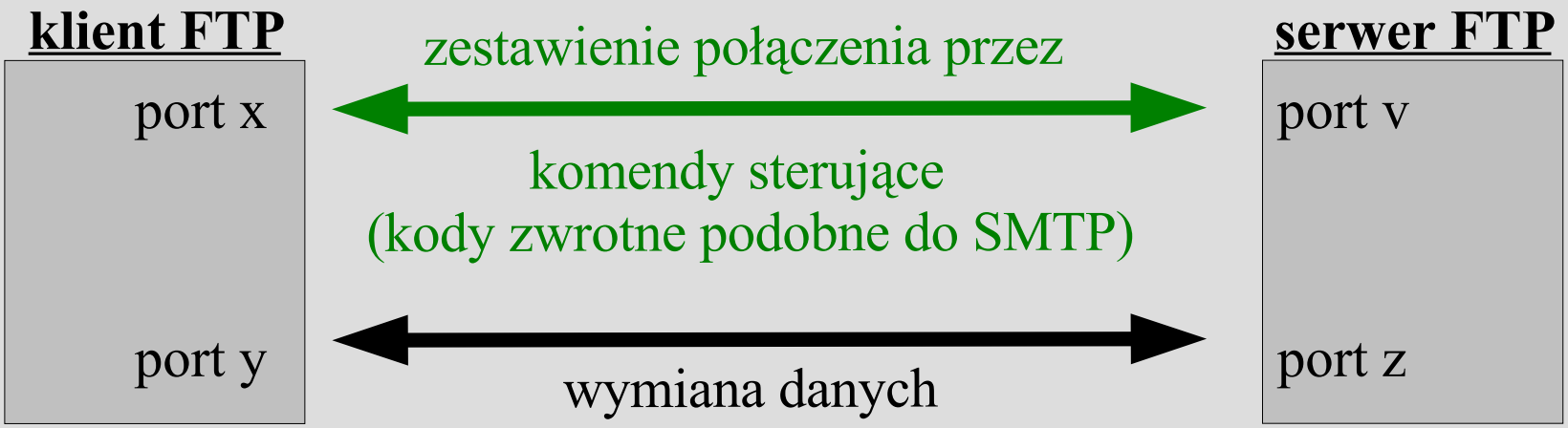
FTP – co to takiego?

FTP – File Transfer Protocol (Protokół Przesyłania Plików) *RFC 114,959*

Protokół niezawodnego przesyłania plików za pomocą prostych komend tekstowych.

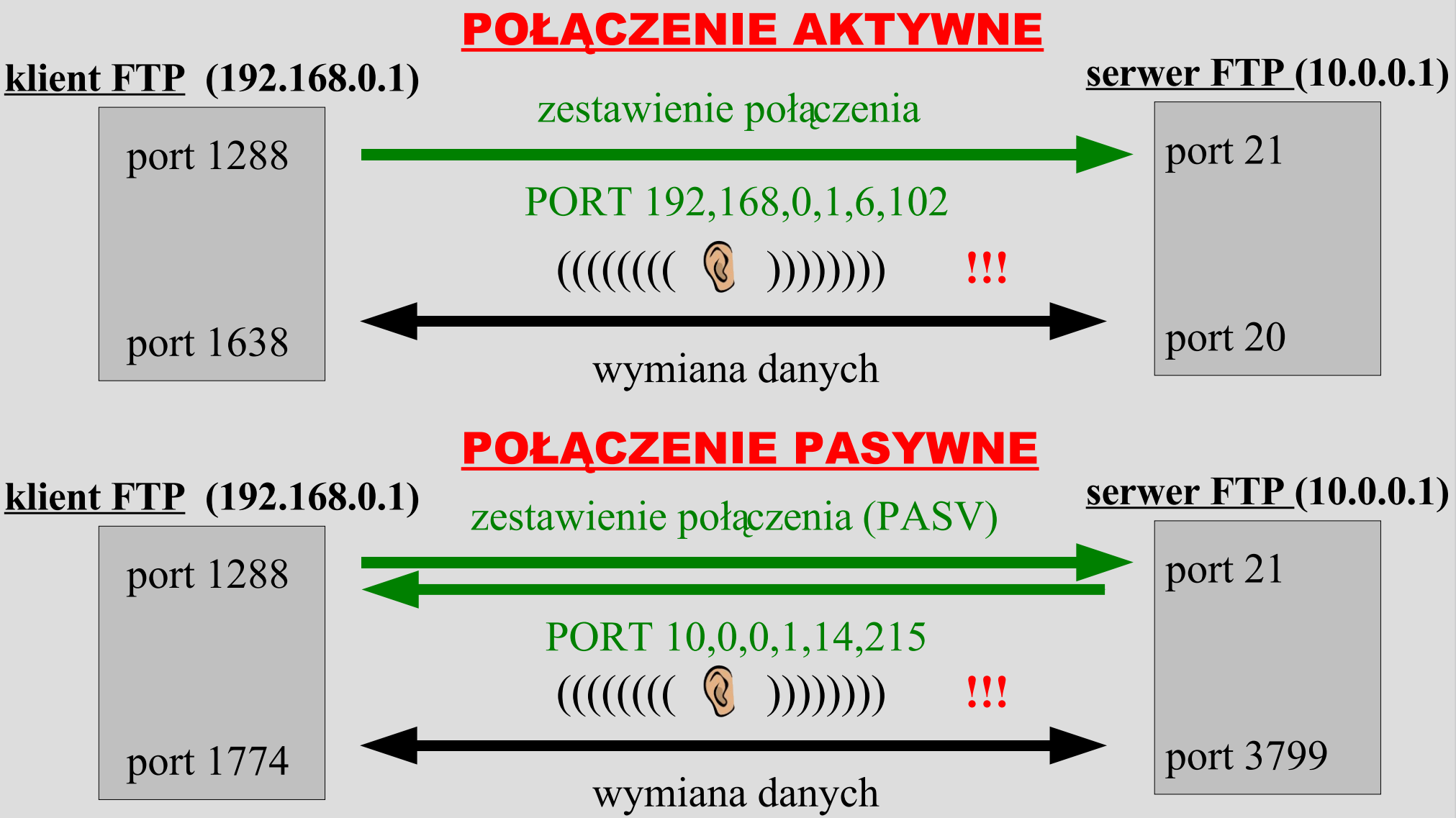
Jeden z najstarszych protokołów stosowanych w Internecie (1971r.)

FTP – jak to działa?



komenda	Opis
USER	nazwa użytkownika
PASS	hasło logowania
LIST	wyświetlenie listy plików i katalogów
ABOR	przerwanie wszystkich połączeń
QUIT	wylogowanie z serwera
RETR	pobieranie pliku
STOR	wgrywanie pliku
PORT	zestawienie połączenia aktywnego
PASV	zestawienie połączenia pasywnego

Rodzaje połączeń FTP



Proftpd – konfiguracja (/etc/proftpd.conf)

- ServerType inetd
- PassivePorts 65100 65200
- DisplayLogin welcome.msg
- Port 2221
- User nobody
  Group nogroup
- <Directory Upload>
  <Limit WRITE> Allow from 192.168.0.0/24 </Limit>
  </Directory>
- <Anonymous /home/ftp>
  MaxClients 10
  ...
  </Anonymous>
- <VirtualHost „ftp.xyz.pl”>
  Port 3221
  ...
  </VirtualHost>

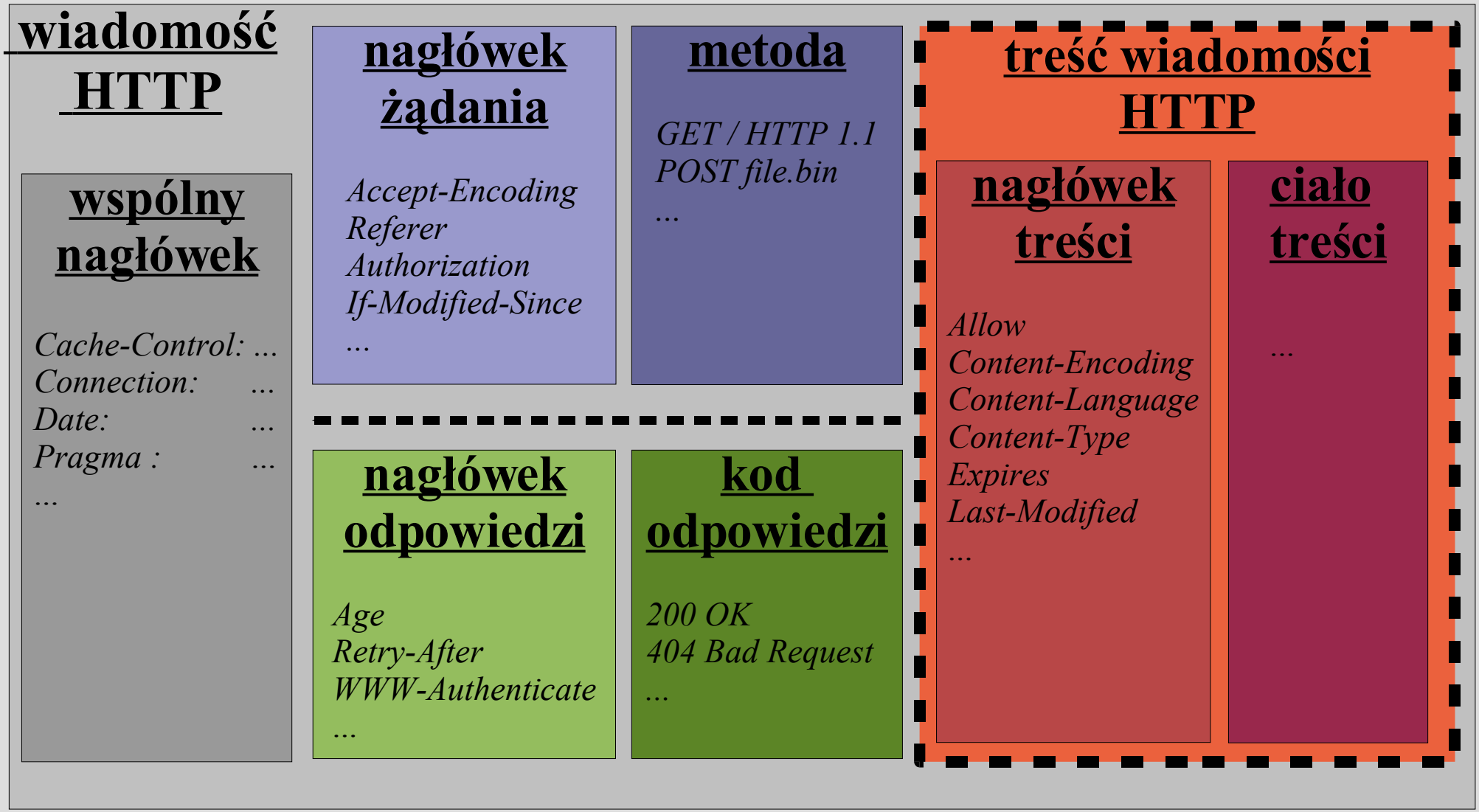
HTTP – co to takiego?

HTTP – HyperText Transfer Protocol (Protokół Przesyłania Hipertekstu) *RFC 2616*

Bezstanowy (bazujący na prostej regule „żądanie-odpowiedź) protokół udostępniania dokumentów WWW (w formie tzw. **„*hipertekstu*”** - tekstu, odnośników, formularzy, tabel, grafiki itp.) za pomocą prostych komend tekstowych.

Jeden z najpopularniejszych protokołów stosowanych w Internecie i rdzeń jego funkcjonalności.

HTTP – rodzaje wiadomości



Metody żądań HTTP

- **OPTIONS**
  Żądanie określenia możliwości serwera (podanych w nagłówku żądania)
- **GET**
  Pobranie wskazanego zasobu
- **HEAD**
  Pobranie tylko nagłówka z meta-informacjami strony WWW
- **POST**
  Przesłanie danych na serwer do podanego zasobu
- **PUT**
  Przesłanie wskazanych danych na serwer
- **DELETE**
  Usunięcie wskazanego zasobu
- **TRACE**
  Odesłania żądania do nadawcy (diagnostyka połączenia)
- **CONNECT**
  Żądanie tunelowania połączenia ze wskazanym proxy

Kody odpowiedzi HTTP

- **1xy** (Informacja)
  Żądanie otrzymane, proces wykonywania w toku
- **2xy** (Akceptacja)
  Żądanie otrzymane i poprawnie wykonane
- **3xy** (Przekierowanie)
  Żądanie otrzymane , ale wymagana dalsza interakcja klienta
- **4xy** (Błąd klienta)
  Żądanie ma złą składnie, bądź nie może być wykonane
- **5xy** (Błąd serwera)
  Serwer nie był w stanie wykonać prawdopodobnie poprawnego żądania

kod	Opis
100 Continue	Serwer wykonuje żądanie,klient może wysyłać kolejne żądania
200 OK	Żądanie wykonane prawidłowo
301 Moved Permanently	Żądany zasób znajduje się pod innym adresem, konieczne podanie nowego URI
400 Bad Request	Nieznane żądanie (błąd syntaktyczny)
401 Unauthorized	Odmowa dostępu do zasobu, z powodu błędnej autoryzacji
404 Not Found	Nie ma takiego zasobu
500 Internal Server Error	Serwer napotkał niespodziewany błąd wewnętrzny
503 Service Unavailable	Serwer jest zbyt obciążony

Charakterystyka serwera Apache

Najpopularniejszy serwer HTTP (62%) w Internecie.

- Z dostępnym kodem źródłowym
- wsparcie dla hostów wirtualnych
- obsługa indywidualnych polityk dostępu do katalogów
- wysoce modularny (php, mysql, proxy, auth, ssl, itp.).
- wszechstronne opcje logowania

Jest poprawioną i bardziej rozbudowaną pochodną serwera NCSA (stąd nazwa: *„a patchy server”*)

konfiguracja serwera HTTP na przykładzie Apache(/etc/apache/httpd.conf)

- **ServerType standalone**
- **ServerRoot /etc/apache**
- **KeepAlive On**
- **StartServers 5**
  MinSpareServers 5
  MaxSpareServers 10
  MaxClients 150
- **User www-data**
  Listen 192.168.0.5:80
  Listen 127.0.0.1:8080
- **Include /etc/apache/modules.conf**
  (LoadModule php4_module /usr/lib/apache/1.3/libphp4.so)
- **DocumentRoot /var/www**
- **AddType application/x-httpd-php3 .php3**
- **DirectoryIndex index.php index.html**
- **Alias /usr/local/httpd/cgi-bin/ /cgi-bin/**
- **LogFormat "%h %l %u %t " common**

- **<Directory /var/www/>**
  Options Indexes FollowSymLinks
  Allow from 192.168.0.0/24
  ...
  </Directory>

- **<IfModule mod_userdir.c>**
  UserDir public_html
  ...
  </IfModule >

- **<VirtualHost *>**
  DocumentRoot /var/virtual
  ServerName wirt.domena.pl
  CustomLog /var/log/virt/access.log common
  ...
  </VirtualHost>

Apache – moduły (fragment)

- **mod_access** - kontrola dostępu do plików w zależności od adresu IP
- **mod_alias** - pozwala mapować część systemu plików w katalogu głównym Apache'a, umożliwia też przekierowywanie adresów URL.
- **mod_auth** - uwierzytelnianie użytkowników
- **mod_cgi** - wykonywanie skryptów CGI po stronie serwera
- **mod_dir** - podstawowe operacje na katalogach.
- **mod_env** -przekazywanie zmiennych środowiskowych do skryptów CGI/SSI
- **mod_expires** - dodaje znacznik Expires do stron WWW przesyłanych klientowi - ważne dla często zmienianych serwisów, które powinny być zawsze aktualne
- **mod_headers** - pozwala na dowolną modyfikację nagłówków HTTP
- **mod_info** - odpowiedzialny za informację o ustawieniach serwera Apache
- **mod_log_agent** - zapisywanie w logach nazw i wersji przeglądarek klientów
- **mod_proxy** – uruchomienie funkcji proxy dla stron WWW
- **mod_rewrite** - modyfikowanie adresów URL w "locie" za pomocą wyrażeń regularnych
- **mod_speling** - poprawianie błędów w adresach URL
- **mod_userdir** - ustawienia dotyczące katalogów domowych użytkowników
- **mod_usertrack** - śledzenie zachowań użytkowników za pomocą Cookies
- **mod_vhost_alias** - konfiguracja serwerów wirtualnych

Apache – konfiguracja autentykacji w dostępie do stron WWW

- stworzenie pliku z listą autoryzowanych użytkowników

  htpasswd -c /katalog_z_dostepem_na_haslo/plik_hasel uzytkownik

- umieszczenie odpowiednich dyrektyw w sekcji **<Directory>** pliku konfiguracji globalnej serwera (httpd.conf) lub w dedykowanym pliku dostępu (**.htaccess**) umieszczonym w katalogu dostępnym po autentykacji:

  AuthType Basic
  AuthName "Podaj hasło"
  AuthUserFile ***/katalog_z_dostepem_na_haslo/plik_hasel***
  Require user ***nazwa_uzytkownika***

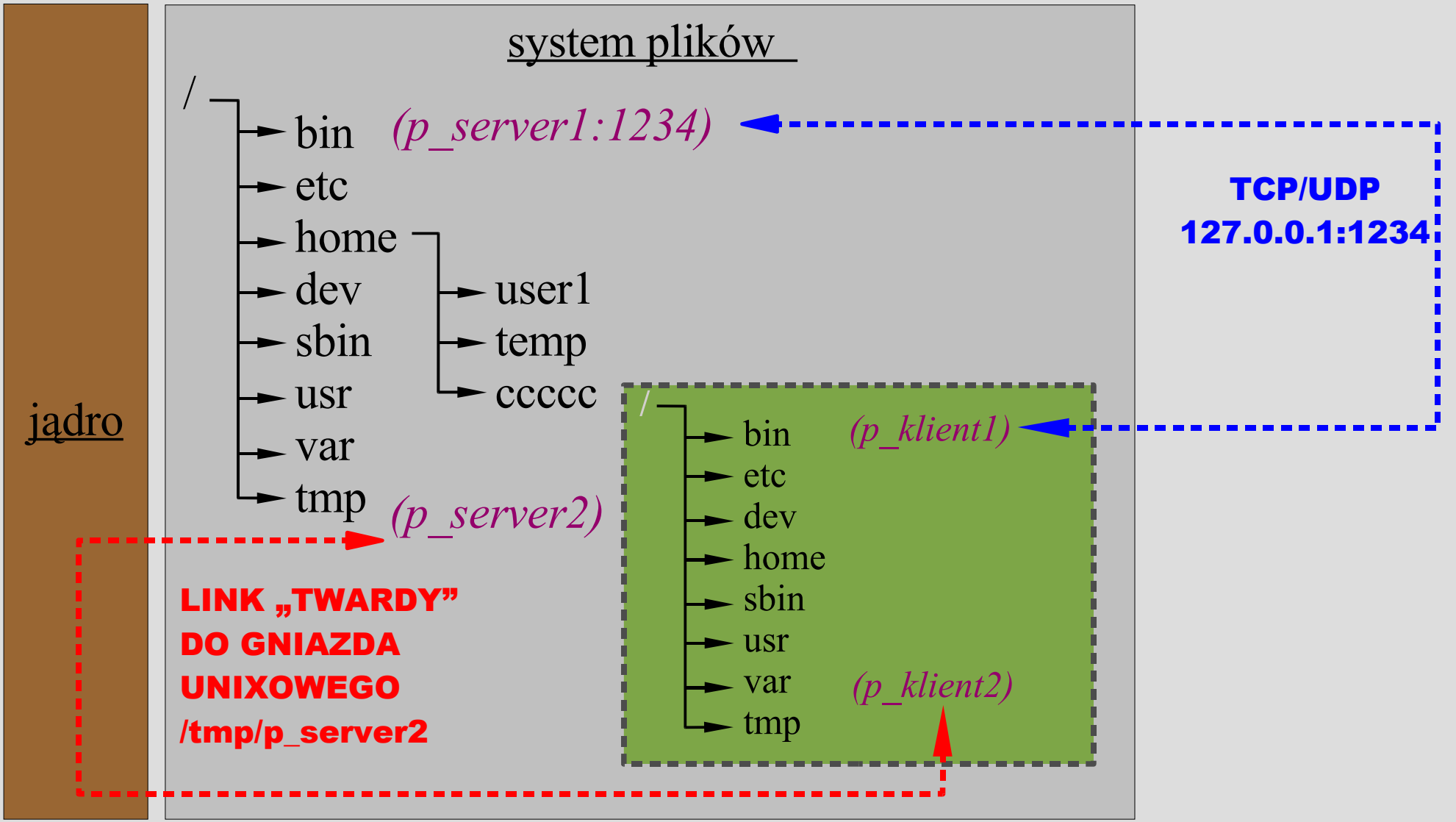
Chroot'owanie – co to takiego?

Chroot (*change root*) –

Operacja która tworzy katalog główny (/) w kontekście uruchomionego procesu (i jego potomków). Taki proces nie może uzyskać dostępu do plików spoza stworzonej w ten sposób „klatki”.

Możliwa do wykonania tylko na systemach uniksowych.

Chroot'owanie – schemat działania i sposoby komunikacji



Chroot'owanie – przygotowanie środowiska

- stworzenie listy plików wymaganych do uruchomienia procesu w izolowanym środowisku *chroot*

  strace *proces* 2>&1 | grep 'open' > *plik*

  fragment stworzonej listy:

```
open("/etc/passwd", O_RDONLY)       = 4
open("/lib/tls/libm.so.6", O_RDONLY)    = 3
open("/lib/tls/libpthread.so.0", O_RDONLY) = 3
```

- stworzenie katalogu w którym będzie rezydowało środowisko izolowane, oraz umieszczenie w nim plików wymaganych przez proces (!)

  mknod */chrootowany_katalog/dev/null* c 2 2

- uruchomienie środowiska izolowanego *chroot*

  chroot */chrootowany_katalog/ chrootowany_proces* lub
  chrootuid */chrootowany_katalog/ użytkownik chrootowany_proces*