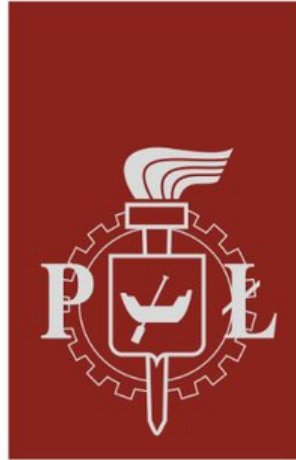


# Politechnika Łódzka

Wydział Elektrotechniki Elektroniki Informatyki i Automatyki



sem, zimowy, r ak. 2024/2025

## **Sprawozdanie z projektu BigData „Predykcja cen samochodów używanych”**



Mateusz Grzybek 240678

Kamil Młynarczyk 240757

16 grudnia 2024

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Założenia projektowe . . . . .	2
1.2	Komponenty . . . . .	2
1.3	Konteneryzacja . . . . .	2
<b>2</b>	<b>Diagramy</b>	<b>3</b>
2.1	Diagram przypadków użycia . . . . .	3
2.2	Diagram sekwencji zdarzeń . . . . .	4
<b>3</b>	<b>Aplikacja kliencka</b>	<b>5</b>
3.1	Opis . . . . .	5
3.2	Technologie . . . . .	5
3.3	Widoki aplikacji . . . . .	6
3.3.1	Strona . . . . .	6
3.3.2	Okno z ceną . . . . .	7
3.3.3	Okno z błędem . . . . .	7
<b>4</b>	<b>Komponent pośredniczący</b>	<b>8</b>
4.1	Opis . . . . .	8
4.2	Technologie . . . . .	8
<b>5</b>	<b>Komponent komunikacyjny</b>	<b>9</b>
5.1	Opis . . . . .	9
5.2	Technologie . . . . .	9
<b>6</b>	<b>Serwis predykcyjny</b>	<b>10</b>
6.1	Opis . . . . .	10
6.2	Technologie . . . . .	10

# Rozdział 1

## Wstęp

### 1.1 Założenia projektowe

Celem projektu jest zaimplementowanie aplikacji webowej pozwalającej użytkownikom na predykcję ceny używanego samochodu na podstawie dostarczonego przez niego zestawu cech. Tematyka projektu daje możliwość wykorzystania różnorodnych technologii z dziedziny uczenia maszynowego, rozwoju aplikacji webowych, komunikacji pomiędzy serwisami, architektury oprogramowania oraz bierania i przetwarzania danych. W celu zrealizowania przewidywanych funkcjonalności, aplikacja została podzielona na cztery komponenty, każdy z nich odpowiedzialny za realizację innego aspektu aplikacji.

### 1.2 Komponenty

- Aplikacja kliencka — Interfejs graficzny użytkownika.
- Pośrednik — Komponent pośredniczący w komunikacji pomiędzy aplikacją kliencką i serwisem predykcyjnym
- Komponent komunikacyjny — Komponent zawierający szyny danych, które są wykorzystywane do dostarczania i odbierania informacji od serwisu predykcyjnego
- Serwis predykcyjny — Komponent dokonujący predykcji na podstawie dostarczonych danych, z wykorzystaniem nauczonego modelu.

### 1.3 Konteneryzacja

Wszystkie komponenty zostały skonteneryzowane za pomocą narzędzi **Docker**<sup>1</sup> i **Docker Compose**<sup>2</sup>, co pozwala na uruchomienie projektu bez konieczności dodatkowej konfiguracji. **Obrazy**<sup>3</sup> **kontenerów**<sup>4</sup> dla aplikacji klienckiej oraz pośrednika zostały zdefiniowane za pomocą plików **Dockerfile**<sup>5</sup>, natomiast dla komponentu komunikacyjnego wykorzystano gotowe obrazy Apache Kafka i Zookeeper z rejestru Docker.io.

---

<sup>1</sup>Narzędzie do tworzenia, uruchamiania i zarządzania aplikacjami w izolowanych środowiskach zwanych kontenerami.

<sup>2</sup>Narzędzie usprawniające zarządzanie wieloma kontenerami jednocześnie.

<sup>3</sup>Gotowy do uruchomienia szablon do tworzenia kontenerów, zawierający system plików, aplikację i jej zależności.

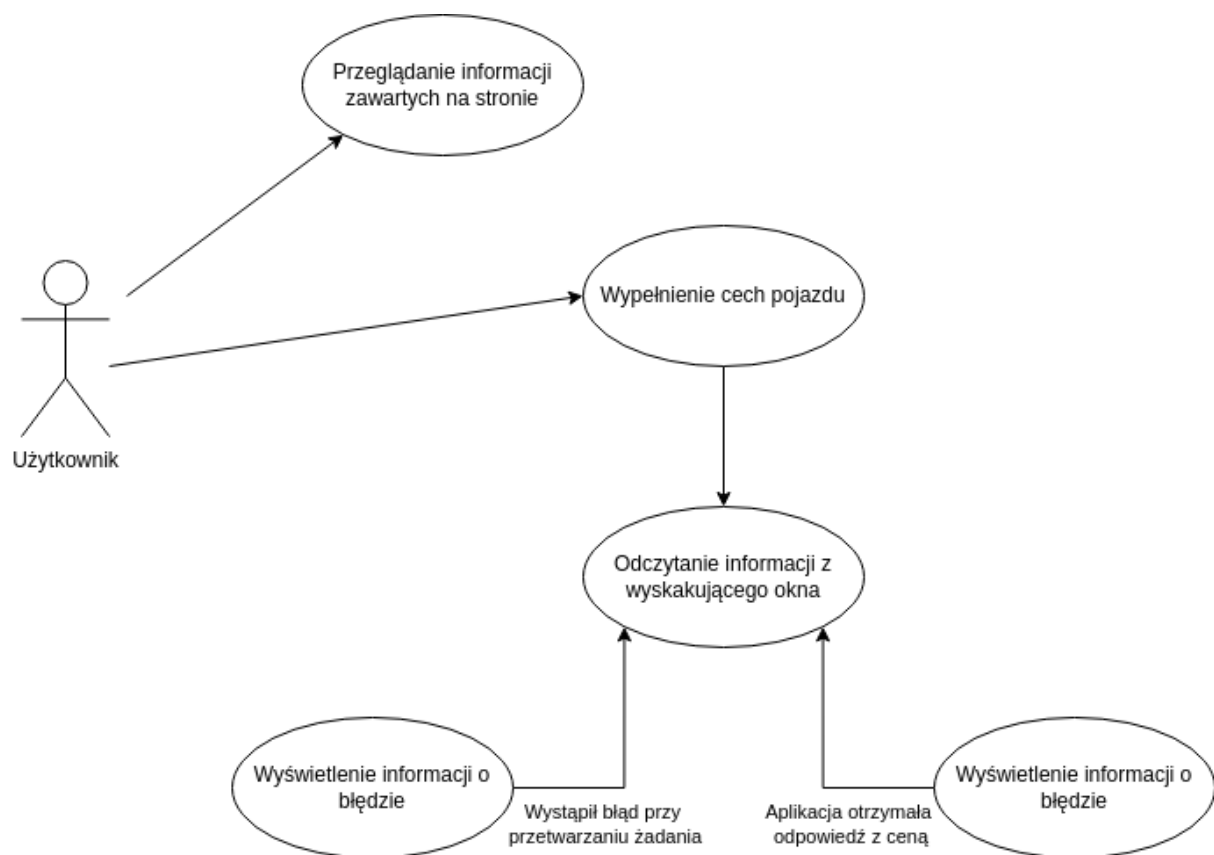
<sup>4</sup>Lekkie, izolowane środowisko uruchomieniowe, które zawiera aplikację wraz z jej zależnościami.

<sup>5</sup>Plik tekstowy zawierający zestaw instrukcji do zbudowania obrazu Docker.

# Rozdział 2

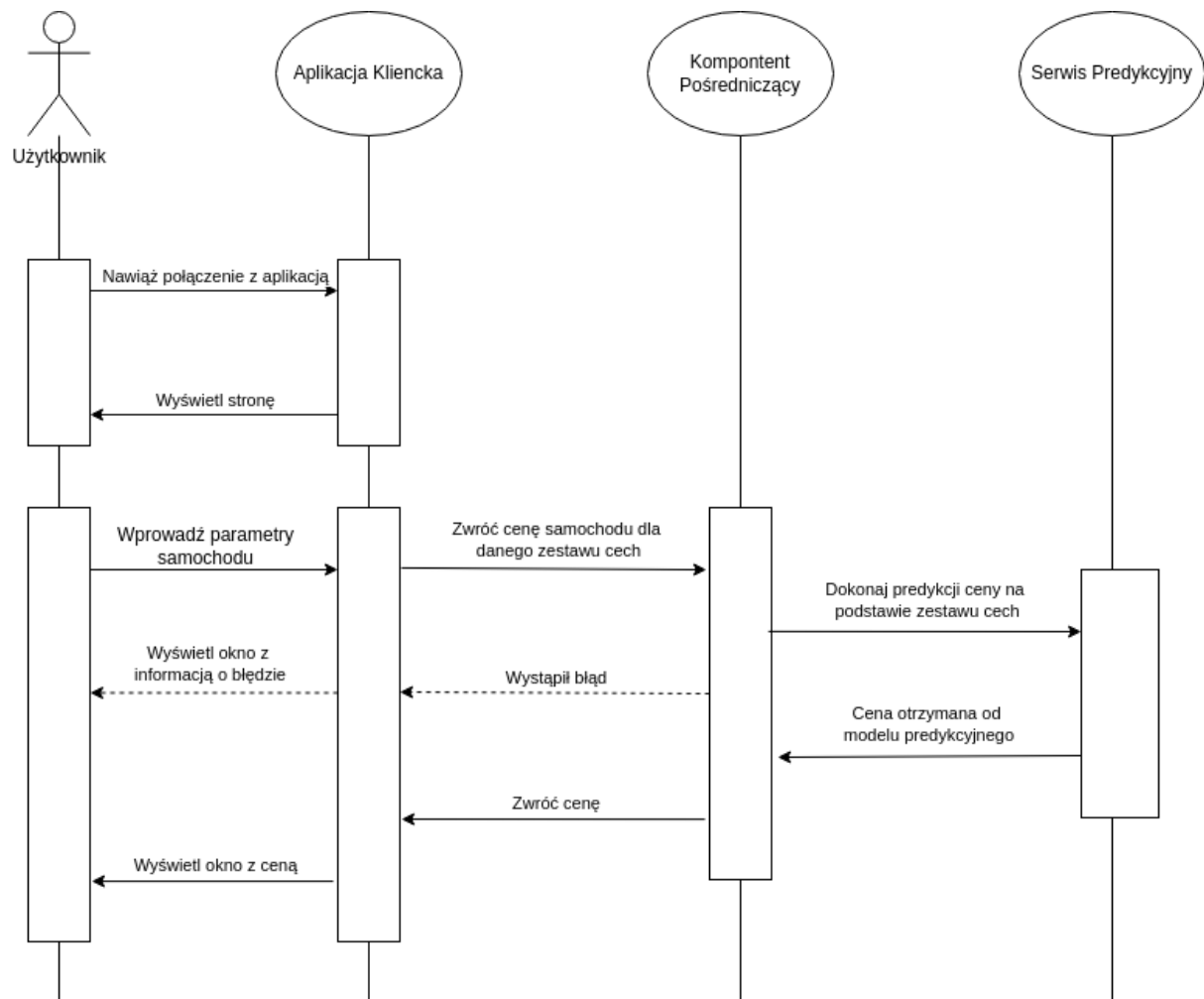
## Diagramy

### 2.1 Diagram przypadków użycia



Rysunek 2.1: Przebieg interakcji użytkownika z aplikacją

## 2.2 Diagram sekwencji zdarzeń



Rysunek 2.2: Przebieg operacji komponentów i działań użytkownika podczas procesu predykcji ceny samochodu

# Rozdział 3

## Aplikacja kliencka

### 3.1 Opis

Aplikacja kliencka stanowi pojedynczą stronę dostępną za pośrednictwem przeglądarki, udostępnianą pod adresem **localhost**<sup>1</sup>, na porcie **9091**. Strona zawiera informacje związane z aplikacją oraz pola do wprowadzania wartości, na podstawie których następnie dokonywana jest predykcja ceny samochodu. Aplikacja łączy się z komponentem middleware za pośrednictwem protokołu **HTTP**<sup>2</sup> w architekturze **REST**<sup>3</sup>.

### 3.2 Technologie

- React — Framework JavaScript do tworzenia interfejsów użytkownika w oparciu o komponenty.
- HTML — Język znaczników do tworzenia struktury strony internetowej.
- CSS — Język stylów wykorzystywany do definiowania wyglądu stron internetowych.
- JavaScript — Język programowania wykorzystywany do tworzenia dynamicznych i interaktywnych elementów stron internetowych.
- Axios — Biblioteka JavaScript służąca do wykonywania zapytań HTTP.
- Vite — Narzędzie do budowania i uruchamiania aplikacji front-endowych.

---

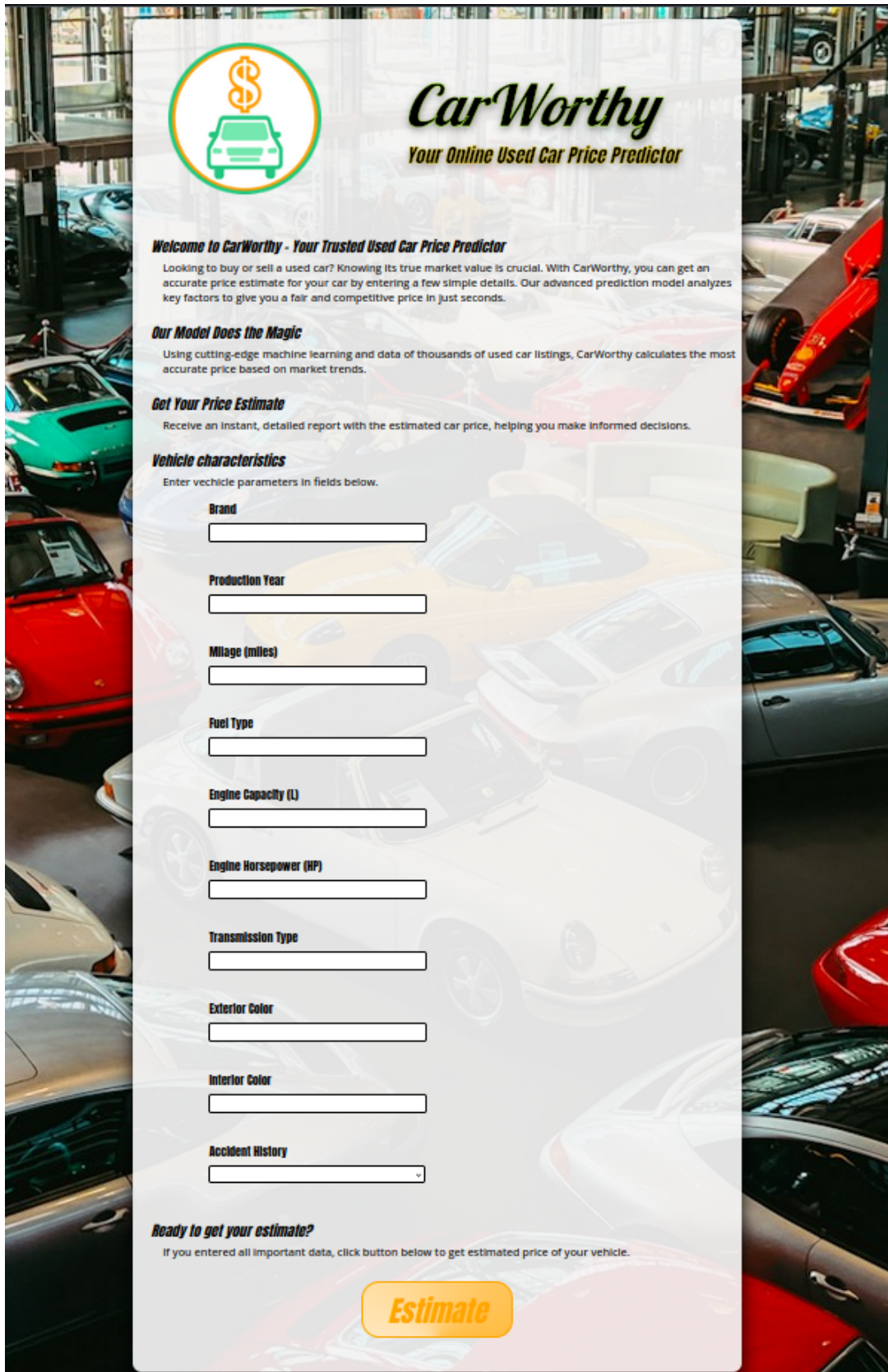
<sup>1</sup>loopback address — adres pętli zwrotnej, który jest wykorzystywany do komunikacji urządzenia z samym sobą.

<sup>2</sup>HyperText Transfer Protocol — protokół komunikacyjny używany do przesyłania danych w sieci.

<sup>3</sup>Representational State Transfer — architektura komunikacji oparta o protokół HTTP definiujący sposoby identyfikacji i manipulacji zasobami za pomocą zapytań HTTP.

## 3.3 Widoki aplikacji

### 3.3.1 Strona



The image shows a web application interface for "CarWorthy", an online used car price predictor. The background is a collage of various cars in a garage. The interface is a white overlay with a green and orange logo at the top left, featuring a car and a dollar sign. The title "CarWorthy" is in a large, stylized font, with the tagline "Your Online Used Car Price Predictor" below it. The main content area includes a welcome message, a description of the model's capabilities, and a section for entering vehicle characteristics. The form fields are labeled: Brand, Production Year, Milage (miles), Fuel Type, Engine Capacity (L), Engine Horsepower (HP), Transmission Type, Exterior Color, Interior Color, and Accident History. A large orange "Estimate" button is at the bottom right.

**CarWorthy**  
*Your Online Used Car Price Predictor*

**Welcome to CarWorthy - Your Trusted Used Car Price Predictor**

Looking to buy or sell a used car? Knowing its true market value is crucial. With CarWorthy, you can get an accurate price estimate for your car by entering a few simple details. Our advanced prediction model analyzes key factors to give you a fair and competitive price in just seconds.

**Our Model Does the Magic**

Using cutting-edge machine learning and data of thousands of used car listings, CarWorthy calculates the most accurate price based on market trends.

**Get Your Price Estimate**

Receive an instant, detailed report with the estimated car price, helping you make informed decisions.

**Vehicle characteristics**

Enter vehicle parameters in fields below.

**Brand**

**Production Year**

**Milage (miles)**

**Fuel Type**

**Engine Capacity (L)**

**Engine Horsepower (HP)**

**Transmission Type**

**Exterior Color**

**Interior Color**

**Accident History**

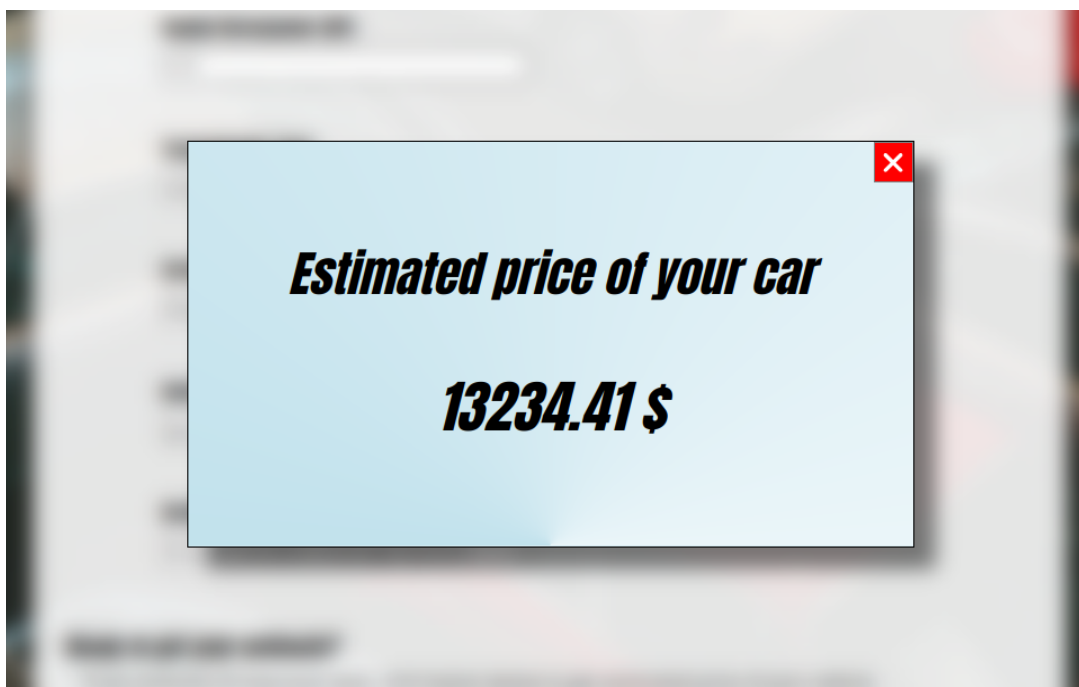
**Ready to get your estimate?**

If you entered all important data, click button below to get estimated price of your vehicle.

**Estimate**

Rysunek 3.1: Widok strony

### 3.3.2 Okno z ceną



Rysunek 3.2: Widok okna z ceną

### 3.3.3 Okno z błędem



Rysunek 3.3: Widok okna z błędem



# Rozdział 4

## Komponent pośredniczący

### 4.1 Opis

Komponent pośredniczący pełni rolę pośrednika pomiędzy aplikacją kliencką i serwisem predykcyjnym. Otrzymywane od **frontendu**<sup>1</sup> dane w formie **JSON**<sup>2</sup> są w tym komponencie przetwarzane na wiadomości w formacie odpowiadającym wejściu modelu, z uwzględnieniem procesu **kodowania liczbowego**<sup>3</sup> pól. Otrzymane w tym procesie wiadomości zapisywane są na **temat**<sup>4</sup> wejściowy Kafki. Pośrednik jest również odpowiedzialny za odczytywanie danych z tematu wyjściowego i przekazywanie uzyskanych z nich informacji do klienta.

### 4.2 Technologie

- Java — Obiektowy język programowania.
- SpringBoot — Framework dla języka Java nastawiony na wytwarzanie aplikacji webowych i mikroservisów
- Gradle — Narzędzie do automatyzacji budowania projektów.

---

<sup>1</sup>Część aplikacji, z którą użytkownik wchodzi w bezpośrednią interakcję, w tym wszystko co widzi oraz elementy wizualne i interaktywne.

<sup>2</sup>JavaScript Object Notation — format danych zapewniający kompaktowe rozmiary i jest czytelny dla ludzi i maszyn.

<sup>3</sup>Technika zamiany wartości danych tekstowych na wartości liczbowe, poprzez przypisanie unikalnej liczby każdej unikalnej wartości tekstowej.

<sup>4</sup>Podstawowy komponent Apache Kafka służący do kategoryzacji napływających wiadomości.

# Rozdział 5

## Komponent komunikacyjny

### 5.1 Opis

Komponent komunikacyjny odpowiedzialny jest za transport danych pomiędzy komponentem pośredniczącym i serwisem predykcyjnym. Wykorzystuje w tym celu skonteneryzowany **broker**<sup>1</sup> wiadomości Apache Kafka wraz z dwoma tematami input oraz output, wykorzystywanych odpowiednio do gromadzenia danych odczytywanych przez serwis predykcyjny i gromadzenia danych odczytywanych przez pośrednika. Do zarządzania brokerem wykorzystywany jest Apache Zookeeper.

### 5.2 Technologie

- Apache Kafka — Platforma przetwarzania danych w czasie rzeczywistym.
- Apache Zookeeper — Usługa koordynacyjna systemów rozproszonych.

---

<sup>1</sup>Serwer Apache Kafka zawierający dane należące do tematów i partycji, na które może być podzielony temat

# Rozdział 6

## Serwis predykcyjny

### 6.1 Opis

Zadaniem serwisu predykcyjnego jest dokonanie predykcji ceny samochodu na podstawie dostarczonego zestawu cech. W tym celu wykorzystuje gotowy, zapisany model przygotowany przy użyciu modułu SparkML. Dane przekazywane do modelu są odczytywane z tematu input za pomocą frameworka Spark. Cena zwrócona przez model zostaje zapisana na temat wyjściowy output.

### 6.2 Technologie

- Python — Język skryptowy.
- Apache Spark — Framework do sprawnego przetwarzania zbiorów danych w pamięci.
- Apache SparkML — Moduł Apache Spark przeznaczony do uczenia maszynowego.