

Tema 8

Redes y Sistemas Distribuidos

Sistemas Distribuidos



*Grado en Ingeniería Informática
Escuela Politécnica Superior*



Universidad
Francisco de Vitoria
UFV Madrid

Contenidos

1. Introducción a las redes de computadores

- Concepto de Red
- Tipos de redes
- Direccionamiento
- Latencia

2. Redes de área local

- Concepto y tipos de redes locales
- Medios de transmisión
- Técnicas de contención

3. Red Ethernet

- Características
- Protocolos
- Estándares
- Direcciones
- Codificación

4. Interconexión de redes

- Modos de interconexión
- Puentes
- *Spanning Tree*
- Switches

5. Red WLAN

- Topologías
- Espectro
- Nivel físico
- Protocolos
- Seguridad

6.1 Nivel del Red Internet

- Encaminamiento
- Fragmentación y reensamblaje

6.2 Direccionamiento IP

7. Arquitectura TCP/IP

- Estructura TCP/IP
- Elementos
- Funcionalidades
- Protocolos
- NAT

8. Sistemas distribuidos

- Concepto
- Tipos
- Arquitecturas



Contenidos

Sistemas distribuídos

- Concepto
- Tipos
- Arquitecturas

Contenidos

Sistemas distribuídos

- Concepto
- Tipos
- Arquitecturas



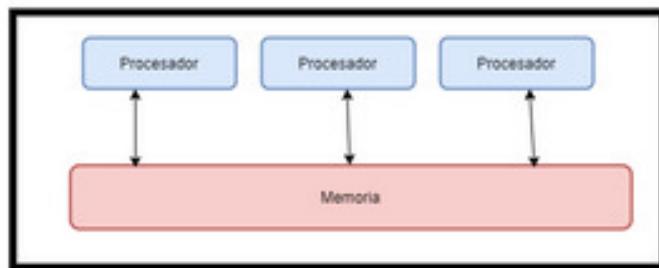
Sistemas distribuidos. Concepto

- Un **sistema distribuido** es un sistema de **software** cuyos componentes están **separados** físicamente y **conectados** entre sí por una **red de computadoras**, se comunican y coordinan entre ellos pasando mensajes
- Dichos componentes **interactúan** entre ellos para lograr una **meta común**.



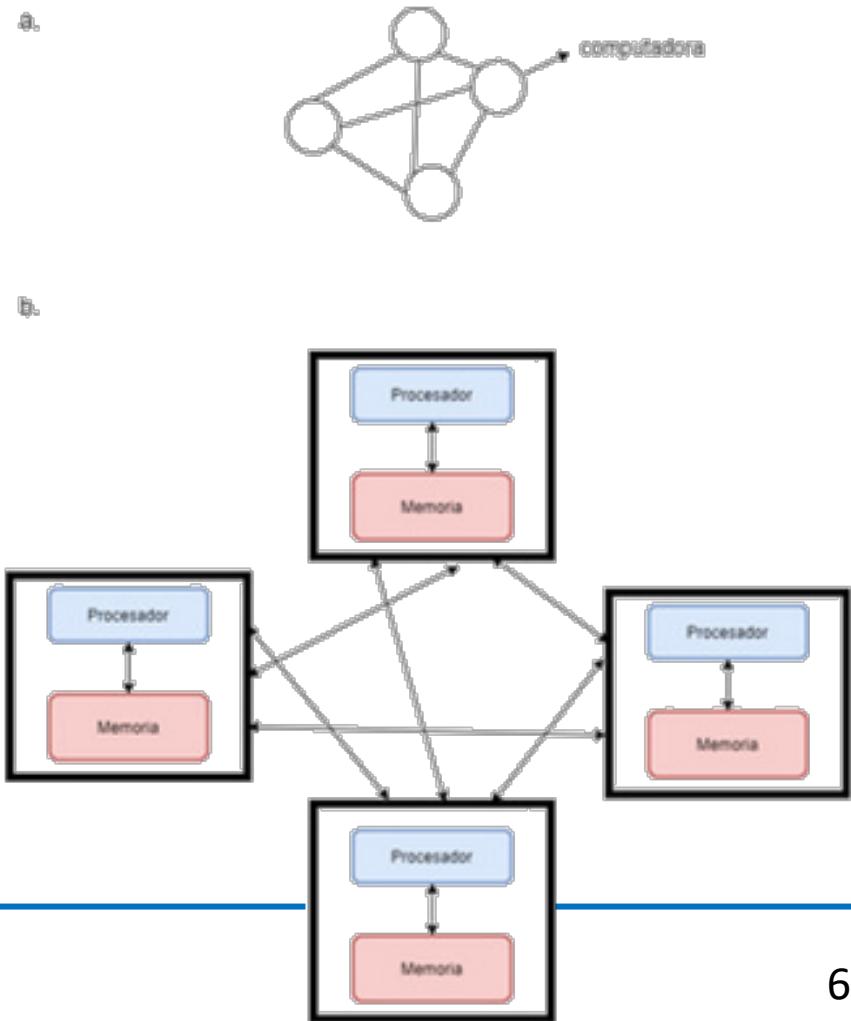
Sistemas distribuidos. Concepto

Sistemas paralelos



Una única computadora

Sistemas distribuidos



Contenidos

Sistemas distribuídos

- Concepto
- **Tipos**
- Arquitecturas

Sistemas distribuidos. Tipos

- Principales tipos de **sistema distribuidos**:
 - Sistemas de información distribuidos
 - Centros de computación distribuida

Sistemas distribuidos. Tipos

- Principales tipos de **sistema distribuidos**:
 - **Sistemas de información distribuidos**
 - Centros de computación distribuida

Sistemas distribuidos. Tipos

- **Sistemas de información distribuidos**
 - Un sistema de **información** distribuido es un conjunto de equipos independientes que actúan de forma transparente **actuando como un único equipo**.
 - Su objetivo es **descentralizar** tanto el **almacenamiento** de la información como el **procesamiento**.

Sistemas distribuidos. Tipos

- **Sistemas de información distribuidos**

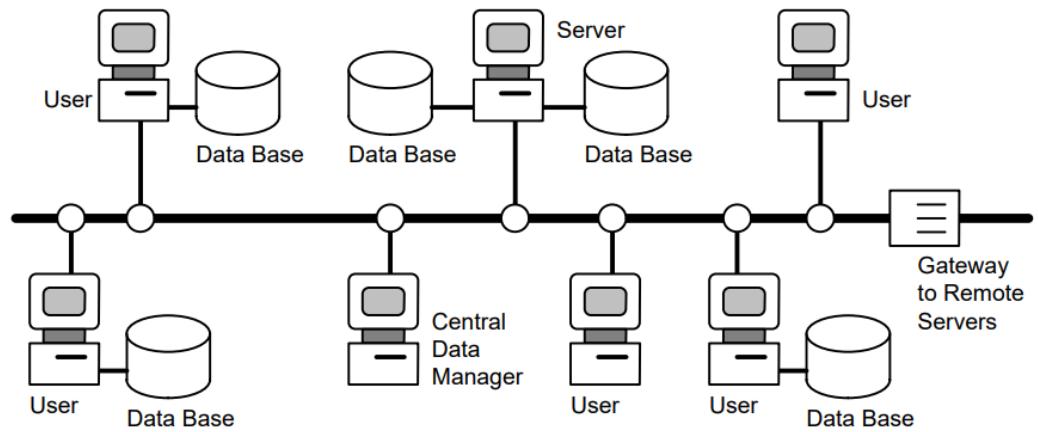
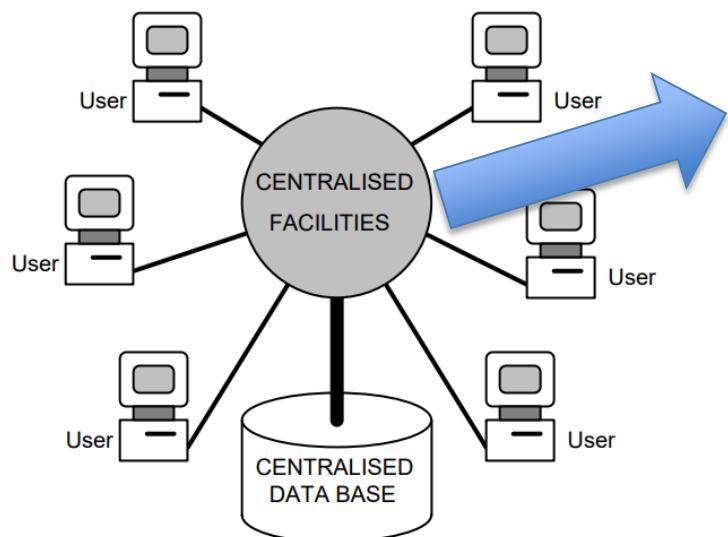


Figure 3. A distributed information system with a central data manager and networked data bases.

Fuente: Zhou, Q., & Evans, A. H. Towards a Multimedia Distributed Spatial Information System.

Figure 2. Centralised facilities are essential for a large stand-alone spatial data base.

Sistemas distribuidos. Tipos

- **Sistemas de información distribuidos**

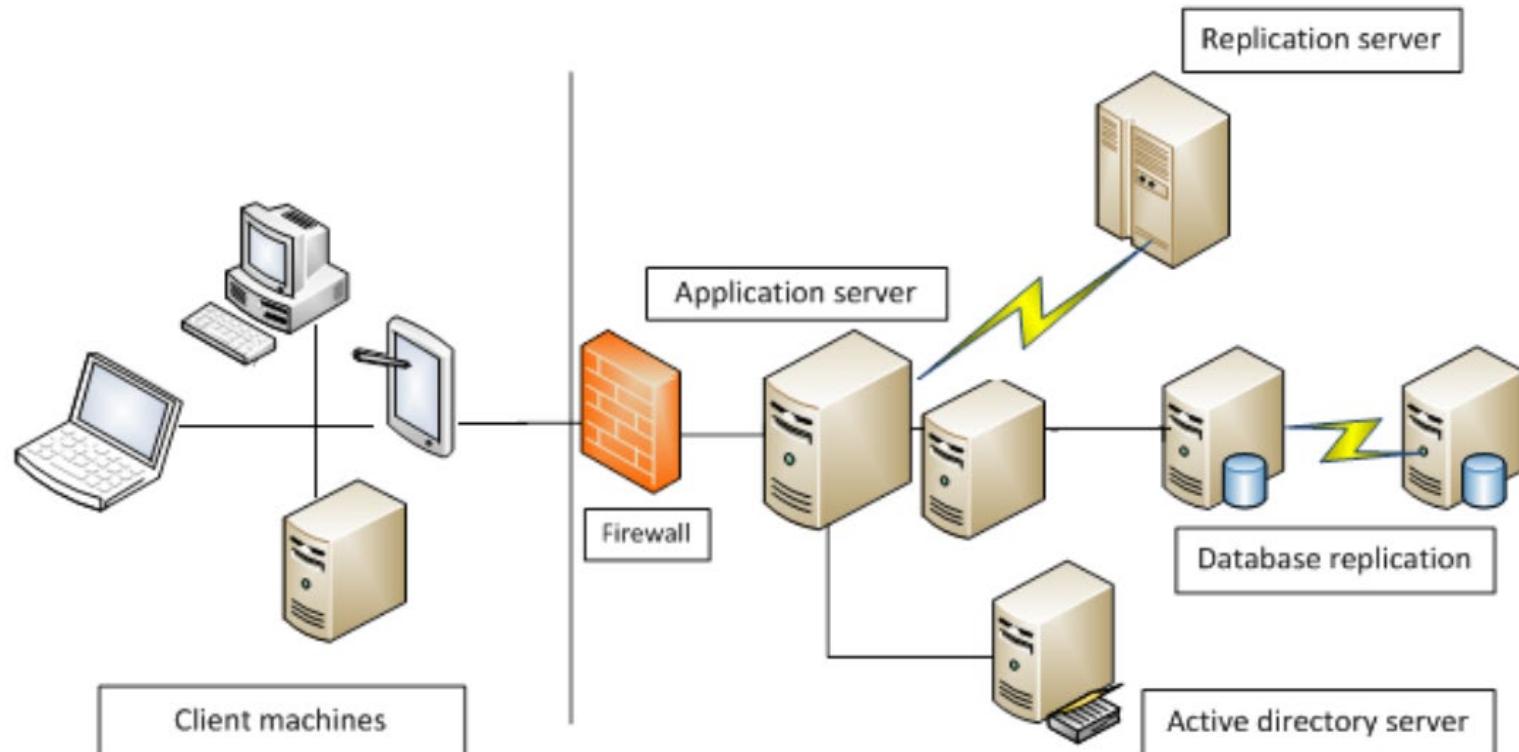


Fig. 1 Distributed information system.

Sistemas distribuidos. Tipos

- Principales tipos de **sistema distribuidos**:
 - Sistemas distribuidos de información
 - **Centros de computación distribuida**

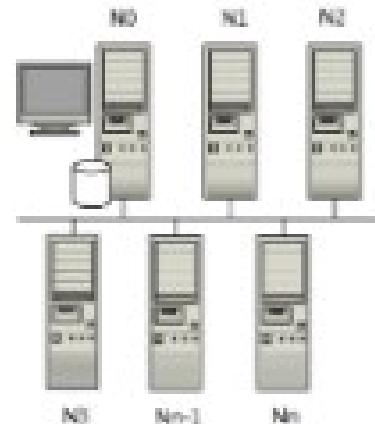
Sistemas distribuidos. Tipos

- **Sistemas de computación distribuida**

- **Objetivo:** realizar tareas de cómputo de **alto rendimiento**

- **Tipos:**

- computación en **clúster**

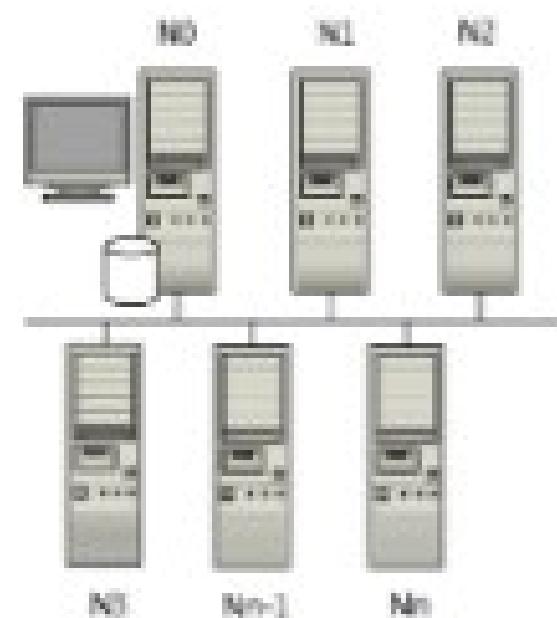


- Computación en malla (**grid**)



Sistemas distribuidos. Tipos

- **Computación en clúster**
 - Ejecución paralela de aplicaciones
 - Entorno **homogéneo** (tipos de máquinas, sistema operativo, etc.)
 - Un **único programa**, de cálculo intensivo, corriendo **en paralelo** en **múltiples máquinas**
 - Un **nodo maestro** gestiona el procesamiento, repartiendo **lotes** a cada máquina.



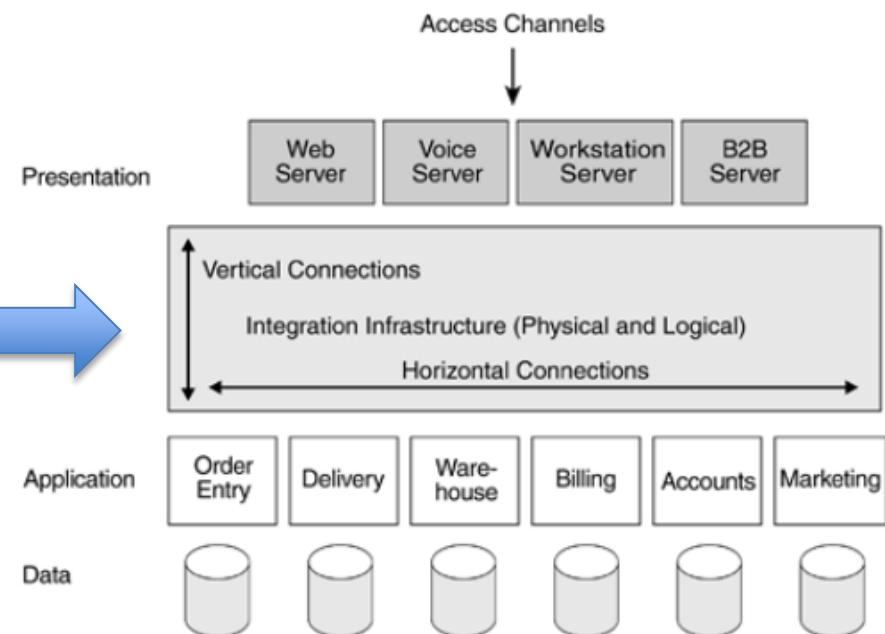
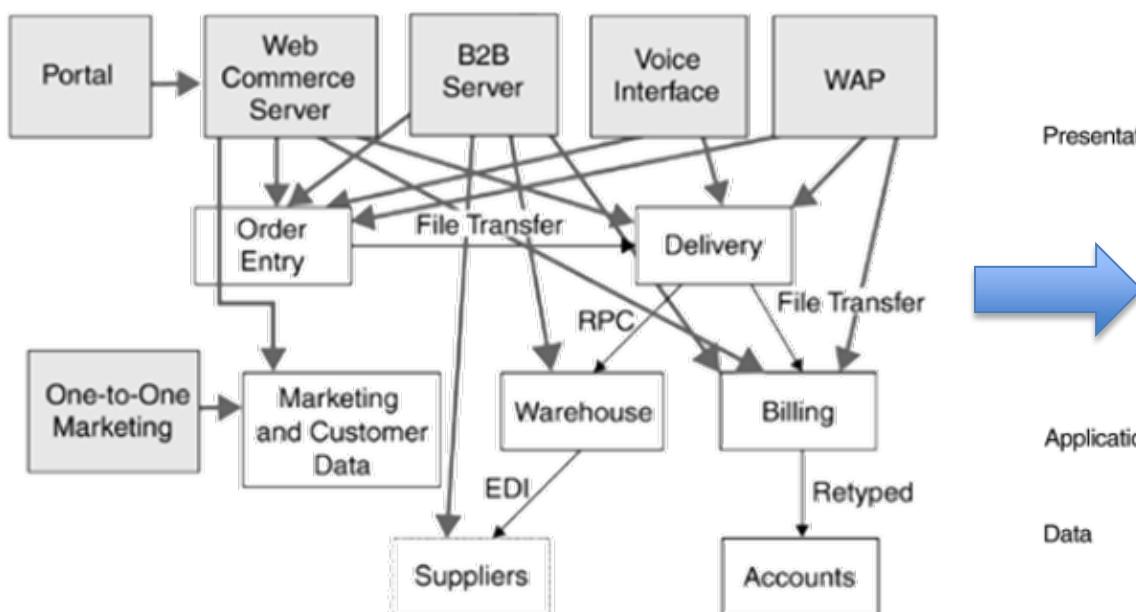
Sistemas distribuidos. Tipos

- **Computación en malla (grid)**
 - Reunir **múltiples programas y recursos de diferentes organizaciones** para permitir la **colaboración** de un grupo de personas o instituciones.
 - Acceso a **recursos compartidos** por ser miembro de la **organización virtual**.
 - Entorno **heterogéneo** (diferentes tipos de máquinas, sistemas operativos, redes, etc.)



Sistemas distribuidos. Tipos

- **Computación en malla (grid)**
 - Requieren **arquitectura**, interfaces y **middleware** de interoperabilidad entre los servicios y entidades participantes



Sistemas distribuídos

- Concepto
- Tipos
- Arquitecturas

Sistemas distribuidos. Arquitecturas

- **Middleware, arquitecturas y protocolos** para interoperabilidad

Point-to-Point



Client/Server



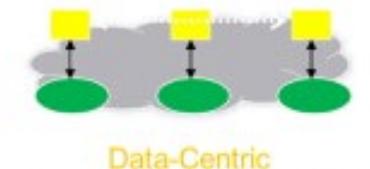
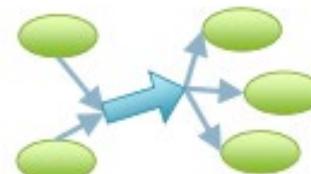
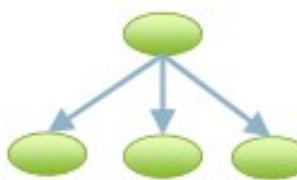
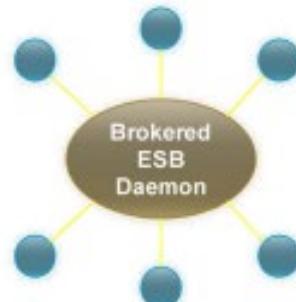
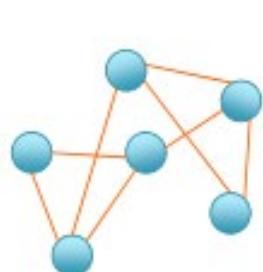
Publish/Subscribe



Queuing



Data-Centric



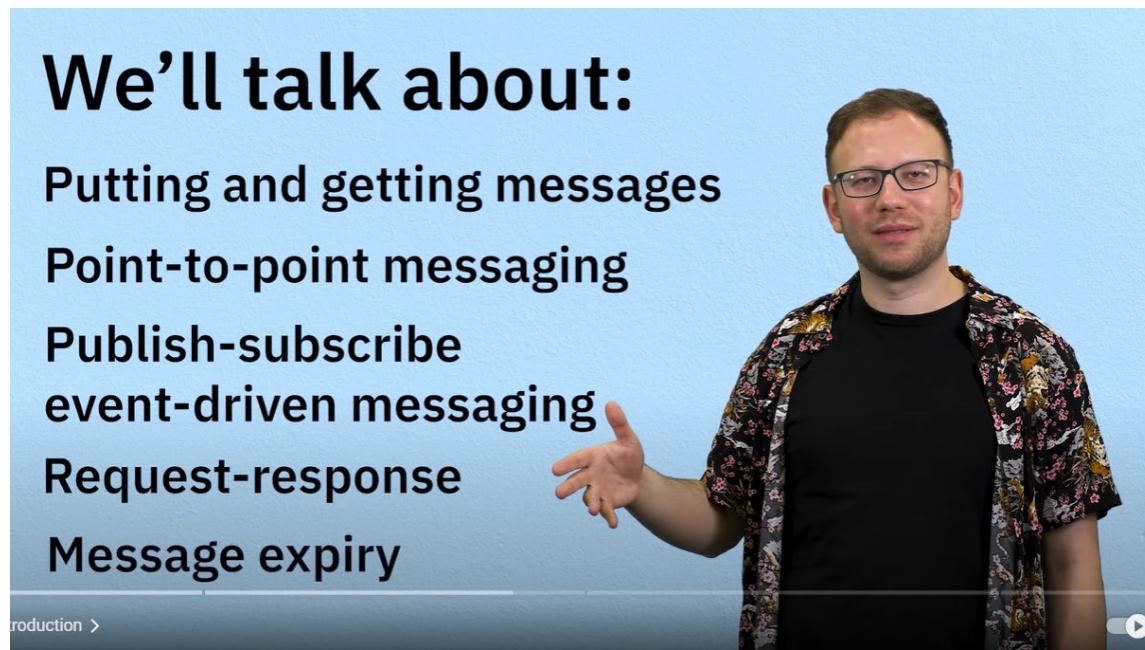
Pub/sub & Queuing:
<https://pandio.com/blog/messaging-patterns-and-data-movement/>

Video: Queues, subscribers, ...

Understanding enterprise messaging patterns

Queues, subscribers, request-response (IBM MQ)

<https://www.youtube.com/watch?v=U03FslufqxY>



We'll talk about:

- Putting and getting messages
- Point-to-point messaging
- Publish-subscribe
- event-driven messaging
- Request-response
- Message expiry

roduction > 



Sistemas distribuidos. Arquitecturas

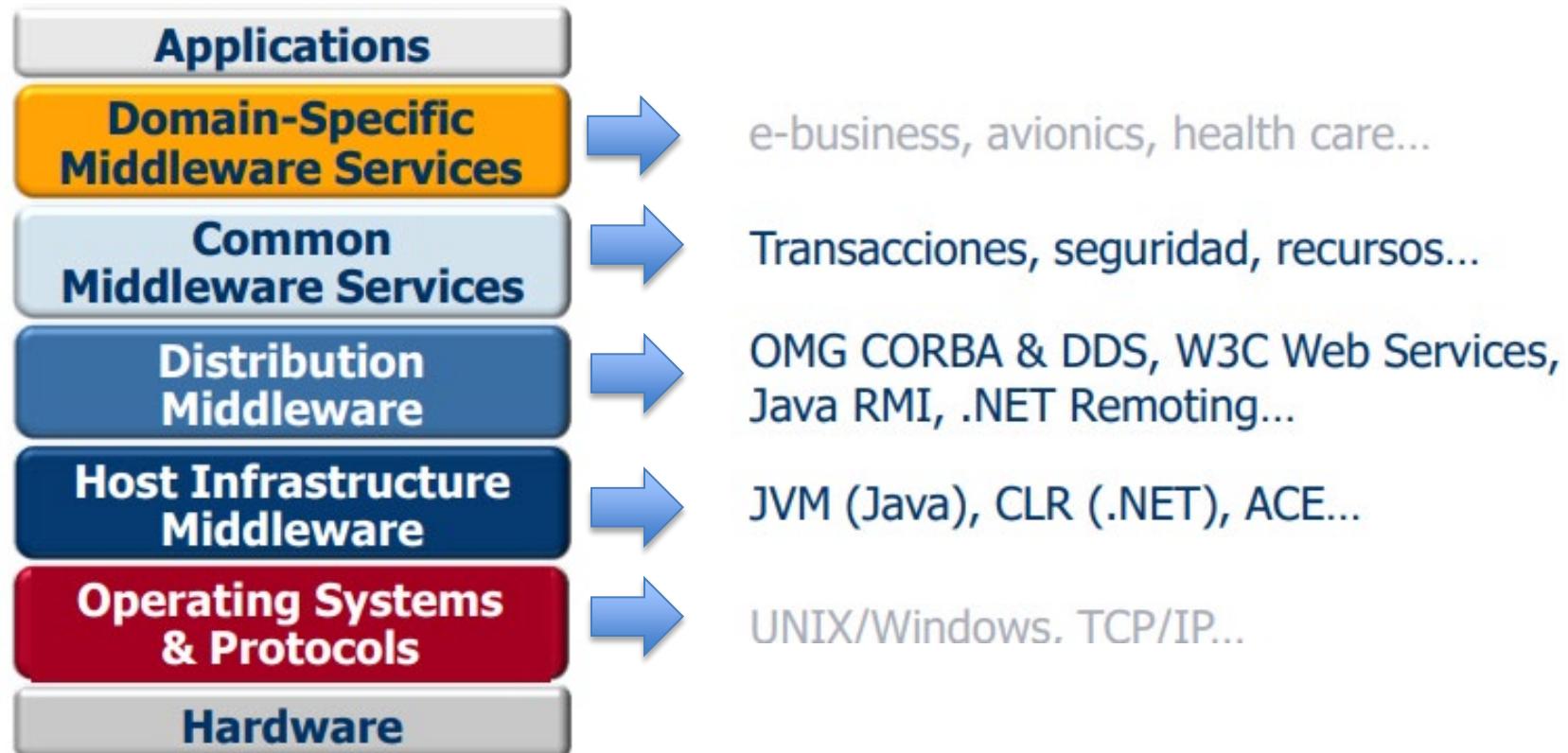
- Relación entre arquitecturas y protocolos
 - Diferentes protocolos (**¡¡a todos los niveles!!**)
 - Orientados a conexión vs no conexión
 - Codificación de datos textual vs binaria
 - Multiplexación (o no) de conexiones
 - Síncrono vs Asíncronos
 - Intercambio de información por mensajes vs memoria compartida
 - Tipo de comunicación (cliente / servidor; peer to peer; broadcast; etc)
 - Etc, etc, etc...



Necesidad de **Middleware** para interoperabilidad

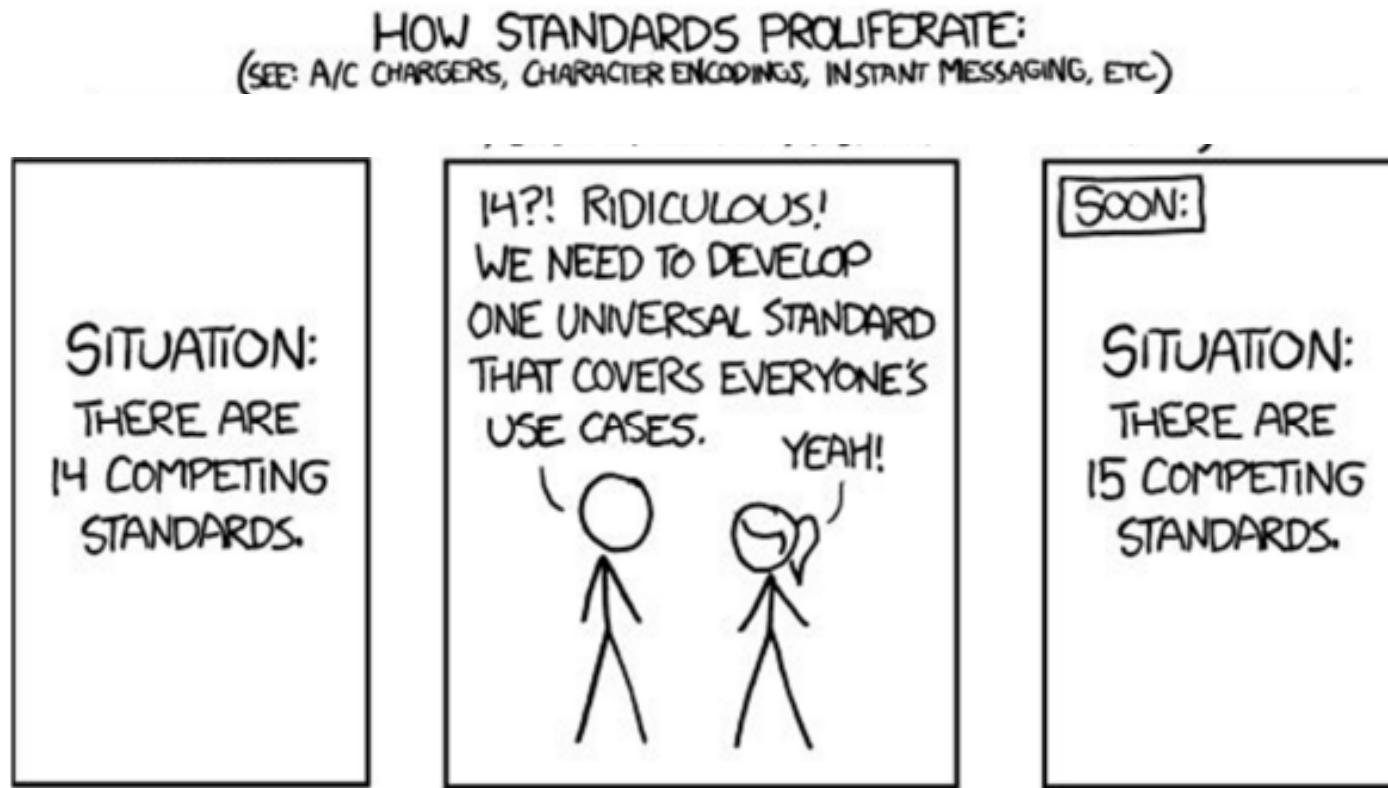
Sistemas distribuidos. Arquitecturas

- Relación entre arquitecturas y protocolos
 - Middleware: arquitectura basada en capas



Sistemas distribuidos. Arquitecturas

- Relación entre arquitecturas y protocolos
 - Middleware: compleja estandarización



Sistemas distribuidos. Arquitecturas

- **Relación entre arquitecturas y protocolos**
 - **Middleware: Ejemplos**
 - **RPC [Remote Procedure Call]**
 - [RPC](#) permite realizar la comunicación entre procesos como si se tratase de simples llamadas a funciones.
 - **Middleware orientado a objetos**
 - [CORBA \(2\)](#) & [DCOM](#)
 - **Middleware basado en componentes**
 - [EJB](#) [Enterprise JavaBeans]
 - [.NET](#) Enterprise Services [COM+]
 - **Servicios**
 - **Servicios web**
 - (WSDL, SOAP, XML,, REST, GraphQL)

Sistemas distribuidos. Arquitecturas

- **Servicio Web: ejemplo**

// Amazon Product Advertising API



```
String AMAZON_ASSOCIATE_TAG = "ikor0c7-20";
String AWS_ACCESS_KEY_ID = "XXXXXXXXXXXXXXXXXXXXXX";
String AWS_SECRET_KEY = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
String ENDPOINT = "webservices.amazon.com"; // .es | .co.uk
```

// Set up the signed requests helper

```
SignedRequestsHelper helper = SignedRequestsHelper.getInstance
    (ENDPOINT, AWS_ACCESS_KEY_ID, AWS_SECRET_KEY);
```

Ejemplos de servicios web para principiantes: <https://programmerclick.com/article/496383664/>

Ejemplos de uso de servicios web Alfresco: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/41>

Sistemas distribuidos. Arquitecturas

• Servicio Web: Más ejemplos

Build the future of live entertainment.

Twitch provides developers products and solutions that extend the lifecycle of entertainment beyond the screen.

Twitch Extensions Twitch Insights **Twitch API** Twitch Game Solutions

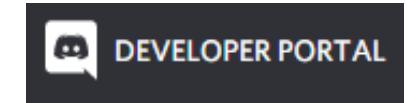
The page displays four main service icons:

- Clips**: Represented by a video camera icon.
- Games**: Represented by a game controller icon.
- Streams**: Represented by a video camera icon with a signal.
- Users**: Represented by a group of people icon.

A purple bar at the bottom right contains the text "Clips API Reference".



Developer Platform



Sistemas distribuidos. Arquitecturas

- Otras alternativas



Lectura: ZooKeeper

Coordinación de procesos distribuidos

<https://zookeeper.apache.org/>



Apache ZooKeeper™ Project Documentation Developers ASF

Welcome to Apache ZooKeeper™

Apache ZooKeeper is an effort to develop and maintain an open-source server which enables highly reliable distributed coordination.

What is ZooKeeper?

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.

Resumen: ZooKeeper

Coordinación de procesos distribuidos

<https://zookeeper.apache.org/>

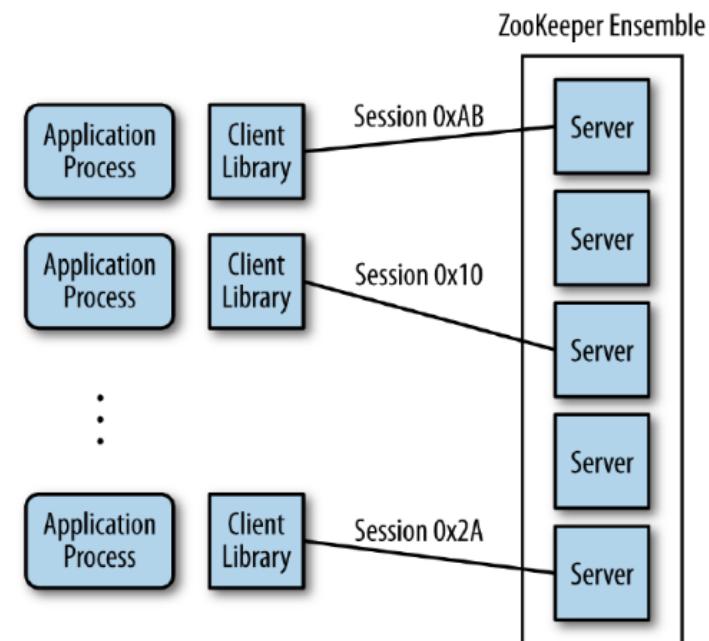
Objetivo:

Coordinación de procesos distribuidos.

Función:

Servicio centralizado para mantener información de configuración, elección de líderes [master election], detección de fallos [crash detection], gestión de grupos [group membership]...

Proporciona algoritmos asíncronos para sistemas distribuidos, difíciles de implementar correctamente.

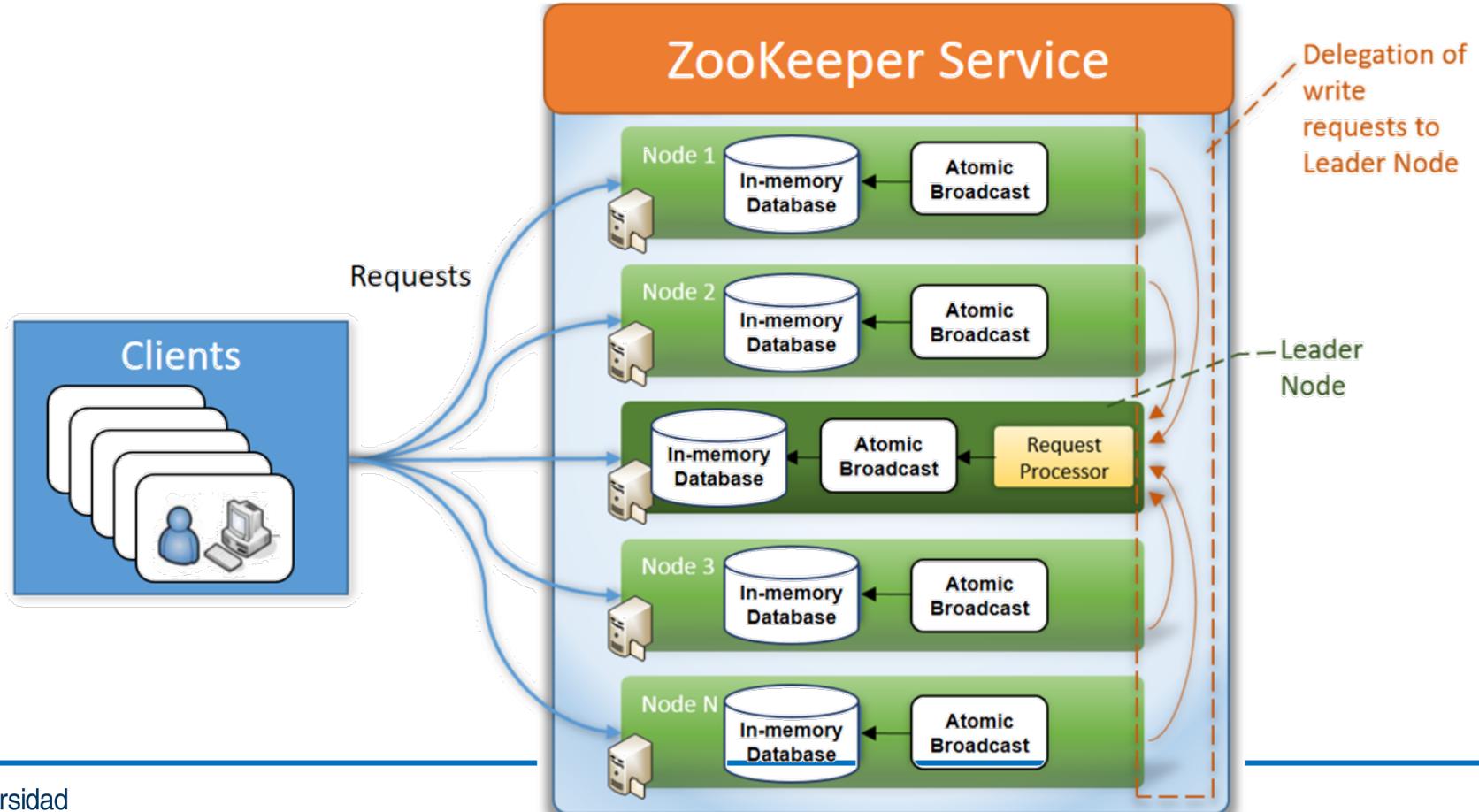


<https://dubbo.apache.org/en/blog/2018/08/07/using-zookeeper-in-dubbo/>

Resumen: ZooKeeper

Coordinación de procesos distribuidos

<https://zookeeper.apache.org/>



Resumen: ZooKeeper

Coordinación de procesos distribuidos

<https://zookeeper.apache.org/>

Usos:

- Apache Hadoop (map-reduce framework): YARN & HDFS
- Apache HBase (NoSQL database)
- Apache Solr (search engine)
- Apache Storm (streaming)
- Yahoo! Fetching Service (crawler)
- Facebook Messages
- eBay
- Twitter
- Netflix
- Zynga
- ...

Para saber más...

Videos: Sistemas distribuidos

Distributed Systems in One Lesson by Tim Berglund

<https://www.youtube.com/watch?v=Y6Ev8Gllbxc>

Distributed Systems Introduction for Beginners

<https://www.youtube.com/watch?v=MwGOjSlld5iE>

¿Qué es la Computación Distribuida? | Domain-Driven Design

https://www.youtube.com/watch?v=ADp7_3ygB2M

¿Qué es un Sistema Distribuido?

https://www.youtube.com/watch?v=MLYM_DeoEoc

QUÉ ES UN SISTEMA DISTRIBUIDO

<https://www.youtube.com/watch?v=lvr9eSk0IkU>

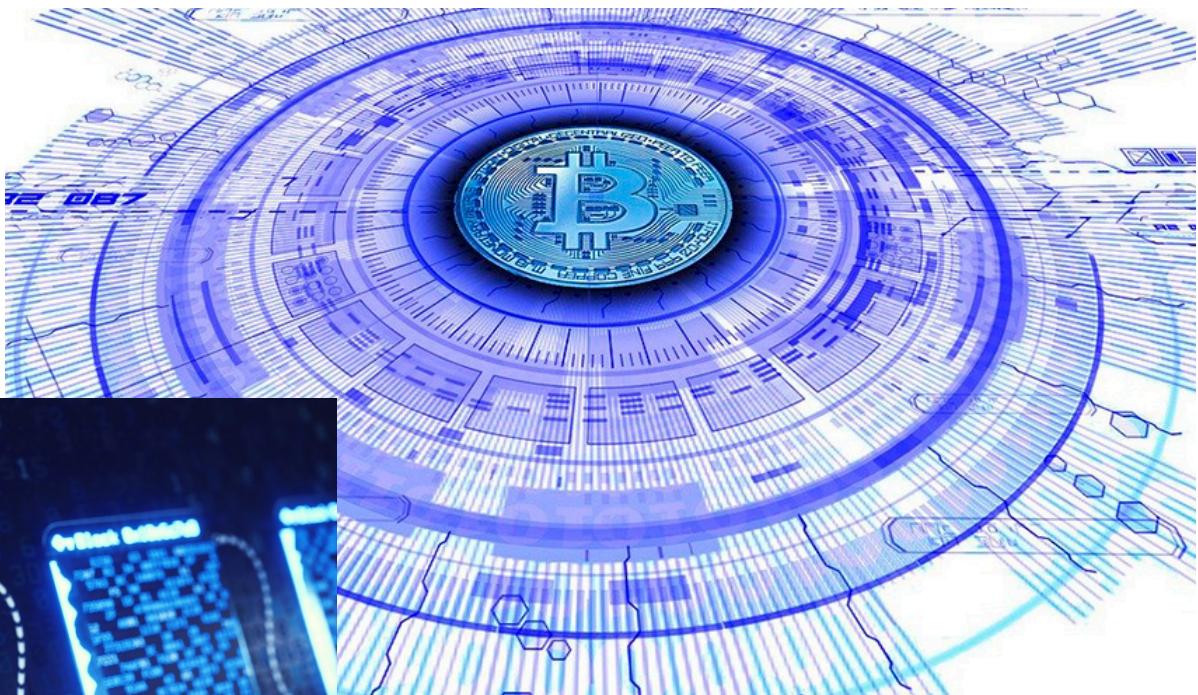
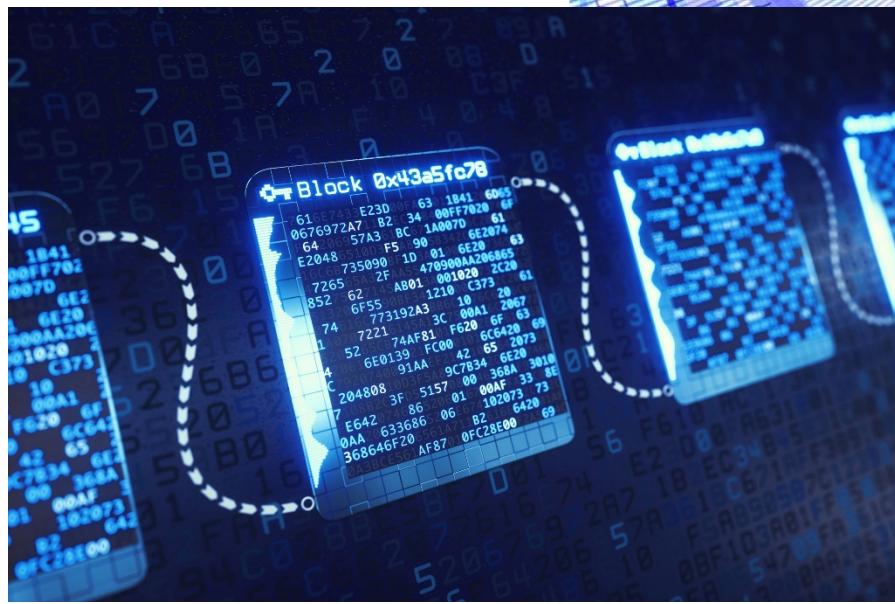
Diferencias entre Librería, Framework y Middleware

<https://www.youtube.com/watch?v=FnaavVdIF1c>



Sistemas distribuidos. Ejemplos

- Blockchain



Sistemas distribuidos. Ejemplos

- **Blockchain**

Blockchain es un **libro de contabilidad compartido e inmutable** que facilita el proceso de registro de transacciones y seguimiento de activos en una red empresarial.

Un **activo** puede ser tangible (una casa, un coche, dinero en efectivo, tierra) o intangible (propiedad intelectual, patentes, derechos de autor, marcas)

Prácticamente cualquier cosa de valor puede rastrearse y negociarse en una red de cadenas de bloques, reduciendo el riesgo y los costes para todos los implicados.

Sistemas distribuidos. Ejemplos

- **Blockchain**

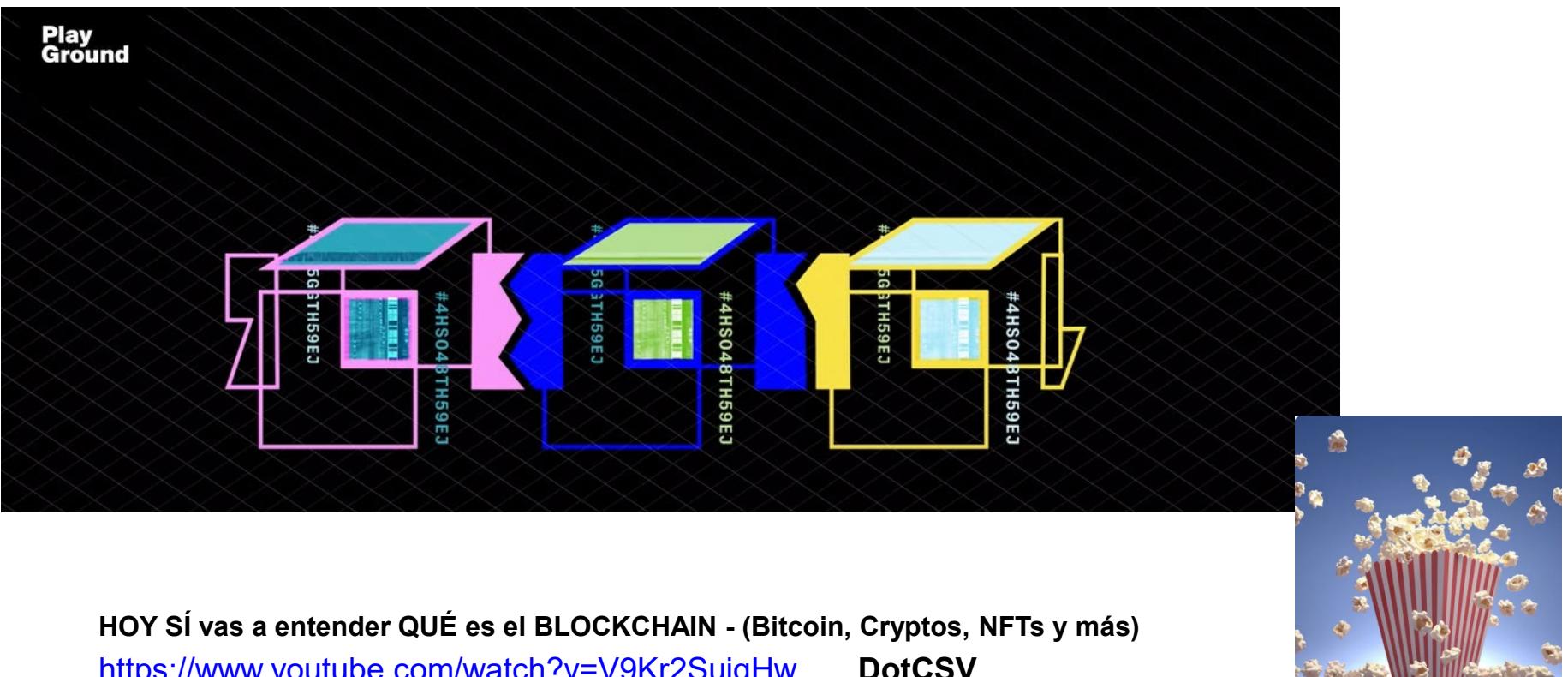
El crecimiento del **comercio electrónico**, la **banca en línea** y las **compras** a través de aplicaciones, junto con la creciente **movilidad** de las personas en todo el mundo, han impulsado el crecimiento del volumen de transacciones.

El volumen de **transacciones** en todo el mundo está creciendo exponencialmente y seguramente aumentarán las complejidades, vulnerabilidades, ineficiencias y costes de los sistemas de transacción actuales.

Video: Qué es "Blockchain" en 5 minutos

Qué es "Blockchain" en 5 minutos (*PlayGround*)

https://www.youtube.com/watch?v=Yn8WGaO_ak



HOY SÍ vas a entender QUÉ es el BLOCKCHAIN - (Bitcoin, Cryptos, NFTs y más)

<https://www.youtube.com/watch?v=V9Kr2SujqHw> DotCSV

Sistemas distribuidos. Ejemplos

- **Blockchain**

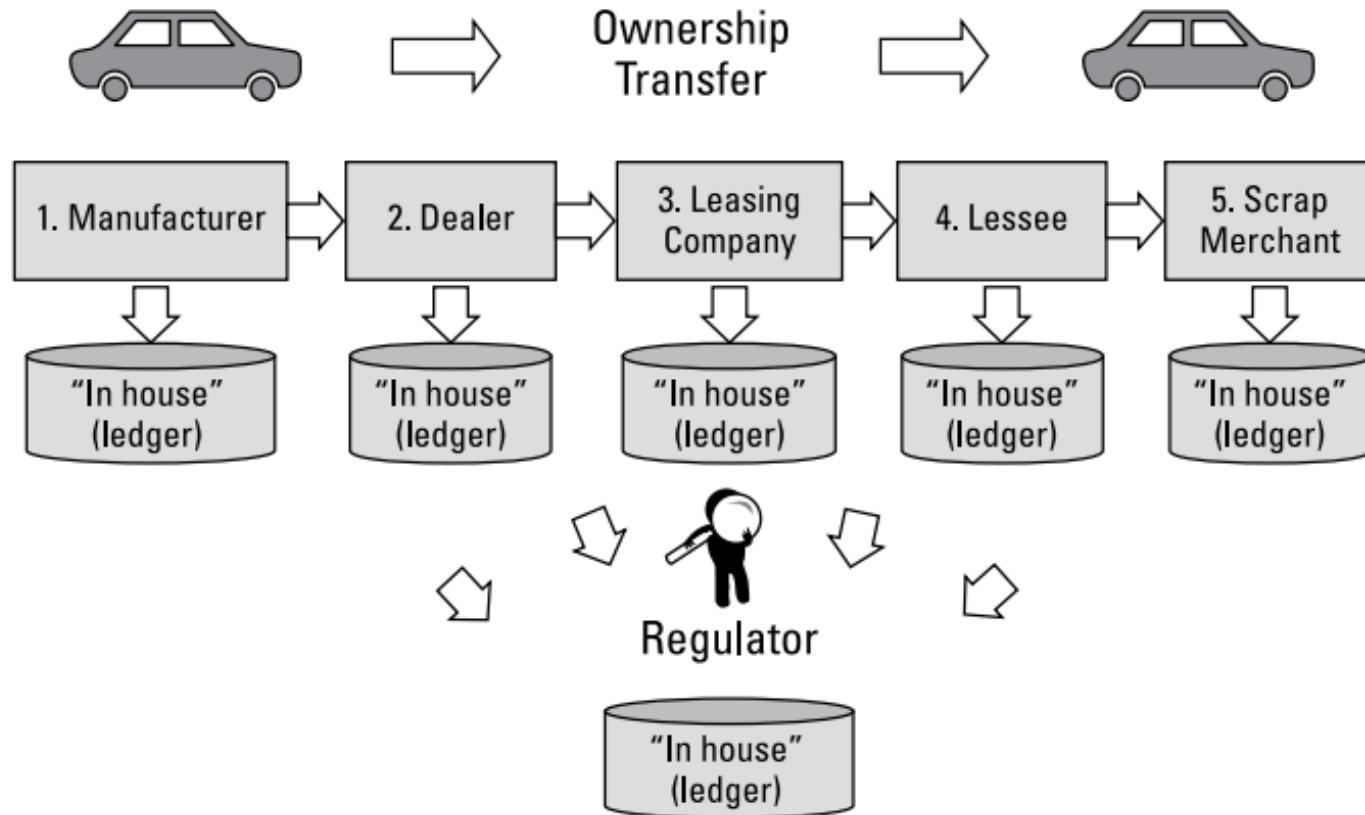


FIGURE 1-1: Tracking vehicle ownership without blockchain.

Sistemas distribuidos. Ejemplos

- **Blockchain**

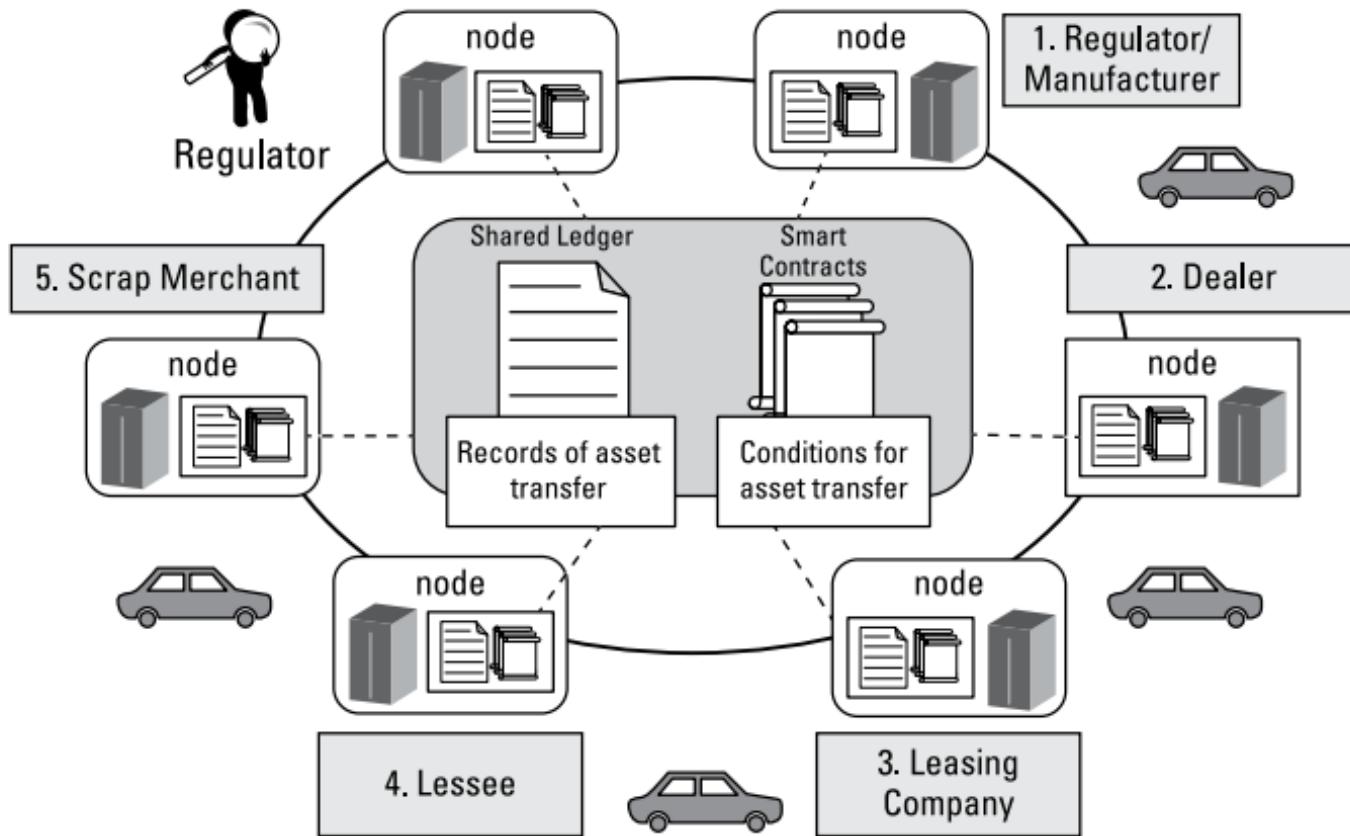


FIGURE 1-2: Tracking vehicle ownership with blockchain.

Sistemas distribuidos. Ejemplos

- **Blockchain**

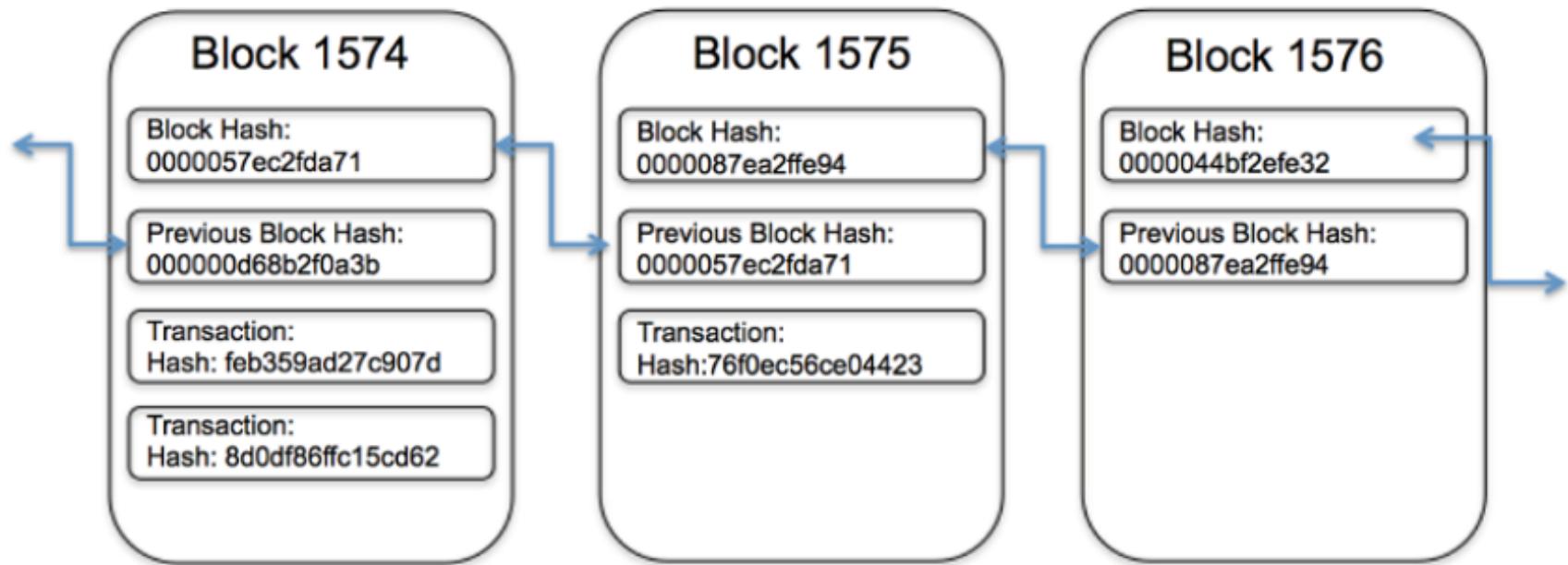


FIGURE 2-1: Blockchain stores transaction records in a series of connected blocks.

Sistemas distribuidos. Ejemplos

• Blockchain: Casos de Uso



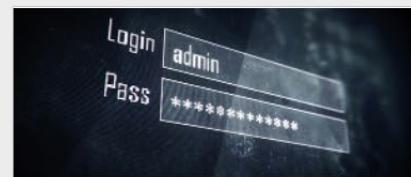
Healthcare

How IBM Blockchain technology powers IBM Digital Health Pass



Banking & Financial Markets

Improving the letter of guarantee banking process with blockchain



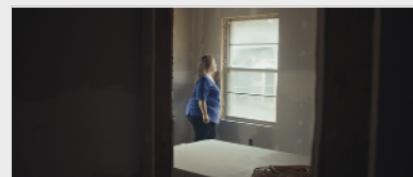
Insurance

AAIS: Enabling regulatory compliance and increased data access using blockchain



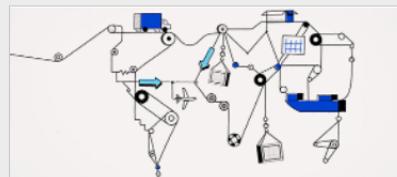
Banking & Financial Markets

ANZ Bank partners with a consortium to transform financial guarantees using IBM Blockchain



Supply Chain, Government, Insurance, Services

Coordinating disaster recovery efforts with blockchain



Banking & Financial Markets

UBS: we.trade offers fast, simple and secure trade transactions based on IBM Blockchain



Supply Chain

Protect Pharmaceutical Product Integrity with the Pharmaceutical Utility Network



Supply Chain

The Vertrax Blockchain is reshaping the oil and gas supply chain with the first multi-cloud deployment of IBM Blockchain Platform

<https://www.ibm.com/blockchain/use-cases/?bkcsol=platform>

Sistemas distribuidos. Ejemplos

- **Blockchain vs Bitcoin**

Para hacer frente a estos y otros retos, el mundo necesita **redes de pago más rápidas**, redes de pago que proporcionen mecanismos para establecer la **confianza**, no requieran equipos especializados, no tengan devoluciones ni cargos mensuales, y ofrezcan una solución de **contabilidad colectiva** para garantizar la transparencia y la confianza.

Una solución que se ha desarrollado para abordar las complejidades, vulnerabilidades, ineficiencias y costes de los actuales sistemas de es el **Bitcoin**

Ejercicio Práctico: Blockchain para Tomates



Ejemplo práctico de Blockchain

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Ejercicio Práctico: Blockchain para Tomates

Objetivo: Conseguir tomates con el mejor sabor, la mejor calidad, y la mayor frescura.

¿qué garantía real tiene el consumidor final sobre la mayor calidad de este producto? ¿qué tiene que ver la frescura de los tomates con la tecnología blockchain?



Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Ejercicio Práctico: Blockchain para Tomates

➤ Escenario:

El caso real de aplicación de blockchain que se propone empieza en los cultivos de tomates de Almería, es decir, en su lugar de **producción**, y terminará en un supermercado de Madrid, **distribuidor** y lugar de adquisición final.

➤ Método:

Empezaremos definiendo los **participantes** de esta transacción para después definir cuáles serán los **activos** a intercambiar en la misma. Por último, vamos a definir la **transacción** que quedará registrada en nuestra blockchain.

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Ejercicio Práctico: Blockchain para Tomates

➤ Participantes

Los **participantes** de esta **transacción** serían **todas las partes interesadas** en tener el tomate con el mejor sabor, y esto va desde el productor al cliente, pasando por diferentes entidades auditadoras, hasta los dispositivos de control de temperatura durante el transporte.

Siendo tan diversos estos participantes, no todos necesitan cada **información** que guardamos en nuestra cadena de bloques, por lo tanto habrá participantes que **sólo conozcan una parte** de esta, mientras que, por ejemplo, el productor y el distribuidor final conocerán toda la cadena:

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Ejercicio Práctico: Blockchain para Tomates

➤ Participantes

Los participantes serían los siguientes:

- ✓ Productor de tomates
- ✓ Empresa de transporte
- ✓ Entidad financiera
- ✓ Dispositivo de control de temperatura y humedad
- ✓ Entidad auditora
- ✓ Entidad certificadora de D.O.
- ✓ Distribuidor

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Caso Práctico: Blockchain para Tomates

➤ Activos

Definiremos los **activos** de la transacción como todos los datos que guardaremos en nuestra blockchain: desde un certificado de denominación de origen, un registro del tiempo de maduración, hasta un historial de temperatura y humedad durante el transporte.

Contaremos con los siguientes **activos**:

- ✓ Certificado de denominación de origen
- ✓ Certificado de maduración óptima
- ✓ Histórico de temperatura y humedad

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Caso Práctico: Blockchain para Tomates

➤ Transacciones

Ahora sólo falta decretar unas **normas** que deben cumplir todos los participantes y que determinarán **qué operación puede realizar cada uno de ellos** y qué parte de información **conocerán** en detalle.

En este caso las **transacciones** podrían ser las siguientes:

- ✓ Certificación de calidad del productor
- ✓ Notificación de recogida para transporte
- ✓ Histórico de temperatura y humedad
- ✓ Certificación de denominación de origen
- ✓ Certificación de calidad de la entidad auditora
- ✓ Notificación de compra del producto

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Caso Práctico: Blockchain para Tomates

➤ Smart contract y su funcionamiento

Una vez conocemos los participantes, los activos y las transacciones, pasamos a **desglosar** todas las **operaciones** que realizaría el smart contract o **contrato inteligente**, o sea, el acuerdo inmutable almacenado y ejecutado en la blockchain.

Fuente: <https://icon-icons.com/es/icono/contrato/194172>



Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Caso Práctico: Blockchain para Tomates

➤ Smart contract y su funcionamiento

1. **Despliegue del smart contract y solicitud del pago al distribuidor:** en este paso el productor de nuestros tomates de Almería despliega en la blockchain el contrato sobre el que vamos a operar. En esta operación únicamente **solicita** el pago.
2. **Controles de calidad:** marcados por el productor, en cuanto a tiempo de recogida, maduración, semillas utilizadas, es decir todo lo referente al producto iría almacenado en este bloque.
3. **Auditoría de calidad:** en este paso la entidad auditora comprobará que el producto cumpla con las normas y lo almacenará en un certificado en el siguiente bloque averiguando que todo lo almacenado en el bloque anterior concuerda con los datos recogidos por ellos.

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Caso Práctico: Blockchain para Tomates

➤ Smart contract y su funcionamiento

4. **La empresa de transporte:** marca el pedido como recogido y verifica que el empaquetado es correcto para su transporte y lo certifica en el siguiente bloque de la cadena.

5. **Control de temperatura y humedad:** un dispositivo conectado a Internet está programado para monitorizar a partir de este punto, por lo tanto en todos los bloques sucesivos enviará los datos para que sean almacenados. Si el transporte fuese largo este nodo se vería replicado tantas veces como sea necesario para almacenar el control constante de los factores de calidad.

6. **Entrega del producto:** en este paso la empresa de transporte marca como entregado el cargamento en Madrid.

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

Caso Práctico: Blockchain para Tomates

➤ Smart contract y su funcionamiento

7. **Certificado de D.O.** : a continuación la entidad certificadora de denominación de origen, en la recepción del cargamento validará que se hayan cumplido todos los requisitos para otorgar el certificado.
8. Por último, el **distribuidor**: nuestro supermercado en Madrid realiza el pago solicitado por el productor y lo marca en la blockchain. En este bloque acaba la comunicación con el smart contract y se realiza el pago.

Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>

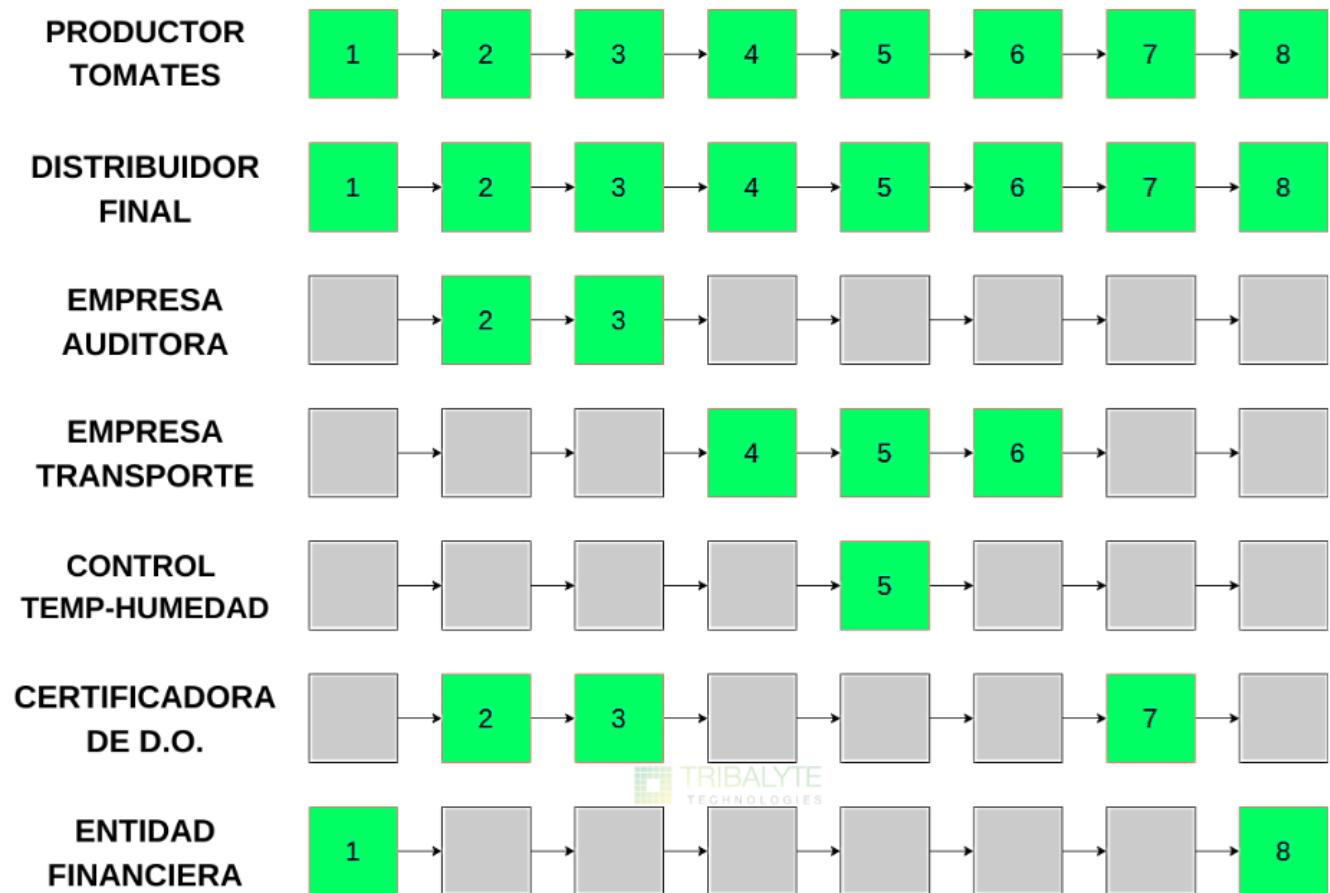
Caso Práctico: Blockchain para Tomates

Leyenda:

Cadena de bloques generada por el smart contract.

En **verde** estarían marcados los bloques que puede ver cada participante y en **gris** los que no.

Los **números** indicarán que **transacción** se está realizando en cada momento.



Fuente: <https://tech.tribalyte.eu/blog-blockchain-ejemplo-facil-y-practico-funcionamiento>



Sistemas distribuidos. Ejemplos

- **Cloud**



Sistemas distribuidos. Ejemplos

• Cloud

La arquitectura **cloud** es la manera en la que los componentes tecnológicos se combinan para construir una nube, en la que los recursos se agrupan mediante la tecnología de virtualización y se comparten en una red.

Los **componentes** de una arquitectura de nube incluyen:

- Una plataforma front-end (el cliente o dispositivo para acceder a la nube)
- Una plataforma back-end (servidores y almacenamiento)
- Un modelo de distribución basado en la nube
- Una red

Esta nube permite la **ejecución** de aplicaciones, lo que brinda a los usuarios finales aprovechar el potencial de los recursos cloud.

Sistemas distribuidos. Ejemplos

- **Cloud: Componentes**

A continuación se indican los componentes fundamentales de la arquitectura de nube:

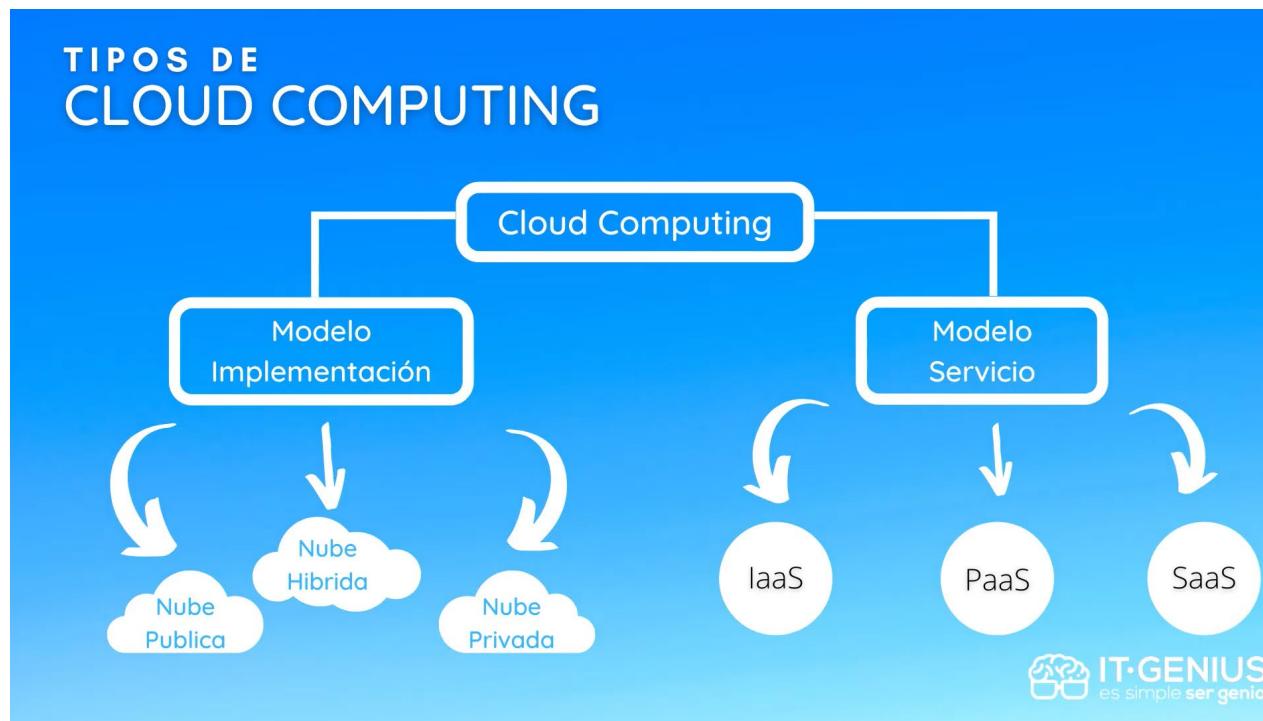
- **Virtualización**
- **Infraestructura**
- **Middleware**
- **Gestión**
- **Software de automatización**

Video: ¿Que Tipos de Cloud Existen?

¿Que Tipos de Cloud Existen?

Nube Publica Privada Hibrida | IaaS SaaS PaaS

<https://www.youtube.com/watch?v=tW2Ut433Mrw>



Sistemas distribuidos. Ejemplos

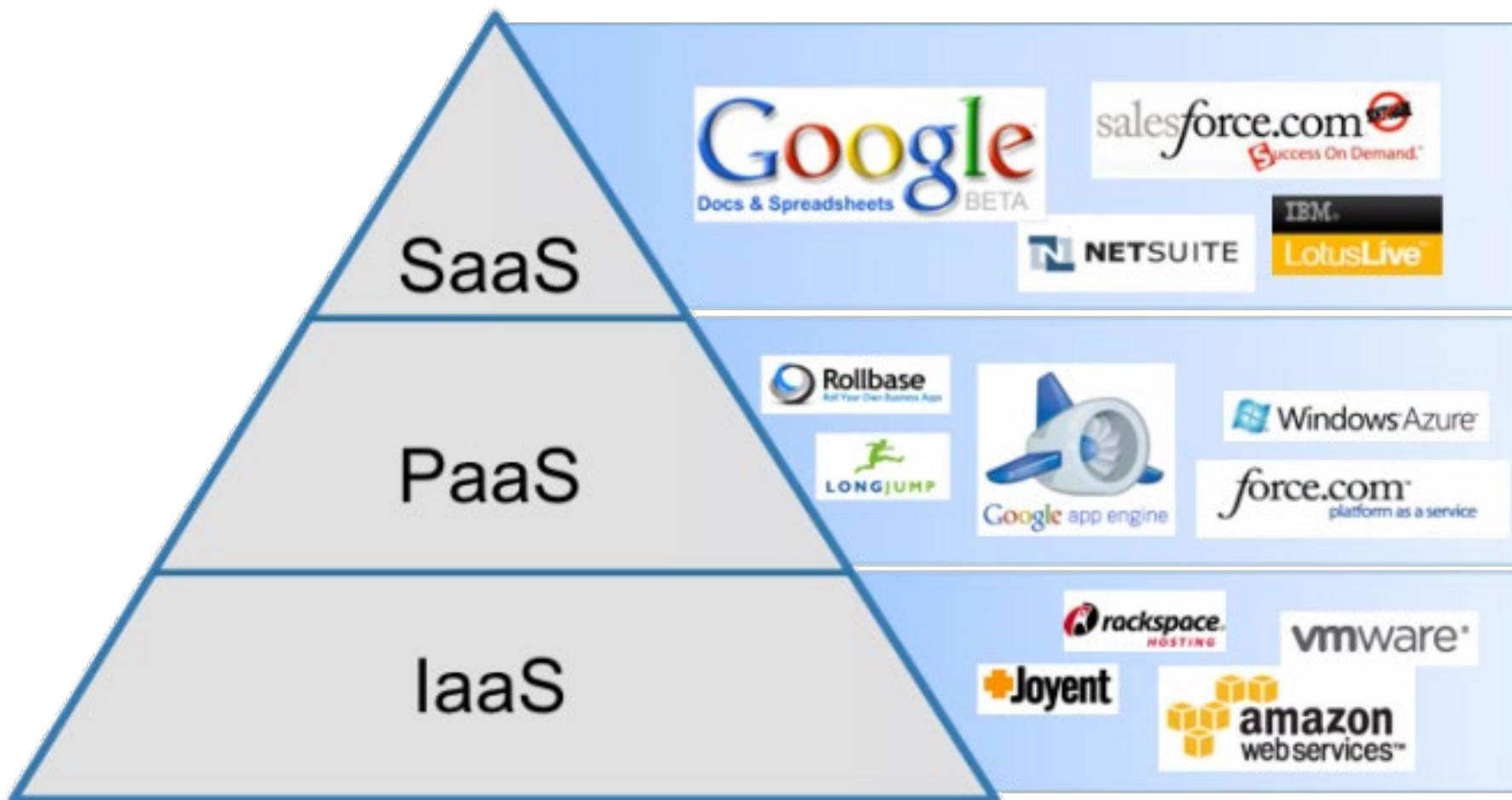
- **Cloud: Modelos**

Hay tres modelos principales de arquitectura de nube que facilitan la migración de las organizaciones a la nube. Cada uno de ellos ofrece determinadas ventajas y prestaciones clave.

1. Software como servicio (**SaaS**)
2. Plataforma como servicio (**PaaS**)
3. Infraestructura como servicio (**IaaS**)

Sistemas distribuidos. Ejemplos

- Cloud: Modelos



Sistemas distribuidos. Ejemplos

- **Cloud: Modelos**

Software como servicio (*SaaS*)

- los **proveedores** de arquitectura SaaS se ocupan de la **distribución y el mantenimiento** de aplicaciones y software para las organizaciones a través de Internet.
- Así, se elimina la necesidad de que los usuarios finales implementen el software localmente.
- Se suele **acceder** a las aplicaciones SaaS a través de una **interfaz web** disponible desde una amplia variedad de dispositivos y sistemas operativos.

Sistemas distribuidos. Ejemplos

<https://aws.amazon.com/solutions/saas/>



Products Solutions Pricing Documentation Learn Partner Network AWS

Software-as-a-Service (SaaS) on AWS

Learn. Build. Innovate.

Más ejemplos: Google Apps, Slack,
Zoom , Square , Jira, Confluence,
Trello, Microsoft Office 365, HipChat,
Hootsuite...

Sistemas distribuidos. Ejemplos

- **Cloud: Modelos**

Plataforma como servicio (*PaaS*) 1/2

- mediante este modelo de nube, el **proveedor de servicios** ofrece una **plataforma** informática y una **pila de soluciones** como servicio, que suele incluir **middleware**.
- Las organizaciones pueden aprovechar esa plataforma para **crear una aplicación o un servicio**.
- El **proveedor** de servicios de nube proporciona las **redes**, los **servidores** y el **almacenamiento** necesarios para alojar una aplicación, mientras que el **usuario final** supervisa la **implementación** del software y la **configuración**.

Sistemas distribuidos. Ejemplos

- **Cloud: Modelos**

Plataforma como servicio (*PaaS*) 2/2

- PaaS es una mejor opción para las organizaciones que tienen menos recursos para desarrollar y administrar aplicaciones.
- PaaS no elimina totalmente la necesidad de desarrolladores, pero **optimiza** las operaciones de desarrollo e implementación y los agrupa con la infraestructura de alojamiento.
- Por esa razón, **PaaS** puede ser **más atractivo para las pequeñas y medianas empresas** que desean que las aplicaciones salgan más rápido.

Sistemas distribuidos. Ejemplos

- **Cloud: Modelos**

Infraestructura como servicio (IaaS)

- según este modelo de la nube en su forma más simple, un **proveedor** externo proporciona la **infraestructura** necesaria, con lo que se **elimina la necesidad** de que las organizaciones **adquieran** servidores, redes o dispositivos de almacenamiento.
- Por su parte, las **organizaciones** gestionan **su software y sus aplicaciones**, y solo **pagan por la capacidad** que necesitan en un momento dado.

Sistemas distribuidos. Ejemplos

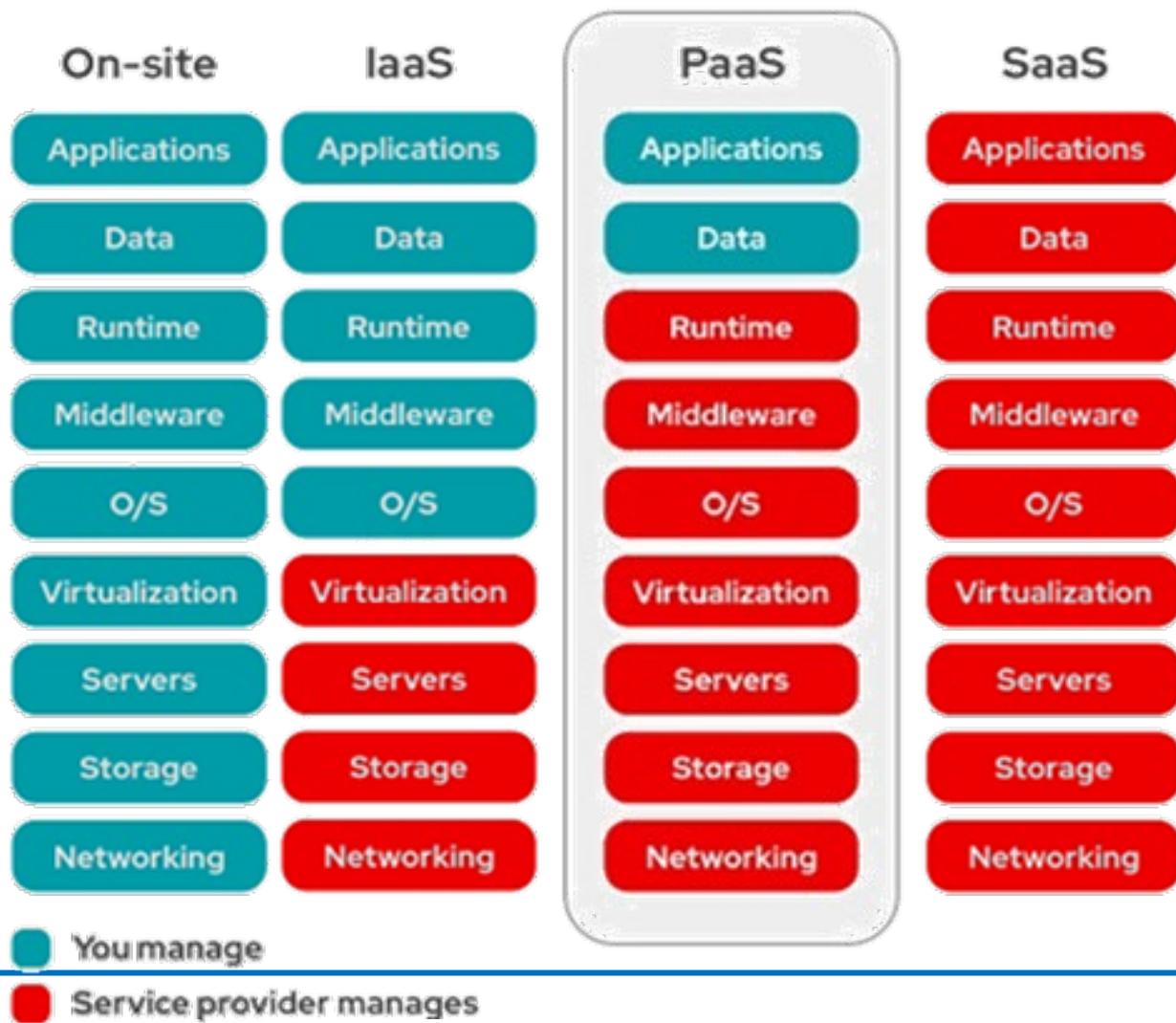
- **Cloud: Modelos**

Infraestructura como servicio (IaaS)

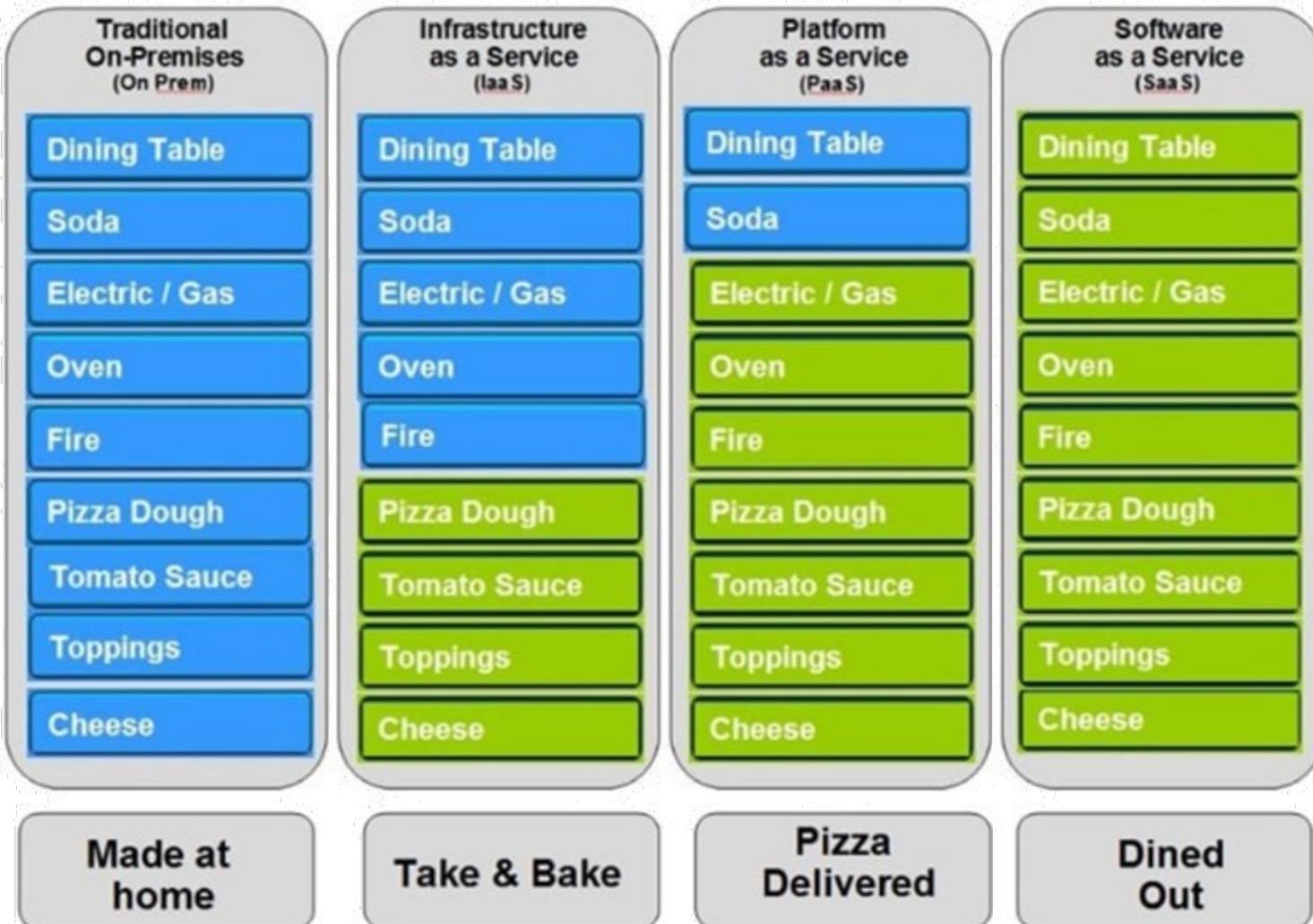
- IaaS suele ser la mejor opción para las empresas que tienen los recursos para **desarrollar, implementar y administrar aplicaciones, sin la ayuda del proveedor.**
- IaaS también es preferible si un usuario requiere componentes que no son proporcionados por una oferta de PaaS, como una herramienta de desarrollo **específica** o un sistema operativo **personalizado.**

Sistemas distribuidos. Ejemplos

Comparativa



Pizza as a Service



You Manage

Vendor Manages



Sistemas distribuidos. Ejemplos

- **Cloud: Ventajas**

Las organizaciones tienen muchas razones para adoptar una arquitectura de nube. A continuación se indican las principales:

- Acelerar la **distribución** de aplicaciones nuevas.
- Aprovechar la arquitectura nativa de nube, como Kubernetes, para **modernizar** las aplicaciones y **acelerar** la transformación digital.
- Garantizar la **conformidad** respecto a las **normativas** más recientes.
- Aportar **transparencia** a los recursos para **reducir costes** y **prevenir** las vulneraciones de datos.

Sistemas distribuidos. Ejemplos

- **Cloud: Ventajas** (continuación)
 - Posibilitar un aprovisionamiento **más rápido** de recursos.
 - Utilizar la arquitectura de **nube híbrida** para aplicar escalabilidad en tiempo real a las aplicaciones a medida que cambian las necesidades empresariales.
 - Cumplir los **objetivos de servicio** de manera constante.
 - Usar la arquitectura de referencia de nube para obtener información sobre los **patrones** de gastos en TI y la utilización de la nube.

Sistemas distribuidos. Ejemplos

• **Cloud: Tipos de Cloud**

Aunque no hay dos nubes iguales, existen varios modelos comunes de arquitectura de nube, entre los que se incluyen:

- Arquitectura de **nube pública**
- Arquitectura de **nube privada**
- Arquitectura de **nube híbrida**
- Arquitectura de **multinube**

Sistemas distribuidos. Ejemplos

- **Cloud: Tipos de Cloud**

Arquitectura de nube pública

- los **recursos** informáticos de una nube pública son **propiedad** de un proveedor de servicios de nube, que también los gestiona.
Proveedor
- Estos **recursos se comparten y redistribuyen** entre varios clientes a través de Internet.
- Entre las **ventajas** de la nube pública se cuentan:
 - costes operativos reducidos
 - escalabilidad sencilla
 - escasa o nula necesidad de mantenimiento.

Sistemas distribuidos. Ejemplos

• Cloud: Tipos de Cloud

Arquitectura de nube privada

- Es una nube de **propiedad y gestión privada**, que generalmente reside en el propio centro de datos local de la empresa.

- Aunque suele ser **más cara** que las soluciones de nube pública, la arquitectura de nube privada es **más personalizable** y puede ofrecer **estrictas opciones de conformidad y seguridad de datos**.



PRIVATE CLOUD

Fuente: <https://apser.es/define-private-cloud-beneficios/>

Sistemas distribuidos. Ejemplos

- **Cloud: Tipos de Cloud**

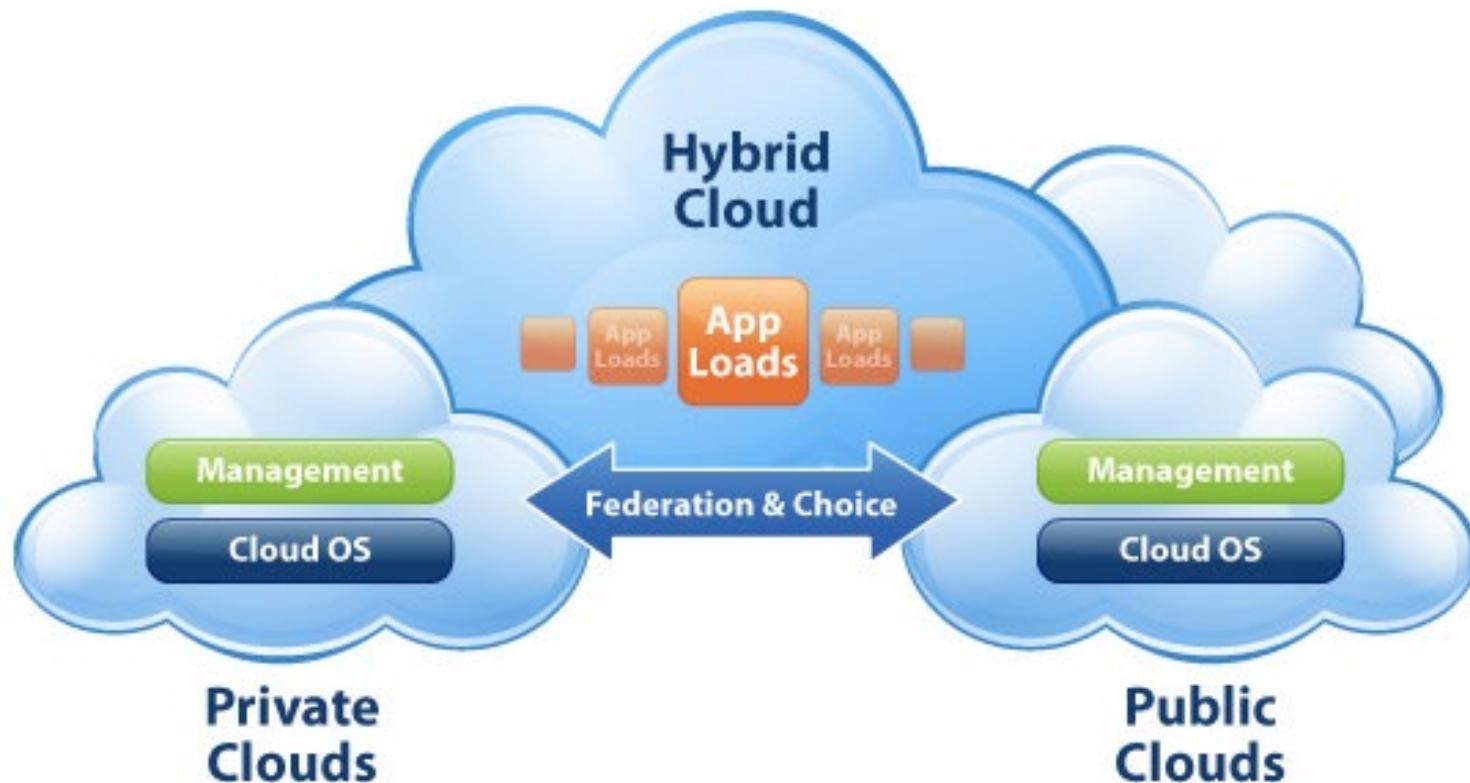
Arquitectura de nube híbrida

- Un entorno de nube híbrida aúna combina la **eficiencia operativa** de la nube pública y las prestaciones de **seguridad de datos** de la nube privada.
- Al usar arquitecturas de nube pública y privada, las nubes híbridas ayudan a consolidar los recursos de TI y permiten a las organizaciones **migrar cargas de trabajo entre entornos** en función de sus requisitos de TI y de seguridad de los datos.

Sistemas distribuidos. Ejemplos

- **Cloud: Tipos de Cloud**

Arquitectura de nube híbrida



Fuente: <https://routingtheworld.files.wordpress.com/2012/03/vmware-stratagen-hybrid-cloud-computing-solution.jpg>

Sistemas distribuidos. Ejemplos

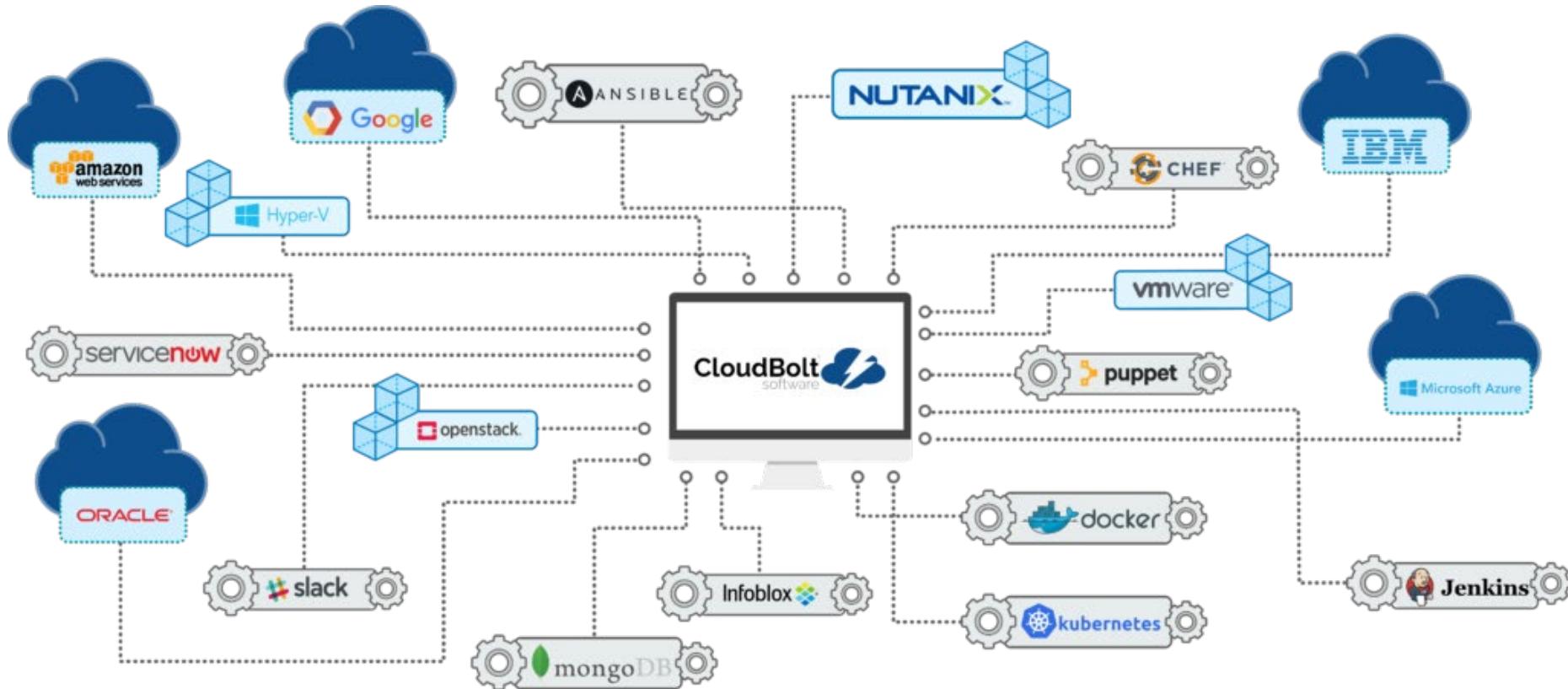
- **Cloud: Tipos de Cloud**

Arquitectura multinube

- Una arquitectura multinube es aquella que **abarca varios servicios de nube pública**.
- **Ventajas** de un entorno multinube:
 - mayor **flexibilidad** para elegir e implementar los servicios de nube para satisfacer los diversos requisitos organizativos.
 - más **independencia**, lo cual permite ahorrar costes y reduce la probabilidad de depender de un solo proveedor de servicios de nube.

Sistemas distribuidos. Ejemplos

- Multicloud



Fuente: <https://www.jorgealdana.pro/blog/informatica-computacion/servidores/multicloud-como-administrar-varias-nubes-en-un-solo-panel/>