

CAMUNDA 8: LOW-CODE UND CONNECTORS

Prozessautomatisierung ohne Programmierung?

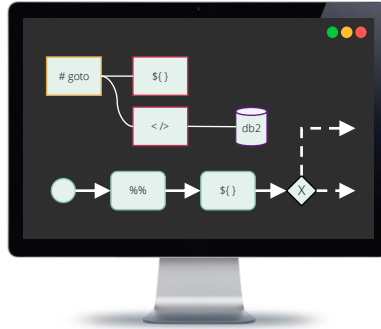
04.04.2023

Florian Runschke

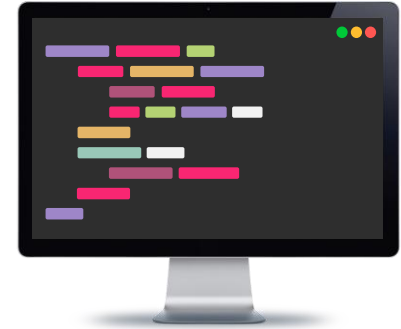
A decorative white line graphic that starts as a horizontal line from the left edge, then dips down at an angle, and finally rises back up to a horizontal line towards the right edge.



NO CODE



LOW CODE



PRO CODE

BPM-SUITE VS -FRAMEWORK

BPM-Suite

- Prozessmodellierung
- Datenmodellierung
- Geschäftsregelmodellierung
- Formular Design
- Benutzerzuweisung und Berechtigungen
- Integration von Schnittstellen

BPM-Framework

- Teil einer eigenen Anwendung
- Konfiguration & Logik im Code
- Unterstützung durch das Framework z.B. bei Retrys und Tests
- Klassischer Entwicklungsprozess

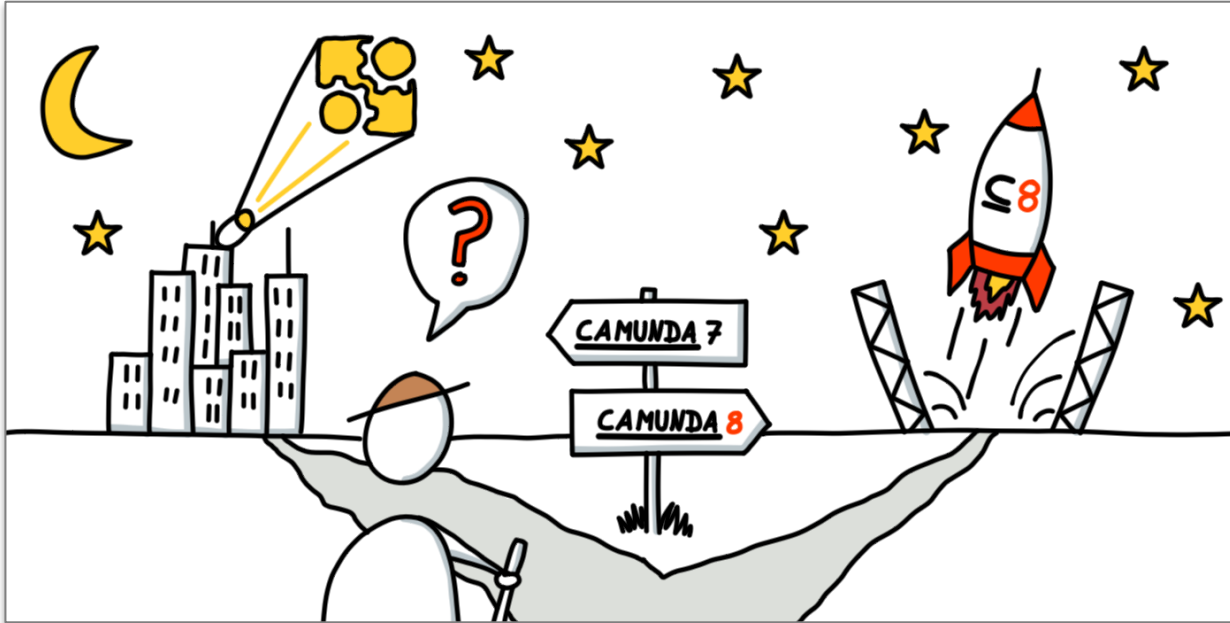
Low Code



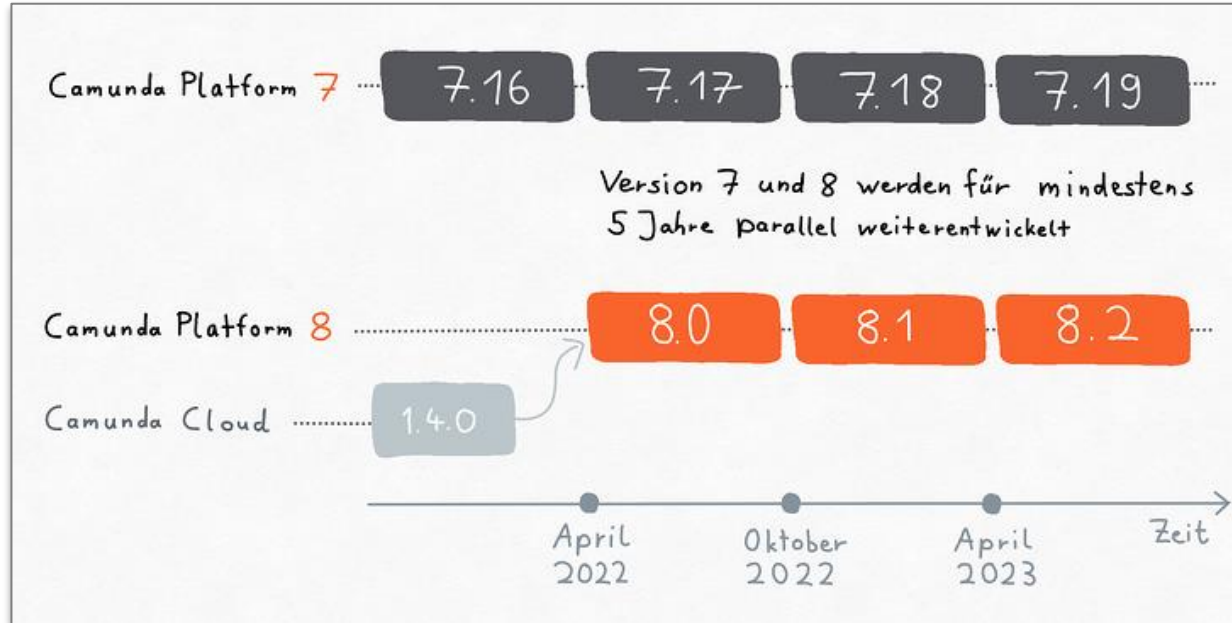
Pro Code

<https://blog.viadee.de/low-code-bpm-suites-vs-bpm-frameworks-welche-bpm-lösung-eignet-sich-für-mich-am-besten>

CAMUNDA PLATFORM 8

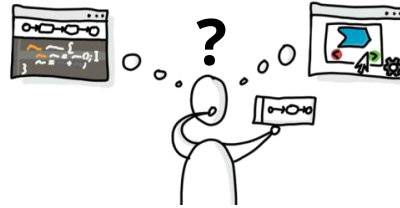


CAMUNDA PLATFORM 8



CAMUNDA PLATFORM 8

- Einordnung: BPM-Suite? BPM-Framework?

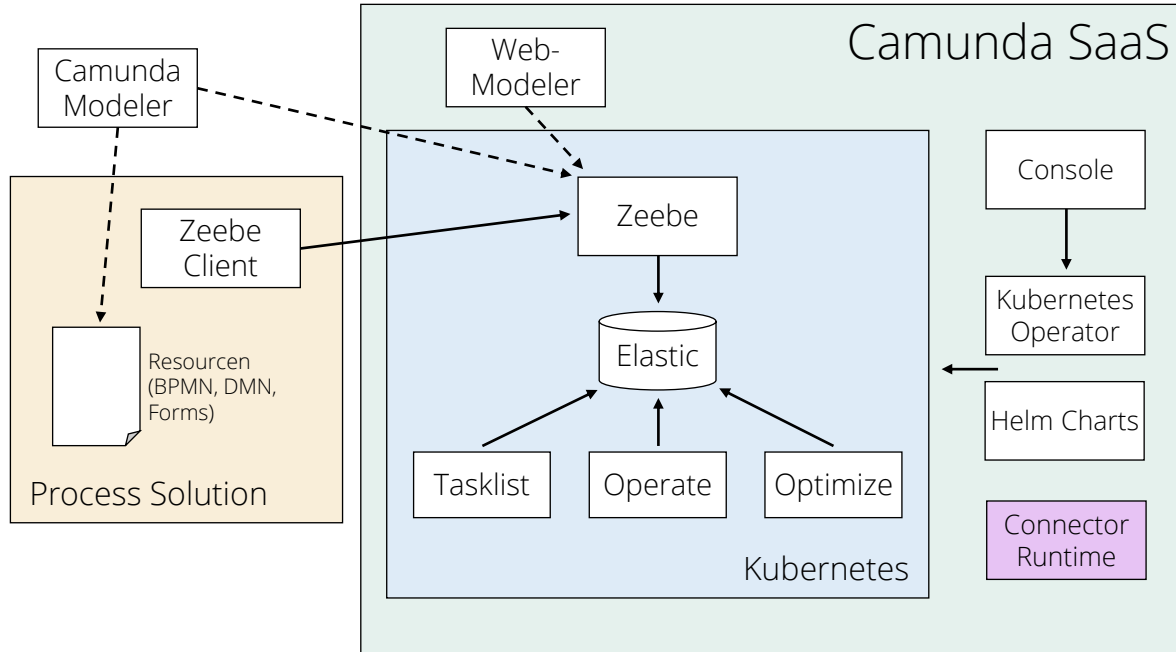


- »Camunda provides a lightweight, developer-friendly, easy-to-integrate solution [...]«
- »Dial-in Low Code«



<https://page.camunda.com/camundacon-2022-ruecker-keynote>
<https://docs.camunda.io/docs/components/concepts/what-is-camunda-platform-8>

CAMUNDA 8 – ARCHITEKTUR




- Neue Technologien
- Einige Komponenten ähneln Camunda 7
- Zeebe Client External-Task Handler
- Connector Runtime: Out-of-the-box Integration


OUT-OF-THE-BOX CONNECTORS


- Integration von Drittsystemen
- 12 out-of-the-box Connectors^{*)}
Mehr oder weniger generisch
- Einfache Kommunikation
mit gängigen Unternehmens-
anwendungen
- Konfiguration im Web-Modeler
- Erweiterung durch Low-Code Attribute

 **REST Connector**
Invoke REST API


 **SendGrid Connector**
Send an Email via SendGrid

 **Slack Connector**
Create a channel or send a

 **Amazon SNS Connector**
Send message to topic

 **Google Drive Connector**
Create folder or a file from

 **RabbitMQ Connector**
Send message to RabbitMQ

 **Amazon SQS Connector**
Send message to queue

 **Kafka Producer Connector**
Produce Kafka message

 **AWS Lambda Connect**
Invoke a function

 **GraphQL Connector**
Execute GraphQL query

 **UiPath Connector**
Orchestrate your UiPath Bots wi

 **Microsoft Teams Connector**
Create, update, and send a mess

^{*)} Stand 03/23, Camunda SaaS, Zeebe 8.2.0-alpha4


OUT-OF-THE-BOX CONNECTORS


- Es tut sich was
 - 17 out-of-the-box Connectors^{**)}




- 2 Inbound-Connectors




 Power Automate
Orchestrate your Power Automate flo...


 Easy Post Connector
Allows you to create addresses, parcel...

 GitHub
GitHub Connector

 Camunda Operate Connector
Fetch data from Camunda Operate API

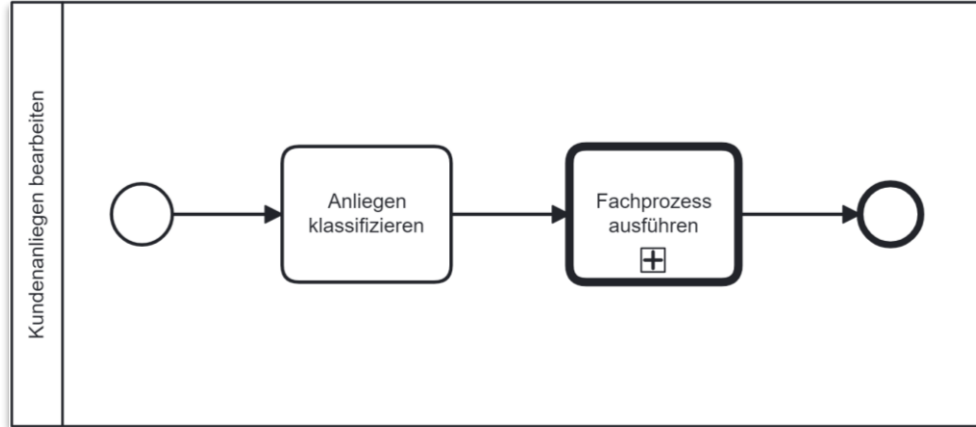
 OpenAI Connector
Interact with ChatGPT or Moderation ...

 Webhook Connector
Configure webhook to receive callbacks

 GitHub Webhook Connector
Receive events from GitHub

^{**)} Stand 04/23, Camunda SaaS, Zeebe 8.2.0-alpha5

KUNDENANLIEGEN BEARBEITEN



- Automatische Klassifizierung des Anliegens
- Entscheidung in Regeltabelle
- Nachbesserung durch Benutzereingabe
- Fachlogik in separaten Prozessen

AUTOMATISCHE KLASSIFIZIERUNG

- Da gibt es doch bestimmt eine Möglichkeit...
- Ausprobiert...



Was ist das Kundenanliegen?

"Sehr geehrte Damen und Herren, hiermit kündige ich an dass ich demnächste umziehen werde. Meine neue Adresse teile ich Ihnen rechtzeitig mit. Mfg K. Lauer"

Kundenanliegen: **Der Kunde möchte seine Adresse ändern.**

Was ist das Kundenanliegen?

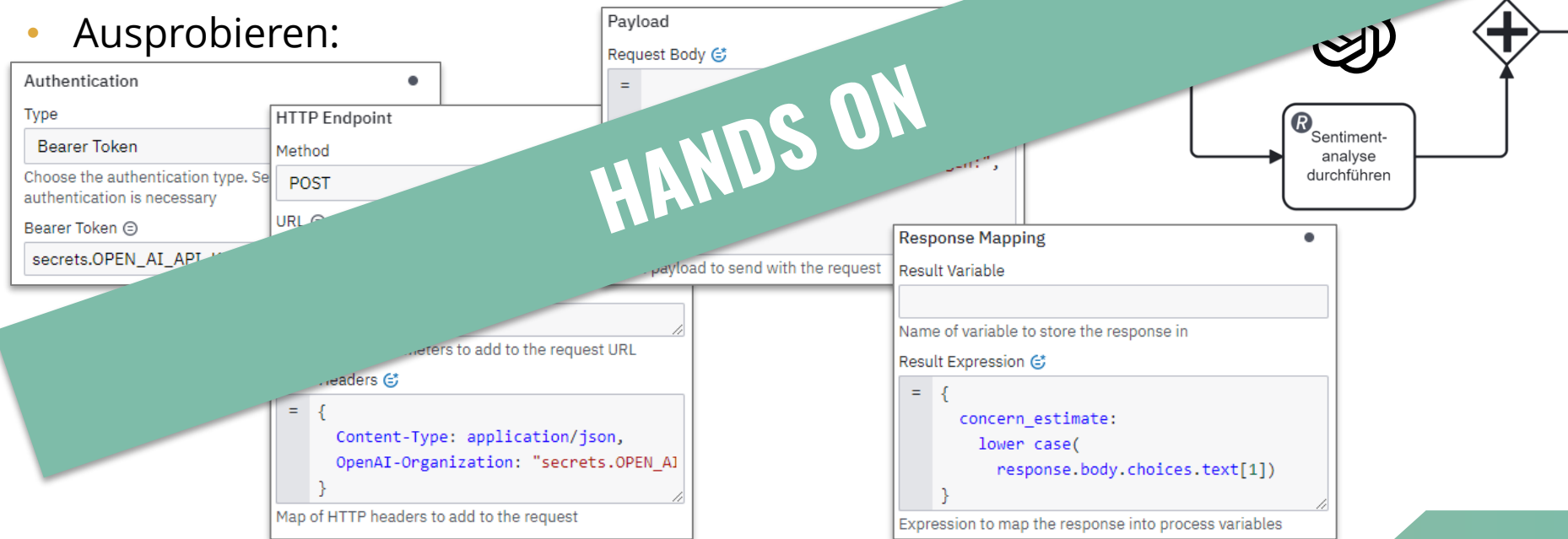
"Sehr geehrte Damen und Herren, das Wetter ist so schön. Ich trinke ein Bier und möchte meinen Vertrag kündigen. Mfg K. Lauer"

Kundenanliegen: **Kündigung des Vertrags.**

- Computer sagt... Ja!

AUTOMATISCHE KLASSIFIZIERUNG

- Account bei OpenAI einrichten
- Getting Started, Guide & Api-Reference beachten
- Ausprobieren:



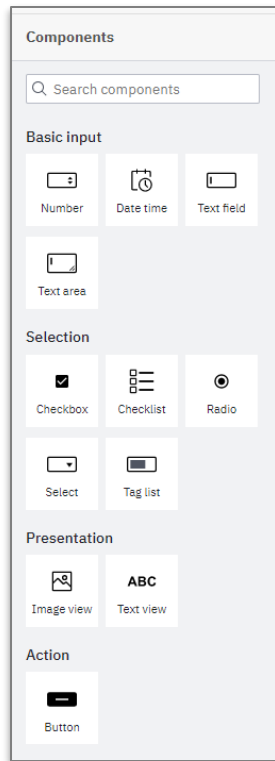
FACHPROZESS BESTIMMEN

- Modellierung im Web-Modeler
- Zugriff auf Prozesskontext / Variablen
- Verwendung von Feel-Funktionen
- Output: Fachprozess

Fachprozess bestimmen				
		Hit Policy: First ▼		
	When	And	Then	
	Einschätzung	Stimmung	Fachprozess	
	string	string	"adresse_aendern","kuendigung..."	
1	-	contains(?, "negativ")	"beschwerde"	
2	contains(?, "beschwerde")	-	"beschwerde"	
3	contains(?, "adresse")	-	"adresse_aendern"	
4	contains(?, "kündigung")	-	"kuendigung"	
5	-	-	"unbekannt"	
+	-	-		

FORMULARE

- Modellierung im Web-Modeler
- Wachsende Auswahl an Komponenten
- Dynamische Inhalte



FORMULARE – FORM DEFINITION

- Modellierung im Web-Modeler
- Wachsende Auswahl an Komponenten
- Dynamische Inhalte
- Form Definition
 - Vertikale Anordnung
 - Form Input als Vorschau

Form Definition

Fachprozess bestimmen

Kundennliegen:

Vorschlag Kundenanliegen

by OpenAI

Sentimentanalyse

by OpenAI

Fachprozess*

Form Input

```
1  v  {
2    "input": "Hallo, ich habe ein Anliegen! Mfg K. Lauer",
3    "concern_estimate": "unbekannt",
4    "
5  }
```

x concern_estimate

x field_0r5nk6j

x input

x sentiment

FORMULARE – FORM PREVIEW

- Modellierung im Web-Modeler
- Wachsende Auswahl an Komponenten
- Dynamische Inhalte
- Form Definition
 - Vertikale Anordnung
 - Form Input als Vorschau
- Form Preview
 - „Was wäre wenn“-Ansicht
 - Form Output für Variablen
- Form wird in das Prozessmodell kopiert!

Form Preview

Fachprozess bestimmen

Kundennliegen:

Hallo, ich habe ein Anliegen! Mfg K. Lauer

Vorschlag Kundenanliegen

unbekannt

by OpenAI

Sentimentanalyse

by OpenAI

Fachprozess*

Kündigung

Form Output

```
1 {  
2   "field_0r5nk6j": "kuendigung"  
3 }
```


FORMULARE – ÜBERSICHT

✓ Autosaved at 7:02:27 PM

Components

Search components

Basic Input

Number

Date time

Text field

Text area

Selection

Checkbox

Checklist

Radio

Select

Tag list

Presentation

Image view

ABC

Text view

Action

Button

Form Definition

Fachprozess bestimmen

Kundennutzen:

Vorschlag Kundenanliegen

by OpenAI

Sentimentanalyse

by OpenAI

Fachprozess*

Search

Form Input

```
1 v [ ]  
2 "input": "Hallo, ich habe ein Anliegen! Mfg K. Lauer",  
3 "concern_estimate": "unbekannt",  
4 {  
5   * concern_estimate  
   * field_0r5nk6j  
   * input  
   * sentiment  
}
```

Form Preview

Fachprozess bestimmen

Kundennutzen:

Hallo, ich habe ein Anliegen! Mfg K. Lauer

Vorschlag Kundenanliegen

unbekannt

by OpenAI

Sentimentanalyse

by OpenAI

Fachprozess*

Kündigung

Form Output

```
1 v {  
2   "field_0r5nk6j": "kuendigung"  
3 }
```

SELECT Fachprozess

General

Field label

Fachprozess

Field description

Key

field_0r5nk6j

Binds to a form variable

Default value

Searchable

Disabled

Condition

Hide if

Condition under which the field is hidden

Options source

Type

Static

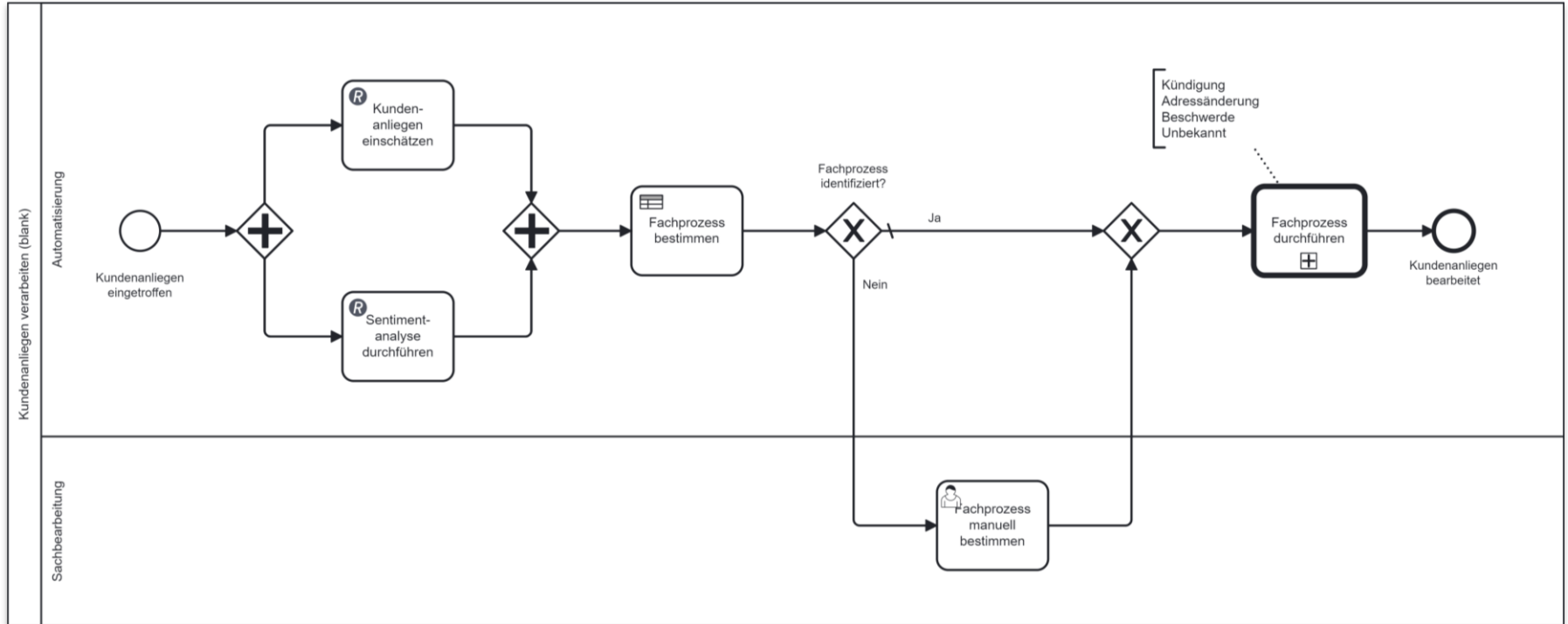
Static options

Validation

Required

Custom properties

KUNDENANLIEGEN BEARBEITEN



PRO UND CONTRA

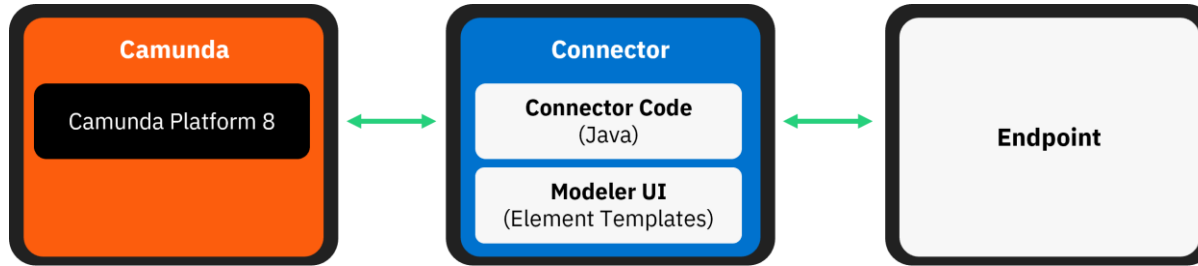
- Rest-Connector
 - ➕ Direkte Umsetzung, out-of-the-box
 - ➕ Keine weitere Software
und keine Laufzeitumgebung!
- Aber
 - ➖ Unübersichtlich
 - ➖ Keine *klassische* Versionskontrolle
 - ➖ Keine einfachen Mocks im Test
 - ➖ Viel Copy / Paste Code

ALTERNATIVE?



CUSTOM CONNECTOR

- Eigene Implementierung für Integration von Drittsystemen
- Template + Code notwendig
- Ausführung von Geschäftslogik



OPEN-AI CONNECTOR - TEMPALTE

- Connector-SDK
- Template Entwicklung im Web-Modeler

The screenshot displays the Camunda Web Modeler interface for configuring a custom connector template. The interface is split into two main panels: 'Raw JSON' on the left and 'Visual Preview' on the right.

Raw JSON Panel: Shows the JSON configuration for the connector. The visible code includes:

```
27 ],
28 "properties": [
29   {
30     "type": "Hidden",
31     "value": "de.viadee.bpm.zeebe.connector:openai:1",
32     "binding": {
33       "type": "zeebe:taskDefinition:type"
34     }
35   },
36   {
37     "label": "Input",
38     "description": "Name of the variable to use as input",
39     "group": "openai",
40     "type": "Text",
41     "feel": "required",
42     "binding": {
43       "type": "zeebe:input",
44       "name": "openai_input"
45     }
46   },
47   {
48     "constraints": {
49       "notEmpty": true
50     }
51   },
52   {
53     "label": "Operation",
54     "description": "Operation to perform",
55     "group": "openai",
56     "type": "Dropdown",
57     "id": "open_ai_type",
58     "choices": [
59       {
60         "name": "Sentiment analysis",
61         "value": "Sentiment analysis"
62       },
63       {
64         "name": "Customer concern recognition",
65         "value": "Customer concern recognition"
66       }
67     ]
68   }
69 ],
70 "output": {
71   "resultVariable": "openai_result",
72   "resultExpression": "={{openai_result}}"
73 }
```

Visual Preview Panel: Shows a visual representation of the connector configuration. It includes a header with the Camunda logo and the text 'OPENAI CONNECTOR ServiceTask'. Below this, there are several sections:

- General:** A section with a right arrow icon.
- Template:** A section with a right arrow icon.
- Open AI:** A section with a dropdown arrow icon.
- Input:** A section with a red border and a red error message: 'Must not be empty.' Below this, there is a text input field with the placeholder 'Name of the variable to use as input'.
- Operation:** A section with a dropdown menu showing 'Sentiment analysis' and 'Customer concern recognition'.
- Output:** A section with a dropdown arrow icon.
- Result Variable:** A section with a text input field.
- Result Expression:** A section with a text input field and a right arrow icon.

OPEN-AI CONNECTOR - CODE

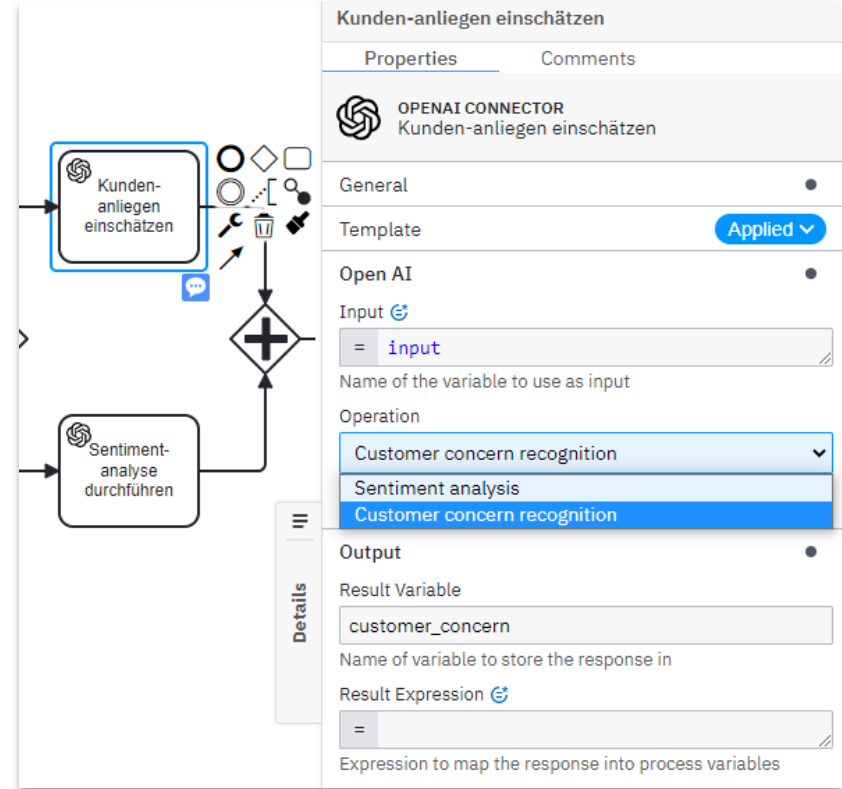
- Connector-SDK
- Template Entwicklung im Web-Modeler
- Connector Entwicklung in Entwicklungsumgebung

```
Powered by KikiManjaro
1  @Component
2  public class OpenAiConnector implements OutboundConnectorFunction {
3      private static final Logger log = LoggerFactory.getLogger(OpenAiConnector.class);
4
5      private final OpenAiAccess openAiAccess;
6
7      public OpenAiConnector(final OpenAiAccess openAiAccess) {
8          this.openAiAccess = openAiAccess;
9      }
10
11     @Override
12     public Object execute(final OutboundConnectorContext context) {
13         log.info("execute open-ai-connector");
14
15         var instructions = context.getVariablesAsType(DataModel.class);
16
17         if (instructions.isSentimentAnalysis()) {
18             var result = openAiAccess.performSentimentAnalysis(instructions.getInput());
19             log.info("sentiment result: {}", result);
20             return result;
21         } else {
22             var result = openAiAccess.determineCustomerConcerns(instructions.getInput());
23             log.info("customer concern result: {}", result);
24             return result;
25         }
26     }
27 }
28
29
```

OPEN-AI CONNECTOR - MODELER

- Connector-SDK
- Template Entwicklung im Web-Modeler
- Connector Entwicklung in Entwicklungsumgebung
- Einbindung im Prozessmodell

<https://github.com/viadee/zeebe-openai-demo-connector>



FAZIT

- Direkter, schneller, einfacher Low-Code-Start
- „Low Code“ ist relativ
- Tipp: Übersicht behalten!
- Licht für die Schatten-IT?
- Rest-Connector ggf. als Übergang,
perspektivisch spezialisierter Connector
- Entwicklung von Camunda 8 im Blick behalten
Weitere Schnittstellen
Inbound-Connectors
- <https://github.com/viadee/zeebe-openai-demo-connector>



OPEN AI



LOW-CODE MIT CAMUNDA 8

Prozessautomatisierung ohne Programmierung?



Florian Runschke

florian.runschke@viadee.de

+49 221 7 888 07 259

viadee Unternehmensberatung AG

Konrad-Adenauer-Ufer 7

50668 Köln

www.viadee.de