BPMN i wybór silnika

Elementy BPMN po ludzku i porównanie silników z licencjami.

- BPMN po ludzku i wybór silnika: Camunda 8, Flowable i reszta

BPMN po ludzku i wybór silnika: Camunda 8, Flowable i reszta

Poziom: podstawy

Dotyczy: BPMN 2.0, DMN, Camunda 8.10 (Zeebe, Operate, Tasklist, Optimize, Camunda Hub z Web Modelerem, Identity, Connectors), Flowable 8.0 i Flowable UI 6.8.0, Camunda 7 i jej rozwidlenia, n8n, Temporal

Czas: ~30 min

Ta strona uczy czytać diagram BPMN bez znajomości programowania: zdarzenia, zadania, bramki, timery i tabele decyzyjne DMN. Na dwóch procesach Zielonej pokazuje, co z czego się składa, a potem porównuje Camundę 8 z Flowable i z narzędziami, które tylko wyglądają podobnie. Wracamy tu, gdy ktoś przynosi diagram do przeglądu albo pyta, który silnik wybrać.

Skróty użyte na tej stronie (16)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
B2B	business to business: sprzedaż i systemy skierowane do innych firm, nie do klientów indywidualnych
BPMN	Business Process Model and Notation: notacja graficzna do rysowania procesów biznesowych, którą wykonuje silnik Camundy
CIB	Cluster Information Base: plik XML, w którym Pacemaker trzyma całą konfigurację klastra i jego aktualny stan; polecenie pcs cluster cib zapisuje go do pliku, a cib-push wgrywa zmiany z powrotem
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
CMMN	Case Management Model and Notation: notacja OMG do spraw, w których kolejność kroków nie jest z góry ustalona; wykonuje ją między innymi Flowable
DMN	Decision Model and Notation: notacja OMG do zapisu decyzji w tabelach, które silnik procesów sprawdza wiersz po wierszu, na przykład kto akceptuje fakturę według kwoty
ISO	International Organization for Standardization: organizacja, która wydaje normy techniczne używane na całym świecie

Skrót	Co znaczy
OIDC	OpenID Connect: protokół logowania przez zewnętrznego dostawcę tożsamości, na którym opiera się funkcja SSO w Vaultwardenie
OMG	Object Management Group: organizacja standaryzacyjna, która rozwija BPMN, DMN i UML
RDBMS	Relational Database Management System: relacyjny system baz danych, na przykład PostgreSQL; w Camundzie 8.10 jedna z opcji magazynu wtórnego
REST	sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu
WMS	Warehouse Management System: system zarządzania magazynem
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS
XML	eXtensible Markup Language: format tekstowy do zapisu ustrukturyzowanych danych w znacznikach, tutaj używany przez PowerShell do zapisu zrzutu obiektów poleceniem Export-Clixml
XOR	Exclusive OR: w BPMN bramka wyłączna, która z kilku dróg wybiera dokładnie jedną według warunku

Fiszka

Komendy wykonujemy z CLI bash na stacji administratora albo na dowolnym linuksie z `curl` i `xmllint` (pakiet `libxml2`). Adresy w przykładach to instalacje z tej książki: Camunda na `bpm.zielona.example` (logowanie Basic ze strony o instalacji) i Flowable REST, który opisuje strona o Flowable jako wariant bez interfejsu.

□□ TYLKO ODCZYT

Sprawdzenie, czy plik BPMN jest poprawnym XML: xmllint --noout

```
xmllint --noout NazwaProcesu.bpmn
```

Brak wyniku oznacza poprawny XML. To nie sprawdza logiki procesu, tylko to, czy plik w ogóle da się wczytać, więc robimy to przed każdym wdrożeniem pliku poprawianego ręcznie.

Przykład:

```
zamowienie-b2b.bpmn:41: parser error : Opening and ending tag mismatch: userTask line 37
and process
```

□□ **BPMN** = Business Process Model and Notation: notacja graficzna do rysowania procesów biznesowych, którą wykonuje silnik Camundy · **XML** = eXtensible Markup Language: format

tekstowy do zapisu ustrukturyzowanych danych w znacznikach, tutaj używany przez PowerShell do zapisu zrzutu obiektów poleceniem Export-Clixml · **xmllint** = Narzędzie do czytania i przeszukiwania plików XML; używamy go z wyrażeniem XPath do wyciągnięcia jednej liczby z wyniku crm_mon w skrypcie dla Zabbixa

□□ TYLKO ODCZYT

Odczyt identyfikatora procesu z pliku: grep process id

```
grep -o '<bpmn:process id="[^"]*"' NazwaProcesu.bpmn
```

Identyfikator procesu (nie nazwa wyświetlana) jest kluczem, którym uruchamiamy egzemplarze w obu silnikach.

Przykład:

```
<bpmn:process id="zamowienie-b2b"
```

□□ **grep** = global regular expression print: szukaj wzorca w tekście

□□ TYLKO ODCZYT

Wdrożenie pliku BPMN do Camundy 8: curl POST /v2/deployments

```
curl -u nazwa_usera -F "resources=@NazwaProcesu.bpmn"
https://bpm.zielona.example/orchestration/v2/deployments
```

Wysyła plik do Orchestration Cluster przez REST API w wersji 2; pole formularza nazywa się `resources` według specyfikacji API 8.10. Ścieżka `/orchestration` wynika z naszego `contextPath`, a przy logowaniu OIDC zamiast `-u` podajemy token.

Przykład (skrócony):

```
{"tenantId":"<default>","deploymentKey":"2251799813685249","deployments":[{"processDefin
ition":{"processDefinitionId":"zamowienie-b2b","processDefinitionVersion":1, ...}}]}
```

□□ **BPMN** = Business Process Model and Notation: notacja graficzna do rysowania procesów biznesowych, którą wykonuje silnik Camundy · **REST** = sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu · **API** = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą · **OIDC** = OpenID Connect: protokół logowania przez zewnętrznego dostawcę tożsamości, na którym opiera się funkcja SSO w Vaultwardenie · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

TYLKO ODCZYT

Uruchomienie egzemplarza procesu w Camundzie 8: curl POST /v2/process-instances

```
curl -u nazwa_usera -H "Content-Type: application/json" -d
'{"processDefinitionId":"IdentyfikatorProcesu","variables":{"odbiorca":"NazwaOdbiorcy"}}
' https://bpm.zielona.example/orchestration/v2/process-instances
```

Bez numeru wersji Camunda bierze najnowszą aktywną wersję procesu. Zmienne trafiają do egzemplarza i są widoczne w Operate.

curl = client URL: pobieranie i wysyłanie przez HTTP z terminala

TYLKO ODCZYT

Wdrożenie pliku BPMN do Flowable REST: curl POST repository/deployments

```
curl -u rest-admin -F "file=@NazwaProcesu.bpmn"
https://flowable.zielona.example/flowable-rest/service/repository/deployments
```

Ta sama czynność w Flowable: aplikacja `flowable-rest` ma ścieżkę `/flowable-rest`, a API silnika procesów leży pod `/service`. Konto `rest-admin` to domyślny administrator obrazu, którego hasło zmieniamy przy instalacji.

BPMN = Business Process Model and Notation: notacja graficzna do rysowania procesów biznesowych, którą wykonuje silnik Camundy · **REST** = sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu · **API** = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

Z życia wzięte

Z ŻYCIA WZIĘTE

Czwartek, 14:00, sala na pierwszym piętrze. Pani Halina rysuje na tablicy obieg faktury kosztowej: „przychodzi faktura, sprawdzam, kierownik podpisuje, a jak powyżej pięćdziesięciu tysięcy, to jeszcze pan Zbigniew”. Ola przerysowuje to w modelerze i pyta, co się dzieje, gdy

kierownik nie podpisze w trzy dni. Zapada cisza, bo dziś „nic się nie dzieje, faktura leży”. Po godzinie na diagramie jest timer, który po trzech dniach przypomina kierownikowi, a po pięciu przekazuje sprawę jego zastępcy, i pani Halina po raz pierwszy widzi swój proces w całości.

Jak to działa

Rozwiń: jak to działa pod spodem

Osiem znaków, które wystarczą na początek

Diagram BPMN czyta się od lewego okręgu do prawego. Okręgi to rzeczy, które się dzieją (zdarzenia), prostokąty to praca do wykonania (zadania), romby to rozwidlenia drogi (bramki). Ikonka w środku elementu mówi, jakiego jest rodzaju.



Rysunek 1. Osiem znaków BPMN: zdarzenie startowe, zadanie użytkownika, zadanie serwisowe, reguła biznesowa DMN, bramka wyłączna XOR, bramka równoległa AND, timer i

zdarzenie końcowe (kliknięcie otwiera pełny rozmiar)

Pojęcie: zdarzenie. Coś, co się zdarza i na co proces reaguje: przyszła wiadomość, minął czas, wystąpił błąd. Zdarzenie startowe zaczyna proces, końcowe go kończy, a pośrednie stoi w środku drogi albo na brzegu zadania.

Pojęcie: zadanie użytkownika. Krok, który wykonuje człowiek. Silnik tworzy go na liście zadań (w Camundzie to Tasklist), przydziela osobie albo grupie i czeka, aż ktoś je zamknie, zwykle wypełniając formularz.

Pojęcie: zadanie serwisowe. Krok, który wykonuje program: wywołanie REST, zapis do bazy, wysłanie maila. W Camundzie 8 robi to osobny proces zwany pracownikiem (job worker) albo gotowy konektor, w Flowable najczęściej klasa w Javie albo wyrażenie.

Pojęcie: bramka. Romb, w którym droga się rozwidla albo łączy. Bramka wyłączna (XOR) wybiera dokładnie jedną drogę według warunku, równoległa (AND) puszcza wszystkie drogi naraz, a przy złączeniu czeka, aż każda dojdzie.

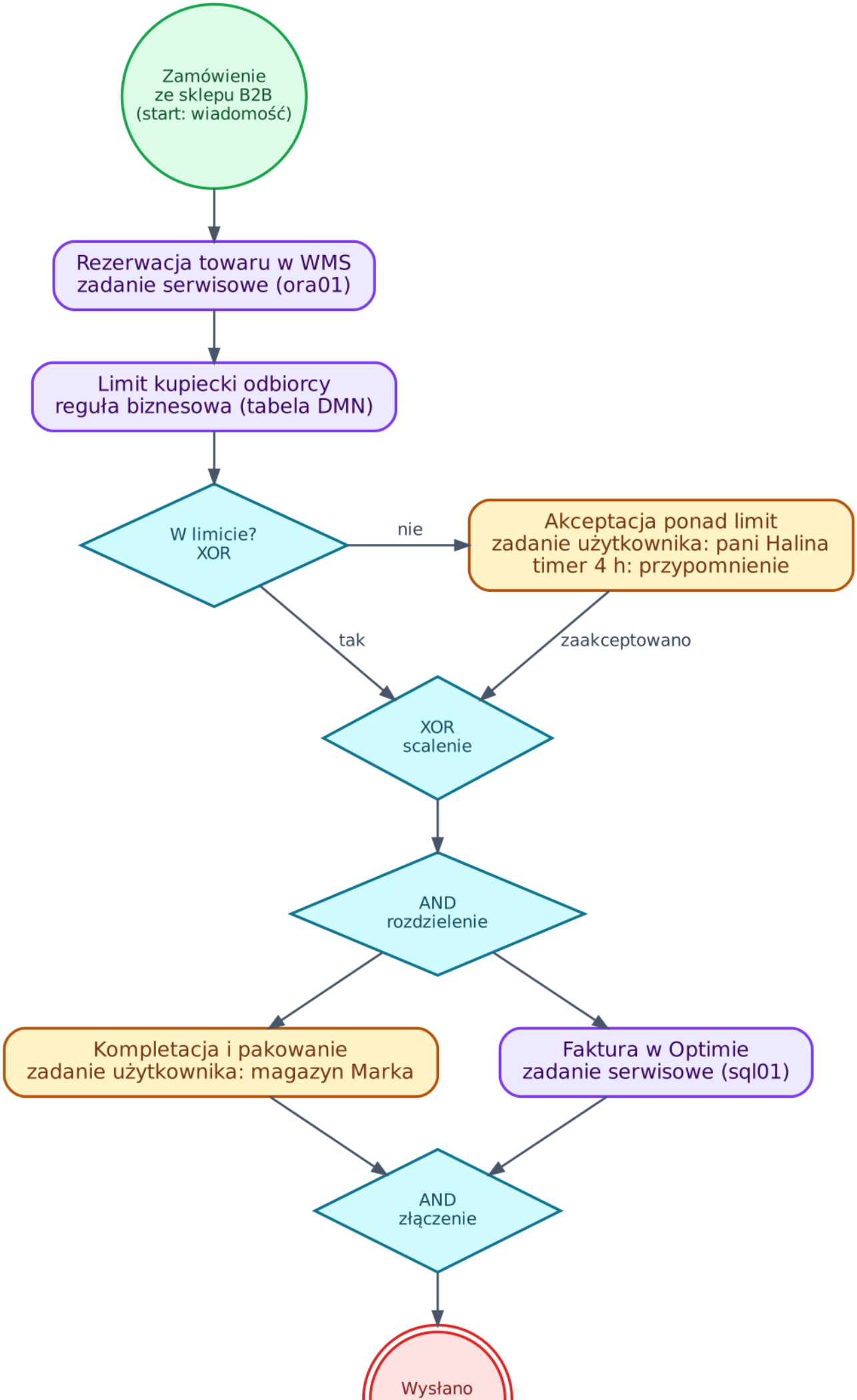
Pojęcie: timer. Zdarzenie, które zachodzi o określonej porze albo po określonym czasie. Przypięty do brzegu zadania działa jak budzik: „jeśli zadanie trwa dłużej niż 4 godziny, przypomnij”. Czas zapisujemy w formacie ISO 8601, na przykład `PT4H` to cztery godziny, a `P3D` trzy dni.

Pojęcie: DMN (Decision Model and Notation). Druga notacja OMG, siostra BPMN, do zapisywania decyzji w tabelach. Wiersz tabeli to reguła „jeśli kwota do 5000 zł, akceptuje kierownik działu”, a silnik sprawdza wiersze i zwraca wynik. Tabelę może zmieniać księgowość bez poprawiania diagramu.

Pojęcie: FEEL. Język prostych wyrażeń używany w DMN i w warunkach na diagramach Camundy: `kwota > 50000`, `[5000..50000]`, `kategoria = "transport"`. Czyta się go prawie jak zdanie.

Proces pierwszy: zamówienie hurtowe od odbiorcy B2B do wysyłki

Zamówienie przychodzi ze sklepu B2B (baza `sklep_b2b` na `my01`) jako wiadomość, która startuje proces. Program rezerwuje towar w WMS na `ora01`, a tabela DMN sprawdza limit kupiecki odbiorcy. Gdy zamówienie mieści się w limicie, idzie dalej od razu, a gdy nie, pani Halina dostaje zadanie z timerem przypominającym po czterech godzinach.


Zamówienie
ze sklepu B2B
(start: wiadomość)
Rezerwacja towaru w WMS
zadanie serwisowe (ora01)
Limit kupiecki odbiorcy
reguła biznesowa (tabela DMN)
W limicie?
XOR
nie
Akceptacja ponad limit
zadanie użytkownika: pani Halina
timer 4 h: przypomnienie
tak
zaakceptowano
XOR
scalenie
AND
rozdzielenie
Kompletacja i pakowanie
zadanie użytkownika: magazyn Marka
Faktura w Optimie
zadanie serwisowe (sql01)
AND
złączenie
Wysłano

Rysunek 2. Proces zamówienia hurtowego narysowany schematem: start wiadomością ze sklepu B2B, rezerwacja w WMS, reguła DMN limitu kupieckiego, bramka XOR z akceptacją pani Haliny, scalenie, bramka AND rozdzielająca kompletację w magazynie Marka i fakturę w Optimie, złączenie i koniec „Wysłano” (kliknięcie otwiera pełny rozmiar)

Po akceptacji dwie rzeczy dzieją się naraz: magazyn Marka kompletuje towar, a program wystawia fakturę w Optimie na `sql01`. Bramka AND na końcu czeka na obie, dopiero potem proces kończy się zdarzeniem „Wysłano”. Brak towaru obsługujemy w wersji drugiej diagramu zdarzeniem błędu na brzegu zadania „Rezerwacja towaru w WMS”, które kieruje sprawę do handlowca z propozycją zamiennika.

Rysunek jest schematem, a nie plikiem z modelera: bramka scalająca XOR przed bramką AND jest tam celowo. Gdyby obie drogi („w limicie” i „po akceptacji”) wchodziły wprost do bramki AND, ta czekałaby na obie i proces stanąłby na zawsze, bo zawsze przychodzi tylko jedna.

Proces drugi: akceptacja faktury kosztowej

Ten proces to prawie same zadania ludzi. Faktura trafia do księgowości, pani Halina sprawdza ją formalnie, a tabela DMN wybiera osobę akceptującą według kwoty i kategorii kosztu.

Kwota (zł)	Kategoria	Akceptujący	Uwagi
`<= 5000`	dowolna	kierownik działu	tylko kierownik zgłaszającego
`[5000..50000]`	dowolna	kierownik działu, potem pani Halina	dwa podpisy
`> 50000`	dowolna	pan Zbigniew	prezes
dowolna	`"transport"`	Marek	koszty magazynu i wysyłek

> **Pojęcie: polityka dopasowania (hit policy).** Zasada, co zrobić, gdy pasuje kilka wierszy tabeli DMN naraz. `UNIQUE` dopuszcza tylko jeden pasujący wiersz, `FIRST` bierze pierwszy od góry, `COLLECT` zbiera wszystkie. Tabela wyżej ma nakładające się wiersze (transport za 60 000 zł pasuje dwa razy), więc przy `UNIQUE` silnik zgłosi błąd i trzeba ją poprawić albo przejść na `FIRST`.

Do zadania „Akceptacja” przypinamy dwa timery: po trzech dniach (`P3D`) przypomnienie mailem, po pięciu (`P5D`) przekazanie zastępcy. Pierwszy nie przerywa zadania, drugi tak.

> **Pojęcie: zdarzenie przerywające i nieprzerywające.** Timer na brzegu zadania może je przerwać (zadanie znika, proces idzie drogą timera) albo nie (zadanie trwa, a obok rusza dodatkowa droga, na przykład mail z przypomnieniem). Na diagramie nieprzerywające ma przerywaną obwódkę.

Camunda 8 z czego się składa

Camunda 8.10 to zestaw aplikacji, z których część jest obowiązkowa, a część dokładamy według potrzeb. Silnik i trzy aplikacje WWW działają w nowych wersjach jako jeden program zwany Orchestration Cluster (od 8.10 nie ma już ich osobnych obrazów), a w 8.10 Web Modeler i Console stały się Camunda Hub.

Składnik	Co robi	Na `bpm01`
Zeebe (w Orchestration Cluster)	silnik: wykonuje procesy, zapisuje dziennik zdarzeń na dysku	tak, jeden broker
Operate (w Orchestration Cluster)	podgląd egzemplarzy, błędów i zmiennych, naprawa utkniętych spraw	tak
Tasklist (w Orchestration Cluster)	lista zadań ludzi z formularzami	tak
Admin (w Orchestration Cluster)	użytkownicy, grupy, role i uprawnienia klastra	tak
Connectors	gotowe zadania serwisowe: REST, poczta, kolejki	tak
Management Identity	logowanie i role dla Optimize i Camunda Hub, połączenie z dostawcą OIDC	na stronie o logowaniu
Camunda Hub (Web Modeler, Console)	modelowanie w przeglądarce, wspólne projekty, wdrażanie	na stronie o logowaniu
Optimize	raporty i analiza czasu procesów	nie, wymaga Elasticsearch albo OpenSearch
Desktop Modeler	modeler na komputer, licencja MIT	na stacjach Oli i pani Haliny

> **Pojęcie: magazyn wtórny (secondary storage).** Baza, z której Operate, Tasklist i API wyszukiwania czytają stan procesów. Zeebe zapisuje wszystko najpierw do swojego dziennika, a potem przepisuje do magazynu wtórnego. W 8.10 może nim być Elasticsearch, OpenSearch albo baza relacyjna (u nas PostgreSQL).

Flowable z czego się składa

Flowable to przede wszystkim biblioteka w Javie: silnik BPMN, silnik DMN i silnik CMMN (sprawy, w których kolejność kroków nie jest z góry ustalona). Programista wkłada ją do własnej aplikacji albo uruchamia gotową aplikację REST.

Składnik	Co robi	Stan w listopadzie 2026
silnik Flowable (biblioteka)	BPMN, DMN, CMMN w aplikacji Javy	8.0.0, Apache 2.0
`flowable/flowable-rest`	silnik z API REST i Swaggerem, bez interfejsu WWW	8.0.0
`flowable/flowable-ui`	modeler, lista zadań, administracja i konta w jednej aplikacji	6.8.0 z grudnia 2022, poza otwartym projektem od 7.0.0
produkty komercyjne Flowable	aplikacje biznesowe i modeler w nowej wersji	warunki sprawdzamy u producenta

Inne: Camunda 7, rozwidlenia i rzeczy, które nie są BPMN

Camunda 7 Community Edition nie dostaje wydań od 7.24 z 14 października 2025 roku, więc nowych procesów na niej nie zaczynamy. Kto ma już procesy w Camundzie 7, może zostać przy rozwidleniach na Apache 2.0 (Operaton, CIB seven, Fluxnova) albo przejść na Camundę 8, co jest projektem migracji, a nie aktualizacją.

Dwa narzędzia wracają w każdej rozmowie o automatyzacji, choć BPMN nie wykonują. Wypisujemy je, żeby było jasne, czym się różnią.

Narzędzie	Czym jest	Licencja (informacyjnie)	Kiedy zamiast silnika BPMN
n8n	automatyzacja w przeglądarce: węzły połączone strzałkami, setki integracji	Sustainable Use License, pliki `.ee.` na licencji Enterprise	proste przepływy „gdy przyjdzie mail, zapisz w arkuszu”, bez zadań ludzi i bez audytu kroków
Temporal	procesy pisane kodem (Go, Java, Python, TypeScript), silnik pilnuje ich trwałości	MIT	zespół programistów, długie procesy w kodzie, brak potrzeby diagramu dla biznesu

Który silnik do czego

Pytanie	Camunda 8	Flowable	n8n, Temporal
Czy pani Halina zobaczy diagram swojego procesu?	tak, w Camunda Hub i w Operate	tak, w Flowable UI	n8n: własny rysunek, nie BPMN; Temporal: nie
Zadania ludzi z formularzami	Tasklist	aplikacja zadań w Flowable UI 6.8 albo własny interfejs nad REST	nie w tej postaci
Tabele DMN	tak	tak	nie
Integracje z WMS i Optimą	konektory i pracownicy w dowolnym języku	Java albo REST	n8n: gotowe węzły; Temporal: kod
Skala i wysoka dostępność	od trzech brokerów, replikacja w Zeebe	baza danych i kilka instancji aplikacji	zależnie od narzędzia
Wymagania na start	Kubernetes, PostgreSQL, sporo pamięci	jeden kontener i PostgreSQL	jeden kontener
Licencja w produkcji	płatna (Enterprise) od 8.6	silnik bezpłatny, aplikacje komercyjne poza 6.8	patrz tabela wyżej

Dla Zielonej wychodzi z tego prosty podział na pilotaż: proces zamówienia, który rozmawia z trzema systemami, próbujemy na Camundzie 8, a akceptację faktur na Flowable. Po pilotażu porównujemy pracę zespołu Oli i koszt licencji, a decyzję zostawiamy autorowi i panu Zbigniewowi.

Częste błędy

Objaw	Przyczyna	Co robimy
Proces staje na bramce AND i nigdy nie rusza dalej	do bramki AND wchodzą drogi, z których przychodzi tylko jedna	przed bramką AND scalamy drogi bramką XOR
Tabela DMN zwraca błąd przy niektórych fakturach	nakładające się wiersze przy polityce `UNIQUE`	poprawiamy zakresy albo zmieniamy politykę na `FIRST`
Proces z modelera Flowable nie wdraża się w Camundzie	rozszerzenia producenta w XML (`flowable:`, `zeebe:`)	traktujemy plik jako szablon i ustawiamy zadania od nowa w modelerze docelowego silnika
Wdrożenie zwraca 401	brak albo zły login przy `curl -u`, a przy OIDC brak tokenu	logowanie ze strony o instalacji albo token od dostawcy OIDC
Timer zadziałał po minucie zamiast po czterech godzinach	`PT4M` zamiast `PT4H` w zapisie ISO 8601	`M` po `T` to minuty, `H` to godziny, `M` przed `T` to miesiące
Egzemplarze stają na zadaniu serwisowym	nikt nie odbiera zadań: nie działa pracownik albo konektor	Operate pokazuje incydent; sprawdzamy, czy działa pod pracownika

Ćwiczenia

1. Na diagramie zamówienia brakuje bramki scalającej XOR, a „w limicie” i „po akceptacji” wchodzą wprost do bramki AND. Co się stanie z zamówieniem w limicie?

Odpowiedź

Bramka AND czeka na wszystkie wchodzące drogi. Zamówienie w limicie przychodzi tylko jedną drogą, więc bramka będzie czekać w nieskończoność na drugą, a egzemplarz utknie. Dlatego drogi wyłączne scalamy bramką XOR, a dopiero potem rozdzielamy bramką AND.

1. Pani Halina chce zmienić próg „powyżej 50 000 zł akceptuje prezes” na 40 000 zł. Czy trzeba poprawiać diagram procesu?

Odpowiedź

Nie, jeśli próg jest w tabeli DMN, a nie w warunku bramki. Zmieniamy jeden wiersz tabeli i wdrażamy nową wersję decyzji; diagram procesu zostaje ten sam.

1. Dlaczego Optimize nie staje na `bpm01` w tej książce?

Odpowiedź

Optimize wymaga Elasticsearch albo OpenSearch, a nasza instalacja próbna używa PostgreSQL jako magazynu wtórnego, żeby nie stawiać dodatkowej wyszukiwarki z dużym apetytem na pamięć. Gdy raporty okażą się potrzebne, dokładamy Elasticsearch albo OpenSearch.

1. Ola proponuje n8n zamiast Camundy, „bo jest prostsze”. Co odpowiadamy w kontekście procesu faktur?

Odpowiedź

n8n dobrze łączy systemy, ale nie wykonuje BPMN, nie ma listy zadań ludzi z formularzami ani tabel DMN, a jego licencja nie jest otwarta. Proces faktur to głównie zadania ludzi z terminami i decyzjami, więc lepiej pasuje do silnika BPMN; n8n może obsłużyć proste powiadomienia obok.

Powiązane: Silniki procesów: od workflow i BPEL do Camundy 8 i Flowable · Camunda 8 Self-Managed przez Helm: instalacja próbna na bpm01 · Flowable w Docker Compose na docker01:

Flowable UI, PostgreSQL i Traefik · Który silnik bazy do czego: przewodnik decyzyjny

Źródła (odczyt 1 listopada 2026): docs.camunda.io, „Licensing” i „What's new in 8.10” (Camunda Hub zastępuje Web Modeler i Console, magazyn wtórny RDBMS); docs.camunda.io, „RDBMS example deployment for Camunda with Helm” (Optimize wymaga Elasticsearch albo OpenSearch); github.com/camunda/camunda, gałąź `stable/8.10`, `zeebe/gateway-protocol/src/main/proto/v2/deployments.yaml` i `process-instances.yaml` (pole `resources`, `processDefinitionId`, `variables`); docs.camunda.io, „Orchestration Cluster REST API” (adres `/v2/`); github.com/flowable/flowable-engine, tagi `flowable-6.8.0` i `flowable-8.0.0`, pliki `docker/README.md` i `flowable-default.properties` (ścieżka `/flowable-rest`, `/service`, konto `rest-admin`); wz-it.com, artykuł o końcu Camundy 7 z 15 października 2026 (rozwidlenia); github.com/n8n-io/n8n, `LICENSE.md`; github.com/temporalio/temporal, `LICENSE`. Obsługę CMMN w Camundzie 8 pomijamy, bo jej nie sprawdzaliśmy.