

CLOUD COMPUTING

DSE-3157

**DEPARTMENT OF DATA SCIENCE AND COMPUTER APPLICATIONS,
MANIPAL INSTITUTE OF TECHNOLOGY,
MANIPAL ACADEMY OF HIGHER EDUCATION**

WHAT IS CLOUD COMPUTING?



On-demand
self-service

No human
intervention
needed to get
resources



Broad network
access

Access
from
anywhere



Resource
pooling

Provider
shares
resources
to
customers



Rapid
elasticity

Get more
resources
quickly as
needed



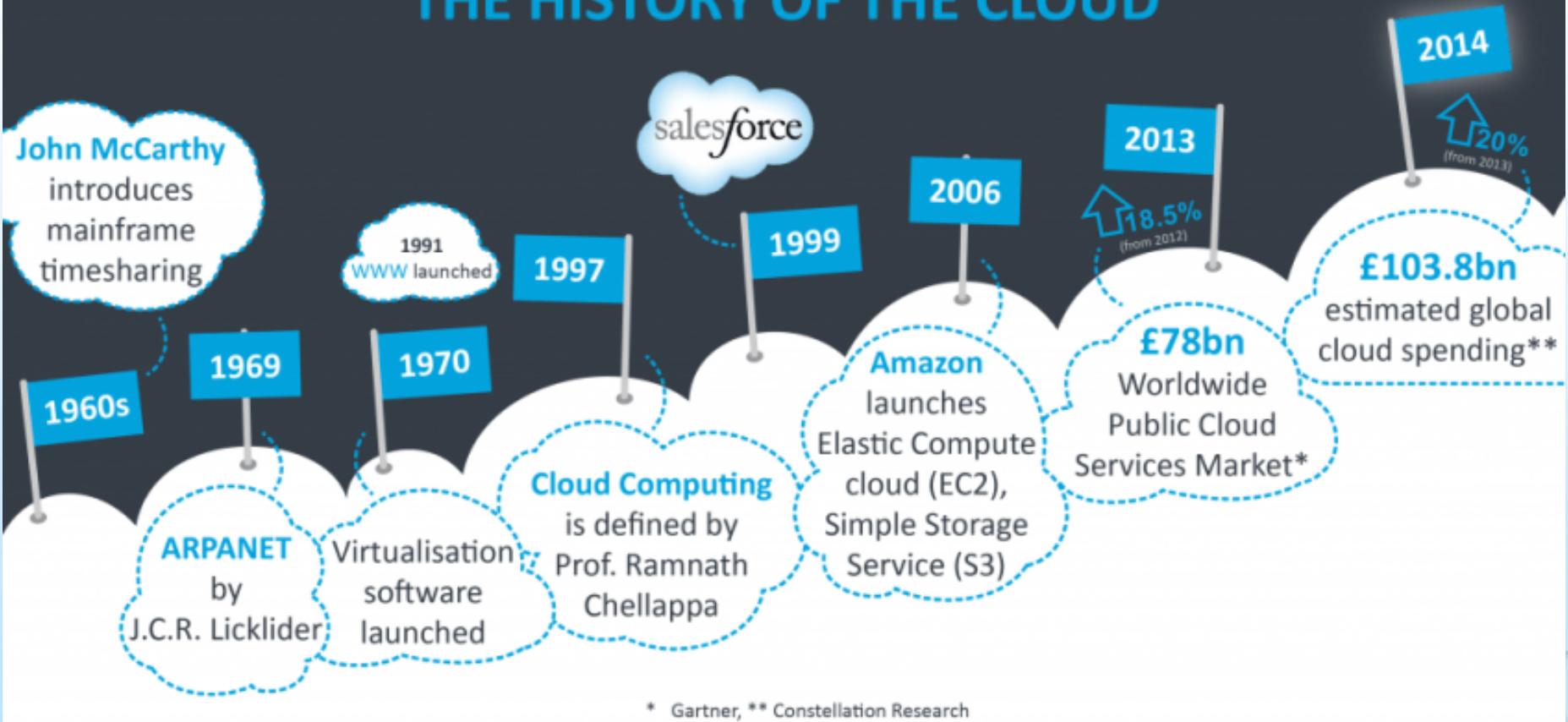
Measured
service

Pay only
for what
you
consume

 Google Cloud

EVOLUTION OF CLOUD

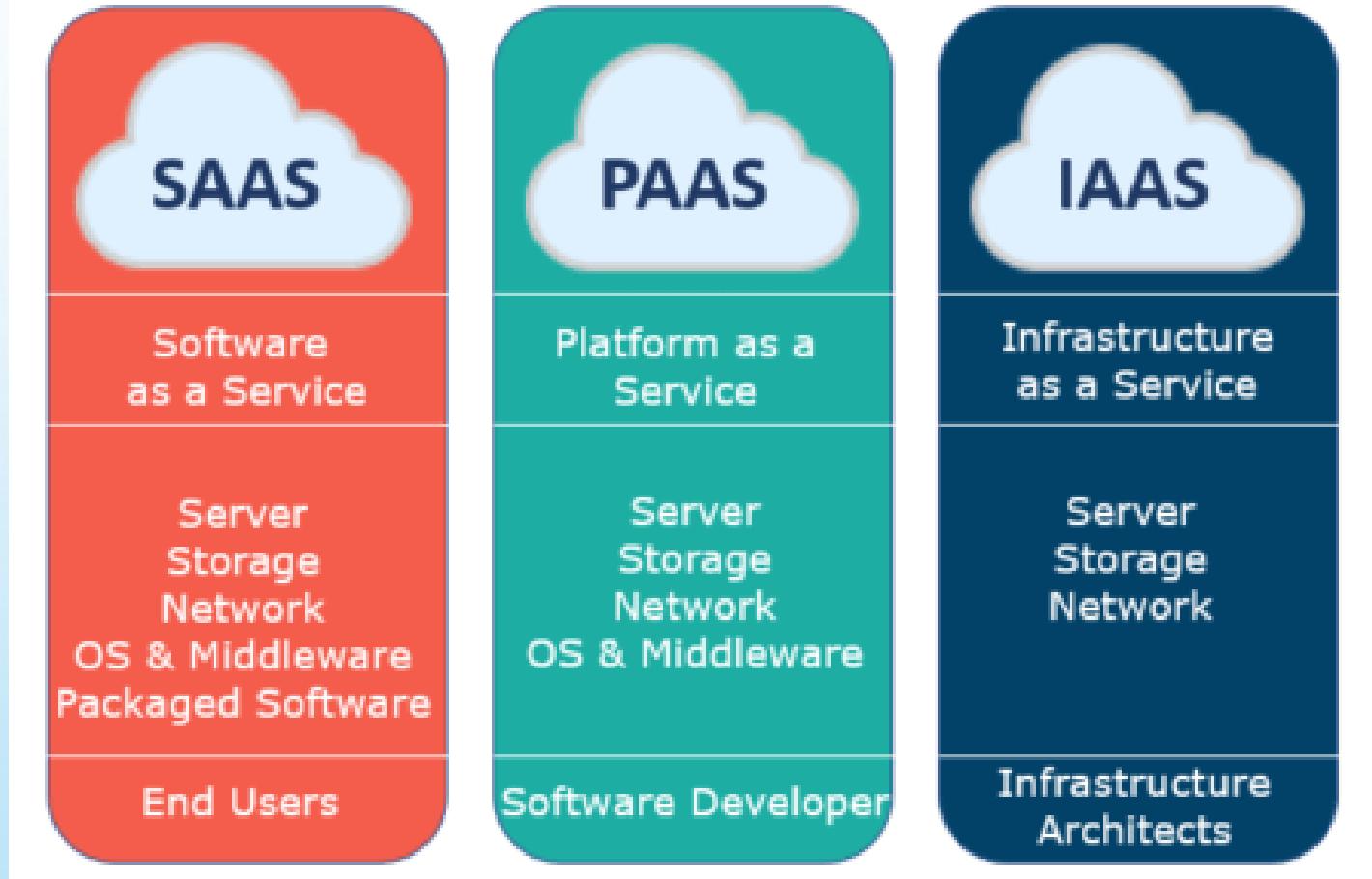
THE HISTORY OF THE CLOUD



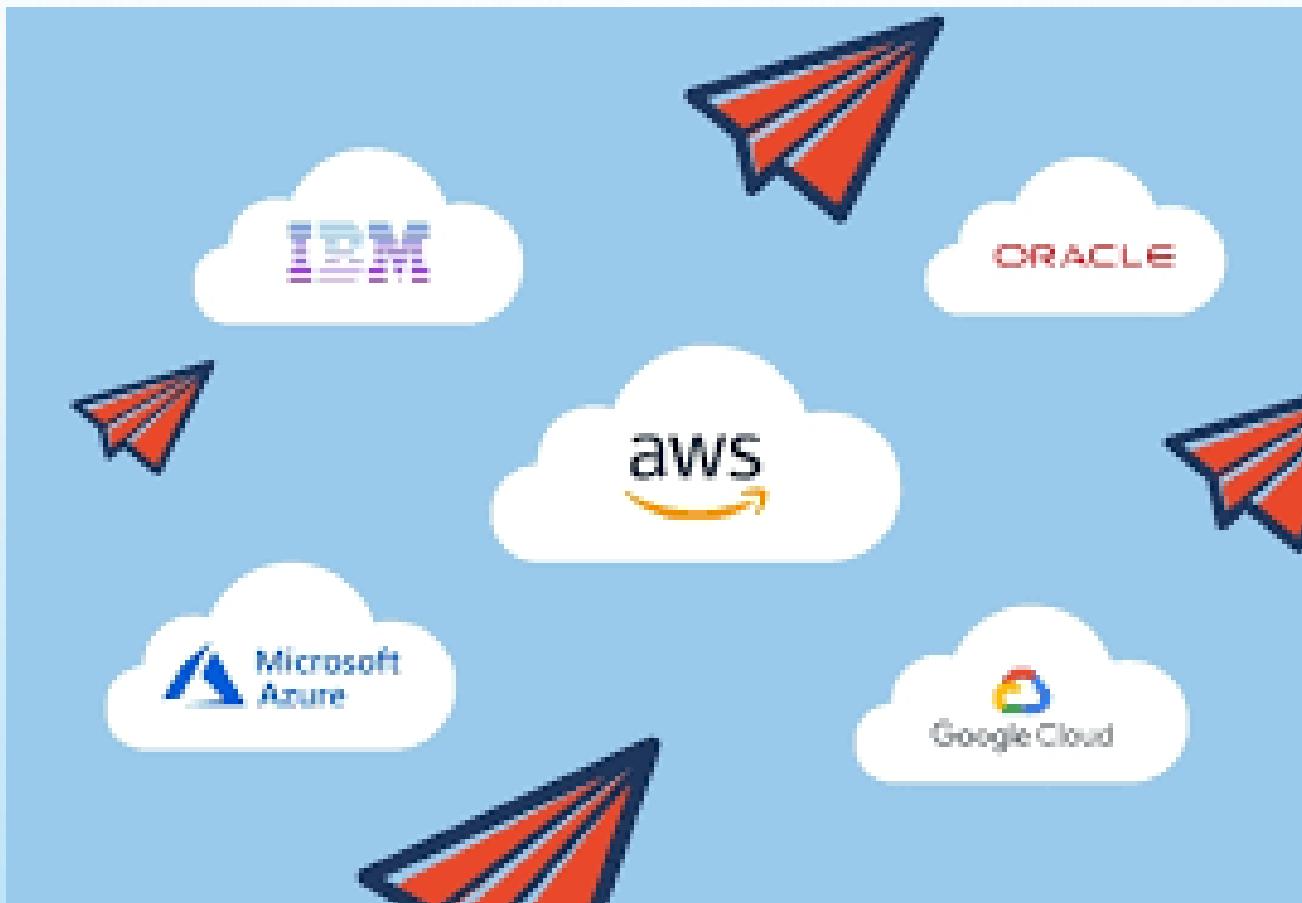
BENEFITS OF CLOUD



TYPES OF SERVICES



KEY CLOUD SERVICE PROVIDERS



COURSE STRUCTURE

Course Objectives

- Differentiate Between Conventional Computing Systems And Cloud Computing Systems.
- Explain The Role And Essentials Of Virtualization In The Cloud-enabling Technologies
- Interpret The Issues Related To Service-oriented Architecture.
- Analyze Various Cloud Programming Models And Their Security To Apply Them To Solve Problems On Real-time Cloud Applications.

COURSE STRUCTURE

INTRODUCTION TO CLOUD COMPUTING

- Cloud Computing In A Nutshell
- Roots Of Cloud Computing
- Layers And Types Of Clouds
- Desired Features Of A Cloud
- Cloud Infrastructure Management
- Infrastructure As A Service Providers , Platform As A Service Providers ,Challenges And Risks, Public Clouds, Private Clouds, Hybrid Clouds.

COURSE STRUCTURE

VIRTUALIZATION & INFRASTRUCTURE AS A SERVICE

- Understanding Virtualization: Describing Virtualization, Importance Of Virtualization, Understanding Virtualization Software Operation.
- Introduction To Hyper Converged Infrastructure: Definition, Resources To Consolidate.
- Architecting The Hyper Converged Data Center: Server Support, Software Defined Storage
- The Role Of Custom Hardware In A Commodity Infrastructure Hyper Convergence And The Public Cloud: Public Cloud, Private Cloud The Intersection Of Cloud And Hyper Converged Infrastructure,
- Hyper Convergence And The Private Cloud Virtual Machines Provisioning And Migration Services:

COURSE STRUCTURE

SERVICE ORIENTED ARCHITECTURES

- Services And Service Oriented Architectures,
- Message-oriented Middleware,
- Portals And Science Gateways, Discovery, Registries, Metadata, And Databases, Workflow In Service-oriented Architectures

COURSE STRUCTURE

CLOUD PROGRAMMING AND SOFTWARE ENVIRONMENTS

- Features Of Cloud And Grid Platforms,
- Parallel And Distributed Programming Paradigms, Programming Support Of Google App Engine, Programming On Amazon AWS And Microsoft Azure
- SLA MANAGEMENT: Inspiration, Traditional Approaches To SLA Management, Types Of SLA, Life Cycle Of SLA, SLA Management In Cloud, Automated Policy-based Management.

COURSE STRUCTURE

CLOUD SECURITY

- Cloud Computing Security Architecture.
- Data Security
- Network Security
- Host Security, Compromise Response

REFERENCE BOOKS

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, *Cloud Computing Principles and Paradigms*, Wiley Publications, 2013.
2. Kai Hwang, Geoffrey Fox, Jack Dongarra, Todd Green, *Distributed and Cloud Computing: From Parallel Processing and the Internet of Things*, Morgan Kaufmann Publishers, 2012
3. George Reese, *Cloud application architectures: building applications and infrastructure in the cloud*, O'Reilly Media, Inc., 2009.
4. Matthew Portnoy, *Virtualization Essentials*, John Wiley and Sons Publication, 2012
5. Scott D Lowe, *Hyper-converged Infrastructure implementation strategies*, Actual Tech media, 2015
6. Thomas Erl, *Service-Oriented Architecture Principles of Service Design*, Prentice-Hall, 2008

NEXT CLASS....

INTRODUCTION TO CLOUD COMPUTING..

INTRODUCTION TO CLOUD COMPUTING

LECTURE -1 TO LECTURE-4

TOPICS

- Introduction to Cloud Computing: Cloud Computing in a Nutshell
- Roots of Cloud Computing,
- Layers and Types of Clouds
- Desired Features of a Cloud,
- Cloud Infrastructure Management
- IaaS,
- PaaS,
- Challenges and Risks

WHAT IS CLOUD COMPUTING?

Cloud Computing is the **Delivery of Computing Services**—servers, storage, databases, networking, software and more—over The Internet.

Companies Offering these Computing Services are called **Cloud Providers** And Typically Charge for Cloud Computing Services Based **On Usage**, Similar to how you are billed for water or electricity at home.
---- **Pay As You Use.**

GENERIC DEFINITIONS

➤ By Buyya et al.,

“Cloud is a **parallel and distributed computing** system consisting of a collection of inter-connected and virtualized computers that are **dynamically provisioned** and presented as one or more unified computing resources **based on service-level agreements (SLA)** established through negotiation between the **service provider and consumers.**”

➤ By Vaquero et al.,

“Clouds are a large pool of easily usable and accessible **virtualized resources** (such as hardware, development platforms and/or services). These resources can be **dynamically reconfigured** to adjust to a variable load (scale), allowing also for optimum resource utilization. This pool of resources is typically exploited by a **pay-per-use model** in which guarantees are offered by the **Infrastructure Provider by means of customized Service Level Agreements.**”

GENERIC DEFINITIONS (CONTD..)

➤ By National Institute of Standards and Technology (NIST)
“...a **pay-per-use model** for enabling available, convenient, **on-demand network access** to a **shared pool of configurable computing resources** (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with **minimal management effort or service provider interaction.**”

➤ Armbrust et al.
“..data center hardware and software that provide services.”

.... Many more definitions are found

CLOUD CHARACTERISTICS

- Pay-per-use (no ongoing commitment, utility prices);
- Elastic capacity and the illusion of infinite resources;
- Self-service interface; and
- Resources that are abstracted or virtualized.

NOTE:

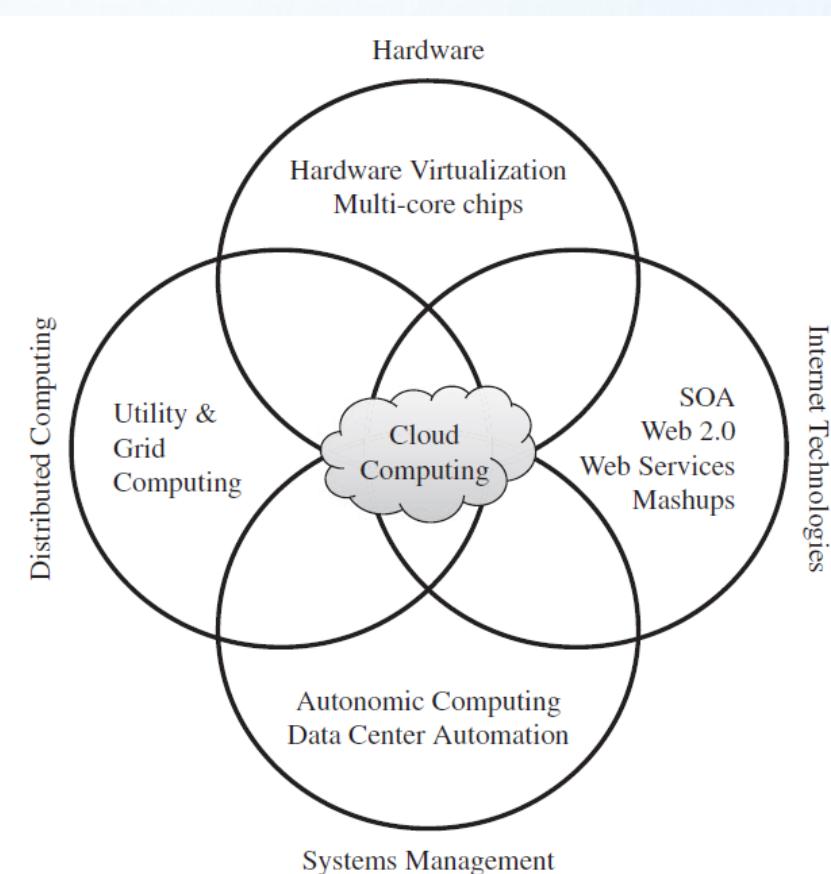
Elasticity is defined as "the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner

Self-service means that the consumer performs all the actions needed to acquire the service himself/herself, instead of going through an IT department

ROOTS OF CLOUD COMPUTING

Roots of cloud computing can be tracked by observing the advancement in technologies....

1. Hardware (virtualization, multi-core chips)
2. Internet technologies (web services, service-oriented architectures, web 2.0)
3. Distributed computing (clusters, grids)
4. Systems management (autonomic computing, data center automation).



ROOTS OF CC (CONTD..)

The technologies that form base of cloud computing ecosystem are:

1. From mainframes to clouds
2. SOA, web services, web 2.0 and mashups
3. Grid computing
4. Utility computing
5. Hardware virtualization
6. Virtual appliances and the open virtualization formats (OVM)
7. Autonomic computing

MAINFRAME COMPUTERS

- Mainframes are a type of computer that generally are known for their **large size, amount of storage, processing power and high level of reliability.**
- A **single mainframe can replace** dozens or even hundreds of smaller servers.
- Widely used areas: **banking, finance, health care, insurance, public utilities, government**
- Mainframes are able to **handle large workloads**, and their **reliability, scalability, and performance** make them the system of choice for many organizations **for bulk processing, handling transactions, and for maintaining vital business records.**

MAINFRAME COMPUTERS (CONTD..)

- Mainframes had to operate at very high utilization rates and were very expensive.
- The mainframe era collapsed with the advent of fast and inexpensive microprocessors.
- Next Cloud Computing have sprung up, allowing applications to be run via third-party infrastructure that requires little or no hardware investment and services to be paid for as they are needed.
- The advent of fast fiber optic networks, helped enabling sharing of computing power over great distances.

• SOA, WEB SERVICES, WEB 2.0, AND MASHUPS

- Service-oriented architecture (SOA) is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications
 - What's key to these services is their loosely coupled nature; i.e., the service interface is independent of the implementation.
 - For example, a service can be implemented either in .Net or J2EE, and the application consuming the service can be on a different platform or language.

• **SOA, WEB SERVICES, WEB 2.0, AND MASHUPS (CONTD..)**

- **WS standards have been created on top of HTTP and XML**
 - Providing a common mechanism for delivering services
 - Making them ideal for implementing a service-oriented architecture (SOA)
- **Web services can glue together applications running on different messaging product platforms**
 - Enabling information from one application to be made available to others

• SOA, WEB SERVICES, WEB 2.0, AND MASHUPS (CONTD..)

- SOA usually linked to Web Services Description Language (WSDL) and Simple Object Access Protocol (SOAP) specifications.
- The WS model of SOA uses the
 - WSDL to connect interfaces with services.
 - SOAP to define procedure or component APIs.
- RESTful web service -- is based on representational state transfer (REST) technology, an architectural style and approach for communications used in web services development.

• SOA, WEB SERVICES, WEB 2.0, AND MASHUPS (CONTD..)

- REST technology is generally preferred to the more robust technology because REST leverages less bandwidth, making it more suitable for internet usage.
- Many service providers make their service APIs publicly accessible using standard protocols like SOAP and REST
- Amazon, Facebook, and Google

• SOA, WEB SERVICES, WEB 2.0, AND MASHUPS (CONTD..)

- In the consumer Web, information and services may be programmatically aggregated
 - Acting as building blocks of complex compositions, called service mashups
 - Services like user authentication, e-mail, payroll management, and calendars are examples
 - Can be reused and combined in a business solution
 - Popular APIs are combined to produce a variety of interesting solutions
 - Google Maps, YouTube, Amazon ecommerce, and Twitter

• SOA, WEB SERVICES, WEB 2.0, AND MASHUPS (CONTD..)

A mashup is a Web page or application that uses and combines data, presentation or functionality from two or more sources to create new services.

Combination with Google Maps HousingMaps: **combines rental listings** (American classified advertisements website with sections devoted to jobs, housing, personals, for sale, items wanted, services, community and discussion forums.) **with Google Maps for a visual representation of local apartments for rent.**

Wikipedia vision: combines Google Map and a Wikipedia API

GRID COMPUTING

- Grid Computing enables aggregation of distributed resources .
- It is a type of parallel and distributed system that enables the sharing, exchange, selection, & aggregation of geographically distributed resources depending on their availability, capability, cost, and user QoS requirements.
- Open Grid Service Architecture(OGSA) – helps in standardization by defining a set of core capabilities and behaviors.

GRID COMPUTING (CONTD..)

Issues in grid

- Guaranteeing execution time for critical applications
 - Availability of resources with diverse configuration
- Virtualization technology helped in finding the solution of some of the issues.

➤ Some characteristics of Grid...

- Numerous
- Owned and managed by different, multiple organisations and individuals.
- Have different security requirements and policies
- Heterogeneous
- Connected by heterogeneous, multilevel networks
- Have different resource management policies
- Likely to be geographically separated

UTILITY COMPUTING

- In utility computing environments, users assign a “utility” value to their jobs, where utility is a fixed or time-varying valuation that captures various QoS constraints (deadline, importance, satisfaction).
- The valuation is the amount they are willing to pay a service provider to satisfy their demands.
- Utility computing is a computing business model in which the provider owns, operates and manages the computing infrastructure and resources, and the subscribers accesses it as and when required on a rental or metered basis.

HARDWARE VIRTUALIZATION

- Cloud computing services are usually backed by large-scale data centers composed of thousands of computers.
- Such data centers are built to serve many users and host many disparate applications.
- For this purpose, **hardware virtualization** can be considered as a perfect fit to overcome most **operational issues of data center building and maintenance**.
- Virtualization enhances :sharing and utilization of computer systems



HARDWARE VIRTUALIZATION (CONTD..)

- H/w virtualization allows running multiple OS and a s/w stacks on a single physical platform
- VMM (virtual machine monitor) also called hypervisor mediates access to physical hardware presenting each guest OS a VM
- A number of VMM platforms like VMWare, Xen.... that are the basis of many utility or cloud computing environments.

Benefits

- Sharing and utilization
- Better manageability
- Higher reliability

HARDWARE VIRTUALIZATION (CONTD..)

Three basic capabilities regarding management of workload:

- **Workload isolation**-A virtual machine should not be able to read RAM that is in use by another virtual machine. It also should not be able to access another virtual machine's disk.
- **Workload migration** - encapsulating guest OS system state within VM and allowing it to suspend or migrate to different platforms and resumed immediately...(facilitates h/w maintenance, load balancing and disaster recovery).
- **Workload consolidation** – Consolidation of several individual and heterogenous workloads onto a single platform leads to better system utilization.

AUTONOMIC COMPUTING

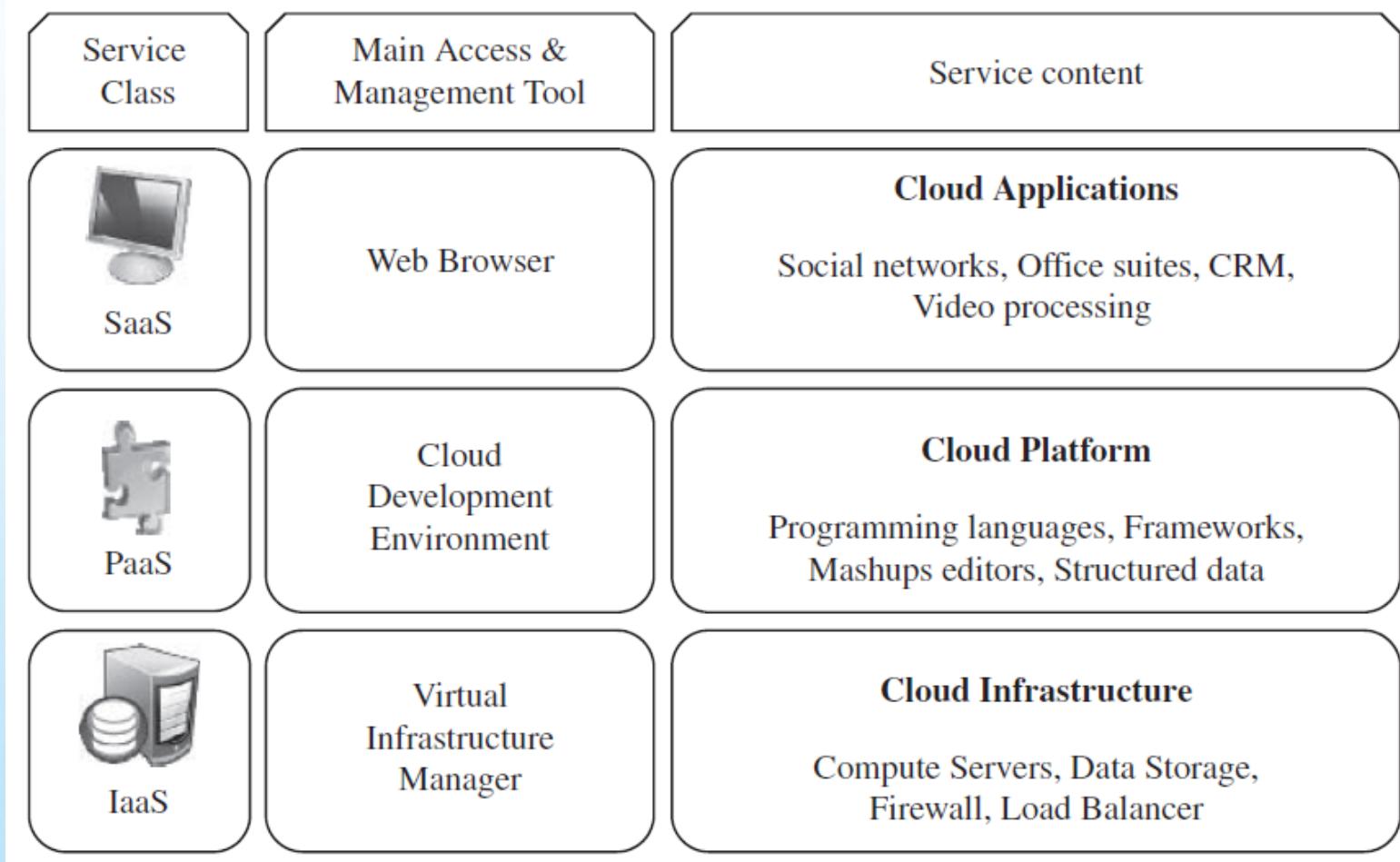
- The increasing complexity of computing systems has motivated research on autonomic computing, which seeks to improve systems by decreasing human involvement in their operation.
- In other words, systems should manage themselves, with high-level guidance from humans
- Autonomic, or self-managing, systems rely on monitoring probes and gauges (**sensors**), on an **adaptation** engine (autonomic manager) for computing optimizations based on monitoring data, and on **effectors** to carry out changes on the system

LAYERS AND TYPES OF CLOUD SERVICES

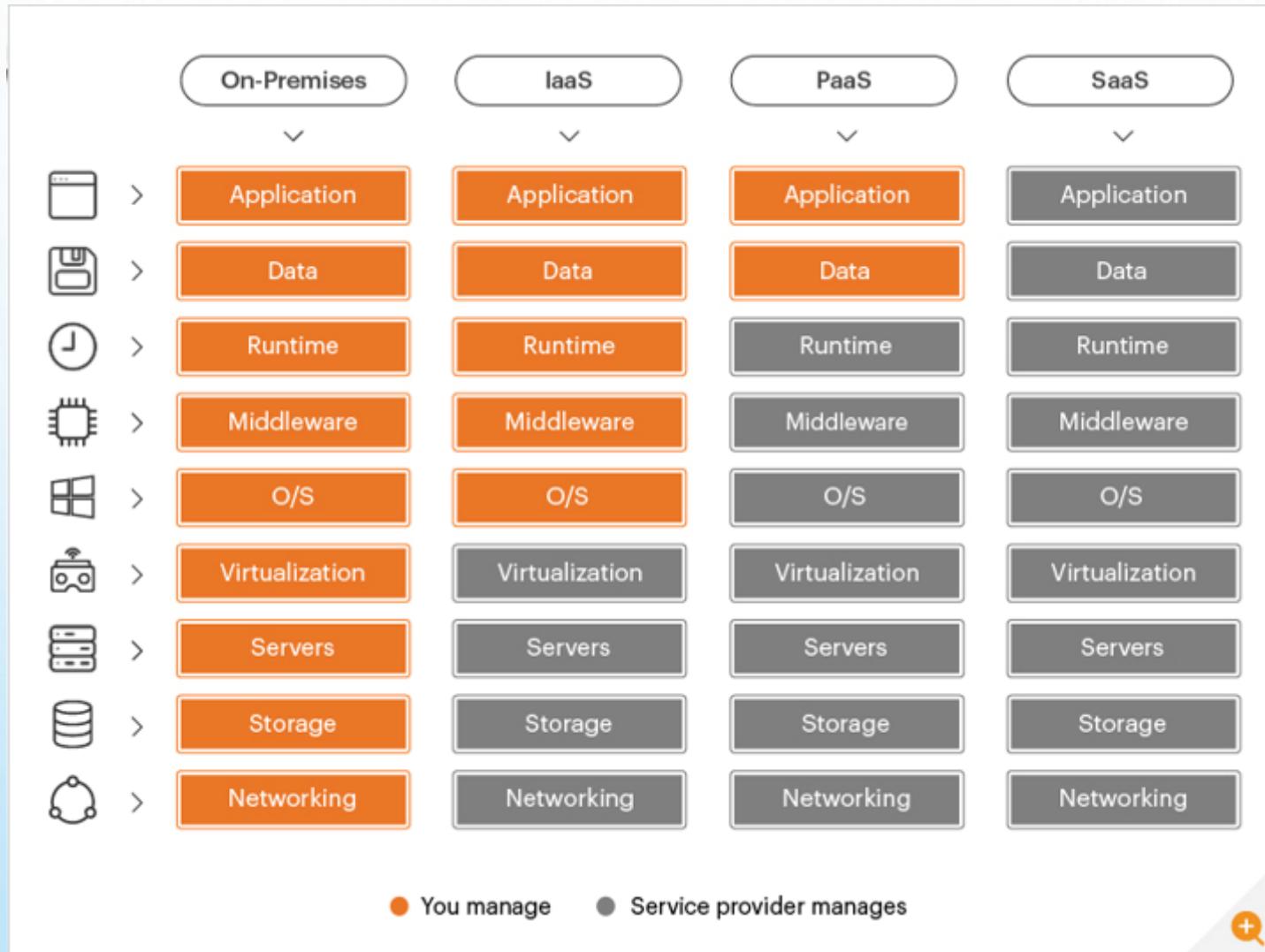
- Cloud computing services are divided into three classes, according to the abstraction level of the capability provided and the service model of providers, namely:
 - **Infrastructure as a Service (IaaS):** hardware is provided by an external provider and managed for the customers
 - **Platform as a Service (PaaS):** in addition to hardware, your operating system layer is managed
 - **Software as a Service (SaaS):** further to the above, an application layer is provided and managed for you

Also called as **The cloud computing stack.**

THE CLOUD COMPUTING STACK



THE CLOUD COMPUTING STACK



<https://www.eginnovations.com/blog/saas-vs-paas-vs-iaas-examples-differences-how-to-choose/>

INFRASTRUCTURE AS A SERVICE (IaaS)

- The capability provided to the consumer is to provision **processing, storage, networks, and other fundamental computing resources** where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.
- The consumer **does not manage or control the underlying cloud infrastructure** but has **control over operating systems and deployed applications**, and possibly limited control of **networking components** .
- The **cloud computing service provider** manages the infrastructure, - customers - install, configure and manage their own software like — operating systems, middleware and applications

IaaS (contd..)

- **Examples :**
- **Amazon EC2** - Amazon Elastic Compute Cloud (EC2) - allowing users to rent virtual computers on which to run their own computer applications.
- **OpenNebula** - OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures
- **Amazon Web Services (AWS)** and **Google Cloud Platform (GCP)** are examples of independent IaaS providers

PLATFORM AS A SERVICE (PaaS)

- The consumer **does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications.**
- Apart from servers, storage and networking—it also includes **middleware, development tools, business intelligence (BI) services, database management systems** and more...
- Can avoid the expense and complexity of buying software licenses, and managing the underlying application infrastructure and middleware or the development tools and other resources

PaaS (contd..)

➤ Examples :

- **Microsoft Windows Azure** - Microsoft Azure is a growing collection of cloud services for building, deploying and managing applications through global network of datacentres.
- **Google App Engine**- Google App Engine is a fully managed platform that completely abstracts away infrastructure so user can focus only on code.
- **Hadoop**- Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment.

SOFTWARE AS A SERVICE

(SaaS)

- The capability provided to the consumer is **to use the provider's applications running on a cloud infrastructure**. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email).
- Often referred to as "**on-demand software**".
- Don't want to worry about the installation, setup and running of the application. Service provider will do that.

SaaS (contd..)

➤ Examples :

- Google Apps (e.g., Gmail, Google Docs, Google sites, ...etc)
- Google Apps, Microsoft Office 365.
- Applications like email (Gmail, Yahoo mail etc), Social Networking sites (Facebook etc)

CLOUD DEPLOYMENT MODELS

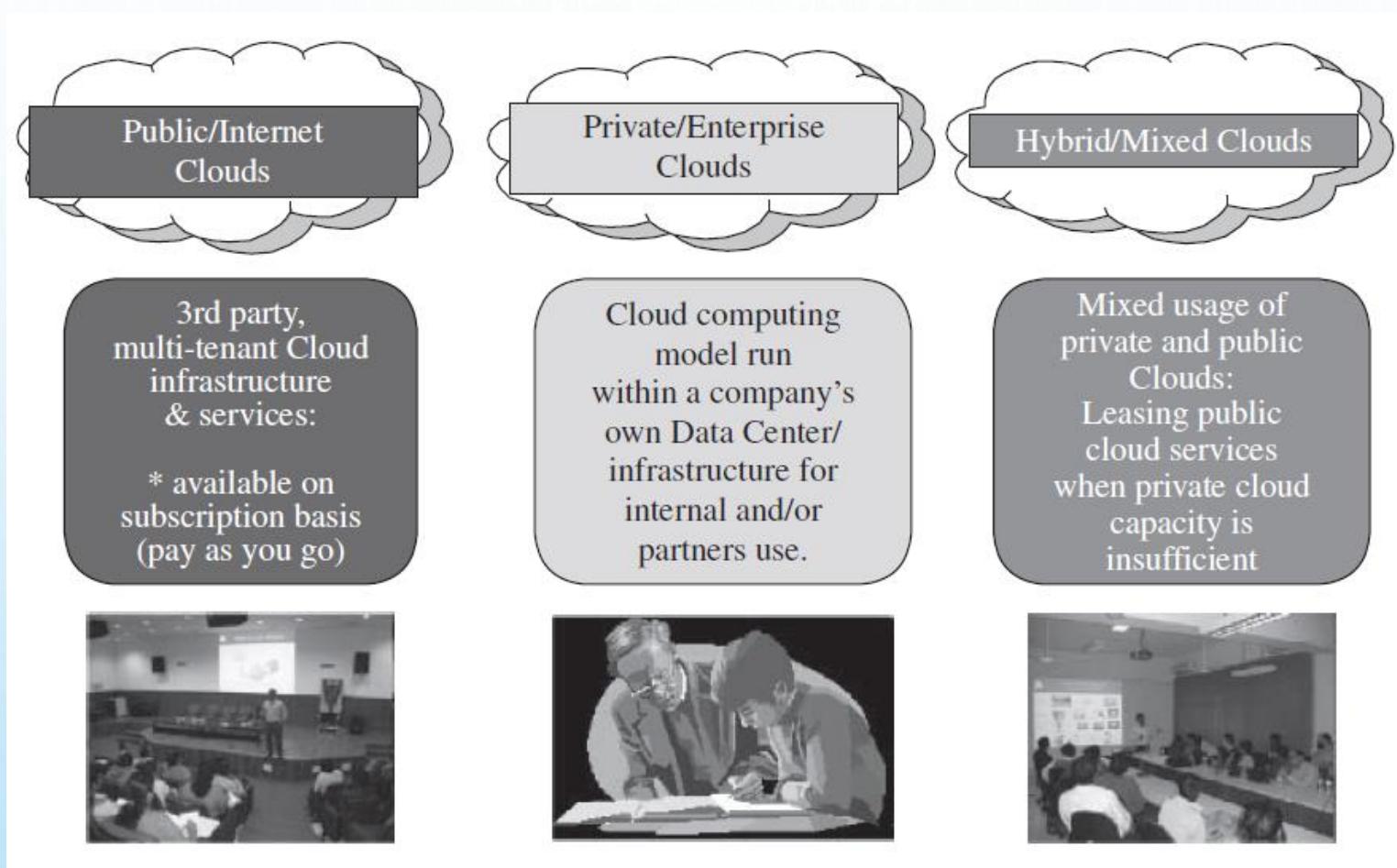


Fig. Types of clouds based on deployment models

PUBLIC CLOUD

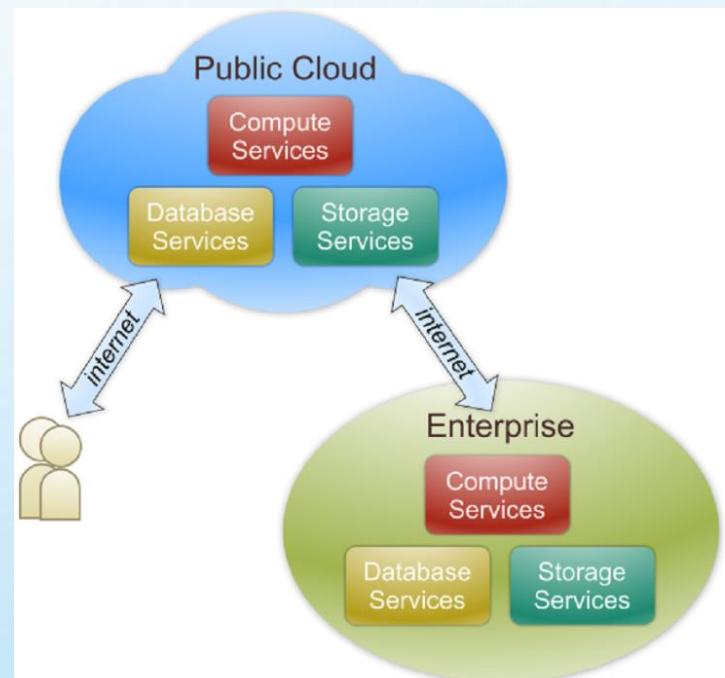
Public cloud definition:

- The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.
- Also known as external cloud or multi-tenant cloud, this model essentially represents a cloud environment that is openly accessible.
- **Basic characteristics :**
 - Homogeneous infrastructure
 - Common policies
 - Shared resources and multi-tenant
 - Leased or rented infrastructure
 - Economies of scale

Examples:

Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform.

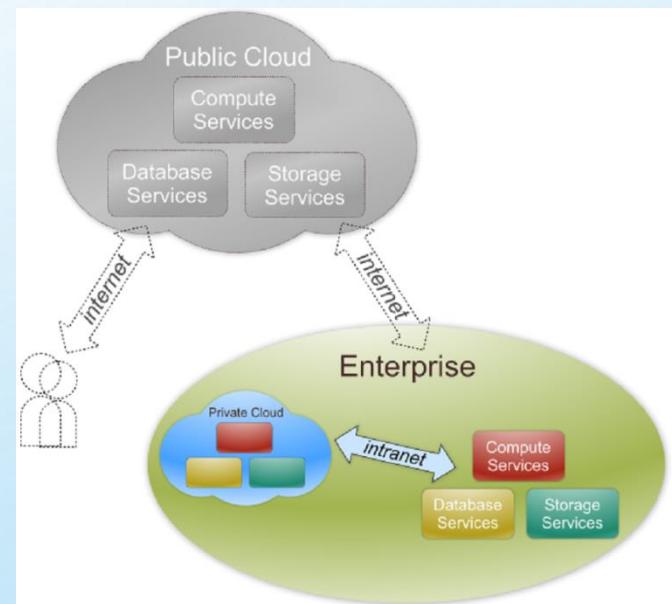
Most of the google services: Gmail, Google Doc, Spreadsheet



PRIVATE CLOUD

Private cloud definition

- The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.
- Also referred to as internal cloud or on-premise cloud, a private cloud intentionally limits access to its resources to service consumers that belong to the same organization that owns the cloud.
- Basic characteristics :
 - Heterogeneous infrastructure
 - Customized and tailored policies
 - Dedicated resources
 - In-house infrastructure
 - End-to-end control

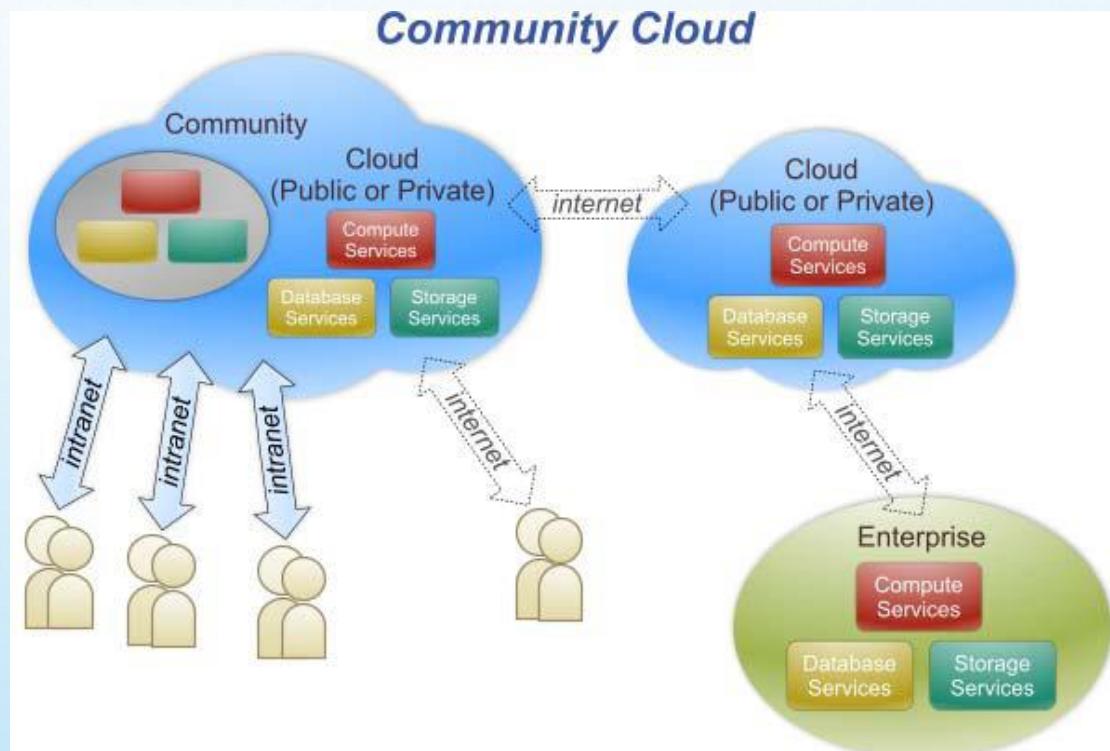


COMMUNITY CLOUD

Community cloud definition

The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations).

Community clouds are often designed for businesses and organizations working on joint projects, applications, or research, which requires a central cloud computing facility for building, managing and executing such projects.

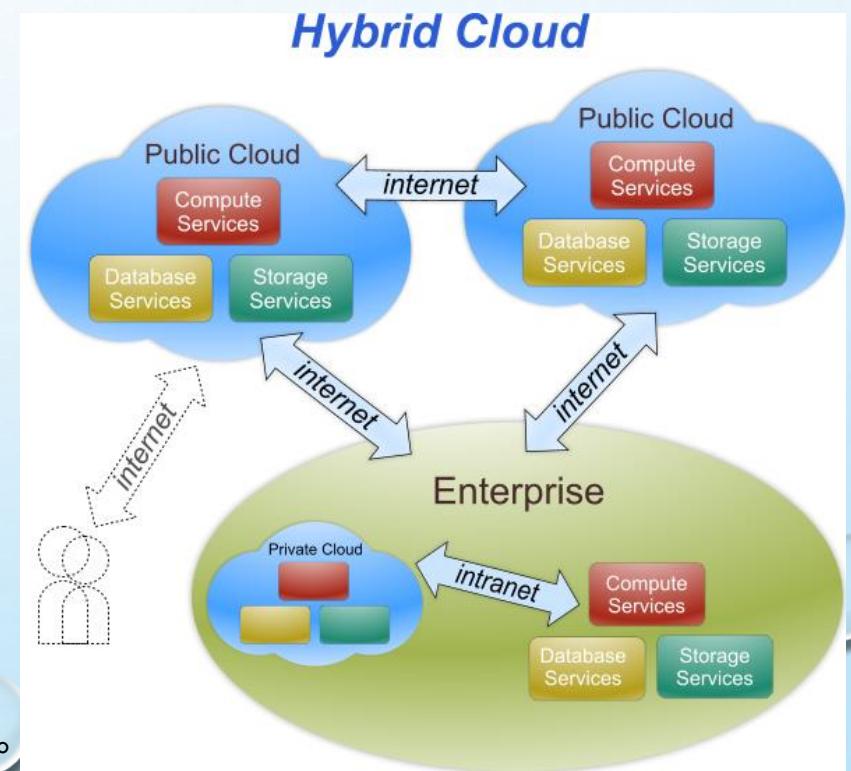


HYBRID CLOUD

Hybrid cloud definition:

The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

- An example of a hybrid cloud solution is an organization that wants to keep confidential information secured on their private cloud, but make more general, customer-facing content on a public cloud.
- Cloud bursting is an application deployment model in which an application runs in a private cloud and burst into a public cloud when the demand for computing capacity spikes.



FEATURES OF CLOUD

- Self Service
- Per usage metering and billing
- Elasticity
- Customization

FEATURES OF CLOUD (CONTD..)

➤ Self Service

Consumers of cloud computing services expect on-demand

Nearly instant access to resources

Clouds must allow self-service access

Customers can request, customize, pay, and use services without intervention of human operators

➤ Per usage metering and billing:

Cloud computing eliminates up-front commitment by users

Allowing them to request and use only the necessary amount

Services must be priced on a short-term basis, e.g., by the hour

Allowing users to release (and not pay for) resources as soon as they are not needed

Clouds must allow efficient trading of service

Such as pricing, accounting, and billing

FEATURES OF CLOUD (CONTD..)

➤ Elasticity

Cloud computing gives the illusion of infinite computing resources available on demand

Users expect clouds to rapidly provide resources in any quantity at any time

The additional resources can be

Provisioned, possibly automatically, when an application load increases

Released when load decreases

Scale up and down

➤ Customization

A multi-tenant cloud often reveals a great disparity between user needs

Resources rented from the cloud must be highly customizable

Infrastructure services allow users

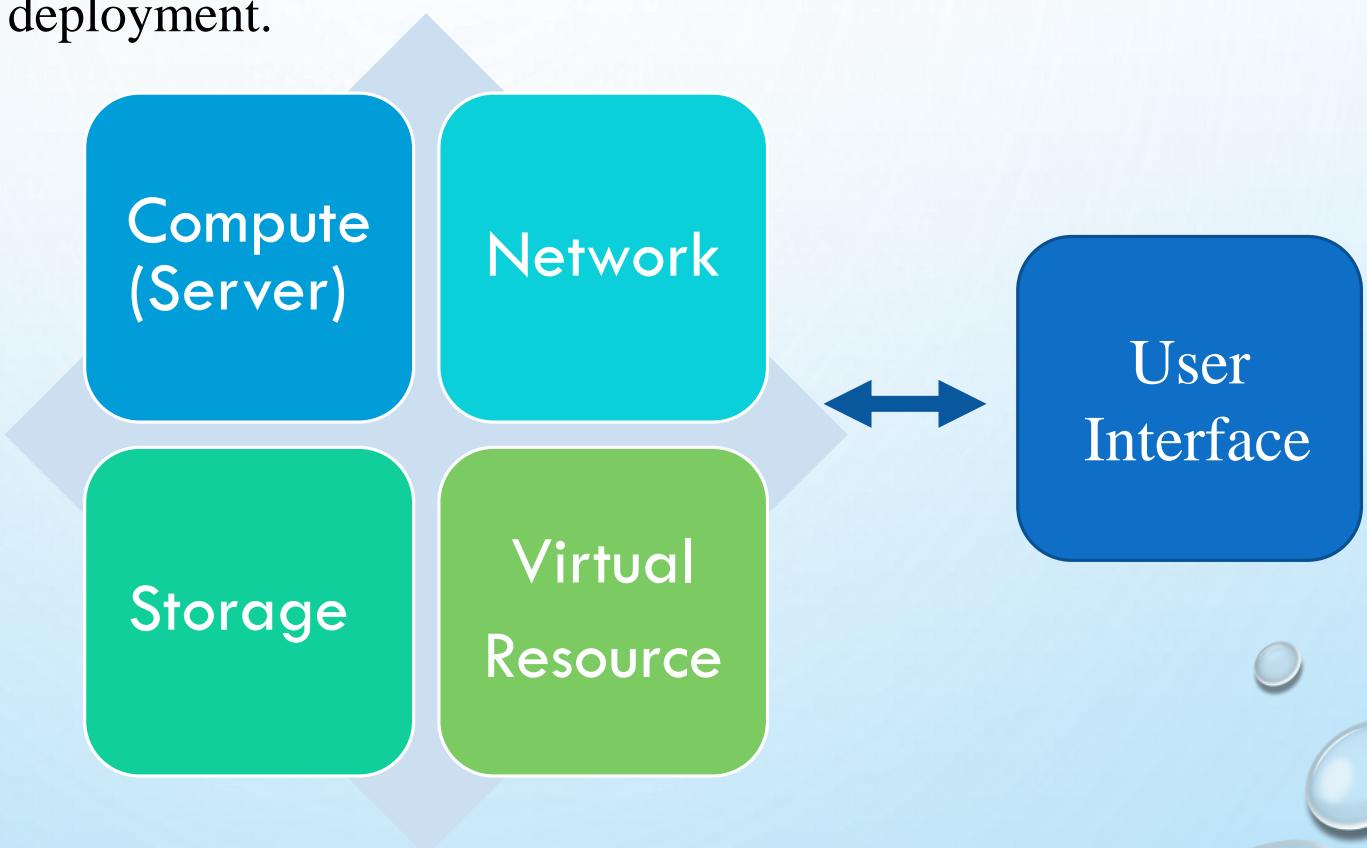
To deploy specialized virtual appliances

To be given privileged (root) access to the virtual servers

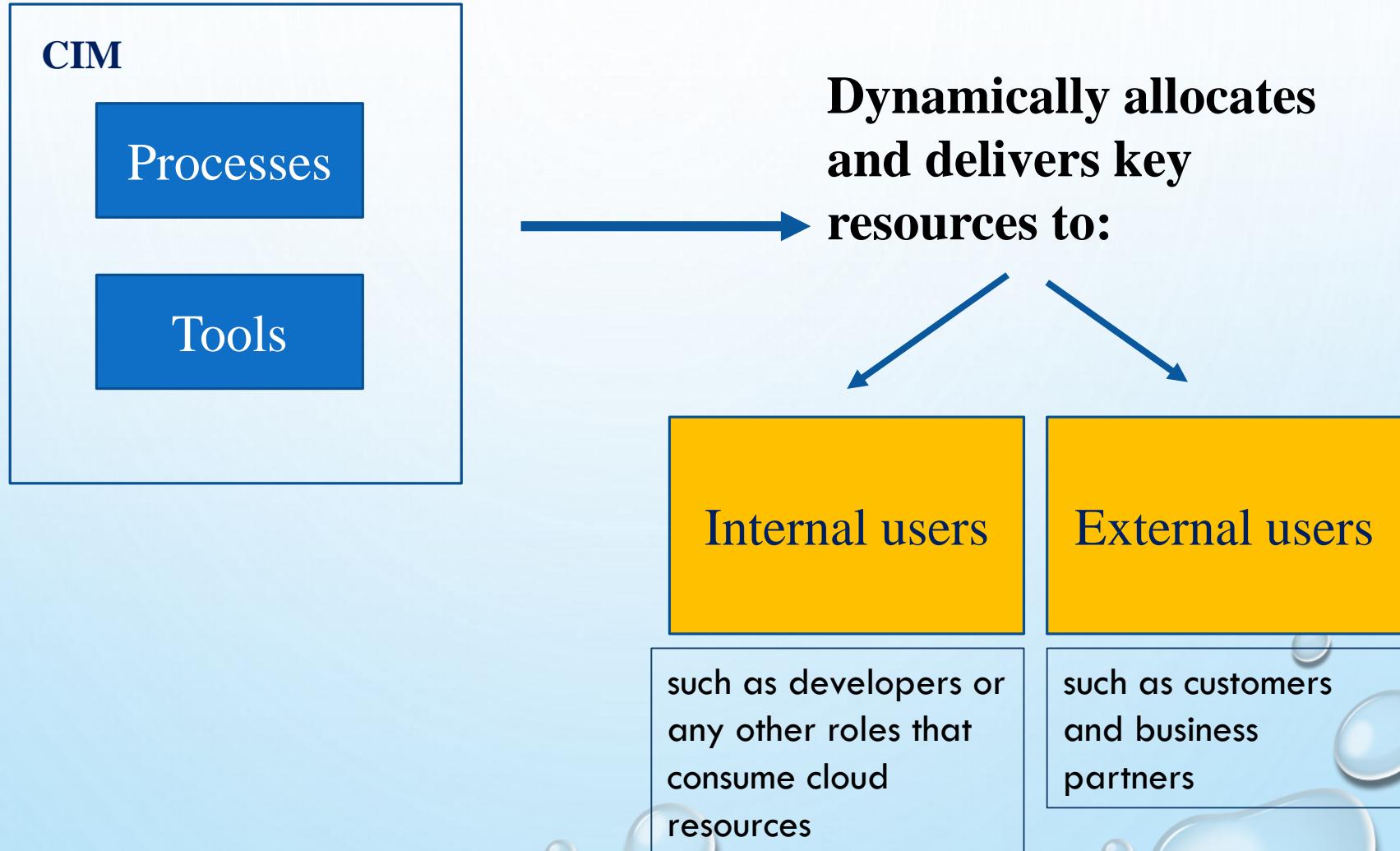
PaaS and SaaS offer less flexibility

CLOUD INFRASTRUCTURE MANAGEMENT

Cloud Infrastructure includes all software and hardware resources needed for cloud deployment.



CLOUD INFRASTRUCTURE MANAGEMENT(CIM) (CONTD..)



CLOUD INFRASTRUCTURE MANAGEMENT (CONTD..)

Why Cloud Infrastructure Management?

- Without appropriate visibility, monitoring and governance, cloud computing costs can increase unnecessarily.
- A typical scenario would be an engineer who leaves a cloud development environment up and running 24/7, even if they only need it for several hours of work.
- In a pay-as-you-go model—which is common in Infrastructure-as-a-Service platforms—that kind of waste can lead to runaway cloud bills.

What does Cloud Infrastructure Management do?

- It's like a central nervous system.
- Tracks cloud usage
- Enables businesses to create, configure, scale and reuse.
- Maximises operational flexibility
- Maintains agility with cost-effectiveness
- Preserves privacy and security of transactions

CLOUD INFRASTRUCTURE MANAGEMENT (CONTD..)

Cloud infrastructure management **tools** as per 2021:

- Apache CloudStack
- BMC Helix Cloud Security
- CloudHealth by VMware
- Microsoft Azure Management Tools
- Morpheus by Morpheus Data
- Terraform Enterprise by HashiCorp
- Turbonomic

<https://www.bmc.com/blogs/hybrid-cloud-management-tools-how-to-choose/>

CLOUD INFRASTRUCTURE MANAGEMENT (CONTD..)

Factors to be considered while choosing the right cloud management tools include:

- Resource management
- Performance monitoring
- Scalability
- Automation and provisioning
- Cross-platform interoperability
- Compliance and governance
- Reporting

CLOUD INFRASTRUCTURE MANAGEMENT

Features available in Virtual Infrastructure Manager(VIM):

- Virtualization Support
- Self Service, On Demand Resource Provisioning
- Multiple backend hypervisors
- Storage Virtualization
- Interface to public clouds
- Virtual Networking
- Dynamic Resource allocation
- Virtual Clusters
- Reservation and negotiation mechanism
- High availability and Data Recovery

FEATURES

Virtualization Support

The multi-tenancy aspect of clouds requires multiple customers with disparate requirements to be served by a single hardware infrastructure

Virtualized resources (CPUs, memory, etc.) Can be sized and resized with certain flexibility

Hardware virtualization

The ideal technology to create a virtual infrastructure that partitions a data center among multiple tenants

FEATURES (CONTD..)

Self-Service, On-Demand Resource Provisioning

- Enables users to directly obtain services from clouds
 - e.g., spawning the creation of a server
 - Tailoring its software, configurations, and security policies
 - Without interacting with a human system administrator
- Eliminates the need for more time-consuming, labor-intensive, human-driven procurement processes
- Users can easily interact with the system

FEATURES (CONTD..)

Multiple Backend Hypervisors

- Some VI managers provide a uniform management layer regardless of the virtualization technology used
- Usually provide pluggable drivers to interact with multiple hypervisors

Storage Virtualization

Abstracting logical storage from physical storage

Consolidating all available storage devices in a data center to create virtual disks independent from device and location

Storage devices are commonly organized in a storage area network (SAN) attached to servers

Via protocols such as Fibre Channel, iSCSI, and NFS

A storage controller provides the layer of abstraction between virtual and physical storage

FEATURES (CONTD..)

Multiple Backend Hypervisors

- Some VI managers provide a uniform management layer regardless of the virtualization technology used
- Usually provide pluggable drivers to interact with multiple hypervisors

Storage Virtualization

Abstracting logical storage from physical storage

Consolidating all available storage devices in a data center to create virtual disks independent from device and location

Storage devices are commonly organized in a storage area network (SAN) attached to servers

Via protocols such as Fibre Channel, iSCSI, and NFS

A storage controller provides the layer of abstraction between virtual and physical storage

FEATURES (CONTD..)

Interface to Public Clouds

Extends the capacity by borrowing resources from public clouds

- Make good use of the available resources

- In case of spikes in demand, extra load can be offloaded to rented resources

A VI manager can be used in a hybrid cloud setup

- Offers a driver to manage the life cycle of virtualized resources obtained from external cloud providers

- Must ideally be transparent

Virtual Clusters

Several VI managers can holistically manage groups of VMs
Useful for provisioning computing virtual clusters on demand
Interconnecting VMs for multi-tier Internet applications

FEATURES (CONTD..)

Virtual Networking

Virtual networks allow creating an isolated network on top of a physical infrastructure

independently from physical topology and locations

A virtual LAN (VLAN) allows isolating traffic that shares a switched network

Allowing VMs to be grouped into the same broadcast domain

Can be configured to block traffic originated from VMs from other networks

The VPN (virtual private network) concept is a secure and private overlay network on top of a public network

Most commonly the public Internet

VI managers support creating and configuring virtual networks to group VMs placed throughout a data center

FEATURES (CONTD..)

Dynamic Resource Allocation

- Increased awareness of energy consumption in data centers has encouraged dynamic consolidating VMs in a fewer number of servers
- Applications have variable and dynamic needs
- Also reallocates available resources among VMs according to application needs
- Energy consumption reduction and better management of SLAs can be achieved
 - By dynamically remapping VMs to physical machines at regular intervals
 - Machines that are not assigned any VM can be turned off or put on a low power state

FEATURES (CONTD..)

Reservation and Negotiation Mechanism

- Requests are termed advance reservations (AR)
Users request computational resources to be available at a specific time
- In contrast to best-effort requests
Users request resources whenever available
The provider can offer a distinct slot that is still satisfactory to the user
- OpenPEX(Open Provisioning and execution system) incorporates a bilateral negotiation protocol
 - OpenPEX, a system that allows users to provision resources ahead of time through advance reservations
 - Allows users and providers to come to an alternative agreement by exchanging offers and counter offers

FEATURES (CONTD..)

High Availability (HA) and Data Recovery

- High availability feature minimizing application downtime and preventing business disruption
- Providing a failover mechanism
 - Detects failure of both physical and virtual servers and restarts VMs on healthy physical servers
 - Protects failures from host
- Frequent backup of a large number of VMs should be done with minimal interference in the systems performance
 - Each one is attached with multiple virtual disks

IAAS FEATURES

- Geographic presence (build data centers)
- User Interface and access to users
- Advance reservation of capacity
- Automatic Scaling and Load Balancing
- Service level Agreement

IaaS FEATURES (CONTD..)

➤ Geographic presence (build data centers)

A provider of worldwide services would typically build several data centers distributed around the world

To improve availability and responsiveness

IaaS FEATURES (CONTD..)

User Interfaces and Access to Servers

A public IaaS provider must provide multiple access means to its cloud

- For various users and their preferences

Different types of user interfaces (UI) provide different levels of abstraction

- The most common being graphical user interfaces (GUI), command-line tools (CLI), and Web service (WS) APIs

GUIs are preferred by end users who need to launch, customize, and monitor a few virtual servers

- Do not necessarily need to repeat the process several times

WS APIs offer programmatic access to a cloud using standard HTTP requests

- Allowing complex services to be built on top of IaaS clouds

IaaS FEATURES (CONTD..)

Advance Reservation of Capacity

Allow users to reserve resources for a specific time frame in the future

Cloud resources will be available at that time

Most clouds only support best-effort requests

Users requests are served whenever resources are available

Amazon Reserved Instances is a form of advance reservation of capacity

Allowing users to pay a fixed amount of money in advance to guarantee resource availability at anytime during an agreed period

Then paying a discounted hourly rate when resources are in use

Only long periods of 1 to 3 years are offered

Users cannot express their reservations in finer granularities, e.g., hours or days

IaaS FEATURES (CONTD..)

Automatic Scaling and Load Balancing

Elasticity is a key characteristic of the cloud

Applications often need to scale up and down to meet varying load conditions

Automatic scaling is a highly desirable feature of IaaS clouds

Allow users to set conditions for when they want their applications to scale up and down

Based on application-specific metrics such as transactions per second, number of simultaneous users, request latency, etc.

Incoming traffic must be automatically distributed among the available servers

Enables applications to promptly respond to traffic increase while also achieving greater fault tolerance

IaaS FEATURES (CONTD..)

Service-Level Agreement

Offered by IaaS providers to express their commitment to delivery of a certain QoS

- To customers it serves as a warranty

- Include availability and performance guarantees

- Metrics must be agreed upon by all parties as well as penalties for violating these expectations

PaaS FEATURES

- Programming Models, Languages and Frameworks – how user can express their application and how efficiently they run in cloud platform.
- Persistence Options – to record their states and recover in case of crashes.

PaaS FEATURES (CONTD..)

Programming Models, Languages, and Frameworks

- Programming models made available by IaaS providers
Define how users can express their applications using higher levels of abstraction and efficiently run them on the cloud platform
- The most common activities that require specialized models are
 - MapReduce model: Processing of large dataset in clusters of computers
 - Development of request-based Web services and applications
 - High-performance distributed execution of various computational tasks
- PaaS providers usually support multiple programming languages
 - Python and Java (e.g., Google appengine)
 - .NET languages (e.g., Microsoft Azure)
 - Ruby (e.g., Heroku)

PaaS FEATURES (CONTD..)

Persistence Options

- A persistence layer is essential to allow applications to record their state
 - Recover it in case of crashes
 - Also store user data
- Web and enterprise application developers have chosen relational databases as the preferred persistence method
 - Offer fast and reliable structured data storage and transaction processing
 - May lack scalability to handle several petabytes (2^{50}) of data stored in commodity computers
- Distributed storage technologies have emerged
 - Seek to be robust and highly scalable
 - At the expense of relational structure and convenient query languages

CHALLENGES AND RISK

- Security , Privacy and Trust
- Data Lock in and standardization
- Availability , Fault Tolerance and Disaster Recovery
- Resource Management and Energy

CHALLENGES AND RISKS (CONTD..)

A significant number of challenges and risks are inherent to cloud computing

- User privacy, data security
- Availability of service, disaster recovery
- Performance, scalability, energy-efficiency
- Programmability

SECURITY, PRIVACY, AND TRUST

Information security is a main issue

Current cloud offerings are essentially public

Exposing the system to more attacks

Need to make cloud computing environments as secure as in-house IT systems

e.g., data encryption, VLANs, and firewalls

The trust toward providers is fundamental

To ensure the desired level of privacy for applications hosted in the cloud

SECURITY, PRIVACY, AND TRUST

Legal and regulatory issues also need attention

- Providers may choose to locate data anywhere
 - The physical location of data centers determines the set of laws applied to the management of data
- Specific cryptography techniques can not be used
 - They are not allowed in some countries
- Country laws can impose that sensitive data are to be stored within national borders
 - e.g., patient health records

DATA LOCK-IN AND STANDARDIZATION

A major concern of cloud computing users

- Having their data locked-in by a certain provider
 - Users may want to move data and applications out from a provider that does not meet their requirements
 - Cloud computing infrastructures and platforms do not employ standard methods of storing user data and applications.
 - User data are not portable

DATA LOCK-IN AND STANDARDIZATION

Open Virtual Format (OVF)

- OVF is a specification that describes an open-standard, secure, efficient, portable and extensible format for packaging and distributing software for virtual machines.
 - Hardware virtualization
 - Aims at facilitating packing and distribution of software to be run on VMs
 - Virtual appliances can be made portable
 - Seamlessly run on hypervisor of different vendors
 - The OVF standard is independent of any particular hypervisor or processor architecture.

AVAILABILITY, FAULT-TOLERANCE, AND DISASTER RECOVERY

- Users will have certain expectations about the service level to be provided
 - Once their applications are moved to the cloud
 - Availability of the service, its overall performance
 - What measures are to be taken when something goes wrong in the system or its components
 - Users seek for a warranty before they can comfortably move their business to the cloud
- SLAs must be ideally set up between customers and cloud computing providers
 - To act as warranty

AVAILABILITY, FAULT-TOLERANCE, AND DISASTER RECOVERY

- Including QoS requirements
- Specifies the details of the service to be provided
 - Including availability and performance guarantees
- Metrics must be agreed upon by all parties
- Penalties for violating the expectations must also be approved

RESOURCE MANAGEMENT AND ENERGY-EFFICIENCY

One important challenge is the efficient management of virtualized resource pools

Physical resources like CPU cores, disk space, and network bandwidth must be sliced and shared among virtual machines

- Running potentially heterogeneous workloads

Dimensions to be considered include

- Number of CPUs, amount of memory, size of virtual disks, and network bandwidth

RESOURCE MANAGEMENT AND ENERGY-EFFICIENCY

Data centers consume large amounts of electricity

100 server racks can consume 1.3 MW of power

Another 1.3 MW are required by the cooling system

Costing USD 2.6 million per year

Data centers significantly impact the environment

In terms of CO₂ emissions from the cooling systems

NEXT SESSION....

VIRTUALIZATION AND HYPERVISORS

VIRTUALIZATION

DR. MANJUNATH V H AND DR. VIDYA RAO

**DEPARTMENT OF DATA SCIENCE AND COMPUTER APPLICATIONS,
MANIPAL INSTITUTE OF TECHNOLOGY,
MANIPAL ACADEMY OF HIGHER EDUCATION**

INTRODUCTION

- Logical abstraction of computing resources
- i.e. separation of resource and/or service request from underlying physical delivery of that service request
- A means of separating hardware from a single operating system
- Allows multiple operating system instances to run concurrently on a single computer.

INTRODUCTION (CONTD..)

- Virtualization is a technique of how to **separate a service from the underlying physical delivery of that service**. It is the process of creating a virtual version of something like computer hardware.
- It was initially developed during the mainframe era.
- It involves using **specialized software** to create a virtual or software-created version of a **computing resource** rather than the actual version of the same resource.
- With the help of **virtualization**, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

INTRODUCTION (CONTD..)

- In other words, **one of the main cost effective, hardware reducing, and energy saving techniques used by cloud providers is virtualization.**
- Virtualization allows to share a single physical instance of a resource or an application among multiple customers and organizations at one time.
- It does this by **assigning a logical name to a physical storage and providing a pointer to that physical resource** on demand.
- The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering infrastructure-as-a-service (IaaS) solutions for cloud computing.
- Moreover, virtualization technologies provide a **virtual environment for not only executing applications but also for storage, memory, and networking.**

BENEFITS

- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of it infrastructure.
- Remote access and rapid scalability.
- High availability and disaster recovery.
- Pay per use of the it infrastructure on demand.
- Enables running multiple operating system.

CONCEPT AND TYPES

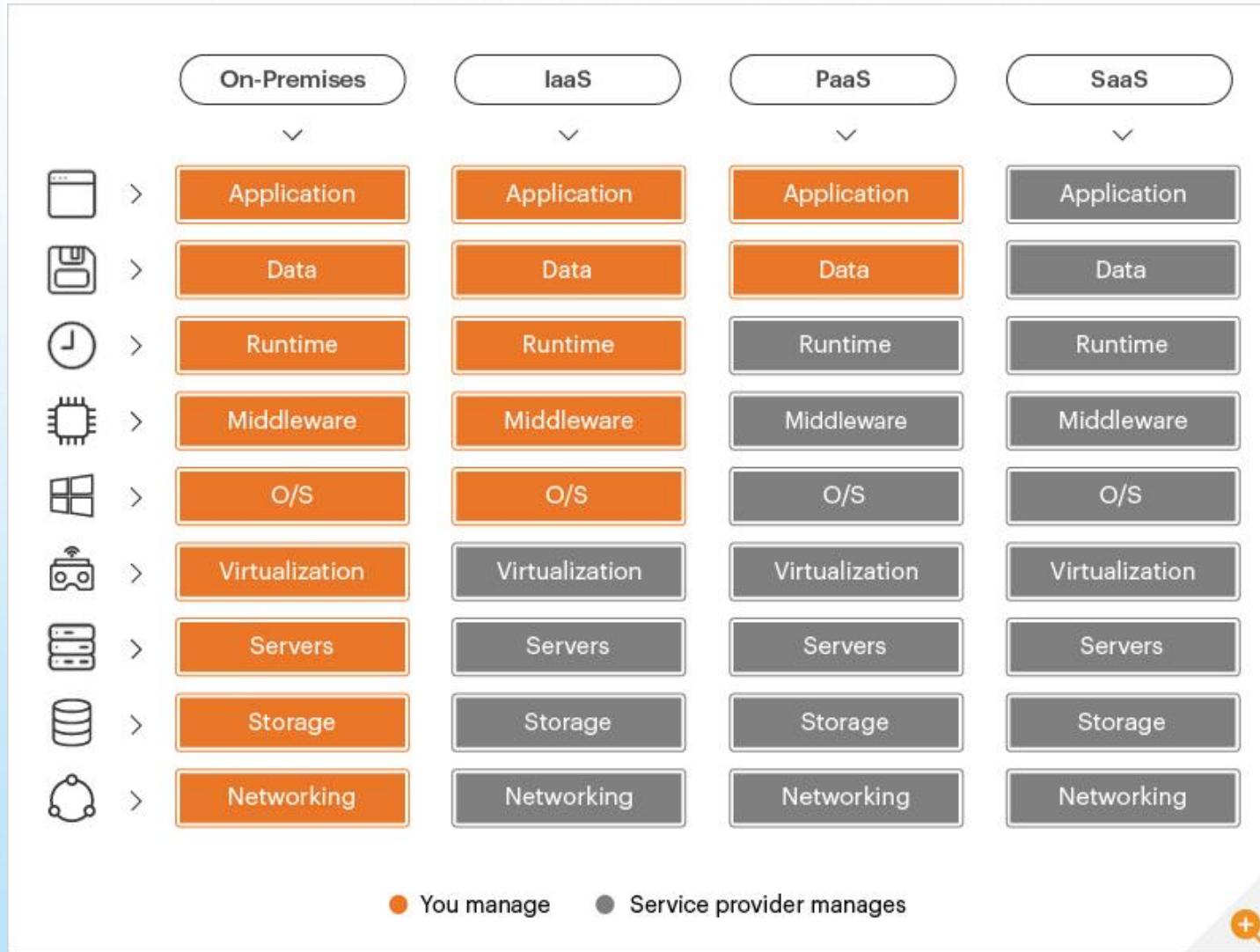
- The machine on which the virtual machine is going to be created is known as **host machine** and that virtual machine is referred as a **guest machine**

Types of virtualization:

- Data virtualization
- Hardware virtualization.
- Application virtualization.
- Network virtualization.
- Desktop virtualization.
- Storage virtualization.
- Server virtualization.
- Operating system virtualization.



CONCEPT AND TYPES



CHARACTERISTICS

- **Increased Security**
- **Managed Execution**
- **Sharing**
- **Aggregation**
- **Emulation**
- **Isolation**
- **Portability**

CHARACTERISTICS (CONTD..)

Increased Security

- The ability to control the execution of guest programs in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment.
- All the operations of the guest programs are generally performed against the virtual machine, which then translates and applies them to the host programs.
- A virtual machine manager can control and filter the activity of the guest programs, thus preventing some harmful operations from being performed.
- Resources exposed by the host can then be hidden or simply protected from the guest.
- Increased security is a requirement when dealing with untrusted code.

CHARACTERISTICS (CONTD..)

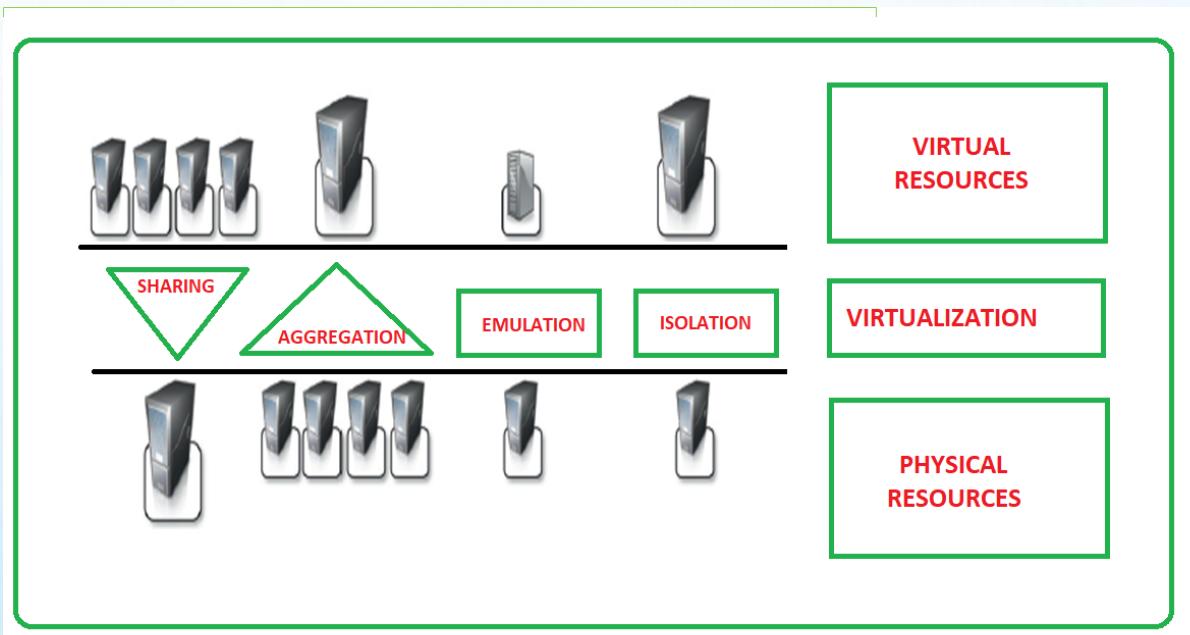
Example

- **Example-1:** untrusted code can be analyzed in cuckoo sandbox environment. The term **sandbox** identifies an isolated execution environment where instructions can be filtered and blocked before being translated and executed in the real execution environment.
- **Example-2:** the expression sandboxed version of the **java virtual machine (JVM)** refers to a particular configuration of the JVM where, by means of security policy, instructions that are considered potentially harmful can be blocked.

CHARACTERISTICS (CONTD..)

Managed Execution

- In particular, sharing, aggregation, emulation, and isolation are the most relevant features.



CHARACTERISTICS (CONTD..)

Sharing

- Virtualization allows the **creation of a separate computing environments** within the same host.
- This basic feature is used to **reduce the number of active servers** and limit power consumption.

Aggregation

- Not only possible to share physical resources among several guests, but virtualization also allows **aggregation**, which is the opposite process.
- A group of separate hosts can be tied together and represented to guests as a single virtual host.
- This functionality is implemented with **cluster management software**, which harnesses the physical resources of a homogeneous group of machines and represents them as a single resource.

CHARACTERISTICS (CONTD..)

Emulation

- Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program.
- Also a completely different environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host.

Isolation

- Virtualization allows providing guests—whether they are operating systems, applications, or other entities—with a completely separate environment, in which they are executed.
- The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources.
- The virtual machine can filter the activity of the guest and prevent harmful operations against the host.

CHARACTERISTICS (CONTD..)

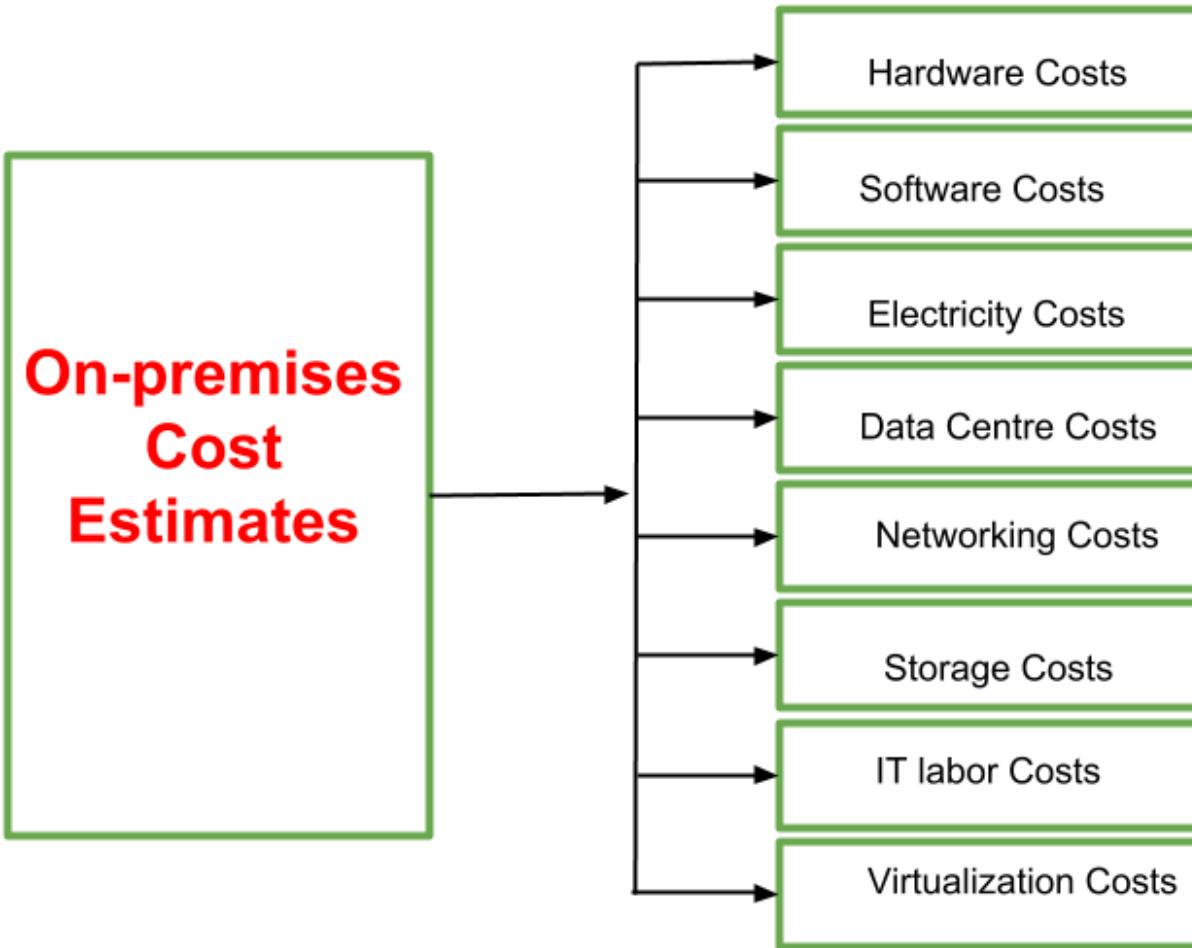
Portability

- The concept of portability applies in different ways according to the specific type of virtualization considered.
- In the case of a hardware virtualization solution, **the guest is packaged into a virtual image that**, in most cases, can be safely moved and executed on top of different virtual machines.
- In the case of programming-level virtualization, as implemented by **the JVM or the .Net runtime, the binary code representing application components** (jars or assemblies) can run without any recompilation on any implementation of the corresponding virtual machine

HYPERVISOR

- A program that allows multiple operating systems to share a single hardware host.
- Also called virtual machine manager/monitor (vmm), or virtualization manager.
- Hypervisor actually controls processor and resources, allocating what is needed to each operating system in turn and making sure that the guest operating systems (vms) don't disrupt each other.
- **Virtual machine:** a self-contained computing environment that behaves as if it is a separate computer.

COST ESTIMATION



COST ESTIMATION (CONTD..)

HARDWARE COSTS

- Based on the user's description, the calculator uses estimated market rates to project the costs of requisite physical hardware.

SOFTWARE COSTS

- If the user specifies an operating system for the environment, then the calculator provides an estimate of the OS licensing costs based on the number of cores required by the user's environment.

ELECTRICITY COSTS

- Estimates the approximate electricity consumption costs. This is done by allocating a power rating to the user's hardware configuration and then multiplying it with an estimate of the power consumed and an industry-standard rate for power consumption.

COST ESTIMATION (CONTD..)

DATA CENTER COSTS

- Based on the user's description, the calculator estimates the requisite amount of normalized rack space.

NETWORKING COSTS

- Networking hardware and software costs are estimated to be a fixed percentage of the on-premises hardware and software costs. The calculator also adds a service provider cost based on the outbound bandwidth requirements specified by the user.

STORAGE COSTS

- The calculator computes storage cost by multiplying an estimated market rate the amount of disk space specified by the user.

COST ESTIMATION (CONTD..)

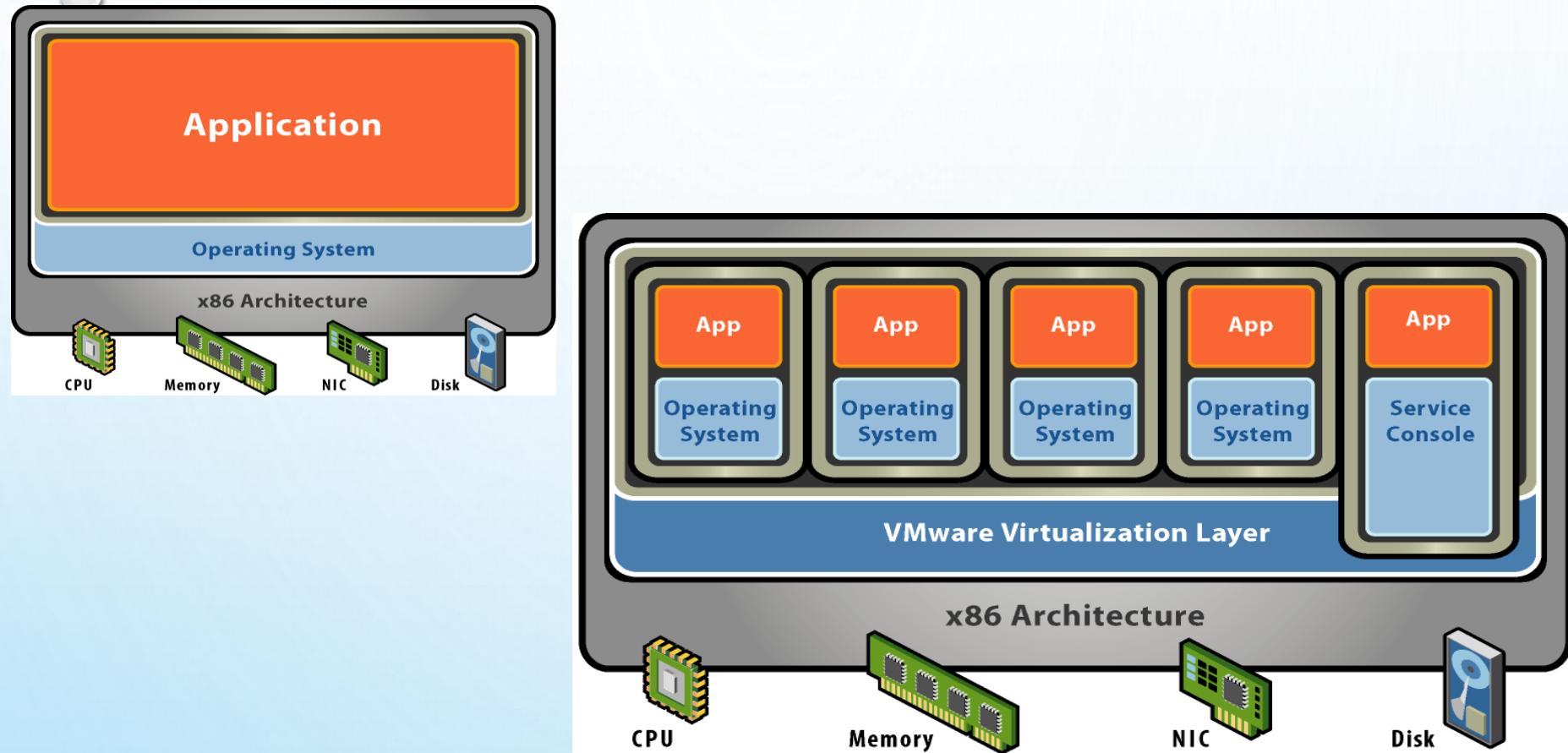
IT LABOR COSTS

- Based on the user's description, the calculator estimates the number of IT administrator man-hours that will be required to maintain the on-premises environment.

VIRTUALIZATION COSTS

- If the user's description includes virtual machines, the calculator uses an industry standard per virtual machine management rate to estimate the total virtualization costs.

ARCHITECTURE DIFFERENCES



DIFFERENCES

BEFORE VIRTUALIZATION:

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure.

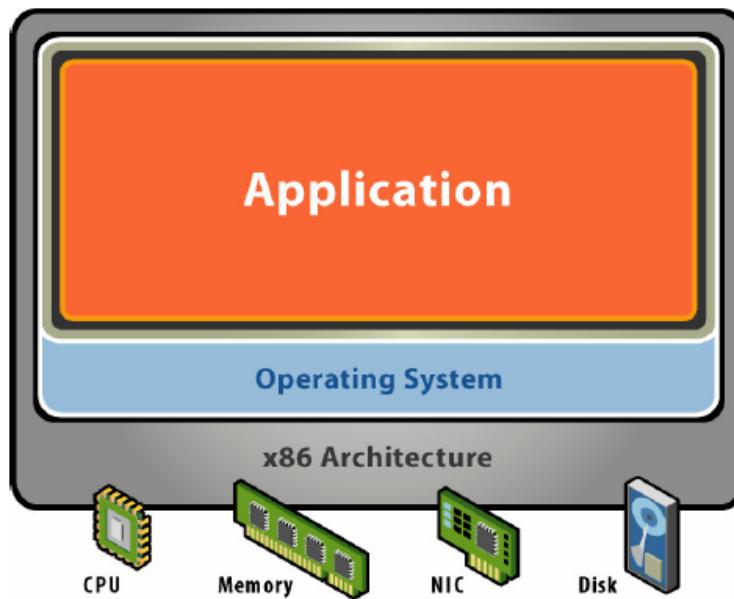
AFTER VIRTUALIZATION:

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

DIFFERENCES

- **TRADITIONAL COMPUTING ARCHITECTURE HAS :**
 - Hardware
 - Operating system
 - Application program(s)
- **VIRTUALIZATION ARCHITECTURE HAS :**
 - Hardware (centralized / decentralized)
 - Virtualization layer (VMM)
 - Host operating system
 - Application program(s) //VM
 - Hosted (guest) operating system(s)
 - Hosted (guest) application program(s)

Starting Point: A Physical Machine



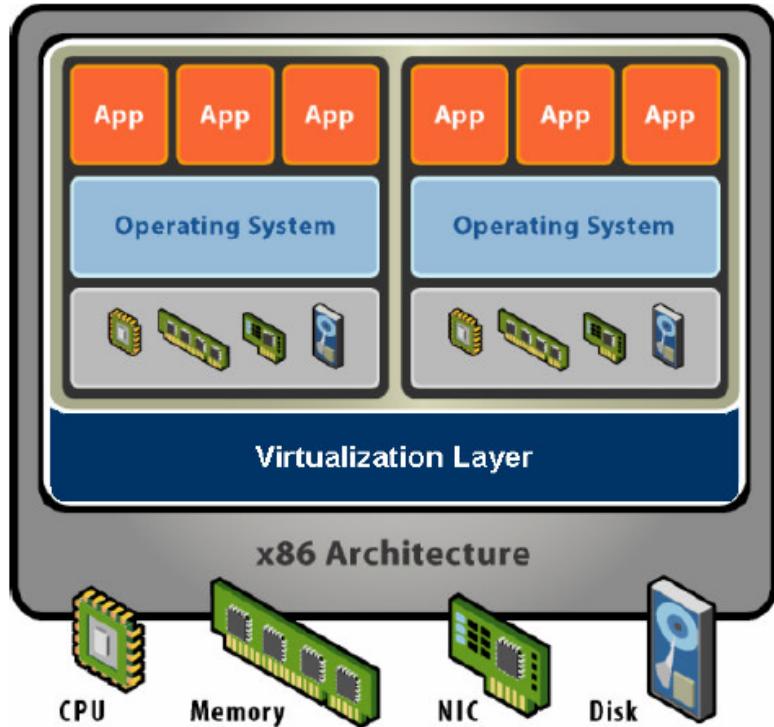
Physical Hardware

- Processors, memory, chipset, I/O bus and devices, etc.
- Physical resources often underutilized

Software

- Tightly coupled to hardware
- Single active OS image
- OS controls hardware

What is a Virtual Machine?



Hardware-Level Abstraction

- Virtual hardware: processors, memory, chipset, I/O devices, etc.
- Encapsulates all OS and application state

Virtualization Software

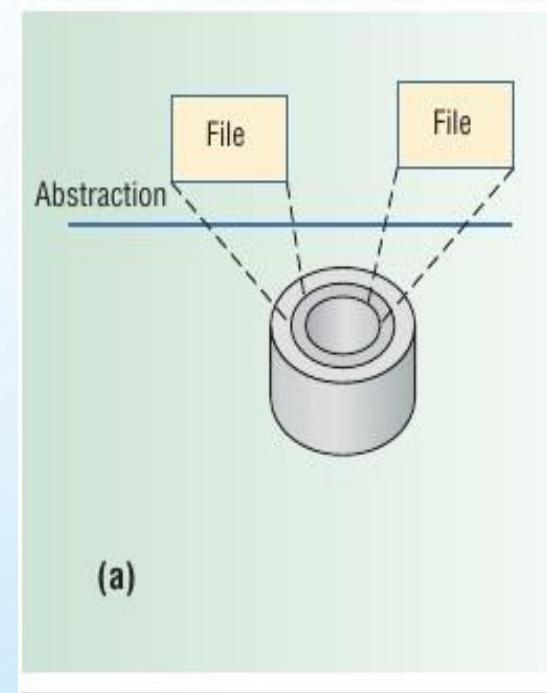
- Extra level of indirection decouples hardware and OS
- Multiplexes physical hardware across multiple “guest” VMs
- Strong isolation between VMs
- Manages physical resources, improves utilization

ARCHITECTURE OF VIRTUAL MACHINES

- VM can support individual processes or a complete system
- Virtualization can be from OS to programming languages to processor architecture.
- VMs enhance
 - Software interoperability (to work together)
 - System impregnability (having strength)
 - Platform versatility

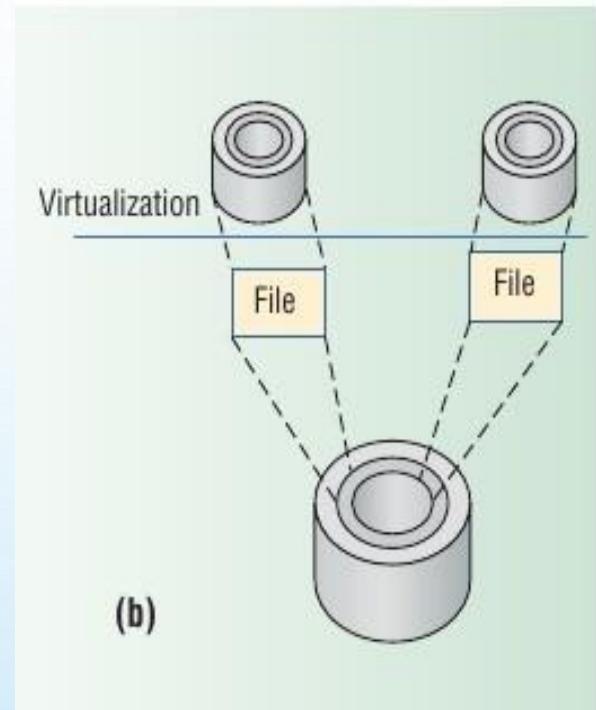
ABSTRACTION AND VIRTUALIZATION

- Computer system is complex, and yet it continue to evolve.
- Computer is designed as hierarchies of well- defined interfaces that separate level of abstraction
- Simplifying abstractions hide lower-level implementation details
- Abstraction
 - Ex. Disk storage
 - Hides hard-disk addressing details (sectors and tracks)
 - It appears to application software as a variable sized files.
 - User can create, write and read files without knowing the underneath details.



VIRTUALIZATION

- Virtualization of system or components like – processor, memory or an I/O device – at a given abstraction level.
- It transforms a entire system or components of the system

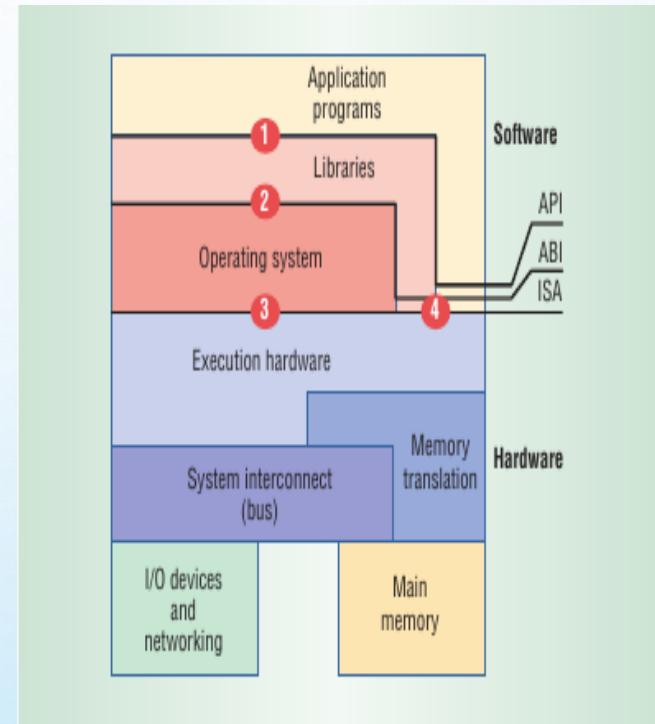


ARCHITECTED INTERFACES

- Architecture, as applied to computer systems, refer to a formal specification to an interface in the system, including the logical behavior of the resources managed via the interface.
- Implementation describes the actual embodiment of an architecture.
- Abstraction levels correspond to implementation layers, having its own interface or architecture.

COMPUTER SYSTEM ARCHITECTURE

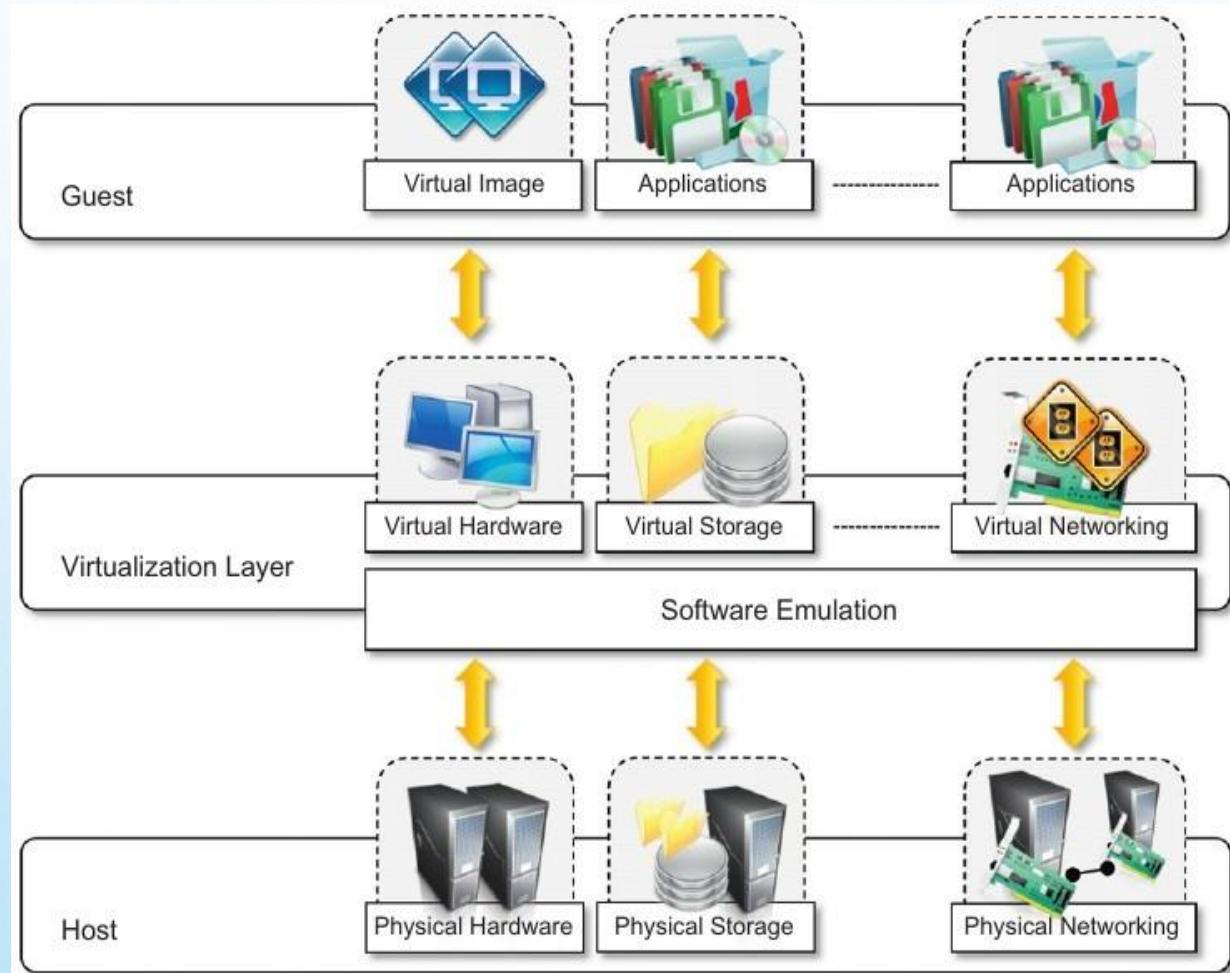
- Interfaces at or near the H/w S/w boundary
- ISA – instruction set architecture.
- API – application program interface
- ABI – application binary interface



VIRTUALIZED ENVIRONMENTS

- Three major components of virtualized environments
- **Guest** – system component that interacts with virtualization layer.
- **Host** – original environment where guest runs.
- **Virtualization layer** – recreate the same or different environment where guest will run.

VIRTUALIZATION REFERENCE MODEL



HYPER-CONVERGED INFRASTRUCTURE (HCI)

L8-L12

DR. MANJUNATH V H AND DR. VIDYA RAO

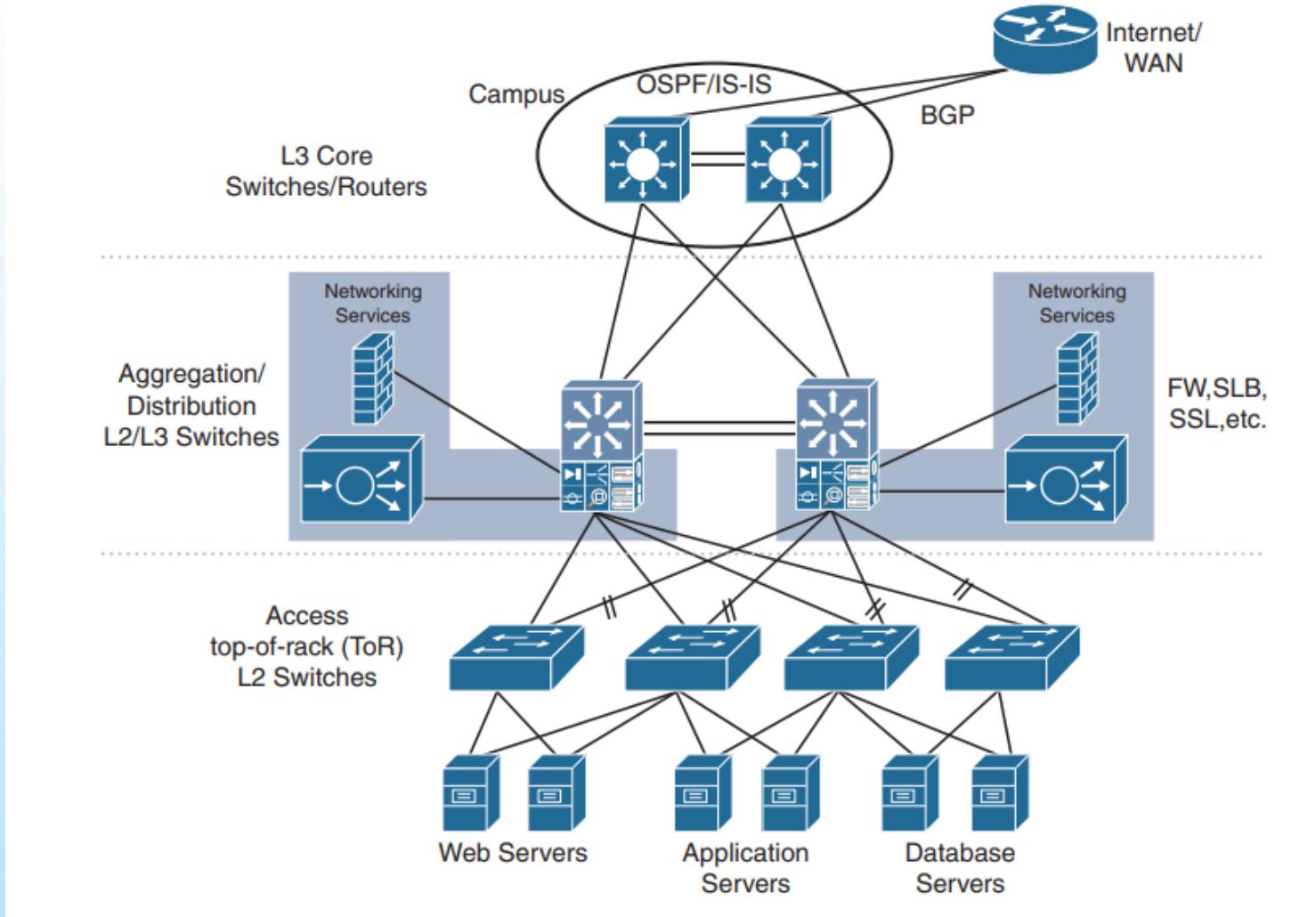
INTRODUCTION TO HYPER CONVERGED INFRASTRUCTURE(HCI)

- Increasing their demands on enterprise IT led to provide **more infrastructure with less cost and time.**
- These demands have resulted in the need for IT organizations to attempt to mimic **NASA's** much-heralded **“Faster, Better, Cheaper”** operational campaign.
- Unfortunately, most of today's data center architectures actively work against these goals, because with increasing complexity comes increased costs — and things have definitely become more complex.
- With virtualization, enterprise IT has moved from physical servers, where storage services could be configured on a per-server basis, **to shared storage systems.**

HCI (CONTD..)

- **Hyperconverged infrastructure (HCI) is a software-defined, unified system that combines all the elements of a traditional data center: storage, compute, networking and management.**
- **Why** are these two resources, storage and compute, at the core of hyperconverged infrastructure?
 - **Storage** has become an incredible challenge for many companies. It's one of—if not the—most expensive resources in the data center and often requires a highly skilled person or team to keep it running
 - **Combining storage with compute** is in many ways a return to the past, but this time many new technologies have been wrapped around it.
- Before virtualization and before SANs, many companies ran physical servers with directly attached storage systems

HCI (CONTD..)



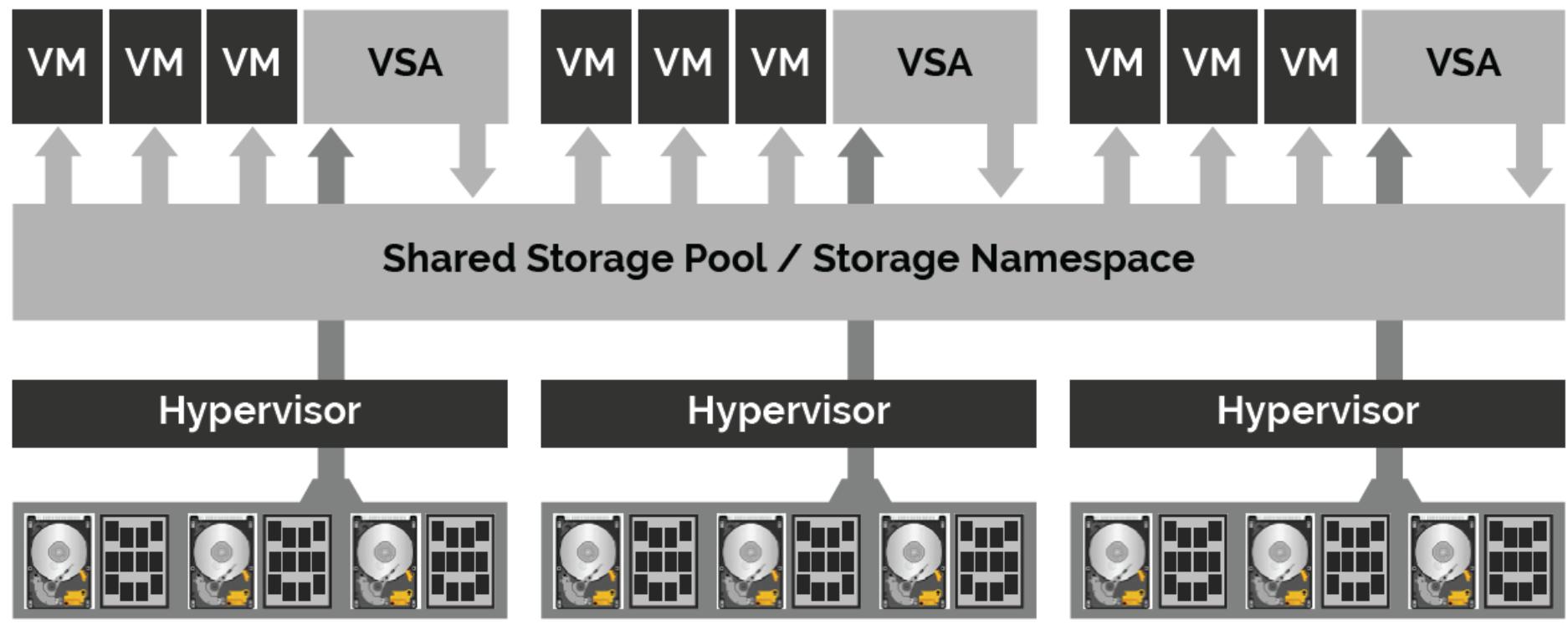
HCI (CONTD..)

- There are other challenges that administrators of legacy data centers need to consider as well:
 - **Hardware sprawl.**
 - **Policy sprawl.**
 - **Scaling challenges.**
 - **Desire for less technical overhead.**

UNDERSTANDING HCI

- Hyperconverged infrastructure **distributes the storage resource** among the various **nodes that comprise a cluster**.
- Often built using commodity server chassis and hardware, hyperconverged infrastructure nodes and appliances are bound together via Ethernet and a powerful software
- The software layer often includes a **virtual storage appliance (VSA)** that runs on each cluster node.
- Each VSA then communicates with all of the other VSAs in the cluster over an Ethernet link, thus forming a distributed file system across which virtual machines are run

UNDERSTANDING HCI (CONTD..)



UNDERSTANDING HCI (CONTD..)

- The fact that these systems leverage commodity hardware is critical.
- The power behind hyperconverged infrastructure lies in its ability to **coral resources** – RAM, compute, and data storage – from hardware that doesn't all have to be custom-engineered.
- This is the basis for hyperconverged infrastructure's ability to **scale granularly and the beginnings of cost reduction processes.**

RESOURCES TO CONSOLIDATE

(CONTD..)

- The basic combination of storage and servers is a good start, but once one looks beyond the confines of this baseline definition, hyper-converged infrastructure begins to reveal its true power.
- The more hardware devices and software systems that can be collapsed(fallen down) into a hyper-converged solution, the easier it becomes to manage the solution and the less expensive it becomes to operate.

RESOURCES TO CONSOLIDATE

(CONTD..)

- Here are some data center elements that can be integrated in a hyperconverged infrastructure:
 - Deduplication Appliances
 - SSD Caches/All-Flash Array
 - Backup Software
 - Data Replication

RESOURCES TO CONSOLIDATE

(CONTD..)

Deduplication Appliances

- Deduplication technologies are common in today's data center. i.e. **eliminating duplicate copies of repeating data**
- **Dedicated appliances** are now available that handle complex and CPU-intensive deduplication tasks, ultimately reducing the amount of data that has to be housed on primary storage thus improving storage utilization
- Successful implementation of the technique can **improve storage utilization**.
- Deduplication works by creating a **data fingerprint** for each object that is written to the storage array.
- As new data is written to the array, if there are matching fingerprints, additional data copies beyond the first are **saved as tiny pointers**.

RESOURCES TO CONSOLIDATE

(CONTD..)

SSD Caches/All-Flash Array

- To address storage performance issues, companies sometimes deploy either solid state disk (SSD)-based caching systems or full SSD/flash-based storage arrays.
- However, both solutions have the potential to increase complexity as well as cost.
- When server-side PCI-e SSD cards are deployed, there also has to be a third-party software layer that allows them to act as a cache, if that is the desired.

Note: PCIe -- Peripheral Component Interconnect-express

RESOURCES TO CONSOLIDATE

(CONTD..)

Backup Software

- Data protection in the form of backup and recovery remains a critical task for IT and is one that's often not meeting organizational needs.
- **Recovery time objectives (RTO)** and **recovery point objectives (RPO)** are both shrinking metrics that IT needs to improve upon.
 - **RTO** is concerned with getting appliances and systems back up and running soon as possible within a predefined time frame.
 - **RPO** is about how much you afford to lose before it impacts business application.
- On the other hand RTO is the time frame within which applications and systems must be restored after an outage.

RESOURCES TO CONSOLIDATE

(CONTD..)

Data Replication

- **Data protection is about far more than just backup and recovery.** What happens if the primary data center is lost? This is where replicated data comes into play.
- **By making copies of data and replicating that data to remote sites,** companies can rest assured that critical data won't be lost.
- To enable these data replication services, companies implement a variety of other **data center services.**
- For example, to minimize replication impact on bandwidth, companies deploy **WAN acceleration** devices intended to reduce the volume of data traversing the Internet to a secondary site.
- **WAN acceleration** remediates network congestion and latency by implementing bandwidth preservation techniques such as **compression, caching, data deduplication, and image optimization.**

ARCHITECTING THE HYPER-CONVERGED DATA CENTER

- One of the primary goals of hyper-convergence is to **simplify infrastructure decisions** in the data center.
- Right hyper-converged infrastructure solution converges of computing and storage resources can be wrapped up into a single hyper-converged appliance.
- Also, include entire backup-and-recovery process, your deduplication and WAN acceleration appliances, and much more

ARCHITECTING THE HYPER-CONVERGED DATA CENTER

- Four decision has to be made while choosing the HCDC:
 - Server support
 - Storage support
 - Data protection services
 - The management layer support

ARCHITECTING THE HYPER-CONVERGED DATA CENTER:

DECISION 1: SERVER SUPPORT

- Not all hyper-converged solutions ship in the same kind of packaging.
- For example, there are **appliance-based hyper-converged** solutions from companies such as SimpliVity, Nutanix, Scale Computing, and Maxta.
- And then there are **software-only solutions** that you install yourself, which include Stratoscale and Maxta. Maxta is on both lists because they support both pre-configured appliances
- **With an appliance-based solution, you're buying the full package, and you just need to plug everything in and turn it on. These are really easy to get going since most things are already done for you.**
- However, with an appliance-based solution, you generally have to live with whatever constraints the vendor has placed on you.

ARCHITECTING THE HYPER-CONVERGED DATA CENTER:

DECISION 1: SERVER SUPPORT

- You need to remain within their hardware specifications, and you don't always get to choose your server platform, although many appliance-based solutions do support servers from multiple vendors
- If you'd rather go your own way with regard to hardware, you can choose a software-based hyper-converged solution.
- Software-based solutions are really good for larger organizations with sufficient staff to install and support the hyper-converged infrastructure.
- Hardware-based solutions are often desired by companies that are looking for a more seamless deployment experience or that do not have sufficient staff to handle these tasks.

ARCHITECTING THE HYPER-CONVERGED DATA CENTER:

DECISION 2: **STORAGE DECISION**

- One of the **main reasons people are dissatisfied** with their data centers is because **their storage solution has failed to keep pace** with the needs of the business.
- It's either too **slow** to support mission critical applications or it doesn't have **data efficiency features**
- **Many storage devices are not well-designed when it comes to supporting virtualized workloads, either.**
- Traditional SANs are challenged when attempting to support the wide array of I/O types that are inherent in heavily virtualized environments.
- At the same time, **storage has become more complex, often requiring specialized skill sets to keep things running.**

ARCHITECTING THE HYPER-CONVERGED DATA CENTER:

DECISION 2: **STORAGE DECISION**

- As companies grow and become more dependent on IT, they also start to have more reliance on data mobility services.
- For some systems, it's not easy to do the basics, which can include managing LUNs, RAID groups, aggregates and more
- **Legacy storage systems don't always do a great job** enabling data mobility and often **don't even support services like remote replication and cloning** or, if they do, it's a paid upgrade service.

Note:

LUN: logical unit number

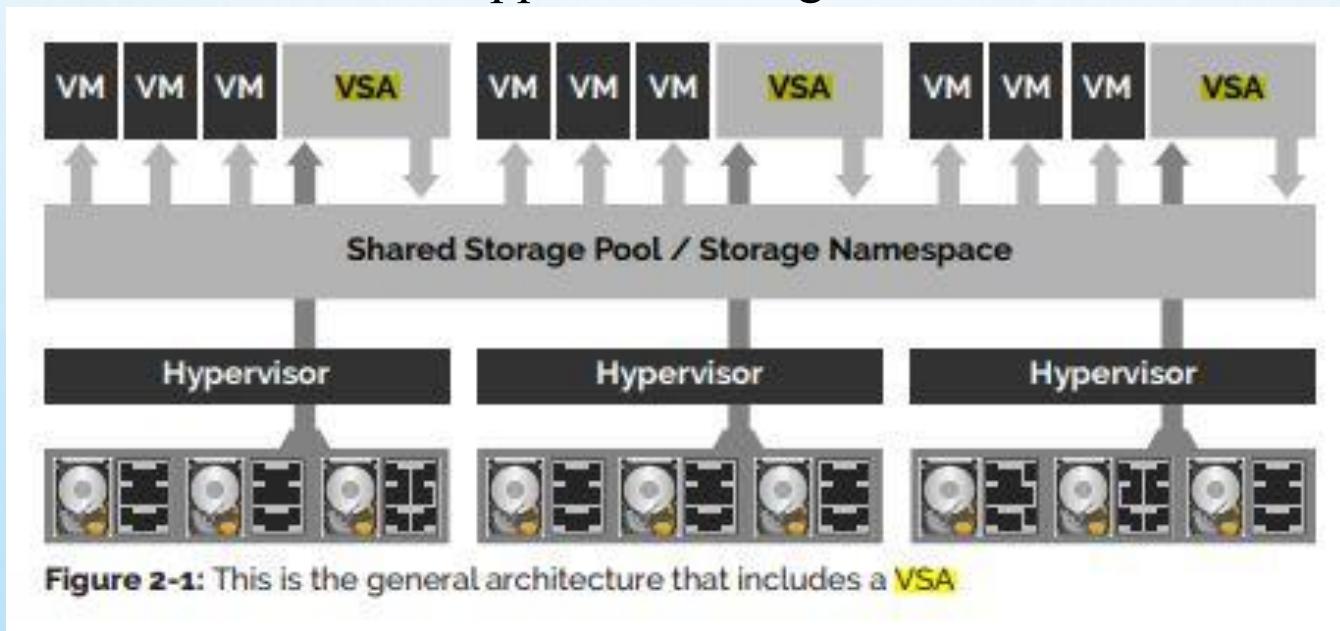
RAID: redundant array of independent disks

SOFTWARE DEFINED STORAGE

- With **storage hardware fully abstracted into software**, it becomes possible to bring policy-based management and APIs to bear in ways that focus efforts on the management on the virtual machine rather than the LUN (logical unit number)
- **The virtual machine (VM) is really the administrative target of interest whereas a LUN is just a supporting element** that contributes to how the virtual machine functions. By moving administration up to the VM level, policies can be applied more evenly across the infrastructure.
- A **VSA (virtual storage appliance)** is a virtual machine that runs on a host computer. This virtual machine's purpose is to manage the storage that is local to that host.

SOFTWARE DEFINED STORAGE (CONTD..)

- The VSAs on individual hosts work together to create a shared storage pool and global namespace. This storage is then presented back to the virtual hosts and used to support virtual machines in the environment.
- Hyperconverged infrastructure companies such as SimpliVity, Nutanix, and Maxta all use VSAs to support the storage element of the solution.



ARCHITECTING THE HYPER-CONVERGED DATA CENTER:

DECISION 3: **DATA PROTECTION SERVICES**

- **Data protection** shouldn't be considered an afterthought in your data center.
- It should be considered a **core service that is central** to how IT operates.
- Recovery Time Objectives (RTOs) and Recovery Point Objectives (RPOs) should be **a key discussion point** as you're considering hyper-converged infrastructure solutions.
- Not all hyper-converged products come with the same levels of data protection.

ARCHITECTING THE HYPER-CONVERGED DATA CENTER:

DECISION 4: THE MANAGEMENT LAYER

- To simplify management in the data center, admins need as few interfaces possible.
- Here are the most common options that you need to be aware of when considering a hyper-converged virtual infrastructure:
- **Virtualization layer management**
- **Orchestration and Automation Layer Management**
 - Vmware's vRealize Automation (vRA), Cisco's Unified Computing System Director (or UCSD), OpenStack — Rapidly growing in popularity, OpenStack
- **Vendor supplied management**
- **REST APIs**

HYPERCONVERGENCE & THE PUBLIC CLOUD

- Public cloud systems are comprised of **multi-tenant environments** operated by a service provider with the hardware and software located in the provider's data center.
- In these environments, **the customer may not always even be aware in which provider data enter the services reside, nor does the customer have to be aware.**
- The beauty of these systems is that workloads can move around as necessary to maintain service-level agreements.
- For the public cloud, there are a number of pros and cons to consider.

HYPERCONVERGENCE & THE PUBLIC CLOUD (CONTD..)

- Public Cloud Advantages**

- Enable immediate implementation.
- Carry low to no initial deployment costs.
- Provide a consumption-based utility cost model.
- Provide more cost effective scale than would be feasible in a private data center.

HYPERCONVERGENCE & THE PUBLIC CLOUD (CONTD..)

• Public Cloud Disadvantages

- Potentially unpredictable ongoing usage charges
- Concerns around data location; many do not want data stored in US-based data centers due to concerns around the NSA and PATRIOT Act
- Charges across every aspect of the environment, from data storage to data transfer and more
- No control over underlying infrastructure
- Care needs to be taken to avoid lock-in

HYPERCONVERGENCE & THE PUBLIC CLOUD (CONTD..)

The faces of the public cloud

Software-as-a-Service (SaaS)

- From a customer perspective, software-as-a-service (SaaS) is the simplest kind of cloud service to consume as it is basically an application all wrapped up and ready to go.
- Common SaaS applications include Salesforce and Office 365.
- **With SaaS applications, the provider controls everything and provides to the customer an application layer interface that only controls very specific configuration items.**

HYPERCONVERGENCE & THE PUBLIC CLOUD (CONTD..)

The faces of the public cloud

Platform-as-a-Service (PaaS)

- Sometimes, you don't need or want a complete application.
- In many cases, you just need a place to install your own applications but you don't want to have to worry at all about the underlying infrastructure or virtualization layers. This is where platform-as-a-service (PaaS) comes into play.
- **PaaS provides you with infrastructure and an application development platform that gives you the ability to automate and deploy applications including your own databases, tools, and services. As a customer, you simply manage the application and data layers.**

HYPERCONVERGENCE & THE PUBLIC CLOUD (CONTD..)

The faces of the public cloud

Infrastructure-as-a-Service (IaaS)

- In other cases, you need a bit more control, but you still may not want to have to directly manage the virtualization, storage, and networking layers.
- However, you need the ability to deploy your own operating systems inside vendor-provided virtual machines. Plus, you want to have the ability to manage operating systems, security, databases, and applications.
- For some, **infrastructure-as-a-service (IaaS) makes the most sense since the provider offers the network, storage, compute resources, and virtualization technology while you manage everything else.**

HYPERCONVERGENCE & THE PRIVATE CLOUD

- A private cloud environment generally resides in a **single tenant environment** that is built out in an on-premises data center, but it can sometimes consist of a single tenant environment in a public data center.
- Private cloud environments are characterized by heavy virtualization which fully abstracts the applications from underlying hardware components.
- Virtualization is absolutely key to these kinds of environments. Some companies go so far as to offer internal service level agreements to internal clients in a cloud-like manner.

HYPERCONVERGENCE & THE PRIVATE CLOUD (CONTD..)

Moving to more of a self-service model has two primary benefits:

- Users get their needs serviced faster
- IT is forced to build or deploy automation tools to enable self-service functionality, thereby streamlining the administrative experience

HYPERCONVERGENCE & THE PRIVATE CLOUD (CONTD..)

• Private Cloud Advantages

- Provides an opportunity to shift workloads between servers to best manage spikes in utilization in a more automated fashion.
- Enables ability to deploy new workloads on a common infrastructure. Again, this comes courtesy of the virtualization layer.
- Provides full control of the entire environment, from hardware to storage to software in a way that enables operational efficiency.
- Allows customers to customize the environment since they own everything.
- Provides additional levels of security and compliance due to the single tenant nature of the infrastructure.
- Private cloud-type environments are often the default due to security concerns.

HYPERCONVERGENCE & THE PRIVATE CLOUD (CONTD..)

- Private Cloud Disadvantages**

- Requiring customers to build, buy, and manage hardware.
This is often something that many companies want to reduce or eliminate.
- Not always resulting in operational efficiency gains.
- Not really providing what is considered a cloud computing economic model.
- You still have to buy and maintain everything.
- Potentially carrying very high acquisition costs.

HYPERCONVERGENCE & THE HYBRID CLOUD

(CONTD..)

- Increasingly, people are choosing both cloud options – public and private – to meet their needs.
- In a hybrid cloud scenario, the company builds its own on-premises private cloud infrastructure to meet local applications needs and also leverages public cloud where reasonable and possible.

THE INTERSECTION OF CLOUD AND HYPERCONVERGED INFRASTRUCTURE (CONTD..)

- Depending on the hyperconverged infrastructure solution you're considering, there are varying degrees of association between the hyperconverged infrastructure product and both public and private clouds
- Virtualization is absolutely key to these kinds of environments. Some companies go so far as to offer internal service level agreements to internal clients in a cloud-like manner.

HCI VS. CLOUD

- An HCI platform is an integrated platform that virtualizes compute, storage and network resources and combines them with a software-defined management system to provide a unified platform for hosting virtualized workloads. The platform is made up of multiple server and storage nodes, along with the necessary networking components, to form a single cluster that IT can easily deploy, maintain and scale.

CONT..

- A cloud computing infrastructure, whether private or public, is made up of physical compute, storage and network resources that are integrated into a single architecture. An abstraction layer pools the physical resources and delivers them as services, which applications and users can easily configure and deploy, either through an API or user interface. However, setting up the infrastructure can be a complex and time-consuming task.

CONT..

- Virtualization lies at the heart of both cloud computing and HCI, but virtualization alone isn't enough to define either one. A cloud environment is directly concerned with the user experience, using advanced automation and orchestration to compose the underlying infrastructure. Hyper-convergence has more to do with simplifying IT, following a rigid node-based architecture that greatly simplifies administration but decreases flexibility. Although some HCI platforms have incorporated cloud-like capabilities, they still remain two different approaches to IT infrastructure.

CONT..

- PERFORMANCE
- SCALABILITY
- RELIABILITY
- WORKLOADS
- STORAGE
- DATA PROTECTION
- MANAGEMENT
- COSTS

REF: [HTTPS://WWW.TECHTARGET.COM/SEARCHDATACENTER/TIP/HCI-VS-CLOUD-THE-MAIN-DIFFERENCES](https://www.techtarget.com/searchdatacenter/tip/hci-vs-cloud-the-main-differences)

CASE STUDY

**Batlivala & Karani Securities migrates from traditional IT to
Hyper-Converged Infrastructure with Galaxy as
Implementation Partner**

CASE STUDY (CONTD..)

- Batlivala & Karani [B & K] Securities India Pvt. Ltd. is a leading financial research and advisory firm in India.
- Established in 1875, B & K Securities leverages their research strengths, keen business insights and years of experience to offer unbiased and in-depth financial advice to institutional and corporate customers.
- Their scope of services includes providing Equity Advisory & Research services, Mutual Fund Analysis & Distribution, Strategic Consulting, Insurance Advisory & Broking, and Investment Banking.
- Since 2017, they've embarked on a digital journey by migrating traditional IT to hyper converged infrastructure.
- Mukesh Mehta says "**Traditional Infrastructures take potentially months of planning prior to an install, and can take weeks to even months whereas Hyper-Converged solutions can be sized to current needs, reducing planning time, installs typically take days or even hours.**"

CASE STUDY (CONTD..)

- **The Challenges: -**

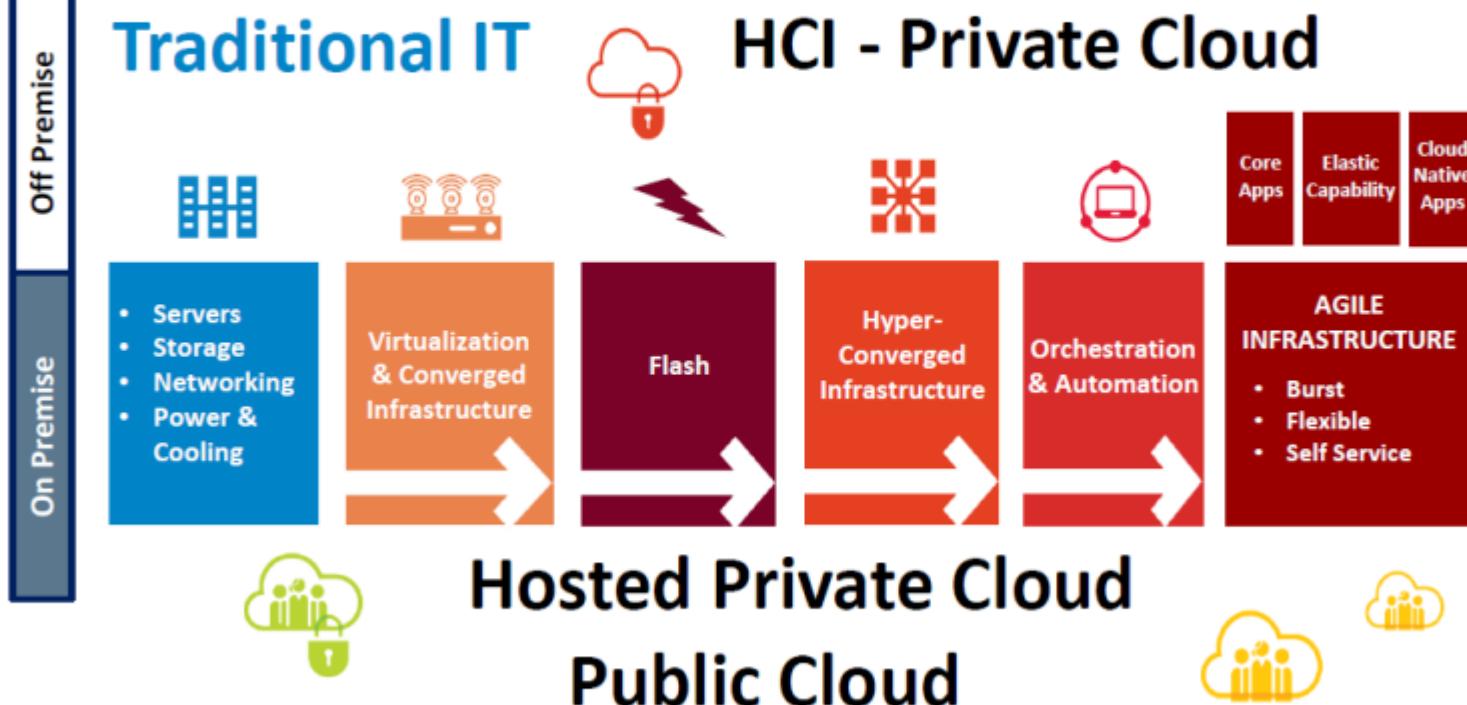
- Prior to virtual DC Infrastructure migration, B & K Securities **had over 30 servers** in their office [both Windows and Linux-based], with mixed bag of virtualized & stand-alone Servers.
- Thus, **Warranty and end-of-life** issues across components, product, as well as software used to be a major issue.
- Further, **individual downtime & dependencies, scalability and handling issues with multiple consoles** like admin/management modules (username/password & respective policies) turned out to be a hurdle in their smooth operations.
- On the other hand, Mukesh Mehta wanted to bring down the **TCO (i.e. IT opex) costs linked** to more space and electrical power consumption, as well as **air-conditioned cooling requirements needed for so many physical servers**.
- In addition, there were other **operational costs linked to periodic maintenance**, and additional resources needed to manage servers in different environments – Windows and Linux

CASE STUDY (CONTD..)



Journey to Integrated Infrastructure

Modern Data Center Environment: Elastic, Scalable, Efficient, Fast



CASE STUDY (CONTD..)

- **The Solution: -**

- Hyper-convergence solutions leverage Intel x86-based servers to natively integrate compute, storage, virtualization and management into a single turnkey platform.
- This replaces silos of servers, storage and virtualization, along with the need for having separate management tools and processes.
- A 100% software-defined storage solution runs on a hypervisor that aggregates the storage and compute performance between nodes and servers, thereby merging these discrete tiers into a single tier.
- As workload requirements grow, organizations simply add another node to an existing hyper-converged deployment to aggregate the additional resources into the cluster.
- A single vendor provides the hardware and software, eliminating all the complexity that comes with procuring and provisioning these components.
-

CASE STUDY (CONTD..)

- **The Solution:** -

-
- Overall management and day-to-day operations are greatly simplified with only one system to manage.
- It's easy to see how businesses can attain enormous cost savings by eliminating expensive storage, overprovisioning, hypervisor licensing costs and the need for specialized skill sets to manage all the complexity.
- The end result of this shift is accelerated deployment, improved performance, and cloud-like consumption and management, all at a fraction of the cost of legacy system architectures.

NEXT CLASS....

VIRTUAL MACHINES PROVISIONING AND MIGRATION SERVICES

L13-L16

DR. MANJUNATH V H AND DR. VIDYA RAO

VIRTUAL MACHINES PROVISIONING AND MANAGEABILITY

Virtual Machine Life Cycle

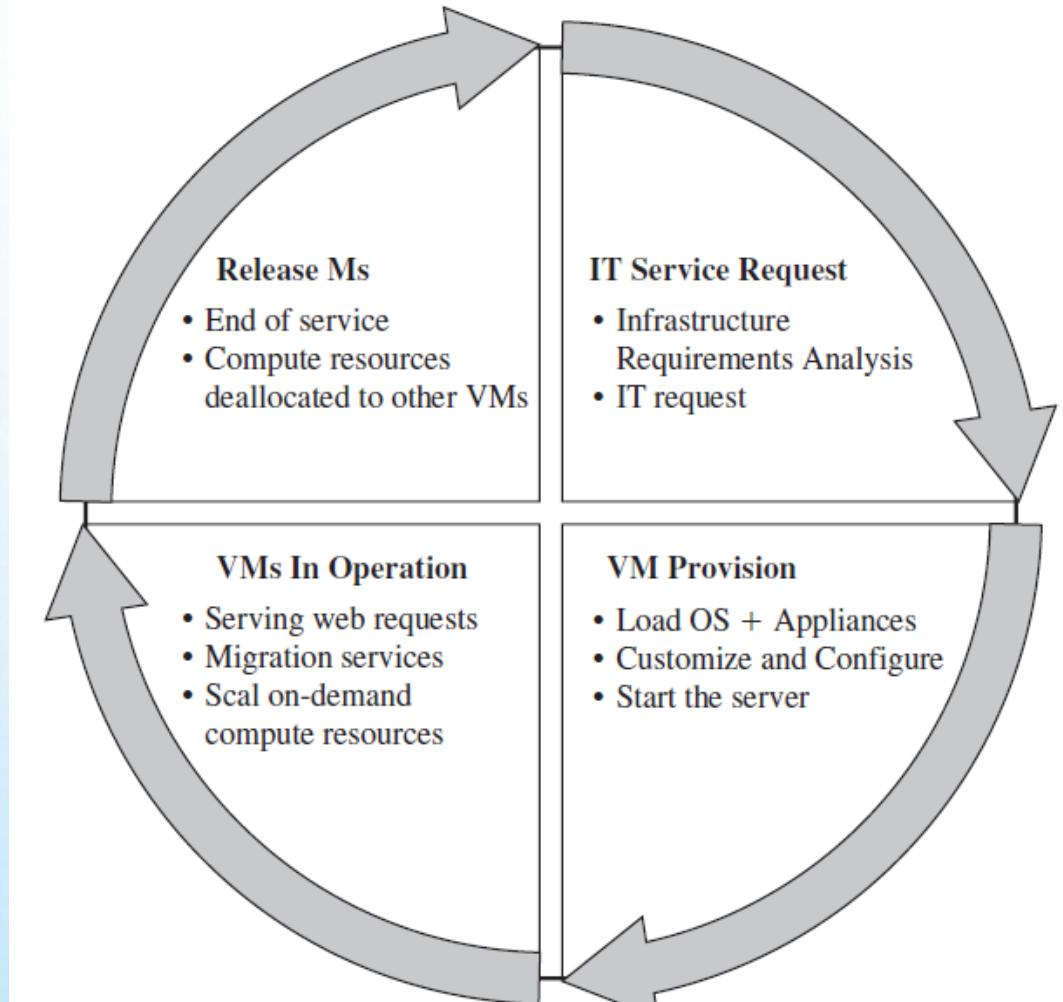


Fig: Virtual Machine Life Cycle

VM PROVISIONING PROCESS



Fig: Virtual machine provision process

MIGRATIONS TECHNIQUES

- Live Migration and High Availability
 - VMware Vmotion
 - Citrix XenServer XenMotion
- Regular/Cold Migration
- Live Storage Migration of Virtual Machine

MIGRATIONS TECHNIQUES

(CONTD..)

Live Migration and High Availability

- aka hot or real-time migration
- **Defined** as the movement of a virtual machine from one physical host to another while being powered on.
- Without any noticeable effect from the end user's point of view (a matter of milliseconds).
- **Advantages**
 - Facilitates proactive maintenance in case of failure, because the potential problem can be resolved before the disruption of service occurs.
 - Load balancing in which work is shared among computers in order to optimize the utilization of available CPU resources

MIGRATIONS TECHNIQUES

(CONTD..)

Live Migration Anatomy, Xen Hypervisor Algorithm

Xen hypervisor: Explain live migration's mechanism and how memory and virtual machine states are being transferred, through the network, from one host A to another host B

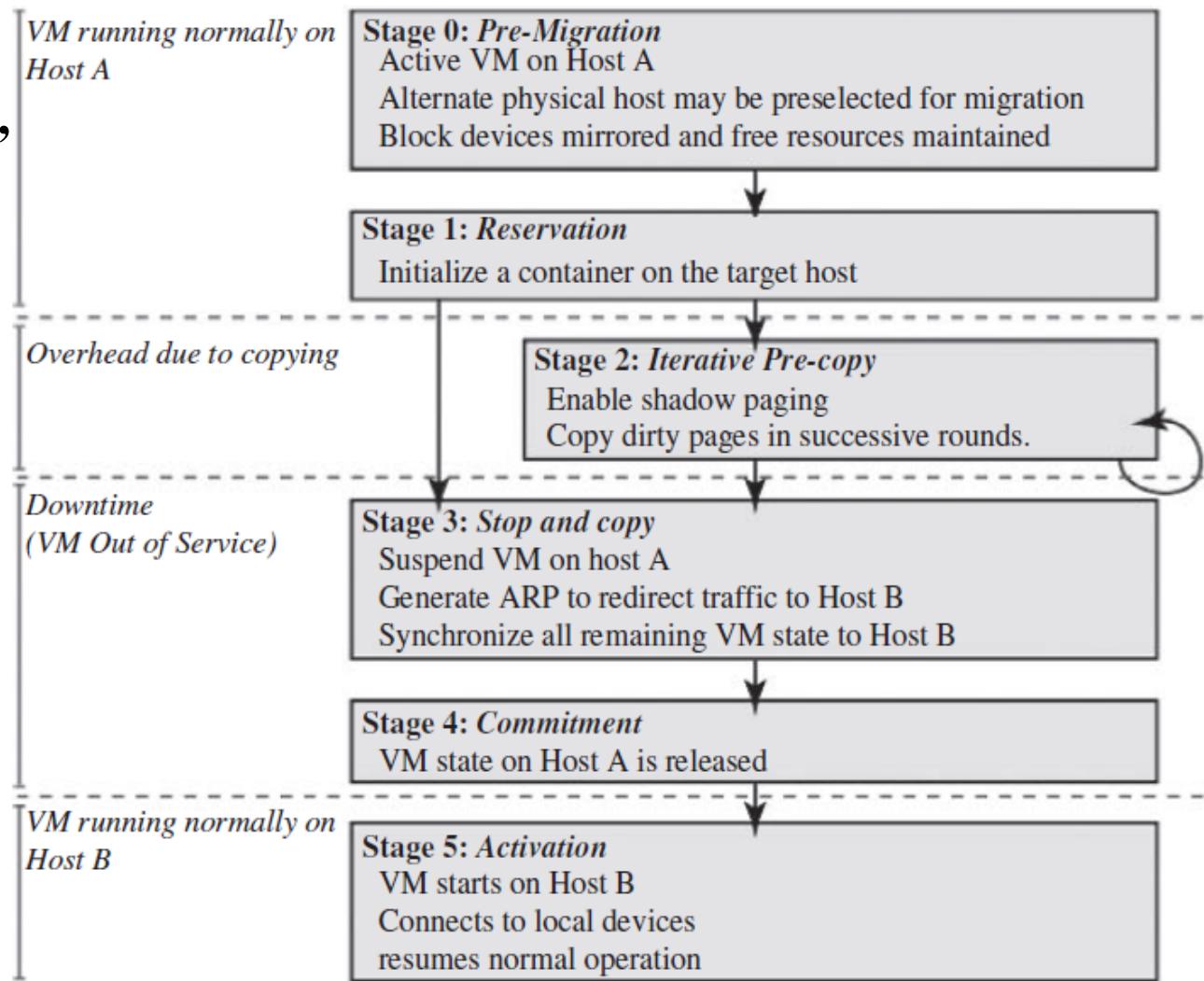


Fig: Live migration timeline

MIGRATIONS TECHNIQUES

(CONTD..)

The migration process has been viewed as a transactional interaction between the two hosts involved:

- **Stage 0:** Pre-Migration. An active virtual machine exists on the physical host A.
- **Stage 1:** Reservation. A request is issued to migrate an OS from host A to host B (a precondition is that the necessary resources exist on B and on a VM container of that size).
- **Stage 2:** Iterative Pre-Copy. During the first iteration, all pages are transferred from A to B. Subsequent iterations copy only those pages dirtied during the previous transfer phase.
- **Stage 3:** Stop-and-Copy. Running OS instance at A is suspended, and its network traffic is redirected to B. As described in reference 21, CPU state and any remaining inconsistent memory pages are then transferred. At the end of this stage, there is a consistent suspended copy of the VM at both A and B. The copy at A is considered primary and is resumed in case of failure
- **Stage 4:** Commitment. Host B indicates to A that it has successfully received a consistent OS image. Host A acknowledges this message as a commitment of the migration transaction. Host A may now discard the original VM, and host B becomes the primary host
- **Stage 5:** Activation. The migrated VM on B is now activated. Post-migration code runs to reattach the device's drivers to the new machine and advertise moved IP addresses

MIGRATIONS TECHNIQUES

(CONTD..)

Live Migration Vendor Implementations Examples

VMware Vmotion

- a) Automatically optimize and allocate an entire pool of resources for maximum hardware utilization, flexibility, and availability.
- b) Perform hardware's maintenance without scheduled downtime along with migrating virtual machines away from failing or underperforming servers

**Citrix XenServer
XenMotion**

- a) Inherited from the Xen live migrate utility, which provides the IT administrator with the facility to move a running VM from one XenServer to another in the same pool without interrupting the service.
- b) Also, balance the workloads on the virtualized environment.

MIGRATIONS TECHNIQUES

(CONTD..)

Regular/Cold Migration

- Cold migration is the migration of a powered-off virtual machine.
- With cold migration, you have the option of moving the associated disks from one data store to another.
- The virtual machines are not required to be on a shared storage.

Live v/s Cold migration

- Live migration needs a shared storage for virtual machines in the server's pool, but cold migration does not;
- In live migration for a virtual machine between two hosts, there would be certain CPU compatibility checks to be applied; while in cold migration this checks do not apply.

MIGRATIONS TECHNIQUES

(CONTD..)

Cold Migration Process:

- The configuration files, including the NVRAM file (BIOS settings), log files, as well as the disks of the virtual machine, are moved from the source host to the destination host's associated storage area.
- The virtual machine is registered with the new host.
- After the migration is completed, the old version of the virtual machine is deleted from the source host

MIGRATIONS TECHNIQUES

(CONTD..)

Live Storage Migration of Virtual Machine:

- This kind of migration constitutes moving the virtual disks or configuration file of a running virtual machine to a new data store without any interruption in the availability of the virtual machine's service.

Migration of Virtual Machines to Alternate Platforms:

- Ability to migrate virtual machines from one platform to another.
- For example, the VMware converter that handles migrations between SX hosts; the VMware server; and the VMware workstation.
- The VMware converter can also import from other virtualization platforms, such as Microsoft virtual server machines

VM PROVISIONING AND MIGRATION IN ACTION

- Deployment scenario
- Installation
- Environment, software and hardware
- Adding managed servers and provisioning VM

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Deployment Scenario

ConVirt deployment consists of at least one ConVirt workstation, where ConVirt is installed and ran, which provides the main console for managing the VM life cycle, managing images, provisioning new VMs, monitoring machine resources, and so on.

Two essential deployment scenarios for ConVirt:

1. Basic configuration in which the Xen or KVM virtualization platform is on the local machine, where ConVirt is already installed;
2. An advanced configuration in which the Xen or KVM is on one or more remote servers.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Installation

- Installing ConVirt on at least one computer.
- Preparing each managed server to be managed by ConVirt.
- Starting ConVirt and discovering the managed servers you have prepared.

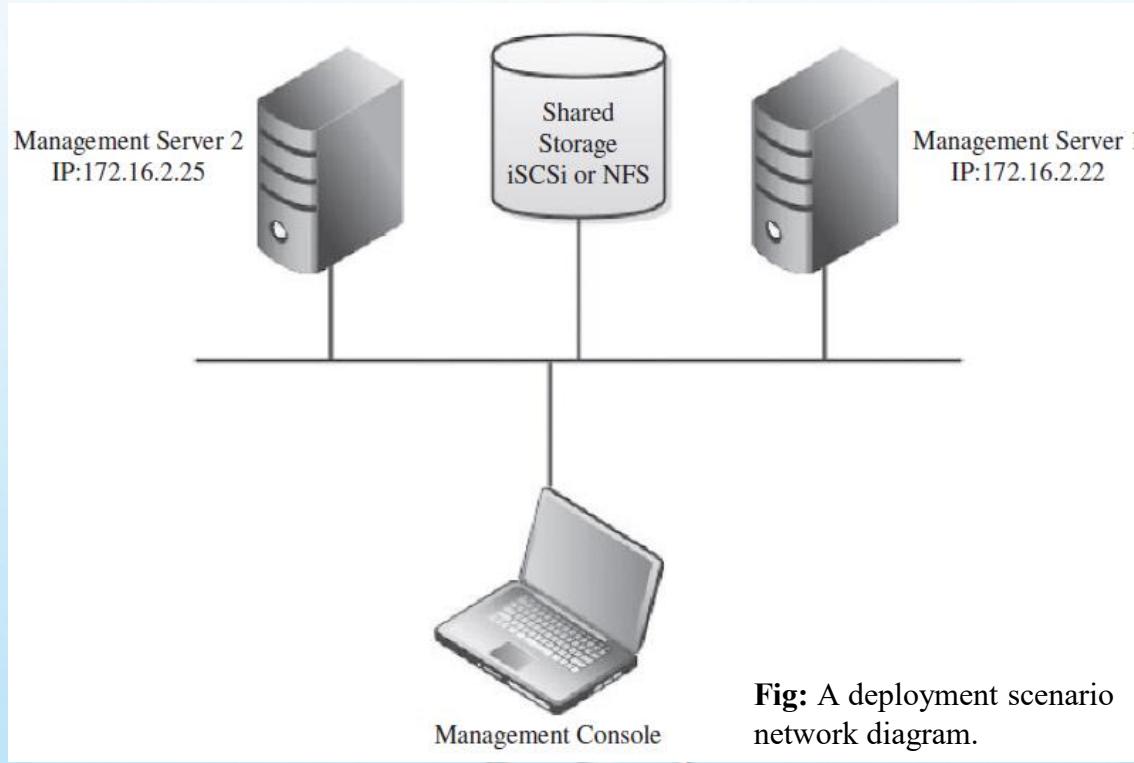


Fig: A deployment scenario network diagram.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Environment

Software

Hardware

ConVirt 1.1

Linux
Ubuntu 8.10

3 machines,
Dell core 2
due
processor,
4G RAM

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

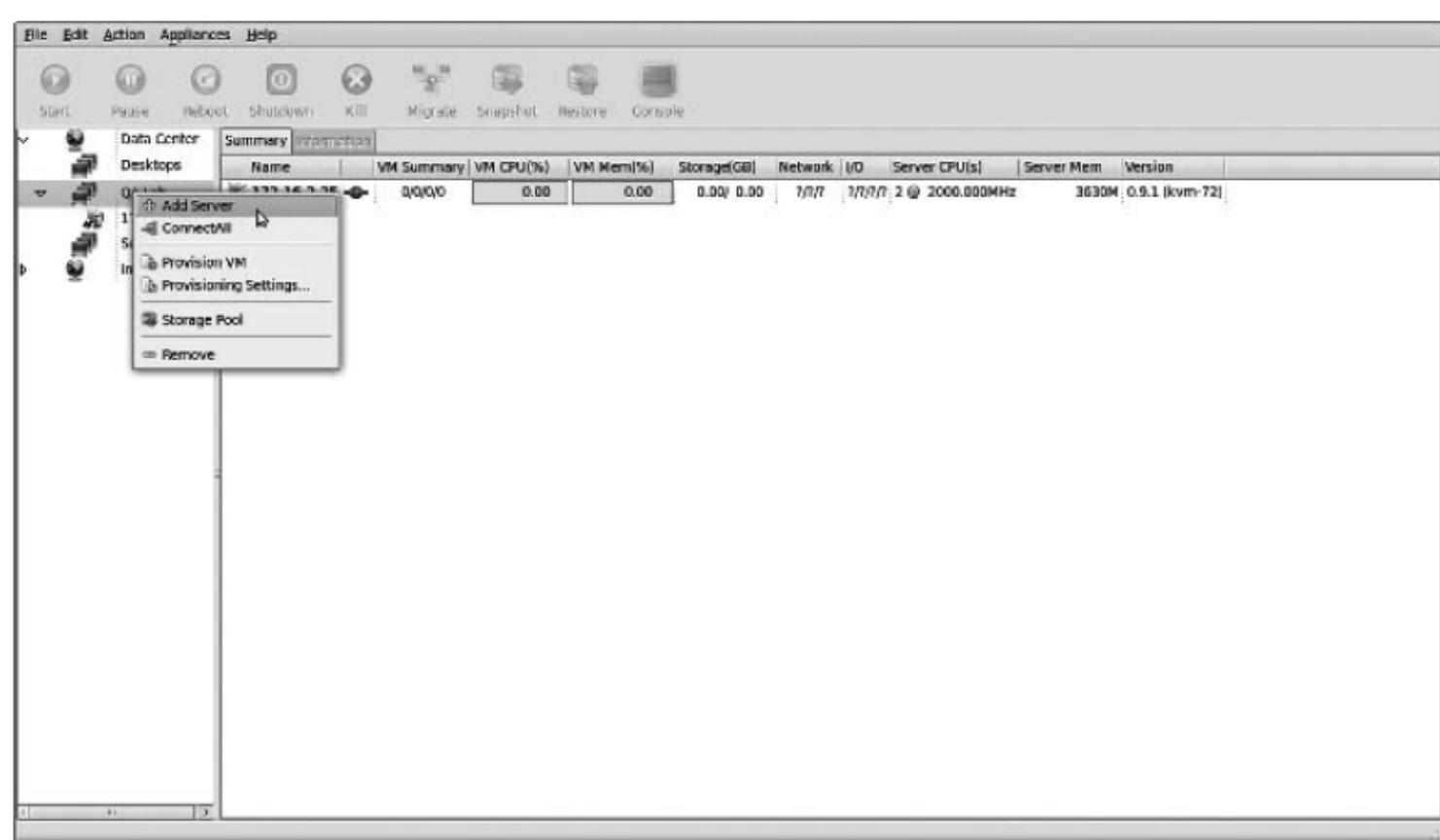


FIGURE 5.8. Adding managed server on the data centre's management console.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

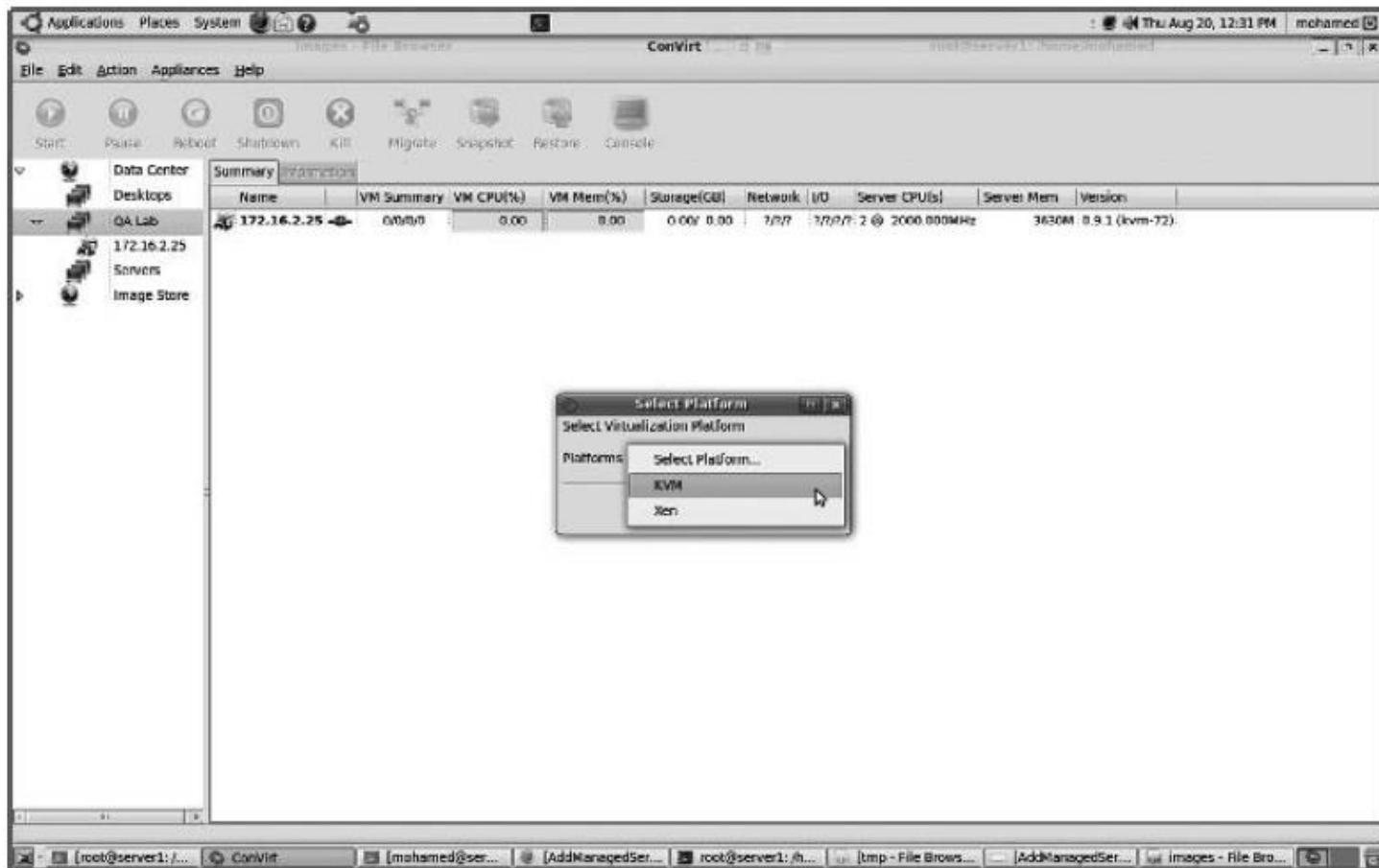


FIGURE 5.9. Select virtualization platform.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

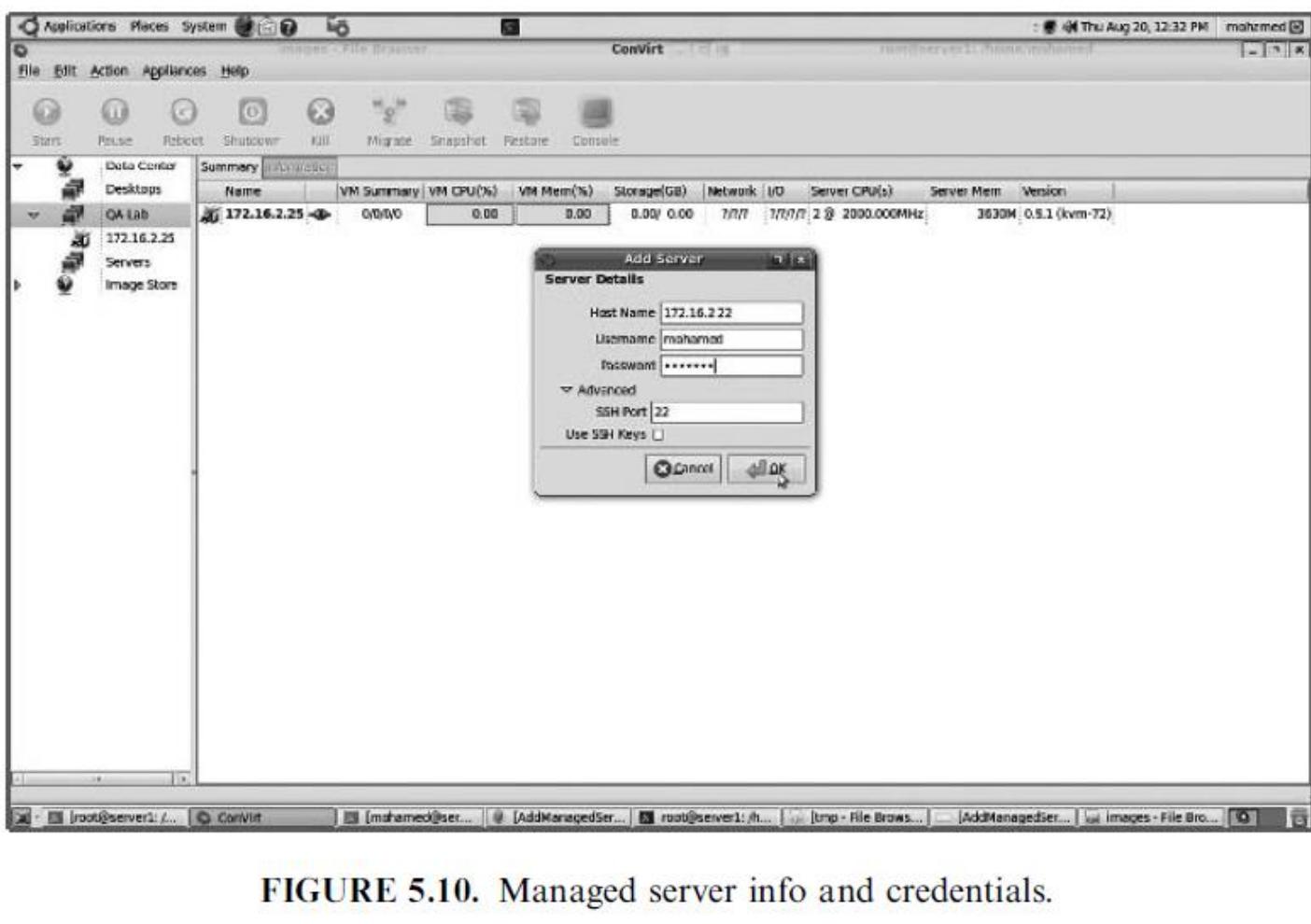


FIGURE 5.10. Managed server info and credentials.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

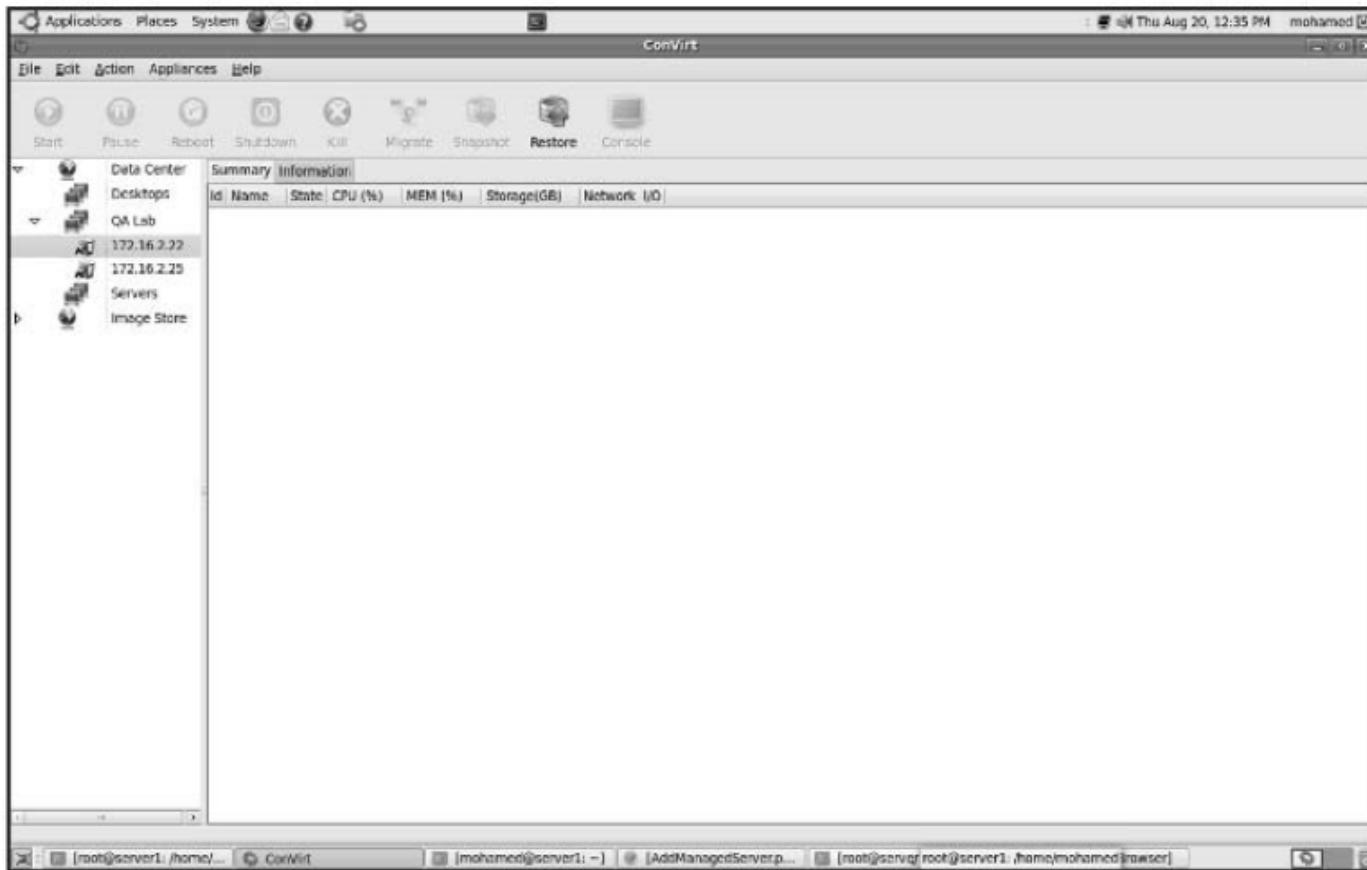


FIGURE 5.11. Managed server has been added.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

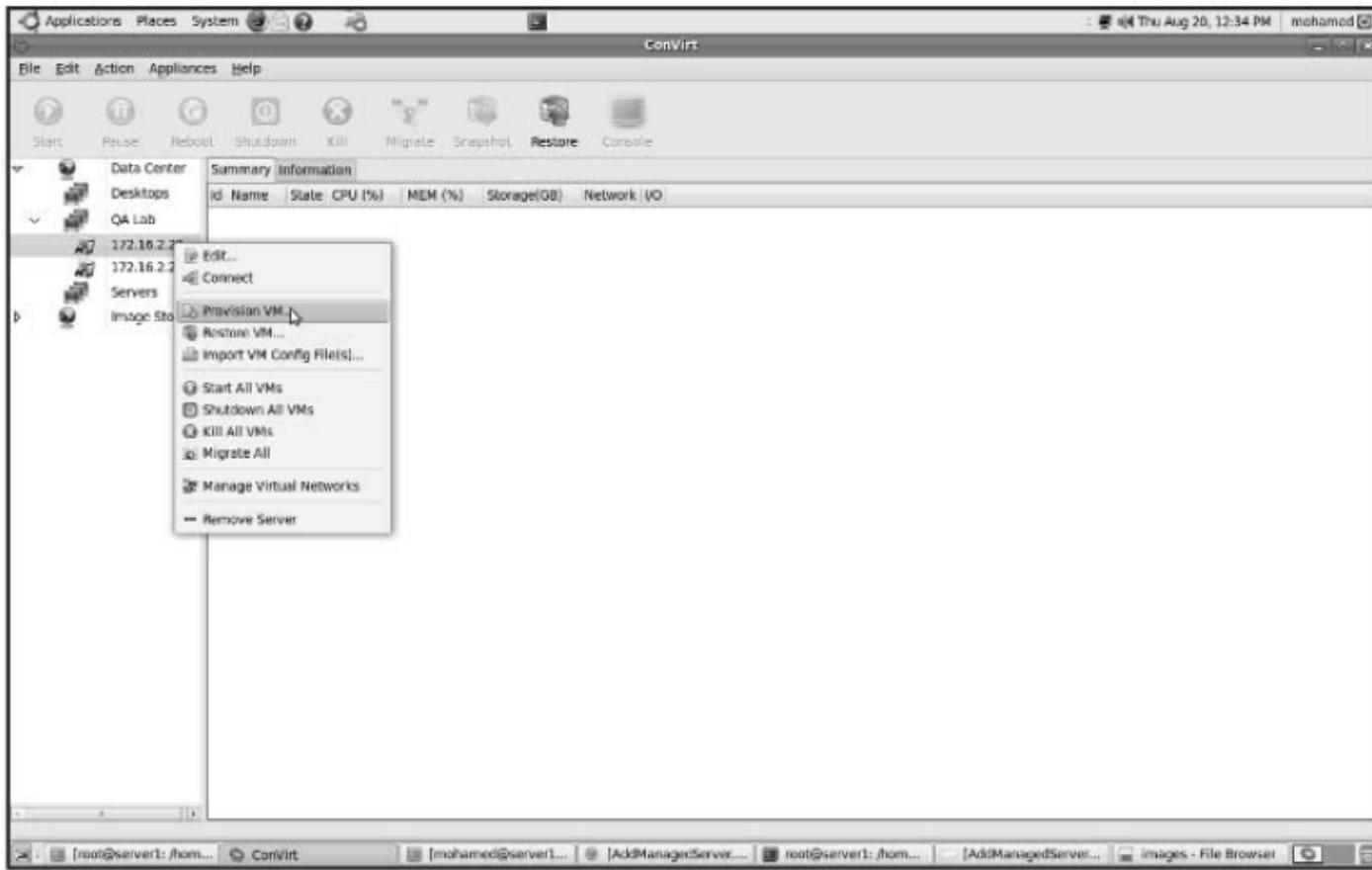


FIGURE 5.12. Provision a virtual machine.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

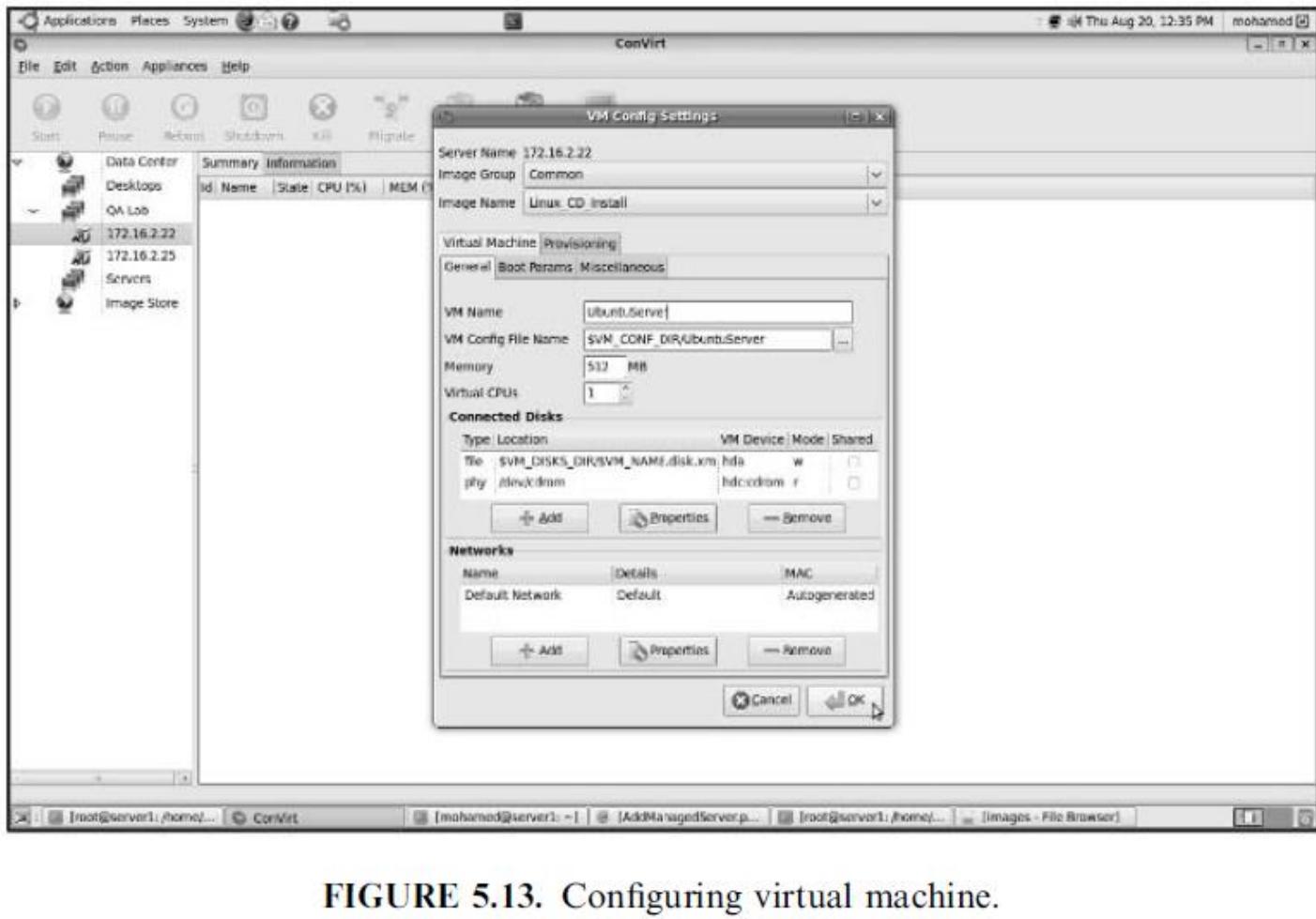


FIGURE 5.13. Configuring virtual machine.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

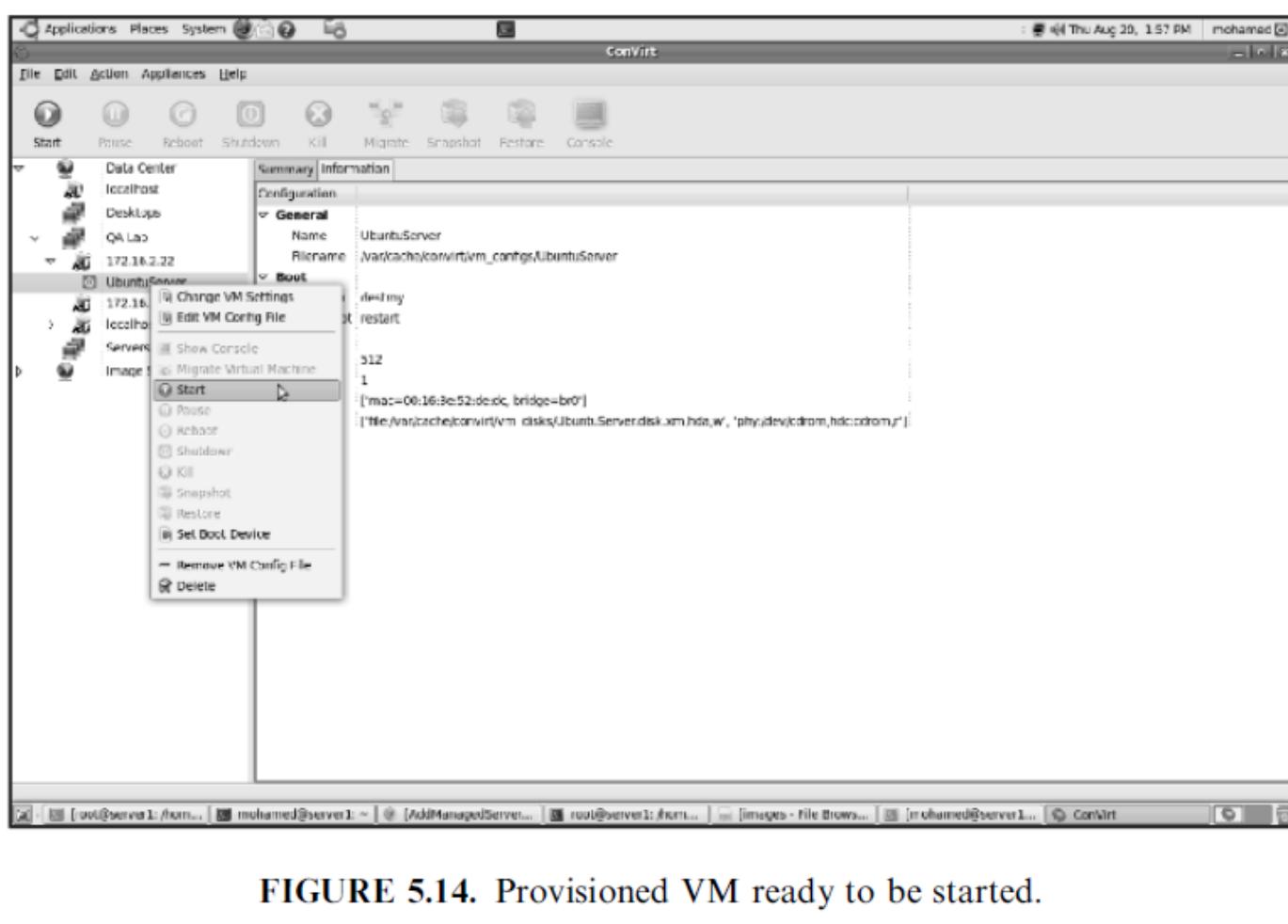


FIGURE 5.14. Provisioned VM ready to be started.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

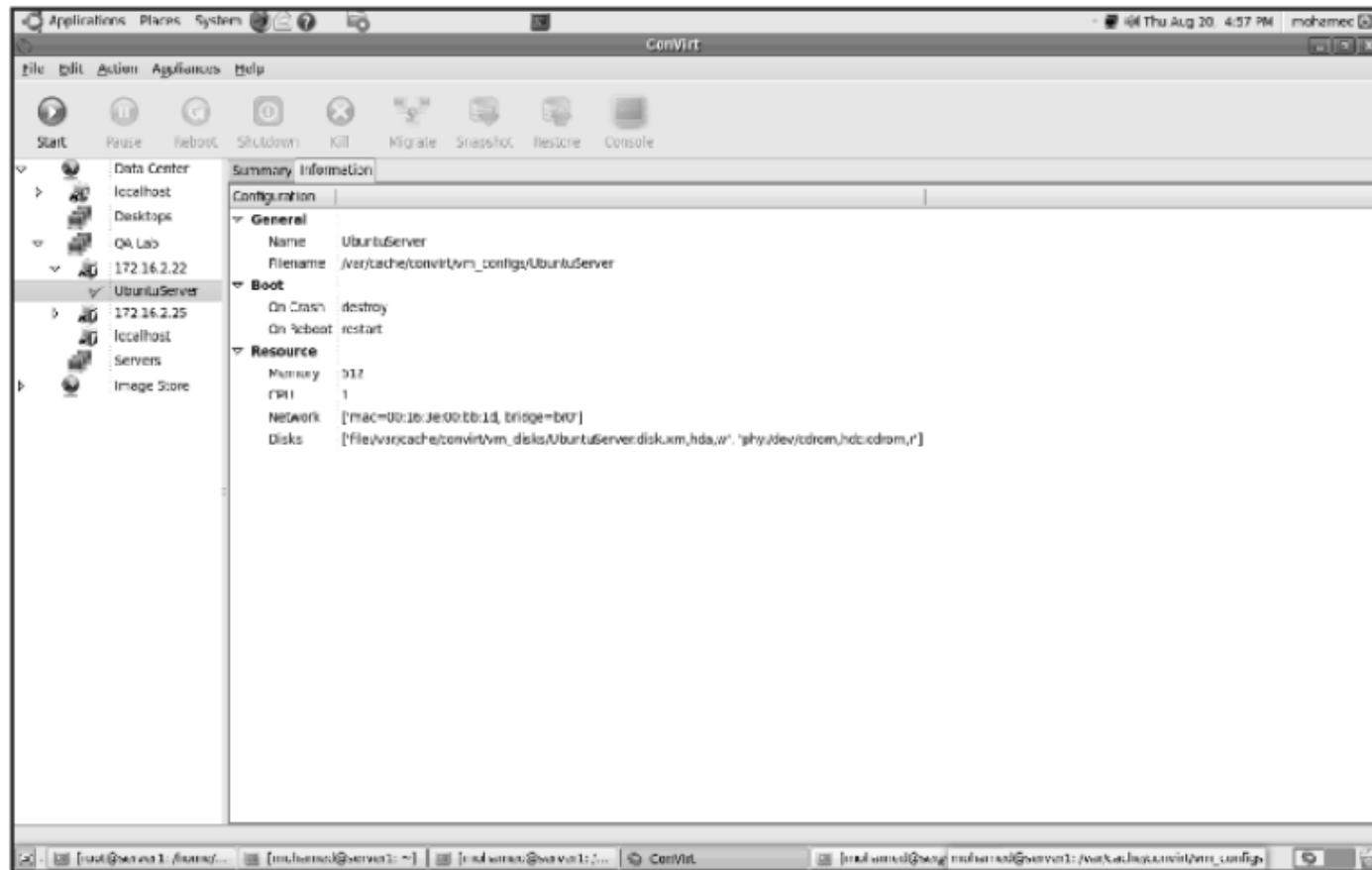


FIGURE 5.15. Provisioned VM started.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Adding Managed Servers and Provisioning VM

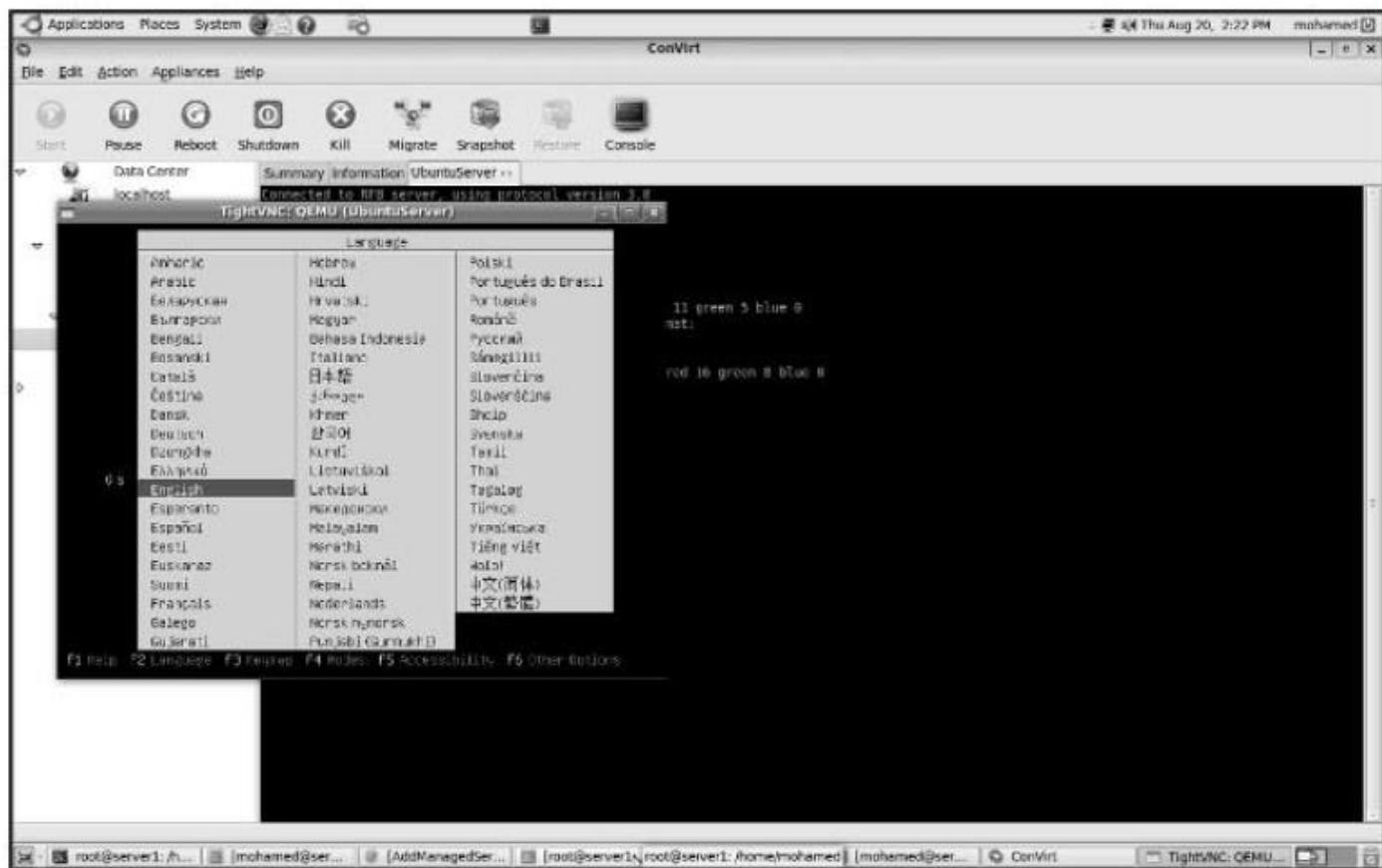


FIGURE 5.16. VM booting from the installation CD to start the installation process.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Live Migration

- ConVirt tool allows running virtual machines to be migrated from one server to another.
- For proper VM migration the following points must be considered:
 - Shared storage for all Guest OS disks (e.g., NFS, or iSCSI).
 - Identical mount points on all servers (hosts).
 - The kernel and ramdisk when using para-virtualized virtual machines should, also, be shared. (This is not required, if pygrub is used.)
 - Centrally accessible installation media (iso).
 - It is preferable to use identical machines with the same version of virtualization platform.
 - Migration needs to be done within the same subnet.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Live Migration Process

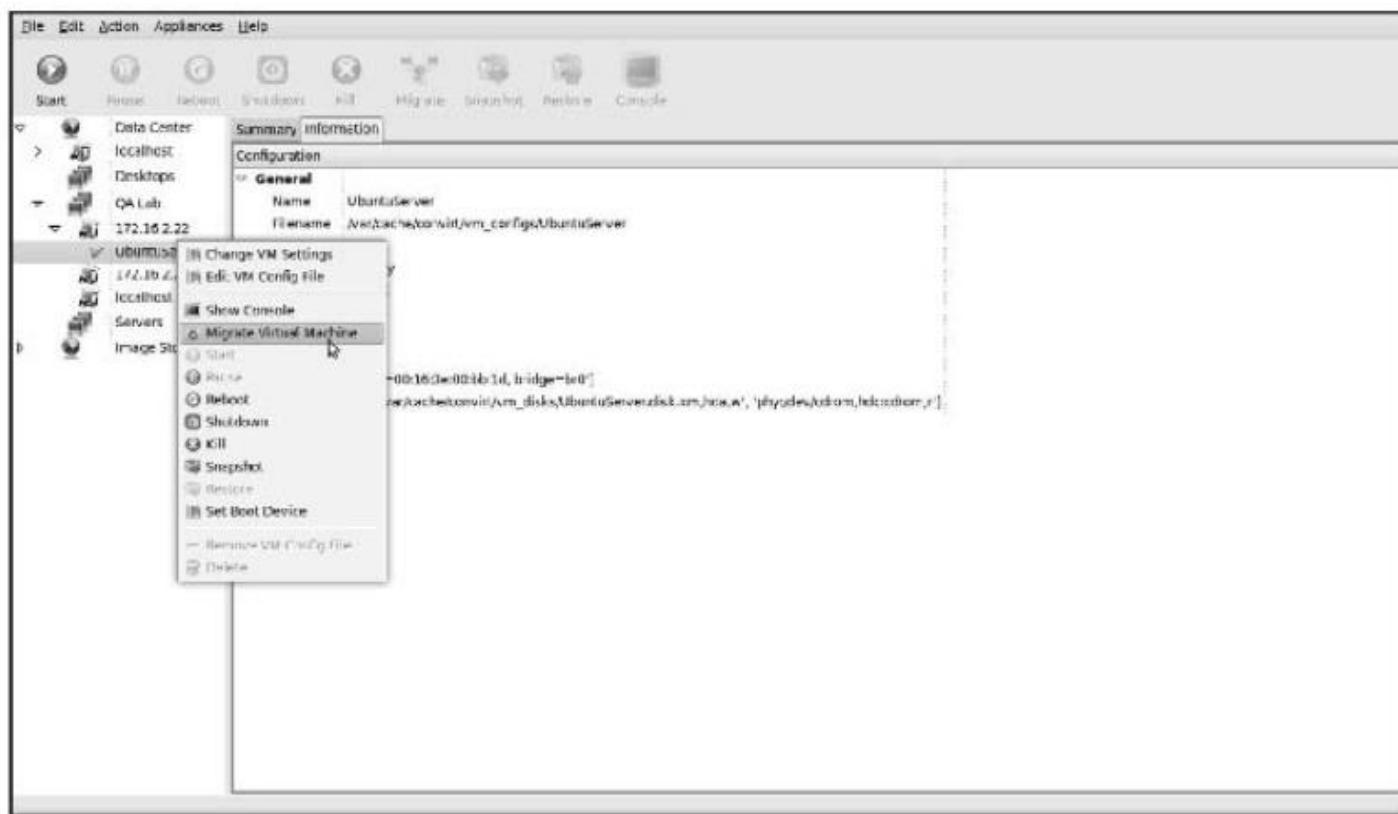


FIGURE 5.17. VM migration.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Live Migration Process

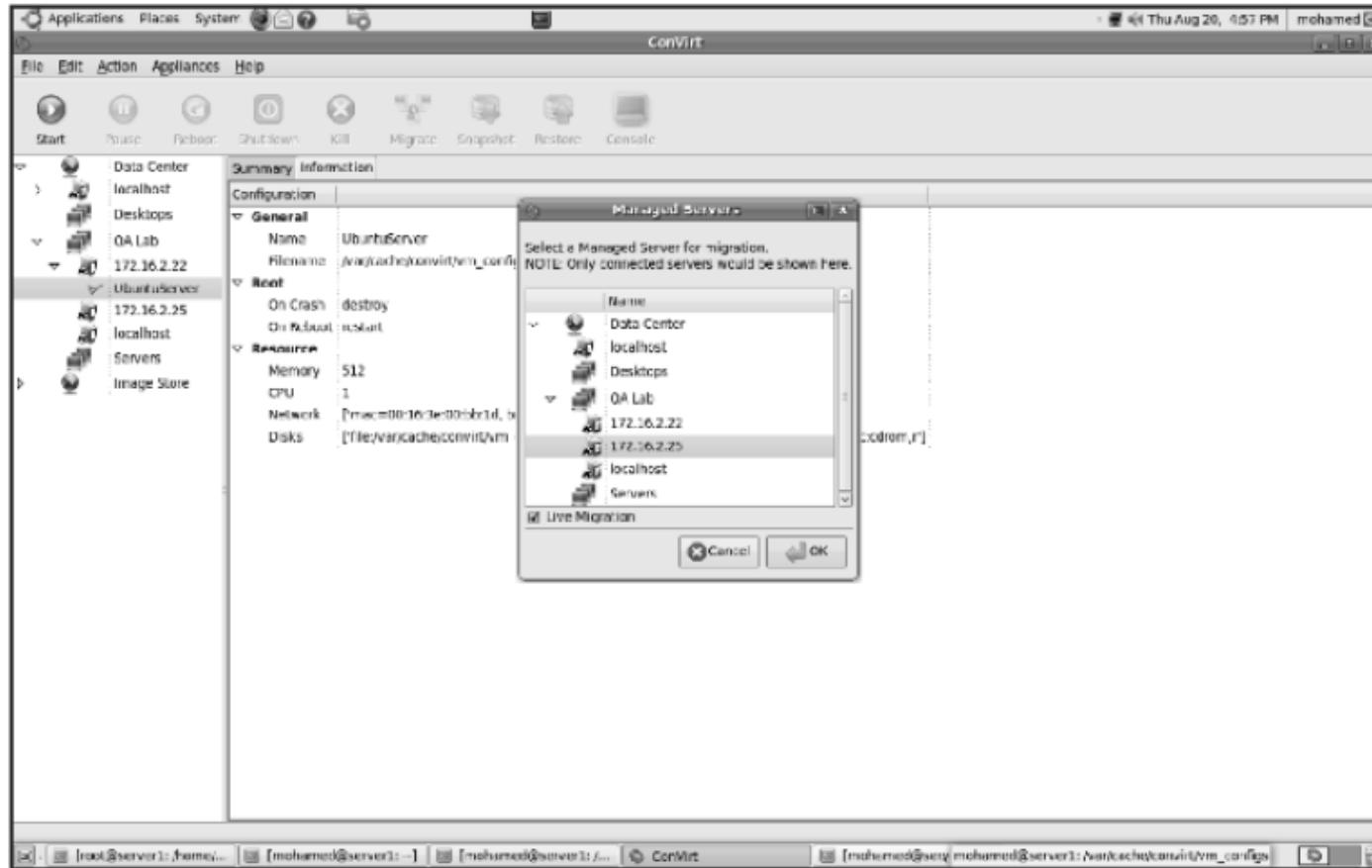


FIGURE 5.18. Select the destination managed server candidate for migration.

VM PROVISIONING AND MIGRATION IN ACTION (CONTD..)

Live Migration Process

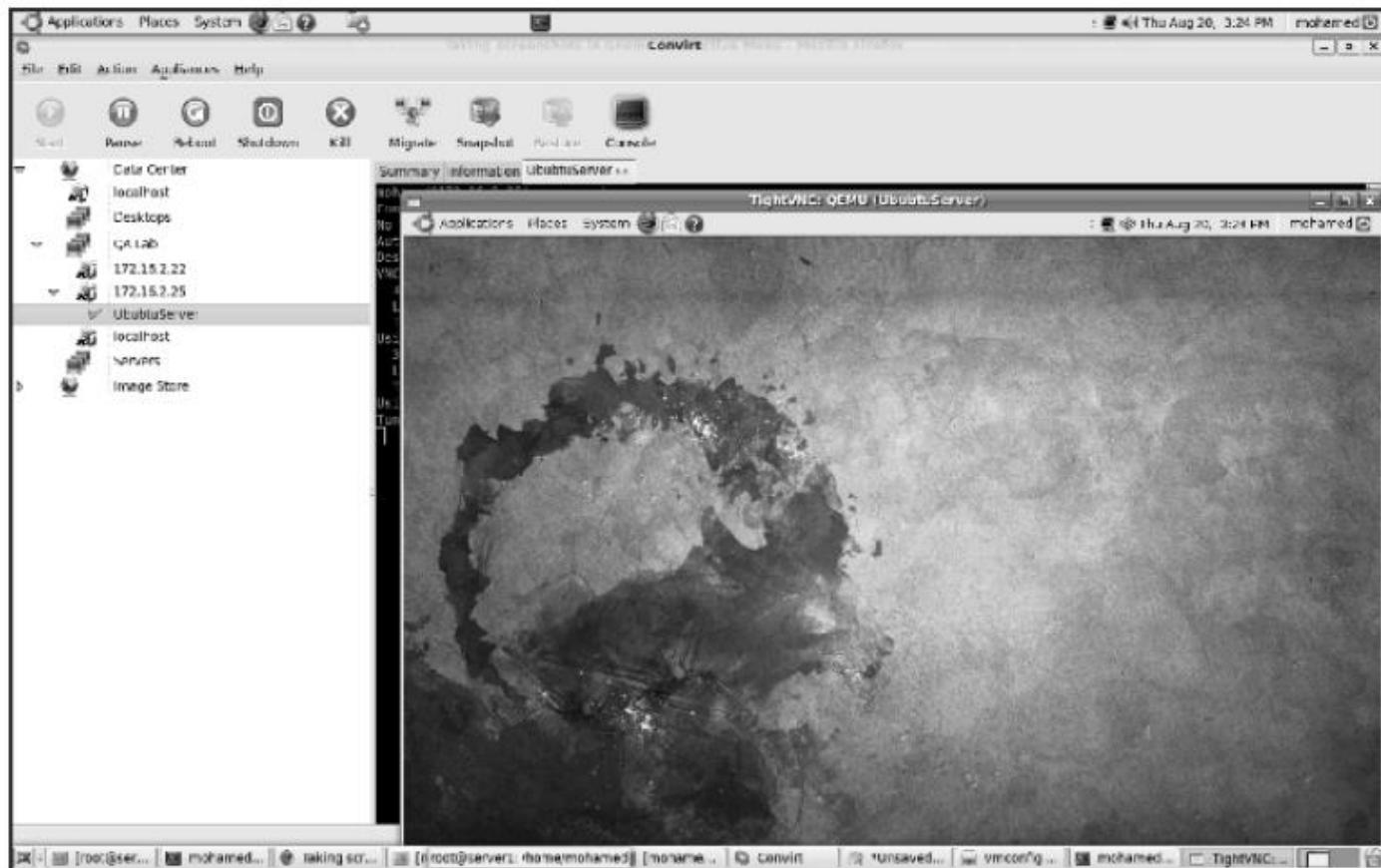


FIGURE 5.19. VM started on the destination server after migration.

PROVISIONING IN THE CLOUD CONTEXT

- Amazon Elastic Compute Cloud (EC2)
- Open source cloud tools play an invaluable role in infrastructure as a service:
 - Eucalyptus
 - Open Nebula
- Plays a prominent role in building private, public, and hybrid cloud architecture.
- The current interface to Eucalyptus is compatible with Amazon's EC2, S3, and EBS interfaces, but the infrastructure is designed to support multiple client-side interfaces.

PROVISIONING IN THE CLOUD CONTEXT

(CONTD..)

Amazon Elastic Compute Cloud (EC2)

- Web service that allows users to provision new machines into Amazon's virtualized infrastructure in a matter of minutes; using a publicly available API.
- Web services APIs allow users to reboot their instances remotely, scale capacity quickly, and use on-demand service when needed; by adding tens, or even hundreds, of machines.
- It is very important to mention that there is no up-front hardware setup and there are no installation costs, because Amazon charges only for the capacity you actually use.
- EC2 instance is typically a virtual machine with a certain amount of RAM, CPU, and storage capacity.

PROVISIONING IN THE CLOUD CONTEXT (CONTD..)

Amazon Elastic Compute Cloud (EC2)

- Amazon EC2 provides its customers with three flexible purchasing models to make it easy for the cost optimization.
 - **On-Demand instances**, which allow you to pay a fixed rate by the hour with no commitment.
 - **Reserved instances**, which allow you to pay a low, one-time fee and in turn receive a significant discount on the hourly usage charge for that instance.
 - **Spot instances**, which enable you to bid whatever price you want for instance capacity, providing for even greater savings, if your applications have flexible start and end times.

PROVISIONING IN THE CLOUD CONTEXT

(CONTD..)

Amazon and Provisioning Services

- ***Amazon Auto Scaling***, is a set of command line tools that allows scaling Amazon EC2 capacity up or down automatically and according to the conditions the end user defines.
- ***Amazon Elastic Load Balancer***, is another service that helps in building fault-tolerant applications by automatically provisioning incoming application workload across available Amazon EC2 instances and in multiple availability zones.

PROVISIONING IN THE CLOUD CONTEXT

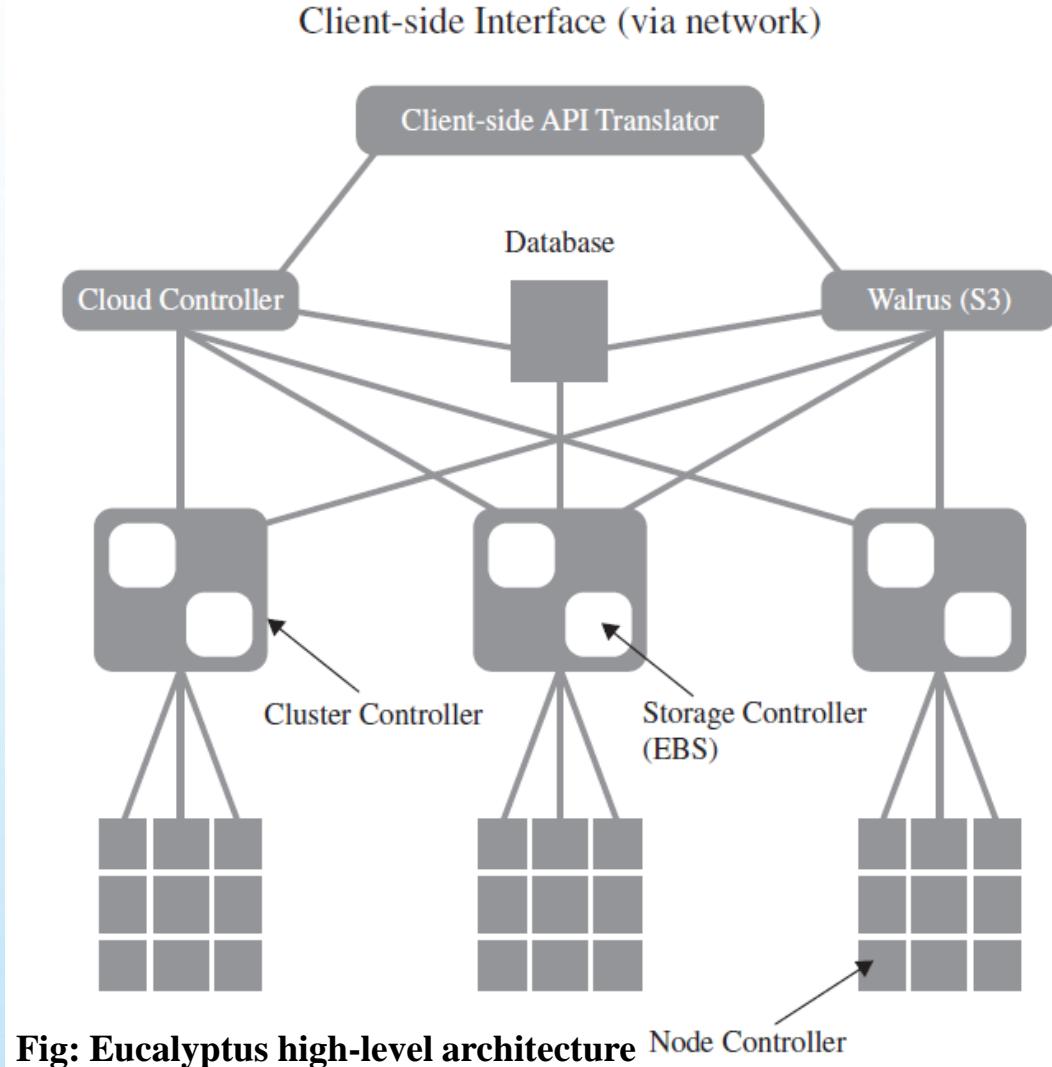
(CONTD..)

Eucalyptus

- Is an open-source infrastructure for the implementation of cloud computing on computer clusters.
- **Eucalyptus**-----“elastic utility computing architecture for linking your programs to useful systems.”
- Features:
 - Interface compatibility with EC2, and S3.
 - Simple installation and deployment.
 - Support for most Linux distributions (source and binary packages).
 - Support for running VMs that run atop the Xen hypervisor or KVM.
 - Support for other kinds of VMs, such as VMware, is targeted for future releases.
 - Secure internal communication using SOAP with WS security.
 - Cloud administrator’s tool for system’s management and user’s accounting.
 - The ability to configure multiple clusters each with private internal network addresses into a single cloud.

PROVISIONING IN THE CLOUD CONTEXT (CONTD..)

Eucalyptus Architecture



PROVISIONING IN THE CLOUD CONTEXT

(CONTD..)

Eucalyptus Architecture components

- **Node controller (NC)** controls the execution, inspection, and termination of VM instances on the host where it runs.
- **Cluster controller (CC)** gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.
- **Storage controller (SC)** is a put/get storage service that implements Amazon's S3 interface and provides a way for storing and accessing VM images and user data.
- **Cloud controller (CLC)** is the entry point into the cloud for users and administrators. It queries node managers for information about resources, makes high-level scheduling decisions, and implements them by making requests to cluster controllers.
- **Walrus (W)** is the controller component that manages access to the storage services within Eucalyptus. Requests are communicated to Walrus using the SOAP or REST-based interface.

PROVISIONING IN THE CLOUD CONTEXT (CONTD..)

VM Dynamic Management Using OpenNebula

- open and flexible tool that fits into existing data center's environments to build any type of cloud deployment.
- Primarily used as a virtualization tool to manage your virtual infrastructure, which is usually referred to as private cloud.
- OpenNebula also supports public clouds by providing cloud's interfaces to expose its functionality for virtual machine, storage, and network management.

PROVISIONING IN THE CLOUD CONTEXT (CONTD..)

OpenNebula Architecture

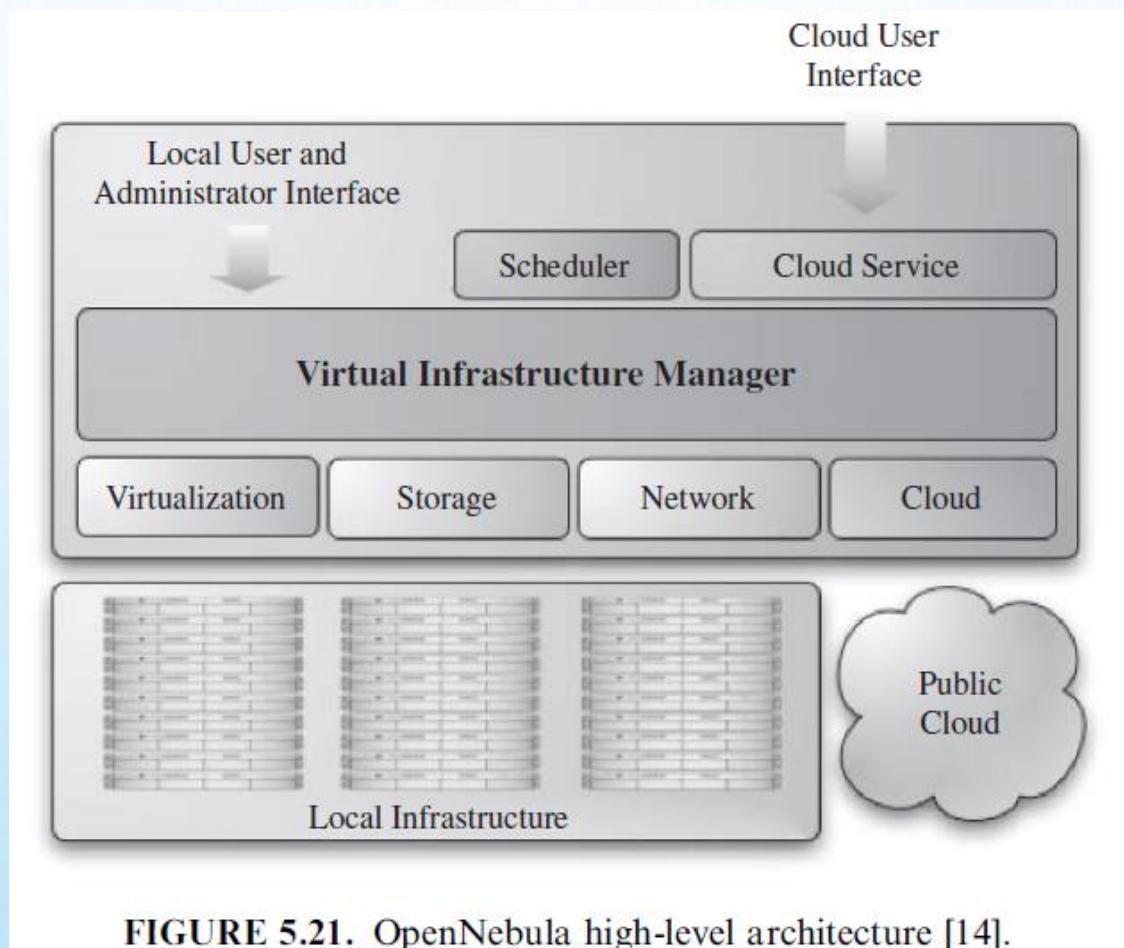


FIGURE 5.21. OpenNebula high-level architecture [14].

NEXT CLASS....

SERVICE-ORIENTED ARCHITECTURE

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L17-L18

- SOA, or service-oriented architecture, defines a way to make software components reusable via service interfaces. These interfaces utilize common communication standards in such a way that they can be rapidly incorporated into new applications without having to perform deep integration each time.

SERVICES AND SERVICE-ORIENTED ARCHITECTURE

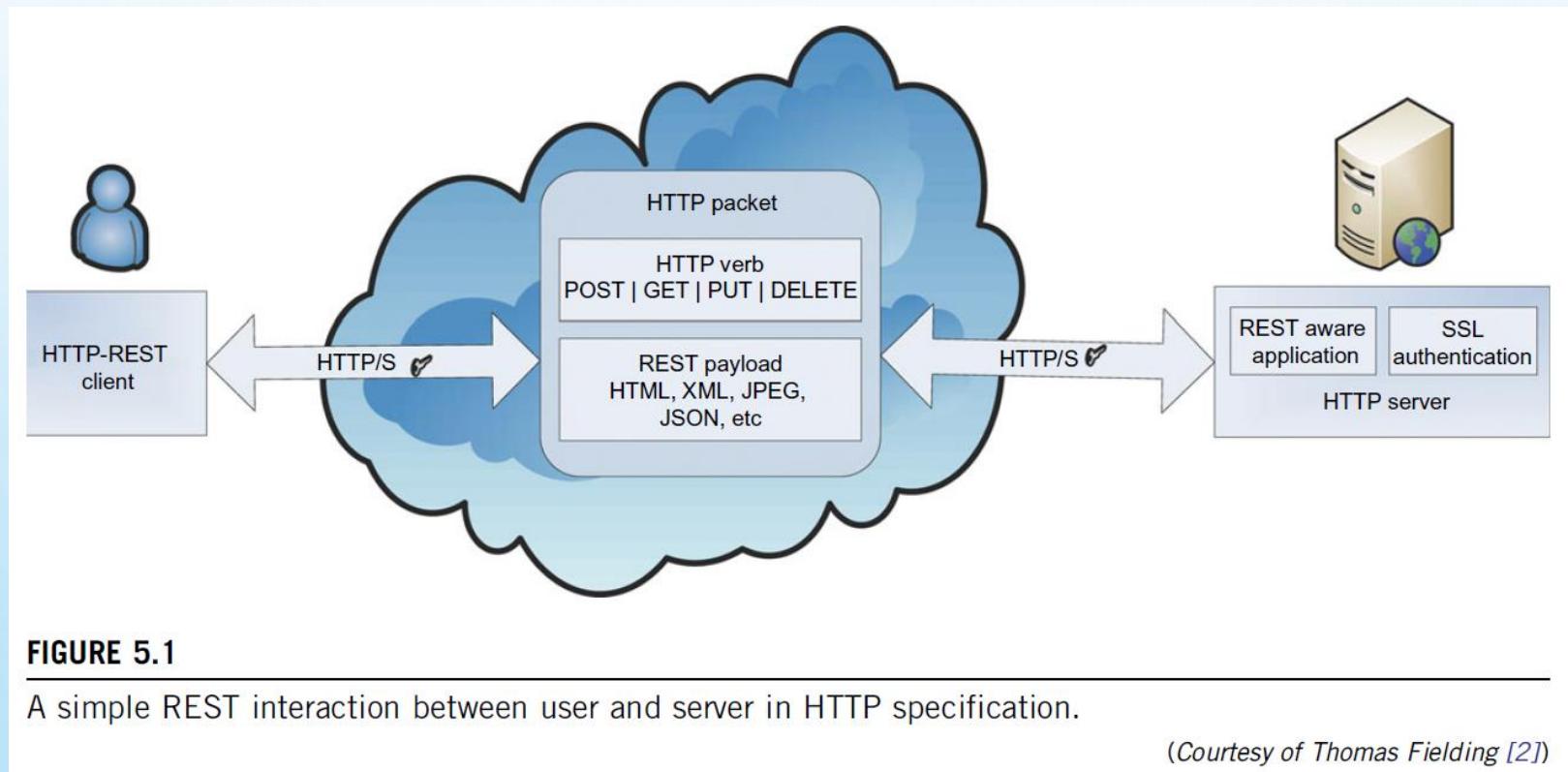
- SOA is about **how to design a software system that makes use of services of new or legacy applications through their published or discoverable interfaces.**
- SOA aims to make **service interoperability extensible and effective.**
- World Wide Web Consortium (W3C) defines SOA as distributed systems with the following properties:
- **Logical view:** The SOA is an abstracted, logical view of actual programs, databases, and business processes.,
- **Message orientation:** The internal structure of providers and requesters includes the implementation language, process structure, and even database structure.
- **Description orientation:** A service is described by machine-executable metadata.
 - Granularity Services.
 - Network orientation Services
 - Platform-neutral Messages

REST AND SYSTEMS OF SYSTEMS

- REST is a software architecture style for distributed systems, particularly distributed hypermedia systems, such as the World Wide Web.
- Popular among: Google, Amazon, Yahoo, Facebook and Twitter (due to simplicity)
- The REST architectural style is based on four principles:
 1. Resource Identification through URIs
 2. Uniform, Constrained Interface
 3. Self-Descriptive Message
 4. Stateless Interactions

RESTFUL WEB SERVICE

RESTful web service is simple, lightweight in nature, and integrates with HTTP.



RESTFUL WEB SERVICE (CONTD..)

- Restlet(Java Framework), implements REST architectural elements such as **resources, representation, connector, and media** type.

Table 5.1 REST Architectural Elements (Adapted from [2])

REST Elements	Elements	Example
Data elements	Resource	The intended conceptual target of a hypertext reference
	Resource identifier	URL
	Representation	HTML document, JPEG image, XML, etc.
	Representation metadata	Media type, last-modified time
	Resource metadata	Source link, alternates, vary
	Control data	If-modified-since, cache-control
Connectors	Client	libwww, libwww-perl
	Server	libwww, Apache API, NSAPI
	Cache	Browser cache, Akamai cache network
	Resolver	Bind (DNS lookup library)
	Tunnel	SSL after HTTP CONNECT
Components	Origin server	Apache httpd, Microsoft IIS
	Gateway	Squid, CGI, Reverse Proxy
	Proxy	CERN Proxy, Netscape Proxy, Gauntlet
	User agent	Netscape Navigator, Lynx, MOMspider

CASE STUDY: RESTFUL WEB SERVICE IN AMAZON S3 INTERFACE

- Amazon S3 (Simple Storage Service) is data storage for Internet applications.
- It provides simple web services to store and retrieves data from anywhere at any time via the web.
- S3 stores metadata in a container called “buckets”

Table 5.2 Sample REST Request-Response for Creating an S3 Bucket

REST Request	REST Response
PUT/[bucket-name] HTTP/1.0 Date: Wed, 15 Mar 2011 14:45:15 GMT Authorization:AWS [aws-access-key-id]: [header-signature] Host: s3.amazonaws.com	HTTP/1.1 200 OK x-amz-id-2: VjzdTviQorQtSjcgLshzCZSzN+7CnewvHA +6sNxR3VRcUPyO5fmSmo8bWnIS52qa x-amz-request-id: 91A8CC60F9FC49E7 Date: Wed, 15 Mar 2010 14:45:20 GMT Location: /[bucket-name] Content-Length: 0 Connection: keep-alive

CASE STUDY: AMAZON S3 (CONTD..)

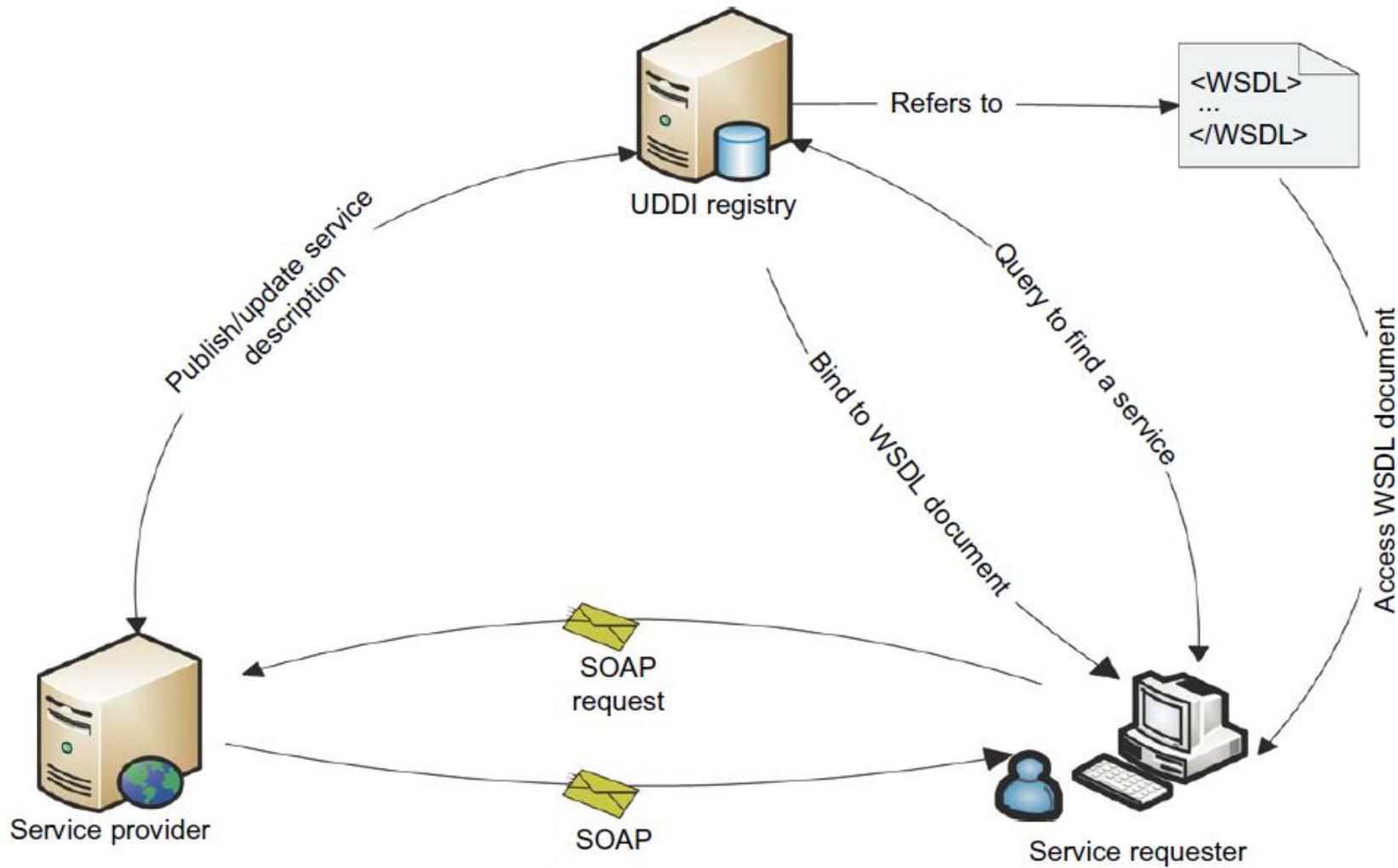
- “buckets” serves several purposes:
 - Organize the Amazon S3 namespace
 - Identify the account responsible for storage
 - Data transfer charges
- Amazon S3 provides three type of resources:
 - a list of user buckets
 - a particular bucket,
 - particular S3 object accessible through:
`https://s3.amazonaws.com/{name-of-bucket}/{name-of-object}.`

SERVICES AND WEB SERVICES

- The term “**web service**” is often referred to a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web.
- **W3C working group** [1] defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network.
- Some of the technologies are:
 1. Simple Object Access Protocol (SOAP)
 2. Web Services Description Language (WSDL)
 3. Universal Description, Discovery, and Integration (UDDI)

SERVICES AND WEB SERVICES

(CONT..)



Simple object access protocol (SOAP)

- Simple object access protocol (SOAP) SOAP provides a standard packaging structure for transmission of XML documents over various internet protocols, such as SMTP, HTTP, and FTP.
- A soap message consists of
- A root element called envelope,
 - A header: a container that can be extended by intermediaries with additional application-level elements such as routing information, authentication,
 - Transaction management,
 - Message parsing instructions
 - Quality of service (qos) configurations,
 - A body element that carries the payload of the message.

WSDL and UDDI

- **Web services description language (WSDL)** WSDL describes the interface, a set of operations supported by a web service in a standard format. It standardizes the representation of Input and output parameters of its operations as well as the service's protocol binding, the way in which the messages will be transferred on the wire. Using WSDL enables disparate clients to automatically understand how to interact with a web service.
- **Universal description, discovery, and integration (UDDI)** UDDI provides a global registry for advertising and discovery of web services, by searching for names, identifiers, categories, or the specification implemented by the web service.

SERVICES AND WEB SERVICES

(CONTD..)

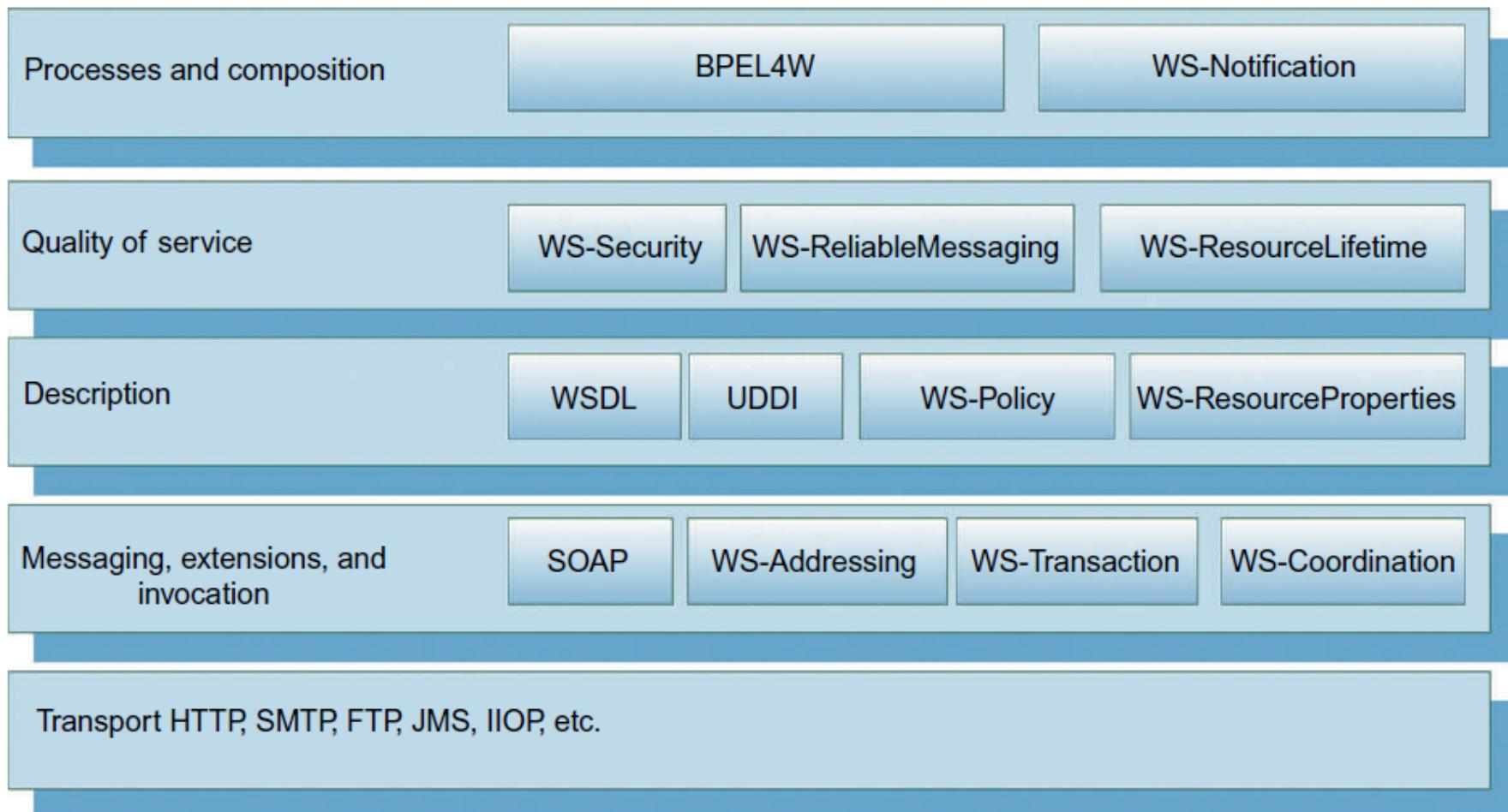


FIGURE 5.3

WS-I protocol stack and its related specifications.

SERVICES AND WEB SERVICES

(CONTD..)

SOAP message consists of an envelope containing a header and a body block

Table 5.3 Sample SOAP Request-Response for Creating an S3 Bucket

SOAP Request	SOAP Response
<pre><soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap- envelope" soap:encodingStyle= "http://www.w3.org/2001/12/soap-encoding"> <soap:Body> <CreateBucket xmlns="http://doc.s3.amazonaws. .com/2010-03-15"> <Bucket>SampleBucket</Bucket> <AWSAccessKeyId> 1B9FVRAYCP1VJEXAMPLE= </AWSAccessKeyId> <Timestamp>2010-03-15T14:40:00.165Z </Timestamp> <Signature>luyz3d3P0aTou39dzbqaEXAMPLE =</Signature> </CreateBucket> </soap:Body> </soap:Envelope></pre>	<pre><soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap- envelope" soap:encodingStyle= "http://www.w3.org/2001/12/soap-encoding"> <soap:Body> <CreateBucket xmlns="http://doc.s3.amazonaws. .com/2010-03-15"> <Bucket>SampleBucket</Bucket> <AWSAccessKeyId>1B9FVRAYCP1VJEXAMPLE= </AWSAccessKeyId> <Timestamp>2010-03-15T14:40:00.165Z </Timestamp> <Signature>luyz3d3P0aTou39dzbqaEXAMPLE =</Signature> </CreateBucket> </soap:Body> </soap:Envelope></pre>

SERVICES AND WEB SERVICES

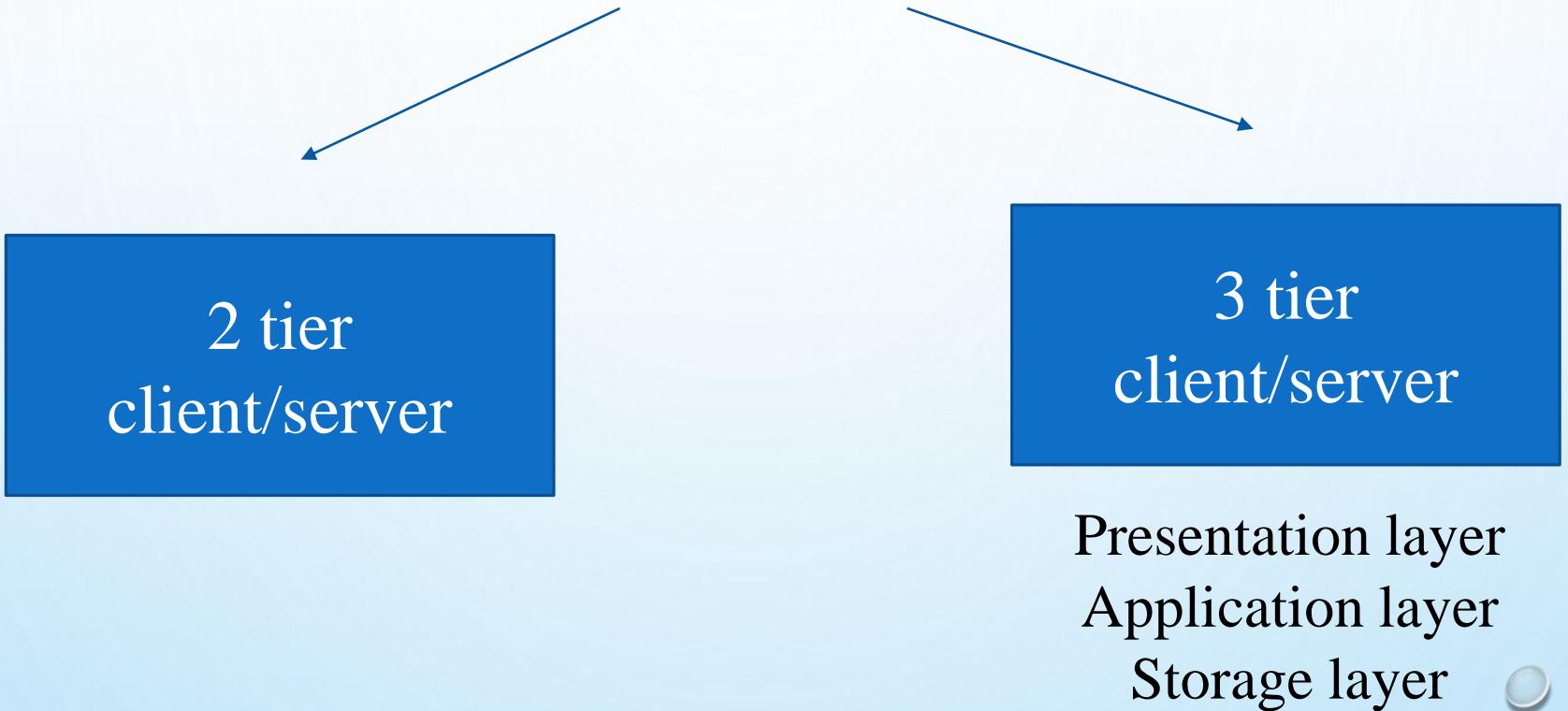
(CONTD..)

WS-* Core SOAP Header Standards

Table 5.4 The 10 Areas Covered by the Core WS-* Specifications

WS-* Specification Area	Examples
1. Core Service Model	XML, WSDL, SOAP
2. Service Internet	WS-Addressing, WS-MessageDelivery, Reliable WSRM, Efficient MOTM
3. Notification	WS-Notification, WS-Eventing (Publish-Subscribe)
4. Workflow and Transactions	BPEL, WS-Choreography, WS-Coordination
5. Security	WS-Security, WS-Trust, WS-Federation, SAML, WS-SecureConversation
6. Service Discovery	UDDI, WS-Discovery
7. System Metadata and State	WSRF, WS-MetadataExchange, WS-Context
8. Management	WSDM, WS-Management, WS-Transfer
9. Policy and Agreements	WS-Policy, WS-Agreement
10. Portals and User Interfaces	WSRP (Remote Portlets)

ENTERPRISE MULTITIER ARCHITECTURE



3 TIER SYSTEM ARCHITECTURE

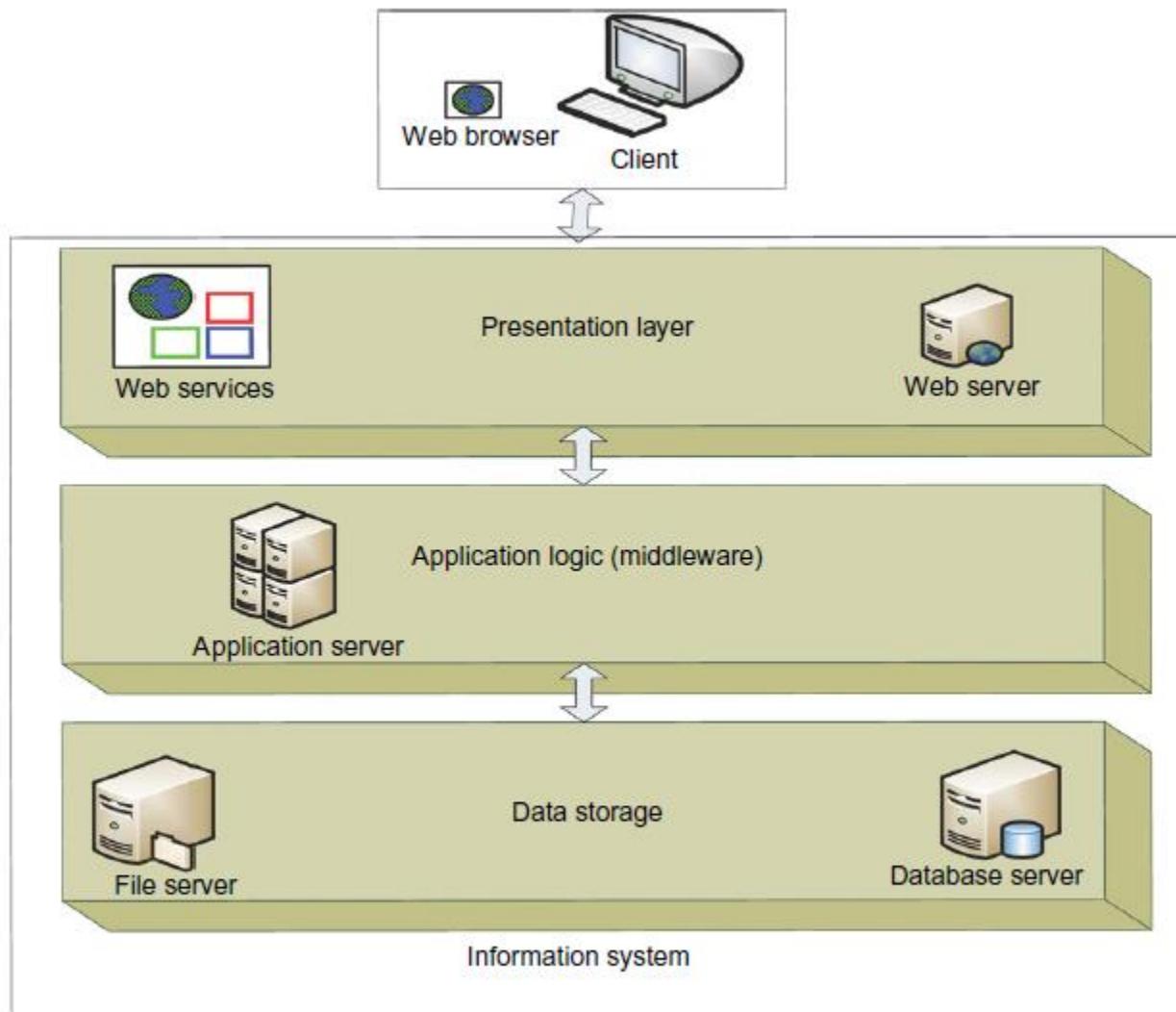


FIGURE 5.4

Three-tier system architecture.

GRID SERVICES AND OGSA

- OGSA is a service-oriented architecture that aims to define a common, standard, and open architecture for grid-based applications.
- “Open” refers to both the process to develop standards and the standards themselves.

Note:

Web Services Description Language
Grid Service Handle (GSH)
Grid Service Reference (GSR),
Open Grid Services Infrastructure (OGSI)
Web Services Resource Framework (WSRF)

GRID SERVICES AND OGSA (CONTD..)

OGSA is intended to:

- Facilitate use and management of resources across distributed, heterogeneous environments.
- Deliver seamless QoS.
- Publishes the interfaces in order to provide interoperability of diverse resources
- Exploit industry-standard integration technologies
- Develop standards that achieve interoperability
- Integrate, virtualize, and manage services and resources in a distributed, heterogeneous environment
- Deliver functionality as loosely coupled, interacting services aligned with industry-accepted web service standards

GRID SERVICES AND OGSA (CONTD..)

OGSA services fall into seven broad areas:

- Infrastructure Services
- Execution Management Services
- Data Management Services
- Resource Management Services
- Security Services
- Information Services
- Self-Management Services

GRID SERVICES AND OGSA (CONTD..)

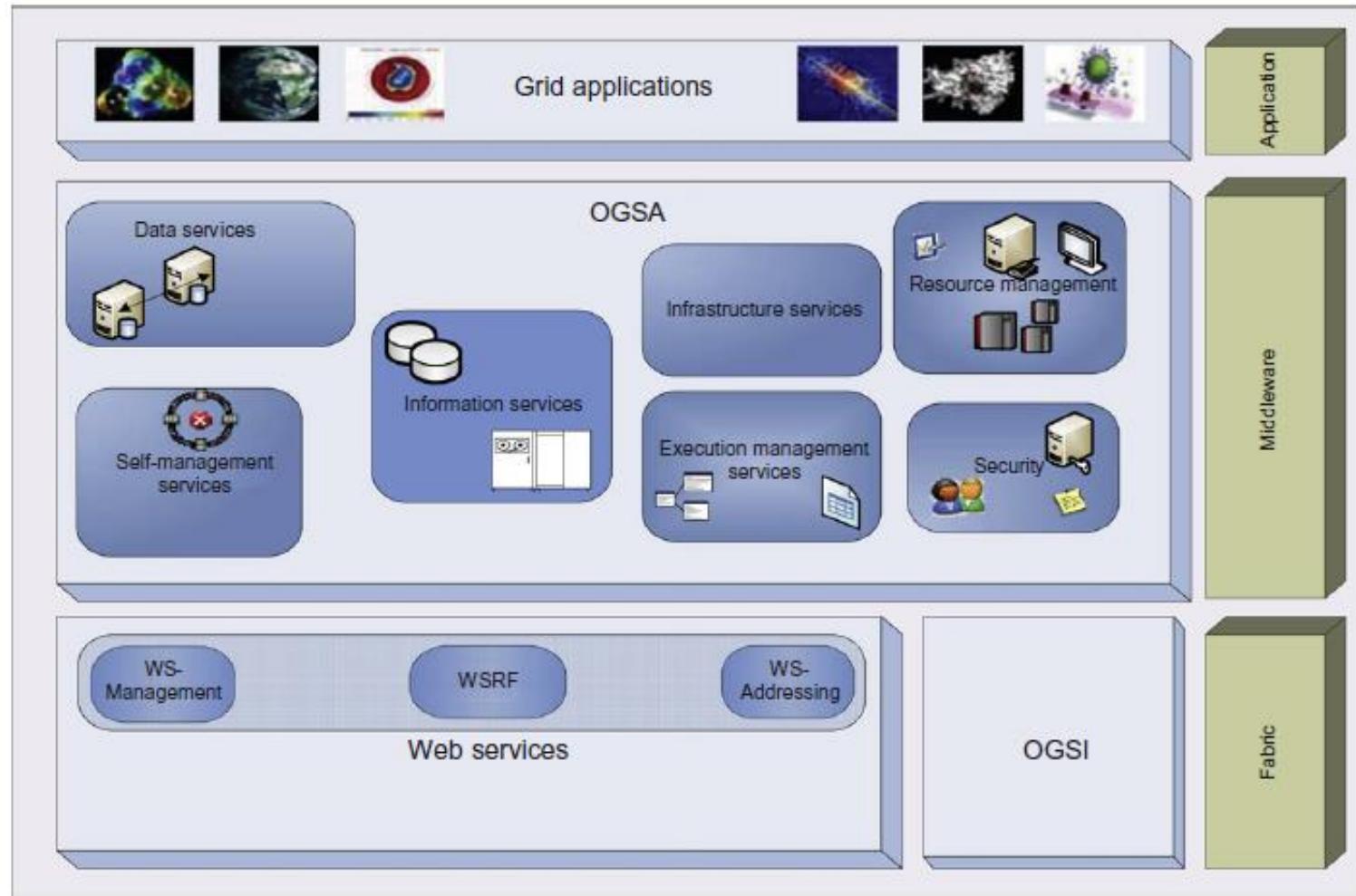


FIGURE 5.5

Dept.

The OGSA architecture.

(Courtesy of Foster, et al. [24], <http://www.ogf.org/documents/GFD.80.pdf>.)

NEXT CLASS....

ABC

SERVICE-ORIENTED ARCHITECTURE

(CONTD..)

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L19

MESSAGE-ORIENTED MIDDLEWARE

- MESSAGE-ORIENTED MIDDLEWARE (MOM) IS SOFTWARE OR HARDWARE INFRASTRUCTURE SUPPORTING SENDING AND RECEIVING MESSAGES BETWEEN DISTRIBUTED SYSTEMS.
- MOM ALLOWS APPLICATION MODULES TO BE DISTRIBUTED OVER HETEROGENEOUS PLATFORMS AND REDUCES THE COMPLEXITY OF DEVELOPING APPLICATIONS THAT SPAN MULTIPLE OPERATING SYSTEMS AND NETWORK PROTOCOLS.

MESSAGE-ORIENTED MIDDLEWARE

The study of MOM includes

- Enterprise Bus
- Publisher-Subscriber Model and Notification
- Queuing and Messaging Systems
- Cloud or Grid Middleware Applications

MESSAGE-ORIENTED MIDDLEWARE

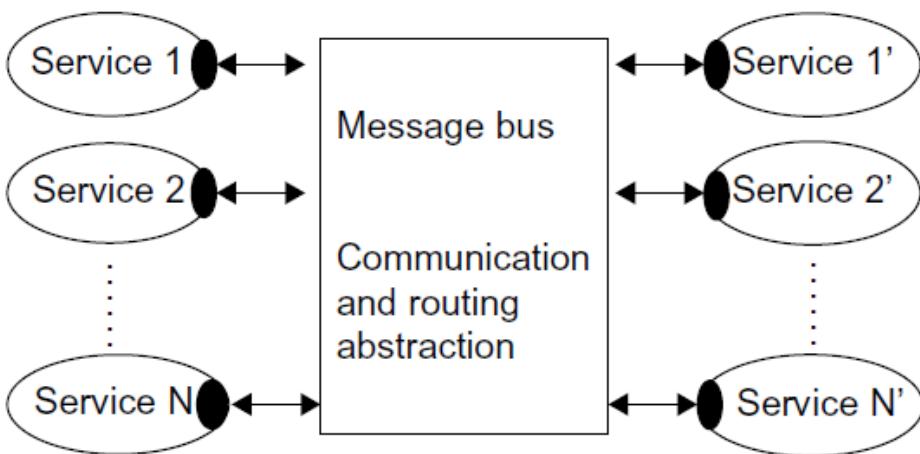
- **Message-oriented middleware (MOM)** is software or hardware infrastructure supporting **sending and receiving messages** between distributed systems.
- MOM allows application modules to be **distributed over heterogeneous platforms** and reduces the complexity of developing applications that span multiple operating systems and network protocols.
- The middleware **creates a distributed communications layer** that insulates the application developer from the details of the various operating systems and network interfaces.

https://en.wikipedia.org/wiki/Message-oriented_middleware

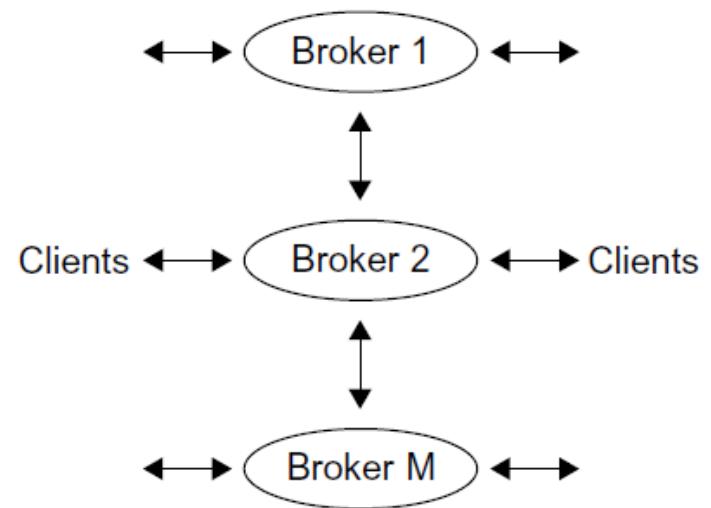
ENTERPRISE BUS

- Also called as **Enterprise Service Bus (ESB)**
- ESB is a centralized software component that performs integrations to backend systems like translations of data models, deep connectivity, routing, and requests.
- Developers can use a single protocol to ‘talk’ to the ESB and issue commands that direct interactions between services and leave it to the ESB to translate the commands, route the messages, and transform the data as required to get the commands executed.
- This enables developers to spend dramatically less time integrating and much more time configuring and improving their applications.
- Ultimately the challenges of maintaining, updating, and scaling a centralized ESB proved so overwhelming and expensive that the ESB often delayed the very productivity gains

ENTERPRISE BUS



(a) Implemented between services



(b) As a network of distributed brokers

FIGURE 5.6

Two message bus implementations between services or using a broker network.

ENTERPRISE BUS

- One may wish to introduce a wrapper so that services expecting messages in different styles (say, SOAP, REST, or Java RMI) can communicate with each other.
- One does not open a channel between source and destination, but rather injects a message into the bus with enough information to allow it to be delivered correctly. This injection is performed by code loaded into each service and represented by the filled ovals as client interfaces in Figure 5.6(a).
- The message bus is shown as linking services in the figure, but it can work with any software or hardware entity sending and receiving messages.

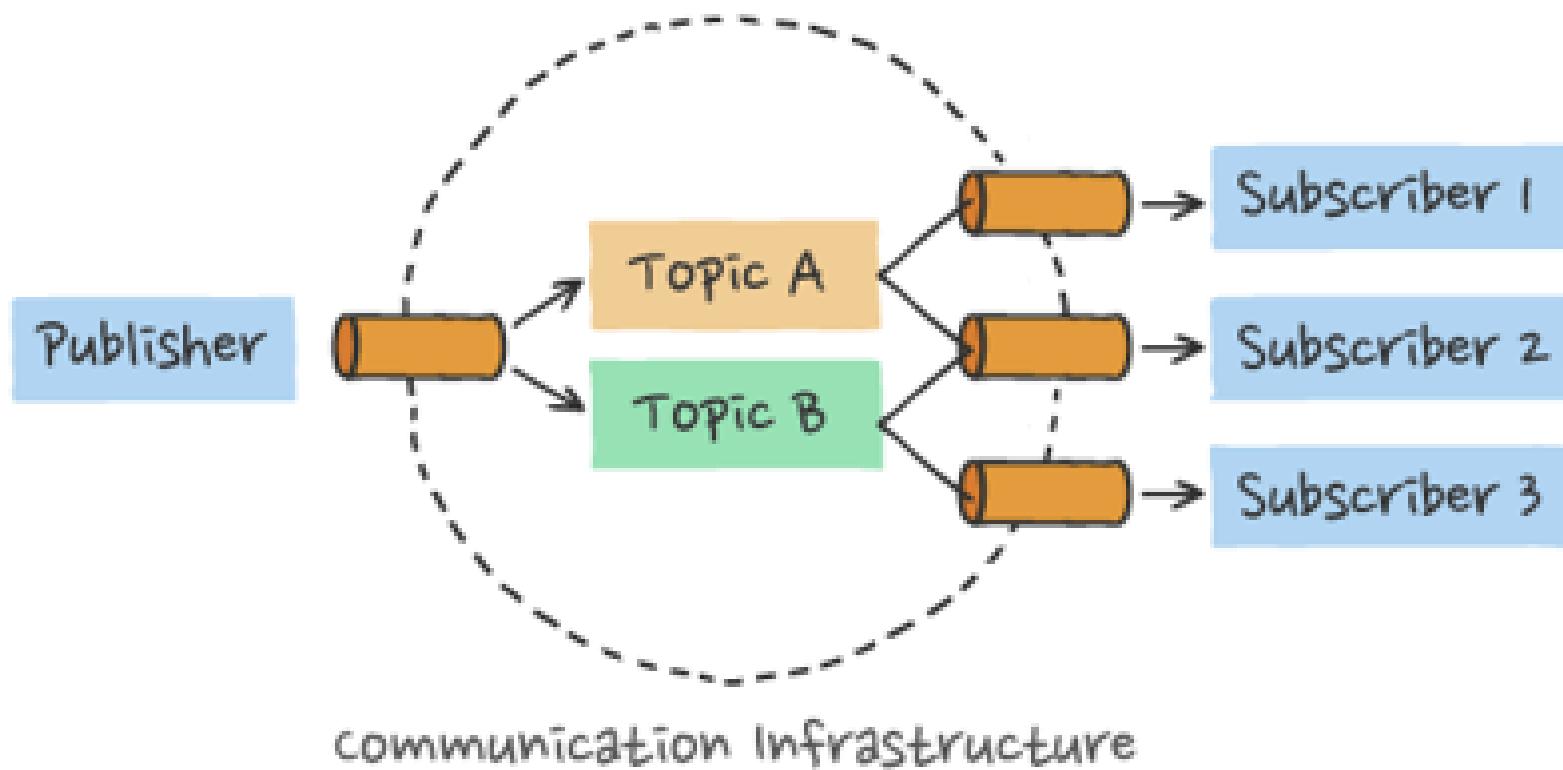
ENTERPRISE BUS

- These buses can be implemented internally to an application, or in a distributed fashion.
- i.e. the message bus typically can be implemented as a set of “brokers”.
- One implements brokers as managers of queues, and software in this area often has MQ or “Message Queue” in its description.

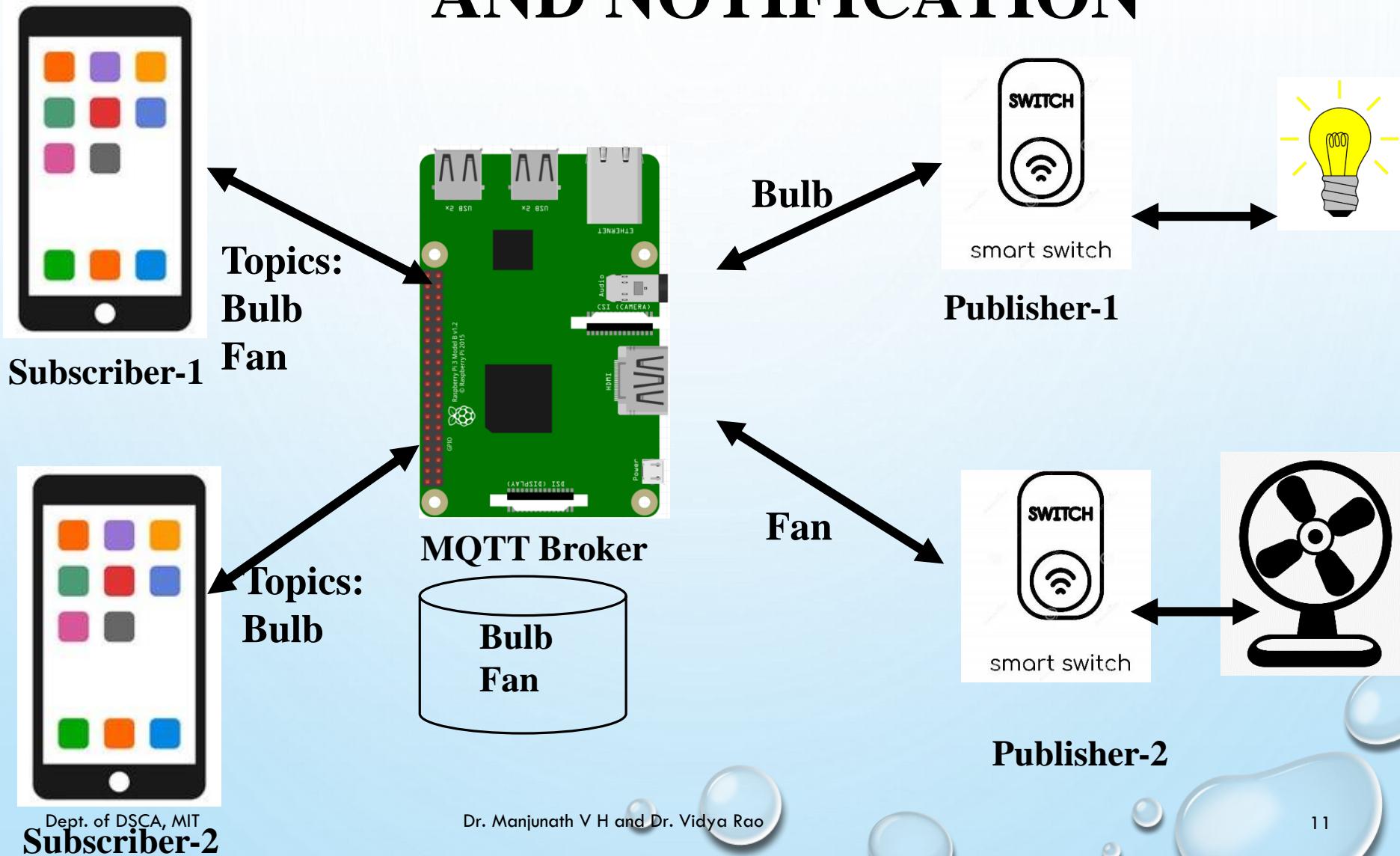
PUBLISHER-SUBSCRIBER MODEL AND NOTIFICATION

- A model for linking source and destination for a message bus.
- Here the producer of the message (publisher) labels the message in some fashion; often this is done by associating one or more topic names from a (controlled) vocabulary.
- Then the receivers of the message (subscriber) will specify the topics for which they wish to receive associated messages.
- The use of topic or content-based message selection is termed message filtering.

PUBLISHER-SUBSCRIBER MODEL AND NOTIFICATION



PUBLISHER-SUBSCRIBER MODEL AND NOTIFICATION



QUEUING AND MESSAGING SYSTEMS

- **Java Message Service (JMS)** - specifies a set of interfaces outlining the communication semantics in pub/sub and queuing systems.
- **Advanced Message Queuing Protocol (AMQP)** - specifies the set of wire formats for communications; unlike APIs, wire formats are cross-platform.
- **MuleMQ**, - is a messaging framework underlying the ESB.
- The focus of Mule is to simplify the integration of existing systems developed using JMS, Web Services, SOAP, JDBC, and traditional HTTP.
- **Protocols supported within Mule** include POP, IMAP, FTP, RMI, SOAP, SSL, and SMTP

CLOUD OR GRID MIDDLEWARE APPLICATIONS

Environmental Monitoring and Internet Conference Using NaradaBrokering:

The GOAT project at Clemson University is part of the Program of Integrated Study for Coastal Environmental Sustainability (PISCES), which addresses environmental sustainability issues that can accompany coastal development. The current study incorporates groundwater monitoring, surface water quality and quantity monitoring, weather, and a variety of ecological measurements. The project utilizes the publishsubscribe messaging system, NaradaBrokering, to provide a flexible and reliable layer to move observation data from a wide variety of sensor sources to users that have diverse data management and processing requirements. NaradaBrokering can display environmental sensors.

CLOUD OR GRID MIDDLEWARE APPLICATIONS

Environmental Monitoring and Internet Conference Using NaradaBrokering: (contd..)

The commercial Internet Meeting software Anabas (www.anabas.com) incorporates support for sharing applications besides incorporating support for shared whiteboards and chat tools. Anabas uses NaradaBrokering for its content dissemination and messaging requirements. On a daily basis, Anabas supports several online meetings in the United States, Hong Kong, and mainland China. Note NaradaBrokering supports audio-video conferencing (using UDP) as well as other collaborative applications using TCP. Dynamic screen display published to NaradaBrokering can be displayed on collaborating clients.

CLOUD OR GRID MIDDLEWARE APPLICATIONS

QuakeSim Project for Earthquake Science using NaradaBrokering:

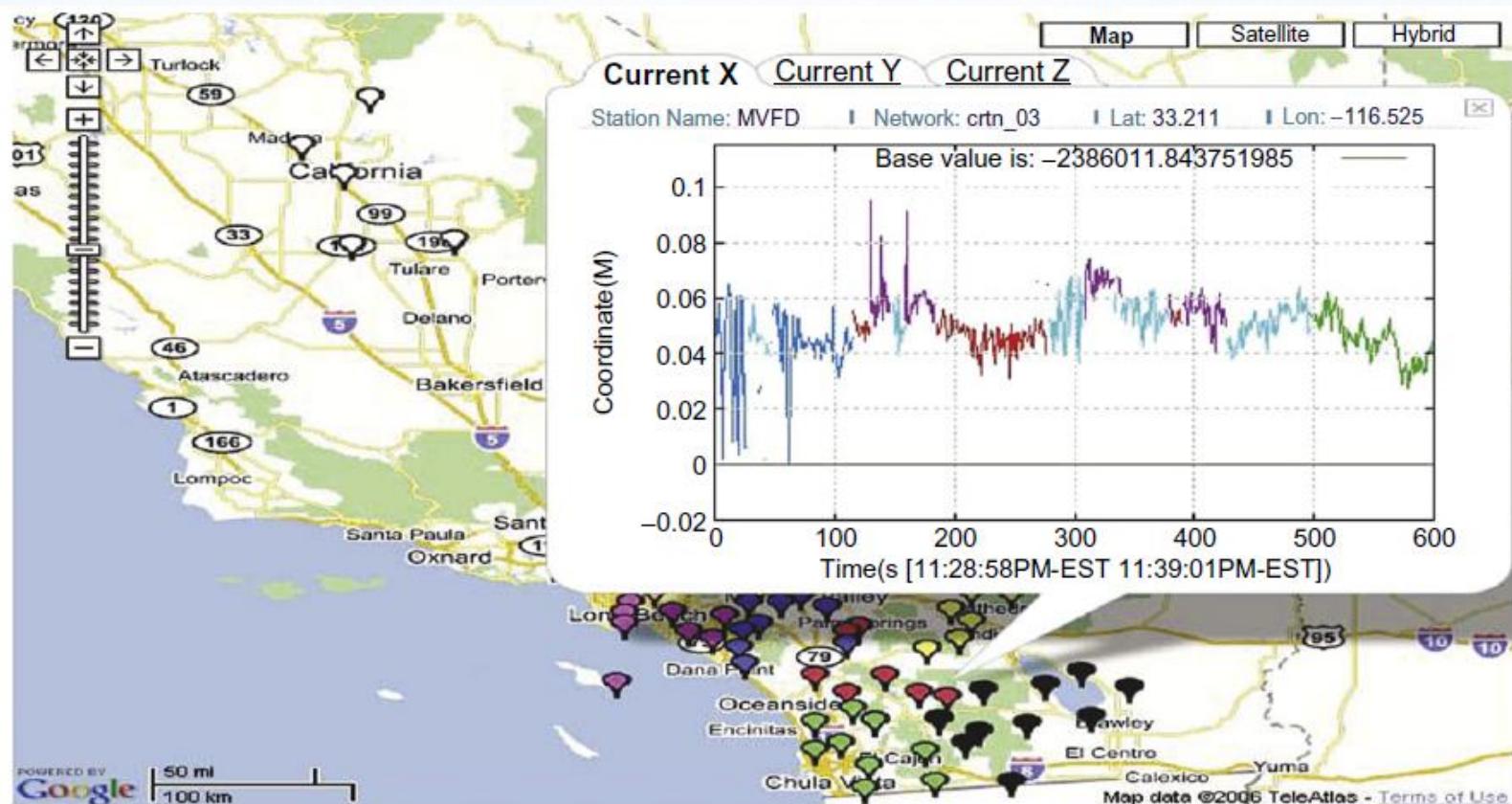


FIGURE 5.7

Display of GPS sensors managed by NaradaBrokering in Southern California; the map displays the time series produced by one of the GPS stations. (<http://quakesim.jpl.nasa.gov/>).

PORtALS AND SCIENCE GATEWAYS

- Science Gateway Exemplars
- HUBzero Platform for Scientific Collaboration
- Open Gateway Computing Environments (OGCE)

PORtALS AND SCIENCE GATEWAYS

Gateways provide user-centric environments for interacting with remote computational resources through user interfaces that are typically (but not exclusively) built with web technologies.

TurnKey Solution

HUBzero

**Provide end-to-end
Solution**

Tool Box Solution

**Open Gateway
Computing
Environment (OGCE)**

**Provides tools to solve a
specific problem**

PORTALS AND SCIENCE GATEWAYS

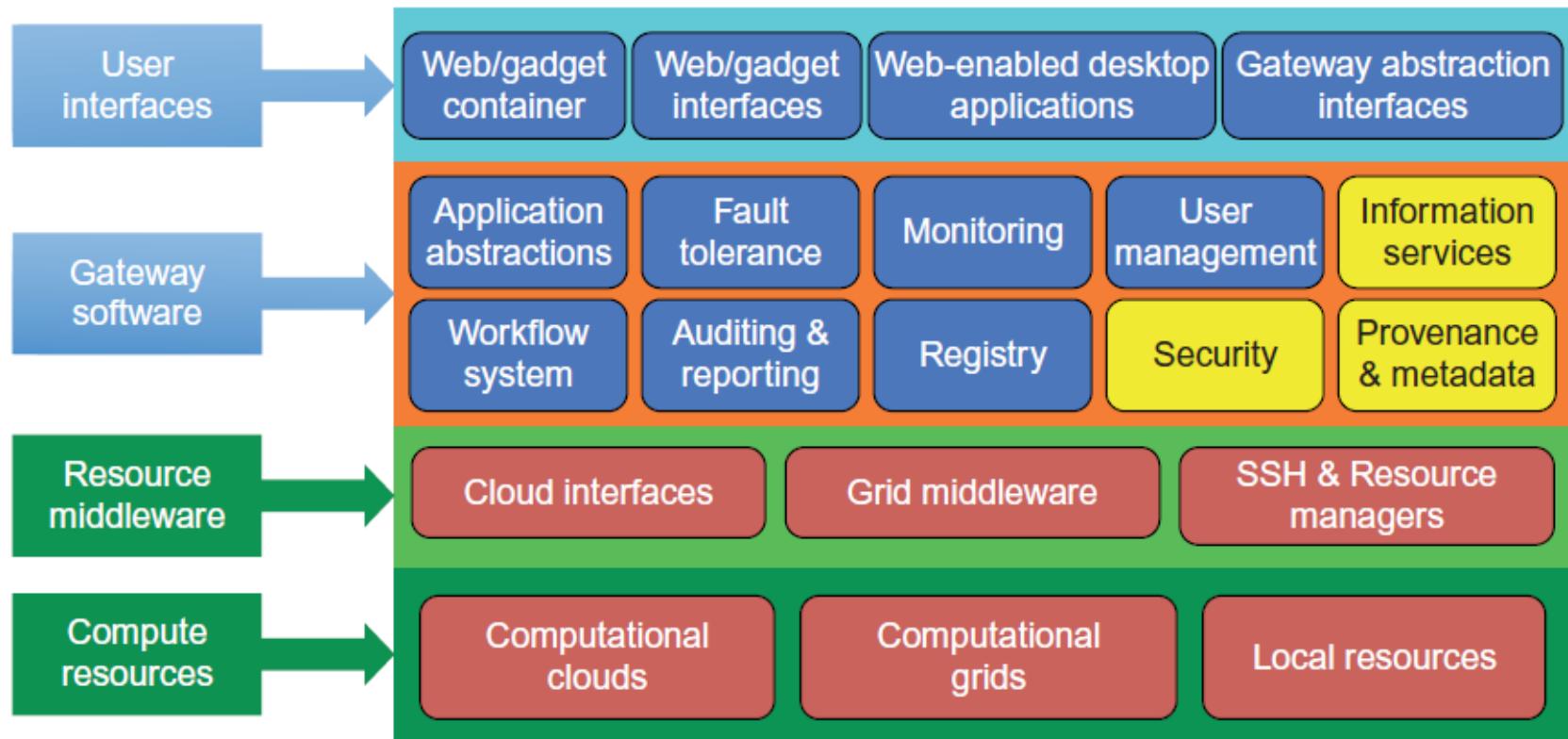


FIGURE 5.8

A gateway component software stack for scientific applications.

NEXT CLASS....

ABC

SERVICE-ORIENTED ARCHITECTURE

(CONTD..)

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L20-L21

PURTALS AND SCIENCE GATEWAYS

- Science Gateway Exemplars
- HUBzero Platform for Scientific Collaboration
- Open Gateway Computing Environments (OGCE)

PORtALS AND SCIENCE GATEWAYS

Gateways provide user-centric environments for interacting with remote computational resources through user interfaces that are typically (but not exclusively) built with web technologies.

TurnKey Solution

HUBzero

Provide end-to-end Solution

Tool Box Solution

Open Gateway Computing Environment (OGCE)

Provides tools to solve a specific problem

PORTALS AND SCIENCE GATEWAYS

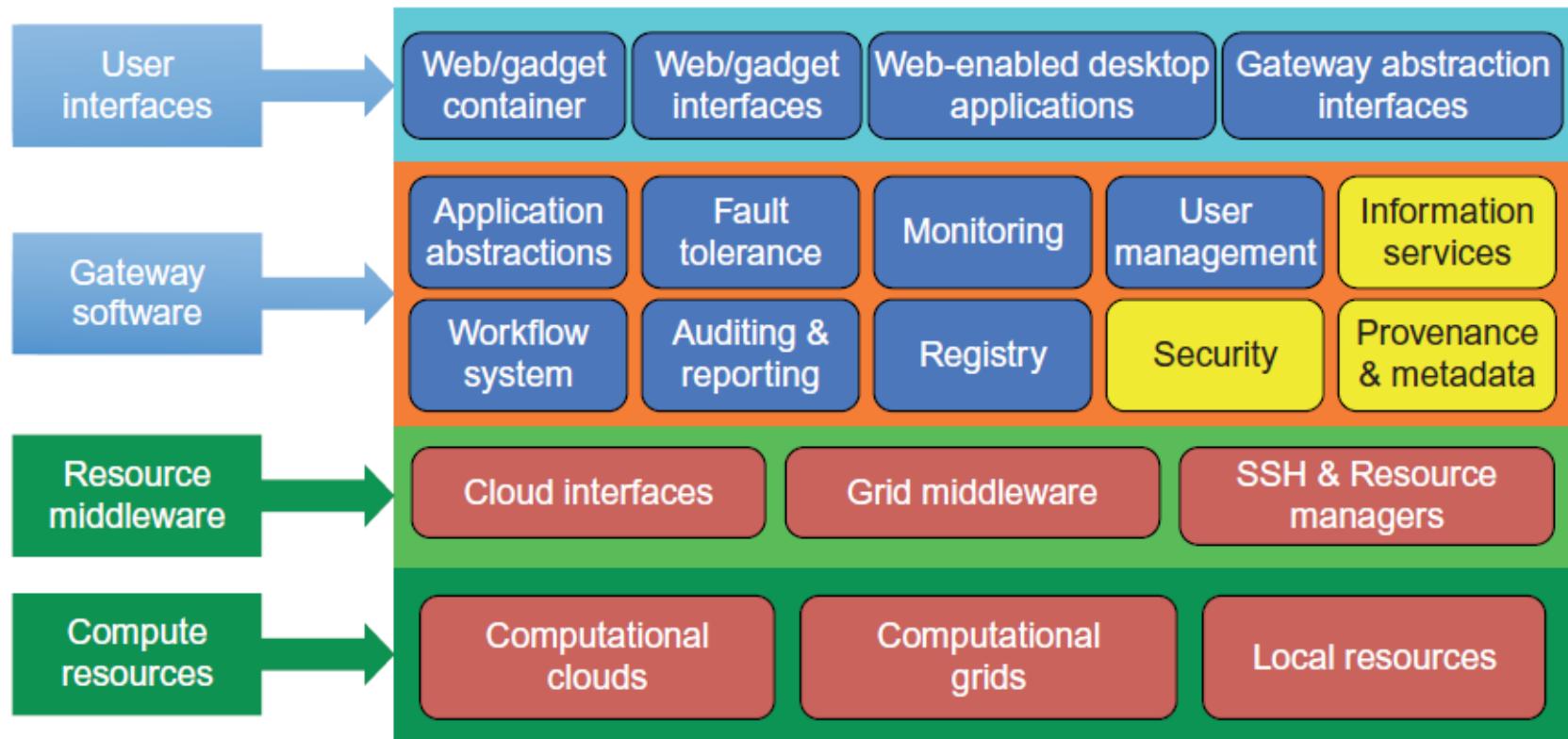


FIGURE 5.8

A gateway component software stack for scientific applications.

HUBzero PLATFORM FOR SCIENTIFIC COLLABORATION

- HUBzero is an open source software platform used to create web sites or “hubs” for scientific collaboration, research, and education.
- Similar to:
 - YouTube.com, HUBzero allows people to **upload content and “publish”** to a wide audience, but instead of being restricted to short video clips, it handles many different kinds of scientific content.
 - Google Groups, HUBzero lets **people work together** in a private space where they can share documents and send messages to one another.
 - SourceForge.net, HUBzero allows researchers to work **collaboratively** on the source code of their simulation programs and share those programs with the community.

HUBzero PLATFORM FOR SCIENTIFIC COLLABORATION

HUBZero Architecture

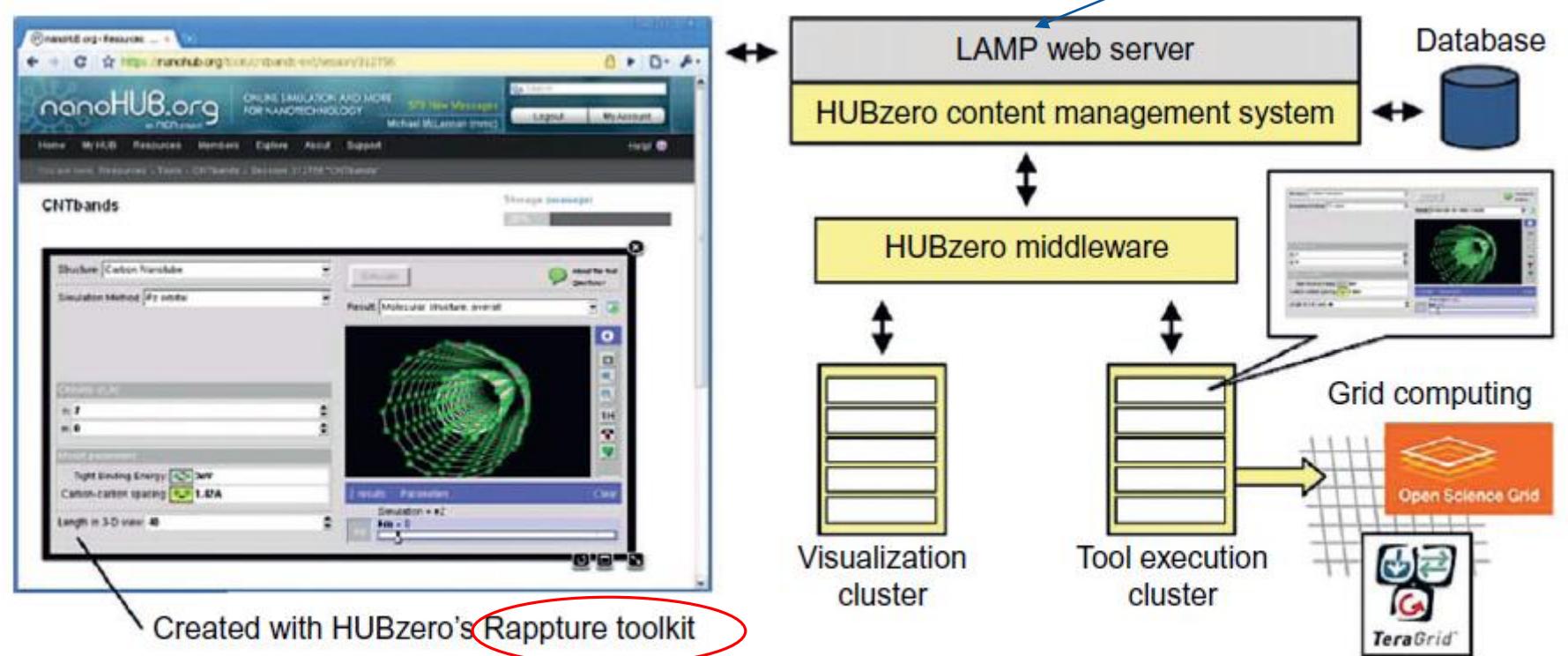


FIGURE 5.9

The HUBzero architecture and its major functional components.

HUBzero PLATFORM FOR SCIENTIFIC COLLABORATION

Rappture toolkit generates GUI containing:

- Operational Features
- Rating and citations
- Content tagging
- User support area
- Wikis and blogs
- Usage metric
- Future design

OPEN GATEWAY COMPUTING ENVIRONMENTS (OGCE)

- Gateways are tools that enable interactive, web-based science, education, and collaboration.
- OGCE open source gateway software that is used in several collaborating gateways.
- OGCE component tools include the following:
 - **OGCE Gadget container**, a Google tool for integrating user interface components
 - **XRegistry**, a registry service for storing information about other online services and workflows
 - **XBaya**, a workflow composer and enactment engine
 - **GFAC**, a factory service that can be used to wrap command-line-driven science applications and make them into robust, network-accessible services
 - **OGCE Messaging Service supports** events and notifications across multiple cooperating services

OPEN GATEWAY COMPUTING ENVIRONMENTS (OGCE)

- OGCE's strategy is based on the toolkit model.
- This strategy has been shaped by the TeraGrid science gateway program and its wide variety of gateways.
- There are obviously many frameworks, programming languages, and tools for building web-based gateways and providing advanced capabilities.
- The OGCE tools focus on scientific application and workflow management and defer issues such as data and metadata management to other projects

OPEN GATEWAY COMPUTING ENVIRONMENTS (OGCE)

Workflows:

The OGCE scientific workflow system provides a programming model that allows the scientist to program experiments using application web services.

Scientific Application Management:

Wrapping scientific applications as remotely accessible services

Gadget Container:

Workflows, registries, and service wrappers have both client- and server-side pieces. The OGCE builds most of its default user interfaces as gadgets.

Packaging:

Applications are packed with required packages that ensure the application is running smooth.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

- In SOA, business services need to discover appropriate services to use and to integrate with.
- A registry requires a set of data structure specifications for the metadata to be stored in the registry.
- Registry operations: **Create, Read, Update, and Delete (CRUD)** for storing, deleting, and querying the data to store metadata for ownership, containment, and categorization of services.
- Registries usually contain three categories of information:
 - **White pages** contain name and general contact information about an entity.
 - **Yellow pages** contain classification information about the types and location of the services the entry offers.
 - **Green pages** contain information about the details of how to invoke the offered services

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

UDDI and Service Registries

- UDDI : Universal Description, Discovery And Integration
- UDDI specifications define a way to describe, publish, and discover information about web services by creating a platform-independent, open framework.

PUBLIC

PRIVATE

Public registry is a logically centralized distributed service

A private registry is only accessible within a single organization or is shared by a group

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Data in a UDDI registry is organized as instance types:

- ***businessEntity*** Describes an organization or a business that provides the web services, including the company name, contact information, industry/product/geographic classification,
- ***businessService*** Describes a collection of related instances of web services offered by an organization, such as the name of the service, a description, and so forth
- ***bindingTemplate*** Describes the technical information necessary to use a particular web service, such as the URL address to access the web service instance and references to its description
- ***tModel*** A generic container for specification of WSDL documents in general web services
- ***publisherAssertion*** Defines a relationship between two or more businessEntity elements

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

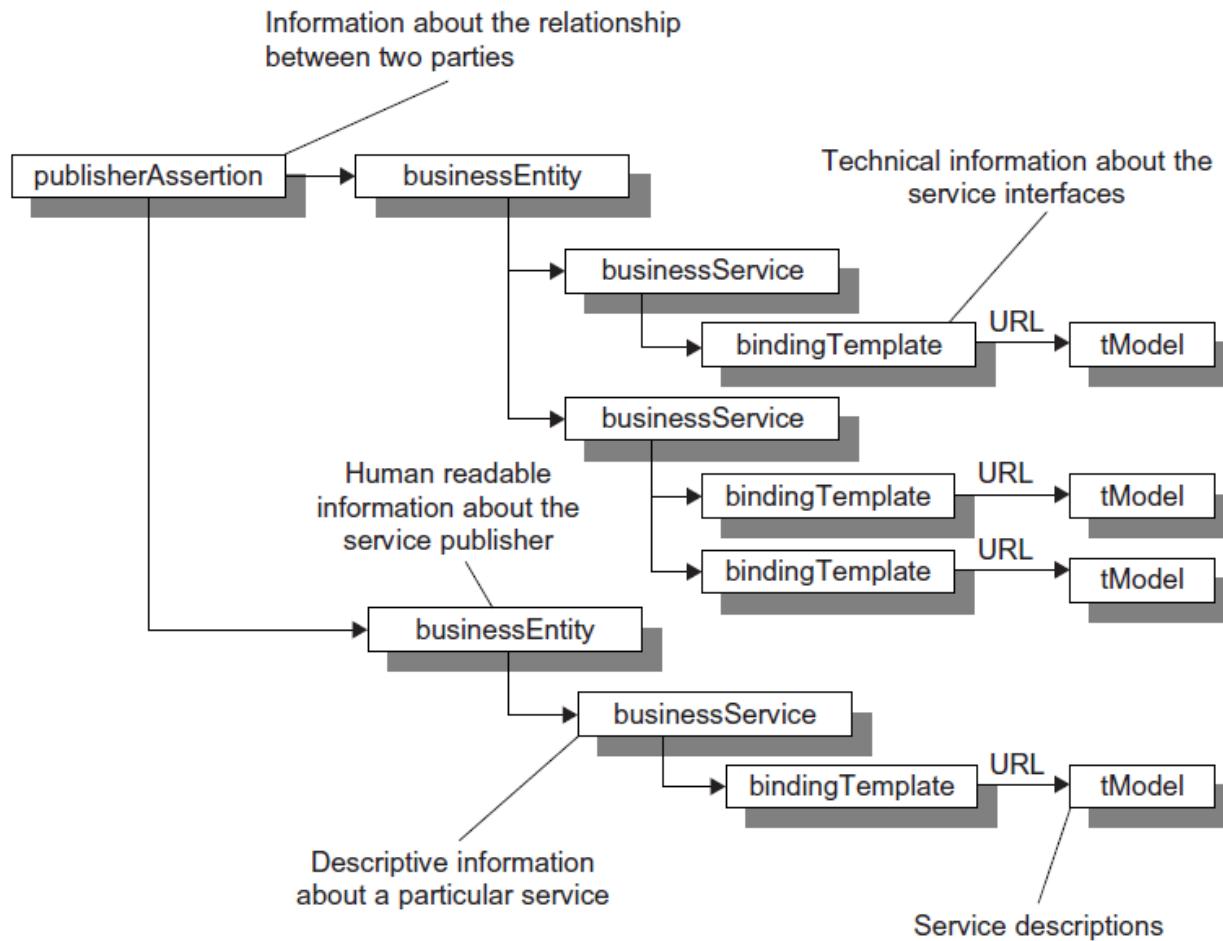


FIGURE 5.10

UDDI entities and their relationship.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

UDDI provides a set of APIs:

- **UDDI Inquiry API** - to find the set of registry entries such as business, service, binding, etc.
- **UDDI Publishers API** - enables add, modify, and delete entries.
- **UDDI Security API** Allows users to get and discard authentication tokens
- **UDDI Custody and Ownership Transfer API** Enables registries to transfer
- **UDDI Subscription API** Enables monitoring of changes in a registry by subscribing to track new, modified, and
- **UDDI Replication API** Supports replication of information between registries so that different registries can be kept synchronized

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Databases and Publish-Subscribe

- Publish-subscribe is a design pattern that enables asynchronous interaction among distributed applications.
- Many high-level applications regularly query the database in order to adapt their execution according to this information.
- Such periodic data polling is **not only inefficient and unscalable but also resource-demanding on both sides.**
- The publish-subscribe mechanism, already largely adopted in the implementation of today's applications, solves this issue.
- In a publish-subscribe interaction, event subscribers register to particular event types and receive notifications from the event publishers when they generate such events.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Databases and Publish-Subscribe(contd..)

- Publish-subscribe systems are classified as either as:

Topic-based

Content-based

- In topic-based systems, publishers generate events with respect to a topic or subject.
- Subscribers then specify their interest in a particular topic and receive all events published on that topic.
- Defining events in terms of topic names only is inflexible and requires subscribers to filter events belonging to general topics.
- Content-based systems solve this problem by introducing a subscription scheme based on the contents of events.
- They give users the ability to express their interest by specifying predicates over the values of a number of well-defined attributes.
- The matching of publications (events) to subscriptions (interest) is done based on the content.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Databases and Publish-Subscribe (contd..)

- Database systems provide many features that a messaging-based architecture can exploit, such as reliable storage, transactions, and triggers.
- It integrates with publish-subscribe capabilities in the database account for information-sharing systems that are simpler to deploy and maintain.
- PostgreSQL open source database system to include publish-subscribe middleware functionality.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Databases and Publish-Subscribe (contd..)

PostgreSQL:

- Based on the integration of active databases and the publish-subscribe communication model to form a global event-based system.
- Databases define and advertise change events, and clients subscribe to events of interest, and can refine their subscriptions through content-based filter expressions.
- This allows a database system in the local domain to function as an event broker (broker), reliably routing events among publishers, subscribers, and other brokers.
- This integration simplifies information management by grouping security, configuration (e.g., type schema), and recovery tasks for database and pub/sub operations under the same interface.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Databases and Publish-Subscribe (contd..)

Oracle Publisher-Subscriber

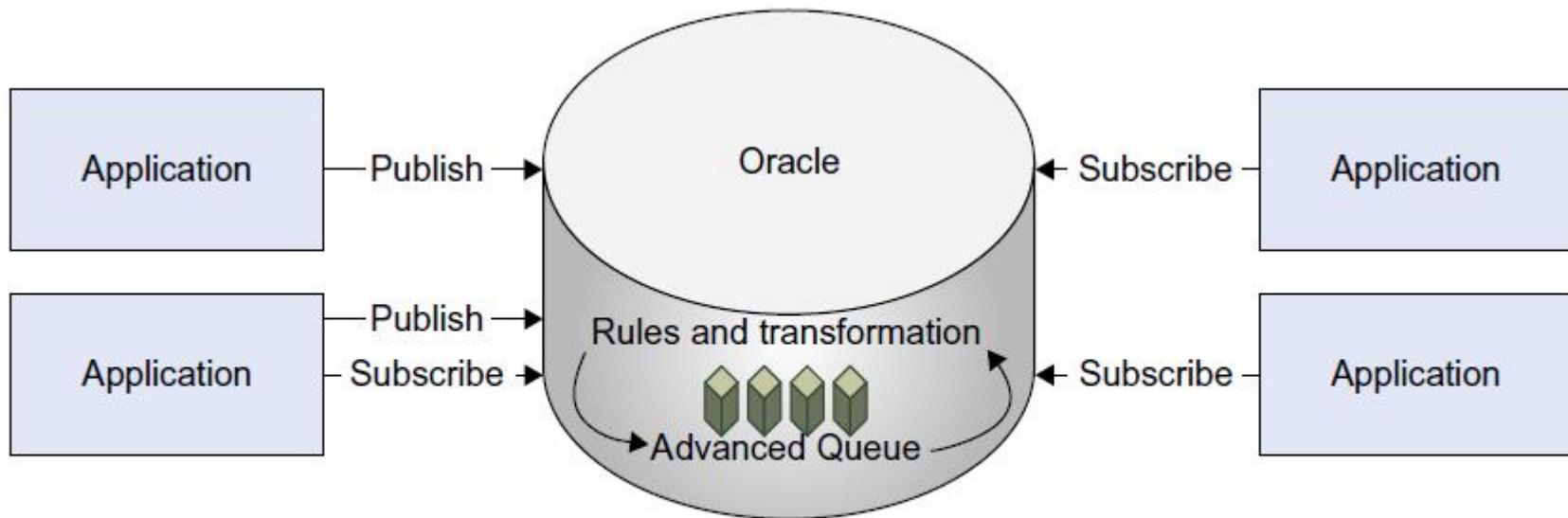


FIGURE 5.11

Oracle publish-subscribe model.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Metadata Catalogs

- Metadata is information about data.
- Metadata is important, since it adds context to the data, in order to identify, locate, and interpret it.
- Play a vital role in distributed heterogeneous environments such as grids by providing users and applications the means to discover and locate the desired data and services among lots of sites in such environments.
- Key metadata on the grid includes the name and location of the data resource, structure of the data held within the data resource, data item names and descriptions, and user information.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Metadata Catalogs

- Metadata services groups
 - Metadata Catalog Service (MCAT)
 - Replica Location Service (RLS)
 - AMGA (the ARDA Metadata for Grid Applications)

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Semantic Web and Grid

- The grid strives to share and access metadata in order to automate information discovery and integration of services and resources in a dynamic, large-scale distributed environment.
- Meanwhile, the Semantic web is all about automation discovery and integration.
- The Semantic web aims to provide an environment where software agents are able to dynamically discover, interrogate, and interoperate resources and perform sophisticated tasks on behalf of humans, which is not far from the ambition of grid computing.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Semantic Web and Grid (contd..)

- **Resource Description Framework (RDF)**, agreed ontologies expressed in a common language.
- RDF is the first language developed for the Semantic web, using XML to represent information (including metadata) about resources on the web.
- RDF uses web identifiers (URIs), and describes resources in terms of simple properties and property values.
- Semantic web services describe and annotate various aspects of a web service using explicit, machine-understandable semantics, that facilitating the discovery, execution, monitoring, and aggregation of resources.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Semantic Web and Grid (contd..)

- **Web Ontology Language (OWL):** enables web services to be described semantically and their descriptions to be processed and understood by software agents.
- Three OWL-S subontologies:
 - **Service profile Expresses** what a service does in order to enable service advertisement and discovery.
 - **Service model Describes** how the service works in order to enable service invocation, composition, monitoring, and recovery.
 - **Service grounding Specifies** the details of how to access the service.

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

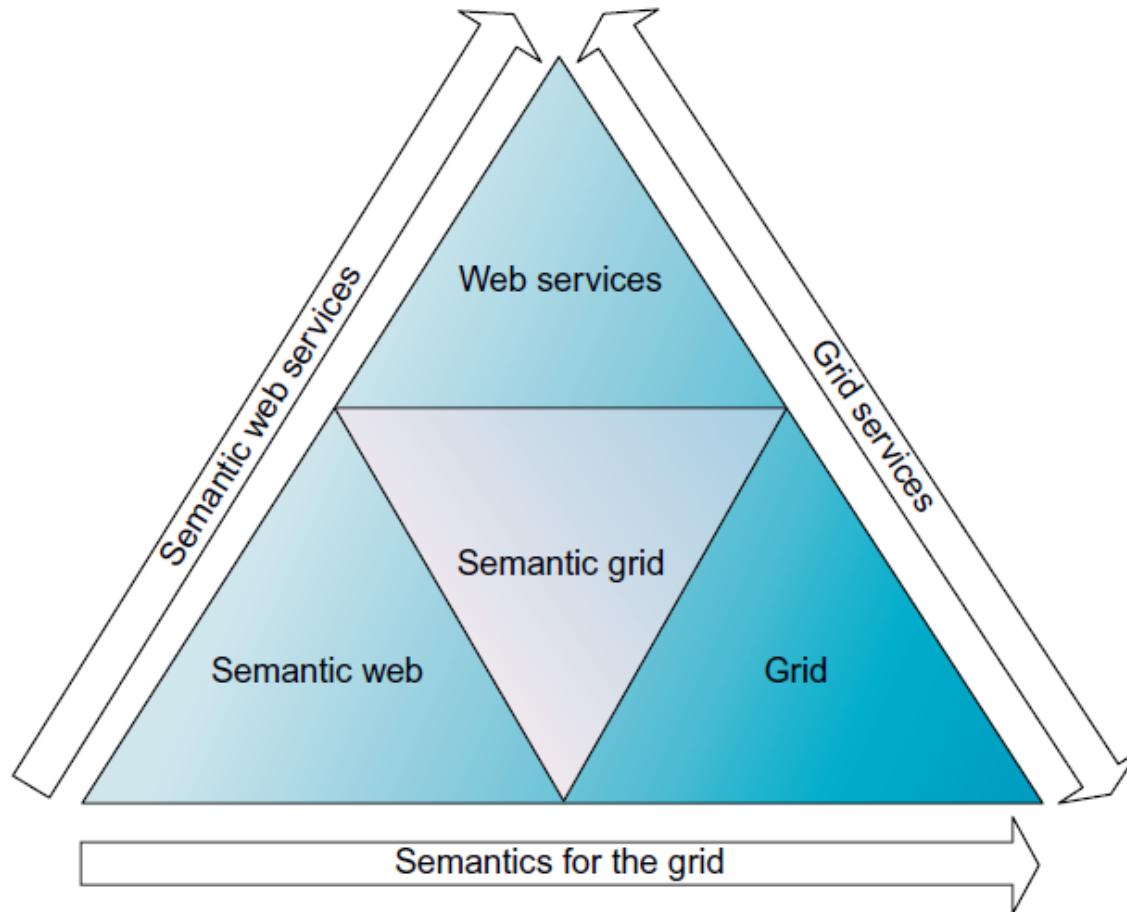


FIGURE 5.12

Semantic grid-related concepts and technologies.

(Courtesy of Goble and Roura, (ECAI-2004), Valencia, Spain, [95])

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

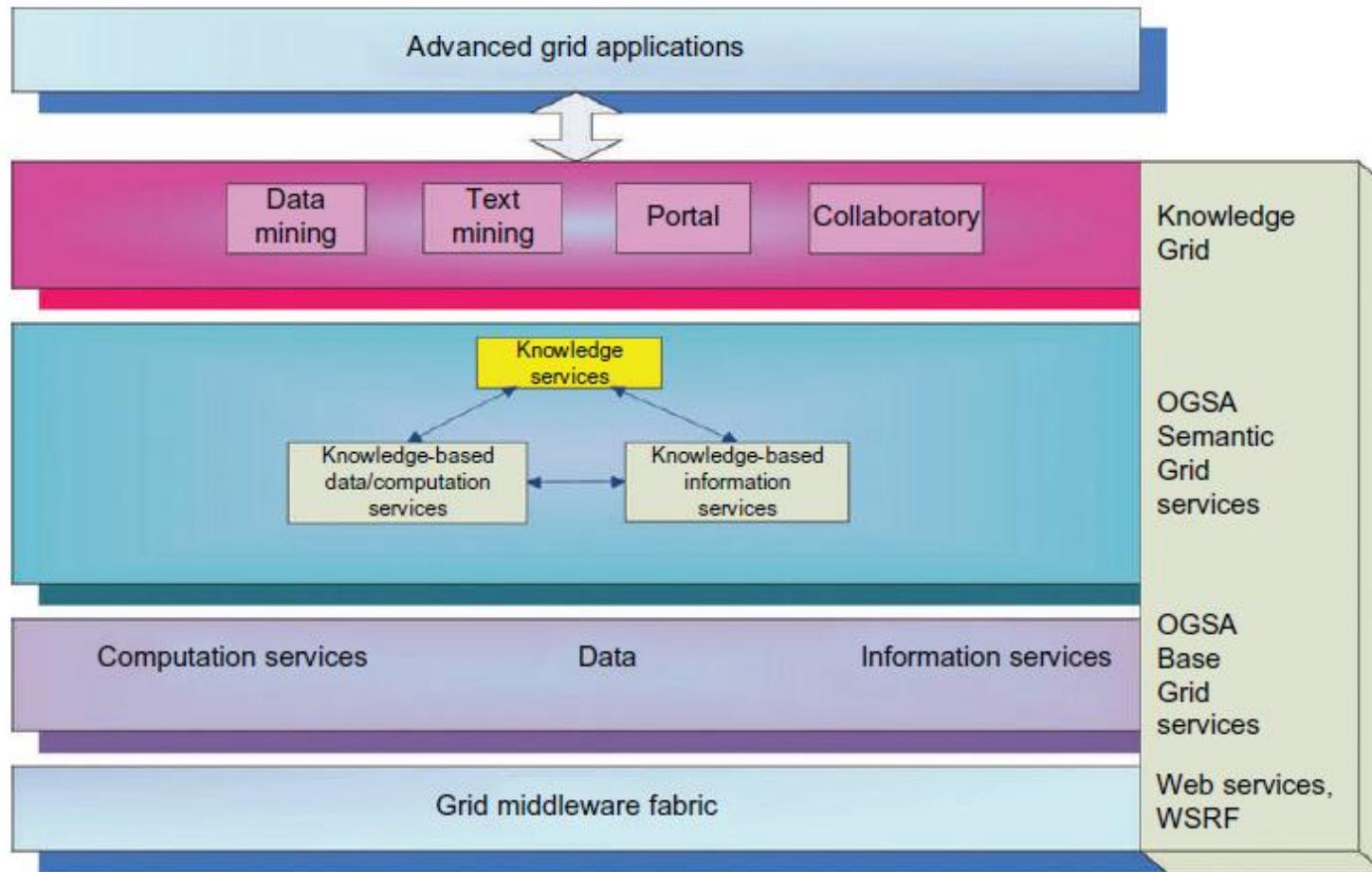


FIGURE 5.13

Semantic grid architecture.

(Courtesy of Goble and Roura, (ECAI-2004), Valencia, Spain, [95])

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Job Execution Environments and Monitoring

- A distributed job execution environment often consists of two components:

**Job execution
engine**

- Job scheduling, resource allocation, and other issues such as fault tolerance

**Distributed data
management system**

Provides an abstraction for jobs to access distributed data

Example:

- Google MapReduce,
- Microsoft Dryad

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Job Execution Environments and Monitoring (contd..)

- **MapReduce:**
- MapReduce was primarily designed to support Google applications that use and generate large data sets.
- It generalizes a map/reduce abstraction from these applications and provides a simple programming model for distributed execution of programs.
- The scheduling mechanism in MapReduce considers data locality by using the data location information provided by the GFS metadata server.
- Similar scheduling strategy is exploited in location aware request distribution algorithms (LARD) for web server clusters

DISCOVERY, REGISTRIES, METADATA, AND DATABASES

Job Execution Environments and Monitoring (contd..)

- **Dryad:**
- Dryad has a similar scheduling mechanism.
- Each execution of a vertex in a DAG has an execution record that contains the state of the execution and the versions of the predecessor vertices providing its input data.
- The scheduler then does the matchmaking by allocating the vertex to the resource.
- The approach has a long history and can be traced back to the Linda programming model.
- As jobs can be dispatched to different nodes, a job execution environment typically needs the support of a distributed data management system for a job to access remote data sets and, sometimes, exchange data with other jobs.

NEXT CLASS....

SERVICE-ORIENTED ARCHITECTURE

(CONTD..)

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L24

WORKFLOW IN SERVICE-ORIENTED ARCHITECTURES

- Basic Workflow Concepts
- Workflow Standards
- Workflow Architecture and Specification
- Workflow Execution Engine
- Scripting Workflow System Swift

WORKFLOW IN SERVICE-ORIENTED ARCHITECTURES

- “real system” consists of multiple interacting
- Grid of services involves:
 - simple sensor (perhaps only an output data stream)
 - a complete grid (a collection of services with multiple input and output message ports)
 - Grid of grid or grid of clouds.
- By adding application-specific services, one builds a distributed system to support the study of flood, gas, or electrical infrastructure.
- Workflow is used to integrate component grids and services.

WORKFLOW IN SERVICE-ORIENTED ARCHITECTURES

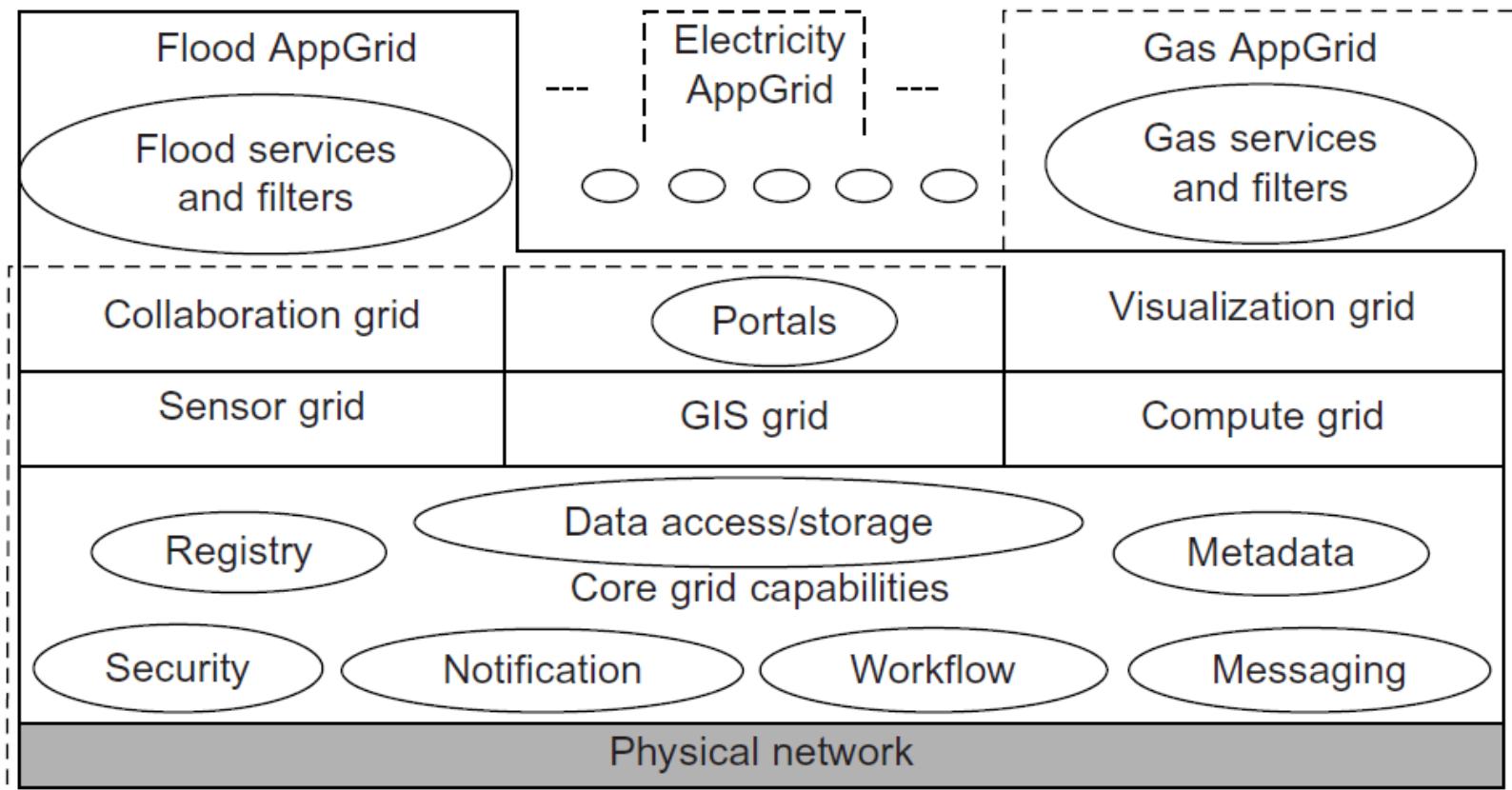


FIGURE 5.14

The concept of the grid of grids of services.

BASIC WORKFLOW CONCEPTS

- Workflow means an approach to program the interaction between the services.
- AKA “software coordination”, “Service orchestration”, “service or process coordination”, “service conversation”, “web or grid scripting”, “application integration,” or “software bus.
- It is an area of active research with different approaches emphasizing control flow, scheduling, and/or data flow.

BASIC WORKFLOW CONCEPTS (CONTD..)

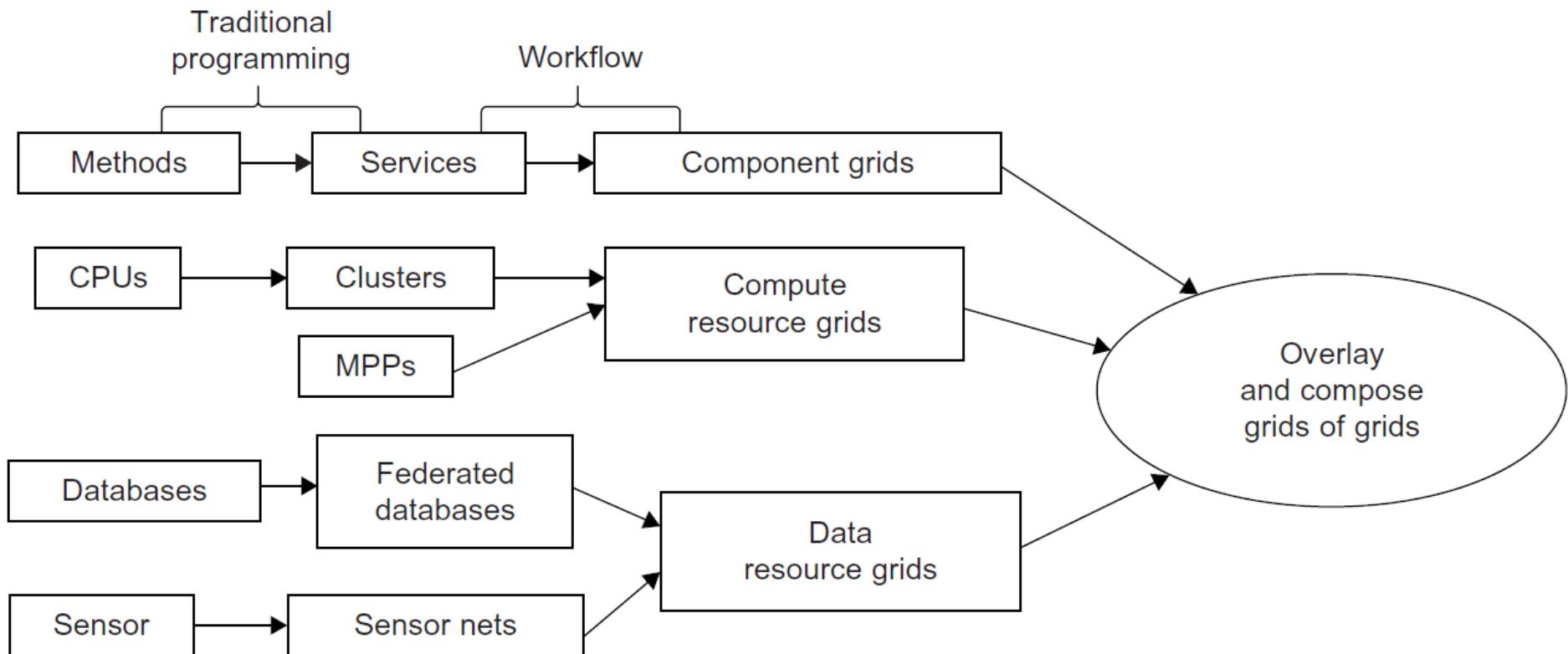


FIGURE 5.15

Hierarchical computing, data, and programming abstraction.

WORKFLOW STANDARDS (CONTD..)

Table 5.9 Workflow Standards, Links, and Status

Standard	Link	Status
BPEL Business Process Execution Language for Web Services (OASIS) V 2.0	http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html ; http://en.wikipedia.org/wiki/BPEL	April 2007
WS-CDL Web Service Choreography Description Language (W3C)	http://www.w3.org/TR/ws-cdl-10/	November 2005, not final
WSCI Web Service Choreography Interface V 1.0 (W3C)	http://www.w3.org/TR/wsci/	August 2002, note only
WSCL Web Services Conversation Language (W3C)	http://www.w3.org/TR/wscl10/	March 2002, note only
WSFL Web Services Flow Language	http://www.ibm.com/developerworks/webservices/library/ws-wsfl2/	Replaced by BPEL
XLANG Web Services for Business Process Design (Microsoft)	http://xml.coverpages.org/XLANG-C-200106.html	June 2001, replaced by BPEL
WS-CAF Web Services Composite Application Framework including WS-CTX , WS-CF , and WS-TXM	http://en.wikipedia.org/wiki/WS-CAF	Unfinished

WORKFLOW STANDARDS

Table 5.9 Workflow Standards, Links, and Status

Standard	Link	Status
WS-CTX Web Services Context (OASIS Web Services Composite Application Framework TC)	http://docs.oasis-open.org/ws-CAF/ws-context/v1.0/OS/wsctx.html	April 2007
WS-Coordination Web Services Coordination (BEA, IBM, Microsoft at OASIS)	http://docs.oasis-open.org/ws-tx/wscoor/2006/06	February 2009
WS-AtomicTransaction Web Services Atomic Transaction (BEA, IBM, Microsoft at OASIS)	http://docs.oasis-open.org/ws-tx/wsat/2006/06	February 2009
WS-BusinessActivity Framework (BEA, IBM, Microsoft at OASIS)	http://docs.oasis-open.org/ws-tx/wsba/2006/06	February 2009
BPMN Business Process Modeling Notation (Object Management Group, OMG)	http://en.wikipedia.org/wiki/BPMN ; http://www.bpmn.org/	Active
BPSS Business Process Specification Schema (OASIS)	http://www.ebxml.org/ ; http://www.ebxml.org/specs/ebBPSS.pdf	May 2001
BTP Business Transaction Protocol (OASIS)	http://www.oasis-open.org/committees/download.php/12449/business_transaction-btp-1.1-spec	Unfinished

WORKFLOW ARCHITECTURE AND SPECIFICATION

- Workflow system have two components:
 - Programming language
 - Runtime components
- Workflow specification
 - Scripting-base workflow – similar to Python, JavaScript, Perl

WORKFLOW ARCHITECTURE AND SPECIFICATION (CONTD..)

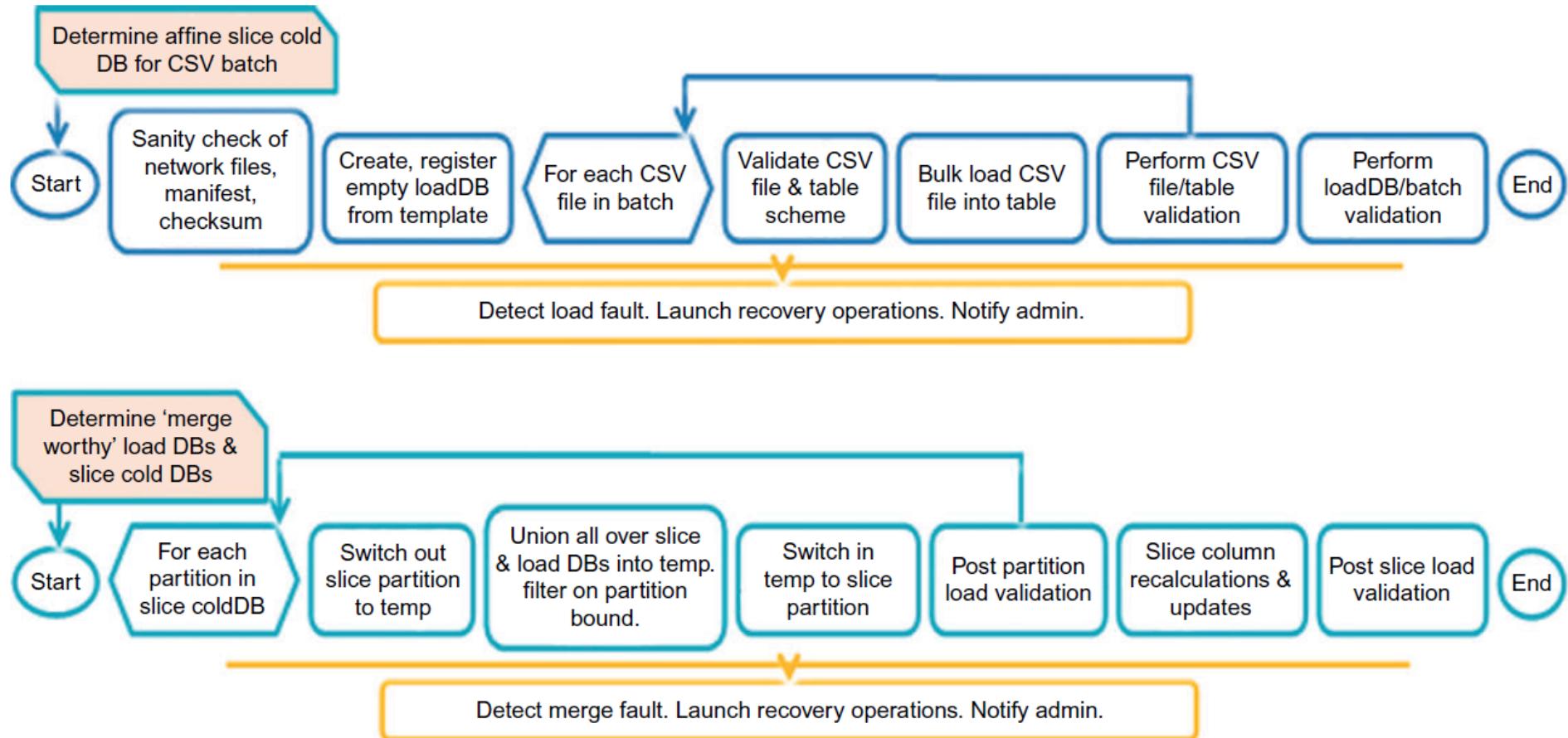


FIGURE 5.16

Two typical (Load and Merge) workflows from the Pan-STARRS astronomy data processing area.

WORKFLOW EXECUTION ENGINE

- Workflows domains are from meteorology and ocean modeling, bioinformatics and biomedical workflows, astronomy and neutron science.
- These are examined according to their size, resource use, graph pattern, data pattern, and usage scenario.
- A more general workflow structure is that of the DAG, which is a collection of vertices and directed edges, each edge connecting one vertex to another such that there are no cycles.
- Specifies the interaction between the constituent services or activities.
- One important technology choice is the mechanism for transferring information between the nodes of the graph.

WORKFLOW EXECUTION ENGINE (CONTD..)

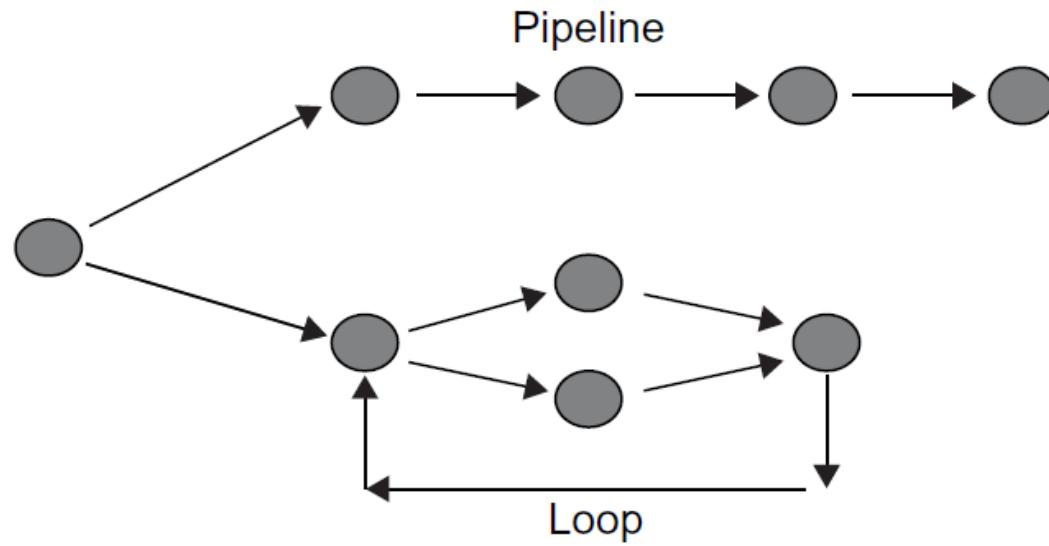


FIGURE 5.18

A workflow graph that includes subgraphs illustrating pipelines and loops.

WORKFLOW EXECUTION ENGINE (CONTD..)

- There are often two communication systems in workflow environments corresponding to “control” and “data,” respectively.
- Obviously, the control communication would usually have small messages and very different requirements from the data network.
- In this regard, one should mention the “proxy model” which is often used in grid architectures and workflow.

SCRIPTING WORKFLOW SYSTEM SWIFT

- Swift is a parallel scripting language in which application programs are represented as functions and variables can be mapped to files.
- Structure and array abstractions are used to manipulate sets of files in parallel.
- Swift has a functional data-flow-based execution model in which all statements are implicitly parallel.
- Parallel looping constructs explicitly specify large-scale parallel processing.
- Figure 5.19 shows the architecture of the Swift workflow system.

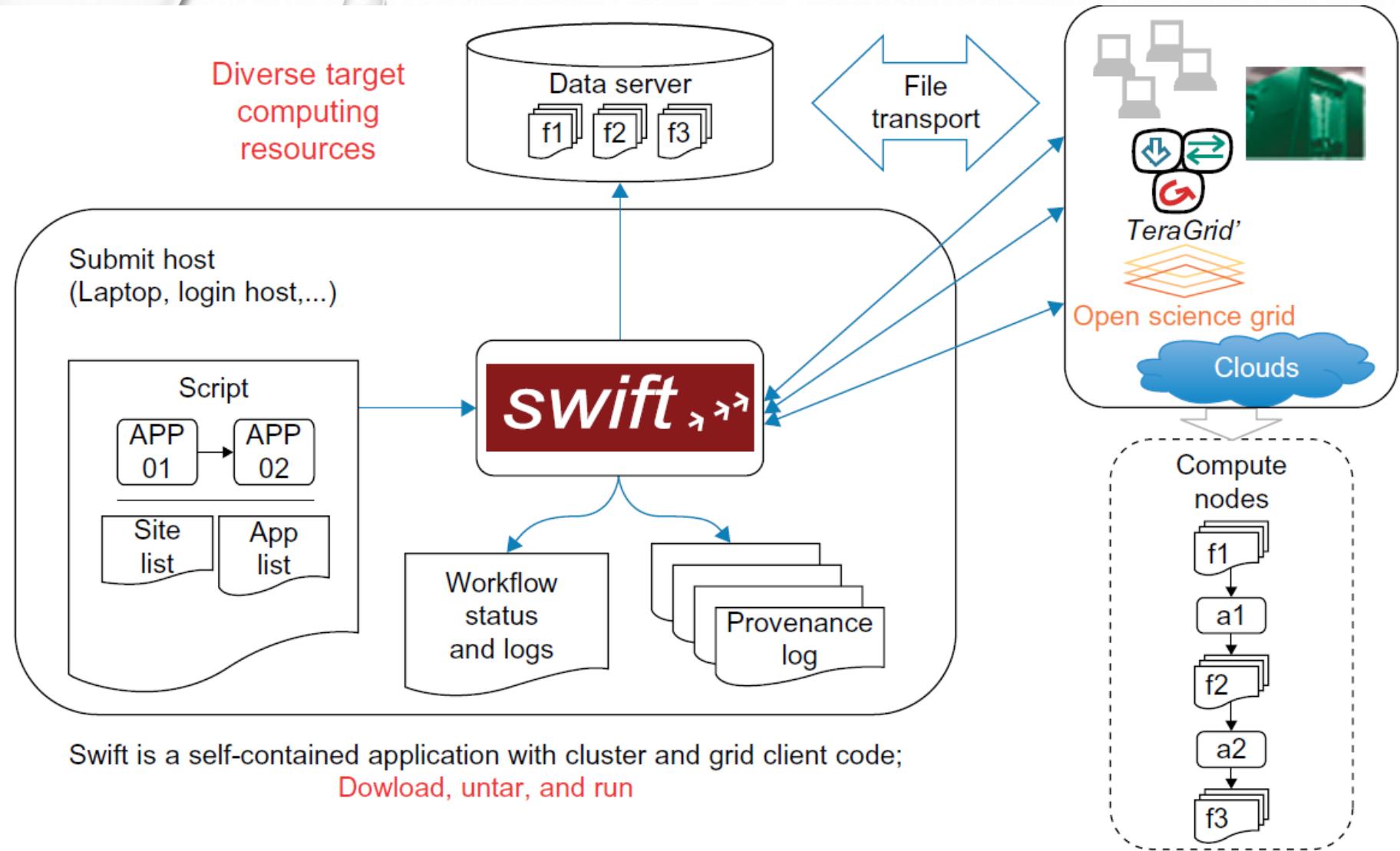


FIGURE 5.19

Swift workflow system architecture.

NEXT CLASS

CLOUD PROGRAMMING ENVIRONMENT

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L-26-29

PRINCIPLES OF PARALLEL AND DISTRIBUTED COMPUTING (CONTD..)

- The term **parallel computing** and **distributed computing** are often used interchangeably, even though they mean slightly different things.
- The term **parallel** implies a **tightly coupled system**, where as **distributed systems** refers to a wider class of system, including those that are **tightly coupled**.
- More precisely, the term **parallel computing** refers to a model in which the computation is divided among several processors sharing the same memory.
- The architecture of **parallel computing system** is often characterized by the **homogeneity of components**: each processor is of the same type and it has the same capability as the others.

PRINCIPLES OF PARALLEL AND DISTRIBUTED COMPUTING (CONTD..)

- The shared memory has a single address space, which is accessible to all the processors.
- Parallel programs are then broken down into several units of execution that can be allocated to different processors and can communicate with each other by means of shared memory.
- Originally parallel systems are considered as those architectures that featured multiple processors sharing the same physical memory and that were considered a single computer.
 - Over time, these restrictions have been relaxed, and parallel systems now include all architectures that are based on the concept of shared memory, whether this is physically present or created with the support of libraries, specific hardware, and a highly efficient networking infrastructure.

PRINCIPLES OF PARALLEL AND DISTRIBUTED COMPUTING (CONTD..)

- The term distributed computing encompasses any architecture or system that allows the computation to be broken down into units and executed concurrently on different computing elements, whether these are processors on different nodes, processors on the same computer, or cores within the same processor.
- Distributed computing includes a wider range of systems and applications than parallel computing and is often considered a more general term.
- Even though it is not a rule, the term distributed often implies that the locations of the computing elements are not the same and such elements might be heterogeneous in terms of hardware and software features.
- Classic examples of distributed computing systems are
 - Computing Grids
 - Internet Computing Systems

WHAT IS PARALLEL PROCESSING?

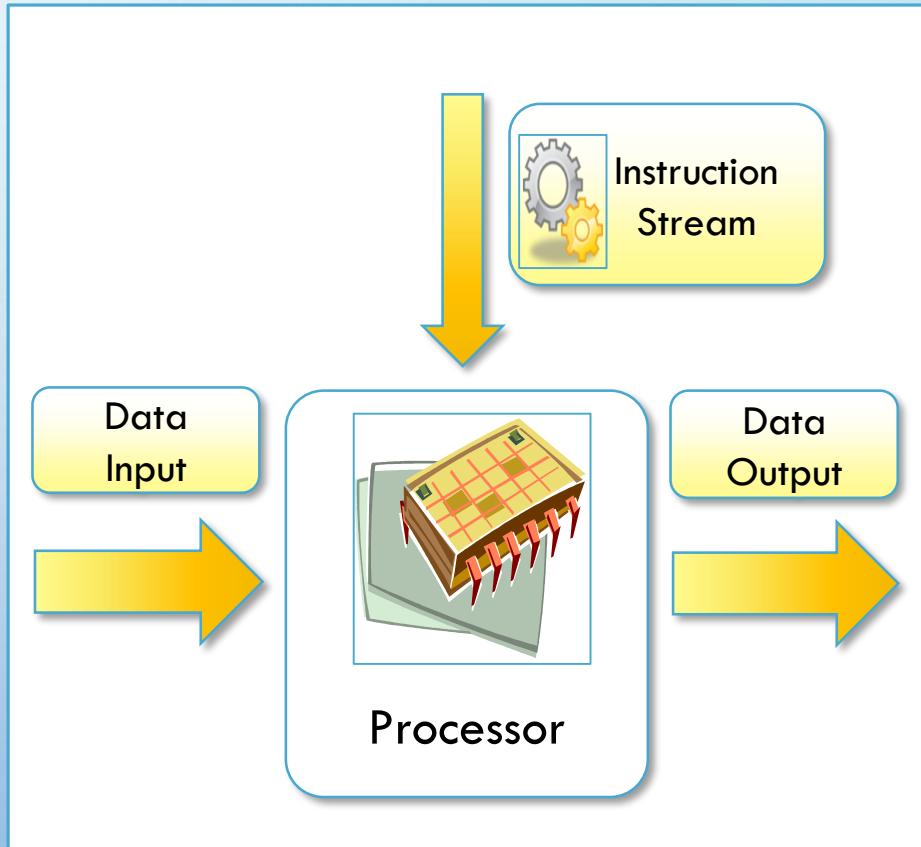
- Processing of multiple tasks simultaneously on multiple processors is called parallel processing.
- The parallel program consists of multiple active processes (tasks) simultaneously solving a given problem.
- A given task is divided into multiple subtasks using a divide-and-conquer technique, and each subtask is processed on a different central processing unit (CPU).
- Programming on multi processor system using the divide-and-conquer technique is called parallel programming.
- Many applications today require more computing power than a traditional sequential computer can offer.
- Parallel Processing provides a cost effective solution to this problem by increasing the number of CPUs in a computer and by adding an efficient communication system between them.
- The workload can then be shared between different processors. This setup results in higher computing power and performance than a single processor a system offers.

HARDWARE ARCHITECTURES FOR PARALLEL PROCESSING

- The core elements of parallel processing are CPUs. Based on the number of instructions and data streams, that can be processed simultaneously, computing systems are classified into the following four categories:
 - Single-instruction, Single-data (SISD) systems
 - Single-instruction, Multiple-data (SIMD) systems
 - Multiple-instruction, Single-data (MISD) systems
 - Multiple-instruction, Multiple-data (MIMD) systems

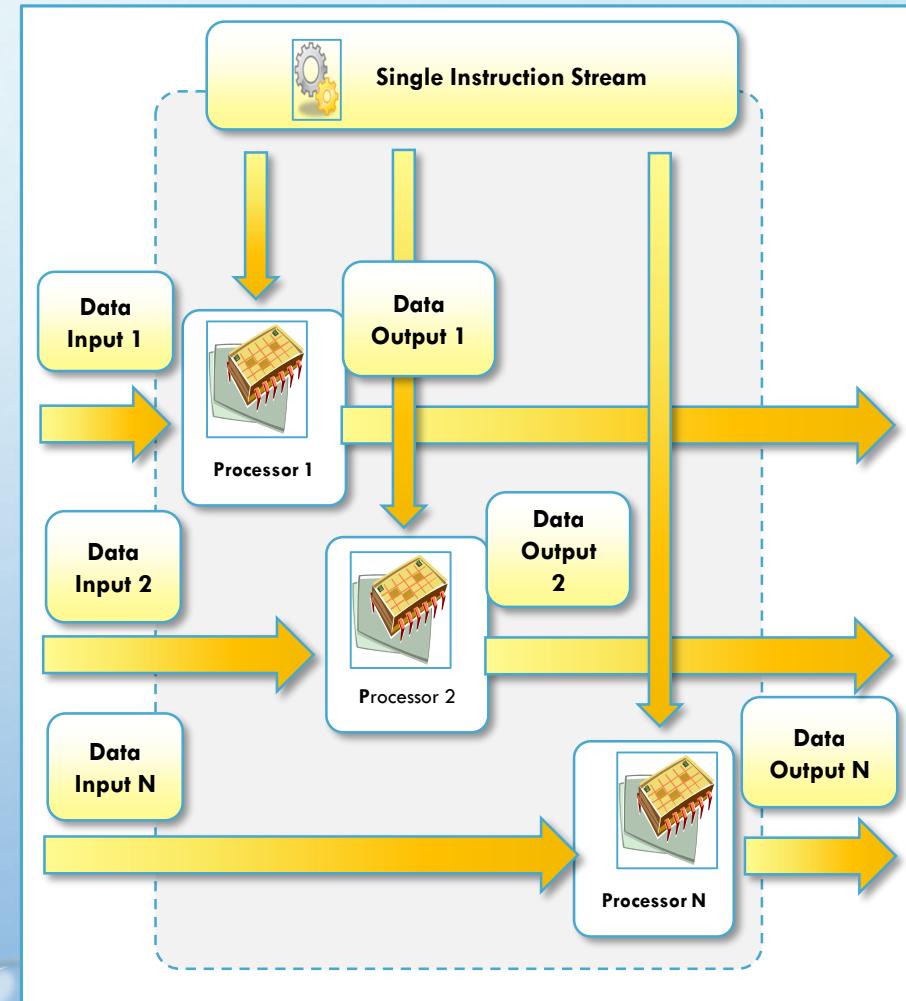
SINGLE – INSTRUCTION , SINGLE DATA (SISD) SYSTEMS

- SISD computing system is a uni-processor machine capable of executing a single instruction, which operates on a single data stream.
- Machine instructions are processed sequentially, hence computers adopting this model are popularly called sequential computers.
- Most conventional computers are built using SISD model.
- All the instructions and data to be processed have to be stored in primary memory.
- The speed of processing element in the SISD model is limited by the rate at which the computer can transfer information internally.
- Dominant representative SISD systems are IBM PC, Macintosh, and workstations.



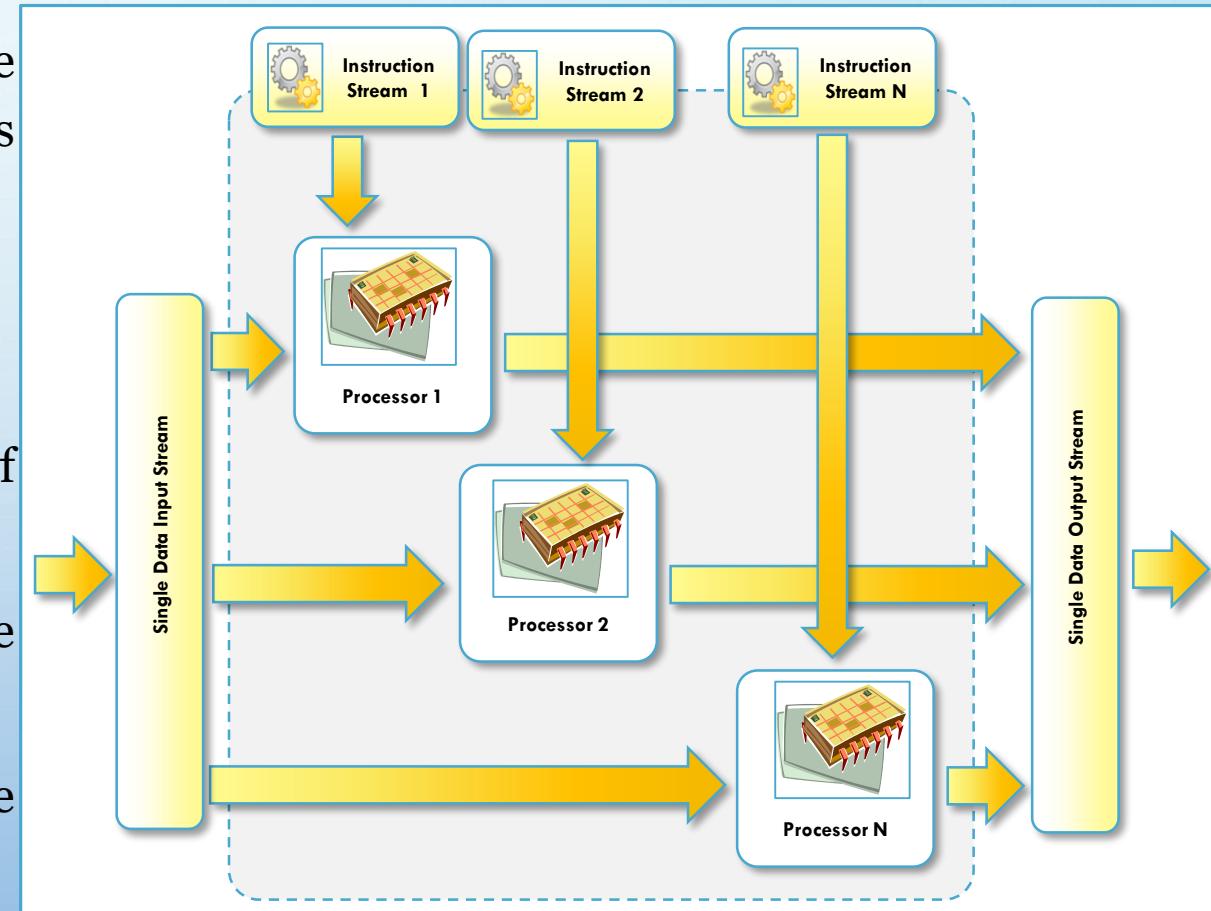
SINGLE – INSTRUCTION , MULTIPLE DATA (SIMD) SYSTEMS

- SIMD computing system is a multiprocessor machine capable of executing the same instruction on all the CPUS but operating on different data streams.
- Machines based on this model are well suited for scientific computing since they involve lots of vector and matrix operations.
- For instance statement $C_i = A_i * B_i$, can be passed to all the processing elements (PEs), organized data elements of vectors A and B can be divided into multiple sets (N- sets for N PE systems), and each pe can process one data set.
- Dominant representative SIMD systems are cray's vector processing machine.



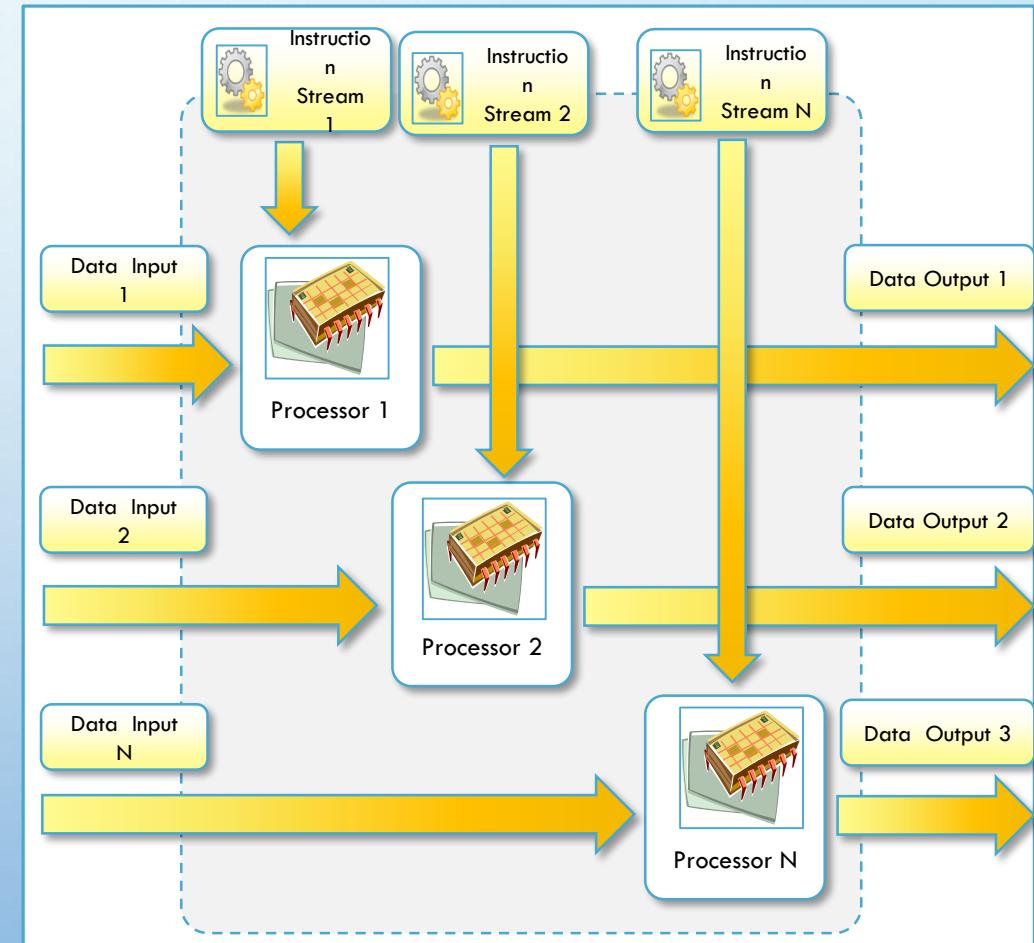
MULTIPLE – INSTRUCTION , SINGLE DATA (MISD) SYSTEMS

- MISD computing system is a multi processor machine capable of executing different instructions on different pes all of them operating on the same data set.
- For example
 - $Y = \sin(x) + \cos(x) + \tan(x)$
- Machines built using MISD model are not useful in most of the applications.
- Few machines are built but none of them available commercially.
- This type of systems are more of an intellectual exercise than a practical configuration.



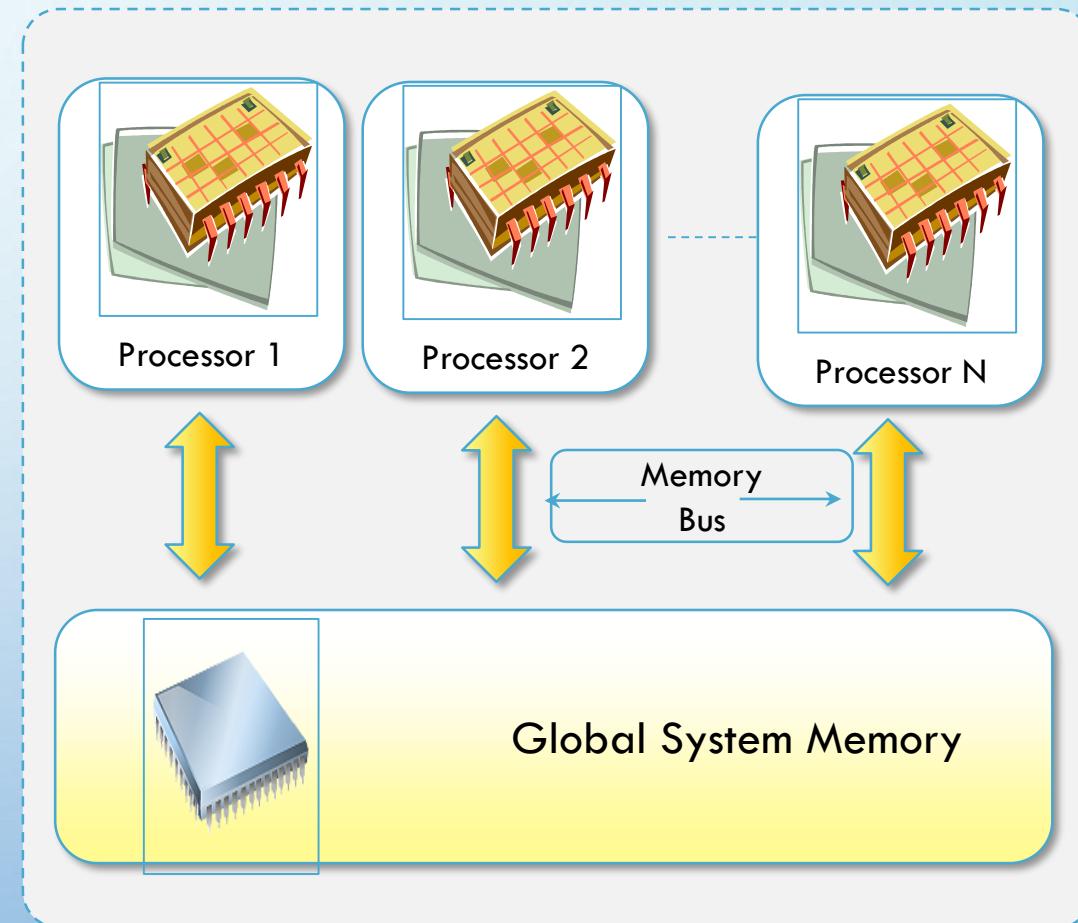
MULTIPLE – INSTRUCTION , MULTIPLE DATA (MIMD) SYSTEMS

- MIMD computing system is a multi processor machine capable of executing multiple instructions on multiple data sets.
- Each pe in the MIMD model has separate instruction and data streams, hence machines built using this model are well suited to any kind of application.
- Unlike SIMD, MISD machine, PEs in MIMD machines work asynchronously,
- MIMD machines are broadly categorized into shared-memory MIMD and distributed memory MIMD based on the way pes are coupled to the main memory.



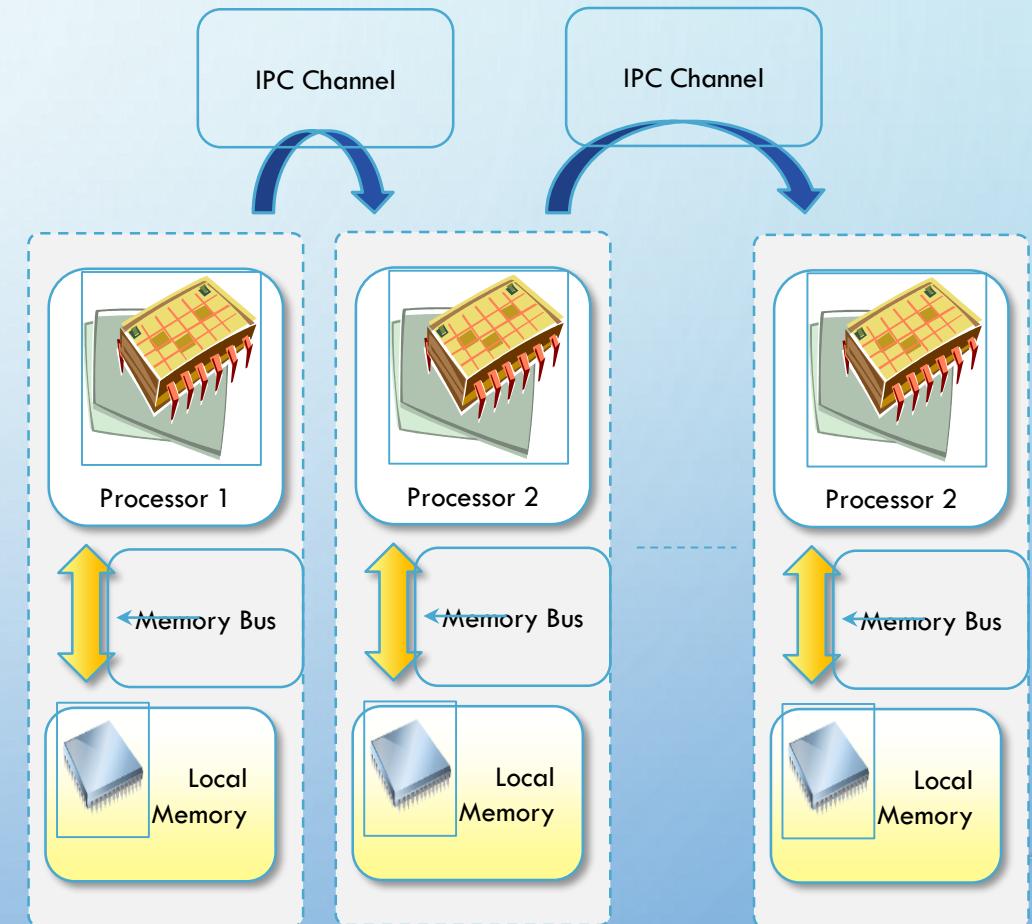
SHARED MEMORY MIMD MACHINES

- All the pes are connected to a single global memory and they all have access to it.
- Systems based on this model are also called tightly coupled multi processor systems.
- The communication between PEs in this model takes place through the shared memory.
- Modification of the data stored in the global memory by one PE is visible to all other PEs.
- Dominant representative shared memory MIMD systems are silicon graphics machines and Sun/IBM SMP (symmetric multi-processing).



DISTRIBUTED MEMORY MIMD MACHINES

- All pes have a local memory. Systems based on this model are also called loosely coupled multi processor systems.
- The communication between PEs in this model takes place through the interconnection network, the inter process communication channel, or IPC.
- The network connecting pes can be configured to tree, mesh, cube, and so on.
- Each PE operates asynchronously, and if communication/synchronization among tasks is necessary, they can do so by exchanging messages between them.



SHARED VS DISTRIBUTED MIMD MODEL

- The shared memory MIMD architecture is easier to program but is less tolerant to failures and harder to extend with respect to the distributed memory MIMD model.
- Failures, in a shared memory MIMD affect the entire system, whereas this is not the case of the distributed model, in which each of the PEs can be easily isolated.
- Moreover, shared memory MIMD architectures are less likely to scale because the addition of more PEs leads to memory contention.
- This is a situation that does not happen in the case of distributed memory, in which each PE has its own memory.
- As a result, distributed memory MIMD architectures are most popular today.

APPROACHES TO PARALLEL PROGRAMMING

- A sequential program is one that runs on a single processor and has a single line of control.
- To make many processors collectively work on a single program, the program must be divided into smaller independent chunks so that each processor can work on separate chunks of the problem.
- The program decomposed in this way is a parallel program.
- A wide variety of parallel programming approaches are available.

APPROACHES TO PARALLEL PROGRAMMING

- The most prominent among them are the following.
 - Data Parallelism
 - Process Parallelism
 - Farmer-and-worker model
- The above said three models are suitable for task-level parallelism. In the case of data level parallelism, the divide-and-conquer technique is used to split data into multiple sets, and each data set is processed on different PEs using the same instruction.
- This approach is highly suitable to processing on machines based on the SIMD model.

APPROACHES TO PARALLEL PROGRAMMING

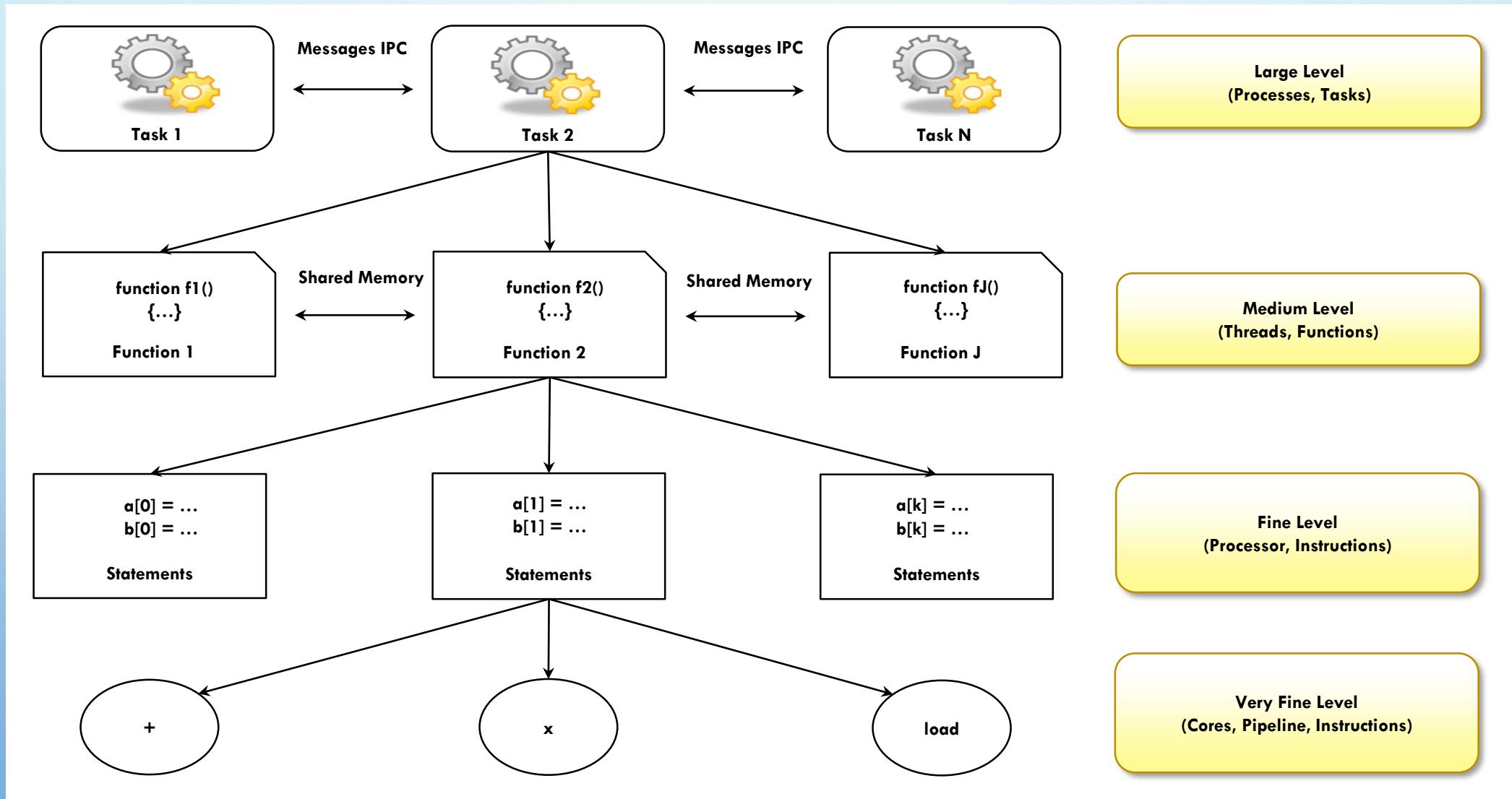
- In the case of Process Parallelism, a given operation has multiple (but distinct) activities that can be processed on multiple processors.
- In the case of Farmer-and-Worker model, a job distribution approach is used, one processor is configured as master and all other remaining PEs are designated as slaves, the master assigns the jobs to slave PEs and, on completion, they inform the master, which in turn collects results.
- These approaches can be utilized in different levels of parallelism.

LEVELS OF PARALLELISM

- Levels of Parallelism are decided on the size of code (grain size) that can be a potential candidate of parallelism.
- The table shows the levels of parallelism.
- All these approaches have a common goal
 - To boost processor efficiency by hiding latency.
 - To conceal latency, there must be another thread ready to run whenever a lengthy operation occurs.
- The idea is to execute concurrently two or more single-threaded applications. Such as compiling, text formatting, database searching, and device simulation.

Grain Size	Code Item	Parallelized By
Large	Separate and heavy weight process	Programmer
Medium	Function or procedure	Programmer
Fine	Loop or instruction block	Parallelizing compiler
Very Fine	Instruction	Processor

LEVELS OF PARALLELISM (CONTD..)



ELEMENTS OF DISTRIBUTED COMPUTING

- In the previous section we discussed techniques and architectures that allow introduction of parallelism within a single machine or system and how parallelism operates at different levels of the computing stack.
- Here extend these concepts and explore how multiple activities can be performed by leveraging systems composed of multiple heterogeneous machines and systems.
- We discuss what is generally referred to as distributed computing and more precisely introduce the most common guidelines and patterns for implementing distributed computing systems from the perspective of the software designer.

GENERAL CONCEPTS AND DEFINITIONS

- Distributed computing studies the models, architectures, and algorithms used for building and managing distributed systems.
- As general definition of the term distributed system, we use the one proposed by Tanenbaum
 - A distributed system is a collection of independent computers that appears to its users as a single coherent system.
- This definition is general enough to include various types of distributed computing systems that are especially focused on unified usage and aggregation of distributed resources.

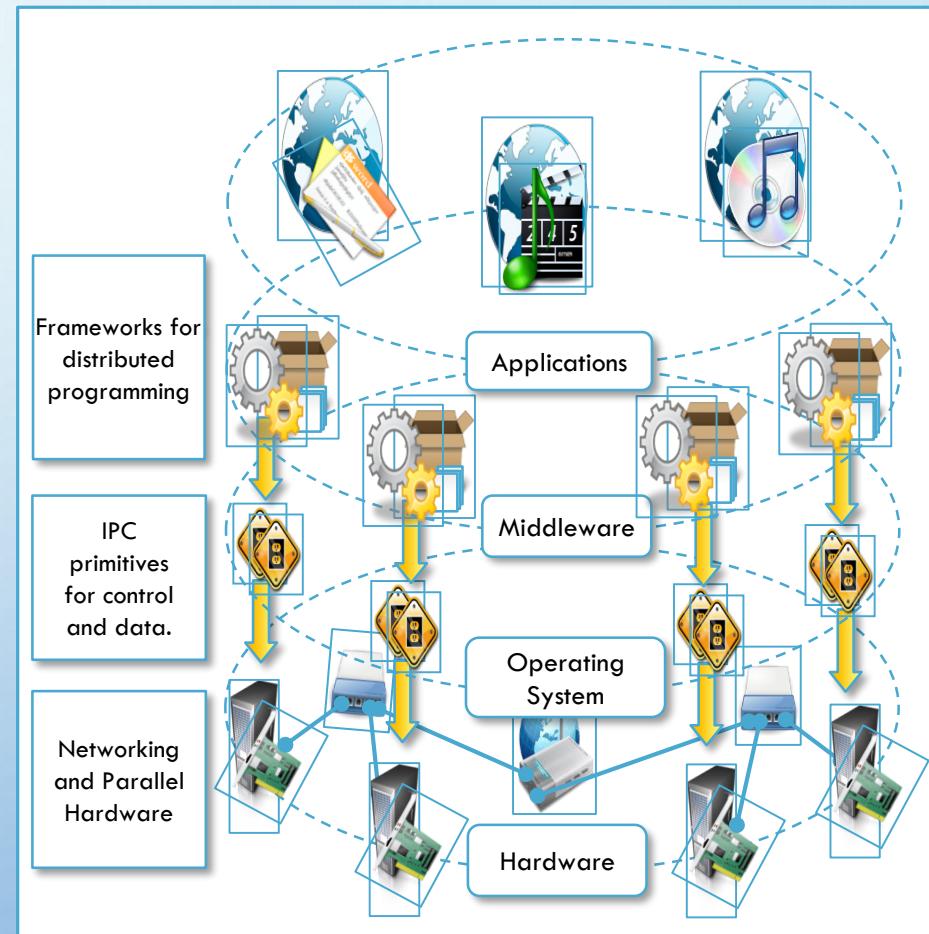
GENERAL CONCEPTS AND DEFINITIONS

(CONTD..)

- Here, we focus on the architectural models, that are used to harness independent computers and present them as a whole coherent system.
- Communications is another fundamental aspect of distributed computing. Since distributed systems are composed of more than one computer that collaborate together, it is necessary to provide some sort of data and information exchange between them, which generally occurs through the network.
 - A distributed system is one in which components located at networked computers communicate and coordinate their action only by passing messages.
 - As specified in this definition, the components of a distributed system communicate with some sort of message passing. This is a term that encompasses several communication models.

COMPONENTS OF DISTRIBUTED SYSTEM

- A distributed system is the result of the interaction of several components that traverse the entire computing stack from hardware to software.
- It emerges from the collaboration of several elements that- by working together- give users the illusion of a single coherent system.
- The figure provides an overview of the different layers that are involved in providing the services of a distributed system.
- At the very bottom layer, computer and network hardware constitute the physical infrastructure; these components are directly managed by the operating system, which provides the basic services for inter process communication (IPC), process scheduling and management, and resource management in terms of file system and local devices.
- Taken together these two layers become the platform on top of which specialized software is deployed to turn a set of networked computers into a distributed system.



TECHNOLOGIES FOR DISTRIBUTED COMPUTING

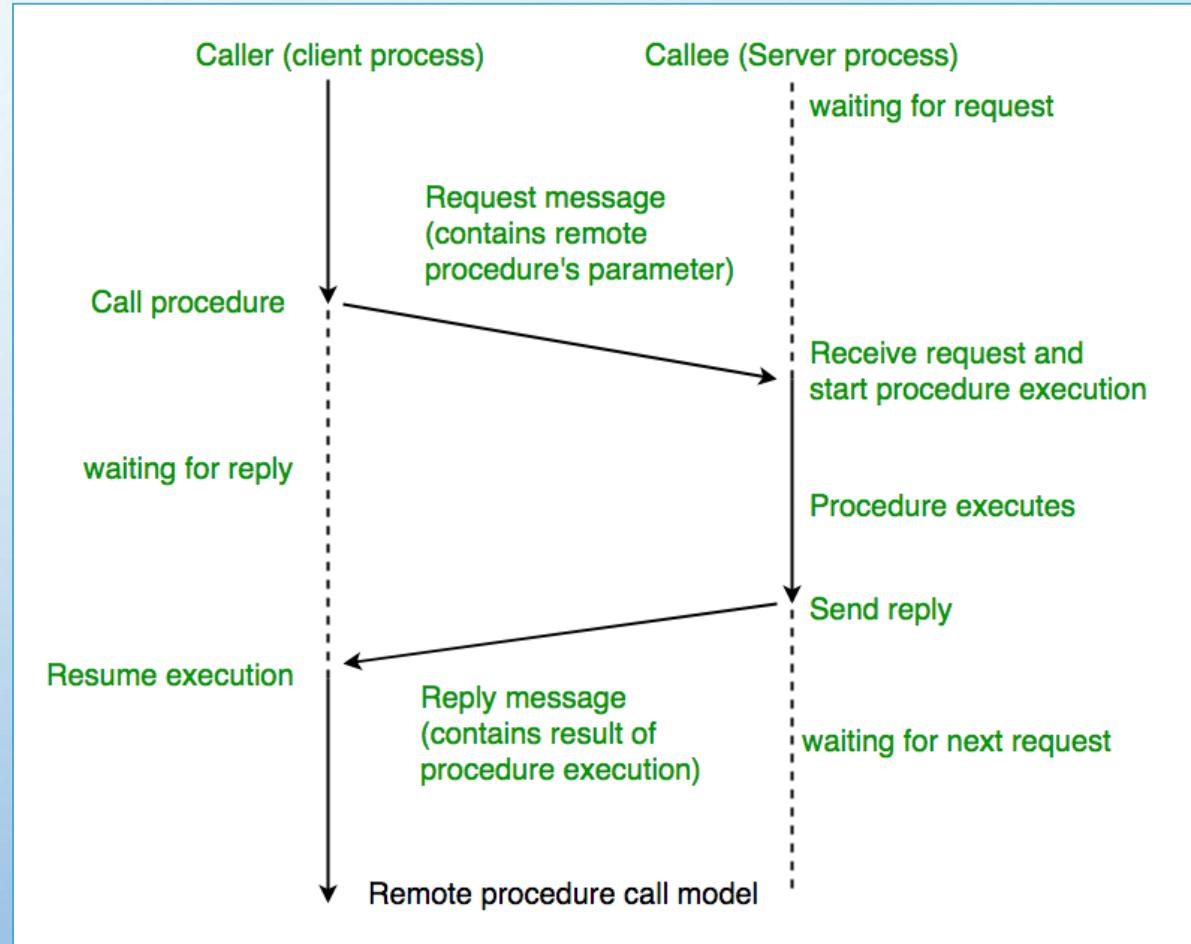
- Remote Procedure Call (RPC)
 - RPC is the fundamental abstraction enabling the execution procedures on clients' request.
 - RPC allows extending the concept of a procedure call beyond the boundaries of a process and a single memory address space.
 - The called procedure and calling procedure may be on the same system or they may be on different systems.
 - The important aspect of RPC is marshalling and unmarshalling.
- Distributed Object Frameworks
 - Extend object-oriented programming systems by allowing objects to be distributed across a heterogeneous network and provide facilities so that they can be coherently act as though they were in the same address space.

REMOTE PROCEDURE CALL (RPC)

- Remote Procedure Call (RPC) is a technique for constructing distributed, client-server based applications. It is based on extending the conventional local procedure calling so that the called procedure need not exist in the same address space as the calling procedure.
- The two processes may be on the same system, or they may be on different systems with a network connecting them.

REMOTE PROCEDURE CALL (RPC)

- The calling environment is suspended, procedure parameters are transferred across the network to the environment where the procedure is to execute, and the procedure is executed there.
- When the procedure finishes and produces its results, its results are transferred back to the calling environment, where execution resumes as if returning from a regular procedure call.

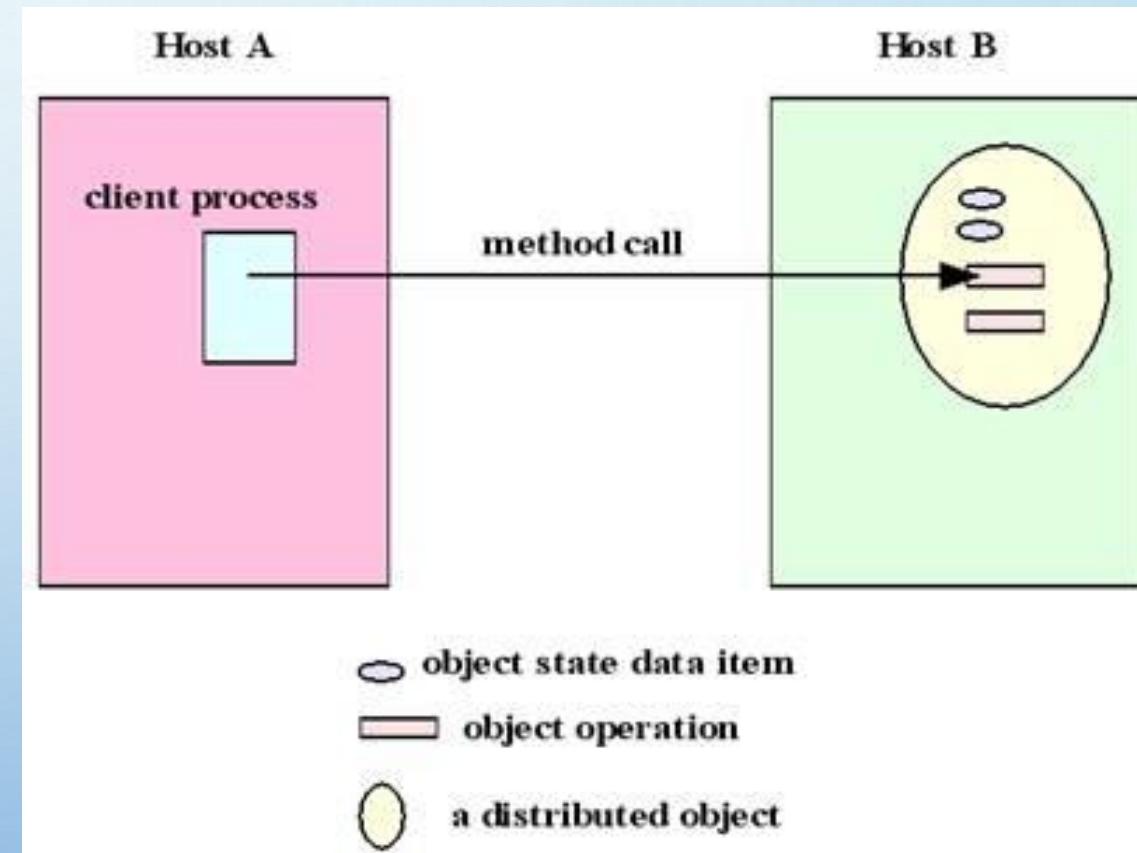


DISTRIBUTED OBJECT

- The distributed object paradigm
 - provides abstractions beyond those of the message-passing model.
 - In object-oriented programming, objects are used to represent an entity significant to an application.
- Each object encapsulates:
 - the state or data of the entity: in Java, such data is contained in the instance variables of each object;
 - the operations of the entity, through which the state of the entity can be accessed or updated.
- Local Objects vs. Distributed Objects
 - Local objects are those whose methods can only be invoked by a local process, a process that runs on the same computer on which the object exists.
 - A distributed object is one whose methods can be invoked by a remote process, a process running on a computer connected via a network to the computer on which the object exists.

THE DISTRIBUTED OBJECT PARADIGM

- In a distributed object paradigm, network resources are represented by distributed objects.
- To request service from a network resource, a process invokes one of its operations or methods, passing data as parameters to the method.
- The method is executed on the remote host, and the response is sent back to the requesting process as a return value.



NEXT CLASS....

SERVICE LEVEL AGREEMENTS (SLA)

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L30-32

INSPIRATION

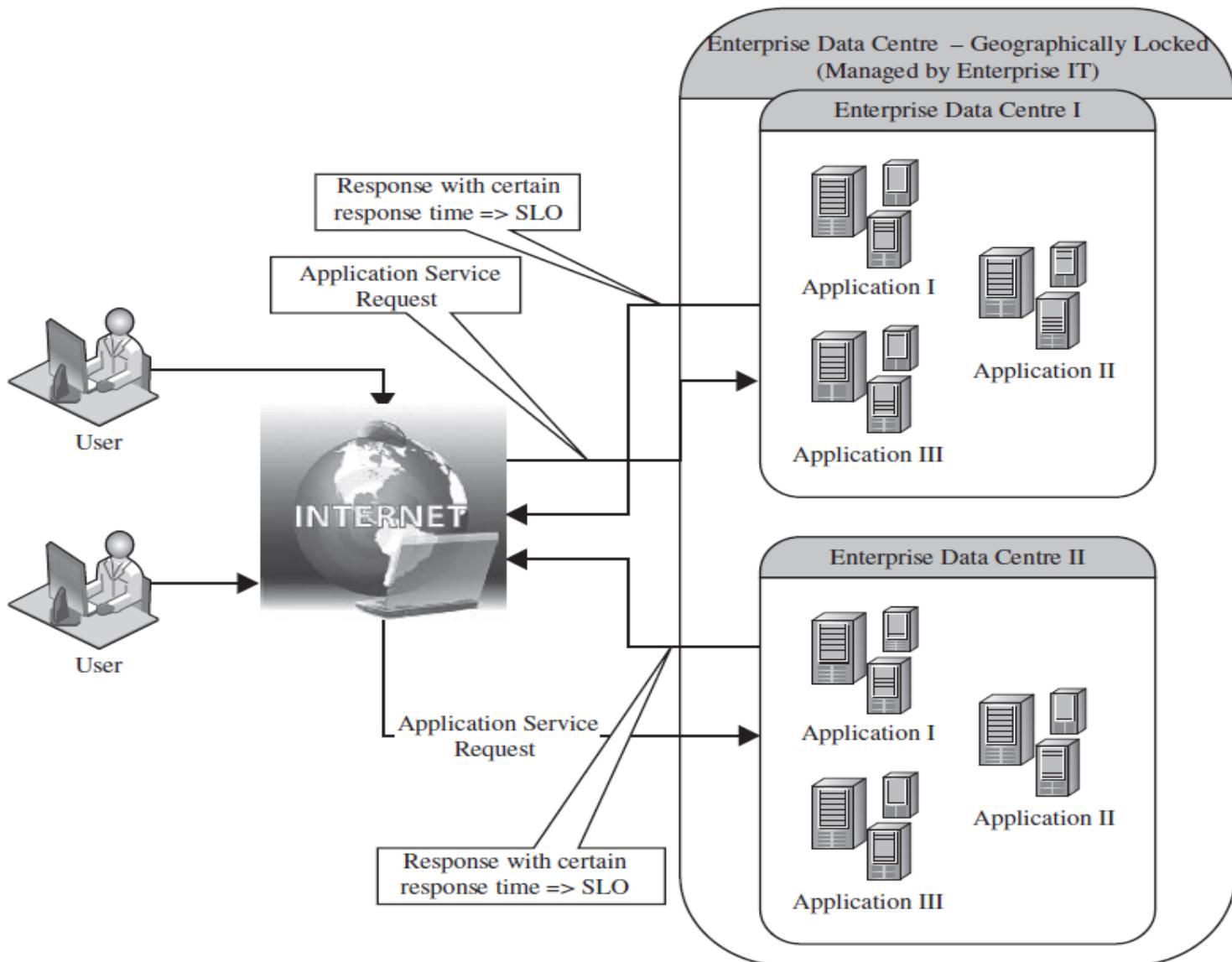


FIGURE 16.1. Hosting of applications on servers within enterprise's data centers.

INSPIRATION (CONTD..)

Enterprises realized that it was economical to outsource the application hosting activity to third-party infrastructure providers because:

- The enterprises need not invest in procuring expensive hardware upfront without knowing the viability of the business.
- The hardware and application maintenance were non-core activities of their business.
- As the number of web applications grew, the level of sophistication required to manage the data centers increased manyfold—hence the cost of maintaining them.

INSPIRATION (CONTD..)

- Enterprises developed the **web applications and deployed** on the infrastructure of the third-party service providers.
- These providers get **the required hardware** and make it available for application hosting.
- It necessitated the enterprises to enter into a **legal agreement** with the infrastructure service providers to guarantee a minimum quality of service (QoS).
- Typically, the **QoS parameters** are related to the **availability of the system CPU, data storage, and network for efficient execution of the application at peak loads.**
- **This legal agreement is known as the service-level agreement (SLA).**

INSPIRATION (CONTD..)

- Consider an example, one SLA may state that the **application's server machine will be available for 99.9% of the key business hours** of the application's end users, also called **core time**, and **85% of the non-core time**.
- Another SLA may state that the service provider would **respond to a reported issue in less than 10 minutes during the core time**, but **would respond in one hour during non-core time**.
- These SLAs are known as the **infrastructure SLAs**, and the infrastructure service providers are known as **Application Service Providers (ASPs)**.

INSPIRATION (CONTD..)

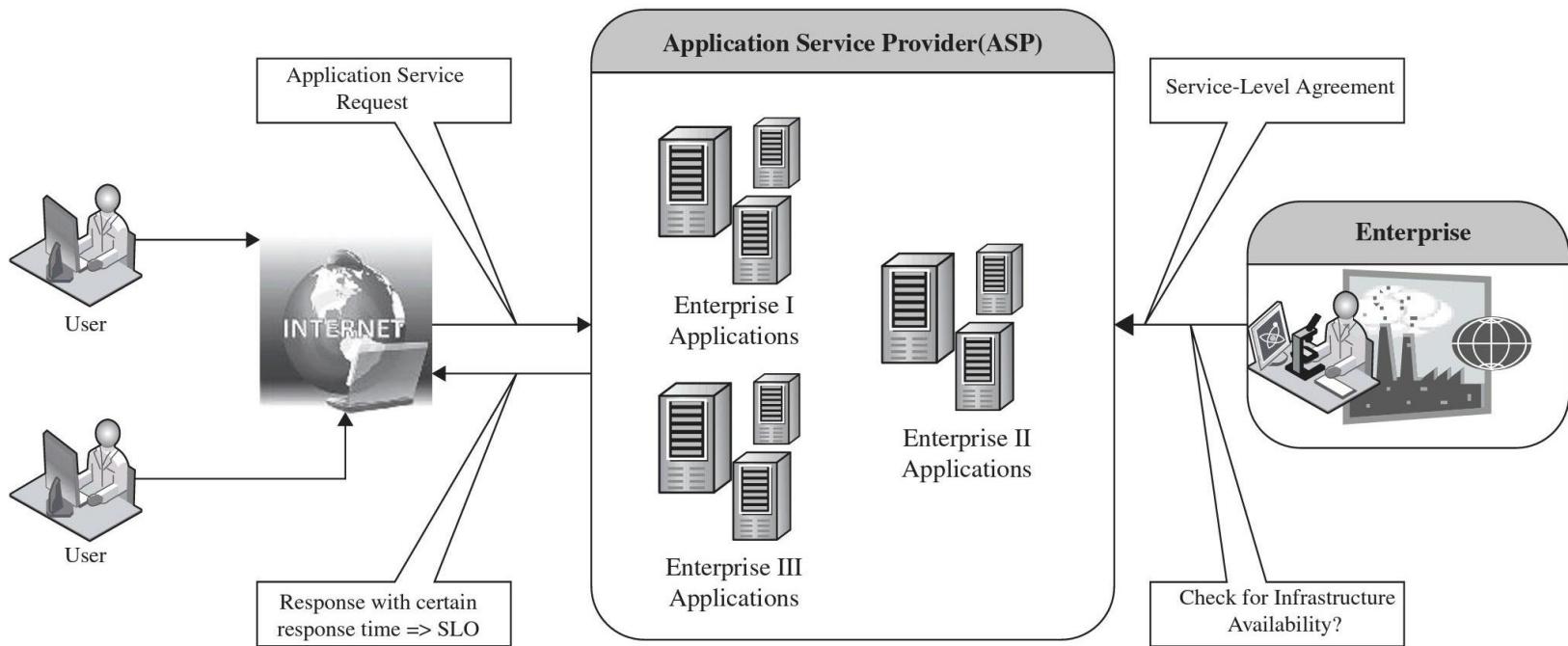


FIGURE 16.2. Dedicated hosting of applications in third party data centers.

INSPIRATION (CONTD..)

- Consequently, a set of tools for **monitoring and measurement** of availability of the infrastructure were required and developed.
- However, availability of the infrastructure doesn't automatically guarantee the availability of the application for its end users.
- These tools helped in tracking the **SLA adherence**.
- The responsibility for making the application available to its end users is with the enterprises.
- Therefore, the enterprises' IT team performs **capacity planning, and the infrastructure** provider procures the same.

INSPIRATION (CONTD..)

- The **dedicated hosting practice resulted in massive redundancies** within the ASP's data centers due to the underutilization of many of their servers.
- This is because the **applications were not fully utilizing their servers' capacity at nonpeak loads.**
- To reduce the redundancies and increase the server utilization in data centers, ASPs started **co-hosting applications** with complementary workload patterns.
- **Co-hosting of applications means deploying more than one application on a single server.**

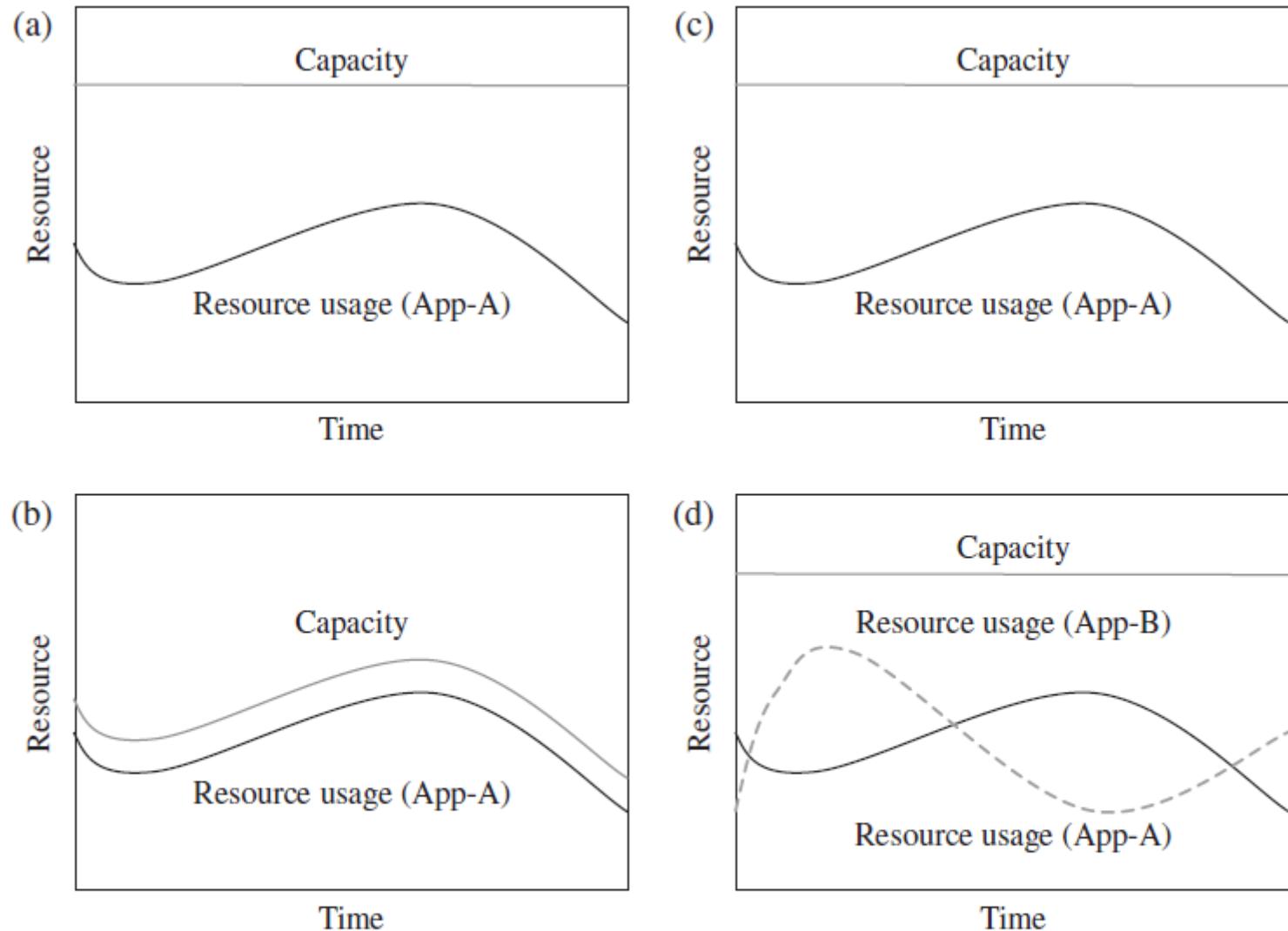


FIGURE 16.3. Service consumer and service provider perspective before and after the MSP's hosting platforms are virtualized and cloud-enabled. (a) Service consumer perspective earlier. (b) Service consumer perspective now. (c) Service provider perspective earlier. (d) Service provider perspective now.

INSPIRATION (CONTD..)

Co-Hosting disadvantages

Performance isolation:

- one application **should not steal the resources** being utilized by other co-located applications.
- For example, assume that application **A** is required to use more quantity of a resource than originally allocated to it for duration of time **t**.
- For that duration the amount of the same resource available to application **B** is decreased.
- This could adversely affect the performance of application **B**.

Performance Isolation

Security

Security

one application should not access and destroy the data and other information of co-located applications.

To handle the above issues

Virtualization technologies

INSPIRATION (CONTD..)

Virtualization technologies

- The applications, instead of being hosted on the physical machines, can be encapsulated using virtual machines.
- These virtual machines are then mapped to the physical machines.
- System resource allocation to these virtual machines can be made in two modes: **(1) conserving and (2) non-conserving**.

A virtual machine demanding more system resources (CPU and memory) than the specified quota cannot be allocated the spare resources that remain unutilized by the other co-hosted virtual machines.

Here, the spare resources that are not utilized by the co-hosted virtual machines can be used by the virtual machine needing the extra amount of resource.

TRADITIONAL APPROACHES TO SLO MANAGEMENT

First attempt to balance QoS among service-level objectives (SLOs)



TRADITIONAL APPROACHES TO SLO MANAGEMENT

Load balancing techniques

- The objective of a load balancing is to distribute the incoming requests onto a set of physical machines, each hosting a replica of an application, so that the load on the machines is equally distributed.
- The load balancing algorithm executes on a physical machine that interfaces with the clients.
- This physical machine, also called the front-end node, receives the incoming requests and distributes these requests to different physical machines for further execution.
- This set of physical machines is responsible for serving the incoming requests and are known as the back-end nodes.
-

TRADITIONAL APPROACHES TO SLO MANAGEMENT

Load balancing techniques

- Typically, the algorithm executing on the front-end node is agnostic to the nature of the request.
- This means that the front-end node is neither aware of the type of client from which the request originates nor aware of the category to which the request belongs to ----- **class agnostic**
- In **class-aware** load balancing and requests distribution, the front-end node must additionally inspect the type of client making the request and/or the type of service requested before deciding which back-end node should service the request.

TRADITIONAL APPROACHES TO SLO MANAGEMENT

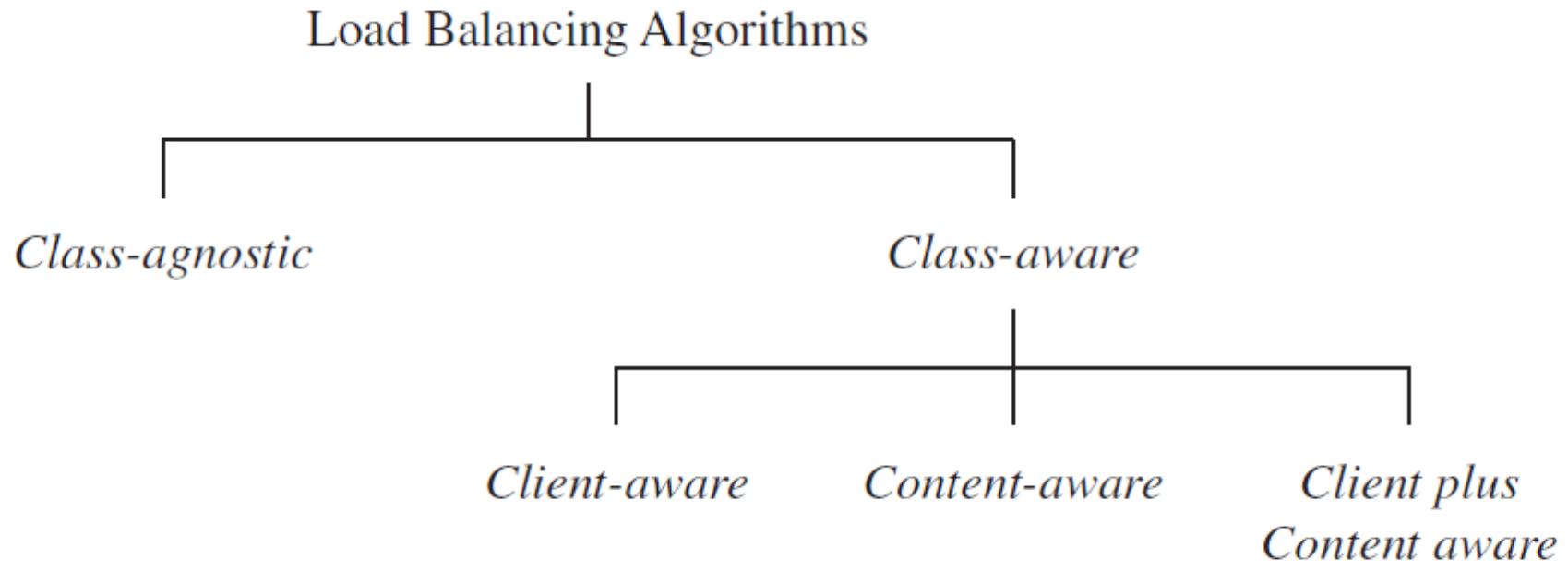


FIGURE 16.5. General taxonomy of load-balancing algorithms.

TRADITIONAL APPROACHES TO SLO MANAGEMENT

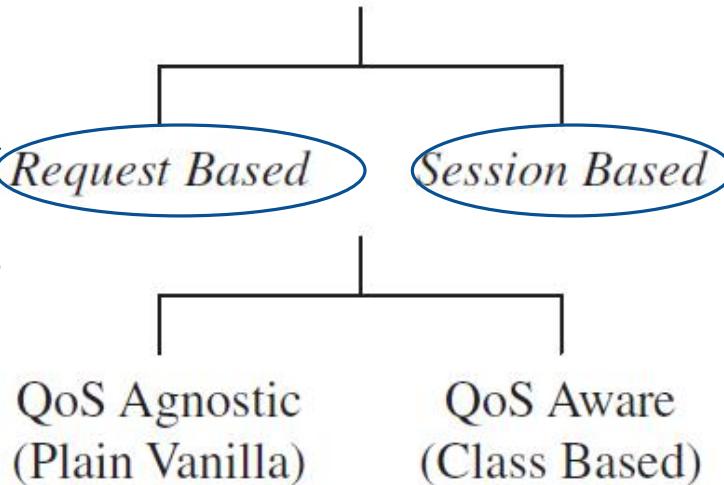
Admission Control

- Admission control algorithms play an important role in deciding the set of requests that should be admitted into the application server when the server experiences “very” heavy loads.
- The objective of admission control mechanisms, therefore, is to police the incoming requests and identify a subset of incoming requests that can be admitted into the system when the system faces overload situations.

TRADITIONAL APPROACHES TO SLO MANAGEMENT

Admission Control

Admission Control Mechanisms



Request-based admission control algorithms reject new requests if the servers are running to their capacity.

Session-based admission control mechanisms try to ensure that longer sessions are completed and any new sessions are rejected.

FIGURE 16.6. General taxonomy for admission control mechanisms.

TYPES OF SLA

Infrastructure SLA

- The infrastructure provider manages and offers guarantees on availability of the infrastructure, namely, server machine, power, network connectivity, and so on.**
- The machines are leased to the customers and are isolated from machines of other customers**

Application SLA

- In the application co-location hosting model, the server capacity is available to the applications based solely on their resource demands.**
- The service providers are flexible in allocating and de-allocating computing resources among the co-located applications.**

TYPES OF SLA

TABLE 16.1. Key Components of a Service-Level Agreement

Service-Level Parameter Metrics	<p>Describes an observable property of a service whose value is measurable.</p> <p>These are definitions of values of service properties that are measured from a service-providing system or computed from other metrics and constants. Metrics are the key instrument to describe exactly what SLA parameters mean by specifying how to measure or compute the parameter values.</p>
Function	A function specifies how to compute a metric's value from the values of other metrics and constants. Functions are central to describing exactly how SLA parameters are computed from resource metrics.
Measurement directives	These specify how to measure a metric.

TYPES OF SLA

TABLE 16.2. Key Contractual Elements of an Infrastructural SLA

<i>Hardware availability</i>	<ul style="list-style-type: none">• 99% uptime in a calendar month
<i>Power availability</i>	<ul style="list-style-type: none">• 99.99% of the time in a calendar month
<i>Data center network availability</i>	<ul style="list-style-type: none">• 99.99% of the time in a calendar month
<i>Backbone network availability</i>	<ul style="list-style-type: none">• 99.999% of the time in a calendar month
<i>Service credit for unavailability</i>	<ul style="list-style-type: none">• Refund of service credit prorated on downtime period
<i>Outage notification guarantee</i>	<ul style="list-style-type: none">• Notification of customer within 1 hr of complete downtime
<i>Internet latency guarantee</i>	<ul style="list-style-type: none">• When latency is measured at 5-min intervals to an upstream provider, the average doesn't exceed 60 msec
<i>Packet loss guarantee</i>	<ul style="list-style-type: none">• Shall not exceed 1% in a calendar month

TYPES OF SLA

TABLE 16.3. Key contractual components of an application SLA

<i>Service-level parameter metric</i>	<ul style="list-style-type: none">• Web site response time (e.g., max of 3.5 sec per user request)
<i>Function</i>	<ul style="list-style-type: none">• Latency of web server (WS) (e.g., max of 0.2 sec per request)• Latency of DB (e.g., max of 0.5 sec per query)• Average latency of WS = (latency of web server 1 + latency of web server 2) /2• Web site response time = Average latency of web server + latency of database
<i>Measurement directive</i>	<ul style="list-style-type: none">• DB latency available via http://mgmtserver/em/latency• WS latency available via http://mgmtserver/ws/instanceno/latency
<i>Service-level objective</i>	<ul style="list-style-type: none">• Service assurance
<i>Penalty</i>	<ul style="list-style-type: none">• web site latency < 1 sec when concurrent connection < 1000• 1000 USD for every minute while the SLO was breached

LIFE CYCLE OF SLA



LIFE CYCLE OF SLA (CONTD..)

Contract Definition.

- Generally, service providers define a set of service offerings and corresponding SLAs using standard templates.
- These service offerings form a catalog. Individual SLAs for enterprises can be derived by customizing these base SLA templates.

Publication and Discovery.

- Service provider advertises these base service offerings through standard publication media, and the customers should be able to locate the service provider by searching the catalog.
- The customers can search different competitive offerings and shortlist a few that fulfill their requirements for further negotiation.

LIFE CYCLE OF SLA (CONTD..)

Negotiation.

- Once the customer has discovered a service provider who can meet their application hosting need, the SLA terms and conditions needs to be mutually agreed upon before signing the agreement for hosting the application.

Operationalization.

- SLA operation consists of SLA monitoring, SLA accounting, and SLA enforcement.
- SLA monitoring involves measuring parameter values and calculating the metrics defined as a part of SLA and determining the deviations.
- On identifying the deviations, the concerned parties are notified.
- SLA accounting involves capturing and archiving the SLA adherence for compliance.

LIFE CYCLE OF SLA (CONTD..)

De-commissioning.

- SLA decommissioning involves termination of all activities performed under a particular SLA when the hosting relationship between the service provider and the service consumer has ended.
- SLA specifies the terms and conditions of contract termination and specifies situations under which the relationship between a service provider and a service consumer can be considered to be legally ended.

SLA MANAGEMENT IN CLOUD

SLA management of applications hosted on cloud platforms involves five phases.

- 1. Feasibility**
- 2. On-boarding**
- 3. Pre-production**
- 4. Production**
- 5. Termination**

SLA MANAGEMENT IN CLOUD

(CONTD..)

Feasibility Analysis

- MSP conducts the feasibility study of hosting an application on their cloud platforms.
- This study involves three kinds of feasibility:
(1) technical feasibility, (2) infrastructure feasibility, (3) financial feasibility.
- The technical feasibility of an application implies determining the following:
 1. Ability of an application to scale out.
 2. Compatibility of the application with the cloud platform.
 3. The need and availability of a specific hardware and software required for hosting and running of the application.
 4. Preliminary information about the application performance.

SLA MANAGEMENT IN CLOUD

(CONTD..)

On-Boarding of Application

On-boarding activity consists of the following steps:

- a. Packing of the application for deploying on physical or virtual environments.
- b. The packaged application is executed directly on the physical servers to capture and analyze the application performance characteristics.
- c. The application is executed on a virtualized platform and the application performance characteristics are noted again.
- d. Based on the measured performance characteristics, different possible SLAs are identified.
- e. Once the customer agrees to the set of SLOs and the cost, the MSP starts creating different policies required by the data center for automated management of the application.
- f. Types of Policies: (1) business, (2) operational, and (3) provisioning

SLA MANAGEMENT IN CLOUD

(CONTD..)

Preproduction:

- Once the determination of policies is completed as discussed in previous phase, the application is hosted in a simulated production environment.
- It facilitates the customer to verify and validate the MSP's findings on application's runtime characteristics and agree on the defined SLA.
- Once both parties agree on the cost and the terms and conditions of the SLA, the customer sign-off is obtained.
- On successful completion of this phase the MSP allows the application to go on-live.

SLA MANAGEMENT IN CLOUD

(CONTD..)

Production:

- The application is made accessible to its end users under the agreed SLA.
- Additionally, customer may request the MSP for inclusion of new terms and conditions in the SLA.

Termination:

- When the customer wishes to withdraw the hosted application and does not wish to continue the termination activity is initiated.
- On initiation of termination, all data related to the application are transferred to the customer and only the essential information is retained for legal compliance.
- This ends the hosting relationship between the two parties for that application, and the customer sign-off is obtained.

NEXT CLASS....

CLOUD SECURITY

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L34-36

CLOUD BREACHES OF 2021

Accenture. In August of 2021, Accenture fell prey to a LockBit ransomware attack. The culprits claimed to have stolen 6TB worth of data, for which they requested a ransom of \$50 million.

Kaseya. In July of 2021, IT solutions provider Kaseya identified an attack on their unified remote monitoring and network perimeter security tool. The attackers aimed to steal administrative control for Kaseya services; from managed service providers to downstream customers.

Cognyte. In May of 2021, the cyber analytics firm Cognyte left a database unsecured without authentication protocols. In turn, hackers managed to expose 5 billion records. Information such as names, email addresses, passwords, and vulnerability data points within their system were leaked.

Facebook. In April of 2021, Facebook reported a breach affecting hundreds of millions of user records, which were publicly exposed on Amazon's cloud computing service. Although Facebook confirmed that it identified and resolved the issue immediately, the attack managed to impact founder Mark Zuckerberg.

Raychat. In February of 2021, Raychat, an online chat application, survived a large-scale cyber attack. A cloud database configuration breach gave hackers free access to 267 million usernames, emails, passwords, metadata and encrypted chats.

THREE BASIC COMPONENTS OF SECURITY



Confidentiality

Keep data and Resources hidden



Integrity

Data Integrity (Integrity)

Origin integrity (Authentication)



Availability

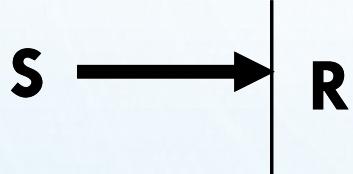
Enabling access to data and resources

SECURITY ATTACKS

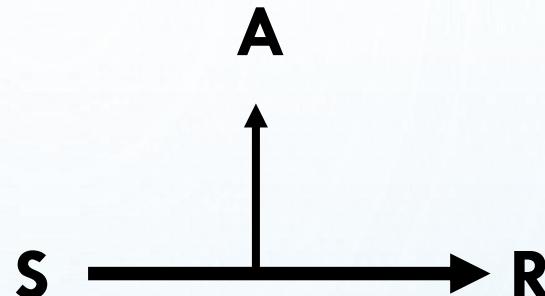
- Any action that compromises the security of information
- Four types of attack
 - **Interruption** - Obstruction during the communication process between the systems.
 - **Interception** - The data or message which is sent by the sender is stolen by an unauthorized individual where the message will be changed to the different form or it will be used by the malicious process
 - **Modification** - Message which is sent by the sender is modified and sent to the destination by an unauthorized user.
 - **Fabrication** - A fake message is inserted into the network by an unauthorized user as if it is a valid user.

BASIC MODEL

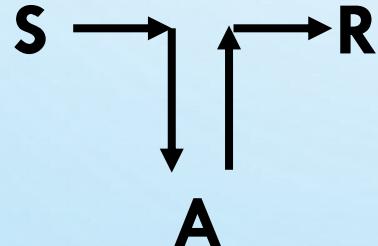
**Interruption –
Attack on
availability**



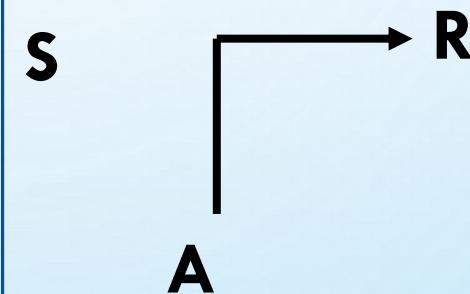
**Interception –
Attack on
Confidentiality**



**Modification –
Attack on
Integrity**



**Fabrication –
Attack on
Authenticity**



KEY TERMS

- **Threat** – is any potential occurrence, malicious or otherwise, that could harm an asset. In other words, a threat is any bad thing that can happen to your assets.
- **Vulnerability** – is a weakness that makes a threat possible. This may be because of poor design, configuration mistakes, or inappropriate and insecure coding techniques.
- **Attack** – is an action that exploits a vulnerability or enacts a threat. Examples of attacks include sending malicious input to an application or flooding a network in an attempt to deny service.

What are the differences between Attack and Threat?

CLASSES OF THREATS

- **Disclosure**
 - Snooping*
- **Deception**
 - Modification, Spoofing**, Repudiation of origin, Denial of receipt
- **Disruption**
 - Modification
- **Usurpation*****
 - Modification, Spoofing, Delay, Denial of service

*snooping is unauthorized access to another person's or company's data

**Spoofing is pretends to be something else in an attempt to gain the confidence to access the system/service

***Unauthorized control of some part of a system

GOALS OF SECURITY

- **Prevention**

- Prevention attack from violating the security policies

- **Detection**

- Prevention attackers who violating the security policies

- **Recovery**

- Stop attack, asses and repair damage
 - Continue to function correctly even if attack succeeds

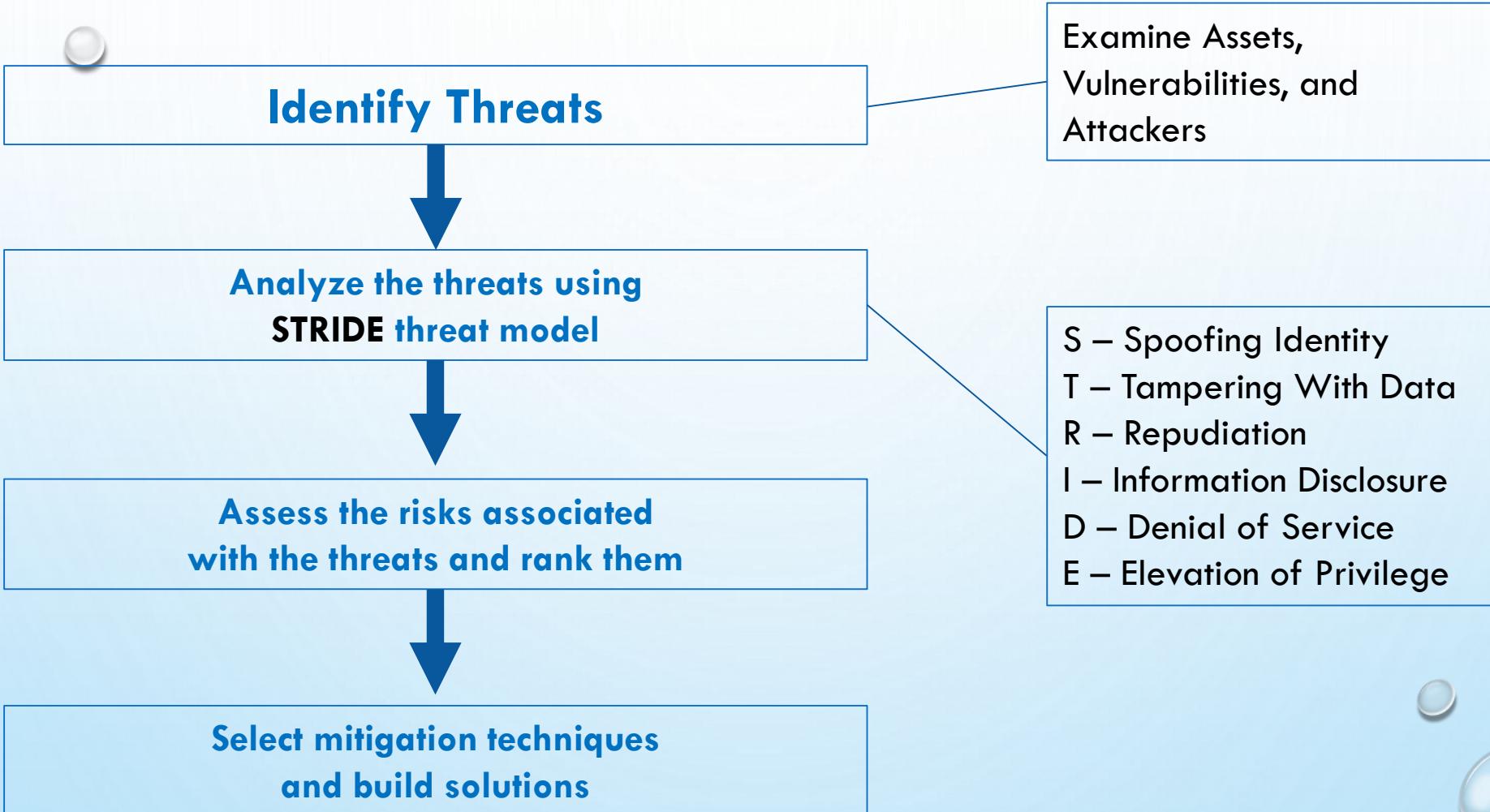
WHAT IS CLOUD SECURITY?

- Cloud security is the protection of data stored online via cloud computing platforms from theft, leakage, and deletion.
- Methods of providing cloud security include
 - Firewalls
 - Penetration testing
 - Tokenization, virtual private networks (VPN)
 - Avoiding public internet connections.
- Cloud security is a form of cybersecurity.

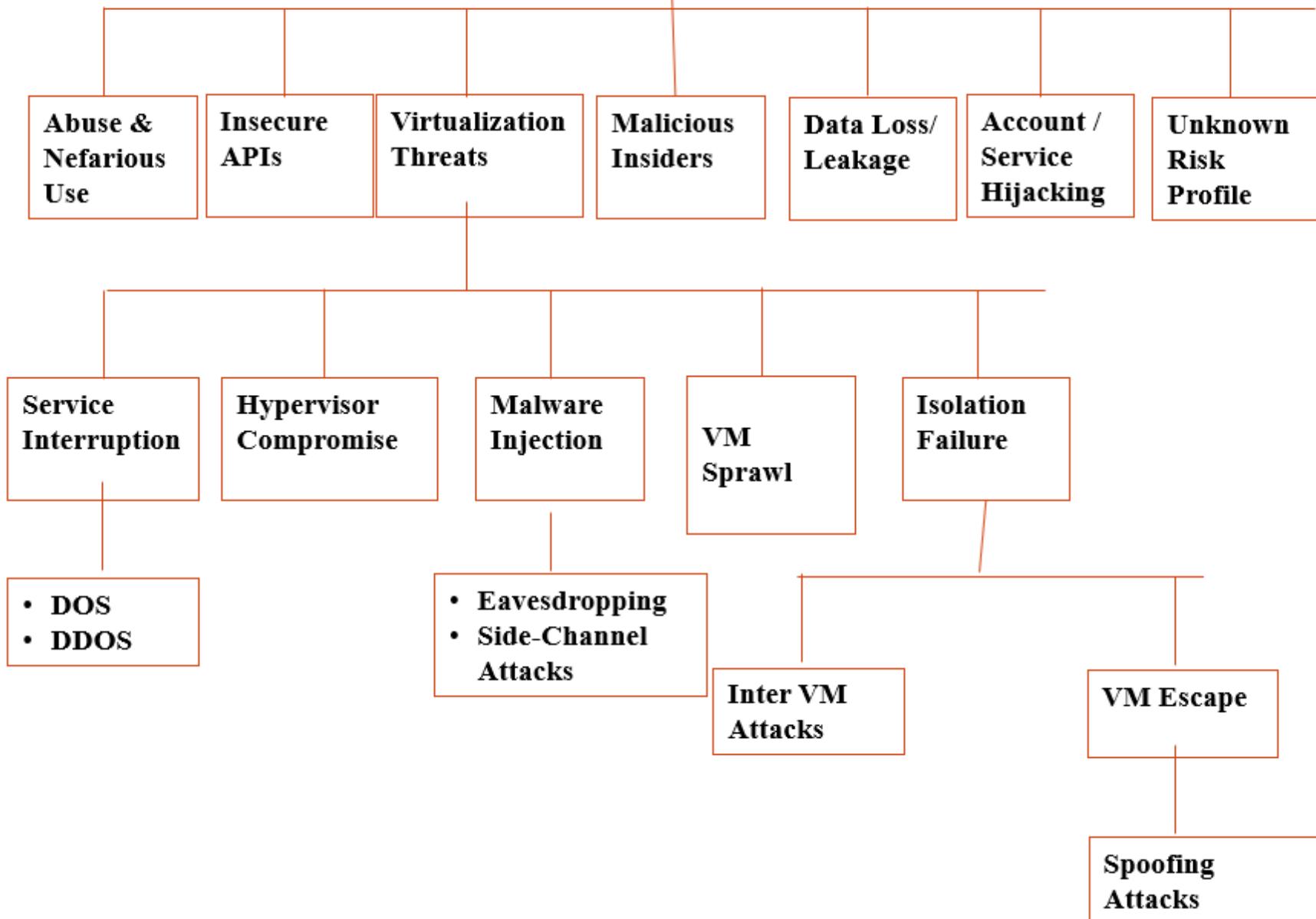
MOTIVATION

- **Cloud provider** - provides resources/services that can be accessed from anywhere in the world by the user.
- **Cloud user** - can be either a single person or any organization.
- Cloud computing has many benefits, but security is a big concern.
- Cloud provider should provide privacy and security to the user's data and applications.

THREAT MODEL



Cloud Specific Threats



- **THREE STAGES OF SECURITY**
- **DATA SECURITY**
- **NETWORK SECURITY**
- **HOST SECURITY**

DATA SECURITY

- Physical storage security defines how you control physical access to the servers that support your infrastructure.
- The cloud still has physical security constraints as there are actual servers running somewhere.
- When selecting a cloud provider, you should understand their physical security protocols.
- To be studied under.....
 - Data Control
 - Encrypt Everything
 - Regulatory and Standards Compliance

DATA CONTROL

- The main practical problem is that, factors that have nothing to do with your business can compromise your operations and your data.
- Some of the events are:
 - The cloud provider declares bankruptcy and its servers are seized or it ceases operations.
 - A third party with no relationship to you (or, worse, a competitor) sues your cloud provider and obtains a blanket subpoena granting access to all servers owned by the cloud provider.
 - Failure of your cloud provider to properly secure portions of its infrastructure—especially in the maintenance of physical access controls—results in the compromise of your systems.

DATA CONTROL (CONTD..)

- The solution is to do two things encrypt everything and keep off-site backups.
 - Encrypt sensitive data in your database and in memory. Decrypt it only in memory for the duration of the need for the data. Encrypt your backups and encrypt all network communications.
 - Choose a second provider and use automated, regular backups (for which many open source and commercial solutions exist) to make sure any current and historical data can be recovered even if your cloud provider were to disappear from the face of the earth.

ENCRYPT EVERYTHING

- In the cloud, your data is stored somewhere; you just don't know exactly where. However, you know some basic parameters:
 - Your data lies within a virtual machine guest operating system, and you control the mechanisms for access to that data.
 - Network traffic exchanging data between instances is not visible to other virtual hosts.
 - For most cloud storage services, access to data is private by default. Many, including Amazon S3, nevertheless allow you to make that data public.

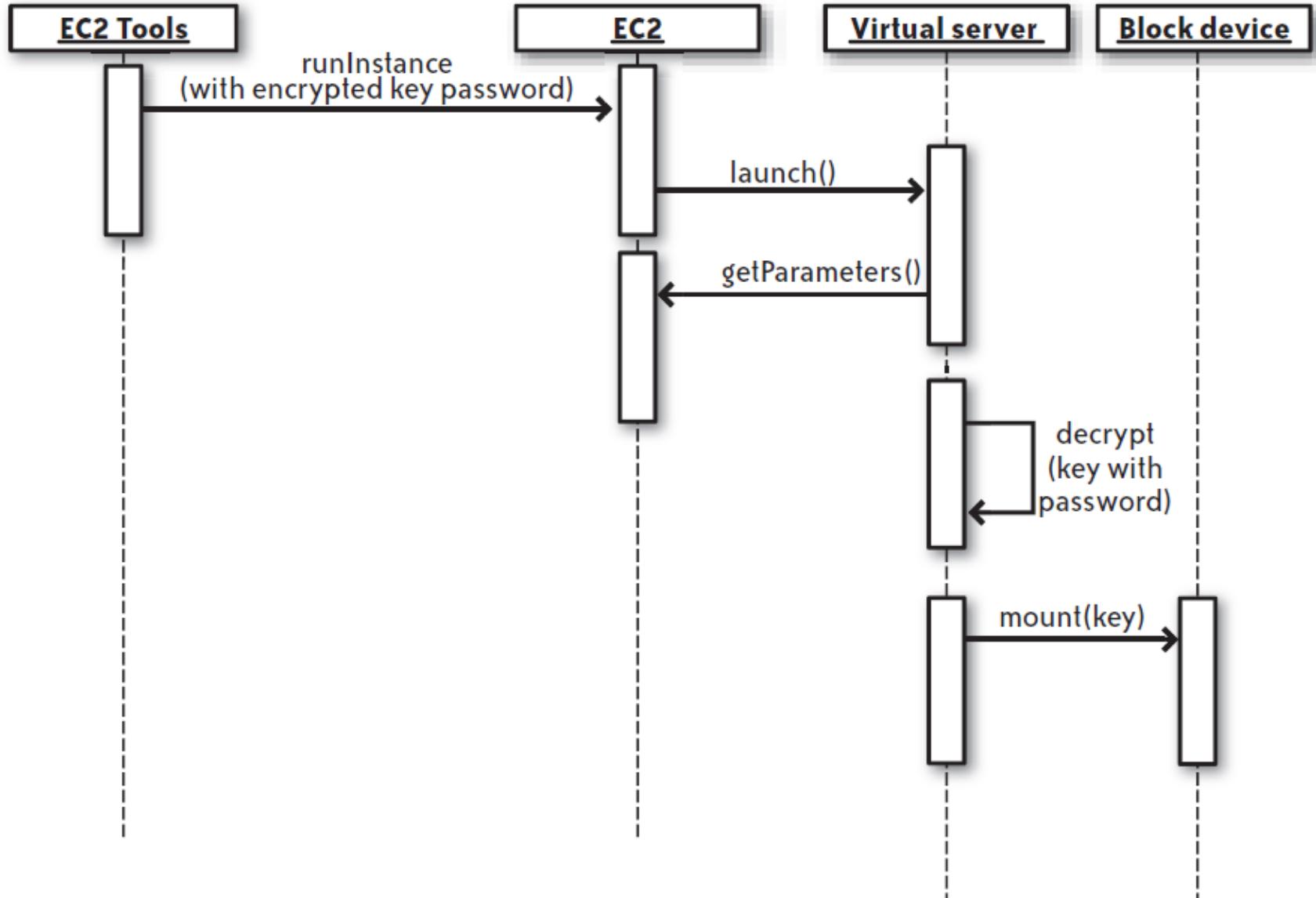


FIGURE 5-1. The process of starting a virtual server with encrypted filesystems

REGULATORY AND STANDARDS COMPLIANCE

From a security perspective, you'll encounter three kinds of issues in standards and regulations:

“How” issues

These result from a standard such as PCI or regulations such as HIPAA or SOX, which govern how an application of a specific type should operate in order to protect certain concerns specific to its problem domain. For example, HIPAA defines how you should handle personally identifying health care data.

“Where” issues

These result from a directive such as Directive 95/46/EC that governs where you can store certain information. One key impact of this particular directive is that the private data on EU citizens may not be stored in the United States (or any other country that does not treat private data in the same way as the EU).

“What” issues

These result from standards prescribing very specific components to your infrastructure. For example, PCI prescribes the use of antivirus software on all servers processing credit card data.

NETWORK SECURITY

Amazon's cloud has no perimeter. Instead, EC2 provides security groups that define firewall-like traffic rules governing what traffic can reach virtual servers in that group due to reasons such as:

- Two servers in two different Amazon EC2 availability zones can operate in the same security group.
- A server may belong to more than one security group.
- Servers in the same security group may not be able to talk to each other at all.
- Servers in the same network segment may not share any IP characteristics—they may even be in different class address spaces.
- No server in EC2 can see the network traffic bound for other servers (this is not necessarily true for other cloud systems). If you try placing your virtual Linux server in promiscuous mode, the only network traffic you will see is traffic originating from or destined for your server.

FIREWALL RULES (CONTD..)

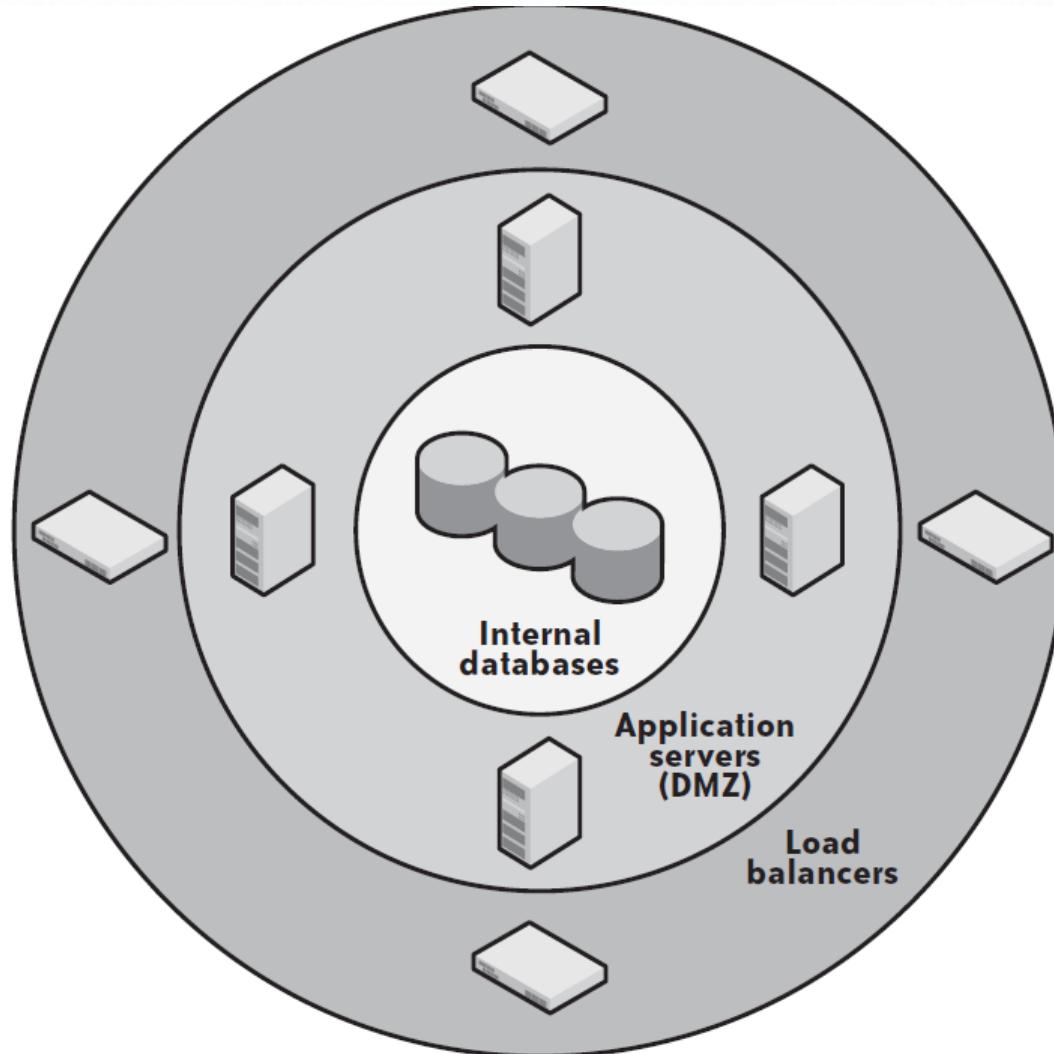


FIGURE 5-2. Firewalls are the primary tool in perimeter security

FIREWALL RULES

- Typically, a firewall protects the perimeter of one or more network segments.
- A main firewall protects the outermost perimeter, allowing only HTTP, HTTPS, and (sometimes) FTP* traffic.
- Within that network segment are border systems, such as load balancers, that route traffic into a DMZ protected by another firewall.
- Finally, within the DMZ are application servers that make database and other requests across a third firewall into protected systems on a highly sensitive internal network.
- This structure requires you to move through several layers—or perimeters—of network protection in the form of firewalls to gain access to increasingly sensitive data.
- The perimeter architecture's chief advantage is that a poorly structured firewall rule on the inner perimeter does not accidentally expose the internal network to the Internet unless the DMZ is already compromised.

FIREWALL RULES_(CONTD..)

- You can set up security groups to help you mimic traditional perimeter security.
- For example, you can create the following:
 - A border security group that listens to all traffic on ports 80 and 443
 - A DMZ security group that listens to traffic from the border group on ports 80 and 443
 - An internal security group that listens to traffic on port 3306 from the DMZ security group

FIREWALL RULES_(CONTD..)

Two other advantages of this security architecture are the following:

- Because you control your firewall rules **remotely**, an intruder does not have a single target to attack, as he does with a physical firewall.
- You don't have the opportunity to accidentally destroy your network rules and thus permanently remove everyone's access to a given network segment.

FIREWALL RULES_(CONTD..)

A few best practices for your network security include:

- Run only one network service (plus necessary administrative services) on each virtual server
- Do not open up direct access to your most sensitive data
- Open only the ports absolutely necessary to support a server's service and nothing more
- Limit access to your services to clients who need to access them
- Even if you are not doing load balancing, use a reverse proxy
- Use the dynamic nature of the cloud to automate your security embarrassments

NETWORK INTRUSION DETECTION

- Perimeter security often involves network intrusion detection systems (NIDS), such as Snort, which monitor local traffic for anything that looks irregular.
- Examples of irregular traffic include:
 - Port scans
 - Denial-of-service attacks
 - Known vulnerability exploit attempts

The purpose of a network intrusion detection system

Network intrusion detection exists **to alert you** of attacks before they happen and, in some cases, foil attacks as they happen.

NIDS (CONTD..)

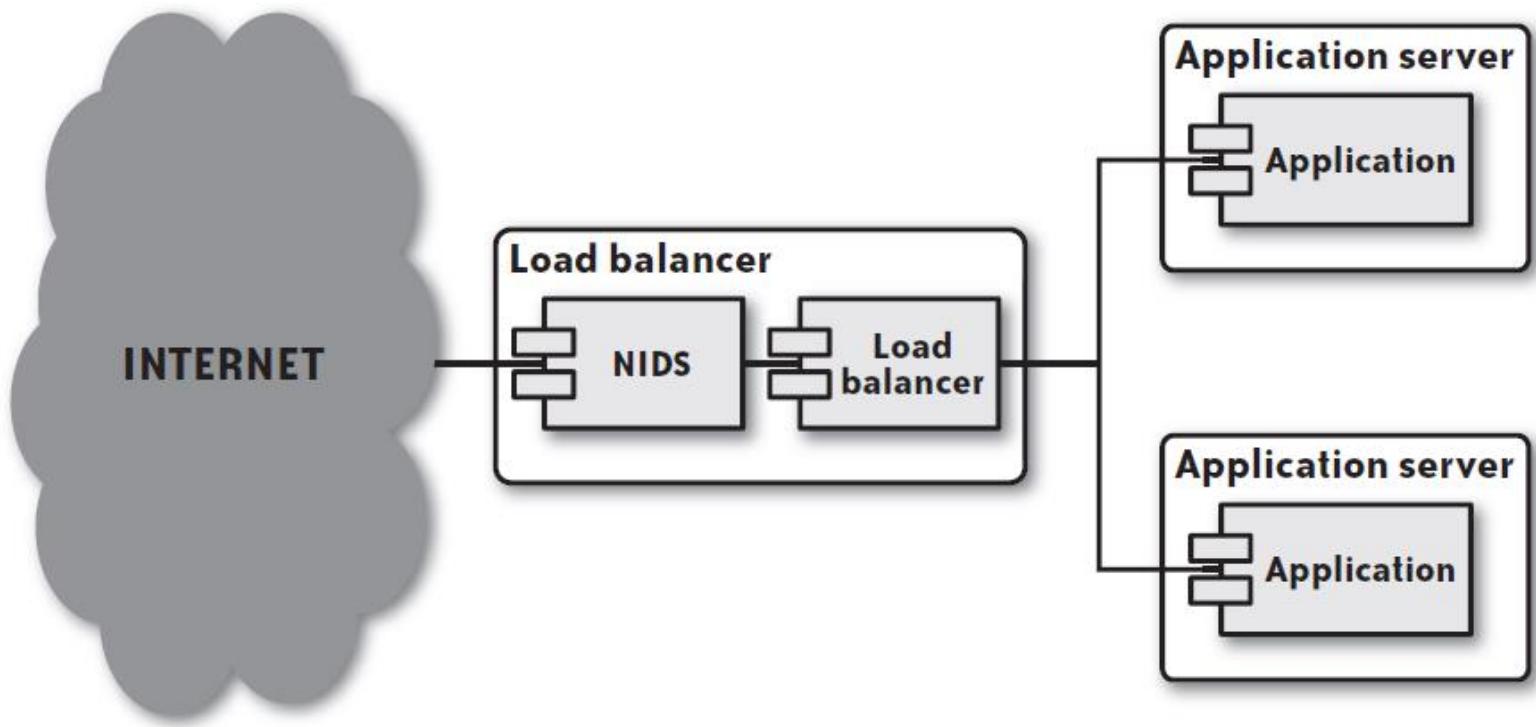


FIGURE 5-4. A network intrusion detection system listening on a load balancer

HOST SECURITY

Host security describes how your server is set up for the following tasks:

- Preventing attacks.
- Minimizing the impact of a successful attack on the overall system.
- Responding to attacks when they occur.

Given the assumption that your services are vulnerable, your most significant tool in preventing attackers from exploiting a vulnerability once it becomes known is the rapid rollout of security patches.

Three steps of rollout patch:

1. Patch your AMI with the new security fixes.
2. Test the results.
3. Relaunch your virtual servers.

SYSTEM HARDENING

Machine hardening is pre-processing of the machine imaging phase.

A hardened system should meet the following criteria:

- No network services are running except those necessary to support the server's function.
- No user accounts are enabled on the server except those necessary to support the services running on the server or to provide access for users who need it.
- All configuration files for common server software are configured to the most secure settings.
- All necessary services run under a nonprivileged role user account (e.g., run MySQL as the mysql user, not root).
- When possible, run services in a restricted filesystem, such as a chroot jail.

HOST INTRUSION DETECTION

- A network intrusion detection system monitors network traffic for suspicious activity, a host intrusion detection system (HIDS) such as OSSEC monitors the state of your server for anything unusual.
- OSSEC has two configuration profiles:
 - Standalone, in which each server scans itself and sends you alerts.
 - Centralized, in which you create a centralized HIDS server to which each of the other servers sends reports.

HOST INTRUSION DETECTION

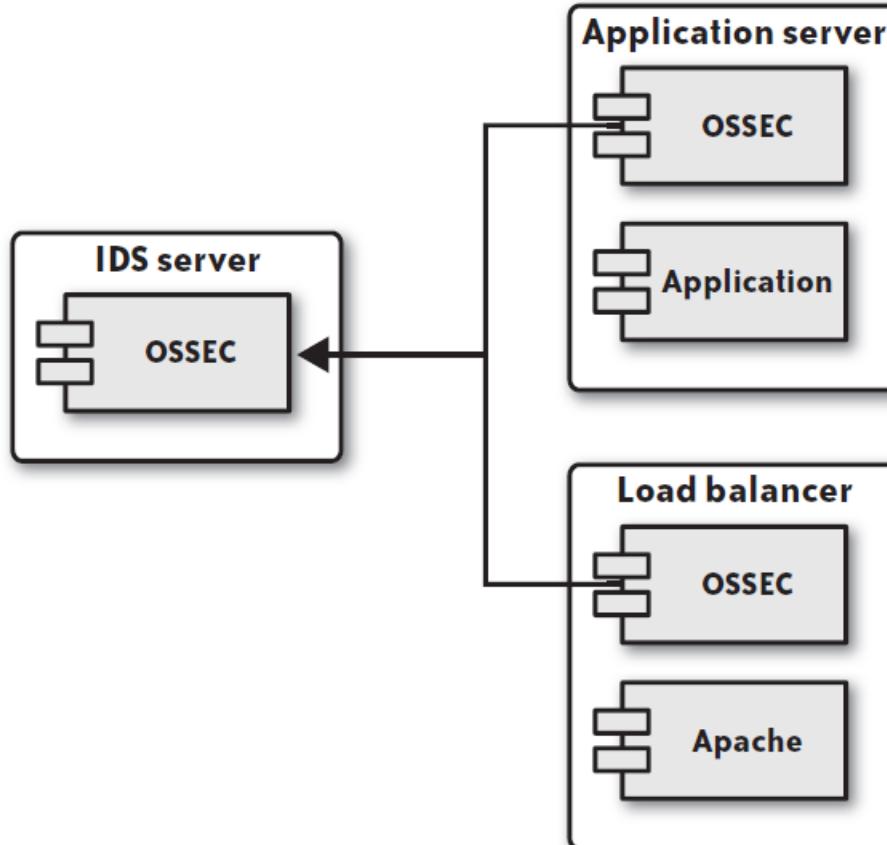


FIGURE 5-5. A HIDS infrastructure reporting to a centralized server

DATA SEGMENTATION

- In addition to assuming that the services on your servers have security exploits, you should further assume that eventually one of them will be compromised.
- Obviously, you never want any server to be compromised. The best infrastructure, however, is tolerant of—in fact, it assumes—the compromise of any individual node.
- This tolerance is not meant to encourage lax security for individual servers, but is meant to minimize the impact of the compromise of specific nodes.
- Making this assumption provides you with a system that has the following advantages:
 - Access to your most sensitive data requires a full system breach.
 - The compromise of the entire system requires multiple attack vectors with potentially different skill sets.
 - The downtime associated with the compromise of an individual node is negligible or non-existent.
- The segmentation of data based on differing levels of sensitivity is your first tool in minimizing the impact of a successful attack.

CREDENTIAL MANAGEMENT

- Your machine images OSSEC profile should have no user accounts embedded in them.
- In fact, you should never allow password-based shell access to your virtual servers.
- The most secure approach to providing access to virtual servers is the dynamic delivery of public SSH keys to target servers.
- In other words, if someone needs access to a server, you should provide her credentials to the server when it starts up or via an administrative interface instead of embedding that information in the machine image.

COMPROMISED RESPONSE

Responding to attack is a major step to be taken by digital forensic professionals. When you detect a compromise on a physical server, the standard operating procedure is a painful, manual process:

1. Remove intruder access to the system, typically by cutting the server off from the rest of the network.
2. Identify the attack vector. You don't want to simply shut down and start over, because
3. the vulnerability in question could be on any number of servers. Furthermore, the intruder very likely left a rootkit or other software to permit a renewed intrusion after you remove the original problem that let him in. It is therefore critical to identify how the intruder compromised the system, if that compromise gave him the ability to compromise other systems, and if other systems have the same vulnerability.
4. Wipe the server clean and start over. This step includes patching the original vulnerability and rebuilding the system from the most recent uncompromised backup.
5. Launch the server back into service and repeat the process for any server that has the same attack vector.