

ДИАНС БАЗА

Прв Колоквиум

1. Во REST, типично го имаме JSON како формат, понекогаш и XML

A. True ***

Б. False

2. Елементи на имплементациската архитектура (Implementation Architecture): *dropdown

а) Конектори – “Uses” врска.

б) Погледи – Разделени.

в) Компоненти – Имплементациски модул (Implementation module).

3. Salesforce.com, Google Apps, Microsoft Office 365, slack, Dropbox, Sensium.io се примери за:

*multiple

A. цевки и филтри (pipes and filter)

Б. BBB

В. R2D2

Г. Saas ***

4. Архитектурите на проток на податоци можат да се класифицираат на 1. Сериски-секвенцијални (batch-sequential) архитектури и 2. Цевки и филтри (pipes and filters)

A.True ***

Б. False

5. GraphQL е јазик за пребарување и манипулација за GUI

A. True

Б. False ***

6. Во архитектурата компонентите можат да бидат: *multiple

A. подсистеми, одделни процеси, пакети на изворен код ***

Б. мрежни протоколи, повикување на методи, асоцијации (ова е за конектори)

7. Тенките клиенти (Thin clients) имаат одлично знаење за апликацијата

A. True

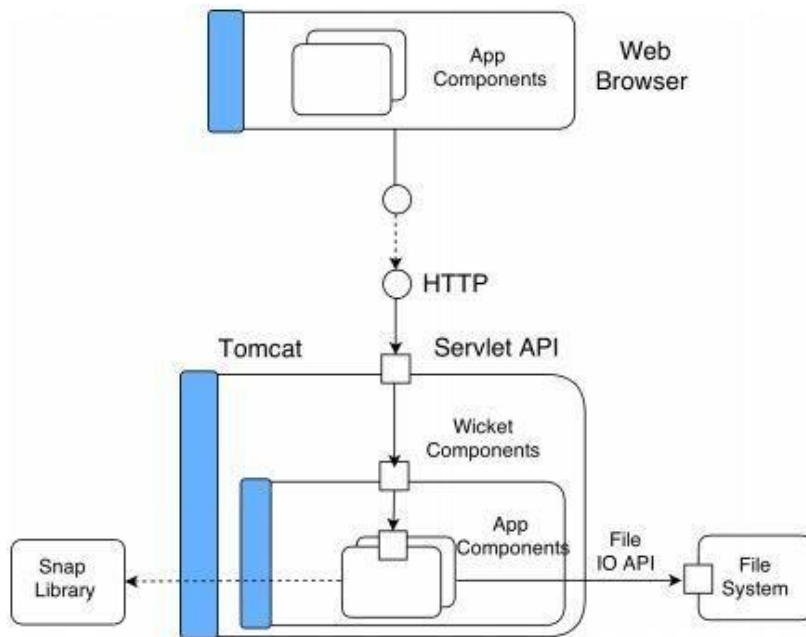
Б. False ***

8. Функционалност е што системот може да направи, а однесување е секвенца на активности

A. True ***

Б. False

9. Ова е пример за: *multiple



а. Имплементациска архитектура (implementation architecture) ***

б. Извршна архитектура (execution architecture)

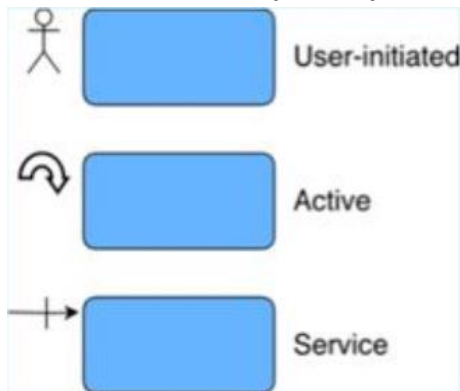
в. Серијализациска архитектура (serialization architecture)

10. Во архитектурите ориентирани кон ресурси (Resource Oriented Architectures), ресурсите (Resources) се изложени преку URLs

A. True ***

Б. False

11. Следната слика претставува: *mutiple



A. Стереотипи на конектори на имплементациската архитектура

Б. Стереотипи на конекторите на извршна архитектура

В. Стереотипи на конекторите на концептуалната архитектура

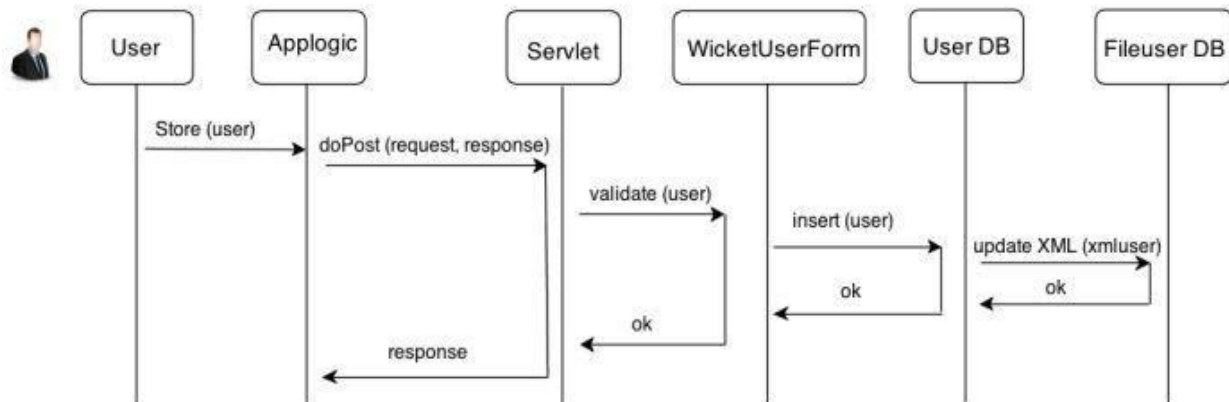
Г. Стереотипи на компоненти на извршната архитектура ***

12. Архитектурите со нотификации (notification architecture) исто така се викаат publish-and-subscribe

A. True ***

B. False

13. Следната слика претставува ():



A. Дијаграм на интеракција (interaction diagram)

B. Дијаграм на проток (flow diagram)

B. Дијаграм на состојби (state diagram)

Г. Runtime diagram

Д. Секвенцен дијаграм (sequence diagram) ***

14. Во имплементациските архитектури, инфраструктурните компоненти се одговорни за имплементација на одговорностите на ниво на домен (domain-level responsibilities)

A. True

B. False ***

15. Конекторите во извршната архитектура (multiple)

A. Шест различни типови на стрелки за шест различни сценарија на повикување

B. Стрелката ја претставува насоката на повикот ***

B. Стрелката покажува од компонентот кој повикува кон компонентот кој се повикува ***

Г. Означуваат дека еден компонент повикува друг ***

16. Што од следното е точно? *multiple

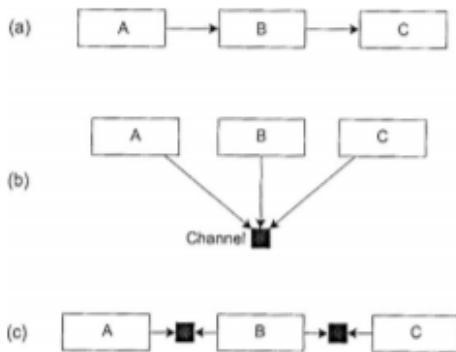
A. Системите имаат однесување ***

B. Системите можат да имаат излези само кон околината

B. Системите имаат структура ***

Г. Системите служат за одредена цел ***

17. Следната слика покажува примери на: *multiple

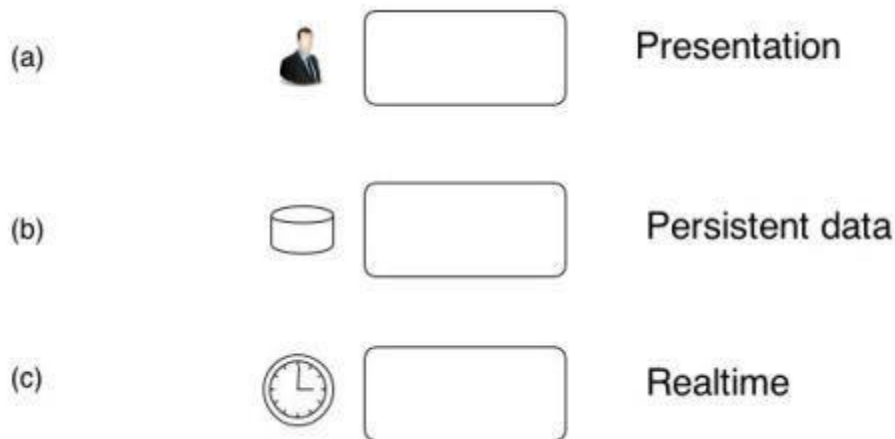


A. Табли (Blackboards)

Б. Нотификации

В. Цевки и филтри (Pipes and filters) ***

18. Следната слика претставува:



A. Стереотипи на компоненти на концептуалната архитектура

Б. Стереотипи на конекторите на извршна архитектура

В. Стереотипи на конекторите на концептуалната архитектура

Г. Стереотипи на компоненти на имплементациската архитектура

19. Сервиси од RESTful стил примаат пливо полно со податоци од клиентот

A. True

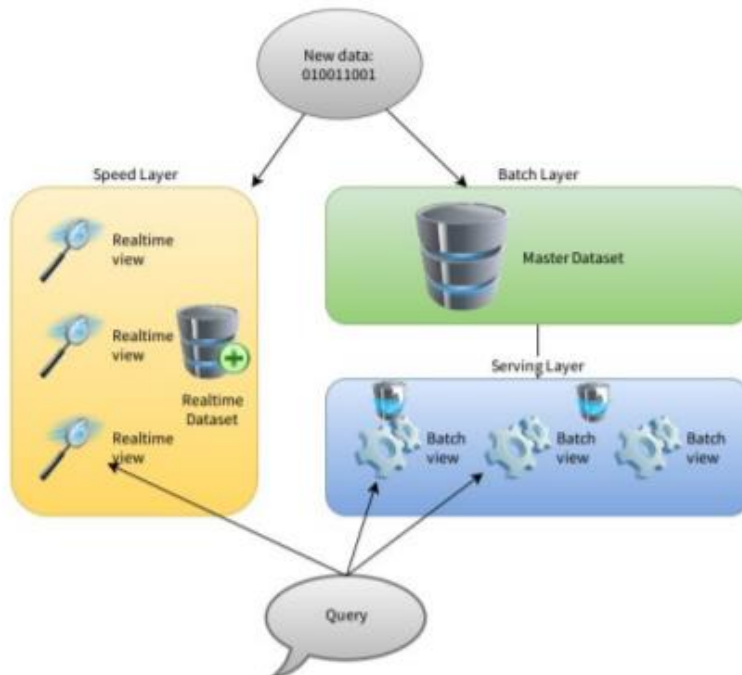
Б. False ***

20. Имплементациската архитектура (implementation architecture) се фокусира на тоа како системот е изграден:

A. True ***

Б. False

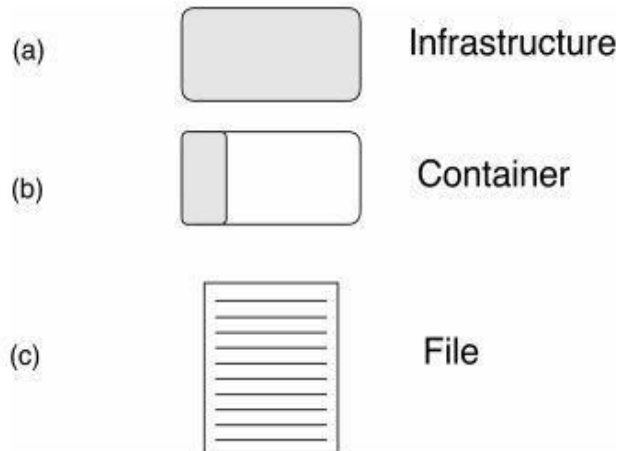
21. Следната слика претставува пример за Lambda архитектура



A. True ***

B. False

22. Следната слика претставува: *multiple



A. Стереотипи на концептуални конектори

B. Стереотипи на извршни компоненти

B. Стереотипи на извршни конектори

Г. Стереотипи на имплементациски компоненти

23. Во процесот на моделирање на извршната архитектура во основа се испитуваат процесите на секој паралелен подсистем

A. True ***

Б. False

24. Резултатите од анализата на барања се: *multiple

A. Извршни барања- execution requirements

Б.Имплементациски барања- Implementation requirements

В. Контекстуални барања- Contextual requirements ***

Г. Функцииски барања- Functional requirements ***

Д. Нефункционални барања- Nonfunctional requirements ***

27. Во имплементациската архитектура, конекторите претставуваат имплементира врска (implements)

A. True

Б. False ***

28. Извршното однесување (execution behavior) се моделира со секвенци дијаграми (sequence diagram)

A. True

Б. False ***

29. Во микросервисните архитектури (microservice architectures) сервисите делат заеднички implementation service bus

A. True

Б. False ***

30. Model-view-controller шемата не е замена за n-слојната архитектура (n-tier architecture)

A. True ***

Б. False

31. За да прикажеме дека архитектурното решение е изводливо имплементираме прототипи

A. True ***

Б. False

32. Грануларност на компонентите претставува колку функционалност е назначена на еден компонент

A. True ***

Б. False

33. Софтверската архитектура е колекција на модели на софтверски системи на повеќе нивоа на апстракција и детали. Моделите опишуват: *multiple

A. Системски компоненти ***

Б. Врските меѓу компонентите ***

В. Како компонентите се во интеракција да ја исполнат целта на системот ***

Г. Системот како целина ***

34. Најдобрата практика за дијзанирање концептуална архитектура (conceptual architecture) е да се започне со клучните концепти на засегнатите страни (stakeholders) и надворешните системи

A. True ***

Б. False

35. One of the most well known examples of data-centered architecture is a database architecture

A. True ***

Б. False

36. Во peer to peer мрежи, секој клиент е истовремено сервер и клиент

A. True ***

Б. False

37. N-слојната архитектура (N-tier architecture) е модерната client-server архитектура

A. True ***

Б. False

38. Sharding ги разделува податоците вериткално

A. True

Б. False ***

39. Интерфејсите се дел од имплементациската архитектура (implementation architecture)

A. True ***

Б. False

40. Почетната софтверска архитектура мора да се дизајнира пред фазата на извлекување на софтверски барања

A. True

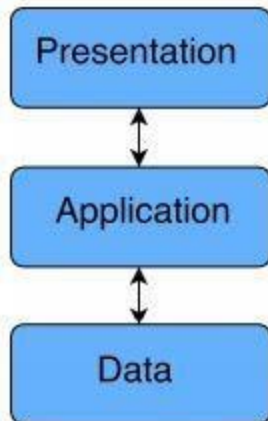
Б. False ***?

41. Weighted Scoring Method се користи за избор на концептуална архитектура

A. True

Б. False ***

42. Следното е: *multiple

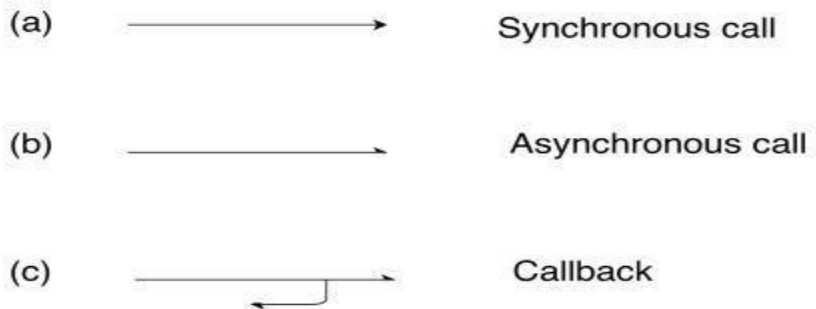


- A. MVCархитектур
- Б. Трислојна архитектура (3-tier architecture) ***
- В. Архитектура на веб сервиси (web services architecture)
- Г. Го нема фали

43. Процесирањето на микро-сериски проток (micro batch stream processing) има помала латентност и голема пропусност

- A. True
- Б. False *** (има само голема пропусност)

44. Следната слика претставува: *multiple



- A. Типови на конектори на извршна архотектура ***
- Б. Типови на конектори на концептуална архитектура
- В. Типови на конектори на имплементациска архитектура

45. Последици на слабо поврзување: *multiple

- A. обично е полесно да се креираат таквите компоненти (последица на тесно поврзување)
- Б. Компонентите се погодни за повторно користење ***
- В. Ако се смени една софтверска компонента, треба да се сменат и другите компоненти (последица на тесно поврзување)
- Г. Компонентите не може да се заменат

46. Во Репозиториум (Repository) системот испраќа нотификации и податоци на претплатниците кога податоците од интерес се менуваат

- A. True
- Б. False ***

47. Изберете ги точните тврдења: *multiple

- А. Деталниот извршен модел вклучува процеси и нишки ***
- Б. Деталниот концептуален модел вклучува процеси и нишки
- В. Деталниот модел на threading/ processes вклучува процеси и нишки

48. Ако комуникацијата клиент-сервер (client-server communication) е stateful, ја следи состојбата на апликација

- А. True ***
- Б. False

49. Поврзувањето (coupling) влијае врз атрибутите на квалитет (quality attributes): *multiple

- А. Слабото поврзување обично ги подобрува перформансите што можат да се постигнат (achievable performance) како и следливоста (traceability)
- Б. Слабото поврзување обично ја подобрува одржливоста (maintainability), еволутивноста (evolvability), можноста за повторна употреба (reusability) ***
- В. Силното поврзување обично ги подобрува одржливоста (maintainability), еволубилноста (evolvability), како и можноста за повторна употреба (reusability)
- Г. Силното поврзување обично ги подобрува перформансите што можат да се постигнат (achievable performance) како и следливоста (traceability) ***

50. Поврзете го соодветното тврдење за конкретниот архитектурен поглед. *dropdown

- а) Извршната архитектура – го разгледува системот во однос на неговата **runtime структура**. (The execution architecture considers the system in terms of its runtime structure).
- б) Имплементациската архитектура – го разгледува системот во однос на неговата **build-time структура**. (The implementation architecture considers the system in terms of its build-time structure).
- в) Концептуалната архитектура – ја разгледува структурата на системот во однос на функционалноста на ниво на домен (**domain-level functionality**). (The conceptual architecture considers the structure of the system in terms of its domain-level functionality).

51. Правила за дизајн на концептуална архитектура *multiple

- А. Избегнувајте кластери од наредби (command clusters) ***
- Б. Избегнувајте концентрирани точки (blobs) ***

52. Што е точно за архитектурните модели? *multiple

А. Архитектурното однесување е динамички модел на системот (како компонентите се во меѓусебна интеракција со цел да извршат некаква работа) ***

Б. Архитектурната структура е статички модел на системот (како системот е поделен на компоненти) ***

53. Мотивација да се воведат ред на пораки (Queues) е да се следи процесирањето на барања

А. True ***

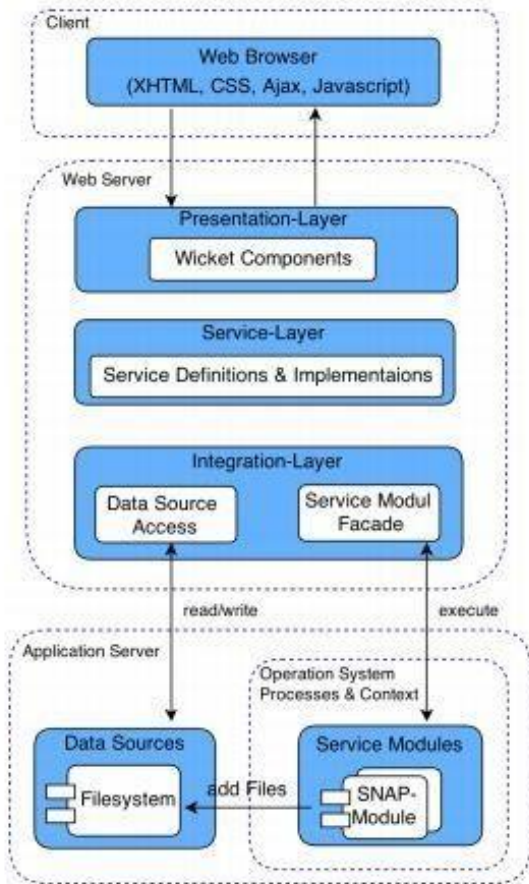
Б. False

54. Архитектурниот стил на поток на податоци (data-flow architecture style) е карактеризиран со тоа што го гледа системот како низа од трансформации на последователните делови на влезните податоци

А. True ***

Б. False

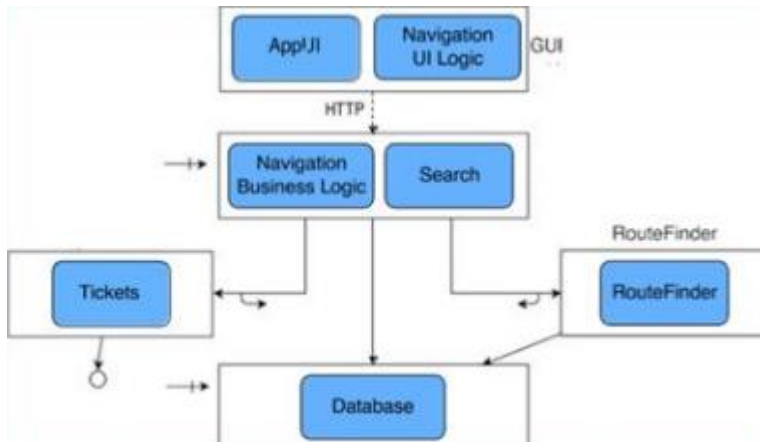
55. Ова е пример на имплементациона архитектура



А. True ***

Б. False

56. Сликата ги поврзува извршните имплементациски модели



A. True

B. False ***

57. Елементи на извршна архитектура (Execution Architecture) *dropdown

а) Конектори – Инвокација (Call).

б) Погледи – Повеќе.

в) Компоненти – Единици на конкурентно однесување (concurrent act).

58. Дали следната класа го исполнува Single Responsibility

```

class Book{
    public string Author { get; set; }
    public string Title { get; set; }
    public void Save() {
        //Save to database code.
        //Open db connection.
        //Make some logic checks.
        //DB CRUD operations.
        //Close connection.
    }
}
  
```

A. ДА

B. НЕ ***





59. Поделбата на одговорности (Separation of Concerns) е:

а) Декомпозиција на проблемот во независни делови.

б) Поделба на грануларни компоненти и поврзани конектори.

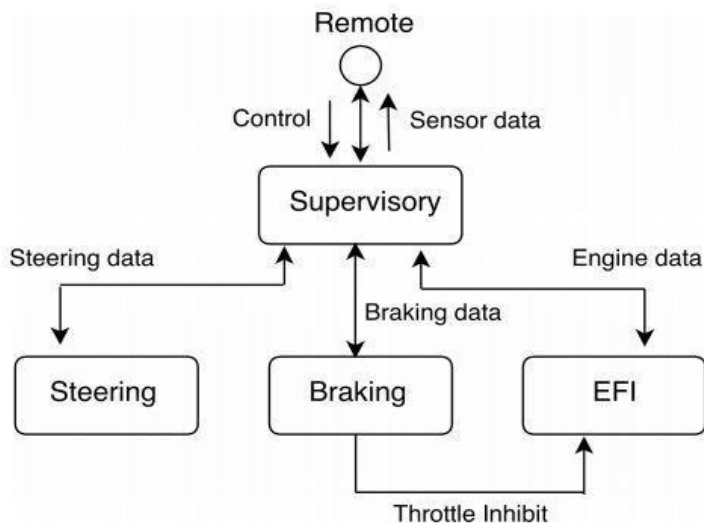
в) Објектно-ориентирана десеријализација на одговорности.

60. Следната слика претставува стереотипи за:

- (a)  API Call
- (b)  Callback
- (c)  Network protocol
- (d)  Signal

- а) имплементациски конектори. ***
- б) извршни конектори.
- в) извршни компоненти.
- г) имплементациски компоненти.
- д) концептуални компоненти.

61. Следниот дијаграм е:



- а) Концептуален поглед (Conceptual view). ***
- б) Извршен поглед (Execution view).
- в) Имплементациски поглед (Implementation view).

62. Слоеви на Облак се:

- a. MiiS
- b. SaaS ***
- c. GiiS
- d. IaaS ***
- e. PaaS ***

63. Главни REST операции се

- a. GET ***
- b. DIR
- c. HASH
- d. POP
- e. POST ***
- f. PUT ***
- g. DELETE ***

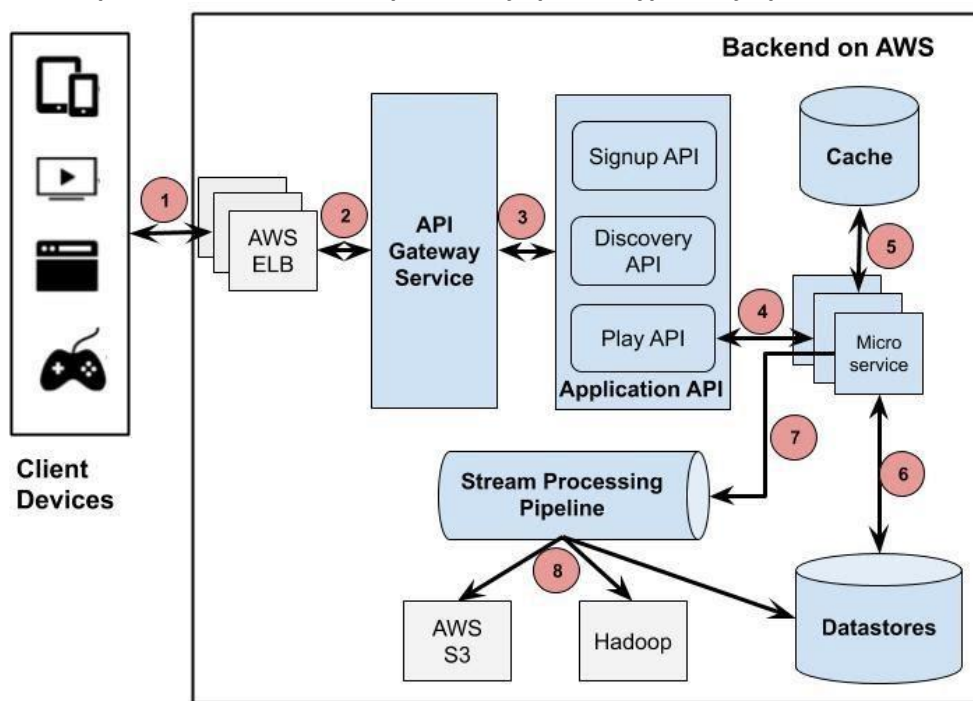
64. Компонента кај софтверска архитектура е:

- a. Архитектурен ентитет со задача да ги регулира и ефектуира интеракциите помеѓу компонентите
- b. Елемент кој енкапсулира податоци и процесирање на ниво на архитектура ***

64.1. Врска кај софтверска архитектура е:

- a. Елемент кој енкапсулира податоци и процесирање на ниво на архитектура
- b. Архитектурен ентитет со задача да ги регулира и ефектуира интеракциите помеѓу компонентите ***

65. Изберете ги точните одговори за овој архитектурен дијаграм.



- A. Оваа архитектура се хостира на Amazon Web Services ***
- Б. Сликата ја претставува позадинската (бекенд) архитектура на youtube
- В. Сликата го претставува компонентниот дијаграм на клиентската апликација на Netflix
- Г. Сликата ја претставува backend архитектурата на Netflix ***
- Д. Сликата ја претставува појачувачот на Google сервисите за стриминг

66. Можноста за аудио конференции во BBB ја овозможува:

- A. Redis DB
- B. FreeSwitch ***
- B. Kurento

67. Секоја архитектура е дизајн, но не секој дизајн е архитектура

- A. True ***
- B. False

68. Компонентата кај софтверска архитектура е:

- a) Елемент кој енкапсулира податоци и процесирање на ниво на архитектура. ***
- б) Архитектурен ентитет со задача да ги регулира и ефектуира интеракциите помеѓу компонентите.

7. Наведете 5 типа нефункционални барања (карактеристики) на еден систем заедно со кусо објаснување на секоја од нив

- a. Безбедност – се одредува колку успешно системот ги крие оние податоци кои се сензитивни
- b. Робустност – справување со грешки на самиот систем
- c. Ефикасност – системот зависи од тоа колку добро ги извршува задачите кои се дефинирани
- d. Одржливост – одржливост на самиот систем и имплементација на некои нови функционалности
- e. Скалабилност – успешноста за прилагодување на некои нови функционалности од страна на системот