

# Współczesne Sieciowe Systemy Operacyjne

**Grzegorz Blinowski**

**"CC"**

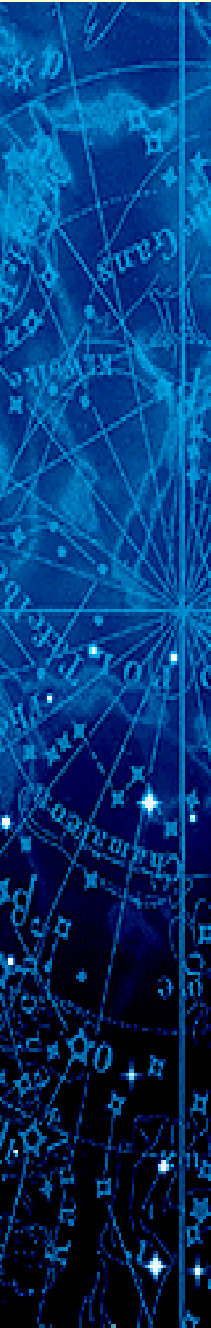
**Grzegorz.Blinowski@cc.com.pl**

**<http://www.cc.com.pl/>**

**tel (22) 646-68-73; faks (22) 606-37-80**

# Zadania SSO

- **Serwer plików**
- **Serwer drukowania**
- **Serwer bazodanowy**
  
- **Serwer katalogowy**
- **Serwer aplikacji**



# **Serwery Aplikacji**

# Co to są usługi katalogowe?

- Spis - książka telefoniczna, typu "Białe" lub "Żółte strony"
- .... i więcej niż książka telefoniczna
- Odzwierciedla strukturę organizacyjną firmy
- Przykłady tradycyjnego systemu katalogowego:
  - Książka adresowa programu pocztowego
  - DNS (translacja nazw na adresy IP)

# Potrzeba zapewnienia usług katalogowych

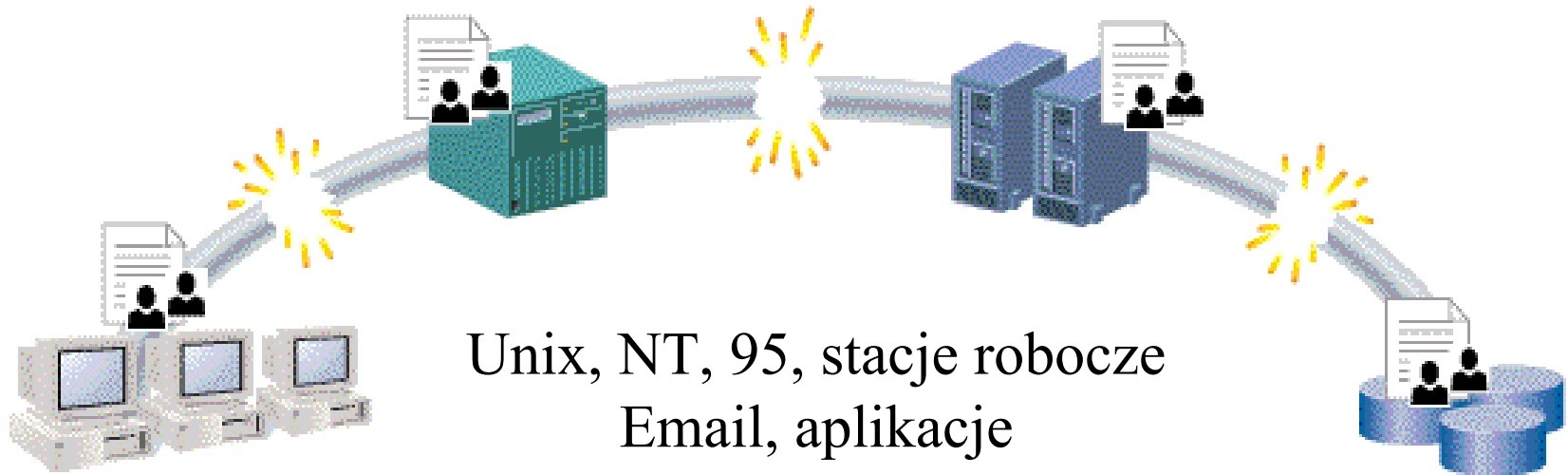
- Intranet
  - Książki adresów e-mail, telefonów, itd.
  - inne atrybuty osób dla aplikacji typu: pracy grupowej, zarządzania procesami, itd.
  - informacja dla PBN i QoS
- Ekstranet
  - autoryzacja i profile partnerów
- Internet
  - handel wirtualny (bezpieczeństwo)

# Potrzeba zapewnienia usług katalogowych

- Dramatyczne zwiększenie rozmiaru sieci:
  - W odseparowanej niewielkiej sieci LAN wystarczą b. proste usługi katalogowe
  - W większych sieciach lokalizacja osób i obiektów zaczyna być problemem
- Rozwój potrzeb:
  - w przeszłości niezbędne było wyłącznie tłumaczenie adresów fizycznych lub sieciowych komputerów na czytelne nazwy oraz książka adresów e-mail
  - obecnie wymagany jest dostęp do wielu atrybutów opisujących obiekty i osoby

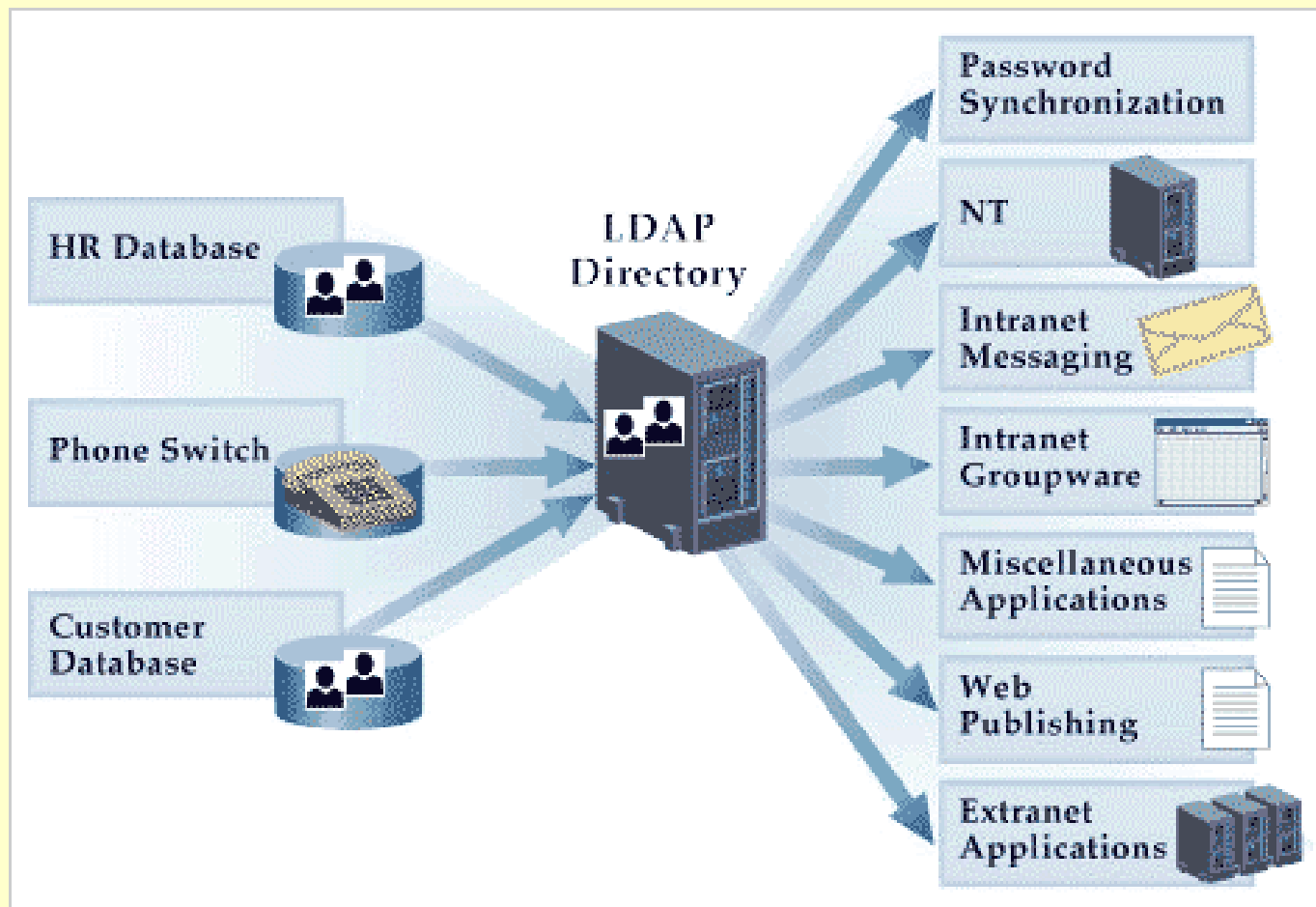
# System informatyczny bez zcentralizowanego serwisu katalogowego

- Każda aplikacja (dział) posiada własny system katalogowy



- Rozwój aplikacji i systemów wymaga rozwiązania zintegrowanego

# System informatyczny ze zcentralizowanym serwisem katalogowym





# "Klienci" usług katalogowych

- **Użytkownicy:** poszukiwanie adresów e-mail, numeru telefonu, itd.
- **Administrator:** jedna "konsola" serwera katalogowego zastępuje wiele "konsol" systemowych
- **Aplikacje:** autoryzacja na podstawie dodatkowych atrybutów
- **Sieci:** realizacja PBN (Policy-based networks) wymaga znajomości profilu osoby

# Usługi katalogowe z historycznego punktu widzenia

- Lokalizacja komputerów
- Określanie adresów e-mail i telefonów
- Aplikacje: serwery WWW, serwery e-mail, serwery aplikacji
- Sprzęt sieciowy (rutery, adaptery sieciowe, modemy)

- 1980

1990

1996

1999 -



# Cztery typy serwisów katalogowych

- Zamknięte
- Oparte na standardach (X.500, LDAP)
- Hybrydowe
- Usługi meta-katalogowe (meta-directory)

# Systemy zamknięte

- Własny standard: katalogu i protokołu komunikacji z klientem
- Zalety: zwykle dość wydajne, choć nie zawsze skalowalne
- Wady: nie umożliwiają rozbudowy i integracji z innymi serwisami
- Obecnie praktycznie w zaniku

# Oparte na standardach

- Całkowicie zgodne z uznanymi standardami (X.500 lub LDAP)
- Zalety: b. dobra integracja z innym oprogramowaniem, dobra skalowalność i wydajność (LDAP)
- Wady: konieczność utrzymania zgodności ze standardem może blokować rozwój cech, których domagają się użytkownicy
- Dynamiczny rozwój (LDAP)

# Hybrydowe

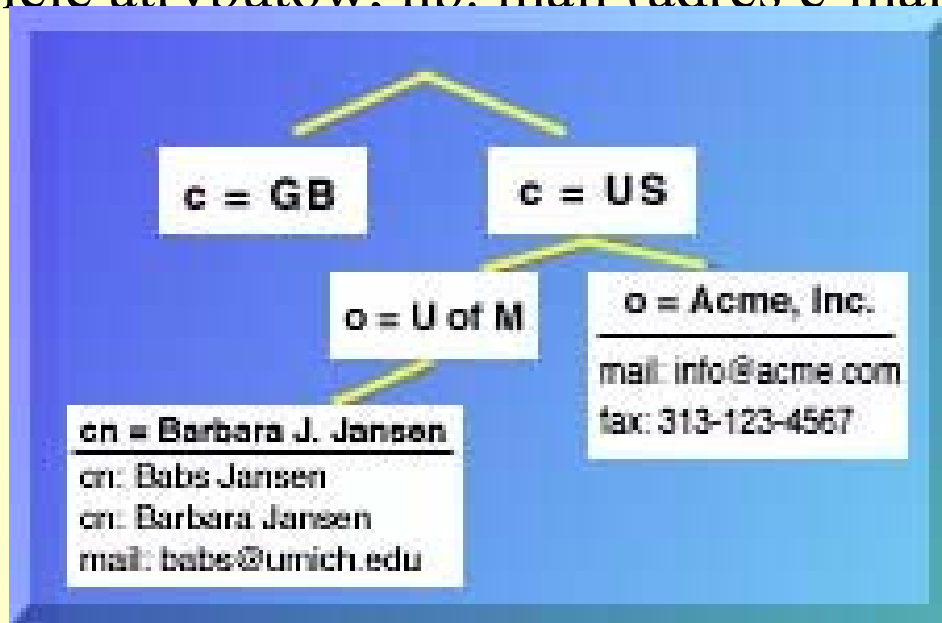
- Łączą zamknięte "jądro" z interfejsem zgodnym ze standardem X.500 i/lub LDAP
- Zalety: dobra integracja z systemem operacyjnym serwera, dość dobra wydajność
- Wady: trudna administracja (podwojenie czynności: katalog "generyczny" i bramka do usług zgodnych ze standardem); przekłamania w bramce X.500/LDAP; ograniczona skalowalność
- Microsoft Exchange, Lotus Notes 5.0, NDS

# Meta-directory

- System zgodny z LDAP lub X.500 replikujący usługi dedykowane
- Zalety: praktycznie jedyne rozwiązanie dla dużych i rozwiniętych systemów, b. dobra skalowalność i wydajność.
- Wady: trudna administracja

# Cechy LDAP

- Standard: RFC-1777, RFC-1778, RFC 1823, RFC 2251-2256
- Serwis TCP/IP, port: 389
- Rekordy (entries) identyfikowane przez DN (*distinguished name*)
- Rekordy posiadają wiele atrybutów. np. mail (adres e-mail)
- Struktura rekordów jest hierarchiczna





# Obiekty katalogu LDAP

- Przykład DN:  
**cn=Grzegorz Blinowski, o=CC,c=PL**
- Atrybuty obiektów mogą być tekstowe lub binarne (kodowane tekstowo) - np. w formacie JPEG
- Typ obiektu określa atrybut *objectclass*  
np. *objectclass=acl*
- Uwaga: DN obiektu nie musi być znany (serwer kat. może obiekt odnaleźć)

# Operacje

- Operacja na katalogu zaczyna się od połączenia (binding) poprzez podanie nazwy domeny bazowej, np. o=CC, c=PL
- Dostępne są operacje: odczytu, wyszukania, zapisu, porównania oraz usuwania obiektów
- Szukanie - do serwera można przekazać filtr określający kryteria szukania (RFC-1558) - operacja przeszukania odbywa się w całości na serwerze

# Kontrola dostępu i bezpieczeństwo

- Katalog może przechowywać atrybuty związane z bezpieczeństwem, np. atrybuty certyfikatu X.509v3
- LDAP może być przenoszony przez SSL
- Katalog może przechowywać listy kontroli dostępu (ACL)

# Implementacja

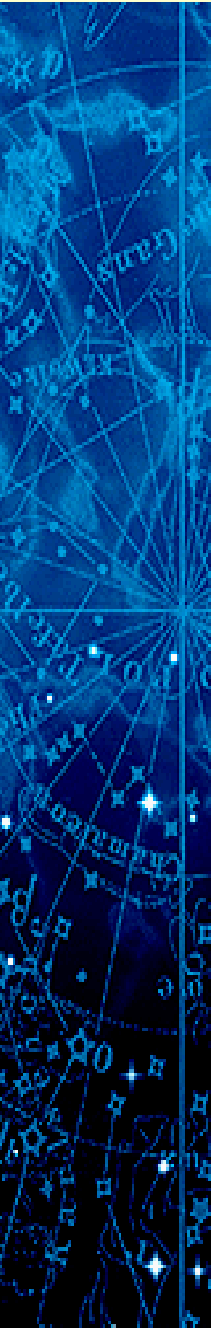
- Serwer:
  - darmowe implementacje dla Unix-a (slpad, ldapd) z uniwersytetu w Michigan
  - Komercyjne: Netscape, Sun
- Klient - dostępne w przeglądarce Netscape Navigator, darmowy klient w źródłach (często w postaci bramki WWW)

# Polityka Firm

- Netscape - własny zgodny ze standardami serwer (Directory Server) zintegrowany z innymi serwerami
- Sun - j.w. - produkt młodszy niż w przypadku Netscape, słabsza integracja z serwerami
- Novell - bramka LDAP do NDS, niejasne plany przeniesienia serwera LDAP Netscape
- Microsoft - bramka LDAP do Active Directory

# Efektywność kosztowa LDAP

- LDAP jest tani w implementacji - protokół bazowy - TCP/IP jest dostępny powszechnie
- Dostępnych jest szereg API dla LDAP
- LDAP jest (stosunkowo) prosty w implementacji oraz przejrzysty dla programisty i administratora
- LDAP jest standardem



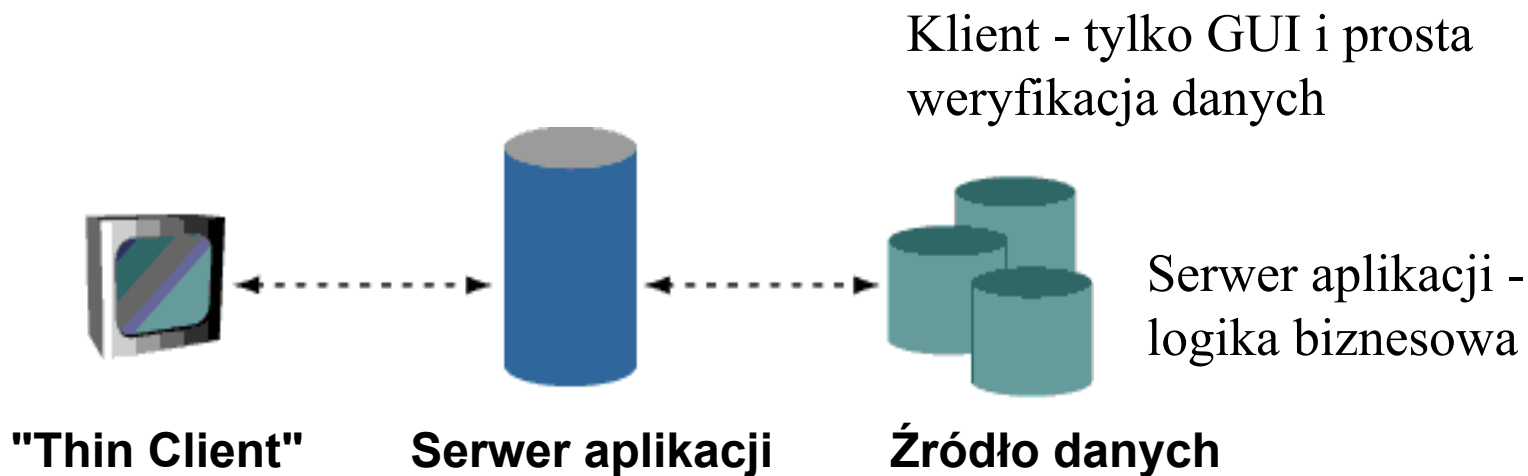
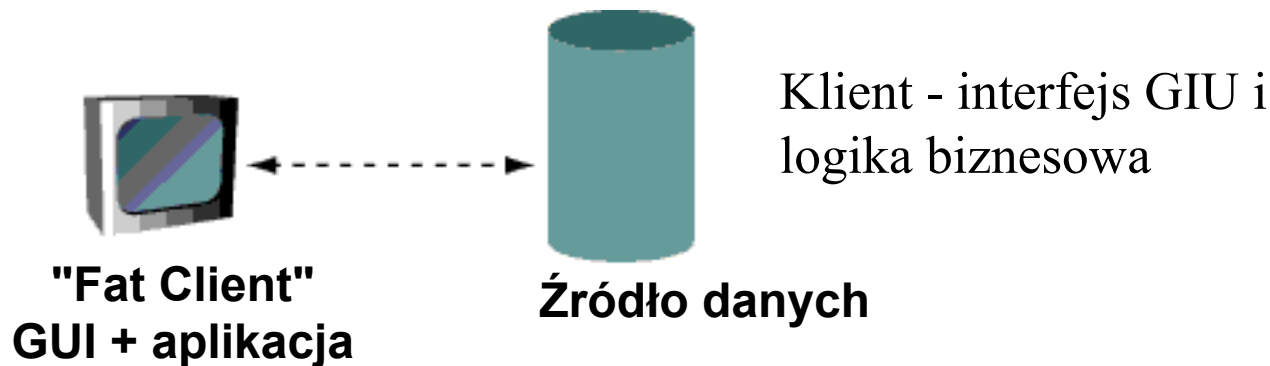
# **Serwery Aplikacji**

# Aplikacje Web

- Aplikacje Web - nowe wcielenie modelu klient-serwer:
  - przeglądarka - klient
  - Serwer: serwer WWW, serwer aplikacji
- Dlaczego aplikacje Web?
  - Tania, dostępna, powszechna, standardowa technologia
  - Możliwość budowy ekstranetu na bazie standardów **internetowych**



# Architektura dwu-, trój- i wielo- warstwowa



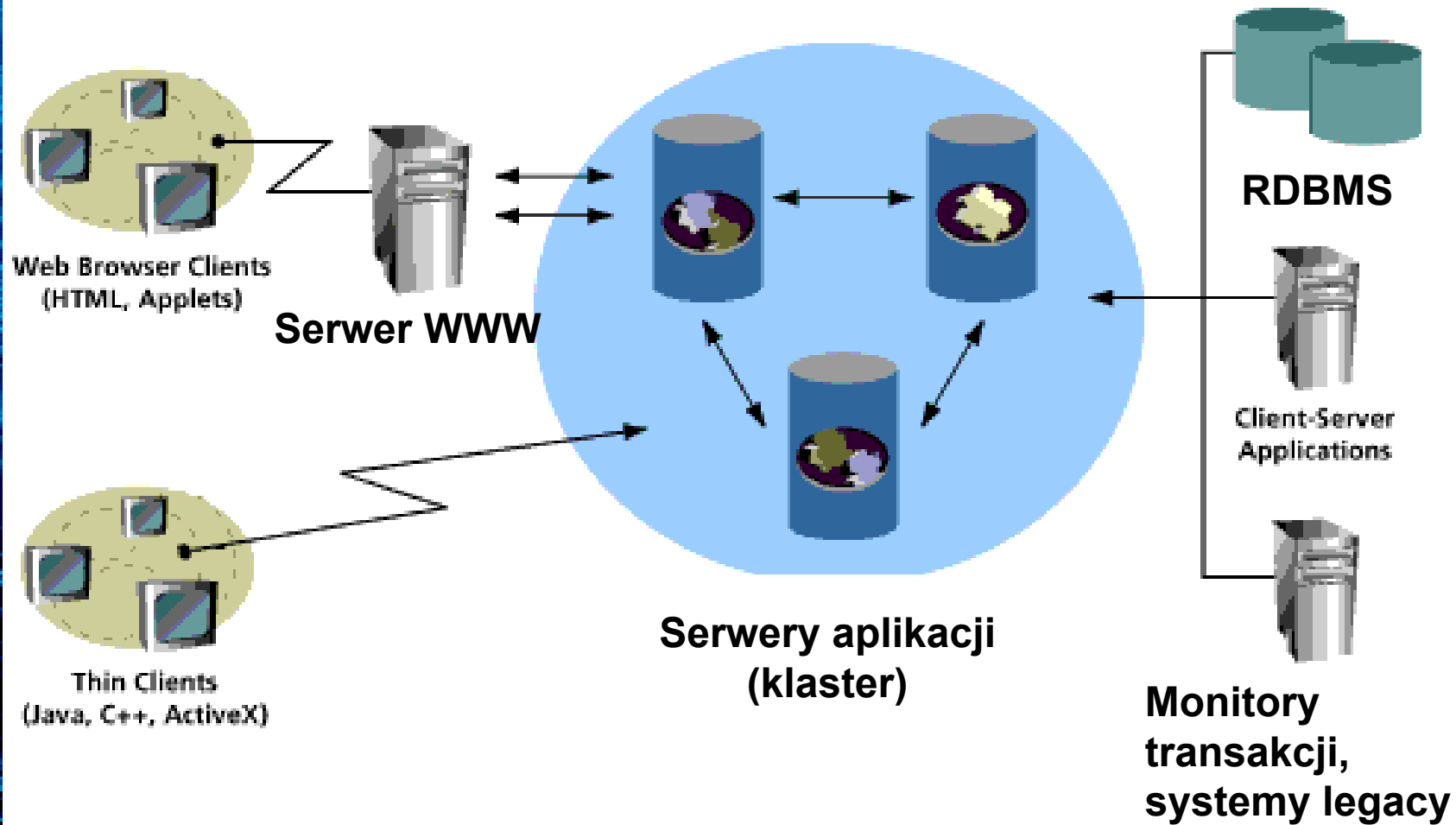
# Klienci serwera aplikacji

- **Klient HTML:**
  - statyczny
  - brak lub minimalna logika (np. JavaScript do weryfikacji formularzy)
  - porozumiewa się za pośrednictwem HTTP
- **Klient Java**
  - aplet java+HTML
  - własna logika
  - porozumiewa się za pośrednictwem IIOP (Internet Inter ORB Protocol) lub JRMI (Java Remote Method Invocation)

# Zalety serwera aplikacji Web

- Niski koszt utrzymania klienta - klientem jest przeglądarka lub applet Java, w przypadku awarii lub złego funkcjonowania reinstalacja jest b. prosta
- Obsługa standardów otwartych - pozwala na wdrożenie intranetu i ekstranetu (nie możemy zakładać, że klienci i partnerzy będą się dostosowywać do naszych rozwiązań)
- Szerokie możliwości generowania i publikowania informacji

# Serwer aplikacji Web



# Wymagania stawiane serwerowi aplikacji

- Integralność transakcji
- Skalowalność, wydajność
- Stabilność
- Otwartość (w sensie obsługi standardów)
- Bezpieczeństwo
- Ochrona inwestycji

# Podstawowe cechy (idealnego) serwera aplikacji

- Obsługuje aplikacje napisane w językach C, C++, Java
- Obsługuje wydajnie transakcyjne relacyjne bazy danych, jest niezależny od producenta bazy
- Oferuje dużą skalowalność (obsługa klastrów)
- Zintegrowane, zcentralizowane zarządzanie
- Obsługuje standardy: HTTP, HTML, CGI, NSAPI (Netscape), ISAPI (Microsoft), Java
- Zawiera środowisko do tworzenia, testowania, wdrażania i zarządzania aplikacjami

# Cechy serwera aplikacji c.d.

- Tworzenie aplikacji:
  - Biblioteki klas (API)
  - środowisko IDE, integracja formularzy, modelu danych i kodu
  - Współpraca z popularnymi narzędziami IDE (JavaSoft, Microsoft, IBM, Borland, itd.)
- Ochrona inwestycji:
  - bramki do innych systemów, Tuxedo, IMS/DC, CICS, MQSeries, itd.





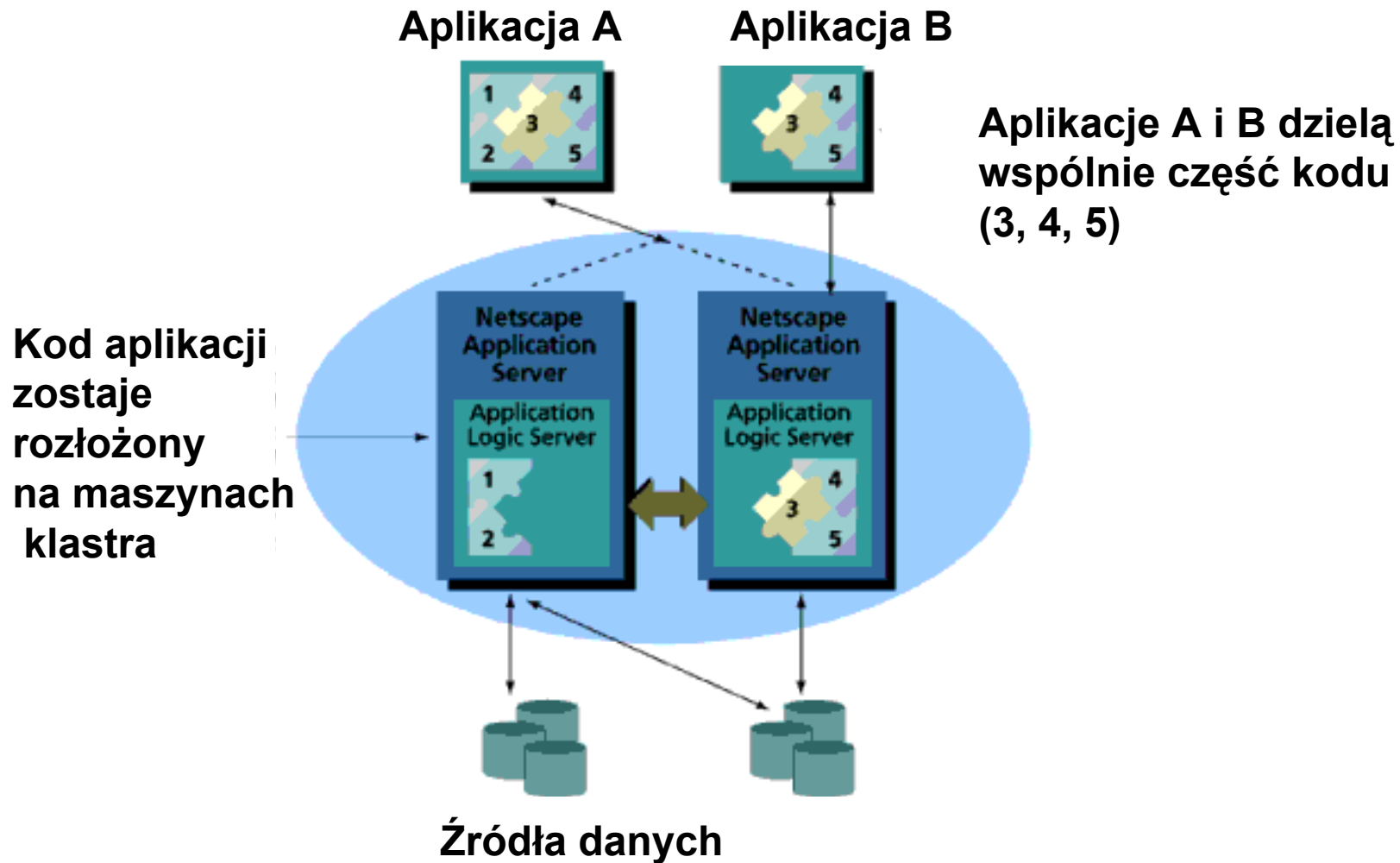
# Cechy serwera aplikacji c.d.

- Cechy API:
  - Zarządzanie sesją
  - Zarządzanie stanem (informacje o kliencie)
  - Obsługa transakcji: auto-commit, rollback, commit
- Skalowalność:
  - Obsługa SMP, wielowątkowość w obrębie jednej maszyny jest zawsze ograniczona
  - Partycjonowanie aplikacji (kodu) i danych
  - Równoważenie obciążenia

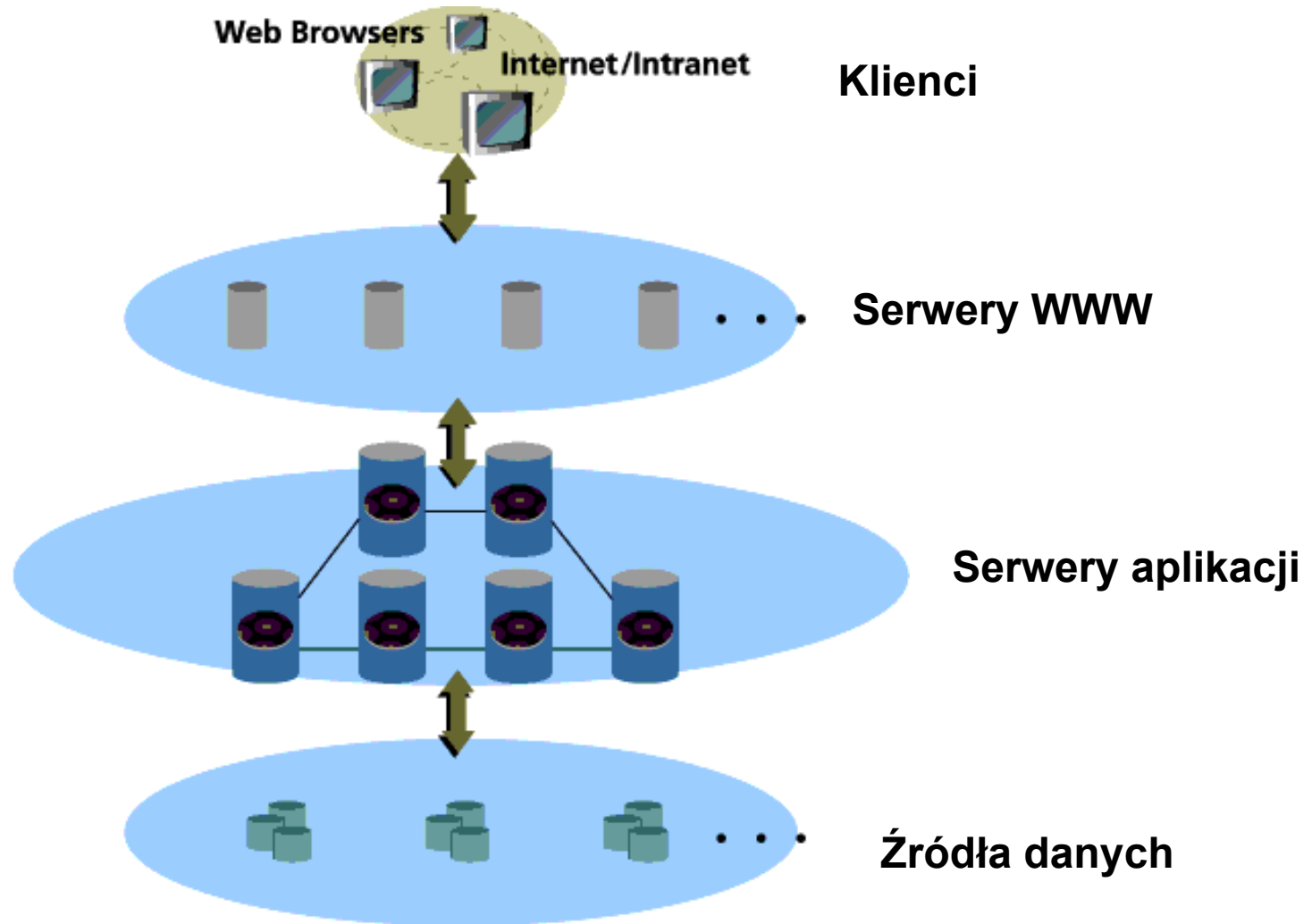
# Cechy serwera aplikacji c.d.

- Stabilność:
  - Autorestart procesów serwera
  - Replikacja danych pomiędzy serwerami w klastrze (serwer działa poprawnie nawet w przypadku awarii składowej klastra)
  - Rekonfiguracja aplikacji bez przerywania pracy
- Bezpieczeństwo:
  - Obsługa SSL
  - Bezpieczne połączenia z bazami danych
  - Logowanie, śledzenie zdarzeń

# Partycjonowanie aplikacji



# Partycjonowanie pozwala nieograniczenie zwiększać moc serwera



# Rynek serwerów aplikacji

- Około 30 firm oferuje serwery aplikacji
- Należy spodziewać się konsolidacji rynku w ciągu następnych 2 lat
- Firmy i produkty:
  - Apple Computer **WebObjects 3.5.1**
  - Bluestone Software **Sapphire/Web 5**
  - NetDynamics, Inc. **NetDynamics 4.1** (kupiony przez Sun)
  - Netscape Communications **Application Server 2.1** (dawniej KIVA)
  - Oracle **Application Server**
  - SilverStream Software Inc. **SilverStream Application Server 1.5**
  - WebLogic **Tengah 3.1**

# Na co zwracać uwagę przy wyborze serwera aplikacji

- Standardy:
  - Zgodność ze standardowymi protokołami (prócz oczywistych): CORBA, DCOM, LDAP
  - Obsługa dynamicznego HTML (DHTML)
  - Programy w C, C++ i Java, obsługa JDK 1.1.5 lub nowszego
  - Obsługa komponentów JavaBeans
  - Obsługa komponentów Enterprise JavaBeans

# Na co zwracać uwagę przy wyborze serwera aplikacji c.d.

- Środowisko IDE
  - Debugger
  - Wsparcie dla pracy grupowej nad projektem, kontrola wersji kodu źródłowego
- Inne programistyczne
  - Obsługa "niestandardowych" systemów i protokołów, CICS, SAP, Tuxedo, itd.
  - Obsługa poczty elektronicznej
  - Współpraca z narzędziami IDE innych producentów

# Na co zwracać uwagę przy wyborze serwera aplikacji c.d.

- Serwisy bazodanowe:
  - Natywne sterowniki: Oracle, Informix, Sybase, MS SQL, DB2
  - Obsługa ODBC na platformach NT i Unix
  - Obsługa JDBC
  - Obsługa procedur wbudowanych (stored procedures)



# Na co zwracać uwagę przy wyborze serwera aplikacji c.d.

- Skalowalność
  - Wielowątkowość
  - Automatyczny restart aplikacji
  - Partycjonowanie aplikacji
  - Równoważenie obciążenia w klastrze
- Stabilność
- Pełna replikacja kodu i danych
- Administracja
  - zdalna konsola

# Na co zwracać uwagę przy wyborze serwera aplikacji c.d.

- Klienci:
  - HTML, Java, CORBA/IOP
- Bezpieczeństwo:
  - SSL, integracja z mechanizmami systemu operacyjnego
- Obsługa platform: Unix, Windows NT
- **Cena:**
  - Różne metody ustalania ceny, np. za stanowisko klienta, za procesor
  - Wydatek rzędu 10 - 50 i więcej tysięcy USD