

# Service Engineering

## Übung 3c – Low Code, Enterprise-Mashups und API-Management

# Motivation

“Organizations are increasingly turning to low-code development technologies to fulfill growing demands for speed application delivery and highly customized automation workflows,” said [Varsha Mehta](#), Senior Market Research Specialist at Gartner. “Equipping both professional IT developers and non-IT personas — business technologists — with diverse low-code tools enables organizations to reach the level of digital competency and speed of delivery required for the modern agile environment.”

*Quelle: <https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023>, Abruf: März 2023*

# Ziele der Übung

- Begriffsklärungen Mashup, Low-/No-Code, API-Management
- Test zweier selbst gewählter No-Code/Low-Code Plattformen
- Aufgaben im API-Management – beteiligte Stakeholder
- Vergleich konkreter Lösungen zum API-Management

# Aufgabenstellungen

# Aufgabenstellung 1

- Es gilt es sowohl den wissenschaftlichen als auch technischen Sachstand zu den Themen No-Code, Low-Code, Mashup und API-Management zu recherchieren sowie konkrete Werkzeuge im Diskurs eigener Servicekompositionen zu testen.
- Führen Sie zunächst eine Begriffsklärung und Abgrenzung der von Ihnen verfolgten Themenstellung durch.
  - Funktionen einer No-Code bzw. Low-Code Plattform (vs. Mashups)
  - Kriterien-basierte Analyse von mind. 5 NoLoCode-Plattformen
- Abgrenzung zu KI-basierten Chatbots (ggf. Low-Code Aspekte)

# Aufgabenstellung 2

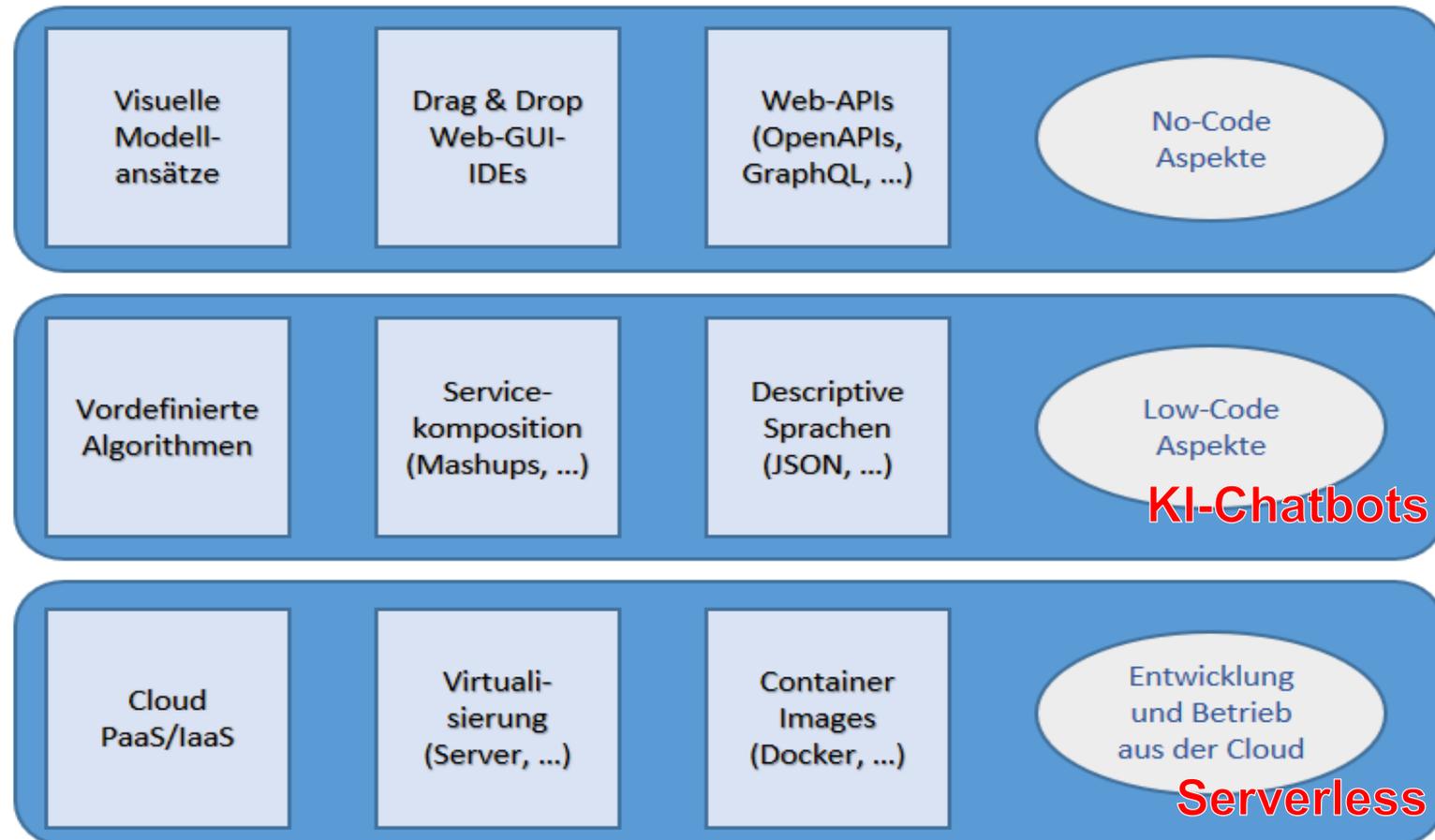
- Prototypischer Einsatz zweier Low-Code (Mashup-) Plattformen zur Servicekomposition:
  - Voraussetzungen für die Nutzung (Skill, Kosten, Lizenz, ...)
  - Möglichkeiten zur Integration eigener Services (Web-APIs)
  - Art der unterstützten Applikationen (mobile Apps, Web-APIs, ...)
  - Vorgehen für die Implementierung eines fachlichen Szenarios
  - Angebotene Import- und Exportfunktionalitäten (inkl. Schnittstellen)
  - Möglichkeiten zur Anpassung/Konfiguration und Versionierung
  - Rahmenbedingungen für die betriebliche Bereitstellung (DevOps)
- Möglichkeiten eines Plattformwechsels
  - (Standard-) Spezifikation der Komposition mit z.B. BPMN, XPDL
  - Mögliche Problembereiche der verwendeten Plattform

# Aufgabenstellung 3

- Ziele und Aufgaben des API-Management
  - Sicht der Geschäftsentwicklung (Prozesse und Produkte)
  - Sicht des Informationsmanagements
  - Sicht der Entwicklung bzw. Integration
  - Sicht des IT-Managements
- Recherche und kriterienbasierter Vergleich von mind. 2 Produkten z.B. API Management *MS Azure, Axway, CA, Apigee*
  - Voraussetzungen für die Nutzung (optional: eigene Tests)
  - Möglichkeiten zur Integration in die Unternehmens-IT-Architektur
  - Funktionsumfang und Einsatzbereiche (ggf. Cloud-Konsolen)
  - Berücksichtigung des Datenschutzes/Datensicherheit

# Weiterführende Hinweise

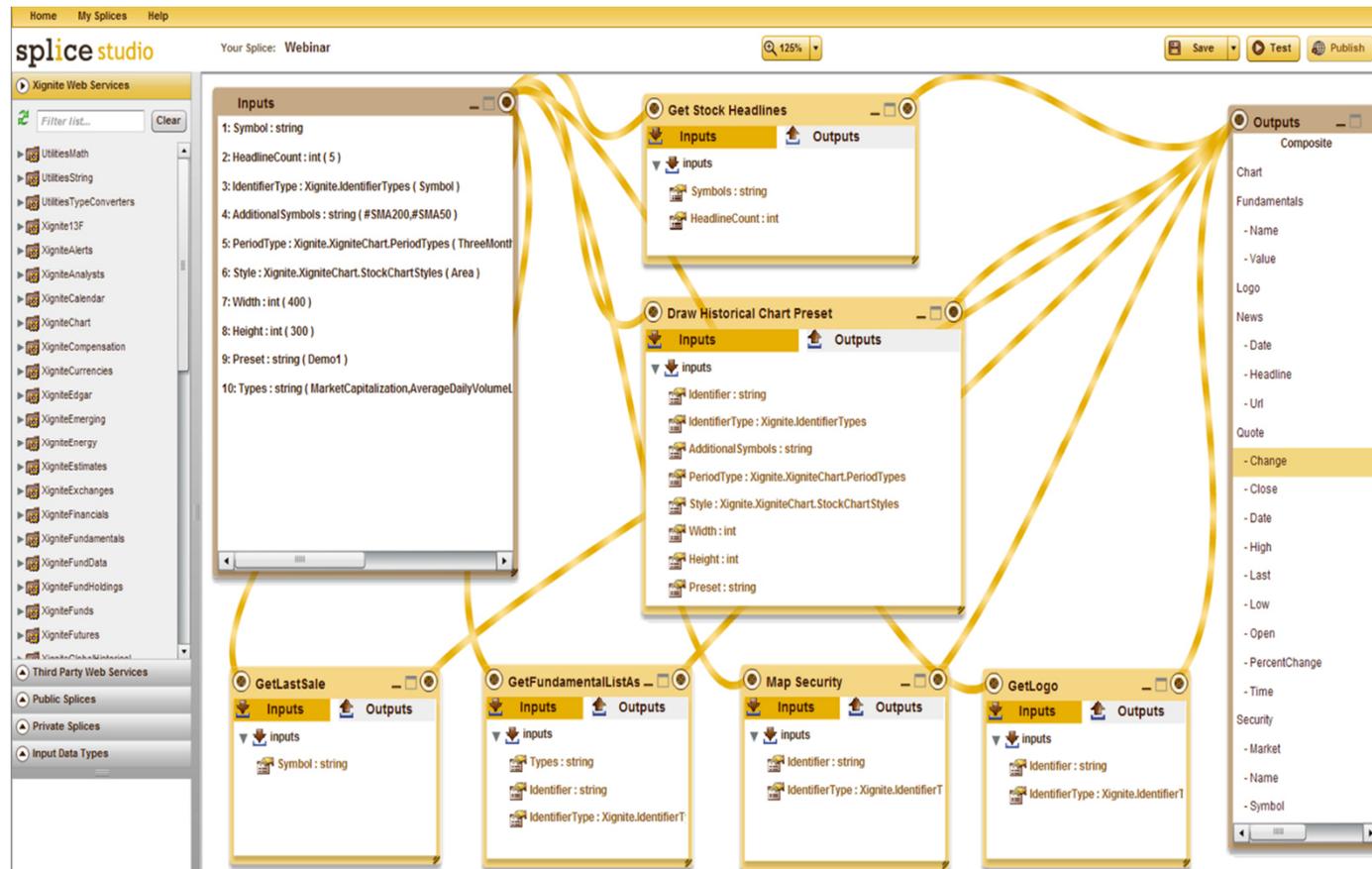
# Kernelemente No-Code/Low-Code



# Beispiel – splice studio

- Komposition von Serviceangeboten
  - Serviceangebote: Strikelron, WebserviceX, CDYNE, ...
  - Voraussetzung: Registrierung als Nutzer (*Splice* Profil & *xignite* Profil)
- Funktionen der SaaS-Anwendung
  - Grafische Präsentation von Serviceangeboten
  - Serviceauswahl per Drag & Drop
  - Anpassung der Input- bzw. Outputparameter gewählter Services
  - Servicekombination und Serviceverkettung
  - Splice-Test und Darstellung der Testergebnisse (XML, Baum, ...)
  - Kommunikations- und Spezifikationsmechanismen (REST, WSDL,...)
  - Veröffentlichung von Splices als neuer Services

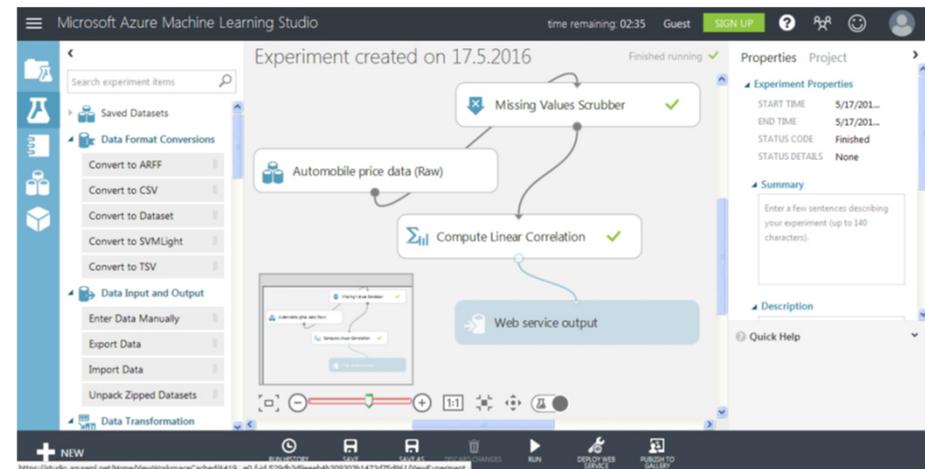
# Beispiel – splice studio



Quelle: <http://xignite.web-services-blog.com/media/splice-web-services-mashup-big.png>

# Beispiel – Azure ML Studio

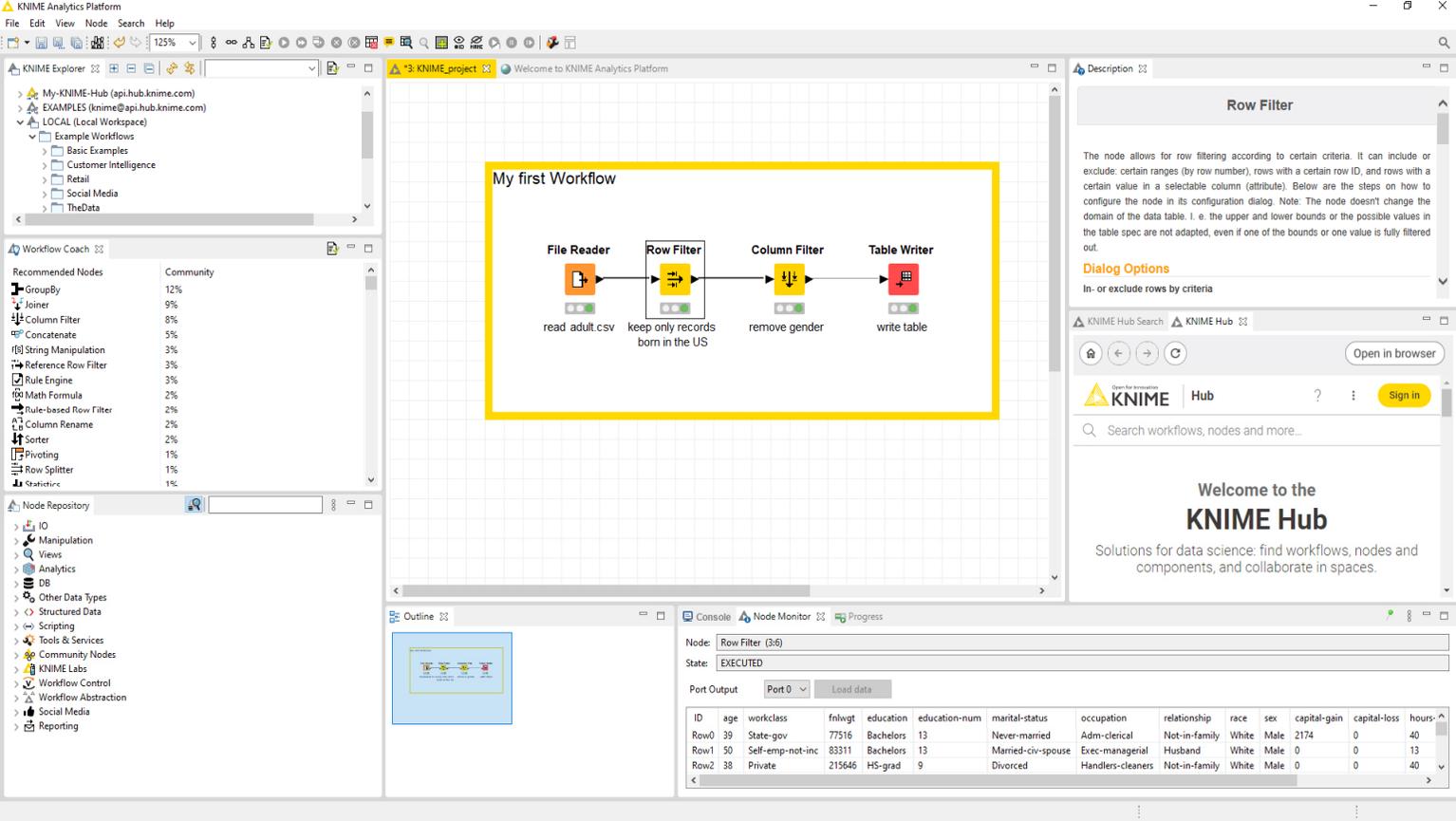
“Azure Machine Learning Studio provides an interactive visual workspace that enables you to easily build, test, and deploy predictive analytic models.”



Textquelle: Barga, R.; Fontama, V.; Tok, W. H.: Predictive Analytics with Microsoft Azure Machine Learning, Apress, Springer Science+Business Media, New York, 2014

Quelle der Abb.: <https://studio.azureml.net>, Erstellung und Abruf des Experiments, Mai 2016

# Beispiel KNIME-Plattform



The screenshot displays the KNIME Analytics Platform interface. The central workspace shows a workflow titled "My first Workflow" with the following steps: File Reader (read adult.csv), Row Filter (keep only records born in the US), Column Filter (remove gender), and Table Writer (write table). The Row Filter node is highlighted with a yellow box.

The right-hand pane shows the configuration for the Row Filter node, including a description and dialog options. Below this, the KNIME Hub search interface is visible, displaying a welcome message and a search bar.

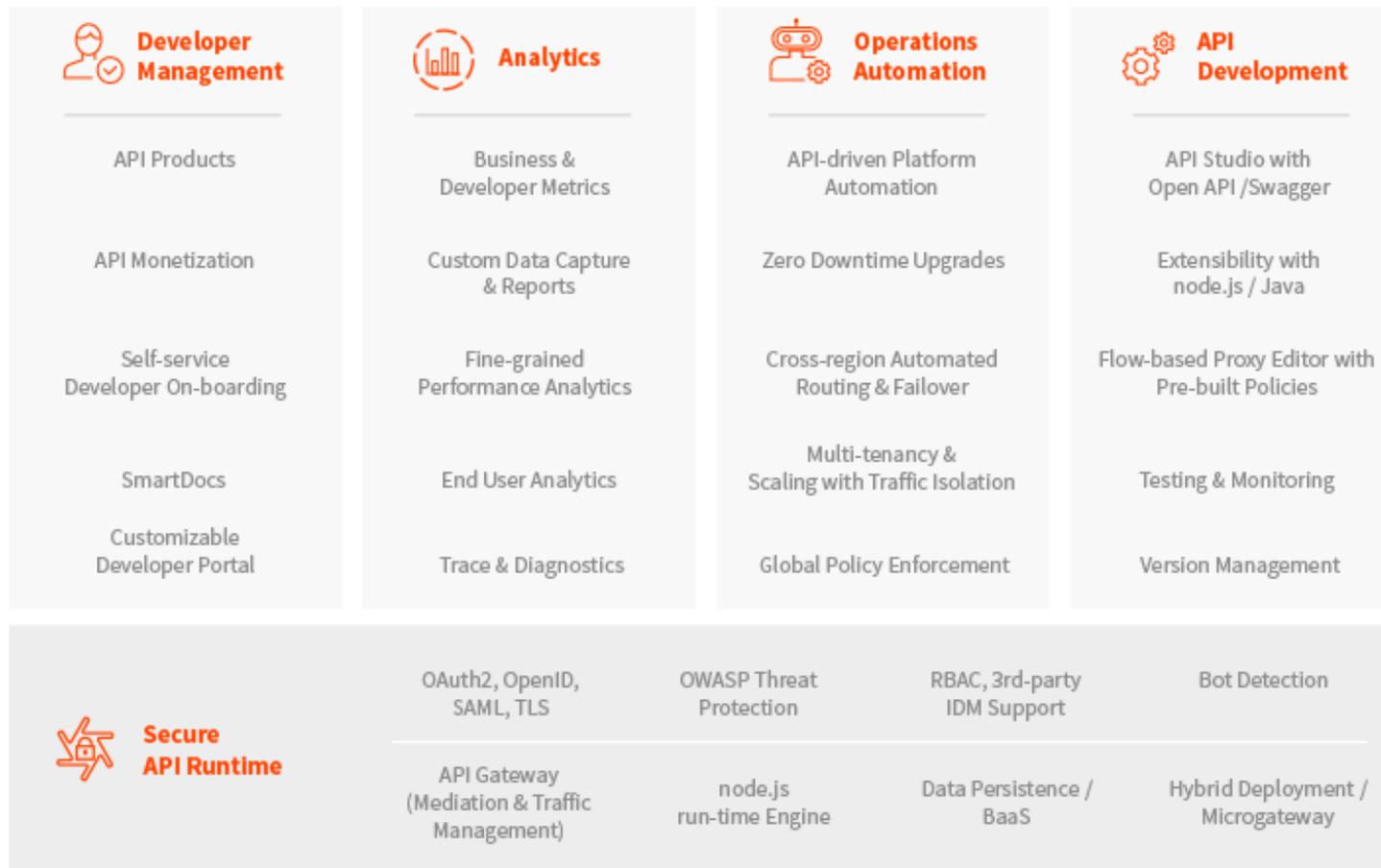
The bottom pane shows the Node Monitor for the Row Filter node, indicating it has been executed. The Port Output table is as follows:

ID	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours
Row0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
Row1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
Row2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40

Quelle: Knime-Workbench (Quelle: <https://www.knime.com>), Mai 2022



# API Management - Funktionen



Quelle: <http://apigee.com/about/products/api-management>, Abruf: Juni 2016

# Organisation der Übung

# Organisation der Übung

Bitte berücksichtigen Sie die folgende Vorgehensweise:

- Die Übung wird in Gruppen durchgeführt!
- Vorgeschlagene Methode zur Aufgabenlösung
  - Abstimmung einer geeigneten Vorgehensweise (15 min)
  - Ausführen der Aufgabenstellungen (120 min)
  - Erstellung eines Protokolls (45 min)
- Gesamtzeit für die Durchführung: ca. 180 min
- Bereitstellung eines entsprechenden Protokolls je Gruppe

# Aufbau des Protokolls

Verwenden Sie bitte des folgende Muster für das Protokoll:

- Allgemeine Angaben
  - Versuch, Beteiligte Studenten, Datum
  - Rahmenbedingungen (Hardware, Software, ...)
  - Methodisches Vorgehen zu Bearbeitung
- Aufgaben des Laborversuchs
  - Entsprechend der Struktur der Aufgabenstellung
  - Textliche Ausführungen zu den Lösungen
  - Verwendung von Grafiken und Screenshots
- Zusammenfassung (Bewertung der erreichten Ergebnisse)
- Genutzte Quellen (z.B. Literatur, Internet, ...)

# Interessante Quellen

- Lauer, B.: Low-Code-Plattform für professionelle Entwickler, <https://www.dotnetpro.de/tools/plattformen/low-code-plattform-professionelle-entwickler-1693306.html>
- Winterheimer, T.: API-Management ohne Code, <https://www.datacenter-insider.de/api-management-ohne-code-a-7547770b4a0c2402c1c38ee199ace055/>
- Ali, A.: 14 Open Source and Managed API Gateway for Modern Applications, <https://geekflare.com/api-gateway/>