Introducción a las arquitecturas C/S

Curso 2022/23

Javier Albert Segui

Universidad de Alcalá

Índice

- Definición del Concepto Cliente/Servidor – C/S
- Arquitectura C/S
- Características Sistema C/S
- Beneficios Sistema C/S
- Problemas Sistema C/S
- Generaciones Sistema C/S
- Modos de llamada C/S
- Modalidades C/S
- Computación en la Nube

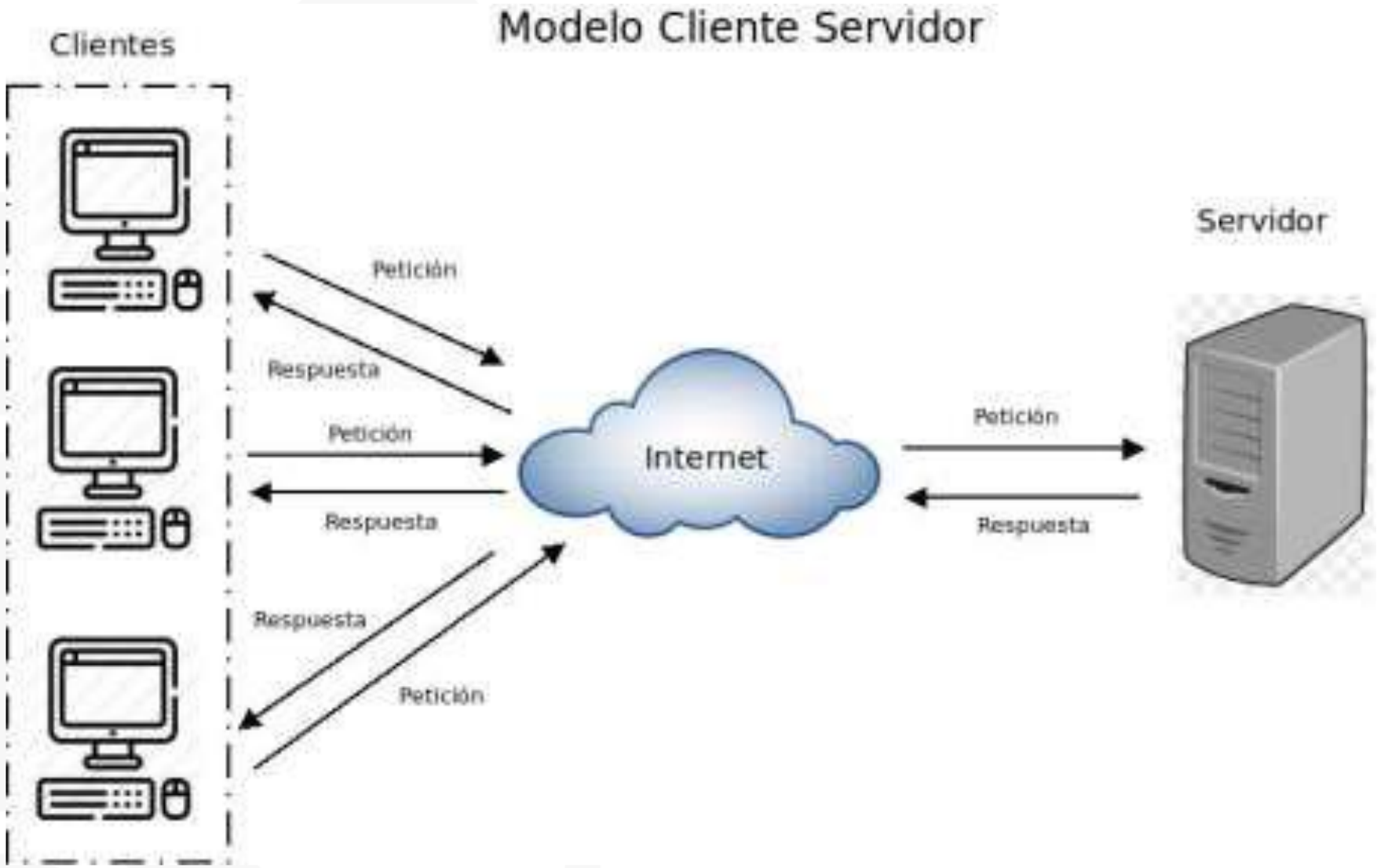
Definición C/S

- Es la integración distribuida de un sistema en red, con los
  - Recursos,
  - Medios
  - Aplicaciones
- Definidos de forma modular para atender las solicitudes de los clientes compartiendo datos, procesos e información de forma transparente

Definiciones C/S

- **Clientes**: son aquellas "entidades" que necesitan Servicios
- **Servidores**: Proporcionan estos Servicios, son objetos separados desde un punto de vista lógico y se comunican a traves de una red de Comunicaciones para realizar una o varias tareas de forma conjunta.
- **Middleware**: es un software distribuido para la interacción entre el cliente y el servidor.

Arquitectura Cliente / Servidor



Características Sistemas C/S

- **Recursos compartidos**: Un único servidor puede interactuar con distintos clientes simultáneamente, por tanto, necesita controlar el acceso a sus recursos para garantizar la seguridad.
- **Transparencia**: Las aplicaciones clientes direccionan el servidor de manera lógica sin importar su localización.
- **Separación de funciones (modularidad)**: La lógica se separa en distintos módulos funcionales para distribuir la carga entre procesadores.
- **Entornos heterogéneos**: Las aplicaciones deberían ser independientes del procesador y el sistema operativo utilizado.

Características Sistemas C/S

- **Encapsulación de servicios**: Las funcionalidades deben ser encapsuladas para que el acceso sea conocido y accesible. Internamente la función puede cambiar sin necesidad de cambiar los clientes.
- **Protocolos asimétricos**: El cliente es quien inicia la comunicación mientras el servidor esta esperando de forma pasiva.
- **Intercambio de mensajes**: La comunicación entre los clientes y servidores se basa en mensajes. El cliente envía **peticiones** y recibe **respuestas** del servidor.
- **Integridad:** El código y los datos se mantienen centralizados por lo tanto es mas sencillo proteger la integridad de ellos.

Beneficios Sistemas C/S

- **Escalabilidad multidimensional**: Los sistemas C/S deben ser escalables tanto horizontal como verticalmente.
  - **Escalabilidad horizontal**: El añadir o eliminar clientes genera impactos de rendimiento
  - **Escalabilidad vertical**: cambio en el servidor.
- Escalabilidad = economía
- HW y SW Heterogéneo = Integración
  - Despliegues independientes del Cliente y Servidor
  - Cliente y Servidor usan el HW y SSOO más adecuados para su función.
- Robustez
- Amigabilidad / Usabilidad

Problemas Sistema C/S

- **Software**: Escasas herramientas para la evaluación de carga
- **Comunicación**: Necesidad de tener mecanismos de comunicación idénticos en distintas plataformas. Sockets, RPC…
- **Seguridad**: Hay que verificarla en ambos extremos.
- **Red**: congestión, pérdida de datos, …..
- **Duplicidad**: Distintas localizaciones, duplicidad de servidores y servicios, actualizaciones…..

Ejemplos C/S

- Fileservers
- Servidores de Bases de datos
- Servidores de transacciones
- Servidores de groupware
- Servidores Web

Generaciones Sistemas C/S

Generaciones Sistemas C/S

- Generación 1- Arquitectura C/S de una capa
  - Basados en Mainframes y miniordenadores a los que se conectaban terminales "tontos"
  - Basados en redes LAN, posteriormente WAN y MAN.
  - Comunicaciones homogéneas
  - Costosos de implementar, desarrollar y mantener.

Generaciones Sistemas C/S

Generaciones Sistemas C/S

- Generación 2 – Arquitectura C/S de 2 capas
  - Funciones distribuidas. Coordinación entre clientes y servidores
  - Algunas funciones se integran en los Gestores de Bases de datos.
  - Arquitecturas complejas, ausencia de estándares, falta de herramientas de gestión.
  - Desarrollo costoso

Generaciones Sistemas C/S

- Generación 3 – Arquitectura C/S de 3 capas
  - Aparece el concepto de *front-end o capa de presentación* que es el responsable de la presentación.
  - El componente *back-end* proporciona acceso a los servicios dedicados
  - Un componente intermedio *middle-tier* que permite compartir y controlar la lógica de negocio.

Funciones del servidor

- Esperar peticiones de los clientes
- Atender solicitudes simultáneas
- Priorizar la atención de las solicitudes
- Capacidad de realizar acciones en segundo plano.
- Robustez
- Escalabilidad y extensibilidad.

Funciones del servidor

- Requisitos del SSOO.
  - Basicos
    - Alto nivel de concurrencia
    - Slots de pequeño tamaño
    - Prioridades
    - Mecanismos de concurrencia
    - Mecanismos de comunicacion entre procesos
    - Threads
    - Sistemas de ficheros de altas prestaciones
    - Sistemas de gestion de memoria eficientes
    - Extensibilidad sin recompilaciones, reinicios....

Funciones del servidor

- Requisitos del SSOO.
  - Extendidos
    - Distintos protocolos de comunicaciones
    - Acceso transparente a recursos compartidos
    - Sistemas de directorio centralizado y global
    - Servicios de autenticación y seguridad
    - Gestión de la configuración, monitorización y alerta

Características del cliente

- Clientes de 3 tipos: sin GUI, con GUI, con OOUI
- Requisitos del SSOO cliente:
  - Implementar mecanismo de envío/recepción de mensajes
  - Implementar mecanismos de transferencia de archivos
  - Multitarea (prioridades, comunicación entre procesos, threads...)
  - Portabilidad
  - Robustez

Middleware

- Su función es que todo funcione de forma TRANSPARENTE
- Este middleware nos da capacidades de:
  - Comunicación a través de la red
  - Envío y recepción de ficheros
  - Servicio de directorios global
  - Mecanismos de seguridad distribuidos

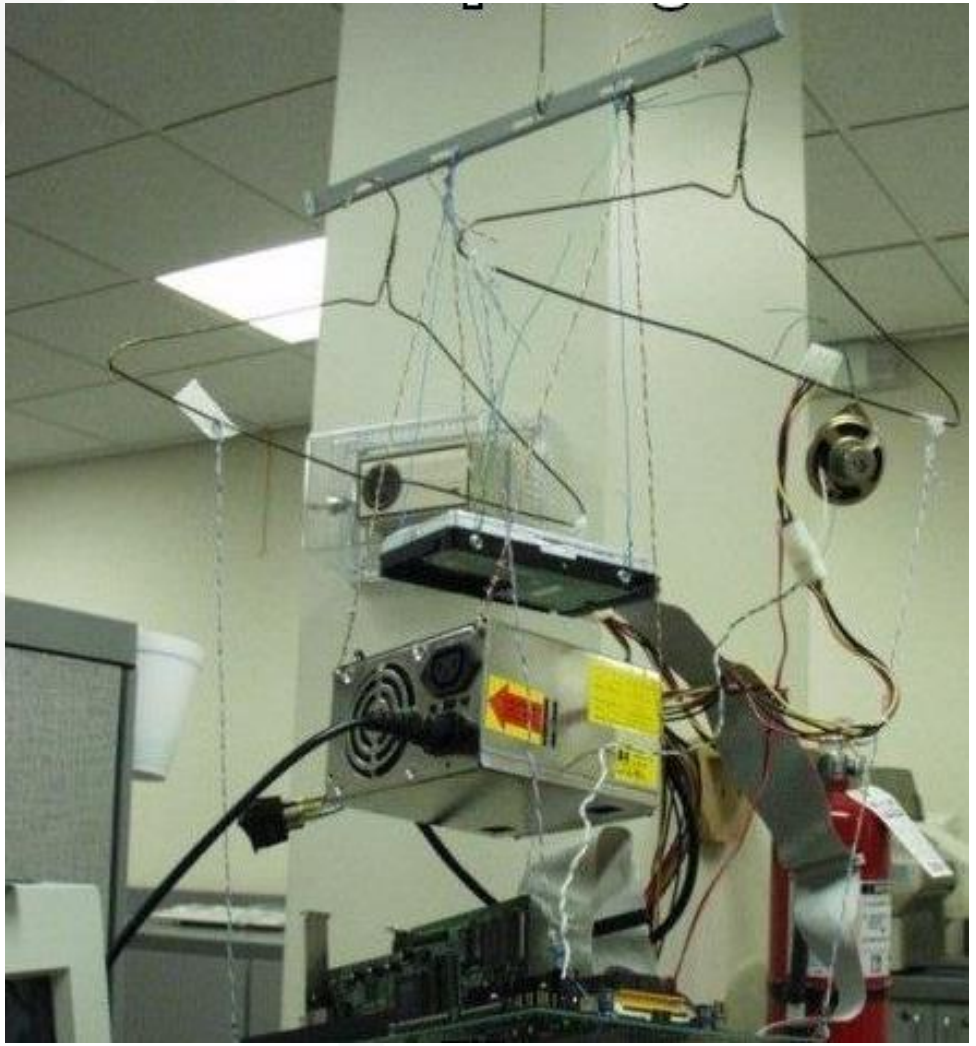
Llamadas en el C/S

- Llamada síncrona: Es el modo mas sencillo de gestionar y usar las llamadas entre Cliente y Servidor.
  - Cuando se realiza una llamada al servidor, el cliente se coloca en modo espera hasta recibir una respuesta.
- Llamada asíncrona: El cliente no entra en modo espera al llamar a un servicio.
  - Cuando el servicio termina, informa a la aplicación de forma automática.
  - La aplicación no tiene que esperar para continuar con otros procesos.

Protocolos C/S

- **Sockets:** Protocolo propio de la comunicación C/S
- **RPC (Remote Procedure Call):** nos permite ejecutar un procedimiento remoto localizado en un servidor.
- **RMI (Remote Method Invocation):** Es un protocolo de invocación de métodos de JAVA
- **CORBA:**
  - Common Object Request Broker Arquitecture.
  - Definicion ESTANDARD
- **Web Services**: Mecanismo para crear Servicios distribuidos basados en el protocolo HTTP mediante SOAP o REST

Cloud computing



Cloud Computing

- Es un paradigma que nos permite ofrecer **servicios** a través de una **red**, generalmente **Internet**

Características

- Agilidad
- Costo
- Escalabilidad
- Independencia
- Rendimiento
- Seguridad
- Mantenimiento

Ventajas

- Integración de servicios
- Rápida implementación
- Servicios ubicuos
- Actualización automática
- Green Computing

Desventajas

- Centralización
- Disponibilidad
- Madurez
- Seguridad
- Escalabilidad

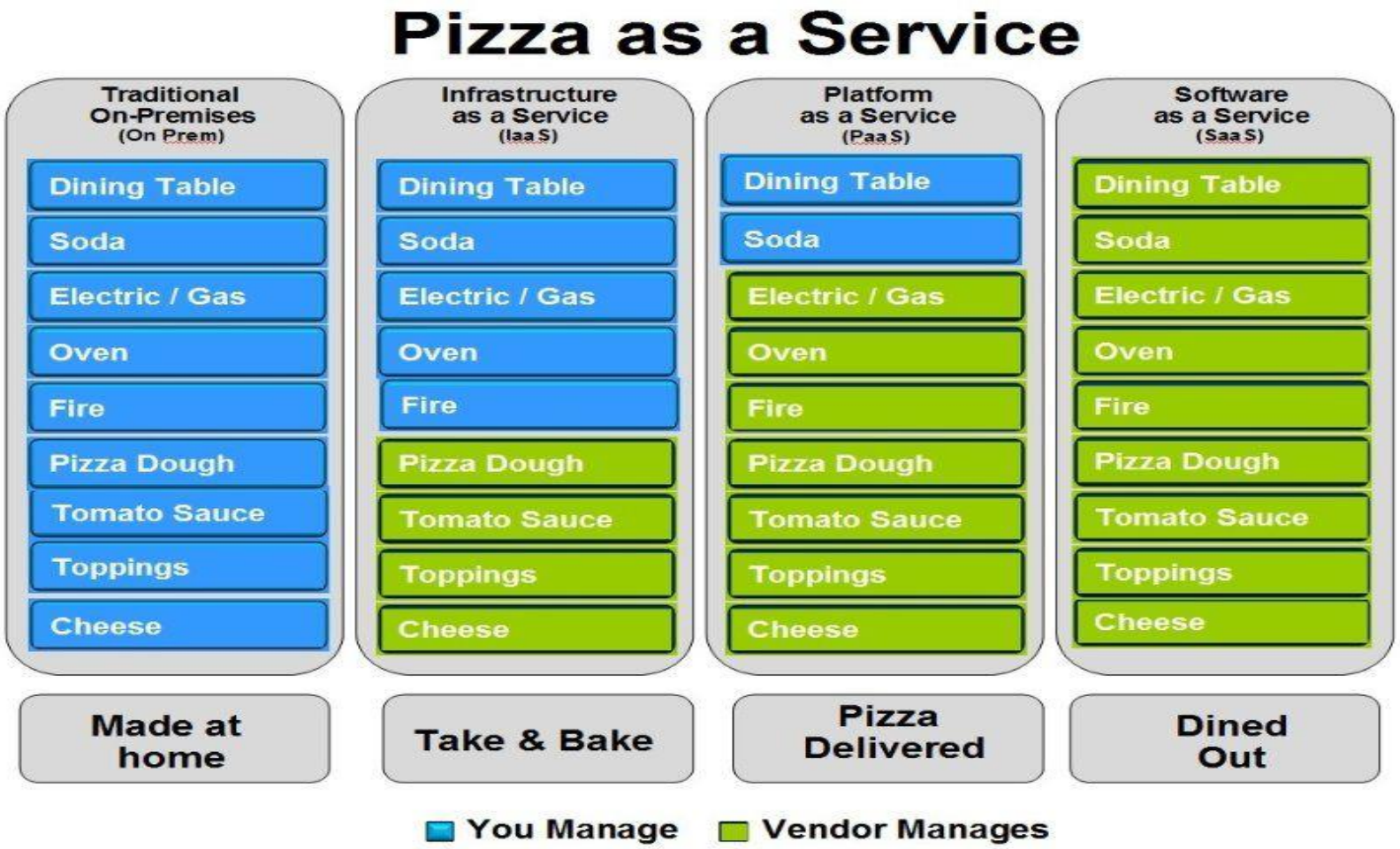
Servicios en la nube

IaaS
Infrastructure as a Service
host

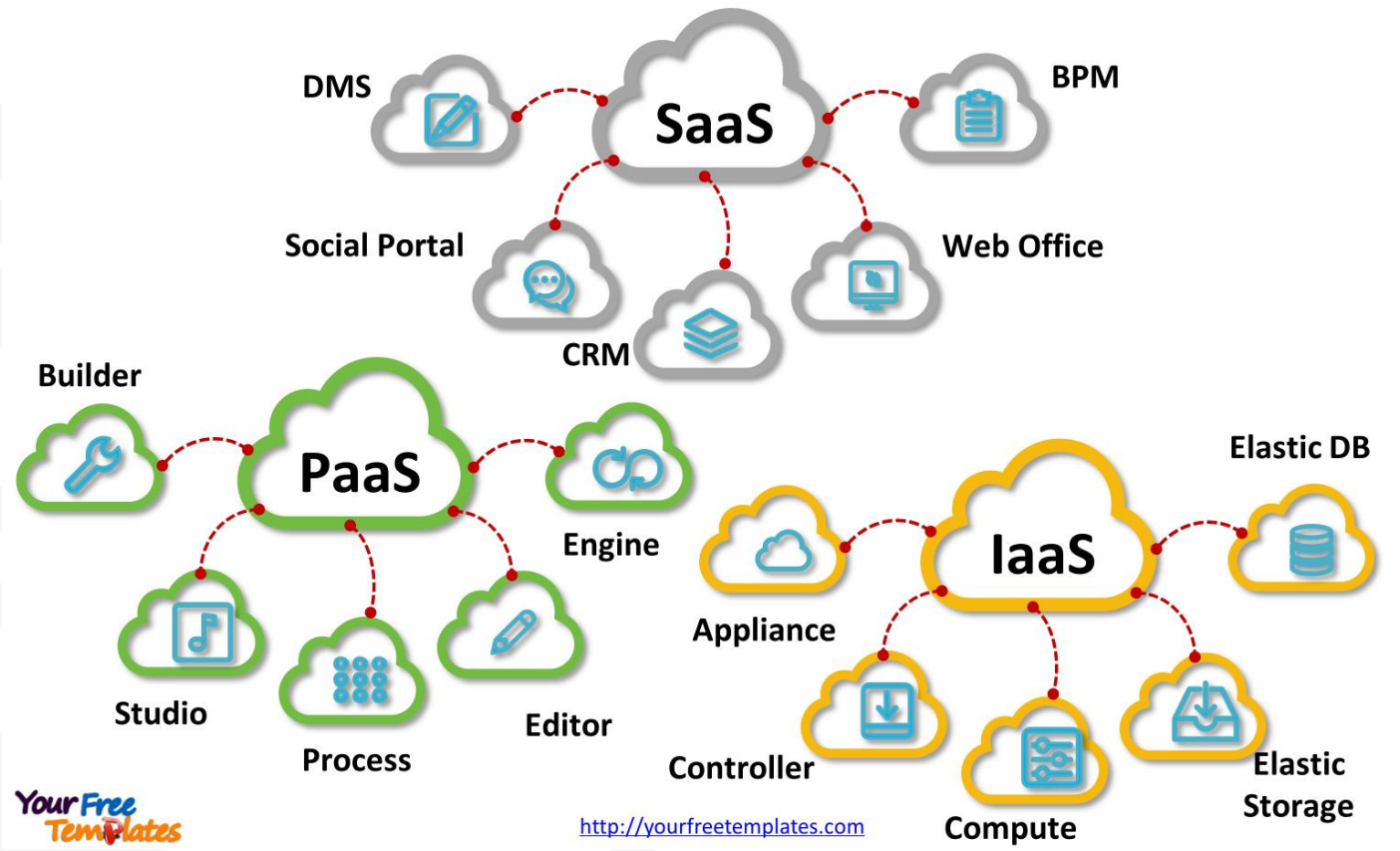
PaaS
Platform as a Service
build

SaaS
Software as a Service
consume

Pizza as a Service

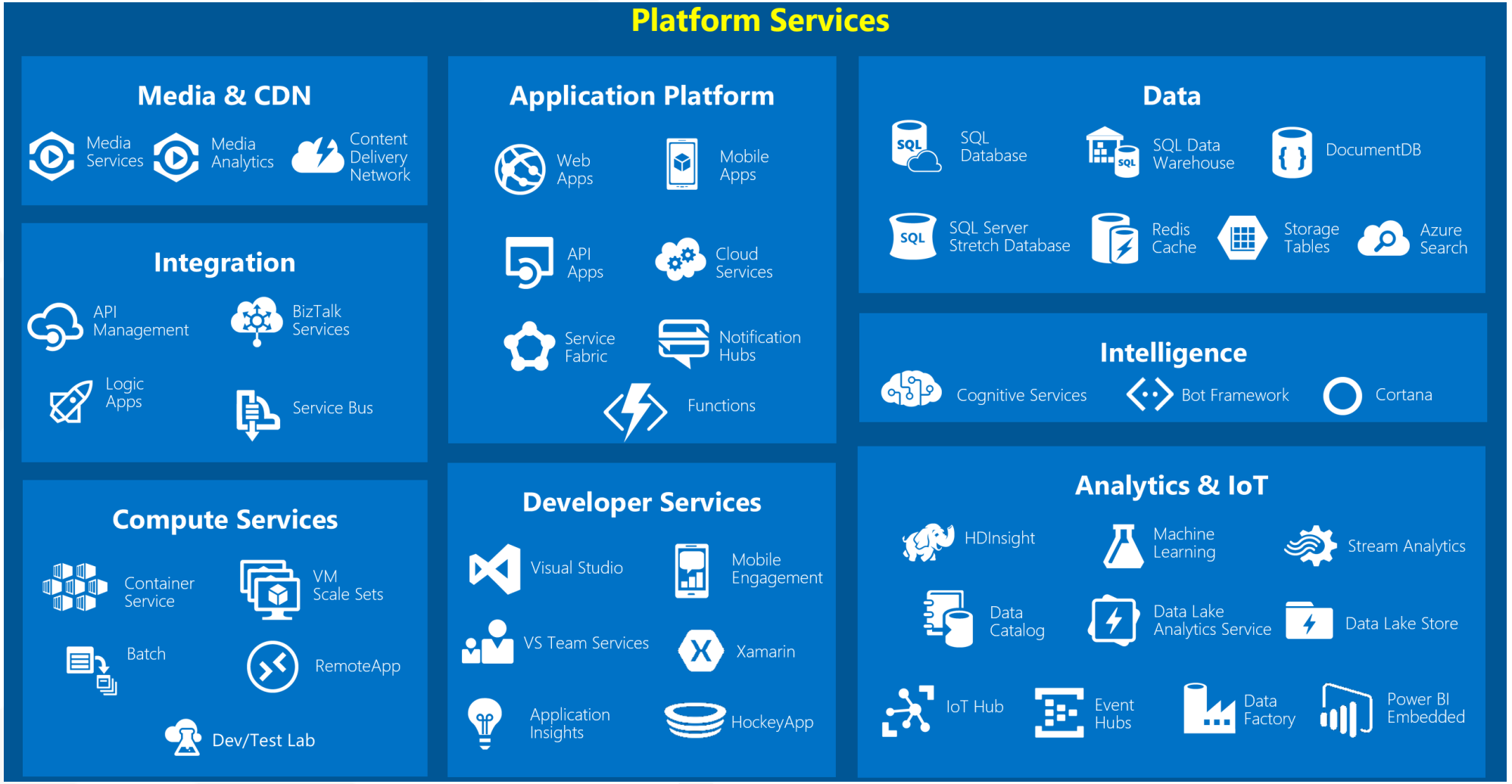


Servicios en la nube



IaaS

PaaS


Platform Services
Media & CDN
Media Services
Media Analytics
Content Delivery Network
Integration
API Management
BizTalk Services
Logic Apps
Service Bus
Compute Services
Container Service
VM Scale Sets
Batch
RemoteApp
Dev/Test Lab
Application Platform
Web Apps
Mobile Apps
API Apps
Cloud Services
Service Fabric
Notification Hubs
Functions
Developer Services
Visual Studio
Mobile Engagement
VS Team Services
Xamarin
Application Insights
HockeyApp
Data
SQL Database
SQL Data Warehouse
DocumentDB
SQL Server Stretch Database
Redis Cache
Storage Tables
Azure Search
Intelligence
Cognitive Services
Bot Framework
Cortana
Analytics & IoT
HDInsight
Machine Learning
Stream Analytics
Data Catalog
Data Lake Analytics Service
Data Lake Store
IoT Hub
Event Hubs
Data Factory
Power BI Embedded

SaaS


Picasa
facebook
msn
Gmail
twitter
Wikipedia
The Free Encyclopedia
flickr
Windows Live
Google buzz
meebo
YouTube
LinkedIn


salesforce.com
Success. Not Software.
skype
angel.com
TypePad
YAHOO
eFax
InterCall
intuit
GoToMeeting
Google
Constant Contact
Connect. Inform. Grow.
facebook
ADP

Responsabilidades