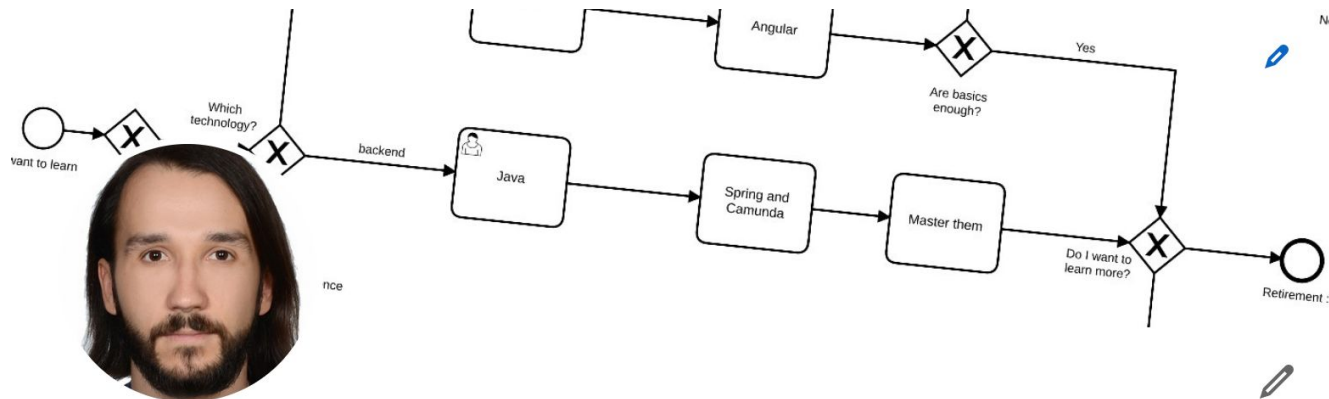

CAMUNDA

NobleProg

O sobie



Tomasz Młynarczyk

Team Lead/Software Architect w Trans.eu Group

Kraków, Woj. Małopolskie, Polska · [Informacje kontaktowe](#)

[121 kontaktów](#)



Trans.eu Group



Politechnika Wroclawska



AGENDA

Camunda (BPMN, DMN, procesy)

Część użytkowa (silnik, ćwiczenia, obsługa błędów)

Dobre praktyki, patterny, wdrożenia produkcyjne

Camunda 7 a Camunda 8

Dzień pierwszy

Możliwości Camundy

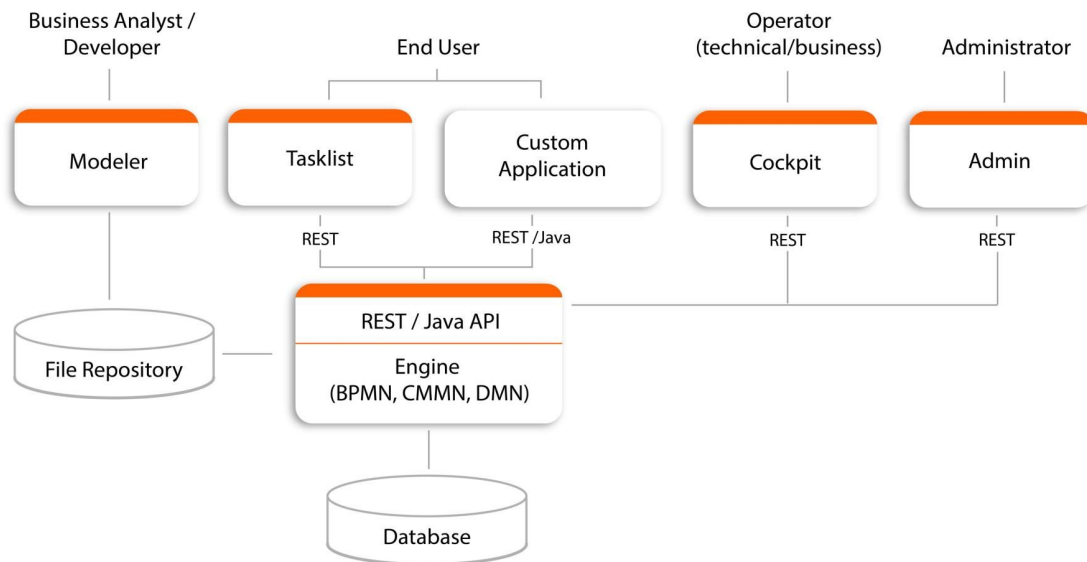
End-to-end Orchestration

Standards-based Business-IT Collaboration

Developer-friendly Approach

Open Architecture








Jakie problemy Camunda rozwiązuje (i jakich dostarcza)



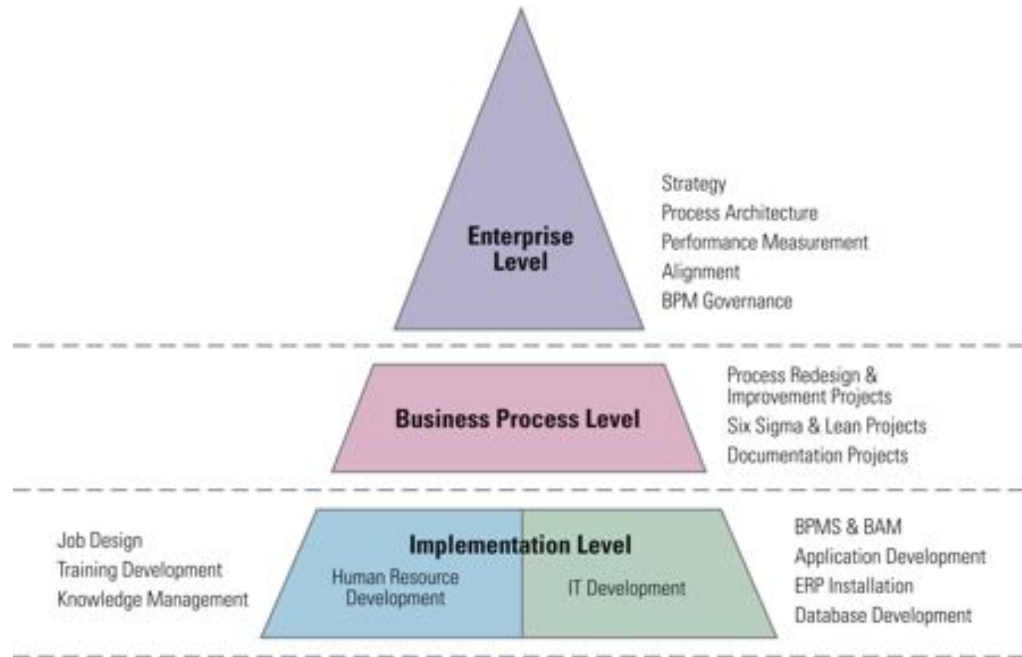
Model

Execute

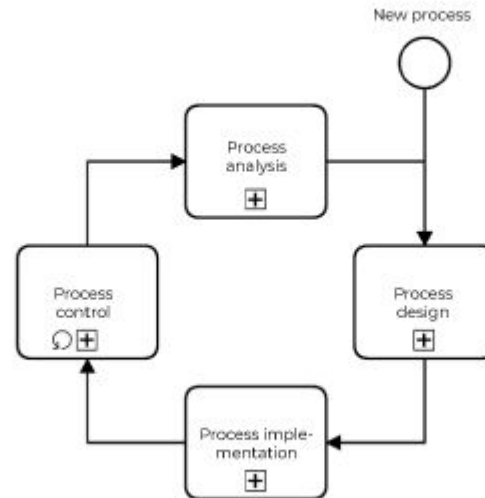
Community vs Enterprise

	Camunda Community	Camunda Enterprise
Design		
 Modeler	✓	✓
 Camunda	✓ basic version	✓ extended version
Automate		
 Cockpit	✓ basic version	✓ extended version
 Workflow Engine	✓	✓
 Tasklist	✓	✓
 Decision Engine	✓	✓
Improve		
 Optimize	✗	✓

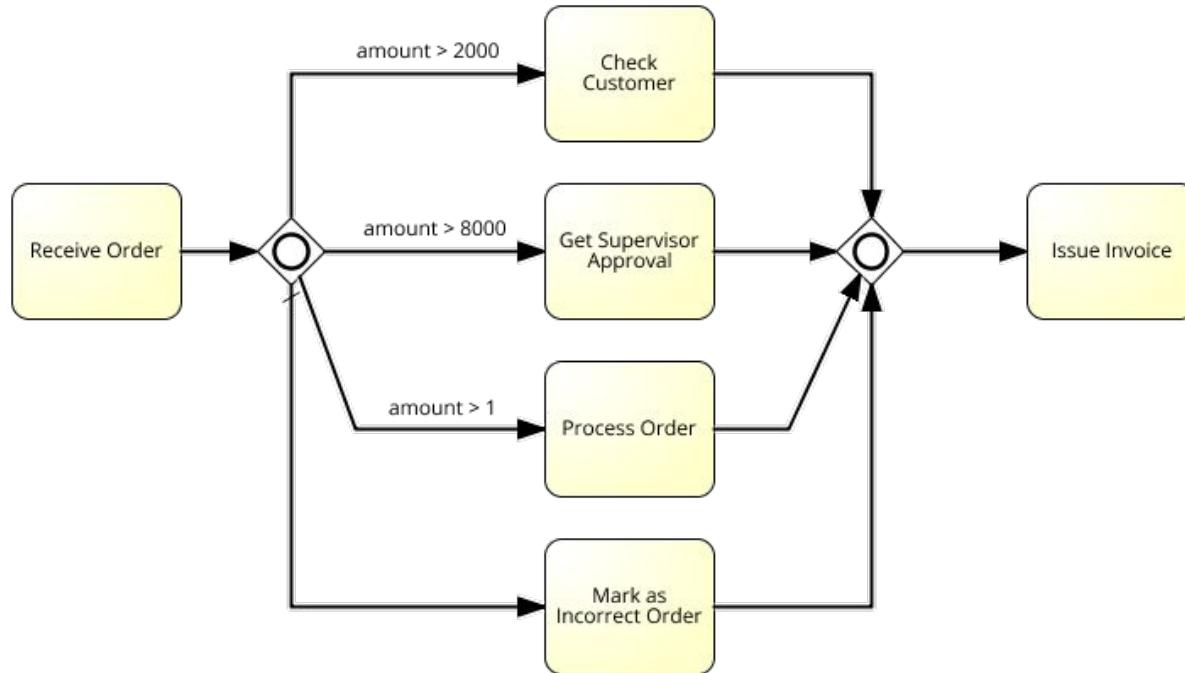
Poziomy implementacji BPMN
























Od pomysłu do implementacji



Od pomysłu do wykresu, czyli jak działa BPMN



Notacja BPMN 2.0 i DMN

	Start			Intermediate				End
Type	Normal	Event Sub process	Event Sub process non-interrupt	Catch	Boundary	Boundary non-interrupt	Throw	
None								
Message								
Timer								
Conditional								

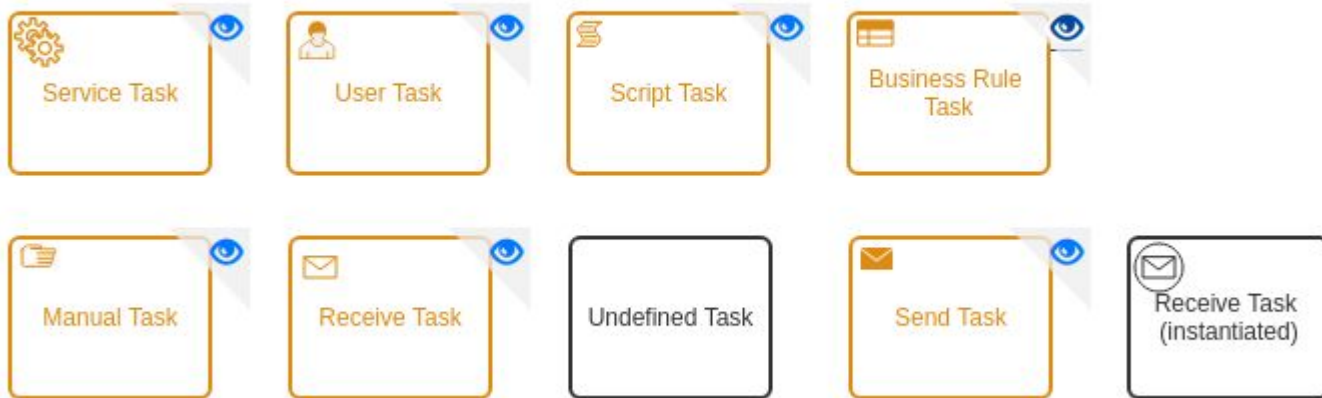
Gateways

Gateways



Tasks

Tasks



Eventy (messaging, timery)



Message



Link



Escalation



Timer



Compensation



Start



Catch event



Signal



End

Camunda Cockpit



Camunda Cockpit

Processes

Decisions

Human Tasks

More ▾

[Dashboard](#) » [Processes](#) » T4 - T6 Access to offers synchronization : Runtime

Definition Version:

1

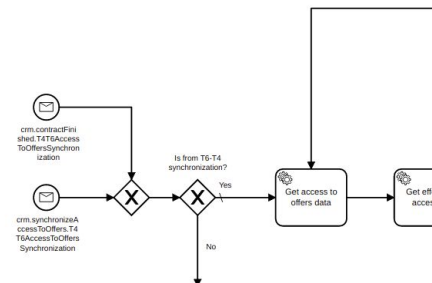
Version Tag:

null

Definition ID:

2fac6fb5-688d-11ec-9b33-465838f...

Definition Key:



Zasady pracy z zadaniem

- będzie potrzebne minimum jedno konto Github per grupa,
 - forkujemy repozytorium podane w linku,
 - po skończeniu zadania wystawiamy MR do głównego repozytorium,
 - jeśli wszystko pójdzie w porządku, pod koniec będziemy mieli 3 kompletne procesy :)
-

Budowanie procesu z opisu biznesowego

- a) Ofertę musi złożyć dział handlowy konkretnej firmy. W tym momencie kierownik handlowy musi przeprowadzić kontrolę wykonalności w oparciu o wymagania biznesowe i potrzeby, aby upewnić się, że oferta jest wykonalna.
 - b) Jeżeli oferta nie jest możliwa do zrealizowania, klient musi zostać o tym poinformowany i w tym momencie proces kończy się po poinformowaniu go.
 - c) Jeśli oferta jest możliwa do zrealizowania, należy sprawdzić potrzeby i dostępność personelu. Po doprecyzowaniu dostępności w oparciu o zasoby biznesowe, oferta musi zostać utworzona przez pracownika komercyjnego i proces kończy się na tworzeniu oferty.
-

Zadanie - budowanie procesu za pomocą modelera

Grupa 1:

- a) Faktura musi zostać sporządzona przez dział rozrachunków z dostawcami (AP). Pierwszym krokiem jest utworzenie dokumentu rozliczeniowego w SAP i sprawdzenie zamówienia za pomocą faktury przez AP Processor.
 - b) Tworząc ten krok, procesor AP musi wybrać liczbę zaksięgowanych dostaw, a następnie wykonać, aby wygenerować dane do faktury.
 - c) Następnie fakturę należy zaksięgować w systemie
 - d) Na tym etapie proces realizacji zamówienia jest zakończony
-

Zadanie - budowanie procesu za pomocą modelera

Grupa 2:

- a) O rejestrację w godzinach nadliczbowych należy poprosić pracownika działu sprzedaży u odpowiedzialnego kierownika.
 - b) Po pierwsze, pracownik musi utworzyć rejestrację w godzinach nadliczbowych, a także musi wysłać tę rejestrację do swojego przełożonego.
 - c) Po otrzymaniu tego wniosku Kierownik ocenia/szacuje wniosek o rejestrację w godzinach nadliczbowych.
 - d) Na tym poziomie Menedżer może albo odrzucić wniosek o rejestrację, a następnie pracownik otrzyma odrzucone zgłoszenie, albo może zatwierdzić wniosek i na tym etapie proces kończy się dla obu wyborów.
-

Zadanie - budowanie procesu za pomocą modelera

Grupa 3:

- a) Po zorientowaniu się, że zatrudnienie osoby na określonym stanowisku jest realizowane przez Kierownika Zatrudnienia, należy stworzyć ogłoszenie o pracę/Zamówienie
 - b) Następnie dział kadr musi znaleźć najlepszych kandydatów na to stanowisko, korzystając z ich CV i listów motywacyjnych.
 - c) Po wybraniu grupy ankietów HR musi zaprosić ankietów na rozmowę kwalifikacyjną.
 - d) HR Manager musi następnie przeprowadzić rozmowy kwalifikacyjne z wybranymi kandydatami. Po przeprowadzeniu rozmowy kwalifikacyjnej, jeśli Kierownik zdecydował się na odpowiednią, kontaktuje się z HR w celu ustalenia Płac oraz kontaktu z Ankietem i przesłania mu oferty/umowy do podpisania przez potencjalnego pracownika. Po podpisaniu umowy zespół HR otrzymuje podpisaną umowę, a proces kończy zatrudnienie osoby na tym poziomie. Jeśli natomiast Menedżer nie znalazł odpowiedniego kandydata pomiędzy ankietami, proces na etapie poszukiwania kandydatów powtarza się.
-

Zadanie - budowanie procesu za pomocą modelera

Grupa 4:

- a) Konserwacja jest wymagana po otrzymaniu awarii sprzętu przez Dział konserwacji. Po otrzymaniu tej wiadomości przez pracownika serwisu należy utworzyć powiadomienie.
 - b) Następnie na podstawie zgłoszeń należy utworzyć zlecenie alimentacyjne.
 - c) W kolejnym kroku pracownik serwisu musi zaplanować/przetworzyć zgłoszenie i zamówienie.
 - d) Następnie prace konserwacyjne należy wykonać w terenie.
 - e) W przypadku, gdy ocena jakości nie została zaliczona, prace konserwacyjne należy wykonać ponownie w terenie. Po przejściu oceny jakości planista konserwacji musi wypełnić zlecenie konserwacji i zamknąć zgłoszenie, a na koniec proces konserwacji jest pomyślnie zakończony.
-

Subproces a call activity

Subprocesses



Subprocesy - zasady podziału

- jeśli jest reużywalny,
 - jeśli proces nadrzędny staje się zbyt skomplikowany,
 - subprocess - jeśli proces może być też niezależny od parenta
 - call-activity - jeśli chcemy uruchomić osobnego BPMNa w tym samym kontekście
-

Ćwiczenie - dodanie prostego subprocesu

Każdy z subprocesów ma zawierać jeden user task do zakończenia.

Proces główny z poprzedniego ćwiczenia ma wywoływać ten subproces.

Serializacja i zasięg zmiennych

- zasięg zmiennych (scopes)
 - propagacja zmiennych (pomiędzy procesami)
 - mapowanie zmiennych
 - dlaczego klucze biznesowe są takie ważne?
 - jakie dane przekazywać do zmiennych?
 - serializacja (Camunda Spin)
 - pobieranie zmiennych w serwisach
-

Camunda Spin

Camunda Spin Dataformat Reference

Camunda Spin is a library for simple XML and JSON processing on the JVM (Java Virtual Machine), targeting Java and JVM-based scripting languages such as Groovy, JRuby, Jython, JavaScript and Java Expression Language. It provides a comprehensible fluent API for working with different data formats through lightweight wrapper objects.

Spin can be used in any Java-based application by adding the following maven dependency to your `pom.xml` file:

Camunda BOM

If you use Spin in combination with the Camunda process engine, please consult the [process engine user guide on Spin integration](#) on how to properly integrate Spin with the engine. Please import the [Camunda BOM](#) to ensure that you use the Camunda Spin version matching your process engine version.

Zmienne procesowe - patterny i przykład

- jak najmniej zmiennych procesowych,
 - używanie typów prostych,
 - w skomplikowanych przypadkach - JSON
 - poleganie na zmiennych globalnych (łatwiejsze testowanie)
 - zmienne lokalne i przekazywanie tylko w przypadku DMNów oraz subprocessów
-



ExchangeClassifierTask

General

Listeners

Input/Output

Field Injections

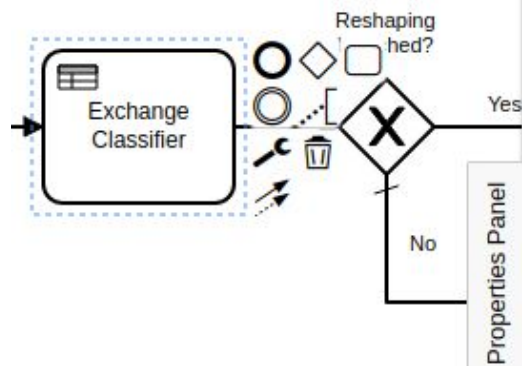
Extensions

Input Parameters

- > **balanceOverdue** ← `execution.getVariableLocal("balanceOverdue") ? execution.getVariableLocal("bala...`
- > **isExchangeBlockedInOrder** ← `${isExchangeBlockedInOrder}`
- > **prepaidEndDate** ← `${exchangePrepaidEndDate}`
- > **contractEndDate** ← `${exchangeContractEndDate}`
- > **exchangePromotionValidTo** ← `${exchangePromotionValidTo}`
- > **currentDate** ← `#{timeProvider.getCurrentLocalDate()}`
- > **hasExchangeOrder** ← `${hasExchangeOrder}`
- > **exchangeAccessSelector** ← `${exchangeAccessSelector}`
- > **isExchangeOrderComponentPresent** ← `${isExchangeOrderComponentPresent}`
- > **isInfinitePromoExchangeAccess** ← `${isInfinitePromoExchangeAccess}`
- > **contractT4ExpirationDate** ← `${contractT4ExpirationDate}`
- > **contractType** ← `${contractType}`

Output Parameters

No variables defined.



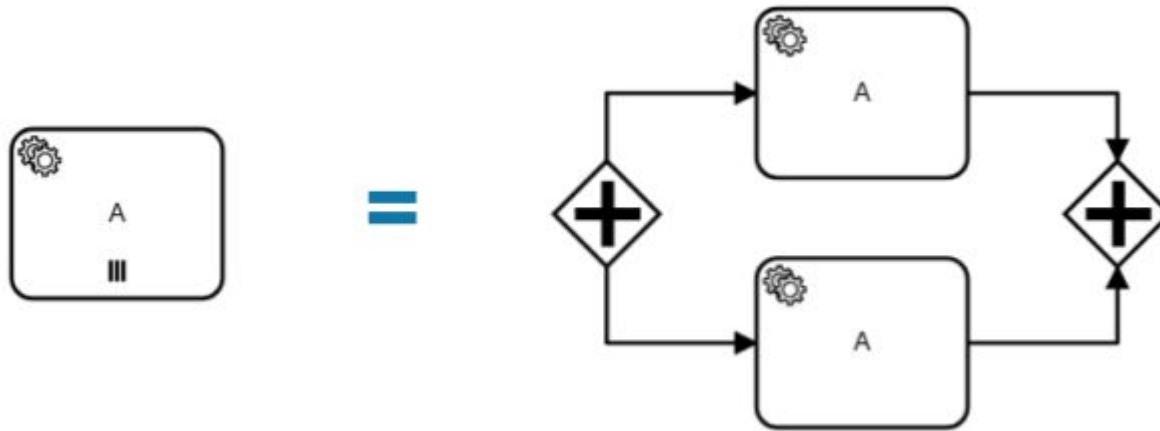
Ćwiczenie - przekazywanie zmiennych

Do poprzedniego subprocesu należy przekazać zmienne podawane z poziomu Camunda Cockpit.

Ograniczamy się do typów prostych.

Dzień drugi

Elementy typu parallel



Ćwiczenie - elementy parallel

W poprzednim procesie należy wykorzystać element typu parallel (user task/subproces).

Listę zmiennych przekazujemy z poziomu CC.

DMN i DRD

Beverages					Show details
C	Input +		Output +	Annotation	
	Dish	Guests with children	Beverages		
1	"Spareribs"	-	"Aecht Schlenkerla Rauchbier"	Tough Stuff	
2	"Stew"	-	"Guinness"	-	
3	"Roastbeef"	-	"Bordeaux"	-	
4	"Steak", "Dry Aged Gourmet Steak", "Light Salad and a nice Steak"	-	"Pinot Noir"	-	
5	-	true	"Apple Juice"	-	
6	-	-	"Water"	-	
+	-	-	-	-	

Podstawy FEEL

<https://docs.camunda.io/docs/components/modeler/feel/what-is-feel/>



Exchange Order Classifier

Hit Policy: First ▼

	asExchangeOrder boolean	And exchangeAccessSele... string	And isExchangeOrderCom... boolean	And isInfinitePromoExcha... boolean	And hasT4ContractDate boolean	And isExchangePrepaidCo... boolean	And contractType string	Then + orderArgumentDescriptor
1		null	false	-	-	-	-	""
2		-	true	-	-	-	-	"exchange{Revoke:[origin=order]]"
3		not("order")	false	-	-	-	-	""
4		"contract"	false	-	-	-	"t4_migrated"	"exchange{Revoke:[origin=contract]],exchange{Gr [origin=order];;" + toString(contractEndDate) + "<";< toString(contractId) + ">}"
5		"order"	true	-	-	-	-	"exchange{Grant:[origin=order];;" + toString(contra <contract_id=" + toString(contractId) + ">}"
6		"prepaid"	-	-	-	-	-	"exchange{Grant:[origin=order];;" + plusDaysAsStri prepaidEndDate, days: 1) + ";;" + toString(contract <contract_id=" + toString(contractId) + ">}"
7		"promotion"	-	true	-	true	-	null
8		"promotion"	true	-	true	false	-	null
9		"promotion"	-	-	-	true	-	"exchange{Grant:[origin=order];;" + plusDaysAsStri prepaidEndDate, days: 1) + ";;" + toString(contract <contract_id=" + toString(contractId) + ">}"
10		"promotion"	-	true	-	-	-	"exchange{Grant:[origin=order];;" + plusDaysAsStri currentDate, days: 2) + ";;" + toString(contractEndC <contract_id=" + toString(contractId) + ">},exchang [origin=promotion];;" + plusDaysAsString(date: cur 1) + "}"

Budowa DMNa z opisu biznesowego

- a) jeśli oferta pochodzi spoza PL, zgłoszenie jest oznaczone jako nieaktywne,
 - b) jeśli kwota zgłoszenia jest większa niż 2000 PLN, zgłoszenie oznaczamy jako przyjęte,
 - c) jeśli kwota zgłoszenia jest mniejsza niż 2000 PLN, zgłoszenie zostanie oznaczone jako zarchiwizowane.
-

Zasady zadania

- wszystkie zmienne dodajemy przez Camunda Cockpit przy starcie procesu
 - DMN implementujemy w procesie nadrzędnym jako ostatni element procesu.
-

Zadanie - budowanie DMNa za pomocą modelera

Grupa 1:

- a) jeśli kwota na fakturze jest większa niż 5000 zł, przypisujemy fakturze priorytet wysoki,
 - b) jeśli zapłaty jest do 10 dni od dnia dzisiejszego, też otrzymujemy priorytet wysoki,
 - c) w pozostałych przypadkach - priorytet niski
-

Zadanie - budowanie DMNa za pomocą modelera

Grupa 2:

- a) jeśli liczba nadgodzin pracownika jest większa niż 30, wniosek jest oznaczany jako odrzucony,
 - b) jeśli data zgłoszenia wniosku mieści się w trzech ostatnich dniach miesiąca (29/30/31) wniosek traktujemy jako odrzucony
 - c) w innych przypadkach otrzymujemy wstępną akceptację
-

Zadanie - budowanie DMNa za pomocą modelera

Grupa 3:

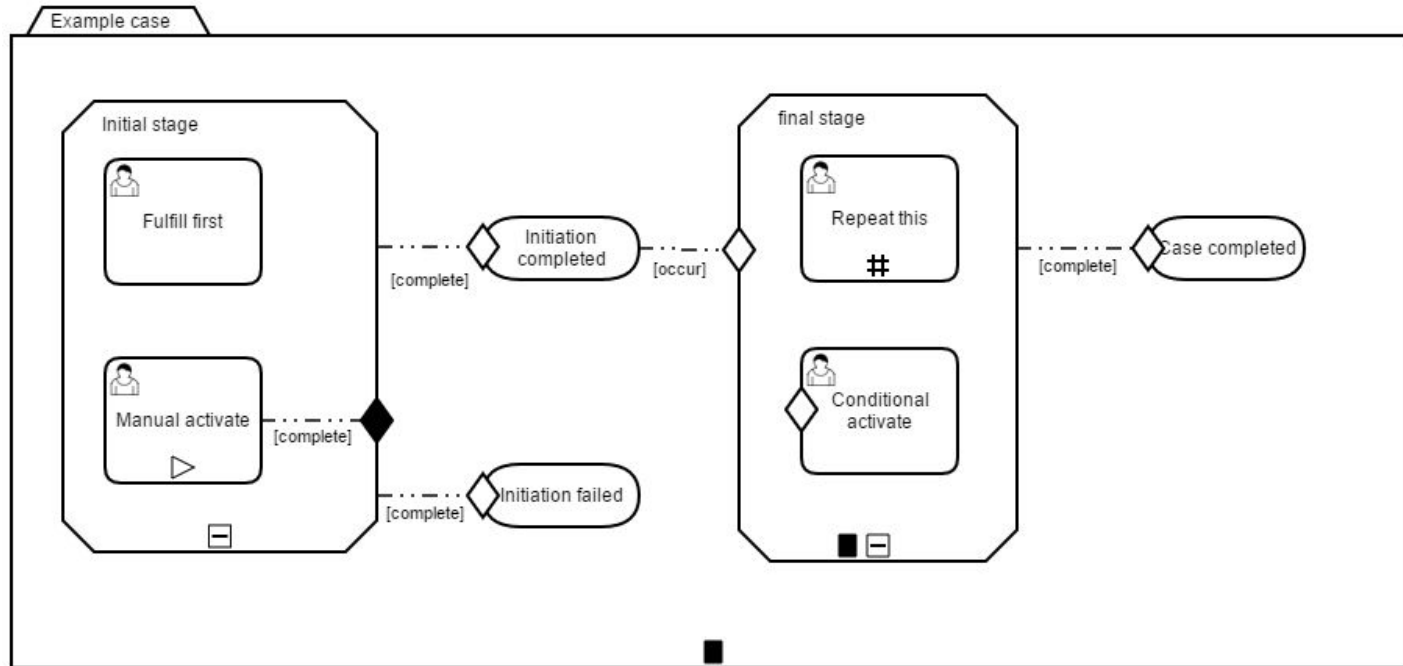
- a) jeśli proponowana płaca pracownika przekracza 20000 zł, zgłoszenie jest traktowane jako wstępnie odrzucone,
 - b) jeśli kandydat był już u nas na rozmowie, także oznaczamy jako wstępnie odrzucone,
 - c) w przeciwnym przypadku jest oznaczone jako wstępnie zaakceptowane
-

Zadanie - budowanie DMNa za pomocą modelera

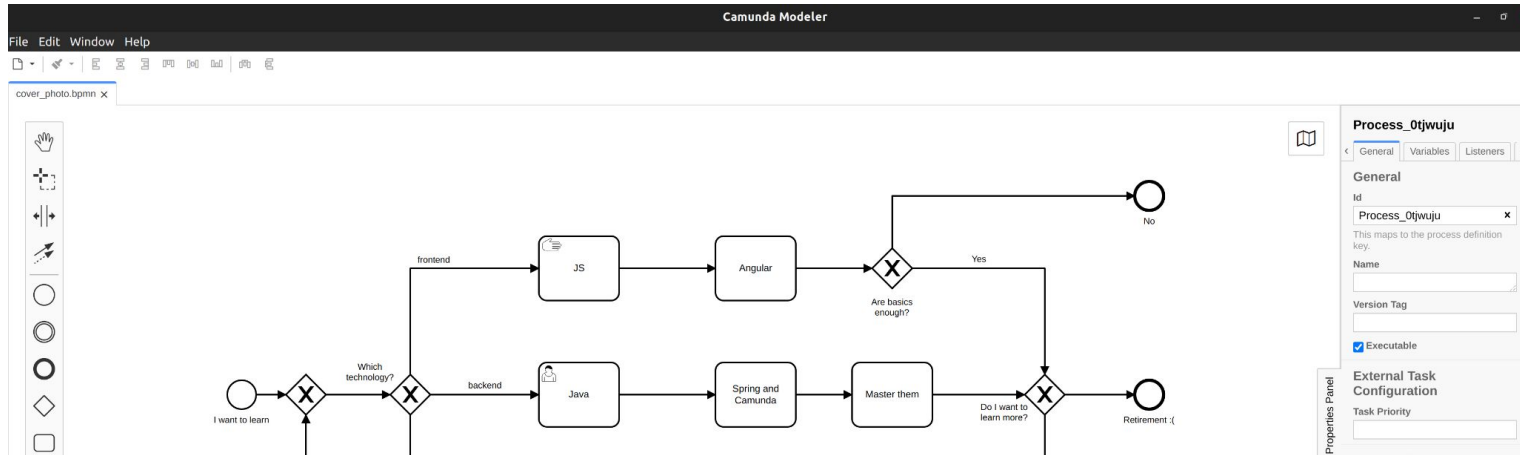
Grupa 4:

- a) Jeśli kraj wykonania konserwacji jest z poza PL, konserwacja dostaje status anulowanej,
 - b) jeśli koszt przekracza 3000 zł, dostaje status do dodatkowej akceptacji,
 - c) w innym przypadku status jest pozytywny
-

Case management - CMMN



Camunda Modeler



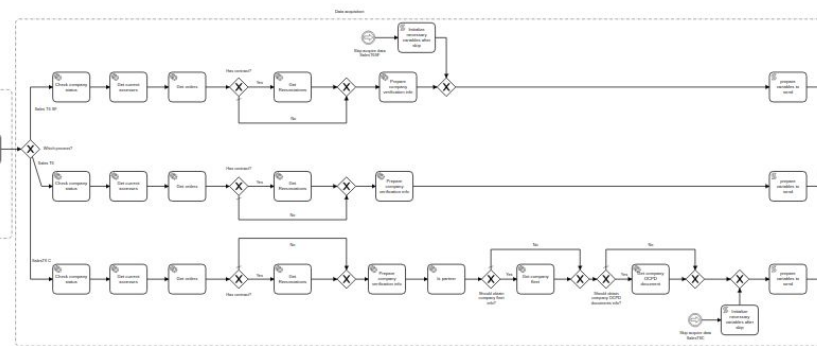
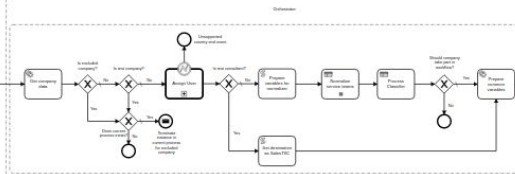
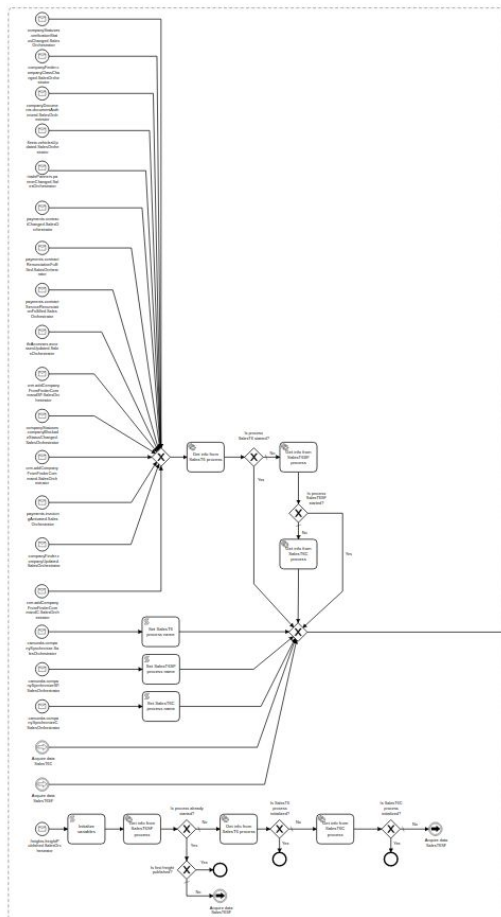
Procesy (pojęcia, budowanie)

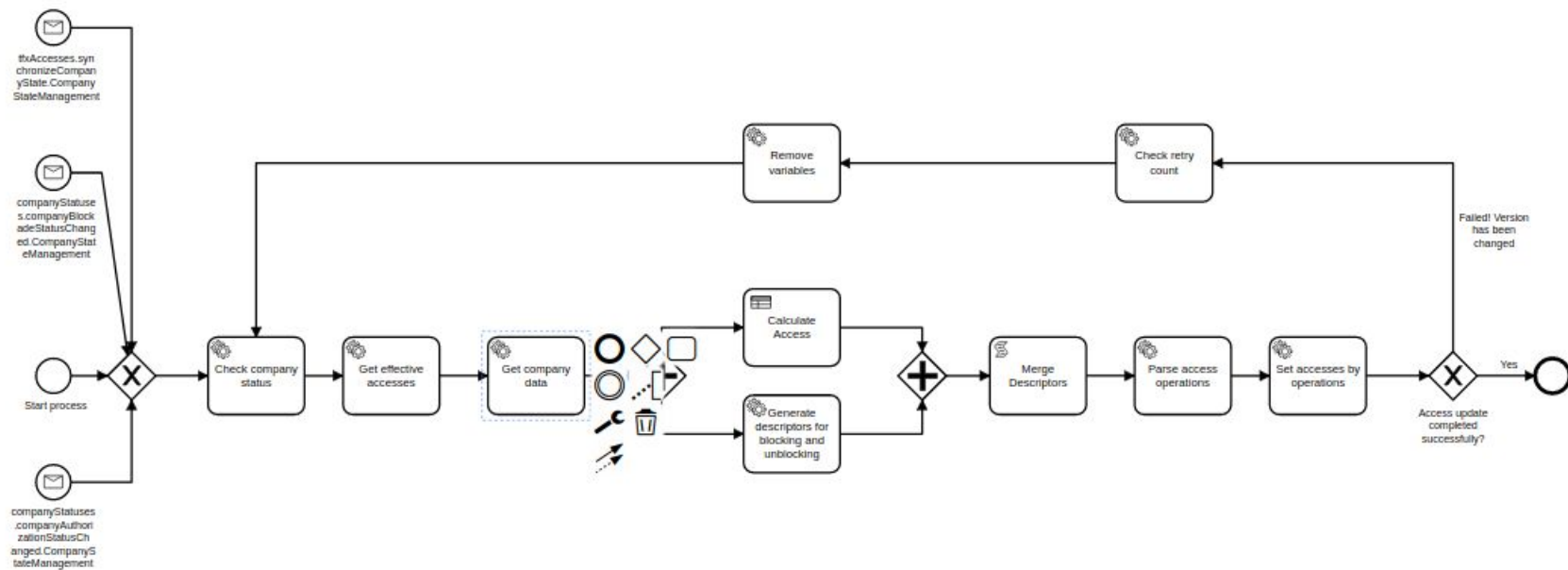
Jak rozpocząć i zakończyć proces?

Elementy BPMN a proces

Cykl życia procesu

Prosty proces vs. skomplikowany proces





Timery i messaging

- sposoby konfiguracji timerów (date, duration, time cycle)

```
managementService.setJobDuedate(String jobId, Date newDuedate)
```

- message a signal

```
// correlate the message
MessageCorrelationResult result = runtimeService.createMessageCorrelation("messageName")
    .processInstanceBusinessKey("AB-123")
    .setVariable("payment_type", "creditCard")
    .correlateWithResult();
```

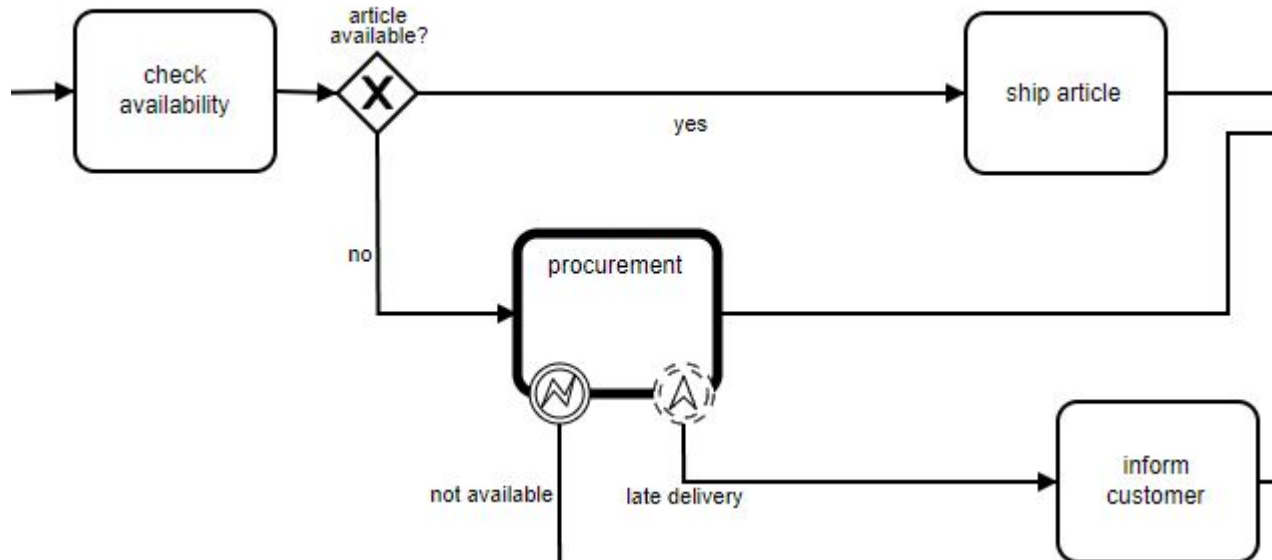
Ćwiczenie - konfiguracja timerów

- dodanie i konfiguracja timera zgodnie z <https://docs.camunda.io/docs/components/modeler/bpmn/timer-events/>

Ćwiczenie - startowanie przez message

- implementacja delegata i startowanie procesu podrzędnego (dodanie prostego bpmna z jednym user taskiem)
-

Boundary events



Budowa przykładowego procesu



Zadanie - implementacja procesu

Description

Jako opiekun procesu systemu sprzedażowego, chciałbym, aby proces obsługi pożyczki był funkcjonalny.

Kryteria akceptacyjne:

1. Proces jest możliwy do wywołania z poziomu cockpitu, ale także z poziomu aplikacji backendowej
2. Service taski nie powinny być operacyjne (można użyć JavaDelegateMock)
3. Manual taski są obsługiwane przez zmienne procesowe
4. Timery konfigurowane są z poziomu procesu
5. Subprocesy chwilowo są zastąpione poprzez manual taski

Podział zadań

- Grupa 1 - proces LoanProcess
 - Grupa 2 - proces LongTermLoanProcess
 - Grupa 3 - proces ShortTermLoanProcess
 - Grupa 4 - proces HouseLoanProcess
-

Interpretacja elementów procesu przez silnik

- kiedy następuje flush i zapisanie stanu?
 - kolejność wykonywania elementów
 - hands on na interfejsy (debug)
-

Q&A

Dzień trzeci

Java delegates/jobs

```
@Slf4j
@Component
public class OrderDataClassifierDelegate implements JavaDelegate {
    private static final String HAS_ORDER_KEY = "hasOrder";
    private static final String ORDER_LINE_KEY = "orderLine";
    private static final String END_DATE_KEY = "endDate";
    private static final String CONTRACT_END_DATE_VARIABLE_NAME = "contractEndDate";
    private static final String ORDERS_VARIABLE_NAME = "orders";
    private static final String END_DATES_VARIABLE_NAME = "endDates";
    private static final String CONTRACT_ID_VARIABLE_NAME = "contractId";
    private static final String ORDER_CLASSIFIERS_VARIABLE_NAME = "orderClassifiers";
    private static final String CONTRACT_TYPE_KEY = "contractType";

    @Override
    public void execute(DelegateExecution execution) throws IOException {
```

Zadanie - napisanie service tasków

Description

Jako właściciel aplikacji wyboru usług, chciałbym żeby service taski były operacyjne

Kryteria Akceptacyjne:

Wszystkie procesy muszą zacząć się w oparciu o zmienną procesową `userIdentificationNumer` (type:String). Dane do service tasków są do pobrania z lokalnej bazy danych poprzez repozytoria dostarczone w aplikacji backendowej.

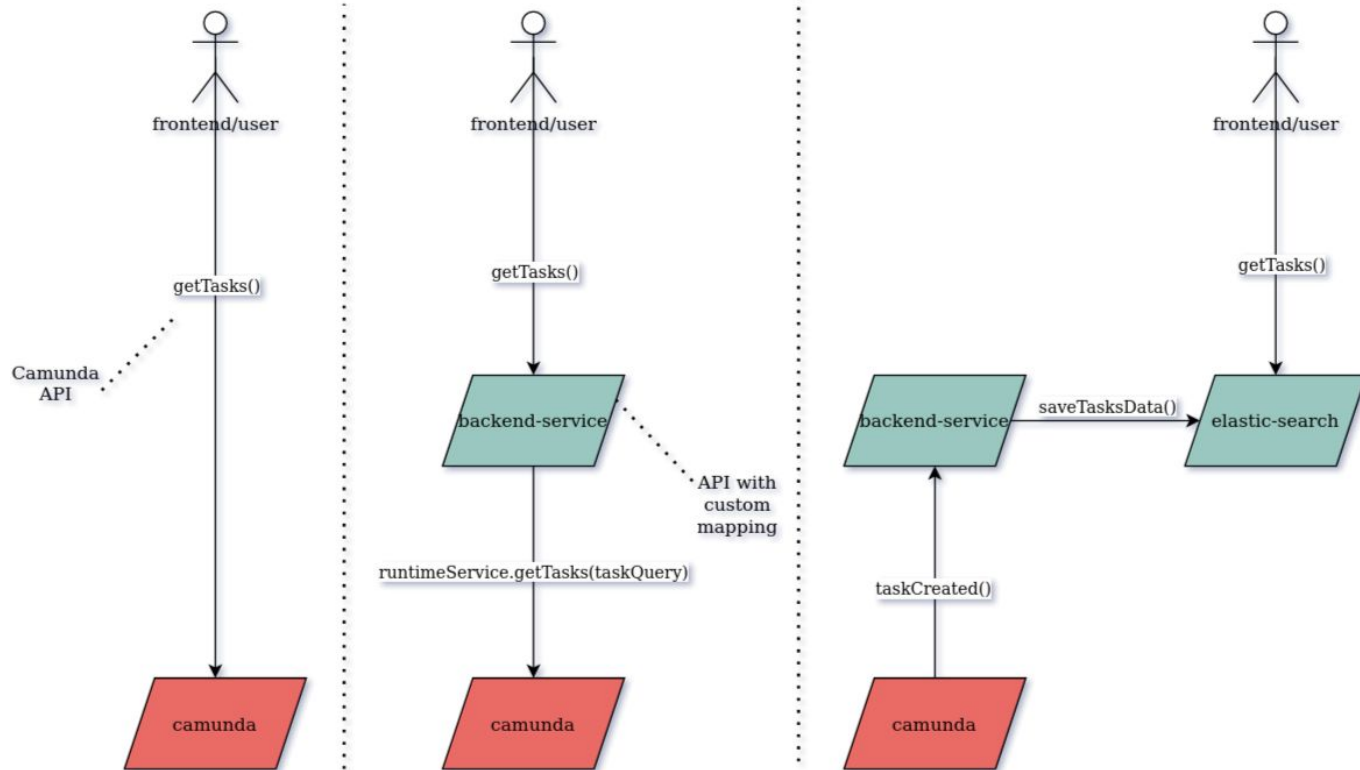
1. Grupa 1 - implementacja `ValidateUserData` (dodanie do zmiennych startowych następujących danych: `street`, `phoneNumber`, `declaredIncome`, `currency` oraz porównanie ich z danymi z bazy danych)
2. Grupa 2 - `GetDebtData` (zaczynamy proces ze zmienną `userId`)
3. Grupa 3 - `GetPreviousLoansData` (zaczynamy proces ze zmienną `userId`)
4. Grupa 4 - `GetPreviousUserDebts` (zaczynamy proces ze zmienną `userId`)
5. Wszystkie grupy - implementacja external taska pobierającego do zmiennej procesowej przelicznika dla waluty bieżącego użytkownika (waluta jako zmienna wejściowa, `currency`)

Tech info: dane pobieramy z <http://api.nbp.pl/>



Zadanie - przekazywanie zmiennych

Custom task structure



Rest API

Najważniejsze do użycia:

- batch
 - case instance
 - external task
 - incident
 - job
 - process instance
 - task
-

Book Requests : Runtime | History

Activity Instance Statistics: on

Set books available statuses

The book is available?

Yes

No

User's ID card for period borrowing

Process Instances Incidents Called Process Definitions Job Definitions Modify Documentation

Message	Process Instance	Timestamp	Activity	Failing Activity	Process Instance ID	Type	Action

Open stacktrace

Unknown property used in expression: #{Available == 'no'}. Cause: Cannot resolve identifier 'Available'

The book is available?

Failed

Zadanie - modyfikacja procesu

Jako opiekun procesu systemu sprzedażowego, chciałbym, aby formularz wyboru procesu był zmodyfikowany o najnowsze wymagania.

Kryteria akceptacyjne:

Grupa 1 - Loan Process:

- Proces może być rozpoczęty przez singal
- Integracja z procesami implementowanymi przez inne grupy
- Proces wykorzystuje inclusive gateway (zamiana gatewaya Which loan option?)

Grupa 2:

- Proces może być rozpoczęty przez event (timer ustawiony co 20 sekund)
- Proces posiada service taska który uruchamia wysyłanie maili do wszystkich userów (parallel task)
- Proces wykorzystuje posiada user task przerywany po 40 sekundach nieaktywności (Accept RODO)

Grupa 3:

- Process może być rozpoczęty przez message
- Proces zawiera call-activity (ekstrakcja external taska)
- Proces zawiera event based gateway (oczekiwanie na signal lub conditon event), do dodania dodatkowe 3 user taski

Grupa 4:

- Process może być rozpoczęty przez conditional start event
 - Proces zawiera subproces (ekstrakcja części z pętlą)
 - Proces zawiera non-interrupting message event, który po odebraniu message na tasku Send Invitation Email wykona dodatkowy serwis task (wygenerowanie przypomnienia wywoływane jako log do konsoli)
-

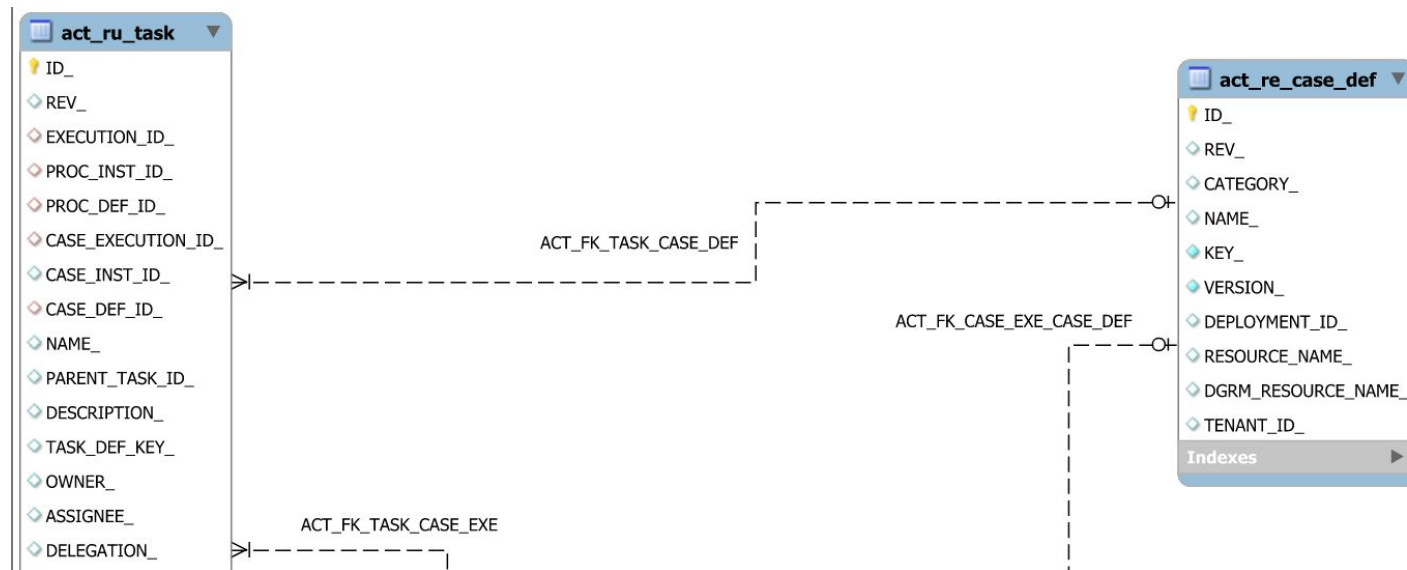
Implementacja DMN

Implementacja DMNa na potrzeby procesu Loan Process, dla zmiennej dla gatewaya Which loan option?. Warunki dla DMN podane są poniżej:

client.declared_income	sum_of(debt.amount)	sum_of(loan_amount)	output
>2000	<5000	<5000	short_term
>4000	>5000	<3000	long_term
>10000	>5000	<8000	house_loan
null	null	null	error

Dzień czwarty

Schematy danych w Camundzie



Schemat danych - ćwiczenia

Wersjonowanie, migracja

Process Definitions

ID_	KEY_	VERSION_	...
invoice:1:1f57588f-afa8-11e1-a01a-08002700282e	invoice	1	
invoice:2:8d55cd80-af15-11e1-8950-08002700282e	invoice	2	

Process Instances

ID_	PROC_DEF_ID_	ACT_ID_	...
71712c34-af1d-11e1-8950-08002700282e	invoice:1:1f57588f-afa8-11e1-a01a-08002700282e	assign-approver	
c80-af15-11e1-8950-0800270028	invoice:2:8d55cd80-af15-11e1-8950-08002700282e	assign-approver	

reference

Migracja - ćwiczenia

Przemigruj wybraną instancję procesu do najnowszej wersji korzystając z REST API.

Testy jednostkowe

Co testujemy?

Testy procesów a CI/CD

Integracja Gitlab/Jenkins

Zadanie - testy jednostkowe

Grupa 1

- test delegata ValidateUserData
- test procesu LoanProcess

Grupa 2

- test delegata GetDebtData
- test procesu LongTermProcess

Grupa 3

- test delegata GetLoanData
- test procesu ShortTermProcess

Grupa 4

- test delegata GetDebtDetails
- test procesu HouseLoanProcess

Testy procesowe obejmują happy path, przepływ jednej ścieżki. Bonus - test DMNa.

Dzień piąty

External Task Handler

Co testujemy?

Testy procesów a CI/CD

Integracja Gitlab/Jenkins

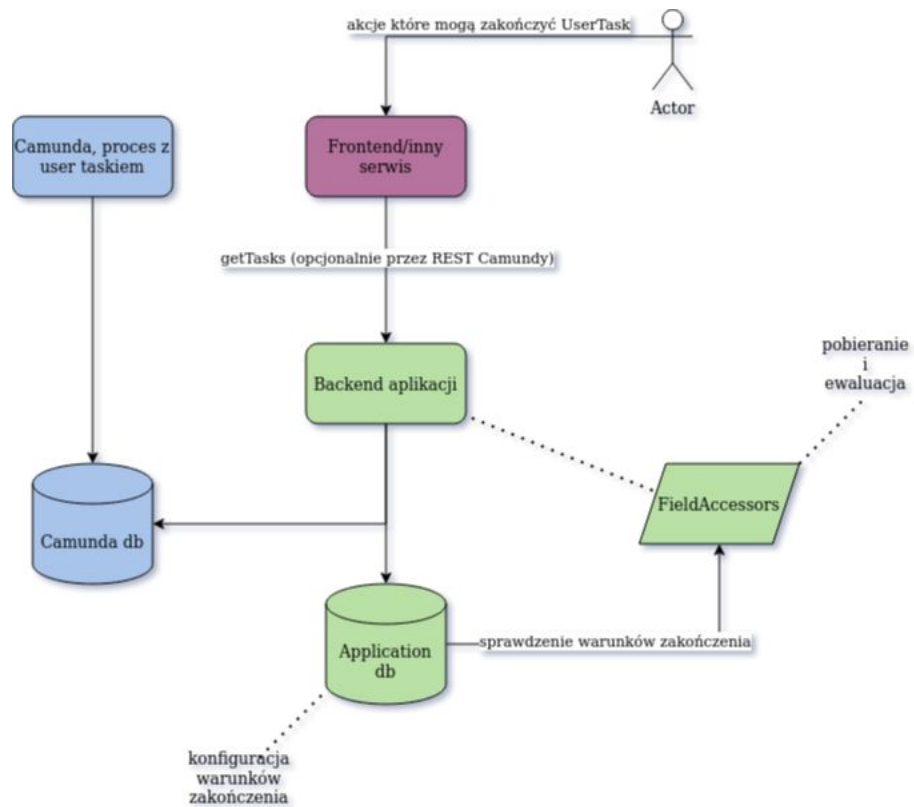
Zadanie - przeniesienie external tasków do drugiej aplikacji

Optymalizacja REST API

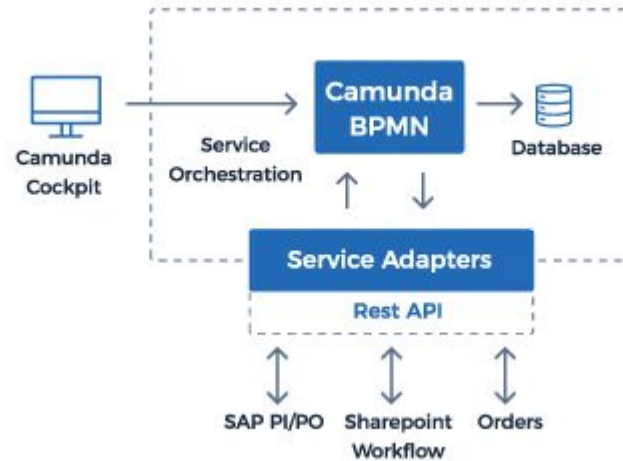
Co testujemy?

Testy procesów a CI/CD

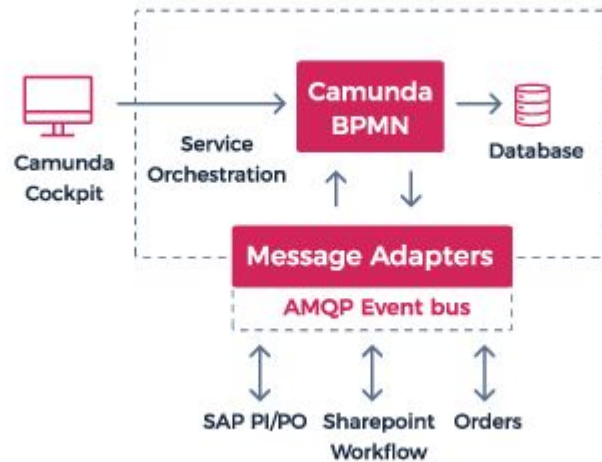
Integracja Gitlab/Jenkins



Komunikacja synchroniczna vs. asynchroniczna

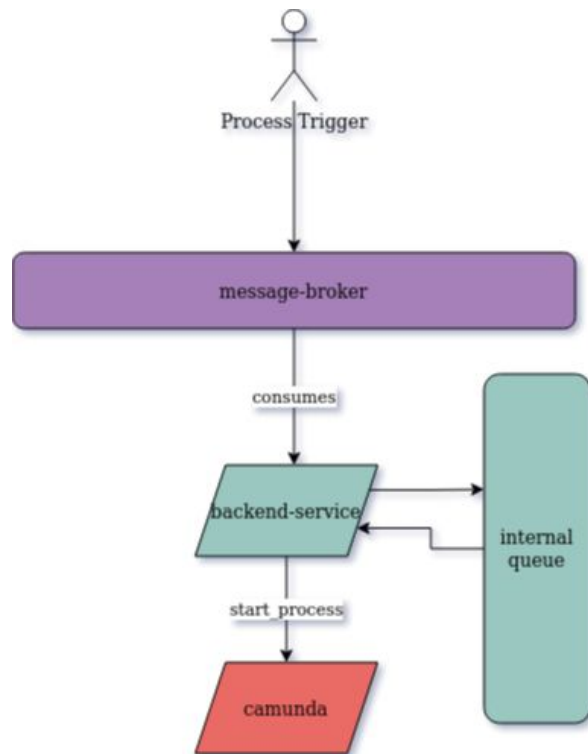


Komunikacja synchroniczna vs. asynchroniczna



Ćwiczenie - AMQP plus Camunda

Asynchroniczny pattern rozpoczynania procesów



Dzień szósty

Patterny/antypatterny procesowe i dobre praktyki

- process discovery,
 - prostota procesów,
 - SOLID,
 - decyzja - jak używamy Cockpitu?
 - nazewnictwo, patterny projektowania procesów, wersjonowanie
 - handlowanie variables w procesach
 - jak długo powinny żyć procesy?
 - Camunda a Elastic Search
 - optymalizacja wydajności
 - manualne działania na produkcji
 - klucze biznesowe
 - Mergowanie procesów
 - Kompatybilność delegatów
-

Pattern Aggregators

<https://www.enterpriseintegrationpatterns.com/Aggregator.html>

Skalowanie aplikacji Camundowych

- patterny
 - bottlenecki
 - Camunda 8
-

User forms

- patterny
 - bottlenecki
 - Camunda 8
-

Ćwiczenie

Handling incydentów



Debugowanie

Runa Cockpit Processes Decisions Human Tasks More

Processes - launchCancel: 17f99a18-47a1-11e7-b4a8-02425f86c310 : Runtime

Filter

ID:

1-47a1-11e7-b4a8-02425f86...

Key:

059312529971d30d9ce773...

Version:

ID:

ancel2:ad507559-47a0-11e...

Key:

ancel

Name:

ancel

:

mt ID:

5-47a0-11e7-b4a8-02425f8...

Inspect "creditPayload" variable

Object Type Name: java.util.LinkedHashMap

Serialization Data Format: application/x-java-serialized-object

Serialized

Deserialized

```
{
  "invoiceNumber": "NXW-1698",
  "subsidiary": {
    "id": null,
    "amount": "9200",
    "vatRate": "2000",
    "amountWoVat": "7668",
    "products": [
      {
        "sku": "mySKU1",
        "name": "dfggff",
        "quantity": "2",
        "type": "product",
        "unitAmount": "2300",
        "unitAmountWoVat": "1917",
        "vatRate": "2000",
        "discount": "0",
        "discountPercent": "0",
        "unitVatAmount": "383",
        "unitTotal": "4600",
        "unitTotalWoVat": "3834"
      },
      {
        "sku": "mySKU2",
        "name": "dfggff",
        "quantity": "2",
        "type": "product",
        "unitAmount": "2300",
        "unitAmountWoVat": "1917",
        "vatRate": "2000",
        "discount": "0",
        "discountPercent": "0",
        "unitVatAmount": "383",
        "unitTotal": "4600",
        "unitTotalWoVat": "3834"
      }
    ],
    "amounts": [],
    "percents": []
  }
}
```

Close

Change

Dzień siódmy

Wdrożenia produkcyjne

- procesy i ich wsteczna kompatybilność
 - aplikacja -> procesy
 - rozwiązywanie incydentów po wdrożeniowych
 - próbne migracje na niższych środowiskach
 - kopia baz produkcyjnych
 - wersjonowanie procesów
-

SpringBoot i Camunda

Distribution Type:
Tomcat

Distribution Type:
Docker

Generate a Spring
Boot Project



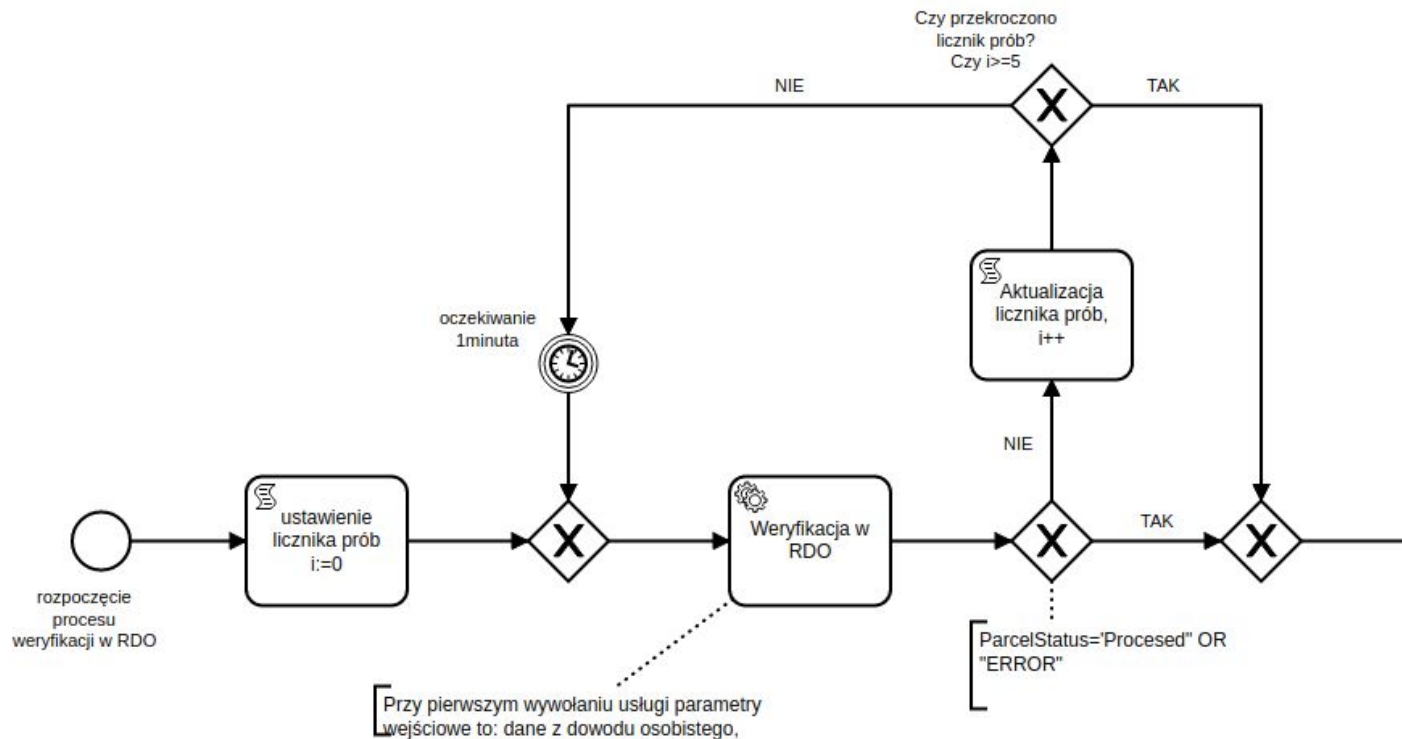
Run with Spring

Camunda Platform Initializr:

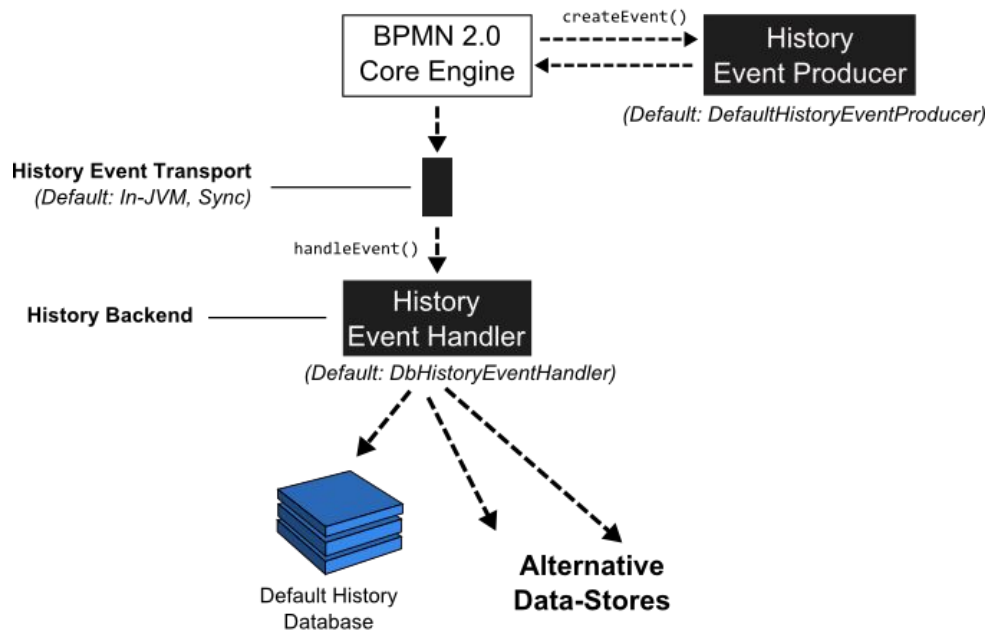
<https://start.camunda.com>

[Spring Boot Integration Documentation](#)

Refactor i usprawnienia procesów



Dane historyczne (Custom History Backend, History API)



Elementy z enterprise do implementacji

Pluginy do modelera

Strongly recommended plugins for the modeler:

Linters: checking missing names, configurations and some violations of best practices -

<https://github.com/camunda/camunda-modeler-linter-plugin>

Tooltip: shows basic info about element without opening the properties -

<https://github.com/viadee/camunda-modeler-tooltip-plugin>

Property Info: shows symbols indicating hidden properties (input/output mapping, start/end listeners, task listeners, documentation, field injection, variable mapping etc.) -

<https://github.com/umb/camunda-modeler-property-info-plugin>

Autosave: saves model automatically in a configured interval -

<https://github.com/pinussilvestrus/camunda-modeler-autosave-plugin>

Advanced plugins:

Script editor: colours the syntax of a script text; works only for script tasks -

<https://github.com/igorsimko/camunda-script-editor-plugin>

Token simulation: shows the behaviour of tokens in the process -

<https://github.com/bpmn-io/bpmn-js-token-simulation>

Transaction boundary: shows where the transactions happen in the process -

<https://github.com/camunda/camunda-modeler-plugins/tree/master/camunda-transaction-boundaries-plugin>

Technical ID generator: regenerates ids of activities for human readability -

<https://github.com/camunda-community-hub/camunda-modeler-plugin-rename-technical-ids>

Dokumentacja i źródła

[https://www.youtube.com/channel/UCTYG6fjGgHg5p47Pqsi
al2w](https://www.youtube.com/channel/UCTYG6fjGgHg5p47Pqsi
al2w)

<https://camunda.com/best-practices/performance-tuning-camunda/>

<https://docs.camunda.org/manual/7.16/>

Dzień ósmy

Camunda 8 - showcase



Camunda Platform 8

The Universal Process Orchestrator

Ćwiczenie - migracja procesu do Camunda 8

Camunda a Activiti

<https://camunda.com/blog/2016/10/camunda-engine-since-activiti-fork/>
