

**كلية** : الحاسبات والمعلومات

**عنوان البحث**

**Distributed Software Engineering**

**ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ**

* **كليـــــــــة : الحاسبات والمعلومات**
* **القسـم / الشعبـــة : علوم الحاسب**

* **الفـــرقــــة : الثالثة**
* **اســم المقـــرر : هندسة برمجيات-2**

* **اســــم الطالــب : حاتم محمد ابراهيم الشريف**
* **الرقم القومــــي : 29905011603795**
* **البريد الاكـاديمي : hatem.mohamed.688@ci.menofia.ed.eg**
* **اشــــــــراف : أ.د / أشرف السيسي**

# **Abstract**

Distributed systems are complex systems because these systems are not depending on single processor for execution but depends on more than devices with different processors, the operations in these systems is difficult and complex as more than systems are work in dependently in network. Client-server computing model and architecture is model that composed of client (user-device) and server (is device has special software that can serve and provide service to client remotely).

# **Introduction and Research Objectives**

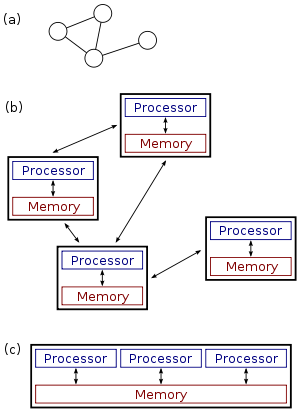
Distributed software engineering we will discuss the distributed systems and the case that make us to go to those systems and some problems and issues that face the Distributed systems like that: transparency, openness, scalability, security policies, quality. We will discuss the client server model architecture and the relation between this model and distributed systems is that most or all distributed system in web are based on client-server model with some layers as presentation, data handling, application processing, database. We will discuss the patterns that commonly used in distributed systems to organize like master-slave architecture, two-tier client-server architecture, multi-tier client-server architecture, distributed component architecture, peer to peer architecture , we will discuss software as a service which mean that: deploy application or software product in cloud or hosting the application and provide away to access it remotely through internet by client pc through any web browser , this services has different limit types and most is paid .you can choice your plane the suitable for your business size and cost.

# **Content**

* Discuss Distributed systems.
* Discuss Distributed system Advantages.
* Discuss Distributed system Drawbacks.
* Discuss How parts of Distributed System Interact.
* Discuss Client Server Architecture.
* Discuss Client Server Layers.
* Discuss Patterns Used in Distributed system.
* Discuss the Notion of (SAAS).
* Discuss Example to Distributed system (cloud computing);
* Discuss Distributed systems.

Distributed system is a software composed of group of devices connected together to

through network or the internet, the connection and interaction (devices call others) between these devices is through passing messages and all devices work to achieve the goal. The idea of distribution is that most processors don't work in utilization mode so that they decided to share processors in additional to effect on the user he using this device through special software that control the device and accept the message with the certain task to execute it and return the required response to integrated with the other response from other devices to achieve the target.



* Discuss Distributed system Advantages.

1. Sharing device resources eg (RAM - CPU)
   1. That improve the performance of system.
2. Fault tolerance
   1. System can still work if any failure happened
3. Scalability
   1. Is one of most challenges that meet the people who work in system development.
   2. We can scale the system through providing.

more resources (infrastructure).

1. Ability to use Devices from multi vendors.
2. Appearing Cloud computing depends on distributed system which provide all type of services remotely as software – storage – infrastructure – data which you can safe your data and increase the bandwidth any time if your app need that.

* Discuss Distributed system Drawbacks.

1. Transparency
   1. Users should use system as a single unit and the distribution process he shouldn't be aware to it and the needed service should know it is not depending on a distributed system.
   2. Because of the network delay users can’t notice this process.
2. openness
3. Number of connected devices un countable (unlimited).
4. Systems can call each other if it's developed by different programming language through web service (SOA) developed to open standard

eg (we can send and receive data to different types of system and application in json format) which most web services use it.

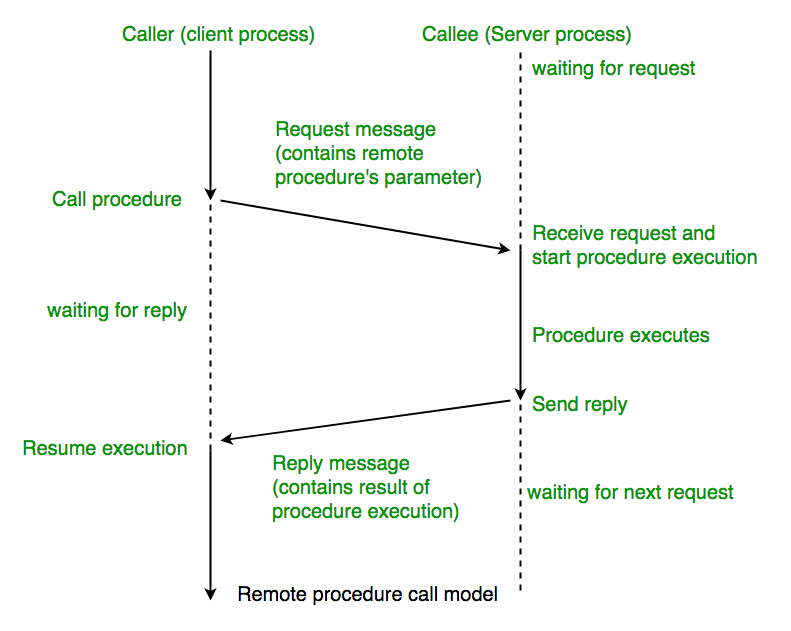
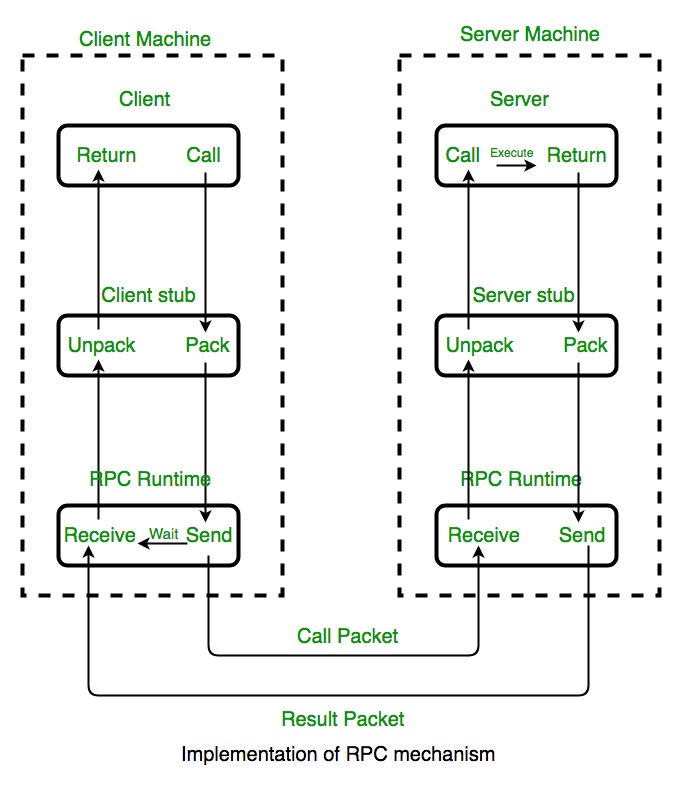
1. Quality of service
2. Quality of service important in streaming operation (video – Audio) and real time application this case is good if client accepted to the time that the system used to return the response.
3. The variety in devices, network, systems, software and operating system is bad

but the get over through some middle software called middleware and protocols.

1. Occurring any failure in the system is a drawback that face distributed system in design phase so that they should provide a tool or solution or mechanism to solve this problem immediately automatically to return the service as if no failure happened.

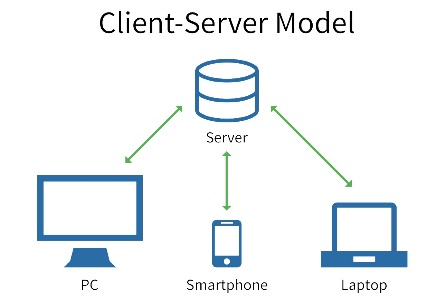
* Discuss How parts of Distributed System Interact.

1. Remote procedure calls or function call used in client server model which client call a service in his device this service is executed in another device or component in network may be is a server provide a certain service this server make processing in this request and return response to the client or to the caller the process may be in the same component or device or different devices but if we have example to RPC in device the two operation not executed at the same time but the sender or the request side blocked after send the request then the receiver execute the request then return response back to sender . this mechanism is powerful in distributed systems but has a problem is that each of sender and receiver must be available in the time that needed to perform the operation.

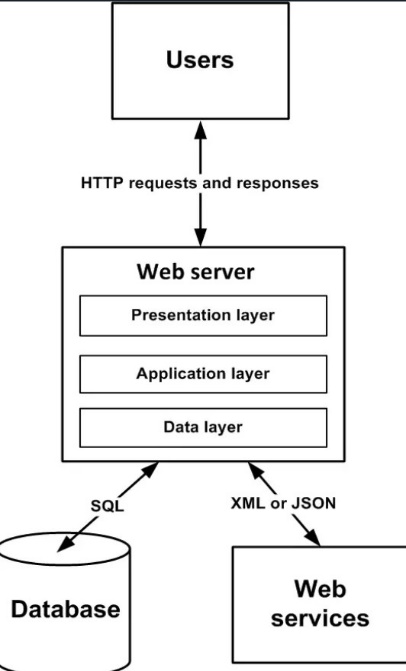


2-Message passing in this interaction there is a middleware is a software responsible for connect between sender and receiver and exchange data between them. sender prepare message that contain all details that required to be performed by the other component the middleware arrive this message to receiver , the receiver or the other part after receive the message make parsing operation to know the required operation and execute it then make or generate the result message that contain the response of the requests. each of sender and receiver may be not aware to the other as this job based on the middleware. the programming language or operating system or processors doesn't important in this case. middleware helps system components to work together providing a layer of functionality for data consistency and b2b integration. Typically, different application can communicate using messaging tool as soap, webs service, rest api, json and xml.

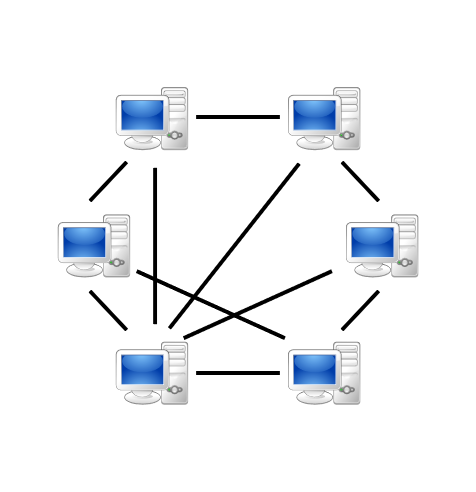
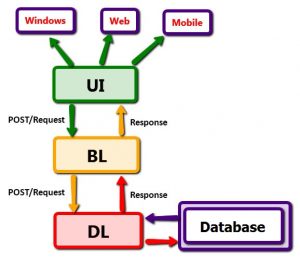
* Discuss Client Server Architecture

****most distributed system in internet based on client server model client refer to end user who need a service or resources and server (device as our special devices but has more powerful capabilities as cpu-ram-gpu-hdd) is the service provider to end user which has the service or resources needed. there is application running in client device as mobile application or web browser through it we can request the certain service and on the other side there is application (software) running in server the server depends on a software that provide only one service as web server that work by http protocol or file server by ftp or mail server with protocols as smtp pop3 imap4 etc.

* Discuss Client Server Layers.
  + - 1. presentation layer

1. manage user interaction.
2. present information to user.
3. Data handling
4. manage the transferred data to and from client.
5. check data and generate web pages.
6. Application processing layer
7. implement application logic.
8. provide functionality to end user.
9. Database
10. store data and provide transaction services.

* Discuss Patterns Used in Distributed system.

1. Master-slave is a model of communication where one device called the master has un directional control to one or more other devices and the master device is elected by some of eligible devices that called slaves. master used as a controller that is responsible for identify which slave will handle the request. This model is used in real time applications which response time in this case is required to be in real time.
2. Two-tier client server architectures composed of two tiers or layers tier 1 is the end user or client who need to access some resources after request this resources wait until the tier 2 or service provider or the server the device or machine with a certain software that different according to the type of service which it provide to response after the server listen to any incoming request it make some process to the this request as example the authorization and some other process before starting in the process of end user after execute the request return response formatted as required or as the development prepared it. the 2 tier has 2 different model thin-client and fat-client. in thin-client model the first layer is implemented on client and the other layers implemented on the server.in fat-client model the first two layer implemented on client and last two layers implemented on the server that due to the business logic is needed to more processing and may be process on client (not needed to devices with high capabilities) to reduce the process to server to make processing in the main function and application core but this model or this systems make loading or more traffic in each network and servers.so that fat client make more processing on client to reduce the traffic for both network and server in fat client model we need the client device with more capabilities than thin but has a problem is that if there is any enhancement in application or development to new release or version we need to remove the last version and install the new in all machines. example for suitable application for thin client is web browsers and in fat client atm machine.
3. Multi-tier client server architecture or n-tier (2-3-4-5-etc.) n refer to the number of layers suitable for application providing solution to scalability, security and fault tolerance, reusability and maintainability in 3 tier model or plus we provide a new server called application server we can use it to make processing in additional to use the model of thin and fat client but most logic in application server so that we can reduce the traffic to server and processing as we have a special machine to process and any new versions in system we change it in one place in application server.
4. ****Peet 2 Peer architecture is a group of computers connected to each other with a certain topology this pcs share it resources as storage or processing, this model used to share files as BitTorrent or other system as database or communication by phone.

* Discuss the Notion of (SAAS)

software and associated data are centrally hosted on the cloud and we can access the resources from any client device through thin client application as web browser. this server that provide this service is managed and owned by a software provider and the organization that use this software or service. software is paid so that users can pay according to the plan the provided by the provider monthly or annually and determine the size of service you need according to the amount of use. we use soa as saas. soa is a some of separated or distributed service it as traditional web service but has different features which through it we can build our service depending on some of other services.

Example to SAAS is Amazon Web Services (AWS) provides a low cost, reliable, and secure foundation for you to use as you build and deliver Software as a Service (SaaS) solutions to customers.

**References**

# **Software Project Management in Practice**

* by Pankaj Jalote
* Released January 2002
* ISBN: 0201737213