CSCI 5409 Adv Topics in Cloud Computing – Fall, 2023
Week 4 – Lecture 2 (Sep 29, 2023)

# Cloud Delivery Models

Dr. Lu Yang
Faculty of Computer Science
Dalhousie University
luyang@dal.ca

# Housekeeping and Feedback

- Final: tentatively 9:30-11:30am, Dec 8.
- First assignment due today.
- GCP credits and K8s assignment will be released this weekend. No double claim of the GCP credits.
- No class on Monday, Oct 2.
- Tech Career Fair
  - AWS vs. Azure 50:50
  - Certifications: Developer vs. Architect CRA
  - Kubernetes ←
  - CRA to cloud ←
  - Career page of companies
  - Resume and first impression
  - High GPA vs. hands-on

# Objectives

- Learning Outcomes
    - Understand and differentiate between cloud delivery models

# Contents

**Section 1.** Cloud Delivery Models

## Cloud Delivery Models

1. Overview
2. Cloud Delivery Models
3. Model Comparisons
4. Model Combination

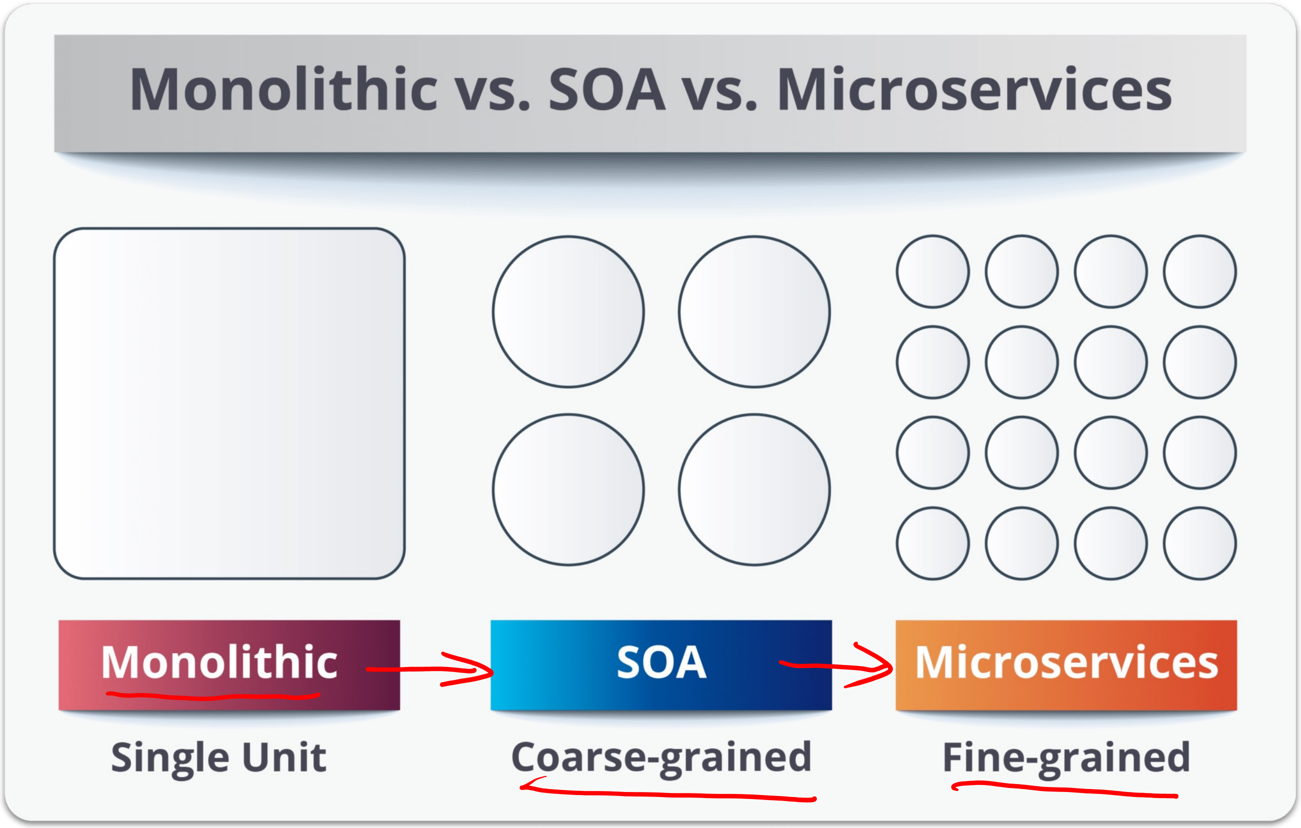
# Overview (1/4)

- "A **cloud delivery model** represents a specific, pre-packaged combination of IT resources offered by a cloud provider."[1]
  - Infrastructure-as-a-Service (IaaS)
  - Platform-as-a-Service (PaaS)
  - Software-as-a-Service (SaaS)
  - Function-as-a-Service (FaaS)
  - Storage-as-a-Service
  - Database-as-a-Service
  - Security-as-a-Service
  - Communication-as-a-Service
  - Integration-as-a-Service
  - Testing-as-a-Service
  - Process-as-a-Service

XaaS

[1]: https://patterns.arcitura.com/cloud-computing-patterns/basics/cloud-delivery-models/overview

# Overview (2/4)



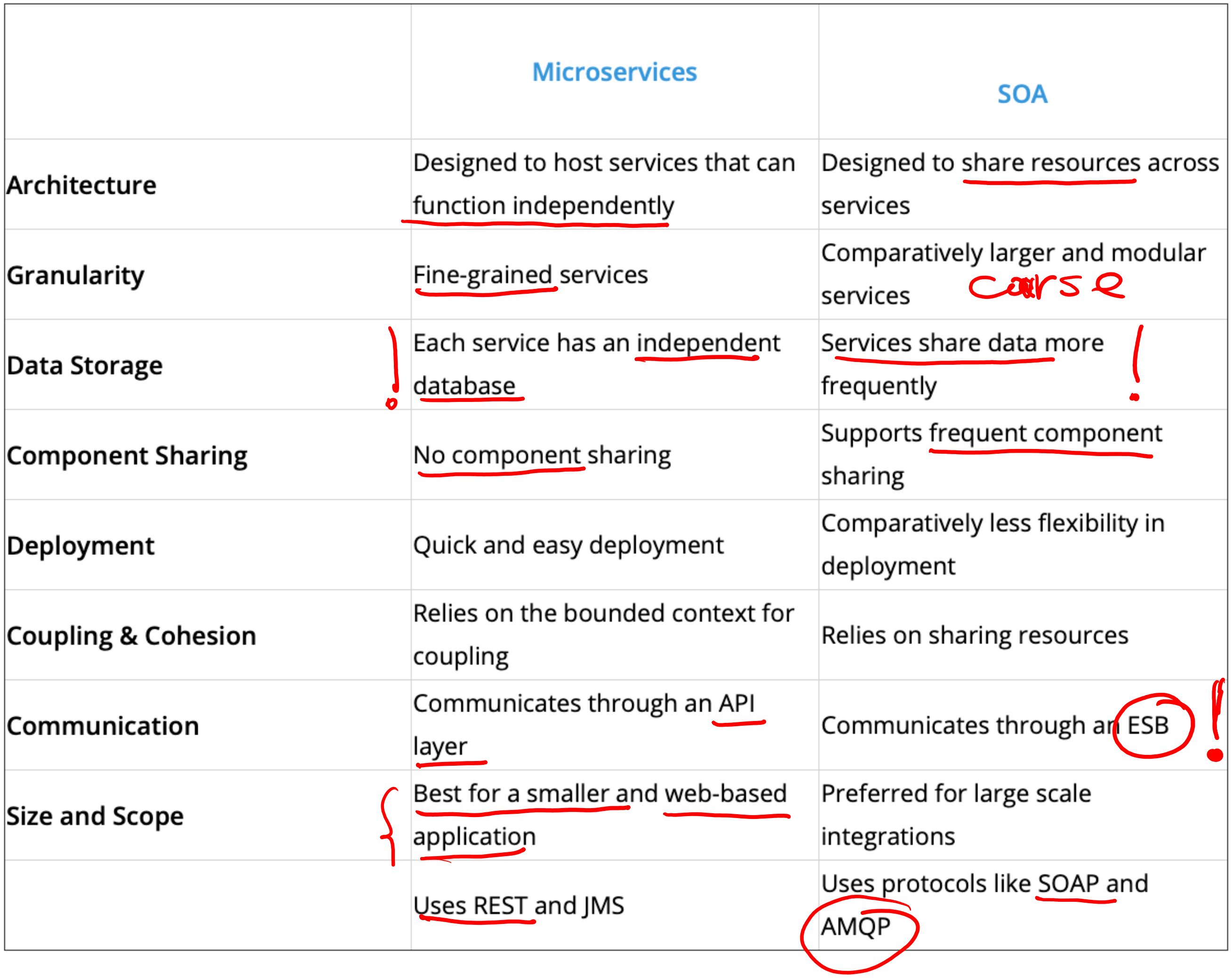
Picture from: https://dzone.com/articles/microservices-vs-soa-whats-the-difference

# Overview (3/4)

- Service-Oriented Architecture (**SOA**) [1]
  - An important stage in the evolution of application development and integration
  - Defines a way to make software components reusable and interoperable via service interfaces
  - SOA breaks up the whole application into separate service modules that interact with one another to carry out the specific business objectives. In SOA, every module is considerably smaller in size when compared with a monolithic application.
  - Each service in an SOA embodies the code and *data* required to execute a complete, discrete business function (e.g., checking a customer's credit, calculating a monthly loan payment, or processing a mortgage application)
  - The service interface is a service contract between the service provider and service consumer. Applications behind the service interface can be written in Java, Microsoft .Net, or any other programming language, supplied as packaged software applications by a vendor (e.g., SAP), SaaS applications (e.g., Microsoft), or obtained as open-source applications.
  - The services are exposed using standard network protocols — such as SOAP (simple object access protocol)/HTTP — to send requests to read or change data

[1]: Cloud Computing (T. Erl, Z. Mahmoud, R. Puttini), pg. 63
[2]:https://www.ibm.com/cloud/learn/soa
[3]: https://readitquik.com/articles/cloud-3/top-10-saas-vendors/

# Overview – Microservices vs. SOP (4/4)



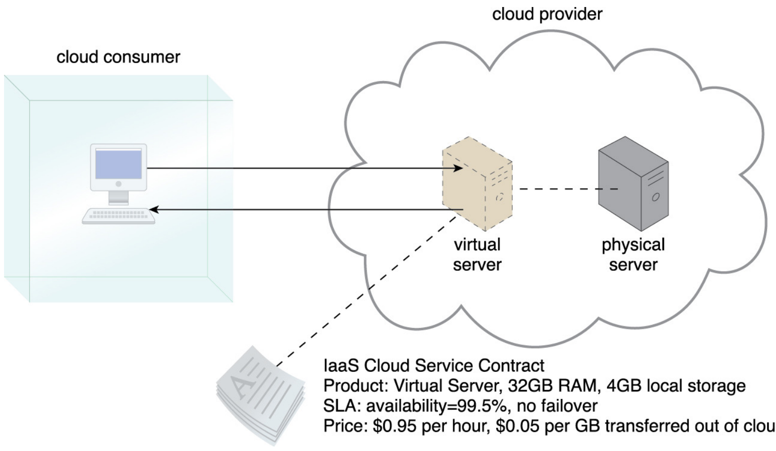
| | Microservices | SOA |
|---|---|---|
| **Architecture** | Designed to host services that can function independently | Designed to share resources across services |
| **Granularity** | Fine-grained services | Comparatively larger and modular services coarse |
| **Data Storage** | ! Each service has an independent database | Services share data more frequently ! |
| **Component Sharing** | No component sharing | Supports frequent component sharing |
| **Deployment** | Quick and easy deployment | Comparatively less flexibility in deployment |
| **Coupling & Cohesion** | Relies on the bounded context for coupling | Relies on sharing resources |
| **Communication** | Communicates through an API layer | Communicates through an ESB ! |
| **Size and Scope** | Best for a smaller and web-based application | Preferred for large scale integrations |
| | Uses REST and JMS | Uses protocols like SOAP and AMQP |

[1]: https://mindmajix.com/soa-vs-microservices

# Cloud Delivery Models — IaaS (1/2)

Infrastructure-as-a-Service (IaaS)

- "The IaaS delivery model represents a self-contained IT environment comprised of infrastructure-centric IT resources that can be accessed and managed via cloud service-based interfaces and tools."[1]



IaaS Cloud Service Contract
Product: Virtual Server, 32GB RAM, 4GB local storage
SLA: availability=99.5%, no failover
Price: $0.95 per hour, $0.05 per GB transferred out of cloud

- **Translation:** Rather than purchase, maintain and upgrade expensive on-premise hardware to provide computing power for the organization's needs, an organization contracts with a cloud provider to deliver these resources virtually.
  rent
- Cloud consumers choose this model when **they want a high level of control and responsibility over configuration and utilization**[1]

[1]: Cloud Computing (T. Erl, Z. Mahmoud, R. Puttini), pg. 64

# Cloud Delivery Models — IaaS (2/2)

- **IaaS Advantages**
  - The most flexible cloud computing model
  - Easy to automate deployment of storage, networking, servers, and processing power
  - Hardware purchases can be based on consumption
  - Clients retain complete control of their infrastructure
  - Resources can be purchased as-needed
  - Highly scalable

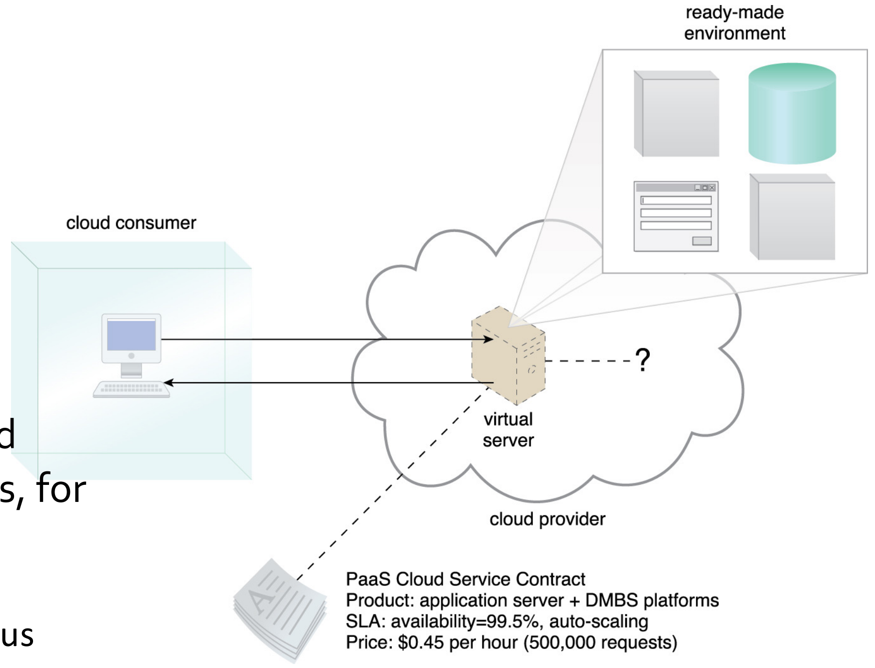
- **IaaS Characteristics**
  - Resources are available as a service
  - Cost varies depending on consumption
  - Services are highly scalable
  - Multiple users on a single piece of hardware
  - Organization retain complete control of the infrastructure
  - Dynamic and flexible

- **When to Use IaaS**
  - **Startups and small companies** may prefer IaaS to avoid spending time and money on purchasing and creating hardware and software.
  - **Larger companies** may prefer to retain complete control over their applications and infrastructure, but they want to purchase only what they actually consume or need.
  - **Companies experiencing rapid growth** like the scalability of IaaS, and they can change out specific hardware and software easily as their needs evolve.

Some texts from: https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/#ref3

# Cloud Delivery Models — PaaS (1/2)

- Platform-as-a-Service
- "The PaaS delivery model represents a pre-defined "ready-to-use" environment typically comprised of already deployed and configured IT resources."[1]
- Why would a cloud consumer choose this model?
  - To extend on-premise environments into the cloud for scalability and economic purposes
  - To entirely substitute an on-premise environment
  - To become a cloud provider and deploy their own cloud services to be made available to other cloud consumers, for example:
    - Twilio (provides a PaaS that enables customers (other businesses) to engage their customers through various communications channel (SMS, voice, WhatsApp, Email, etc.)
    - AWS Elastic Beanstalk, Sagemaker



PaaS Cloud Service Contract
Product: application server + DMBS platforms
SLA: availability=99.5%, auto-scaling
Price: $0.45 per hour (500,000 requests)

[1]: Cloud Computing (T. Erl, Z. Mahmoud, R. Puttini), pg. 65

# Cloud Delivery Models — PaaS (2/2)

- **PaaS Advantages**
  - Simple, cost-effective development and deployment of apps
  - Scalable
  - Highly available
  - Developers can customize apps without the headache of maintaining the software
  - Automation of business policy
  - Easy migration to the hybrid model

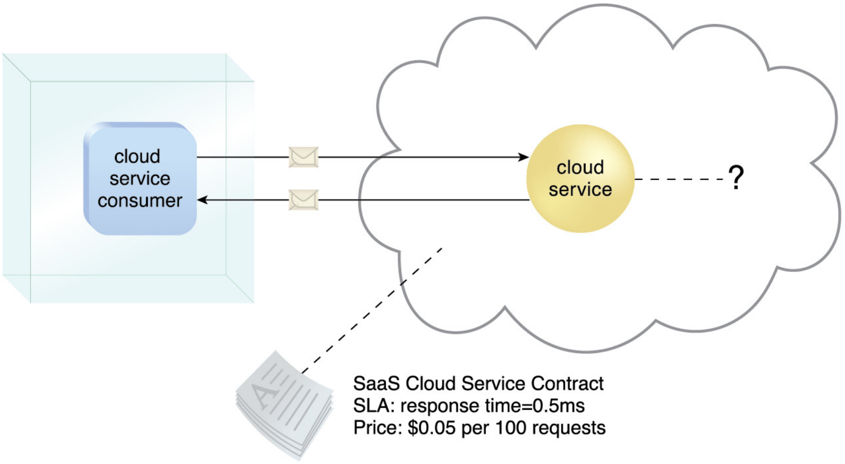
- **PaaS Characteristics**
  - Builds on virtualization technology, so resources can easily be scaled up or down as your business changes
  - Provides a variety of services to assist with the development, testing, and deployment of apps
  - Accessible to numerous users via the same development application
  - Integrates web services and databases

- **When to Use PaaS**
  - Multiple developers working on the same development project.
  - Quickly and flexibly get required other vendors included into the entire process.
  - Create customized applications.
  - Rapidly develop and deploy an application.

Some texts from: https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/#ref3

# Cloud Delivery Models — SaaS (1/2)

- Software-as-a-Service (SaaS)
- SaaS represents a "software program positioned as a shared cloud service and made available as a "product" or generic utility."[1]
- SaaS is typically used to make a reusable cloud service widely available (even commercially) to a range of cloud consumers
  - An entire industry shift occurred to leasing software, paying only for the features your users need
- Examples:
  - Gmail and Google docs
  - Office365
  - Dropbox



[1]: Cloud Computing (T. Erl, Z. Mahmoud, R. Puttini), pg. 66

# Cloud Delivery Models — SaaS (2/2)

- **SaaS Advantages**
  - Reduces money and tasks of installing, managing, and upgrading software
  - Most can be run directly on web browser

- **SaaS Characteristics**
  - Managed from a central location
  - Hosted on a remote server
  - Accessible over the internet
  - Users not responsible for hardware or software updates

- **When to Use SaaS**
  - Startups or small companies that need to launch ecommerce quickly and don't have time for server issues or software
  - Short-term projects that require quick, easy, and affordable collaboration
  - Applications that aren't needed too often, such as tax software
  - Applications that need both web and mobile access

Some texts from: https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/#ref3

# Cloud Delivery Models — FaaS (1/2)

- Function-as-a-Service (FaaS)
- "FaaS provides a platform allowing cloud consumers to develop, run and manage application functionalities without the complexity of building and maintaining the infrastructure to typically associated with developing and launching an app."[1]
- Allows cloud consumers to:
  - Run code only when needed, nothing is spinning (sitting idle waiting)
  - Build entirely serverless applications
- FaaS is an event-driven execution model that runs in stateless containers and those functions manage server-side logic and state through the use of services from a FaaS provider.[2]
  - For example, uploading a file could trigger custom code that transcodes the file into a variety of formats.
- A core component of serverless computing
  - FaaS and serverless meant more or less the same thing, but serverless has expanded to mean a larger set of architectural patterns and practices that make extensive use of common services in addition to custom business logic encoded in FaaS.
  - The term "serverless" is also being used to represent managed services, like databases and messaging systems, where a developer or administrator is not required to operate the system since it's being managed by a cloud provider or third-party company.

[1]: https://en.wikipedia.org/wiki/Function_as_a_service
[2]: https://www.redhat.com/en/topics/cloud-native-apps/what-is-faas

# Cloud Delivery Models — FaaS (2/2)

- **FaaS Advantages[1]**
  - Write code, upload, deploy
  - Extend infinitely, anytime
  - Save money, save the planet
  - Scale your app to the moon and back
  - High performance, low latency

- **FaaS Characteristics[1]**
  - Short-lived
  - Event-driven
  - Stateless
  - Scalable
  - Easily-integrated

- **When to Use FaaS [2]**
  - High volumn parallel workloads
  - Creation of back-end systems and for data processing, encoding, format conversion, and data aggregation
  - Tool for back-ends, data-processing, web apps, or to create chatbots for IoT devices
  - Help to manage and oversee third-party services

[1] https://www.pubnub.com/blog/what-is-functions-as-a-service-faas/

[2] https://www.ringcentral.co.uk/gb/en/blog/definitions/function-as-a-service-faas/#FaaS_use_cases

# Cloud Delivery Models — Other Models

- **Database-as-a-Service**: This is just PaaS but for a database.
- **Security-as-a-Service**: A form of SaaS, a cloud consumer moves all of their authentication, anti-virus, anti-malware, intrusion detection software into the cloud (e.g. single sign-on with your NetID through my.dal.ca)
- **Communication-as-a-Service**: Business to business solutions like VOIP
- ***X-as-a-Service*** has become a popular way to sell your app concept, business deciders have picked up on the cost savings of cloud computing and are desperate to get out of their premises, they're looking for existing solutions that save them time and money

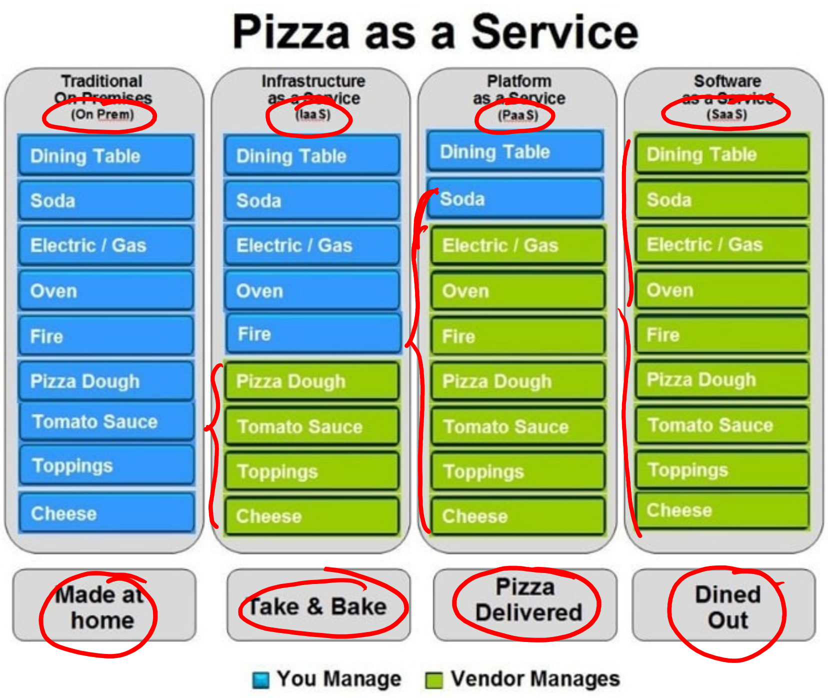
# Cloud Delivery Models — Comparison (1/2)

| Cloud Delivery Model | Typical Level of Control Granted to Cloud Consumer | Typical Functionality Made Available to Cloud Consumer | Common Cloud Consumer Activities | Common Cloud Provider Activities |
|---|---|---|---|---|
| **SaaS** | Usage and usage-related configuration | Access to front-end user-interface | Using and configuring the cloud service | Implementing, managing and maintaining the cloud service, monitoring usage by cloud consumers |
| **PaaS** | Limited administrative | Moderate level of administrative control over IT resources relevant to usage of the platform | Developing, testing, deploying, managing cloud services and cloud-based solutions | Pre-configuring platforms and provisioning underlying infrastructure, middleware and other resources as necessary, monitoring usage by cloud consumers |
| **IaaS** | Full administrative | Full access to virtualized infrastructure-related IT resources | Setting up and configuring virtual infrastructure, managing and monitoring any needed software | Provisioning and managing the physical IT infrastructure, monitoring usage by cloud consumers |
| **FaaS** | Limited administrative (create, update, execute) | Control over the function code, some providers allow control over memory limits | Developing, testing, deploying, connecting functions to events/triggers | Provisioning and managing the physical IT infrastructure, monitoring usage by cloud consumers, building / supporting the software infrastructure that executes functions |

Table from: Cloud Computing (T. Erl, Z. Mahmoud, R. Puttini). Pg. 68, with additions of FaaS by Rob

# Cloud Delivery Models — Comparison (2/2)

## Better examples?

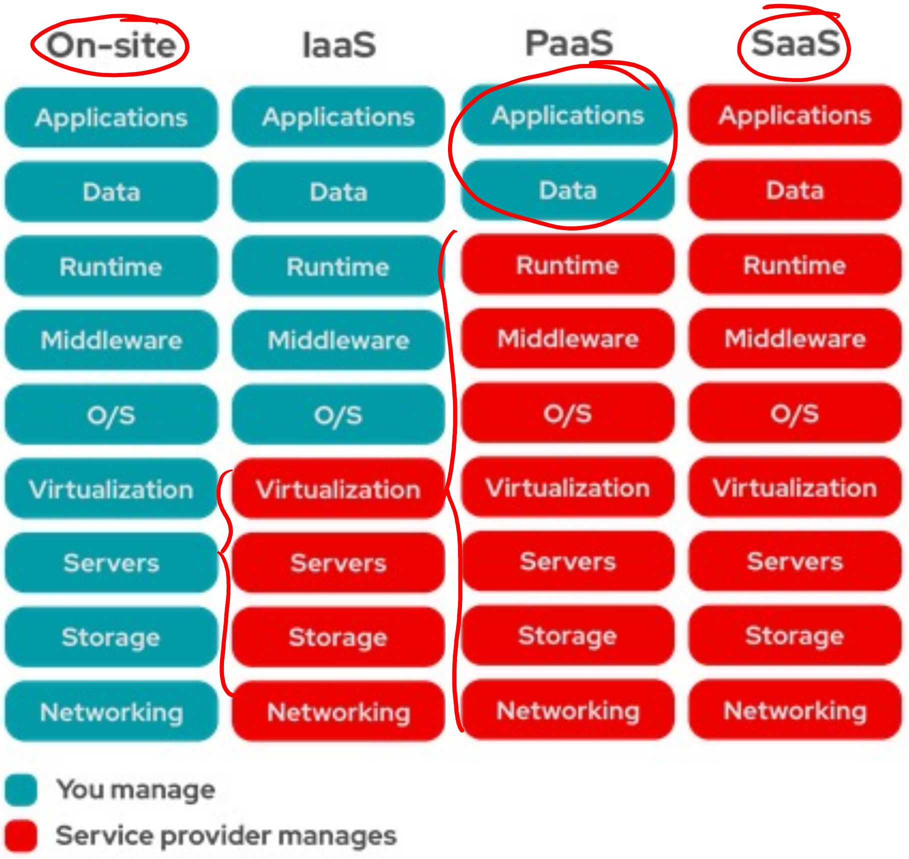


### Pizza as a Service

| Traditional On-Premises (On Prem) | Infrastructure as a Service (IaaS) | Platform as a Service (PaaS) | Software as a Service (SaaS) |
|---|---|---|---|
| 🟦 Dining Table | 🟦 Dining Table | 🟦 Dining Table | 🟩 Dining Table |
| 🟦 Soda | 🟦 Soda | 🟦 Soda | 🟩 Soda |
| 🟦 Electric / Gas | 🟦 Electric / Gas | 🟩 Electric / Gas | 🟩 Electric / Gas |
| 🟦 Oven | 🟦 Oven | 🟩 Oven | 🟩 Oven |
| 🟦 Fire | 🟦 Fire | 🟩 Fire | 🟩 Fire |
| 🟦 Pizza Dough | 🟩 Pizza Dough | 🟩 Pizza Dough | 🟩 Pizza Dough |
| 🟦 Tomato Sauce | 🟩 Tomato Sauce | 🟩 Tomato Sauce | 🟩 Tomato Sauce |
| 🟦 Toppings | 🟩 Toppings | 🟩 Toppings | 🟩 Toppings |
| 🟦 Cheese | 🟩 Cheese | 🟩 Cheese | 🟩 Cheese |
| Made at home | Take & Bake | Pizza Delivered | Dined Out |

🟦 You Manage 🟩 Vendor Manages

From: [https://www.bigcommerce.com/blog/saas-vs-paas-vs-iaas/#executive-summary-summing-up-saas-vs-paas-vs-iaas](https://www.bigcommerce.com/blog/saas-vs-paas-vs-iaas/#executive-summary-summing-up-saas-vs-paas-vs-iaas)



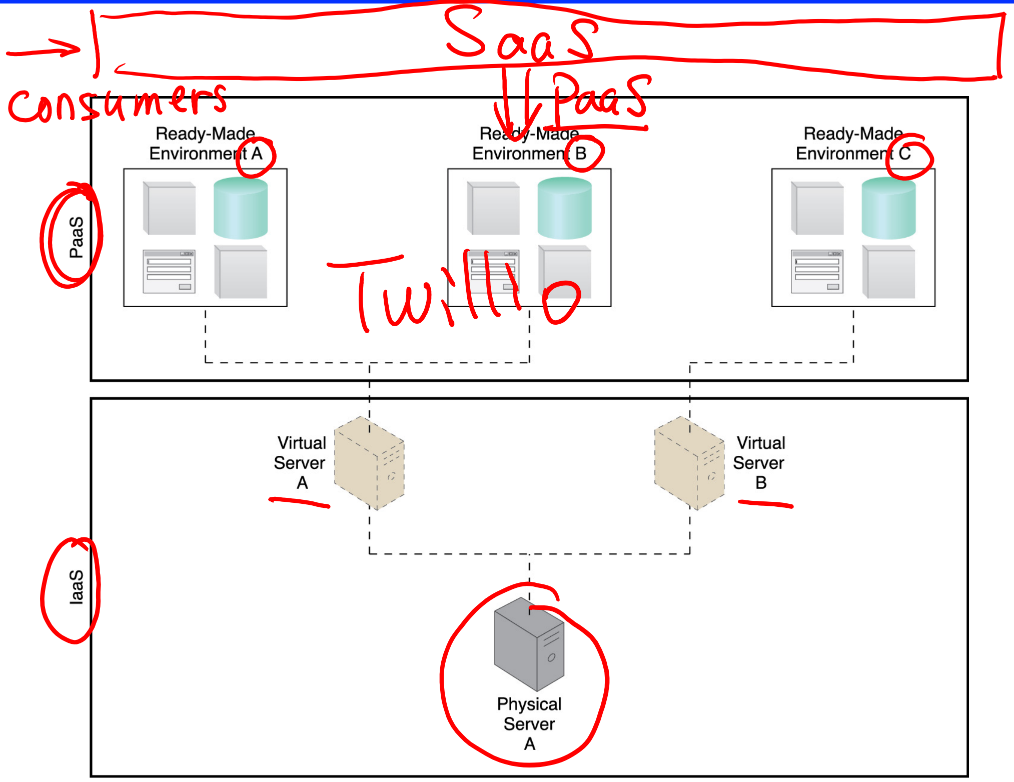
| On-site | IaaS | PaaS | SaaS |
|---|---|---|---|
| 🟦 Applications | 🟦 Applications | 🟦 Applications | 🟥 Applications |
| 🟦 Data | 🟦 Data | 🟦 Data | 🟥 Data |
| 🟦 Runtime | 🟦 Runtime | 🟥 Runtime | 🟥 Runtime |
| 🟦 Middleware | 🟦 Middleware | 🟥 Middleware | 🟥 Middleware |
| 🟦 O/S | 🟦 O/S | 🟥 O/S | 🟥 O/S |
| 🟦 Virtualization | 🟥 Virtualization | 🟥 Virtualization | 🟥 Virtualization |
| 🟦 Servers | 🟥 Servers | 🟥 Servers | 🟥 Servers |
| 🟦 Storage | 🟥 Storage | 🟥 Storage | 🟥 Storage |
| 🟦 Networking | 🟥 Networking | 🟥 Networking | 🟥 Networking |

🟦 You manage

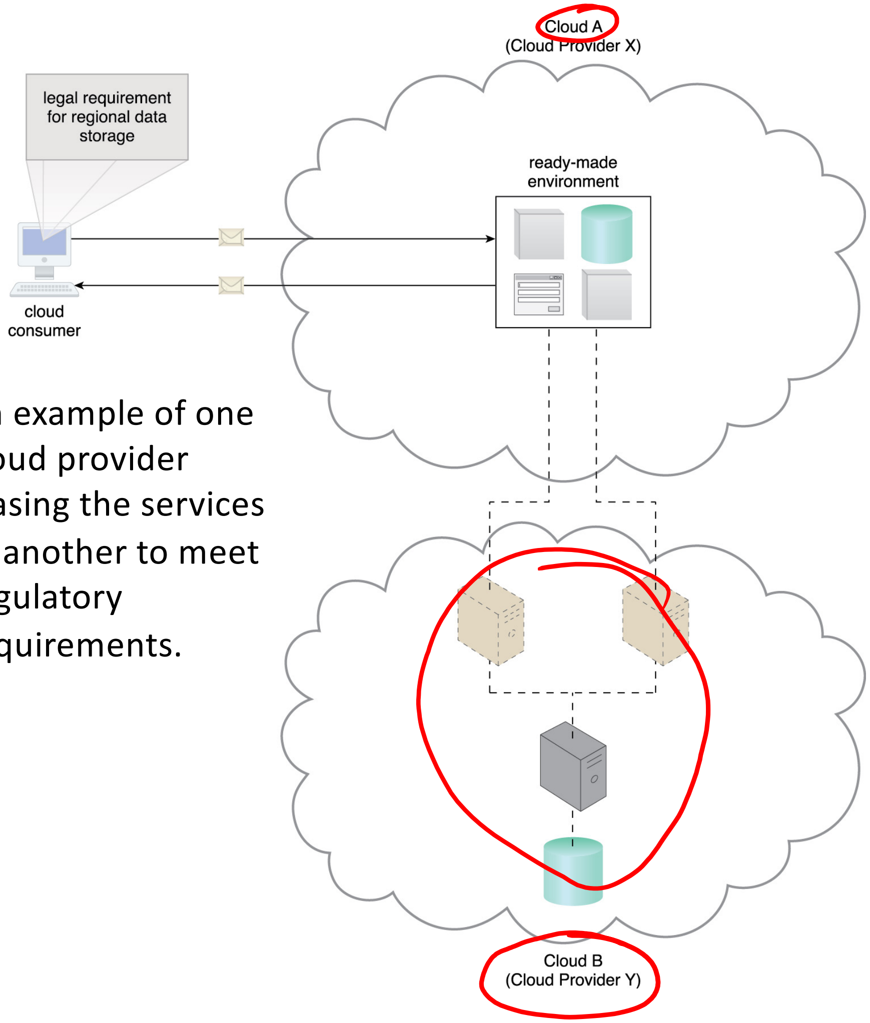
🟥 Service provider manages

From: [https://www.redhat.com/en/topics/cloud-computing/what-is-iaas](https://www.redhat.com/en/topics/cloud-computing/what-is-iaas)

# Cloud Delivery Models — Combination of Models



A PaaS environment based on the IT resources provided by an underlying IaaS environment.



An example of one cloud provider leasing the services of another to meet regulatory requirements.

Figures from: Cloud Computing (T. Erl, Z. Mahmoud, R. Puttini), pg. 70-71

Kahoot!