

Lecture 3 - Cloud Delivery Models

Cloud Delivery Models

Pizza as a service

The variety of cloud services can obfuscate the level of an organization's ownership of the stack.
Albert Barron, executive software client architect at IBM, uses this analogy to provide clarity:



ON-PREMISES MANAGEMENT

Making a pizza
at home



IAAS

Take-and-bake
service



PAAS

Delivery



SAAS

Eat-in
restaurant



Cloud Delivery Models....

Cloud Service Models

**Software as a Service
(SaaS)**

A software distribution model in which applications are hosted by a service provider and made available to customers over Internet

**Platform as a Service
(PaaS)**

Web servers, Application Server, operating system (OS), middleware (e.g. Java runtime, .NET runtime, integration, etc.), and other services to host the consumer's application

**Infrastructure as a Service
(IaaS)**

A provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components.



Google™ Apps



G APP



GCE



Cloud Delivery Models...

On-site

IaaS

PaaS

SaaS

Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking



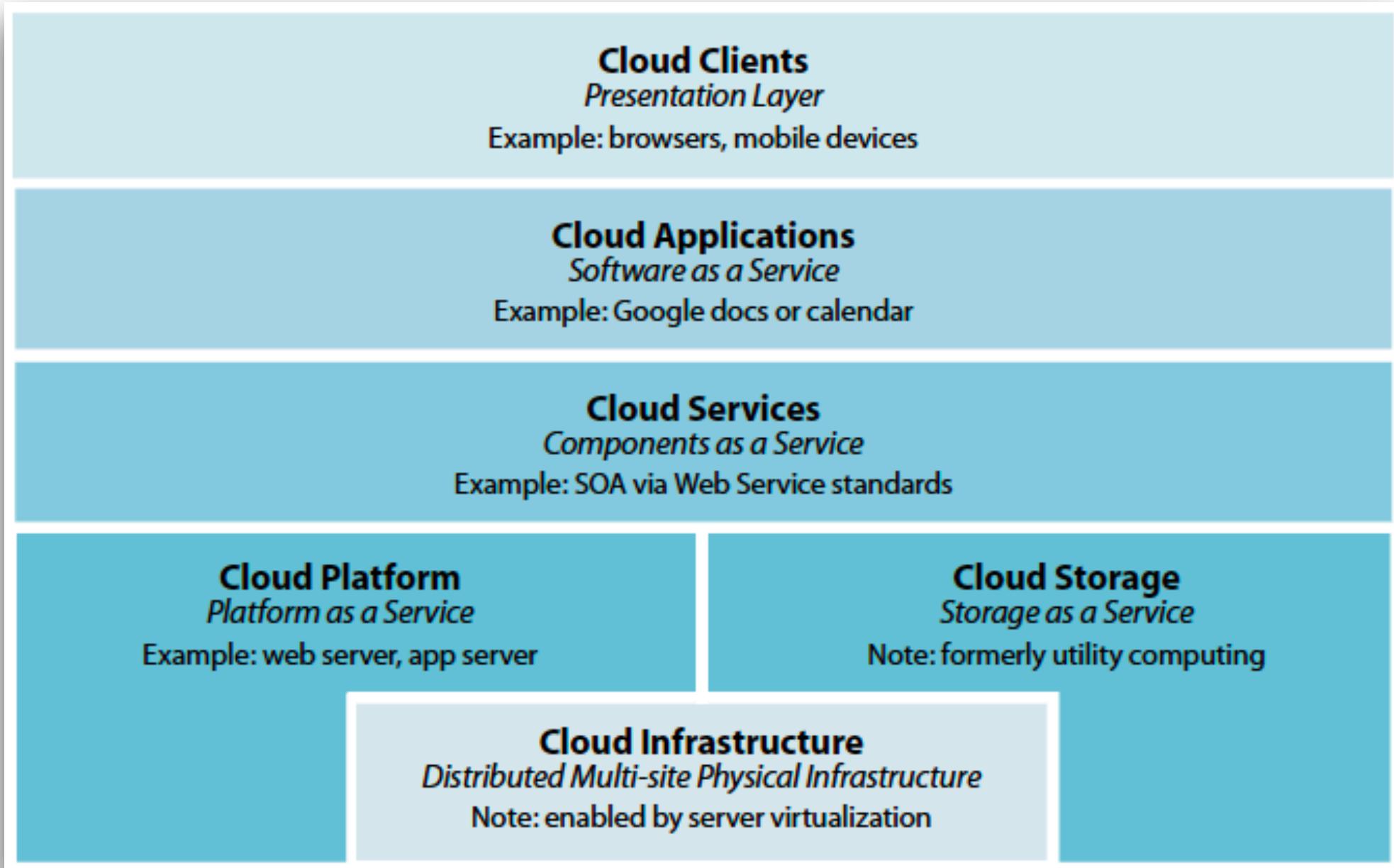
You manage



Service provider manages



Mitre's Cloud Stack



Cloud Delivery Models Examples



SaaS Software

VS



Traditional Software

Subscription

Users subscribe to the software without paying any money up front



One-time fee

Users purchase the software up front and install it on their own computers

Multiple devices

Applications can be used across multiple devices with a single login, the application can be updated online instantaneously



Single device

Licensed individually and usually limited to a single device and when updates come out, they must be downloaded or purchased and installed



Cloud Delivery Models Examples

IaaS	PaaS	BaaS	SaaS
 Amazon EC2	 HEROKU	 Firebase	 zendesk
 DigitalOcean	 salesforce	 Skygear	 PayPal
 rackspace <small>the open cloud company</small>		 Parse Server	
⋮	⋮	⋮	⋮



XaaS

- “Anything as a service”
- Improving the expense model
- Speeding new apps and business processes
- Shifting IT resources to higher-value projects

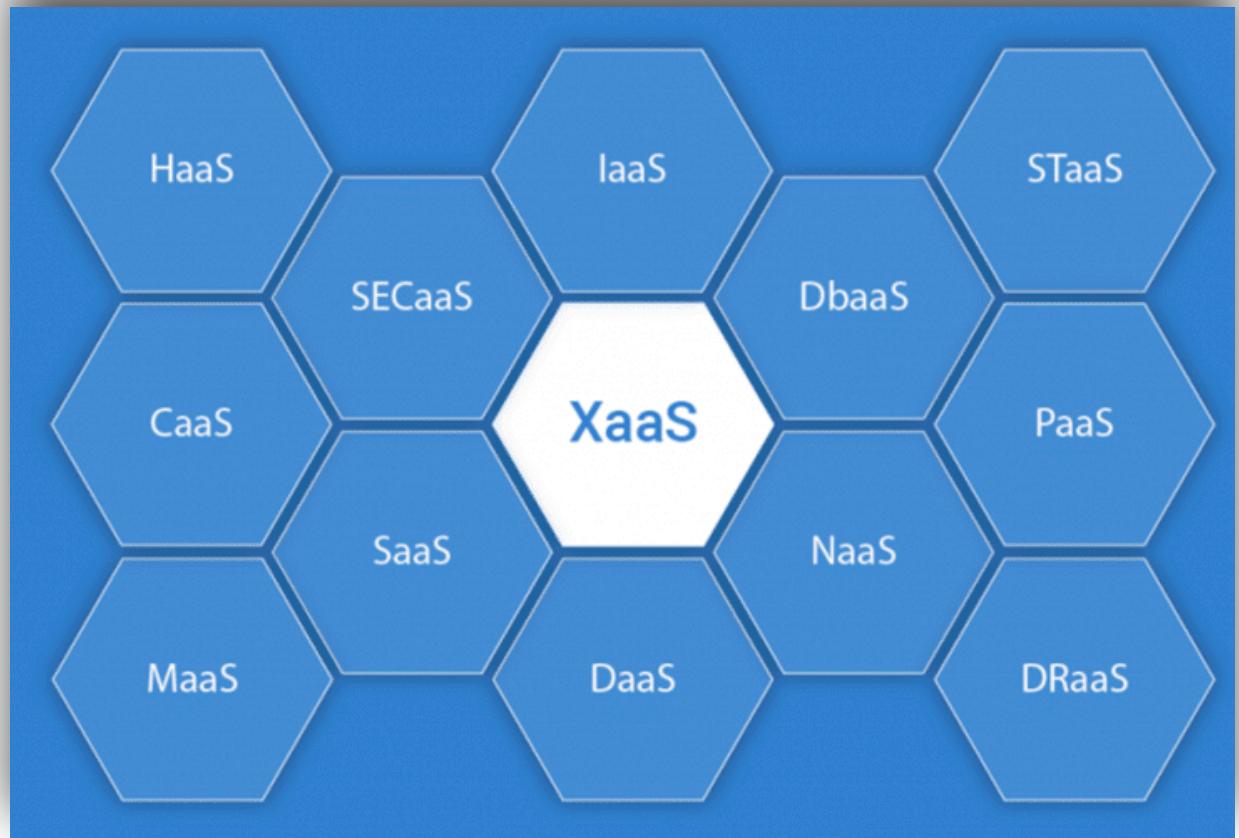


XaaS - Examples



XaaS - Examples

- Disaster recovery as a service (DRaaS)
- Communications as a service (CaaS)
- Network as a Service (NaaS)
- Database as a Service (DBaaS)
- Storage as a service (STaaS)
- Desktop as a service (DaaS)
- Monitoring as a service (MaaS)
- IT as a service (ITaaS)



XaaS - Disadvantages

- Possible downtime
- Performance issues
- Complexity impacts

Key Characteristics of Cloud Computing

NIST Definition

- On-demand self-service: consumers can unilaterally provision computing capabilities as needed automatically without requiring human interaction with each service provider.
- Broad network access: capabilities are available over the network and accessed through standard mechanism that promote use by heterogeneous thin or thick client platforms.
- Resources pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand
- Rapid elasticity: capabilities can be elastically provisioned and released to scale rapidly outward and inward commensurate with demand.
- Measured service: cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

NIST Definition

- Service Models
 - Software as a Service (SaaS)
 - Platform as a Service (PaaS)
 - Infrastructure as a Service (IaaS)
- Deployment Models
 - Private cloud
 - Community cloud
 - Public cloud
 - Hybrid cloud

Terms in Computing – Scalability and Elasticity

- Scalability
 - A desirable property of a system, a network, or a process, which indicates its ability to either handle growing amount of work in a graceful manner or to be readily enlarged.
- Elasticity
 - The ability to apply a quantifiable methodology that allows for the basis of an adaptive introspection within a real time infrastructure.
- But how to achieve these properties ?
 - ✓ Dynamic provisioning



Terms in Computing – Availability and Reliability

- Availability
 - Degree to which a system, subsystem, or equipment is in a specified operable and committable state at the start of the mission, when a mission is called for at an unknown time.
- Reliability
 - The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- Availability and Reliability can be achieved by ,
 - ✓ Fault-tolerant systems
 - ✓ System Resilience
 - ✓ Reliable system security



Terms in Computing – Manageability and Interoperability

- Manageability
 - Systems manageability is strongly affected by network management initiatives in telecommunications.
- Interoperability
 - A property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementations
- But how to achieve above two ?
 - System control automation
 - System state monitoring



Terms in Computing – Performance & Optimization

Performance

- Application performance should be guaranteed.

Optimization

- Use of powerful infrastructure or other underlining resources to build up a highly performed and highly optimized environment, and then deliver the complete services to end users.

- ❖ Parallel computing
- ❖ Load balancing
- ❖ Job scheduling



Terms in Computing – Accessibility & Portability

- Accessibility
 - A general term used to describe the degree to which a product, service, or environment is accessible by as many people as possible.
- Portability
 - Service portability is the ability to access services using any device, anywhere, continuously with mobility support and dynamic adaptation to resource variations

How can we achieve this??

- ✓ Uniform access
- ✓ Thin client



Thanks