Clase 2 sistemas distribuidos

70-80 filse system arquitectura jerarquica

90 c/s-arquitectura (cliente servidor)

2000 2p2

2010 microservicios

Sistemas distribuidos para manejos de grandes volúmenes de datos

Bd relacionales normalmente se usa para estructurar una catidad de datos no tan grande

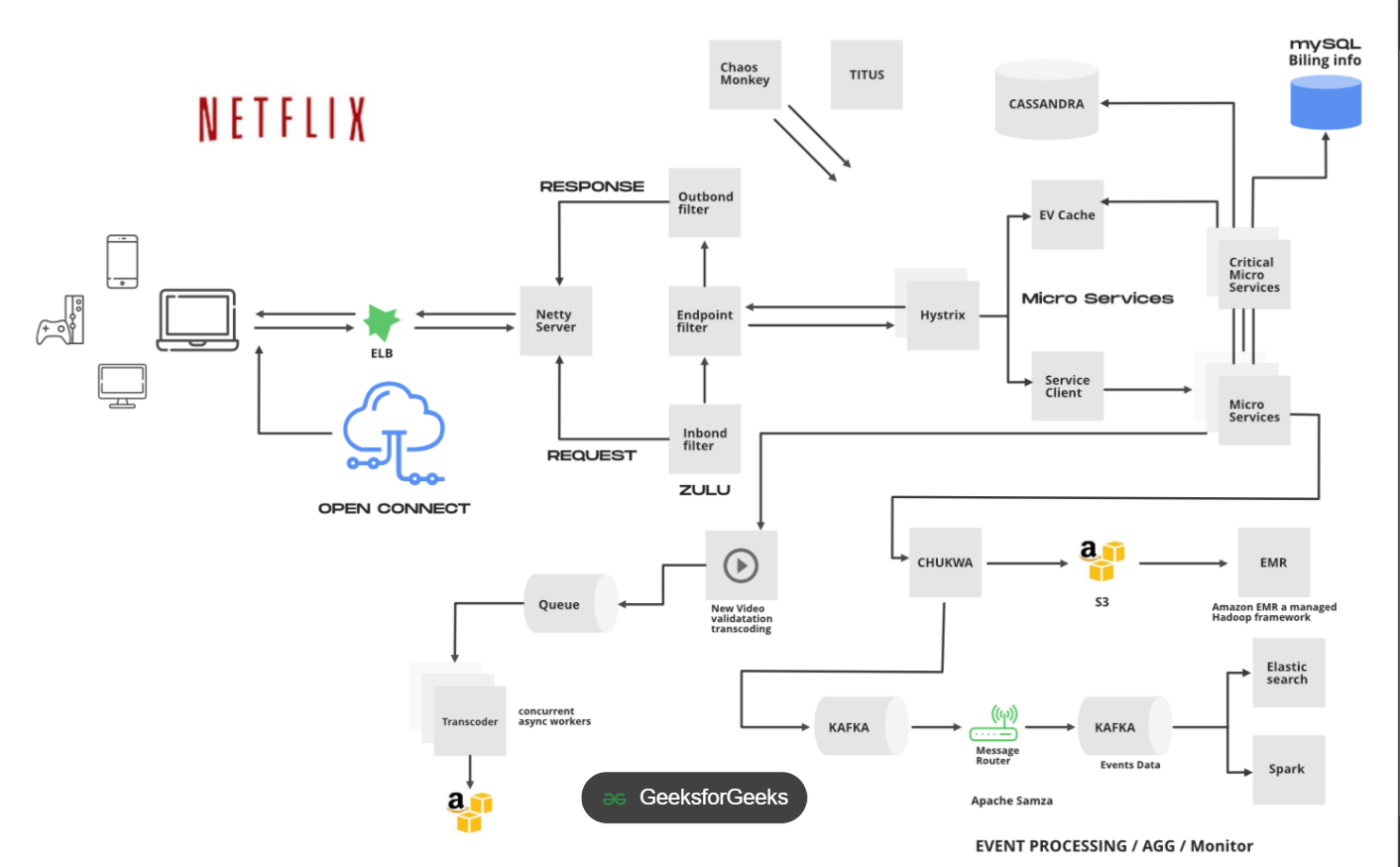
Bd no relacionales son flexibles para manejar una gran cantidad de datos

En arquitectura DE MICROSERVICIOS hay una base de datos dedicada por servicio, hay un despliegue autónomo de cada servicio, y pues servicios pequeños e independientes (API GATEWAY, service Discovery, Contenedores y orquestación) algunas desventajas podría ser la latencia para la comunicación entre los microservicios y que las transacciones son de manera distribuida

<https://www.cloudmasters.es/wp-content/uploads/2019/11/imagen-12-1024x653.png>

para el despliegue de paginas web o servicios en la nube, hoy en dia se usan los principales servidores como (azure, amazon, gogle) los cuales ofrecen los diferente microservicios, son configurables y el precio varia según el nivel de transacción.

Ahora vamos a ver la arquitectura de una plataforma con sus microservicios, como Netflix



Cap NoSQL bd no relacionales

ACID SQL bd relacionales

En el teorema de CAP hay algunas variantes o bases tradicionales como:

CA: SQLserver, MySQL (consistency, availability)

AP: cassandra, couchDB, Amazon DynamoDB (availability partition tolerance)

CP: redis, mongo DB (consistency, partition tolerance)

**CP (Consistencia + Tolerancia a particiones):**  
Prioriza que todos vean datos iguales, pero puede sacrificar disponibilidad.  
Ejemplo: bases como HBase, MongoDB (modo fuerte).

**AP (Disponibilidad + Tolerancia a particiones):**  
Siempre responde, pero los datos pueden no estar actualizados inmediatamente (eventual consistency).  
Ejemplo: Cassandra, DynamoDB.

**CA (Consistencia + Disponibilidad):**  
Posible en sistemas **no distribuidos** o sin particiones de red.  
Ejemplo: bases SQL tradicionales en un solo servidor.

 Si necesitas **exactitud total** (ejemplo: transacciones bancarias) → prefieres **CP**.

 Si necesitas **alta disponibilidad** (ejemplo: redes sociales, catálogos de productos) → prefieres **AP**.

Tener en cuenta palabras clave como arquitectura del sistema

Arquitectura de datos: bd relacionales y modelar data set

SD: bd no relacinales

IA: trabajar con datos

----- ACTIVIDAD

Tener en cuenta el proyecto final

Hacer el catalogo con base de datos MONGO DB