Department of Computer Engineering

Academic Term: January - May 2023

Class: ***B.E (Computer) Semester VIII*** **Division:** ***B***

Subject Name: ***Distributed Systems CSC 801***

Assignment I

1. Discuss the applications of the following computing paradigms.
   I) **High performance distributed computing and Pervasive systems**
      a) Grid Computing b) Cluster Computing c) Cloud Computing
      d) Distributed Computing e) Ubiquitous Cloud Computing f) Parallel Computing

2. Give some examples of multi-tiered architectures applicable to distributed systems [Draw diagrams, mention the applications, advantages and disadvantages of the given architectures]

Roll No:	**8940**
Name of the Student:	**Warren Fernandes**
Date of Submission	***08/02/2023***

Evaluation:

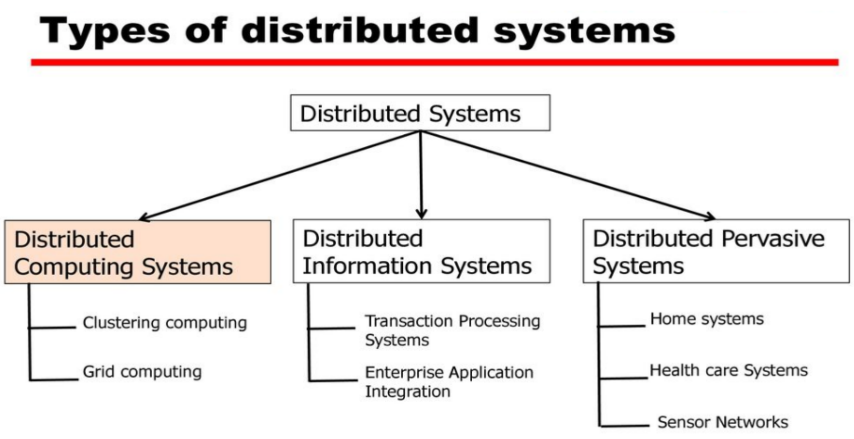
Sr. No	**Rubric**	**Grade**
1	**Timeliness**	**/2**
2	**Level of Content [clearly mention the source of your literature]**	**/4**
3	**Knowledge/Understanding**	**/4**

Signature of the Teacher :

	CO Mapping of the Tool
Assessment Tool	Assignment I
CSC 801.1	Demonstrate the knowledge of basic elements and concepts related to distributed system technologies.
Module	1

Rubrics for Assignment Grading:

Indicator				
Timeline (2)		More than 2 days late (0)	Max 2 days late (1)	On time (2)
Level of content (4)	Just Managed (1)	Major points are addressed minimally (2)	Only major topics are covered (3)	Most major and some minor criteria are included. Information is Adequate (4)
Reading and Understanding (4)	Just Managed (1)	Superficial at most (2)	Understood concepts but no related topics (3)	Understood concepts and related topics (4)



1. Discuss the applications of the following computing paradigms.

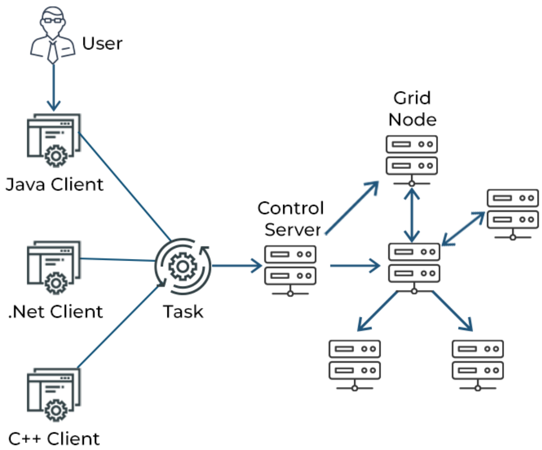
   I) **High performance distributed computing and Pervasive systems**

   a) Grid Computing b) Cluster Computing c) Cloud Computing
   d) Distributed Computing e) Ubiquitous Cloud Computing f) Parallel Computing

Ans. High performance distributed computing and pervasive systems refer to the integration of advanced computing technologies, communication networks, and ubiquitous computing devices to build highly scalable and efficient systems for solving complex problems and performing demanding computational tasks. High performance distributed computing refers to the use of a large number of computing nodes that work together in a coordinated manner to perform a common task. The nodes can be geographically dispersed and connected via a high-speed communication network. The goal of high-performance distributed computing is to provide solutions that are faster and more reliable than those provided by traditional computing systems. Applications of high-performance distributed computing include scientific simulations, data analysis, and machine learning. Pervasive systems, on the other hand, refer to the integration of advanced computing technologies into everyday objects and environments to make them more intelligent and responsive.

I. <u>Grid Computing</u>

Grid computing enables the coordinated use of multiple computing resources to solve complex problems and perform demanding computational tasks. A grid computing system is composed of a large number of nodes, which can be geographically dispersed and connected via a high-speed communication network. The nodes can be computers, servers, or other computing devices that can be used to perform tasks and share resources in a coordinated manner.



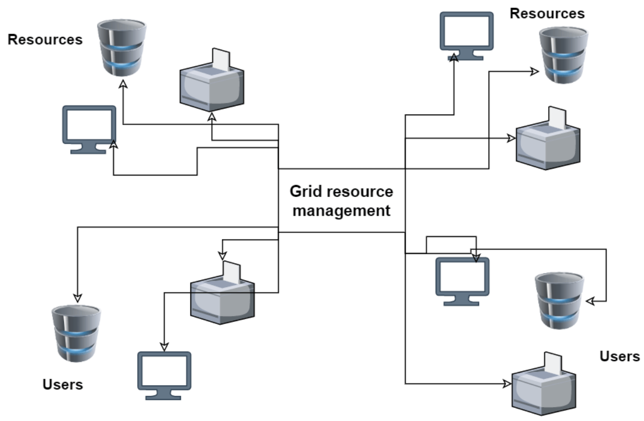
Grid computing has a wide range of applications in various fields, some of which are:

- Scientific simulations: Grid computing enables researchers to perform complex simulations and modeling of physical and biological systems by leveraging the power of multiple computing resources.

- Machine learning: Grid computing can be used to train large machine learning models on massive datasets, enabling organizations to build advanced artificial intelligence systems.
- Financial services: Grid computing is used in finance to perform complex financial simulations, risk assessments, and high-frequency trading.

II. <u>Cluster Computing</u>

Cluster computing involves connecting multiple computers or computing nodes to work together as a single system to perform a common task. The nodes in a cluster computing system are typically connected via a high-speed communication network and can be used to perform tasks and share resources in a coordinated manner. Cluster computing provides a way to increase the processing power and reliability of a computing system, by distributing the workload across multiple nodes. This allows for the completion of demanding computational tasks that would not be possible with a single computer.

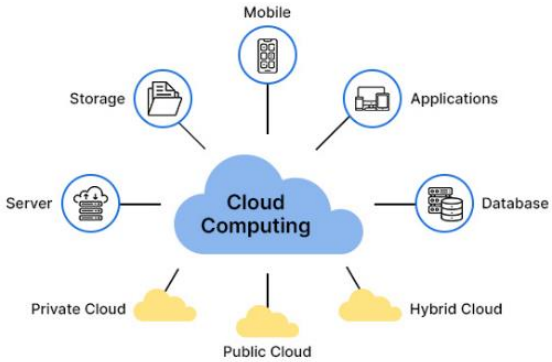


Cluster computing has a wide range of applications in various fields, some of which are:

- Data analysis: Cluster computing allows for the processing of large datasets in a parallel and distributed manner, enabling organizations to extract meaningful insights from vast amounts of data.
- Healthcare: Cluster computing is used in healthcare to support large-scale data analysis and medical imaging, helping healthcare organizations make better decisions and provide better care.
- Climate modeling: Cluster computing is used in climate science to perform complex climate simulations and modeling, helping researchers understand the impacts of climate change and develop solutions.

III. <u>Cloud Computing</u>

Cloud computing is a model for delivering computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the internet to provide faster innovation, flexible resources, and economies of scale. With cloud computing, organizations can access and use these services on demand, paying only for what they use. Cloud computing enables organizations to focus on their core business and not worry about the infrastructure, maintenance, and management of their computing resources. Instead, the infrastructure and maintenance are managed by the cloud service provider. The benefits of cloud computing include increased agility and innovation, improved collaboration, and enhanced security and compliance.

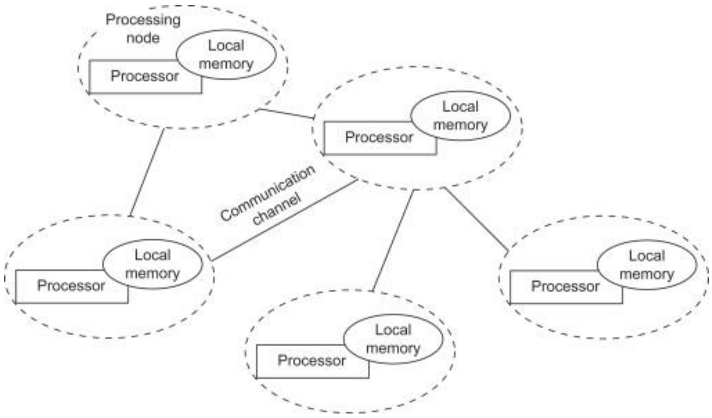


Cloud computing has a wide range of applications across various industries and business domains, some of which are:

- Web-based Applications: Cloud computing provides a platform for delivering web-based applications and services over the internet, allowing organizations to reduce the costs of software development and deployment.
- Business Productivity: Cloud computing offers a range of business productivity applications, such as email, file sharing, and project management, helping organizations to collaborate and work more efficiently.
- Data Storage and Backup: Cloud computing provides scalable and secure data storage solutions, enabling organizations to store and backup their data in the cloud, reducing the risk of data loss and improving disaster recovery capabilities..

IV. Distributed Computing

Distributed computing is a field of computer science that focuses on the development of systems and algorithms for the distribution of computational tasks and data across multiple nodes connected in a network. These nodes work together as a system to solve complex computational problems that cannot be solved by a single computer. In a distributed computing system, tasks are divided into smaller subtasks, which are then assigned to individual nodes for processing. The results of these subtasks are then combined to produce the final result.

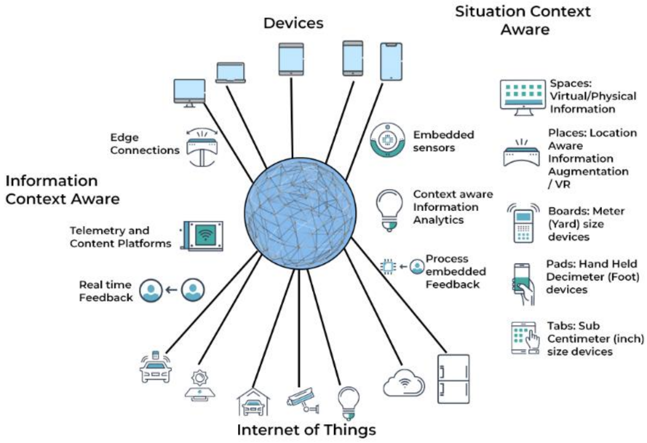


Distributed computing has a wide range of applications in various fields, including:

- Machine learning: Distributed computing provides the infrastructure for training large machine learning models, reducing the time it takes to train models and enabling organizations to tackle more complex problems.
- Energy and Climate modeling: Distributed computing enables organizations to run large-scale simulations to understand the impact of human activities on the environment and to develop strategies for mitigating these impacts.
- Grid computing: Distributed computing provides the infrastructure for grid computing, enabling organizations to solve large-scale scientific and engineering problems by tapping into a global network of computing resources.

V. <u>Ubiquitous cloud computing</u>

Ubiquitous cloud computing aims to make computing resources and services accessible from anywhere, at any time, and from any device. It refers to the integration of cloud computing into everyday life, where users can access cloud services seamlessly and effortlessly. In a ubiquitous cloud computing environment, computing resources and services are delivered from the cloud and accessed through various devices, such as smartphones, laptops, and tablets. This allows users to work and play from anywhere, at any time, without being limited by the computational capabilities of their device.



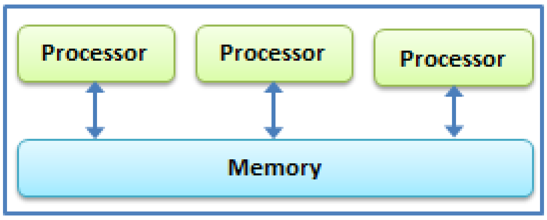
Ubiquitous cloud computing has a range of applications, including:

- Mobile computing: Ubiquitous cloud computing enables users to access cloud services and applications from their mobile devices, providing them with the ability to work and play from anywhere, at any time.
- Virtual and augmented reality: Ubiquitous cloud computing provides the infrastructure for virtual and augmented reality applications, allowing users to interact with digital content in new and innovative ways.
- IoT: Ubiquitous cloud computing provides the infrastructure for the Internet of Things (IoT), enabling organizations to manage and analyze large amounts of data generated by connected devices.
- E-commerce: Ubiquitous cloud computing provides e-commerce organizations with the ability to store and process large amounts of customer data, enabling them to deliver personalized experiences and improve customer engagement.

VI. Parallel Computing

Parallel computing is a type of computing architecture in which multiple processors or cores work together to solve a computational problem. The goal of parallel computing is to increase computational performance by breaking down a large computational problem into smaller, more manageable tasks that can be solved simultaneously by multiple processors. In parallel computing, each processor or core is assigned a specific task and operates independently of other processors, working in parallel to solve the overall computational problem. The results from each processor are then combined to produce the final solution.

Parallel Computing



Parallel computing has a wide range of applications, including:

- Gaming: Parallel computing is used in gaming for applications such as 3D graphics rendering, physics simulation, and artificial intelligence.
- Weather forecasting: Parallel computing is used for weather forecasting, enabling meteorologists to process large amounts of data in parallel to make more accurate predictions.
- Manufacturing: Parallel computing is used in manufacturing for applications such as product design, simulation, and optimization.

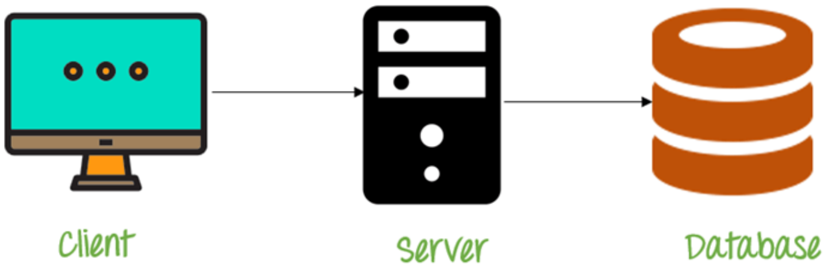
2. Give some examples of multi-tiered architectures applicable to distributed systems [Draw diagrams, mention the applications, advantages and disadvantages of the given architectures]

Multi-tiered architecture is a common approach in distributed systems, where the system is divided into multiple layers or tiers, each with a specific function. Some examples of multi-tiered architectures in distributed systems are:

I. **Three-tier architecture**:

Three-tier architecture consists of three tiers: the client tier, the application tier, and the database tier.

- Client tier: This tier represents the user interface and is responsible for user interaction. It provides a graphical user interface (GUI) through which users can interact with the system.
- Application tier: This tier is responsible for processing and handling business logic. It communicates with the client tier to receive requests and with the database tier to retrieve and store data.
- Database tier: This tier stores and manages data. The database tier is responsible for data persistence and provides the necessary data to the application tier.



Applications: Three-tier architecture is widely used in various applications, such as e-commerce websites, customer relationship management (CRM) systems, and enterprise resource planning (ERP) systems.

Advantages:

- Scalability: Three-tier architecture allows organizations to add more resources to the system as needed, making it scalable.
- Maintainability: The separation of the application and database tiers makes it easier to maintain the system, as changes to one tier do not affect the other tiers.
- Security: Three-tier architecture provides a higher level of security, as the client tier communicates with the application tier through a secure connection, and the database tier is protected by security measures such as firewalls and authentication mechanisms.

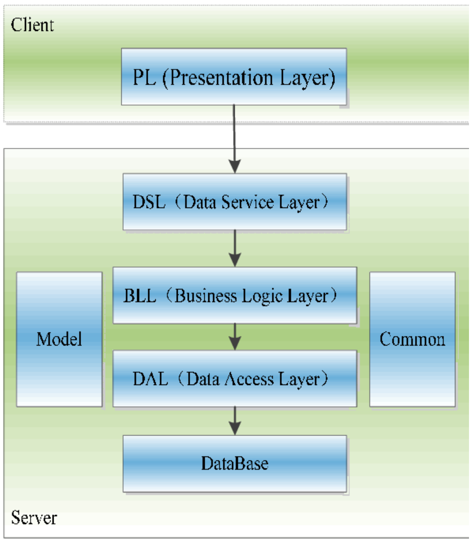
Disadvantages:

- Complexity: The increased number of tiers in the system can make it more complex, which can increase the cost and time required to develop and maintain the system.
- Performance: The added layer of communication between the client and application tiers can increase latency, reducing the performance of the system.
- Cost: The added infrastructure required for the three-tier architecture, such as the application and database servers, can increase the cost of the system.

II. **<u>Four-tier architecture</u>**:

Four-tier architecture consists of four tiers: the client tier, the presentation tier, the application tier, and the database tier.

- Client tier: This tier represents the user interface and is responsible for user interaction. It provides a graphical user interface (GUI) through which users can interact with the system.
- Presentation tier: This tier is responsible for formatting and displaying data to the user. It communicates with the client tier to receive requests and with the application tier to retrieve data.
- Application tier: This tier is responsible for processing and handling business logic. It communicates with the presentation tier to receive requests and with the database tier to retrieve and store data.
- Database tier: This tier stores and manages data. The database tier is responsible for data persistence and provides the necessary data to the application tier.



Applications: Four-tier architecture is widely used in various applications, such as e-commerce websites, customer relationship management (CRM) systems, and enterprise resource planning (ERP) systems.

Advantages:

- Maintainability: The separation of the presentation, application, and database tiers makes it easier to maintain the system, as changes to one tier do not affect the other tiers.
- Security: Four-tier architecture provides a higher level of security, as the client tier communicates with the presentation tier through a secure connection, and the database tier is protected by security measures such as firewalls and authentication mechanisms.
- Performance: The separation of the client and database tiers allows for better performance, as the presentation and application tiers can handle the processing, freeing up the database tier to handle data storage and retrieval.

Disadvantages:

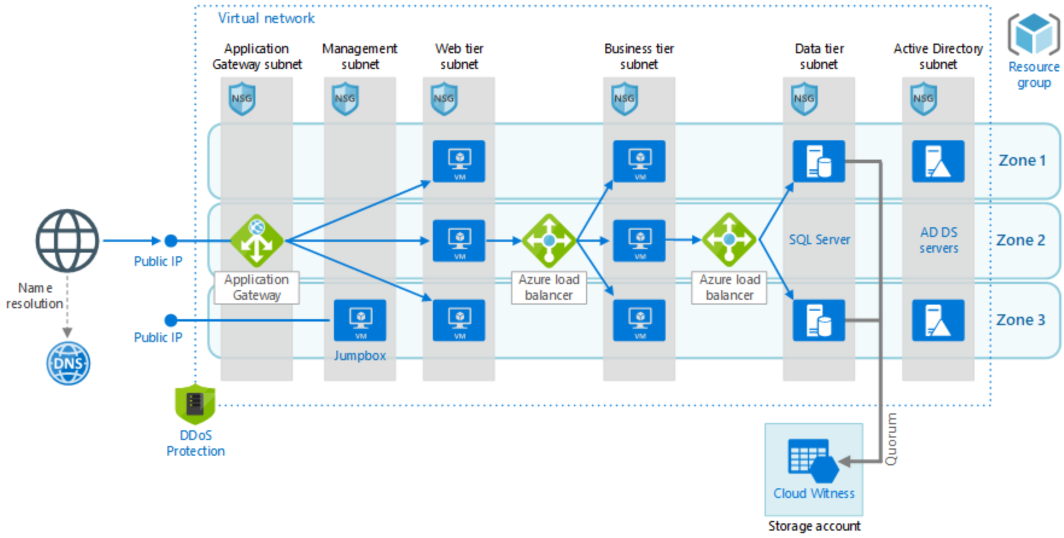
- Complexity: The increased number of tiers in the system can make it more complex, which can increase the cost and time required to develop and maintain the system.
- Performance: The added layer of communication between the client and presentation tiers can increase latency, reducing the performance of the system.
- Cost: The added infrastructure required for the four-tier architecture, such as the presentation, application, and database servers, can increase the cost of the system.

III. **N-tier architecture**:

N-tier architecture consists of multiple tiers, where each tier provides a specific service. The number of tiers can vary, but common configurations include two-tier, three-tier, and four-tier architectures. In an N-tier architecture, each tier is responsible for a specific function, and the tiers communicate with each other to provide the overall system functionality.

- Client tier: This tier represents the user interface and is responsible for user interaction. It provides a graphical user interface (GUI) through which users can interact with the system.
- Presentation tier: This tier is responsible for formatting and displaying data to the user. It communicates with the client tier to receive requests and with the application tier to retrieve data.
- Application tier: This tier is responsible for processing and handling business logic. It communicates with the presentation tier to receive requests and with the database tier to retrieve and store data.

- Database tier: This tier stores and manages data. The database tier is responsible for data persistence and provides the necessary data to the application tier.



Applications: N-tier architecture is widely used in various applications, such as e-commerce websites, customer relationship management (CRM) systems, and enterprise resource planning (ERP) systems.

Advantages:

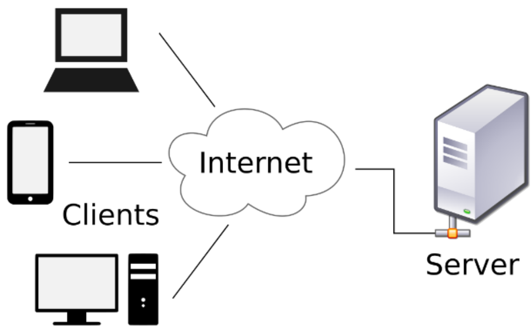
- Scalability: N-tier architecture allows organizations to add more resources to the system as needed, making it scalable.
- Security: N-tier architecture provides a higher level of security, as the client tier communicates with the presentation tier through a secure connection, and the database tier is protected by security measures such as firewalls and authentication mechanisms.
- Performance: The separation of the client and database tiers allows for better performance, as the presentation and application tiers can handle the processing, freeing up the database tier to handle data storage and retrieval.

Disadvantages:

- Complexity: The increased number of tiers in the system can make it more complex, which can increase the cost and time required to develop and maintain the system.
- Performance: The added layer of communication between the client and presentation tiers can increase latency, reducing the performance of the system.
- Cost: The added infrastructure required for the N-tier architecture, such as the presentation, application, and database servers, can increase the cost of the system.

IV. <u>Client-server architecture:</u>

Client-server architecture is a type of computing architecture that involves a client device (the client) and a server device (the server), which communicate with each other to provide a service. The client device sends requests to the server, which processes the request and returns a response.



Applications: Client-server architecture is widely used in various applications, such as email, file sharing, and web applications. In these applications, the client device is typically a personal computer or a mobile device, and the server is a computer that provides the requested service.

Advantages:

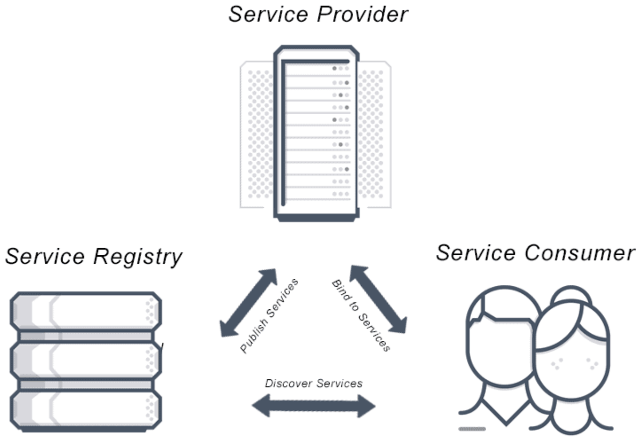
- Scalability: The client-server architecture allows organizations to add more resources to the system as needed, making it scalable.
- Centralized data management: The server device is responsible for managing the data, which makes it easier to maintain data consistency and security.
- Improved performance: The server device can provide faster processing and data retrieval than the client device, which can improve the overall performance of the system.
- Improved security: The server device can be protected by security measures such as firewalls and authentication mechanisms, which can increase the security of the system.

Disadvantages:

- Dependence on server: The client device is dependent on the server for processing and data retrieval, which can lead to downtime or slow performance if the server is unavailable.
- Increased complexity: The client-server architecture can increase the complexity of the system, which can increase the cost and time required to develop and maintain the system.
- Security risks: The client-server architecture can introduce security risks, as the server device is responsible for managing sensitive data, and it can be vulnerable to hacking or other security breaches.

V. **Service-oriented architecture (SOA):**

Service-oriented architecture (SOA) is a software design pattern that organizes software systems into services that can be reused and composed to create new applications. These services can be independently deployed and managed, and they communicate with each other through standard protocols, such as XML or HTTP.



Applications: SOA is widely used in various applications, such as e-commerce systems, customer relationship management systems, and supply chain management systems. In these applications, SOA provides a flexible and scalable way to integrate different systems and services.

Advantages:

- Reusability: Services can be reused across different applications, reducing the cost and time required to develop and maintain the system.
- Flexibility: SOA allows organizations to add or change services as needed, making it flexible and adaptable to changing business requirements.
- Interoperability: SOA enables communication between different systems and services through standard protocols, increasing the interoperability of the system.

Disadvantages:

- Complexity: SOA can increase the complexity of the system, which can increase the cost and time required to develop and maintain the system.
- Security risks: SOA can introduce security risks, as the communication between services is vulnerable to hacking or other security breaches.
- Integration challenges: Integrating different systems and services can be challenging, and it requires a well-defined strategy and proper testing.

These are some of the examples of multi-tiered architectures that are commonly used in distributed systems. The choice of architecture depends on the specific requirements of the system, such as scalability, security, and performance.

References

1. Sharma, Neha & Prabha, Chander. (2021). Computing Paradigms: An Overview. 1-6. 10.1109/ASIANCON51346.2021.9545007.
2. A. Waheed et al., "A Comprehensive Review of Computing Paradigms, Enabling Computation Offloading and Task Execution in Vehicular Networks," in IEEE Access, vol. 10, pp. 3580-3600, 2022, doi: 10.1109/ACCESS.2021.3138219.
3. https://www.geeksforgeeks.org/different-computing-paradigms/
4. https://www.newagepublishers.com/samplechapter/002724.pdf
5. https://www.infor.uva.es/~jjalvarez/asignaturas/SD/lectures/chapter3.pdf
6. Wang, Yu-Ling. (2014). Research on multi-layer distributed system architecture based on middleware technology. 687-693. 10.2495/CECNet130801.
7. https://www.sciencedirect.com/topics/computer-science/multitier-architecture
8. Kistijantoro, Achmad Imam & Morgan, Graham & Shrivastava, Santosh & Little, M.. (2008). Enhancing an Application Server to Support Available Components. Software Engineering, IEEE Transactions on. 34. 531-545. 10.1109/TSE.2008.38.
9. Erickson, John & Siau, Keng. (2009). Service Oriented Architecture: A Research Review from the Software and Applications Perspective. Innovations in Information Systems Modeling: Methods and Best Practices. 190-203. 10.4018/978-1-60566-278-7.ch010.
10. https://www.javatpoint.com/service-oriented-architecture
11. https://intellipaat.com/blog/what-is-client-server-architecture/
12. Yuxin Liu, Kaixue Yao, Daoyun Xu. Automatic Code Generation Based in Template on .Net Framework of Three Layers Architecture [J]. Computer Technology and Development, 2012, 22(8) 13-14