



Politechnika Wroclawska

Internetowe Bazy Danych

dr inż. Roman Ptak

Katedra Informatyki Technicznej

roman.ptak@pwr.edu.pl



Plan wykładu 1.

- Sprawy organizacyjne
- Wprowadzenie w zagadnienie
- Projektowanie systemów informatycznych
 - Analiza ekonomiczna SWOT
- Przykłady serwisów internetowych
- Wybór technologii oraz architektury
 - Języki programowania
 - Serwery baz danych
 - Technologie udostępniania informacji
 - Technologie dostępu do danych



Sprawy organizacyjne

- Forma zajęć:
 - wykład (15 h)
 - projekt (15 h)
- Wykłady: poniedziałek (TP) 9:30-11:45
 - 12.10, 26.10, 9.11, 23.11, 7.12
- Materiały odstępne pod adresem:
roman.ptak.staff.iiar.pwr.wroc.pl



Konsultacje

dr inż. Roman Ptak

e-mail: roman.ptak@pwr.edu.pl

bud. C3, pok. 321

- ?



Warunki zaliczenia przedmiotu

- Zaliczenie wykładu na ocenę na podstawie kolokwium zaliczeniowego (i obecności na wykładzie)
- Zwolnienie z kolokwium na podstawie oceny z projektów
- Zaliczenie projektów na ocenę



Program wykładu

- Wprowadzenie - architektura internetowych systemów z bazami danych
 - Technologie bazodanowe
 - Technologie webowe
- Bezpieczeństwo internetowych baz danych
- Zasady używania języka PHP, Python (i innych technologii) w projektach internetowych baz danych
- Budowa systemów z użyciem technologii MySQL, ...
- Przygotowywanie dokumentacji projektowej

Literatura podstawowa

- Tim Converse, Joyce Park, *Clark Morgan*, *PHP5 i MySQL. Biblia*, Helion, 2005.
- Luke Welling, Laura Thomson, *PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty*, Helion, 2009.
- Julie C. Meloni, *PHP, MySQL i Apache dla każdego. Wydanie III*, Helion 2007.
- Thomas Connolly, Carolyn Begg, *Systemy baz danych. Praktyczne metody projektowania, implementacji i zarządzania. Tom 2*, RM, 2004.



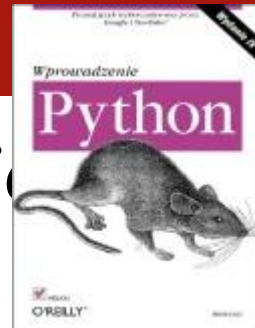


Literatura uzupełniająca

- Marcin Lis, *PHP. 101 praktycznych skryptów. Wydanie II*, Helion, 2007.
- Łukasz Sosna, *101 porad. PHP i MySQL*, Mikom, 2005.
- Stephan Schmidt et al., *PEAR. Programowanie w PHP*, Helion, 2007.
- Jacek Matulewski, Sławomir Orłowski, *Technologie ASP.NET i ADO.NET w Visual Web Developer*, Helion, 2007.

Literatura uzupełniająca - c.d.

- Mark Lutz, *Python. Wprowadzenie*. Wydanie Helion, 2010.
- Jeff Forcier, Paul Bissex, Wesley Chun, *Python i Django. Programowanie aplikacji webowych*, Helion, 2009.
- Aidas Bendoraitis, *Aplikacje internetowe z Django. Najlepsze receptury*, Helion, 2015.
- Larry Ullman, *Ruby. Szybki start*, Helion, 2009.
- Edward Benson, *Rails. Sztuka programowania*, Helion, 2009.





WPROWADZENIE



Czym są internetowe bazy danych?

- Zbiór danych dostępny w Internecie z możliwością edycji i zarządzania nim.
- Uporządkowany zbiór danych z pewnej dziedziny tematycznej, zorganizowany w sposób umożliwiający ich wyszukiwanie według zadanych kryteriów.
- Często stosowane są relacyjne bazy danych - podzbiory danych są ze sobą powiązane, współpracujące ze sobą.



Cele projektów informatycznych wykorzystujących bazy danych:

- zastąpienie dotychczasowego systemu w celu poprawienia wydajności (gdy system „ręczny” lub komputerowy nie nadąża z przetwarzaniem dużej ilości transakcji lub nie umożliwia ich realizacji na dużą skalę),
- obniżkę kosztów związanych z realizacją transakcji lub dystrybucją informacji (np. rezygnacja z „papierowego” przekazywania informacji w dużych firmach), a w przypadku sklepów internetowych - brak kosztów związanych z obsługą osób nie dokonujących żadnych transakcji,
- uzyskanie lepszego obiegu informacji w przypadku systemów związanych z jej dystrybucją lub zwiększenie liczby klientów.



Cele projektów informatycznych wykorzystujących bazy danych:

- poprawę *image* przedsiębiorstwa,
- zwiększenie „dostępności” firmy dla klienta (nie jest problemem 24 h dostęp),
- uzyskanie możliwości zbierania i wykorzystywania informacji o gęście i upodobaniach klientów
(w tym zakresie stosowane są nawet techniki znane wcześniej ze zwykłych sklepów, np. ustawianie przy kasach odpowiednich produktów = np. wyświetlanie w odpowiedni sposób informacji w oknie koszyka).



Typowe zastosowania internetowych baz danych:

- serwisy WWW - m. in. uniezależnienie prezentowanych treści od „wyglądu” witryny,
 - część informacyjna i transakcyjna serwisu
- e - commerce,
- inne usługi np. serwer poczty WWW, forum internetowe, baza dokumentów.



ANALIZA SYSTEMÓW INFORMATYCZNYCH



Zakres funkcjonalny systemów informatycznych:

- misja i produkty firmy, rynki i klienci,
- analiza otoczenia i czynników zewnętrznych związanych z technologią informatyczną,
- analiza potencjału firmy z uwzględnieniem wsparcia informatycznego,
- analiza technologii informatycznej.

Ryzyka i zagrożenia związane z projektem:

- złe oszacowanie kosztów przedsięwzięcia,
- niedopracowany lub nieprzemyślany projekt interfejsu,
- trudności z implementacją nieprzewidzianych wcześniej elementów - skalowalność systemu,
- niezrozumienie lub zła specyfikacja potrzeb zleceniodawcy i odbiorcy projektu przez „informatyków”.



Ryzyka i zagrożenia (inne):

- trudności z „dotarciem” odbiorcy do naszej witryny,
- działania hakerów (crackerów),
- awarie sprzętu oraz błędy w oprogramowaniu,
- często zmieniające się przepisy prawne,
- zawodność dostawców,
- aktualizacja danych (np. modyfikacja cen).



Analiza SWOT

- Jedna z technik analitycznych, służących do porządkowania wszelkich informacji na temat danej firmy.
- Pomaga w analizie zewnętrznej jak i wewnętrznej danego środowiska.



Analiza SWOT





Przykłady

- Ogrodnictwo
- Zakład fryzjerski
 - [przykład2](#)
- Kręgielnia
 - [przykład3](#)
- źródła:
 - <http://www.megapedia.pl/analiza-swot.htm>
 - <http://analiza-swot.pl/zobacz-przyklady>
 - <http://analiza-swot.pl/zobacz-przyklady>

MOCNE STRONY (Strengths)

Klimat — produkcja w okresie od marca do października umożliwia eksport pomidorów na rynki zachodnioeuropejskie

Położenie — możliwość uprawy w gruncie przy paliku

Lokalizacja — bliskość największego w Europie rynku zbytu, jakim są Niemcy

Robocizna — nadal tania siła robocza

SZANSE (Opportunities)

Sprzedaż — możliwość eksportu w okresie od maja do listopada

Jakość — duże walory smakowe owoców z Polski

Marka — kreowana przez duże zakłady produkcyjne

Specjalizacja — na przykład w produkcji pomidorów wielkoowocowych

SŁABE STRONY (Weaknesses)

Technologia — brak dobrych specjalistów do prowadzenia produkcji

Sprzedaż i marketing — małe szanse rozwoju przy sprzedaży indywidualnej

Do inwestowanie — brak środków na wdrożenie nowych technologii

Wydajność pracowników — niedostateczna wiedza i umiejętności pracowników sezonowych

Nawożenie — używanie gotowych mieszanek o wiele droższych niż nawozy pojedyncze

Powolne zmiany — konserwatywny sposób myślenia wielu producentów, brak grup producenckich

ZAGROŻENIA (Threats)

Import z krajów o cieplejszym klimacie, obniżający opłacalność produkcji szklarniowej w cyklu całorocznym

Rosnące **ceny energii**

Kontrola jakości — jest prowadzona przez wielu dużych odbiorców

Migracja pracowników — coraz więcej osób wyjeżdża do pracy na Zachód



Przykładowe czynniki - system informatyczny

	Pozytywne	Negatywne
Wewnętrzne	<ul style="list-style-type: none">• Doświadczenie i wiedza specjalistyczna pracowników• Możliwości realizacji usług w szeroki zakresie• Renoma firmy na rynku	<ul style="list-style-type: none">• Brak systemów klasy CRM• Brak aplikacji bazodanowych• Zbytne obciążenie pracą• Brak strony WWW
Zewnętrzne	<ul style="list-style-type: none">• Nawiązanie współpracy z innymi firmami• Nowe technologie na rynku• Możliwość outsorsingu	<ul style="list-style-type: none">• Zmiana przepisów prawnych• Kryzys gospodarczy• Ataki crackerów na system



ANALIZA POWIĄZAŃ SWOT

- Czy określona mocna strona pozwala wykorzystać daną szansę? ($S \rightarrow O$)
- Czy określona mocna strona pozwala ograniczyć dane zagrożenie? ($S \rightarrow T$)
- Czy określona słaba strona ogranicza możliwość wykorzystania danej szansy? ($W \rightarrow O$)
- Czy określona słaba strona potęguje dane zagrożenie? ($W \rightarrow T$)



ANALIZA POWIĄZAŃ TOWS

- Czy określona szansa potęguje daną silną stronę? ($O \rightarrow S$)
- Czy określone zagrożenie ogranicza daną silną stronę? ($T \rightarrow S$)
- Czy określona szansa pozwala osłabić daną słabą stronę? ($O \rightarrow W$)
- Czy określone zagrożenie wzmacnia daną słabą stronę? ($T \rightarrow W$)



Strategie organizacji

- Agresywna (S-O)
- Konserwatywna (S-T)
- Konkurencyjna (W-O)
- Defensywna (W-T)



PRZYKŁADY SERWISÓW



Allegro.pl

Strefa marek

Inspiracje

moda.allegro

wystaw przedmiot

moje allegro

wyloguj

allegro

wszystkie działy



koszyk jest pusty

Elektronika

Moda
i uroda

Dom
i zdrowie

Dziecko

Kultura
i rozrywka

Sport
i wypoczynek

Motoryzacja

Kolekcje
i sztuka

Firma
i usługi

Strefa
okazji

LAPTOP HP 250

- PROCESOR INTEL CELERON 1000M
- PAMIĘĆ 2 GB RAM
- DYSK TWARDY 320 GB
- WINDOWS 8



~~1299 zł~~
999 zł

wysyłka gratis



Laptop HP

Kupony Allegro

LG L90

Tanie wystawianie

Sukienki

wyjątkowe **okazje** dla Ciebie



Allegro.pl

Adres:	www.allegro.pl
System operacyjny:	LinuxBSD
Serwer WWW:	Apache
Baza danych:	MySQL
Autoryzacja:	Login i hasło - przesłane niekodowanym tekstem, możliwości SSL
Technologia:	PHP
Śledzenie sesji:	ID sesji przekazywany w cookie
Liczba ofert / użytkowników:	48,1 mln / 11,5* mln (* 2011 r.)

Realizacje IBD

- *Database Driven Websites*
- *sklepy internetowe, blogi, CMS i inne*

ebay Google

amazon

● onet

interia



WP



YAHOO!



Joomla!



WYBÓR TECHNOLOGII ORAZ ARCHITEKTURY



Rozwój systemów komputerowych

- Lata 30/40 XX w. - pierwsze komputery
- Systemy scentralizowane (np. IBM Mainframe od 1952 r.)
 - dostęp poprzez terminale
- Systemy sieciowe
 - Architektury klient-server
 - wykorzystanie sieci komputerowych (LAN, WAN)
 - dwuwarstwowe, trójwarstwowe, wielowarstwowe
 - rozproszone BD | rozproszone przetwarzanie danych (scentralizowana BD w sieci WAN, np. Internet)
 - przetwarzanie w „chmurze” (ang. *cloud computing*) XXI w.



Systemy scentralizowane

- Oparte o duże komputery (*mainframes*); rozwój od początku 1950 roku.
- Użytkownicy korzystają z systemu poprzez zdalne terminale znakowe lub sieć lokalną;
- Posiadają scentralizowane systemy operacyjne.



Systemy sieciowe

- Początek Internetu: 29 października 1969 r. - powstały pierwsze węzły sieci ARPANET (ang. *Advanced Research Projects Agency Network*)
- Wykorzystują sieciowe systemy operacyjne - Unix BSD (rozwijany od 1969 roku)
- Różne rodzaje systemów:
 - Unix/Linux,
 - Windows (serwery i stacje robocze),
 - system Novell NetWare.



Cele projektantów systemów sieciowych

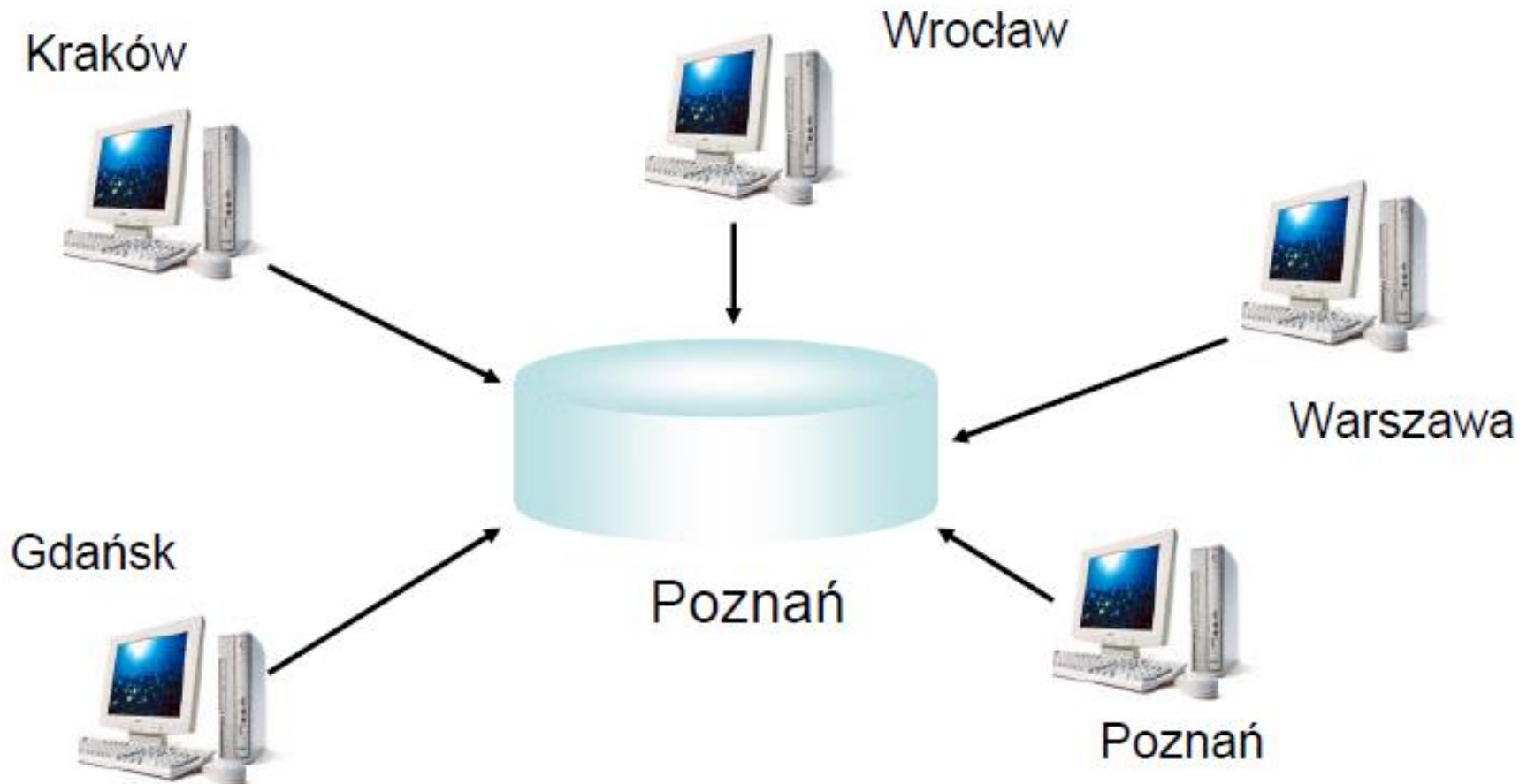
- Udostępnienie użytkownikom zdalnym zasobów i usług zarządzanych lokalnie przez serwery
- Zapewnienie udogodnień w dostępie do rozproszonych zasobów
- Zwiększenie efektywności i niezawodności przetwarzania.



Zalety systemów sieciowych:

- Współdzielenie zasobów;
- Ułatwiona komunikacja sieciowa;
- Zwiększona wydajność przetwarzania;
- Zwiększona niezawodność.

Scentralizowana BD - przykład





ROZPROSZONE BAZY DANYCH



Rozproszona baza danych

- **Rozproszona baza danych** - logicznie powiazany zbior danych, wspoluzytkowany przez wiele osob, fizycznie rozproszony w sieci komputerowej
- **Rozproszony system zarzadzania baza danych (RSZBD)** - oprogramowanie umozliwiajace zarzadzenie rozproszona baza danych
 - Rozproszenie danych jest niewidoczne (przezroczyste) dla uzytkownika

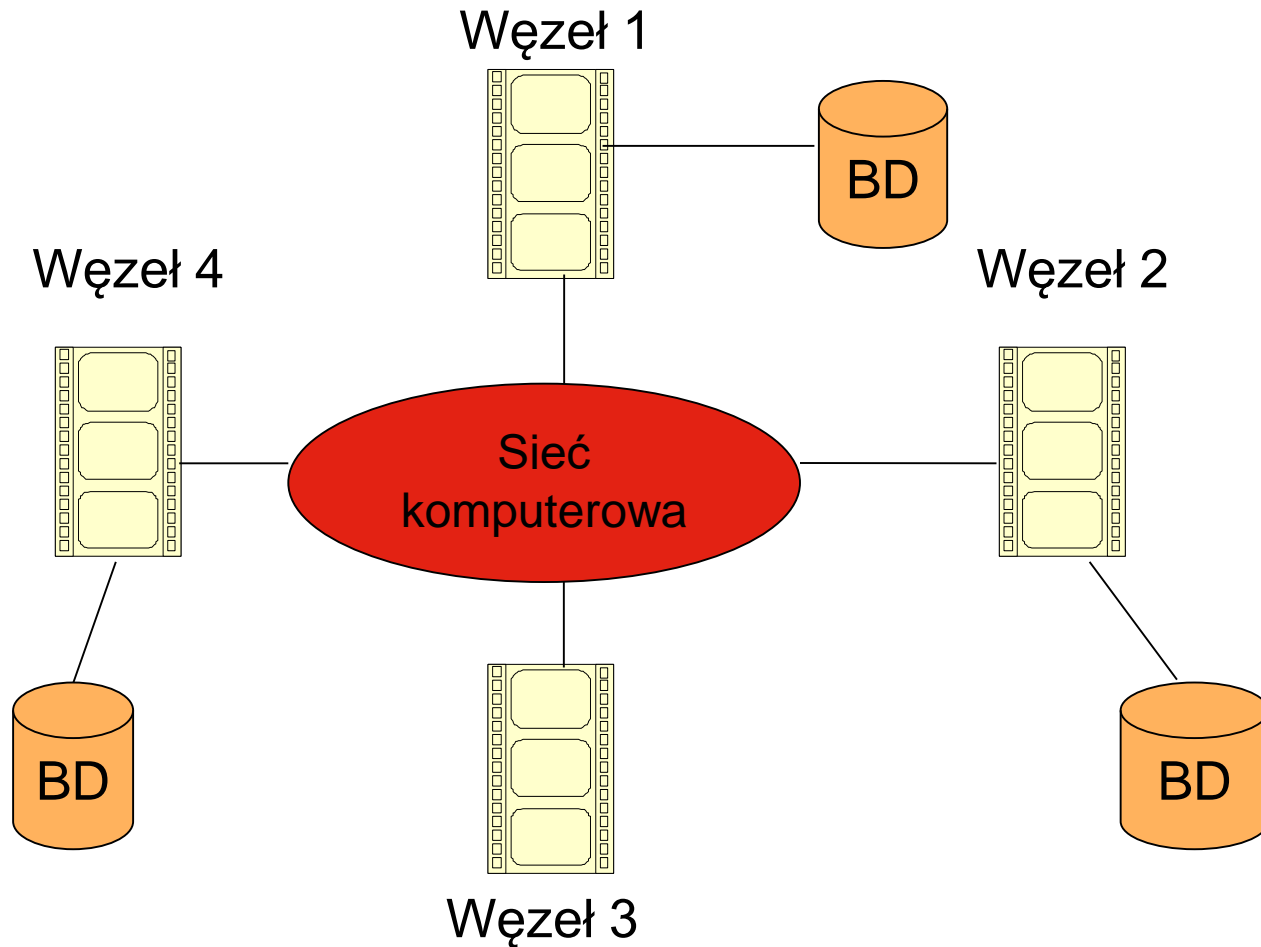


Rozproszona baza danych

- Zbiór współpracujących z sobą baz danych (lokalne BD)
- Każda z baz lokalnych znajduje się na innym serwerze
- Z punktu widzenia użytkownika bazy lokalne logicznie stanowią jedną BD

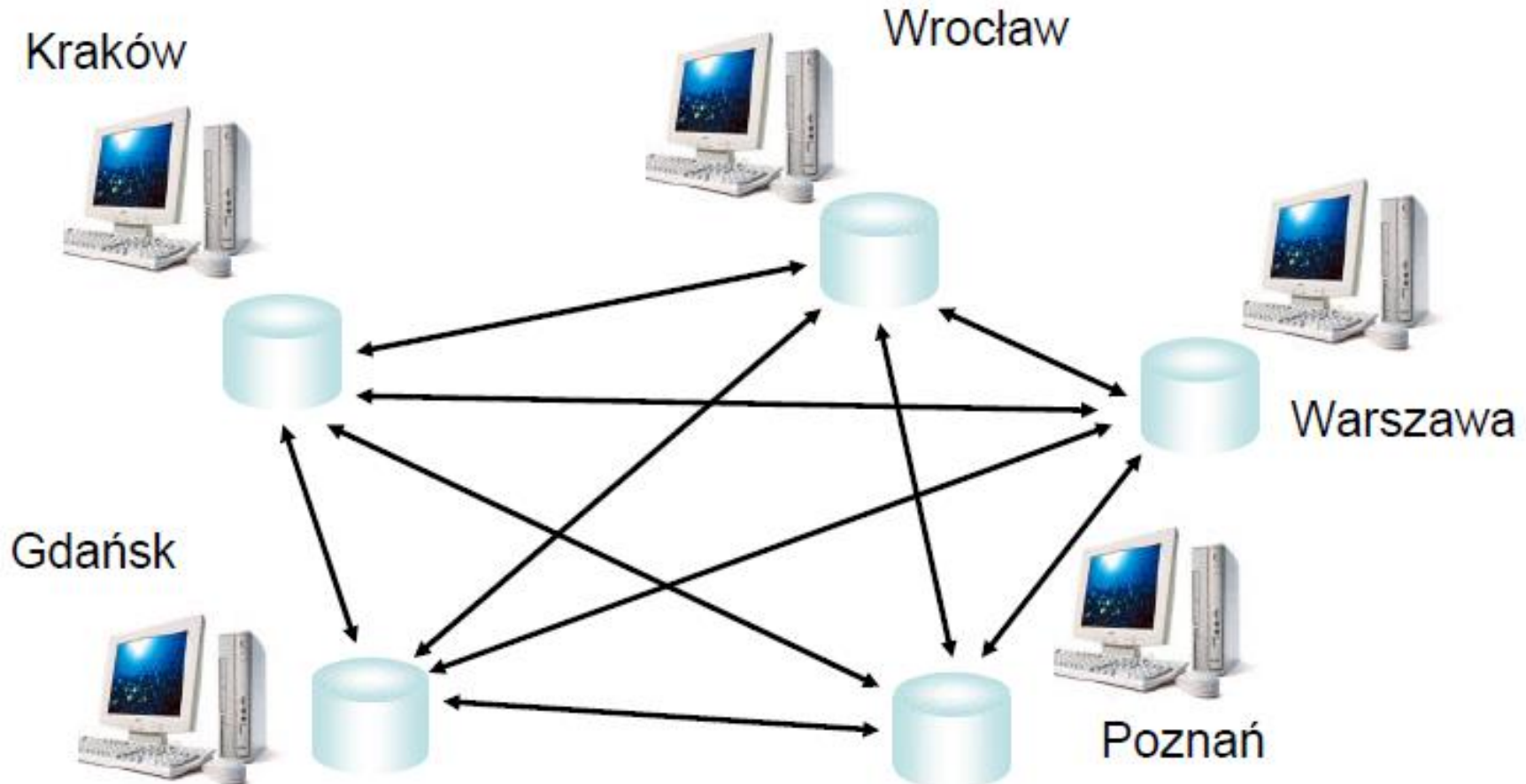


Rozproszona baza danych





Rozproszona BD - przykład





Zalety rozproszonych BD

- dane umieszczone „blisko” użytkownika → skrócenie czasu transmisji sieciowej
- mniejsze ryzyko utraty wszystkich danych na skutek awarii systemu
- wzrost niezawodności całego systemu



Wady rozproszonych BD

- utrudniony dostęp do pełnego (zintegrowanego) zbioru danych
- konieczność utrzymywania kopii danych (replik) i ich odświeżania



Komponenty architektury

- Sprzętowe komponenty rozproszonej bazy danych (RBD)
 - węzły - komputery, na których działa lokalna BD
 - sieć komputerowa
- Programowe komponenty RBD
 - protokoły sieciowe, np. TCP/IP, IPX/SP, IBM LU6.2, DEC Net
 - dedykowane oprogramowanie realizujące dostęp z jednej bazy danych do drugiej



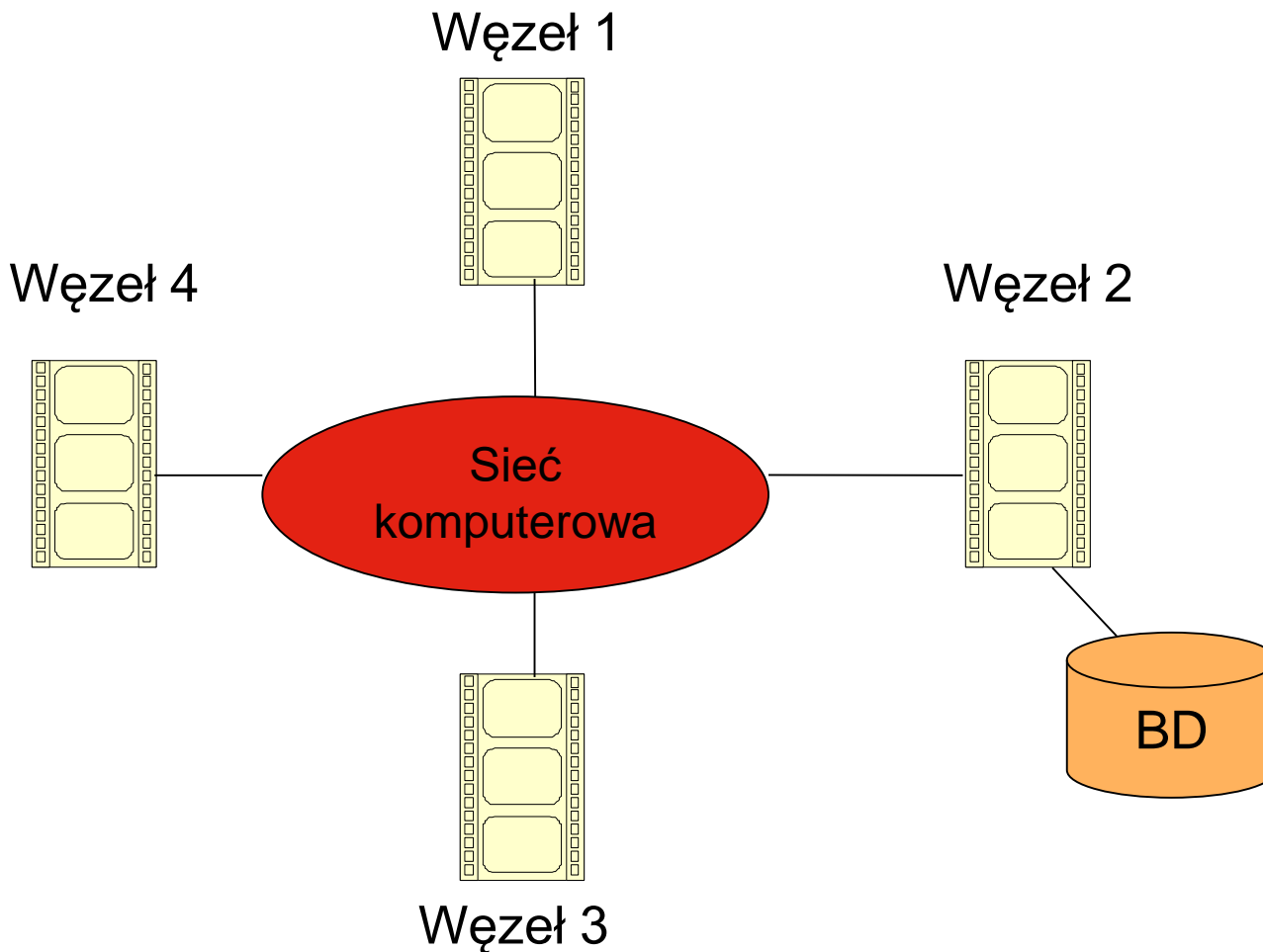
PRZETWARZANIE ROZPROSZONE

Przetwarzanie rozproszone

- Różnica pomiędzy rozproszonymi SZBD a przetwarzaniem rozproszonym
 - **Przetwarzanie rozproszone** - scentralizowana baza danych, do której można uzyskać dostęp poprzez sieć komputerową np. WAN - Internet
- ➔ Internetowe bazy danych



Przetwarzanie rozproszone





BAZY DANYCH W ARCHITEKTURACH SIECIOWYCH



Bazy danych w architekturze klient/serwer

- Aplikacja klienta i serwer bazy danych - znana jako dwuwarstwowa architektura aplikacji.
- Jednak model dwuwarstwowy nie jest odpowiedni dla WWW.
 - Duża liczba użytkowników, działających za pośrednictwem WWW, generujących ogromną liczbę transakcji.
 - Obciążenie dystrybucją, instalacją, uaktualnianiem i zarządzaniem kodem klienta na setkach PC.



Zastosowanie przeglądarek webowych

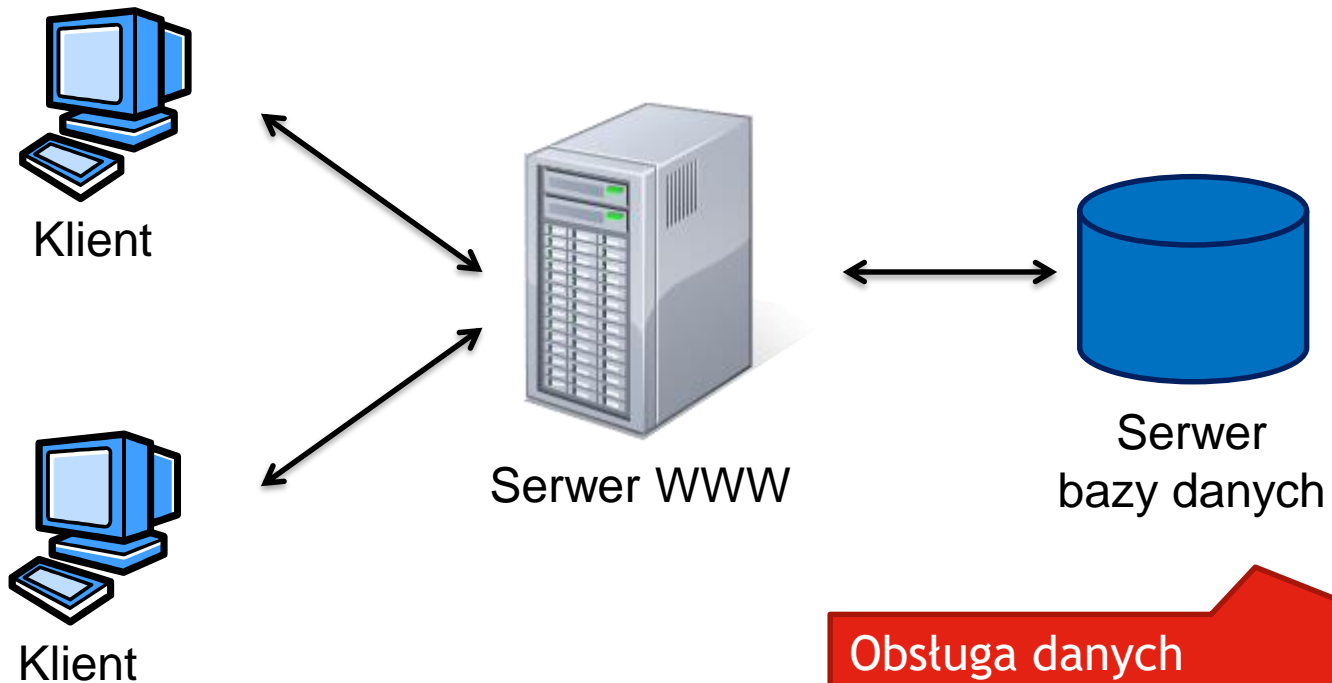
- Problemy te rozwiązuje się przechodząc na przeglądarki webowe, stanowiące szeroko rozpowszechniony i ogólnodostępny interfejs.
- Aplikacje korzystające z tej technologii mają architekturę trzywarstwową:
 - 1) dane na serwerze wspierającym (*back-end*),
 - 2) logika aplikacji i kod dostępu do danych na serwerze warstwy pośredniej,
 - 3) aplikacja klienta (*front-end*).



Architektura trójwarstwowa

Interfejs użytkownika
Kontrola wprowadzonych danych

Zadania aplikacji
Reguły biznesowe



Obsługa danych
Przechowywanie danych
Reguły integralności



Sposoby połączenia systemu klient/serwer z WWW

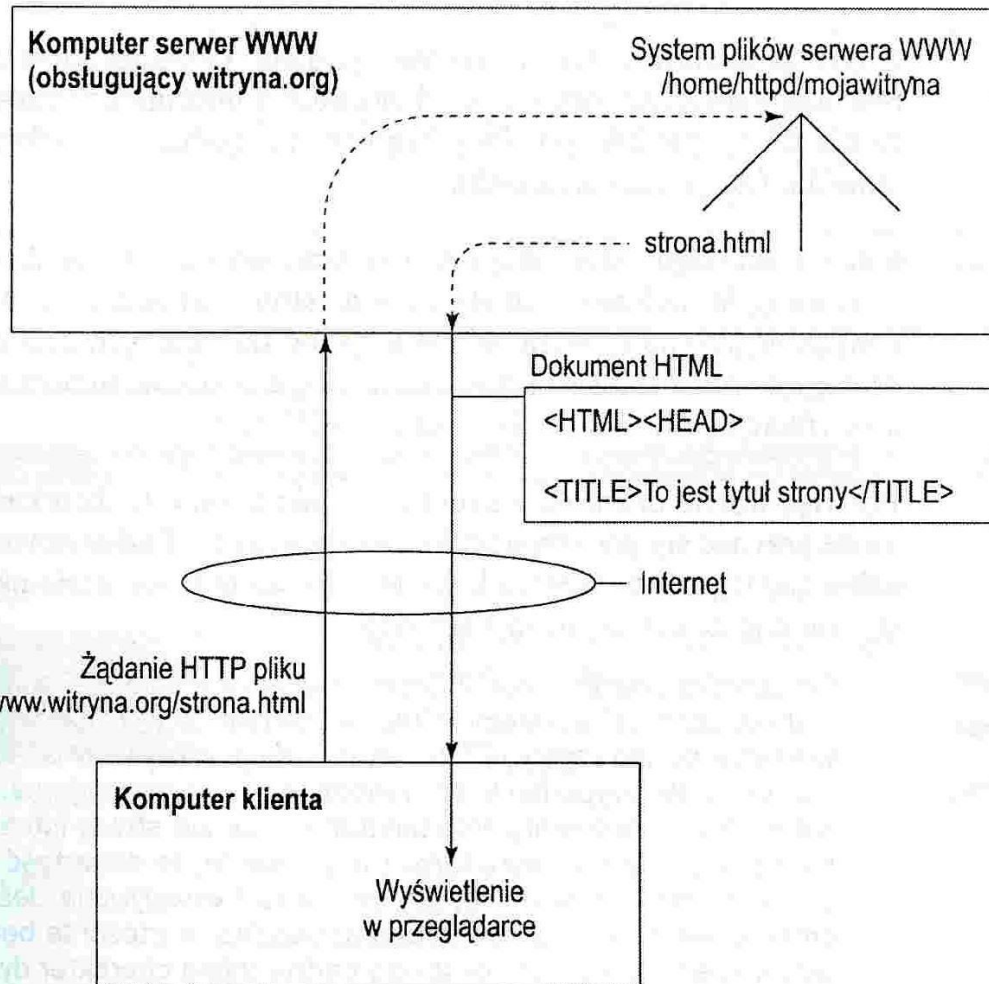
- 1) Przepisanie aplikacji w technologii internetowej, wykorzystującej interfejsy Java, HTML, HTTP i serwer webowy.
- 2) Model *data publishing*, który zakłada pobieranie istniejących danych klient/serwer i ich konwersję w serwerze webowym na format HTML.
- 3) Przemieszczanie części logiki aplikacji do przeglądarki klienta w formie apletów Javy lub ActiveX.



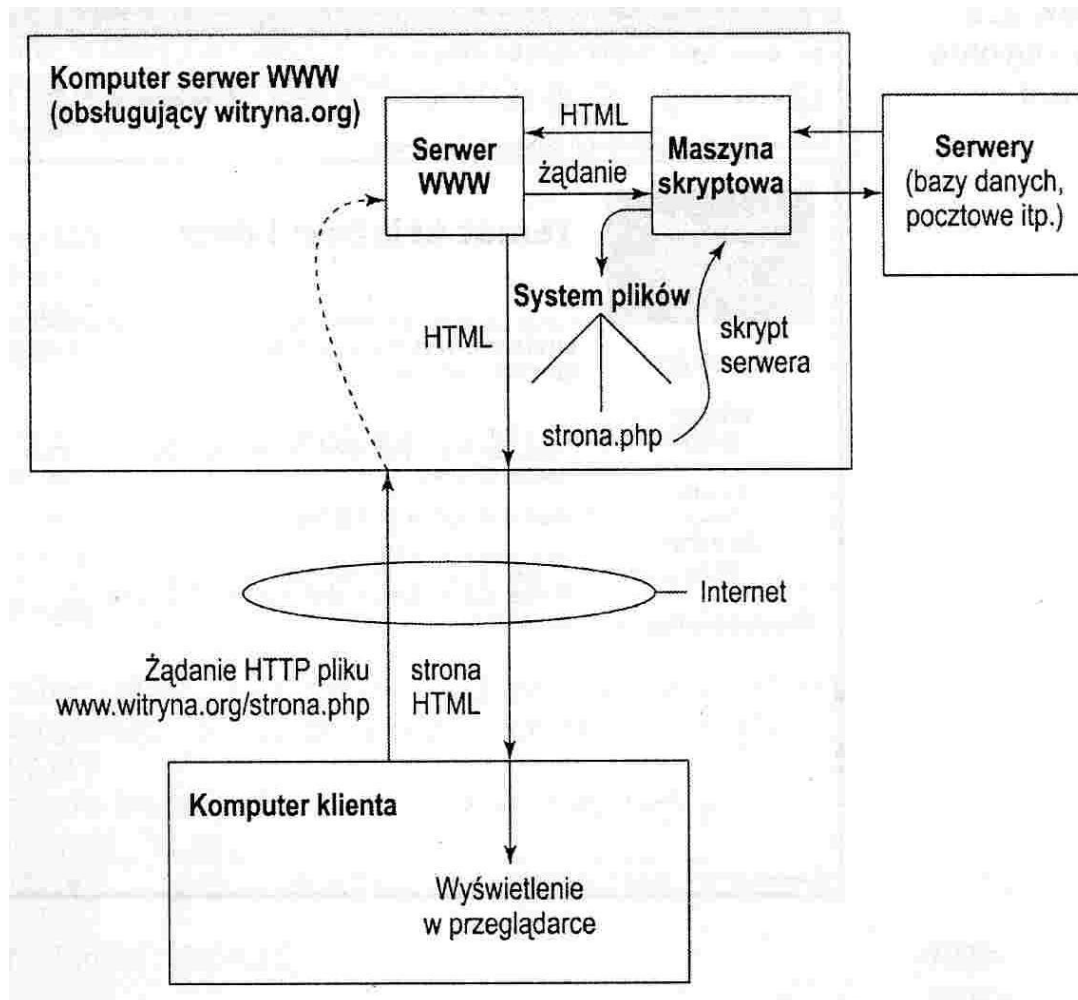
Cienki vs. Gruby klient

- Terminy funkcjonujące w sieciach o architekturze klient-serwer
- Cienki klient (ang. *lean, zero, slim, thin client*)
 - np. przeglądarka internetowa
- Gruby klient (ang. *heavy, rich, thick, fat client*)
 - np. Swing, ActiveX

Realizacja żądań HTTP



Działanie skryptów na serwerze





Rich Internet Application (RIA)

- Aplikacje internetowe, oferujące bogaty, dynamiczny, jednoekranowy interfejs, tzw. (ang.) *one-screen-application*.
- Rozwiązaniem trójwarstwowe:
 - baza danych,
 - oprogramowanie na serwerze i
 - przeglądarka internetowa.
- Każda z tych warstw jest budowana z użyciem innych technologii. Kluczowy jest interfejs uruchamiany w przeglądarce. Oprogramowanie to jest tworzone głównie w JavaScript.

Model trójwarstwowy

Warstwa prezentacji



Warstwa biznesowa



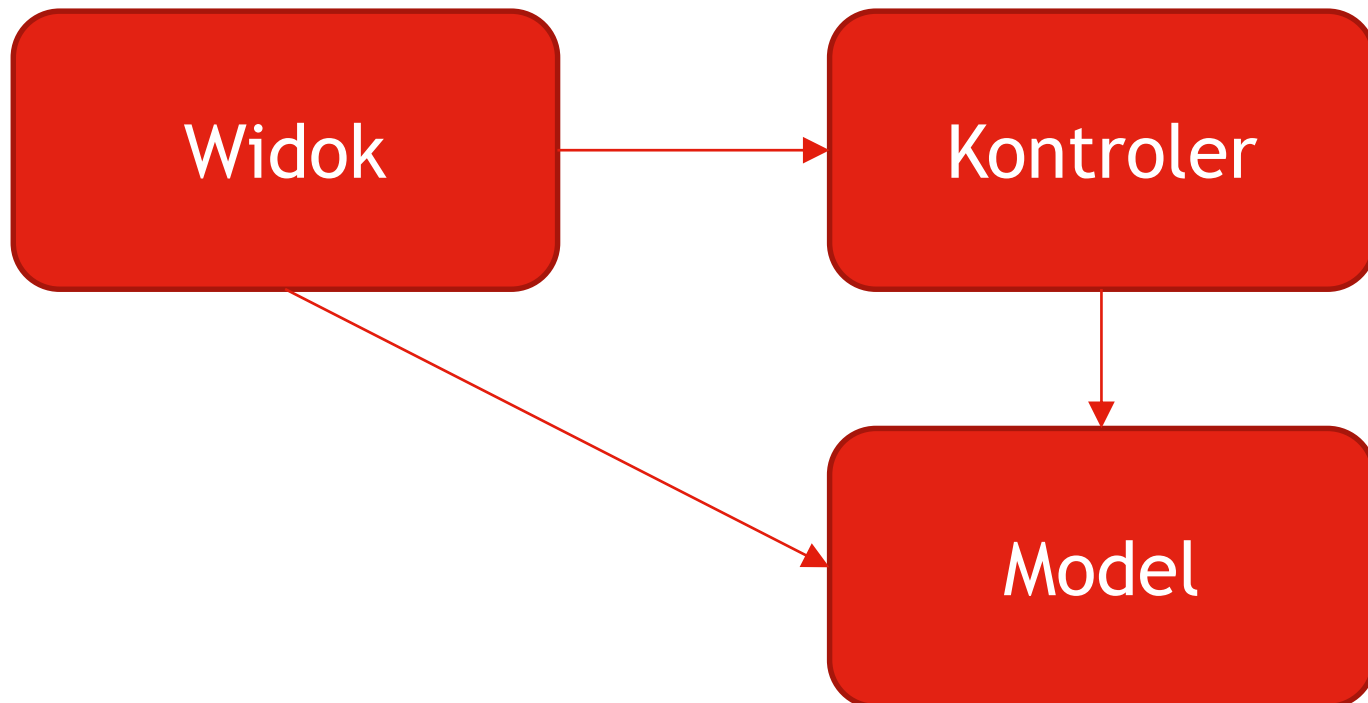
Warstwa integracji



Baza danych

Model-widok-kontroler (MVC)

- Popularny wzorzec projektowy
 - frameworki: Rails, ...
- Witryny internetowe to aplikacje MVC





Technologie

JĘZYKI PROGRAMOWANIA



TIOBE Index for October 2020

Oct 2020	Oct 2019	Change	Programming Language	Ratings	Change
1	2	↑	C	16.95%	+0.77%
2	1	↓	Java	12.56%	-4.32%
3	3		Python	11.28%	+2.19%
4	4		C++	6.94%	+0.71%
5	5		C#	4.16%	+0.30%
6	6		Virtual Basic .NET	3.97%	+0.23%
7	7		JavaScript	2.14%	+0.06%
8	9	↑	PHP	2.09%	+0.18%
9	15	↑↑	R	1.99%	+0.73%
10	8	↓	SQL	1.57%	-0.37%



TIOBE Index for September 2019

Sep 2019	Sep 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.661%	-0.78%
2	2		C	15.205%	-0.24%
3	3		Python	9.874%	+2.22%
4	4		C++	5.635%	-1.76%
5	6	↑	C#	3.399%	+0.10%
6	5	↓	Virtual Basic .NET	3.291%	-2.02%
7	8	↑	JavaScript	2.128%	-0.00%
8	9	↑	SQL	1.944%	-0.12%
9	7	↓	PHP	1.863%	-0.91%
10	10		Objective-C	1.840%	+0.33%



TIOBE Index for October 2018

Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.801%	+5.37%
2	2		C	15.376%	+7.00%
3	3		C++	7.593%	+2.59%
4	5	↑	Python	7.156%	+3.35%
5	8	↑	Virtual Basic .NET	5.884%	+3.15%
6	4	↓	C#	3.485%	-0.37%
7	7		PHP	2.794%	+0.00%
8	6	↓	JavaScript	2.280%	-0.37%
9	-	↑↑	SQL	2.038%	+2.04%
10	16	↑↑	Swift	1.500%	-0.17%



TIOBE Index for September 2017

Sep 2017	Sep 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.687%	-5.55%
2	2		C	7.382%	-3.57%
3	3		C++	5.565%	-1.09%
4	4		C#	4.779%	-0.71%
5	5		Python	2.983%	1.32%
6	7	↑	PHP	2.210%	-0.64%
7	6	↓	JavaScript	2.017%	-0.91%
8	9	↑	Virtual Basic .NET	1.982%	-0.36%
9	10	↑	Perl	1.952%	-0.38%
10	12	↑	Ruby	1.933%	-0.03%



TIOBE Index for October 2016

Oct 2016	Oct 2015	Change	Programming Language	Ratings	Change
1	1		Java	18.799%	-0.74%
2	2		C	9.835%	-6.35%
3	3		C++	5.797%	+0.05%
4	4		C#	4.367%	-0.46%
5	5		Python	3.775%	-0.74%
6	8	↑	JavaScript	2.751%	+0.46%
7	6	↓	PHP	2.741%	+0.18%
8	7	↓	Virtual Basic .NET	2.660%	+0.20%
9	9		Perl	2.495%	+0.25%
10	14	↑↑	Objective-C	2.263%	+0.84%



TIOBE Index for October 2015

Oct 2015	Oct 2014	Change	Programming Language	Ratings	Change
1	2	↑	Java	19.543%	+6.04%
2	1	↓	C	16.190%	-1.47%
3	4	↑	C++	5.749%	+0.88%
4	5	↑	C#	4.825%	+0.08%
5	8	↑	Python	4.512%	+2.18%
6	7	↑	PHP	2.561%	-0.38%
7	13	↑↑	Visual Basic .NET	2.462%	+0.71%
8	12	↑↑	JavaScript	2.292%	+0.52%
9	9		Perl	2.247%	+0.13%
10	16	↑↑	Ruby	1.825%	+0.70%

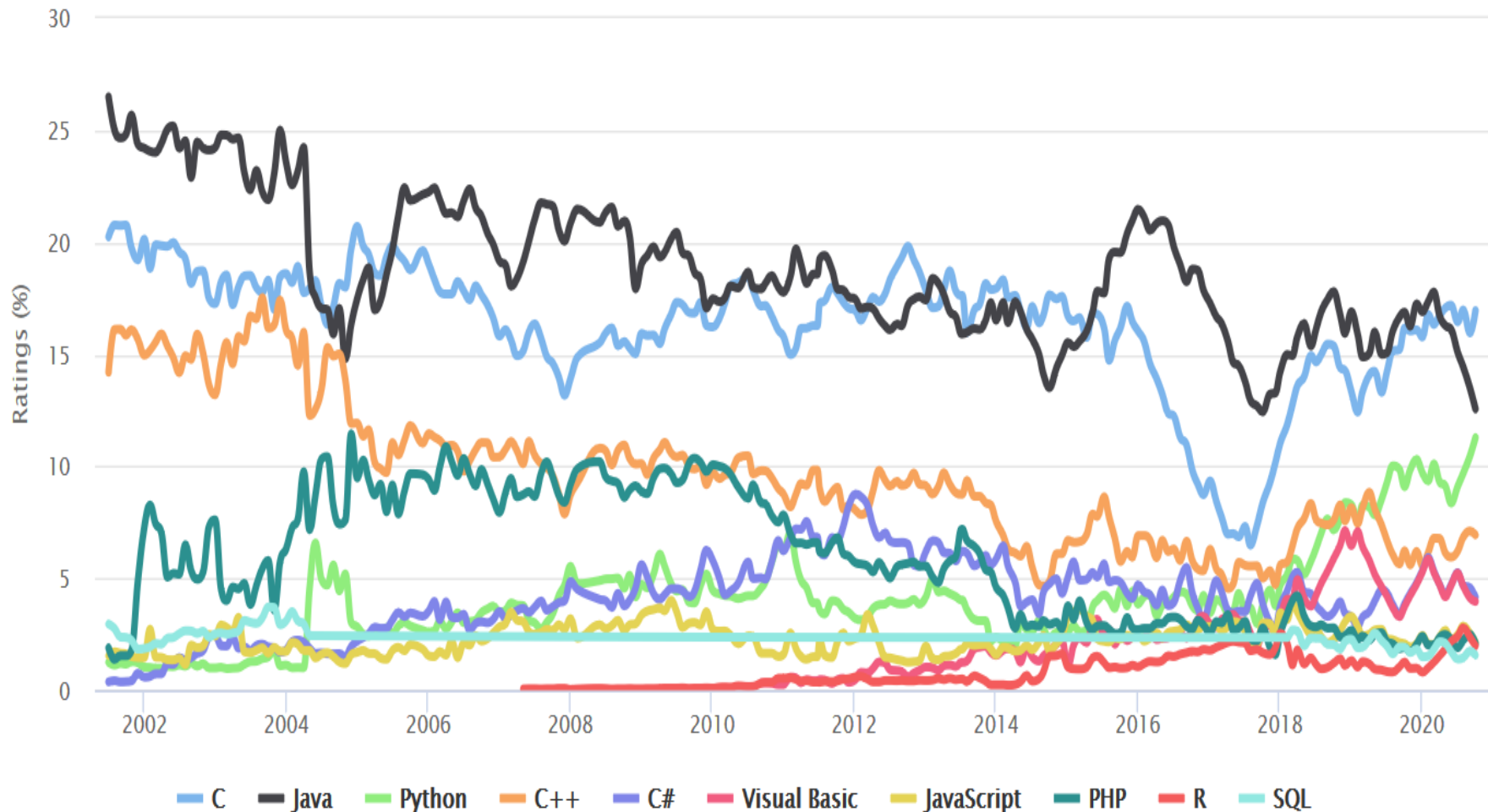


TIOBE Index for September 2014

Sep 2014	Sep 2013	Change	Programming Language	Ratings	Change
1	1		C	16.721%	-0.25%
2	2		Java	14.140%	-2.01%
3	4	↑	Objective-C	9.935%	+1.37%
4	3	↓	C++	4.674%	-3.99%
5	6	↑	C#	4.352%	-1.21%
6	7	↑	Basic	3.547%	-1.29%
7	5	↓	PHP	3.121%	-3.31%
8	8		Python	2.782%	-0.39%
9	9		JavaScript	2.448%	+0.43%
10	10		Transact-SQL	1.675%	-0.32%



TIOBE Programming Community Index



źródło: www.tiobe.com/tiobe-index/



Ranking SPECTRUM

Oct 2014	Programming Language	Ratings
1	Python	100.0%
2	C++	99.7%
3	Java	97.5%
4	C	96.7%
5	C#	89.4%
6	PHP	84.9%
7	R	82.9%
8	JavaScript	82.6%
9	Go	76.4%
10	Assembly	74.1%





Wielość języków/technologii

- PHP
- JavaScript
 - obecnie także po stronie serwerowej, z użyciem silnika nazywanego: Node.js
- Java / JSP
- Python
- Ruby
- ASP.NET - zwykle z językiem programowania C#
- inne



SERWERY BAZ DANYCH



Wersje BD (X'2020)

Produkt	Producent	Najnowsza wersja	Licencja
MySQL	Oracle	8.0.21	GPL lub komercyjna
MariaDB	Monty Program Ab	10.5.6	GPL v. 2
Microsoft SQL Server	Microsoft	2019 (15.0.2000.5)	Microsoft EULA
Oracle Database	Oracle	19c (12.2.0.3)	komercyjna
PostgreSQL	PostgreSQL Global Development Group	13.0	PostgreSQL
SQLite	D. Richard Hipp	3.33.0	Public domain
Informix	IBM	14.10.FC3	EULA
DB2	IBM	11.5	EULA



DB-Engines Ranking X 2019

Oct 2019	Oct 2018	Change	Programming Language	Ratings	Change
1	1		Oracle	16.661%	-0.78%
2	2		MySQL	15.205%	-0.24%
3	3		Microsoft SQL Server	9.874%	+2.22%
4	5		PostgreSQL	5.635%	-1.76%
5	4		MongoDB	3.399%	+0.10%
6	6		IBM DB2	3.291%	-2.02%
7	8	↑	Elasticsearch	2.128%	-0.00%
8	7	↓	Redis	1.944%	-0.12%
9	9		Microsoft Access	1.863%	-0.91%
10	11		Cassandra	1.840%	+0.33%



DB-Engines Ranking X 2018

Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Oracle	1319.27	-29.54
2	2		MySQL	1178.12	-120.71
3	3		Microsoft SQL Server	1058.33	-151.99
4	5		PostgreSQL	419.39	+46.12
5	4		MongoDB	363.19	+33.79
6	6		DB2	179.69	-14.90
7	8	↑	Redis	145.29	+23.24
8	7	↑	Elasticsearch	142.33	+22.09
9	9	↓	Microsoft Access	136.80	+7.35
10	11	↓	Cassandra	123.39	-1.40



Duże bazy danych

- Microsoft SQL Server
- Oracle DB
- IBM Informix Extended Parallel Server (XPS)
- SAP Sybase Adaptive Server Enterprise 16.0 (ASE)
- InterBase 2020



Średnie bazy danych

- MySQL
- MariaDB
- PostgreSQL
- Firebird 3.0.5
- Microsoft Visual FoxPro 9.0
- MS Access 2019 / Office 365



Małe bazy danych

- mSQL 4.2
- SQLite

- Nie nadają się do tworzenia wielodostępowych aplikacji internetowych.



MySQL

- Silniki baz danych
 - ISAM
 - MyISAM
 - **InnoDB**
 - BerkeleyDB (BDB)
 - MERGE
 - HEAP



Kryteria wyboru składowych systemu dla potrzeb IBD

- Ilość i rodzaj danych przechowywanych w systemie.
- Szacowana liczba użytkowników mogących jednocześnie korzystać z systemu.
- Skalowalność i wieloplatformowość.
- Bezpieczeństwo i ochrona danych.



Kryteria wyboru składowych systemu dla potrzeb IBD (2)

- Zabezpieczenie przed kradzieżą kodu oraz utratą danych.
- Środowisko programistyczne.
- Obsługiwana platforma sprzętowa (najczęściej pod kątem wykorzystania nowych możliwości SO).
- Koszt.

Zestaw oprogramowania

- LAMP - Linux + Apache + MySQL/MariaDB + PHP/Perl/Python
- WAMP - Windows + Apache + MySQL + PHP
- MAMP - Mac OS X + Apache + MySQL + PHP
- XAMPP - X (Cross-platform) Apache, MySQL, PHP, Perl

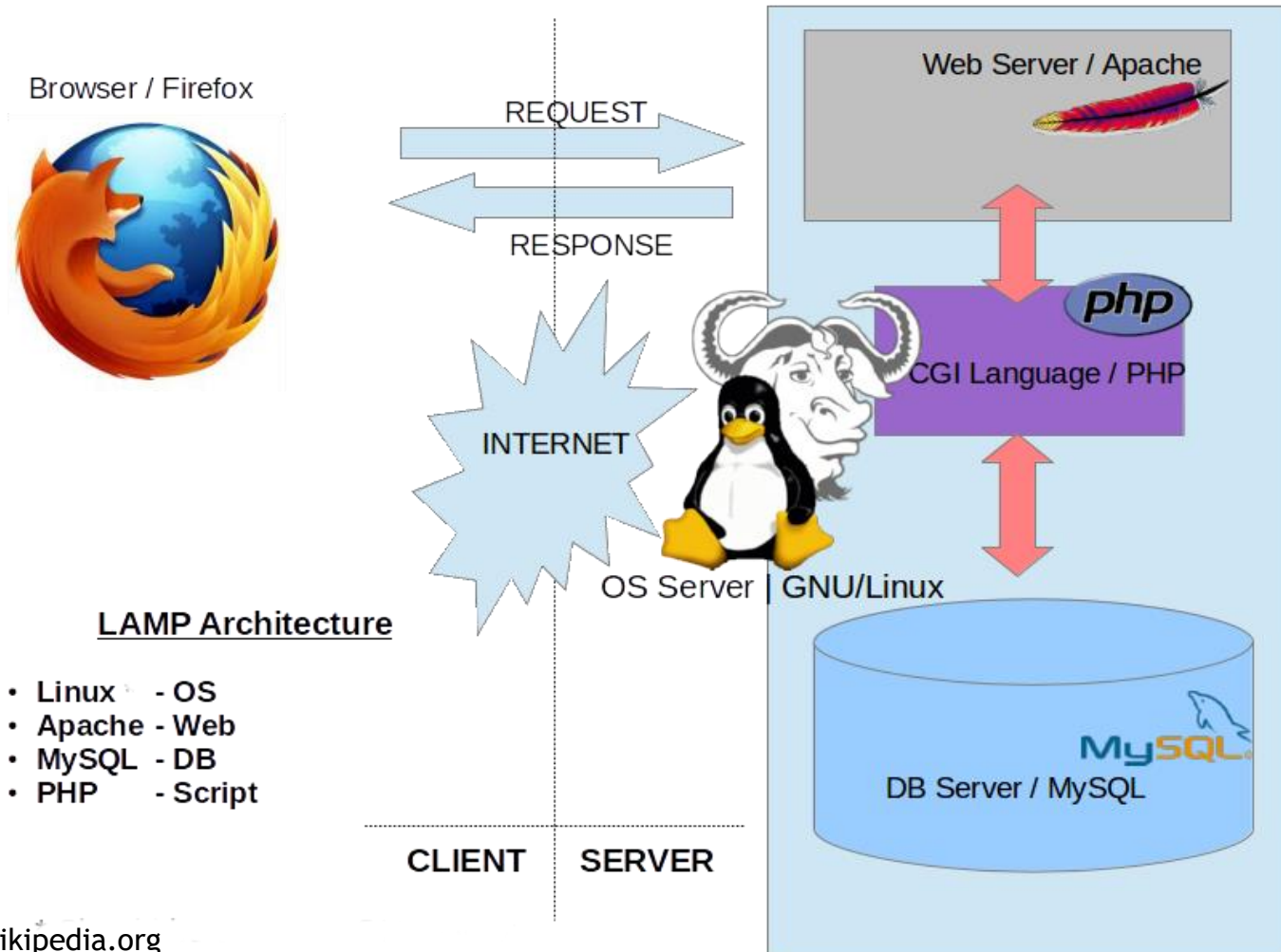


LAMP:





Uproszczony schemat architektury LAMP



LAMP Architecture

- Linux - OS
- Apache - Web
- MySQL - DB
- PHP - Script



Bazy danych obsługiwane przez PHP


Baza danych	Typ	Obsługa	Platforma	Licencja	Uwagi
Adabas D	R	ODBC	U, W	Komercyjna	Niemiecka, rozprowadzana z SuSE Linux
DBA/DBM	Płaski plik	Warstwa abstrakcji	U	Open source, komercyjna	Sleepycat, Gnu DBM, cdb
dBase	P	Tylko import	W	Komercyjna	Bez SQL
Empress	R	ODBC	U, W	Komercyjna	Korporacyjna, dostępne JDBC
filepro	R	Tylko import	U, W	Komercyjna	Nie do zadań produkcyjnych
IBM DB2	R	ODBC	U, W	Komercyjna	Korporacyjna, dostępne JDBC
Informix	R	Własna	U, W	Komercyjna	Korporacyjna
Interbase	R	Własna	U, W	Komercyjna	Korporacyjna, dostępne JDBC



Bazy danych obsługiwane przez PHP (2)

Baza danych	Typ	Obsługa	Platforma	Licencja	Uwagi
MS Access	R	ODBC	W	Komercyjna	Nie do zadań produkcyjnych
MS SQL Server	R	Własna	W	Komercyjna	Korporacyjna
mSQL	R	Własna	U	Shereware	Bardzo mała
MySQL	R	ODBC	U, W	Komercyjna, Shareware	Kilka licencji
Oracle	R	Własna	U, W	Komercyjna	Korporacyjna
Oracle8	R	Własna	U, W	Komercyjna	Korporacyjna, integracja z Java
PostgreSQL	O-R	Własna	U	Open source	Wsparcie komercyjne
Solid	R	ODBC	U, W	Komercyjna	Wbudowana baza danych, firma fińska
SQLite	FF	Własna	U, W	Open source	Wbudowana baza danych
Sybase	R	Własna	U, W	Komercyjna	Korporacyjna

Warstwa abstrakcji bazy danych

- DBAL (*Database abstraction layer*)
- w PHP:
 - AdoDB (*Active Data Objects Database*)
 - PEAR DB - biblioteki PEAR (PHP Extension and Application Repository)
 - Metabase 
 - ➔ PEAR MDB2
 - PDO (*PHP Data Objects*) - obiekty danych PHP



PEAR DB

- Nakładka na rozszerzenia bazodanowe PHP
- Pakiet zorientowany obiektowo
- Cel: uogólnienie procesu komunikacji z bazami danych
- Dokumentacja i przykłady:
<http://pear.php.net/manual/pl/package.database.db.intro-connect.php>



Szablony PHP

- Smarty
- <http://www.smarty.net>
- Oddzielenie kodu PHP od statycznego kodu HTML (trójwarstwowa architektura)



Przykład użycia Smartów

```
// ----- PHP page -----  
include('Smarty.class.php');  
  
// create object  
$smarty = new Smarty;  
  
// assign some content. This would typically come from  
// a database or other source, but we'll use static  
// values for the purpose of this example.  
$smarty->assign('name', 'george smith');  
$smarty->assign('address', '45th & Harris');  
  
// display it  
$smarty->display('index.tpl');
```



Przykład użycia Smartów (cd.)

```
// ----- smarty template -----
```

```
<html>  
<head>  
<title>User Info</title>  
</head>  
<body>  
  
User Information:<p>  
  
Name: {$name}<br>  
Address: {$address}<br>  
  
</body>  
</html>
```




PHP i Smarty

- Przykładowa implementacja architektury MVC dla aplikacji opartej na PHP może wyglądać następująco:
 - Kontroler - skrypty PHP.
 - Model - obiektowe klasy języka PHP.
 - Widok - szablony Smarty, wspomagane HTML, CSS i innymi podobnymi technologiami.



Frameworki PHP

- Symfony
- Zend
- CodeIgniter
- Kohana
- PRADO Framework



TECHNOLOGIE UDOSTĘPNIANIA INFORMACJI



Technologie udostępniania informacji

- HTML (HyperText Markup Language)
- ISAPI (Internet Server Application Programming Interface)
- CGI (Common Gateway Interface)
- ASP (Active Server Pages)
- PHP (PHP Hypertext Preprocessor)
- JSP (JavaServer Pages)
- XML (Extensible Markup Language)





HTML (HyperText Markup Language)

- Hipertekstowy język znaczników (ang. *tags*) stosowany do pisania stron WWW

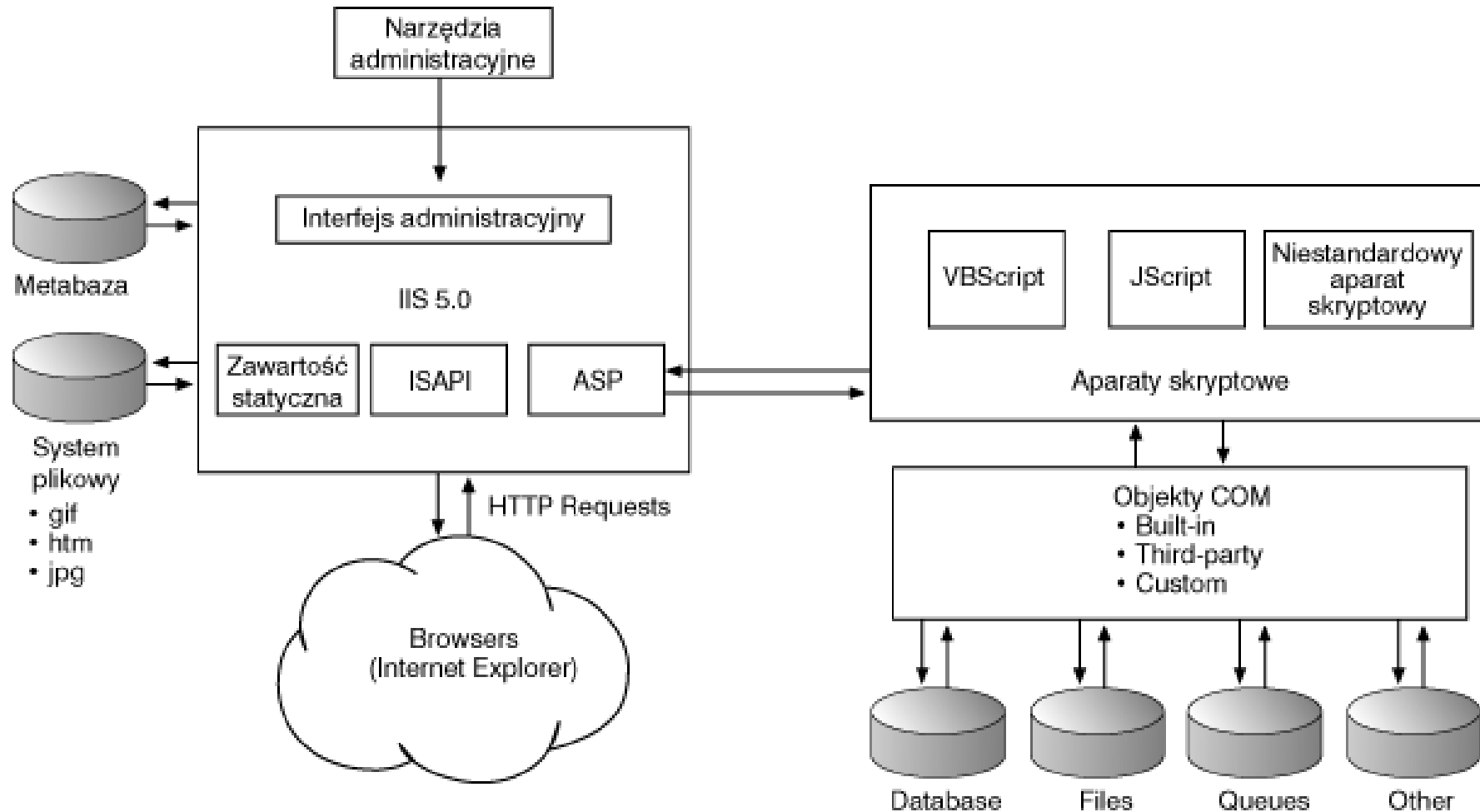
```
<html>
  <head>
    <title> Strona internetowa </title>
  </head>
  <body>
    <p> Witamy... </p>
  </body>
</html>
```



ISAPI (Internet Server API)

- API (*Application Programming Interface*) - rozszerzenie programistyczne firmy Microsoft dla serwera WWW o nazwie IIS - *Internet Information Services* (np. składnik Windows 2016 Server).
- Główne zastosowania: jako wstępny filtr lub aplikacja na serwerze.
- Wydajniejsze od CGI (m.in. jednokrotne załadowanie biblioteki DLL - jeden proces i uruchamianie wątków).
- Zadania wykonywane są po stronie serwera.

Architektura IIS 5.0





CGI (*Common Gateway Interface*)

- Standard wymiany informacji (1993):
przeładowarka WWW → usługa WWW → skrypt CGI
→ pamięć
- Główne zastosowania: przetwarzanie danych
(formularze).
- Zadania wykonywane są po stronie serwera.
- Wada: mała wydajność (ale oprogramowanie
FastCGI)
- Zagrożenie dla bezpieczeństwa (przy nieuwadze
programisty).

PHP

- *PHP Hypertext Preprocessor*
- Dawniej skrót rozwijano jako *Personal Home Page*
- Skryptowy język programowania stosowany głównie do tworzenia dynamicznych stron WWW
- Wykonywany po stronie serwera





PHP

- Zalety:
 - Obsługa bardzo wielu platform,
 - Licencjonowanie - OpenSource,
 - Zachowanie poufności kodu - wykonywanie po stronie serwera,
 - Dostępność narzędzi wspomagających.
- Wady:
 - Trudności z przenaszalnością (ze względu na bezpośredni dostęp do bazy danych),
 - Obsługi wyjątków dopiero od wersji 5 i to częściowo.



PHP

```
<html>
<head>
  <title>Przyklad</title>
</head>
<body>
  <?php echo "Witaj swiecie!"; ?>
</body>
</html>
```



Style PHP

- Minimalny

```
<?php  
require_once("dompdf_config.inc.php");
```

- Maksymalny

```
$html =  
    '<html><body>'.  
    '<p>Hello World!</p>'.  
    '</body></html>';
```

- Średni

```
$dompdf = new DOMPDF();  
$dompdf->load_html($html);  
  
$dompdf->render();  
$dompdf->stream("hello_world.pdf");  
  
?>
```

Źródło: <https://code.google.com/p/dompdf/wiki/FAQ>



ASP (Active Server Pages)

- Standard firmy Microsoft
- W plikach *.asp umieszczony jest skrypt VBS (Visual Basic Script)
- Technologia umożliwiająca połączenie HTML'a, specjalnych skryptów i komponentów COM
- Serwer wykonuje napotkane polecenia i do przeglądarki zwraca HTML (Uwaga: komponenty COM mogą być wykonywane również po stronie klienta)
- Współpraca z bazami danych: ODBC, OLE DB, ADO
- Wada: głównie związane z jednym systemem operacyjnym



ASP

```
<html>
<head>
  <title>Przyklad</title>
</head>
<body>
  <%
    response.write("Hello World!")
  %>
</body>
</html>
```



J2EE

- Servlety
- JSP
- Enterprise JavaBeans (EJB)
 - sesyjne EJB (ang. session EJB) bezstanowe i stanowe
 - sterowane komunikatami EJB (ang. message-driven EJB)
 - encyjne EJB (ang. entity EJB)
- inne



JSP (Java Server Pages)

- Technologia tworzenia dynamicznych stron WWW z wykorzystaniem języka wplecionego w kod strony
- Odmiana servletów - aplikacji w Javie uruchamianych po stronie serwera
- Strona JSP zamieniana jest na servlet, który wykonuje operacje i każde kolejne zapytania do tej strony (prekompilacja - zamiana na servlety już przy uruchamianiu)

JSP

- Zastosowania:
 - Duże serwisy - oddzielenie warstwy prezentacji (kod HTML) od warstwy logiki (kod Javy),
 - Niezależność od platformy,
 - Modelowanie z użyciem narzędzi CASE (język obiektowy).
- Wady
 - Duża pracochłonność,
 - Raczej nie wykorzystana w małych serwisach (duże wymagania - instalacja oraz wyższy koszt).



XML

- Język znaczników umożliwiający tworzenie swoich własnych znaczników formatujących, definiujących w DTD (Document Type Definition) dokumentu.
- Niektóre odmiany związane z XMLem:
 - Określające strukturę dokumentu: DTD, XML-Data, XML-Schema,
 - Określające dowiązania: XLink, Xpath
 - W zakresie prezentacji i przetwarzania: XSL (Extensible Stylesheet Language), XSLT (XSL Transforming), DOM (Document Object Model).



XML

- Wady:
 - nie wspomagany przez niektóre przeglądarki
 - wymaga większej dokładności przy pisaniu niż HTML



Rozszerzenie HTML po stronie klienta

- Uruchamianie aplikacji po stronie klienta (ang. *client-side*)
- CSS, Dynamic HTML
- Skrypty wykonywane po stronie klienta
 - Java Script
 - VBScript
- Aplety Java
- Animacje Flash



Aplety Java

- Przesyłane wraz z dokumentem HTML i wykonywane na wirtualnej maszynie wbudowanej w przeglądarkę
- Zastosowania:
 - Efekty graficzne
 - Narzędzia pomocnicze np. dla chata
 - Realizacja tzw. cienkich klientów
- Brak możliwości dostępu do zasobów lokalnych komputera



Java Script

- Język skryptowy interpretowany przez przeglądarkę.
- Zastosowania:
 - Programy reagujące na zdarzenia,
 - Elementy animacyjne,
 - Weryfikacja poprawności formularzy

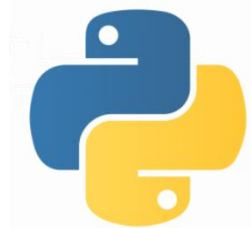
PERL

- Interpretowany język programowania.
- Program jest wcześniej kompilowany do kodu pośredniego.



Python

- Język programowania wysokiego poziomu ogólnego przeznaczenia.
- Często używany jako język skryptowy.
- Django - wysokopoziomowy, opensource'owy framework przeznaczony do tworzenia aplikacji internetowych, napisany w Pythonie.
- Django opiera się na wzorcu projektowym podobnym do MVC nazywanym MVT (Model-View-Template).



Ruby On Rails (RoR)

- Framework do szybkiego tworzenia aplikacji webowych.
- Został napisany w języku Ruby z użyciem architektury MVC (ang. *Model-View-Controller*).
- RoR może współpracować z różnymi odmianami relacyjnych baz danych.





TECHNOLOGIE DOSTĘPU DO DANYCH



Technologie dostępu do danych

- ODBC (*Open DataBase Connectivity*)
- JDBC (*Java DataBase Connectivity*)
- OLE DB (*Object Linking and Embedding for DataBases*)
- DAO (*Data Access Object*)
- ADO (*ActiveX Data Object*)
- ADO.NET (*ActiveX Data Objects for .NET*)

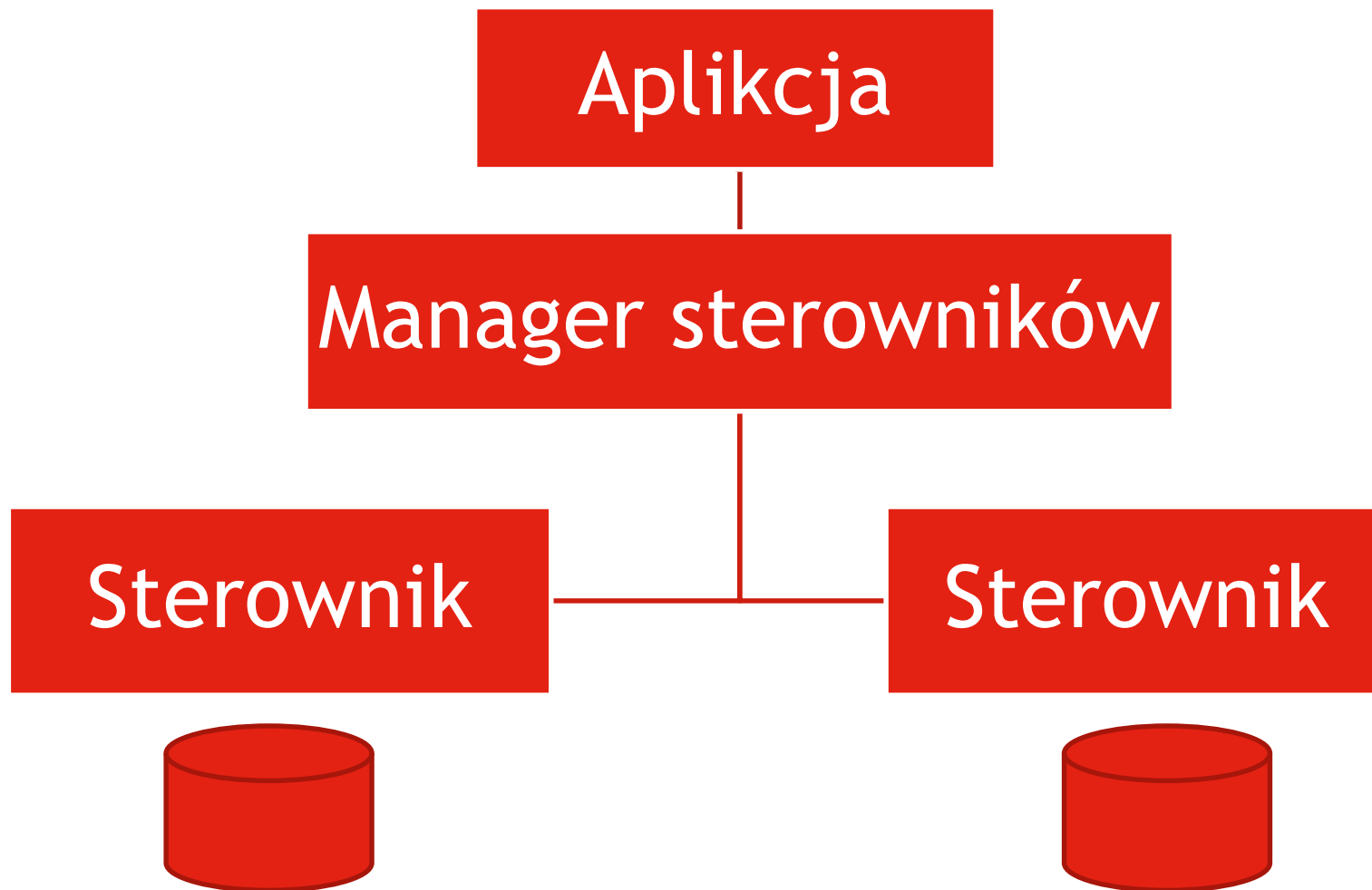


ODBC (*Open DataBase Connectivity*)

- Standard interfejsu (w oparciu o SQL) do heterogenicznych baz danych - możliwość zastosowania jednej aplikacji do różnych systemów zarządzania bazą danych.
- API niezależne od języka programowania, systemu operacyjnego i bazy danych.
- Ok. 200 sterowników do różnego rodzaju źródeł danych.
- Zastosowanie: dostęp do relacyjnych baz danych.
- Ograniczenia: złożone wykonywanie operacji, duża ilość kodu → wpływ na przetwarzanie.



ODBC (Open DataBase Connectivity)





JDBC (*Java DataBase Connectivity*)

- Standard interfejsu łączący aplikację Javy z zewnętrzną bazą danych.
- JDBC API (aplikacja → Driver Manager) → sterownik (np. JDBC-NET, JDBC-ODBC, sterownik wbudowany w API).
- SQLJ - standard umożliwiający osadzanie instrukcji SQL w plikach źródłowych (ORACLE, Sybase, IBM, Informix, Tandem).
 - Cel: łatwiejsze pisanie kodu, eliminacja błędów, translacja na JDBC.



OLE DB (*Object Linking and Embedding for DataBases*)

- Interfejs programistyczny firmy Microsoft służący do uzyskania dostępu do danych.
- Jest to obiekt COM, który funkcjonuje w podobny sposób jak ODBC, ale w odniesieniu do dowolnego źródła danych, a nie tylko do baz danych SQL.
- Bezpośredni dostęp lub poprzez ODBC.



ADO (*ActiveX Data Object*)

- Zbiór interfejsów (poziom aplikacji) korzystających z OLE DB.
- ActiveX - komponenty do użycia w różnych narzędziach, m.in. Delphi, VB, C++, Java.
- Przeznaczenie: system Windows.
- Uzależnienie od dostawcy (poprzez nakładkę OLE-DB).

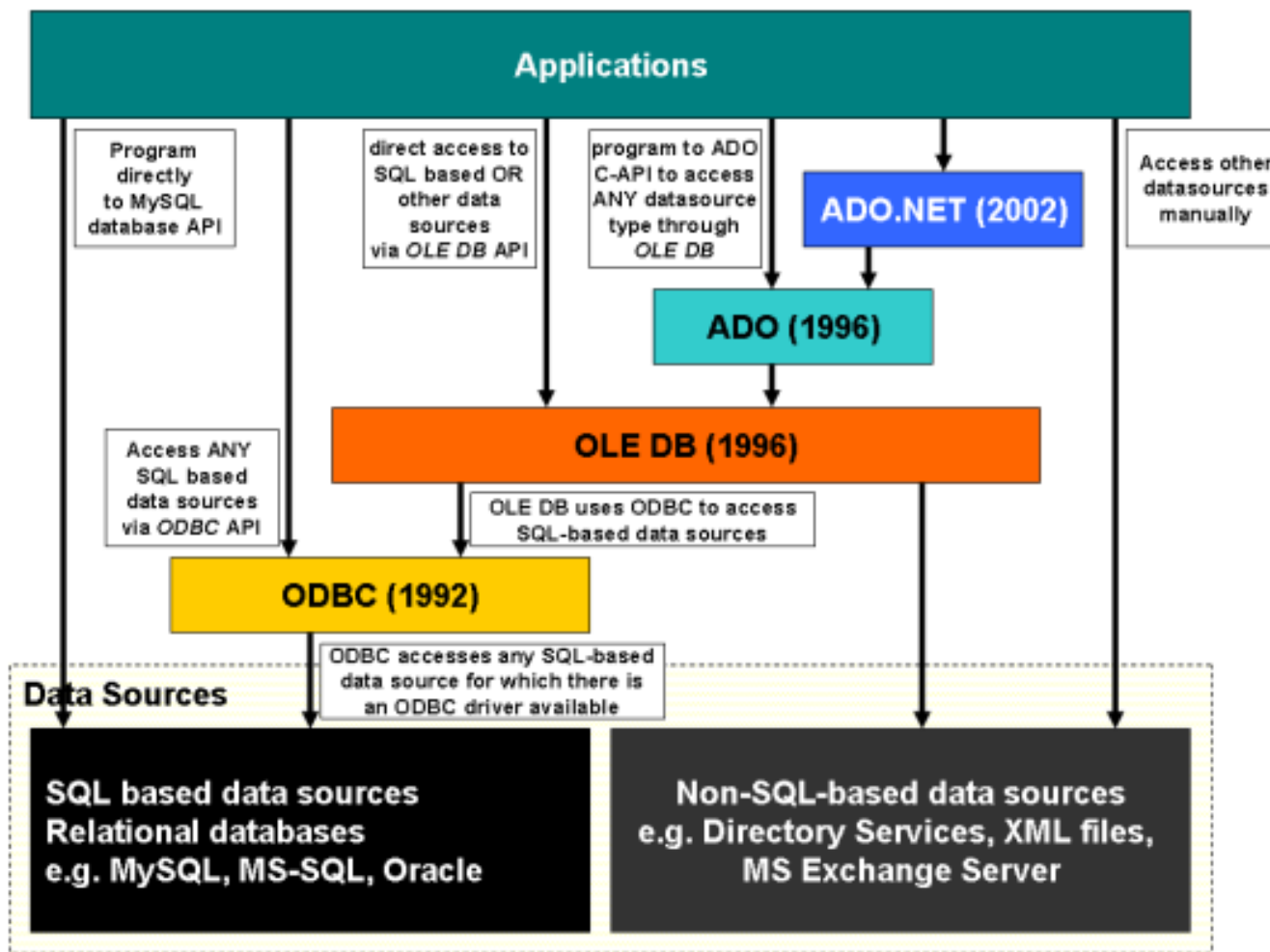


ADO.NET (*ActiveX Data Objects for .NET*)

- Produkt Microsoftu wywodzący się z ADO.
- Zbiór programów i bibliotek używanych w celu dostępu do baz danych zintegrowanych ze środowiskiem .NET.
- Framework ORM.



Zależność technologii





Pytania?

DZIĘKUJĘ ZA UWAGĘ