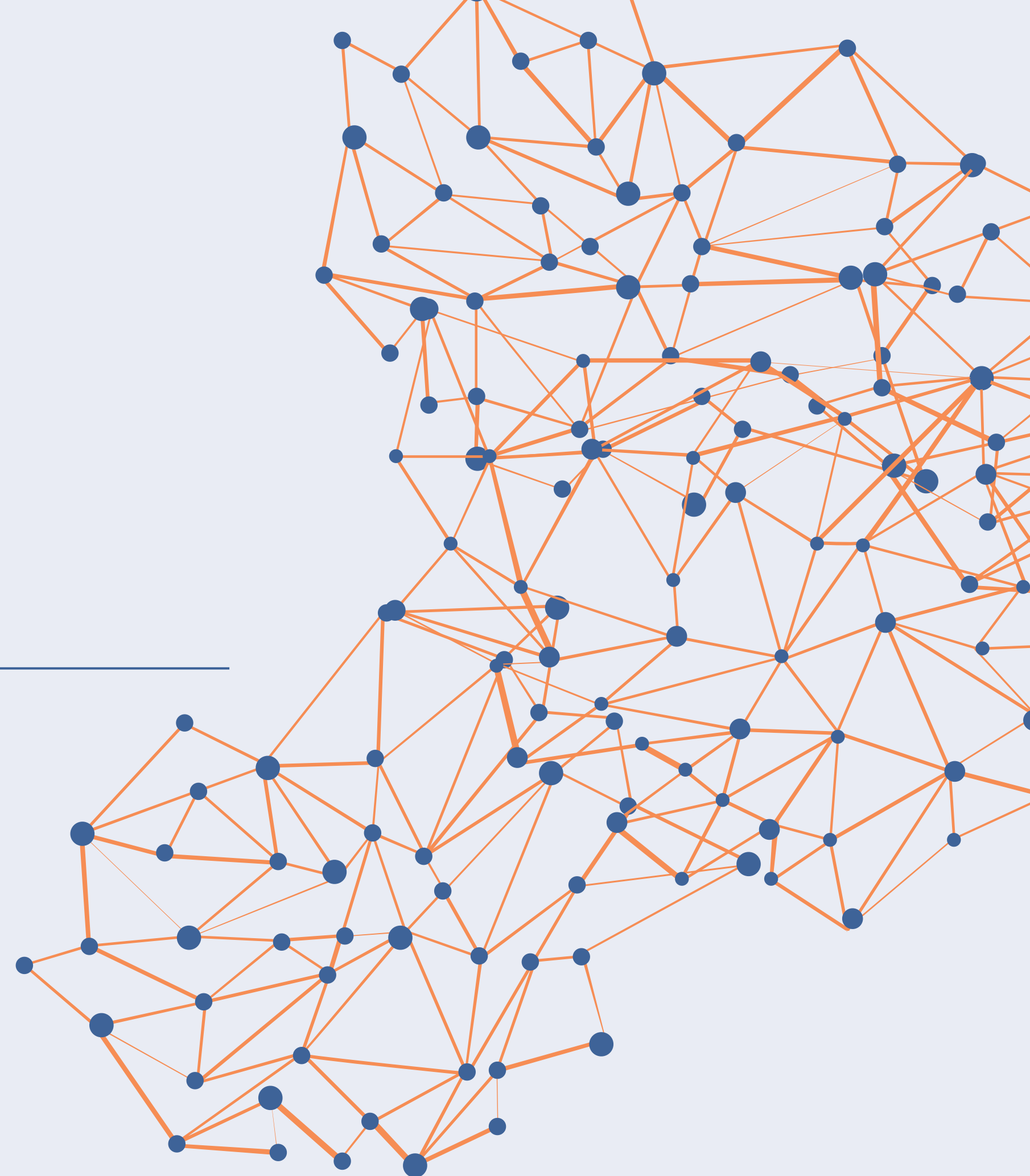


# Narzędzia i metody prototypowania systemów geoprzestrzennych

---

Inżynieria oprogramowania

Aleksandra Buńko  
Julia Goska



# Spis Treści

**1 Wprowadzenie**

**2 Czym są systemy geoprzestrzenne?**

**3 Dlaczego prototypowanie GIS jest wyzwaniem?**

**4 Streamlit – narzędzie interaktywnego interfejsu**

**5 Właściwości i zalety Streamlit**

**6 Folium – narzędzie GIS w Pythonie**

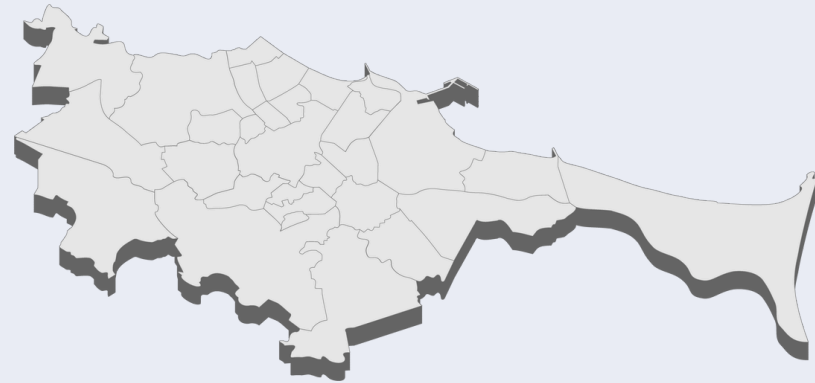
**7 Właściwości i zalety Folium**

**8 Integracja narzędzi w procesie prototypowania**

**9 Przykłady zastosowań**

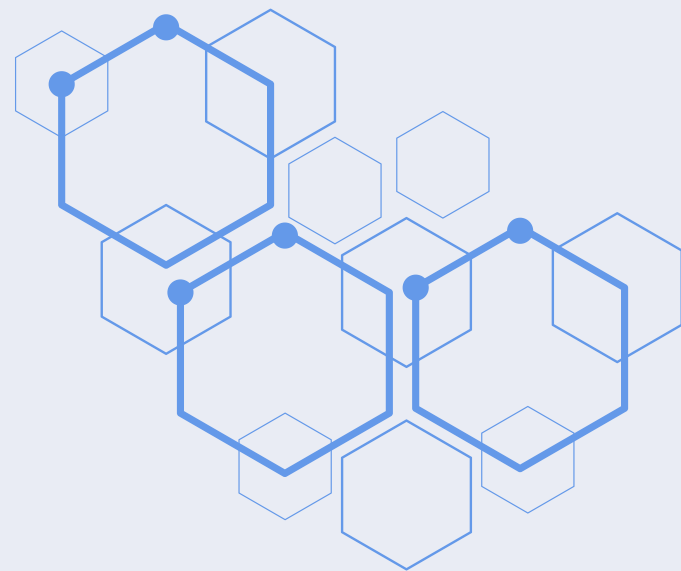
**10 Podsumowanie i wnioski**

# Wprowadzenie



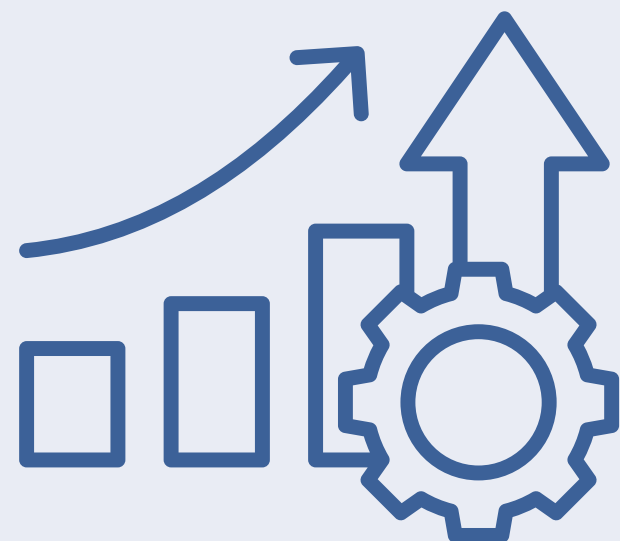
## Dane Geoprzestrzenne

To dane, które oprócz wartości opisowej zawierają informację o położeniu na Ziemi, mogą odnosić się do punktów, tras, obszarów oraz zjawisk środowiskowych



## GIS — Systemy Informacji Geograficznej

Technologia pozwalająca łączyć dane z lokalizacją. Wykorzystuje takie narzędzia jak m.in. streamlit oraz foliom



## Prototypowanie GIS

Umożliwia szybkie tworzenie wstępnych wersji systemów geoprzestrzennych. Pozwala testować funkcje i analizować dane jeszcze przed pełnym wdrożeniem. Ogranicza koszty i przyspiesza pracę nad rozwiązaniem

# Czym są systemy geoprzestrzenne?

Systemy geoprzestrzenne GIS to technologie pozwalające na:



**Gromadzenie danych związanych z lokalizacją**



**Analizę przestrzenną**



**Wizualizacje wyników na mapach**



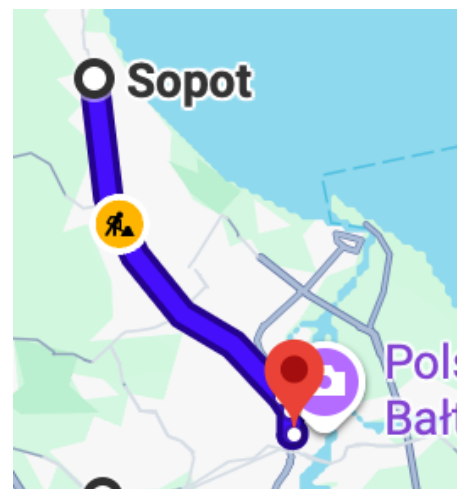
**Podjęmowanie decyzji na podstawie lokalizacji**

**Przykłady danych:**

## Punkty



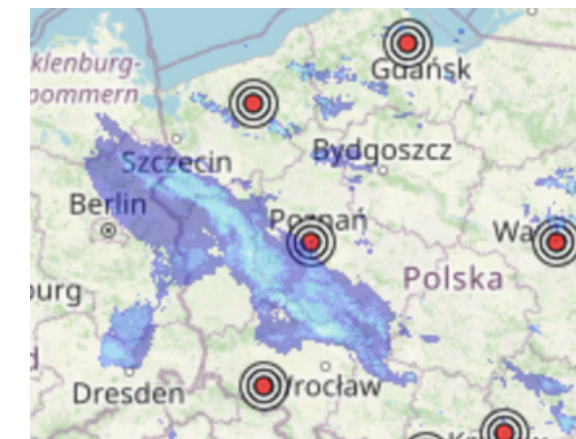
## Linie



## Obszary



## Zjawiska przestrzenne



# Dlaczego prototypowanie GIS jest wyzwaniem?

GIS jest trudny, bo łączy największe wyzwania danych: skalę, dokładność i interakcję.  
Dlatego prototypowanie musi być szybkie, webowe i elastyczne.



Dane



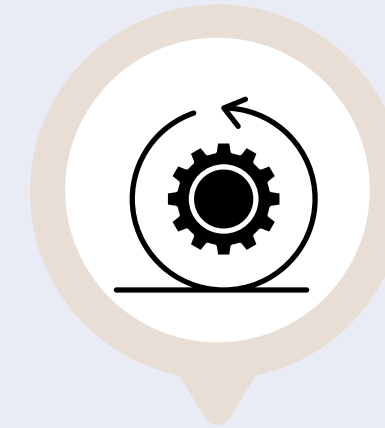
Interakcja



Efekt na mapie



Wnioski



Iteracja

Ogromna skala danych  
Złożone formaty  
danych przestrzennych

Konwersje między  
układami  
współrzędnych

Aktualność danych

Niejednorodność źródeł

Optymalizacja

Użytkownik wybiera  
obiekt na mapie

Dynamiczne  
parametry analizy

Różne poziomy  
szczegółowości map

Testy użyteczności

Responsywność

Wydajność w wizualizacji

Warstwy GIS

Czytelność

Renderowanie po każdej  
zmianie danych

Przestrzenna dokładność  
odzworowania

Analiza zależności  
przestrzennych

Wpływ błędów  
lokalizacji

Interpretacja przez  
osoby nietechniczne

Zaufanie do wyników

Zmieniające się  
założenia użytkownika

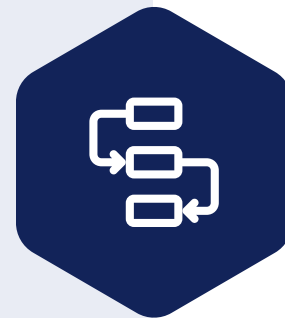
Szybkie zmiany  
funkcjonalności

Niska tolerancja  
klienta na czekanie

Koszt iteracji przy  
narzędziach  
desktopowych

# Streamlit

Framework do budowania interaktywnych aplikacji webowych w języku Python, bez potrzeby pisania kodu front-end (HTML/CSS/JavaScript).



## Architektura działania Streamlit

```
import streamlit as st
import pandas as pd
import numpy as np

@st.cache_data
def load_data():
    rng = np.random.default_rng(42)
    return pd.DataFrame({
        "lat": 52 + rng.normal(0, 0.1, 500),
        "lon": 21 + rng.normal(0, 0.1, 500)
    })

st.title("Interaktywna aplikacja Streamlit")

threshold = st.slider("Liczba punktów", 50, 500, 200)
data = load_data().head(threshold)

st.dataframe(data)
st.map(data)
```

1. Model klient–serwer

2. Deklaracyjny model budowy interfejsu

3. Mechanizm reaktywnego wykonywania skryptu

4. Brak trwałego stanu aplikacji  
model stateless  
– cache’owanie (st.cache, st.cache\_data)  
– kontrolowane przechowywanie stanu (st.session\_state)



1

## Minimalizacja kosztu implementacji UI

Brak konieczności stosowania technologii front-endowych  
— interfejs tworzony bezpośrednio z kodu Python.

2

## Reaktywne przetwarzanie danych

Automatyczna aktualizacja wyników wizualizacji przy  
każdej zmianie parametrów wejściowych.

3

## Webowa dostępność i wysoka użyteczność

Aplikacje uruchamiane w przeglądarce umożliwiają  
łatwą walidację koncepcji z użytkownikami.

4

## Integracja z komponentami analitycznymi

Wsparcie dla bibliotek obliczeniowych i GIS umożliwia  
implementację pełnego modelu w jednym środowisku.

5

## Model open-source

Darmowa dostępność oraz szybki rozwój narzędzia  
dzięki aktywnej społeczności.

**Właściwości  
i zalety Streamlit**

# Czym jest Folium?

Biblioteka Python do map interaktywnych

01

**Most między Pythonem a JavaScript**

02

**Generator interaktywnych obiektów HTML**

03

**Narzędzie lekkiego GIS**



```
import folium

# Inicjalizacja mapy (Centrum: Warszawa)
m = folium.Map(location=[52.229, 21.012], zoom_start=13)

# Dodanie znacznika (Twoja inwestycja)
folium.Marker(
    [52.229, 21.012],
    popup="Lokalizacja Inwestycji",
    icon=folium.Icon(color="green", icon="sun")
).add_to(m)
```



## Architektura Zdarzeniowa

Kluczowa cecha dla nas: mapa obsługuje zdarzenia (np. Click, Hover). Dzięki temu służy nie tylko do wyświetlania danych, ale też do ich wprowadzania (geokodowanie kliknięć).

## Architektura Kafelkowa

Mapa nie jest pobierana jako jeden gigantyczny plik. Folium wykorzystuje standard XYZ Tiles (np. OpenStreetMap), doczytując dynamicznie tylko te fragmenty świata, które ogląda użytkownik.

## Integracja z Pandas

Bezpośrednie wizualizowanie ramek danych (DataFrame) na mapie. Folium natywnie obsługuje formaty używane w analityce danych bez ręcznej konwersji.

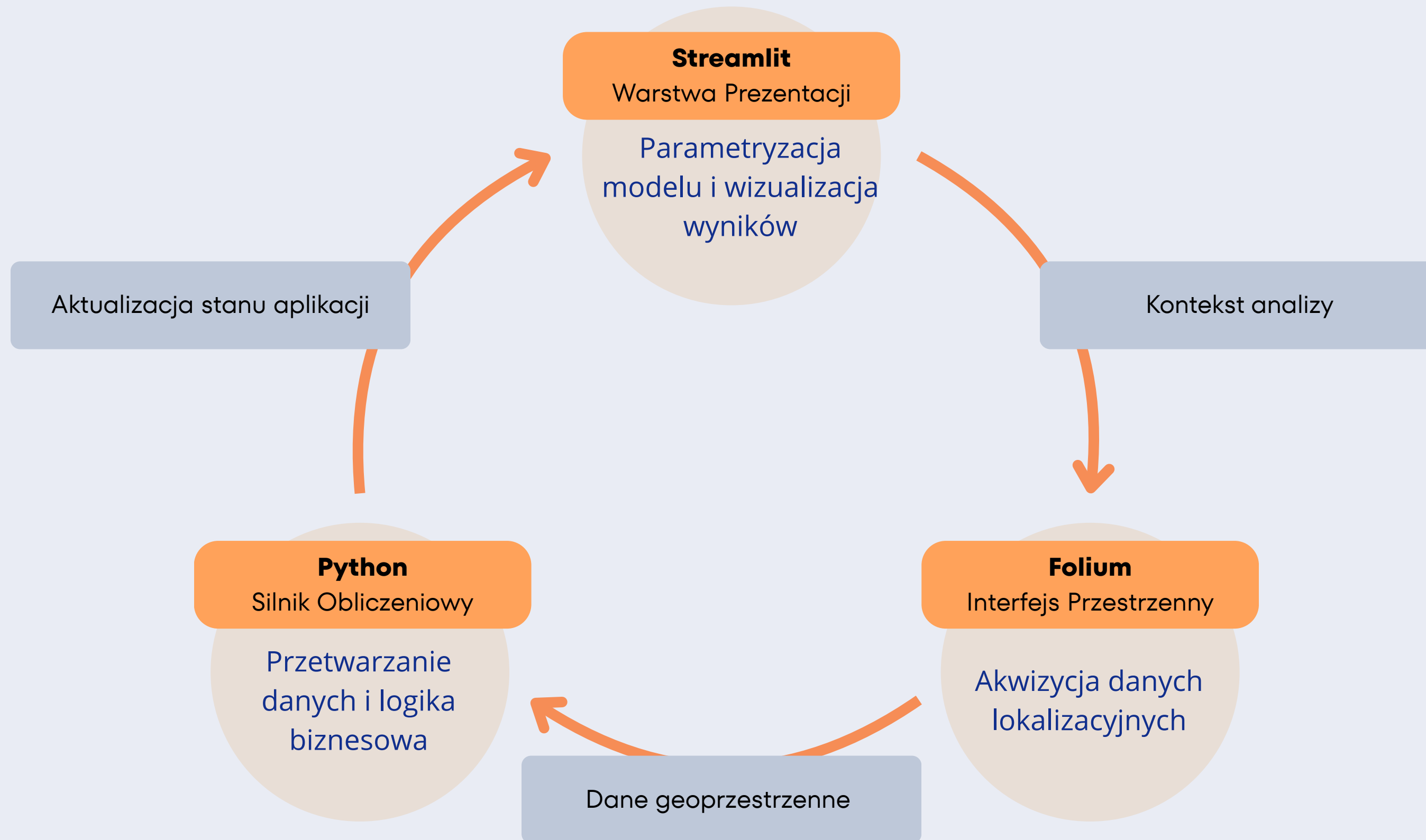
## Abstrakcja Warstwy Frontend

Folium działa jak kompilator: przyjmuje kod Python i wypływa gotowy, zoptymalizowany kod JavaScript. Pozwala backendowcom tworzyć frontend bez znajomości JS.



**Właściwości  
i zalety**

# Integracja narzędzi w procesie prototypowania



# Przykłady zastosowań narzędzi geoprzestrzennych



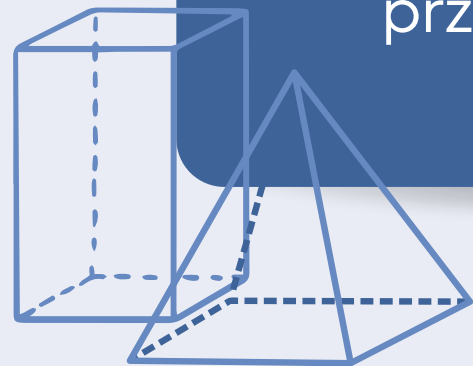
Energetyka i  
optymalizacja  
lokalizacji



Logistyka i trasy  
transportowe



Narzędzia  
edukacyjne



Planowanie  
przestrzenne



Mapy zagrożeń i  
analiz  
środowiskowych

01

02

03

04

05

# Podsumowanie



Odchodzimy od statycznych map na rzecz interaktywnych systemów decyzyjnych. Kluczem jest połączenie twardych danych z intuicyjną wizualizacją.

**Nowoczesny GIS**



Połączenie Streamlit i Folium w ekosystemie Python tworzy kompletne środowisko do budowy aplikacji analitycznych bez narzutu frontendowego.

**Stos Technologiczny**



Wybrane narzędzia znacząco skracają czas Time-to-Market. Pozwalają na natychmiastową weryfikację założeń biznesowych (MVP) niskim kosztem.

**Rapid Prototyping**



Ten stos technologiczny jest idealny dla projektów inżynierskich wymagających złożonych obliczeń (ekonometria) prezentowanych w prostej formie.

**Moc Obliczeniowa**



**Dziękujemy za uwagę**

Źródła:

- Streamlit Inc. (2024). Streamlit Documentation. Dostępne pod adresem: [docs.streamlit.io](https://docs.streamlit.io)
- Python Visualization. (2023). Folium 0.15.0 Documentation. Dostępne pod adresem: [python-visualization.github.io/folium/](https://python-visualization.github.io/folium/)
- OpenStreetMap Contributors. (2024). OpenStreetMap Data. Dostępne pod adresem: [openstreetmap.org](https://openstreetmap.org)