



# Wprowadzenie do protokołu HTTP

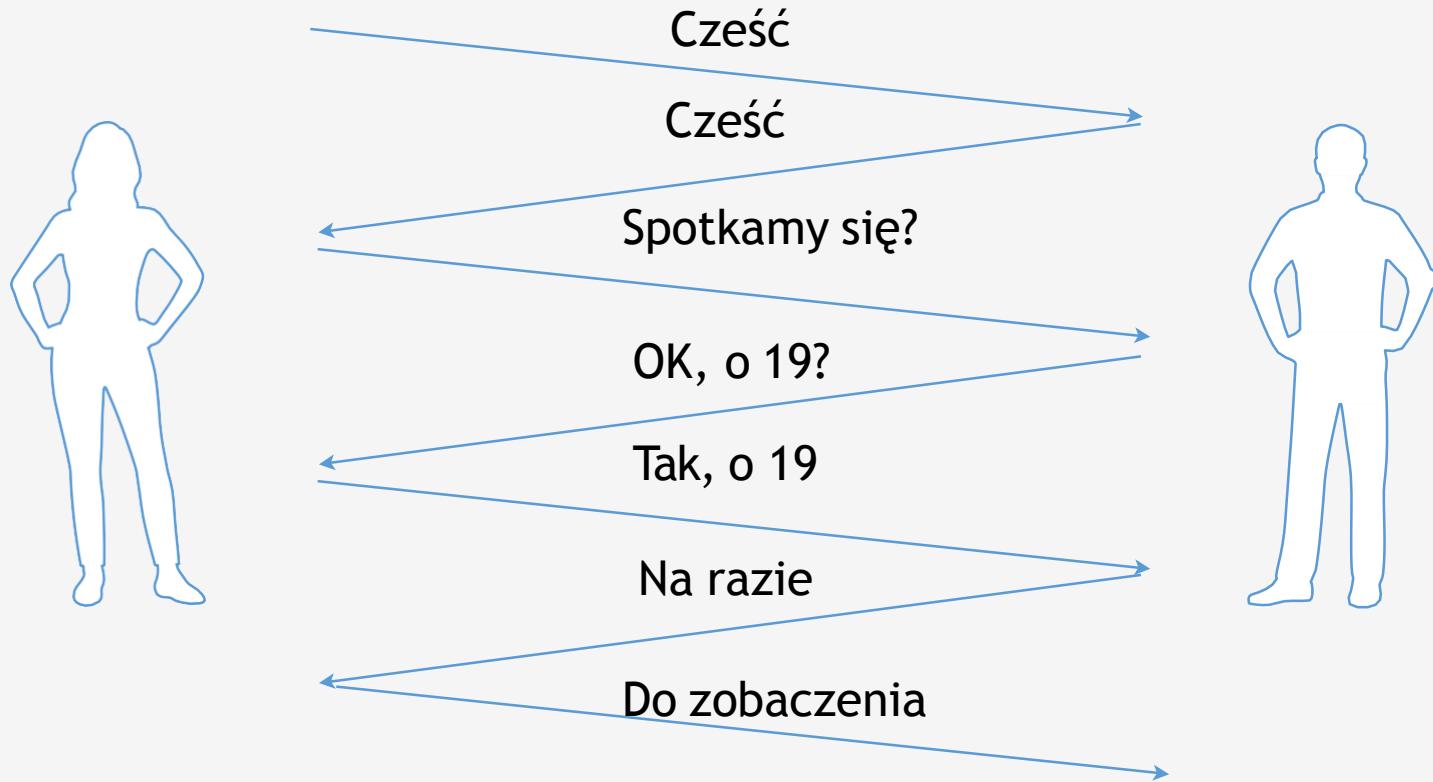
Piotr Najecki



# Agenda

- Protokół
  - Zasada działania sieci
  - DNS
  - HTTP
  - Narzędzia sieciowe
  - Webserwisy
  - Ćwiczenia

# Protokół





Artykuł [Dyskusja](#)

Czytaj [Edytuj](#) [Edytuj kod źródłowy](#) Historia i autorzy [Przeszukaj Wikipedię](#)

Nie 4 dni temu Późnie

**Protokół** [edytuj]

To jest strona ujednoznaczająca. Poniżej znajdują się różne znaczenia hasła: **Protokół**.

• W najbardziej ogólnym sensie: zestaw reguł umożliwiających porozumienie. Przykładem może być [protokół dyplomatyczny](#), wypracowany w ciągu stuleci w celu zażegnania [konfliktów](#) występujących w trakcie oficjalnych wizyt przedstawicieli obcych [państw](#) u przedstawicieli [władz](#). W tym sensie słowo to jest też używane zamiennie z [etykieta](#).

• Oficjalne, pisane na bieżąco sprawozdanie przebiegu rozmaitych posiedzeń, zebrań, [obrad](#), [wyborów](#) itp. Sprawozdania takie służą później do sprawdzania legalności decyzji podejmowanych przez protokołowane [gremia](#) i ew. stanowią materiał odwoławczy.

• Ścisła specyfikacja działań jakie podejmują urzędzenia komunikacyjne ([faksy](#), [modemy](#), [komputery](#) itp.) aby ustawić między sobą połączenie a następnie móc przekazywać dane. Zobacz [protokół komunikacyjny](#).

Zobacz hasło [protokół](#) w [Wikisłowniku](#)

Źródło: <https://pl.wikipedia.org/wiki/protokol>



# Protokół sieciowy

Zbiór zasad, procedur i reguł określających sposób komunikacji między węzłami sieciowymi.

Aby dwa węzły mogły się komunikować muszą one używać tego samego protokołu.

A czym jest węzeł sieciowy?

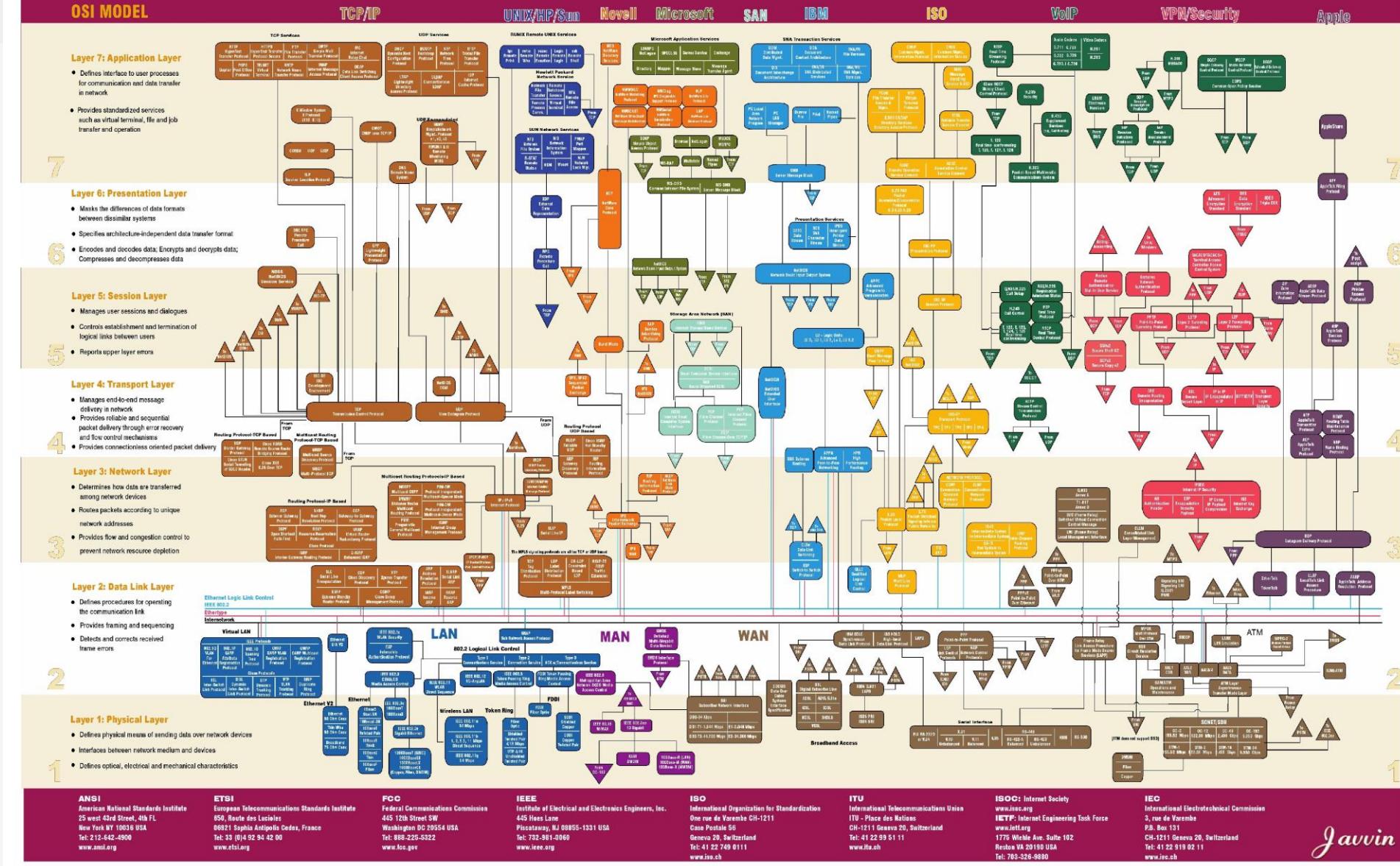


# Węzeł sieciowy

**Węzeł sieci komputerowej** – przyłączone do sieci aktywne urządzenie elektroniczne, które pozwala na wysyłanie, odbieranie i przekazywanie informacji przez [kanał komunikacyjny](#). Pasywne elementy sieci, takie jak np. [panel krosowniczy](#), nie są zaliczane do węzłów sieci komputerowej, natomiast mogą nimi być urządzenia transmisji danych takie jak [modemy](#), [koncentratory](#), [mosty](#) i [przełączniki](#) bądź urządzenia końcowe jak [telefony cyfrowe](#) i [drukarki sieciowe](#), a także [hosty](#) – tzn. [routery](#), [stacje robocze](#) lub [serwery](#).

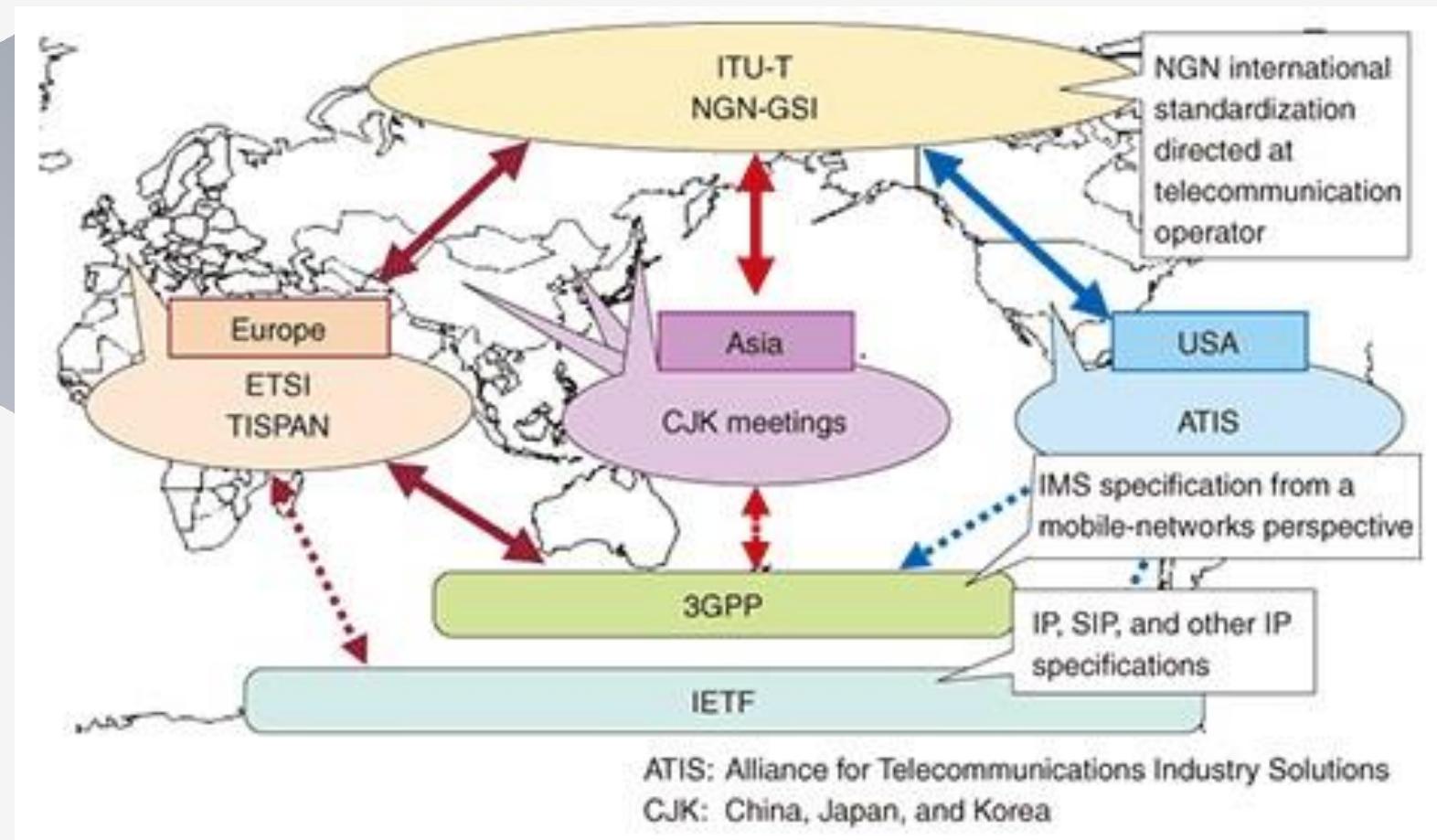


# NETWORK COMMUNICATION PROTOCOLS MAP



Źródło:<http://img.softpicks.fr/screenshots/Network-Protocols-Map-Poster.gif>

Javvin





# Sieć komputerowa

Łączy ze sobą komputery oraz inne urządzenia za pomocą kanałów komunikacyjnych oraz oprogramowania, aby: ...



# Elementy sieci

- host – komputer zapewniający użytkownikowi dostęp do zasobów sieci,
- serwer – maszyna na stałe podłączona do sieci udostępniająca swoje zasoby jej uczestnikom,
- medium transmisyjne – nośnik zapewniający kanał komunikacji pomiędzy urządzeniami działającymi w sieci (kable, światłowody, fale radiowe, satelity, ...)
- sprzęt sieciowy – urządzenia umożliwiające komunikację pomiędzy urządzeniami w sieci,
- oprogramowanie – programy zainstalowane na komputerach i urządzeniach sieciowych.



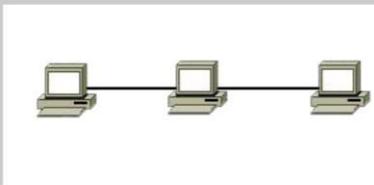
# Podział sieci ze względu na zasięg



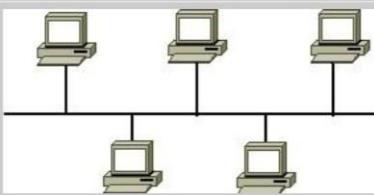
# Topologia sieci



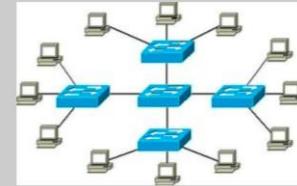
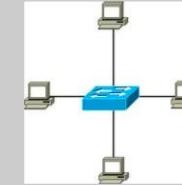
Liniowa



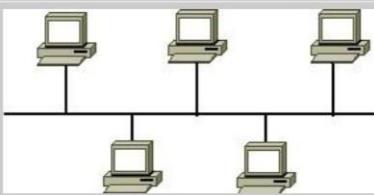
Gwiazdy



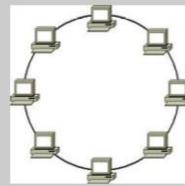
Gwiazdy  
rozszerzonej



Magistrali



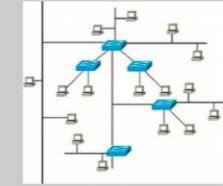
Pierścienia



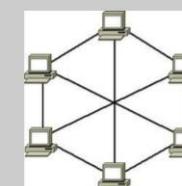
Podwójny  
pierścień



Hierarchiczna /  
Drzewa



Siatki  
(redundancja  
łączy)





# Model ISO/OSI

- *Open Systems Interconnection*
- Model sieciowy dzięki któremu producenci mogą tworzyć współpracujące ze sobą elementy
- Przyjęty przez ISO w 1984
- Przyjęty w Polsce w 1995
- Składa się z 7 warstw
- Każda warstwa posiada specyficzne dla siebie protokoły.
- Protokoły niższej warstwy świadczą usługi warstwie wyższej.



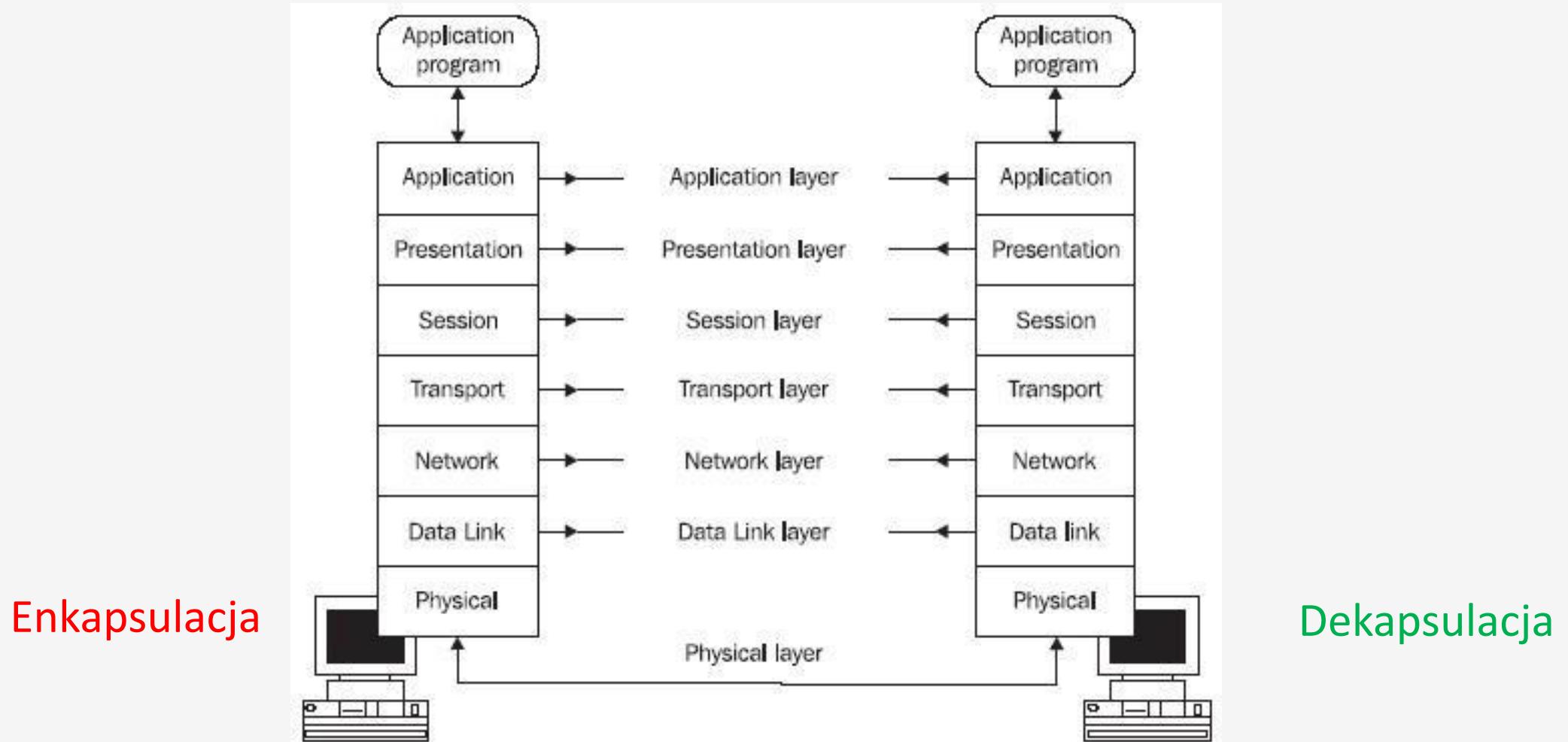
# Model ISO/OSI

ISO/OSI	TCP/IP	Niektóre protokoły Internetu
Warstwa aplikacji	Warstwa aplikacji	Telnet FTP HTTP SMTP POP
Warstwa prezentacji		DNS NFS SNMP
Warstwa sesji		RIP
Warstwa transportowa	Warstwa transportowa	TCP UDP
Warstwa sieciowa	Warstwa Internetu	IP ICMP
Warstwa łącza danych	Warstwa dostępu do sieci	ARP PPP SLIP Inne ...
Warstwa sprzętowa		CSMA/CD Ethernet

Źródło: [https://pl.wikipedia.org/wiki/Model\\_OSI](https://pl.wikipedia.org/wiki/Model_OSI)

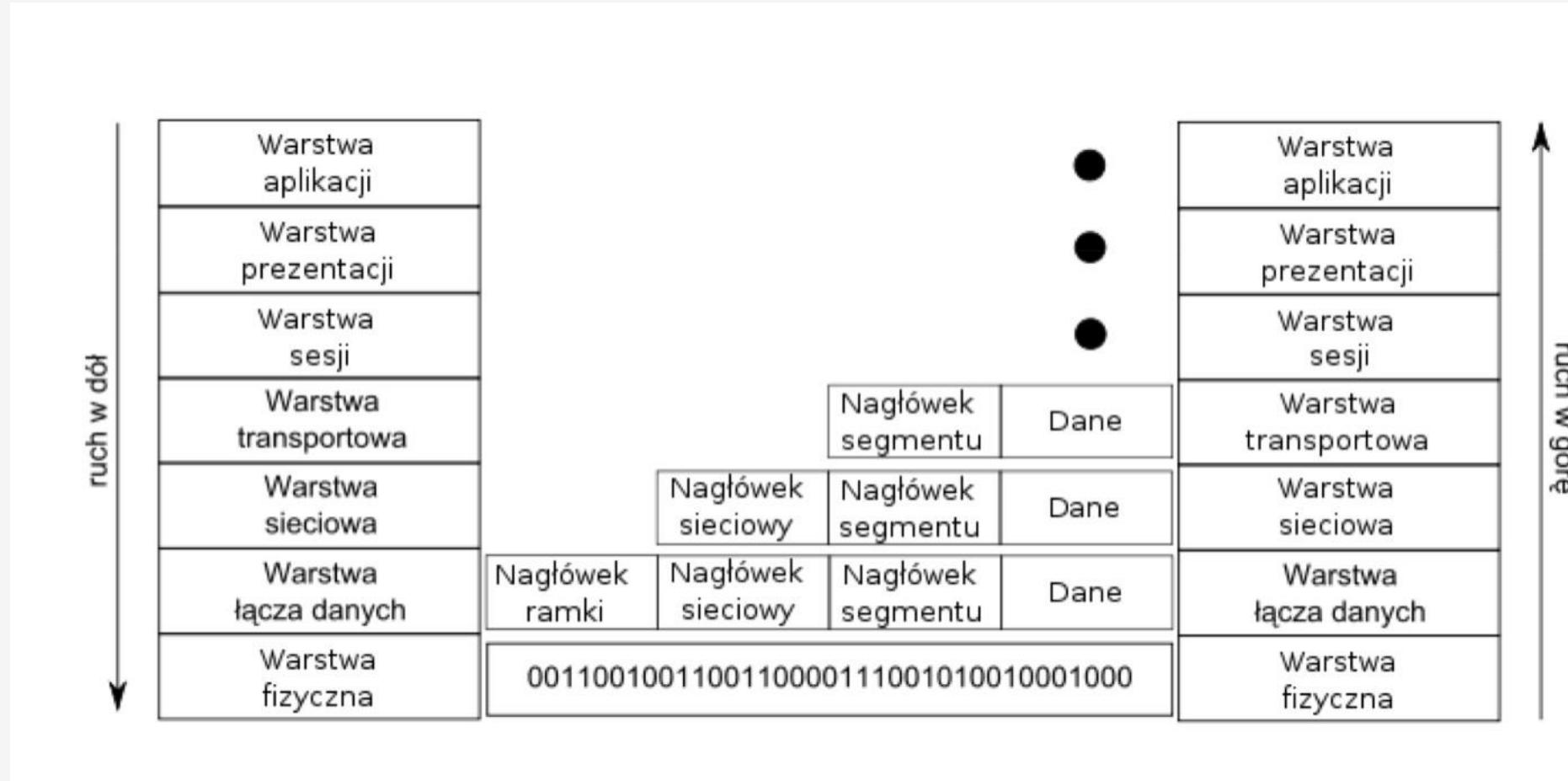
[https://en.wikipedia.org/wiki/List\\_of\\_network\\_protocols\\_\(OSI\\_model\)](https://en.wikipedia.org/wiki/List_of_network_protocols_(OSI_model))

# Przesyłanie danych między węzłami





# Przesyłanie danych między warstwami





# Warstwa fizyczna

Określa ona wszystkie składniki sieci niezbędne do obsługi elektrycznego, optycznego, radiowego wysyłania i odbierania sygnałów

- Warstwa odpowiada za fizyczne przesłanie ramki przez medium transmisyjne (np. miedziany kabel lub światłowód).
- Nie posiada mechanizmów weryfikacji poprawności przesyłanych danych.
- Określa ona wszystkie składniki sieci niezbędne do obsługi elektrycznego, optycznego, radiowego, itd. wysyłania i odbierania sygnałów.
- Urządzenia działające w tej warstwie: karty sieciowe, repeater'y, hub'y, modemy.

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Warstwa łączy danych

- Nadzoruje jakość przekazywanych danych przez warstwę fizyczną
- Ma możliwość korekcji błędów występujących w warstwie 1.
- Odpowiada za pakowanie danych w ramki przekazywane do warstwy 1.
- Urządzenia działające w tej warstwie: most (bridge), przełącznik (switch).

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Warstwa sieciowa

- Warstwa sieciowa jako jedyna dysponuje wiedzą dotyczącą fizycznej **topologii sieci**.
- Rozpoznaje, jakie drogi łączą poszczególne komputery (**trasowanie**) i decyduje, ile informacji należy przesyłać jednym z połączeń, a ile innym.
- Jeżeli danych do przesyłania jest zbyt wiele, to warstwa sieciowa po prostu je ignoruje.
- Nie musi zapewniać pewności transmisji, więc w razie błędu pomija niepoprawne pakiety danych.

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Warstwa transportowa

- Warstwa transportowa segmentuje dane oraz składa je w tzw. strumień.
- Warstwa ta zapewnia całościowe połączenie między stacjami: źródłową oraz docelową, które obejmuje całą drogę transmisji.
- Następuje tutaj podział danych na części, które są kolejno indeksowane i wysyłane do docelowej stacji.
- Na poziomie tej warstwy do transmisji danych wykorzystuje się dwa protokoły **TCP** (ang. *Transmission Control Protocol*) oraz **UDP** (ang. *User Datagram Protocol*).

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Warstwa sesji

- Warstwa sesji otrzymuje od różnych aplikacji dane, które muszą zostać odpowiednio zsynchronizowane.
- Synchronizacja występuje między warstwami sesji systemu nadawcy i odbiorcy.
- Warstwa sesji „wie”, która aplikacja łączy się z którą, dzięki czemu może zapewnić właściwy kierunek przepływu danych – nadzoruje połączenie.
- Wznawia połączenie po przerwaniu.

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Warstwa prezentacji

- Warstwa odpowiada za konwersję danych otrzymanych z aplikacji do postaci kanonicznej, zgodnej ze specyfikacją OSI Reference Model.
- Dzięki niej niższe warstwy otrzymują dane w spójnym formacie możliwym do przesłania przez sieć.
- W tej warstwie odbywa się np.: kodowanie obrazu (MPEG, JPG, GIF itd.), szyfrowanie/deszyfrowanie danych, kompresja/dekompresja.

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Warstwa aplikacji

- Warstwa aplikacji jest warstwą najwyższą, zajmuje się specyfikacją interfejsu, który wykorzystują aplikacje do przesyłania danych do sieci (poprzez kolejne warstwy modelu ISO/OSI).
- W przypadku sieci komputerowych aplikacje są zwykle **procesami** uruchomionymi na odległych **hostach**.
- Interfejs udostępniający programistom usługi dostarczane przez warstwę aplikacji opiera się na obiektach nazywanych **gniazdami** (ang. *socket*).

Warstwa aplikacji

Warstwa prezentacji

Warstwa sesji

Warstwa transportowa

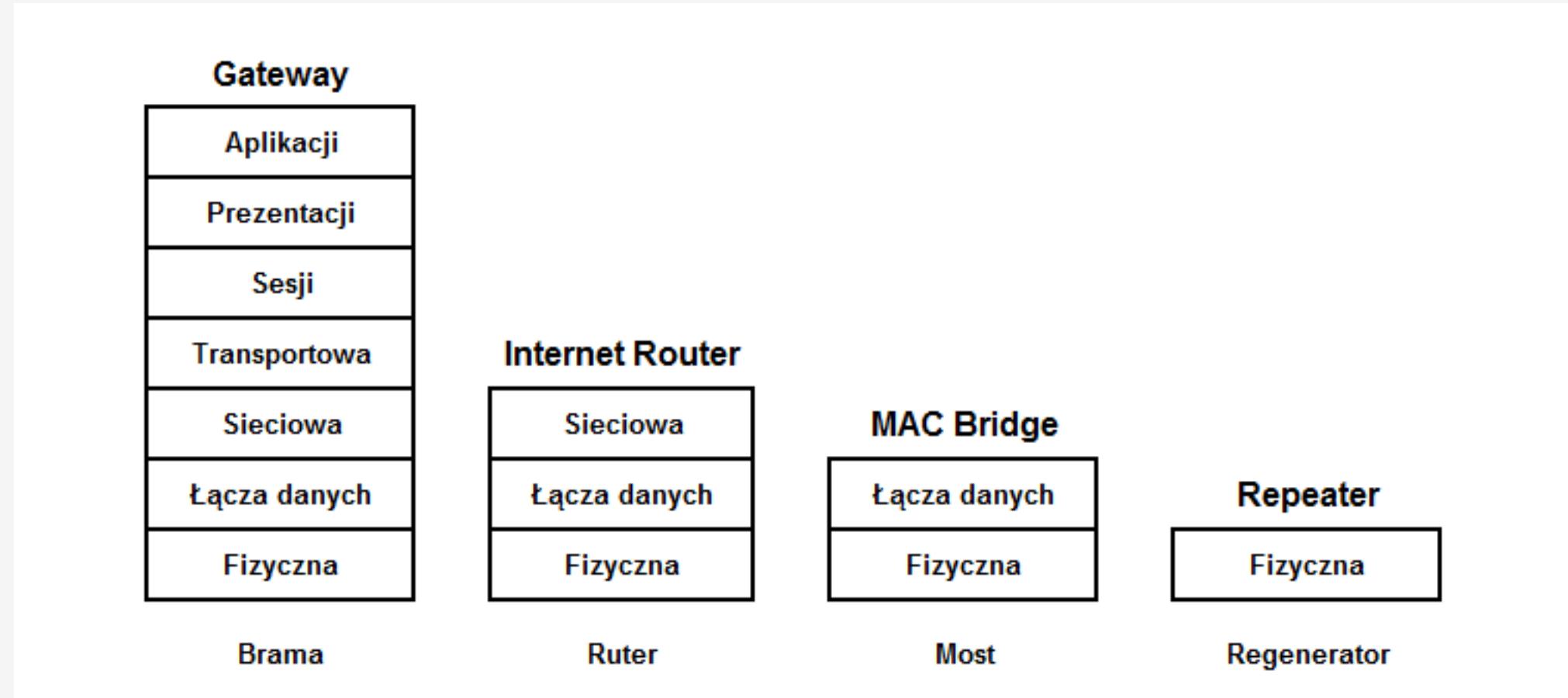
Warstwa sieciowa

Warstwa łączy danych

Warstwa fizyczna



# Urządzenia sieciowe





# Model TCP/IP

- Uproszczony, w stosunku do modelu OSI, model do opisu komunikacji w ramach internetu
- Nie należy mylić Modelu OSI-RM z TCP/IP. Oba te modele nie są w pełni zgodne.
- Początek standaryzacji w latach 70
- Podział na 4 warstwy
- Współistnieje z modelem OSI

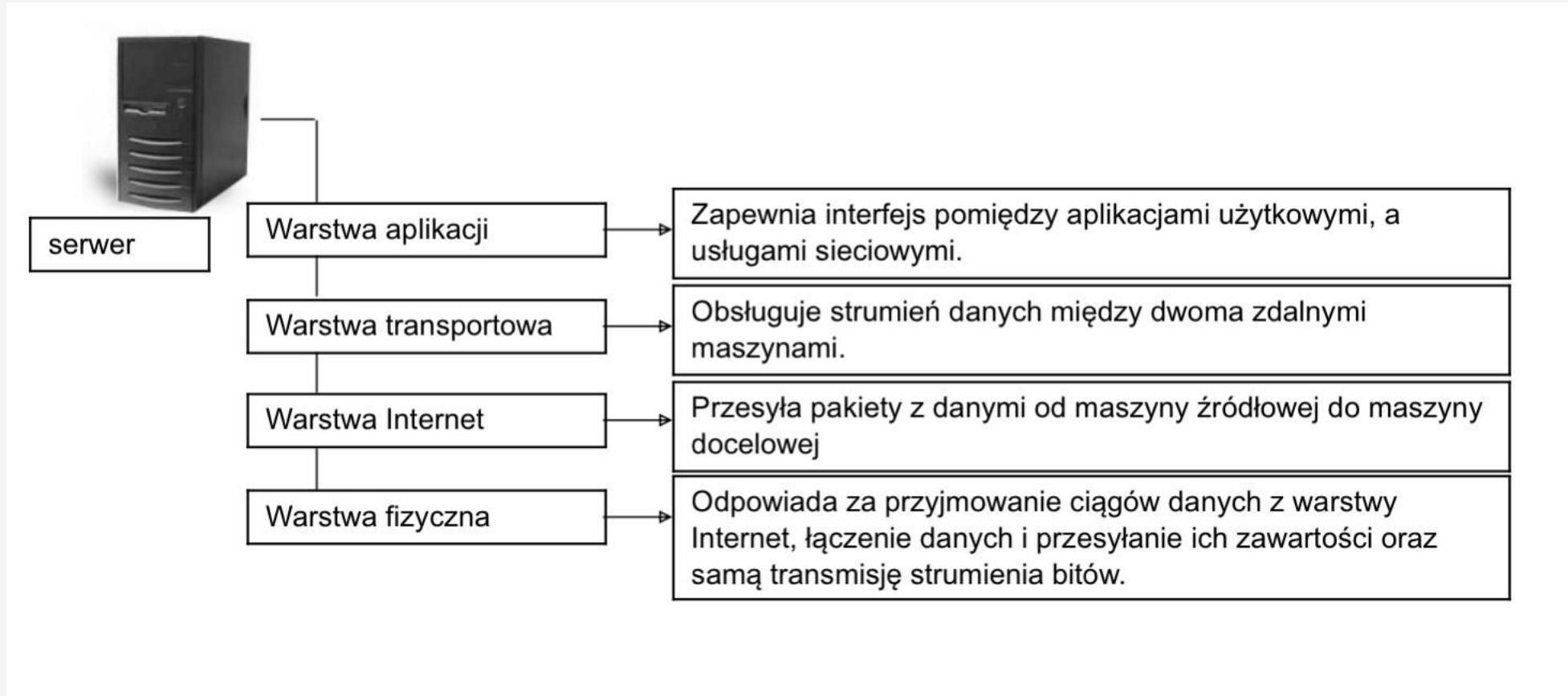


# Model TCP/IP

ISO/OSI	TCP/IP	Niektóre protokły Internetu		
Warstwa aplikacji	Warstwa aplikacji	Telnet	DNS	RIP
Warstwa prezentacji		FTP	NFS	
Warstwa sesji		HTTP	SNMP	
Warstwa transportowa	Warstwa transportowa	SMTP		
Warstwa sieciowa	Warstwa Internetu	POP		
Warstwa łącza danych	Warstwa dostępu do sieci	TCP	UDP	
Warstwa sprzętowa		IP		ICMP
		ARP	PPP	Inne ...
		CSMA/CD Ethernet	SLIP	



# Model TCP/IP





# Model TCP/IP

ISO/OSI	TCP/IP	Niektóre protokły Internetu		
Warstwa aplikacji	Warstwa aplikacji	Telnet	DNS	RIP
Warstwa prezentacji		FTP	NFS	
Warstwa sesji		HTTP	SNMP	
Warstwa transportowa	Warstwa transportowa	SMTP		
Warstwa sieciowa	Warstwa Internetu	POP		
Warstwa łącza danych	Warstwa dostępu do sieci	TCP	UDP	
Warstwa sprzętowa		IP		ICMP
		ARP	PPP	Inne ...
		CSMA/CD Ethernet	SLIP	

Źródło: [https://pl.wikipedia.org/wiki/Model\\_OSI](https://pl.wikipedia.org/wiki/Model_OSI)



# IP (Internet Protocol)

Co robi:

- zarządza odnajdywaniem urządzeń sieciowych oraz przesyłaniem danych z punktu do punktu

Cechy

- dobra odtwarzalność po awarii
- wysoki współczynnik korekcji błędów
- mały narzut na rozmiar pakietów
- nie ma potrzeby restartu sieci po dodaniu nowych elementów



Co robi:

- uzgadnianie tożsamości
- zarządza pakietami
- steruje przepływem pakietów
- wykrywanie i obsługa błędów

Cechy

- wysoki współczynnik korekcji błędów
- kontrola utrzymywania połączenia mimo braku świadomej przesyłanych danych



# UDP (User Datagram Protocol)

Co robi:

- warstwa transportowa
- fire and forget ;)

Cechy

- nie gwarantuje dostarczenia
- nie gwarantuje kolejnosci
- nie wymaga nawiazywania połączenia (tylko fizyczne)



# TCP vs UDP

Protokoły TCP oraz UDP służą do komunikacji między procesami na zdalnych maszynach

## **Protokół UDP (User Datagram Protocol):**

- Nie kontroluje poprawności i prędkości przesyłu. Minimalizuje przesył dodatkowych danych
- Gdy zależy nam na maksymalnej wydajności i tolerujemy utratę danych
- Odpowiednie dla: strumieniowe multimedia, telefonia VOIP, routing, DNS, sieci P2P

## **Protokół TCP (Transfer Control Protocol)**

- Niezawodne przesyłanie danych (strumień). Wykorzystuje potwierdzenie z retransmisją (wysyła dane aż do otrzymania potwierdzenia)
- Gdy zależy nam na pewności połączenia i nie tolerujemy utraty danych
- Odpowiednie dla: poczta, zdalne logowanie, WWW, transfer plików



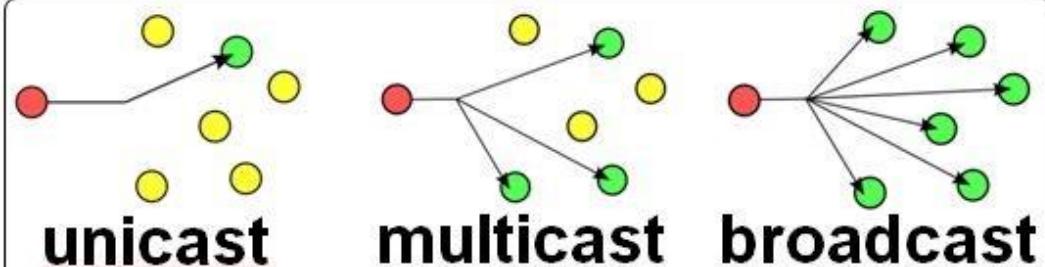
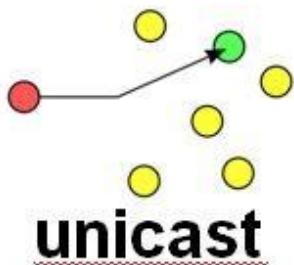
TCP

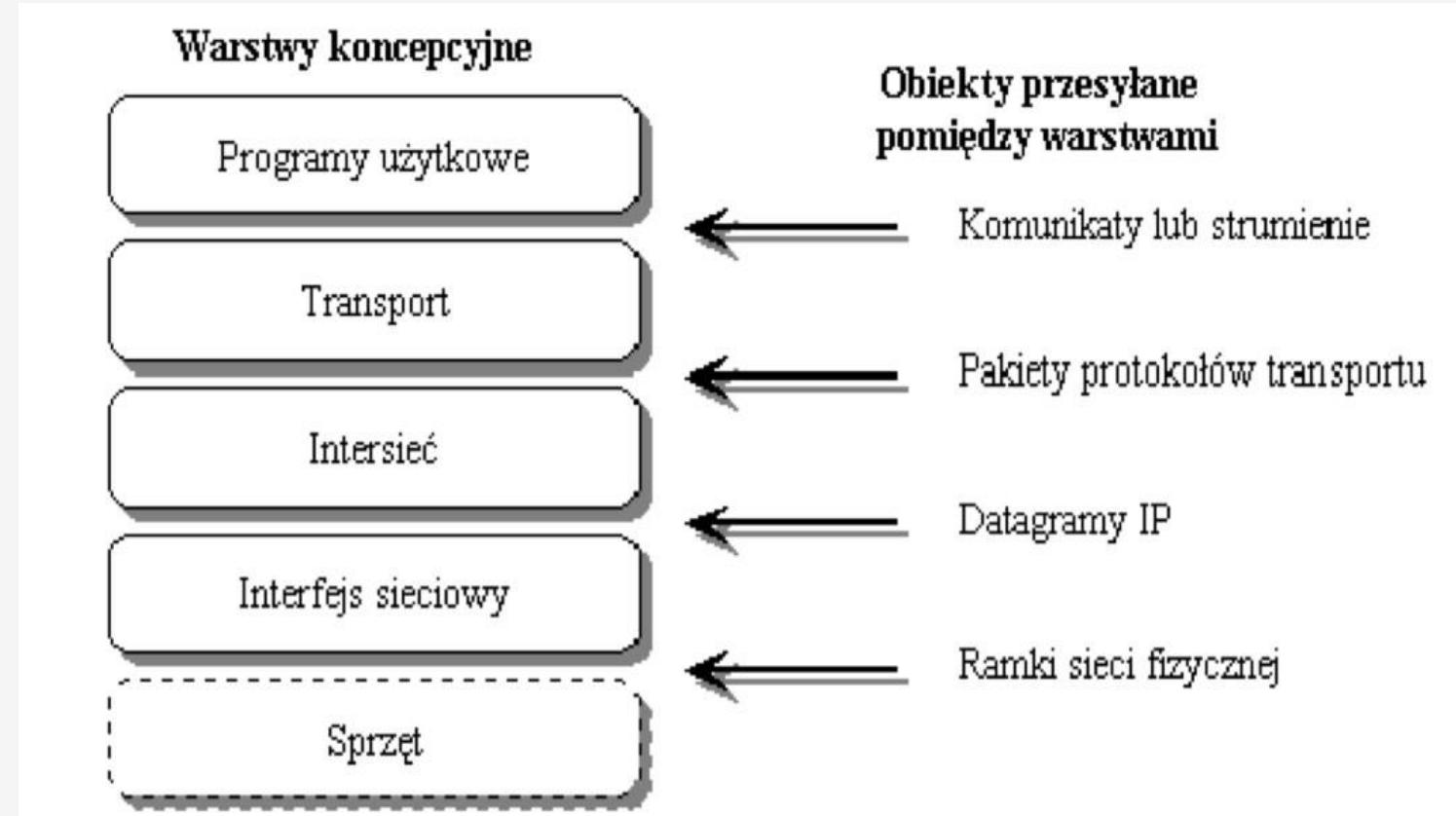


UDP

- **Slower but reliable transfers**
- **Typical applications:**
  - Email
  - Web browsing

- **Fast but non-guaranteed transfers (“best effort”)**
- **Typical applications:**
  - VoIP
  - Music streaming





# TCP/IP = współczesny internet



- dobra odtwarzalność po awarii,
- możliwość dodawania nowych elementów sieci bez przerywania pracy istniejących, wysoki współczynnik korekcji błędów
- niezależność od platformy
- mały stopień obciążenia danych własnymi strukturami
- duża wydajność.



# Adresacja IPv4

- Adres IP jest adresem logicznym interfejsu sieciowego.
- Adres IP to 32-bitowy ciąg zer i jedynek ale dla wygody przedstawiamy go w formie dziesiętnej jako 4 liczby (oktety), każda wielkości 1 bajta, np.:
  - 192.112.20.101 11000000.01110000.00010100.01100101
- Pewna część bitów od lewej strony określa adres sieci. Reszta to adres host'a (urządzenia) w danej sieci.
- Podział ten określa maska sieci. Definiuje wielkość sieci oraz możliwą do zaadresowania ilość hostów.
- Dla każdej sieci definiujemy 3 adresy:
  - adres sieciowy – określa całą sieć,
  - adres rozgłoszeniowy – używany w celu wysłania danych do wszystkich hostów w danej sieci,
  - adres host'a – adres urządzenia z danej sieci.



# Adresacja IPv4

	DZIESIĘTNIE	BINARNE
adres IP	<b>195.225.71.72</b>	11000011.11100001.01000111.0   1001000
maska	255.255.255.128	11111111.11111111.11111111.1   0000000
adres sieciowy (AND)	195.225.71.0/25	11000011.11100001.01000111.0   0000000
adres rozgłoszeniowy (NOT maski + adres sieci)	195.225.71.127	11000011.11100001.01000111.0   1111111
pierwszy host	195.225.71.1	11000011.11100001.01000111.0   0000001
ostatni host	195.225.71.126	11000011.11100001.01000111.0   1111110



# Adresacja IPv4 binarnie

	128	64	32	16	8	4	2	1
200	1	1	0	0	1	0	0	0
17	0	0	0	1	0	0	0	1
100	0	1	1	0	0	1	0	0
3	0	0	0	0	0	0	1	I

200.17.100.3

# Operacja AND, OR & NOT



A	B	C
---	---	---

**0 AND 0 -> 0**

**0 AND 1 -> 0**

**1 AND 0 -> 0**

**1 AND 1 -> 1**

A	B	C
---	---	---

**0 OR 0 -> 0**

**0 OR 1 -> 1**

**1 OR 0 -> 1**

**1 OR 1 -> 1**

A	B
---	---

**NOT 0 -> 1**

**NOT 1 -> 0**



# Obliczanie adresu (identyfikatora) sieci

Maska	11111111	11111111	11111111	00000000
	255	255	255	0
Mój IP	11000000	10101000	00000001	00000011
	192	168	1	3
Wynik a	11000000	10101000	00000001	00000000
	192	168	1	0
Maska	11111111	11111111	11111111	00000000
	255	255	255	0
Docelowy IP	11000000	10101000	00000001	00000010
	192	168	1	2
Wynik b	11000000	10101000	00000001	00000000
	192	168	1	0



# Obliczanie adresu rozgłoszeniowego sieci

Adres IP: 212.51.219.32

Maska: 255.255.255.192

Adres IP binarnie: 11010100.00110011.11011011.00100000

Maska binarnie: 11111111.11111111.11111111.11000000

!Maska binarnie: 00000000.00000000.00000000.00111111

Adres broadcast = Adres IP binarnie OR !Maska binarnie:

1	1	0	1	0	1	00.	0	0	1	1	00	.	1	1	0	1	1	01	11.	0	0	1	0	0000
0	0	0	0	0	0	00	0	0	0	0	0000	.	0	0	0	0	0000	00	.	0	0	1	1	1111

---

1	1	0	1	0	1	00.	0	0	1	1	0011,	1	1	0	1	1	011.	0	0	1	1	1111
---	---	---	---	---	---	-----	---	---	---	---	-------	---	---	---	---	---	------	---	---	---	---	------

Adres broadcast:

212.51.219.63



# Adres IPv4 ćwiczenie

Dla podanych adresów IP i masek podsieci wyznacz identyfikatory sieci:

1. 185.158.109.145    255.255.255.0
  
2. 191.15.56.194                          255.255.224.0
  
3. 157.83.85.231                          255.255.255.192
  
4. 173.124.19.182                          255.255.192.0
  
5. 152.173.243.72                          255.255.240.0



# Adresacja IPv6

- Wielkość adresu IPv6 wynosi 128 bitów.
- Zapis w systemie szesnastkowym
- Adresy IPv6 są z zakresu od  
0000:0000:0000:0000:0000:0000:0000:0000 do ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff.
- W adresacji



# IPv4 vs IPv6

- Przy użyciu protokołu IPv4 można utworzyć **4,294,967,296** adresów
- Przy użyciu protokołu IPv6 można utworzyć  
**340,282,366,920,938,463,463,374,607,431,768,211,456** adresów.



# Adresy IPv4

- Publiczne
- Prywatne
- Localhost



# Adres IP vs adres domenowy

google.pl:

- Adres IP to 216.58.209.163 (,numer telefonu')
- Adres domenowy to google.pl (,nazwa kontaktu')



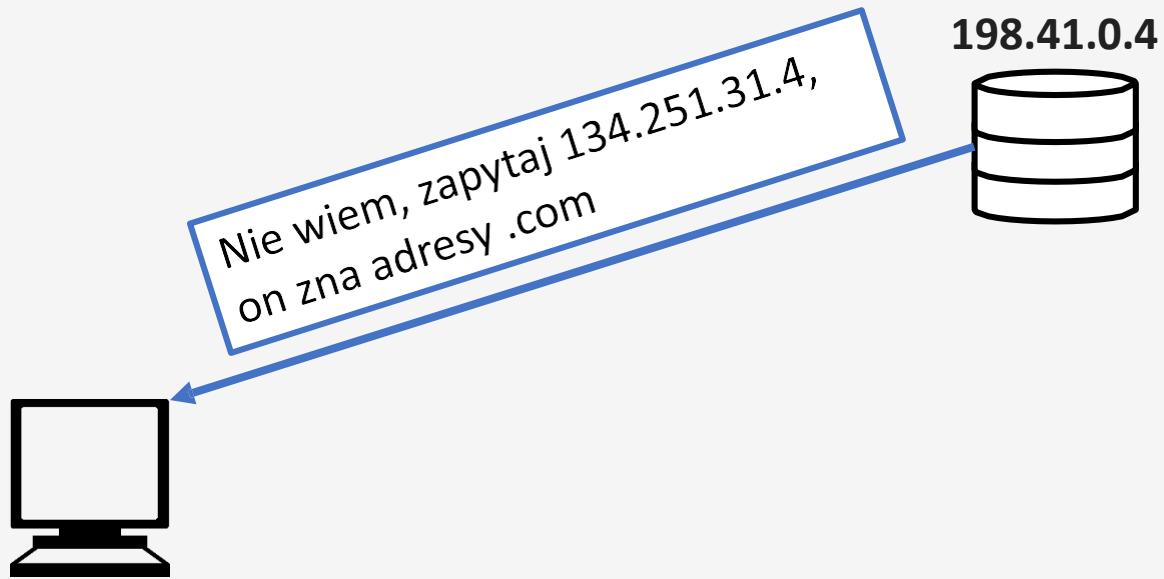
- Domain Name System
- Tłumaczy adresy WWW na adresy IP
- Oparty na 13 głównych serwerach



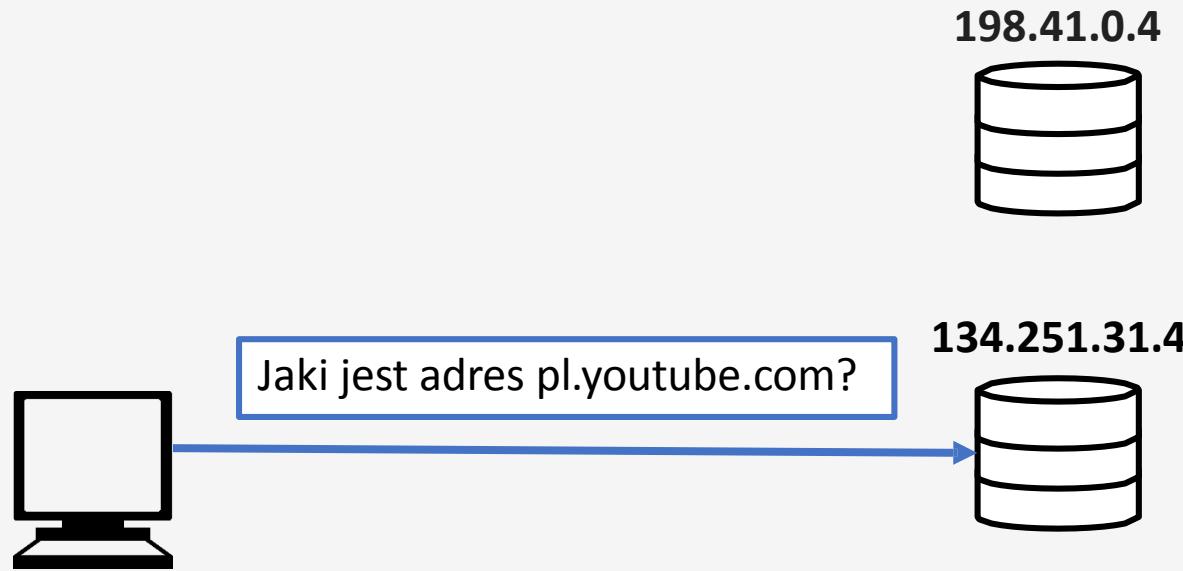
# DNS



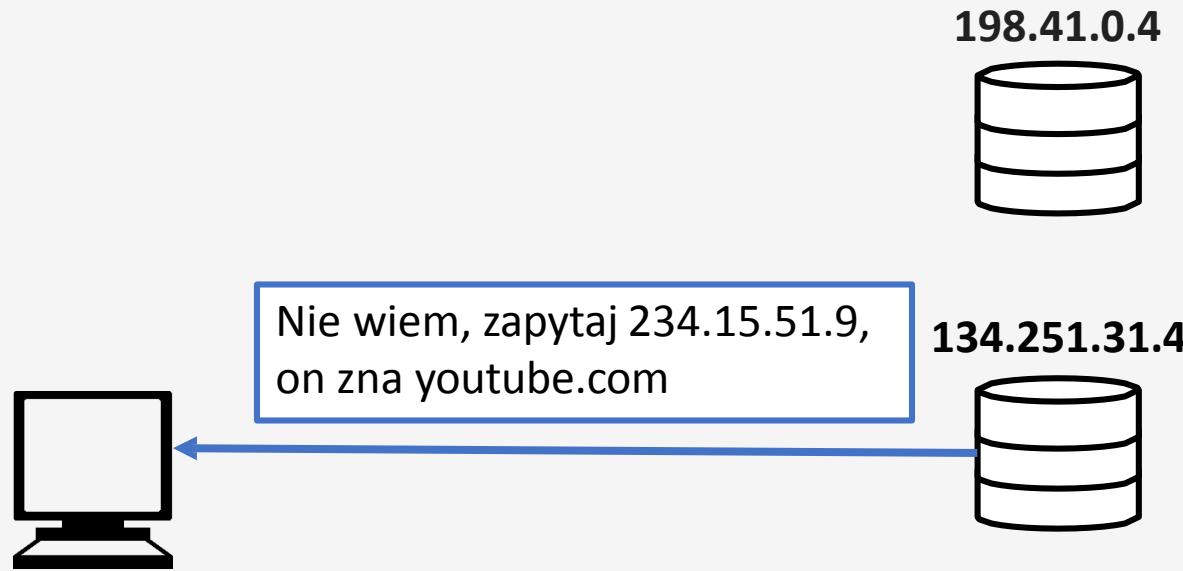
# DNS



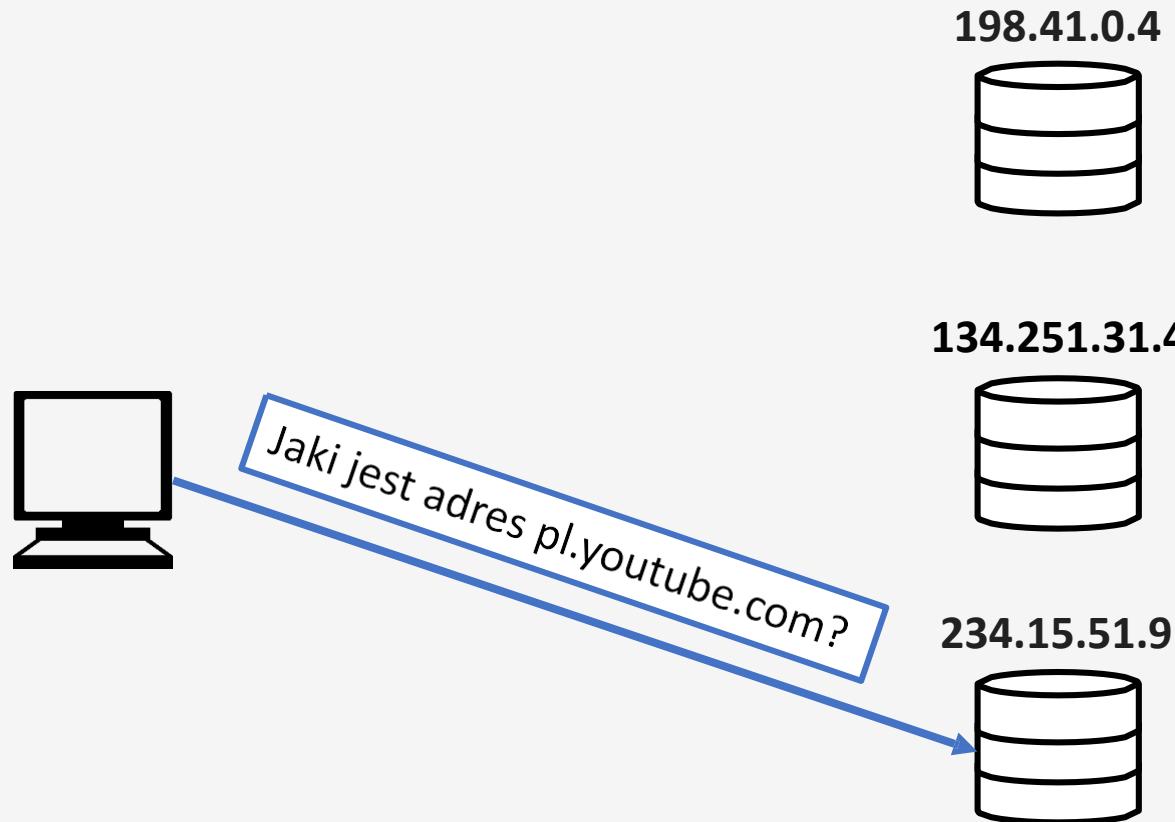
# DNS



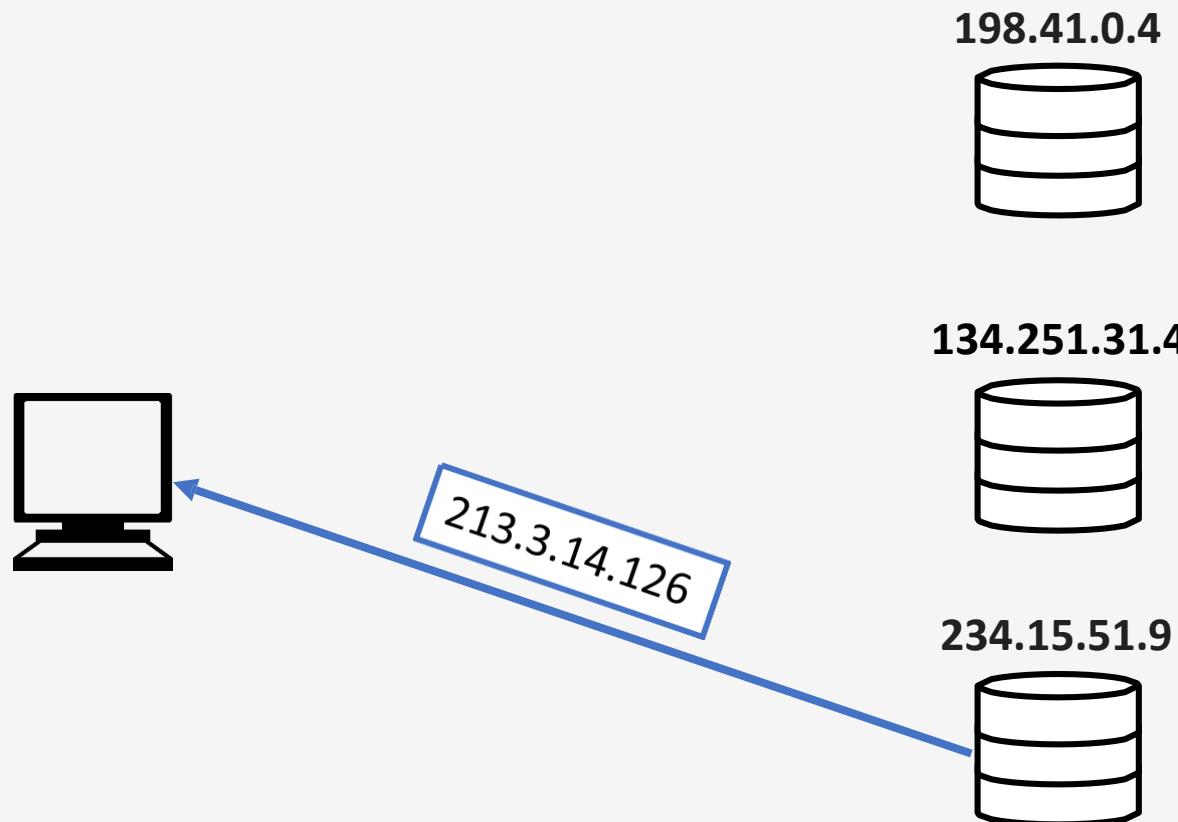
# DNS



# DNS



# DNS





- Spójność danych
- Skalowalność
- Odporność
- Wydajność
- Decentralizacja (danych i zarządzania)



# Narzędzia diagnostyczne

## PING

- Służy do sprawdzania połączenia do zdalnego hosta
- Odpytuje serwery DNS – służy często do sprawdzania numeru IP
- Mierzy czasy oczekiwania na odpowiedź (lagi)
- Wykonajmy:
- ***ping sdacademy.pl***
- ***ping wp.pl***

## TRACEROUTE (TRACERT)

- Służy do sprawdzania trasy jaką przemierzają pakiety w drodze do celu
- Może służyć do diagnozowania problemów pomiędzy konkretnymi przeskokami
- Sprawdźmy przez jakie serwery łączymy się ze stroną [google.com](http://google.com):
  - ***traceroute google.com***
  - Wyniki wklejamy na stronę, która przedstawi je w postaci mapy:

# Wireshark



- Umożliwia przechwytywanie oraz rejestrowanie ruchu na interfejsach sieciowych.
- Liczne dodatki pozwalają dekodować różne protokoły komunikacyjne.

The screenshot shows the Wireshark interface with a list of captured network frames. Frame 76 is selected, showing its details and structure. The packet list pane shows several HTTP requests and responses between 192.168.3.103 and 91.208.189.175. The details pane provides a breakdown of the selected frame, including its bytes on wire and captured, protocol stack, and specific fields like source and destination addresses, port numbers, and payload. The bytes pane shows the raw hex and ASCII data of the selected frame. The bottom pane displays the expanded HTTP request, revealing the GET / HTTP/1.1\r\n header and the Mozilla/5.0 User-Agent string.

No.	Time	Source	Destination	Protocol	Length	Info
76	1.615516	192.168.3.103	91.208.189.175	HTTP	978	GET / HTTP/1.1
126	1.658903	91.208.189.175	192.168.3.103	HTTP	329	HTTP/1.1 200 OK (text/html)
158	2.196719	193.200.227.13	192.168.3.103	HTTP	253	HTTP/1.0 400 Bad request (text/html)
165	2.304147	192.168.3.103	91.208.189.175	HTTP	957	GET /js/routing?callback=fos.Router.setData HTTP/1.1
168	2.323592	91.208.189.175	192.168.3.103	HTTP	732	HTTP/1.1 200 OK (application/javascript)
381	3.689382	192.168.3.103	89.25.223.83	HTTP	1221	GET /api/r.gif?uri=%2F&location=www.focus.pl&uuid=160fbc24d
388	3.711002	89.25.223.83	192.168.3.103	HTTP	344	HTTP/1.1 200 OK (GIF89a)
390	3.711234	192.168.3.103	89.25.223.83	HTTP	401	GET /api/vs?location=www.focus.pl&referrer= HTTP/1.1
391	3.711265	192.168.3.103	89.25.223.83	HTTP	202	GET /api/vs?location=www.focus.pl&referrer= HTTP/1.1

Frame 76: 978 bytes on wire (7824 bits), 978 bytes captured (7824 bits) on interface 0  
Ethernet II, Src: Apple\_28:1d:40 (80:e6:50:28:1d:40), Dst: Tp-LinkT\_46:d3:06 (c0:4a:00:46:d3:06)  
Internet Protocol Version 4, Src: 192.168.3.103, Dst: 91.208.189.175  
Transmission Control Protocol, Src Port: 52519, Dst Port: 80, Seq: 1, Ack: 1, Len: 912

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]  
Request Method: GET  
Request URI: /  
Request Version: HTTP/1.1  
Host: www.focus.pl\r\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:57.0) Gecko/20100101 Firefox/57.0\r\n

## CURL I WGET

Narzędzia do odpytywania sieci

Obsługują różne protokoły

Wymagają zainstalowania na Windows

Przydatne przy różnych skryptach oraz automatyzacji





## CURL I WGET

### cURL

1996/1997

Biblioteka (ma własne API)

Wspiera kilkanaście protokołów

Mnóstwo parametrów i opcji

### Wget

1995

Program (wywoływany za pomocą komendy)

HTTP, HTTPS, FTP

Proste narzędzie



# Curl

```
Piotr@seqr_asus ~
$ curl -I https://www.o2.pl
HTTP/2 405
server: nginx
date: Sun, 14 Jan 2018 23:50:52 GMT
content-type: text/plain; charset=utf-8
content-length: 23
set-cookie: SID=883863; path=/
```

Pobieranie pliku: *curl http://.../file.txt -o plik.txt*

```
Piotr@seqr_asus ~
$ curl http://mirror.onet.pl/pub/mirrors/fedora/linux/releases/27/Workstation/x8
6_64/iso/Fedora-Workstation-netinst-x86_64-27-1.6.iso -o fed.iso
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload Total   Spent    Left Speed
 7  508M    7 39.6M    0      0  2137k      0  0:04:03  0:00:19  0:03:44 1837k
```



# Wget

```
[ ~/tmp$ wget https://wp.pl
--2018-01-15 14:11:04--  https://wp.pl/
Resolving wp.pl (wp.pl)... 212.77.98.9
Connecting to wp.pl (wp.pl)|212.77.98.9|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.wp.pl/ [following]
--2018-01-15 14:11:05--  https://www.wp.pl/
Resolving www.wp.pl (www.wp.pl)... 212.77.98.9
Connecting to www.wp.pl (www.wp.pl)|212.77.98.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Cookie coming from www.wp.pl attempted to set domain to wp.pl
Length: unspecified [text/html]
Saving to: 'index.html'

index.html          [ <=>                               ]  74.90K  464KB/s  in 0.2s

2018-01-15 14:11:05 (464 KB/s) - 'index.html' saved [237505]
```



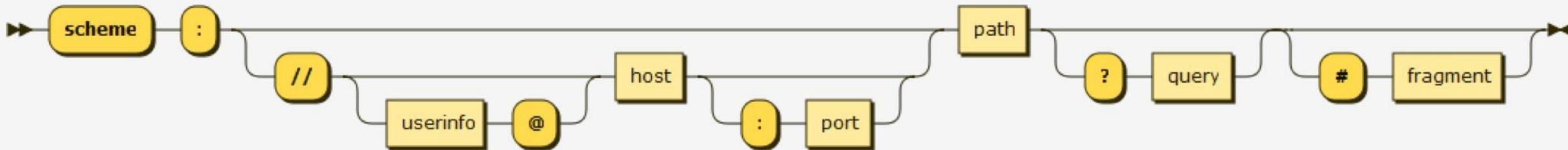
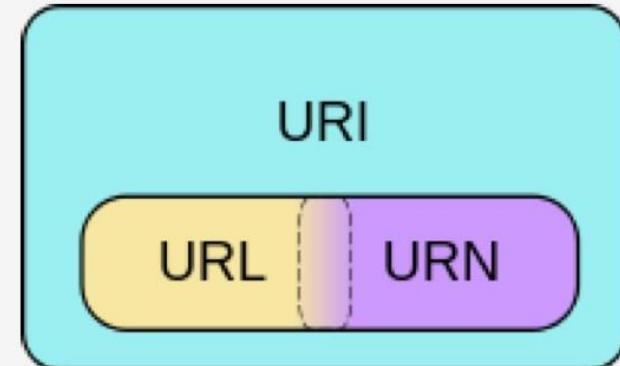
# Dodatkowe polecenia diagnostyczne

- Ipconfig
- telnet
- Tcpdump



# URI

**URI - Uniform Resource Identifier** - standard sieciowy definiujący sposób metodę zapisu identyfikatorów umożliwiających łatwą i jednoznaczna identyfikację zasobów.



np: [http://sda-http.herokuapp.com/add\\_names](http://sda-http.herokuapp.com/add_names)  
<sip:joe.bloggs@212.123.1.213>

tel:7042;phone-context=example.com

Źródło: [11]



# URL

**URL - Unified Resource Locator** - Ujednolicony format adresowania zasobów umożliwiający łatwą i jednoznaczną lokalizację zasobu. Rodzaj URI (najczęściej z URL utoższamiane jest URI)

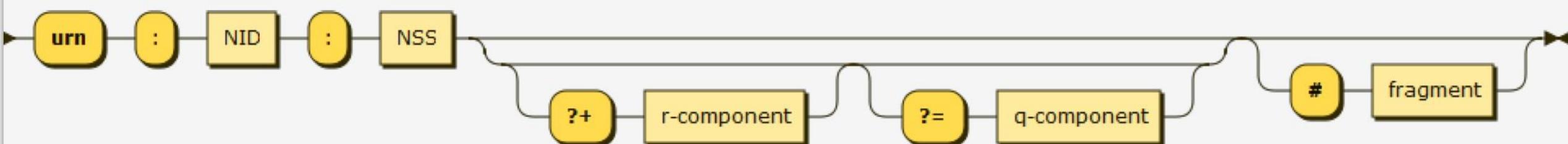
np: <https://pl.wikipedia.org:443/wiki/>

[Uniform\\_Resource\\_Identifier#URI\\_a\\_URL](#)

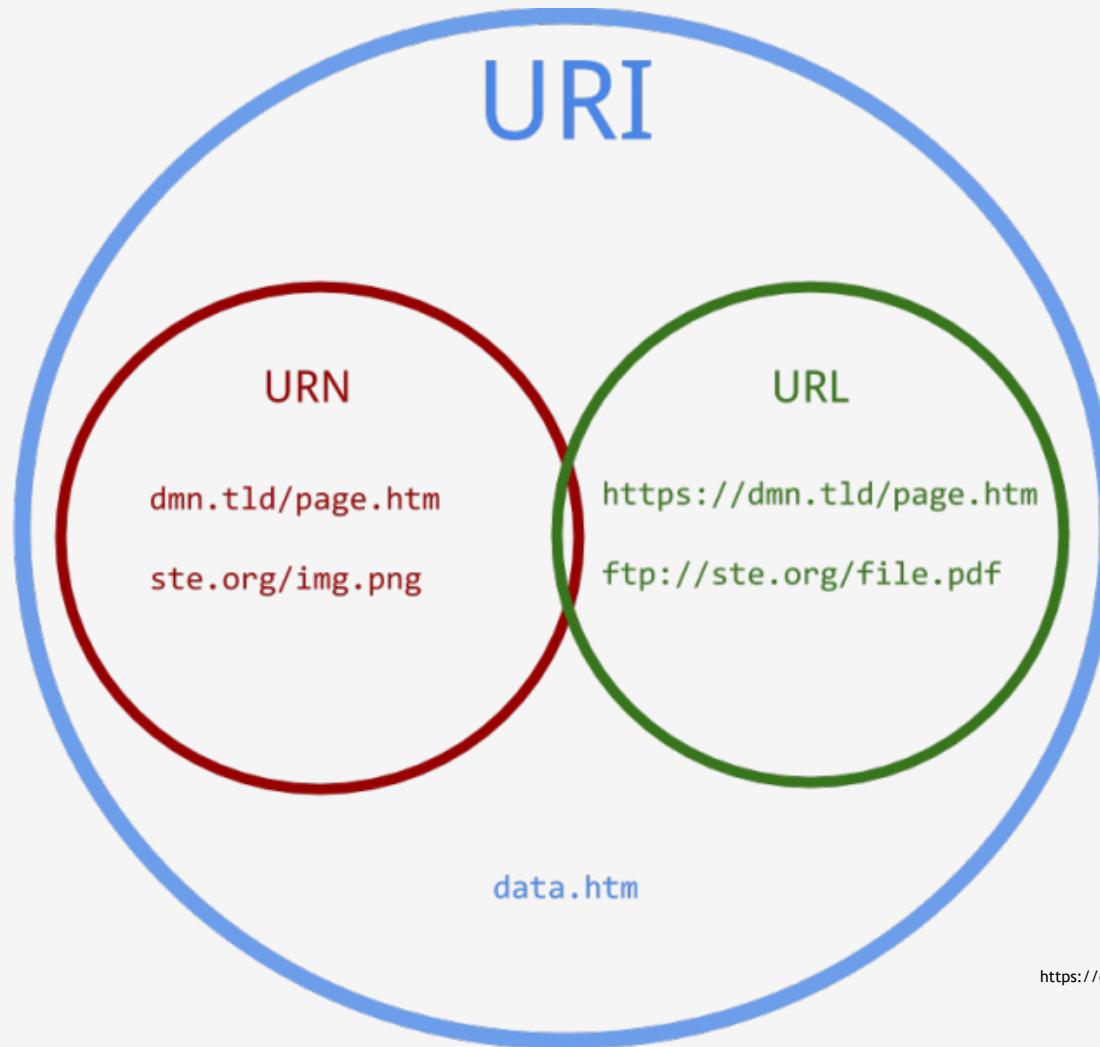
**URN - Unified Resource Name** - ujednolicony format nazw zasobów.

Składa się z identyfikatora przestrzeni nazw (NID) oraz ciągu znaków specyficznego dla danej przestrzeni (NSS).

Np: urn:lsid:zoobank.org:pub:CDC8D258-8F57-41DC-B560-247E17D3DC8C



Źródło: [12]





# Protokół HTTP (Hypertext Transfer Protocol)

- Protokół przesyłania dokumentów hipertekstowych (HTML).
- Zaprojektowany do komunikacji pomiędzy klientem (przeglądarką) a serwerem WWW ale może być także używany do innych celów.
- Działa w oparciu o protokół TCP/IP.
- Jest protokołem bezstanowym – serwer nie trzyma żadnych danych pomiędzy kolejnymi wywołaniami. Obejściami tego ograniczenia są:
  - parametry w URL'u, np.: /index.php?userid=7
  - ukryte pola formularzy
  - ciasteczka
  - sesje
- Standardowo korzysta z portu 80 TCP.



# HTTPS

Tak naprawde to HTTP, przy czym w warstwie transportu tressz zadania i odpowiedzi jest szyfrowana.

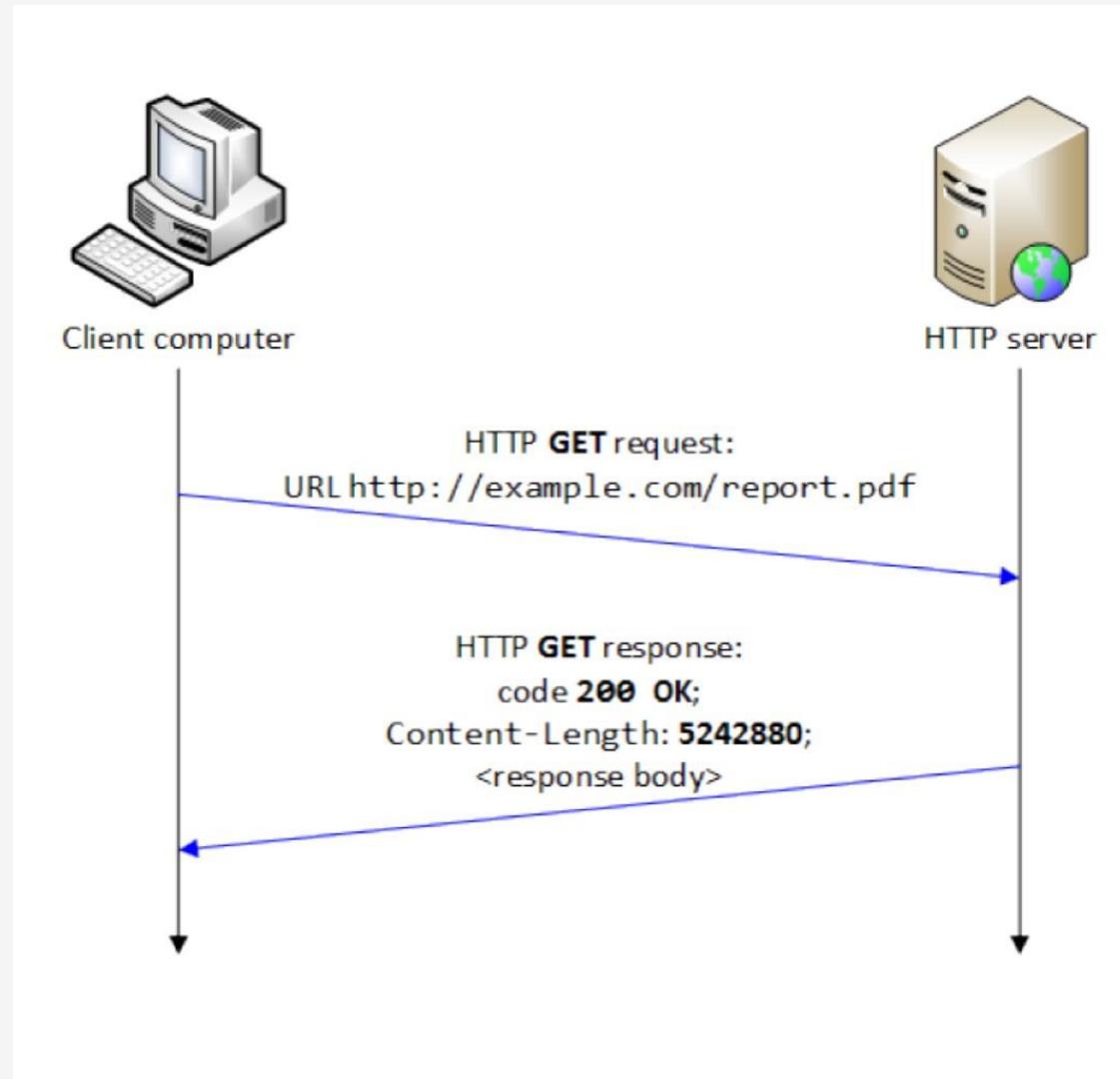
Kryptografia symetryczna – jednym kluczem można zaszyfrować i odszyfrować dane

HyperText Transfer Protocol

Kryptografia asymetryczna – jednym kluczem dane można zaszyfrować a innym odszyfrować



# Dialog HTTP



# Konstrukcja wiadomości HTTP



Linia statusowa

Nagłówki

Wiadomość (opcjonalnie)



# Żądanie HTTP

Żądanie HTTP ma postać:

Metoda Zasób Wersja-http

Nagłówek

Nagłówek

[pusta linia]

Zawartość (jeżeli potrzebna)

Przykład:

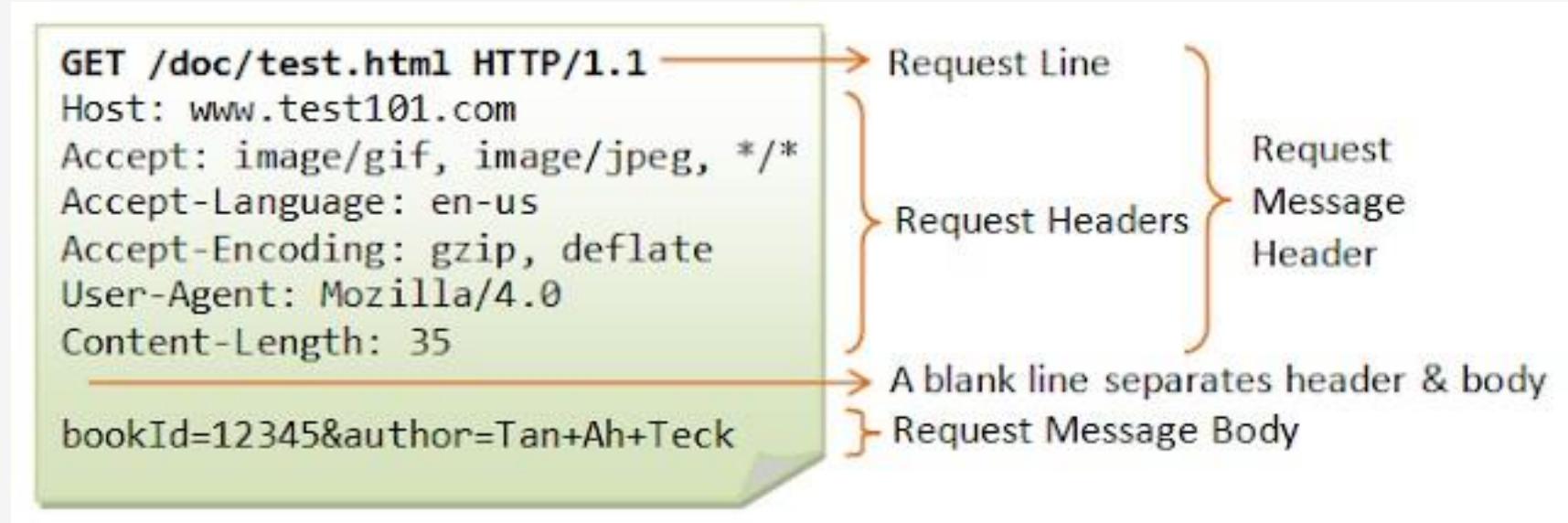
GET / HTTP/1.1

Host: www.domena.com

Connection: keep-alive



# Żądanie HTTP





# Odpowiedź HTTP

Odpowiedź HTTP ma postać:

Wersja-http   Kod-odpowiedzi   Uzasadnienie  
Nagłówek

Nagłówek

[pusta linia]

Zawartość (jeżeli potrzebna)

**Przykład:**

HTTP/1.1 200 OK.

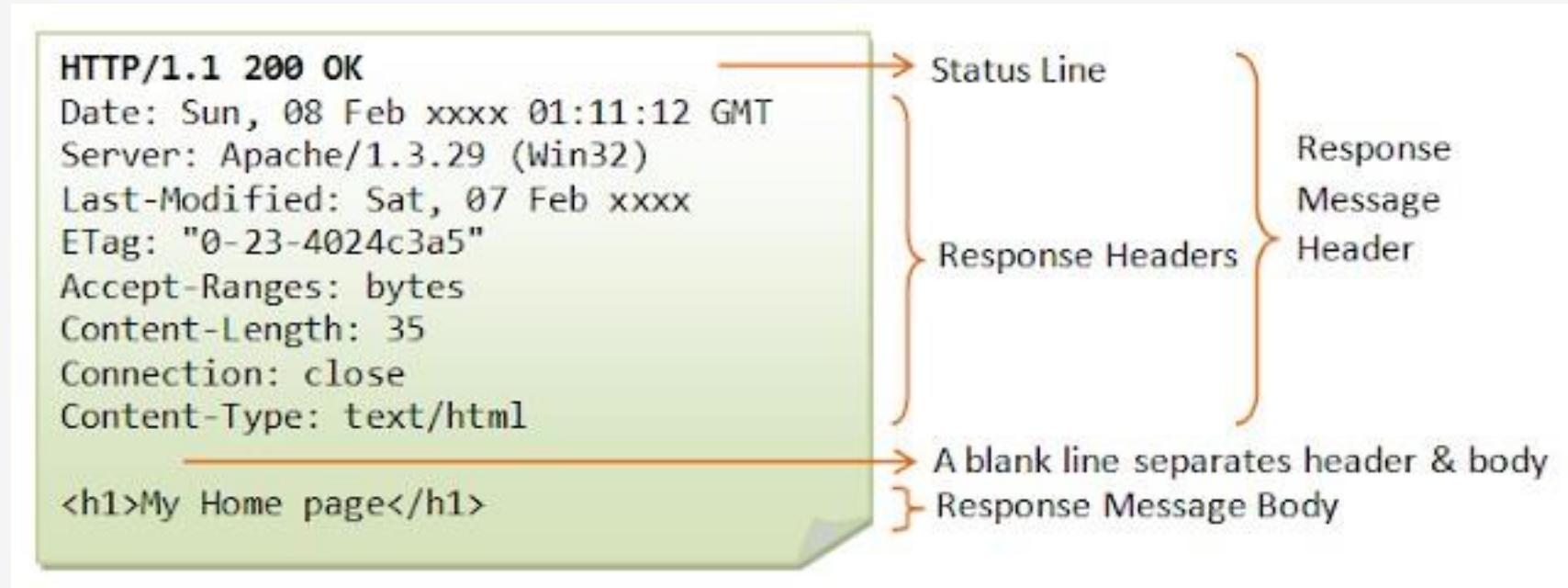
Server: Apache

Content-type: text/html

[Dokument HTML]



# Odpowiedź HTTP



# Jeszcze trochę o req/res headers w HTTP



[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_header\\_fields](https://en.wikipedia.org/wiki/List_of_HTTP_header_fields)



# Metody HTTP

Metoda	Opis
DELETE	Żądanie usunięcia zasobu (dokumentu) z serwera
GET	Żądanie zasobu od serwera w formie nagłówka i treści
HEAD	Żądanie zasobu od serwera w formie nagłówka
LINK	Żądanie ustanowienia relacji między istniejącymi zasobami
OPTIONS	Żądanie od serwera identyfikacji obsługiwanych metod
POST	Żądanie odebrania przez serwer danych od klienta
PUT	Żądanie odebrania przez serwer od klienta pliku . :
TRACE	Żądanie zwrócenia przez serwer nagłówków wiadomości wysłanej od klienta (w celach testowania)
UNLINK	Żądanie usunięcia relacji między istniejącymi zasobami



# HTTP - metody

HTTP Method	RFC	Request Has Body	Response Has Body	Idempotent
<b>GET</b>	<a href="#">RFC 7231</a>	No	Yes	Yes
HEAD	<a href="#">RFC 7231</a>	No	No	Yes
<b>POST</b>	<a href="#">RFC 7231</a>	Yes	Yes	No
<b>PUT</b>	<a href="#">RFC 7231</a>	Yes	Yes	Yes
<b>DELETE</b>	<a href="#">RFC 7231</a>	No	Yes	Yes
CONNECT	<a href="#">RFC 7231</a>	Yes	Yes	No
<b>OPTIONS</b>	<a href="#">RFC 7231</a>	Optional	Yes	Yes
TRACE	<a href="#">RFC 7231</a>	No	Yes	Yes
<b>PATCH</b>	<a href="#">RFC 5789</a>	Yes	Yes	No



# Metoda GET

Request URL: [http://kis.p.lodz.pl/media/js/jquery.dataTables\\_pl.txt](http://kis.p.lodz.pl/media/js/jquery.dataTables_pl.txt)

Request Method: GET

Status Code: 200 OK

▼ Request Headers [view source](#)

Accept: application/json, text/javascript, \*/\*

Accept-Encoding: gzip,deflate,sdch

Accept-Language: pl-PL,pl;q=0.8,en-US;q=0.6,en;q=0.4

Connection: keep-alive

Cookie: PHPSESSID=92oqujt1ksnr3jh5v1ku0vijv3; KISlang=pl; last\_banner=9

Host: kis.p.lodz.pl

Referer: <http://kis.p.lodz.pl/news.html?lang=pl>

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML

X-Requested-With: XMLHttpRequest



# Metoda POST

Request URL: [https://www.orange.pl/prt/pl/order/process/?\\_DARGS=/gear/order-service/process/ord](https://www.orange.pl/prt/pl/order/process/?_DARGS=/gear/order-service/process/ord)

Request Method: POST

Status Code: 200 OK

▼ Request Headers [view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Encoding: gzip,deflate, sdch

Accept-Language: pl-PL,pl;q=0.8,en-US;q=0.6,en;q=0.4

Connection: keep-alive

Content-Length: 1570

Content-Type: application/x-www-form-urlencoded

Cookie: TPC=31.183.97.73.1380639569430854; ALLOW\_COOKIES=cookiesAllowed; \_\_lc.visitor\_id.1325064E5EB29BECCC2369C1828E4D474.ocpwww609; rateOrange=true; MSID=16514D373F9627DCE38EBB45A6128AD=5AF951EBC049D2BDE681DEC3369DDEAB.ptpportal1406; ATG\_SESSION\_ID=5AF951EBC049D2BDE681DEC3369D1406; tp\_CheckCookie="Your viewer attends cookies"; \_\_utma=132817616.1246361074.1380639569.132817616; \_\_utmz=132817616.1383224083.4.1.utmcsrc=(direct)|utmccn=(direct)|utmcmd=(none)

Host: www.orange.pl

Origin: <https://www.orange.pl>

Referer: <https://www.orange.pl/prt/pl/order/process/?formId=neo-ins&minimize=additionalproductIO=ADD&modemParam=01&periodParam=01&promotionParam=01&additionalParam=01>

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/

## NAGŁÓWKI HTTP

Przenoszą dodatkowe informacje o żądaniu oraz odpowiedzi

Przesyłane jako pola klucz:wartość

Posiada wiele zdefiniowanych nagłówków

Można tworzyć własne nagłówki

Maksymalna ilość nagłówków zależy od implementacji serwera





# HTTP - nagłówki

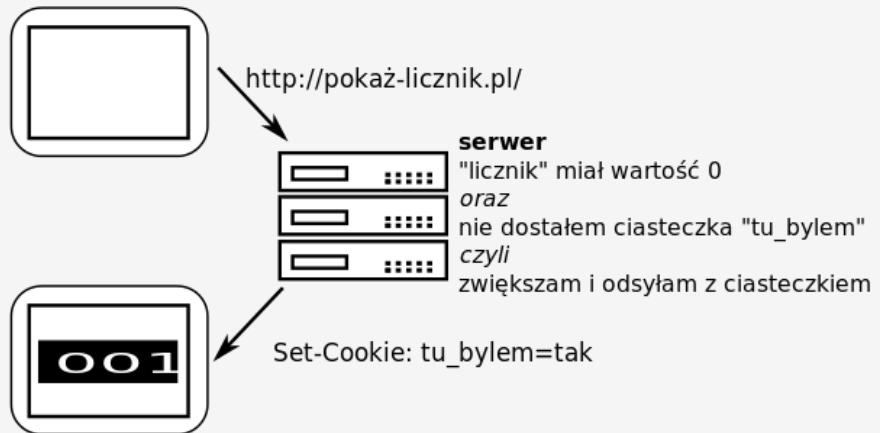
Najczęściej używane:

- Content-type
- Accept
- Authorization
- Location
- Accept-Charset
- User-agent
- Set-Cookie
- Cookie

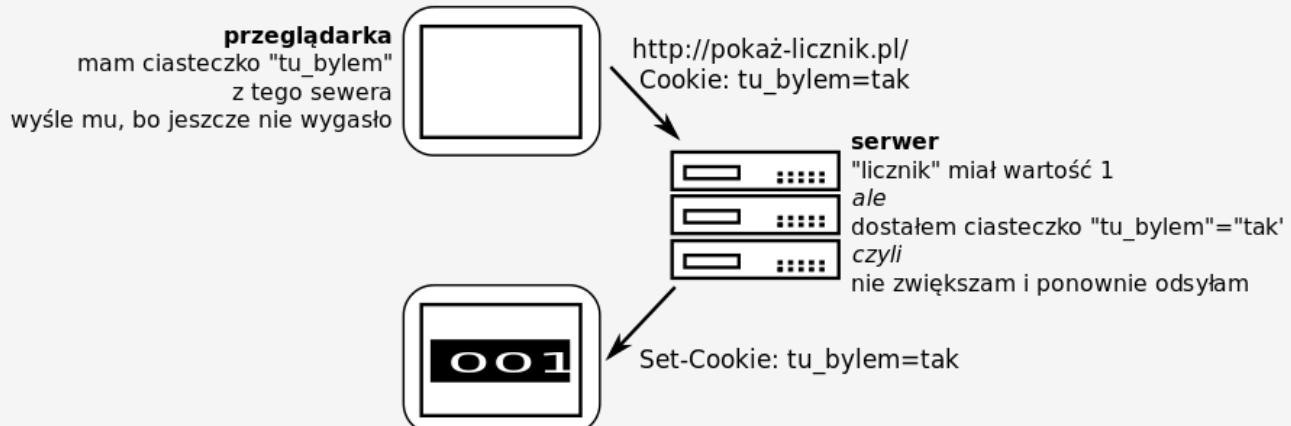


# Cookies

## Pierwsze odwiedziny



## Kolejne odwiedziny



Źródło: [https://pl.wikipedia.org/wiki/HTTP\\_cookie](https://pl.wikipedia.org/wiki/HTTP_cookie)



# HTTP – ciasteczka (cookies)

Przy komunikacji z serwerem negocjacja ciasteczek odbywa się za pomocą nagłówków. Ciasteczka jednak są przechowywane przez przeglądarki.

Maksymalny rozmiar jednego ciasteczka to 4096 bajtów

Mozna ustawić czas zycia (expires)

**Ciasteczka używane są również do trzymania sesji (JSESSIONID !!)**

# HTTP kody odpowiedzi





# HTTP – kody odpowiedzi

Informują o statusie obsługi zadania. Znajdują się w odpowiedzi. Występują zawsze w odpowiedzi nawet jeśli odpowiedź jest „pusta”. Kody błędów to 3 cyfry.

Są kategoryzowane wg. pierwszej cyfry:

- 1xx – kody informacyjne
- 2xx – kody powodzenia
- 3xx – kody przekierowań
- 4xx – kody błędu klienta
- 5xx – kody błędu serwera



- 100 - Continue – „Prześlij resztę danych”
- 101 - Switching Protocols – przetworze to żądanie jeśli wyślesz je za pomocą protokołu z nagłówka Upgrade



- 200 - OK - Żądanie wykonane prawidłowo
- 201 - Created – Nowy zasób został utworzony
- 202 - Accepted – Zrozumiałem i przyjąłem Twoje żądanie



- 301 - Moved Permanently – Kiedyś był tutaj zasób ale już go nie ma i nie będzie
- 302 - Moved Temporarily – Zasób chwilowo nieobecny pod tą lokalizacją
- 303 - See other – Zasób którego szukasz nie znajduje się tutaj ale sprawdź tam



- 400 - Bad Request – żądanie nie może być obsłużone przez serwer z powodu nieprawidłowości postrzeganej jako błąd użytkownika
- 401 - Unauthorized - Nieautoryzowany dostęp – żądanie zasobu, który wymaga uwierzytelnienia
- 403 - Forbidden - Zabroniony – serwer zrozumiał zapytanie, lecz konfiguracja bezpieczeństwa zabrania mu zwrócić żądanego zasób
- 404 - Not Found – Wskazany zasób nie istnieje
- 405 - Method Not Allowed – Zła metoda do wywołania tego adresu
- 415 - Unsupported Media Type – Typ zawartości żądania nie jest obsługiwany



- 500 - Internal Server Error – Wewnętrzny błąd serwera – serwer napotkał niespodziewane trudności, które uniemożliwiły zrealizowanie żądania
- 503 - Service Unavailable – Serwer nie był w stanie odpowiedzieć z powodu przeciążenia
- 504 - Gateway Timeout – Przesyłanie danych pomiędzy serwerami trwało zbyt długo

# Narzędzia developerskie w przeglądarkach



Screenshot of a web browser showing developer tools open over a Google Doodle for Wilbur Scoville's 151st birthday.

The main content area displays the Google homepage with the doodle. The doodle features a man in a lab coat holding a test tube, set against a background of various chili peppers. Below the doodle, a sidebar shows a smaller version of the doodle and some text about Lotte Reiniger's 117th birthday.

The browser's address bar shows the URL: <https://www.google.com/doodles/wilbur-scovilles-151st-birthday>.

The developer tools Network tab is active, showing a timeline of requests. One request is selected: "wilbur-scovilles-151st-birthday".

**Request URL:** <https://lh3.googleusercontent.cJJJcwsxr0tRG6PDUB-IjTGgPhAMu93FjYGYMeKHntxeGZajIG2JeSGFwsBNqjRIQ6RJoQN90q3K6fsXkH7kzgZ3p0ebq=s0>

**Request Method:** GET

**Status Code:** 200

**Remote Address:** 172.217.16.1:443

**Referrer Policy:** no-referrer-when-downgrade

**Response Headers:**

- access-control-allow-origin: \*
- access-control-expose-headers: Content-Length
- age: 7703
- alt-svc: quic=":443"; ma=2592000; v="44,43,39"
- cache-control: public, max-age=86400, no-tran...
- content-disposition: inline; filename="unnamed.p...
- content-length: 80312
- content-type: image/png



# HTTP – do czego najczęściej używane

- HTML oraz dowolne inne pliki.
- Webserwisy
  - SOAP
  - REST



- Wymaga wiedzy i narzędzi do otworzenia i odebrania wiadomości
- Interoperacyjny, ale tylko na poziomie transportu
- Nastawiony na operacje
- Korzysta z SOA (Service Oriented Architecture)
- Używa jedynie metody HTTP POST
- Format: tylko XML
- Bardzo sformalizowana komunikacja



# SOAP - przykład

```
<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <ns1:GetCDLPDF xmlns:ns1="http://cropscape.csiss.gmu.edu/CDLService/">
            <year>2011</year>
            <fips>19037</fips>
            <papersize>Letter</papersize>
            <title>2011 Chickasaw County, Iowa</title>
            <boundaryflag>true</boundaryflag>
        </ns1:GetCDLPDF>
    </soapenv:Body>
</soapenv:Envelope>
```

(a)

```
<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <ns1:GetCDLPDFResponse xmlns:ns1="http://cropscape.csiss.gmu.edu/CDLService/">
            <returnURL>
                http://nassgeodata.gmu.edu/nass_data_cache/CDL_2011_19037.pdf
            </returnURL>
        </ns1:GetCDLPDFResponse>
    </soapenv:Body>
</soapenv:Envelope>
```



# SOAP - przykład

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
    xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

    <soap:Body xmlns:m="http://www.example.org/stock">
        <m:GetStockPrice>
            <m:StockName>IBM</m:StockName>
        </m:GetStockPrice>
    </soap:Body>

</soap:Envelope>
```

<https://www.w3.org/2001/12/soap-envelope>



- Bezstanowa komunikacja
- Korzystanie z zasobów (resources)
- W całości oparty na protokole HTTP
- Reprezentacja zasobu pozwala na wykonywać operacje CRUD
- Dowolny format danych, najczęściej JSON
- Klarowna komunikacja
- prostota (dobry serwis REST opisuje się sam)
- skalowalność i modyfikowalność
- przenośność (brak zależności od platformy)
- .



# REST - zasoby

- Elementy wystawiane przez system na zewnątrz
- Wszystko co może mieć reprezentację, może być zasobem (osoba, rezerwacja, wyjątek)
- Reprezentowane przez URI
- Mogą być reprezentowane w różnych formatach
- Nie można wyszukać żadnych powiązań pomiędzy żądaniami

# REST





# REST - zasoby

Request URL	POST	GET	PUT	DELETE
/fabryka	Stwórz nowy samochód	Pobierz wszystkie samochody	Zamień/aktualizuj wszystkie samochody	Usuń wszystkie samochody
/fabryka/ford	Stwórz nowy samochód marki ford	Pobierz wszystkie samochody marki ford	Zamień/aktualizuj wszystkie samochody marki ford	Usuń wszystkie samochody marki <b>ford</b>

Źródło: [http://www.moseleians.co.uk/wp-content/uploads/cmdm/9632/1422444257\\_api-restowe-whitepaper.pdf](http://www.moseleians.co.uk/wp-content/uploads/cmdm/9632/1422444257_api-restowe-whitepaper.pdf)

# Postman



The screenshot shows the Postman application interface. At the top, there are buttons for NEW, Runner, Import, and a search bar. The title bar indicates "Builder" and "Team Library". A notification bar at the top says "Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more".

The main area has tabs for History, Collections, and a Filter search bar. The History tab is selected, showing a list of recent requests:

- Today: GET https://jsonplaceholder.typicode.com/todos
- August 9:
  - GET https://jsonplaceholder.typicode.com/todos/1
  - GET https://sucharry.pl/20322
  - GET http://piszsuchary.pl/losuj
  - GET https://http.cat/100
  - GET google.com

A modal window for a specific request is open:

- Method: GET
- URL: https://jsonplaceholder.typicode.com/todos
- Params: None
- Send and Save buttons
- Authorization tab (selected): Type No Auth
- Headers tab: Headers (17)
- Body tab (selected): Pretty, Raw, Preview, JSON (selected), and a large JSON response pane.
- Test Results tab
- Status: 200 OK, Time: 215 ms

The JSON response pane displays the following data:

```
[{"id": 1, "userId": 1, "title": "delectus aut autem", "completed": false}, {"id": 2, "userId": 1, "title": "quis ut nam facilis et officia qui", "completed": false}, {"id": 3, "userId": 1, "title": "fugiat veniam minus", "completed": false}, {"id": 4, "userId": 1, "title": "et porro tempora", "completed": true}]
```



# GET - ZADANIE

1. Wejdź na stronę <https://punkapi.com/documentation/v2>
2. Za pomocą dokumentacji wykonaj następujące zapytania:
  1. Pobranie wszystkich piw z bazy
  2. Pobranie piwa o ID 27
  3. Pobranie drugiej strony z 15 wynikami
3. Wszystkie zapytania zapisz w Postman w kolekcji Beer API

Dodatkowe:

1. Napisz zapytanie, które zwróci kod odpowiedzi HTTP 400
2. Napisz zapytanie, które zwróci piwa o IBU większym niż 50
3. Napisz zapytanie, które zwróci drugą pięcioelementową stronę z piwami, które pasują do barbecue i zostały uwarzone po 11/2011 a ich IBU należy do zakresu (2:100)
4. Sprawdź czy możesz pobrać losowe piwo z użyciem filtru. Czy filtr działa?

# API ćwiczenia



1. Załącz konto na stronie <https://gorest.co.in>
2. Zapoznaj się z dokumentacją API
3. Pobierz listę wszystkich:
  1. Użytkowników
  2. Postów
  3. Zdjęć
4. Wypisz listę wszystkich użytkowników ale tym razem pobierz tylko część danych o użytkownikach (data urodzenia, telefon, status, id)
5. Pobierz użytkownika z konkretnym id
6. Pobierz albumy użytkownika o konkretnym id
7. Utwórz nowego użytkownika
8. Pobierz wcześniej utworzonego użytkownika
9. Uaktualnij użytkownika z punktu 6, zmieniając jego adres email
10. Pobierz uaktualnianego użytkownika
11. Dodaj nowy post do użytkownika
12. Pobierz użytkownika
13. Usuń użytkownika



# Źródła

- Wikipedia - pojęcia „Protokół” i „Model ISO/OSI”, „DNS”
- Prezentacje z poprzednich edycji kursów SDA, udostępnione przez SDA
- „Wprowadzenie do architektury REST,” [http://www.moseleians.co.uk/wp-content/uploads/cmdm/9632/1422444257\\_api-restowe-whitepaper.pdf](http://www.moseleians.co.uk/wp-content/uploads/cmdm/9632/1422444257_api-restowe-whitepaper.pdf)
- „Narzędzia TCP/IP”, [http://soisk.info/index.php/Narzędzia\\_TCP/IP](http://soisk.info/index.php/Narzędzia_TCP/IP)
- „Diagnostyka portów i protokołów”, [http://www.iniejawna.pl/doc/programy/diagnostyka\\_%20port.pdf](http://www.iniejawna.pl/doc/programy/diagnostyka_%20port.pdf)
- „Mikroserwery TCP/IP - podstawy TCP/IP”, <http://www.pg.gda.pl/~zbczaja/pdf/mse/podstawyt>
- „Protokół HTTP ,omówienie standardu z analizą ruchu sieciowego”,  
[http://stud.ics.p.lodz.pl/~adrstud/files/http\\_protocol.pdf](http://stud.ics.p.lodz.pl/~adrstud/files/http_protocol.pdf)
- „Prezentacja TCP/IP”, <http://www.metal.agh.edu.pl/~blzielin/TCP.ppt>
- „#Niezbędnik Juniora. Protokół HTTP”, <https://kobietydokodu.pl/niezbiednik-juniora-protokol-http/>