



SITA1503 FOG AND CLOUD COMPUTING

UNIT -II



COURSE OBJECTIVES

- To understand the basic concepts of cloud computing and cloud enables.
- To understand cloud services and Multi-tenancy computing.
- To study about various models of cloud environments and virtualization.
- To manage the cloud computing infrastructure with security.
- To gain knowledge of cloud and to understand about Fog, edge computing.



UNIT 2 CLOUD SERVICE MODELS

- Software as a Service (SaaS)
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Web services
- Service Oriented Architecture (SoA)
- Elastic Computing
- On Demand Computing
- Service Management in Cloud Computing
- Multi-tenancy computing , architecture

Software As A Service

Contents

- Introduction
- Characteristics of SaaS
- Types of SaaS
- Architecture of SaaS
- Software as a service providers
- Advantages
- Disadvantages
- Conclusion

- What is a cloud?
- What is cloud computing?
- What are cloud computing Applications?
- What are services of cloud computing?



Applications

- Google (Google Apps)
- Salesforce.com
- Microsoft (OfficeLive)
- Yahoo (Zimbra)
- Concur
- Taleo
- NetSuite

Platforms

- Google- App Engine
- Salesforce.com (Force.com)
- Microsoft (Windows Live)
- Amazon (EC2)
- NetSuite

Google
Microsoft

Infrastructure

- Google
- Microsoft
- Amazon (S3)
- IBM (Blue cloud)
- EMC
- Sun (Project Carline)

Enablers : VMWare, Citrix, Akamai, Dell, HP, EMC, NetApp, Sun, IBM, Adobe



SaaS

(Software as a Service)

Applications, typically available via the browser:

- Google Apps
- Salesforce.com

PaaS

(Platform as a Service)

Hosted application environment for building and deploying cloud applications:

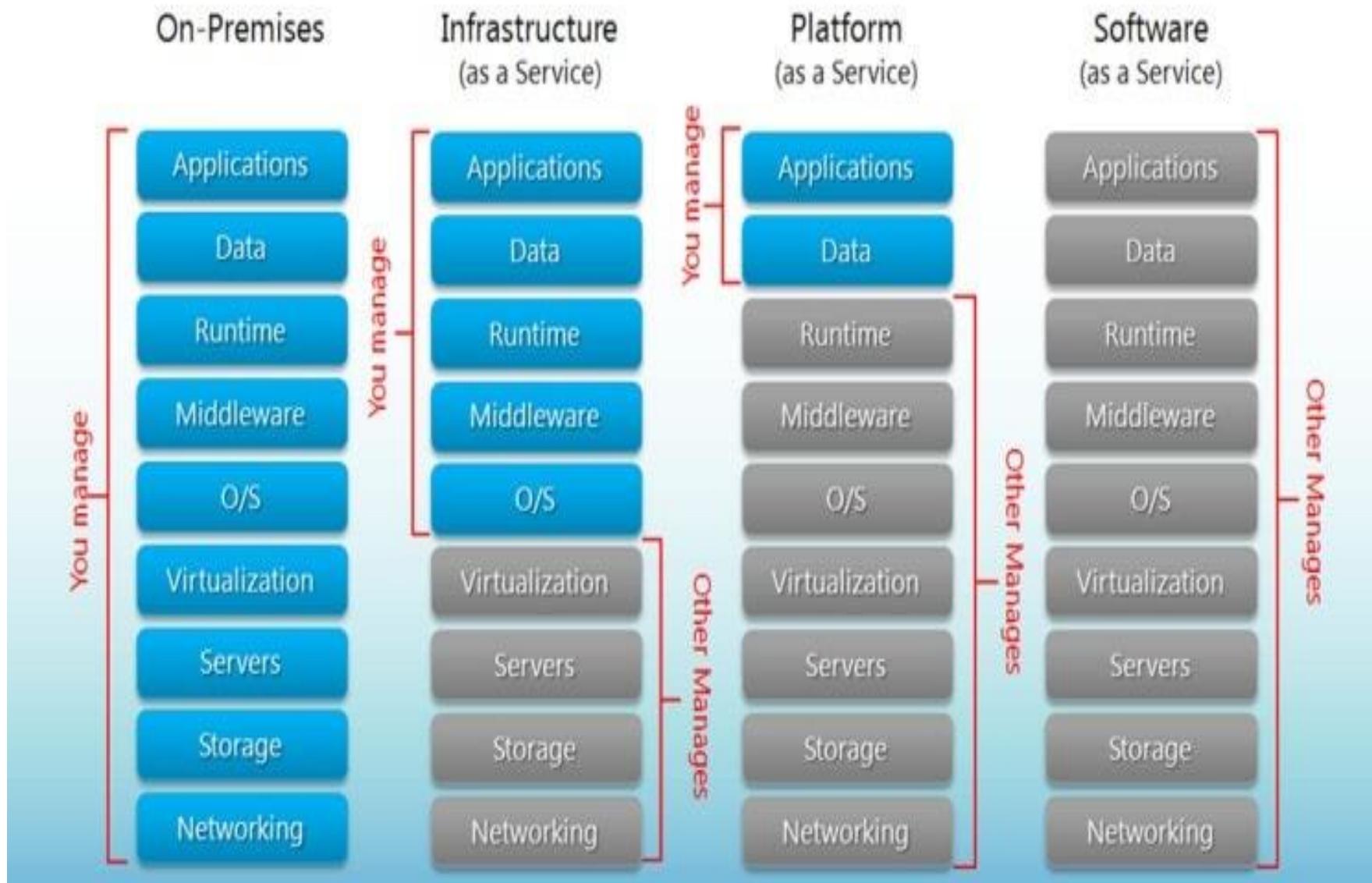
- Salesforce.com
- Amazon E2C
- Microsoft Azure

IaaS

(Infrastructure as a Service)

Utility computing data center providing on demand server resources:

- HP Adaptive Infrastructure as a Service
- Rackspace
- Amazon E2C & S3



History of SaaS

- The term “SaaS” or software-as-a-service was first mentioned in an internal document called “Software as a Service: Strategic Backgrounder” by the Software & Information Industry Association in 2000.

Introduction to SaaS

Software as a Service (SaaS), on demand software, is a software delivery model in which software and its associated data are hosted **centrally** and accessed using a thin-client, usually a web browser over the internet.

Eg

- Gmail(Google)
- Fb

Characteristics of SaaS

- Easy customization
- Better access
- Multitenant architecture
- Integration
- Operational performance

characteristics

- **Multitenancy:** This is an architecture whereby a single instance of a software app is meant to serve multiple customers.
- **Security and Compliance:** The responsibility of the SaaS application and hardware security solely lies with the application developer. He should ensure that multiple tenants on a shared infrastructure have their data virtually segregated and made accessible to those individuals intended to access it.



Service Offered by SaaS

Enterprise Service	Web 2.0
Workflow Management	Desktop software
Groupware & Collaborative	Financial Management
Supply chain	Geo Spatial & Search
Digital Signation	Blogs
CRM	Wiki Services
Communications	Portal Services
	Social Networking
	Meta data Management

Types of SaaS

- There are two types of SaaS

1. The Ups of SaaS

- ❖ No need to buy the applications, deploy them and maintain them
- ❖ No need to maintain the equipment and infrastructure.
- ❖ Lower costs

2. The Downs of SaaS

- ❖ Dependency on the SaaS provider
- ❖ Security issues
- ❖ Limited applications

Architecture of SaaS

Application design for SaaS or cloud infrastructure requires a number of key considerations to leverage the benefits delivered by the cloud. The design of the traditional applications considered limited resources.

Architecture of SaaS

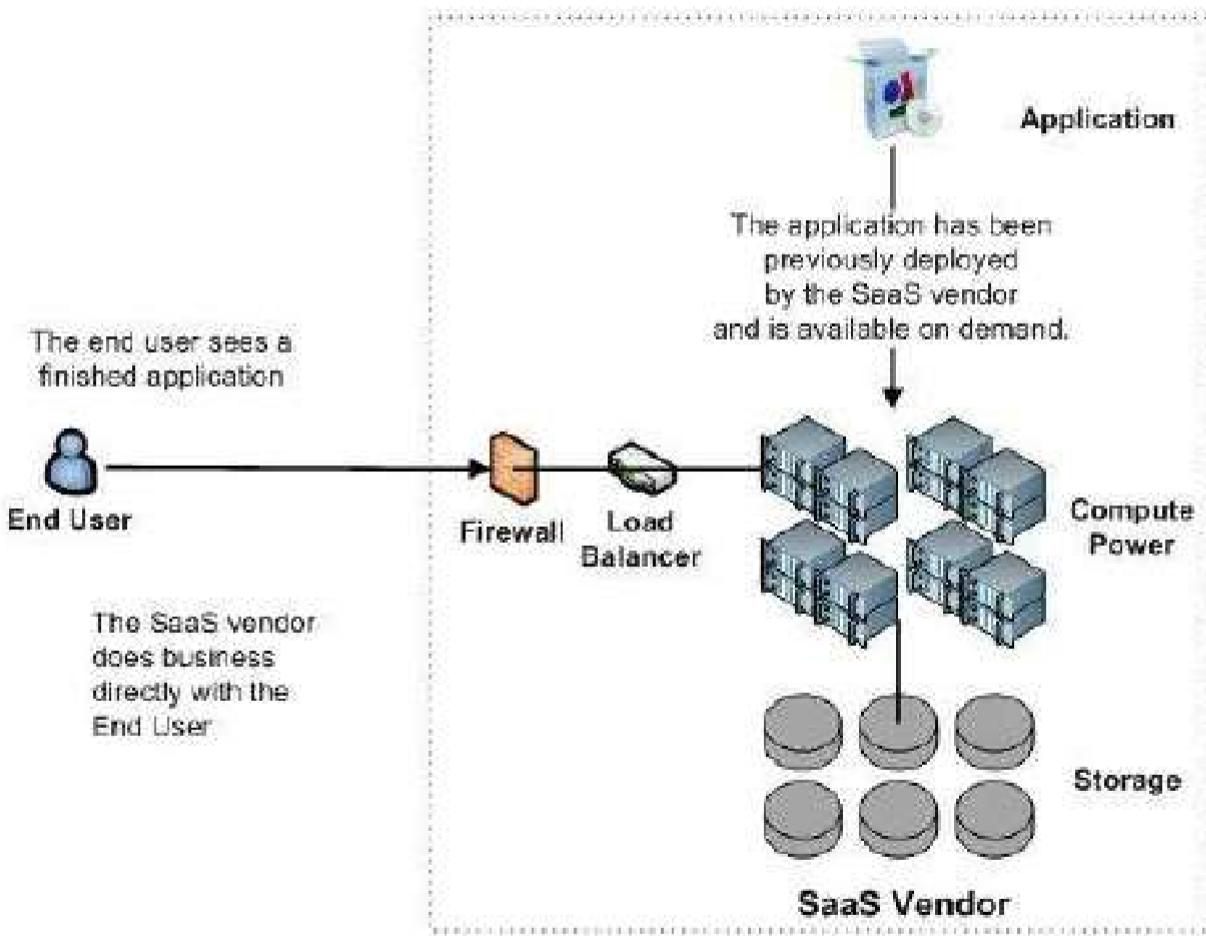
□ Vertical SaaS

A Software which answers the needs of a specific industry (e.g., software for the healthcare, agriculture, real estate, finance industries)

□ Horizontal SaaS

The products which focus on a software category (marketing, sales, developer tools, HR) but are industry agnostic

Software as a Service (SaaS)



**Business
Model**

**Software
Services**

**Application
Architecture**

**Operational
Structure**

SaaS providers

- SalesForce
- LinkedIn
- Workday
- Netsuite
- ServiceNow
- Concur Technologies

Advantages

- Easy to use
- Scalability
- Vendor accountability

Disadvantages

- Robustness
- Privacy
- Security
- Reliability
- Compliance
- Performance

Conclusion

- Use of SaaS services is pervasive in the development and deployment of modern applications
- In conclusion, the SaaS methodology is very mature and would play a central role to the future of computing.



Platform as a Service(PaaS)





Platform-as-a-Service (PaaS)
is a cloud delivery model for
applications composed of
services managed by a 3rd
party.



PaaS





PaaS in Cloud Computing

- Includes services for developing and deploying applications.
- It is an abstracted and integrated cloud-based computing environment that supports the development, running, and management of applications.
- A primary value of a PaaS environment is that developers don't have to be concerned with some of the lower-level details of the environment.





What is PaaS?

- PaaS, is a category of **cloud computing** that provides a **platform** and environment to allow developers to build applications and services over **the Internet**.
- A “**cloud aware**” **application development and deployment** environment.
- An **abstraction layer** between your cloud application and your IaaS provider.



What is PaaS?

- **PaaS services** are hosted in the **cloud** and accessed by users simply via their web browser.
- Fundamentally provides **elastic scaling of your application**.
- Deployments include **public**, **private** and **hybrid** cloud configurations.
- Example PaaS services:
 - Data services,
 - Application runtime,
 - Messaging & queueing,
 - Application management.



PaaS

- Platform as a Service (PaaS) is a way to rent hardware, operating systems, storage and network capacity over the Internet.
- The service delivery model allows the customer to rent virtualized servers and associated services for running existing applications or developing and testing new ones.
- It is an outgrowth of SaaS, a software distribution model in which hosted software applications are made available to customers over the Internet.



PaaS Goal

- Create an abstracted environment that supports an efficient, cost-effective, and repeatable process for the creation and deployment of high-quality applications.
- Focus on Development, not Ops
 - Programmers' development environment
 - Presentation layer: HTML, CSS, JavaScript
 - Control layer: Web Server code
 - Data layer: Data Model
 - Optionally, analytics



PaaS Goal

- **Cloud operations (CloudOps)** is the management, delivery and consumption of software in a computing environment where there is limited visibility into an app's underlying infrastructure
- Ops below
 - Made visible through a web interface
 - configuration management
 - resource allocation
 - optimizing performance capacity;
 - managing compliance burden
 - Operating system
 - File system
 - User authentication
 - Utilities (cron, etc.)
 - Logs
 - Database maintenance, backups, etc.



How PaaS works

- Platform as a Service allows users to create software applications using tools supplied by the provider.
- PaaS services can consist of preconfigured features that customers can subscribe to; they can choose to include the features that meet their requirements while discarding those that do not.
- The infrastructure and applications are managed for customers and support is available.
- Services are constantly updated, with existing features upgraded and additional features added.



PaaS does that

- PaaS enables an organization to do the following:
- **Leverage key middleware services** without having to deal with the complexities of managing individual hardware and software elements.
- Access a complete stack of development and deployment tools via a web browser, a middleware environment where APIs can be used to plug into selected development and deployment tools. A developer might also leverage a full desktop development environment.
- Overcome the challenges of managing lots of individual development and deployment tools by providing a suite of integrated and standardized tools — operating systems, security products, and the like — that meet company requirements.



Payment

- As with most cloud offerings, PaaS services are generally **paid for on a subscription basis** with clients ultimately paying just for what they use.
- Clients also benefit from the economies of scale that arise from the sharing of the underlying physical infrastructure between users, and that results in lower costs.



PaaS offers

- Below are some of the features that can be included with a PaaS offering:
 - Operating system
 - Server-side scripting environment
 - Database management system
 - Server Software
 - Support
 - Storage
 - Network access
 - Tools for design and development & Hosting



Who can use PaaServices?

- Software developers
- web developers and
- businesses can benefit from PaaS.
- For example, web developers can use individual PaaS environments at every stage of the process to develop, test and ultimately host their websites. However, businesses that are developing their own internal software can also utilise Platform as a Service, particularly to create distinct ring-fenced development and testing environments.

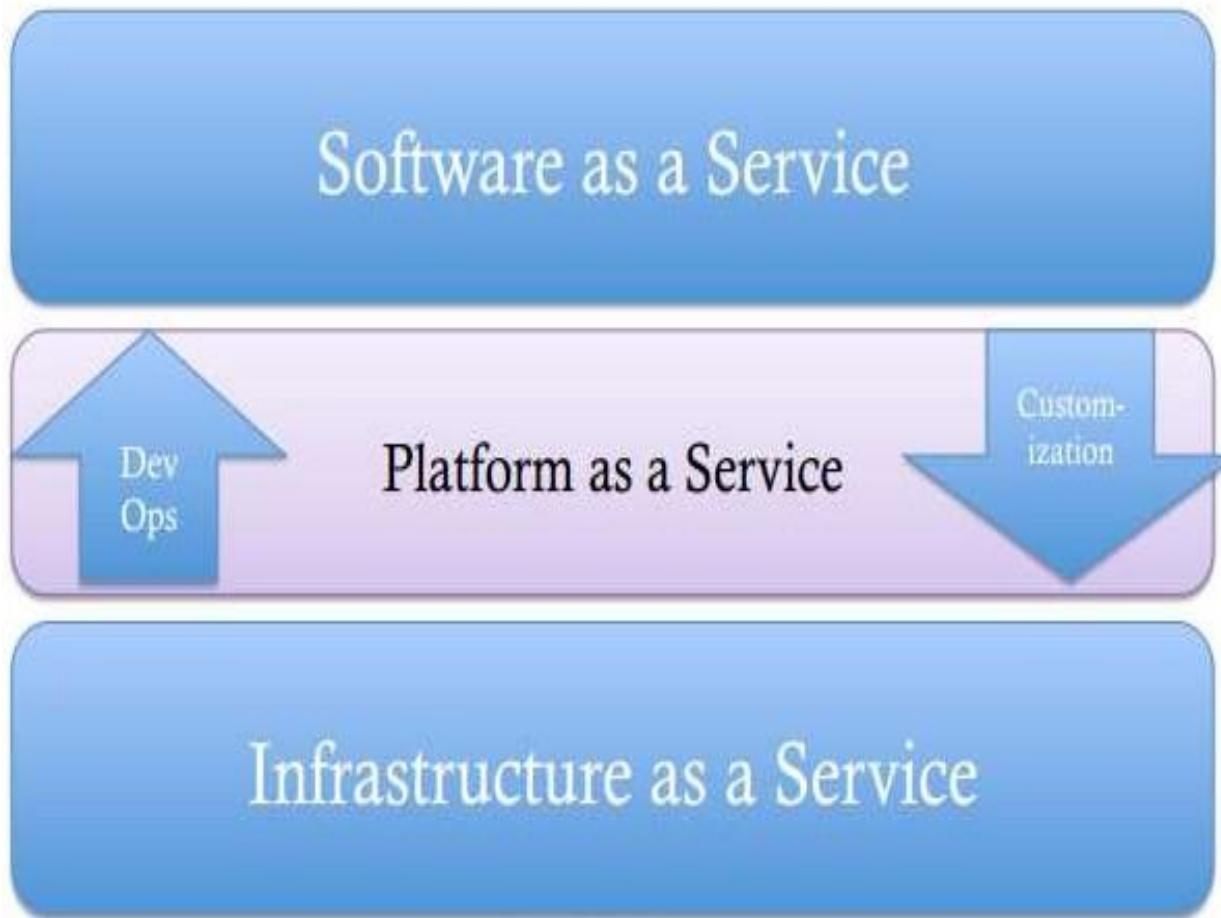


Seperation of Duties

- **Development** and **infrastructure** teams often clash when building software.
- Ideally, there is a clear **separation of duties** and clean handoffs.



PaaS



Evolving from different standards



- Evolving “upwards” from IaaS
 - Amazon (Mail, Notification, Events, Databases, Workflow, etc.)
- Evolving “downwards” from SaaS
 - Force.com – a place to host additional per-tenant logic.
 - Google App Engine
- Evolving “sideways” from middleware platforms
 - WSO2, Tibco, vmWare, Oracle, IBM



Deployment System

- Methods for continuous deployment
 - Upload
 - Version management



What do I care about (As a developer) ?

- My code – running
 - Not a “VM” but a Virtual App Server
- Not just code
 - I like Queues and Topics, ESB flows, Workflows, Databases, Logs, Portals, etc.
- Not just Runtime
 - I like SVN, Git, build, continuous integration, code coverage, automated test
- Moreover, if you are a manager of a group of developers, you probably like governance.

Beginning to use PaaS



- One of the decisions you need to make when beginning to use a PaaS is whether you want to maintain the software or if you want the vendor to be the administrator.

Maintain the software yourself



- If you choose to maintain the software yourself, you must set up, configure, maintain, and administer the PaaS yourself (either on a public or private cloud).

Let the vendor be the administrator



- Alternatively, you can have the vendor to provide these services. The result is reduced friction between the development and deployment teams.
There will, of course, be situations in which it's critical for the internal team to control and manage a complex software environment.



Best Practices

- Start with the data, and work up to the services and UI. No matter what the PaaS provider suggests.
- Define a staging and testing strategy before you begin development.
- Consider SOA approaches in the design and deployment of the PaaS-based application.
- Make sure to do load testing along with functional testing.
- Make sure to model performance.
- Don't fall in love with a PaaS player, you may need to use several.



Primary Benefits of PaaS

- Lower development costs (%30+)
- Lower deployment and DevOps costs (Nearly eliminate)
- Application portability (depends on the PaaS)

Benefits for Developers



- Below are some of the benefits of PaaS to application developers:
 - They don't have to invest in physical infrastructure (being able to 'rent' virtual infrastructure)
 - They don't have to be concerned with some of the lower-level details of the environment.
 - Makes development possible for 'non-experts'
 - Flexibility
 - Adaptability
 - Teams in various locations can work together & Security



IaaS

Infrastructure as a Service



Infrastructure as a Service (IaaS)



Learning Objectives

At the end of this lesson learner be able to

- Articulate the meaning of IaaS,
- Appreciate its characteristics
- know some of its appropriate uses and
- where IaaS inappropriate



Infrastructure as a Service

- Infrastructure as a service (IaaS) is a means of delivering **computing infrastructure** as an **on- demand** service.
- It is one of the three fundamental cloud service models



Computing Infrastructure

IaaS provides the following

- Servers- compute, machines
- Storage
- Network
- Operating system



How does IaaS Works?

The user instead of **purchasing-** servers, software, data center space or network equipment, **rent** those resources as a fully **outsourced** service on-demand model.



IaaS- Categories

IaaS can be obtained as

- (1) Public or
- (2) Private infrastructure or combination of both
- (3)

Public Cloud is considered infrastructure that consists of shared resources, deployed on a self-service basis over the Internet.

Private cloud incorporates some of cloud computing features such as Virtualization, but does so on a private network



IaaS- Characteristics

- Resources are distributed as a service
- Allows dynamic scaling (1...10....100.....)
- Has a variable costs-
- Generally includes multiple-users on a single piece of hardware. (**multi-tenancy**)



IaaS- Appropriate uses

IaaS is useful in the following situations

1.

- where demand is very **volatile**- any time there are significant **spikes and troughs** in terms of demand on the infracture
 - amazon.in, snapdeal, flipkart- during festival season



IaaS- Appropriate uses

IaaS is useful in the following situations

2.

- For new enterprise without **capital to invest in hardware**
 - entrepreneurs starting on a shoestring budget



IaaS- Appropriate uses

IaaS is useful in the following situations

3.

- Where the enterprise is growing rapidly and scaling hardware would be problematic
 - a company that experience huge success immediately



IaaS- Appropriate uses

IaaS is useful in the following situations

4.

- Where the enterprise is growing rapidly and scaling hardware would be problematic
 - a company that experience huge success immediately - animato, pinterest



IaaS- Appropriate uses

IaaS is useful in the following situations

5.

For specific line of business, trial or temporary infrastructural needs



IaaS- inappropriate use

- Where **regulatory compliance** makes the offshoring or outsourcing of data storage and processing difficult
 - personal information, medical information can not be stored in another country
- Where the highest levels of performance are required, and on-premise or dedicated hosted infrastructure has the capacity to meet the organization's needs
 - network latency,



Conclusion

- IaaS is one of three fundamental cloud service models
- IaaS provides basic computing resources- storage, servers, software networks
- IaaS resources are provided on-demand and utility model



Introduction to IaaS

- **IaaS:**
 - *Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet.*
- IaaS is one of three main categories of cloud computing services, alongside
- In an IaaS model, a third-party provider hosts hardware, software, servers, storage and other infrastructure components on behalf of its users.
- IaaS providers also host users' applications and handle tasks including system maintenance, backup and resiliency planning. Software as a Service (SaaS) and Platform as a Service (PaaS)..

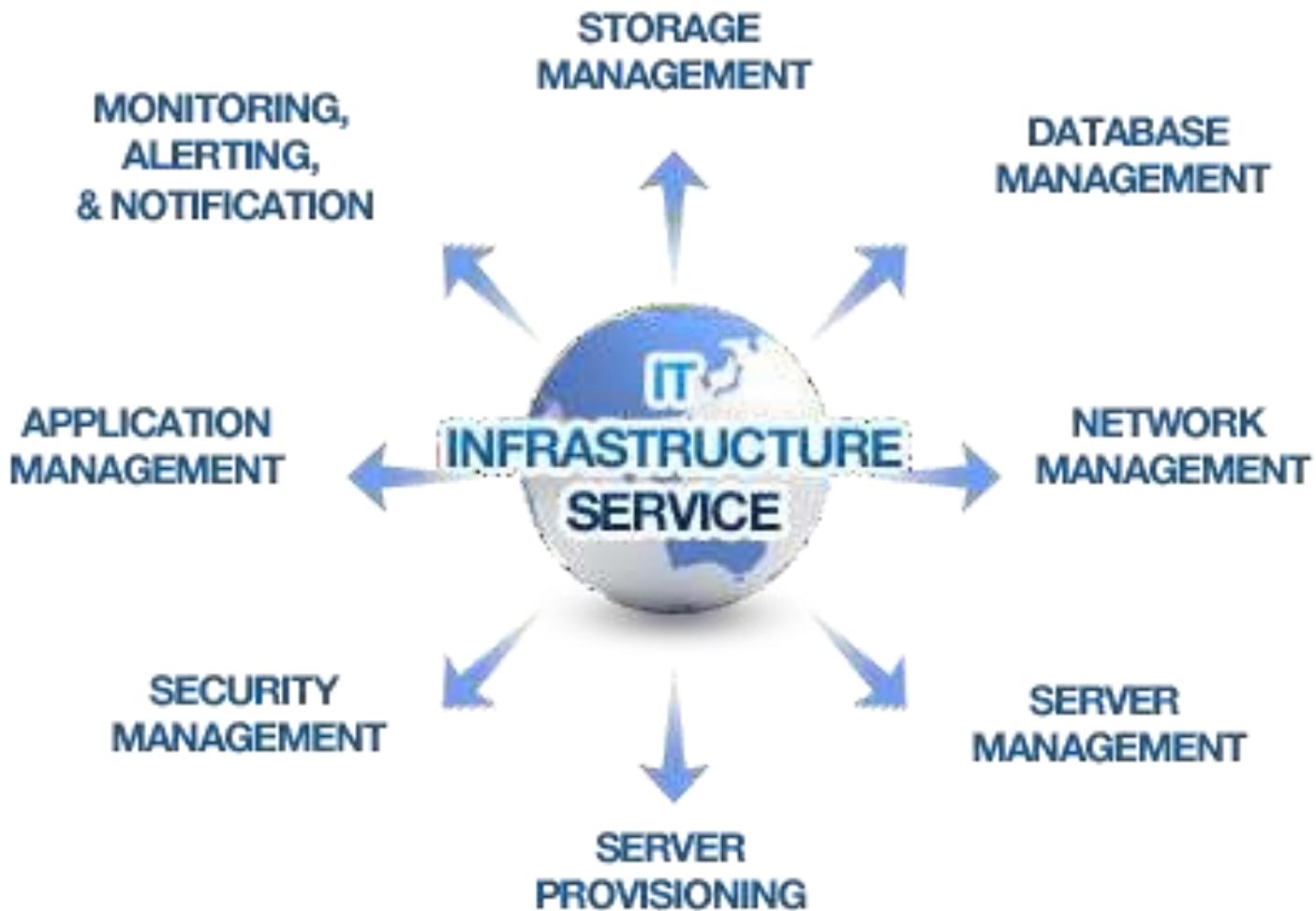


Advantages of IaaS

- Eliminates capital expense and reduces ongoing cost.
- Improves business continuity and disaster recovery.
- Innovate rapidly.
- Respond quicker to shifting business conditions.
- Focus on your core business.
- Increase stability, reliability and supportability.
- Better security.
- Gets new apps to users faster.



IaaS





Common IaaS business scenarios

- ***Typical things businesses do with IaaS include:***
- **Test and development.** Teams can quickly set up and dismantle test and development environments, bringing new applications to market faster. IaaS makes it quick and economical to scale up dev-test environments up and down.
- **Website hosting.** Running websites using IaaS can be less expensive than traditional web hosting.
- **Storage, backup and recovery.** Organizations avoid the capital outlay for storage and complexity of storage management, which typically requires a skilled staff to manage data and meet legal and compliance requirements. IaaS is useful for handling unpredictable demand and steadily growing storage needs. It can also simplify planning and management of backup and recovery systems.



Common IaaS Business Scenarios

- **Web apps.** IaaS provides all the infrastructure to support web apps, including storage, web and application servers and networking resources. Organisations can quickly deploy web apps on IaaS and easily scale infrastructure up and down when demand for the apps is unpredictable.
- **High-performance computing.** High-performance computing (HPC) on supercomputers, computer grids or computer clusters helps solve complex problems involving millions of variables or calculations. Examples include earthquake and protein folding simulations, climate and weather predictions, financial modeling and evaluating product designs.
- **Big data analysis.** Big data is a popular term for massive data sets that contain potentially valuable patterns, trends and associations. Mining data sets to locate or tease out these hidden patterns requires a huge amount of processing power, which IaaS economically provides.



Characteristics of SaaS, PaaS & IaaS

SaaS

- Web access to commercial software
- S/w is managed from central location
- s/w is delivered in a one to many model
- Uses not required to handle s/w upgrades and patches
- API allow for integration between different pieces of s/w

PaaS

- Multitenant Architecture
- Built in scalability of deployed s/w including load balancing and failover
- Integration with web services and databases via common stds.
- Support for development team collaboration-solutions include project planning & communication tools.

IaaS

- Resources are distributed as a service
- Allows for dynamic scaling



UNIT-II

Web Services

List of Topics

- What are Web Services ?
- How it works ?
- Why Web Services?
- Architecture
- Components
- Security
- How WebService is implemented?
- Web services samples
- REST

What are web services?

- Web Services can convert your application into a Web-application which can publish its function or message to the rest of the world.
- The basic Web Services platform is XML + HTTP
- A application which run on web (Internet or Intranet) and provides generic services.
- The services provided are through the web and in a standardized format which makes it generic and independent on the platform or the protocol on which the service was requested.

What are Web Services cont...

- Web services are open standard (XML, SOAP, HTTP etc.) based Web applications that interact with other web applications for the purpose of exchanging data
- There are two types of Webservices
 - **SOAP** (JAX-WS , Java API for XML Web Services)
 - **REST** (JAX-RS, Java API for RESTful Web Services)

What are Web Services cont...

To summarize, a complete web service is, therefore, any service that:

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

Why Web services ?

□ Exposing the existing function on to network:

A Web service is a unit of managed code that can be remotely invoked using HTTP, that is, it can be activated using HTTP requests. So, Web Services allows you to expose the functionality of your existing code over the network. Once it is exposed on the network, other application can use the functionality of your program.

Why Web services ?

□ Connecting Different Applications i.e. Interoperability:

Web Services allows different applications to talk to each other and share data and services among themselves. Other applications can also use the services of the web services.

For example VB or .NET application can talk to java web services and vice versa. So, Web services is used to make the application platform and technology independent.

Why Web services continued...

□ Standardized Protocol:

Web Services uses standardized industry standard protocol for the communication.

All the four layers (Service Transport, XML Messaging, Service Description and Service Discovery layers) uses the well defined protocol in the Web Services protocol stack.

This **standardization of protocol stack** gives the business many advantages like **wide range of choices, reduction in the cost** due to competition and increase in the quality.

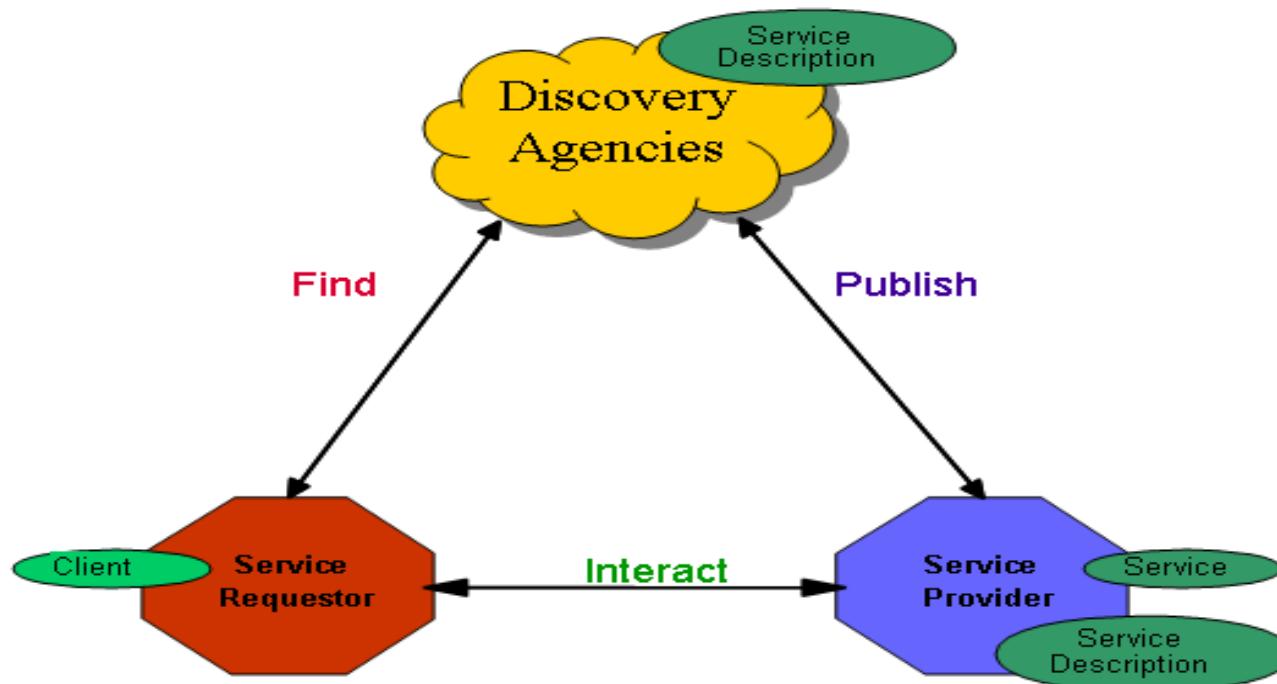
Why Web services continued...

Low Cost of communication:

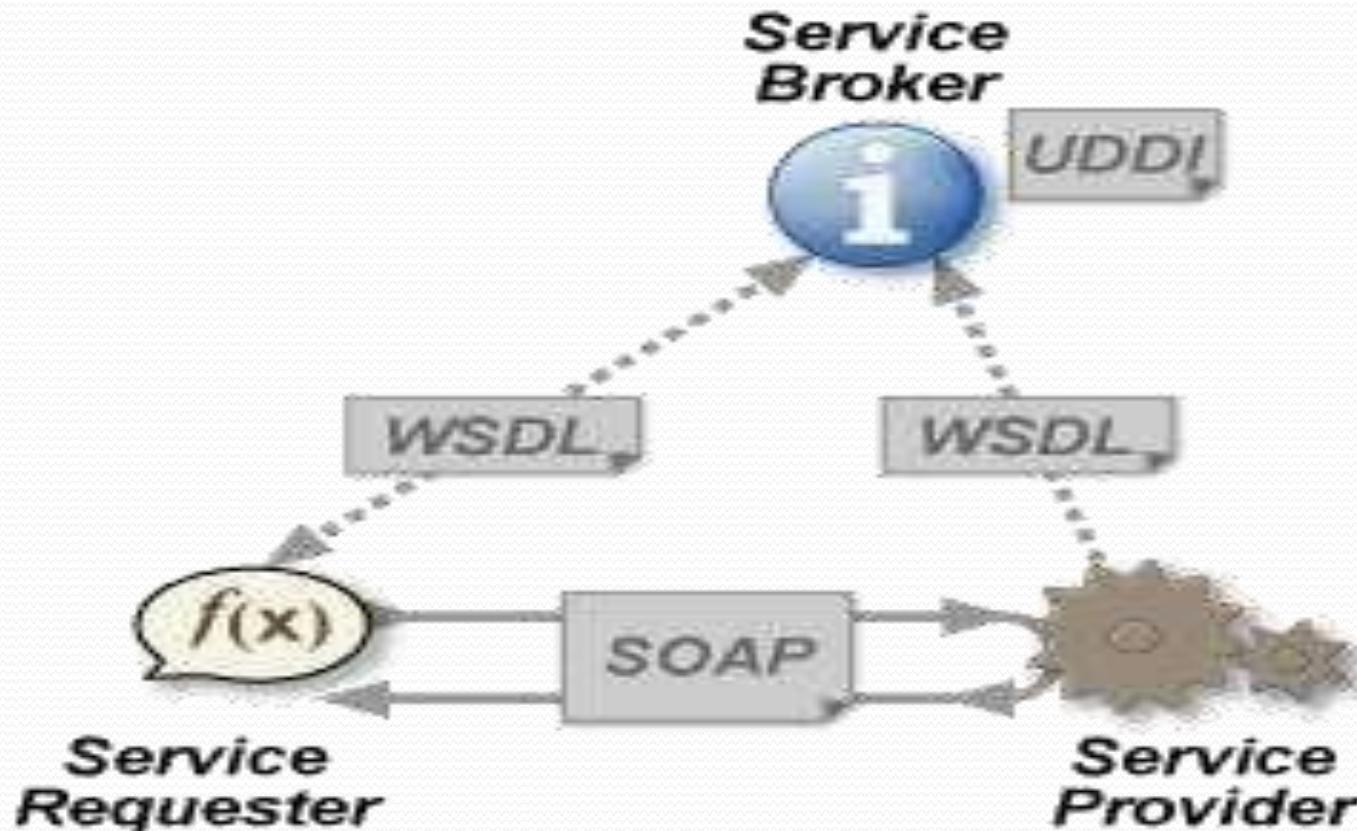
Web Services uses SOAP over HTTP protocol for the communication, so you can use your existing low cost internet for implementing Web Services. Beside SOAP over HTTP, Web Services can also be implemented on other reliable transport mechanisms like FTP etc.

Architecture

Service Oriented Architecture



Architecture cont...

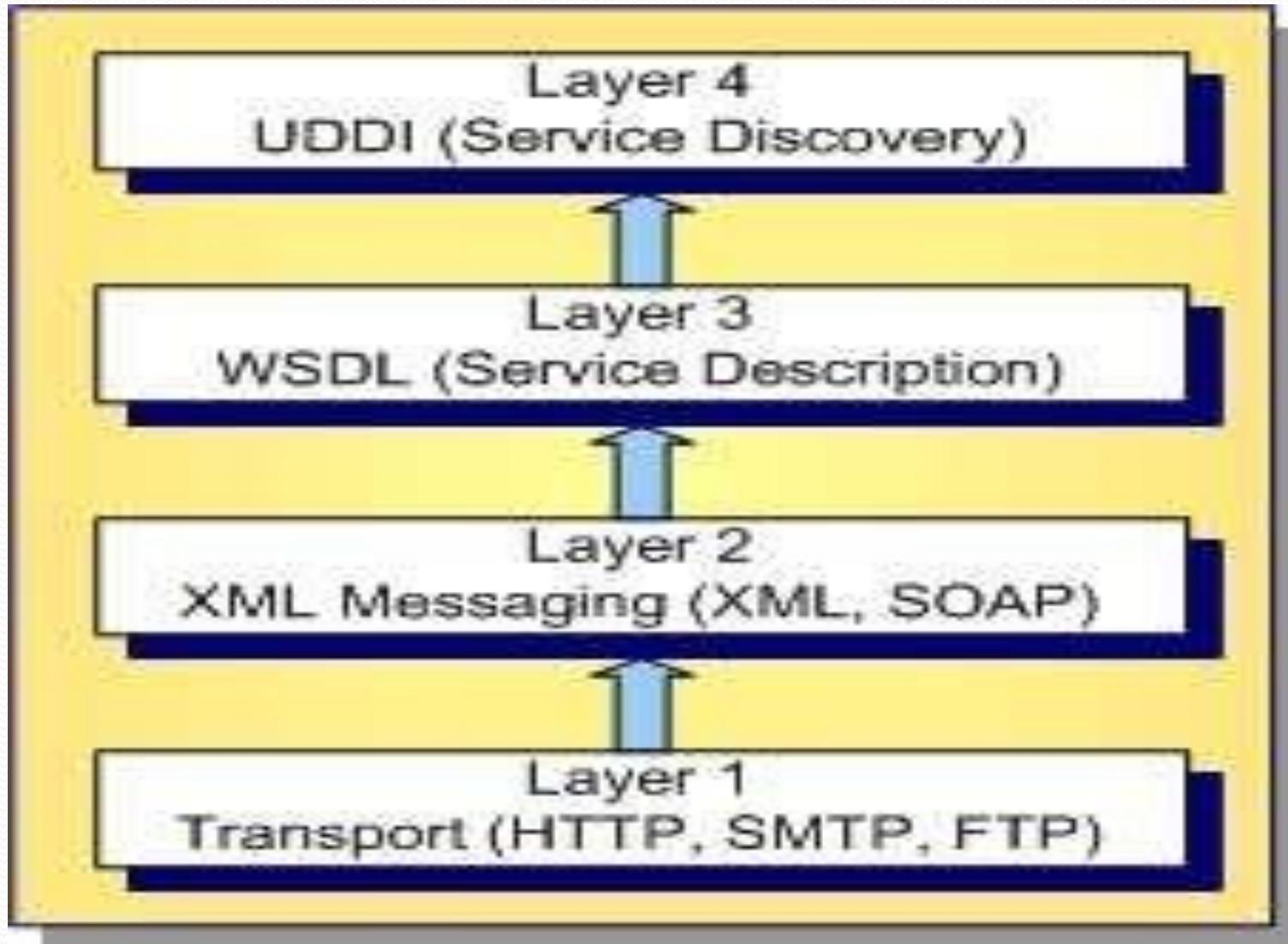


Web service roles

There are **three major roles** within the web service architecture

- **Service provider:** This is the provider of the web service. The service provider implements the service and makes it available on the Internet.
- **Service requestor:** This is any consumer of the web service. The requestor utilizes an existing web service by opening a network connection and sending an XML request.
- **Service registry:** This is a logically centralized directory of services.
The registry provides a central place where developers can publish new services or find existing ones.

Web Service protocol stack



Protocol stack cont...

- **Service transport:** This layer is responsible for transporting messages between applications. Currently, this layer includes hypertext transfer protocol (HTTP), Simple Mail Transfer Protocol (SMTP), file transfer protocol (FTP), and newer protocols, such as Blocks Extensible Exchange Protocol (BEEP).
- **XML messaging:** This layer is responsible for encoding messages in a common XML format so that messages can be understood at either end. Currently, this layer includes XML-RPC and SOAP.

Protocol stack cont...

- **Service description:** This layer is responsible for describing the public interface to a specific web service. Currently, service description is handled via the Web Service Description Language (WSDL).
- **Service discovery:** This layer is responsible for centralizing services into a common registry, and providing easy publish/find functionality. Currently, service discovery is handled via Universal Description, Discovery, and Integration (UDDI).

Components of Web Service

- SOAP (Simple Object Access Protocol)
 - Protocol based on XML , used for message transfer
- WSDL (Web Service Description Language)
 - XML file used to describe the Web service and how to access them
- UDDI (Universal Description and Discovery Integration)
 - Used to register and search for web service
 - Directory of web service
- JAX-RPC
 - For intercommunication
- HTTP
 - For message transfer

What is SOAP?

SOAP is an XML-based protocol to let applications exchange information over HTTP.

Or more simple: SOAP is a protocol for accessing a Web Service.

- SOAP stands for Simple Object Access Protocol
- SOAP is a communication protocol
- SOAP is a format for sending messages
- SOAP is designed to communicate via Internet
- SOAP is platform independent
- SOAP is language independent
- SOAP is based on XML
- SOAP is simple and extensible
- SOAP allows you to get around firewalls
- SOAP is a W3C standard

What is WSDL?

WSDL is an XML-based language for locating and describing Web services.

- WSDL stands for Web Services Description Language
- WSDL is based on XML
- WSDL is used to describe Web services
- WSDL is used to locate Web services
- WSDL is a W3C standard

What is UDDI?

UDDI is a directory service where companies can register and search for Web services.

- UDDI stands for Universal Description, Discovery and Integration
- UDDI is a directory for storing information about web services
- UDDI is a directory of web service interfaces described by WSDL
- UDDI communicates via SOAP

XML-RPC

This is the simplest XML based protocol for exchanging information between computers.

- XML-RPC is a simple protocol that uses XML messages to perform RPCs.
- Requests are encoded in XML and sent via HTTP POST.
- XML responses are embedded in the body of the HTTP response.
- XML-RPC is platform-independent.
- XML-RPC allows diverse applications to communicate.
- XML-RPC is the easiest way to get started with web services.

Security

□ Confidentiality

If a client sends an XML request to a server, then question is that can we ensure that the communication remains confidential?

Answer lies here

- XML-RPC and SOAP run primarily on top of HTTP.
- HTTP has support for Secure Sockets Layer (SSL).
- Communication can be encrypted via the SSL.
- SSL is a proven technology and widely deployed.



Security

□ Authentication

If a client connects to a web service, how do we identify the user? And is the user authorized to use the service?

Following options can be considered but there is no clear consensus on a strong authentication scheme.

- HTTP includes built-in support for Basic and Digest authentication, and services can therefore be protected in much the same manner as HTML documents are currently protected.
- SOAP Security Extensions: Digital Signature (SOAP-DSIG). DSIG leverages public key cryptography to digitally sign SOAP messages. This enables the client or server to validate the identity of the other party.
- The Organization for the Advancement of Structured Information Standards (OASIS) is working on the Security Assertion Markup Language (SAML, It is an XML-based open standard data format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider).

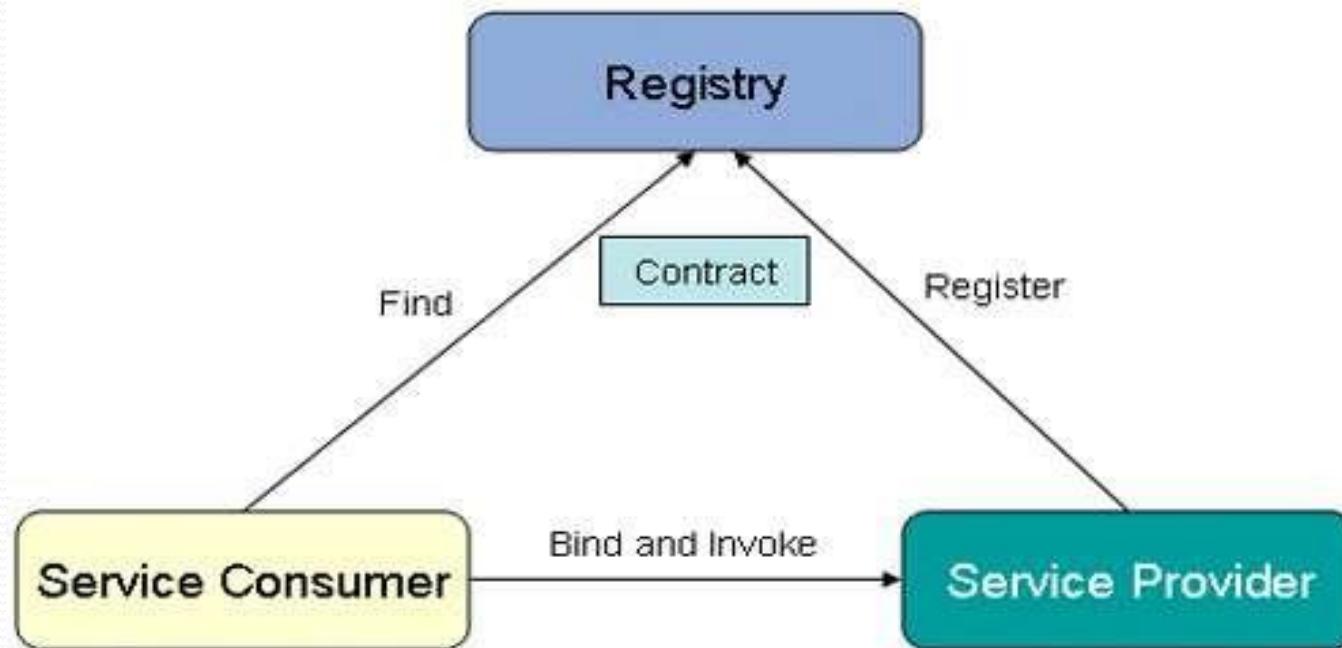
Security cont...

□ Network Security

There is currently no easy answer to this problem, and it has been the subject of much debate. For now, if you are truly intent on filtering out SOAP or XML-RPC messages, one possibility is to filter out all HTTP POST requests that set their content type to text/xml.

How Web Service is implemented ?

- Build and Publish
- Find
- Bind and Invoke



Web Service Implementation Cont...

- **Pre Conditions**
 - No one will be able to use your service
- **Step 1:Build and Publish**
 - *Build*
 - Create your application
 - Create your contract file (WSDL)
 - *Publish*
 - Register your application as a web service onto any registry
 - This process happens on UDDI using a separate SOAP request
 - This process is useful only if your web service should be accessible using Internet
- **Post conditions**
 - Your web service will be available to the Public (If published to any registry) or will be accessible to intranet users

Web Service Implementation

Cont...

□ Pre Conditions

- The user needs to access a service but is not aware of the service details

□ Step 2 :Find

- Find
 - Search in the registry for a service which provides your needs
 - Obtain the necessary details about the service

□ Post conditions

- The user will have all the details about the service and gets ready to contact the service

Web Service Implementation

Cont...

□ Pre Conditions

- The user will have the contract necessary to identify and call the service

□ Step 3:Bind

□ Bind

- Use the contract file to build the request message.

□ Invoke

- Send a request to the service and request for the necessary operation available from the service.
- The request should be sent in the protocol which is required by the service

□ Post conditions

- The user will receive the response from the service in a format specified in the contract file.

Web Services samples

- Eclipse tutorial link :

<http://www.eclipse.org/webtools/jst/components/ws/1.5/tutorials/index.html>

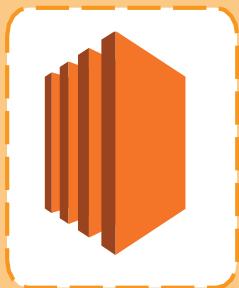
- Bottom to Top approach
 - JAVAto WSDL
- Top to bottom approach
 - WSDL to JAVA



Amazon Elastic Compute Cloud (EC2)



Amazon Elastic Compute Cloud (EC2)



Amazon
EC2

- **Resizable** compute capacity
- Complete control of your computing resources
- **Reduces the time required** to obtain and boot new server instances to minutes

Amazon EC2 Facts



- Scale capacity as your computing requirements change
- Pay only for capacity that you actually use
- Choose Linux or Windows
- Deploy across AWS Regions and Availability Zones for reliability

-



Amazon's service lets customers choose **from three sizes of virtual servers**:

Small, which offers the equivalent of a system with 1.7GB of memory, 160GB of storage, and one virtual 32-bit core processor

Large, which offers the equivalent of a system with 7.5GB of memory, 850GB of storage, and two 64-bit virtual core processors

Extra large, which offers the equivalent of a system with 15GB of memory, 1.7TB of storage, and four virtual 64-bit core processors

Launching an Amazon EC2 Instance via the Web Console



1. Determine the AWS Region in which you want to launch the Amazon EC2 instance.
2. Launch an Amazon EC2 instance from a pre-configured Amazon Machine Image (AMI).
3. Choose an instance type based on CPU, memory, storage, and network requirements.
4. Configure network, IP address, security groups, storage volume, tags, and key pair.

AMI Details



An AMI includes the following:

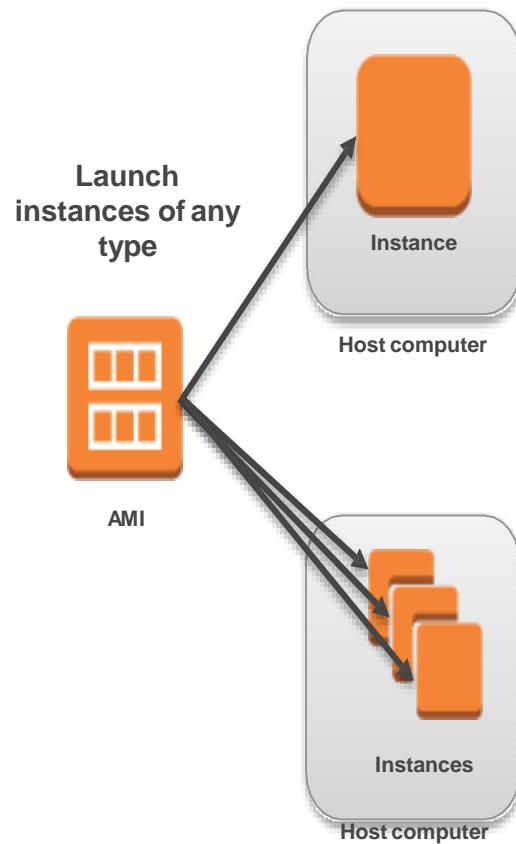
- ─ A template for the root volume for the instance (for example, an operating system, an application server, and applications).
- ─ Launch permissions that control which AWS accounts can use the AMI to launch instances.
- ─ A block device mapping that specifies the volumes to attach to the instance when it's launched.
- ─

Instances and AMIs

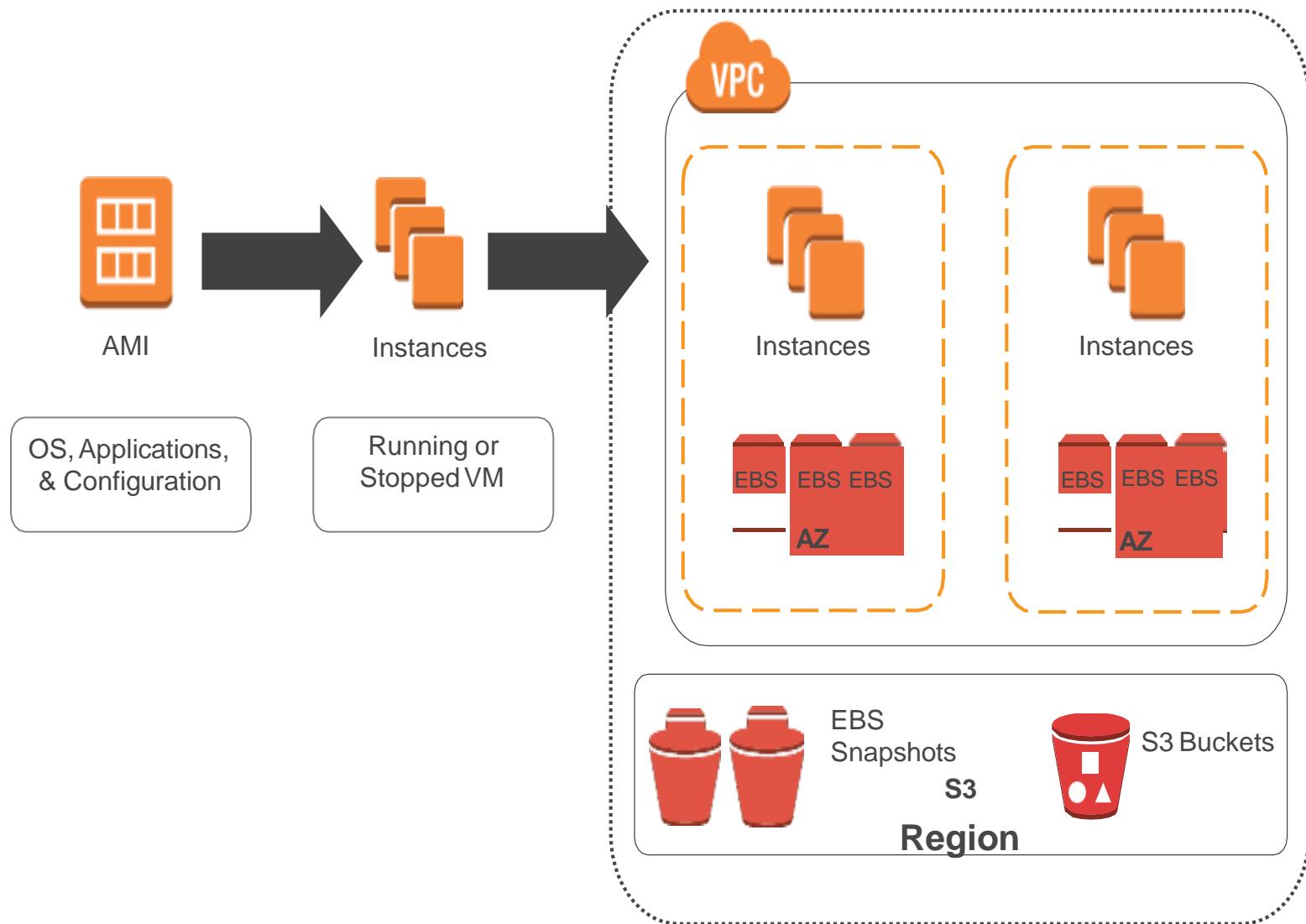


Select an AMI based on:

- Region
- Operating system
- Architecture (32-bit or 64-bit)
- Launch permissions
- Storage for the root device
- Network interface



Amazon EC2 Instances



Amazon EBS vs. Amazon EC2 Instance Store



Amazon EBS

- Data stored on an Amazon EBS volume can persist independently of the life of the instance.
- Storage is persistent.

Amazon EC2 Instance Store

- ### Amazon EC2 Instance Store
- Data stored on a local instance store persists only as long as the instance is alive.
 - Storage is ephemeral.

AMI Types - Storage for the Root Device



Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually < 1 minute	Usually < 5 minutes
Size limit	16 TiB	10 GiB
Data persistence	The root volume is deleted when the instance terminates. Data on any other Amazon EBS volumes persists after instance termination.	Data on any instance store volumes persists only during the life of the instance.
Charges	Instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	Instance usage and storing your AMI in Amazon S3.
Stopped state	Can be stopped.	Cannot be stopped.

AWS Marketplace – IT Software Optimized for the Cloud



AWS Marketplace:

Is an online store to discover, purchase, and deploy IT software on top of the AWS infrastructure.

- Catalog of 2300+ IT software solutions
 - Including Paid, BYOL, Open Source, SaaS, & free to try options
- Pre-configured to operate on AWS
 - Software checked by AWS for security and operability
- Deploys to AWS environment in minutes
- Flexible, usage-based billing models
- Software charges billed to AWS account

Includes AWS Test Drive.

<https://aws.amazon.com/marketplace>

The screenshot shows the AWS Marketplace homepage. At the top, there's a navigation bar with links for 'Amazon Web Services Home', 'Your Account', 'Help', and 'Sell on AWS Marketplace'. Below the navigation is a search bar labeled 'Search AWS Marketplace' and a button for 'Sign in or Create a new account'. A large central banner features a diagram of a server node connected to a network and the text: 'Production-ready cluster deployments in minutes with AWS Marketplace and AWS CloudFormation'. It also mentions 'Deploy AMI clusters with multiple AWS resources, all in one easy transaction' with a 'Learn more' link. To the right of the banner is a section titled 'Operating Systems' with icons for Amazon Linux AMI (x86_64) and CentOS 7 (x86_64) with Update RPM. Below the banner are sections for 'Featured Products' and 'Popular Products'. The 'Featured Products' section includes links for WebSphere Application Server, MaxCompute ETL for Redshift, TIBCO Clarity, and Oracle Linux 6.5. The 'Popular Products' section lists Sophos UTM 9, SoftNAS Cloud Standard - High-Performance, TIBCO Jaspersoft for AWS with Multi-Tenant Software, and Ubuntu Server 14.04 LTS. Each product listing includes a logo, name, description, price range, and a 'Free Trial' button.

AWS Marketplace



Enable success in the cloud with software access across your technology stack.

Software Categories

Business Intelligence

Media

Dev Ops

Security

Open Source & OS

Database

Network

Storage



And Many More



DevOps

- DevOps is a set of practices that works to automate and integrate the processes between software development and IT teams, so they can build, test, and release software faster and more reliably.
- The term DevOps was formed by combining the words “development” and “operations” and signifies a cultural shift that bridges the gap between development and operation teams, which historically functioned in siloes.

AWS Marketplace



Leverage a broad catalog of IT software to support your workload needs.



Choosing the Right Amazon EC2 Instance



EC2 instance types are optimized for different use cases and come in multiple sizes. This allows you to optimally scale resources to your workload requirements.

AWS uses Intel® Xeon® processors for EC2 instances, providing customers with high performance and value.

Consider the following when choosing your instances: Core count, memory size, storage size and type, network performance, and CPU technologies.

Hurry Up and Go Idle - A larger compute instance can save you time and money, therefore paying more per hour for a shorter amount of time can be less expensive.



Get the Intel® Advantage



Intel's latest 22nm Haswell microarchitecture on new C4 instances, with custom **Intel® Xeon® v3** processors, provides new features:

- **Haswell microarchitecture** has better branch prediction; greater efficiency at prefetching instructions and data; along with other improvements that can **boost existing applications' performance** by **30% or more**
- **P state and C state control** provides the ability to individually tune each cores performance and sleep states to improve application performance
- **Intel® AVX2.0** instructions can double the floating-point performance for compute-intensive workloads over Intel® AVX, and provide additional instructions useful for compression and encryption



Intel® Processor Technologies



■ **Intel® AVX** – Get dramatically better performance for highly parallel HPC workloads such as life science engineering, data mining, financial analysis, or other technical computing applications. AVX also enhances image, video, and audio processing.

■ **Intel® AES-NI** – Enhance your security with these new encryption instructions that reduce the performance penalty associated with encrypting/decrypting data.

■ **Intel® Turbo Boost Technology** – Get more computing power when you need it with performance that adapts to spikes in your workload with Intel® Turbo Boost Technology 2.0



EC2 Instances with Intel® Technologies



	Burstable	Balanced	Compute	Memory	GPU	I/O	Storage
AWS Instance Type	T2	M4	C4	R3	G2	I2	D2
Intel® processor	Intel® Xeon® family	Intel® Xeon® E5-2676 v3	Intel® Xeon® E5-2666 v3	Intel® Xeon® E5-2670 v2	Intel® Xeon® E5-2670	Intel® Xeon® E5-2670 v2	Intel® Xeon® E5-2676 v3
Intel® process technology		22nm Haswell	22nm Haswell	22nm Ivy Bridge	32nm Sandy Bridge	22nm Ivy Bridge	22nm Haswell
Intel® AVX							
Intel® AVX2							
Intel® Turbo Boost							
Storage	EBS only	EBS only	EBS only	SSD	SSD	SSD	HDD

Current Generation Instances



Instance Family	Some Use Cases
General purpose (t2, m4, m3)	<ul style="list-style-type: none">• Low-traffic websites and web applications• Small databases and mid-size databases
Compute optimized (c4, c3)	<ul style="list-style-type: none">• High performance front-end fleets• Video-encoding
Memory optimized (r3)	<ul style="list-style-type: none">• High performance databases• Distributed memory caches
Storage optimized (i2, d2)	<ul style="list-style-type: none">• Data warehousing• Log or data-processing applications
GPU instances (g2)	<ul style="list-style-type: none">• 3D application streaming• Machine learning

Instance Metadata & User Data



Instance Metadata:

- Is data about your instance.
- Can be used to configure or manage a running instance.

Instance User Data:

- Can be passed to the instance at launch.
- Can be used to perform common automated configuration tasks.
- Runs scripts after the instance starts.



Retrieving Instance Metadata



- To view all categories of instance metadata from within a running instance, use the following URI:

<http://169.254.169.254/latest/meta-data/>

On a Linux instance, you can use:



```
➤ $ curl http://169.254.169.254/latest/meta-data/
➤ $ GET http://169.254.169.254/latest/meta-data/
```

- All metadata is returned as text (content type text/plain).

The screenshot shows a browser window with the URL <http://169.254.169.254/latest/meta-data/>. The page content is a list of metadata categories:

- ami-id
- ami-launch-index
- ami-manifest-path
- block-device-mapping/
- hostname
- instance-action
- instance-id
- instance-type
- local-hostname
- local-ipv4
- mac
- metrics/
- network/
- placement/
- profile
- public-hostname
- public-ipv4
- public-keys/
- reservation-id
- security-groups
- services/

Adding User Data



- 💡 You can specify user data when launching an instance.

User data can be:

- 💡 ➤ Linux script – executed by **cloud-init**
- Windows batch or PowerShell scripts – executed by **EC2Config** service

User data scripts run once per instance-id by default.



User Data Example Linux



```
#!/bin/sh
```

User data shell scripts must start with the #! characters and the path to the interpreter you want to read the script.

```
yum -y install httpd  
chkconfig httpd on  
/etc/init.d/httpd  
start
```

Install Apache web server
Enable the web server
Start the web server

User Data Example Windows



```
<powershell>
```

```
Import-Module ServerManager
```

Import the Server Manager module
for Windows PowerShell.

```
Install-WindowsFeature web-server, web-webserver  
Install-WindowsFeature web-mgmt-tools
```

```
</powershell>
```

Install IIS
Install Web Management Tools

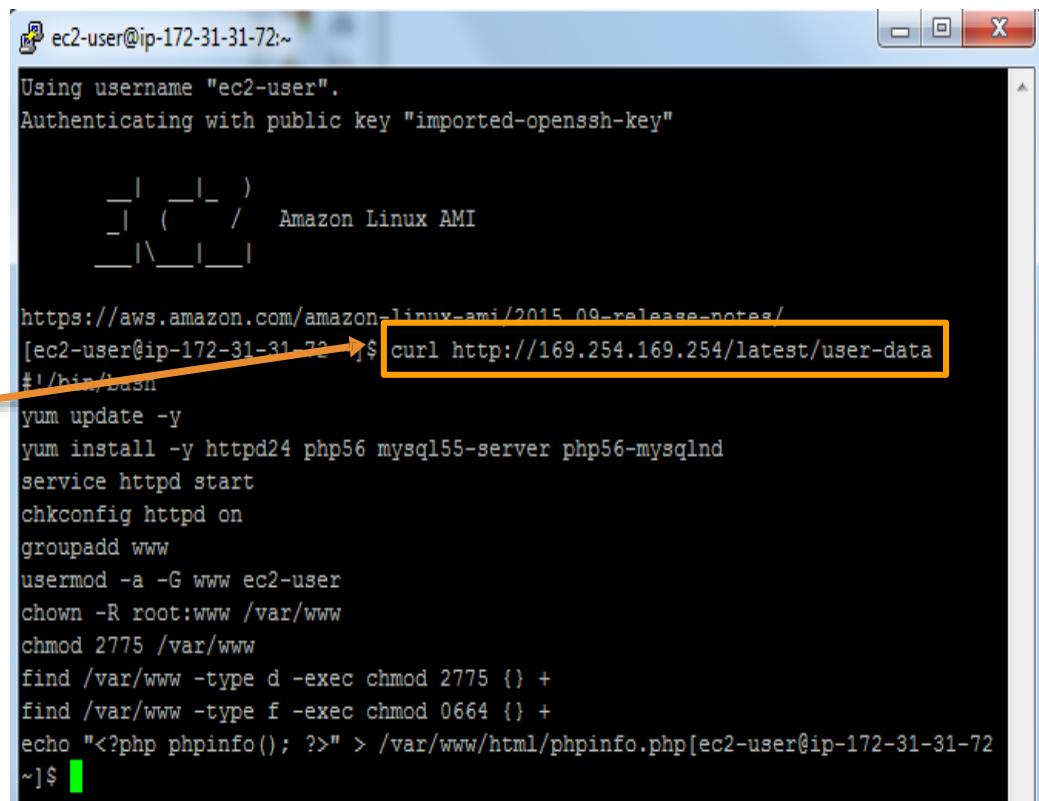
Retrieving User Data



- To retrieve user data, use the following URI:
<http://169.254.169.254/latest/user-data>

On a Linux instance, you can use:

- \$ curl <http://169.254.169.254/latest/user-data/>
- \$ GET <http://169.254.169.254/latest/user-data/>



The screenshot shows a terminal window titled "ec2-user@ip-172-31-31-72:~". The session starts with the message "Using username "ec2-user". Authenticating with public key "imported-openssh-key"". It then displays the Amazon Linux AMI logo. The command "curl http://169.254.169.254/latest/user-data" is highlighted with a yellow box and an arrow pointing from the text above it. The terminal then lists several system updates and configurations, ending with the command "echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php[ec2-user@ip-172-31-31-72 ~]\$".

Amazon EC2 Purchasing Options



On-Demand Instances

Pay by the hour.

Reserved Instances

Purchase at significant discount.
Instances are always available.

1-year to 3-year terms.

Scheduled Instances

Purchase a 1-year RI for a recurring period of time.

Spot Instances

Highest bidder uses instance at a significant discount.
Spot blocks supported.

Dedicated Hosts

Physical host is fully dedicated to run your instances. Bring your per-socket, per-core, or per-VM software licenses to reduce cost.



Multi-entity support

Multi-schema approach

Multi-Tenant Architecture

Cloud Service Management

On Demand Computing



Multi-entity support

- A multi-tenant cloud is a cloud computing architecture that allows customers to share computing resources in a public or private cloud.
- Each tenant's data is isolated and remains invisible to other tenants.
- In a multi-tenant cloud system, users have individualized space for storing their projects and data.



Benefits of multitenancy

Better use of resources:

One machine reserved for one tenant isn't efficient, as that one tenant is not likely to use all of the machine's computing power.

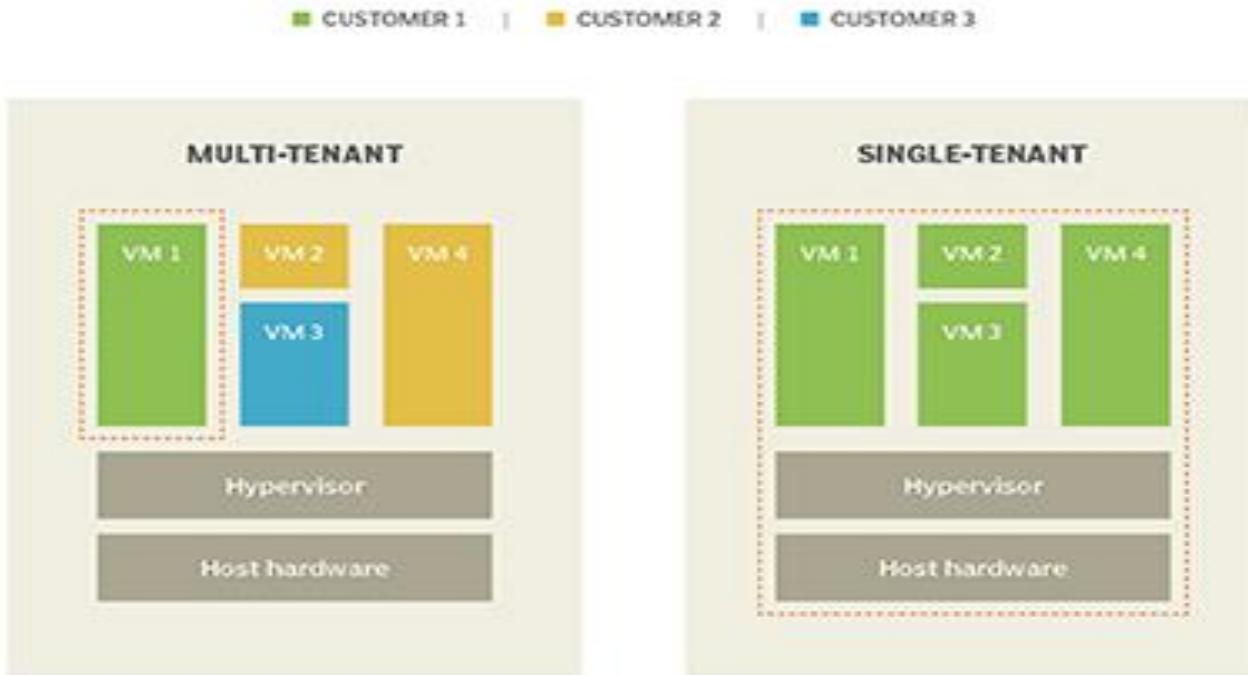
By sharing machines among multiple tenants, use of available resources is maximized.

Lower costs:

With multiple customers sharing resources, a cloud vendor can offer their services to many customers at a much lower cost than if each customer required their own dedicated infrastructure.

Single-Tenant vs. Multi-Tenant

Multi-tenant vs. single-tenant





Single-Tenant vs. Multi-Tenant

Single-Tenant Hosting	Multi-Tenant Hosting
*a client operates in a solely dedicated environment.	*each tenant is isolated from other tenants.
Advantage:	Advantage:
Greater customization	Simplified hosting
Greater isolation from security risks	Better protection of systems
Faster recovery	Upgrading software is no longer your problem
Better control	
Avoiding “noisy neighbor” syndrome	
Disadvantage:	Disadvantage:
Cost	They have their own security risks
Client responsibility	Downtime
Less efficient	Noise neighbors



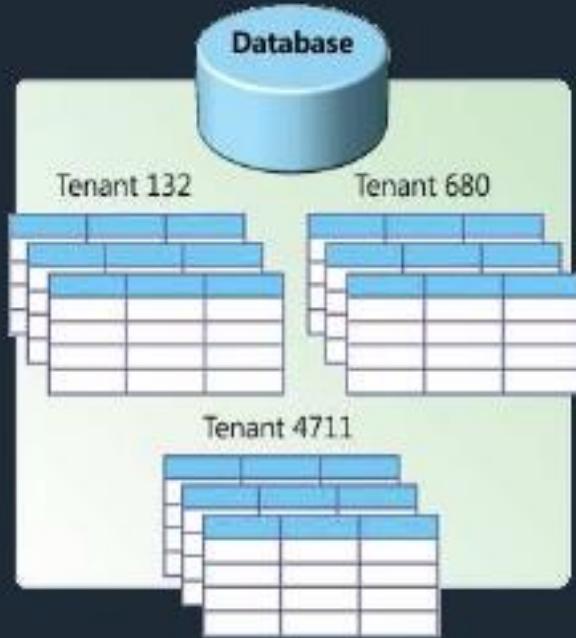
Multitenancy in private cloud computing

- Private cloud computing uses multitenant architecture that the other tenants are from the internal organizations.
- In private cloud computing, different teams within Company A share infrastructure with each other.
- Multitenancy is providers
 - Amazon Web Services and Microsoft Azure.
 - multitenancy is a , software as a service, is deployed by many software as a service companies.

Multiple tenants in the same DB



Separate schema



Shared Database, Separate Schemas

This approach involves housing multiple tenants in the same database, with each tenant having its own set of tables that are grouped into a schema created specifically for the tenant.

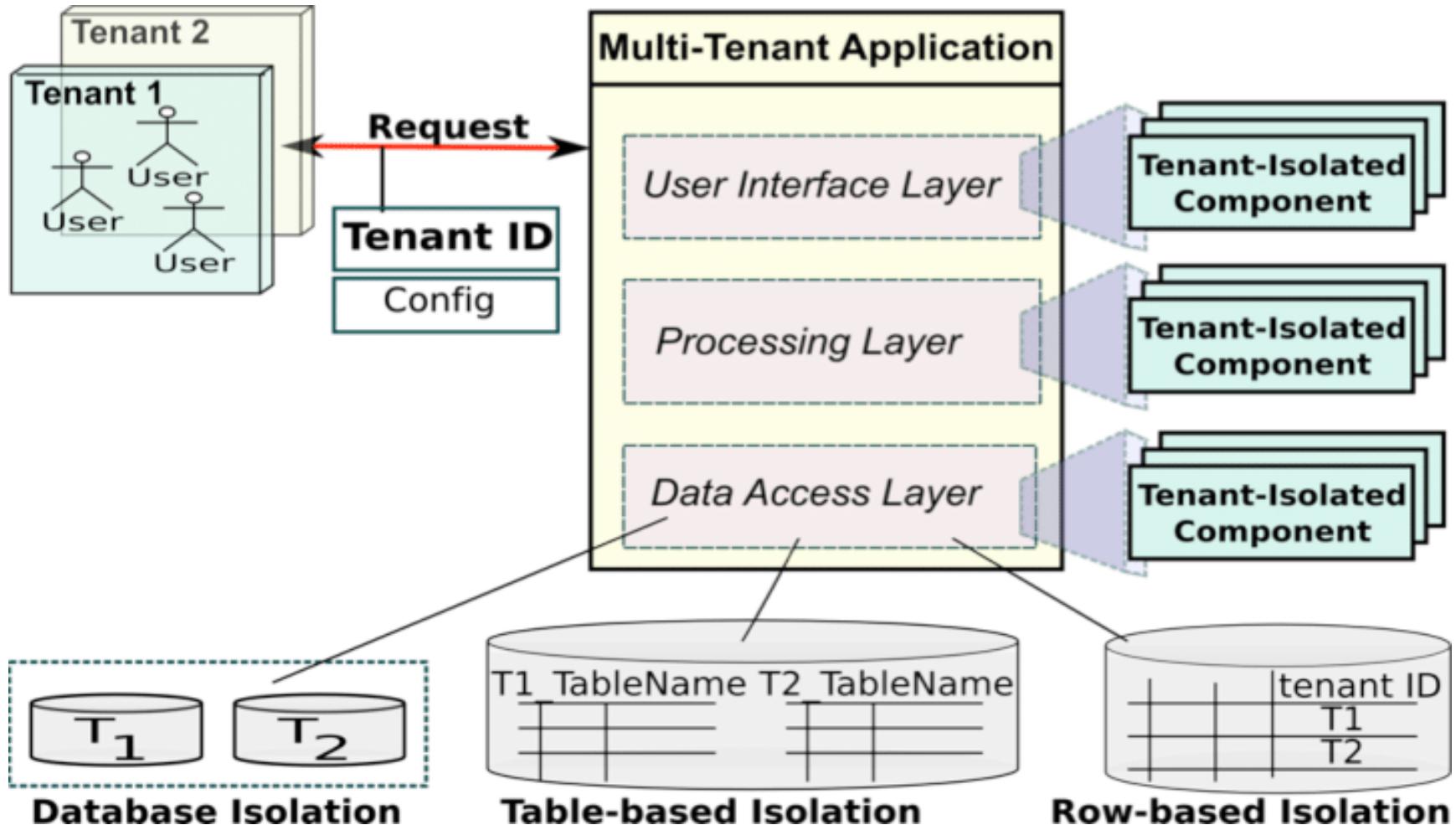


Development of Multi-Tenant Architecture

- In a multitenancy environment, multiple customers share the same application, in the same operating environment, on the same hardware, with the same storage mechanism.
- In virtualization, every application runs on a separate virtual machine with its own OS.



Multi-tenant architecture with tenant-isolated components





Multi-tenancy using cloud data stores

- The cloud storage has both financial and security advantages.
- Financial advantages in the case that the virtual resources are cheaper than dedicated physical resources connected to a personal computer or network.



Types of cloud storage

- Personal Cloud Storage
- Public Cloud Storage
- Private Cloud Storage
- Hybrid Cloud Storage



Types of cloud storage

Personal Cloud Storage:

- It is also called mobile cloud storage which is a subset of public cloud storage that applies to storing an individual's data in the cloud and provides access to data from anywhere.
- Apple's iCloud is an example of personal cloud storage.



Types of cloud storage

Public Cloud Storage:

- Public cloud storage is where the **storage service providers and enterprises are separate** and there aren't any cloud resources stored in the enterprise's data center.
- The enterprise data management is done by cloud storage provider.



Types of cloud storage

Private Cloud Storage:

- A form of cloud storage where the **data storage is done within the enterprise.**
- This helps to resolve the potential risk for performance and security concerns while still providing the advantages of cloud storage.



Types of cloud storage

Hybrid Cloud Storage:

- Hybrid cloud storage is a combination of private and public cloud storage where some data that is critical resides in the enterprise's private cloud while other data is stored and accessible from a public storage provider.



Tenant Data Management in Multi-Tenant Cloud Systems

- Multi-tenancy enables the client service providers to serve multiple clients by a single application instance, isolating each tenant's data.
- The benefits of this method include increased utilization of available hardware resources and improved ease of maintenance and deployment.
- These results are in lower overall application costs.

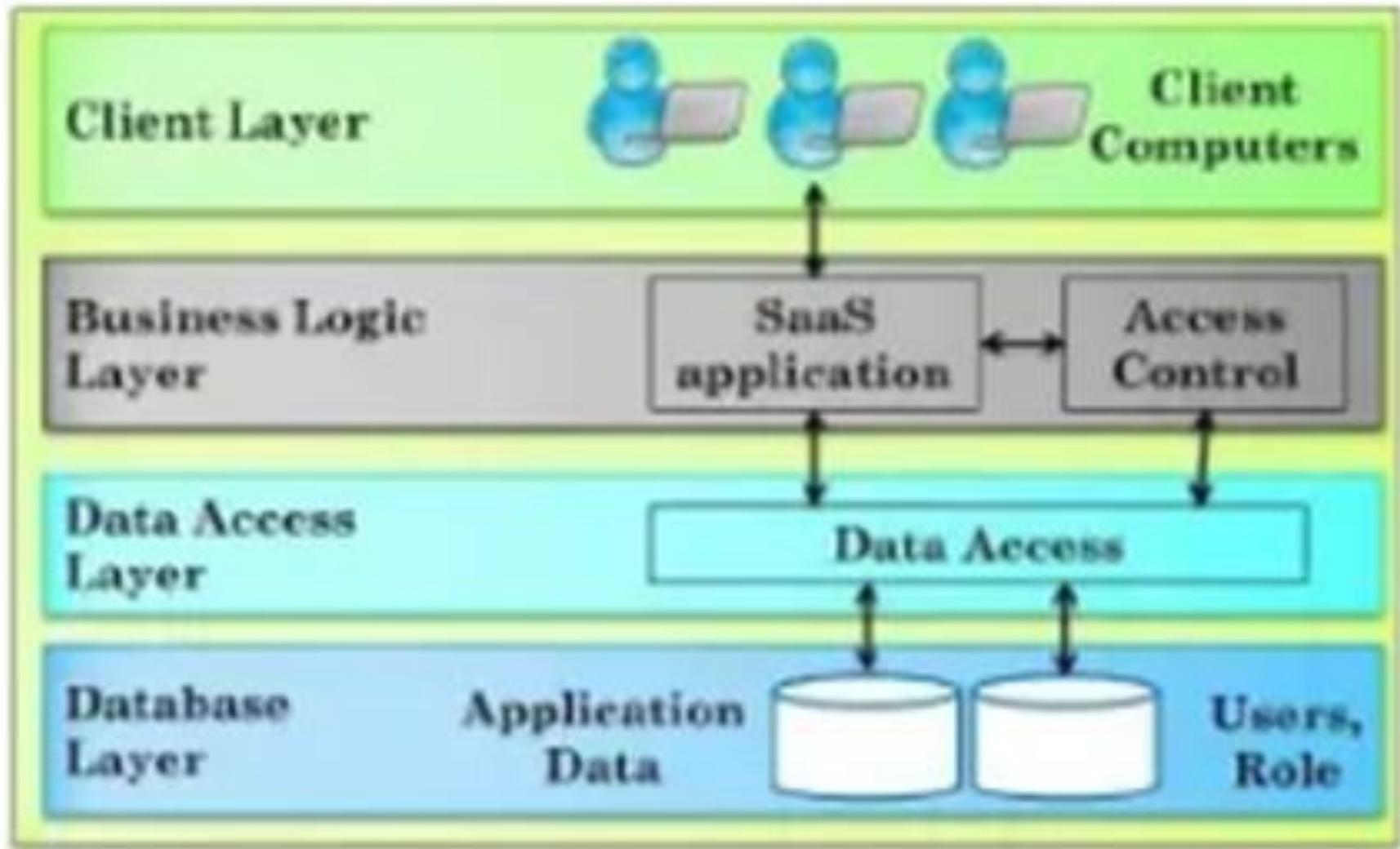


Tenant Data Management in Multi-Tenant Cloud Systems

- In a multi-tenant architecture, a software application is designed virtually to partition data.
- Each tenant works in a virtual application instance in which some tenants may be divided into multiple subtenants, each having its own users.
- As the number of tenants increases, a scalable architecture is needed



Tenant Data Management in Multi-Tenant Cloud Systems





Tenant Data Management in Multi-Tenant Cloud Systems

The business logic layer provides security by the access control component which should be isolated from the offered services.

The database layer stores the application data. To enforce scalability of the database layer a logical hierarchical representation of the tenants and subtenants, and making a mapping to the physical storage is followed.

The data access layer provides load balancing between the different data stores and stores the decision support for managing the data store.



Tenant Data Management in Multi-Tenant Cloud Systems

- Tenants and subtenants in a multi-tenant organization can be structured hierarchically in the form of **a tenant tree**.
- At the top level of the tree is the SaaS provider, which is the root node of the tenant tree.
- Various tenants using the applications are located at the next level, and is the child nodes of SaaS provider.
- All tenants share the same parent or root node.
- Some tenants may be even divided into multiple subtenants.



Tenant Tree

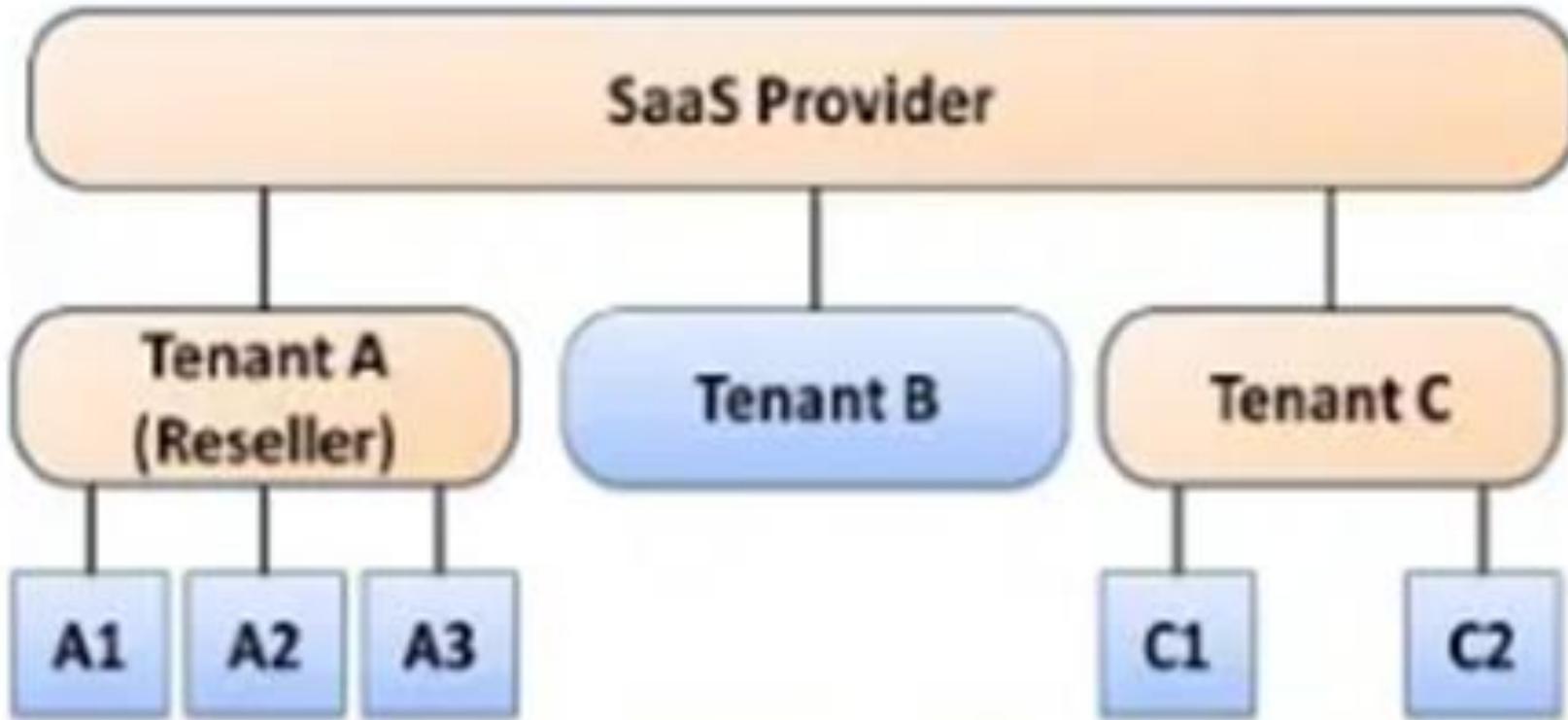


Fig 3 Tenant Tree



Tenant Data Management in Multi-Tenant Cloud Systems

- The representation of tenants is done by logical hierarchical representation.
- The data could be split as in the logical representation, by creating a data stores for tenants or merging smaller databases.
- For small applications with a less number of tenants, can share a single data store.
- There are **3 different models** for load balancing



Cloud Service Management

- The management of cloud infrastructure products and services is cloud management.
- Public clouds are operated by public cloud service providers, which provide the servers, storage, networking and data centre operations of the public cloud environment.
- With a third-party cloud management tool, users can also choose to manage their public cloud services.



Cloud Service Management

- Public cloud service users can typically choose from three categories of specific cloud provisioning:
- **User self-provisioning:** Users, usually via a web form or console interface, buy cloud services directly from the provider. On a per-transaction basis, the client pays.

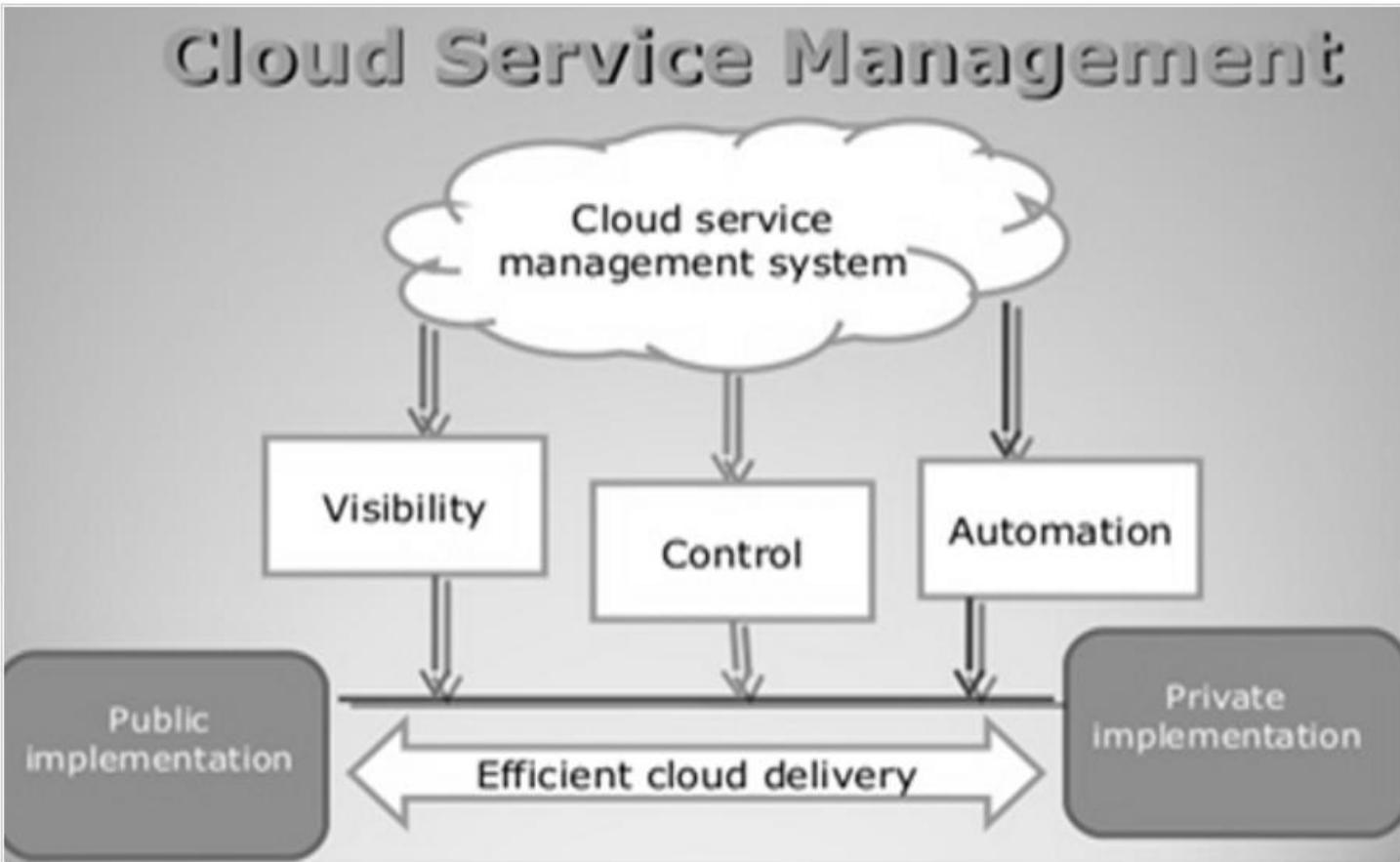


Cloud Service Management

- **Advanced provisioning:** A pre-determined sum of services scheduled in advance of operation is contracted in advance by customers. A flat fee or a monthly fee is charged by the consumer.
- **Dynamic provisioning:** When the client requires them, the provider allocates resources, and then decommissions them when they are no longer required. On a pay-per-use basis, the client is paid.



Cloud Service Management





On Demand Computing

Cloud Computing vs. Grid Computing

Grid Computing is computing technology that includes a combination of computer resources that offers seamless access to computing power and data storage capacity distributed over the globe.

Cloud Computing Developed from grid computing technology, cloud computing technology offers highly flexible on-demand provisioning of its resources



On-demand computing (ODC)

On-demand computing (ODC) is a business computing model that allows companies to provide access to computing resources as they become necessary, rather than full time

- a computing and communications infrastructure that facilitates flexible business service delivery and provides the basis for:
- Autonomic computing
- Fast response to external business-affecting changes.
- Adaptive business processes to protect revenues and contain costs
- Complex interactions inside and outside of organizational boundaries
- Resilience against external threats such as viruses, intrusions, and power outages.



How 'Cloud' differs from the 'Grid'?

Grid computing is focused towards solving a computational problem whereas Cloud computing is focused towards providing 'On Demand' services.

Types of services provided by Cloud Infrastructure

- a) SaaS (Software as a Service)
- b) PaaS (Platform as a Service)
- c) IaaS (Infrastructure as a Service)

Grid Computing is possible only with applications that can be parallelized whereas Cloud computing does not have any such restriction.



Grid Computing

- a special type of parallel computing that relies on complete connected to a network by a conventional network interface, such as Ethernet.
- combination of computer resources from multiple administrative domains to reach a common goal
- Grid size can vary by a considerable amount



Grid

In Grid, resource allocation is primarily controlled by the contract between the user (known as Virtual Organization) and allocator (known as Admin Domain).

Therefore a user, irrespective of his requirement will always get the resources ONLY as agreed upon in the contract. On the other hand in Cloud there is no such restriction.

Cloud supports 'On-Demand' provisioning as it follows the principle of 'Pay as you go'. Based on the user's requirements Cloud will allocate the required resources thus enabling easy scalability in order to achieve the desired SLA.



Cluster Computing

A group of linked computers, working together closely thus in many respects forming a single computer

- Cluster computing category
- High-availability (HA) clusters
- Load-balancing clusters
- Compute clusters Source:

http://www.rdi.ku.ac.th/Techno_ku60/res-96/index96.html



Cluster Computing categories

High-availability (HA):

clusters are linked together to share computational workload or function as a single virtual computer

- **Load-balancing clusters:** multiple computers are linked together to share computational workload or function as a single virtual computer
- **Compute clusters:** Often clusters are used primarily for computational purposes, rather than handling IO-oriented operations such as web service or databases



Cluster computing categorized by connection

- Closed system cluster computing
- Internal Connected
- high security
- only 1 IP address, each node cannot manage external data
- Open system cluster computing
- Direct connect to external network
- each node has its own IP address
- suitable for being web servers or ftp servers

Thank you

Any queries?