

Architektury rozproszonych baz danych

Robert A. Kłopotek

r.klopotek@uksw.edu.pl

Wydział Matematyczno-Przyrodniczy. Szkoła Nauk Ścisłych, UKSW

Rozproszona baza danych

- ▶ Rozproszoną bazą danych (RBD) nazywamy:
Logicznie połączone ze sobą zasoby współdzielonych danych, które fizycznie rozproszone są pomiędzy węzłami w sieci komputerowej.
- ▶ System zarządzania rozproszoną bazą danych (SZRBD) DDBMS (ang. Distributed Database Management System) definiowany jest jako:
Oprogramowanie umożliwiające zarządzanie rozproszoną bazą danych, w sposób, w którym rozproszenie danych jest niewidoczne dla użytkowników.
- ▶ Dokonując interpretacji definicji można stwierdzić, że DDB jest po prostu rozproszonym systemem plików DSF (ang. Distributed File Systems)

Różnice pomiędzy DDB i DFS (1)

- ▶ **DFS** zapewnia użytkownikom prosty interfejs dostępu do plików rozmieszczonych na komputerach połączonych w sieć. Pliki te mają prostą strukturę (zazwyczaj są płaskie). Powiązania pomiędzy kilkoma plikami (o ile w ogóle istnieją) nie są zarządzane przez system, pozostawiając to użytkownikom.
- ▶ W przeciwieństwie do tego, **DDB** są zorganizowane zgodnie ze schematem, który definiuje strukturę danych oraz relacje między nimi. Za zarządzanie strukturą oraz relacjami odpowiedzialny jest system.

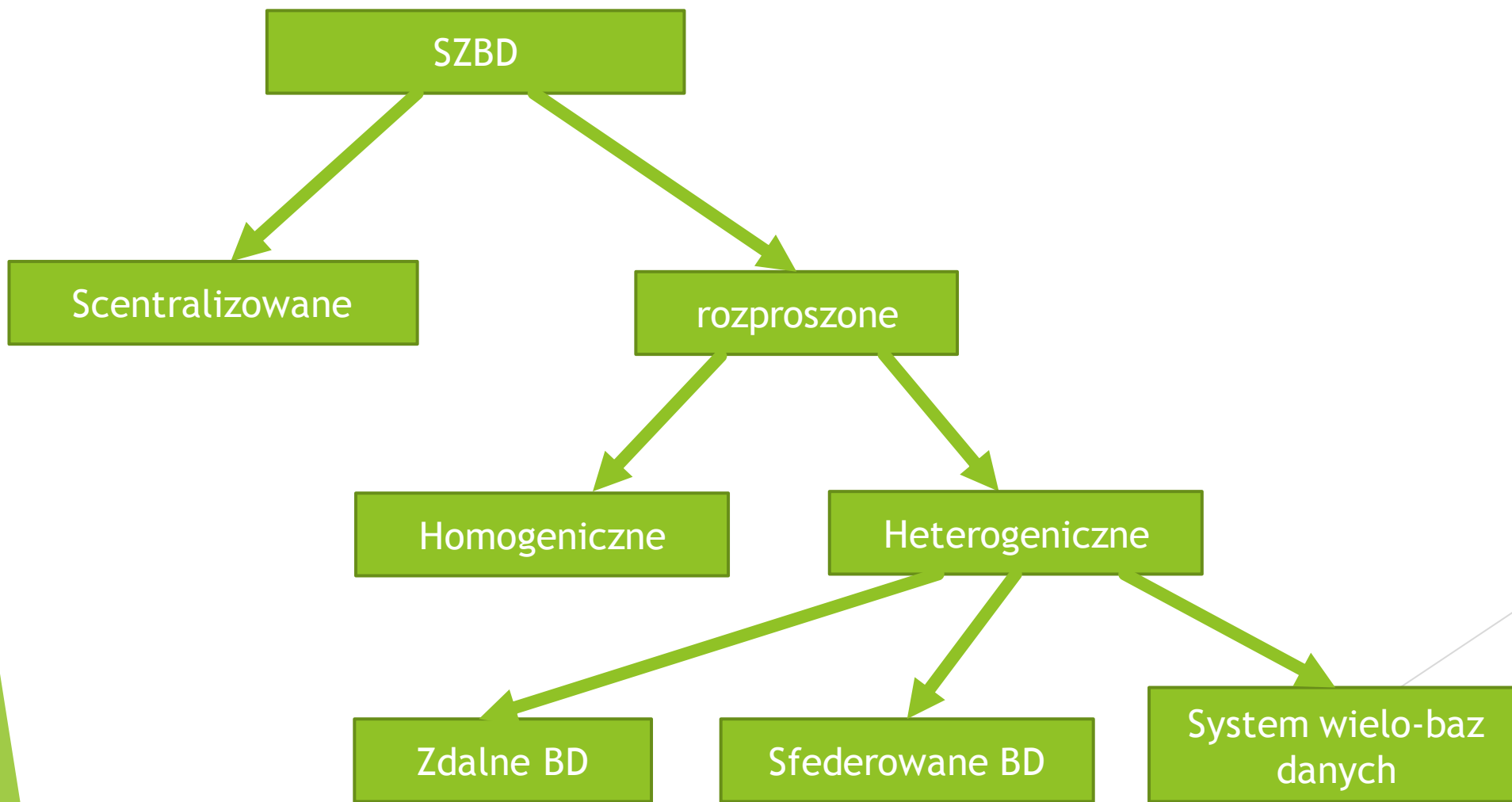
Różnice pomiędzy DDB i DFS (2)

- ▶ Interfejs użytkownika w **DFS** umożliwia jedynie wykonywanie podstawowych operacji na plikach, typu: otwarcie, odczyt lub zapis oraz zamknięcie pliku.
- ▶ **DDB** natomiast posiadają pełną funkcjonalność centralnej bazy danych. Oznacza to, że umożliwiają wykonywanie operacji na danych z użyciem języków wysokiego poziomu (np. SQL lub OQL ang. Object Query Language), wsparcie dla transakcji (zbioru powiązanych zapytań) oraz implementację mechanizmów wielodostępu i odzyskiwania danych.

Różnice pomiędzy DDB i DFS (3)

- ▶ W **DDB** zapewniona jest przezroczystość dostępu do danych, co oznacza, iż użytkownik nie zdaje sobie sprawy z tego, że dane są rozmieszone na innych komputerach niż jego.
- ▶ W **DFS** użytkownik musi znać rozmieszczenie plików.

Typy rozproszonych baz danych



Systemy homogeniczne vs heterogeniczne

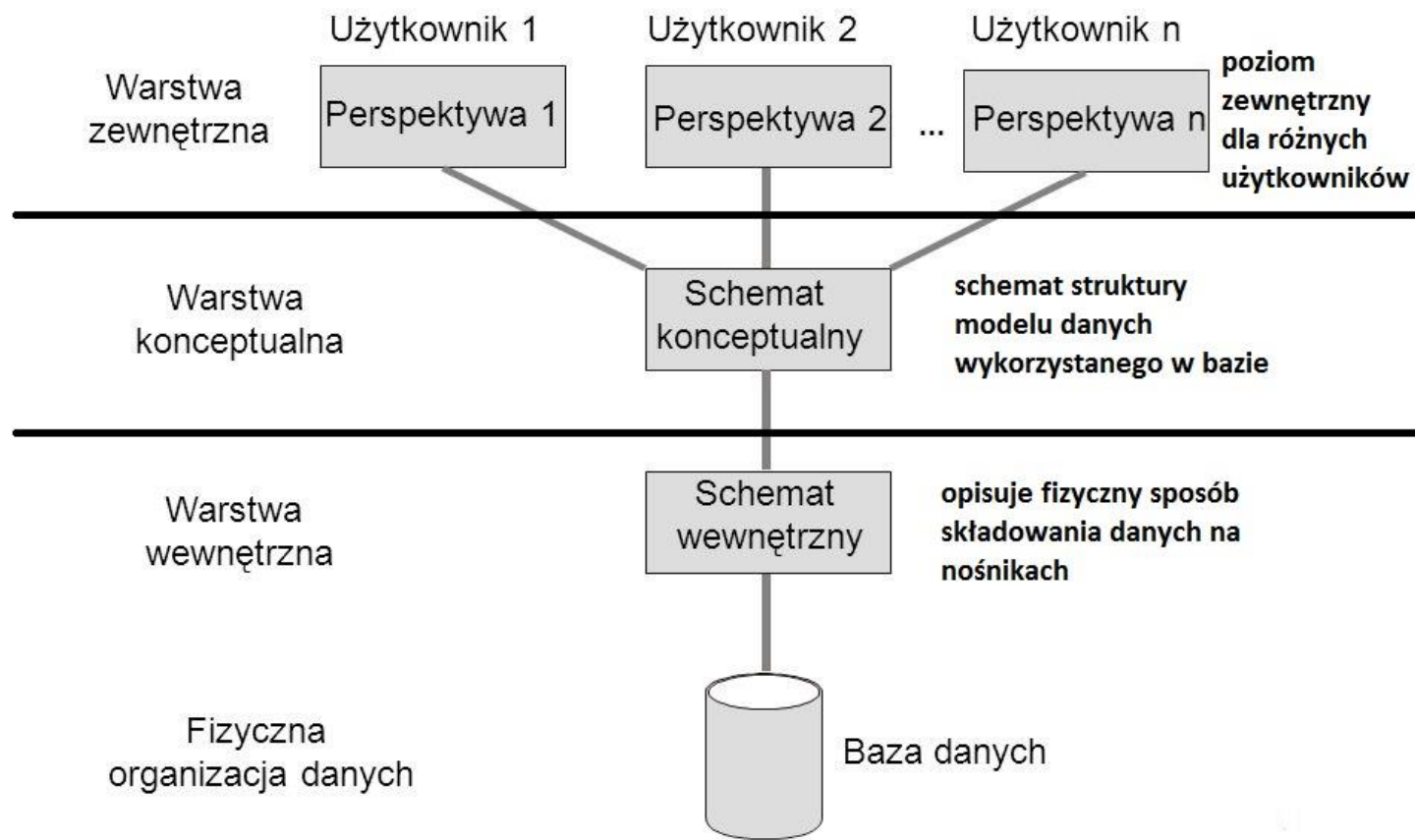
- ▶ Jeśli wszystkie serwery (lub indywidualne lokalne SZBD) i klienci wykorzystują identyczne oprogramowanie, to SZRBD jest nazywany systemem homogenicznym
- ▶ System zarządzania heterogeniczną rozproszoną bazą danych (SZHRBD) jest systemem integrującym istniejące, rozwijane autonomicznie, heterogeniczne systemy baz danych, nazywane lokalnymi systemami baz danych (LSBD)

Systemy heterogeniczne

- ▶ **Heterogeniczność *LSBD*** wchodzących w skład *SHRBD* oznacza, że systemy te stosują różne interfejsy użytkowe, różne modele danych na których oparte są te systemy, różne języki dostępu, systemy te mogą różnić się semantyką i syntaktyką przechowywanych danych oraz mechanizmami zarządzania współbieżnością.
- ▶ **Problem lokalnej autonomii:** Jeśli wszystkie operacje dostępu do *SZRBD* muszą odbywać się poprzez „klientów”, to taki system nie ma lokalnej autonomii, natomiast jeśli jest możliwe wykonywanie bezpośredniego dostępu do danych przez lokalne transakcje (z pominięciem *SZRBD*), to jest to system z lokalną autonomią
- ▶ **Systemy bez lokalnej autonomii** - klasyczne systemy rozproszone
- ▶ **Systemy z lokalną autonomią** - systemy sfederowanych baz danych i systemy wielo-baz danych

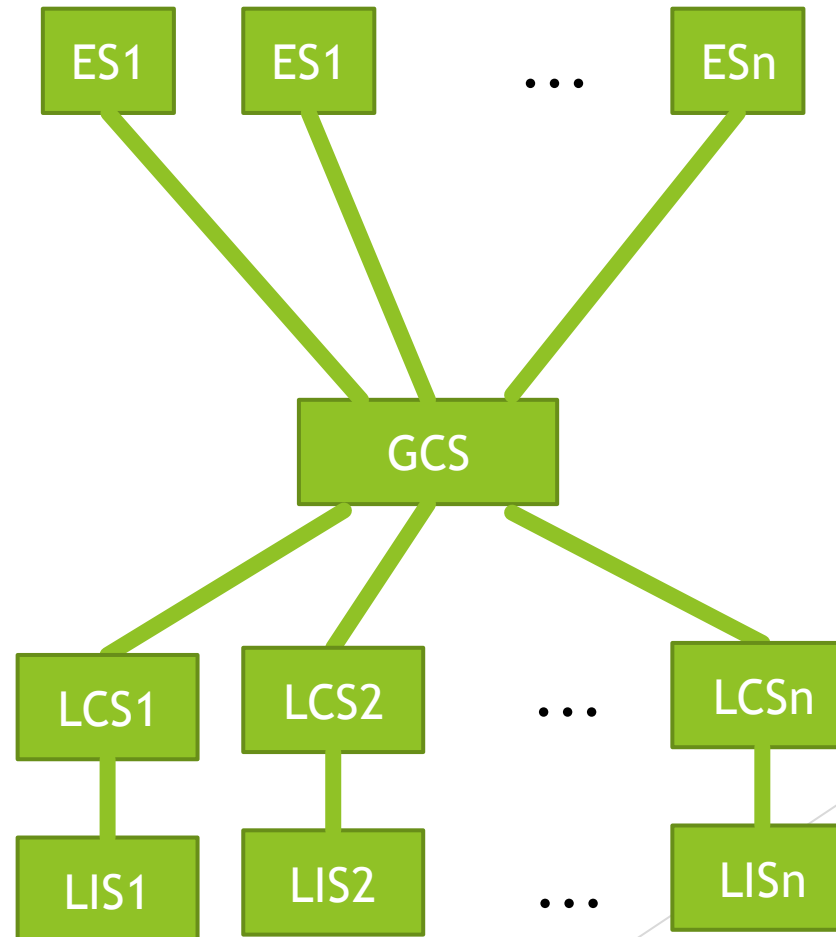
Architektura ANSI/SPARC

Architektura trójwarstwowa ANSI-SPARC



Architektura SZRBD zalecana przez ANSI

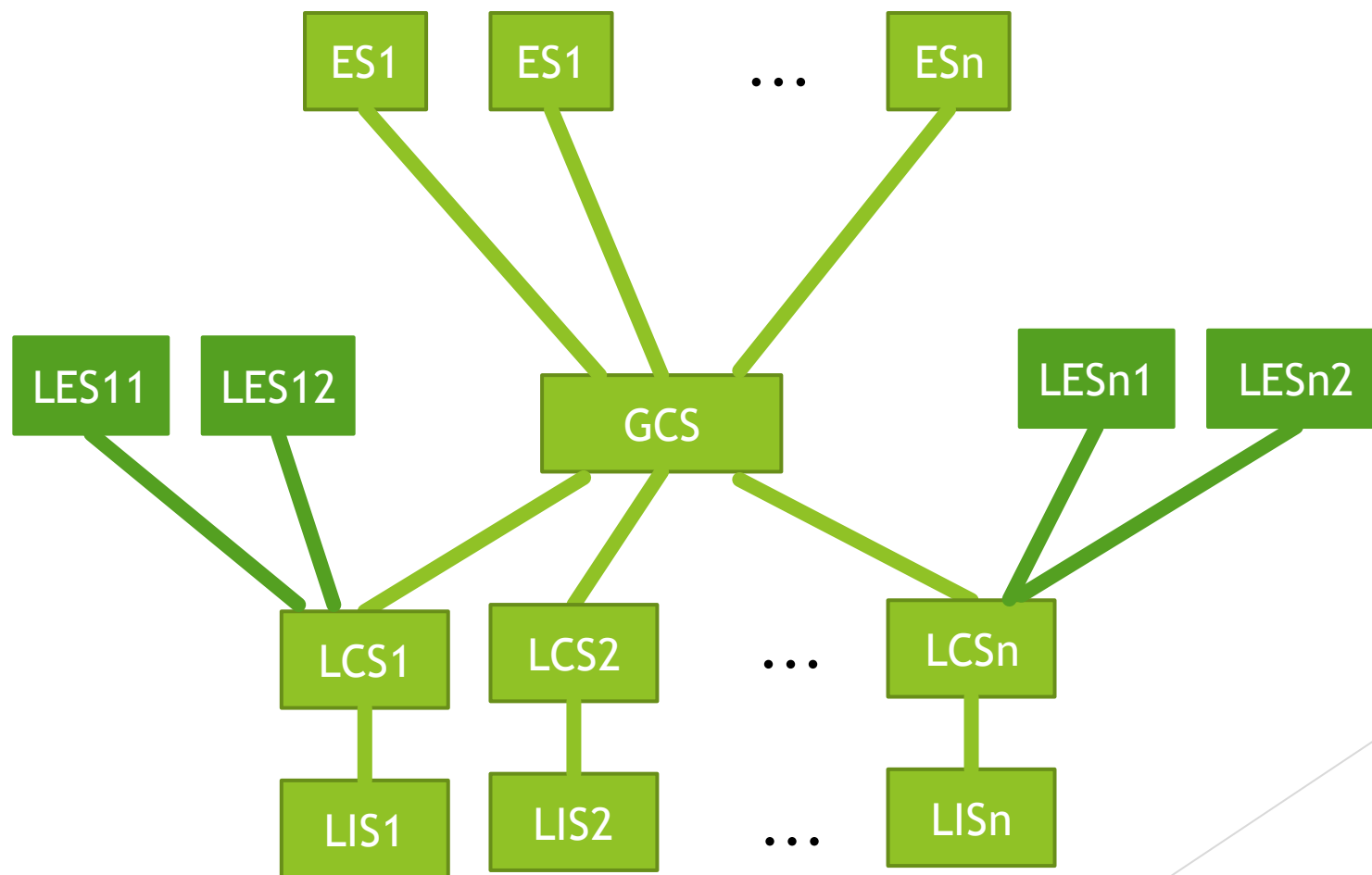
- ▶ Bazuje na idei tworzenia globalnego schematu koncepcyjnego w oparciu o schematy lokalne
- ▶ Integracja schematów to proces syntezy pojedynczego schematu globalnego na podstawie 2 lub więcej schematów lokalnych
- ▶ **Legenda:**
 - ▶ ES1 - External Schema
 - ▶ GCS - Global Conceptual Schema
 - ▶ LCS1 - Local Conceptual Schema
 - ▶ LIS1 - Local Internal Schema



System sfederowanych baz danych

- ▶ **System sfederowanych baz danych** to system składający się z co najmniej dwóch niezależnych, różnych systemów baz danych oraz odpowiedniego mechanizmu konsolidującego wszystkie ich komponenty
- ▶ Każdy serwer jest niezależnym i autonomicznym scentralizowanym SZBD, który ma swoich własnych lokalnych użytkowników.
- ▶ Każdy serwer może autoryzować dostęp do określonej porcji bazy danych przez wyspecyfikowanie schematu eksportowego, który określa część bazy danych, do której mogą mieć dostęp użytkownicy globalni.
- ▶ Użytkownik globalny poprzez GCS (tworzony na podstawie schematów eksportowych) posiada w ograniczonym zakresie informacje co w każdej bazie danych się znajduje.

Architektura systemu sfederowanych baz danych



Systemy sfederowanych baz danych - komentarz

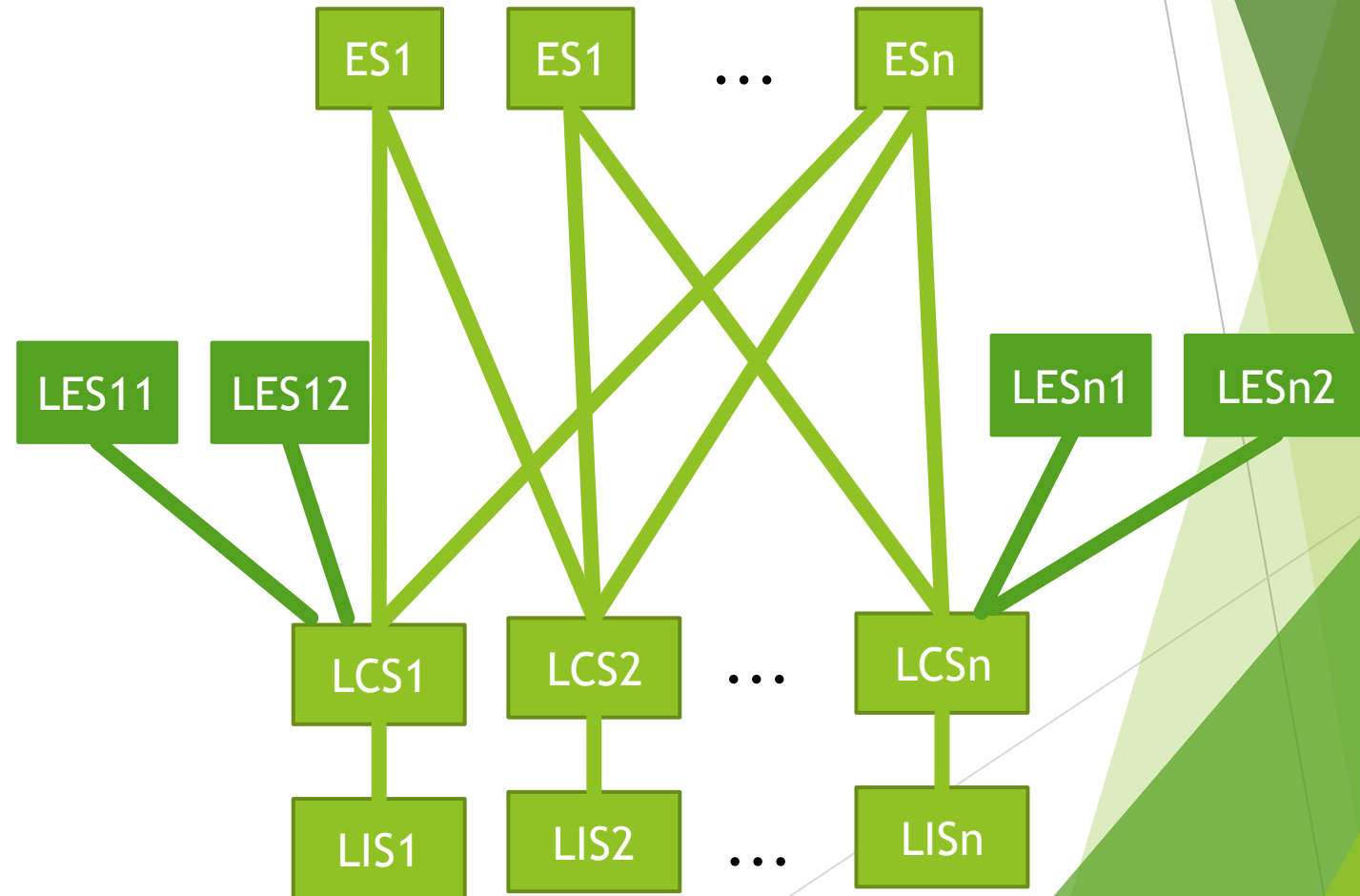
- ▶ System taki jest hybrydą pomiędzy systemem scentralizowanym i rozproszonym
- ▶ Poszczególne systemy baz danych należące do systemu sfederowanego są całkowicie niezależne od siebie i mogą pracować jako samodzielne jednostki
- ▶ Wykorzystywanie systemów sfederowanych może mieć niemalże nieograniczony zasięg
- ▶ Integracją komponentów systemu sfederowanego mogą zajmować się użytkownicy we współpracy z administratorami i projektantami aplikacji

Systemy sfederowanych baz danych - problemy

- ▶ Konieczność wielokrotnego tłumaczenia formatów danych
- ▶ Realizacja transakcji, która wymaga dostępu do innych baz danych, wiąże się z koniecznością nie tylko pobrania danych, ale również dopasowania do schematu bazy danych, w której transakcja jest przetwarzana
- ▶ Podstawowe problemy związane z zarządzaniem sfederowanymi bazami danych:
 - ▶ propagacja modyfikacji z poziomu globalnego na poziomy lokalne,
 - ▶ spójność bazy danych na poziomie globalnym.

System wielo-baz danych

- ▶ **System wielo-baz danych** (ang. *multidatabase system*) jest zbiorem autonomicznych i heterogenicznych baz danych, które umożliwiają realizację pewnych operacji globalnych, do których wykonania jest konieczne niezależne i asynchroniczne wykonanie zbioru transakcji w ogólności w różnych bazach danych, będących składnikami wielobazy.



Systemy wielo-baz danych - przykład

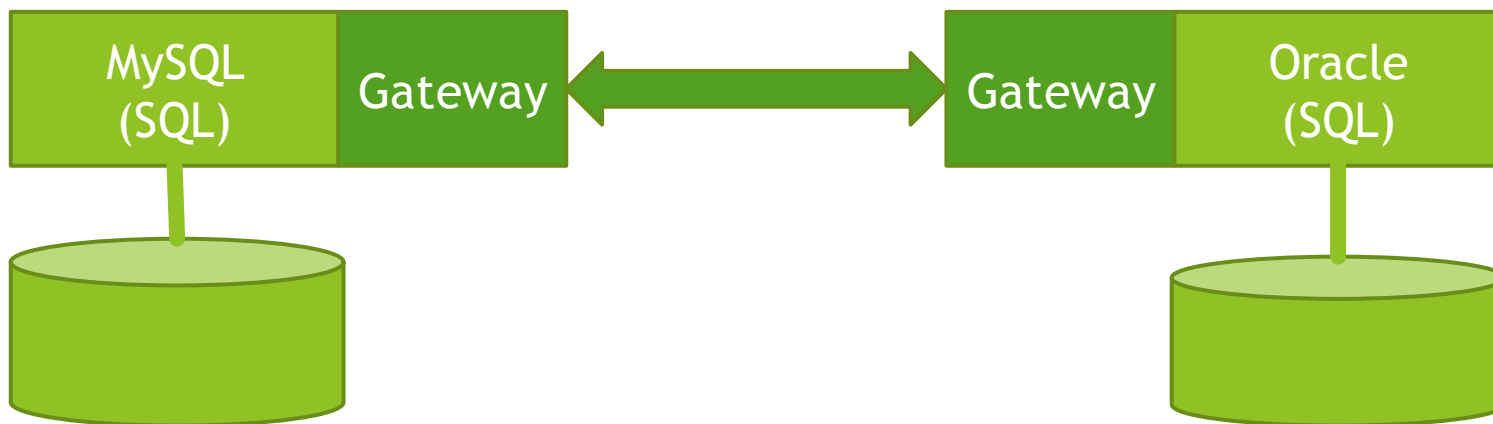
- ▶ System rezerwacji połączeń komunikacyjnych różnymi środkami lokomocji między dowolnymi miastami
- ▶ Wielo-baza danych składa się z następujących baz danych: rezerwacji połączeń lotniczych, rezerwacji połączeń kolejowych, rezerwacji i wynajmu samochodów, rezerwacji hotelowej.
- ▶ Użytkownik wielo-baza danych *Kowalski* planuje dotarcie z Poznania do Zanzibar Town szukając najszybszego i najbardziej komfortowego połączenia, przy jednoczesnym dopasowaniu go do jego ograniczeń czasowych

Systemy wielo-baz danych - charakterystyka

- ▶ Poszczególne bazy danych są wzajemnie niezależne, tzn. że zawarte w nich dane nie mają wpływu na zawartość i transakcje realizowane w innych bazach danych
- ▶ Podobnie jak w systemach sfederowanych wyróżnia się użytkowników lokalnych i globalnych
- ▶ Ze względu na autonomiczność tworzących wielo-bazę węzłów zakłada się, że każdy z nich musi być w stanie spójnym, natomiast stan całej wielo-bazy w ogólności nie jest spójny
- ▶ W celu realizacji zleceń globalnych w wielo-bazach wykorzystuje się tzw. transakcje kooperujące (ang. cooperating transactions) - realizowane w różnych węzłach wielo-bazy i przekazujące między sobą wyniki. Na przykład, rezerwacja lotnicza może być parametrem rezerwacji kolejowej

Bramy i zdalne bazy danych (ang. gateways and remote databases)

- ▶ **Zdalna baza danych** jest to baza ulokowana na stanowisku komputerowym innym niż stanowisko na którym pracuje dany użytkownik - użytkownik jest świadom, że pracuje ze zdalną bazą danych
- ▶ Konieczny jest specjalny interfejs do zdalnego heterogenicznego systemu bazy danych nazywany **bramą**
- ▶ Jest to odmienne od koncepcji systemów rozproszonych, sfederowanych i wielo-baz.



Standardy łączenia rozproszonych danych (1)

- ▶ **Open DataBase Connectivity (ODBC):** standard [zdalnego] dostępu do relacyjnych baz danych bazuje na Call Level Interface (CLI) opracowanym przez konsorcjum X/Open definiuje API oraz cechy SQL które muszą być zapewnione na różnych poziomach zgodności.
- ▶ **Java DataBase Connectivity (JDBC):** analogiczny do ODBC standard dla Java.
- ▶ **OLE-DB:** API podobne do ODBC, ale wspomagające źródła nie-bazodanowe, takie jak płaskie pliki. OLE-DB program może negocjować ze źródłem danych aby znaleźć właściwości, które ono podtrzymuje. API jest podzbiorem SQL
- ▶ **ADO (Active Data Objects):** łatwy interfejs do funkcji OLE-DB

Standardy łączenia rozproszonych danych (2)

- ▶ **Kilka standardów bazujących na XML dla E-commerce**
 - ▶ Np. RosettaNet (łańcuchy dostaw),
 - ▶ BizTalk Definiują katalogi, opisy usług, faktury, zamówienia, itd.
 - ▶ osłony XML są używane do eksportu informacji z relacyjnej BD do XML
- ▶ **Resource Description Framework (RDF):** specyfikacja ontologii dla zasobów Web.
- ▶ **Web Services i Simple Object Access Protocol (SOAP):** bazujący na XML standard dla zdalnego wołania usług.
 - ▶ SOAP jest mniej elastyczny i uniwersalny w stosunku do CORBA.
 - ▶ Używa XML do zakodowania danych, HTTP jako protokołu transportowego
 - ▶ Kilka dalszych standardów: WSDL (opis danych i usług), UDDI (rejstry usług), itd. Dalsze standardy są oparte na SOAP dla specyficznych aplikacji, np. OLAP i Data Mining (standardy Microsoft'u)

Replikacja

- ▶ **Replikowanie danych** - w systemie utrzymywanych jest wiele kopii tych samych danych (relacji lub fragmentów relacji)
- ▶ **Cel:**
 - ▶ zwiększenie niezawodności systemu (uszkodzenie określonej kopii lub stanowiska nie musi oznaczać załamania się całego systemu)
 - ▶ dostęp do określonego pliku danych jest możliwy nawet w przypadku awarii serwera na którym ten plik był przechowywany
 - ▶ równomierne rozłożenie obciążenia pomiędzy poszczególne serwery i równoległe wykonywanie zadań, a tym samym zwiększenie efektywności pracy całego systemu.
 - ▶ zwiększenie efektywności operacji dostępu do danych - system może wybrać kopię, która jest usytuowana najbliżej stanowiska użytkownika

Metody replikowanie danych

- ▶ **replikacja pełna** - fully replicated distributed database - na każdym stanowisku jest przechowywana cała baza danych
 - ▶ maksymalizacja dostępu do danych
 - ▶ wysoka efektywność wykonywania zapytań globalnych
 - ▶ drastyczne spowolnienie transakcji aktualizacyjnych
- ▶ **replikacja częściowa** - partially replication - pewne fragmenty bazy danych mogą być replikowane, inne - nie.
 - ▶ liczba kopii relacji może się zmieniać od 1 do liczby stanowisk w systemie
 - ▶ fakt replikowania plików musi być transparentny dla użytkownika - rozszerzony mechanizm nazywania plików

Sposoby wykonywania kopii fragmentów bazy danych

- ▶ **Explicit replication** - kopia danych jest najpierw tworzona na lokalnym serwerze, a następnie na kolejnych
 - ▶ wykonywanie operacji modyfikacji bazy danych jest realizowane w ten sam sposób
 - ▶ podczas realizacji zapytań, wykonywane są próby dostępu do poszczególnych kopii danych, tak długo aż zostanie znaleziona kopia dostępna.
- ▶ **Lazy replication** - tworzenie kolejnych kopii na innych stanowiskach, gdy „serwer ma trochę czasu”
 - ▶ problemy z utrzymywaniem spójności bazy danych

Wady replikacji

- ▶ Konieczna dodatkowa przestrzeń dyskowa
- ▶ Możliwość powstania niespójności pomiędzy repliką i oryginałem - zagrożenie dla spójności procesów biznesowych
 - ▶ Np. jeżeli dane o wysokości kont są powtórzone w wielu miejscach, wówczas ktoś korzystając z (celowo) uszkodzonych linii transmisyjnych może dokonać nielegalnych wypłat.
- ▶ Zwiększenie kosztów aktualizacji danych:
 - ▶ aktualizacja oryginału pociąga za sobą dodatkowy koszt aktualizacji replik

Rozproszone zarządzanie transakcjami bazujące na wyróżnionej kopii danych

- ▶ Główna idea - wyznaczenie w systemie wyróżnionej kopii danych
- ▶ Blokady danych są związane z wyróżnioną kopią - wszystkie żądania założenia lub zdjęcia blokad są wysyłane na stanowisko, które przechowuje tą kopię
- ▶ Szereg metod bazuje na tej idei, ale różnią się one metodą wyboru kopii wyróżnionej

Pytania?