

Cloud Computing

Architectures & Models

Alex Mwotil

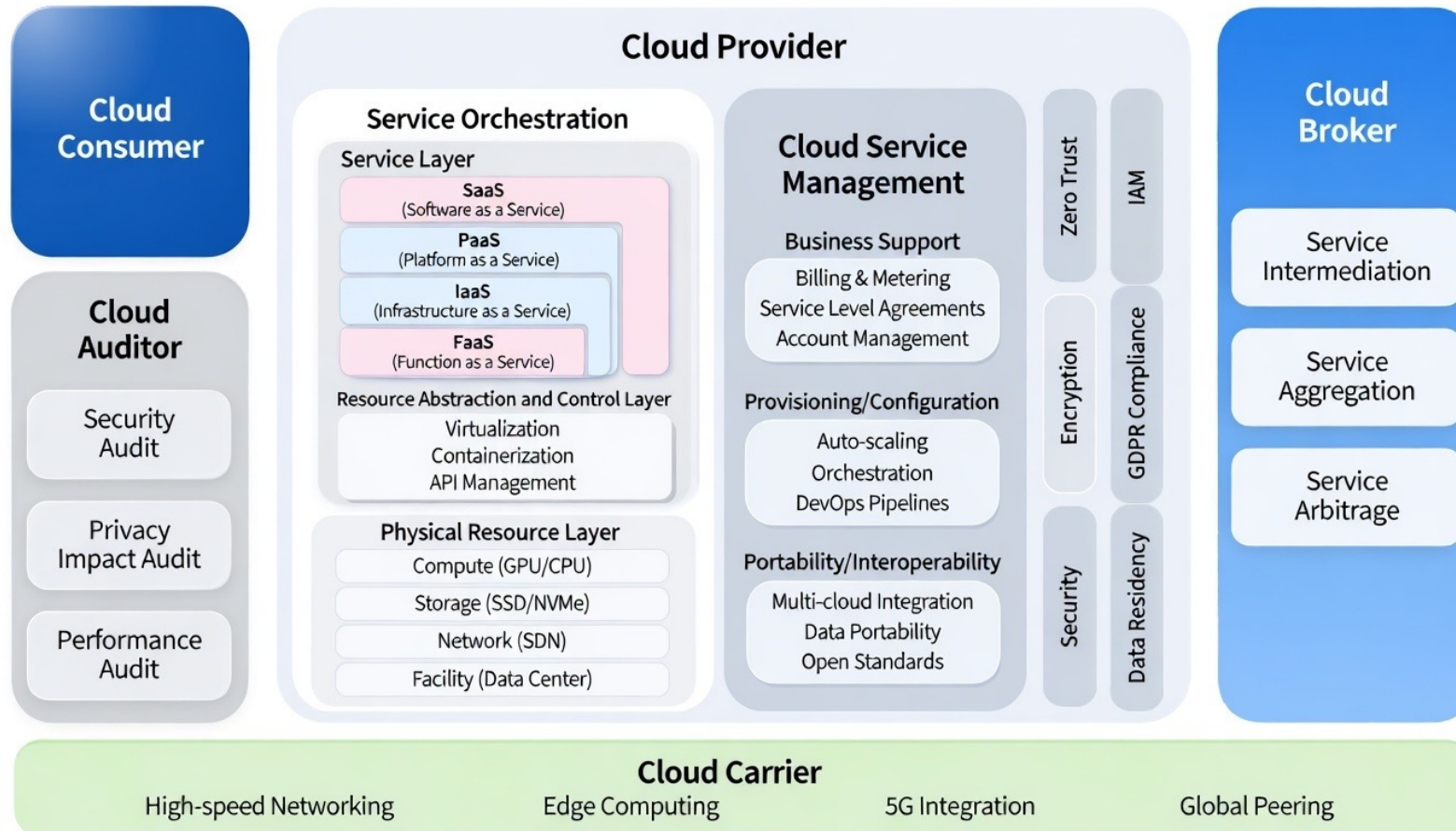
Overview

- Architecture
- Deployment Models
- Service Models
- Conclusion

Architecture

- **Google:** Layout that connects all the necessary components and technologies required for cloud computing.
- **IBM:** Fundamental components of a cloud computing environment
- **AWS:** the structured design, organization, and implementation of cloud
- **Azure:** the design and structure of how cloud services are delivered – the components that work together

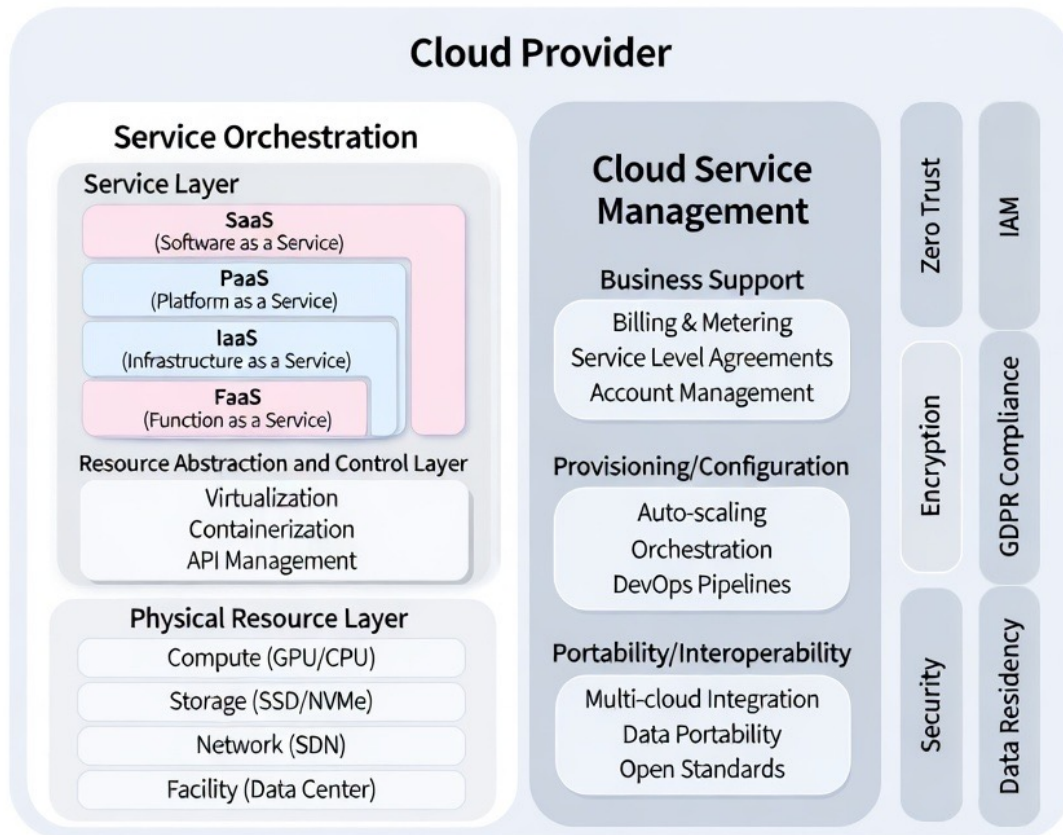
Architecture



Actor/role-based architecture of Cloud Computing

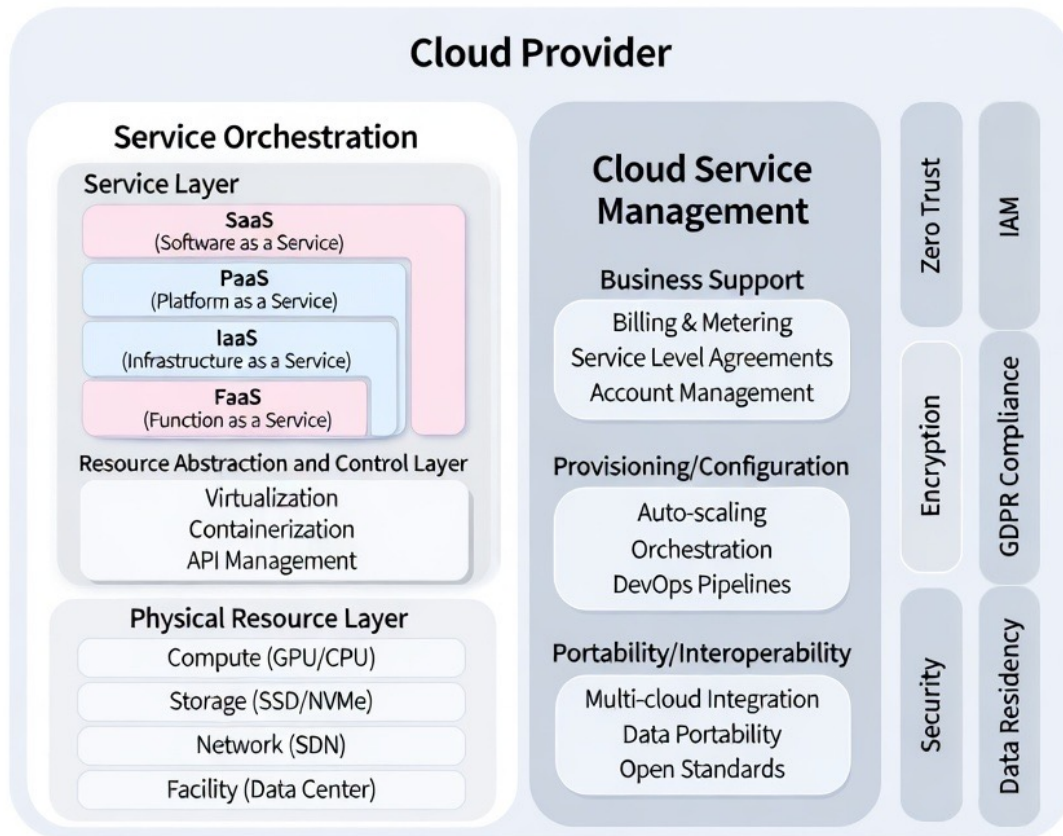
- **Cloud Provider**
- **Cloud Consumer**
- **Cloud Auditor**
- **Cloud Broker**
- **Cloud Carrier**

Architecture



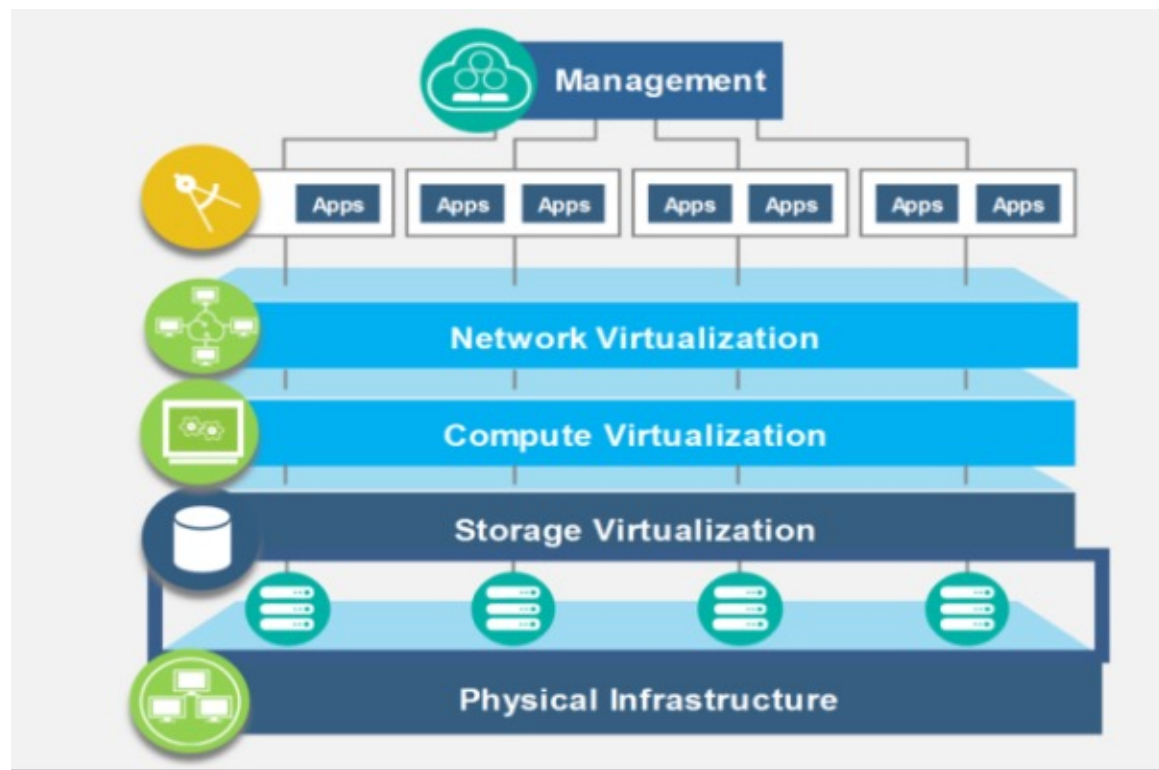
- A **person, organization, or entity** responsible for making a service available to interested parties.
- **Service Layer** — Delivers the actual cloud service models to consumers (SaaS, PaaS, IaaS, FaaS)
- **Resource Abstraction and Control Layer** — The virtualization/control plane that hides physical hardware details.
- **Physical Resource Layer** — The real hardware foundation in data centers.
 - Pooled and virtualized resources (servers, disks, switches)

Architecture



- **Cloud Service Management**
 - Provisioning/Configuration, Auto-scaling, Orchestration, DevOps Pipelines, Portability/Interoperability, Billing & Metering, SLAs, Account Management.
- **Security & Compliance**
 - IAM, Encryption, Zero Trust, GDPR/compliance, Data Residency — embedded throughout provider operations.

Cloud Service Provider

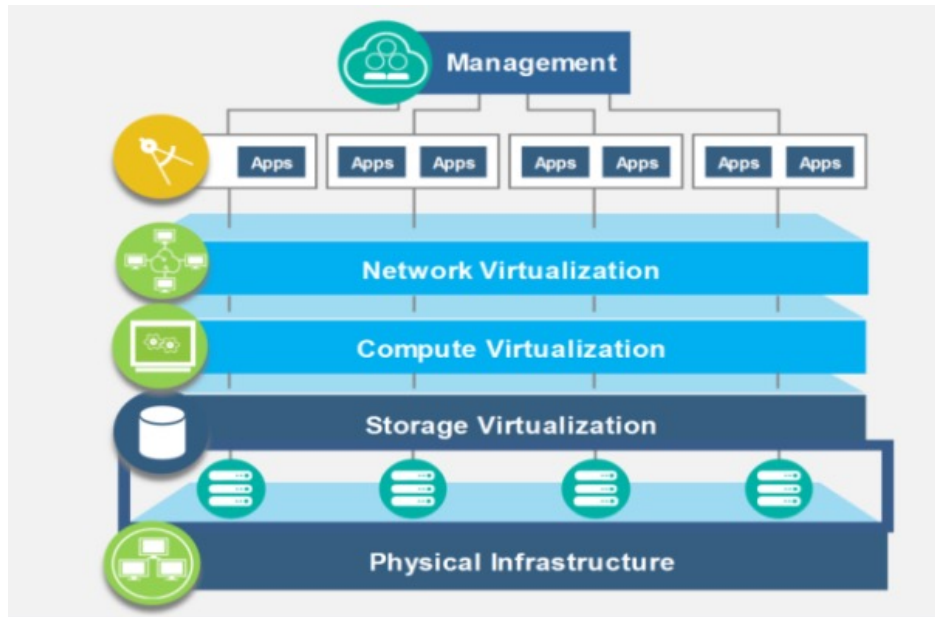


- **Physical Infrastructure**
 - Real, tangible hardware in 1+ data centers.
 - Servers (with CPUs/GPUs), memory (RAM), physical storage devices (SSDs, NVMe, HDDs)
 - Networking gear (switches, routers, cables), power/cooling systems, and facilities.
 - Raw compute, storage, and connectivity.
 - Virtualized – multiple users

Cloud Service Provider

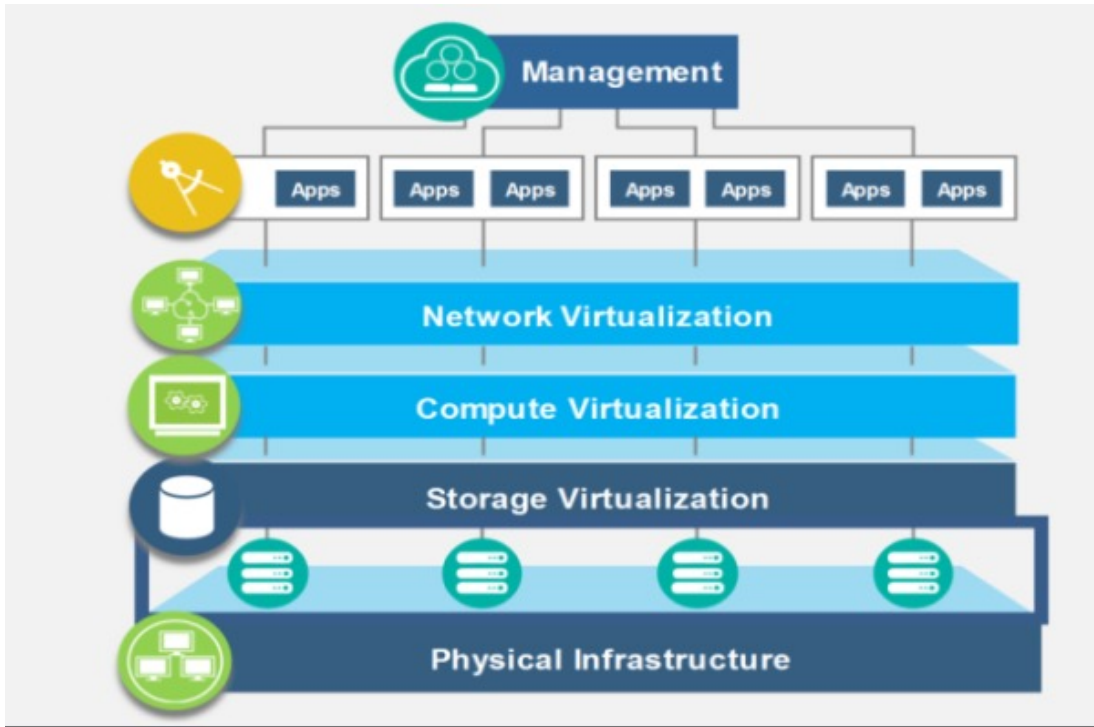


Cloud Service Provider



- **Virtualization**
 - **Storage** – Virtual disks, distributed file systems, object storage, block storage
 - Abstracts and pool → scalable virtual storage
 - **Compute** – Convert physical servers into multiple isolated VMs or containers
 - **Network** – Decouple the network from physical hardware (use SDN)
 - Virtual networks, subnets, VLANs/VXLANs, virtual routers, load balancers, firewalls, VPNs, and overlays.

Cloud Service Provider



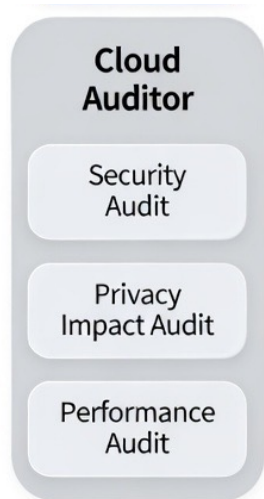
- **Management**
 - Orchestration, monitoring, and control plane
 - Tools for provisioning, configuration, auto-scaling, billing/metering, monitoring, logging, security policies, and API gateways.
- **Apps**
 - User services/applications (SaaS apps, PaaS platforms, custom workloads)
 - Deployed and managed on top of the virtualized layers.

Cloud Consumer



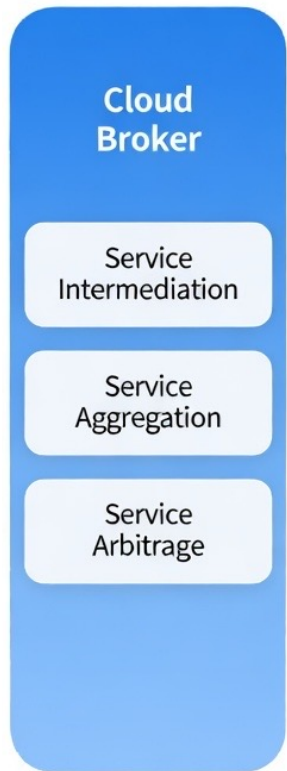
- A **person** or **organization** that maintains a business relationship with, and uses service from, **Cloud Providers**.
 - **Browses** service catalogue
 - **Requests / provisions** services
 - **Uses** the cloud services
 - **Pays / manages** billing
 - **Monitors** usage, performance, and compliance (SLAs)
 - **Manages** own data, applications, identities, and security

Cloud Auditor



- **Independent** Third-Party Entity / Organization
 - Firms (e.g., Deloitte, PwC, KPMG, Ernst & Young).
- **Conducts** independent assessments
- **Collects** evidence from Provider / Consumer
- **Issues** reports / opinions
- **Focus:** Security, Privacy, Performance, Compliance

Cloud Broker



- **Intermediary** Entity / Organization
- **Acts** as a single point of entry for Consumers
- **Manages** use, performance, delivery of services
- **Negotiates** relationships between Providers & Consumers
- **Provides** value-added capabilities

Architecture

Global Peering

5G Integration

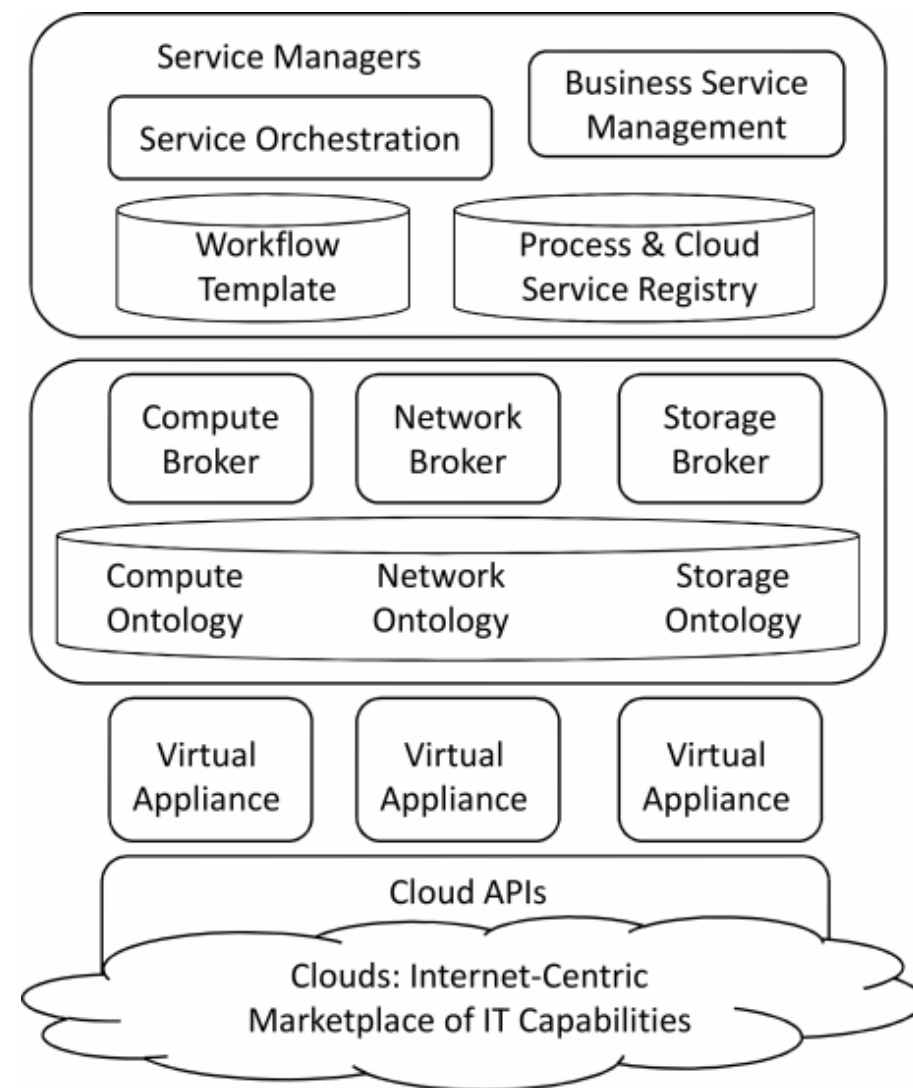
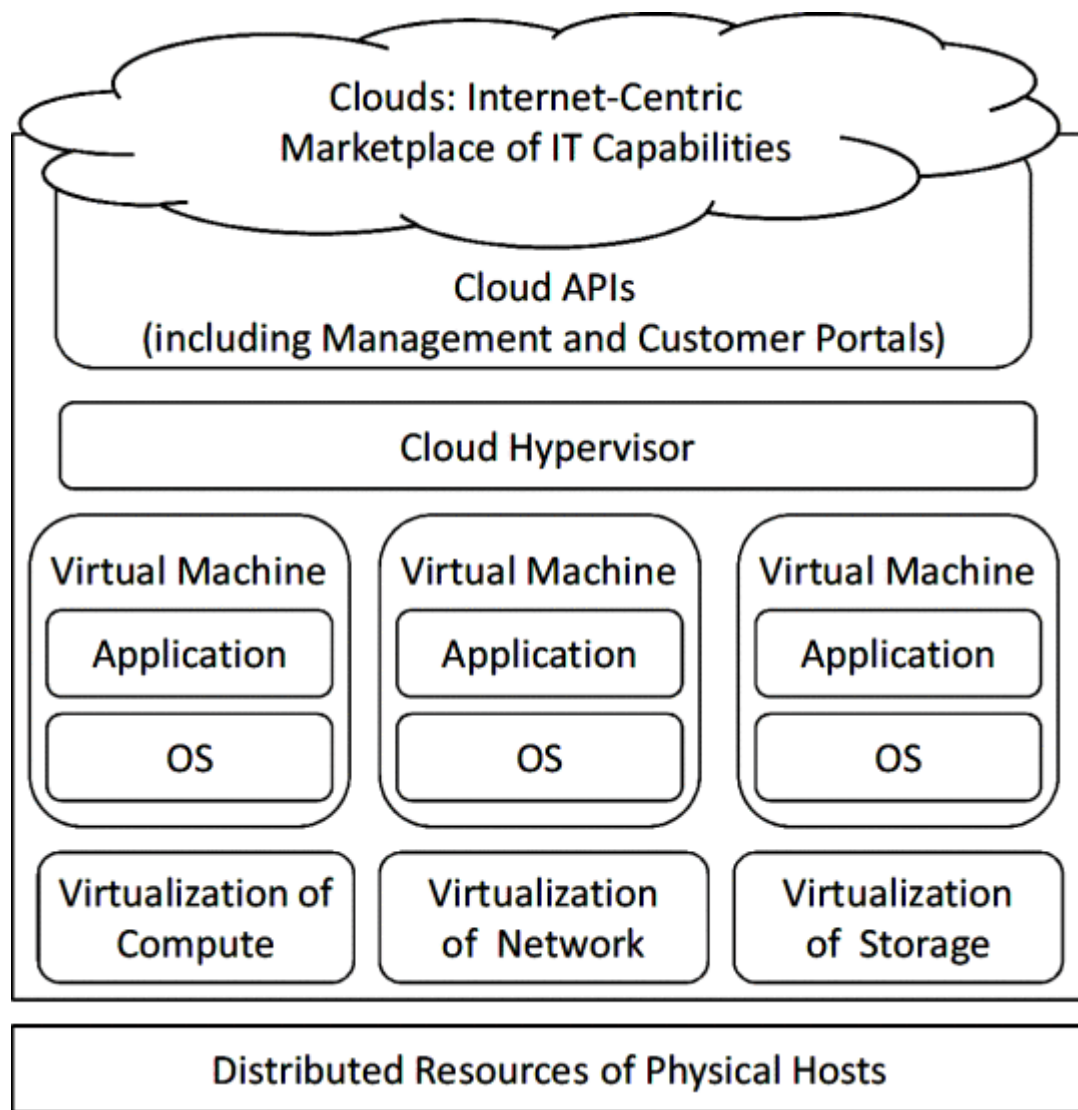
Cloud Carrier

Edge Computing

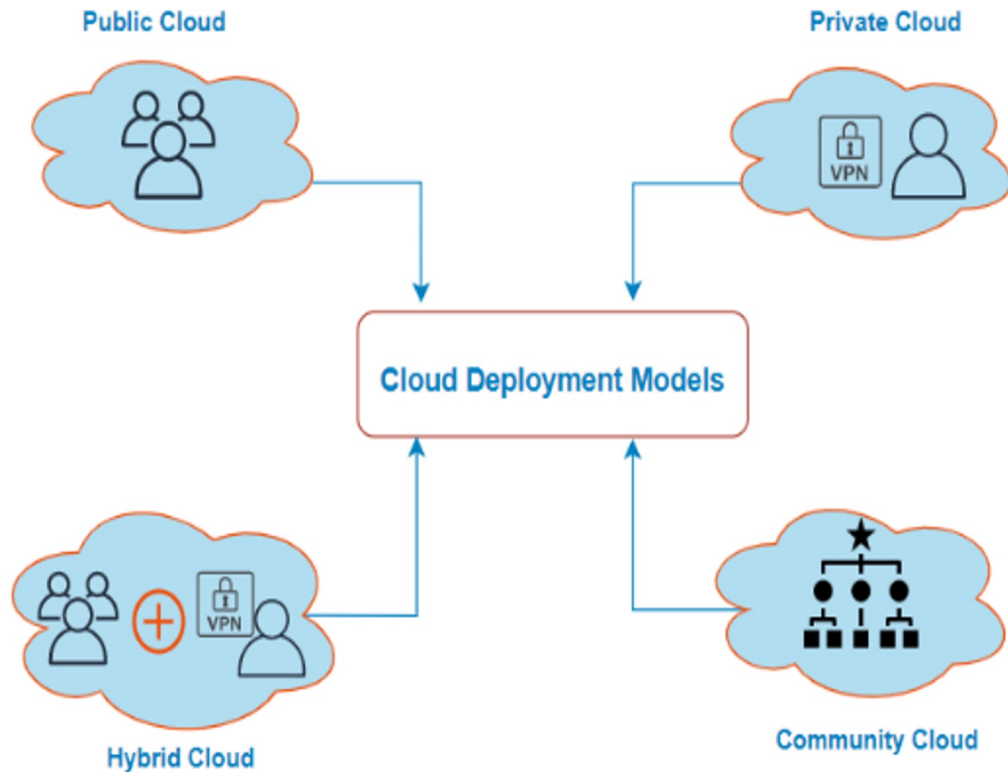
High-speed Networking

- **Network / Telecom Provider** or Transport Agent
- **Provides** connectivity & transport
- **Ensures** data transmission between Provider & Consumer
- **Maintains** network infrastructure & access points
- **Supports** SLAs for performance & security

Architecture

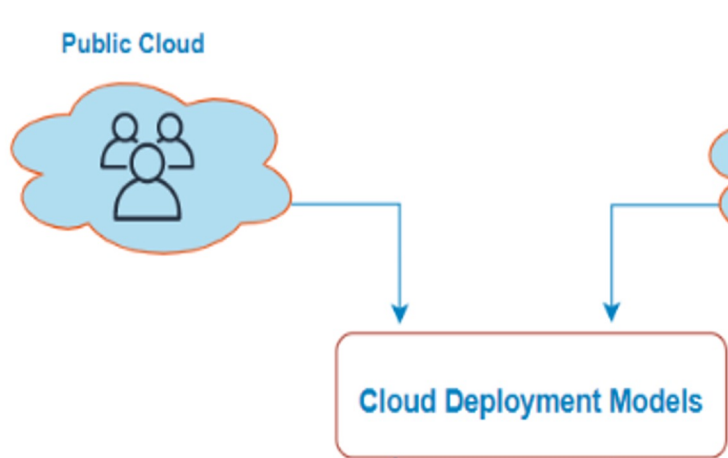


Deployment Models



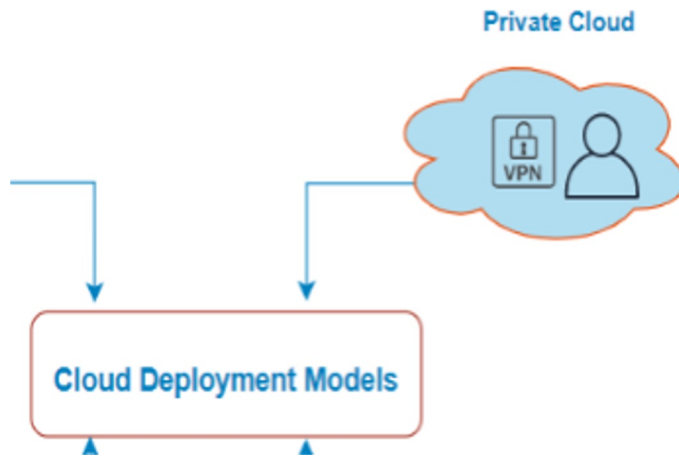
- The **virtual computing environment**
 - Choice of the model depends on how much data you want to store and who has access to the infrastructure.
- Four deployment models
 - **Public**
 - **Private**
 - **Community**
 - **Hybrid**

Deployment Models



- **Public Cloud**
 - Generally hosted for the client/users with easy access.
 - Better suited for commercial purposes
 - Economical due to the very low cost of their services.
- **Examples**
 - IBM, Google, Amazon, Microsoft,

Deployment Models

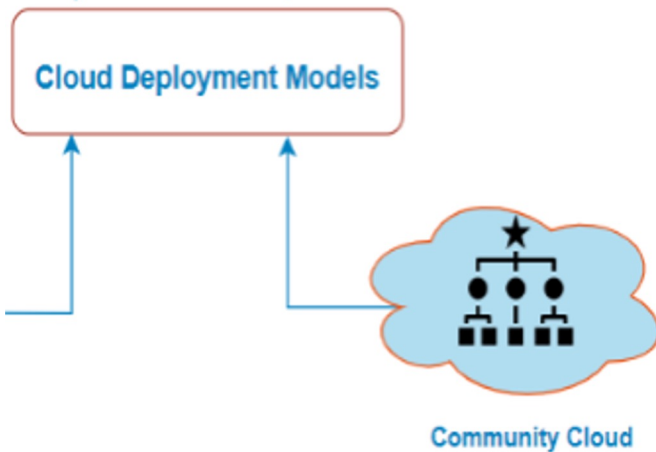


- **Private Cloud**
 - Generally hosted on the enterprise's premises.
 - Many organizations with security concerns on Public
 - Under control of only specific executives.
 - AKA corporate or internal cloud
 - **Examples**
 - MUK Cloud

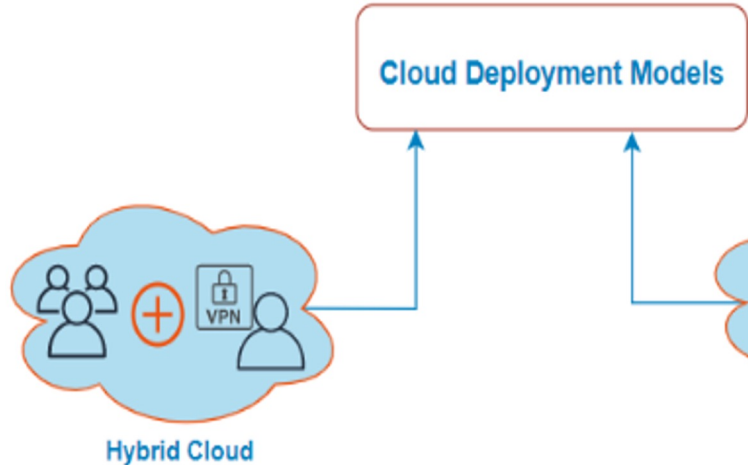
Deployment Models

- **Community Cloud**

- Similar to the private cloud, but..
- Used and maintained by similar need organizations
- Consistent security, privacy, and performance requirements
- Suitable for organizations on joint projects.
- Supports the development, management, and implementation of the project.
- All users share the costs.

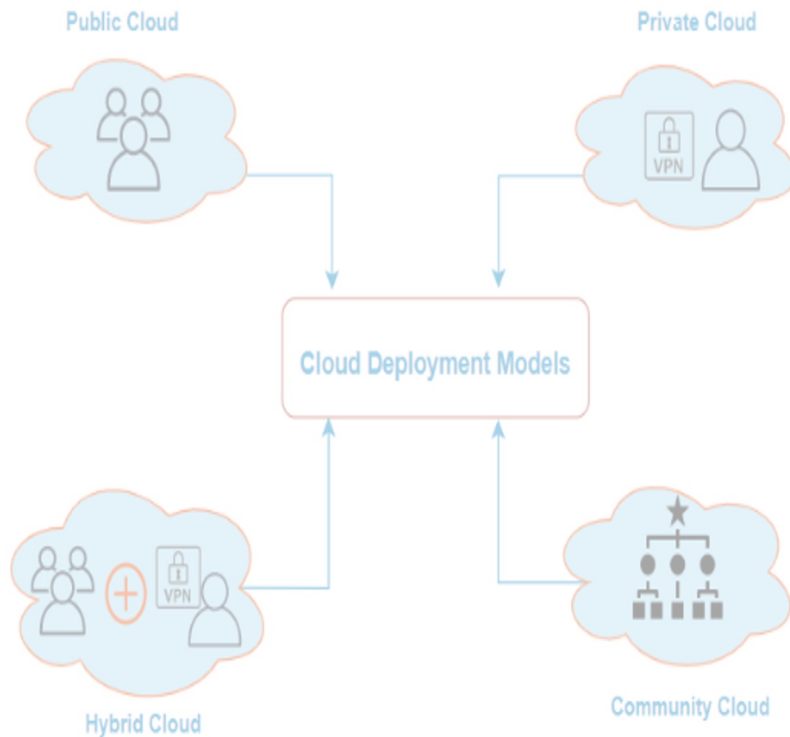


Deployment Models



- **Hybrid Cloud**
 - Combination of 2+ cloud servers
 - Private, public or community combined in a single architecture
 - Managed as a single individual unit.
 - Important data is generally preferred in a private cloud

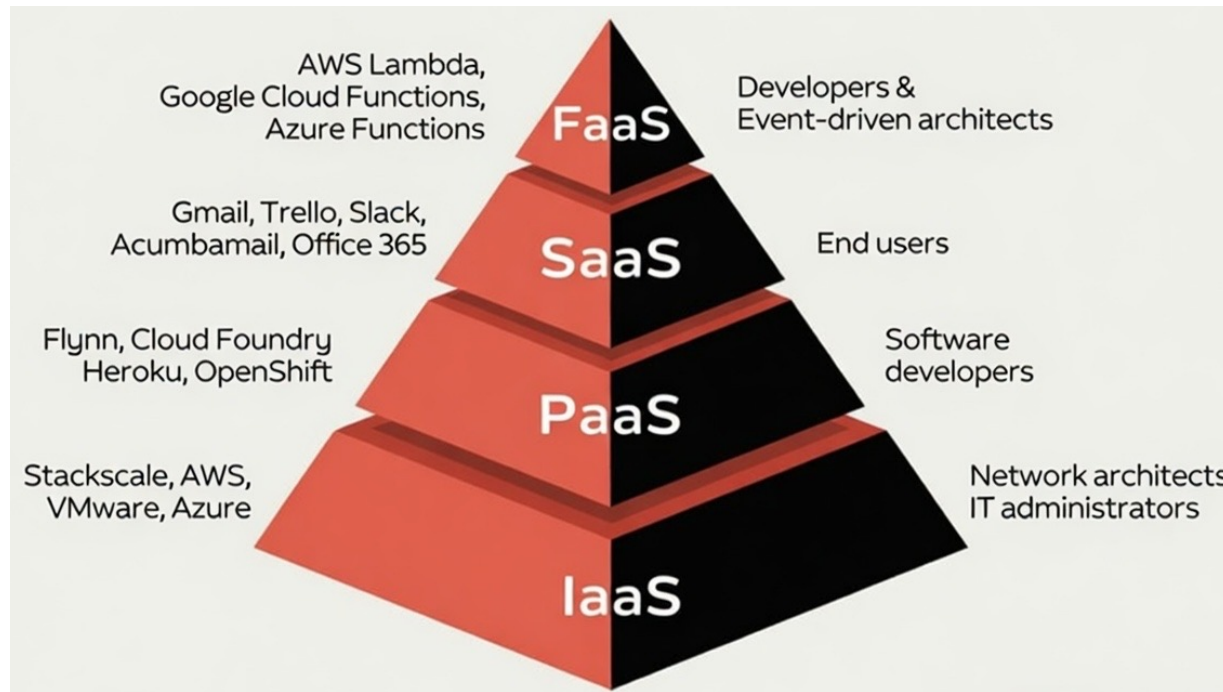
Deployment Models



- **Conclusion**

- Each Model has a unique offering and can immensely add value to a business.
- For small to medium-sized businesses, a public cloud is an ideal model to start with.
- As requirements change, switch to another model
- Model depends on needs or requirements

Service Models



- Define **offerings** of a cloud service provider
- **Control & responsibility** between customer & provider
- A **hierarchy** of increasing **abstraction** and decreasing mgmt **overhead**
 - FaaS → Highest abstraction
 - IaaS → Increased overhead
- The higher you go in the pyramid, the less you manage infrastructure
 - More focus on the business value

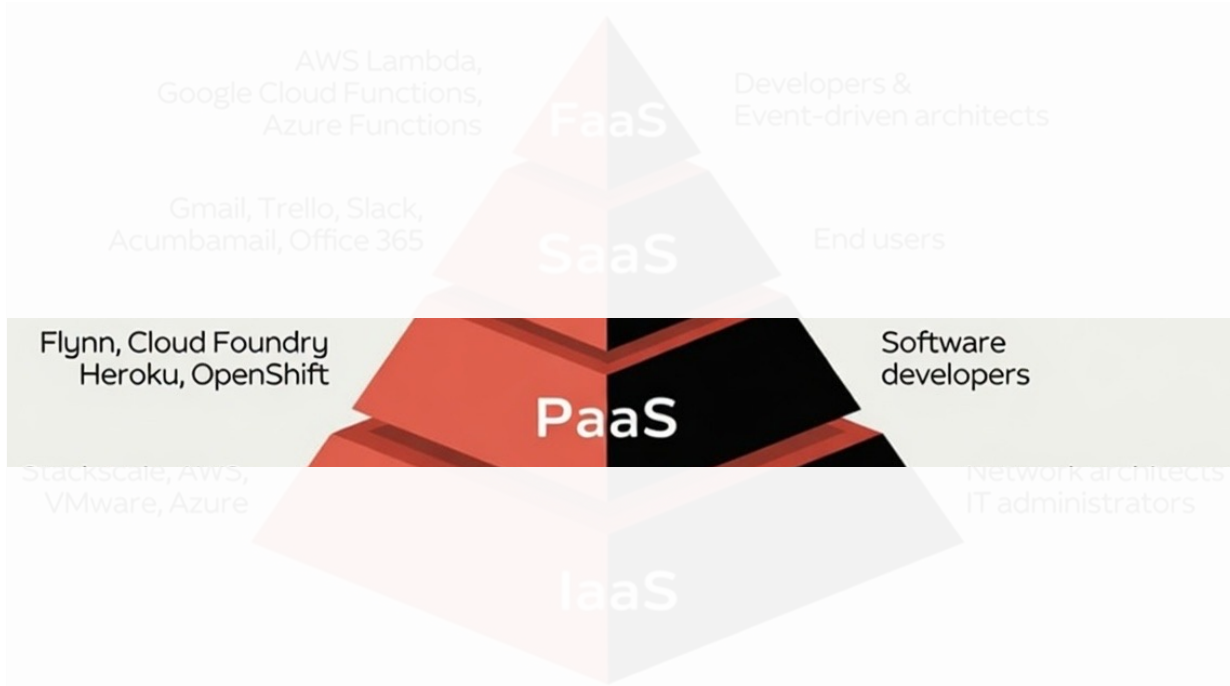
Service Models



Infrastructure as a Service (IaaS)

- Provides **virtualized computing** resources over the internet
- **Rent** virtual machines, storage, networks, load balancers on-demand
- Customer - **Full control** over operating system, middleware, runtime
- Flexible — **install** any software, **configure** servers as needed
- **Provider manages:** physical hardware, virtualization layer, DC
- **Customer manages:** OS, patches, applications, data, security, backups

Service Models



Platform as a Service

- Provides a complete development and deployment **environment**
- Developers **upload code** — platform handles runtime, middleware, OS, scaling
- **Built-in tools:** databases, caching, queues, monitoring, CI/CD integration
- **Faster development** — focus on code and business logic, not infrastructure
- **Provider manages:** servers, OS patches, training
- **Customer manages:** code, data, configuration

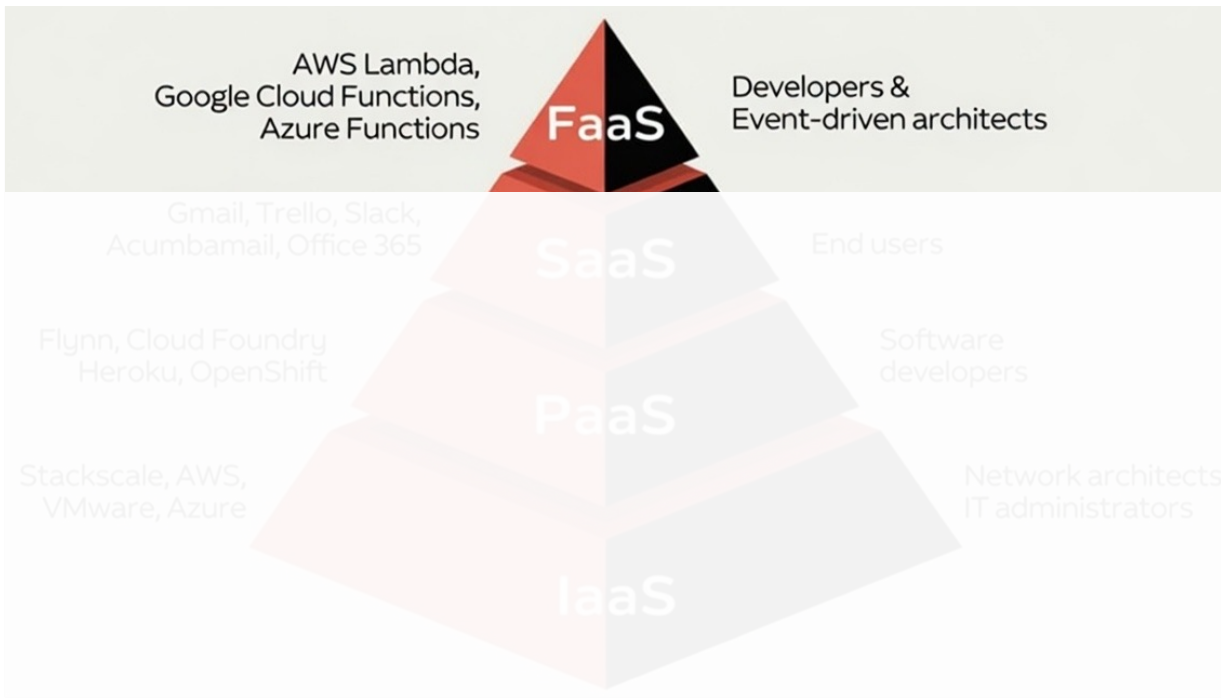
Service Models



Software as Service

- Complete, **ready-to-use** applications delivered over the internet
- Accessed via web browser or mobile app — no installation required
- **Provider handles everything:** application, updates, security, servers,
- **Customer manages only:** user accounts, data/content, basic configuration
- **Subscription-based** pricing (per user/month)

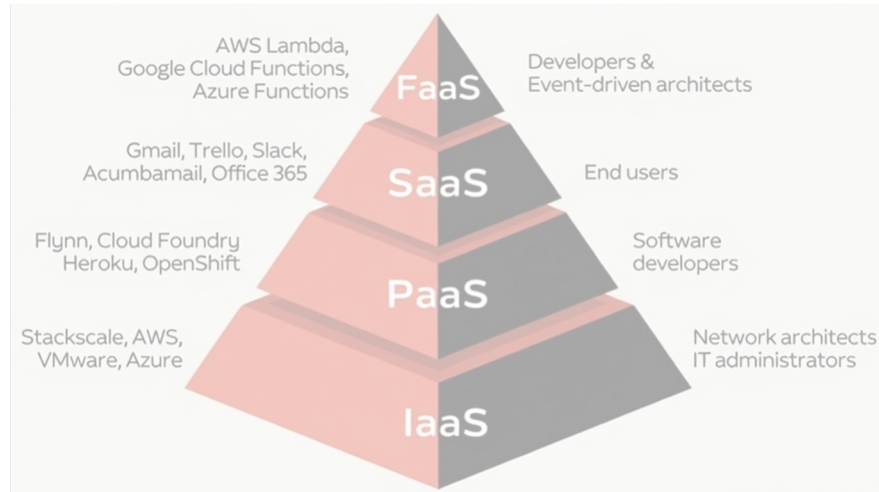
Service Models



Function as a Service (FaaS) | Serverless

- Developers write and deploy small, single-purpose **functions**
- Code runs only when **triggered** by events (HTTP request, file upload, message queue, timer, etc.)
- **Automatic scaling** — from zero to thousands of instances instantly
- **Pay only** for actual execution time (milliseconds), not idle servers
- Provider manages runtime, servers, patching, scaling, availability

Service Models



Conclusion

- Cloud service models are a spectrum of **responsibility**: Full infrastructure control (IaaS) → zero infrastructure management (FaaS/SaaS).
- The higher in the pyramid → the less you manage → the faster you innovate → the lower the operational overhead.
 - IaaS → Maximum flexibility & control (ideal for custom environments, migrations)
 - FaaS → Ultimate efficiency for event-driven, scalable workloads (pay-per-use, auto-scaling)

Conclusion

- **Cloud Architecture**

- Actor/role based: provider, consumer, auditor, broker & carrier

- **Cloud Deployment Models**

- Public/Private/Community/Hybrid

- Cloud Service Models

- IaaS/PaaS/SaaS/FaaS

- **Next Week**

- Cloud Enabling Technologies