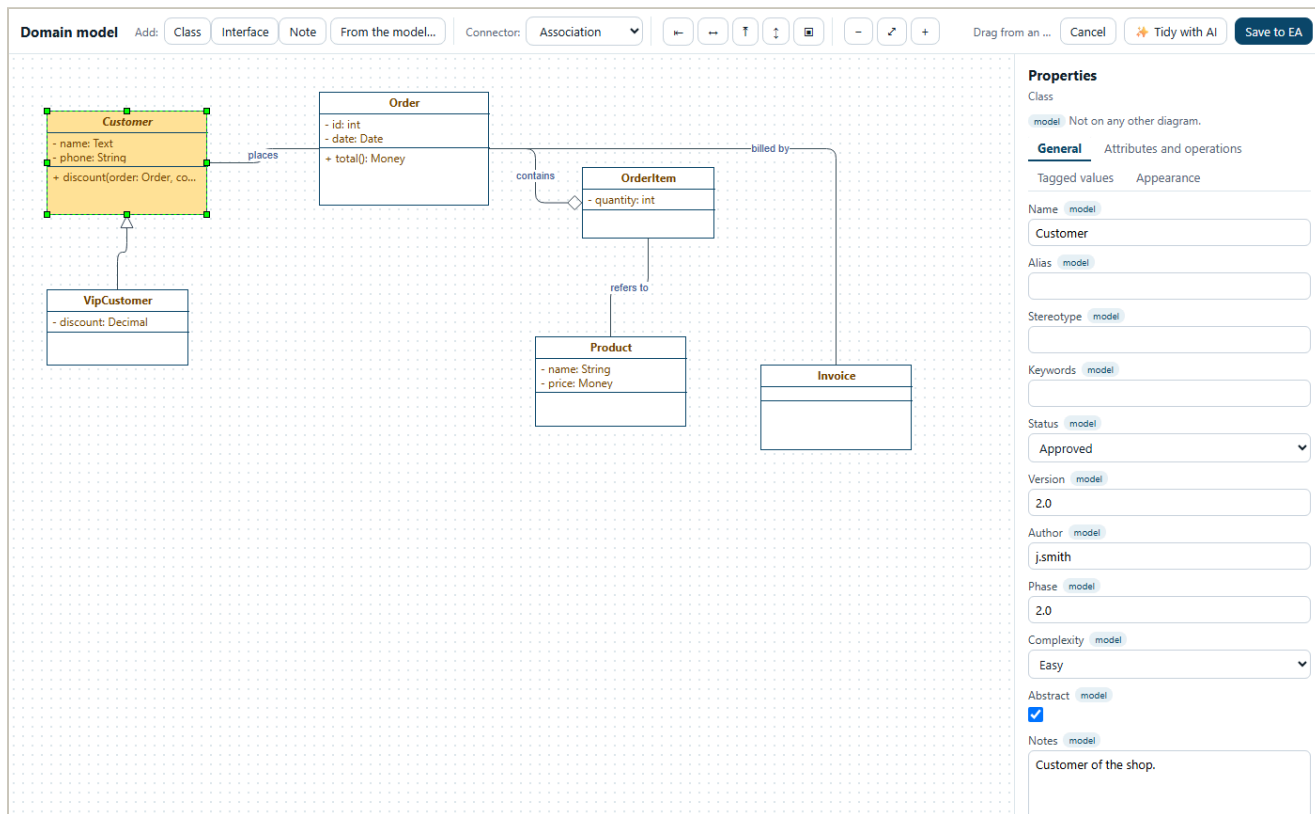
Kroki asystenta w System Output

Okno **System Output** w EA ma zakładkę **Loonar Compass**: Twoje wiadomości i każdy krok asystenta (odczyty modelu, zapisy do EA, wyniki) z godziną. Czat w oknie Compass pokazuje w tym czasie tylko ostatni krok, w jednej linii.

Zamykanie. Compass zamyka się razem z Enterprise Architect i zapisuje rozmowę. Po zamknięciu projektu w EA Compass prosi o otwarcie projektu.

11. Edytor diagramów

Każdy diagram z EA pokazany w rozmowie ma przycisk **Edytuj diagram**. Otwiera on diagram w oknie edytora, w którym zmieniasz go myszą. Edytor działa na modelu, nie na obrazku: to, co zapiszesz, trafia do EA przez te same operacje, których używa asystent, z panelem zatwierdzania i dziennikiem zmian (rozdział 16). Edytor potrzebuje WebView2, który jest częścią Windows 10 i 11. W Compass w EA przycisk **Edytuj diagram** otwiera diagram w samym Enterprise Architect (rozdział 10).



Edytor diagramów: pasek narzędzi, diagram i panel Właściwości

Rysowanie

- **Przesuwanie, zmiana rozmiaru i wyrównywanie** elementów myszą i przyciskami wyrównania; **Ctrl** + **Z** i **Ctrl** + **Y** cofają i ponawiają.
- **Dodaj**: przyciski zależą od rodzaju diagramu (klas, komponentów, przypadków użycia, aktywności, BPMN, maszyny stanów, wdrożenia, pakietów, sekwencji). Nowy element trafia w wolne miejsce blisko środka widoku; od razu wpisz jego nazwę.
- **Łączenie**: przeciągnij od jednego elementu do drugiego. Rodzaj relacji wybierasz na liście **Połączenie**.
- **Z modelu...** wstawia na diagram element, który już jest w modelu (wyszukiwanie po nazwie, typie lub pakiecie). Jego relacje z elementami diagramu pojawią się po zapisie.
- Dwuklik na elemencie lub relacji zmienia nazwę.

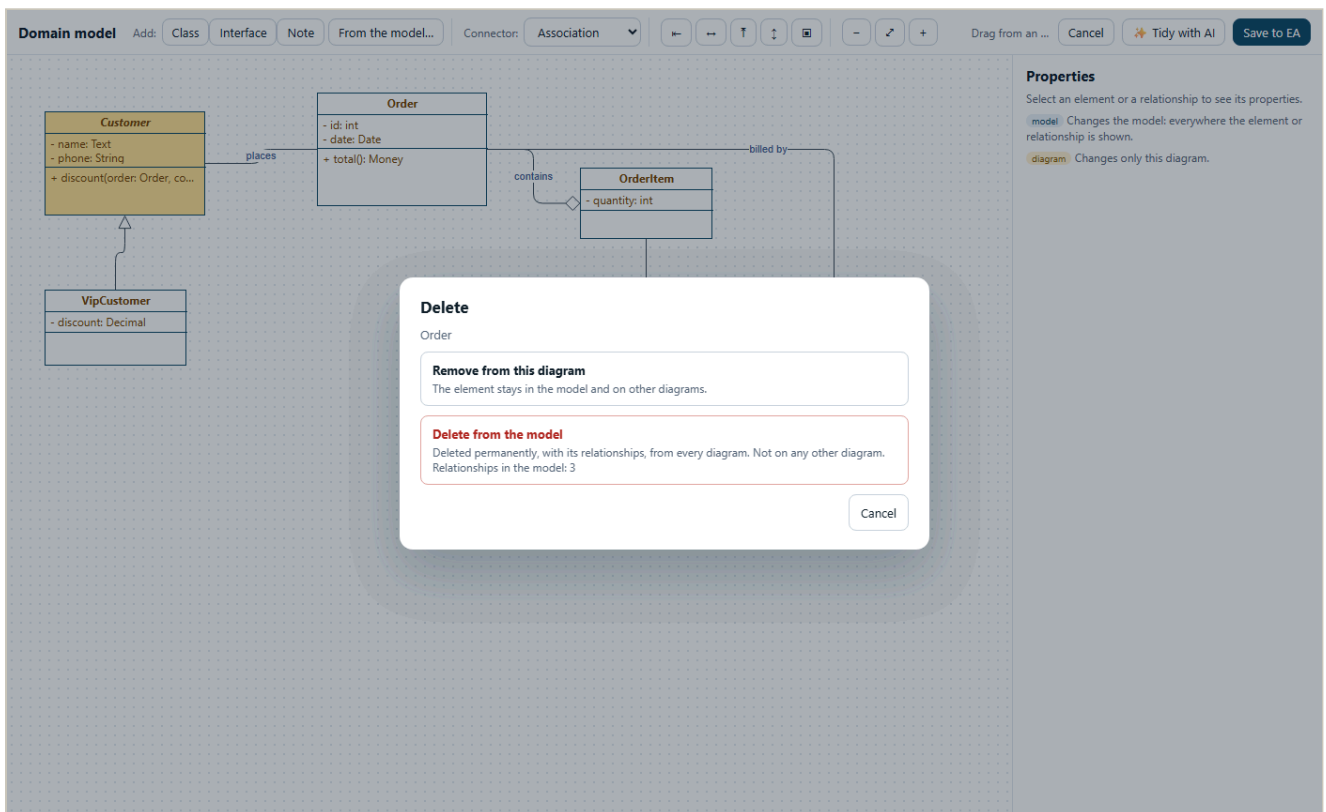
Właściwości

Panel **Właściwości** pokazuje zaznaczony element lub relację. Każde pole ma znaczek **model** (zmiana obowiązuje wszędzie, gdzie obiekt jest pokazany) albo **diagram** (tylko ten diagram).

- **Ogólne:** nazwa, alias, stereotyp, słowa kluczowe, status, wersja, autor, faza, złożoność, abstrakcyjny i notatki; pakiet, GUID i daty tylko do odczytu. Dla relacji: nazwa, stereotyp, kierunek, krotności i role obu końców, notatki.
- **Atrybuty i operacje** (klasy, interfejsy, komponenty): dodawanie, zmiana (widoczność, nazwa, typ) i usuwanie; nowa operacja przyjmuje parametry w postaci `nazwa: Typ, inna: Typ`.
- **Tagi:** zmiana, dodawanie i usuwanie; gdy profil podaje dozwolone wartości (na przykład typy zadań BPMN), wybierasz z listy.
- **Wygląd:** kolor wypełnienia na tym diagramie albo domyślny kolor elementu oraz styl linii relacji na tym diagramie.

Usuwanie

Klawisz `Delete` usuwa bez pytania to, co dodałeś w edytorze. Przy obiekcie, który już jest w modelu, edytor pyta: **Usuń z tego diagramu** (zostaje w modelu i na innych diagramach) albo **Usuń z modelu** (trwale, razem z relacjami, ze wszystkich diagramów). Pokazuje, na jakich innych diagramach jest element i ile ma relacji.



Usuwanie: z tego diagramu albo z modelu, z listą innych diagramów elementu

Diagramy sekwencji

Linie życia mają nagłówek i przerywaną linię; przesuwają się tylko w poziomie. Komunikaty idą od góry w swojej kolejności, z numerami, tak jak rysuje je EA. Kolejność zaznaczonego komunikatu zmieniasz przyciskami **▲ ▼** w panelu albo `Alt + ↑` / `Alt + ↓`. Nowy komunikat (przeciągnij od jednej linii życia do drugiej) trafia zaraz po zaznaczonym. W panelu ustawiasz rodzaj (synchroniczny albo asynchroniczny), czy to odpowiedź, i możesz nazwać go operacją odbiorcy.

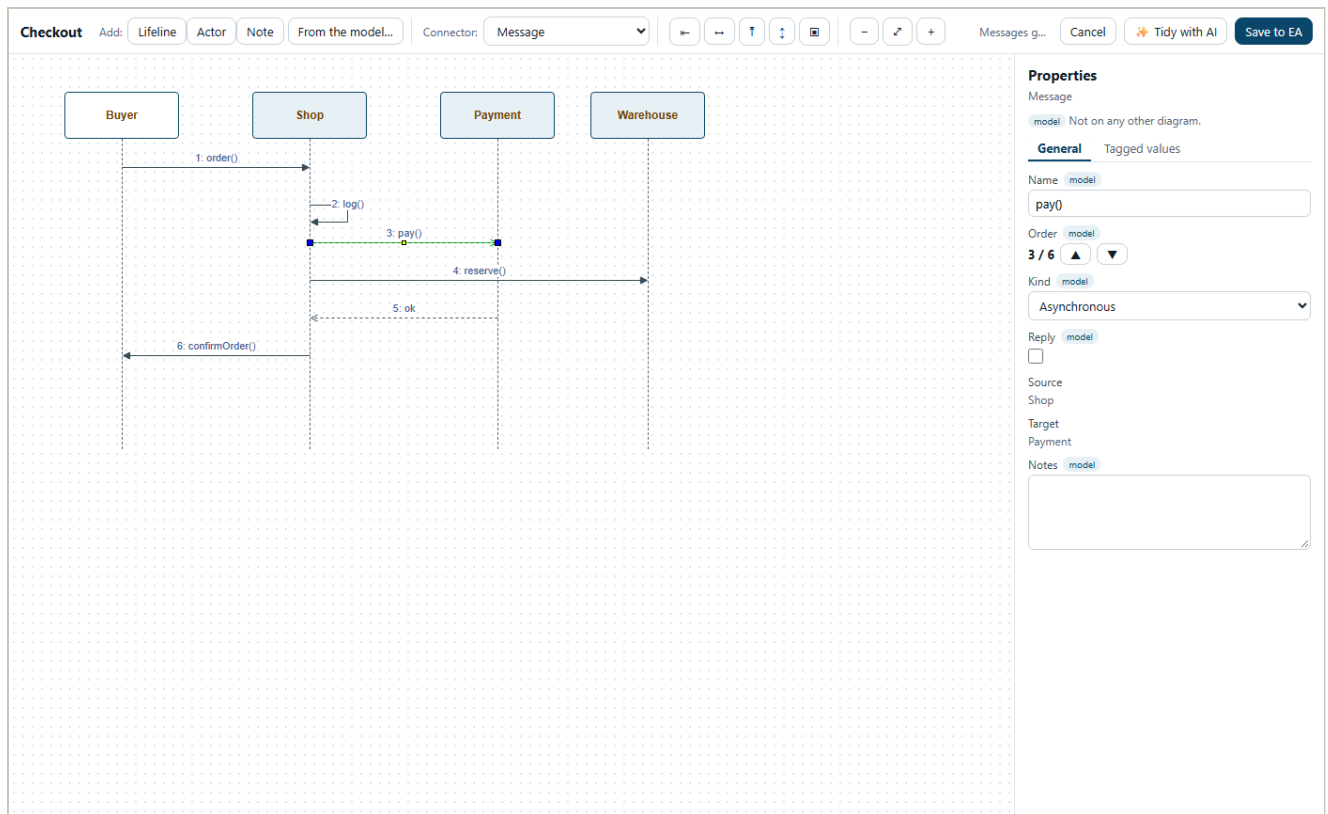


Diagram sekwencji: ponumerowane komunikaty, kolejność zmieniana przyciskami ▲ ▼

Zapis

- **Zapisz do EA** zapisuje zmiany i zamyka edytor; wynik pojawia się w rozmowie. **Anuluj** zamyka go bez zapisu.
- **Uporządkuj z AI** zapisuje zmiany i prosi asystenta w rozmowie o takie rozmieszczenie diagramu, żeby dobrze się czytał, bez dodawania, usuwania i zmiany nazw.

12. Wysyłanie diagramów do Miro

Diagram z EA może trafić na tablicę Miro jako edytowalne kształty i połączenia w ramce z nazwą diagramu. Tylko z EA do Miro: to, co ktoś zmieni na tablicy, nie wraca do modelu.

Konfiguracja, jednorazowo:

1. W Miro otwórz ustawienia profilu → **Your apps** → **Create new app**. Miro najpierw prosi o założenie **Developer team** (darmowej piaskownicy na Twoje aplikacje): zaakceptuj warunki i utwórz go. W organizacji na planie Enterprise zakłada go administrator organizacji.
2. Nadaj aplikacji nazwę (na przykład Loonar Compass) i **nie zaznaczaj „Expire user authorization token”**: wygasający token przestaje działać po godzinie.
3. W sekcji **Permissions** zaznacz `boards:read` i `boards:write`.
4. Kliknij **Install app and get OAuth token** i wybierz organizację oraz zespół, których tablic aplikacja może używać (na pierwszy test: Developer team). Skopiuj token.
5. Wklej go w Ustawieniach, karta „Miro”, i kliknij **Zapisz i sprawdź połączenie z Miro**. Token jest przechowywany w Menedżerze poświadczeń Windows.

Diagramy trafiają tylko na tablice zespołu, w którym aplikacja jest zainstalowana; żeby wysłać do innej organizacji, zainstaluj aplikację także tam (tym samym przyciskiem).

Wysyłanie:

1. Kliknij **Wyślij do Miro** pod diagramem w rozmowie.
2. Wybierz tablicę (pierwsza jest ta użyta ostatnio; pole wyszukiwania filtruje po nazwie).
3. Miniatura pokazuje, co już jest na tablicy (szare), i gdzie trafi diagram (niebieska ramka, czerwona, gdy coś by zasłoniła). Aplikacja sama znajduje wolne miejsce; przeciągnij ramkę, żeby wybrać inne.
4. Kliknij **Wyślij do Miro**. Po wysłaniu w rozmowie pojawia się wiadomość z linkiem, który otwiera tablicę na nowej ramce.

Diagramy sekwencji trafiają jako linie życia z ponumerowanymi komunikatami w kolejności z EA (odpowiedzi przerywaną linią). Ponowne wysłanie tego samego diagramu stawia nową ramkę obok poprzedniej, więc nic, co zmieniono na tablicy, nie zostaje nadpisane.

13. Dokumenty: Word, Excel, obrazki PNG, Confluence

Asystent przygotowuje dokumenty z treści modelu i rozmowy. Gotowy plik pojawia się na czacie jako karta z przyciskami, a sam plik zostaje w folderze `attachments` w folderze danych (rozdział 18).

Rodzaj	Co zawiera	Przyciski na czacie
Word (DOCX)	Nagłówki, akapity, listy, tabele, diagramy z EA i obrazki narysowane wcześniej w rozmowie	Otwórz, Zapisz jako...
Excel (XLSX)	Kilka arkuszy, nagłówki z filtrem i zamrożonym pierwszym wierszem, liczby jako liczby; na przykład lista elementów, macierz, wynik analizy wpływu	Otwórz, Zapisz jako...
Obrazek (PNG)	Macierz (na przykład przypadków użycia i wymagań), tabela, wykres, schemat. Asystent widzi wynik i może go poprawić.	Zapisz jako..., Otwórz w pełnym rozmiarze
Strona Confluence	Kod strony dla edytora kodu źródłowego oraz diagramy i obrazki jako pliki w folderze <code><strona>_images</code>	Kopiuj kod strony, Otwórz folder z obrazkami, Zapisz jako...

Przygotuj dokument Word z opisem pakietu „Sklep”: lista przypadków użycia z opisami i diagram przypadków użycia.

Narysuj macierz przypadków użycia i wymagań z pakietu „Sklep” jako obrazek.

Strony Confluence

Asystent przygotowuje stronę w formacie edytora kodu źródłowego Confluence (storage XHTML): nagłówki, tabele, listy, spis treści, panele (informacja, uwaga, ostrzeżenie, wskazówka, sekcja rozwijana), statusy, zgłoszenia Jira, wzmianki o osobach i bloki kodu. Diagramy z EA i obrazki są dołączane jako załączniki.

Wklejanie ręczne (bez ustawień Confluence):

1. Kliknij **Kopiuj kod strony** na karcie strony.
2. W Confluence otwórz edycję strony i edytor kodu źródłowego (Source Editor), wklej kod i zatwierdź.
3. Kliknij **Otwórz folder z obrazkami** i dodaj obrazki do strony jako załączniki.

Publikacja przez asystenta. Po uzupełnieniu karty „**Confluence**” asystent na Twoją prośbę publikuje stronę przez REST API, wyłącznie przez HTTPS: tworzy nową stronę pod stroną nadrzędną albo aktualizuje istniejącą (wskazaną po ID albo tytule). Obrazki trafiają jako załączniki; istniejące dostają nową wersję. Diagramy są przy każdej publikacji eksportowane z EA na nowo, więc ponowna publikacja odświeża stronę po zmianach w modelu. Asystent pamięta strony, które opublikował.

Publikacja zawsze wymaga zatwierdzenia, także w trybie **auto**. Przed wysłaniem strony asystent sprawdza, czy w przestrzeni jest już strona o tym samym tytule albo wskazanym ID. Jeśli tak, panel zgody mówi wprost, że publikacja **ZASTĄPI** treść istniejącej strony, i podaje jej ID oraz wersję; poprzednia wersja zostaje w historii strony. Stronę wysyłasz przyciskiem **Zatwierdź**, a **Odrzuć** anuluje publikację.

Pole karty „Confluence”	Co wpisać
Adres Confluence	Na serwerach firmy (Data Center), na przykład <code>https://confluence.firma.pl</code> ; w chmurze (Cloud) <code>https://firma.atlassian.net/wiki</code> . Adres musi zaczynać się od <code>https://</code> .
E-mail (tylko Cloud)	Dla Confluence Cloud: e-mail konta. Dla Data Center zostaw puste.
Token	Data Center: token osobisty (Profil → Personal Access Tokens). Cloud: token API konta. Token jest zapisywany w Menedżerze poświadczeń Windows.
Przestrzeń (klucz)	Klucz przestrzeni, na przykład <code>ARCH</code>
Strona nadrzędna (ID)	Opcjonalnie: ID strony, pod którą powstają nowe strony

Confluence Cloud: skąd wziąć dane.

1. **Token**: tworzy się go na koncie Atlassian, nie w Confluence. Otwórz `https://id.atlassian.com/manage-profile/security/api-tokens` (konto Atlassian → Security → API tokens). Atlassian może najpierw poprosić o potwierdzenie tożsamości kodem wysłanym e-mailem.
2. Kliknij **Create API token** (zwykły, nie **Create API token with scopes**), nadaj nazwę, na przykład Loonar Compass, i wybierz ważność: domyślnie tydzień, najwyżej rok. Kliknij **Create** i skopiuj token: Atlassian pokazuje go tylko raz. Gdy wygaśnie, utwórz nowy i wklej go w karcie „Confluence”.
3. **E-mail**: adres konta Atlassian, do którego należy token.
4. **Adres**: Twoja witryna z `/wiki`, na przykład `https://twoja-witryna.atlassian.net/wiki`.
5. **Przestrzeń (klucz)**: jest w adresie każdej strony przestrzeni, po `/spaces/` (na przykład `.../wiki/spaces/PM/...` daje `PM`), oraz w **Ustawieniach przestrzeni**.
6. **Strona nadrzędna (ID)** (opcjonalnie): liczba w adresie strony, po `/pages/` (na przykład `.../pages/65818/...` daje `65818`).

Zapisz kartę przyciskiem **Zapisz i sprawdź połączenie z Confluence**.

Przygotuj stronę Confluence z opisem procesu „Obsługa reklamacji” z diagramem i opublikuj ją w przestrzeni.

14. Szablony dokumentów

Szablon mówi asystentowi, jak zbudować dokument: jakie sekcje ma mieć i co do nich wpisać. Asystent wypełnia go treścią z modelu i rozmowy i przygotowuje stronę Confluence albo dokument Word. Gdy prosisz o dokument, asystent sam sprawdza, czy pasuje do niego któryś szablon; czego nie wie, oznacza jako „do uzupełnienia”, zamiast zgadywać.

Po instalacji są dostępne szablony domyślne, w języku aplikacji:

- **Wstępna analiza wykonalności,**
- **BRD - Wymagania biznesowe,**
- **Opis zmian w systemie,**
- **Opis procesu.**

Widok „Szablony dokumentów”

- Lista szablonów z krótkim opisem.
- **Użyj** wstawia do czatu polecenie „Przygotuj stronę Confluence według szablonu »...« dla: ”. Uzupełnij je (na przykład o nazwę pakietu albo procesu) i wyślij. Jeśli wolisz dokument Word, zmień polecenie przed wysłaniem.
- **Edytuj** otwiera plik szablonu w Notatniku.
- **Otwórz folder szablonów** otwiera folder `templates` w Eksploratorze (starsze instalacje mogą nadal używać folderu `szablony`).

Własne szablony

Szablony to pliki Markdown (`.md`) ze wskazówkami dla asystenta. Pierwszy wiersz w postaci `> Szablon: opis` jest opisem widocznym na liście, a kolejne wiersze zaczynające się od `>` to wskazówki dla asystenta. Możesz zmieniać istniejące szablony, dodawać własne pliki do folderu szablonów albo poprosić asystenta, żeby zrobił szablon z przykładowego dokumentu:

Zrób szablon z załączonego dokumentu: zachowaj układ sekcji i dopisz, co w każdej powinno się znaleźć.

Zapis szablonu przez asystenta zawsze wymaga Twojej zgody, także w trybie **auto**, a panel zgody pokazuje cały tekst szablonu (rozdział 16).

15. Pamięć asystenta

Asystent pamięta ustalenia między rozmowami: konwencje modelowania, Twoje preferencje i poprawki oraz fakty o projektach i modelu. Zna je w każdej nowej rozmowie. Ustalenie na ten sam temat aktualizuje, zamiast dublować, a nieaktualne usuwa. Haseł, kluczy ani innych sekretów nie zapamiętuje.

Ustalenia asystent zapisuje sam, gdy uzna, że przydadzą się w kolejnych rozmowach, i mówi o tym jednym zdaniem. Możesz też poprosić o to wprost, zaczynając od „zapamiętaj...”:

Zapamiętaj: wymagania numerujemy FR01, FR02... i trzymamy w pakiecie „Wymagania”.

Zapamiętaj, że na diagramach BPMN zadania systemowe kolorujemy na zielono.

Ustalenia są wspólne dla wszystkich modeli. Jeśli chcesz, żeby ustalenie dotyczyło tylko jednego modelu, powiedz to wprost, na przykład „zapamiętaj to tylko dla tego modelu”: wtedy asystent korzysta z niego wyłącznie w rozmowach z tym modelem, a widok „Pamięć asystenta” oznacza je znacznikiem „Tylko dla modelu ...”.

Zapis ustalenia w pamięci (i jego usunięcie przez asystenta) zawsze wymaga Twojej zgody, w każdym trybie, a panel zgody pokazuje nazwę i całą treść notatki (rozdział 16). Zapis ustalenia nie przerywa pracy asystenta nad bieżącym zadaniem. Ustalenia dotyczące modelu mogą być nieaktualne, dlatego przed zapisem do modelu asystent sprawdza je w bazie. Pełna treść pamięci trafia do modelu AI w każdym zapytaniu (rozdział 18).

Widok „Pamięć asystenta”

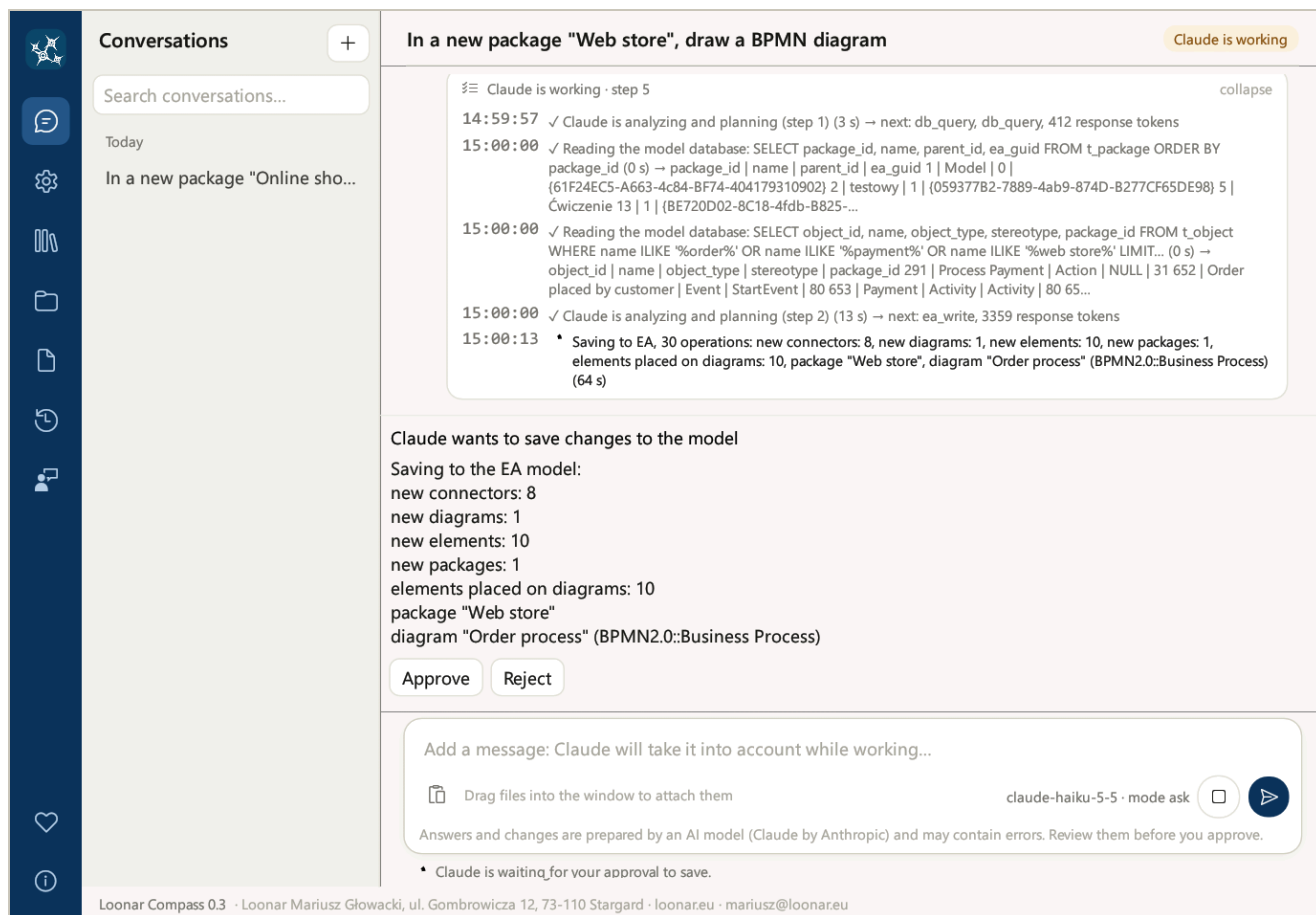
- Lista ustaleń z datą.
- **Usuń** od razu kasuje wybrane ustalenie.
- **Otwórz folder pamięci** otwiera folder `memory` w Eksploratorze. Ustalenia są w nim plikami Markdown, a spis jest w pliku `MEMORY.md`.

16. Zatwierdzanie zmian i dziennik zmian

Tryb ustawiasz w widoku **Ustawienia**, w karcie „Język i zatwierdzanie zmian”. Decyduje on, czy i kiedy aplikacja zapyta o zgodę przed zapisem do EA (zapis do pamięci asystenta i zapis szablonu zawsze wymagają zgody). Odczyt modelu i pokazywanie diagramów nigdy nie wymagają zgody. Bieżący tryb widać też w karcie wiadomości.

Tryb	Kiedy pyta o zgodę	Dla kogo
ask	Przed każdym zapisem	Praca na ważnym modelu; pełna kontrola
plan	Raz, przed pierwszym zapisem w danej odpowiedzi asystenta; kolejne zapisy tej odpowiedzi idą bez pytania. Wyjątek: usuwanie obiektów, przenoszenie ich do innego pakietu i kopiowanie pakietów zawsze wymagają osobnej zgody.	Większe zadania, gdy pierwszy zapis pokazuje kierunek
auto	Tylko przy działaniach, które zawsze wymagają zgody (poniżej)	Model testowy albo praca, którą łatwo cofnąć

Gdy potrzebna jest zgoda, nad kartą wiadomości pojawia się panel **„Asystent chce zapisać zmiany w modelu”** z opisem zmian i przyciskami **Zatwierdź** oraz **Odrzuć**. Opis podaje, ile pakietów, diagramów, elementów i połączeń powstanie albo się zmieni, nazwy nowych pakietów i diagramów, a przy usuwaniu, przenoszeniu i kopiowaniu także osobne wiersze z nazwami tych obiektów (USUNIĘCIE, PRZENIESIENIE, KOPIOWANIE). Ten sam panel pyta o zgodę na zapis ustalenia w pamięci, nadpisanie szablonu i publikację w Confluence. Odrzucony zapis niczego nie zmienia, a asystent dostaje informację o odrzuceniu i może zaproponować coś innego.



Opis zmian w panelu zgody ma część „**Przed** → **po:**”: co powstaje (utworzone obiekty), co zmienia się z jakiej wartości na jaką (pole: stara → nowa wartość) i co jest usuwane. Dzięki temu widzisz skutek zapisu, zanim klikniesz **Zatwierdź**.

Zawsze, w każdym trybie, Compass pyta też przed: publikacją w Confluence, cofnięciem zapisów, utworzeniem zgłoszeń, komentarzy i linków w Jirze, zapisem plików (eksport XMI, dokumenty z szablonów EA) i importem XMI, zapisem ustalenia w pamięci albo szablonu dokumentu, zapisem lub uruchomieniem skryptu EA, wywołaniem narzędzia serwera MCP, który nie jest zaufany, oraz przeczytaniem pliku, który nie jest załącznikiem rozmowy (rozdział 7).

Każdy zapis asystenta do modelu EA, także nieudany, trafia do dziennika zmian (plik `audit_log.jsonl` w folderze danych). Nie ma w nim zapisów odrzuconych przez Ciebie ani operacji, których aplikacja nie dopuściła do zapisu; dziennik nie obejmuje też pamięci, szablonów ani publikacji w Confluence.

17. Rozmowy: historia, wyszukiwanie, praca w tle

- **Nowa rozmowa:** przycisk + nad listą rozmów. Każda rozmowa jest zapisywana i wraca po ponownym uruchomieniu aplikacji.
- **Grupy:** rozmowy są pogrupowane na „Dzisiaj”, „Wczoraj”, „Ostatnie 7 dni” i „Wcześniej”.
- **Wyszukiwanie:** pole „Szukaj w rozmowach...” filtruje listę po tytule i po treści wiadomości.
- **Zmiana nazwy:** kliknij ołówek przy wybranej rozmowie, wpisz nazwę, **Enter** zapisuje, **Esc** anuluje.
- **Kilka rozmów naraz:** w trakcie pracy asystenta możesz zacząć nową rozmowę albo przejść do innej. Rozpoczęta praca toczy się w tle.
- **Usuwanie:** w aplikacji nie można usuwać rozmów. Każda rozmowa to plik JSON w folderze **history** w folderze danych (nazwą pliku jest identyfikator rozmowy, a tytuł jest w jego treści). Jeśli musisz usunąć rozmowę, zamknij aplikację i usuń jej plik; załączniki zostają w folderze **attachments**.
- **Model rozmowy:** każda rozmowa pracuje z własnym modelem EA, a nowa rozmowa zaczyna się od zatwierdzenia tego modelu. Ekran nowej rozmowy pokazuje model domyślny z ustawień. Przycisk **Wybierz inny model...** otwiera okno „Model tej rozmowy”, w którym wybierasz **Domyślny z ustawień** albo **Inny model:** **Baza PostgreSQL** (serwer, port, nazwa bazy, użytkownik, hasło, szyfrowanie połączenia) albo **Plik EA (.qea, .qeax)**, z kontem EA i opcjami zabezpieczeń jak w karcie Model. **Sprawdź połączenie** otwiera model, liczy pakiety, elementy i diagramy oraz zapisuje w nim pakiet próbny, który sprawdza i od razu usuwa. Przycisk **Zatwierdź model i rozpocznij** ustala model rozmowy: potem nie można go już zmienić, a przycisk **Model: ...** w nagłówku pokazuje go tylko do podglądu. Aby pracować na innym modelu, zacznij nową rozmowę. Zmiana modelu domyślnego w ustawieniach nie zmienia trwających rozmów. Hasła połączenia innego niż w ustawieniach trafiają do Menedżera poświadczeń Windows. W Compass w EA modelem rozmowy jest zawsze projekt otwarty w EA (rozdział 10).
- **Plik EA upuszczony na okno:** plik .qea lub .qeax upuszczony na czat staje się modelem rozmowy, a nie załącznikiem. Jeśli rozmowa ma już wiadomości, z tym plikiem zaczyna się nowa rozmowa.
- **Rozmowy z wcześniejszych wersji** przy pierwszym uruchomieniu wersji 0.5 lub nowszej dostają kopię modelu z ustawień, więc pracują z nim dalej, nawet gdy później zmienisz ustawienia. Po najechniu myszą na rozmowę na liście widać jej model.

Kropka przy rozmowie	Znaczenie
● pomarańczowa	Asystent w tej rozmowie pracuje (albo czeka na Twoją zgodę na zapis)
● zielona	Asystent skończył pracę w tle, a Ty jeszcze nie zajrzałeś do wyniku

Zapisy do EA z różnych rozmów wykonują się po kolei, nie jednocześnie. EA w tle ma otwarty jeden model naraz: przejście między rozmowami z różnymi modelami otwiera go ponownie, co trwa od kilku do około 25 sekund. Gdy czat jest przewinięty wyżej, kółko ze strzałką nad polem wiadomości przenosi na koniec rozmowy.

18. Dane, bezpieczeństwo i prywatność

Gdzie są Twoje dane

Co	Gdzie
Program	<code>%LOCALAPPDATA%\Programs\Loonar Compass</code> (instalacja „Tylko dla mnie”) albo <code>%ProgramFiles%\Loonar Compass</code> (instalacja dla wszystkich użytkowników): aplikacja, licencja, lista zmian, <code>THIRD-PARTY-NOTICES.txt</code> ; instrukcja użytkownika jest na loonar.eu (menu Start → Instrukcja użytkownika)
Folder danych użytkownika	<code>%APPDATA%\Loonar Compass</code> , osobny dla każdego użytkownika komputera: <ul style="list-style-type: none"><code>app.properties</code> : ustawienia, bez haseł, kluczy i tokenów,<code>history\</code> : historia rozmów,<code>attachments\</code> : załączniki i pliki przygotowane przez asystenta (Word, Excel, PNG, strony Confluence z folderami obrazków <code><strona>_images</code>),<code>memory\</code> : pamięć asystenta, ze spisem <code>MEMORY.md</code> ,<code>templates\</code> : szablony dokumentów,<code>audit_log.jsonl</code> : dziennik zmian w modelu,<code>confluence_pages.json</code> : strony opublikowane przez asystenta.
Klucz API Claude, klucz API OpenAI, hasło konta EA, hasło do bazy, tokeny Confluence i Miro, tokeny serwerów MCP (<code>mcp_...</code>)	Menedżer poświadczeń Windows (wpisy <code>loonar-compass</code> , zaszyfrowane przez system). W pliku ustawień nie ma żadnych sekretów.
Opis nieoczekiwanego błędu	<code>%TEMP%\loonar-compass-crash.log</code> (powstaje tylko po takim błędzie)

Historia rozmów to zwykłe pliki w Twoim profilu Windows, chronione Twoim kontem Windows; aplikacja ich nie szyfruje. Chronić dostęp do swojego konta i nie udostępniaj folderu danych.

Co trafia do dostawcy modelu AI

Żeby asystent mógł odpowiadać, w każdym zapytaniu do wybranego dostawcy modelu AI (Anthropic albo OpenAI, karta Model AI) aplikacja wysyła: treść rozmowy, załączone pliki, pełną treść pamięci asystenta, listę szablonów z opisami oraz to, co asystent odczytał z modelu w danej rozmowie, czyli wyniki jego zapytań do bazy (na przykład nazwy i opisy pakietów, elementów i diagramów) i obrazki diagramów. Nie jest wysyłana cała baza, tylko to, o co asystent zapytał. Treść szablonu trafia do modelu, gdy asystent go użyje, a po zmianie języka trafiają też pamięć i szablony do przetłumaczenia.

Dane trafiają bezpośrednio do wybranego dostawcy (api.anthropic.com albo api.openai.com) z użyciem Twojego klucza API; drugi dostawca nie dostaje niczego. Loonar ich nie otrzymuje, a aplikacja nie zbiera żadnych danych o korzystaniu (brak telemetrii).

Zanim użyjesz aplikacji na modelu firmowym, sprawdź, czy zasady Twojej organizacji i umowy z klientami pozwalają przekazywać treść modelu zewnętrznej usłudze AI, oraz jakie warunki przetwarzania danych obowiązują na Twoim koncie u wybranego dostawcy (Anthropic albo OpenAI).

Uprawnienia asystenta

- Bazę asystent czyta w sesji tylko do odczytu, a plik modelu otwiera tylko do odczytu: wyłącznie zapytaniem SELECT i WITH, z limitem czasu zapytania i bez funkcji serwera, które czytają pliki albo wpływają na inne sesje. Wszystkie zmiany idą przez Enterprise Architect, na koncie EA modelu (z karty Model albo z modelu rozmowy), więc obowiązują je uprawnienia i blokady tego konta. Compass w EA czyta i zapisuje model przez okno EA, w którym jest otwarty, na koncie, na którym zalogowałeś się w EA.
- Treść załączników, dokumentów, stron i modelu asystent traktuje jako dane, a nie jako polecenia: nie wykonuje znalezionych tam instrukcji i nie wysyła danych w miejsca, które ta treść wskazuje.
- W Confluence asystent działa z uprawnieniami tokena z karty „Confluence” i publikuje tylko po Twoim zatwierdzeniu.
- Asystent nie ma dostępu do plików na dysku poza tymi, które sam mu załączysz, oraz poza swoimi folderami pamięci i szablonów; każdy inny plik (dokument wymagań, folder z kodem) czyta dopiero po zapytaniu Cię o zgodę. Obrazki rysowane przez asystenta nie mogą wczytać plików z dysku ani z sieci.
- Login i hasło z ciągu połączenia otwartego okna EA nigdy nie trafiają do modelu językowego.

Połączenia i szyfrowanie

- **API modelu AI (Anthropic albo OpenAI), Confluence i Miro:** wyłącznie HTTPS (dla Confluence adres `http://` jest dopuszczalny tylko dla serwera na tym samym komputerze).
- **Baza modelu:** połączenie może być szyfrowane TLS, zależnie od ustawienia w karcie Model (rozdział 5). Administrator bazy może wymusić TLS dla wszystkich klientów wpisami `hostssl` w `pg_hba.conf`. Połączenie samego Enterprise Architect z bazą zależy od ustawień EA.
- **Proxy:** aplikacja korzysta z proxy z ustawień internetowych Windows oraz ze zmiennych środowiskowych `HTTPS_PROXY` i `ALL_PROXY`. Skrypty PAC i uwierzytelnianie NTLM/Kerberos na proxy nie są obsługiwane.
- **Chwilowe błędy API modelu AI** (przeciążenie 529 i inne błędy 5xx, limit zapytań 429) aplikacja ponawia automatycznie.

19. Rozwiązywanie problemów

Objaw	Co zrobić
Windows SmartScreen ostrzega przed instalatorem („System Windows ochronił ten komputer”)	Instalator nie jest jeszcze podpisany cyfrowo. Kliknij „Więcej informacji”, a potem „Uruchom mimo to”.
Windows blokuje instalator bez możliwości uruchomienia	Włączona jest Inteligentna kontrola aplikacji (Smart App Control), która blokuje niepodpisane aplikacje. Poczekać na wersję podpisaną cyfrowo.
Aplikacja otwiera się w widoku „Ustawienia”	Brakuje ustawień, hasła albo klucza; pasek stanu podaje, co uzupełnić. Uzupełnij karty i zapisz każdą przyciskiem „Zapisz i sprawdź...”.
„Plik ustawień ... zawiera błąd”	Komunikat podaje błędną wartość. Popraw ją w odpowiedniej karcie i zapisz kartę ponownie.
„Program napotkał nieoczekiwany błąd i musi zostać zamknięty”	Rozmowy i ustawienia są zapisane. Zgłoś problem (widok „Zgłoś błąd” albo mariusz@loonar.eu) i dołącz plik <code>%TEMP%\loonar-compass-crash.log</code> .
„Claude API odrzuciło klucz API (401)” albo „API OpenAI odrzuciło klucz (401)”	Klucz jest błędny albo unieważniony. Wklej aktualny klucz wybranego dostawcy (z konsoli Anthropic albo z platformy OpenAI) w karcie „Model AI”.
„API OpenAI nie zna modelu ... (404)”	Identyfikator modelu w karcie „Model AI” jest błędny albo Twoje konto OpenAI nie ma dostępu do tego modelu. Wybierz model z listy ▼.
Odpowiedź „która nie pochodzi z API” (dostęp do api.anthropic.com może blokować proxy lub zaporę)	Między komputerem a Internetem jest proxy albo zaporę. Poproś administratora o dostęp do api.anthropic.com (albo api.openai.com , gdy dostawcą jest OpenAI). Aplikacja korzysta z proxy z ustawień Windows oraz ze zmiennych <code>HTTPS_PROXY</code> i <code>ALL_PROXY</code> ; skrypty PAC i uwierzytelnianie NTLM/Kerberos nie są obsługiwane.
„Osiągnięto limit zapytań Claude API (429)” albo „Claude API jest przeciążone lub chwilowo niedostępne”	Aplikacja sama ponawia takie zapytania kilka razy. Jeśli komunikat mimo to się pojawi, odczekaj chwilę i wyślij wiadomość ponownie.
Test pliku: „to nie wygląda na plik modelu Enterprise Architect”	Wybierz plik projektu <code>.qea</code> lub <code>.qeax</code> utworzony w Enterprise Architect 16 lub nowszym. Starsze pliki <code>.eap/.eapx</code> najpierw przekonwertuj w EA do <code>.qea</code> (transfer projektu).
Test bazy: brak połączenia	Sprawdź serwer, port, nazwę bazy, konto i hasło oraz czy serwer PostgreSQL przyjmuje połączenia z tego komputera.

„Połączenie NIE jest szyfrowane: serwer nie oferuje TLS” albo brak połączenia przy ustawieniu „Wymagane”	Serwer bazy nie oferuje TLS. Poproś administratora bazy o włączenie TLS albo wybierz „Preferowane” (wtedy połączenie nie będzie szyfrowane).
Nie udaje się weryfikacja certyfikatu („Wymagane, z weryfikacją certyfikatu”)	Certyfikat serwera (albo urzędu, który go wystawił) musi być w magazynie certyfikatów Windows, a nazwa w polu Serwer musi się zgadzać z nazwą w certyfikacie. Poproś administratora o certyfikat albo wybierz „Wymagane” (bez weryfikacji certyfikatu).
Test bazy ostrzega, że konto ma uprawnienia superużytkownika	Uprawnienia superużytkownika nie są potrzebne. Poproś administratora o zwykłe konto z prawem odczytu i zapisu tabel modelu: asystent tylko czyta bazę, ale Enterprise Architect zapisuje zmiany tym samym kontem.
„Hasło do bazy nie może zawierać średnika (;)”	Ustaw dla konta bazy hasło bez średnika: Enterprise Architect nie przekaże go w swoim ciągu połączenia.
„EA nie odpowiedział w ciągu 300 s”	Najczęściej błędne konto lub hasło użytkownika modelu (EA czeka wtedy na niewidocznym oknie logowania), błędne dane bazy albo brak wolnej licencji. Popraw dane w karcie Model (konto EA, dane bazy) i użyj „Zapisz i sprawdź dostęp do EA” w karcie Enterprise Architect.
Pierwszy zapis albo pierwszy diagram trwa długo	To start Enterprise Architect w tle (od kilku do kilkunastu sekund). Kolejne wywołania są szybkie.
„Odpowiedź modelu została ucięta na limicie ... tokenów”	Zwiększ „Max tokenów” w karcie „Model AI” albo poproś o wykonanie zadania w mniejszych częściach.
„Asystent wykonał maksymalną liczbę kroków (24) i nie skończył”	Jedna odpowiedź może mieć najwyżej 24 kroki. Poproś w nowej wiadomości, żeby asystent kontynuował; duże zadania zlecaj częściami.
Nie widać zmian w Enterprise Architect	Jeśli w EA jest otwarty ten sam model, asystent odświeża go sam. W przeciwnym razie odśwież pakiet w przeglądarce projektu (F5) albo otwórz diagram ponownie.
„Nie można zablokować obiektu”	Obiekt jest zablokowany przez innego użytkownika modelu; komunikat podaje kto. Zdejmij blokadę w EA albo poproś o to jej właściciela.
Pomarańczowy „!” na ikonach Ustawienia, Pamięć asystenta, Szablony dokumentów	Po zmianie języka tłumaczenie czeka na połączenie z modelem AI. Sprawdź klucz API w karcie „Model AI” i kliknij „Przetłumacz teraz” w karcie „Język i zatwierdzanie zmian”.
Publikacja w Confluence się nie udaje	Sprawdź kartę „Confluence” przyciskiem „Zapisz i sprawdź połączenie z Confluence”: adres (zaczynający się od https://), token, klucz przestrzeni, a dla Cloud także e-mail. Stronę możesz też wkleić ręcznie przyciskiem „Kopiuje kod strony”.

Trzeba przywrócić wcześniejszą wersję pakietu z baseline'u	W EA: prawy przycisk na pakiecie → Manage Baselines → Compare/Restore.
Asystent twierdzi, że czegoś nie potrafi	Zapytaj: „Co potrafisz w tej aplikacji?”. Jeśli funkcja jest na liście w widoku „Changelog”, napisz mu to wprost i poproś o ponowną próbę.
Na wstążce Start w EA nie ma przycisku Loonar Compass albo jest nieaktywny	Uruchom instalator ponownie z zaznaczoną opcją „Dodaj Loonar Compass do Enterprise Architect”, a potem uruchom EA ponownie. Przycisk działa tylko przy otwartym projekcie (rozdział 10).
Nie da się przeciągnąć plików do okna	Aplikacja została uruchomiona jako administrator. Zamknij ją i uruchom zwykłym skrótem.
<code>LoonarCompass.exe --check</code> nic nie pokazuje w PowerShell albo <code>\$LASTEXITCODE</code> jest puste	Compass jest programem okienkowym, więc PowerShell na niego nie czeka. Przekaż jego wyjście dalej: <code>... Out-Host</code> (rozdział 23).
Gotowy serwer MCP uruchamiany przez <code>npx</code> nie startuje	Zainstaluj Node.js (zawiera npx) i uruchom Compass ponownie. Sprawdź organizację albo folder we wpisie serwera.
Nie da się zalogować do widoku webowego huba	Sprawdź nazwę użytkownika i hasło. Po 20 błędnych próbach z jednego adresu hub odrzuca go przez 10 minut. Administrator może ustawić nowe hasło w zakładce Users (rozdział 22).

20. Zgłaszanie błędów i nowe wydania

Zgłoś błąd

1. Kliknij ikonę **Zgłoś błąd** w pasku nawigacji.
2. Wpisz temat i opis: co robiłeś, co się stało, czego się spodziewałeś. Opis jest wymagany; możesz w nim wkleić treść błędu z czatu.
3. Dodaj zdjęcia, najlepiej zrzuty ekranu z zaznaczonym miejscem błędu: przeciągnij je do okna, wklej ze schowka (**Ctrl** + **V**) albo wybierz z dysku przyciskiem + **Dodaj zdjęcie**. Dodane zdjęcia widać jako miniatury; razem mogą mieć najwyżej 15 MB.
4. Opcjonalnie podaj e-mail do odpowiedzi. Opcja **Dołącz informacje techniczne** (domyślnie zaznaczona) dodaje wersję aplikacji, system, model, tryb zatwierdzania i ostatni komunikat, bez haseł i kluczy.
5. Kliknij **Wyślij zgłoszenie**. W programie pocztowym Windows otworzy się gotowa wiadomość do autora (mariusz@loonar.eu) ze zdjęciami jako załącznikami. Sprawdź ją i wyślij.

Zgłoszenie (opis, zrzuty ekranu, opcjonalne informacje techniczne i Twój adres e-mail) trafia do Loonar Mariusz Głowacki i służy wyłącznie do odpowiedzi i poprawy aplikacji; jest przechowywane do 2 lat po zamknięciu sprawy. Przed wysłaniem usuń ze zrzutów dane osobowe i informacje poufne. Szczegóły są w polityce prywatności, do której link jest w formularzu.

Zdjęcia trafiają do programu pocztowego przez tymczasowe kopie, które aplikacja usuwa, gdy program pocztowy je przejmie. Jeśli program pocztowy nie przyjmuje załączników z innych aplikacji, otworzy się wiadomość bez zdjęć oraz folder z ich kopiami: przeciągnij zdjęcia do wiadomości. Tylko w tym przypadku kopie zostają w folderze tymczasowym.

Nowe wydania (newsletter)

W karcie „**Nowe wydania (newsletter)**” w ustawieniach wpisz adres e-mail, zaznacz „Chcę dostawać informacje o nowych wydaniach” i kliknij **Zapisz**. Dostaniesz krótki e-mail, gdy wyjdzie nowa wersja: co nowego i skąd ją pobrać, bez reklam. Zapis i rezygnacja trafiają do autora wiadomości z programu pocztowego. Adres jest wykorzystywany wyłącznie do informacji o nowych wersjach, na podstawie Twojej zgody, do czasu jej wycofania; administratorem danych jest Loonar Mariusz Głowacki. Zgodę wycofasz w każdej chwili, odznaczając opcję i zapisując kartę albo pisząc e-mail. Pod informacją w karcie jest link do polityki prywatności.

21. Aktualizacja i odinstalowanie

Aktualizacja. Zamknij aplikację i uruchom instalator nowszej wersji. Kreator pokaże stronę „Co nowego” ze zmianami od zainstalowanej wersji, zaproponuje ten sam rodzaj instalacji i folder oraz zachowa Twoje ustawienia, hasła, historię rozmów, załączniki, pamięć, szablony i dziennik zmian. Aktualizację z wersji o dawnej nazwie opisuje rozdział 3.

Zamykanie aplikacji. Zamknięcie okna zamyka też sesję Enterprise Architect uruchomioną w tle przez aplikację, więc licencja EA jest od razu zwalniana.

Odinstalowanie. Zamknij aplikację. Użyj skrótu „Odinstaluj Loonar Compass” w menu Start albo pozycji „Loonar Compass” w ustawieniach Windows (Aplikacje → Zainstalowane aplikacje). Przy instalacji dla

wszystkich użytkowników Windows poprosi o uprawnienia administratora. Program odinstalowujący usuwa pliki programu, skróty i wpis w systemie, pokazując postęp na pasku. Twoje dane zostają na komputerze, chyba że zaznaczysz:

- „Usuń ustawienia, rozmowy, załączniki, pamięć asystenta, szablony dokumentów i dziennik zmian”: usuwa cały folder `%APPDATA%\Loonar Compass`,
- „Usuń klucz API, hasła i tokeny z Menedżera poświadczeń”: usuwa wszystkie sekrety aplikacji, także hasło do bazy i token Confluence.

Usuwane są tylko dane użytkownika, który uruchomił odinstalowanie; dane innych użytkowników zostają w ich profilach.

22. Compass Hub (usługa zespołowa)

Compass Hub to opcjonalna usługa dla zespołu, uruchamiana na komputerze lub serwerze zespołu. Daje wspólną politykę (reguły kontroli modelu i minimalny tryb zatwierdzania), centralny dziennik zmian ze wszystkich stanowisk, wspólne notatki, zestawy zmian do przeglądu, widok webowy modeli dla osób bez Enterprise Architect, zgłoszenia Jira i strony Confluence tworzone z widoku webowego oraz serwer MCP (tylko odczyt) dla innych narzędzi AI. Bez huba Compass działa samodzielnie, jak dotąd.

Dla użytkownika

Adres huba i swój token API dostajesz od administratora (zakładka **Users** widoku webowego). Wpisz je w karcie „**Zespół (Compass Hub)**” w widoku Ustawienia: **Adres huba** (musi zaczynać się od `https://`; `http://` tylko na tym komputerze), **Token** i opcjonalnie **Ten model w hubie** (identyfikator źródła, czyli modelu huba odpowiadającego Twojemu), a potem kliknij **Zapisz i sprawdź połączenie z hubem**. Test pokazuje Twoje konto, rolę i politykę zespołu. Z hubem:

- polityka zespołu `min_approval` może tylko zaostrzyć lokalny tryb zatwierdzania (od najluźniejszego: auto, plan, ask);
- reguły kontroli modelu z huba obowiązują, gdy nie ma lokalnego pliku `governance.json` (rozdział 7);
- każda zapisana zmiana trafia do zespołowego dziennika zmian; wysyłanie odbywa się w tle, więc niedostępny hub nie zatrzymuje pracy, a ostatni problem widać na karcie.

Instalacja huba

Hub to osobny program, `compass-hub`, dla Windows i Linuksa. Czyta modele wprost (baza PostgreSQL albo pliki `.qea/.qeax`), więc na serwerze nie jest potrzebny Enterprise Architect. Do pobrania:

- **Windows:** `CompassHub-0.8.2-windows-x64.zip` z plikiem `compass-hub.exe`, przykładowym `hub.json` i plikiem README.
- **Docker:** obraz `ghcr.io/loonar-eu/compass-hub:0.8.2` albo własna kompilacja w głównym folderze repozytorium: `docker build -f hub/Dockerfile -t compass-hub .`

W konsoli (Windows albo Linux). Hub działa, dopóki nie zamkniesz konsoli:

```
compass-hub hub.json
```

Jako usługa Windows, w konsoli uruchomionej jako administrator. Usługa „CompassHub” startuje od razu i razem z Windows; zatrzymujesz ją w Usługach albo poleceniem `sc stop CompassHub`, a usuwasz opcją `--service-uninstall`:

```
compass-hub --service-install C:\Hub\hub.json
compass-hub --service-uninstall
```

W Dockerze. Plik `hub.json` z `"listen": "0.0.0.0:8730"` umieść w folderze danych, który kontener widzi jako `/data`:

```
docker run -d --name compass-hub -p 8730:8730 -v /srv/compass-hub:/data \
-e LOONAR_SECRET_CLAUDE_API_KEY=... ghcr.io/loonar-eu/compass-hub:0.8.2
```

- Kontener działa jako użytkownik o identyfikatorze 10001: folder danych musi być dla niego zapisywalny, bo hub zapisuje tam `hub.json`, dziennik zmian, notatki, zestawy zmian i `hub.log`.
- Modele: pliki `.qea/.qeax` w folderze danych, ze ścieżkami widzianymi z wnętrza kontenera (na przykład `model.file=/data/models/shop.qea` w `app.properties` źródła), albo baza PostgreSQL.

Polecenie `LoonarCompass.exe --hub C:\Hub\hub.json` nadal działa: aplikacja desktopowa uruchamia wtedy hub bez okna. Hub zapisuje dziennik pracy w pliku `hub.log` obok `hub.json`. `compass-hub --version` wypisuje wersję.

Sekrety huba

Klucze, tokeny i hasła nie leżą w `hub.json`. `compass-hub` bierze każdy z nich ze zmiennej środowiska `LOONAR_SECRET_<NAZWA>` albo z pliku wskazanego w `LOONAR_SECRET_<NAZWA>_FILE` (na przykład sekretu Dockera); w Windows także z Menedżera poświadczeń konta, na którym działa hub. Dla usługi Windows ustaw je jako systemowe zmienne środowiska.

Sekret	Do czego
<code>LOONAR_SECRET_CLAUDE_API_KEY</code> albo <code>LOONAR_SECRET_OPENAI_API_KEY</code>	Model AI dla zakładki Ask
<code>LOONAR_SECRET_JIRA_TOKEN</code>	Jira zespołu (<code>integrations</code>)
<code>LOONAR_SECRET_CONFLUENCE_TOKEN</code>	Confluence zespołu (<code>integrations</code>)
<code>LOONAR_SECRET_DB_PASSWORD</code>	Hasło do bazy PostgreSQL z modelem

Konta i logowanie

Do widoku webowego (`https://<hub>/` ; `http://` tylko na tym samym komputerze) ludzie logują się nazwą użytkownika i hasłem. Compass i klienci MCP używają tokenu API.

Konto domyślne. Nowy hub (`hub.json` bez użytkowników) tworzy konto `admin` z hasłem `password` (z zerem). Przy pierwszym logowaniu hub prosi o zmianę hasła i do tego czasu nie pozwala na nic innego. Zrób to, zanim hub będzie dostępny z innych komputerów.

- Hasło ma co najmniej 10 znaków i nie może być hasłem domyślnym ani nazwą użytkownika. Hub przechowuje hasła tylko jako skróty PBKDF2.
- Zalogowanie trwa 12 godzin; **Sign out** kończy je od razu.
- Po 20 błędnych hasłach albo tokenach z jednego adresu ten adres jest odrzucany przez 10 minut.
- Link **Sign in with a token instead** na stronie logowania pozwala zalogować się tokenem API.

Zakładka **Users** (tylko dla administratorów):

- **Add a user** (dodaj użytkownika): nazwa, rola i pierwsze hasło, które użytkownik musi zmienić przy pierwszym logowaniu.
- **Role**: zmienia rolę (viewer, architect albo admin).
- **Reset password**: ustawia nowe hasło, które użytkownik musi zmienić przy następnym logowaniu.
- **New API token**: token dla Compassa (Ustawienia → Zespół (Compass Hub)) i dla klientów MCP. Jest pokazywany tylko raz, a poprzedni token użytkownika przestaje działać.

- **Remove:** usuwa użytkownika. Hub zawsze zachowuje co najmniej jednego administratora, a siebie usunąć nie możesz.

Zmiana albo usunięcie użytkownika kończy jego zalogowania. Każda zmiana użytkowników trafia do dziennika zmian.

Konfiguracja (hub.json)

```
{
  "listen": "127.0.0.1:8730",
  "sources": [
    { "id": "shop-db", "name": "Shop (PostgreSQL)", "properties": "shop-db/app.properties" },
    { "id": "shop-file", "name": "Shop (file)", "properties": "shop-file/app.properties" }
  ],
  "users": [
    { "name": "anna", "role": "admin", "password_pbkdf2": "pbkdf2-sha256$210000$. . . " },
    { "name": "bob", "role": "architect", "token_sha256": "<SHA-256 token>" }
  ],
  "integrations": {
    "jira": {
      "url": "https://firma.atlassian.net", "user": "hub@firma.pl", "project": "ARCH", "issue_type": "Task",
      "confluence": {
        "url": "https://firma.atlassian.net/wiki", "user": "hub@firma.pl", "space": "ARCH", "parent": "65818"
      }
    },
    "policy": { "min_approval": "ask", "ask_min_role": "viewer", "ask_per_hour": 20,
      "governance": { ... governance.json ... } }
  }
}
```

- **listen:** adres i port; domyślnie `127.0.0.1:8730`, w Dockerze `0.0.0.0:8730`. Poza jednym komputerem postaw hub za HTTPS (zobacz „Bezpieczeństwo huba”).
- **sources:** modele zespołu. `properties` to plik `app.properties` danego modelu (w tym samym formacie co plik ustawień Compassa), ścieżka względem `hub.json`.
- **users:** nie musisz ich wpisywać. Pomiń `users`, żeby dostać domyślnego administratora, a użytkownikami zarządzaj w zakładce **Users**: hub zapisuje ich z powrotem w `hub.json` tylko jako skróty (`password_pbkdf2`, `token_sha256`). Wpis ze zwykłym `token` (co najmniej 16 losowych znaków) jest nadal przyjmowany i zamieniany na `token_sha256`, gdy hub zapisuje użytkowników. `compass-hub --hub-token` (albo `LoonarCompass.exe --hub-token`) wypisuje nowy losowy token i jego skrót. Nazwy i tokeny muszą być unikalne.
- **integrations:** Jira i Confluence zespołu (opis niżej); tokeny pochodzą z sekretów huba.
- **policy:** `min_approval` (`ask`, `plan` lub `auto`), `ask_min_role` (kto może korzystać z **Ask** w widoku webowym; domyślnie `viewer`), `ask_per_hour` (pytania na użytkownika i godzinę; domyślnie 20) i `governance` w formacie `governance.json`.

Rola	Uprawnienia
viewer	Odczyt: widok webowy, REST, MCP
architect	To, co viewer, oraz wysyłanie wpisów do dziennika, edycja notatek zespołu, decyzje o zestawach zmian innych osób oraz tworzenie zgłoszeń Jira i stron Confluence z widoku webowego
admin	To, co architect, oraz zarządzanie użytkownikami (zakładka Users), zmiana polityki (<code>PUT /api/policy</code> , zapisywana w <code>hub.json</code>) i zwolnienie zestawu zmian, który został w stanie „wprowadzany”, bo Compass przestał działać: po sprawdzeniu modelu Not applied – free it (nie wprowadzono, zwolnij) albo It is in the model – mark applied (jest w modelu, oznacz jako wprowadzony)

Nikt, nawet admin, nie zatwierdza własnego zestawu zmian. Zestaw zmian jest wprowadzany w dwóch krokach: jeden Compass go rezerwuje, a po zapisie zgłasza wynik, więc nie da się go wprowadzić dwa razy.

Jira i Confluence z huba

Z `integrations` w `hub.json` architekci i administratorzy dostają przyciski w widoku webowym. Hub działa na własnym koncie technicznym: `user` (e-mail dla Cloud; puste dla Data Center z osobistym tokenem dostępu) i tokeny z sekretów `JIRA_TOKEN` i `CONFLUENCE_TOKEN`.

- **Create Jira issue** przy elemencie (**Search** → element): zgłoszenie w projekcie `project`, typu `issue_type` (domyślnie Task), ze szczegółami elementu w opisie.
- **Create Jira issue for this change set** w zestawie zmian: jedno zgłoszenie na zestaw zmian, z jego opisem i podglądem; link do zgłoszenia widać w zestawie zmian.
- **Publish this package to Confluence** w kontroli modelu (**Model check**) pakietu: strona tekstowa w przestrzeni `space` (pod stroną `parent`, jeśli ją podano) z elementami, ich opisami i relacjami. Nie ma na niej obrazków diagramów, bo hub nie ma EA. Kolejna publikacja pakietu zastępuje stronę.

Każda taka akcja trafia do dziennika zmian. Asystent w **Ask** nadal tylko czyta: nie może tworzyć zgłoszeń ani stron.

Widok webowy

Pod adresem `https://<hub>/` (`http://` tylko na tym samym komputerze) logujesz się nazwą użytkownika i hasłem. Widok pokazuje drzewo pakietów, wyszukiwanie, szczegóły elementu z analizą wpływu, kontrolę modelu według reguł zespołu, luki, dziennik zmian, notatki zespołu i zestawy zmian, a administratorom także użytkowników. W zakładce **Ask** asystent odpowiada na pytania o model. Tylko czyta model: nie ma dostępu do plików, Jiry, Confluence ani serwerów MCP i nie może niczego zaproponować ani zapisać; odpowiada na jedno pytanie naraz.

Bezpieczeństwo huba

Hub komunikuje się zwykłym HTTP. Do dostępu z innych komputerów postaw go za reverse proxy z HTTPS (na przykład IIS albo nginx) i podawaj ludziom adres `https://`. Od razu zmień hasło konta domyślnego. Chroń plik `hub.json` i folder danych uprawnieniami do plików. Zapytania mają ograniczony rozmiar i czas. Po 20 błędnych hasłach albo tokenach z jednego adresu ten adres jest odrzucany przez 10 minut; poprawny token nadal działa. Błędy wewnętrzne są pokazywane tylko jako identyfikator, a szczegóły są w `hub.log`. Widok webowy trzyma zalogowanie tylko na czas sesji przeglądarki i ma przycisk **Sign out** (wyloguj).

Serwer MCP

Serwer MCP (tylko odczyt) działa jako JSON-RPC przez HTTP POST pod adresem `/mcp`, z nagłówkiem `Authorization: Bearer <token>`. Narzędzia: `list_sources`, `list_packages`, `search_elements`, `element_details`, `impact_analysis`, `model_lint`, `gap_analysis`.

23. Tryb sprawdzania w potokach CI

Z opcją `--check` Loonar Compass sprawdza model bez okna i bez modelu AI, na przykład w potoku CI po każdej zmianie modelu albo kodu. Sprawdzenia wykonuje program, więc ten sam model zawsze daje ten sam wynik, a w modelu nic nie jest zapisywane.

```
LoonarCompass.exe --check --model <plik.qea|app.properties>
  [--package <id|nazwa>] [--rules governance.json] [--code <folder>]
  [--requirements <dokument>] [--cmdb <eksport.csv|json>] [--report raport.md]
  [--junit junit.xml] [--fail-on error|warning|info|never] [--lang en|pl|de|fr]
```

Opcja	Znaczenie
<code>--model</code>	Plik EA .qea albo .qeax, czytany wprost i tylko do odczytu (Enterprise Architect nie jest potrzebny), albo plik <code>app.properties</code> instalacji Compassa, na przykład z modelem w PostgreSQL. Bez tej opcji: ustawienia użytkownika Windows, który uruchamia polecenie.
<code>--package</code>	Sprawdzany pakiet (id albo nazwa), razem z podpakietami. Bez tej opcji: wszystkie pakiety główne modelu.
<code>--rules</code>	Plik <code>governance.json</code> z regułami zespołu (rozdział 7). Bez tej opcji: <code>governance.json</code> obok podanego <code>app.properties</code> albo polityka jego huba; dla pliku modelu tylko reguły wbudowane.
<code>--code</code>	Folder z kodem: model jest porównywany z kodem (podrozdział 8.6), a z <code>--requirements</code> każde wymaganie jest też prowadzone do kodu i testów (rozdział 7, „łańcuch wymagań”).
<code>--requirements</code>	Dokument wymagań: .docx, .md, .txt albo .html (podrozdział 8.5).
<code>--cmdb</code>	Eksport z CMDB, CSV albo JSON (rozdział 7, „CMDB a model”).
<code>--report</code>	Zapisuje raport Markdown także do tego pliku.
<code>--junit</code>	Zapisuje plik JUnit XML dla potoku.
<code>--fail-on</code>	Najniższa waga, przy której sprawdzenie się nie udaje: <code>error</code> (domyślnie), <code>warning</code> , <code>info</code> albo <code>never</code> .
<code>--lang</code>	Język raportu: <code>en</code> (domyślnie), <code>pl</code> , <code>de</code> albo <code>fr</code> .

Co jest sprawdzane

Część	Kiedy	Waga
Kontrola modelu: kontrole wbudowane, relacje ArchiMate 3.2, reguły zespołu	zawsze	jak w kontroli modelu (rozdział 7): błąd, ostrzeżenie albo informacja
Model a kod	<code>--code</code>	różnice: ostrzeżenie
CMDB a model	<code>--cmdb</code>	różnice: ostrzeżenie
Wymagania	<code>--requirements</code>	wymaganie, którego nie ma w modelu, które ma tylko kandydatów (bez relacji śledzenia) albo ma zgodny identyfikator, ale inną treść: ostrzeżenie; z <code>--code</code> wymaganie bez kodu albo bez testów: informacja

Wynik

- **Raport:** Markdown na standardowym wyjściu. Zaczyna się od wyniku, PASSED albo FAILED, i liczby grup błędów i ostrzeżeń, a dalej ma sekcję dla każdej części.
- **JUnit:** zestaw testów (test suite) dla każdej części i przypadek testowy dla każdej grupy ustaleń. Grupy z wagą na progu `--fail-on` albo wyżej są niepowodzeniami, więc potok pokazuje je jako nieudane testy.
- **Kod wyjścia:** 0 sprawdzenie przeszło; 1 ustalenia na progu `--fail-on` albo wyżej; 2 błędne argumenty, model, którego nie da się odczytać, albo plik, którego nie da się zapisać (przyczyna trafia na wyjście błędów).

Program okienkowy w PowerShell. `LoonarCompass.exe` jest programem okienkowym, więc PowerShell na niego nie czeka. Przekaż jego wyjście dalej, na przykład `... | Out-Host : PowerShell` wtedy czeka, pokazuje raport i ustawia `$LASTEXITCODE`. Hasło do bazy PostgreSQL z modelem pochodzi ze zmiennej środowiska `LOONAR_DB_PASSWORD` (zmiennej tajnej potoku) i ma pierwszeństwo przed Menedżerem poświadczeń.

GitLab CI

Zadanie na runnerze Windows (PowerShell) z Loonar Compass zainstalowanym dla wszystkich użytkowników. Zadanie uruchamia sprawdzenie i publikuje `junit.xml` jako raport testów:

```
compass-check:
  tags: [windows]
  script:
    - '& "C:\Program Files\Loonar Compass\LoonarCompass.exe" --check --model model\shop.qea --rules governance.json --code src --junit junit.xml --report report.md | Out-Host'
    - 'if ($LASTEXITCODE -ne 0) { exit $LASTEXITCODE }'
  artifacts:
    when: always
    paths: [report.md]
    reports:
      junit: junit.xml
```

GitHub Actions

Zadanie na własnym runnerze Windows (self-hosted) z Loonar Compass zainstalowanym dla wszystkich użytkowników. Zadanie uruchamia sprawdzenie i przesyła `junit.xml` oraz raport jako artefakt, który wybrany krok raportu JUnit może pokazać jako wyniki testów:

```
jobs:
  compass-check:
    runs-on: [self-hosted, windows]
    steps:
      - uses: actions/checkout@v4
      - name: Model check
        shell: pwsh
        run: |
          & "C:\Program Files\Loonar Compass\LoonarCompass.exe" --check --model model\shop.qea --
rules governance.json --code src --junit junit.xml --report report.md | Out-Host
          exit $LASTEXITCODE
      - name: Publish the results
        if: always()
        uses: actions/upload-artifact@v4
        with:
          name: compass-check
          path: |
            junit.xml
            report.md
```

Dla modelu w PostgreSQL podaj w `--model` plik `app.properties` bez haseł i ustaw `LOONAR_DB_PASSWORD`: w GitLabie jako maskowaną zmienną CI/CD, w GitHubie z sekretu (`env: LOONAR_DB_PASSWORD: ${ secrets.EA_DB_PASSWORD }`).

24. Licencja, znaki towarowe i kontakt

Licencja. Loonar Compass jest oprogramowaniem typu freeware. W skrócie:

- korzystanie jest bezpłatne, także zawodowe i komercyjne, na dowolnej liczbie komputerów, bez ograniczeń czasu i terytorium,
- aplikację wolno kopiować i przekazywać innym wyłącznie bezpłatnie, w całości i bez zmian, razem z plikami LICENSE i THIRD-PARTY-NOTICES,
- bez pisemnej zgody właściciela nie wolno jej zmieniać, dekompilować, sprzedawać ani wynajmować (z wyjątkiem uprawnień, których nie można wyłączyć umową),
- odpowiedzi, zmiany w modelu i dokumenty przygotowuje model AI: sprawdzaj je przed zatwierdzeniem, tryb **auto** włączasz na własne ryzyko, a przed pracą rób kopie zapasowe modelu,
- aplikacja nie zbiera danych (brak telemetrii); treść potrzebna do odpowiedzi trafia bezpośrednio do wybranego dostawcy modelu AI (Anthropic albo OpenAI) na podstawie Twojej umowy z tym dostawcą,
- Enterprise Architect, Claude API, OpenAI API, Confluence, Miro i inne usługi wymagają odrębnych licencji i kont ich dostawców; opłaty za nie ponosisz Ty.

Wiążąca jest polska wersja licencji; wersje angielska, niemiecka i francuska są tłumaczeniami. Pełna treść jest w widoku **Informacje** oraz w pliku `LICENSE.txt` w folderze programu. Lista zmian i znanych ograniczeń tej wersji jest w widoku **Changelog**.

Prywatność i nota prawna. Polityka prywatności: <https://loonar.eu/pl/privacy/>. Nota prawna: <https://loonar.eu/pl/legal/>. Link do polityki prywatności jest też w widoku Informacje, w formularzu

zgłoszenia błędu i w karcie newslettera.

Kod źródłowy i komponenty open source. Kod źródłowy aplikacji jest publiczny:

<https://github.com/loonar-eu/loonar-compass>. Aplikacja zawiera biblioteki i czcionki open source; każda z nich ma własną licencję. Ich lista jest w pliku `THIRD-PARTY-NOTICES.txt` w folderze programu, a w widoku Informacje otwiera ją przycisk **Pokaż licencje komponentów** w części „Komponenty open source”.

Znaki towarowe. Sparx Systems i Enterprise Architect są zarejestrowanymi znakami towarowymi Sparx Systems Pty Ltd. Claude i Anthropic są znakami towarowymi Anthropic, PBC. OpenAI i GPT są znakami towarowymi OpenAI. Atlassian, Confluence i Jira są znakami towarowymi Atlassian. Microsoft, Windows, Word, Excel, Azure DevOps i GitHub są znakami towarowymi grupy Microsoft. GitLab jest znakiem towarowym GitLab Inc., ServiceNow – ServiceNow, Inc., a Docker – Docker, Inc. PostgreSQL jest zarejestrowanym znakiem towarowym PostgreSQL Community Association. Loonar Compass jest niezależnym produktem, niepowiązanym z tymi firmami, nieautoryzowanym ani niesponsorowanym przez nie.

Właściciel i kontakt. Loonar Mariusz Głowacki, ul. Gombrowicza 12, 73-110 Stargard.

E-mail: mariusz@loonar.eu, strona: <https://loonar.eu> (z pytaniami i odpowiedziami). Błędy najwygodniej zgłaszać przez widok **Zgłoś błąd** (rozdział 20). Problemy z bezpieczeństwem zgłaszaj prywatnie e-mailem na mariusz@loonar.eu, z tematem „Security: Loonar Compass”, a nie w publicznym zgłoszeniu na GitHubie.

Wsparcie projektu. Rozwój aplikacji możesz dobrowolnie wesprzeć na <https://suppi.pl/loonar> (widok **Wesprzyj projekt**, rozdział 4).

Loonar Compass 0.8.2 · © 2026 Loonar Mariusz Głowacki. Wszelkie prawa zastrzeżone.