РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

Павел Кюркчиев
Ас. към ПУ „Паисий Хилендарски“
@pkyurkchiev

РАЗПРЕДЕЛЕНИ АРХИТЕКТУРИ CLIENT-SERVER И N-TIER

Софтуерна архитектура

- Софтуерната архитектура е съвкупност от важни решения за организацията на програмните системи.

История

- Софтуерна архитектура като концепция има своите корени в изследванията на Едсгер Дейкстра през 1968 г. и Дейвид Парнас в началото на 1970. Добива широка популярност като термин в началото на 1990 години.

Архитектура

- Софтуерните архитектури са необходими за аргументиране на софтуерните системи. Всяка структура съставлява софтуерни елементи, връзки между тях и свойствата на двете-елементи и връзки.

Основни твърдения:

- Софтуерните архитектури са важни за успешното разработване на софтуерни системи.
- Има достатъчно и добре обобщени знания за софтуерните архитектури.
- Софтуерните системи се изграждат за да удовлетворят бизнес целите на организациите.
- Архитектурата е мост между тези (често абстрактни) бизнес цели и крайната (конкретна)система.
- Софтуерните архитектури могат да бъдат проектирани, анализирани, документирани и реализирани с определени техники, така че да бъдат постигнати целите на бизнеса.

Архитектурни дейности

- Архитектурен анализ
  - *процес на разбиране на средата, в която ще работят планирана система или системи и определяне на изискванията за системата.*
- Архитектурен синтез
  - *или дизайн е процес на създаване на архитектура.*
- Оценяване на архитектура
  - *процес на определяне на това колко добре настоящия дизайн или част от него отговаря на изискванията, получени по време на анализа*
- Еволюция на архитектура
  - *процесът на поддържане и адаптиране на съществуваща софтуерна архитектура, за да отговаря на изискванията и промените в околната среда*

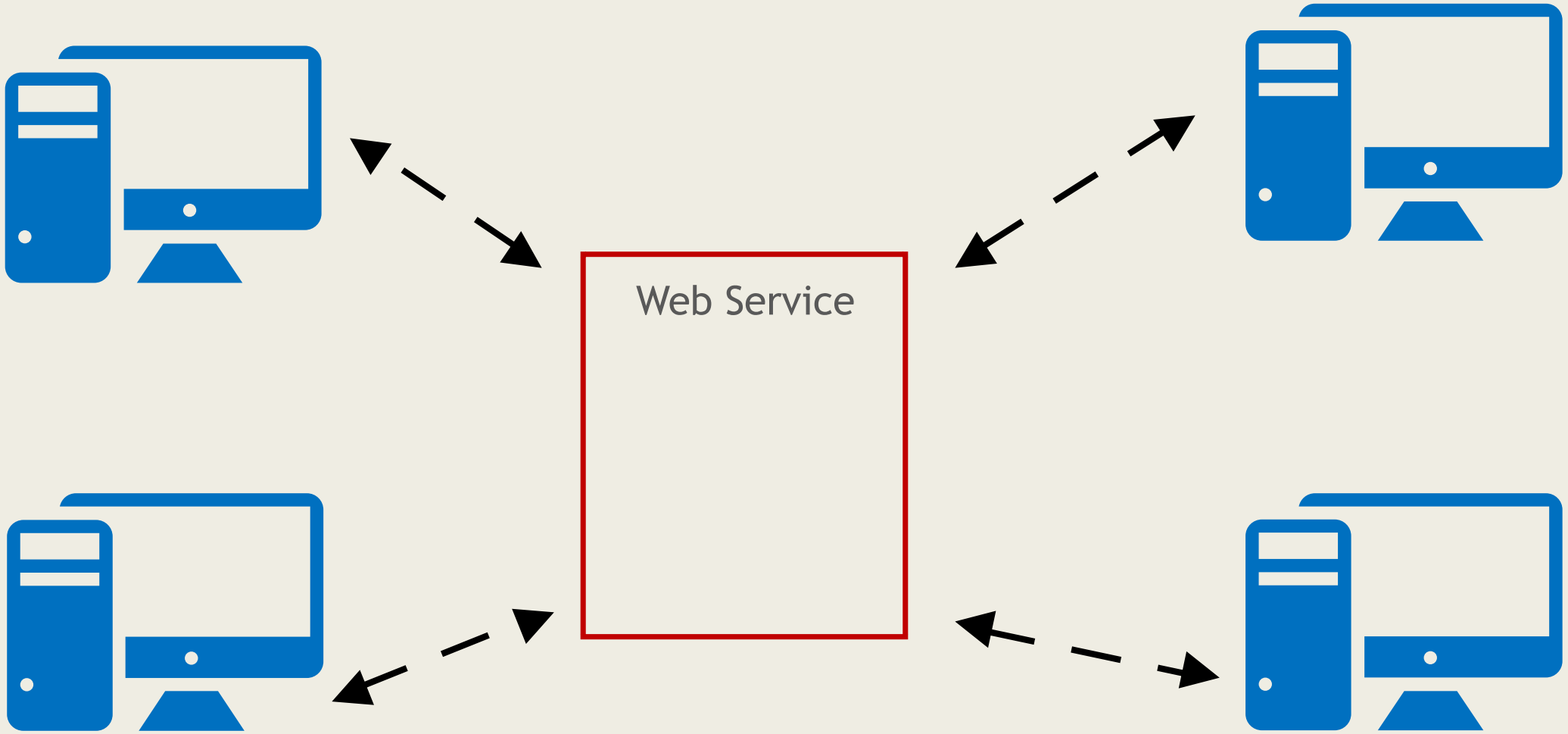
Архитектури

- Client-Server
- N-tiers (3-tiers)
- SOA
- Microservice

Client-Server

- Моделът клиент-сървър е разпределена архитектура на приложение, която разделя задачите(или натоварването) между доставчиците на ресурс (или услуга), наречени сървъри и инициатори, наречени клиенти.

Какво е Client-Server?



Какво е клиент-сървър приложение?

- Посредством клиента, потребителят взаимодейства с приложението
- Един сървър може да обслужва много клиенти, и да предоставя функционалности, които не могат да се реализират само при клиента в изолация
- Когато клиента има нужда от информация, или, да предаде на сървъра информация в следствие интеракцията на потребителя с приложението, клиента може да изпрати съобщение към сървъра
- В отговор на това съобщение, сървъра изпълнява необходимите действия и може да върне съобщение-отговор
- При някои клиент-сървър приложения сървъра може да изпраща съобщения към клиента, които да не са задължително отговор на съобщения изпратени от клиента

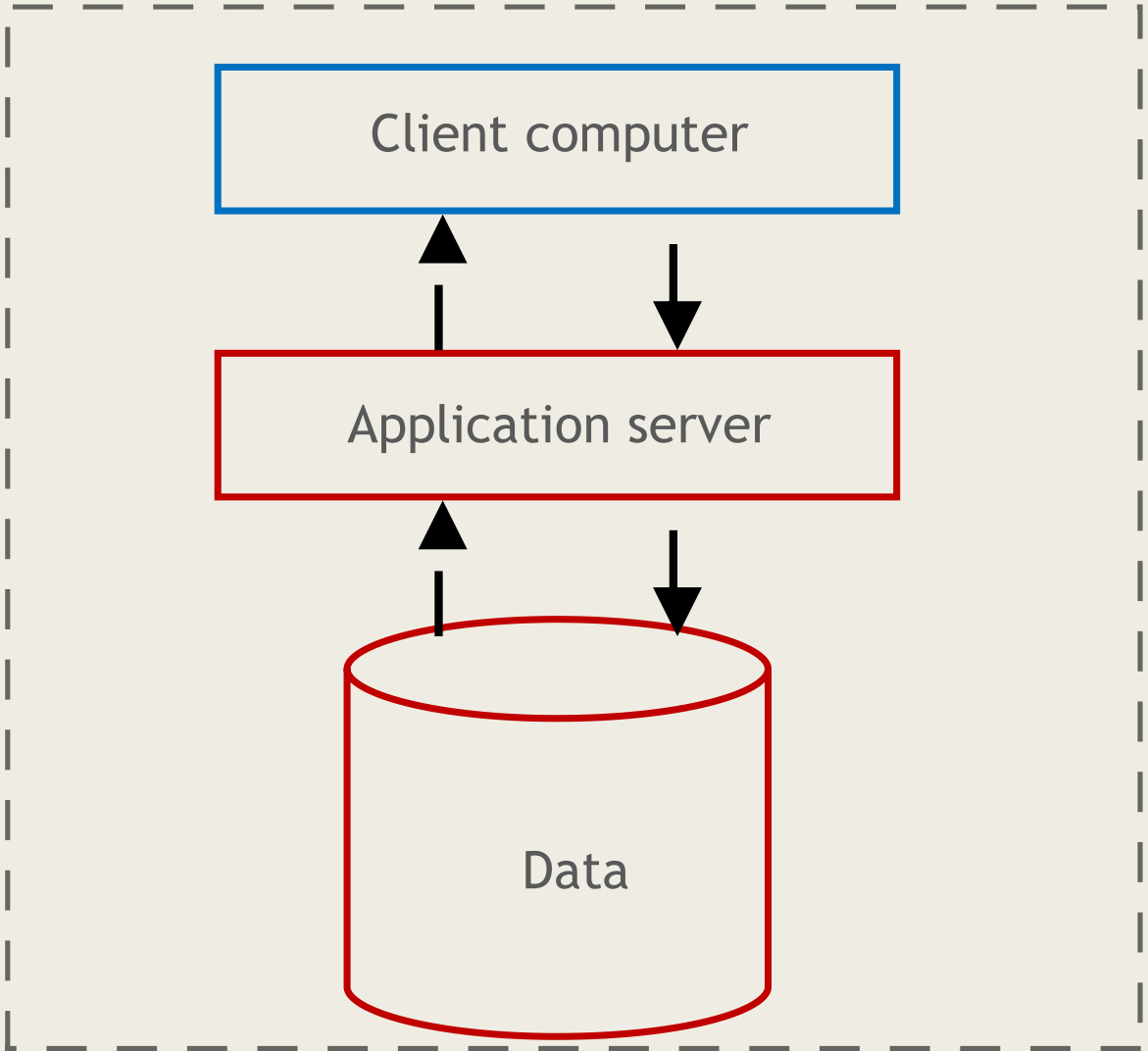
Какво е клиент-сървър приложение?

- Често клиента и сървъра комуникират през компютърна мрежа (локална мрежа, интранет или интернет) и се намират на отделен хардуер

N-tier(3-tier)

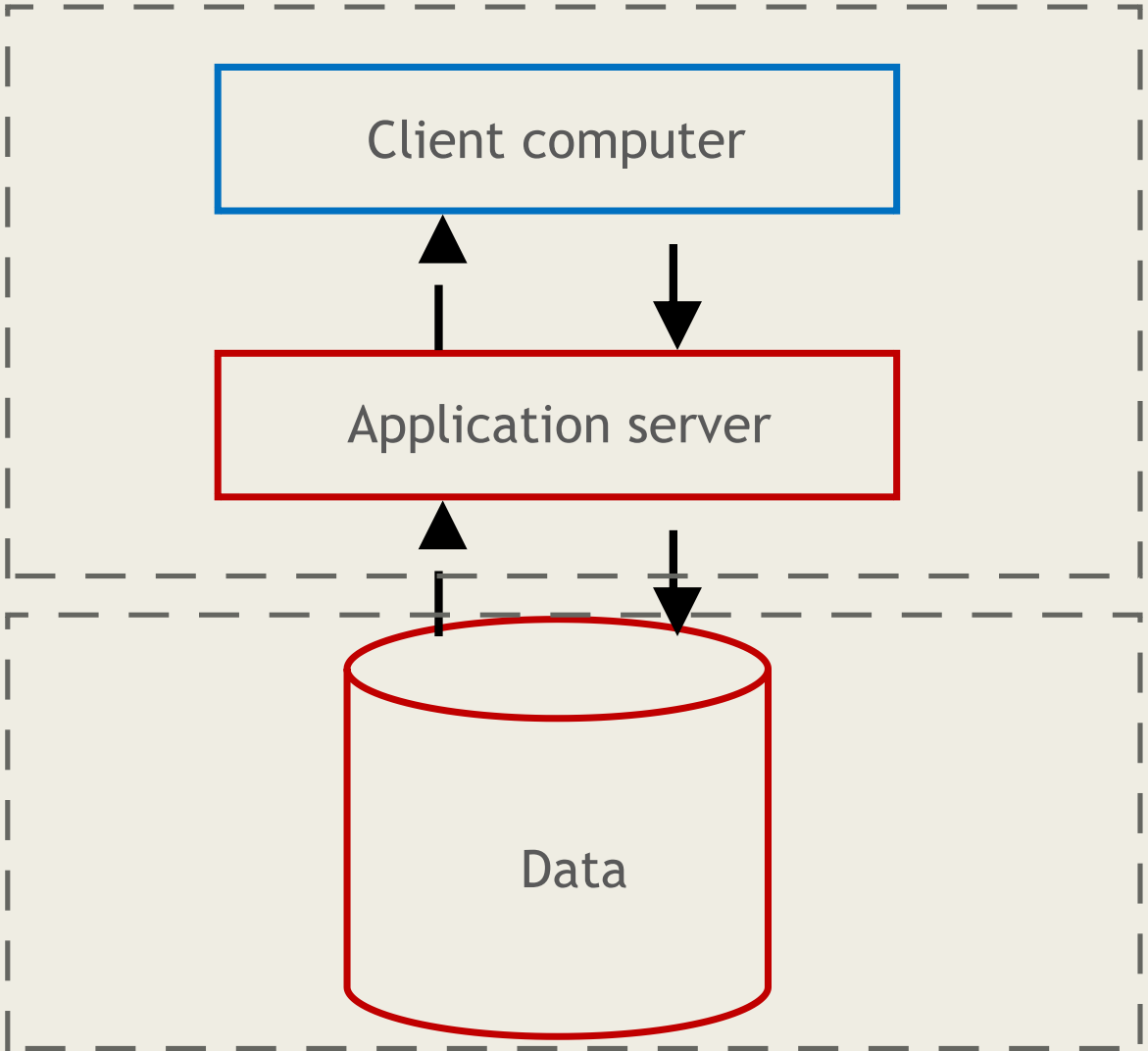
- Многослойната архитектура е вид клиент – сървър архитектура, на която презентационния, бизнес логиката и слоя за управление на информацията са отделени физически

1-tier architecture



Client tier

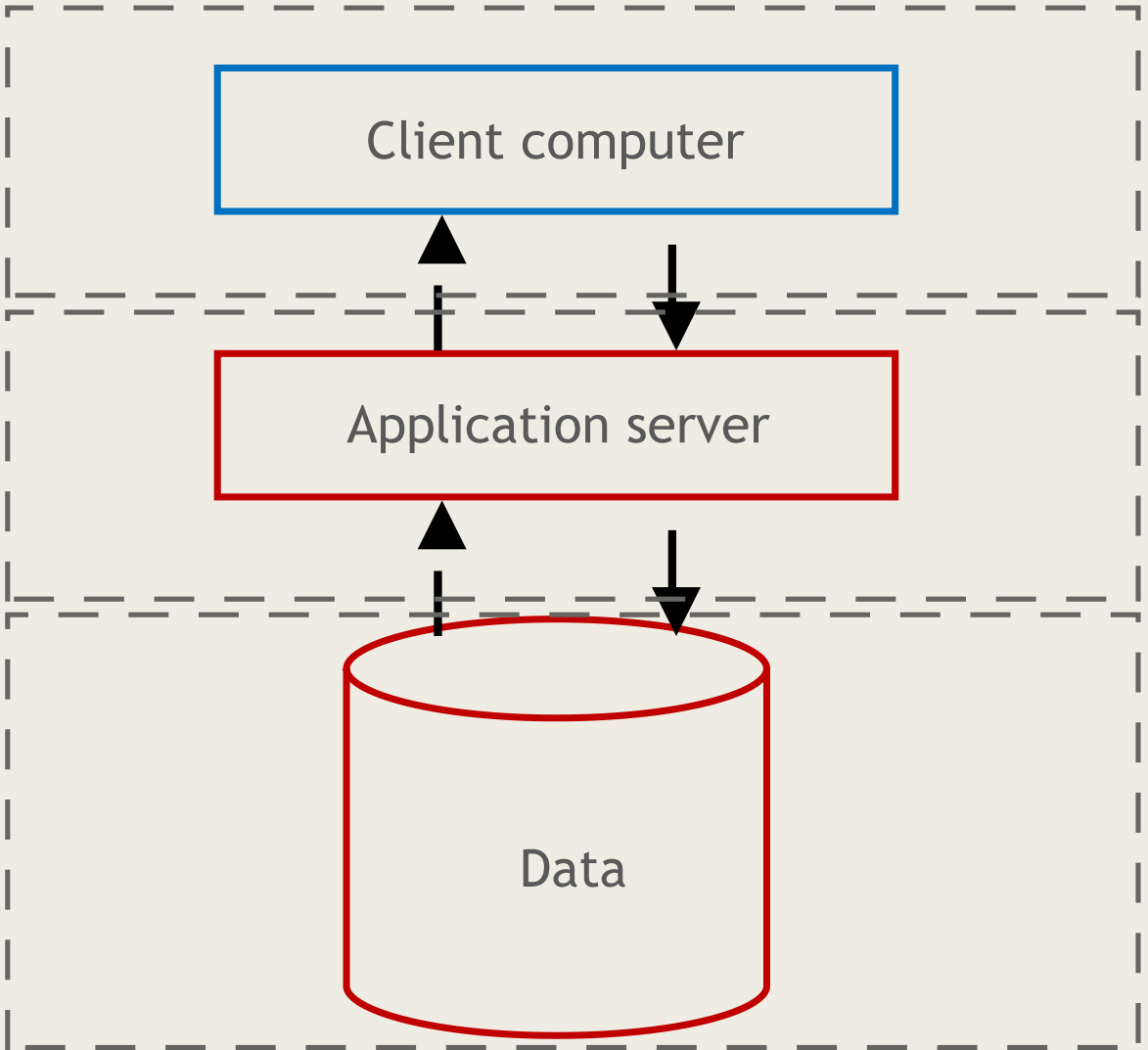
2-tier architecture



Трислойна архитектура

- Презентационен слой
  - *Презентационният слой е на най-високо ниво в приложението и потребителят има директен достъп до него*
- Слой за бизнес логика
  - *Този слой е изтеглен от презентационния слой, и като отделен такъв, контролира функционалността на приложението като извършва различни процеси по обработката на данните.*
- Слой за данните
  - *Този слой се състой от сървър база данни. Тук информацията се съхранява и достъпва*

3-tier architecture



Client tier

Application tier

Database tier

Типове N-tier

- Closed layer architecture
  - *В затворената архитектура, слой може да се свърже само с най – близкия до него.*
- Open layer architecture
  - *В отворената архитектура, слой може да се свърже с всеки слой намиращ се под него.*

Кога да използваме N-Tier?

- Опростени уеб приложения
- Миграция на локални приложения към облачна среда (AWS, Azure, Google cloud)
- Обединена разработка на локално и облачно базирано приложения

Предимства

- Преносимост на едно приложение между облачна и локална платформа, или различни облачни платформи
- По – малко време за обучение на голяма част от разработчиците
- Естествена еволюция от традиционния модел на разработка
- Отворен за различни среди на разработка (Windows, Linux, Unix …)

DEMO N-TIER

https://github.com/pkyurkchiev/n-layer_skeleton_net-core