

UNIT III
UNDERSTANDING CLOUD SERVICES, APPLICATIONS AND CAPACITY PLANNING

3.1 Types of Services provided by Cloud

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

Service Oriented Architecture

- Elastic Computing
- On Demand Computing

3.2 Cloud services

3.2.1 Software as a Service

Software as a Service (SaaS) is a cloud computing model where software applications are hosted and maintained by a third-party provider and made available to customers over the internet. Here's an overview of SaaS and its key characteristics:

Characteristics of Software as a Service (SaaS):

1. Accessibility and Delivery:

- **Internet-Based Access:** Applications are accessed through a web browser or dedicated software interface, eliminating the need for on-premises installation.
- **Subscription Model:** Typically offered on a subscription basis, where customers pay a recurring fee (monthly or annually) for access to the software.

2. Managed by the Provider:

- **Infrastructure and Maintenance:** The SaaS provider manages the infrastructure, including servers, databases, and software updates.
- **Security and Compliance:** Providers ensure data security, backups, and compliance with industry regulations (e.g., GDPR, HIPAA).

3. Scalability and Flexibility:

- **Elasticity:** SaaS applications can scale horizontally to handle varying workloads and user demands without user intervention.
- **Customization:** Often allows some degree of customization through configuration options rather than full-scale code modifications.

4. Multi-Tenancy:

- **Shared Infrastructure:** Multiple customers (tenants) access the same instance of the software, benefiting from economies of scale and shared resources.
- **Isolation:** Ensures data security and privacy through logical and physical isolation mechanisms.

5. Examples of SaaS Applications:

- **Productivity Tools:** Microsoft Office 365, Google Workspace (formerly G Suite).
- **Customer Relationship Management (CRM):** Salesforce, HubSpot.
- **Enterprise Resource Planning (ERP):** SAP Business ByDesign, Oracle NetSuite.
- **Communication and Collaboration:** Slack, Zoom, Dropbox.

Advantages of SaaS:

- **Cost Efficiency:** Eliminates upfront costs for hardware and software licenses, with predictable subscription-based pricing.
- **Accessibility:** Accessible from any device with an internet connection, facilitating remote work and collaboration.
- **Scalability:** Easily scale up or down based on business needs without

additional infrastructure investments.

- **Maintenance and Updates:** Providers handle maintenance, updates, and security patches, ensuring optimal performance and security.

Considerations:

- **Data Security:** Relies on the provider's security measures and compliance certifications. Organizations must assess data handling practices.
- **Integration:** Compatibility with existing IT systems and APIs for seamless integration with other applications.
- **Vendor Lock-in:** Potential dependency on a single provider's ecosystem, affecting flexibility and data portability.

3.2.2 Platform as a Service

Platform as a Service (PaaS) is a cloud computing model that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the underlying infrastructure. Here's a comprehensive overview of PaaS and its key characteristics:

Characteristics of Platform as a Service (PaaS):

1. **Development Tools and Frameworks:**
 - **Integrated Environment:** PaaS offers a comprehensive development environment with tools, libraries, and frameworks necessary for application development.
 - **Support for Multiple Languages:** Typically supports various programming languages (Java, Python, .NET, Node.js, etc.) and development frameworks.
2. **Middleware Capabilities:**
 - **Built-In Services:** Includes middleware services such as databases, messaging queues, caching, and identity management.
 - **API Integration:** Enables integration with other services and APIs provided by the PaaS provider or third-party vendors.
3. **Deployment and Scalability:**
 - **Automated Deployment:** Simplifies deployment processes with automated provisioning and configuration management.
 - **Scalability:** Provides automatic scaling capabilities to handle varying workload demands, ensuring performance and availability.
4. **Managed Services:**
 - **Infrastructure Management:** PaaS providers manage underlying infrastructure components, including servers, networking, and storage.
 - **Maintenance and Updates:** Handles maintenance tasks such as software updates, security patches, and system monitoring.
5. **Benefits of PaaS:**
 - **Speed and Efficiency:** Accelerates application development and deployment timelines by providing ready-to-use development environments and services.
 - **Cost Savings:** Reduces upfront infrastructure costs and operational expenses associated with managing hardware and software infrastructure.
 - **Focus on Innovation:** Enables developers to focus on application logic and innovation rather than infrastructure management.
6. **Examples of PaaS Offerings:**
 - **Microsoft Azure App Service:** Enables developers to build, deploy, and scale web apps and APIs without managing infrastructure.
 - **Google App Engine (GAE):** Provides a platform for developing and hosting web applications using Google's infrastructure.
 - **AWS Elastic Beanstalk:** Allows quick deployment and management of applications using AWS cloud services.

Use Cases for Platform as a Service (PaaS):

- **Web Application Development:** Rapidly develop and deploy web applications and APIs using pre-built components and services.
- **Microservices Architecture:** Build and manage microservices-based applications with ease of scalability and integration.
- **DevOps Integration:** Facilitate continuous integration and deployment (CI/CD) workflows with built-in automation and monitoring tools.
- **Data Analytics and IoT:** Utilize PaaS for processing and analyzing large volumes of data or managing IoT devices and applications.

Considerations:

- **Vendor Lock-in:** Evaluate compatibility and portability of applications and data when choosing a PaaS provider.
- **Security and Compliance:** Assess security measures, data protection practices, and compliance certifications offered by the PaaS provider.
- **Integration with Existing Systems:** Ensure seamless integration with existing IT infrastructure, databases, and external APIs.

3.2.3 Open SaaS and SOA

- Open SaaS (Software as a Service) and SOA (Service-Oriented Architecture) are two related concepts in the realm of software development and cloud computing. Here's an overview of each concept and how they relate to modern software architecture:
- **Open SaaS (Software as a Service):**
- **Definition:**
- **SaaS Model:** Open SaaS refers to a SaaS (Software as a Service) application that offers greater flexibility, customization, and interoperability compared to traditional closed or proprietary SaaS offerings.
- **Customization:** Allows customers to customize and extend the functionality of the SaaS application through open APIs, plugins, or integrations with third-party services.
- **Open APIs:** Provides well-documented APIs that enable developers to integrate the SaaS application with other systems or build custom extensions.
- **Key Characteristics:**
- **Interoperability:** Supports integration with external applications and services through open standards and APIs.
- **Flexibility:** Offers configurable options and tools that enable users to tailor the SaaS solution to meet specific business needs.
- **Community Collaboration:** Encourages collaboration and community-driven innovation through open-source components, plugins, or marketplace ecosystems.
- **Examples:**
- **WordPress:** An open SaaS platform for website creation and content management, allowing users to extend functionality through plugins and themes.
- **Salesforce AppExchange:** A marketplace for extending Salesforce CRM with third-party apps and integrations, showcasing the open SaaS model.
- **Service-Oriented Architecture (SOA):**
- **Definition:**
- **Architectural Approach:** SOA is an architectural style that structures software applications as a collection of loosely coupled services.
- **Service:** Each service implements a specific business functionality and communicates with other services through well-defined interfaces (typically APIs).
- **Interoperability:** Promotes interoperability and reuse of services across different applications and platforms.

- **Key Characteristics:**
- **Modularity:** Decomposes complex applications into smaller, manageable services that can be developed, deployed, and maintained independently.
- **Scalability:** Supports horizontal scaling by distributing services across multiple nodes or containers.
- **Flexibility:** Allows services to be composed and orchestrated to fulfill specific business processes or workflows dynamically.
- **Examples:**
- **Microservices:** A modern implementation of SOA where applications are built as a collection of small, independent services that communicate through APIs.
- **Enterprise Service Bus (ESB):** Middleware technology that facilitates communication and integration between diverse applications and services in an SOA environment.
- **Relationship Between Open SaaS and SOA:**
- **Interoperability:** Both concepts emphasize the importance of interoperability and integration through well-defined APIs and standards.
- **Flexibility and Customization:** Open SaaS leverages SOA principles to provide customizable and extensible services that can be integrated with other systems.
- **Modularity and Reusability:** SOA's modularity and service reusability principles support the development and deployment of open SaaS applications that can adapt to evolving business needs.
-

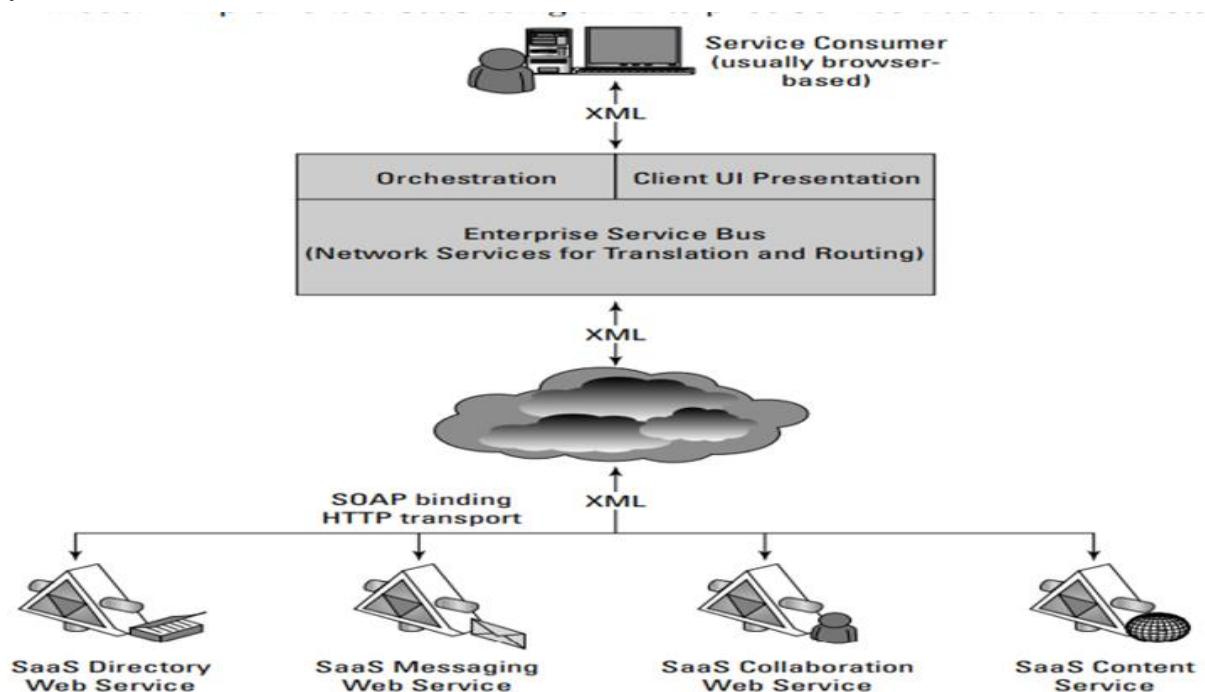


Fig 3.1 A modern implementation of SaaS using Enterprise service bus architected with SOA.

3.2.4 Defining Identity as a Service (IDaaS)

Identity as a Service (IDaaS) is a cloud-based service that provides identity and access management (IAM) capabilities to users and organizations. Here's a comprehensive definition and overview of IDaaS:

Definition of Identity as a Service (IDaaS):

1. Cloud-Based Identity Management:

- **Service Model:** IDaaS is a cloud-delivered service model that simplifies identity and access management (IAM) for users, applications, and devices.
- **Core Functions:** Provides centralized authentication, authorization, user provisioning, and access governance functionalities.

2. Key Features and Capabilities:

- **Single Sign-On (SSO):** Enables users to access multiple applications and services with a single set of credentials.
- **Identity Federation:** Integrates with external identity providers (IdPs) using protocols like SAML, OAuth, and OpenID Connect for seamless authentication across different domains.
- **Multi-Factor Authentication (MFA):** Enhances security by requiring additional verification methods (e.g., SMS, biometrics) beyond passwords.
- **User Provisioning and De-Provisioning:** Automates user lifecycle management, including onboarding, offboarding, and role-based access control (RBAC).
- **Access Governance:** Monitors and audits user activities, enforces security policies, and ensures compliance with regulatory requirements.

3. Benefits of IDaaS:

- **Scalability:** Easily scales to support growing numbers of users and applications without the need for additional infrastructure investments.
- **Ease of Integration:** Integrates with existing IT systems, applications, and cloud services through standardized protocols and APIs.
- **Cost Efficiency:** Reduces operational costs associated with managing on-premises IAM infrastructure and support.
- **Enhanced Security:** Provides robust security controls, including MFA, adaptive authentication, and continuous monitoring, to protect against unauthorized access and data breaches.

4. Use Cases for IDaaS:

- **Enterprise SSO:** Streamlines access to corporate applications and resources for employees, partners, and contractors.
- **Customer Identity and Access Management (CIAM):** Manages user identities and access across customer-facing applications and services.
- **Cloud Application Integration:** Secures access to cloud-based applications (SaaS) while ensuring compliance and data protection.

5. Examples of IDaaS Providers:

- **Okta:** Offers a comprehensive IDaaS platform with SSO, MFA, lifecycle management, and API access management capabilities.
- **Azure Active Directory (Azure AD):** Microsoft's cloud-based identity and access management service integrated with Microsoft 365 and Azure services.
- **Ping Identity:** Provides IDaaS solutions for secure access management across hybrid IT environments.

Figure 3.3 shows how these different standards form an identity service framework

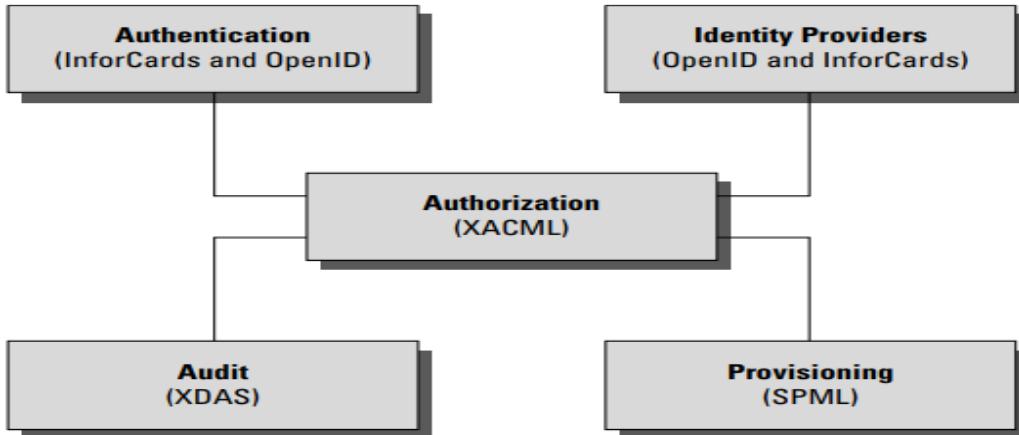


Fig. 3.3 Open standards that support an IDaaS infrastructure for cloud computing

3.2.5 Defining Compliance as a Service (CaaS)

Compliance as a Service (CaaS) is a cloud-based service model that helps organizations manage and maintain regulatory compliance requirements efficiently. Here's an overview and definition of Compliance as a Service:

Definition of Compliance as a Service (CaaS):

1. Cloud-Based Compliance Management:

- **Service Model:** CaaS leverages cloud computing to provide tools, resources, and expertise for managing regulatory compliance requirements.
- **Core Functionality:** Offers automated solutions, workflows, and documentation to ensure adherence to industry regulations, standards, and policies.

2. Key Features and Capabilities:

- **Regulatory Monitoring:** Monitors and tracks changes in regulatory requirements, ensuring organizations stay updated with compliance obligations.
- **Audit and Assessment Management:** Facilitates internal and external audits, assessments, and certifications to verify compliance status.
- **Policy and Procedure Management:** Centralizes policies, procedures, and controls to align with regulatory frameworks and industry best practices.
- **Risk Management:** Identifies, assesses, and mitigates compliance risks through proactive monitoring and reporting.
- **Incident Response and Remediation:** Provides frameworks and tools to respond to compliance incidents, breaches, and violations promptly.

3. Benefits of Compliance as a Service:

- **Cost Efficiency:** Reduces costs associated with manual compliance management processes and dedicated compliance teams.
- **Scalability:** Scales resources and capabilities based on organizational needs and regulatory changes without additional infrastructure investments.

- **Automation and Efficiency:** Automates compliance workflows, tasks, and reporting, improving operational efficiency and accuracy.
- **Expertise and Guidance:** Access to specialized compliance expertise, advice, and best practices from CaaS providers.
- **Continuous Monitoring:** Offers continuous monitoring and real-time alerts on compliance status and deviations.

4. Use Cases for Compliance as a Service:

- **Financial Services:** Ensures compliance with regulations like GDPR, PCI DSS, and SOX for financial transactions and data protection.
- **Healthcare:** Manages HIPAA compliance for protecting patient health information and maintaining data security.
- **Government and Public Sector:** Achieves regulatory compliance with standards and policies governing public sector operations.
- **Global Organizations:** Supports multinational compliance requirements across different jurisdictions and legal frameworks.

5. Examples of CaaS Providers:

- **Datica:** Specializes in compliance and security solutions for healthcare organizations, offering HIPAA compliance as a service.
- **CompliancePoint:** Provides CaaS solutions for data privacy, GDPR compliance, and cybersecurity standards across industries.
- **AWS Compliance Center:** Amazon Web Services offers resources and tools to help customers achieve and maintain regulatory compliance in the cloud.

3.2. 6 Capacity Planning

3.6.1 Baseline measurements

There are two important overall workload metrics in this LAMP system:
 Page views or hits on the Web site, as measured in hits per second
 Transactions completed on the database server, as measured by transactions per second.

In Figure 3.4, the historical record for the Web server page views over a hypothetical day, week, and year are graphed. These graphs are created by summing the data from the different servers

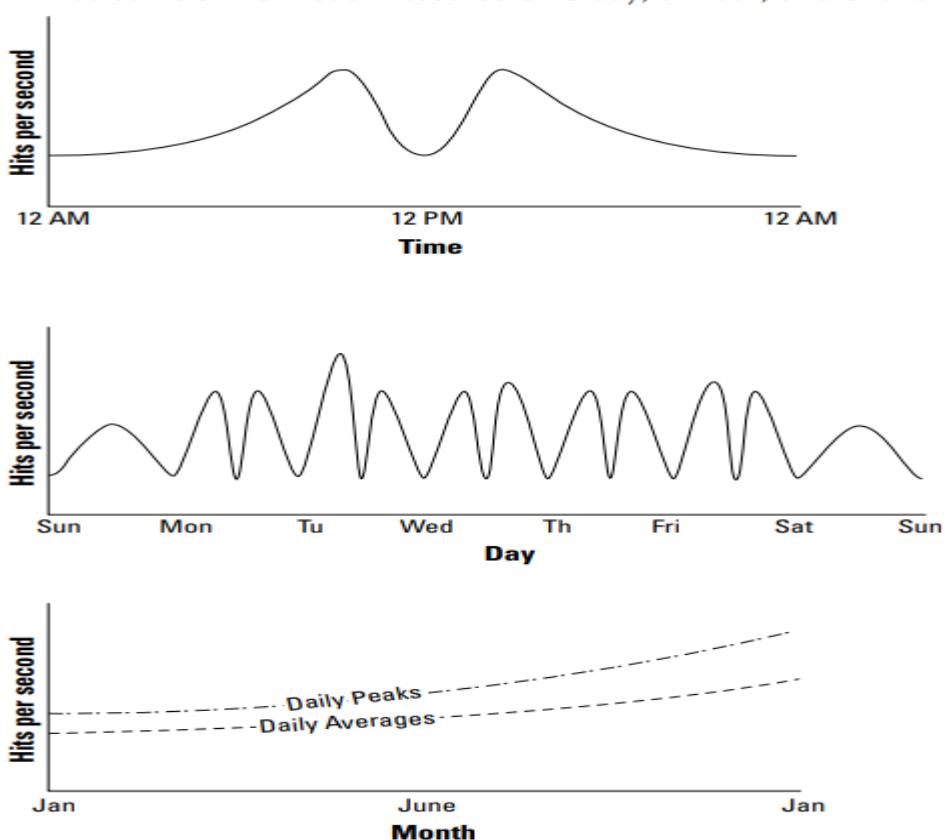


Fig 3.4 A Web servers' workload measured on a day, a week, and over the course of a year

3.6.2 System metrics Capacity planning must measure system-level statistics, determining what each system is capable of, and how resources of a system affect system-level performance.

A machine instance (physical or virtual) is primarily defined by four essential resources:

- CPU
- Memory (RAM)
- Disk
- Network connectivity.

Each of these resources can be measured by tools that are operating-system-specific, but for which tools that are their counterparts exist for all operating systems.

RRDTool is a utility that can capture time-dependent performance data from resources such as a CPU load, network utilization (bandwidth), and so on and store the data in a circular buffer. It is commonly used in performance analysis work. Figure 3.5 shows some of the examples from a gallery of RRDTool graphs found at <http://oss.oetiker.ch/rrdtool/gallery/>.

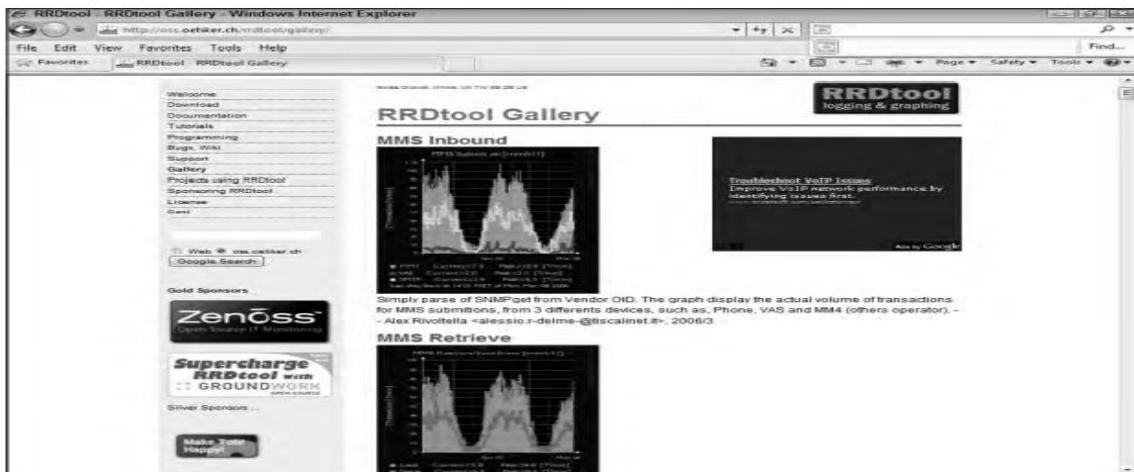


Fig 3.5 RRDTool lets you create historical graphs of a wide variety of performance data. Some samples are shown in the gallery at <http://oss.oetiker.ch/rrdtool/gallery/>. Table 3.1 lists some LAMP performance testing tools

Tool Name	Web Site	Developer	Description
Alertra	http://www.alertra.com	Alertra	Web site monitoring service
Cacti	http://www.cacti.net	Cacti	Open source RRDTool graphing module
Collectd	http://www.collectd.org/	collectd	System statistics collection daemon
Dstat	http://dag.wieers.com/home-made/dstat/	DAG	System statistics utility; replaces vmstat, iostat, netstat, and dstat
Ganglia	http://www.ganglia.info	Ganglia	Open source distributed monitoring system
Gomez	http://www.gomez.com	Gomez	Commercial third-party Web site performance monitor
GraphClick	http://www.arizona-software.ch/graphclick/	Arizona	A digitizer that can create a graph from an image
GroundWork	http://www.groundworkopensource.com/	Groundwork's Open Source	Network monitoring solution
Hyperic HQ	http://www.hyperic.com	Spring Source	Monitoring and alert package for virtualized environments
Keynote	http://www.keynote.com	Keynote	Commercial third-party Web site performance monitor
Monit	http://www.tildeslash.com/monit	Monit	Open source process manager
Munin	http://munin.projects.linpro.no/	Munin	Open source network resource monitoring tool
Nagios	http://www.nagios.org	Nagios	Metrics collection and event notification tool
OpenNMS	http://www.opennms.org	OpenNMS	Open source network management platform
Pingdom	http://www.pingdom.com	Pingdom	Uptime and performance monitor
RRDTool	http://www.RRDTool.org/	Oetiker+Partner AG	Graphing and performance metrics storage utility
SiteUpTime	http://www.siteuptime.com	SiteUpTime	Web site monitoring service
Zabbix	http://www.zabbix.com	Zabbix	Performance monitor
ZenOSS	http://www.zenoss.com/	Zenoss	Operations monitor, both open source and commercial versions

Table 3.1 LAMP Performance Monitoring Tools