

# Baza wiedzy Jarvis — ZETTELKASTEN + Źródła wiedzy

## Metodologia zarządzania wiedzą z research, dyskusji i transkrypcji

**Data:** 17.01.2026 **Autor:** Research session \_jarvis **Status:** Do analizy i przemyślenia

**Kontekst:** Projekt \_jarvis — system akumulacji wiedzy dla asystenta AI

---

### Streszczenie

Dokument opisuje dwuczęściową architekturę bazy wiedzy dla projektu Jarvis:

1. **ZETTELKASTEN** — system atomowych notatek z wagami pewności i powiązaniemi
2. **Źródła wiedzy** — architektura podłączania zewnętrznych i budowania własnych źródeł (w tym transkrypcje YT)

**Kluczowy problem:** Jak zapewnić, że błędne wnioski z dyskusji nie zniekształcają całego systemu?

**Rozwiązanie:** System wag (confidence scoring) + deprecacja + sieć powiązań.

---

### Część 1: ZETTELKASTEN — Atomowe notatki

#### Czym jest Zettelkasten?

Zettelkasten (niem. “pudełko na kartki”) to metoda zarządzania wiedzą opracowana przez Niklasa Luhmanna. Kluczowe zasady:

1. **Atomowość** — jedna notatka = jeden fakt/wniosek
2. **Powiązania** — notatki łączą się w sieć (nie hierarchię)
3. **Identyfikatory** — każda notatka ma unikalny ID
4. **Kontekst** — notatka wie skąd pochodzi i z czym się łączy

#### Dlaczego Zettelkasten dla Jarvisa?

Problem	Rozwiązanie Zettelkasten
Wiedza ginie w długich dokumentach	Atomowe notatki — łatwe do znalezienia
Brak śladu źródła	Każda notatka ma source
Nieaktualne informacje	System deprecacji + superseded_by
Błędne wnioski z dyskusji	Wagi pewności (confidence scoring)
Brak powiązań między faktami	Sieć linków między notatkami

## Struktura katalogów

```
docs/central/ZETTELKASTEN/
└── _README.md          # Instrukcja systemu
└── _INDEX.md           # Spis wszystkich notatek
└── 202601-001.md       # Pierwsza notatka ze stycznia 2026
└── 202601-002.md
└── ...
└── 202601-NNN.md
```

## Format pojedynczej notatki

# YYYYMM-NNN: [Tytuł]

```
**confidence:** [0.0-1.0]
**source:** [URL, transcript-ID, user-discussion-DATE]
**version:** 1
**deprecated:** false
**superseded_by:** []
**tags:** [@projekt, @domena, @temat]
```

### ## Fakt

[Atomowy fragment wiedzy – 1-3 akapity max]

### ## Kontekst

[Opcjonalnie: dlaczego to ważne, kiedy użyć]

### ## Powiązania

- → 202601-XXX: [opis relacji – "ta notatka wskazuje na"]
- ← 202601-YYY: [opis relacji – "inna notatka wskazuje tu"]

### ## Źródło surowe

> [Cytat ze źródła, jeśli istnieje]

## Przykład notatki

# 202601-004: Prompt Caching – 90% oszczędności na powtarzalnych kontekstach

```
**confidence:** 0.95
**source:** https://platform.claude.com/docs/en/build-with-claude/prompt-caching
**version:** 1
**deprecated:** false
**superseded_by:** []
**tags:** [@_jarvis, @claude-api, @optymalizacja, @koszty]
```

### ## Fakt

Prompt caching pozwala oznaczyć statyczną zawartość jako cacheowaną.  
Kolejne requesty czytają z cache za 10% ceny.

### Parametry:

- Cache write: \$3.75/MTok (1.25x cena bazowa)
- Cache read: \$0.30/MTok (0.1x cena bazowa)
- TTL: 5 minut
- ROI: po ~13 requestach

## ## Kontekst

Dla Jarvisa – cache CLAUDE\_RULES.md, KNOWLEDGE.md.  
Przy 100+ requestów/dzień = znacząca oszczędność.

## ## Powiązania

- → 202601-002: Tool Runner + caching = optymalna kombinacja
- → 202601-005: W L2+ cache na pamięć długoterminową

## ## Źródło surowe

> "Prompt caching allows you to cache the prefix of your prompts,  
> reducing costs by up to 90% for repeated content."

---

## System wag pewności (Confidence Scoring)

### Skala pewności

Poziom	Zakres	Opis	Przykład źródła
<b>VERY HIGH</b>	0.90-1.0	Zweryfikowana, oficjalne	Dokumentacja API, kod źródłowy
<b>HIGH</b>	0.75-0.89	Wiarygodne, ale mogą być nieaktualne	Blogi techniczne, GitHub issues
<b>MEDIUM</b>	0.50-0.74	Opinie, częściowo zweryfikowane	Stack Overflow, dyskusje
<b>LOW</b>	0.25-0.49	Hipotezy, mogą być błędy	Transkrypcje YT, wstępne wnioski
<b>SPECULATIVE</b>	0.0-0.24	Domysły, nieweryfikowalne	Luźne pomysły, spekulacje

## Matryca pewności źródeł

Źródło	Avg Confidence	Uzasadnienie
platform.claude.com	0.95	Oficjalna dokumentacja Anthropic
modelcontextprotocol.io	0.92	Oficjalna dokumentacja MCP
docs.n8n.io	0.90	Oficjalna dokumentacja n8n
github.com (issues)	0.80	Weryfikowalne, ale mogą być nieaktualne
tech blogs	0.75	Opinie ekspertów
user-discussion	0.60-0.70	Hipotezy, niezweryfikowane
youtube-transcript	0.65	Mogą być błędy w transkrypcji

## Jak wagi chronią przed błędami?

**Scenariusz:** W dyskusji z użytkownikiem pada wniosek “n8n nie obsługuje webhooków z Telegrama”.

1. **Bez systemu wag:** Wniosek trafia do bazy wiedzy jako fakt → Jarvis powtarza błąd

2. **Z systemem wag:**

- Notatka: confidence: 0.60 (user-discussion)
- Przy podejmowaniu decyzji Jarvis może filtrować: “użyj tylko high-confidence”
- Gdy okaże się że to błąd → deprecated: true, superseded\_by: [202601-XXX]

## Deprecacja i rewizja

Gdy informacja okazuje się błędna lub nieaktualna:

```
---  
deprecated: true  
version: 2  
superseded_by: [202601-052]  
revision_date: 2026-02-15  
---
```

```
# 202601-023: [Tytuł] [DEPRECATED]
```

**\*\*UWAGA:\*\*** Ta notatka została zdeprecjonowana przez 202601-052.  
Powód: API zostało zaktualizowane, stary wniosek już nie obowiązuje.

[\[Oryginalna treść zachowana dla audytu\]](#)

**Zasada:** Nie usuwamy — archiwizujemy z powodem. Git history = pełny audit trail.

---

## KNOWLEDGE\_GRAPH — Mapa powiązań

### Struktura grafu

```
# Domeny wiedzy (nodes)
```

```
jarvis-core:
```

- architecture
- roadmap
- fazy
- knowledge
- transcripts

```
anthropic-api:
```

```
models:
```

- claude-opus-4-5
- claude-sonnet-4-5
- claude-haiku-4-5

```
features:
```

- tool-use
- streaming
- prompt-caching

```
# Powiązania (edges)
202601-001 --"foundational"--> anthropic-api
202601-002 --"extends"--> 202601-001
202601-003 --"enables"--> jarvis-core.architecture
```

## Typy relacji

Symbol	Relacja	Opis
→	points_to	Ta notatka wskazuje na inną
←	pointed_by	Inna notatka wskazuje na tę
↔	bidirectional	Wzajemne powiązanie
⊗	supersedes	Ta notatka zastępuje/unieważnia inną
∈	belongs_to	Należy do domeny/klastra

## Klastry tematyczne

### ### Cluster: Anthropic Technology Stack

- 202601-001 (MCP)
- 202601-002 (Tool Use)
- 202601-004 (Caching)

### ### Cluster: Jarvis Architecture

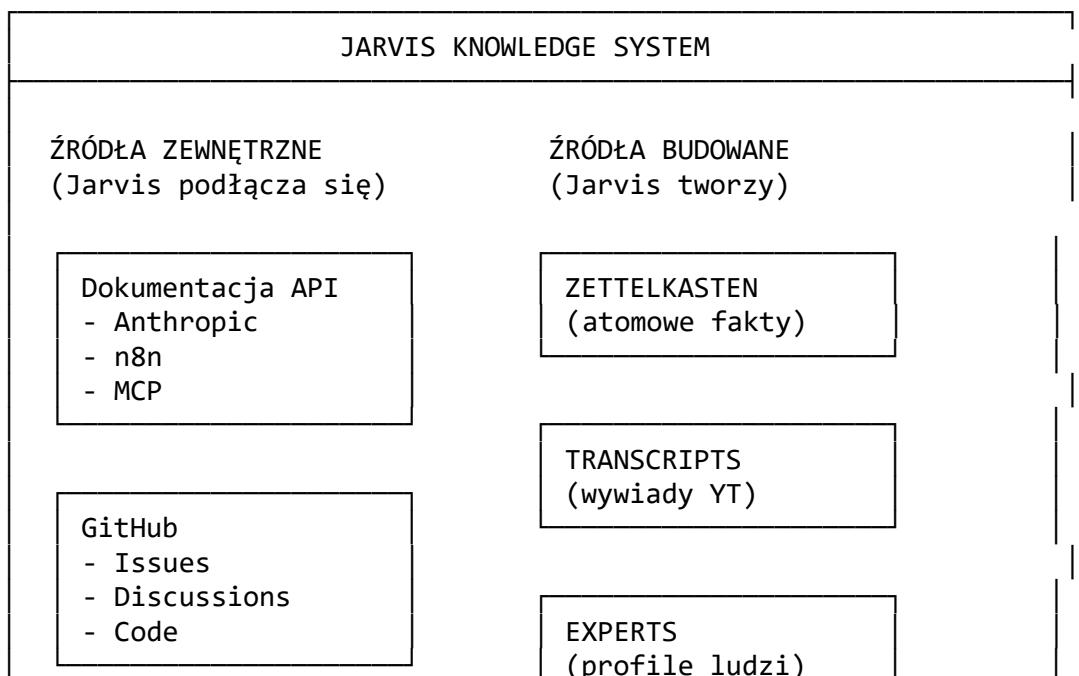
- 202601-003 (Orchestrator)
- 202601-005 (Autonomy levels)

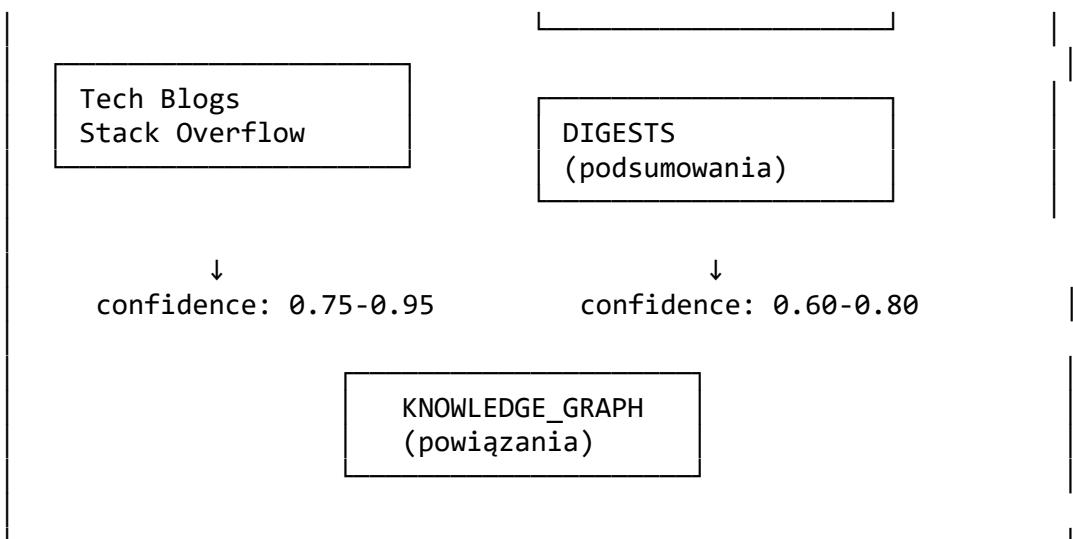
### ### Cluster: Knowledge Sources

- 202601-006 (Architecture)
- 202601-007 (YT Transcripts)

## Część 2: Źródła wiedzy — Architektura podłączania

### Dwa kierunki pozyskiwania wiedzy





## Lista źródeł do połączenia

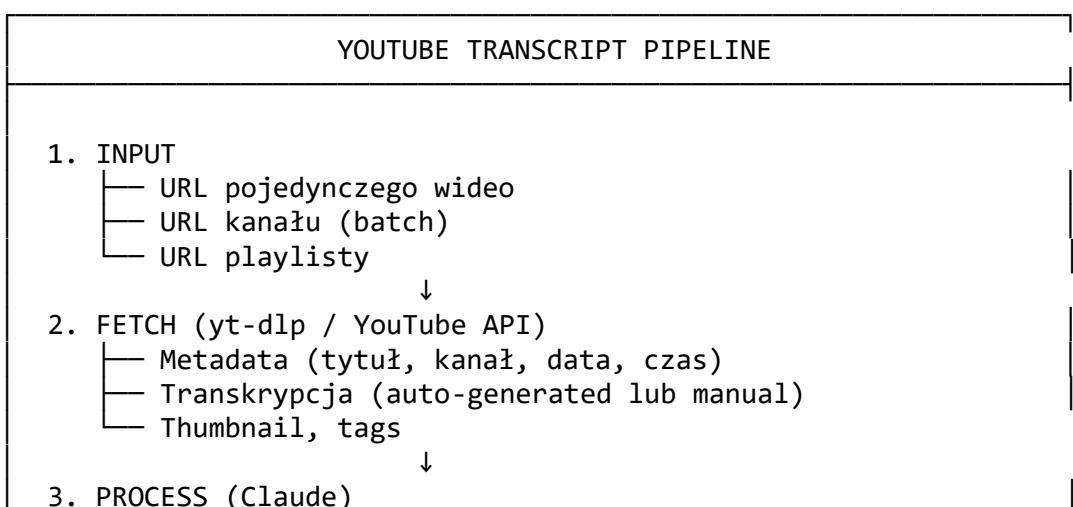
Priorytet	Źródło	Typ	Base Confidence
●	platform.claude.com	Zewnętrzne	0.95
●	docs.n8n.io	Zewnętrzne	0.90
●	modelcontextprotocol.io	Zewnętrzne	0.92
■	YouTube (AI, biznes)	Budowane	0.65
■	GitHub issues	Zewnętrzne	0.80
■■	Tech blogs	Zewnętrzne	0.75

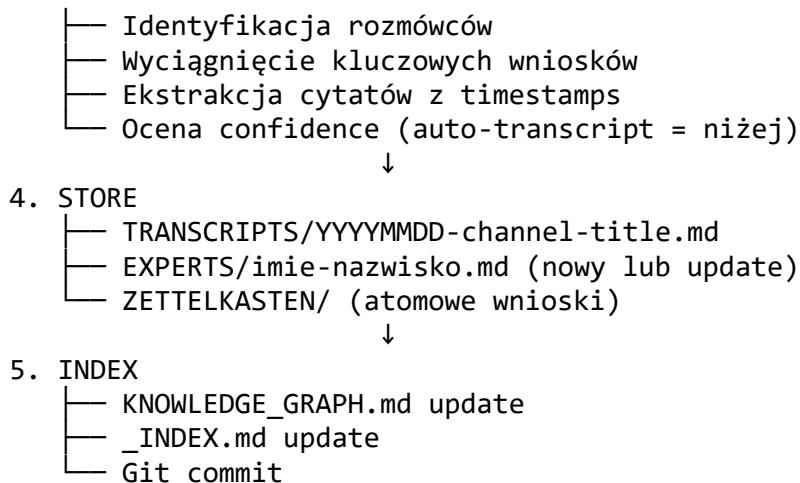
## Część 3: Proces budowy bazy transkrypcji YT

### Cel

Wyciąganie wiedzy z wywiadów i tutoriali na YouTube: - **Wnioski merytoryczne** — co ekspert powiedział o temacie - **Profile ekspertów** — kim jest, jakie ma poglądy, gdzie się specjalizuje - **Cytaty** — dosłowne wypowiedzi z timestampami

### Pipeline transkrypcji





## Struktura katalogów

```

docs/central/TRANSCRIPTS/
└── _INDEX.md                      # Spis transkrypcji
└── 20260115-lex-fridman-sam-altman.md # Pełna transkrypcja
└── 20260110-andrej-karpathy-llm.md
EXPERTS/
    └── _INDEX.md                      # Spis ekspertów
    └── sam-altman.md                  # Profil eksperta
    └── andrej-karpathy.md
    └── lex-fridman.md

```

## Format transkrypcji

```

# [Tytuł wideo]

**source:** https://youtube.com/watch?v=XXXXXX
**channel:** [Nazwa kanału]
**date:** YYYY-MM-DD
**duration:** HH:MM:SS
**confidence:** 0.65
**transcript_type:** auto-generated | manual
---
```

## ## Rozmówcy

Osoba	Rola	Profil
Sam Altman	Guest	[→ EXPERTS/sam-altman.md]
Lex Fridman	Host	[→ EXPERTS/lex-fridman.md]

## ## Kluczowe wnioski

```

### 1. [Tytuł wniosku]
**timestamp:** 05:23
**speaker:** Sam Altman

```

\*\*confidence:\*\* 0.70

### [Opis wniosku – 2-3 zdania]

### 2. [Tytuł wniosku]

\*\*timestamp:\*\* 12:45

\*\*speaker:\*\* Sam Altman

### [Opis wniosku]

---

## ## Cytaty

> "AGI is closer than people think, but the definition keeps moving."  
-> Sam Altman, 08:15

> "The key to AI safety is not stopping progress, but steering it."  
-> Sam Altman, 23:40

---

## ## Pełna transkrypcja

<**details**>  
<**summary**>Rozwiń pełną transkrypcję</**summary**>

[00:00] Lex: Welcome to the podcast...

[00:15] Sam: Thank you for having me...

[...]

</**details**>

---

## ## Powiązania

- → 202601-XXX: [wniosek wyciągnięty do ZETTELKASTEN]
- → EXPERTS/sam-altman.md

## Format profilu eksperta

# Sam Altman

\*\*role:\*\* CEO OpenAI  
\*\*expertise:\*\* AI, Startups, Venture Capital, AGI  
\*\*confidence:\*\* 0.75  
\*\*last\_updated:\*\* 2026-01-15

---

## ## Bio

Sam Altman (ur. 1985) – amerykański przedsiębiorca, CEO OpenAI od 2019.

Wcześniej prezes Y Combinator (2014-2019). Współzałożyciel Loopt.

---

## ## Kluczowe poglądy

### ### Na temat AGI

- AGI jest bliżej niż ludzie myślą (confidence: 0.70)
- Definicja AGI ciągle się zmienia
- → 202601-XXX

### ### Na temat AI Safety

- Nie zatrzymywać postępu, ale kierować go
- Governance > moratorium
- → 202601-YYY

---

## ## Cytaty

> "AGI is closer than people think, but the definition keeps moving."  
-> Lex Fridman Podcast, 2026-01-15

> "Move fast and be responsible."  
-> TED Talk, 2025

---

## ## Źródła

Data	Źródło	Link
2026-01-15	Lex Fridman Podcast	[→ TRANSCRIPTS/20260115-lex-fridman-sam-al tman.md]
2025-11-20	TED Talk	[→ external URL]

---

## ## Powiązania

- ↔ Ilya Sutskever (współpraca OpenAI)
- ↔ Elon Musk (współzałożyciel OpenAI, konflikt)
- → OpenAI (organizacja)
- → Y Combinator (poprzednia rola)

## Confidence dla transkrypcji

Typ transkrypcji	Base Confidence	Uzasadnienie
Manual (oficjalne)	0.80	Zweryfikowane przez człowieka
Auto-generated (YouTube)	0.65	Mogą być błędy
Auto + korekta Claude	0.70	Częściowo poprawione
Wywiad > tutorial	+0.05	Bezpośrednie słowa eksperta

Typ transkrypcji	Base Confidence	Uzasadnienie
Stare wideo (>2 lata)	-0.10	Informacje mogą być nieaktualne

## Workflow: Weekly Review

### Co tydzień

1. **Przegląd nowych zettels**
  - Czy confidence jest prawidłowe?
  - Czy są wszystkie powiązania?
2. **Deprecjacja nieaktualnych**
  - API się zmieniło?
  - Wniosek okazał się błędny?
3. **Uzupełnianie powiązań**
  - Nowe notatki → linki do starych
  - Klastry → czy są kompletne?
4. **Matryca źródeł**
  - Update avg confidence
  - Które źródła były używane?

### Checklist

```
## Weekly Review - [DATA]

### Nowe zettels (ostatni tydzień)
- [ ] 202601-XXX: [tytuł] – confidence OK?
- [ ] 202601-YYY: [tytuł] – powiązania?

### Deprecjacja
- [ ] Sprawdzić zettels >30 dni – czy aktualne?
- [ ] Oznaczyć deprecated jeśli nie

### Powiązania
- [ ] Nowe → stare połączenia
- [ ] Bidirectional links

### Źródła
- [ ] Update KNOWLEDGE_GRAPH matryca
```

---

## Narzędzia do implementacji

### Opcja 1: MCP Skill /zettel

```
/zettel-save "Tytuł wniosku" \
--confidence 0.85 \
--source "https://..." \
--tags "@claude-api,@tool-use" \
--relates-to "202601-001,202601-003"
```

Automatycznie: - Generuje ID (YYYYMM-NNN) - Tworzy plik w ZETTELKASTEN/ - Update \_INDEX.md - Update KNOWLEDGE\_GRAPH.md - Git commit

### Opcja 2: MCP Skill /transcript

```
/transcript "https://youtube.com/watch?v=XXXXXX"
```

Automatycznie: - Fetch via yt-dlp - Process via Claude - Store w TRANSCRIPTS/ - Create/update EXPERTS/ - Extract zettels do ZETTELKASTEN/

### Opcja 3: n8n Workflow

```
Trigger: Webhook z URL  
↓  
yt-dlp: Fetch metadata + transcript  
↓  
Claude: Process + extract  
↓  
GitHub: Commit files  
↓  
Notify: Telegram "Nowa transkrypcja dodana"
```

---

## Pytania otwarte

1. **Versjonowanie** — czy każdy commit = 1 zettel, czy batch'ować?
  2. **Search** — full-text search czy wystarczy grep + \_INDEX.md?
  3. **Powiązania** — automatyczne bidirectional czy manualne?
  4. **Eksperci** — jak rozwiązać konflikty poglądów między ekspertami?
  5. **Stare transkrypcje** — jak oznaczać że informacje mogą być nieaktualne?
  6. **Confidence calibration** — jak weryfikować czy nasze wagи są prawidłowe?
- 

## Referencje

### Metodologie

- **Zettelkasten Method** — oficjalna strona metody
- **PARA Method** — Tiago Forte
- **ADR (Architecture Decision Records)**
- **Building a Second Brain**

### Narzędzia

- **yt-dlp** — pobieranie z YouTube
- **Obsidian** — edytor markdown z linkami
- **Logseq** — alternatywa dla Obsidian

### Teoria

- **Knowledge Graphs for AI** — uncertainty-aware KG

- Atomic Note-Taking Guide
- 

## Podsumowanie

Element	Opis	Lokalizacja
<b>ZETTELKASTEN</b>	Atomowe notatki z wagami	docs/central/ZETTELKASTEN/
<b>KNOWLEDGE_GRAPH</b>	Mapa powiązań	docs/central/KNOWLEDGE_GRAPH.md
<b>TRANSCRIPTS</b>	Transkrypcje YT	docs/central/TRANSCRIPTS/
<b>EXPERTS</b>	Profile ekspertów	docs/central/TRANSCRIPTS/EXPERTS/

**Kluczowa wartość:** - Wiedza nie ginie — każdy fakt ma swoje miejsce - Błędy nie zniekształcają — system wag + deprecjacja - Powiązania widoczne — Knowledge Graph - Źródła śledzalne — każda notatka wie skąd pochodzi

---

**Status dokumentu:** Do analizy i przemyślenia **Następny krok:** Decyzja o implementacji (MCP Skill vs n8n Workflow) **Powiązane:** [ROADMAP.md](#) Faza 3 (MÓZG)