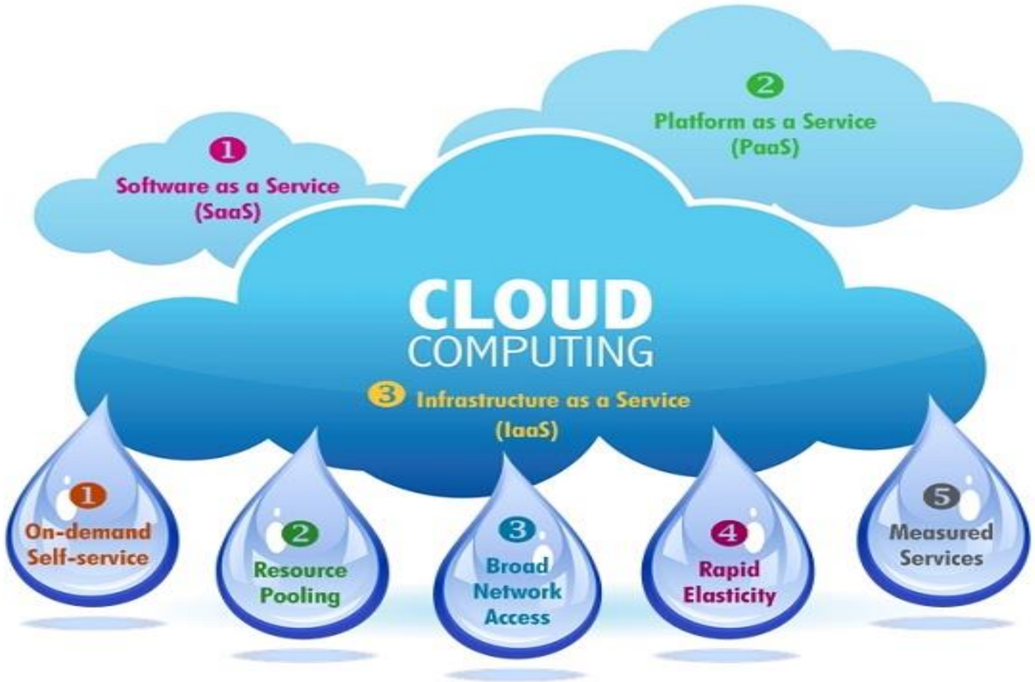
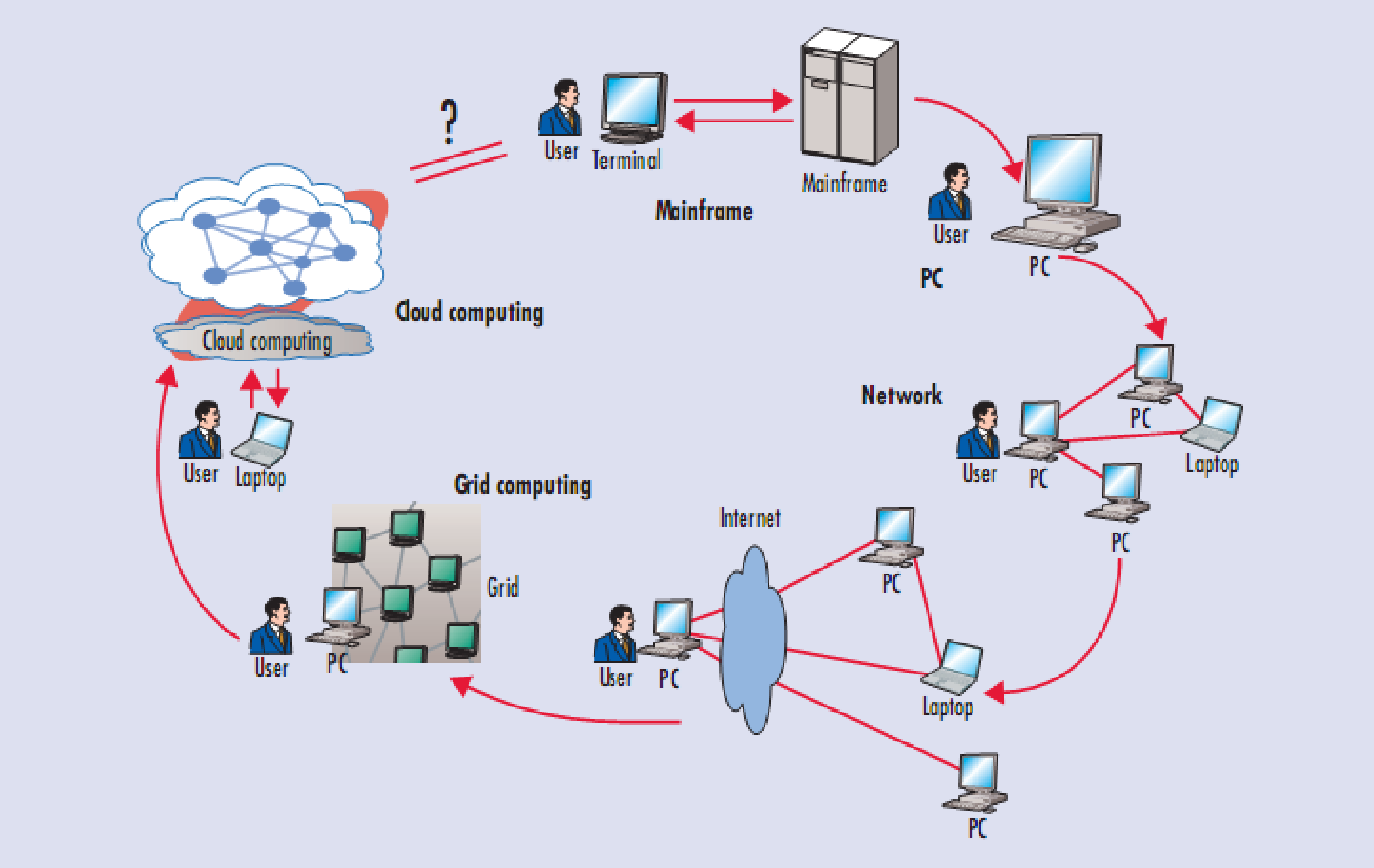


PMCA506L: Cloud Computing

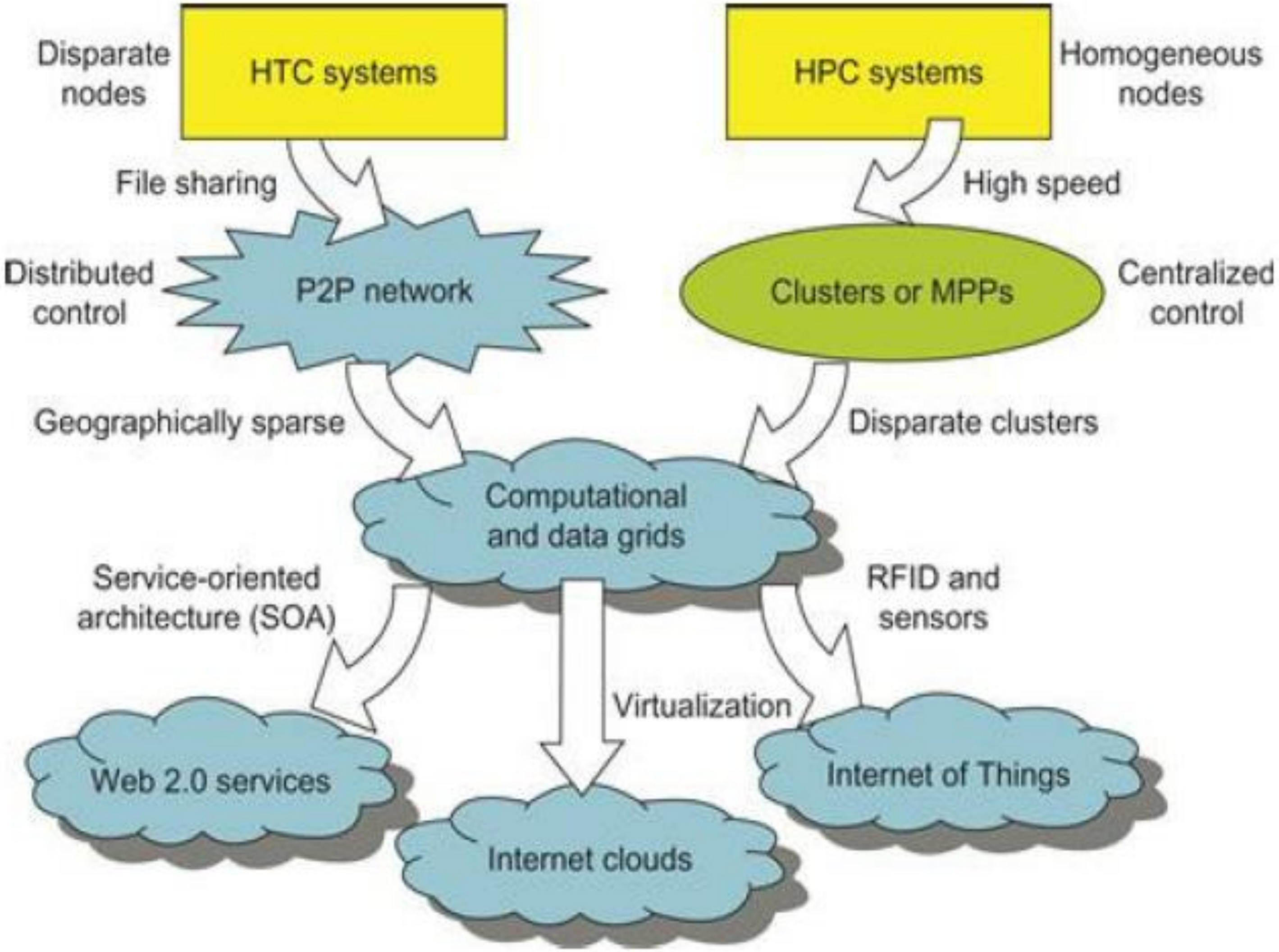
Module 1 : Cloud Computing Paradigms

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Computing Paradigm Shift



HTC and HPC


Disparate nodes
HTC systems
HPC systems
Homogeneous nodes
File sharing
High speed
Distributed control
P2P network
Clusters or MPPs
Centralized control
Geographically sparse
Disparate clusters
Computational and data grids
Service-oriented architecture (SOA)
RFID and sensors
Virtualization
Web 2.0 services
Internet of Things
Internet clouds

High Performance Computing

- HPC systems emphasize the raw speed performance.

- The speed of HPC systems has increased from Gflops –Pflops- Tflops – Eflops-Zflops-Yflops

- **Floating Point Operations Per Second**

- SuperComputer - https://www.datacenterknowledge.com/cloud/5-reasons-cloud-repatriation-should-be-part-digital-transformation

- https://en.wikipedia.org/wiki/FLOPS

- The fastest high-performance computing system in the world is currently the Frontier-Cray system at Oakridge National Laboratory, United States. This has a peak speed of **1 exa-flop (or about 1,000 petaflops)**.

High-Throughput Computing

- Measure *high throughput* or the number of tasks completed per unit of time.

- The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously

Computing Paradigm Distinctions

- Centralized computing
- Parallel computing
- Distributed computing
- Cloud computing

Centralized computing

- Computing paradigm by which all computer resources are centralized in one physical system.

- All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS.

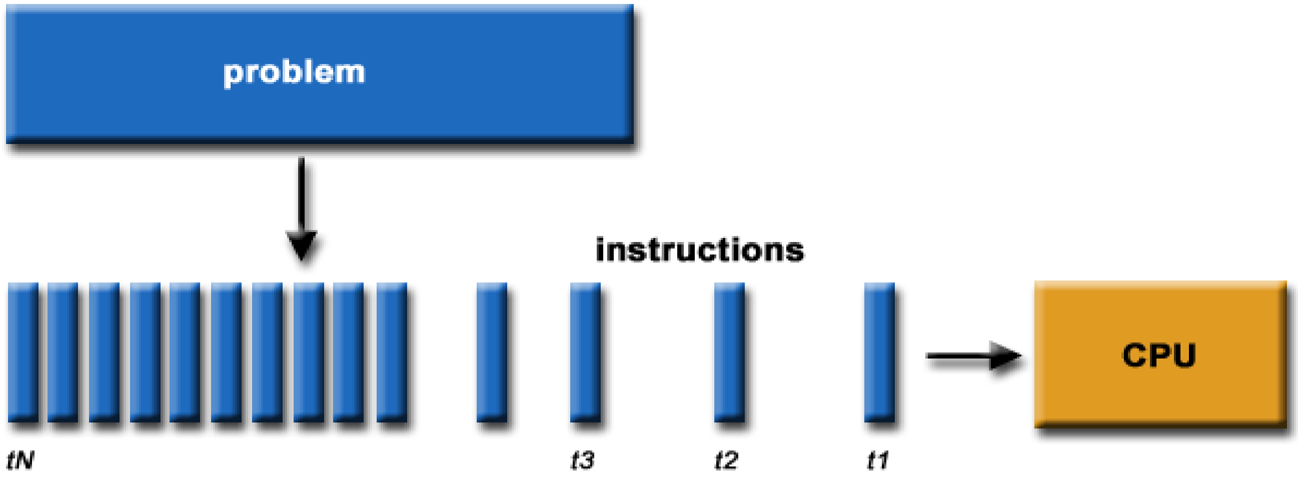
- Many data centers and supercomputers are *centralized systems*, but they are used in parallel, distributed, and cloud computing applications

Parallel Computing

- In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory.

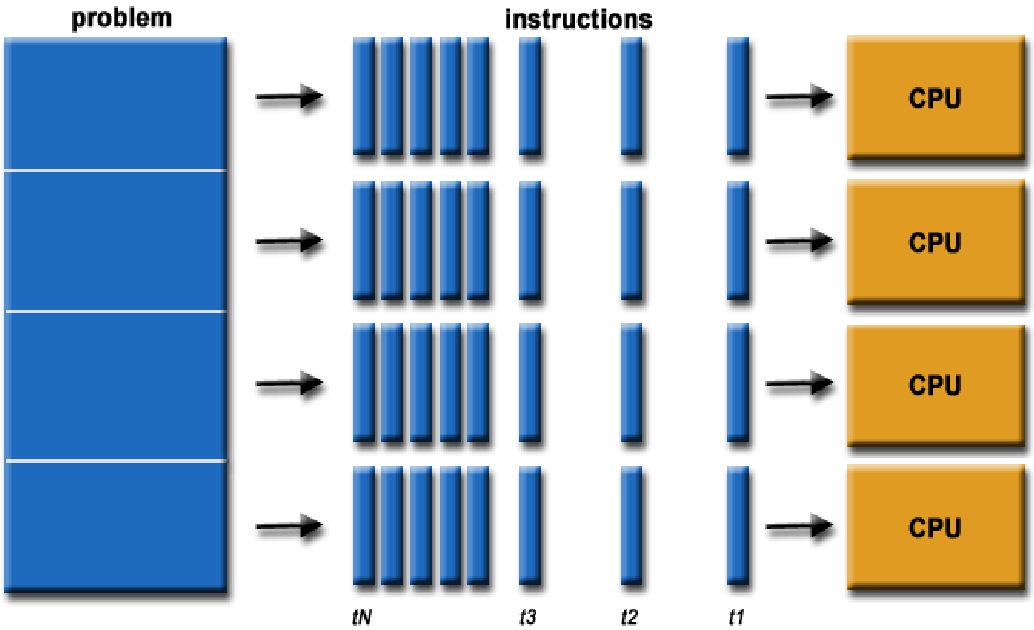
What is Parallel Computing?

- Traditionally, software has been written for ***serial*** computation:
  - To be run on a single computer having a single Central Processing Unit (CPU);
  - A problem is broken into a discrete series of instructions.
  - Instructions are executed one after another.
  - Only one instruction may execute at any moment in time.



What is Parallel Computing?

- In the simplest sense, ***parallel computing*** is the simultaneous use of multiple compute resources to solve a computational problem.
  - To be run using multiple CPUs
  - A problem is broken into discrete parts that can be solved concurrently
  - Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different CPUs

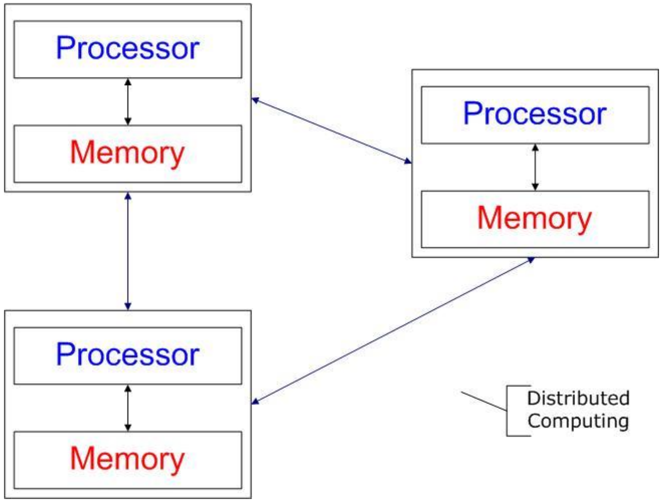


Parallel Computing: Resources

- The compute resources can include:
  - A single computer with multiple processors;
  - A single computer with (multiple) processor(s) and some specialized computer resources ((GPU, Field Programmable Gate Arrays(FPGA)…)
  - An arbitrary number of computers connected by a network;
  - A combination of both.

Distributed computing

- Multiple autonomous computers, each having its own private memory, communicating through a computer network.

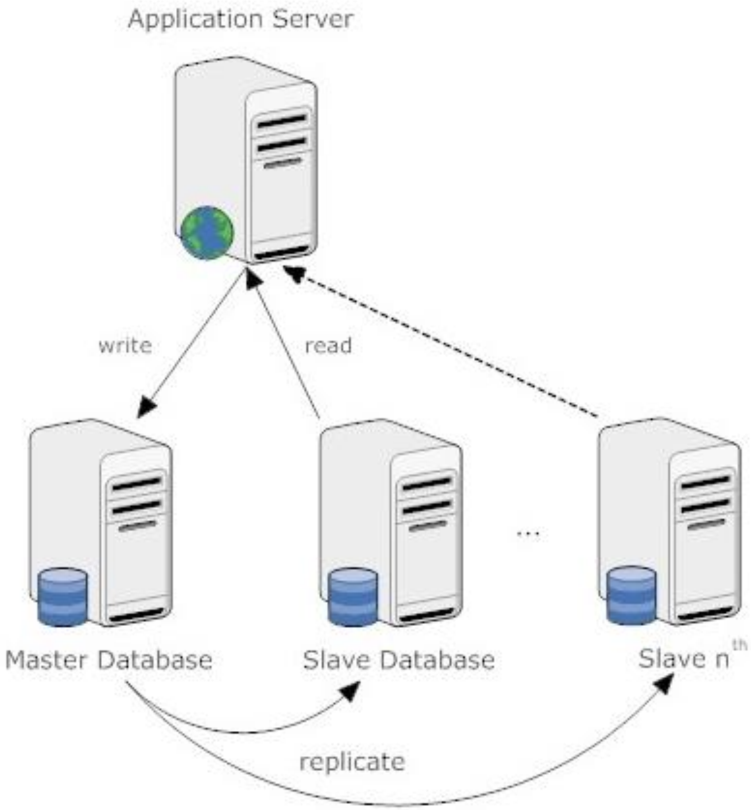


What is Distributed Computing/System?

- Common properties
  - Fault tolerance
    - When one or some nodes fails, the whole system can still work fine except performance.
    - Need to check the status of each node
  - Resource sharing
    - Each user can share the computing power and storage resource in the system with other users
  - Load Sharing
    - Dispatching several tasks to each nodes can help share loading to the whole system.
  - Easy to expand
    - We expect to use few time when adding nodes. Hope to spend no time if possible.

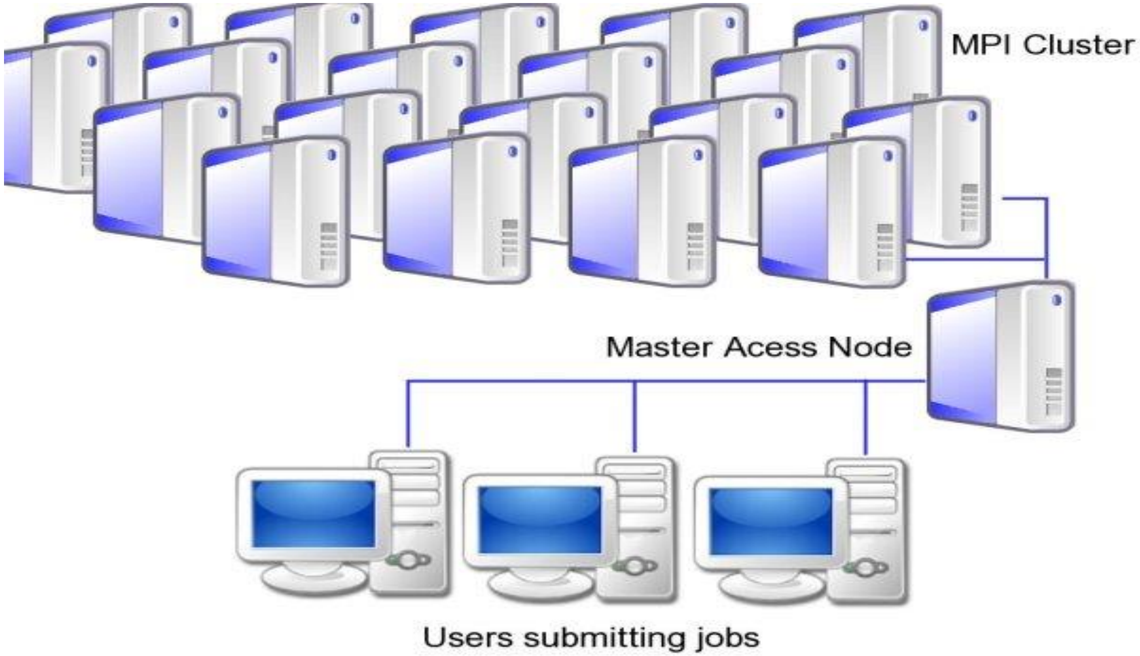
Common Architectures

- Master/Slave architecture
  - Master/slave is a model of communication where one device or process has unidirectional control over one or more other devices
    - Database replication
      - Source database can be treated as a master and the destination database can treated as a slave.
    - Client-server
      - web browsers and web servers



Classification of Distributed Computing Systems

- These can be classified into 4 groups: clusters, peer-to-peer networks, grids, and clouds.
- A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource. The network of compute nodes are connected by LAN/SAN and are typically homogeneous with distributed control running Unix/Linux. They are suited to HPC.
- **Message Passing Interface** (MPI)



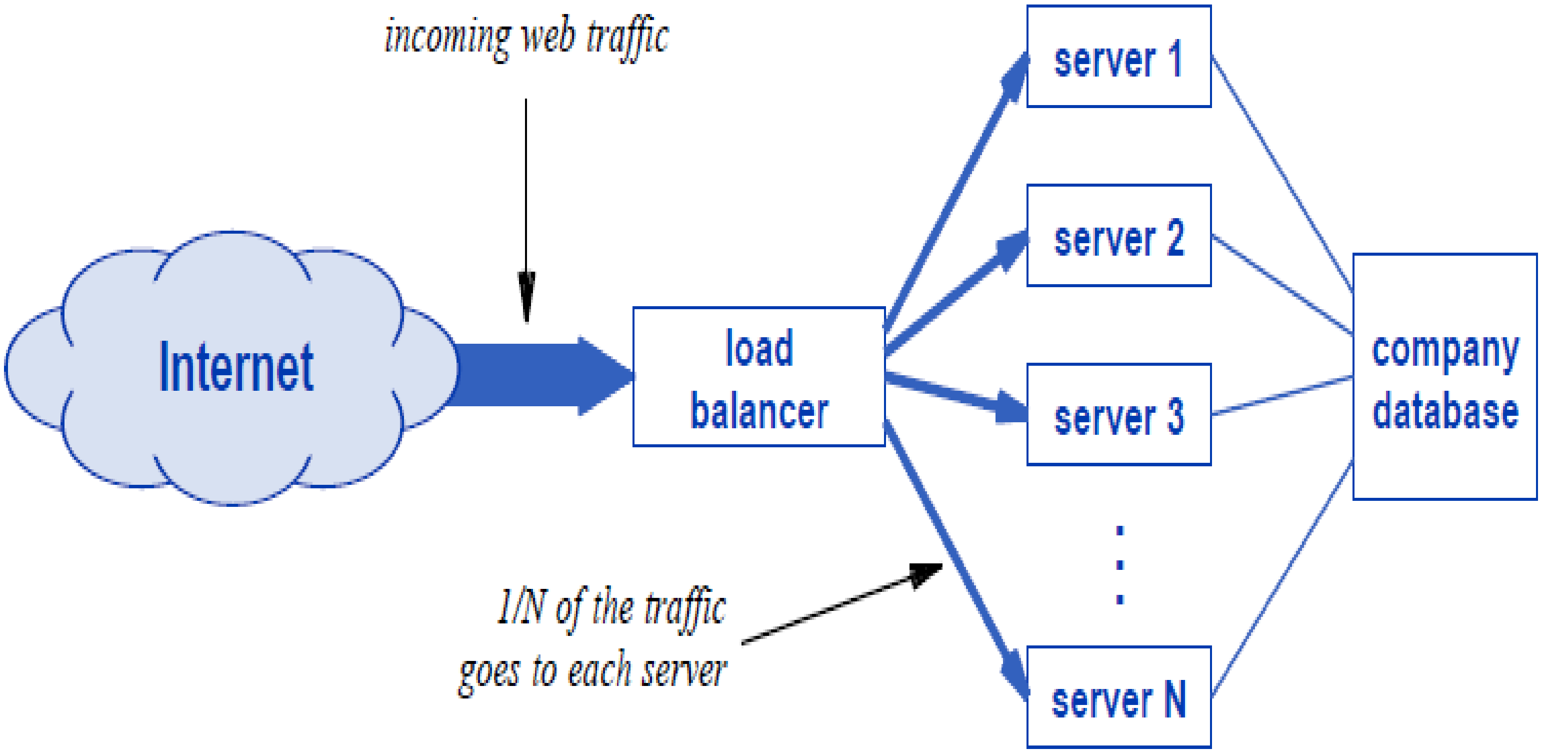
What is a cluster?

- A cluster is a type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers cooperatively working together as a single, integrated computing resource.
- A typical cluster:
  - Network: Faster, closer connection than a typical network (LAN)
  - Low latency communication protocols
  - Looser connection than SMP (symmetric multiprocessing-SMP
  - (symmetric multiprocessing is computer processing done by multiple processors that share a common operating system (OS) and memory))

From Clusters To Web Sites And Load Balancing

- Web sites and scientific computing systems differ in a fundamental way.
- Super- computer clusters intended for scientific calculations are designed so that small computers can work together on one computation at a time.
- In contrast, a web site must be designed to process many independent requests simultaneously.

Load Balancer



Racks Of Server Computers

- A server rack houses and organizes critical IT systems, which can be configured to support a wide range of requirements.

Data Center

- A data center is a physical room, building or facility that houses IT infrastructure for building, running, and delivering applications and services.

- Storing and managing the data associated with those applications and services.

- Availability of high-speed computer networks

Economic Motivation For A Centralized Data Center

- Operating expenses (opex):

  lower recurringcost

- Capital expenses(capex):

  lower equipmentcost

Cloud computing

- An *Internet cloud* of resources can be either a centralized or a distributed computing system.
- The cloud applies parallel or distributed computing, or both.
- Clouds can be built with physical or virtualized resources over large data centers that are centralized or distributed
- *Elastic Computing*

Advantage of Clouds over Traditional Distributed Systems

- Traditional distributed computing systems provided for on-premise computing and were owned and operated by autonomous administrative domains (e.g. a company).

- These traditional systems encountered performance bottlenecks, constant system maintenance, poor server (and other resource) utilization, and increasing costs associated with hardware/software upgrades.

- Cloud computing as an on-demand computing paradigm resolves or relieves many of these problems.

Multi-Tenant Clouds

- *Multi-tenant* refer to a datacenter that serves customers from multiple organizations.

- Cloud provider builds a data center (or multiple data centers) that can handle computing for many customers.

- Technologies used in cloud systems are designed to support multi-tenant computing and keep the data of each customer safe.

Situations Where Latency Matters

- Is latency important?
- When a corporation performs routine business (e.g., recording sales transactions or submitting monthly payroll information), a slight delay is unnoticeable.
- Small delay in making stock trades can result in a huge loss.
- Small delay in receiving data from a patient monitor can delay activation of an implanted medical treatment device.

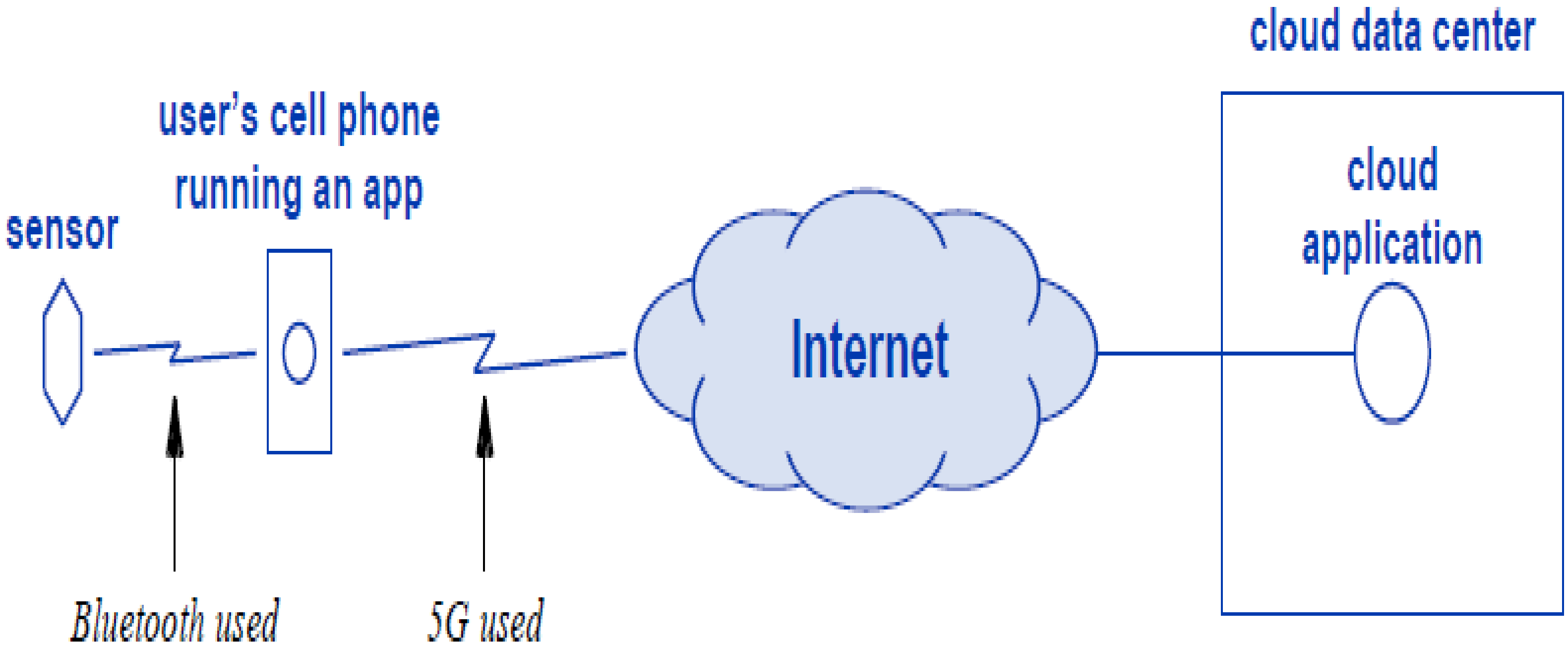
Moving Computing To The Edge

- How can cloud computing be adapted to meet the requirements for low latency? ***Edge Computing***

- Keep computing facilities near each source of information, and perform initial processing locally.

- Simultaneously run applications in a cloud data center, and use the cloud applications to handle computational-intensive tasks.

Moving Computing To The Edge



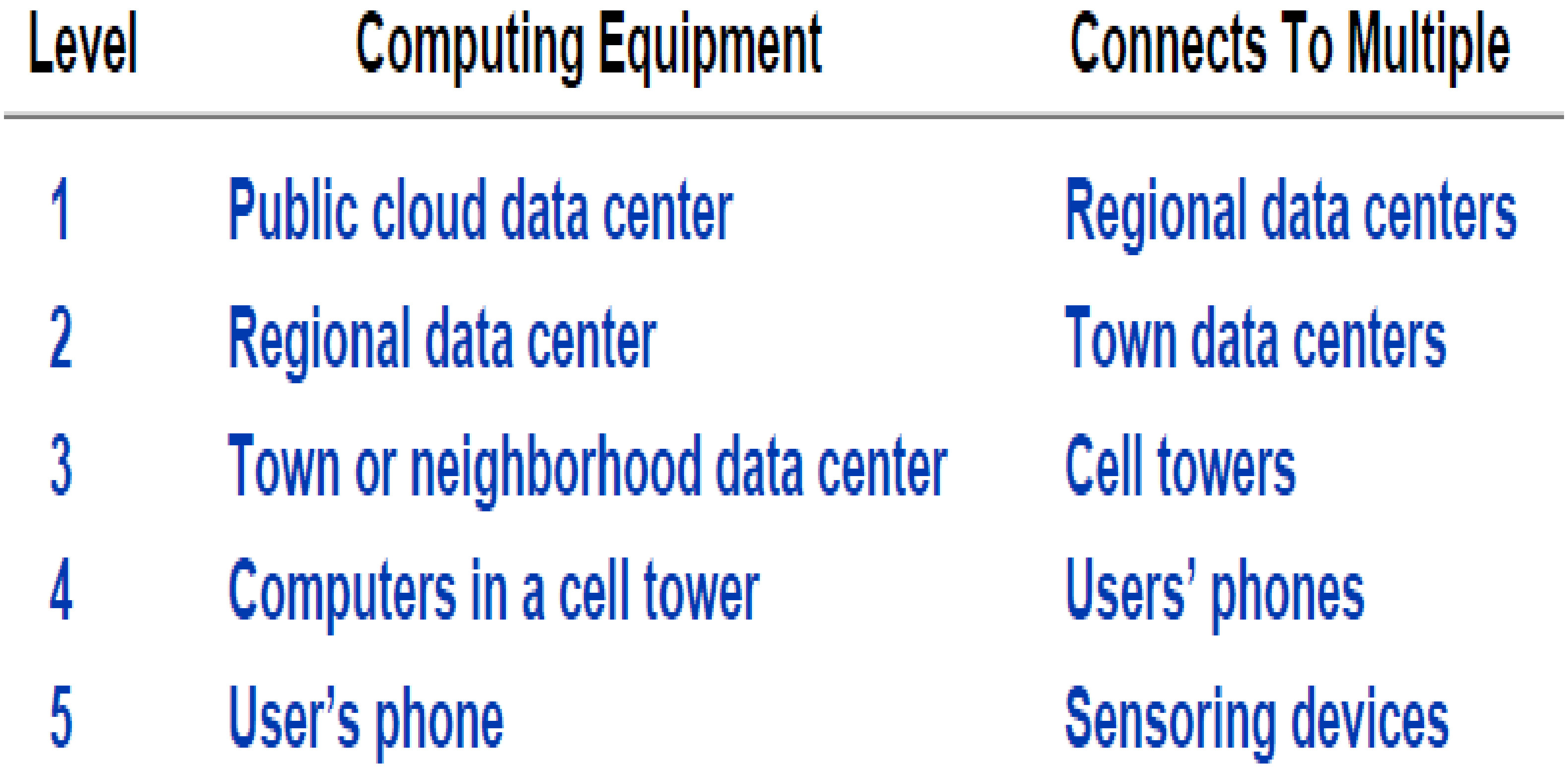
Extending Edge Computing To A Fog Hierarchy

Where should edge datacenters be placed?

- The locations and sizes depend on the applications being supported and the latency requirements.

- To achieve the lowest possible latency ,an edge facility must be as close to each user as possible.

Edge Computing Hierarchy



Level	Computing Equipment	Connects To Multiple
1	Public cloud data center	Regional data centers
2	Regional data center	Town data centers
3	Town or neighborhood data center	Cell towers
4	Computers in a cell tower	Users' phones
5	User's phone	Sensoring devices

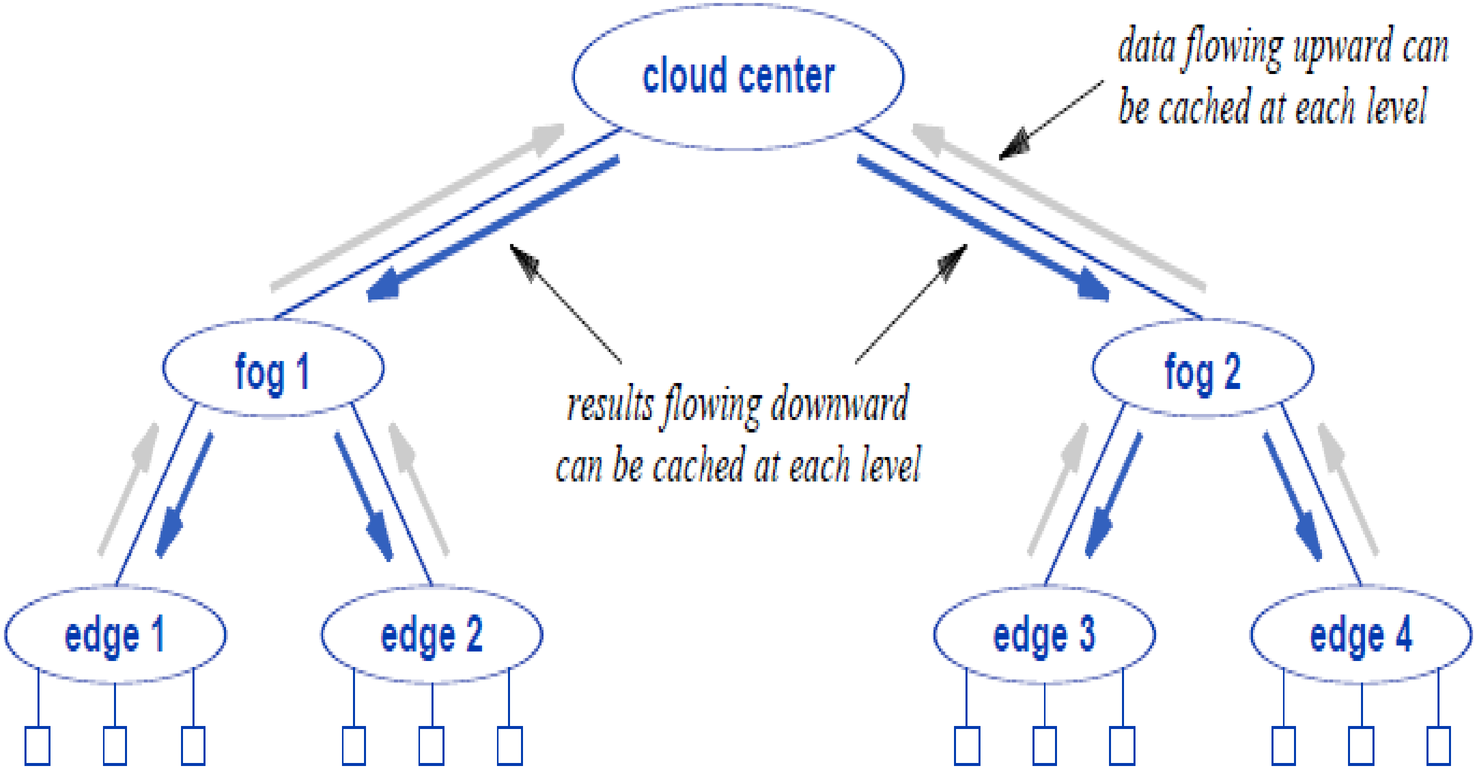
Fog & Edge Data Center

- Edge datacenter for a small datacenter directly adjacent to endpoints.

- Fog datacenter refer to an intermediate datacenter in an edge hierarchy.

- This distinguish between edge facilities located adjacent to end users and edge facilities that serve larger geographic regions

Caching At Multiple Levels Of A Hierarchy


cloud center
data flowing upward can be cached at each level
fog 1
fog 2
results flowing downward can be cached at each level
edge 1
edge 2
edge 3
edge 4

An Automotive Example

- *Connected Vehicles*

- Once the system becomes operational, each vehicle, whether self- driven or driven by a human, will communicate with near by vehicles as well as with communication facilities permanent placed near roadways

- Three aspects of the connected vehicle system lend themselves to the edge computing approach.

Connected Vehicles- Edge Computing Approach

- Low latency/real-time requirements
- Geographic locality and awareness
- The wide scope needed for route planning and navigation

The envisioned system for connected vehicles illustrates how a hierarchy of small edge and fog data centers can provide low-latency responses and manage information over a range of geographic areas.

Web x.x

- Web 0.0 – Developing the internet
- Web 1.0 – The shopping carts & static web
- Web 2.0 – The writing and participating web
- Web 3.0 – The semantic executing web
- Web 4.0 – “Mobile Web”
- Web 5.0- Open, Linked and Intelligent Web = Emotional Web

Web Service

“Web services” is an effort to build a distributed computing platform for the Web.

Applications that enable remote procedure calls over a network or the Internet often using XML and HTTP

This allows us to hide the details of how a service is implemented; only URL and data types are required

It is largely irrelevant to the client whether the service is developed with Java or ASP.NET or if it is running on Windows, Linux or any other platform

Concepts of Web services

- Web services is a messaging system which allows communication between objects.

- Messages can be synchronous or asynchronous.

- This system is loosely coupled (ie. Services should not be dependent on each other).

Contd..

- Services offered by one application to another through www
- A business application sends a request to a service at a given URL using SOAP (Simple Object Access Protocol) over HTTP.
- Service receives the request ,process and return as response.

  Example : Stock Quote Price
- **Users are mainly from B2B transactions**

Service Oriented Architecture (SOA)

- Webservices – Shared Organizing Prinicples
- SOA organizes as - (YP)

  Service Provider

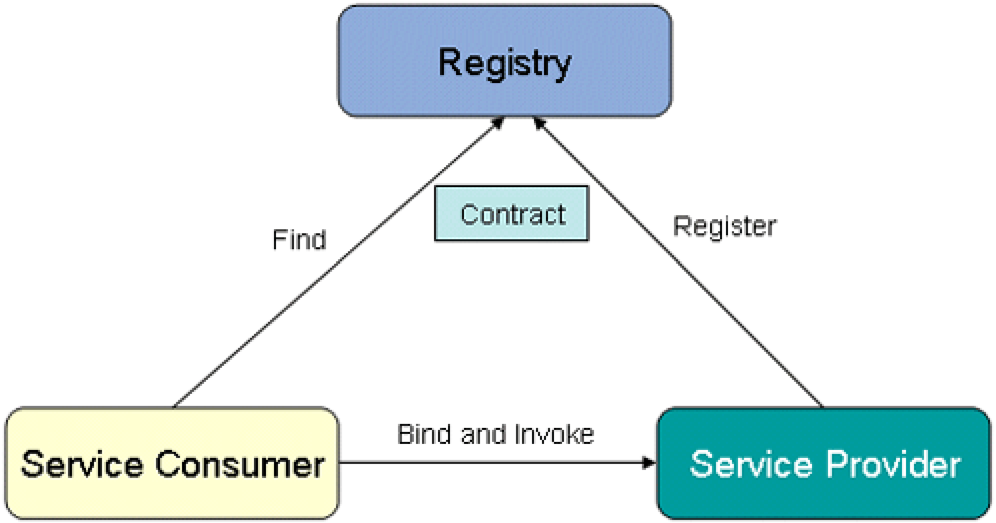
  Service Registry

  Service Requester

(Collection of Services)

Web services roles and relationship

- Just in Time Integration



Key Functional Components (4)

- **Service Implementation**
  ->Develop WS & Interface
  ->Publish Interface & Deploy WS

- **Publication**
  -> Author WS Description Document
  -> WSDL
  -> UDDI
  (Universal Description Discovery Integration)

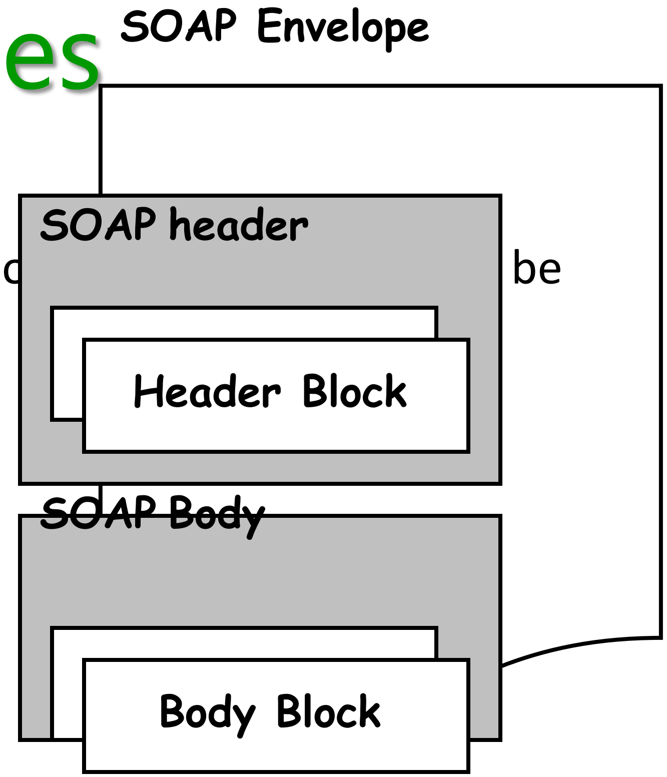
- **Discovery** --(UDDI looks up client query)

- **Invocation** – (CLIENT –SOAP – Remote Web service)

Web service Building Blocks - SOAP

- A “wrapper” protocol
- Written in XML
- Independent of the wrapped data
- Independent of the transport protocol
- Efficient (according to the W3C)
- A uni-directional message exchange paradigm

SOAP messages



- SOAP is based on message exchanges
- Messages are seen as envelops where the applicatio be sent
- A message has two main parts:
  - header: which can be divided into blocks
  - body: which can be divided into blocks
- SOAP does not say what to do with the header and the body, it only states that the header is optional and the body is mandatory
- Use of header and body, however, is implicit. The body is for application level data. The header is for infrastructure level data

Why XML?

- Simple text markup language
- Platform, language and vendor agnostic
- Easily extensible
- Capable of solving interoperability problem

Summary

- Computing Paradigm Shift
- Centralized, Parallel and Distributed Systems
- Cluster, Grid, P2P Systems
- Cloud Computing- Multitenant Cloud
- Edge and Fog Computing
- Web Services