

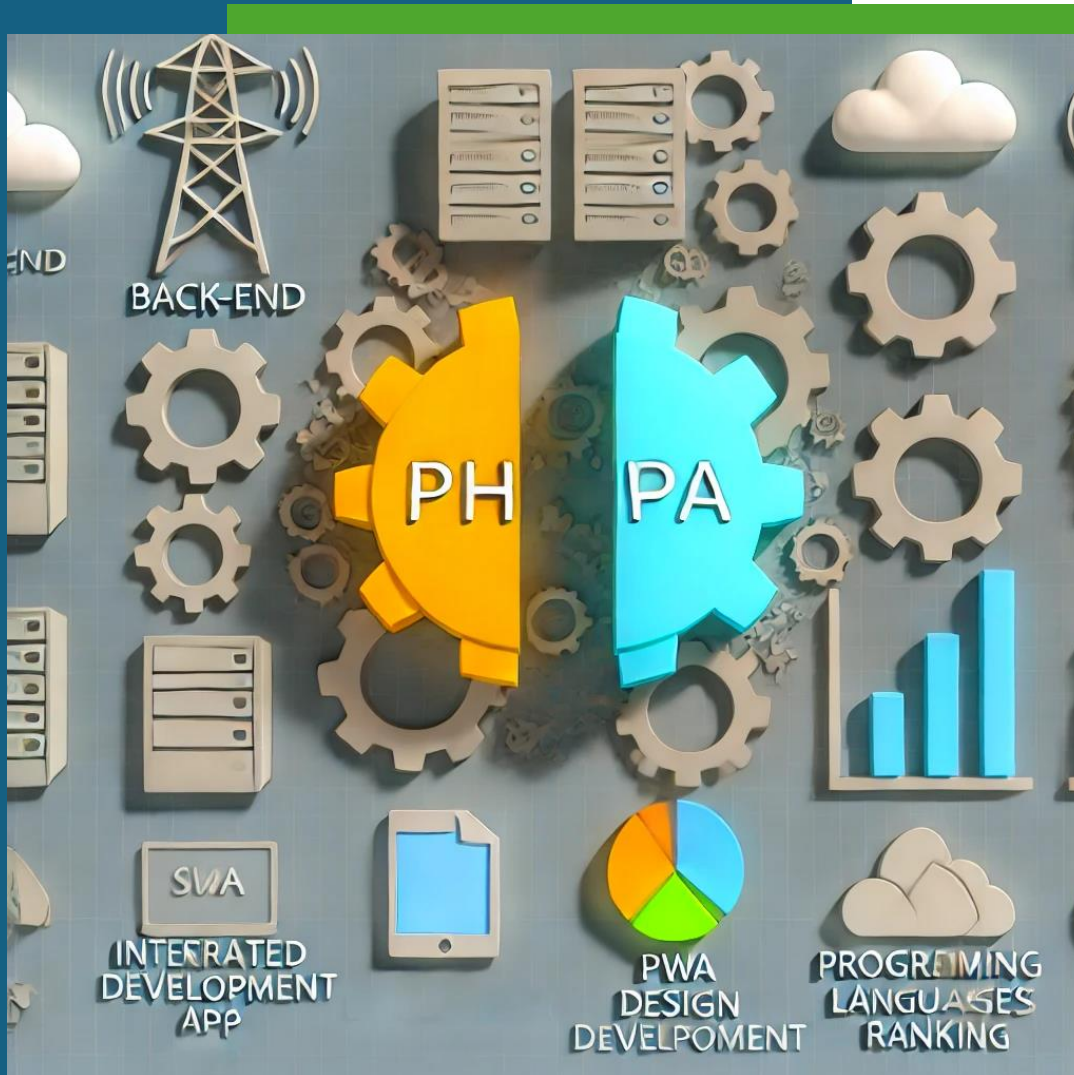
Temat 5 .

Logika i dane aplikacji
(back-end, baza danych)

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

TECHNIKI INTERNETOWE



Zagadnienia:

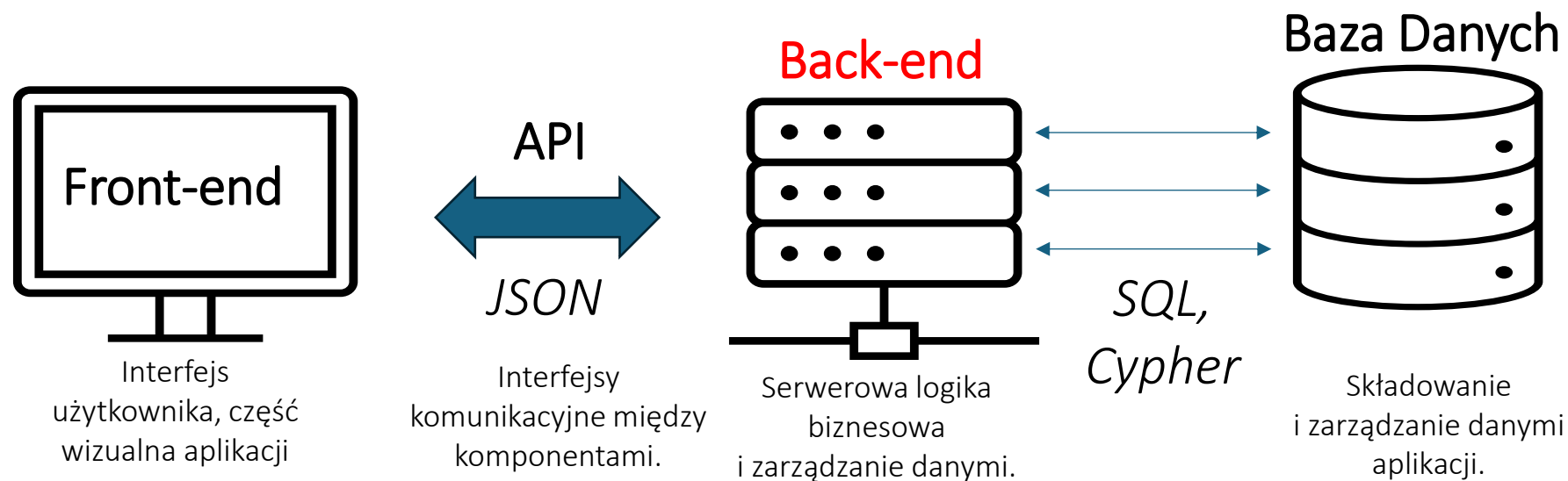
- Algorytmy wyszukiwarek
- Back-end
- PHP
- MPA, SPA
- Aplikacje mobilne
- PWA
- Wzorce projektowe
- Bazy danych
- Zintegrowane środowiska deweloperskie

Back-end

Back-end

to część serwerowa, która jest odpowiedzialna za zarządzanie danymi, logiką biznesową, autoryzacją użytkowników, a także komunikację z bazą danych i innymi serwisami zewnętrznymi.

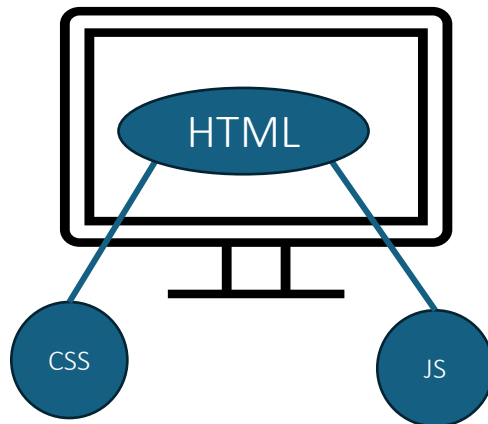
Działa "za kulisami" aplikacji, obsługując żądania od front-end (interfejsu użytkownika) i wysyłając odpowiedzi.



Back-end

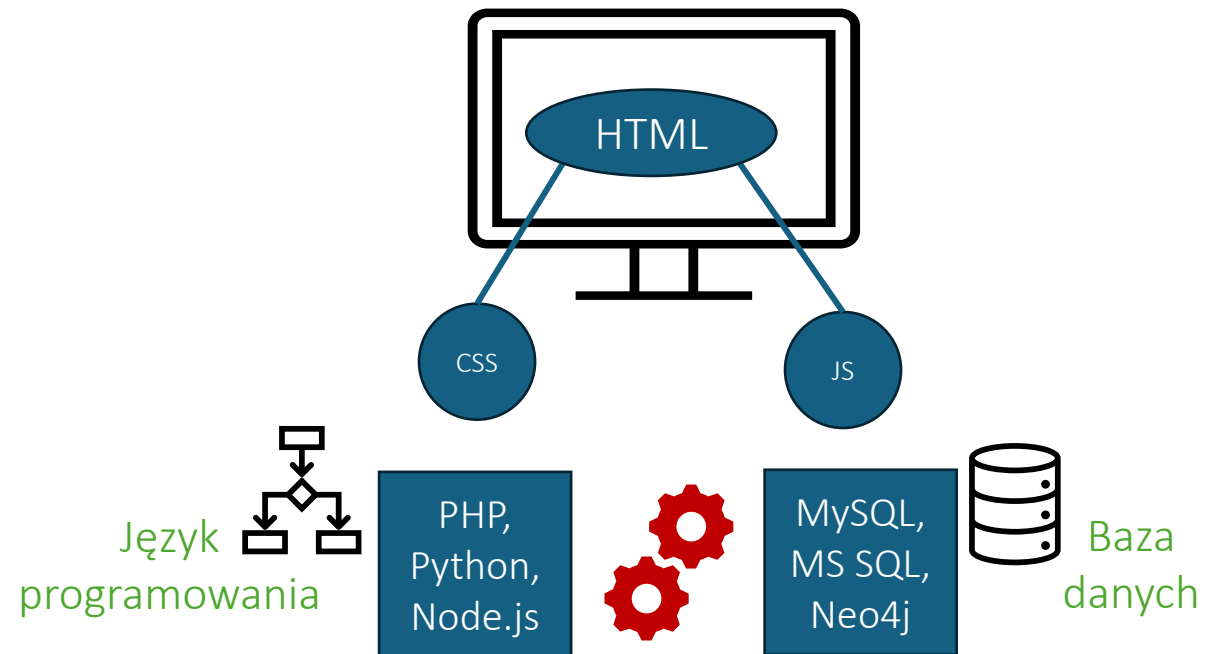
Strony

Statyczne



Zawartość stała, niezmienna dla każdego użytkownika

Dynamiczne



Zawartość zmienia się w odpowiedzi na działania użytkownika

Back-end

Statyczne vs dynamiczne strony

Kategoria	Statyczne Strony WWW	Dynamiczne Strony WWW
Technologie	HTML, CSS, czasami JavaScript	HTML, CSS, JavaScript, języki skryptowe (np. PHP, Python), bazy danych (np. MySQL, PostgreSQL)
Szybkość	Szybsze ładowanie, mniej zasobów	Wolniejsze ładowanie, więcej zasobów
Bezpieczeństwo	Mniej punktów narażonych na ataki	Więcej punktów narażonych na ataki (bazy danych, serwery)
Łatwość wdrożenia	Łatwiejsze do hostowania, mniej skomplikowane	Bardziej złożone, wymagają więcej zasobów i konfiguracji
Interaktywność	Ograniczona, brak interakcji użytkownika	Wysoka, użytkownik może wchodzić w interakcję z treścią
Aktualizacja	Ręczna aktualizacja treści	Automatyczna aktualizacja treści, np. wiadomości, kursy walut
Przykłady	Strony wizytówki, blogi bez interakcji użytkowników, strony informacyjne, portfolio	Serwisy społecznościowe, sklepy internetowe, platformy e-learningowe, portale informacyjne
Złożoność	Prostota	Złożoność

Back-end

Języki skryptowe

to klasy języków programowania wysokopoziomowych, których podstawową cechą jest interpretacja kodu w czasie wykonywania, a nie kompilacja do formy binarnej przed uruchomieniem.

Oznacza to, że kod źródłowy – zwany skrypcem – jest zwykle przetwarzany „na bieżąco” przez interpreter, co umożliwia szybką iterację, prototypowanie oraz łatwe łączenie istniejących komponentów oprogramowania.

Język Skryptowy	Serwer WWW	Framework	Rok
Python	Apache, Nginx	Django, Flask, FastAPI	1991
PHP	Apache, Nginx	Laravel, Symfony, CodeIgniter	1995
Java	Apache Tomcat, Jetty	Spring Boot, Grails	1995
C#	IIS (Internet Information Services)	ASP.NET Core, Nancy	2000
Perl	Apache	Catalyst, Dancer	1987
Node.js	Apache, Nginx	Express.js, Koa, Hapi	2009

PHP

PHP - Hypertext Preprocessor

<https://www.php.net/>

jest językiem skryptowym stosowanym głównie do tworzenia dynamicznych stron internetowych i aplikacji webowych.

PHP jest wykonywany po stronie serwera, co oznacza, że kod PHP jest przetwarzany przez serwer przed wysłaniem wyniku do przeglądarki użytkownika.



Wbudowane wsparcie dla baz danych:

obsługuje wiele popularnych baz danych, takich jak MySQL, PostgreSQL, SQLite.

Łatwa integracja z HTML:

można łatwo osadzać w kodzie HTML, co ułatwia tworzenie dynamicznych stron internetowych.

Wieloplatformowy:

działa na wielu systemach operacyjnych, w tym Windows, Linux, macOS.

Bogata biblioteka funkcji:

posiada rozbudowaną bibliotekę wbudowanych funkcji, które ułatwiają wykonywanie wielu zadań programistycznych.

Możliwość generowania grafik:

może tworzyć i modyfikować obrazy dzięki rozszerzeniu GD Library, co jest przydatne w tworzeniu dynamicznych grafik na stronach internetowych.

Odbieranie przesyłanych plików:

umożliwia przysyłanie i przetwarzanie plików od użytkowników.

PHP

Skrypt

http://localhost/index.php?a=3&b=5

Wywołanie w przeglądarce

Odbieranie danych z GET

Dodawanie liczb

Modyfikacja daty

Pętla od 1 do 10

```
<!DOCTYPE html>
<html>
<head>
    <title>Prosty Skrypt PHP</title>
</head>
<body>
    <?php
        $a = isset($_GET['a']) ? intval($_GET['a']) : 0;
        $b = isset($_GET['b']) ? intval($_GET['b']) : 0;
        $suma = $a + $b;
        $aktualnaData = new DateTime();
        $aktualnaData->modify("+$suma days");
        echo "Data po dodaniu sumy ($suma) dni: " . $aktualnaData->format('Y-m-d') . "<br>";
        for ($i = 1; $i <= 10; $i++) {
            echo "$i<br>";
        }
    ?>
</body>
</html>
```

PHP

Zmienne serwera

to zmienne globalne, które są automatycznie ustawiane przez serwer webowy i dostarczają informacji o środowisku serwera, klienta, a także o żądaniach HTTP.

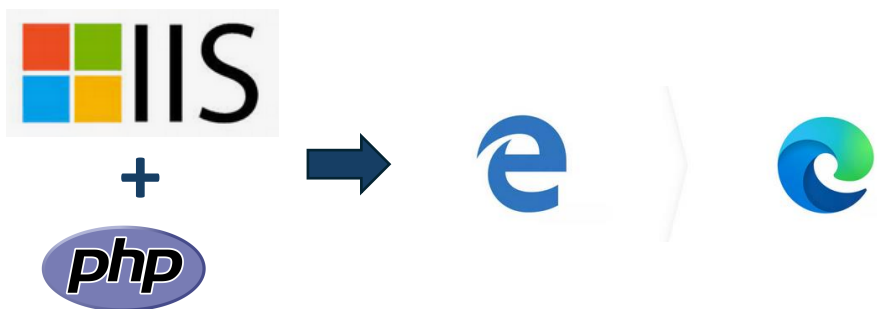
Są one dostępne w superglobalnej tablicy `$_SERVER`.

Zmienna	Opis
<code>\$_SERVER['PHP_SELF']</code>	Zawiera ścieżkę do aktualnie wykonywanego skryptu.
<code>\$_SERVER['SERVER_NAME']</code>	Nazwa hosta serwera, na którym uruchomiony jest skrypt.
<code>\$_SERVER['HTTP_HOST']</code>	Zawiera zawartość nagłówka HTTP Host wysłanego przez klienta.
<code>\$_SERVER['HTTP_USER_AGENT']</code>	Zawiera ciąg określający przeglądarkę klienta.
<code>\$_SERVER['SERVER_ADDR']</code>	Adres IP serwera, na którym uruchomiony jest skrypt.
<code>\$_SERVER['REMOTE_ADDR']</code>	Adres IP komputera klienta, który żądał strony.
<code>\$_SERVER['REQUEST_METHOD']</code>	Metoda żądania użyta do uzyskania dostępu do strony (np. GET, POST).
<code>\$_SERVER['QUERY_STRING']</code>	Ciąg zapytania, jeśli jest obecny w adresie URL.
<code>\$_SERVER['HTTP_REFERER']</code>	Adres URL strony, z której przyszedł użytkownik (jeśli jest dostępny).

PHP

Dane o użytkowniku

W środowisku serwerowym opartym na IIS (Internet Information Services) z włączonym uwierzytelnianiem zintegrowanym systemu Windows, PHP może udostępniać informację o aktualnie zalogowanym użytkowniku domenowym za pośrednictwem zmiennej serwera `$_SERVER['LOGON_USER']`

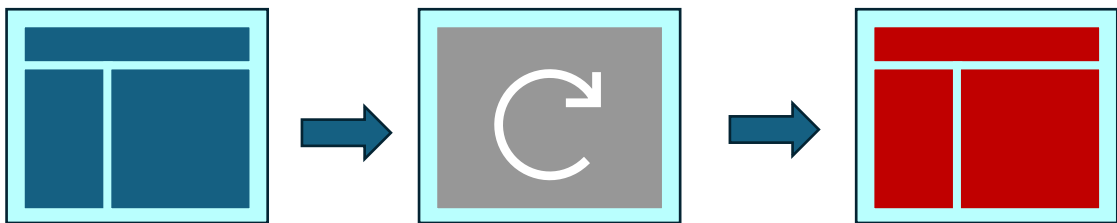


```
<?php
if (isset($_SERVER['LOGON_USER'])) {
    echo "Zalogowany użytkownik: " . $_SERVER['LOGON_USER'];
} else {
    echo "Brak informacji o zalogowanym użytkowniku.";
}
?>
```

MPA, SPA

MPA (Multi-Page Application)

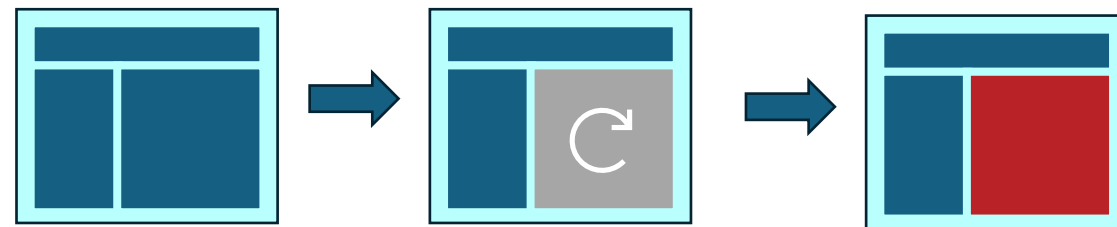
to tradycyjny rodzaj aplikacji webowej, w której każda zmiana lub interakcja użytkownika prowadzi do przeładowania całej strony lub załadowania nowej strony. Każda strona w MPA jest oddzielnym dokumentem HTML, który jest pobierany z serwera. Chociaż MPA może być mniej responsywna w porównaniu do SPA ze względu na częste przeładowania stron, jest ona łatwiejsza do zarządzania w kontekście SEO i może być bardziej odpowiednia dla aplikacji z dużą ilością treści.



SPA (Single Page Application)

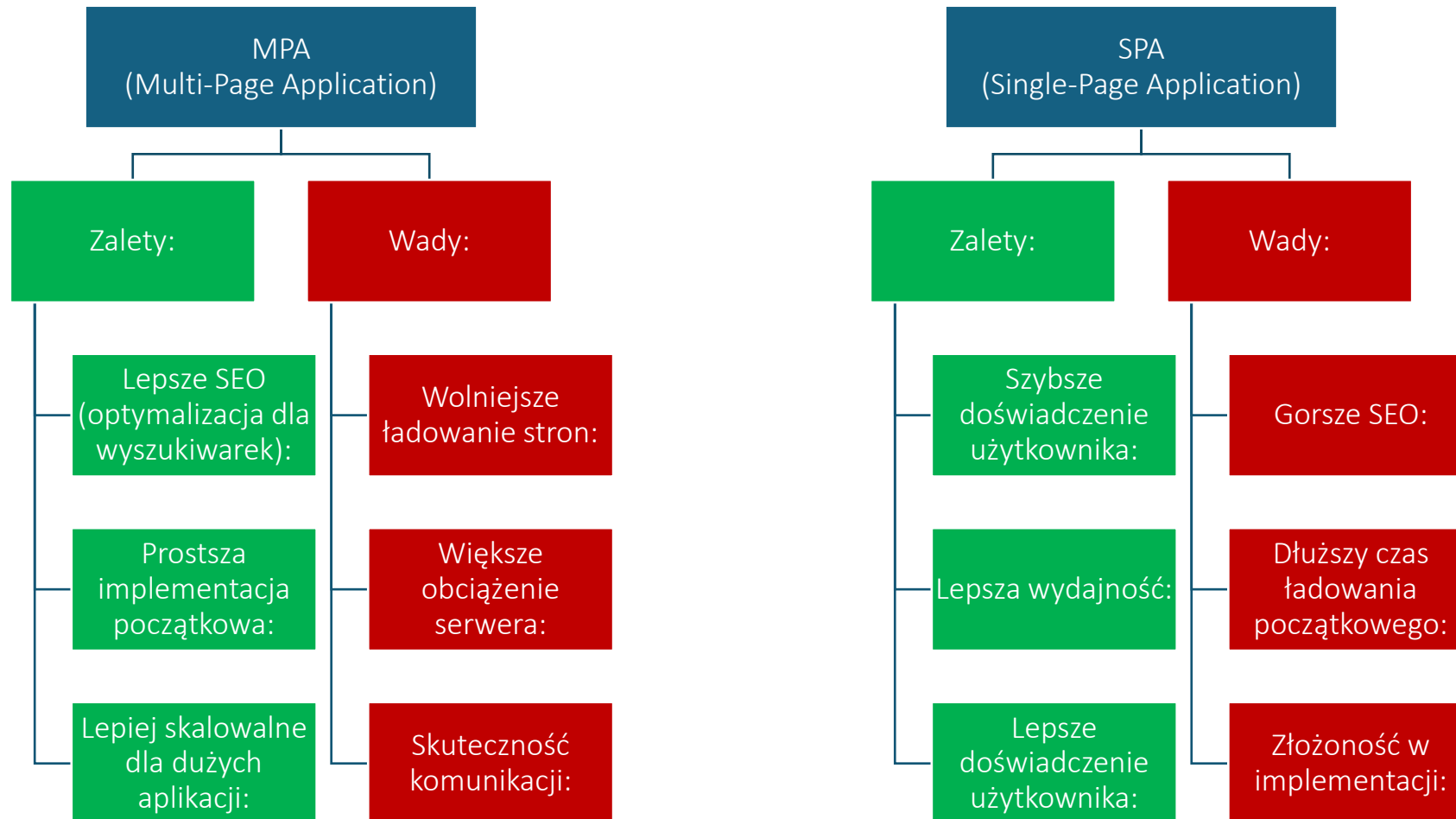
to rodzaj aplikacji webowej, która ładuje pojedynczy plik HTML i dynamicznie aktualizuje jego zawartość w miarę interakcji użytkownika, zamiast przeładowywać całą stronę.

Dzięki temu SPA oferują szybsze i bardziej responsywne doświadczenie użytkownika, ponieważ komunikacja z serwerem odbywa się asynchronicznie za pomocą AJAX lub WebSocketów, co pozwala na pobieranie tylko niezbędnych danych.



MPA, SPA

MPA vs SPA



Aplikacje mobilne

Aplikacje mobilne

to programy zaprojektowane do działania na urządzeniach mobilnych, takich jak smartfony i tablety. Mogą być preinstalowane na urządzeniu lub pobierane i instalowane przez użytkowników z różnych sklepów z aplikacjami.

Aplikacje Natywne (Native Apps):

- Tworzone specjalnie dla konkretnego systemu operacyjnego (iOS, Android).

Aplikacje Hybrydowe (Hybrid Apps):

- Łączą elementy aplikacji natywnych i webowych, uruchamiane wewnątrz przeglądarki osadzonej w aplikacji natywnej.

Aplikacje Webowe (Web Apps):

- Strony internetowe dostosowane do wyświetlania na urządzeniach mobilnych, działające w przeglądarce.

Aplikacje Progresywne (PWA, Progressive Web Apps):

- Nowoczesne aplikacje webowe, które łączą zalety aplikacji webowych i natywnych, działające w przeglądarkach z funkcjami offline i instalacji.

Aplikacje mobilne

Architektura

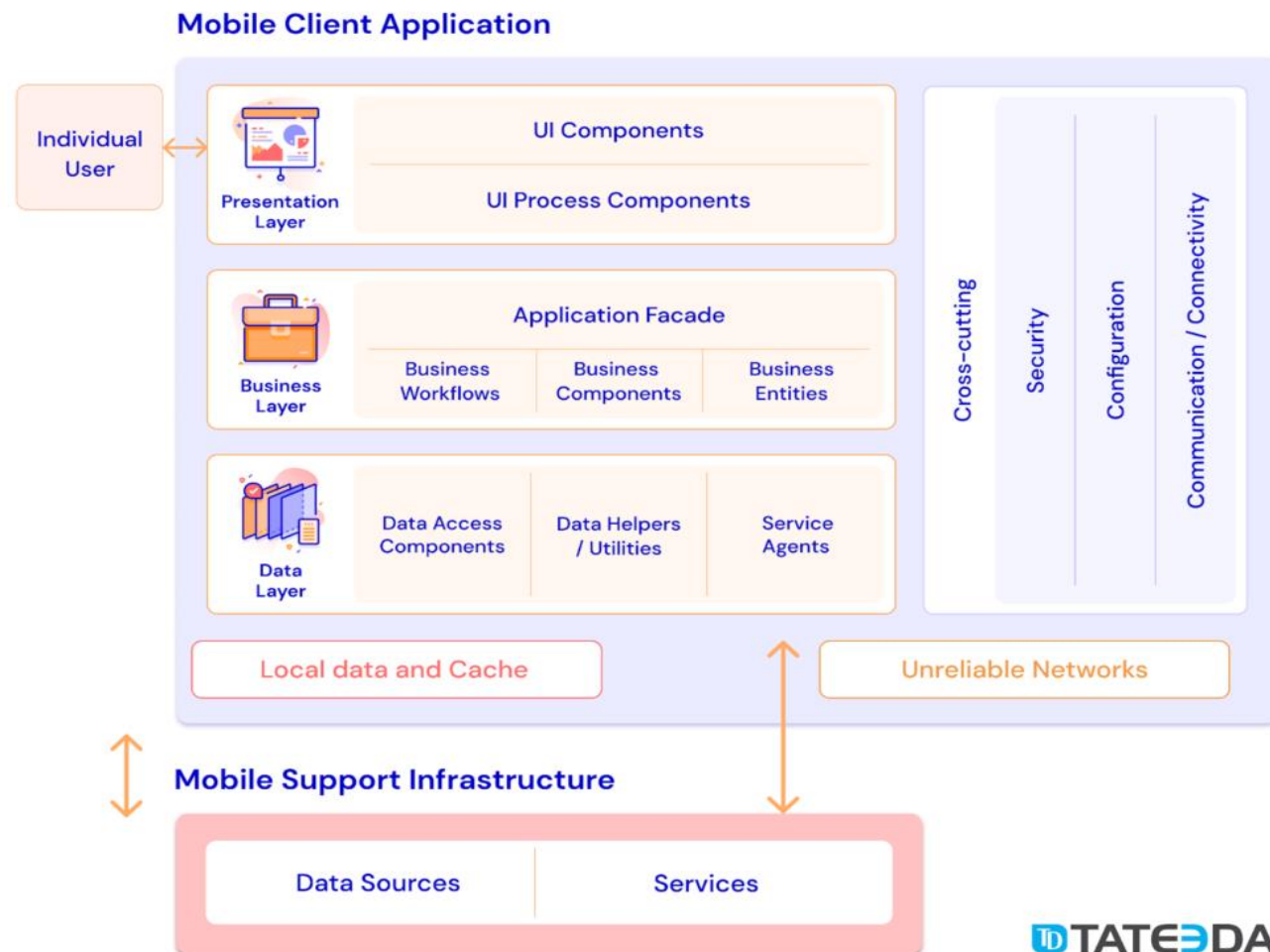
Warstwa Prezentacji (Presentation Layer)

Warstwa Biznesowa (Business Layer)

Warstwa Dostępu do Danych (Data Layer)

Lokalna Pamięć i Cache (Local Data and Cache)
Sieci (Unreliable Networks)

Źródła Danych (Data Sources)
Usługi (Services)



PWA

Aplikacje Progresywne (PWA, Progressive Web App)

to strony webowe (HTML, CSS, JavaScript), które wykorzystują nowoczesne technologie webowe, aby dostarczyć użytkownikom doświadczenia zbliżone do natywnych aplikacji mobilnych.

PWAs łączą najlepsze cechy aplikacji webowych i mobilnych, oferując szybkie, niezawodne i angażujące doświadczenia.

Cechy

Responsywność:

- Dostosowują się do różnych rozmiarów ekranów i urządzeń.

Niezależność od sieci:

- Mogą działać offline lub przy słabym połączeniu sieciowym dzięki technologii Service Workers.

Instalowalność:

- Mogą być instalowane na urządzeniach użytkowników bez konieczności przechodzenia przez sklep aplikacji.

Szybkość:

- Szybkie ładowanie i responsywność, nawet w trudnych warunkach sieciowych.

Bezpieczeństwo:

- Muszą być serwowane przez HTTPS, co zapewnia bezpieczne połączenie i chroni integralność danych.

PWA

Czy aplikacje webowe to PWAs?

Nie każda aplikacja webowa jest PWA. Aby aplikacja webowa była uznana za PWA, musi spełniać powyższe cechy i dodatkowe wymagania:

Manifest JSON:

- Plik manifestu (np. manifest.json), który zawiera metadane aplikacji, takie jak jej nazwa, ikony, start URL, kolory, itp.

Service Workers:

- Skrypty działające w tle przeglądarki, które umożliwiają funkcje offline, push notifications, i inne zaawansowane funkcje.

```
{
  "name": "Przykładowa PWA",
  "short_name": "PWA",
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "description": "Przykładowa Progressive Web App",
  "icons": [
    {
      "src": "/images/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/images/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

manifest.json

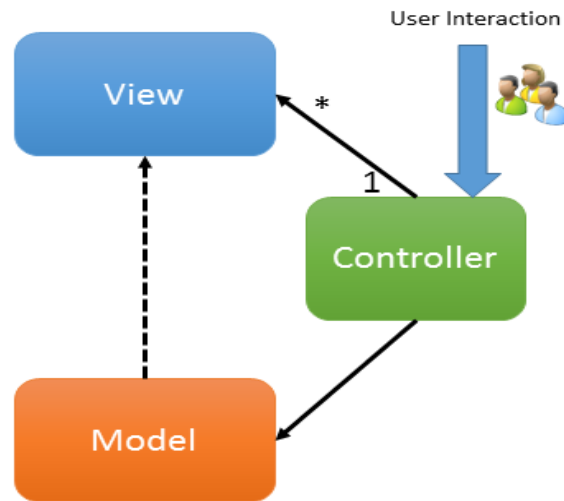
```
self.addEventListener('install', function(event) {
  event.waitUntil(
    caches.open('static-cache').then(function(cache) {
      return cache.addAll([
        '/',
        '/index.html',
        '/styles.css',
        '/app.js',
        '/images/icon-192x192.png',
        '/images/icon-512x512.png'
      ]);
    })
  );
});

self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.match(event.request).then(function(response) {
      return response || fetch(event.request);
    })
  );
});
```

service-worker.js

Wzorce projektowe

MVC (Model-View-Controller)

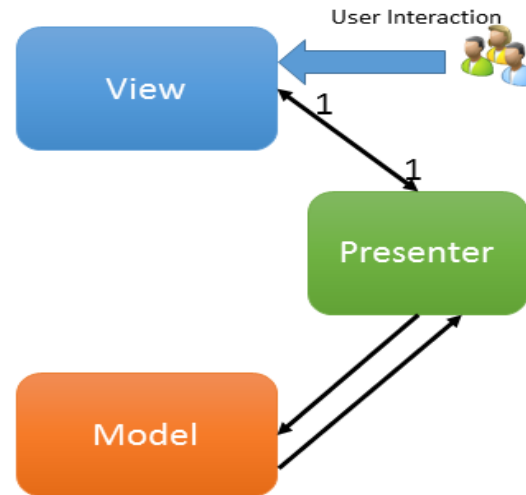


Model: Reprezentuje dane i logikę biznesową aplikacji. Zarządza stanem aplikacji i komunikuje się z bazą danych.

View: Odpowiada za prezentację danych użytkownikowi. Renderuje dane dostarczane przez kontroler.

Controller: Przechwytyuje dane wejściowe od użytkownika, przetwarza je i aktualizuje Model oraz View.

MVP (Model-View-Presenter)

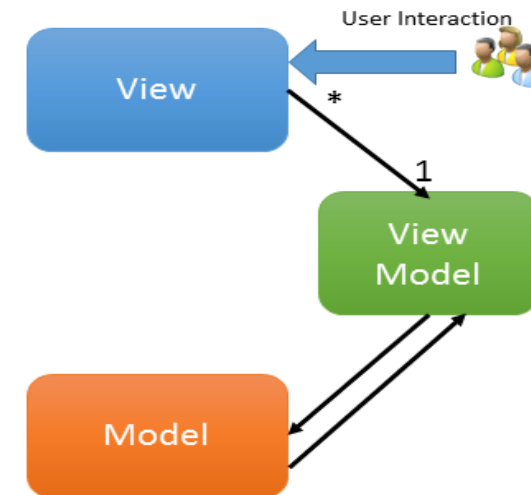


Model: Reprezentuje dane i logikę biznesową aplikacji.

View: Odpowiada za prezentację danych. Jest pasywna i nie zawiera logiki.

Presenter: Obsługuje logikę prezentacji i działa jako mediator między Modelem a View.

MVVM (Model-View-ViewModel)



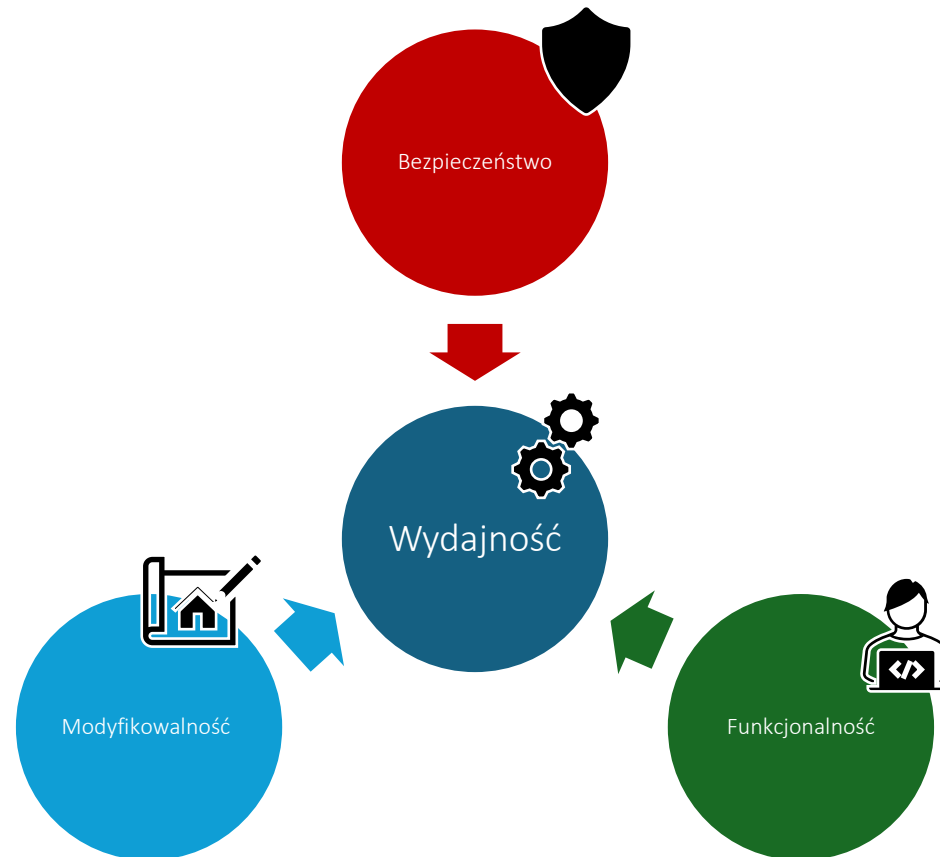
Model: Reprezentuje dane i logikę biznesową aplikacji.

View: Odpowiada za prezentację danych użytkownikowi.

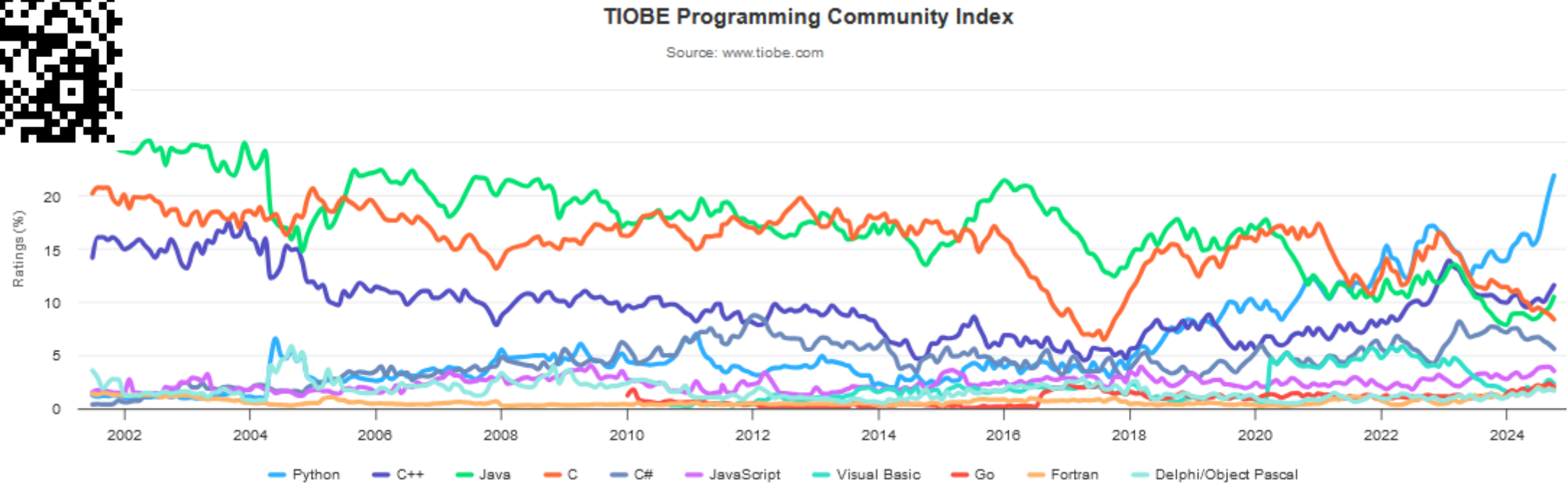
ViewModel: Zarządza logiką prezentacji i stanem widoku. Jest pośrednikiem między Modelem a View.

Wzorce projektowe

Wydajność aplikacji



Ranking - Języki programowania



TIOBE Index

<https://www.tiobe.com/tiobe-index/>

Ranking - Języki programowania



Oct 2024	Oct 2023	Change	Programming Language		Ratings	Change
1	1			Python	21.90%	+7.08%
2	3	^		C++	11.60%	+0.93%
3	4	^		Java	10.51%	+1.59%
4	2	v		C	8.38%	-3.70%
5	5			C#	5.62%	-2.09%
6	6			JavaScript	3.54%	+0.64%
7	7			Visual Basic	2.35%	+0.22%
8	11	^		Go	2.02%	+0.65%
9	16	^^		Fortran	1.80%	+0.78%
10	13	^		Delphi/Object Pascal	1.68%	+0.38%
11	9	v		SQL	1.64%	-0.15%
12	14	^		MATLAB	1.48%	+0.22%
13	20	^^		Rust	1.45%	+0.53%
14	12	v		Scratch	1.41%	+0.05%
15	8	vv		PHP	1.21%	-0.69%
16	10	vv		Assembly language	1.13%	-0.51%
17	17			R	1.09%	+0.12%
18	19	^		Ruby	0.99%	+0.07%
19	24	^^		COBOL	0.99%	+0.23%
20	15	vv		Swift	0.98%	-0.09%

TIOBE Index

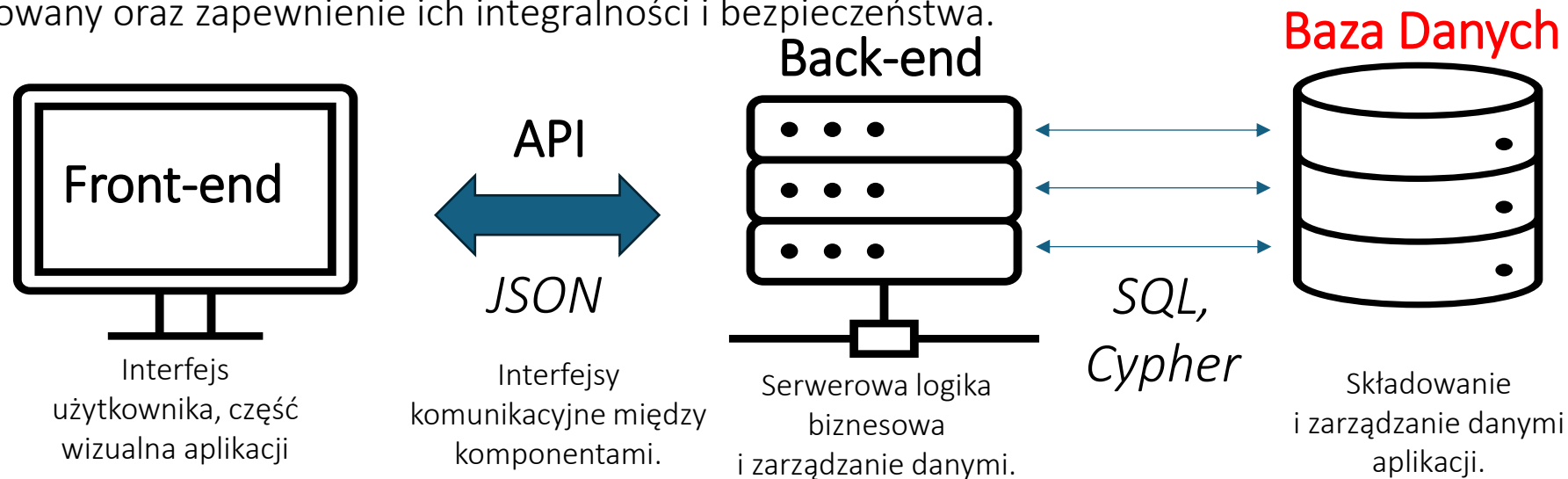
<https://www.tiobe.com/tiobe-index/>

Bazy danych

Baza Danych

to system składowania danych, który umożliwia przechowywanie, organizowanie oraz zarządzanie dużymi ilościami informacji.

Jest kluczowym elementem aplikacji, odpowiedzialnym za gromadzenie i przechowywanie danych w sposób uporządkowany oraz zapewnienie ich integralności i bezpieczeństwa.



Bazy danych

Systemy Baz Danych

Typ Bazy Danych	Nazwa	Opis
Relacyjne	MySQL	Open-source system zarządzania bazą danych, popularny w aplikacjach webowych.
	Oracle Database	Jeden z najbardziej zaawansowanych systemów baz danych, używany w dużych aplikacjach korporacyjnych.
	Microsoft SQL Server	System zarządzania bazą danych opracowany przez Microsoft, szeroko stosowany w korporacjach.
	PostgreSQL	Zaawansowany system zarządzania bazą danych, znany z obsługi skomplikowanych zapytań SQL.
	SQLite	Lekka, wbudowana baza danych, idealna do aplikacji mobilnych i desktopowych.
Nierelacyjne (NoSQL)	Neo4j	Baza danych grafowa, która przechowuje dane w postaci węzłów, relacji i właściwości.
	MongoDB	Baza danych dokumentowa, która przechowuje dane w formacie JSON.
	Cassandra	Baza danych kolumnowa, zaprojektowana do obsługi dużych ilości danych rozproszonych.
	Redis	Baza danych klucz-wartość, znana z wysokiej wydajności i wsparcia dla różnych struktur danych.
	CouchDB	Baza danych dokumentowa, która przechowuje dane w formacie JSON z obsługą wersjonowania.

Modele baz danych

SQL (80%)

Bazy relacyjne *

Przechowują dane w tabelach powiązanych kluczami głównymi i obcymi, zapewniając integralność i spójność danych.

Przykłady: Oracle, MSSQL, MySQL

Zastosowanie: Systemy transakcyjne, raportowanie, analizy danych

NoSQL (20%)

Bazy dokumentowe

Przechowują dane w formie dokumentów, zazwyczaj w formacie JSON, BSON lub XML.

Przykłady: MongoDB, CouchDB.

Zastosowanie: Idealne do przechowywania danych o zmiennej strukturze, takich jak dokumenty, profile użytkowników.

Bazy klucz-wartość

Przechowują dane jako pary klucz-wartość, gdzie klucz jest unikalnym identyfikatorem, a wartość może być dowolnym typem danych.

Przykłady: Redis, DynamoDB.

Zastosowanie: Świetne do szybkiego przechowywania i pobierania dużych ilości danych, takich jak sesje użytkowników, dane cache.

Bazy wektorowe

Organizują dane w formie wektorów, co umożliwia szybkie dopasowywanie podobieństw między danymi oraz efektywne wyszukiwanie.

Przykłady: Pinecone, Milvus, Qdrant

Idealne do wyszukiwania semantycznego, systemów rekomendacji oraz analiz AI, gdzie kluczowe jest porównywanie wektorów i odnajdywanie podobieństw.

Bazy grafowe *

Przechowują dane w postaci węzłów, krawędzi i właściwości, co umożliwia modelowanie złożonych relacji między danymi.

Przykłady: Neo4j, ArangoDB.

Zastosowanie: Używane w aplikacjach wymagających analizy grafów, takich jak sieci społecznościowe, zarządzanie zasobami, rekomendacje.

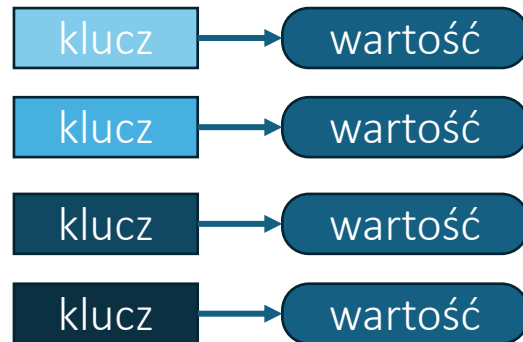
Wprowadzenie do baz NoSQL

Bazy klucz-wartość

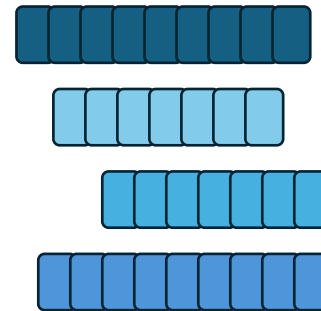
(Key-Value): gdzie wszystko przechowywane jest jako para klucz-wartość.

Przykłady: Redis, Amazon DynamoDB

Bazy klucz-wartość



Bazy wektorowe



Bazy wektorowe

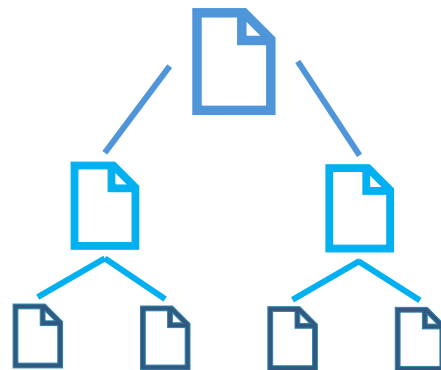
(Vectors): dane są przechowywane i wyszukiwane jako wektory, co umożliwia szybkie dopasowywanie podobieństw między obiektami.
Przykłady: Pinecone, Milvus, Qdrant.

Bazy dokumentowe

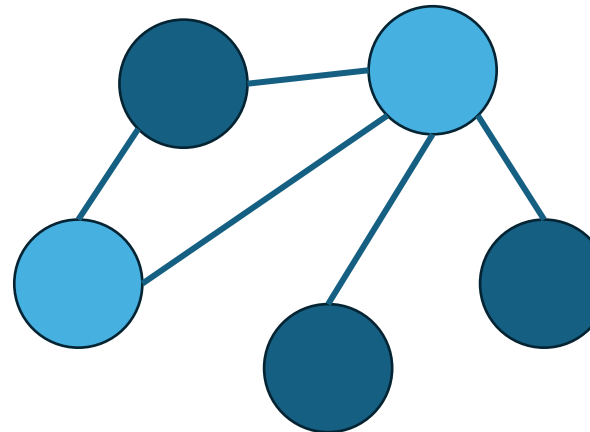
(Document): przechowują informacje jako dokumenty, często w formatach takich jak JSON.

Przykłady: MongoDB, Couchbase.

Bazy dokumentów



Bazy grafowe



Bazy grafowe

(Graph): skoncentrowane na przechowywaniu informacji o węzłach i relacjach między nimi.
Przykład: Neo4j, JanusGraph

Bazy klucz-wartość

Bazy klucz-wartość

to jeden z najprostszych i najbardziej efektywnych modeli przechowywania danych, który polega na skojarzeniu unikalnego klucza z odpowiadającą mu wartością.

Chociaż ich struktura jest wyjątkowo prosta, bazy te stanowią fundament dla wielu zaawansowanych systemów, które obsługują ogromne ilości danych w czasie rzeczywistym.

Przestrzeń: Klient		Przestrzeń: Piłkarz		Przestrzeń: Polityk	
osoba:1	Nowak	osoba:1	Smolarek	osoba:1	Sikorski
osoba:2	Kowalski	osoba:2	Boniek	osoba:2	Kuroń
osoba:3	Malinowski	osoba:3	Lato	osoba:3	Mazowiecki



Bazy wektorowe

Wektorowa baza danych

- to systemy przechowywania i organizacji danych, które wykorzystują matematyczne wektory do reprezentacji informacji.
- Wektor to zbiór liczb opisujących pewne cechy obiektu, np. punktów w przestrzeni, słów w tekście lub właściwości obrazów. Dzięki temu można analizować i porównywać dane, szukając między nimi podobieństw. W praktyce bazy wektorowe stanowią kluczowe narzędzie w teleinformatyce oraz w obszarach związanych ze sztuczną inteligencją.

Zalety baz wektorowych

Szybkie wyszukiwanie podobieństw:

Idealne do analizy danych wielowymiarowych.

Skalowalność:

Możliwość pracy z ogromnymi zbiorami danych, np. miliony obrazów czy dokumentów.

Uniwersalność:

Obsługa różnych typów danych, takich jak tekst, obrazy, dźwięki czy dane przestrzenne.

Optymalizacja dla AI:

Umożliwiają integrację z modelami uczenia maszynowego i głębokiego.

Wady baz wektorowych

Wysokie zużycie pamięci

Wielowymiarowe wektory wymagają dużej przestrzeni dyskowej i pamięci operacyjnej.

Złożoność obliczeniowa

Wyszukiwanie i porównywanie wektorów w przestrzeni o dużej liczbie wymiarów jest czasochłonne.

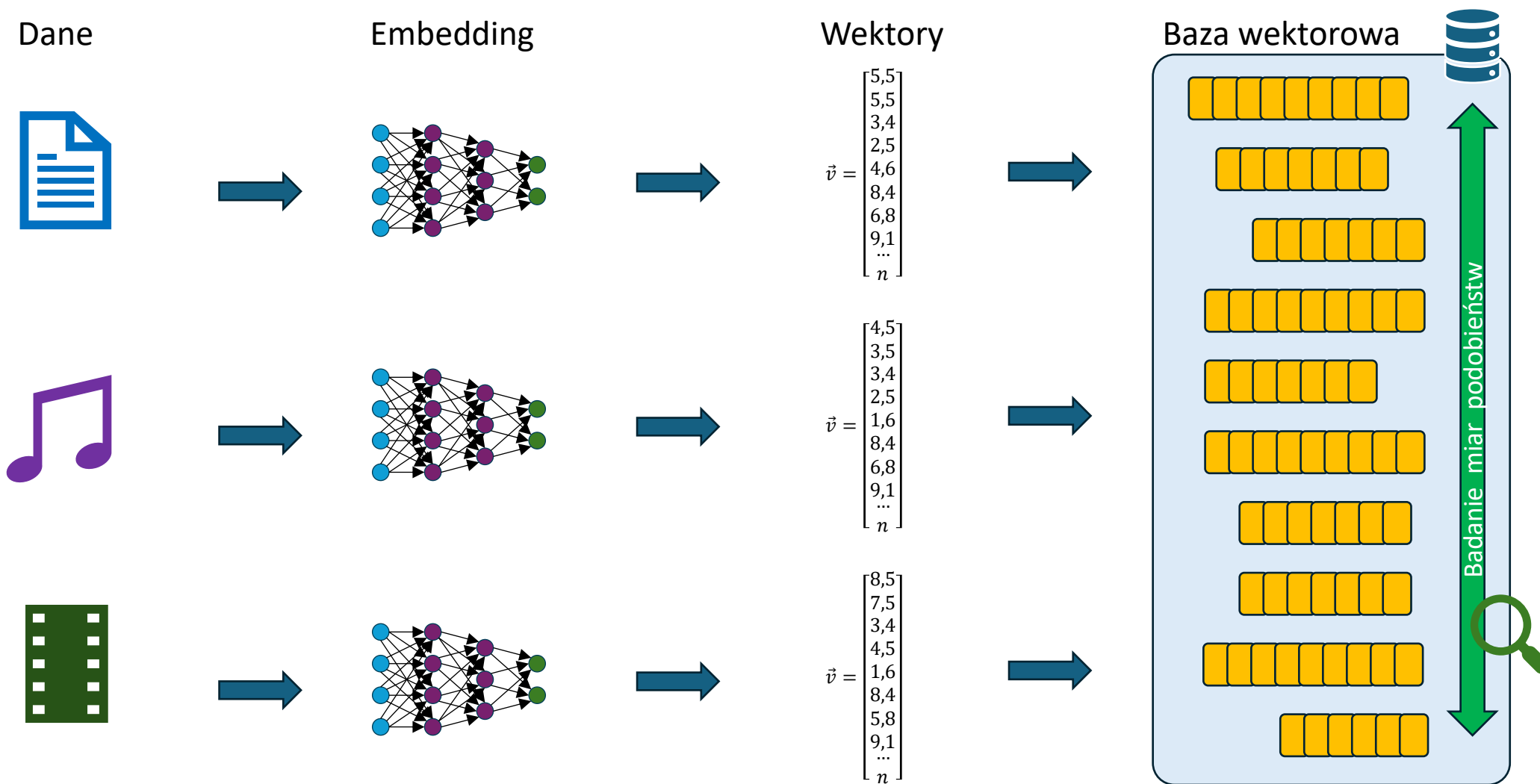
Przybliżone wyniki wyszukiwania

Algorytmy **Approximate Nearest Neighbor (ANN)** często zwracają wyniki przybliżone, a nie dokładne.

Trudność interpretacji

Wysokowymiarowe przestrzenie trudno jest wizualizować i zrozumieć intuicyjnie.

Bazy wektorowe



Bazy grafowe

Grafowa baza danych

to system zarządzania bazą danych zaprojektowany do przechowywania, zarządzania i wyszukiwania informacji w strukturach grafowych.



Elastyczność

Łatwość modelowania
złożonych relacji
i dynamicznych
schematów.



Wydajność

Szybka nawigacja
i zapytania w
złożonych sieciach
relacji.



Intuicyjność

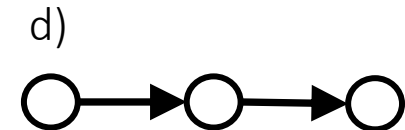
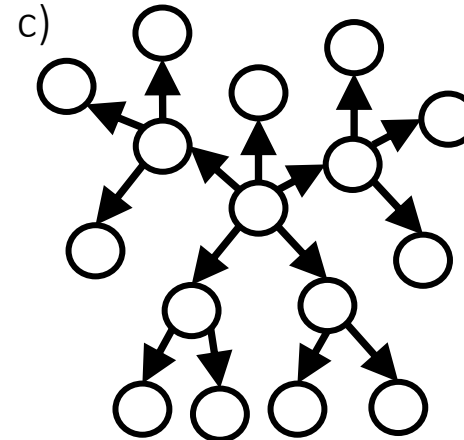
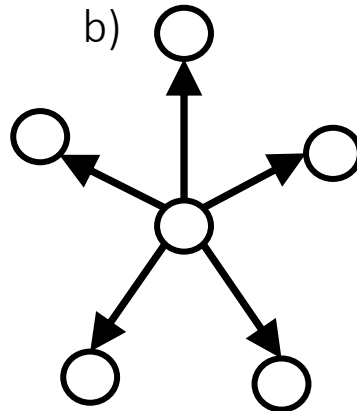
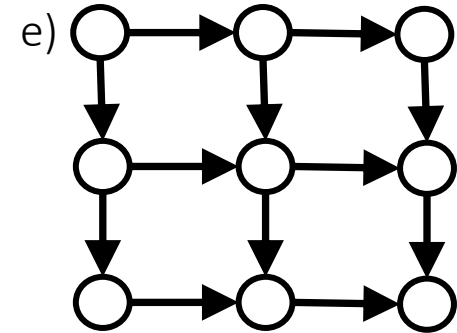
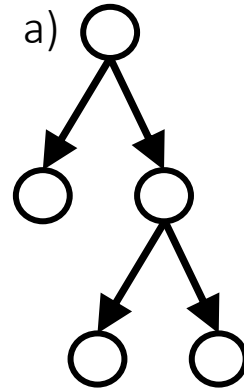
Naturalne
odzworowanie
złożonych struktur
i relacji świata
rzeczywistego.

Bazy grafowe należą do klasy baz **NoSQL**

Wybrane formy grafów

Przykładowe formy grafów

- a) Drzewo np.: struktura katalogów w systemie plików
- b) Gwiazda np.: komputerowa sieć lokalna
- c) Płatek śniegu np.: organizacja Państwa (województwa, powiaty, gminy, miasta...)
- d) Łańcuch np.: rzeka
- e) Siatka np.: sieć uliczna



Bazy grafowe vs. Bazy relacyjne



Relacyjna baza danych

Osoba

id	nazwisko
1	Nowak
2	Kowalski

Firma

id	Nazwa
1	Alfa
2	Beta

Miasto

id	nazwa
1	Warszawa
2	Kraków

Mieszka_w

idosoba	idmiasto	od_roku
1	2	1990
2	1	2011

Pracuje_w

idosoba	idfirma	jako
1	2	Mechanik
2	1	Księgowy

Ma_lokalizację_w

idfirma	idmiasto	dzielnica
1	1	Mokotów
1	2	Kazimierz
2	1	Ochota

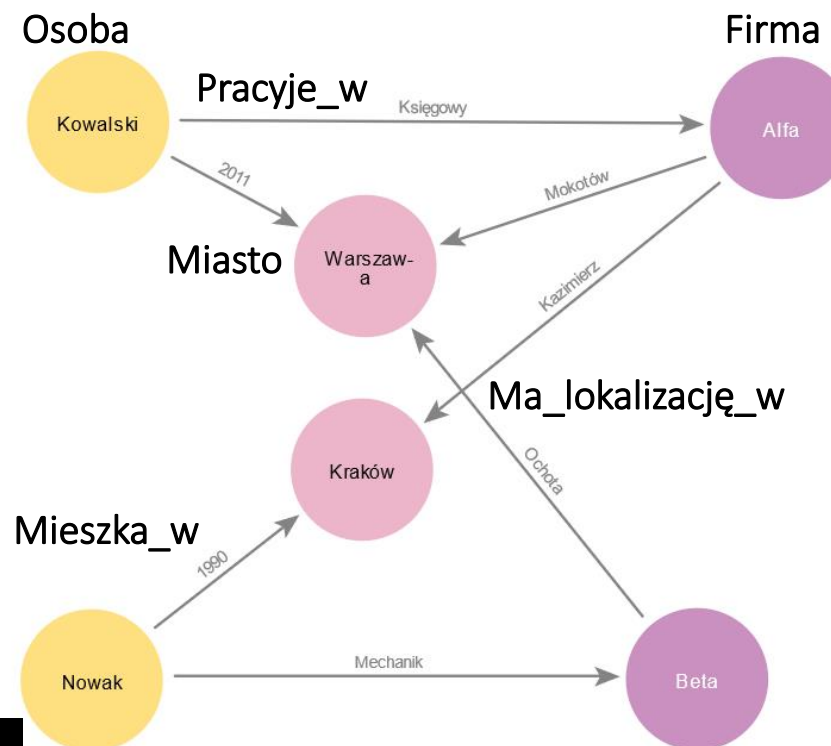
SQL

```
SELECT Firma.nazwa, Pracuje_w.jako FROM Firma
JOIN Pracuje_w ON Pracuje_w.idfirma = Firma.id
JOIN Osoba ON Osoba.id = Pracuje_w.idosoba
WHERE Osoba.nazwisko = "Kowalski"
```

Jako kto i gdzie pracuje Kowalski?



Grafowa baza danych

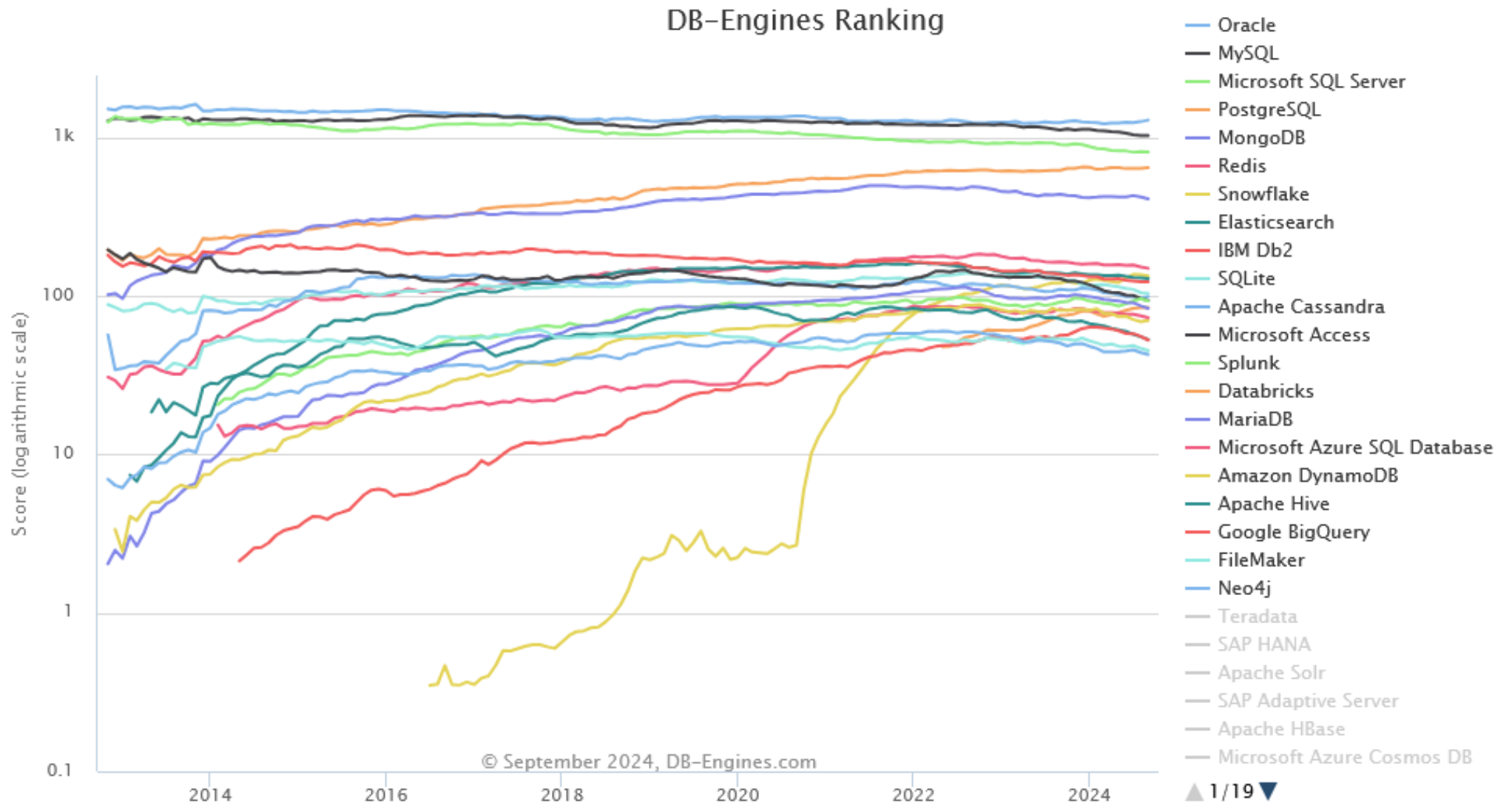


Cypher

```
MATCH (o:Osoba {nazwisko:'Kowalski'})-[p:Pracuje_w]->(f:Firma)
RETURN f.nazwa, p.jako
```

nazwa	jako
Alfa	Księgowy

Ranking - Systemy baz danych



DB-Engines Ranking

<https://db-engines.com/en/ranking>

Ranking - Systemy baz danych



423 systems in ranking, September 2024

Rank			DBMS	Database Model	Score		
Sep 2024	Aug 2024	Sep 2023			Sep 2024	Aug 2024	Sep 2023
1.	1.	1.	Oracle +	Relational, Multi-model i	1286.59	+28.11	+45.72
2.	2.	2.	MySQL +	Relational, Multi-model i	1029.49	+2.63	-82.00
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	807.76	-7.41	-94.45
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	644.36	+6.97	+23.61
5.	5.	5.	MongoDB +	Document, Multi-model i	410.24	-10.74	-29.18
6.	6.	6.	Redis +	Key-value, Multi-model i	149.43	-3.28	-14.26
7.	7.	↑ 11.	Snowflake +	Relational	133.72	-2.25	+12.83
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model i	128.79	-1.04	-10.20
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model i	123.05	+0.04	-13.67
10.	10.	↓ 9.	SQLite +	Relational	103.35	-1.44	-25.85
11.	11.	↑ 12.	Apache Cassandra +	Wide column, Multi-model i	98.94	+1.94	-11.11
12.	12.	↓ 10.	Microsoft Access	Relational	93.76	-2.61	-34.81
13.	13.	↑ 14.	Splunk	Search engine	93.02	-3.08	+1.63
14.	↑ 15.	↑ 17.	Databricks +	Multi-model i	84.24	-0.22	+9.06
15.	↓ 14.	↓ 13.	MariaDB +	Relational, Multi-model i	83.44	-3.09	-17.01
16.	16.	↓ 15.	Microsoft Azure SQL Database	Relational, Multi-model i	72.95	-2.08	-9.78
17.	17.	↓ 16.	Amazon DynamoDB +	Multi-model i	70.06	+1.15	-10.85
18.	↑ 19.	18.	Apache Hive	Relational	53.07	-2.17	-18.76
19.	↓ 18.	↑ 20.	Google BigQuery +	Relational	52.67	-2.86	-3.80
20.	20.	↑ 21.	FileMaker	Relational	45.20	-1.47	-8.40
21.	21.	↑ 23.	Neo4j +	Graph	42.68	-1.22	-7.71
22.	↑ 23.	↓ 19.	Teradata	Relational, Multi-model i	41.47	-0.78	-18.86
23.	↓ 22.	↓ 22.	SAP HANA +	Relational, Multi-model i	41.32	-1.04	-9.29
24.	24.	24.	Apache Solr	Search engine, Multi-model i	34.15	-2.14	-12.94
25.	25.	25.	SAP Adaptive Server	Relational, Multi-model i	32.67	-0.61	-10.64

DB-Engines Ranking

<https://db-engines.com/en/ranking>

Bazy danych

Metody dostępu

PHP + MySQL

Logowanie do bazy danych

```
<?php
$conn = new mysqli("localhost", "username", "password", "database");
if ($conn->connect_error) die("Connection failed: " . $conn->connect_error);
$sql = "SELECT id, username, email FROM users";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["username"]. $row["email"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

SQL



Python + Neo4j

```
from neo4j import GraphDatabase

class Neo4jExample:
    def __init__(self, uri, user, password):
        self.driver = GraphDatabase.driver(uri, auth=(user, password))
    def close(self):
        self.driver.close()
    def read_data(self):
        with self.driver.session() as session:
            result = session.run("MATCH (u:User) RETURN u.id AS id, u.username AS username")
            for record in result:
                print(f"id: {record['id']} - Username: {record['username']}")

neo4j_example = Neo4jExample("bolt://localhost:7687", "username", "password")
neo4j_example.read_data()
neo4j_example.close()
```

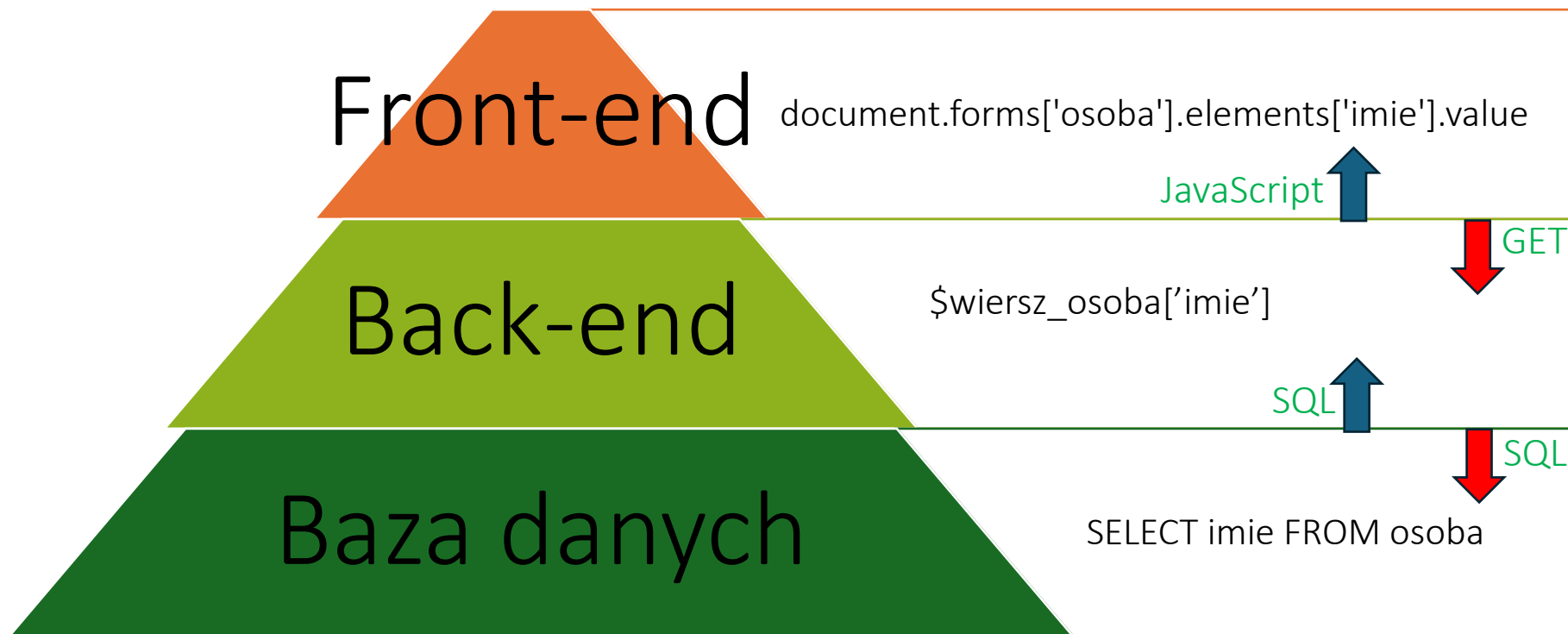
Cypher



Logowanie do bazy danych

Bazy danych

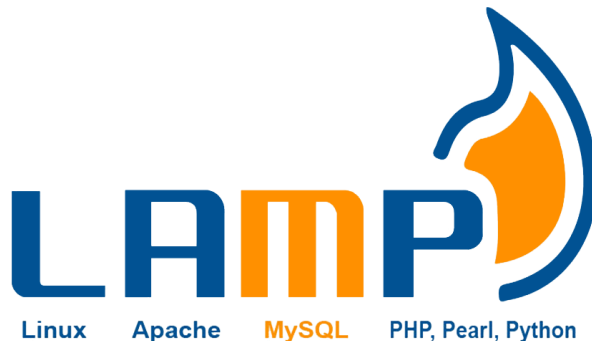
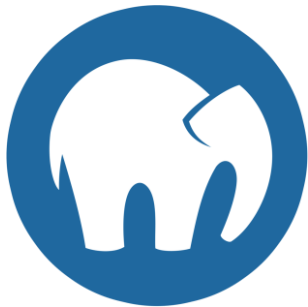
Trzy warstwy zarządzania wartościami danych



Zintegrowane środowiska deweloperskie

Zintegrowane środowiska deweloperskie

Środowisko	Komponenty	Systemy Operacyjne	Linki
XAMPP	Apache, MySQL, PHP i Perl.	Linux, Windows, macOS	https://www.apachefriends.org/pl/index.html
MAMP	Apache, MySQL i PHP.	macOS, Windows	https://www.mamp.info/en/windows/
LAMP	Linux, Apache, MySQL, PHP	Linux	https://help.ubuntu.com/community/ApacheMySQLPHP



Zintegrowane środowiska deweloperskie

XAMPP i LAMP

How to use XAMPP?



LAMP architecture

