

Архітектура та технології клієнт- сервер

{

Особливості клієнт-серверної архітектури

- Як правило комп'ютери та програми, що входять до складу інформаційної системи, не є рівноправними.
- Деякі з них володіють ресурсами (файлова система, процесор, принтер, база даних і т.д.), інші мають можливість звертатися до цих ресурсів.
- *Комп'ютер (або програму), що керують ресурсом, називають сервером цього ресурсу (файл-сервер, сервер бази даних, обчислювальний сервер).*
- Клієнт і сервер якого-небудь ресурсу можуть перебувати як на одному комп'ютері, так і на різних комп'ютерах, пов'язаних мережею.

Три групи функцій для вирішення різних підзадач в клієнт-серверній системі

- ▣ .Функції введення і відображення даних (забезпечують взаємодію з користувачем);
- ▣ 2.Прикладні функції, характерні для даної предметної області;
- ▣ 3. Службові функції.

Мережевий додаток	Представлення даних
	Прикладна логіка
	Доступ до ресурсі

які можна представити у вигляді взаємопов'язаних компонентів (рис. 2).

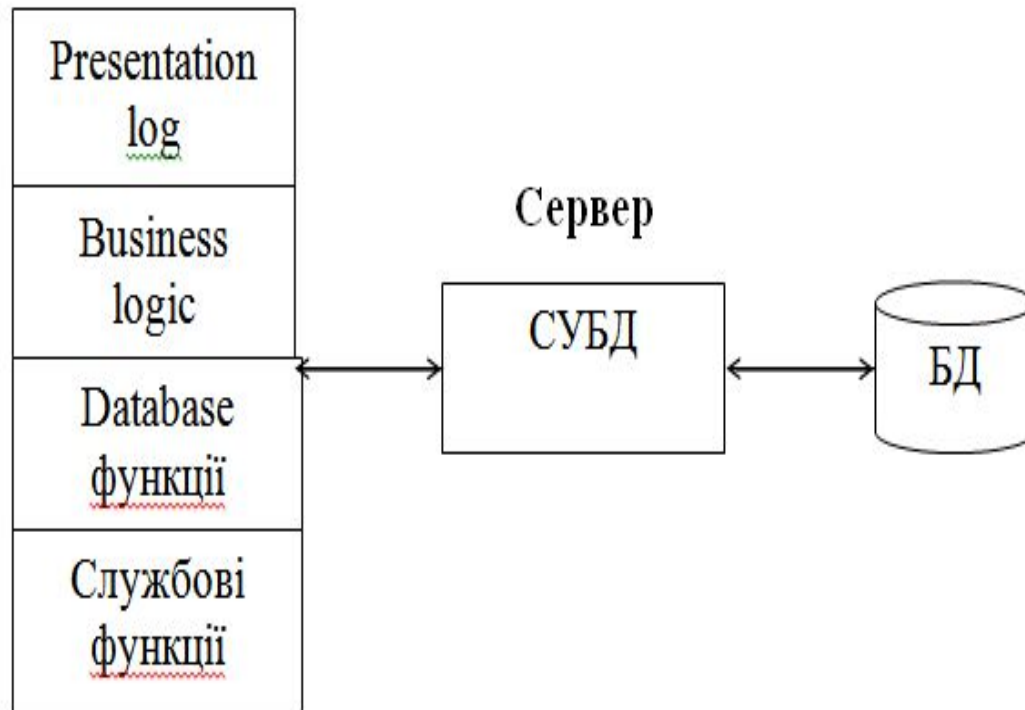


Рис. 2. Структура типового інтерактивного додатку, що працює з базою даних

- Де:
- Presentation logic - компонента подання даних, що відповідає за інтерфейс користувача;
- Business logic - компонента прикладної логіки реалізує алгоритм вирішення конкретного завдання;
- Database функції - компонента управління ресурсом, що забезпечує доступ до необхідних даних.
- Автономна система (комп'ютер, не підключений до мережі) представляє всі ці компоненти як на різних рівнях (ОС, службове ПО і утиліти, прикладне ПЗ), так і на рівні додатків.
- Так само і мережа - вона представляє всі ці компоненти, але, в загальному випадку, розподілені між вузлами. Завдання зводиться до забезпечення мережної взаємодії між цими компонентами.

Поняття про клієнт-серверну архітектуру та технології

- Архітектура «клієнт-сервер» визначає загальні принципи організації взаємодії в мережі, де є *сервери* - вузли-постачальники деяких специфічних функцій (сервісів) і *клієнти*, споживачі цих функцій.
- Практичні реалізації такої архітектури називаються *клієнт-серверними технологіями*. Кожна технологія визначає власні або використовує наявні *правила* взаємодії між клієнтом і сервером, які називаються *протоколом обміну* (*протоколом взаємодії*).

Дволанкова архітектура

- Дволанковою (two-tier, 2-tier) вона називається через необхідність розподілу трьох базових компонентів між двома вузлами (клієнтом і сервером).
- Дволанкова архітектура використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому використовуючи лише власні ресурси.
- Тобто сервер не викликає сторонні мережеві програми, не звертається до сторонніх ресурсів для виконання будь-якої частини запиту

Моделі дволанкової архітектури

- Розташування компонентів на стороні клієнта або сервера визначає такі основні моделі їх взаємодії в рамках дволанкової архітектури:
- *-сервер терміналів - розподілене представлення даних;*
- *-файл-сервер - доступ до віддаленої бази даних і файлових ресурсів;*
- *-сервер БД - віддалене представлення даних;*
- *-активний сервер БД – обробка даних на сервері.*

Мейнфреймова архітектура

- Історично першою з'явилася модель розподіленого представлення даних (модель сервер терміналів).
- Вона реалізовувалася на універсальній ЕОМ (*мейнфрейм*), що виступала в ролі сервера, з підключеними до неї алфавітно-цифровими терміналами.
- Користувачі виконували введення даних з клавіатури терміналу.
- Дані, по провідних каналах зв'язку, передавалися на мейнфрейм і там виконувалася їх обробка, включаючи формування зображення з результатами.
- Це зображення і поверталось користувачеві на екран терміналу.

Файл –серверна архітектура

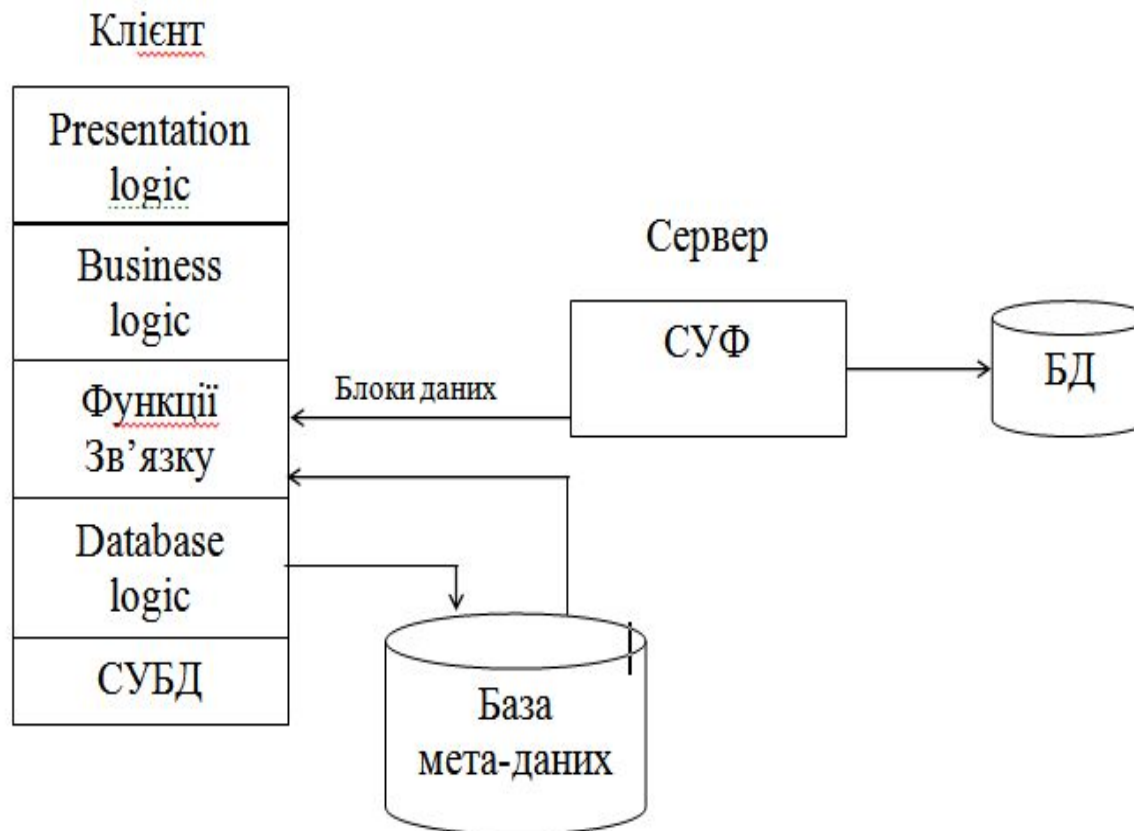


Рис. 3. Модель файлового сервера

- Відповідно до запитів користувачів файли з файл-сервера передаються на робочі станції користувачів, де здійснюється основна обробка даних.
- Централізований сервер виконує, в основному, лише роль сховища файлів, не приймаючи участі в обробці самих даних. Після завершення роботи користувачі копіюють файли з обробленими даними назад на сервер, звідки їх можуть копіювати і обробляти інші користувачі.
- Файл-серверна архітектура має ряд недоліків, а саме, при одночасному зверненні до одних тих самих даних користувачі змушені очікувати вивільнення даних, в результаті чого різко падає продуктивність роботи інформаційної системи.
- В протилежному випадку можливе взаємне затирання змін зроблених різними користувачами.
- Частковим рішенням є підтримка тиражування (реплікації) таблиць і запитів. У цьому випадку, наприклад при зміні даних, оновлюється не вся таблиця, а тільки модифікована її частина.

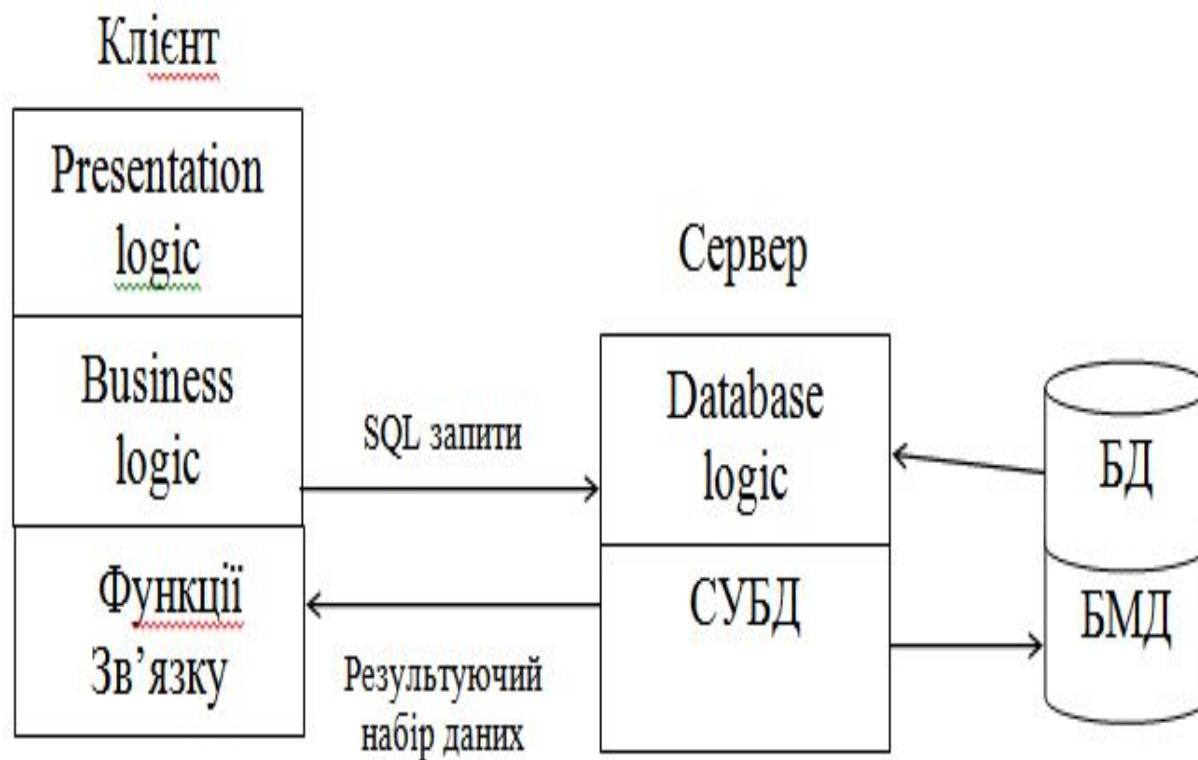


Рис. 4. Модель віддаленого доступу (RDA)

Модель сервера баз даних

- З появою спеціалізованих СУБД з'явилася можливість реалізації іншої моделі доступу до віддаленої бази даних - *моделі сервера баз даних*
- В цьому випадку ядро СУБД функціонує на сервері, прикладна програма на клієнті, а протокол обміну забезпечується за допомогою мови SQL.
- На відміну від технології "файл-сервер", між клієнтом і сервером передається не весь набір даних, а лише дані, які потрібні користувачеві.
- Такий підхід у порівнянні з файловим сервером веде до зменшення завантаження мережі й уніфікації інтерфейсу "клієнт-сервер».
- Однак, мережевий трафік залишається досить високим, крім того, як і раніше неможливо задовільне адміністрування додатків, оскільки в одній програмі поєднуються різні функції.

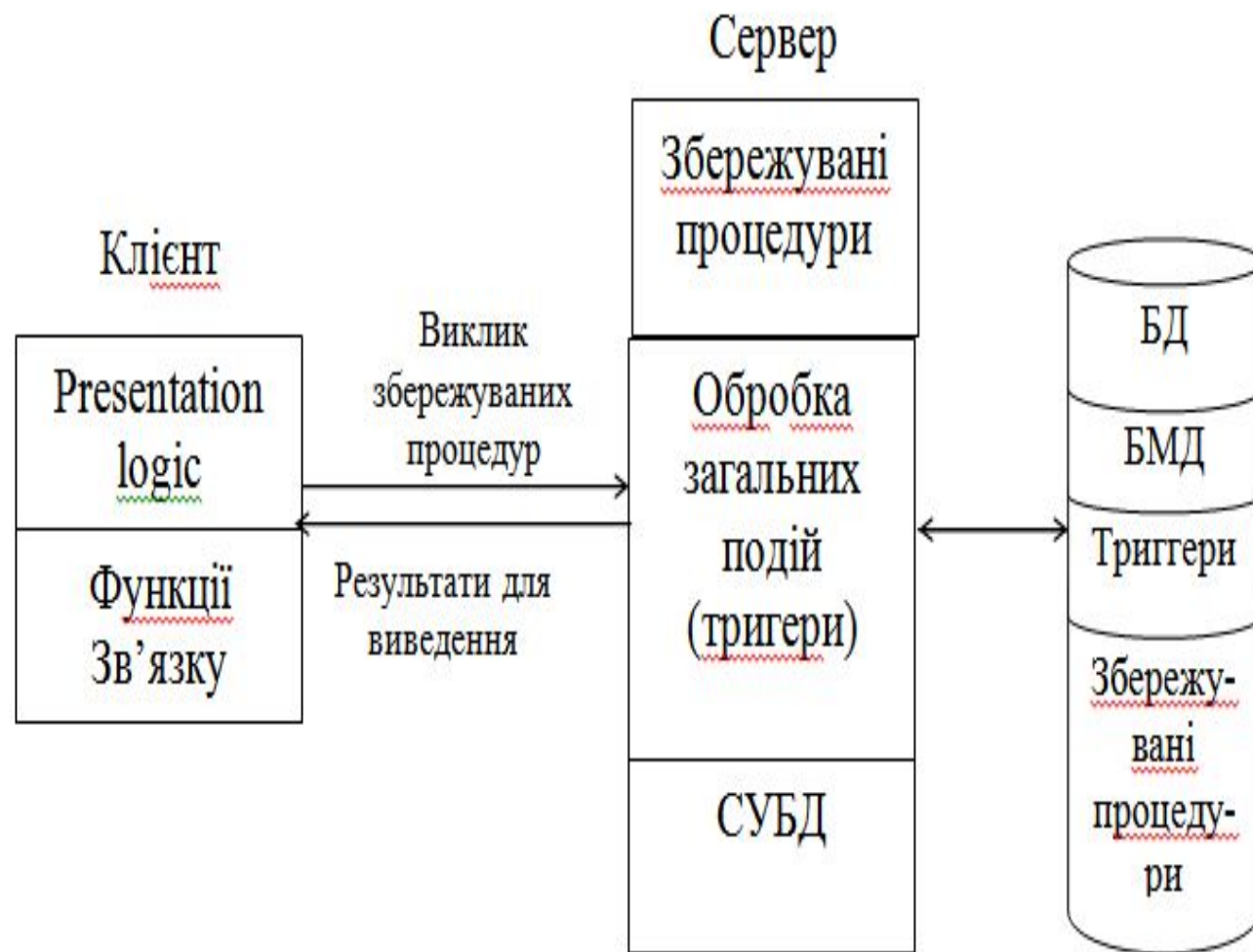


Рис. 5. Модель активного сервера

Модель активного сервера БД

- З розробкою і впровадженням на рівні серверів баз даних механізму збережуваних процедур з'явилася концепція *активного сервера БД*.
- У цьому випадку частина функцій прикладного компонента реалізовані у вигляді *збережуваних процедур*, що виконуються на стороні сервера.
- Решта прикладної логіки виконується на стороні клієнта.

- В основу *архітектури активного сервера* закладена ідея про те, що крім зберігання файлів бази даних, центральний сервер повинен виконувати і основну частину обробки даних.
- Звернення користувачів приймаються сервером і породжують на ньому процеси обробки даних.
- В результаті користувач отримує уже оброблений набір даних.
- Технологія "клієнт-сервер" дає можливість уникнути передачі по мережі великих обсягів інформації завдяки перекладанню всієї обробки інформації на центральний сервер.
- Крім того, даний підхід дає можливість уникнути конфліктних ситуацій при зміні одних тих самих даних різними користувачами, що автоматично підтримує цілісність даних.
- Хоча слід відмітити, що в клієнт-серверних системах існують спеціальні засоби для підтримки цілісності даних.

Модель активного сервера БД

- Переваги такого підходу очевидні :
- -централізоване адміністрування прикладних функцій;
- -зниження вартості володіння системою (ТОС, total cost of ownership) за рахунок оренди сервера, а не його покупки;
- -значне зниження мережевого трафіку (передаються не SQL-запити , а виклики збережуваних процедур) .
- Основний недолік - обмеженість засобів розробки збережуваних процедур у порівнянні з мовами високого рівня та високі вимоги до продуктивності центрального сервера.

Триланкова архітектура

- ▣ Реалізація прикладного компонента на стороні сервера надає наступну модель - сервер додатків.
- ▣ Перенесення функцій прикладного компонента на сервер знижує вимоги до конфігурації клієнтів і спрощує адміністрування, але представляє підвищені вимоги до продуктивності, безпеки і надійності сервера.
- ▣ В даний час намічається тенденція повернення до того, з чого починалася клієнт-серверна архітектура - до централізації обчислень на основі моделі термінал-сервера.
- ▣ У сучасній реінкарнації термінали відрізняються від своїх алфавітно-цифрових предків тим, що маючи мінімум програмних і апаратних засобів, представляють мультимедійні можливості (в т.ч. графічний інтерфейс користувача).
- ▣ Роботу терміналів забезпечує високопродуктивний сервер, куди винесено все, аж до віртуальних драйверів пристроїв, включаючи драйвери відео підсистеми.

підсистеми.

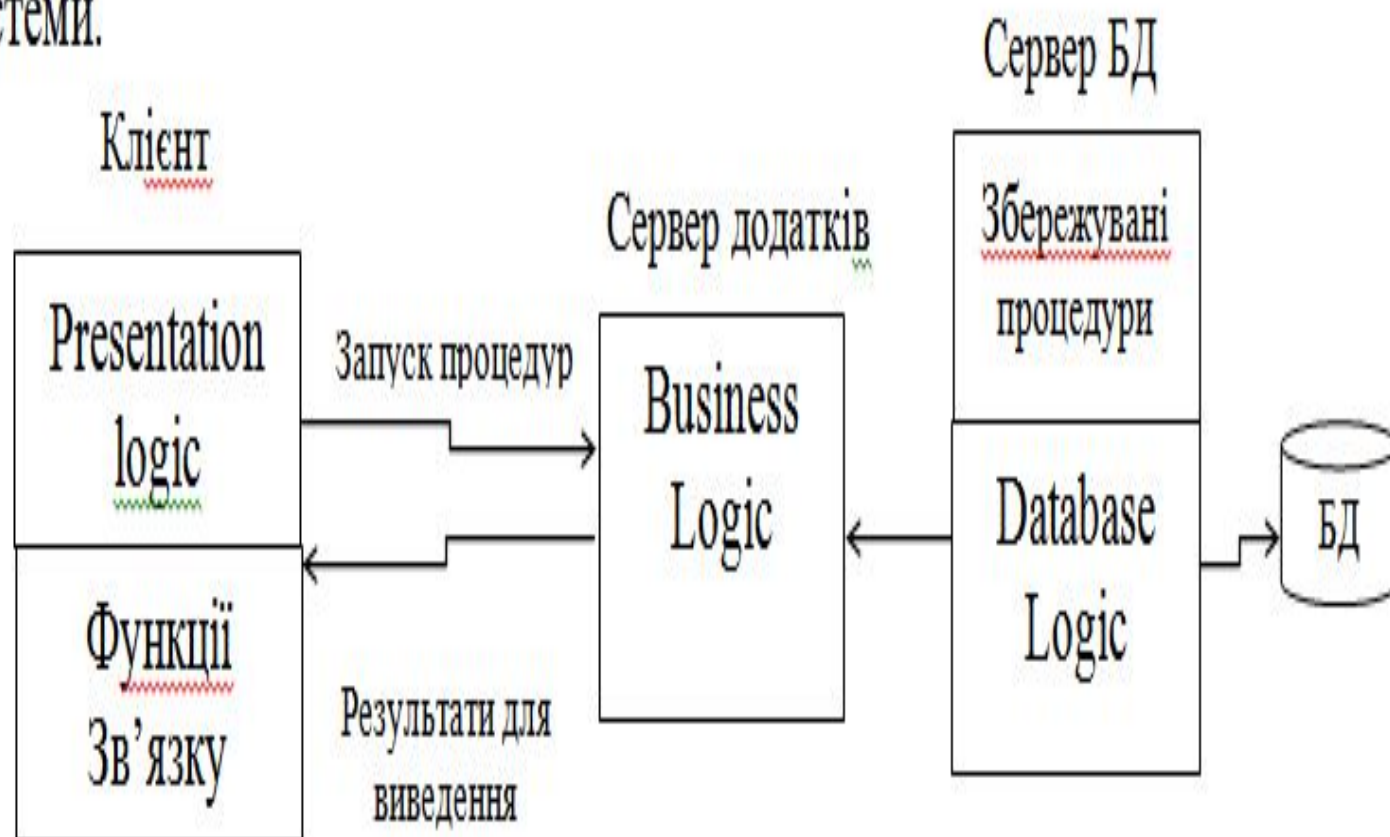


Рис.6. Триланкова клієнт-серверна архітектура

- Ще одна тенденція у клієнт-серверних технологіях пов'язана з усе більшим використанням розподілених обчислень.
- Вони реалізуються на основі моделі *сервера додатків*, де мережевий додаток розділений на дві і більше частин, кожна з яких може виконуватися на окремому комп'ютері.
- Виділені частини програми взаємодіють одна з одною, обмінюючись повідомленнями у заздалегідь узгодженому форматі. У цьому випадку *дволанкова клієнт-серверна архітектура* стає *три ланковою*.
- Як правило, третьою ланкою в триланковій архітектурі стає сервер додатків, тобто компоненти розподіляються наступним чином (рис. 6):
 - 1. Представлення даних (Presentation logic)- на стороні клієнта.
 - 2. Прикладний компонент (Business logic) - на виділеному сервері додатків (як варіант, що виконує функції проміжного ПО).
 - 3. Управління ресурсами - на сервері БД, який і представляє запитувані дані.

- Триланкова архітектура може бути розширена до багатоланкової (N1-tier, Multi-tier) шляхом виділення додаткових серверів, кожен з яких буде представляти власні сервіси і користуватися послугами інших серверів різного рівня.

Порівняння архітектур

- Дволанкова архітектура простіше, так як всі запити обслуговуються одним сервером, але саме через це вона менш надійна і пред'являє підвищені вимоги до продуктивності сервера.
- Триланкова архітектура складніша, але завдяки тому, що функції розподілені між серверами другого і третього рівня, ця архітектура представляє:
 - 1. Високий ступінь гнучкості.
 - 2. Високу безпеку (тому захист можна визначити для кожного сервісу або рівня).
 - 3. Високу продуктивність (завдання розподілені між серверами).

Клієнт-серверні технології

- ▣ Архітектура клієнт-сервер застосовується у великому числі мережевих технологій, що використовуються для доступу до різних мережевих сервісів.
- ▣ **Web-сервери**
- ▣ Спочатку представляли доступ до гіпертекстових документів по протоколу HTTP (Hyper Text Transfer Protocol). Зараз підтримують розширені можливості, зокрема роботу з бінарними файлами (зображення, мультимедіа і т.п.)
- ▣ **Сервери додатків**
- ▣ Призначені для централізованого вирішення прикладних завдань в деякій предметній області. Для цього користувачі мають право запускати серверні програми на виконання. Використання серверів додатків дозволяє знизити вимоги до конфігурації клієнтів і спрощує загальне управління мережею.

Клієнт-серверні технології

▣ Сервери баз даних

- ▣ Сервери баз даних використовуються для обробки запитів на мові SQL. При цьому СУБД знаходиться на сервері, до якого і підключаються клієнтські програми .

▣ Файл-сервери

- ▣ Файл-сервер зберігає інформацію у вигляді файлів і представляє користувачам доступ до неї. Як правило файл-сервер забезпечує й певний рівень захисту від несанкціонованого доступу.

▣ Проксі-сервер

- ▣ По-перше, діє як посередник, допомагаючи користувачам отримати інформацію з Інтернету і при цьому забезпечуючи захист мережі. По-друге, зберігає часто запитувану інформацію в кеш-пам'яті на локальному диску, швидко доставляючи її користувачам без повторного звернення до Інтернету.

Клієнт-серверні технології

- ▣ **Файрволи (брандмауери)**
- ▣ **Міжмережеві екрани**, що аналізують та фільтрують дані і відслідковують як проходить мережевий трафік, з метою забезпечення безпеки мережі.
- ▣ **Поштові сервери**
- ▣ Представляють послуги з відправлення та одержання електронних поштових повідомлень.
- ▣ **Сервери віддаленого доступу (RAS)**
- ▣ Ці системи забезпечують зв'язок з мережею по комутованих лініях. Віддалений співробітник може використовувати ресурси корпоративної ЛВС, підключившись до неї за допомогою звичайного модему.
- ▣ Це лише декілька типів з усього різноманіття клієнт-серверних технологій, що використовуються як в локальних, так і в глобальних мережах.

Клієнти

- Для доступу до тих чи інших мережеских сервісів використовуються клієнти, можливості яких характеризуються поняттям «товщини». Воно визначає конфігурацію устаткування і програмне забезпечення, наявні у клієнта. Розглянемо можливі граничні значення:
- **«Тонкий» клієнт**
- Цей термін визначає клієнта, обчислювальних ресурсів якого достатньо лише для запуску необхідної мережескої програми через Web-інтерфейс. Інтерфейс користувача такого додатка формується засобами статичної HTML (виконання JavaScript не передбачається), вся прикладна логіка виконується на сервері
- Для роботи тонкого клієнта досить лише забезпечити можливість запуску web-браузера, у вікні якого і здійснюються всі дії. З цієї причини web-браузер часто називають "універсальним клієнтом".

▣ «Товстий» клієнт

- ▣ Таким є робоча станція або персональний комп'ютер, що працюють під управлінням власної дискової операційної системи та мають необхідний набір програмного забезпечення.
- ▣ До мережевих серверів «товсті» клієнти звертаються в основному за додатковими послугами (наприклад, доступ до WEB-серверу або корпоративної бази даних).
- ▣ Так само під «товстим» клієнтом мається на увазі і клієнтський мережевий додаток, запущений під управлінням локальної ОС.
- ▣ Такий додаток поєднує компонент представлення даних (графічний користувацький інтерфейс ОС) і прикладний компонент (обчислювальні потужності комп'ютера клієнта).
- ▣ Останнім часом все частіше використовується ще один термін: «rich»-client. «Rich»-клієнт свого роду компроміс між «товстим» і «тонким» клієнтом. Як і «тонкий» клієнт, «Rich»-клієнт також представляє графічний інтерфейс, описуваний вже засобами XML і включає деяку функціональність товстих клієнтів. Прикладна логіка «Rich»-клієнта також реалізована на сервері.

ДЯКУЮ ЗА УВАГУ !