

# OPIS PRZEDMIOTU ZAMÓWIENIA

Narzędzie wspierające proces oceny formalnej odwołań do Krajowej Izby Odwoławczej (KIO)

## SPIS TREŚCI

1. Definicje i skróty
2. Cel i zakres zamówienia
3. Architektura rozwiązania i środowisko docelowe
4. Wymagania funkcjonalne
5. Integracja z biblioteką do klasyfikacji i ekstrakcji metadanych
6. Integracje z systemami zewnętrznymi
7. Integracja z SharePoint Online
8. Weryfikacja podpisów elektronicznych (DSS)
9. Ocena formalna odwołania
10. Generowanie Karty Odwołania
11. Obsługa braków formalnych — wezwanie do uzupełnienia
12. Wymagania dotyczące testowania i odbioru
13. Produkty i rezultaty zamówienia (Deliverables)
14. Hardening środowiska produkcyjnego
15. Warunki realizacji zamówienia
16. Umowa powierzenia przetwarzania danych osobowych
17. Kryteria akceptacji i mierniki sukcesu
18. Załączniki

## 1. Definicje i skróty

Termin	Definicja
<b>KIO</b>	Krajowa Izba Odwoławcza — organ rozstrzygający odwołania w zamówieniach publicznych
<b>PZP</b>	Ustawa z dnia 11 września 2019 r. — Prawo zamówień publicznych (t.j. Dz. U. z 2024 r. poz. 1320 z późn. zm.)
<b>Karta Odwołania</b>	Dokument wyjściowy systemu agregujący dane o sprawie odwoławczej, weryfikujący wymogi formalne art. 516–518 PZP
<b>Biblioteka do klasyfikacji i ekstrakcji metadanych</b>	System analizy dokumentów (PDF → dane strukturalne JSON), zainstalowany i udostępniający

Termin	Definicja
<b>n8n</b>	API REST na infrastrukturze Zamawiającego Platforma orkiestracji workflow (open-source), stanowiąca silnik automatyzacji procesów
<b>n8n node</b>	Pojedynczy komponent integracyjny w ekosystemie n8n, realizujący określone zadanie (np. wywołanie API, transformacja danych)
<b>DSS</b>	Digital Signature Service — otwarte oprogramowanie Komisji Europejskiej do weryfikacji podpisów elektronicznych
<b>docling-serve</b>	Serwer konwersji dokumentów PDF na tekst, działający w kontenerze Docker z silnikiem OCR Tesseract
<b>LiteLLM</b>	Biblioteka proxy umożliwiająca routing i testowanie różnych dostawców AI przez jednolity interfejs
<b>n8n-mcp</b>	Serwer MCP (Model Context Protocol) wspierający tworzenie i modyfikację workflow n8n z użyciem AI
<b>SIWZ</b>	Specyfikacja Istotnych Warunków Zamówienia
<b>OPZ</b>	Opis Przedmiotu Zamówienia (niniejszy dokument)
<b>KRS</b>	Krajowy Rejestr Sądowy
<b>CEIDG</b>	Centralna Ewidencja i Informacja o Działalności Gospodarczej
<b>REGON</b>	Rejestr Gospodarki Narodowej (GUS)
<b>BZP</b>	Biuletyn Zamówień Publicznych (platforma eZamówienia)
<b>SharePoint Online</b>	Usługa Microsoft 365 służąca jako warstwa storage i interfejs użytkownika
<b>PAdES / XAdES / CAdES</b>	Formaty kwalifikowanych podpisów elektronicznych zgodne z eIDAS
<b>ground_truth.yaml</b>	Plik referencyjny zawierający oczekiwane dane ekstrakcji dla danego zbioru dokumentów
<b>Zamawiający</b>	Urząd Zamówień Publicznych (UZP)
<b>Wykonawca</b>	Podmiot realizujący zamówienie

## 2. Cel i zakres zamówienia

### 2.1. Cel zamówienia

Przedmiotem zamówienia jest zaprojektowanie, implementacja i dostarczenie narzędzia wspierającego proces oceny formalnej odwołań wnoszonych do Krajowej Izby Odwoławczej (KIO), zgodnie z wymogami art. 516–518 ustawy Prawo zamówień publicznych. Narzędzie wykorzystuje platformę n8n zintegrowaną z istniejącą biblioteką do klasyfikacji i ekstrakcji metadanych w celu automatyzacji tego procesu.

### 2.2. Zakres zamówienia

Zamówienie obejmuje:

1. **Dostarczenie workflow oceny formalnej dokumentacji odwoławczej** — automatyczna weryfikacja wymogów art. 516–518 PZP (12 kryteriów F1–F12), generowanie wyniku oceny z uzasadnieniem i odwołaniem do źródeł (szczegóły w rozdziale 9).
2. **Dostarczenie mechanizmu generowania Karty Odwołania** — dokument podsumowujący stan formalny sprawy odwoławczej w formatach DOCX i JSON, z systemem cytowania źródeł (szczegóły w rozdziale 10).
3. **Dostarczenie mechanizmu obsługi braków formalnych** — automatyczne generowanie wezwania do uzupełnienia braków i dostarczenie go wskazanymi kanałami (SharePoint, e-mail) (szczegóły w rozdziale 11).
4. **Dostarczenie narzędzi testowych i ewaluacyjnych** — skrypty do generowania ground truth, uruchamiania testów akceptacyjnych i raportowania accuracy (szczegóły w rozdziale 12).
5. **Dostarczenie kodów źródłowych i workflow** — komplet n8n nodes (dedykowane komponenty integracyjne n8n) realizujących pełny proces przetwarzania odwołań, wraz z workflow JSON i kodem pomocniczym.
6. **Dostarczenie konfiguracji środowiska uruchomieniowego** — pełna konfiguracja docker-compose obejmująca n8n (w konfiguracji skalowalnej z Redis i PostgreSQL), DSS, docling-serve, serwer MCP (n8n-mcp), LiteLLM oraz wymagane zależności.
7. **Dokumentacja techniczna i wdrożeniowa.**
8. **Zawarcie umowy powierzenia przetwarzania danych osobowych.**
9. **Hardening środowiska produkcyjnego** — podłączenie do domeny, konfiguracja kontroli dostępu, zabezpieczenie portów (szczegóły w rozdziale 14).

### 2.3. Założenia

1. Biblioteka do klasyfikacji i ekstrakcji metadanych jest zainstalowana na infrastrukturze Zamawiającego i udostępnia swoje metody przez API REST (endpoint POST /v1/extract), zgodnie ze specyfikacją w Załączniku A.

- Zamawiający dostarczy **20 zbiorów dokumentów** składających się na odwołanie oraz odpowiadające im **karty odwołań** (w formacie DOC lub DOCX) jako materiał referencyjny do testów akceptacyjnych.
- Zamawiający posiada klucze API do systemów zewnętrznych (KRS, CEIDG, REGON, eZamówienia/BZP).
- Zamawiający dysponuje licencją Microsoft 365 z dostępem do SharePoint Online.
- Wykonawca jest ekspertem w zakresie platformy n8n i ma doświadczenie w budowie integracji workflow.

## 2.4. Wyłączenia z zakresu

- Instalacja, konfiguracja i utrzymanie biblioteki do klasyfikacji i ekstrakcji metadanych — system jest już wdrożony.
- Dostarczenie lub konfiguracja licencji Microsoft 365 / SharePoint Online.

## 3. Architektura rozwiązania i środowisko docelowe

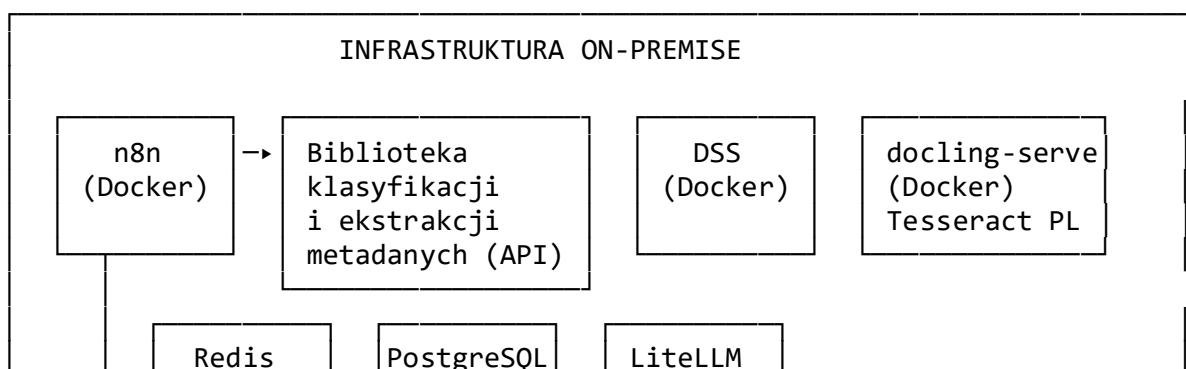
### 3.1. Środowisko infrastrukturalne

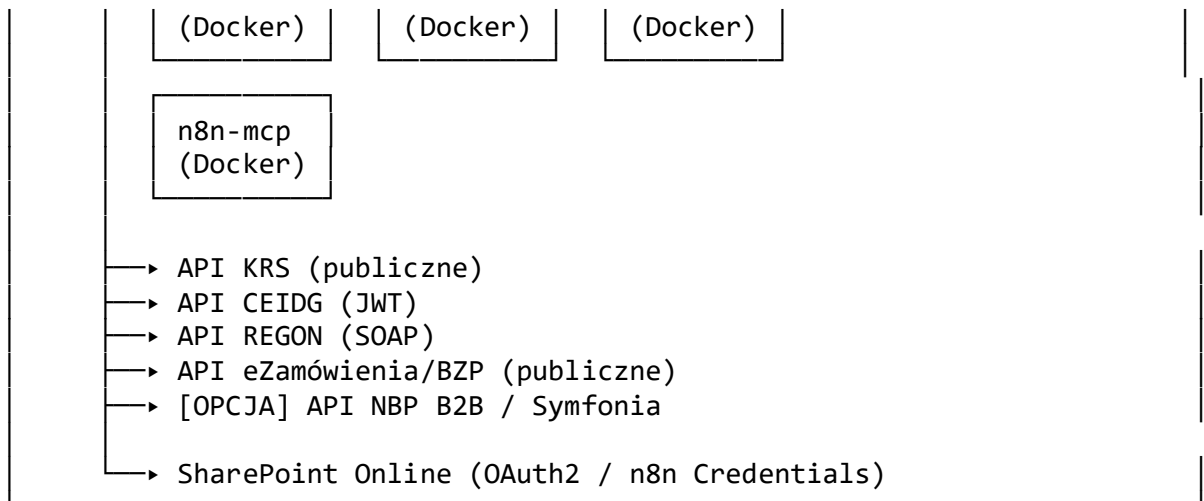
Parametr	Wartość
<b>Typ wdrożenia</b>	On-premise
<b>System operacyjny</b>	Ubuntu Server (wersja LTS)
<b>Konteneryzacja</b>	Docker / Docker Compose
<b>Parametry maszyn</b>	Do ustalenia w trakcie realizacji projektu — Wykonawca przedstawi rekomendacje minimalne i zalecane

Wykonawca dostarczy rekomendacje dotyczące minimalnych i zalecanych parametrów sprzętowych w ramach dokumentacji wdrożeniowej (Deliverable D11).

### 3.2. Architektura logiczna

Rozwiązanie opiera się na architekturze hybrydowej łączącej przetwarzanie on-premise z usługami Microsoft 365:





### 3.3. Komponenty Docker

Wykonawca musi dostarczyć konfigurację `docker-compose.yml` obejmującą następujące kontenery, połączone wewnętrzną siecią Docker i umożliwiające rozproszenie na różne maszyny wirtualne:

Kontener	Obraz / Technologia	Cel	Sieć
<b>n8n</b>	n8nio/n8n (wersja stabilna)	Orkiestracja workflow w, w konfiguracji akcji skalowalnej (queue mode) zgodnie z dokumentacją producenta	Wewnętrzna sieć Docker
<b>Redis</b>	redis:7 (lub nowszy LTS)	Broker kolejki n8n (wymagany dla trybu skalowalnego)	Wewnętrzna sieć Docker

Kontener	Obraz / Technologia	Cel	Sieć
<b>PostgreSQL</b>	postgres:16 (lub nowszy LTS)	nego — queue mode) Baza danych n8n (trwałe przechowywanie workflow, credentials, execution history)	Wewnętrzna sieć Docker
<b>DSS</b>	cefdigital/dss-standalone-app:5.13 (lub nowszy)	Weryfikacja podpisów elektronicznych	Wewnętrzna sieć Docker
<b>docling-serve</b>	ds4sd/docling-serve (lub równoważny)	Konwersja dokumentów PDF na formę tekstową, konfiguracja z Tessera ct OCR dla języka polskiego	Wewnętrzna sieć Docker
<b>n8n-mcp</b>	ghcr.io/czlonkowski/n8n-mcp (lub build z <a href="https://github.com/czlonkowski/n8n-mcp">https://github.com/czlonkowski/n8n-mcp</a> )	Serwer MCP wspierający	Wewnętrzna sieć Docker

Kontener	Obraz / Technologia	Cel	Sieć
		tworzenie i modyfikację workflow w n8n z użyciem AI	
<b>LiteLLM</b>	ghcr.io/berriai/litellm (lub równoważny)	Proxy umożliwiające routing i testowanie różnych dostawców AI przez jednolity interfejs	Wewnętrzna sieć Docker

Konfiguracja musi umożliwiać:

1. Uruchomienie wszystkich kontenerów na jednej maszynie wirtualnej (docker compose up).
2. Rozproszenie kontenerów na różne maszyny wirtualne (np. n8n + Redis + PostgreSQL na jednej, DSS + docling-serve na drugiej) — poprzez konfigurację zmiennych środowiskowych i adresów sieciowych.
3. Skalowanie n8n w trybie queue mode (wiele workerów) zgodnie z oficjalną dokumentacją producenta.

### 3.4. Konfiguracja docling-serve

Kontener docling-serve musi być skonfigurowany do obsługi języka polskiego:

Parametr	Wartość
<b>Silnik OCR</b>	Tesseract
<b>Język OCR</b>	Polski (po1)
<b>Endpoint</b>	Dostępny z poziomu workflow n8n (wewnętrzna sieć Docker)
<b>Zastosowanie</b>	Konwersja dokumentów PDF na formę tekstową bezpośrednio z workflow n8n

Workflow n8n musi wykorzystywać docling-serve do konwersji dokumentów PDF na tekst.

### 3.5. Struktura pluginów (n8n nodes)

Każdy plugin integracyjny musi posiadać następującą strukturę katalogową:

```
integrations_n8n/  
├── [nazwa_pluginu]/  
│   ├── workflow.json           # Eksport workflow z n8n  
│   └── manifest.yaml           # Metadane: wersja, opis, kontrakt I/O, zmie  
nne  
├── README.md                   # Dokumentacja dla dewelopera  
└── tests/  
    ├── input_happy_path.json  
    ├── input_error.json  
    └── output_expected.json
```

---

## 4. Wymagania funkcjonalne

### 4.1. Przeptyw główny (End-to-End)

System musi realizować następujący przeptyw przetwarzania, składający się z 10 faz:

Faza	Nazwa	Opis
1	<b>Monitoring i detekcja</b>	Automatyczne wykrywanie nowych plików ZIP w katalogu SharePoint Do obsłużenia/
2	<b>Rekurencyjne rozpakowywanie</b>	Ekstrakcja plików z archiwów ZIP (obsługa zagnieżdżonych archiwów)
3	<b>Konwersja PDF na tekst</b>	Wywołanie docling-serve (Tesseract, język polski) w celu konwersji dokumentów PDF na formę tekstową
4	<b>Klasyfikacja plików</b>	Kategoryzacja dokumentów do 8 kategorii (patrz 4.2) z <b>wykorzystaniem biblioteki do klasyfikacji i ekstrakcji metadanych</b> — zadaniem Wykonawcy jest integracja z biblioteką

Faza	Nazwa	Opis
5	<b>Ekstrakcja metadanych</b>	poprzez jej API Wywołanie API biblioteki do klasyfikacji i ekstrakcji metadanych dla dokumentów wymagających analizy
6	<b>Weryfikacja podpisów</b>	Walidacja podpisów elektronicznych przez DSS
7	<b>Pobieranie danych z rejestrów</b>	Wzbogacanie danych z KRS, CEIDG, REGON, eZamówienia; opcjonalnie: weryfikacja wpisu przez NBP B2B lub Symfonię
8	<b>Ocena formalna odwołania</b>	Automatyczna weryfikacja spełnienia wymogów formalnych art. 516–518 PZP (szczegóły w rozdziale 9, kryteria w Załączniku G)
9	<b>Generowanie Karty Odwołania / Wezwania do uzupełnienia</b>	Agregacja danych i generowanie DOCX + JSON; w przypadku stwierdzenia braków — generowanie wezwania do uzupełnienia (rozdział 11)
10	<b>Finalizacja</b>	Przeniesienie do Przetworzone/ lub Błąd/ w SharePoint

## 4.2. Kategorie dokumentów

System musi klasyfikować dokumenty do następujących kategorii:

Kategoria	Katalog docelowy	Opis
Odwołanie	Odwołanie/	Główny dokument odwołania
Przystąpienie	Przystąpienie/	Dokumenty przystąpienia do postępowania
Dokumenty rejestrowe	Dokumenty_rejestrowe/	Odpisy KRS,

Kategoria	Katalog docelowy	Opis
		wydruki CEIDG
Pełnomocnictwo	Pełnomocnictwo/	Pełnomocnictwa procesowe
Wpis	Wpis/	Dowody uiszczenia wpisu
Potwierdzenia przelewów	Potwierdzenia_przelewow/	Potwierdzenia bankowe
Weryfikacja podpisów	Weryfikacja_podpisow/	Raporty weryfikacji podpisów
Inne	Inne/	Dokumenty niesklasyfikowane



### 4.3. Struktura katalogowa sprawy

Po przetworzeniu, katalog sprawy w SharePoint musi mieć następującą strukturę:

```
Przetworzone/[case_id]/
├── Odwołanie/
│   ├── odwołanie.pdf
│   └── odwołanie_metadata.json
├── Dokumenty_rejestrowe/
│   ├── krs_XXXXXXXXXX.pdf
│   └── krs_XXXXXXXXXX_api.json
├── Pełnomocnictwo/
│   └── pelnomocnictwo.pdf
├── Wpis/
│   └── wpis.pdf
├── Potwierdzenia_przelewow/
│   └── przelew.pdf
├── Weryfikacja_podpisow/
│   ├── odwołanie_verification.pdf
│   └── pelnomocnictwo_verification.pdf
├── Karta_Odwołania_[case_id].docx
└── Karta_Odwołania_[case_id].json
```

### 4.4. Obsługa błędów

System musi implementować następujący model obsługi błędów:

1. **Błąd krytyczny** (niemożliwość przetworzenia): przeniesienie pliku ZIP do katalogu Błąd/ z raportem:
  - o `error_report.json` — strukturalny opis błędu (kod, faza, timestamp)
  - o `error_description.txt` — opis tekstowy dla użytkownika
2. **Błąd niekrytyczny** (częściowe przetworzenie): kontynuacja procesu z oznaczeniem brakujących danych statusem  lub  w Karcie Odwołania.
3. **Retry logic**: automatyczne ponowienie dla błędów przejściowych (timeout API, HTTP 5xx) z wykładniczym backoff (minimum 3 próby).

## 4.5. Optymalizacja pobierania danych z rejestrów

System musi implementować strategię „Smart Registry Lookup”:

1. Jeżeli dokument odwołania zawiera identyfikatory (NIP, KRS, REGON) — bezpośrednio odpytanie API rejestrów na podstawie tych identyfikatorów.
2. Odpytanie API z dokumentów rejestrowych tylko gdy identyfikatory nie zostały znalezione w odwołaniu.
3. Wszystkie odpowiedzi API zapisywane jako pliki `*_api.json` w katalogu `Dokumenty_rejestrowe/`.

## 5. Integracja z biblioteką do klasyfikacji i ekstrakcji metadanych

### 5.1. Specyfikacja API

Biblioteka do klasyfikacji i ekstrakcji metadanych udostępnia API REST zgodnie ze specyfikacją Swagger (udostępnianą przez Zamawiającego). Kluczowe elementy:

Parametr	Wartość
<b>Endpoint główny</b>	POST <code>/v1/extract</code>
<b>Autoryzacja</b>	Nagłówek <code>X-API-Key</code>
<b>Content-Type</b>	<code>multipart/form-data</code>
<b>Parametry</b>	<code>file</code> (wymagany), <code>stage2_method</code> (opcjonalny), <code>locale_hint</code> (opcjonalny, domyślnie <code>pl-PL</code> ), <code>return_mode</code> (opcjonalny: <code>auto/sync/async</code> )

### 5.2. Tryby odpowiedzi

API biblioteki do klasyfikacji i ekstrakcji metadanych obsługuje tryb hybrydowy:

Tryb	Kod HTTP	Opis
<b>Synchroniczny</b>	<code>200 OK</code>	Wynik zwracany bezpośrednio

Tryb	Kod HTTP	Opis
<b>Asynchroniczny</b>	202 Accepted	(przetwarzanie < 800s) Job utworzony, wynik dostępny przez polling

### 5.3. Endpointy polling (tryb asynchroniczny)

Endpoint	Metoda	Opis
/v1/jobs/{job_id}	GET	Status joba (queued/running/succeeded/failed/canceled/expired)
/v1/jobs/{job_id}/result	GET	Pobranie wyniku (dostępny jednokrotnie — delete-on-fetch)
/v1/jobs/{job_id}	DELETE	Anulowanie i czyszczenie joba

### 5.4. Wymagania integracyjne

1. Workflow n8n musi obsługiwać oba tryby odpowiedzi (200 i 202).
2. Dla trybu 202: implementacja polling z respektowaniem nagłówka Retry-After.
3. TTL joba: 24 godziny — workflow musi pobrać wynik przed upływem TTL.
4. Obsługa kodów błędów: 401 (brak autoryzacji), 409 (wynik niegotowy), 404 (job nieznalesiony), 410 (job wygasł).

### 5.5. Kontrakt danych wyjściowych

Wynik ekstrakcji z biblioteki do klasyfikacji i ekstrakcji metadanych zawiera strukturę zgodną z kontraktem FinalResult v0:

- schema\_version — wersja kontraktu
- run\_id — identyfikator uruchomienia
- files[].classification — kategoria dokumentu z oceną pewności
- files[].extraction.objects — wyekstrahowane dane pogrupowane wg ról (np. Odwołujący, Zamawiający, OdwołanieKIO)
- Każde pole zawiera data (wartość), quality (pewność) i evidence (odniesienie do źródła)

Szczegółowy opis kontraktu stanowi Załącznik A.

---

## 6. Integracje z systemami zewnętrznymi

### 6.1. Wykaz integracji

System	Typ API	Autoryzacja	Cel
<b>KRS</b> (api-krs.ms.gov.pl)	REST/JSON	Publiczne (brak)	Dane podmiotów, zasady reprezentacji, skład organów
<b>CEIDG</b> (dane.biznes.gov.pl)	REST/JSON	Token JWT	Dane przedsiębiorców (osoby fizyczne)
<b>REGON</b> (wyszukiwarkaregon.stat.gov.pl)	SOAP/XML	Klucz API	Dane statystyczne podmiotów
<b>eZamówienia/BZP</b> (ezamowienia.gov.pl)	REST/JSON	Publiczne (brak)	Dane postępowań, daty publikacji, dane zamawiającego
<b>[OPCJA A] NBP B2B</b> (nbp.pl)	REST/XML	Certyfikat / klucz API	Weryfikacja wpisu — zapytanie o wyciąg z rachunku bankowego
<b>[OPCJA B] Symfonia</b>	API (do ustalenia)	Klucze API	Weryfikacja wpisu — sprawdzenie zaksięgowania przelewu

Zamawiający dostarczy klucze API do systemów KRS, CEIDG, REGON i eZamówienia/BZP.

**Uwaga dotycząca weryfikacji wpisu (OPCJA A / OPCJA B):** Integracja z systemem weryfikacji wpisu (NBP B2B lub Symfonia) stanowi **element opcjonalny**, podlegający **odrębnej wycenie**. Zamawiający zastrzega sobie prawo do nieskorzystania z tej opcji. Wykonawca musi wycenić tę opcję osobno w ofercie. Wybór wariantu (NBP B2B lub Symfonia) zostanie dokonany przez Zamawiającego.

## 6.2. Wymagania dla integracji KRS

1. **Endpoint:** GET /api/krs/OdpisAktualny/{krs}?rejestr={rejestr}&format=json
2. **Fallback:** w przypadku HTTP 404 dla rejestr=P (Przedsiębiorcy), ponowienie z rejestr=S (Stowarzyszenia).
3. **Mapowanie na model kanoniczny:** nazwa, adres, NIP, organ reprezentacji, sposób reprezentacji, skład organu (imiona, nazwiska, funkcje).
4. **Caching:** wyniki dla danego numeru KRS POWINNY być cache'owane na min. 24h w ramach sprawy.
5. Implementacja jako sub-workflow: func\_get\_krs\_data.

## 6.3. Wymagania dla integracji CEIDG

1. **Endpoint:** GET /api/ceidg/v3/firmy?nip={nip}
2. **Autoryzacja:** Authorization: Bearer <JWT\_TOKEN> — token zarządzany przez n8n Credentials.
3. **Rate limiting:** max 50 żądań / 3 min (≈1 req / 3,6s). Workflow musi implementować throttling (Wait node ≥ 4s).
4. **Mapowanie:** nazwa firmy, właściciel (imię, nazwisko), NIP, adres, status aktywności.
5. Implementacja jako sub-workflow: func\_get\_ceidg\_data.

## 6.4. Wymagania dla integracji REGON

1. **Protokół:** SOAP/XML.
2. **Autoryzacja:** klucz API w nagłówku żądania SOAP.
3. **Mapowanie:** numer REGON, forma prawna, PKD, adres siedziby.
4. Implementacja jako sub-workflow: func\_get\_regon\_data.

## 6.5. Wymagania dla integracji eZamówienia/BZP

1. **Endpoint:** GET /mo-board/api/v1/notice
2. **Parametry wymagane:** NoticeNumber, NoticeType, PublicationDateFrom, PublicationDateTo.
3. **Strategia wyszukiwania:** iteracja po typach (ContractNotice → TenderResultNotice → ContractPerformingNotice) z dynamicznym zakresem dat (90 dni wstecz).
4. **Normalizacja numerów:** obsługa różnych formatów zapisu numeru BZP (ze spacjami, bez, z rokiem).

5. **Mapowanie:** numer ogłoszenia, data publikacji, nazwa postępowania, dane zamawiającego (nazwa, NIP), termin składania ofert.
6. Implementacja jako sub-workflow: `func_get_procurement_data`.

## 6.6. [OPCJA] Weryfikacja wpisu od odwołania

Integracja z systemem weryfikacji wpisu stanowi **element opcjonalny realizowany w ramach prawa opcji**, podlegający **odrębnej wycenie** przez Wykonawcę. Zamawiający zastrzega sobie prawo do nieskorzystania z tej opcji lub do wyboru jednego z dwóch wariantów.

### 6.6.1. OPCJA A: Integracja z NBP B2B

1. **Cel:** weryfikacja uiszczenia wpisu od odwołania (czy deklarowany przelew został faktycznie zaksięgowany na rachunku bankowym).
2. **Dokumentacja API:** <https://nbp.pl/interfejs-b2b/>
3. **Zakres integracji:** „zapytanie o wyciąg, w tym w ramach zapytania o wyciąg — zapytanie o listę operacji bieżących na rachunku”, pozwalające na sprawdzenie, czy deklarowany przelew za wpis został faktycznie zaksięgowany.
4. **Mapowanie:** kwota przelewu, data zaksięgowania, dane nadawcy, tytuł przelewu.
5. Implementacja jako sub-workflow: `func_verify_payment_nbp`.

### 6.6.2. OPCJA B: Integracja z systemem Symfonia

1. **Cel:** weryfikacja uiszczenia wpisu od odwołania — analogiczny cel jak w OPCJI A, realizowany przez integrację z systemem finansowo-księgowym Symfonia.
2. **Specyfikacja API:** do uzgodnienia z Zamawiającym w fazie analizy (Zamawiający dostarczyć dokumentację API Symfonii).
3. **Mapowanie:** kwota przelewu, data zaksięgowania, dane nadawcy, tytuł przelewu.
4. Implementacja jako sub-workflow: `func_verify_payment_symfonia`.

## 6.7. Wymagania wspólne dla integracji

1. Wszystkie wrażliwe dane dostępne (klucze API, tokeny) **MUSZĄ** być przechowywane wyłącznie w **n8n Credentials Store** — nigdy w kodzie workflow ani zmiennych środowiskowych.
2. Każda integracja musi zwracać dane w modelu kanonicznym (Canonical Model) — niezależnym od formatu API źródłowego.
3. Każda integracja musi implementować obsługę błędów: timeout, HTTP 4xx/5xx, brak danych.
4. Odpowiedzi API **MUSZĄ** być zapisywane jako pliki `*_api.json` w strukturze katalogowej sprawy.

## 7. Integracja z SharePoint Online

### 7.1. Mechanizm autoryzacji

Integracja z SharePoint Online musi wykorzystywać mechanizm **n8n Credentials** (wbudowany connector SharePoint / Microsoft Graph API z konfiguracją OAuth2 w n8n Credentials Store).

Zamawiający wymaga wykorzystania tego mechanizmu. Wykonawca NIE MOŻE stosować bezpośredniego zarządzania tokenami OAuth2 poza n8n Credentials Store.

### 7.2. Struktura katalogów SharePoint

Katalog	Funkcja
Do obsłużenia/	Katalog wejściowy — monitorowany przez n8n (trigger)
W trakcie/	Sprawy w trakcie przetwarzania
Przetworzone/	Sprawy przetworzone pomyślnie
Błąd/	Sprawy z błędami przetwarzania
Temp/	Pliki tymczasowe (usuwane po przetworzeniu)

### 7.3. Wymagane operacje

1. **Monitoring** — wykrywanie nowych plików ZIP w Do obsłużenia/.
2. **Download** — pobieranie plików do przetwarzania.
3. **Upload** — zapisywanie wyników (Karta Odwołania DOCX/JSON, raporty weryfikacji, pliki \*\_api.json).
4. **Move** — przenoszenie plików między katalogami (np. z Do obsłużenia/ do W trakcie/, a następnie do Przetworzone/ lub Błąd/).
5. **Delete** — usuwanie pliku źródłowego ZIP po pomyślnym przetworzeniu.

## 8. Weryfikacja podpisów elektronicznych (DSS)

### 8.1. Rozwiązanie techniczne

System musi wykorzystywać **Digital Signature Service (DSS)** — oprogramowanie open-source Komisji Europejskiej — jako silnik weryfikacji podpisów elektronicznych.

### 8.2. Wymagania funkcjonalne

1. **Obsługiwane formaty podpisów:** PAdES, XAdES, CAdES, ASiC.
2. **Walidacja zgodna z eIDAS:** weryfikacja certyfikatów wobec europejskich Trusted Service Lists (TSL/LOTL).
3. **Automatyczne odświeżanie TSL:** cron co 24h.

4. **Polityki walidacji:** konfigurowalne (BASELINE-B, BASELINE-T, BASELINE-LT, BASELINE-LTA).

### 8.3. Ekstrakcja danych z podpisów

Dla każdego zweryfikowanego podpisu system musi wyekstrahować:

Pole	Opis
<code>is_valid</code>	Czy podpis jest ważny (true/false)
<code>signers[].name</code>	Imię i nazwisko z certyfikatu
<code>signers[].signing_time</code>	Data i czas złożenia podpisu (priorytet: <code>bestSignatureTime</code> z TSA, fallback: <code>claimedSigningTime</code> )
<code>signers[].is_qualified</code>	Czy podpis kwalifikowany (QES)
<code>signers[].valid_at_signing</code>	Czy certyfikat był ważny w momencie podpisu
<code>signature_format</code>	Format podpisu (np. PAdES-BASELINE-LT)

### 8.4. Architektura (Strategy Pattern)

Weryfikacja podpisów musi być zaimplementowana zgodnie z wzorcem Strategy Pattern:

1. **Router workflow** (`func_verify_signature`): Facade wybierający strategię na podstawie zmiennej `SIGNATURE_VERIFICATION_STRATEGY`.
2. **Adapter DSS** (`func_verify_signature_dss`): implementacja wywołania DSS REST API i transformacji odpowiedzi na model kanoniczny.
3. **Model kanoniczny:** jednolity format wyjściowy niezależny od dostawcy (patrz Załącznik B).

### 8.5. Wdrożenie DSS

Wykonawca musi dostarczyć konfigurację DSS jako kontener Docker w ramach wspólnej konfiguracji docker-compose (patrz rozdział 3.3):

Parametr	Wartość
<b>Obraz</b>	<code>cefdigital/dss-standalone-app:5.13</code> (lub nowszy stabilny)
<b>RAM</b>	min. 2 GB
<b>Storage</b>	min. 500 MB (cache TSL)
<b>Sieć Docker</b>	Wspólna sieć z n8n
<b>Health check</b>	Endpoint <code>/services/rest/server/info</code>

## 9. Ocena formalna odwołania

### 9.1. Cel

Przed wygenerowaniem Karty Odwołania system musi przeprowadzić automatyczną ocenę formalną odwołania, weryfikując spełnienie wymogów określonych w art. 516–518 ustawy Prawo zamówień publicznych. Zadaniem Wykonawcy jest stworzenie workflow n8n realizującego tę ocenę na podstawie danych zebranych w fazach 3–7.

### 9.2. Kryteria weryfikacji formalnej

Workflow oceny formalnej musi weryfikować następujące kryteria (pełna specyfikacja w Załączniku G):

ID	Kryterium	Podstawa prawna	Źródło danych
F1	Odwołanie wniesione w terminie	art. 515 PZP	Data podpisu (DSS) vs termin obliczony z daty czynności
F2	Wpis od odwołania uiszczony w prawidłowej kwocie	art. 516 PZP	Potwierdzenie przelewu / [OPCJA] NBP B2B / Symfonia
F3	Wpis wniesiony przed upływem terminu na wniesienie odwołania	art. 516 PZP	Data przelewu vs termin na wniesienie
F4	Odwołanie podpisane podpisem kwalifikowanym lub zaufanym	art. 508 ust. 1 PZP	Weryfikacja DSS
F5	Podpis złożony przez osobę uprawnioną do reprezentacji	art. 516 ust. 1 PZP	Porównanie: sygnatariusz (DSS) vs skład organu (KRS/CEIDG) lub pełnomocnictwo
F6	Pełnomocnictwo załączone (jeśli wymagane)	art. 516 ust. 1 PZP	Klasyfikacja dokumentów (biblioteka)
F7	Opłata skarbową od pełnomocnictwa (jeśli dotyczy)	ustawa o opłacie skarbowej	Potwierdzenie przelewu

ID	Kryterium	Podstawa prawna	Źródło danych
F8	Dowód przekazania kopii odwołania zamawiającemu	art. 516 ust. 1 PZP	Klasyfikacja dokumentów (biblioteka)
F9	Wskazanie czynności lub zaniechania zamawiającego	art. 516 ust. 1 pkt 1 PZP	Ekstrakcja z odwołania (biblioteka)
F10	Wskazanie żądania	art. 516 ust. 1 pkt 3 PZP	Ekstrakcja z odwołania (biblioteka)
F11	Wskazanie okoliczności faktycznych i prawnych	art. 516 ust. 1 pkt 4 PZP	Ekstrakcja z odwołania (biblioteka)
F12	Dane odwołującego (nazwa/imię, adres, NIP/KRS)	art. 516 ust. 1 PZP	Ekstrakcja z odwołania + weryfikacja KRS/CEIDG

### 9.3. Wynik oceny formalnej

Workflow musi wygenerować strukturę danych zawierającą:

1. **Wynik ogólny:** `braki_formalne`: TAK/NIE
2. **Lista zweryfikowanych kryteriów:** dla każdego kryterium F1–F12:
  - o `status`: spełnione / niespełnione / `do_weryfikacji_ręcznej` / `nie_dotyczy`
  - o `uzasadnienie`: tekstowe uzasadnienie wyniku
  - o `źródło`: cytat źródłowy (plik, strona, API)
3. **Lista stwierdzonych braków:** jeśli `braki_formalne`: TAK — lista braków z odwołaniem do kryteriów.

### 9.4. Scenariusze po ocenie formalnej

Wynik oceny	Działanie systemu
Brak braków formalnych	Przejdzie do Fazy 9 — generowanie Karty Odwołania
Stwierdzono braki formalne	Przejdzie do Fazy 9 — generowanie Karty Odwołania z oznaczeniem braków ORAZ generowanie wezwania do uzupełnienia (rozdział 11)
Nie można jednoznacznie ocenić	Oznaczenie kryteriów statusem <code>do_weryfikacji_ręcznej</code> w Karcie Odwołania

## 10. Generowanie Karty Odwołania

### 10.1. Cel

Karta Odwołania jest głównym produktem systemu — kompleksowym dokumentem podsumowującym stan formalny sprawy odwoławczej.

### 10.2. Formaty wyjściowe

System musi generować Kartę Odwołania w dwóch równoległych formatach:

Format	Zastosowanie	Technologia
<b>DOCX</b>	Dokument do wydruku, edycji i archiwizacji przez pracowników KIO	Python 3.11+ z biblioteką python-docx
<b>JSON</b>	Dane strukturalne do integracji z innymi systemami	JavaScript (n8n Code Node)

Oba formaty MUSZĄ zawierać identyczne dane.

### 10.3. Struktura Karty Odwołania

Karta musi zawierać następujące sekcje:

#### 10.3.1. Nagłówek

- Sygnatura akt: KIO [NR]/[ROK]
- Numer DzK, sekretariat, data
- **Podsumowanie: Braki formalne: TAK/NIE**

#### 10.3.2. Tabela Informacji Ogólnych

- ZAMÓWIENIE (typ, nazwa postępowania)
- TRYB POSTĘPOWANIA
- PUBLIKATOR (TED/BZP z numerem ogłoszenia)
- CZYNNOŚĆ (opis zaskarżonej czynności z datą)
- TERMIN NA WNIESIENIE (data obliczona)
- WNIESIONE W TERMINIE (ocena: TAK/NIE z datą)
- WPIS (kwota wymagana, wniesiona, data, ocena)

#### 10.3.3. Sekcja 1: ODWOŁUJĄCY

- Konsorcjum (TAK/NIE)
- Nazwa podmiotu, forma prawna
- Adres siedziby, dane kontaktowe (email, telefon)
- Identyfikatory (NIP, REGON, KRS)
- Adres do doręczeń elektronicznych (BAE)

- ePUAP ID

#### 10.3.4. Sekcja 2: *REPREZENTACJA ODWOŁUJĄCEGO*

- Zasady reprezentacji (z KRS/CEIDG)
- Osoby uprawnione do reprezentacji (zarząd/wspólnicy)
- Osoba podpisująca odwołanie (z weryfikacji podpisu)
- Typ podpisu (zaufany/kwalifikowany)
- Pełnomocnictwo (czy wymagane, czy załączone)
- Opłata skarbową od pełnomocnictwa

#### 10.3.5. Sekcja 3: *ZAMAWIAJĄCY*

- Nazwa, adres, dane kontaktowe
- Identyfikatory (NIP, REGON)
- Pełnomocnik zamawiającego (jeśli dotyczy)

#### 10.3.6. Sekcja 4: *WERYFIKACJA FORMALNA*

- Numer ogłoszenia TED/BZP
- Data czynności zamawiającego
- Terminy (ustawowy, wyliczony, faktyczny)
- Wniesione w terminie (TAK/NIE)
- Dowód uiszczenia wpisu (kwoty, daty, ocena)
- Dowód przekazania kopii zamawiającemu
- Dokument potwierdzający umocowanie
- **Ocena braków formalnych (art. 516–518 PZP)**

#### 10.3.7. *Stopka*

- Skład orzekający KIO (do wypełnienia ręcznie)
- Data i podpis Prezesa KIO

### 10.4. System cytowania źródeł

Każda informacja w Karcie Odwołania musi posiadać cytat źródłowy:

Typ źródła	Format cytatu
Dokument PDF	Nazwa_pliku.pdf > strona: X
API rejestrów	Odpis_KRS_XXX.pdf > strona: X
Ustawa PZP	ustawa PZP.pdf > art. X ust. Y pkt Z
Weryfikacja podpisu	Weryfikacja podpisu (ekran) > strona: X
Brak danych	BRAK DANYCH

### 10.5. Ikony statusu

Ikona	Znaczenie	Status
-------	-----------	--------

Ikona	Znaczenie	Status
✓	Dane znalezione i zweryfikowane	verified
⚠	Dane częściowe lub wymagają weryfikacji	warning
✗	Dane wymagane, ale brak w dokumentacji	missing

## 10.6. Priorytet źródeł danych (przy konflikcie)

1. **API rejestrów** (KRS, CEIDG, REGON) — najwyższa wiarygodność
2. **Weryfikacja podpisów** (DSS) — dla danych o podpisach
3. **Ekstrakcja z dokumentów oryginalnych** (odwołanie, pełnomocnictwo)
4. **Ekstrakcja z dokumentów wtórnych** (ogłoszenie, potwierdzenia)

## 10.7. Formatowanie dokumentu DOCX

Element	Specyfikacja
Czcionka treści	Calibri, 11pt
Czcionka nagłówek	Calibri Bold, 12pt
Marginesy	2 cm (góra, dół, lewo, prawo)
Tabele	Obramowanie 0,5pt czarne, nagłówki z tłem #F2F2F2
Numeracja stron	W stopce

## 11. Obsługa braków formalnych — wezwanie do uzupełnienia

### 11.1. Cel

W przypadku stwierdzenia braków formalnych odwołania (wynik oceny z rozdziału 9), system musi automatycznie wygenerować wezwanie do uzupełnienia braków i dostarczyć je wskazanymi kanałami.

### 11.2. Szablony wezwania

Wykonawca musi dostarczyć konfigurowalne szablony wezwania do uzupełnienia braków, zawierające:

1. **Dane adresata** (odwołujący — z ekstrakcji).
2. **Sygnaturę sprawy** (KIO [NR]/[ROK]).
3. **Listę stwierdzonych braków** — automatycznie wypełnianą na podstawie wyników oceny formalnej (kryteria F1–F12 z Załącznika G).
4. **Podstawę prawną wezwania** — art. 518 ust. 1 PZP.
5. **Termin na uzupełnienie** — konfigurowalny parametr (domyślnie: 3 dni robocze).
6. **Dane KIO** — nagłówek, adres, dane kontaktowe.

Szablony MUSZA być edytowalne przez Zamawiającego (format DOCX z polami merge lub konfiguracja w n8n).

### 11.3. Scenariusz 1: Umieszczenie dokumentu na SharePoint

Workflow musi:

1. Wygenerować dokument wezwania w formacie DOCX na podstawie szablonu.
2. Umieścić dokument w katalogu sprawy na SharePoint:

```
Przetworzone/[case_id]/Wezwanie_do_uzupelnienia/  
├─ wezwanie_[case_id]_[data].docx  
└─ wezwanie_[case_id]_[data].json      # Dane strukturalne wezwania
```

3. Oznaczyć sprawę odpowiednim statusem (np. metadata/tag w SharePoint).

### 11.4. Scenariusz 2: Wystanie wezwania e-mailem

Workflow musi:

1. Wygenerować treść wezwania w formacie HTML (do treści maila) oraz załącznik DOCX.
2. Wysłać e-mail za pośrednictwem skonfigurowanego serwera SMTP (lub Microsoft Graph API — zależnie od konfiguracji n8n Credentials):
  - o **Do:** adres e-mail odwołującego (z ekstrakcji)
  - o **Temat:** Wezwanie do uzupełnienia braków formalnych – [sygnatura]
  - o **Treść:** wezwanie w formacie HTML z listą braków
  - o **Załącznik:** wezwanie w formacie DOCX
3. Zapisać potwierdzenie wysłania (message ID, timestamp) w katalogu sprawy na SharePoint.

### 11.5. Konfiguracja

Parametr	Opis	Domyślna
DEFICIENCY_TEMPLATE_PATH	Ścieżka do szablonu DOCX wezwania	templates/wezwanie_template.docx
DEFICIENCY_DEADLINE_DAYS	Termin na uzupełnienie (dni robocze)	3
DEFICIENCY_EMAIL_ENABLED	Czy wysyłać e-mail (true/false)	true
DEFICIENCY_SHAREPOINT_ENABLE	Czy umieszczać na SharePoint (true/false)	true

---

## 12. Wymagania dotyczące testowania i odbioru

### 12.1. Zbiory testowe

Zamawiający dostarczy:

1. **20 zbiorów dokumentów** — każdy zbiór stanowi kompletne odwołanie do KIO (pliki ZIP zawierające: odwołanie, dokumenty rejestrowe, pełnomocnictwa, dowody wpisu, potwierdzenia przelewów).
2. **20 kart odwołań referencyjnych** — w formacie DOC lub DOCX, odpowiadające zbiorom dokumentów. Karty te stanowią wzorzec (ground truth) poprawnie wypełnionej Karty Odwołania.

### 12.2. Plik ground\_truth.yaml

#### 12.2.1. Wymaganie

Dla każdego z 20 zbiorów testowych Wykonawca musi dostarczyć plik `ground_truth.yaml` będący odwzorowaniem oczekiwanych informacji na Karcie Odwołania.

#### 12.2.2. Format pliku

Plik `ground_truth.yaml` musi zawierać ścieżkowe asercje do wszystkich pól Karty Odwołania, zgodnie z formatem stosowanym w repozytorium biblioteki do klasyfikacji i ekstrakcji metadanych:

```
version: 2
```

```
min_similarity: 0.55
```

```
assert:
```

```
  # Nagłówek
```

```
  naglowek.sygnatura: "KIO 123/26"
```

```
  naglowek.braki_formalne: false
```

```
  # Informacje ogólne
```

```
  informacje_ogolne.zamowienie: "Usługi KLASYCZNE: ..."
```

```
  informacje_ogolne.tryb_postepowania: "Przetarg nieograniczony"
```

```
  informacje_ogolne.publikator: "TED powyżej progu UE: 487788-2025"
```

```
  informacje_ogolne.wniesione_w_terminie: true
```

```
  informacje_ogolne.wpis.wymagane: 15000.00
```

```
  informacje_ogolne.wpis.wniesione: 15000.00
```

```
  # Sekcja 1: Odwołujący
```

```
  odwolujacy.konsorcjum: false
```

```
  odwolujacy.nazwa: "BUD-POL Spółka z o.o."
```

```
  odwolujacy.nip: "5252001234"
```

```
  odwolujacy.krs: "0000123456"
```

```
  odwolujacy.adres.miasto: "Warszawa"
```

```
odwolujacy.email: "biuro@budpol.pl"
```

#### # Sekcja 2: Reprezentacja

```
reprezentacja.zasady: "Samodzielnie - Prezes Zarządu"  
reprezentacja.podpisujacy.imie_nazwisko: "Jan Kowalski"  
reprezentacja.podpisujacy.typ_podpisu: "kwalifikowany"  
reprezentacja.pelnomocnictwo_wymagane: false
```

#### # Sekcja 3: Zamawiający

```
zamawiajacy.nazwa: "Gmina Przykładowo"  
zamawiajacy.nip: "1234567890"
```

#### # Sekcja 4: Weryfikacja formalna

```
weryfikacja.termin_ustawowy: "2026-01-09"  
weryfikacja.wniesione_w_terminie: true  
weryfikacja.wpis_prawidlowy: true  
weryfikacja.braki_formalne: false
```

### 12.2.3. Struktura katalogowa zbiorów testowych

```
test_cases/  
├── case_001/  
│   ├── input/  
│   │   └── odwolanie_001.zip  
│   ├── reference/  
│   │   └── karta_odwolania_001.docx # Dostarczona przez Zamawiającego  
│   ├── expected/  
│   │   └── ground_truth.yaml # Wygenerowana przez skrypt Wykonaw  
├── cy  
│   ├── case_002/  
│   │   └── ...  
│   ├── case_020/  
│   │   └── ...
```

## 12.3. Skrypt generowania ground\_truth.yaml

### 12.3.1. Wymaganie

Wykonawca musi dostarczyć skrypt w języku Python, który na podstawie dokumentu DOC lub DOCX (referencyjnej Karty Odwołania dostarczonej przez Zamawiającego) generuje plik ground\_truth.yaml.

### 12.3.2. Specyfikacja skryptu

Parametr	Wartość
<b>Nazwa</b>	generate_ground_truth.py
<b>Język</b>	Python 3.11+
<b>Wejście</b>	Ścieżka do pliku DOC/DOCX (referencyjna Karta Odwołania)
<b>Wyjście</b>	Plik ground_truth.yaml w formacie

Parametr	Wartość
	opisanym w punkcie 12.2.2
<b>Zależności</b>	python-docx, pyyaml (lub równoważne)

### 12.3.3. Wymagania funkcjonalne skryptu

1. Parsowanie struktury dokumentu DOCX (tabele, sekcje, nagłówki).
2. Ekstrakcja wartości pól dla wszystkich 4 sekcji Karty Odwołania.
3. Mapowanie wyekstrahowanych wartości na ścieżki YAML zgodne ze schematem `ground_truth.yaml`.
4. Obsługa wartości pustych (mapowanie na `null`).
5. Obsługa formatu DOC (konwersja do DOCX lub bezpośrednia ekstrakcja).

### 12.3.4. Przykład wywołania

```
python generate_ground_truth.py \
  --input reference/karta_odwolania_001.docx \
  --output expected/ground_truth.yaml
```

## 12.4. Skrypt ewaluacji accuracy

### 12.4.1. Wymaganie

Wykonawca musi dostarczyć skrypt w języku Python, który:

1. Wywołuje workflow n8n dla wskazanego zbioru dokumentów testowych.
2. Porównuje wynik (wygenerowaną Kartę Odwołania) z plikiem `ground_truth.yaml`.
3. Generuje raport accuracy z historią uruchomień.

### 12.4.2. Specyfikacja skryptu

Parametr	Wartość
<b>Nazwa</b>	<code>evaluate_accuracy.py</code>
<b>Język</b>	Python 3.11+
<b>Wejście</b>	Katalog ze zbiorami testowymi (struktura wg 10.2.3), URL instancji n8n, dane uwierzytelniające
<b>Wyjście</b>	Raport w formacie JSON + podsumowanie tekstowe na stdout
<b>Zależności</b>	<code>requests</code> , <code>pyyaml</code> , <code>deepdiff</code> (lub równoważne)

### 12.4.3. Wymagania funkcjonalne skryptu

Skrypt musi realizować następujące operacje:

1. **Trigger workflow n8n** — wywołanie workflow dla każdego zbioru testowego (upload ZIP do SharePoint lub bezpośrednio wywołanie webhook n8n).
2. **Oczekiwanie na wynik** — polling statusu przetwarzania z konfigurowalnym timeout.

3. **Pobranie wyniku** — pobranie wygenerowanej Karty Odwołania (JSON) z SharePoint lub bezpośrednio z n8n.
4. **Porównanie z ground truth** — porównanie każdego pola `ground_truth.yaml` z odpowiadającym polem w wygenerowanej Karcie.
5. **Obliczenie accuracy** — stosunek pól zgodnych do łącznej liczby pól (per zbiór i globalnie).
6. **Dopisywanie historii** — każde uruchomienie skryptu musi dopisywać nowy wiersz do pliku historii (`accuracy_history.json`), zawierający:
  - Datę i czas uruchomienia (ISO 8601)
  - Globalny wynik accuracy (%)
  - Liczbę zbiorów testowych
  - Liczbę rozbieżności ogółem
  - Wersję workflow (jeśli dostępna)

#### 12.4.4. Zawartość raportu

Raport musi zawierać następujące sekcje:

##### A. Podsumowanie bieżącego uruchomienia:

Metryka	Opis
<code>accuracy</code>	Globalny wynik accuracy (%)
<code>total_fields</code>	Łączna liczba porównywanych pól
<code>matching_fields</code>	Liczba pól zgodnych
<code>discrepancies</code>	Liczba rozbieżności
<code>timestamp</code>	Data i czas uruchomienia

**B. Historia uruchomień** — tabela ze wszystkimi poprzednimi uruchomieniami (wczytywana z `accuracy_history.json`), umożliwiająca śledzenie trendu accuracy w czasie.

##### C. TOP 10 pól YAML z największą liczbą rozbieżności:

Ranking 10 elementów YAML (ścieżek pól), dla których najczęściej występują rozbieżności między ground truth a wynikiem workflow, np.:

Rank	Pole YAML	Rozbieżności	% zbiorów
1	<code>reprezentacja.podpisujacy.imie_nazwisko</code>	14/20	70%
2	<code>odwolujacy.email</code>	10/20	50%
3	<code>weryfikacja.termin_ustawowy</code>	8/20	40%
...			

##### D. TOP 10 dokumentów z największą liczbą rozbieżności:

Ranking 10 zbiorów testowych (odwołań), których format uniemożliwia precyzyjne przetworzenie, np.:

Rank	Zbiór	Accuracy	Rozbieżności	Uwagi
1	case_007	45%	22/40	Skan niskiej jakości
2	case_003	62%	15/40	Nietypowy format odwołania
3	case_019	70%	12/40	Brak podpisu elektronicznego
...				

#### 12.4.5. Przykład wywołania

```
python evaluate_accuracy.py \
  --test-dir ./test_cases/ \
  --n8n-url http://localhost:5678 \
  --n8n-api-key "your-api-key" \
  --history-file ./accuracy_history.json \
  --output-report ./report_2026-02-09.json
```

#### 12.4.6. Format pliku historii (accuracy\_history.json)

```
[
  {
    "timestamp": "2026-02-01T10:00:00Z",
    "accuracy": 0.85,
    "total_fields": 800,
    "matching_fields": 680,
    "discrepancies": 120,
    "cases_count": 20,
    "workflow_version": "1.0.0",
    "top10_fields": [...],
    "top10_documents": [...]
  },
  {
    "timestamp": "2026-02-09T14:00:00Z",
    "accuracy": 0.92,
    "total_fields": 800,
    "matching_fields": 736,
    "discrepancies": 64,
    "cases_count": 20,
    "workflow_version": "1.1.0",
    "top10_fields": [...],
    "top10_documents": [...]
  }
]
```

## 12.5. Testy akceptacyjne

Odbiór zamówienia wymaga przeprowadzenia testów akceptacyjnych na wszystkich 20 zbiorach testowych:

1. **Test przetwarzania End-to-End:** każdy z 20 zbiorów musi zostać przetworzony przez system bez błędów krytycznych.
2. **Test accuracy:** globalny wynik accuracy musi wynosić co najmniej **80%** (mierzony skryptem `evaluate_accuracy.py`).

3. **Test generowania Karty Odwołania:** Karta musi być wygenerowana w obu formatach (DOCX + JSON) dla każdego zbioru.
4. **Test integralności danych:** dane w DOCX i JSON MUSZĄ być identyczne.
5. **Test obsługi błędów:** co najmniej 2 scenariusze z celowo wprowadzonymi brakami (brak pliku, uszkodzony ZIP) MUSZĄ zakończyć się poprawną obsługą błędu.
6. **Test oceny formalnej:** workflow oceny formalnej musi poprawnie identyfikować braki formalne zgodnie z kryteriami z Załącznika G.
7. **Test wezwania do uzupełnienia:** w przypadku stwierdzonych braków, system musi wygenerować wezwanie (DOCX na SharePoint + e-mail).

### 13. Produkty i rezultaty zamówienia (Deliverables)

ID	Deliverable	Opis	Format
<b>D1</b>	Workflow oceny formalnej	Workflow n8n realizujący ocenę formalną wg kryteriów Załącznika G	JSON (n8n export)
<b>D2</b>	Generator Karty Odwołania	Kod Python generujący Kartę Odwołania w formacie DOCX	Python 3.11+
<b>D3</b>	Szablony wezwań do uzupełnienia	Szablony DOCX wezwania do uzupełnienia braków formalnych	DOCX + konfiguracja
<b>D4</b>	Workflow przetwarzania odwołań (n8n)	Eksportowane workflow n8n realizujące pełny proces end-to-end (fazy 1–10), gotowe do importu	JSON (n8n export)
<b>D5</b>	Kody źródłowe integracji (n8n nodes)	Komplet n8n nodes (workflow.json, manifest.yaml, README, testy)	Repozytorium Git

ID	Deliverable	Opis	Format
<b>D6</b>	Konfiguracja środowiska (Docker)	Plik docker-compose.yml z pełną konfiguracją (n8n + Redis + PostgreSQL + DSS + docling-serve + n8n-mcp + LiteLLM)	Docker Compose YAML
<b>D7</b>	Pliki ground_truth.yaml	20 plików ground truth dla zbiorów testowych	YAML
<b>D8</b>	Skrypt generowania ground truth	generate_ground_truth.py	Python 3.11+
<b>D9</b>	Skrypt ewaluacji accuracy	evaluate_accuracy.py z historią uruchomień	Python 3.11+
<b>D10</b>	Raport z testów akceptacyjnych	Wyniki testów na 20 zbiorach z raportem accuracy	JSON + PDF
<b>D11</b>	Dokumentacja wdrożeniowa	Instrukcja instalacji, konfiguracji i uruchomienia (w tym hardening)	Markdown
<b>D12</b>	Dokumentacja techniczna	Opis architektury, przepływów danych, kontraktów API	Markdown
<b>D13</b>	Konfiguracja zmiennych środowiskowych	Lista wymaganych zmiennych z opisem i wartościami domyślnymi	Markdown /.env.example
<b>D14</b>	[OPCJA] Integracja weryfikacji wpisu	n8n node	n8n node + workflow

ID	Deliverable	Opis	Format
		integracji z NBP B2B lub Symfonią (do odrębnej wyceny)	

## 14. Hardening środowiska produkcyjnego

### 14.1. Cel

Wykonawca musi przeprowadzić hardening środowiska produkcyjnego w celu zabezpieczenia systemu przed nieautoryzowanym dostępem.

### 14.2. Wymagania

ID	Wymaganie	Opis
H1	<b>Podłączenie do domeny</b>	Serwery produkcyjne MUSZĄ być podłączone do domeny Active Directory Zamawiającego
H2	<b>Kontrola dostępu</b>	Dostęp do interfejsu n8n, konsoli Docker i SSH musi być ograniczony wyłącznie do osób posiadających uprawnienia administracyjne w domenie
H3	<b>Zamknięcie portów</b>	Wszystkie porty sieciowe inne niż niezbędne do funkcjonowania systemu MUSZĄ być zamknięte (firewall)
H4	<b>Komunikacja wewnętrzna</b>	Kontenery Docker MUSZĄ komunikować się wyłącznie przez wewnętrzną sieć Docker — porty usług (Redis, PostgreSQL, DSS, docling-serve) NIE

ID	Wymaganie	Opis
H5	<b>HTTPS</b>	MOGĄ być eksponowane na zewnątrz Interfejs n8n musi być dostępny wyłącznie przez HTTPS (certyfikat SSL/TLS)
H6	<b>Logowanie dostępu</b>	System musi logować próby dostępu (udane i nieudane) do interfejsu n8n

### 14.3. Dokumentacja

Wykonawca musi dostarczyć w ramach dokumentacji wdrożeniowej (D11):

1. Listę portów wymaganych do funkcjonowania systemu.
2. Instrukcję podłączenia do domeny AD.
3. Konfigurację firewalla (iptables/ufw).
4. Konfigurację reverse proxy z SSL (np. nginx/Caddy).
5. Procedurę weryfikacji poprawności hardeningu.

## 15. Warunki realizacji zamówienia

### 15.1. Harmonogram

Etap	Opis	Czas trwania
<b>Etap 1: Analiza</b>	Analiza procesu oceny formalnej, uzgodnienie wymagań biznesowych, dostęp do środowisk i API (w tym opcjonalnie NBP B2B / Symfonia)	2 tygodnie
<b>Etap 2: Implementacja</b>	Budowa workflow oceny formalnej, generatora Karty Odwołania, obsługi braków formalnych,	6 tygodni

Etap	Opis	Czas trwania
	n8n nodes i integracji	
<b>Etap 3: Integracja i testy</b>	Integracja komponentów, testy jednostkowe i integracyjne, konfiguracja środowiska i hardening	2 tygodnie
<b>Etap 4: Testy akceptacyjne</b>	Testy na zbiorach Zamawiającego, weryfikacja poprawności oceny formalnej, korekty, dokumentacja	2 tygodnie

Łączny czas realizacji: **12 tygodni** od daty podpisania umowy.

## 15.2. Środowisko pracy

1. Zamawiający udostępni środowisko deweloperskie/testowe (serwer Ubuntu z Docker) w ciągu 5 dni roboczych od podpisania umowy.
2. Zamawiający udostępni dostęp do instancji biblioteki do klasyfikacji i ekstrakcji metadanych (API) oraz dokumentację Swagger.
3. Zamawiający dostarczy klucze API do systemów zewnętrznych.
4. Zamawiający dostarczy 20 zbiorów testowych oraz odpowiadające im karty odwołań (DOC/DOCX) w ciągu 10 dni roboczych od podpisania umowy.

## 15.3. Gwarancja i wsparcie

1. Wykonawca udziela **6-miesięcznej gwarancji** na dostarczone oprogramowanie, liczonej od daty odbioru końcowego.
2. W okresie gwarancji Wykonawca zobowiązuje się do usunięcia wad w terminie:
  - o Błędy krytyczne (brak możliwości przetwarzania): 2 dni robocze
  - o Błędy niekrytyczne: 5 dni roboczych
3. Wsparcie techniczne w okresie gwarancji: email + wideokonferencja, czas reakcji max 1 dzień roboczy.

## 15.4. Przeniesienie praw

1. Wykonawca przeniesie na Zamawiającego pełne majątkowe prawa autorskie do wszystkich dostarczonych kodów źródłowych, workflow, skryptów i dokumentacji.
2. Przeniesienie praw obejmuje prawo do modyfikacji, rozpowszechniania i dalszego rozwoju.

## 16. Umowa powierzenia przetwarzania danych osobowych

### 16.1. Wymaganie

Zawarcie umowy o powierzeniu przetwarzania danych osobowych jest warunkiem koniecznym realizacji zamówienia.

### 16.2. Zakres powierzenia

Element	Wartość
<b>Administrator danych</b>	Zamawiający (Urząd Zamówień Publicznych)
<b>Podmiot przetwarzający</b>	Wykonawca
<b>Kategorie osób</b>	Strony postępowań odwoławczych (odwołujący, zamawiający, pełnomocnicy, członkowie organów spółek)
<b>Kategorie danych</b>	Imiona i nazwiska, NIP, REGON, KRS, PESEL (z certyfikatów podpisów), adresy, dane kontaktowe, numery rachunków bankowych
<b>Cel przetwarzania</b>	Realizacja zamówienia: budowa, testowanie i wdrożenie narzędzia wspierającego ocenę formalną odwołań do KIO
<b>Czas przetwarzania</b>	Na czas realizacji umowy + 30 dni na usunięcie danych

### 16.3. Obowiązki Wykonawcy

1. Przetwarzanie danych wyłącznie w zakresie i celu określonym w umowie.
2. Zapewnienie, że osoby upoważnione do przetwarzania zobowiążą się do zachowania poufności.
3. Niezwłoczne informowanie Zamawiającego o naruszeniu ochrony danych osobowych (max 24h).
4. Po zakończeniu umowy: trwałe usunięcie wszystkich danych osobowych z systemów Wykonawcy i potwierdzenie tego faktu na piśmie.

## 17. Kryteria akceptacji i mierniki sukcesu

### 17.1. Kryteria odbioru

ID	Kryterium	Sposób weryfikacji
K1	Workflow oceny formalnej poprawnie identyfikuje braki	Weryfikacja na zbiorach testowych

ID	Kryterium	Sposób weryfikacji
K2	Karta Odwołania generowana w DOCX i JSON	Weryfikacja obecności plików
K3	Zgodność danych DOCX ↔ JSON	Automatyczne porównanie
K4	Wezwanie do uzupełnienia generowane na SharePoint i wysłane e-mailem	Test scenariuszy z brakami formalnymi
K5	Przetworzenie 20/20 zbiorów testowych bez błędów krytycznych	Raport z testów akceptacyjnych
K6	Globalny accuracy ≥ 80%	Raport skryptu evaluate_accuracy.py
K7	Wszystkie integracje podstawowe (KRS, CEIDG, REGON, BZP) działające	Logi n8n, pliki *_api.json
K8	Weryfikacja podpisów DSS działająca	Raporty weryfikacji w katalogu weryfikacja_podpisow/
K9	Docker-compose uruchamia pełne środowisko (n8n + Redis + PostgreSQL + DSS + docling-serve + n8n-mcp + LiteLLM)	Test docker compose up + health checks
K10	Dostarczenie plików ground_truth.yaml dla 20 zbiorów	Weryfikacja obecności i poprawności formatu
K11	Skrypt generate_ground_truth.py poprawnie parsuje DOCX	Test na dostarczonej karcie referencyjnej
K12	Skrypt evaluate_accuracy.py generuje kompletny raport	Weryfikacja raportu: accuracy, historia, TOP10 pól, TOP10 dokumentów
K13	Hardening środowiska wykonany (AD, porty, HTTPS)	Weryfikacja konfiguracji
K14	Dokumentacja wdrożeniowa kompletna i wystarczająca do samodzielnego uruchomienia	Przegląd przez Zamawiającego

ID	Kryterium	Sposób weryfikacji
K15	Umowa powierzenia danych osobowych podpisana	Dokument prawny

## 17.2. Procedura odbioru

1. Wykonawca zgłasza gotowość do odbioru.
2. Zamawiający weryfikuje deliverables D1–D14 (max 5 dni roboczych).
3. Zamawiający uruchamia skrypt `evaluate_accuracy.py` na 20 zbiorach testowych.
4. Jeżeli `accuracy`  $\geq 80\%$  i kryteria K1–K15 spełnione — podpisanie protokołu odbioru.
5. Jeżeli `accuracy`  $< 80\%$  lub kryteria niespełnione — Wykonawca ma 10 dni roboczych na korektę i ponowne zgłoszenie.

## 18. Załączniki

Integralną częścią niniejszego OPZ są następujące załączniki:

Załącznik	Tytuł	Opis
<b>Załącznik A</b>	Specyfikacja API biblioteki do klasyfikacji i ekstrakcji metadanych	Kontrakty endpointów, formaty request/response, kody błędów
<b>Załącznik B</b>	Model kanoniczny weryfikacji podpisów	Schemat JSON modelu wyjściowego adaptera podpisów
<b>Załącznik C</b>	Struktura Karty Odwołania (JSON)	Pełny schemat JSON Karty Odwołania z opisem pól
<b>Załącznik D</b>	Schemat <code>ground_truth.yaml</code>	Pełny schemat pliku ground truth z opisem wszystkich pól
<b>Załącznik E</b>	Specyfikacja raportu <code>accuracy</code>	Schemat JSON raportu generowanego przez <code>evaluate_accuracy.py</code>
<b>Załącznik F</b>	Wzór umowy powierzenia przetwarzania danych osobowych	Wzór umowy RODO

Załącznik

Tytuł

Opis

---

**Załącznik G**

Kryteria oceny formalnej  
odwołania

Szczegółowa  
specyfikacja  
kryteriów F1–F12  
weryfikacji formalnej

# Załącznik A — Specyfikacja API biblioteki do klasyfikacji i ekstrakcji metadanych

## 1. Informacje ogólne

Parametr	Wartość
<b>Typ API</b>	REST (HTTP/JSON)
<b>Wersja</b>	v1
<b>Autoryzacja</b>	Nagłówek X-API-Key (wymagany dla wszystkich endpointów /v1/*)
<b>Dokumentacja interaktywna</b>	Swagger UI udostępniany przez Zamawiającego
<b>Base URL</b>	Konfigurowany zmienną środowiskową (np. http://extraction-api:8080)

## 2. Endpointy

### 2.1. POST /v1/extract — Ekstrakcja danych z dokumentu

**Opis:** Główny endpoint do przetwarzania dokumentów. Przyjmuje plik (PDF, DOCX, JPG, PNG) i zwraca wyekstrahowane dane strukturalne.

**Request:**

```
POST /v1/extract HTTP/1.1
Host: extraction-api:8080
X-API-Key: <klucz_api>
Content-Type: multipart/form-data
```

```
--boundary
Content-Disposition: form-data; name="file"; filename="odwolanie.pdf"
Content-Type: application/pdf
```

```
<zawartość pliku>
--boundary
Content-Disposition: form-data; name="locale_hint"
```

```
pl-PL
--boundary
Content-Disposition: form-data; name="return_mode"
```

```
auto
--boundary--
```

**Parametry formularza:**

Parametr	Typ	Wymagany	Domyślna	Opis
file	binary	TAK	—	Dokument wejściowy (PDF, DOCX, JPG, PNG)
filename	string	NIE	nazwa z file	Nazwa logiczna pliku
stage2_method	string	NIE	lapdoc_spatial	Strategia OCR/ekstrakcji Stage2
locale_hint	string	NIE	pl-PL	Locale dokumentu
return_mode	string	NIE	auto	Tryb zwracania wyniku: auto, sync, async

### Tryby odpowiedzi:

Tryb	Zachowanie
auto	Serwer sam decyduje: jeśli przetwarzanie < 800s → 200, w przeciwnym razie → 202
sync	Wymuszenie próby 200 (do budżetu czasu 800s)
async	Zawsze zwraca 202 (asynchronicznie, wynik przez polling)

## 2.2. Odpowiedź 200 OK — Wynik synchroniczny (FinalResult v0)

```
{
  "schema_version": "v0",
  "run_id": "uuid-abcd-1234",
  "result": {
    "workspace": {
      "name": "case_001"
    },
    "files": [
      {
        "file_id": "odwolanie.pdf",
        "classification": {
          "best_category": "OdwolanieKIO",
          "confidence": 0.95,

```

```

    "candidates": [
      {"category": "OdwolanieKIO", "confidence": 0.95},
      {"category": "Inne", "confidence": 0.05}
    ]
  },
  "extraction": {
    "objects": {
      "OdwolanieKIO": [
        {
          "data": {
            "data": "2026-01-09",
            "miejscowosc": "Warszawa"
          },
          "quality": {
            "data": {"confidence": 0.92},
            "miejscowosc": {"confidence": 0.88}
          },
          "metadata": {
            "role_id": "OdwolanieKIO",
            "cardinality": "single"
          },
          "evidence": {
            "data": {"page_number": 1, "file_id": "odwolanie.pdf"},
            "miejscowosc": {"page_number": 1, "file_id": "odwolanie.pdf"}
          }
        }
      ]
    }
  },
  "Odwolujacy": [
    {
      "data": {
        "nazwa": "BUD-POL Sp. z o.o.",
        "nip": "5252001234",
        "krs": "0000123456",
        "adres": "ul. Mokra 12, 00-100 Warszawa"
      },
      "quality": {
        "nazwa": {"confidence": 0.95},
        "nip": {"confidence": 0.99, "verified": true},
        "krs": {"confidence": 0.99, "verified": true}
      }
    }
  ],
  "Zamawiajacy": [
    {
      "data": {
        "nazwa": "Gmina Przykładowo",
        "nip": "1234567890",
        "adres": "ul. Główna 1, 00-001 Przykładowo"
      }
    }
  ]
}

```

```
}
  ]
}
}
}
}
}
```

### 2.3. Odpowiedź 202 Accepted — Job asynchroniczny

```
{
  "job_id": "job-uuid-5678",
  "run_id": "uuid-abcd-1234",
  "status": "queued",
  "created_at": "2026-02-09T10:00:00Z",
  "expires_at": "2026-02-10T10:00:00Z",
  "status_url": "/v1/jobs/job-uuid-5678",
  "result_url": "/v1/jobs/job-uuid-5678/result",
  "input_sha256": "a1b2c3d4...",
  "context_sha256": "e5f6g7h8..."
}
```

**Nagłówki odpowiedzi 202:** - Location: /v1/jobs/job-uuid-5678 - Retry-After: 30

### 2.4. GET /v1/jobs/{job\_id} — Status joba

#### Odpowiedź:

```
{
  "job_id": "job-uuid-5678",
  "run_id": "uuid-abcd-1234",
  "status": "running",
  "created_at": "2026-02-09T10:00:00Z",
  "expires_at": "2026-02-10T10:00:00Z",
  "input_sha256": "a1b2c3d4..."
}
```

#### Statusy joba:

Status	Opis
queued	Job oczekuje na przetwarzanie
running	Przetwarzanie w toku
succeeded	Przetwarzanie zakończone pomyślnie
failed	Przetwarzanie zakończone błędem

Status	Opis
canceled	Job anulowany
expired	Job wygał (TTL 24h)

## 2.5. GET /v1/jobs/{job\_id}/result — Pobranie wyniku

Kod HTTP	Warunek	Opis
200 OK	Job succeeded	Zwraca FinalResult v0 (identyczny format jak w 2.2)
404 Not Found	Job nie istnieje	—
409 Conflict	Job istnieje, ale wynik niegotowy	—
410 Gone	Job expired	TTL upłynął
500 Internal Server Error	Job failed	—

**Uwaga:** Wynik jest usuwany po pierwszym udanym pobraniu (delete-on-fetch). Kolejne próby zwrócą 404.

## 2.6. DELETE /v1/jobs/{job\_id} — Anulowanie joba

Idempotentna operacja: anuluje job (jeśli możliwe) i usuwa artefakty z dysku.

Kod HTTP	Opis
200 OK	Job anulowany/wyczyszczony
404 Not Found	Job nie istnieje

## 3. Kody błędów

Kod HTTP	Opis
200	Sukces
202	Job utworzony (tryb asynchroniczny)
400	Nieprawidłowe żądanie (brak pliku, nieobsługiwany typ)
401	Brak nagłówka X-API-Key lub klucz nieprawidłowy
404	Zasób nie znaleziony
409	Konflikt (wynik niegotowy, job anulowany)
410	Zasób wygał
500	Błąd wewnętrzny serwera

## 4. Limity i parametry operacyjne

Parametr	Wartość
Budżet czasu sync	800 sekund
TTL joba	24 godziny
Retencja wyniku	Delete-on-fetch
Retencja pliku wejściowego	Usuwany po przetworzeniu

# Załącznik B — Model kanoniczny weryfikacji podpisów elektronicznych

## 1. Opis

Każdy adapter weryfikacji podpisów (DSS, WebNotarius lub inny) musi zwracać wynik w poniższym formacie kanonicznym. Format jest niezależny od dostawcy usługi weryfikacyjnej i stanowi kontrakt wejściowy dla dalszych kroków workflow (generowanie Karty Odwołania).

## 2. Schemat JSON

```
{
  "is_valid": true,
  "validation_time": "2026-02-08T12:00:00Z",
  "signers": [
    {
      "name": "Jan Kowalski",
      "certificate_serial": "1234567890ABC",
      "valid_at_signing": true,
      "is_qualified": true,
      "signing_time": "2026-02-07T15:30:00Z"
    }
  ],
  "signature_format": "PAdES-BASELINE-LT",
  "evidence_report_url": "Weryfikacja_podpisow/odwolanie_verification.pdf",
  "diagnostics": {
    "provider": "dss",
    "validation_policy": "BASELINE_LT",
    "tsl_version": "EU-LOTL-2026-02-01"
  }
}
```

### 3. Opis pól

#### 3.1. Pola główne

Pole	Typ	Wymagane	Opis
is_valid	boolean	TAK	Czy podpis jest ważny i integralny
validation_time	string (ISO 8601)	TAK	Czas wykonania walidacji
signers	array	TAK	Lista sygnatariuszy (≥ 1 element)
signature_format	string	TAK	Zidentyfikowany format podpisu (np. PAdES-BASELINE-LT, XAdES-BASELINE-T)
evidence_report_url	string	NIE	Ścieżka lub URL do raportu weryfikacji (PDF) w SharePoint
diagnostics	object	NIE	Dane diagnostyczne (opcjonalne)

#### 3.2. Pola obiektu signer

Pole	Typ	Wymagane	Opis
name	string	TAK	Imię i nazwisko z certyfikatu (CommonName)
certificate_serial	string	NIE	Numer seryjny certyfikatu
valid_at_signing	boolean	TAK	Czy certyfikat był ważny w momencie składania podpisu
is_qualified	boolean	TAK	Czy podpis kwalifikowany (QES) zgodnie z eIDAS
signing_time	string (ISO	TAK	Data i czas

Pole	Typ	Wymagane	Opis
	8601)		złożenia podpisu. Priorytet: bestSignatureTime (TSA) > claimedSigningTime

### 3.3. Pola obiektu diagnostics

Pole	Typ	Wymagane	Opis
provider	string	TAK	Identyfikator dostawcy (dss, webnotarius)
validation_policy	string	NIE	Polityka walidacji DSS (np. BASELINE_LT)
tsl_version	string	NIE	Wersja/data Trusted Service List użytej do walidacji

## 4. Mapowanie z DSS REST API

### 4.1. Odpowiedź DSS → Model kanoniczny

Pole DSS (JSON)	Pole kanoniczne	Transformacja
diagnosticData.signatureData[].indication — (timestamp walidacji)	is_valid validation_time	"TOTAL_PASSED" → true, inne → false Generowany przez adapter (UTC)
diagnosticData.usedCertificates[].commonName	signers[].name	Bezpośrednie mapowanie
diagnosticData.usedCertificates[].serialNumber	signers[].certificate_serial	Bezpośrednie mapowanie
diagnosticData.signatureData[].bestSignatureTime	signers[].signing_time	Priorytet 1 (TSA-verified)
diagnosticData.signatureData[].claimedSigningTime	signers[].signing_time	Fallback (gdy brak bestSignatureTime)
diagnosticData.signatureData[].signatureFormat	signature_format	Bezpośrednie mapowanie
diagnosticData.signatureData	signers[].is_qualified	Logika: QESIG / QESEAL →

Pole DSS (JSON)	Pole kanoniczne	Transformacja
ta[].signatureLevel		true

## 5. Obsługa wielu podpisów

Dokument może zawierać wiele podpisów. Adapter musi:

1. Wyekstrahować dane każdego sygnatariusza jako osobny element tablicy signers.
2. Ustawić `is_valid` na `true` TYLKO gdy WSZYSTKIE podpisy są ważne.
3. Zachować kolejność sygnatariuszy zgodną z kolejnością podpisów w dokumencie.

## Załącznik C — Struktura Karty Odwołania (JSON)

### 1. Opis

Poniższy schemat opisuje strukturę pliku `Karta_Odwołania_[case_id].json` generowanego jako jeden z dwóch głównych produktów systemu. Schemat pokrywa wszystkie sekcje Karty Odwołania opisane w rozdziale 9 OPZ.

### 2. Pełny schemat JSON

```
{
  "metadata": {
    "version": "1.0",
    "generated_at": "2026-02-09T12:00:00Z",
    "case_id": "KIO_123_26",
    "generator": "n8n-appeal-card-generator",
    "workflow_version": "1.0.0"
  },
  "naglowek": {
    "sygnatura": "KIO 123/26",
    "numer_dzk": "DzK-12345/26",
    "sekretariat": "Sekretariat I",
    "data": "2026-02-09",
    "braki_formalne": false,
    "braki_formalne_opis": "BRAK BRAKÓW FORMALNYCH"
  },
  "informacje_ogolne": {
    "zamowienie": {
      "value": "Usługi KLASYCZNE: Zadanie BOM nr 55...",
      "status": "verified",
      "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 1"
    }
  },
}
```

```
"tryb_postepowania": {
  "value": "Przetarg nieograniczony",
  "status": "verified",
  "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 1"
},
"publikator": {
  "value": "TED powyżej progu UE: 487788-2025",
  "status": "verified",
  "source": "eZamowienia API"
},
"czynnosc": {
  "value": "Czynność z dnia 2025-12-30: odrzucenie oferty",
  "status": "verified",
  "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 2"
},
"termin_na_wniesienie": {
  "value": "2026-01-09",
  "status": "verified",
  "source": "Obliczenie: art. 515 ust. 1 pkt 1 lit. a PZP"
},
"wniesione_w_terminie": {
  "value": true,
  "opis": "ODWOŁANIE W TERMINIE: 2026-01-09",
  "status": "verified",
  "source": "Weryfikacja podpisu (ekran) > strona: 1"
},
"wpis": {
  "wymagane": 15000.00,
  "wniesione": 15000.00,
  "data_wplaty": "2026-01-08",
  "waluta": "PLN",
  "prawidlowy": true,
  "status": "verified",
  "source": "Potwierdzenie przelewu > strona: 1"
}
},
"odwołujący": {
  "konsorcjum": {
    "value": false,
    "status": "verified",
    "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 1"
  },
  "nazwa": {
    "value": "BUD-POL Spółka z o.o.",
    "status": "verified",
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 1"
  },
  "forma_prawna": {
    "value": "Spółka z ograniczoną odpowiedzialnością",
```

```
    "status": "verified",
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 1"
  },
  "adres": {
    "ulica": {"value": "ul. Mokra 12", "status": "verified"},
    "kod_pocztowy": {"value": "00-100", "status": "verified"},
    "miasto": {"value": "Warszawa", "status": "verified"},
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 1"
  },
  "email": {
    "value": "biuro@budpol.pl",
    "status": "verified",
    "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 1"
  },
  "telefon": {
    "value": "+48 22 123 45 67",
    "status": "verified",
    "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 1"
  },
  "nip": {
    "value": "5252001234",
    "status": "verified",
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 1"
  },
  "regon": {
    "value": "123456789",
    "status": "verified",
    "source": "REGON API"
  },
  "krs": {
    "value": "0000123456",
    "status": "verified",
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 1"
  },
  "bae": {
    "value": null,
    "status": "missing",
    "source": "BRAK DANYCH"
  },
  "epuap": {
    "value": "/budpol/SkrytkaESP",
    "status": "verified",
    "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 1"
  }
},

"reprezentacja": {
  "zasady": {
    "value": "Do składania oświadczeń woli w imieniu Spółki upoważniony jest każdy Członek Zarządu samodzielnie",

```

```
    "status": "verified",
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 2"
  },
  "organ": {
    "value": "ZARZĄD",
    "status": "verified",
    "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 2"
  },
  "czlonkowie": [
    {
      "imie": "Jan",
      "nazwisko": "Kowalski",
      "funkcja": "PREZES ZARZĄDU",
      "status": "verified",
      "source": "Odpis_Aktualny_KRS_0000123456.pdf > strona: 2"
    }
  ],
  "podpisujacy": {
    "imie_nazwisko": {
      "value": "Jan Kowalski",
      "status": "verified",
      "source": "Weryfikacja podpisu (ekran) > strona: 1"
    },
    "typ_podpisu": {
      "value": "kwalifikowany",
      "status": "verified",
      "source": "Weryfikacja podpisu (ekran) > strona: 1"
    },
    "data_podpisu": {
      "value": "2026-01-09T13:20:32.256Z",
      "status": "verified",
      "source": "Weryfikacja podpisu (ekran) > strona: 1"
    },
    "podpis_wazny": {
      "value": true,
      "status": "verified",
      "source": "DSS Validation Report"
    }
  },
  "pelnomocnictwo_wymagane": {
    "value": false,
    "status": "verified",
    "source": "Analiza: podpisujący jest w składzie organu uprawnionego do reprezentacji"
  },
  "pelnomocnictwo_zalaczone": {
    "value": null,
    "status": "not_applicable"
  },
  "opлата_skarbowa": {
```

```
    "value": null,
    "status": "not_applicable"
  }
},

"zamawiajacy": {
  "nazwa": {
    "value": "Gmina Przykładowo",
    "status": "verified",
    "source": "eZamowienia API"
  },
  "adres": {
    "ulica": {"value": "ul. Główna 1", "status": "verified"},
    "kod_pocztowy": {"value": "00-001", "status": "verified"},
    "miasto": {"value": "Przykładowo", "status": "verified"},
    "source": "eZamowienia API"
  },
  "nip": {
    "value": "1234567890",
    "status": "verified",
    "source": "eZamowienia API"
  },
  "regon": {
    "value": "987654321",
    "status": "verified",
    "source": "REGON API"
  },
  "pelnomocnik": {
    "value": null,
    "status": "not_applicable"
  }
},

"weryfikacja": {
  "numer_ogloszenia": {
    "value": "487788-2025",
    "status": "verified",
    "source": "eZamowienia API"
  },
  "data_czynnosci": {
    "value": "2025-12-30",
    "status": "verified",
    "source": "Odwołanie IBiosfera 9.01.2026.pdf > strona: 2"
  },
  "termin_ustawowy": {
    "value": "10 dni (art. 515 ust. 1 pkt 1 lit. a PZP)",
    "status": "verified",
    "source": "ustawa PZP.pdf > art. 515 ust. 1 pkt 1 lit. a"
  },
  "termin_wyliczony": {
```

```
    "value": "2026-01-09",
    "status": "verified",
    "source": "Obliczenie: 2025-12-30 + 10 dni"
  },
  "termin_faktyczny": {
    "value": "2026-01-09",
    "status": "verified",
    "source": "Weryfikacja podpisu (ekran) > strona: 1"
  },
  "wniesione_w_terminie": {
    "value": true,
    "status": "verified"
  },
  "wpis_prawidlowy": {
    "value": true,
    "status": "verified",
    "source": "Potwierdzenie przelewu > strona: 1"
  },
  "dowod_przekazania_kopii": {
    "value": true,
    "status": "verified",
    "source": "Potwierdzenie przekazania > strona: 1"
  },
  "dokument_umocowania": {
    "value": "Nie wymagany (podpisujący w składzie organu)",
    "status": "verified"
  },
  "braki_formalne": {
    "value": false,
    "lista_brakow": [],
    "status": "verified"
  }
},

"stopka": {
  "sklad_orzekajacy": null,
  "data_podpisu_prezesa": null,
  "uwaga": "Do wypełnienia ręcznie przez pracownika KIO"
},

"audit_trail": {
  "processing_start": "2026-02-09T11:55:00Z",
  "processing_end": "2026-02-09T12:00:00Z",
  "sources_used": [
    "Biblioteka do klasyfikacji i ekstrakcji metadanych (API)",
    "KRS API",
    "REGON API",
    "eZamowienia API",
    "DSS Signature Verification"
  ],
}
```

```
    "warnings": [],  
    "errors": []  
  }  
}
```

### 3. Opis sekcji

Sekcja	Klucz JSON	Opis
Metadane	metadata	Wersja schematu, data generowania, identyfikator sprawy
Nagłówek	naglowek	Sygnatura akt, podsumowanie braków formalnych
Informacje ogólne	informacje_ogolne	Dane o zamówieniu, terminach, wpisie
Odwołujący	odwolujacy	Dane strony odwołującej
Reprezentacja	reprezentacja	Zasady i weryfikacja reprezentacji
Zamawiający	zamawiajacy	Dane zamawiającego
Weryfikacja formalna	weryfikacja	Ocena spełnienia wymogów formalnych
Stopka	stopka	Pola do ręcznego uzupełnienia
Audit trail	audit_trail	Dane audytowe przetwarzania

### 4. Konwencje wartości pól

#### 4.1. Obiekt z wartością i statusem

Większość pól w schemacie ma postać obiektu:

```
{  
  "value": "<wartość>",  
  "status": "<status>",  
  "source": "<źródło>"  
}
```

## 4.2. Statusy pól

Status	Znaczenie
verified	Dane zweryfikowane, kompletne
warning	Dane częściowe lub wymagające ręcznej weryfikacji
missing	Dane wymagane, ale nieznalesione w dokumentacji
not_applicable	Pole nie dotyczy danej sprawy

## 4.3. Wartości null

Wartość null oznacza brak danych. W połączeniu ze statusem: - "status": "missing" — dane powinny być obecne, ale ich nie znaleziono - "status": "not\_applicable" — pole nie dotyczy (np. pełnomocnictwo gdy podpisujący jest w zarządzie)

# Załącznik D — Schemat ground\_truth.yaml

## 1. Opis

Plik `ground_truth.yaml` stanowi referencję (wzorzec) oczekiwanych wartości pól Karty Odwołania dla danego zbioru testowego. Jest generowany na podstawie referencyjnej Karty Odwołania (DOC/DOCX) dostarczonej przez Zamawiającego i służy do automatycznej weryfikacji poprawności przetwarzania (accuracy).

---

## 2. Format pliku

```
version: 2
min_similarity: 0.55

assert:
  # — NAGŁÓWEK —
  naglowek.sygnatura: "KIO 123/26"
  naglowek.braki_formalne: false

  # — INFORMACJE OGÓLNE —
  informacje_ogolne.zamowienie: "Usługi KLASYCZNE: Zadanie BOM nr 55..."
  informacje_ogolne.tryb_postepowania: "Przetarg nieograniczony"
  informacje_ogolne.publikator: "TED powyżej progu UE: 487788-2025"
  informacje_ogolne.czynnosc: "Czynność z dnia 2025-12-30: odrzucenie oferty"
  informacje_ogolne.termin_na_wniesienie: "2026-01-09"
  informacje_ogolne.wniesione_w_terminie: true
  informacje_ogolne.wpis.wymagane: 15000.00
  informacje_ogolne.wpis.wniesione: 15000.00
  informacje_ogolne.wpis.prawidlowy: true

  # — SEKCJA 1: ODWOŁUJĄCY —
  odwolujacy.konsorcjum: false
  odwolujacy.nazwa: "BUD-POL Spółka z o.o."
  odwolujacy.forma_prawna: "Spółka z ograniczoną odpowiedzialnością"
  odwolujacy.adres.ulica: "ul. Mokra 12"
  odwolujacy.adres.kod_pocztowy: "00-100"
  odwolujacy.adres.miasto: "Warszawa"
  odwolujacy.email: "biuro@budpol.pl"
  odwolujacy.telefon: "+48 22 123 45 67"
  odwolujacy.nip: "5252001234"
  odwolujacy.regon: "123456789"
  odwolujacy.krs: "0000123456"
  odwolujacy.bae: null
  odwolujacy.epuap: "/budpol/SkrytkaESP"

  # — SEKCJA 2: REPREZENTACJA —
  reprezentacja.zasady: "Do składania oświadczeń woli w imieniu Spółki upoważ"
```

```
niony jest każdy Członek Zarządu samodzielnie"
  reprezentacja.org: "ZARZĄD"
  reprezentacja.podpisujacy.imie_nazwisko: "Jan Kowalski"
  reprezentacja.podpisujacy.typ_podpisu: "kwalifikowany"
  reprezentacja.pelnomocnictwo_wymagane: false
```

```
# — SEKCJA 3: ZAMAWIAJĄCY —
zamawiajacy.nazwa: "Gmina Przykładowo"
zamawiajacy.adres.miasto: "Przykładowo"
zamawiajacy.nip: "1234567890"
```

```
# — SEKCJA 4: WERYFIKACJA FORMALNA —
weryfikacja.numer_ogloszenia: "487788-2025"
weryfikacja.data_czynnosci: "2025-12-30"
weryfikacja.termin_ustawowy: "10 dni (art. 515 ust. 1 pkt 1 lit. a PZP)"
weryfikacja.termin_wyliczony: "2026-01-09"
weryfikacja.wniesione_w_terminie: true
weryfikacja.wpis_prawidlowy: true
weryfikacja.dowod_przekazania_kopii: true
weryfikacja.braki_formalne: false
```

### 3. Specyfikacja pól

#### 3.1. Pola nagłówkowe pliku

Pole	Typ	Wymagane	Opis
version	integer	TAK	Wersja formatu (aktualnie: 2)
min_similarity	float	NIE	Minimalny próg podobieństwa dla porównań tekstowych (0.0–1.0, domyślnie: 0.55)

#### 3.2. Sekcja assert

Sekcja assert zawiera ścieżkowe asercje w formacie klucz: wartość, gdzie klucz jest ścieżką pola oddzieloną kropkami.

## 4. Kompletna lista pól

### 4.1. Nagłówek (naglowek.\*)

Ścieżka	Typ	Opis
naglowek.sygnatura	string	Sygnatura akt KIO
naglowek.braki_formalne	boolean	Czy stwierdzono braki formalne

### 4.2. Informacje ogólne (informacje\_ogolne.\*)

Ścieżka	Typ	Opis
informacje_ogolne.zamowienie	string	Typ i nazwa zamówienia
informacje_ogolne.tryb_postepowania	string	Tryb postępowania
informacje_ogolne.publikator	string	Publikator (TED/BZP z numerem)
informacje_ogolne.czynnosc	string	Zaskarżona czynność z datą
informacje_ogolne.termin_na_wniesienie	string (date)	Obliczony termin na wniesienie
informacje_ogolne.wniesione_w_terminie	boolean	Czy wniesione w terminie
informacje_ogolne.wpis.wymagane	float	Wymagana kwota wpisu (PLN)
informacje_ogolne.wpis.wniesione	float	Wniesiona kwota wpisu (PLN)
informacje_ogolne.wpis.prawidlowy	boolean	Czy wpis prawidłowy

### 4.3. Odwołujący (odwolujacy.\*)

Ścieżka	Typ	Opis
odwolujacy.konsorcjum	boolean	Czy odwołujący to konsorcjum
odwolujacy.nazwa	string	Nazwa podmiotu
odwolujacy.forma_prawna	string	Forma prawna
odwolujacy.adres.ulica	string	Ulica
odwolujacy.adres.kod_pocztowy	string	Kod pocztowy
odwolujacy.adres.miasto	string	Miasto
odwolujacy.email	string   null	Adres email
odwolujacy.telefon	string   null	Numer telefonu
odwolujacy.nip	string   null	NIP

Ścieżka	Typ	Opis
odwolujacy.regon	string   null	REGON
odwolujacy.krs	string   null	KRS
odwolujacy.bae	string   null	Adres do doręczeń elektronicznych
odwolujacy.epuap	string   null	Adres ePUAP

#### 4.4. Reprezentacja (reprezentacja.\*)

Ścieżka	Typ	Opis
reprezentacja.zasady	string	Zasady reprezentacji (z KRS/CEIDG)
reprezentacja.organ	string	Nazwa organu reprezentacji
reprezentacja.podpisujacy.imie_nazwisko	string	Osoba podpisująca odwołanie
reprezentacja.podpisujacy.typ_podpisu	string	Typ podpisu (kwalifikowany / zaufany)
reprezentacja.pelnomocnictwo_wymagane	boolean	Czy pełnomocnictwo jest wymagane

#### 4.5. Zamawiający (zamawiajacy.\*)

Ścieżka	Typ	Opis
zamawiajacy.nazwa	string	Nazwa zamawiającego
zamawiajacy.adres.ulica	string   null	Ulica
zamawiajacy.adres.miasto	string	Miasto
zamawiajacy.nip	string   null	NIP zamawiającego

#### 4.6. Weryfikacja formalna (weryfikacja.\*)

Ścieżka	Typ	Opis
weryfikacja.numer_ogloszenia	string	Numer ogłoszenia TED/BZP
weryfikacja.data_czynnosci	string (date)	Data zaskarżonej czynności
weryfikacja.termin_ustawowy	string	Termin ustawowy z podstawą prawną
weryfikacja.termin_wyliczony	string (date)	Wyliczony termin na odwołanie
weryfikacja.wniesione_w_terminie	boolean	Ocena terminowości

Ścieżka	Typ	Opis
weryfikacja.wpis_prawidlowy	boolean	Ocena prawidłowości wpisu
weryfikacja.dowod_przekazania_kopii	boolean   null	Czy załączono dowód przekazania kopii
weryfikacja.braki_formalne	boolean	Ostateczna ocena braków formalnych

## 5. Typy wartości

Typ	Opis	Przykład
string	Tekst	"BUD-POL Sp. z o.o."
string (date)	Data w formacie ISO	"2026-01-09"
boolean	Wartość logiczna	true / false
float	Liczba zmiennoprzecinkowa	15000.00
null	Brak danych	null

## 6. Reguły porównywania

### 6.1. Wartości tekstowe

- Porównanie z uwzględnieniem progu `min_similarity` (domyślnie 0.55).
- Algorytm: znormalizowana odległość Levenshteina lub token-set ratio.
- Ignorowane różnice: wielkość liter, nadmiarowe spacje, polskie znaki diakrytyczne (normalizacja Unicode NFKD).

### 6.2. Wartości liczbowe

- Dokładne porównanie wartości numerycznej (tolerancja:  $\pm 0.01$  dla kwot).

### 6.3. Wartości logiczne

- Dokładne porównanie (true  $\neq$  false).

### 6.4. Wartości null

- null w ground truth oznacza, że pole powinno być puste / nieznalezione.
- Jeśli system zwróci wartość dla pola oczekiwanego jako null — to rozbieżność.

# Załącznik E — Specyfikacja raportu accuracy

## 1. Opis

Niniejszy załącznik opisuje format raportu generowanego przez skrypt `evaluate_accuracy.py` (Deliverable D9). Raport służy do obiektywnej oceny jakości przetwarzania odwołań przez system.

## 2. Schemat JSON raportu

```
{
  "report_metadata": {
    "generated_at": "2026-02-09T14:00:00Z",
    "script_version": "1.0.0",
    "url": "http://localhost:5678",
    "test_dir": "./test_cases/",
    "workflow_version": "1.1.0"
  },
  "summary": {
    "accuracy": 0.92,
    "total_fields": 800,
    "matching_fields": 736,
    "discrepancies": 64,
    "cases_total": 20,
    "cases_passed": 16,
    "cases_failed": 4,
    "processing_errors": 0
  },
  "per_case_results": [
    {
      "case_id": "case_001",
      "accuracy": 0.95,
      "total_fields": 40,
      "matching_fields": 38,
      "discrepancies": 2,
      "discrepancy_details": [
        {
          "field": "odwolujacy.telefon",
          "expected": "+48 22 123 45 67",
          "actual": "+48221234567",
          "match_type": "similarity",
          "similarity_score": 0.48
        },
        {
          "field": "odwolujacy.bae",

```

```

        "expected": null,
        "actual": "AE:PL-12345-67890-ABCDE-12",
        "match_type": "exact",
        "similarity_score": 0.0
    }
],
"processing_time_s": 45.2,
"status": "completed"
}
],

"top10_fields": [
{
    "rank": 1,
    "field": "reprezentacja.podpisujacy.imie_nazwisko",
    "discrepancy_count": 14,
    "discrepancy_rate": 0.70,
    "cases_affected": ["case_001", "case_002", "case_003", "case_005", "case_006", "case_008", "case_010", "case_012", "case_014", "case_015", "case_017", "case_018", "case_019", "case_020"]
},
{
    "rank": 2,
    "field": "odwolujacy.email",
    "discrepancy_count": 10,
    "discrepancy_rate": 0.50,
    "cases_affected": ["case_002", "case_004", "case_006", "case_007", "case_009", "case_011", "case_013", "case_015", "case_017", "case_019"]
},
{
    "rank": 3,
    "field": "weryfikacja.termin_ustawowy",
    "discrepancy_count": 8,
    "discrepancy_rate": 0.40,
    "cases_affected": ["case_003", "case_005", "case_007", "case_010", "case_012", "case_014", "case_016", "case_020"]
}
],

"top10_documents": [
{
    "rank": 1,
    "case_id": "case_007",
    "accuracy": 0.45,
    "total_fields": 40,
    "discrepancies": 22,
    "notes": "Skan niskiej jakości, nieczytelne fragmenty"
},
{
    "rank": 2,

```

```

    "case_id": "case_003",
    "accuracy": 0.625,
    "total_fields": 40,
    "discrepancies": 15,
    "notes": "Nietypowy format odwołania (tabelaryczny)"
  },
  {
    "rank": 3,
    "case_id": "case_009",
    "accuracy": 0.70,
    "total_fields": 40,
    "discrepancies": 12,
    "notes": "Brak podpisu elektronicznego w pliku"
  }
],

"history_reference": {
  "history_file": "./accuracy_history.json",
  "total_runs": 5,
  "trend": "improving",
  "previous_accuracy": 0.85,
  "accuracy_delta": 0.07
}
}

```

### 3. Opis sekcji raportu

#### 3.1. Metadane raportu (report\_metadata)

Pole	Typ	Opis
generated_at	string (ISO 8601)	Data i czas wygenerowania raportu
script_version	string	Wersja skryptu ewaluacyjnego
n8n_url	string	URL instancji n8n użytej w teście
test_dir	string	Ścieżka do katalogu ze zbiorami testowymi
workflow_version	string	Wersja workflow n8n (jeśli dostępna)

#### 3.2. Podsumowanie (summary)

Pole	Typ	Opis
accuracy	float	Globalny wynik accuracy (0.0–1.0)

Pole	Typ	Opis
total_fields	integer	Łączna liczba porównywanych pól
matching_fields	integer	Liczba pól zgodnych
discrepancies	integer	Łączna liczba rozbieżności
cases_total	integer	Łączna liczba zbiorów testowych
cases_passed	integer	Liczba zbiorów z accuracy $\geq$ próg
cases_failed	integer	Liczba zbiorów z accuracy $<$ próg
processing_errors	integer	Liczba zbiorów z błędami przetwarzania

### 3.3. Wyniki per zbiór (per\_case\_results[])

Pole	Typ	Opis
case_id	string	Identyfikator zbioru testowego
accuracy	float	Accuracy dla danego zbioru
total_fields	integer	Liczba porównywanych pól
matching_fields	integer	Pola zgodne
discrepancies	integer	Pola rozbieżne
discrepancy_details[]	array	Szczegóły każdej rozbieżności
processing_time_s	float	Czas przetwarzania (sekundy)
status	string	Status: completed, error, timeout

### 3.4. Szczegóły rozbieżności (discrepancy\_details[])

Pole	Typ	Opis
field	string	Ścieżka pola YAML (np. odwołujacy.nip)
expected	any	Wartość oczekiwana (z ground_truth.yaml)
actual	any	Wartość uzyskana z systemu
match_type	string	Typ porównania: exact, similarity, numeric
similarity_score	float	Wynik podobieństwa (0.0–1.0) dla porównań tekstowych

### 3.5. TOP 10 pól z rozbieżnościami (top10\_fields[])

Pole	Typ	Opis
rank	integer	Pozycja w rankingu (1–10)
field	string	Ścieżka pola YAML
discrepancy_count	integer	Liczba zbiorów z rozbieżnością dla tego pola
discrepancy_rate	float	Odsetek zbiorów z rozbieżnością (0.0–1.0)
cases_affected	array[string]	Lista identyfikatorów zbiorów z rozbieżnością

### 3.6. TOP 10 dokumentów z rozbieżnościami (top10\_documents[])

Pole	Typ	Opis
rank	integer	Pozycja w rankingu (1–10)
case_id	string	Identyfikator zbioru testowego
accuracy	float	Accuracy zbioru
total_fields	integer	Łączna liczba porównywanych pól
discrepancies	integer	Liczba rozbieżności
notes	string	Uwagi dotyczące przyczyn rozbieżności

### 3.7. Odniesienie do historii (history\_reference)

Pole	Typ	Opis
history_file	string	Ścieżka do pliku historii
total_runs	integer	Łączna liczba uruchomień w historii
trend	string	Trend: improving, stable, declining
previous_accuracy	float	Accuracy z poprzedniego uruchomienia
accuracy_delta	float	Zmiana accuracy vs poprzedni run

## 4. Format pliku historii (accuracy\_history.json)

Plik historii jest tablicą JSON, do której każde uruchomienie skryptu dopisuje nowy element:

```
[  
  {  
    "timestamp": "2026-02-01T10:00:00Z",  
    "accuracy": 0.85,  
    "total_fields": 800,  
    "matching_fields": 680,  
    "discrepancies": 120,  
    "cases_count": 20,  
  }  
]
```

```

"workflow_version": "1.0.0",
"top10_fields": [
  {
    "field": "reprezentacja.podpisujacy.imie_nazwisko",
    "discrepancy_count": 16,
    "discrepancy_rate": 0.80
  }
],
"top10_documents": [
  {
    "case_id": "case_007",
    "accuracy": 0.35,
    "discrepancies": 26
  }
]
}
]

```

#### 4.1. Reguły zarządzania historią

1. Skrypt musi odczytać istniejący plik `accuracy_history.json` (jeśli istnieje).
2. Skrypt musi dopisać nowy element na końcu tablicy.
3. Skrypt NIE MOŻE modyfikować ani usuwać istniejących wpisów historycznych.
4. Jeśli plik nie istnieje — skrypt tworzy nowy z jednym elementem.
5. Format daty: ISO 8601 z UTC (np. `2026-02-09T14:00:00Z`).

## 5. Wymagania dotyczące wyświetlania na stdout

Skrypt musi wypisać podsumowanie na standardowe wyjście w formacie czytelnym dla użytkownika:

---



---

RAPORT ACCURACY – 2026-02-09 14:00:00 UTC

---



---

Globalny accuracy: 92.0% (736/800 pól)  
 Zbiorów testowych: 20  
 Rozbieżności: 64  
 Trend: ↑ +7.0% vs poprzedni run (85.0%)

— TOP 10 PÓL Z ROZBIEŻNOŚCIAMI —

#	Pole	Rozb.	%
1	reprezentacja.podpisujacy.imie_nazwisko	14/20	70%
2	odwolujacy.email	10/20	50%
3	weryfikacja.termin_ustawowy	8/20	40%
...			

— TOP 10 DOKUMENTÓW Z ROZBIEŻNOŚCIAMI —

#	Zbiór	Accuracy	Rozbieżności
1	case_007	45.0%	22/40
2	case_003	62.5%	15/40
3	case_019	70.0%	12/40
...			



# Załącznik F — Wzór umowy powierzenia przetwarzania danych osobowych

## UMOWA POWIERZENIA PRZETWARZANIA DANYCH OSOBOWYCH

(DO UZUPEŁNIENIA)

# Załącznik G — Kryteria oceny formalnej odwołania

## 1. Opis

Niniejszy załącznik zawiera szczegółową specyfikację kryteriów weryfikacji formalnej odwołania do KIO, które workflow n8n musi automatycznie oceniać przed wygenerowaniem Karty Odwołania. Kryteria wynikają z wymogów art. 516–518 ustawy z dnia 11 września 2019 r. — Prawo zamówień publicznych (Dz.U. 2019 poz. 2019 ze zm.).

## 2. Kryteria weryfikacji formalnej

### F1. Terminowość wniesienia odwołania

Element	Wartość
<b>Podstawa prawna</b>	Art. 515 ust. 1 PZP
<b>Opis</b>	Odwołanie wniesione w ustawowym terminie od dnia przestania informacji o czynności zamawiającego
<b>Źródło danych</b>	Data podpisu elektronicznego (z DSS — <code>signers[].signing_time</code> ) oraz data czynności zamawiającego (z ekstrakcji odwołania)
<b>Algorytm</b>	1. Ustal datę czynności zamawiającego (z ekstrakcji). 2. Ustal termin ustawowy (5/10/15 dni w zależności od trybu i progu — patrz tabela poniżej). 3. Oblicz datę graniczną. 4. Porównaj z datą podpisu (najwcześniejszy podpis z DSS).
<b>Wynik</b>	spełnione / niespełnione / <code>do_weryfikacji_ręcznej</code>
<b>Tabela terminów (art. 515 PZP):</b>	
Warunek	Termin

Warunek	Termin
Zamówienie powyżej progów UE, powiadomienie elektroniczne	10 dni
Zamówienie powyżej progów UE, powiadomienie inne	15 dni
Zamówienie poniżej progów UE, powiadomienie elektroniczne	5 dni
Zamówienie poniżej progów UE, powiadomienie inne	10 dni
Zaniechanie czynności (powyżej progów)	10 dni
Zaniechanie czynności (poniżej progów)	5 dni

## F2. Wpis od odwołania — prawidłowa kwota

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 PZP, Rozporządzenie Prezesa Rady Ministrów w sprawie wpisu
<b>Opis</b>	Wpis uiszczony w prawidłowej kwocie
<b>Źródło danych</b>	Potwierdzenie przelewu (ekstrakcja kwoty), typ zamówienia (z odwołania/eZamówienia)
<b>Algorytm</b>	1. Ustal typ zamówienia (dostawy/usługi vs roboty budowlane). 2. Ustal czy powyżej/poniżej progów UE. 3. Oblicz wymaganą kwotę wpisu. 4. Porównaj z kwotą na potwierdzeniu przelewu.
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej

### Tabela kwot wpisu:

Typ zamówienia	Poniżej progów UE	Powyżej progów UE
Dostawy / Usługi	7 500 PLN	15 000 PLN
Roboty budowlane	10 000 PLN	20 000 PLN

## F3. Wpis wniesiony przed terminem

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 PZP
<b>Opis</b>	Wpis wniesiony najpóźniej w dniu upływu terminu do wniesienia odwołania
<b>Źródło danych</b>	Data przelewu (z potwierdzenia), termin na wniesienie (obliczony w F1)
<b>Algorytm</b>	Porównanie: data przelewu ≤ data graniczna

Element	Wartość
	z F1
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej

#### F4. Podpis kwalifikowany lub zaufany

Element	Wartość
<b>Podstawa prawna</b>	Art. 508 ust. 1 PZP
<b>Opis</b>	Odwołanie wniesione w formie elektronicznej (podpis kwalifikowany) lub w postaci elektronicznej opatrzone podpisem zaufanym
<b>Źródło danych</b>	Weryfikacja DSS — is_valid, signers[].is_qualified
<b>Algorytm</b>	1. Sprawdź czy is_valid = true. 2. Sprawdź typ podpisu (kwalifikowany/zaufany).
<b>Wynik</b>	spełnione / niespełnione

#### F5. Podpis osoby uprawnionej

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 PZP
<b>Opis</b>	Odwołanie podpisane przez osobę uprawnioną do reprezentacji odwołującego
<b>Źródło danych</b>	Sygnatariusz z DSS (signers[].name) vs skład organu z KRS/CEIDG vs pełnomocnictwo
<b>Algorytm</b>	1. Pobierz imię i nazwisko sygnatariusza z DSS. 2. Sprawdź czy figuruje w składzie organu reprezentacji (KRS/CEIDG). 3. Jeśli nie — sprawdź czy jest pełnomocnictwo. 4. Jeśli pełnomocnictwo jest — sprawdź czy sygnatariusz jest w nim wymieniony.
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej
<b>Uwagi</b>	Porównanie imion i nazwisk z uwzględnieniem polskich znaków diakrytycznych, różnic w zapisie (np. imiona

Element	Wartość
	złożone), normalizacji

## F6. Pełnomocnictwo (jeśli wymagane)

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 PZP
<b>Opis</b>	Jeżeli odwołanie podpisuje pełnomocnik — wymagane jest pełnomocnictwo
<b>Źródło danych</b>	Wynik F5 (czy sygnatariusz jest w KRS/CEIDG), klasyfikacja dokumentów (biblioteka — czy jest plik w kategorii Pełnomocnictwo)
<b>Algorytm</b>	1. Jeśli sygnatariusz jest w składzie organu → nie_dotyczy. 2. Jeśli sygnatariusz nie jest w składzie organu → sprawdź czy jest pełnomocnictwo wśród dokumentów. 3. Jeśli jest pełnomocnictwo → spełnione. 4. Jeśli brak → niespełnione.
<b>Wynik</b>	spełnione / niespełnione / nie_dotyczy

## F7. Opłata skarbową od pełnomocnictwa

Element	Wartość
<b>Podstawa prawna</b>	Ustawa o opłacie skarbowej (17 PLN)
<b>Opis</b>	Jeżeli załączono pełnomocnictwo — wymagany dowód uiszczenia opłaty skarbowej
<b>Źródło danych</b>	Klasyfikacja dokumentów, potwierdzenia przelewów
<b>Algorytm</b>	1. Jeśli nie ma pełnomocnictwa → nie_dotyczy. 2. Jeśli jest pełnomocnictwo → szukaj potwierdzenia przelewu na 17 PLN z tytułem zawierającym „opłata skarbową” lub „pełnomocnictwo”.
<b>Wynik</b>	spełnione / niespełnione / nie_dotyczy / do_weryfikacji_ręcznej

## F8. Dowód przekazania kopii zamawiającemu

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 PZP
<b>Opis</b>	Odwołujący przesyła kopię odwołania zamawiającemu
<b>Źródło danych</b>	Klasyfikacja dokumentów (biblioteka), ekstrakcja z odwołania
<b>Algorytm</b>	Sprawdź czy wśród dokumentów jest dowód przekazania kopii (np. potwierdzenie wysłania ePUAP, potwierdzenie nadania, wydruk e-mail)
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej

## F9. Wskazanie czynności lub zaniechania zamawiającego

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 pkt 1 PZP
<b>Opis</b>	Odwołanie musi wskazywać czynność lub zaniechanie czynności zamawiającego, której zarzuca się niezgodność z przepisami ustawy
<b>Źródło danych</b>	Ekstrakcja z odwołania (biblioteka) — pole „czynność” / „zaniechanie”
<b>Algorytm</b>	Sprawdź czy ekstrakcja z odwołania zawiera niepuste pole opisujące czynność lub zaniechanie
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej

## F10. Wskazanie żądania

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 pkt 3 PZP
<b>Opis</b>	Odwołanie musi zawierać żądanie co do sposobu rozstrzygnięcia
<b>Źródło danych</b>	Ekstrakcja z odwołania (biblioteka)
<b>Algorytm</b>	Sprawdź czy ekstrakcja z odwołania zawiera niepuste pole opisujące żądanie

Element	Wartość
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej


### F11. Wskazanie okoliczności faktycznych i prawnych

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 pkt 4 PZP
<b>Opis</b>	Odwołanie musi zawierać uzasadnienie — związane przedstawienie zarzutów, okoliczności faktycznych i prawnych
<b>Źródło danych</b>	Ekstrakcja z odwołania (biblioteka)
<b>Algorytm</b>	Sprawdź czy ekstrakcja z odwołania zawiera niepuste pole opisujące uzasadnienie/okoliczności
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej

### F12. Dane identyfikacyjne odwołującego

Element	Wartość
<b>Podstawa prawna</b>	Art. 516 ust. 1 PZP
<b>Opis</b>	Odwołanie musi zawierać dane odwołującego: nazwę/imię i nazwisko, adres, NIP/KRS
<b>Źródło danych</b>	Ekstrakcja z odwołania (biblioteka) — role Odwołujący
<b>Algorytm</b>	1. Sprawdź obecność nazwy/imienia i nazwiska. 2. Sprawdź obecność adresu. 3. Sprawdź obecność NIP lub KRS. 4. Zweryfikuj NIP/KRS z danymi z rejestrów (KRS/CEIDG).
<b>Wynik</b>	spełnione / niespełnione / do_weryfikacji_ręcznej

## 3. Statusy wyników weryfikacji

Status	Opis	Działanie systemu
spełnione	Kryterium spełnione —	Oznaczenie  w Karcie Odwołania

Status	Opis	Działanie systemu
niespełnione	dane potwierdzone Kryterium niespełnione — brak formalny	Oznaczenie <b>✗</b> w Karcie Odwołania + dodanie do listy braków w wezwaniu
do_weryfikacji_ręcznej	Nie można jednoznacznie ocenić automatycznie	Oznaczenie <b>⚠</b> w Karcie Odwołania
nie_dotyczy	Kryterium nie ma zastosowania w danej sprawie	Brak oznaczenia / pominięcie

#### 4. Struktura danych wyjściowych oceny formalnej

```
{
  "braki_formalne": false,
  "timestamp": "2026-02-09T12:00:00Z",
  "kryteria": [
    {
      "id": "F1",
      "nazwa": "Terminowość wniesienia odwołania",
      "status": "spełnione",
      "uzasadnienie": "Odwołanie wniesione 2026-01-09, termin upływał 2026-01-09. Odwołanie złożone w terminie.",
      "zrodlo": "DSS: signing_time=2026-01-09T13:20:32Z, Odwołanie > strona: 2 (data czynności: 2025-12-30)",
      "dane_szczegolowe": {
        "data_czynnosci": "2025-12-30",
        "termin_ustawowy_dni": 10,
        "data_graniczna": "2026-01-09",
        "data_wniesienia": "2026-01-09"
      }
    },
    {
      "id": "F5",
      "nazwa": "Podpis osoby uprawnionej",
      "status": "do_weryfikacji_ręcznej",
      "uzasadnienie": "Sygnatariusz 'Jan A. Kowalski' nie został jednoznacznie dopasowany do składu organu. Możliwe warianty: 'Jan Adam Kowalski' (Prezes Zarządu).",
    }
  ]
}
```

```
    "zrodlo": "DSS: signer.name='Jan A. Kowalski', KRS API: member.name='Jan Adam Kowalski'",
    "dane_szczegolowe": {
      "sygnatariusz": "Jan A. Kowalski",
      "kandydaci_krs": ["Jan Adam Kowalski (Prezes Zarządu)"],
      "dopasowanie_score": 0.85
    }
  ],
  "lista_brakow": []
}
```

## 5. Uwagi implementacyjne

1. **Porównanie imion i nazwisk:** algorytm musi uwzględniać normalizację Unicode (NFKD), warianty zapisów (imiona złożone, inicjały), polskie znaki diakrytyczne. Próg podobieństwa:  $\geq 0.85$  (token-set ratio).
2. **Obliczanie terminów:** dni ustawowe to dni kalendarzowe; jeśli termin wypada w sobotę, niedzielę lub święto — przesuwa się na najbliższy dzień roboczy (art. 115 Kodeksu cywilnego w zw. z art. 8 ust. 4 PZP).
3. **Weryfikacja wpisu:** jeśli opcjonalna integracja z NBP B2B / Symfonią nie jest aktywna, system weryfikuje wpis wyłącznie na podstawie danych z potwierdzenia przelewu (ekstrakcja).
4. **Status do weryfikacji ręcznej:** system musi stosować ten status, gdy automatyczna weryfikacja nie daje jednoznacznego wyniku — zamiast ryzykować błędne niespełnione.