La Evolución del Big Data

Durante los últimos 20 años, los avances tecnológicos crearon una nueva forma de procesar y entender los datos:

🔹 MapReduce (Google, 2004)

El primer modelo que permitió procesar enormes volúmenes de datos distribuidos en múltiples máquinas.
Fue el origen conceptual del Big Data moderno.

🔹 Las 3V del Big Data (Doug Laney, 2001–2012)

Para describir el nuevo fenómeno de datos masivos:

- **Volumen**: demasiados datos para sistemas tradicionales
- **Velocidad**: llegan continuamente, en tiempo real
- **Variedad**: estructurados, no estructurados y semiestructurados

🔹 Bases de Datos NoSQL (2007 en adelante)

Surgen porque los modelos relacionales no podían escalar:
MongoDB, Cassandra, HBase → diseñadas para grandes volúmenes y alta disponibilidad.

🔹 Hadoop & YARN (2008–2012)

El ecosistema Hadoop democratiza MapReduce y habilita clusters distribuidos de bajo costo.

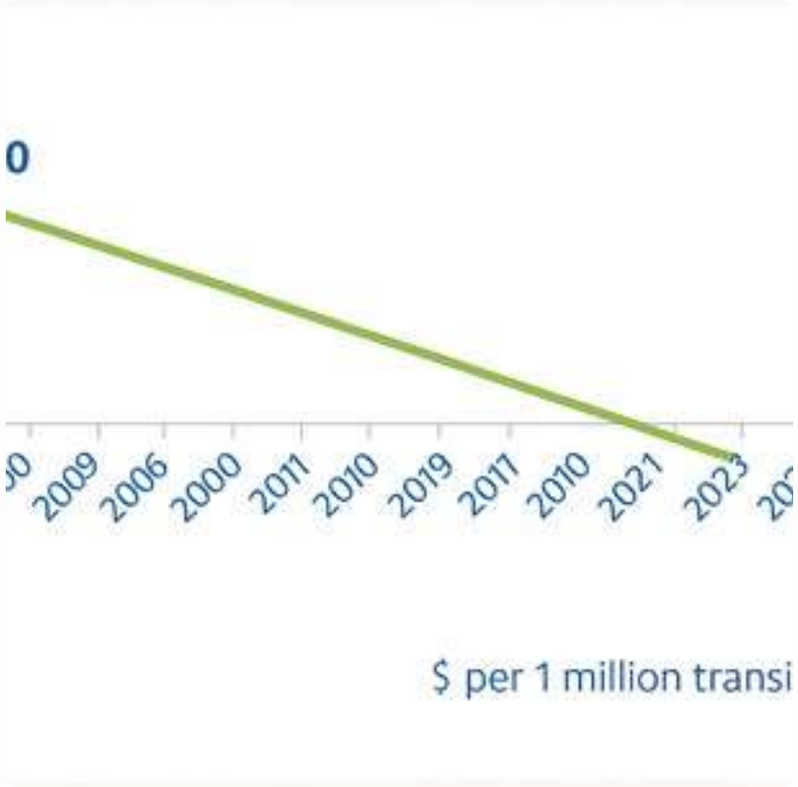
🔹 Apache Spark (2013)

Modelo de procesamiento en memoria, mucho más rápido que Hadoop y clave para machine learning y streaming.

🔹 Apache Kafka (2014)

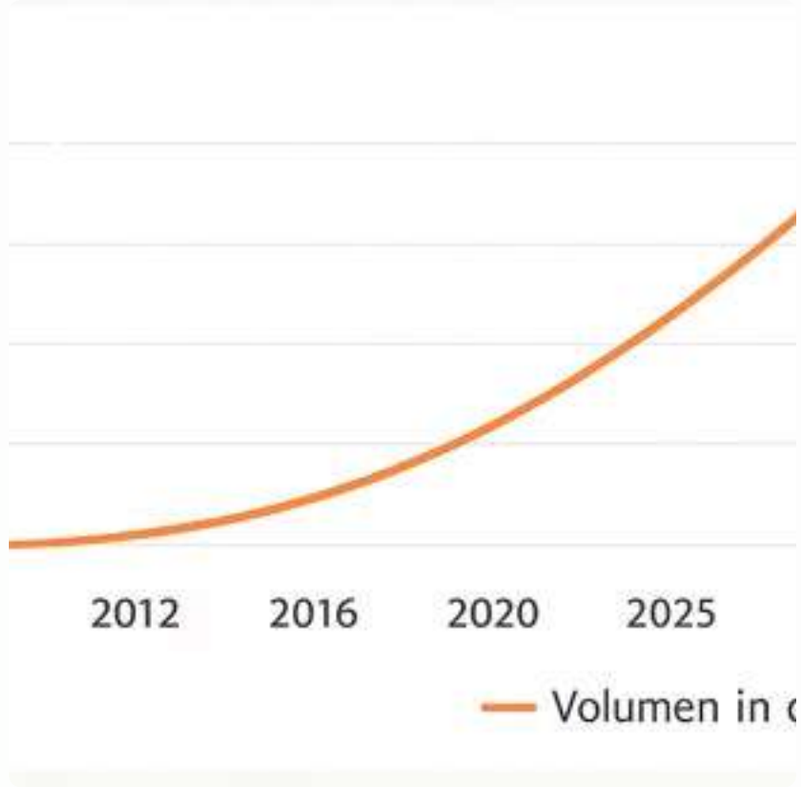
La revolución del *streaming*: flujos de datos en tiempo real desde sensores, apps, logs, IoT.

Big Data



La drástica disminución del costo del procesamiento y del almacenamiento de datos ha cambiado completamente la forma en que las organizaciones capturan y guardan información.

Hoy almacenar grandes volúmenes de datos es más accesible que nunca, lo que ha habilitado nuevos modelos de análisis y gestión de información.



Sin embargo, mientras los costos bajan, el volumen, la velocidad y la complejidad de los datos crecen de manera exponencial.

Esta combinación —más datos, más diversos, generados más rápido— supera las capacidades de los sistemas tradicionales y exige arquitecturas diseñadas específicamente para Big Data.

Cómo Google revolucionó el procesamiento de datos con MapReduce

1

En 2004, dos ingenieros de Google, Jeff Dean y Sanjay Ghemawat, presentaron una idea que cambiaría la forma de procesar enormes cantidades de información: **MapReduce**. ("MapReduce: una infraestructura para el procesamiento paralelo de grandes conjuntos de datos").

2

MapReduce es una forma sencilla de dividir un problema grande en muchos problemas pequeños que pueden resolverse al mismo tiempo.

3

Su funcionamiento se basa en dos pasos:

- **Map:** separar y organizar los datos.
- **Reduce:** juntar los resultados y producir una respuesta final.

4

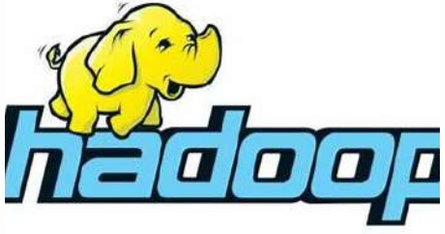
Este enfoque permite que miles de computadoras trabajen a la vez, lo que hace que el procesamiento de datos sea mucho más rápido y eficiente.

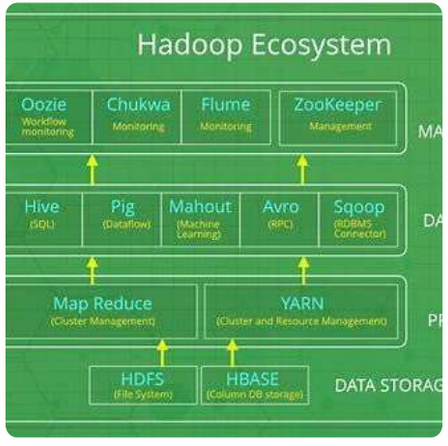
5

Gracias a MapReduce, Google pudo mejorar procesos como la búsqueda web, el análisis de tendencias y muchas tareas relacionadas con Big Data. Hoy su idea sigue siendo la base de tecnologías modernas como Hadoop.

Hadoop: Una herramienta para trabajar con grandes cantidades de datos

El siguiente paso relevante en la historia de Big Data fue el desarrollo de Hadoop, un marco de código abierto para el procesamiento de datos a gran escala. Hadoop fue creado por Doug Cutting y Mike Cafarella en 2005, y se basa en el modelo MapReduce.





Procesamiento de información masiva: Muchas empresas usan Hadoop para manejar grandes volúmenes de datos, como registros de navegación, resultados de búsqueda o información generada en sitios web.

Análisis de datos empresariales

Hadoop permite analizar patrones de tráfico, comportamiento de usuarios y datos de clientes. Esto ayuda a tomar decisiones basadas en información real y detallada.

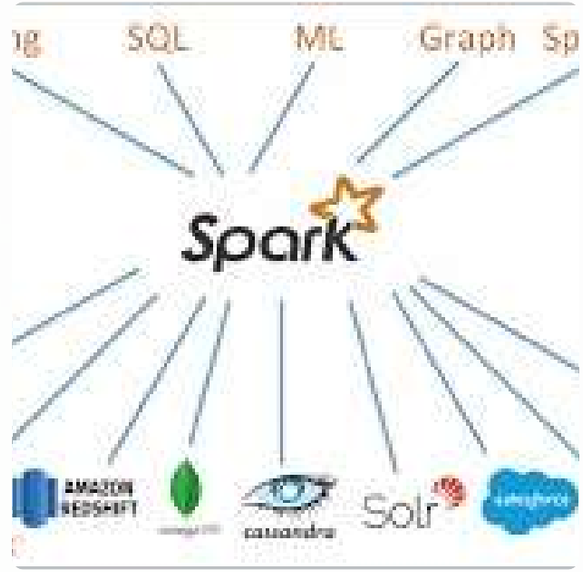
Ciencia de datos y análisis avanzado: Los científicos de datos utilizan Hadoop para estudiar conjuntos enormes de información: encuestas, sensores, transacciones y más. Su capacidad para trabajar en paralelo lo hace ideal para tareas que requieren mucha potencia de procesamiento.

Big Data: Una visión general de las 7 V

- **Volumen:** La cantidad de datos que se genera y almacena crece de forma exponencial día a día.
- **Velocidad:** Los datos se producen y se deben procesar cada vez más rápido, muchas veces en tiempo casi real.
- **Variedad:** Los datos provienen de muchas fuentes distintas: sensores, redes sociales, dispositivos móviles, registros de transacciones, entre otros.
- **Veracidad:** Los datos deben ser confiables y de calidad; de lo contrario, las conclusiones pueden ser erróneas
- **Viabilidad:** Importa que los datos sean realmente útiles para la organización: que valga la pena almacenarlos, procesarlos y analizarlos.
- **Visualización de datos**: Es el proceso de representar los datos de forma gráfica (gráficos, cuadros, tableros) para facilitar su comprensión y la detección de patrones.
- **Valor:** El valor de los datos puede verse en:
  - **Valor tangible:** ahorro de costos, aumento de ingresos, mejora de la eficiencia.
  - **Valor intangible:** mejor toma de decisiones, innovación y creación de nuevos productos o servicios.

Spark: Procesamiento distribuido en memoria, la siguiente evolución del Big Data

Luego de Hadoop, el siguiente gran avance en Big Data fue el desarrollo de **Apache Spark**, un marco de código abierto creado por *Matei Zaharia* y su equipo en la Universidad de Berkeley en 2009.
Spark nació para superar las limitaciones de Hadoop y aprovechar mejor la memoria, permitiendo procesar datos de forma **más rápida, flexible y eficiente**.



Mucho mayor velocidad: Spark puede trabajar almacenando los datos en memoria (RAM), lo que lo hace **hasta 100 veces más rápido** que Hadoop en ciertas tareas.
Ideal para análisis en tiempo real, flujos de datos y aplicaciones interactivas.

Más flexibilidad: Spark no solo sirve para procesamiento por lotes: también soporta

- análisis de streaming,
- machine learning,
- consultas SQL,
- grafos.

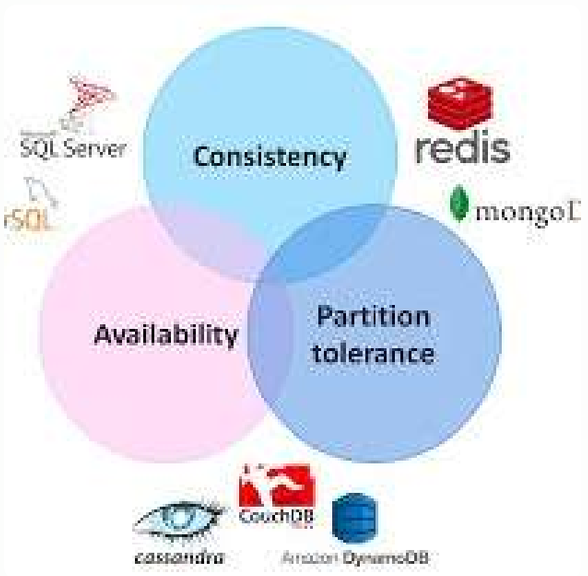
Mejor escalabilidad: Al igual que Hadoop, Spark puede trabajar con miles de nodos, pero lo hace de forma más eficiente.
Es capaz de manejar volúmenes enormes de información y escalar según las necesidades del proyecto.

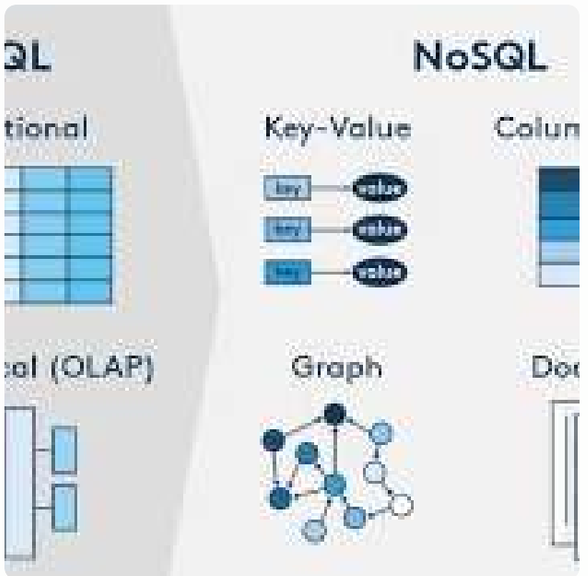
El teorema CAP y las bases de datos NoSQL

El teorema CAP, propuesto por Eric Brewer en el año 2000, afirma que en un sistema distribuido es imposible garantizar **al mismo tiempo** estas tres propiedades:

- **C**onsistencia
- **A**disponibilidad (Availability)
- **P**artición (Partition tolerance)

Un sistema distribuido solo puede ofrecer **dos de ellas a la vez**, pero nunca las tres de forma completa.





Consistencia: Todos los nodos del sistema ven **la misma información al mismo tiempo**. Si un usuario hace un cambio, ese cambio debe reflejarse inmediatamente en todo el sistema.

Disponibilidad: El sistema **siempre responde**, aunque sea con una versión desactualizada de los datos. No importa si un nodo falla: el sistema sigue funcionando para los usuarios

Tolerancia a particiones: El sistema sigue operativo aunque se produzcan **fallos de comunicación** entre los nodos. Es decir, aunque la red se “parta”, el sistema no se cae.

Propiedades de las base de datos no SQL

Escalabilidad (Ventaja): Las bases de datos NoSQL suelen escalar de forma horizontal, añadiendo más nodos al clúster. Esto les permite manejar grandes volúmenes de datos y alto tráfico de manera eficiente.

Flexibilidad (Ventaja): NoSQL permite almacenar datos en múltiples formatos (documentos, grafos, columnas, clave–valor). Esto facilita adaptarse a datos semiestructurados o no estructurados sin necesidad de un esquema rígido.

Alta velocidad y eficiencia (Ventaja): Diseñadas para grandes cargas distribuidas, muchas bases NoSQL ofrecen un rendimiento superior en consultas simples, procesamiento masivo o lectura/escritura a gran escala

Consistencia eventual (Limitación): A diferencia de las bases relacionales, que priorizan consistencia fuerte, muchas bases NoSQL optan por consistencia eventual para aumentar disponibilidad y tolerancia a fallos (según el teorema CAP).

Seguridad variable (Limitación): Históricamente, algunos sistemas NoSQL no tenían opciones avanzadas de seguridad por defecto (autenticación, cifrado, roles).
Esto ha mejorado mucho, pero aún puede ser inferior a la madurez de las bases relacionales tradicionales.

Menor madurez y estandarización (Limitación)

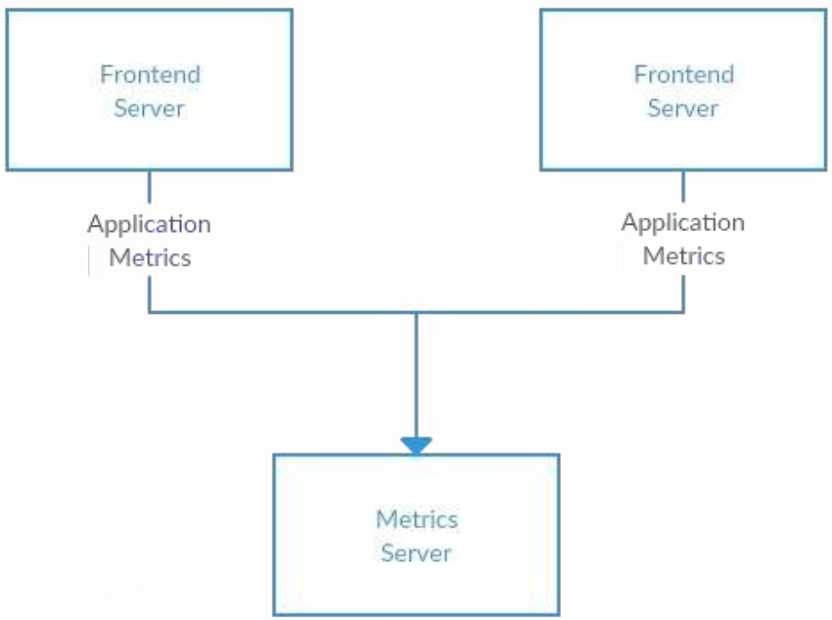
Las bases NoSQL son más recientes que las bases SQL, por lo que

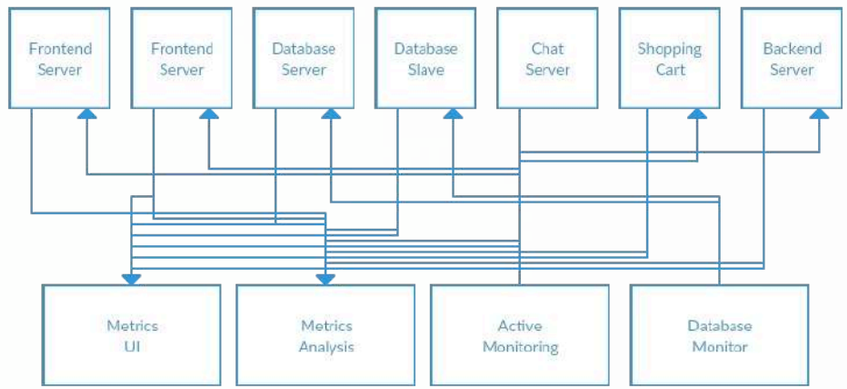
- algunas tienen menor documentación,
- menor estandarización,
- diferencias importantes entre proveedores.

Kafka y la estructura publicador-suscriptor

Con la aparición de nuevas fuentes de información —redes sociales, sensores, dispositivos móviles, sistemas IoT, logs de aplicaciones— la **cantidad y la velocidad** de los datos crecieron de forma exponencial.
El modelo tradicional **cliente-servidor**, basado en peticiones puntuales de un cliente a un servidor, funciona bien para interacciones *uno a uno*, pero:

- obliga a que el cliente conozca directamente al servidor
- no soporta bien muchos productores y muchos consumidores a la vez
- se vuelve difícil de escalar cuando los datos llegan de forma continua y en tiempo real



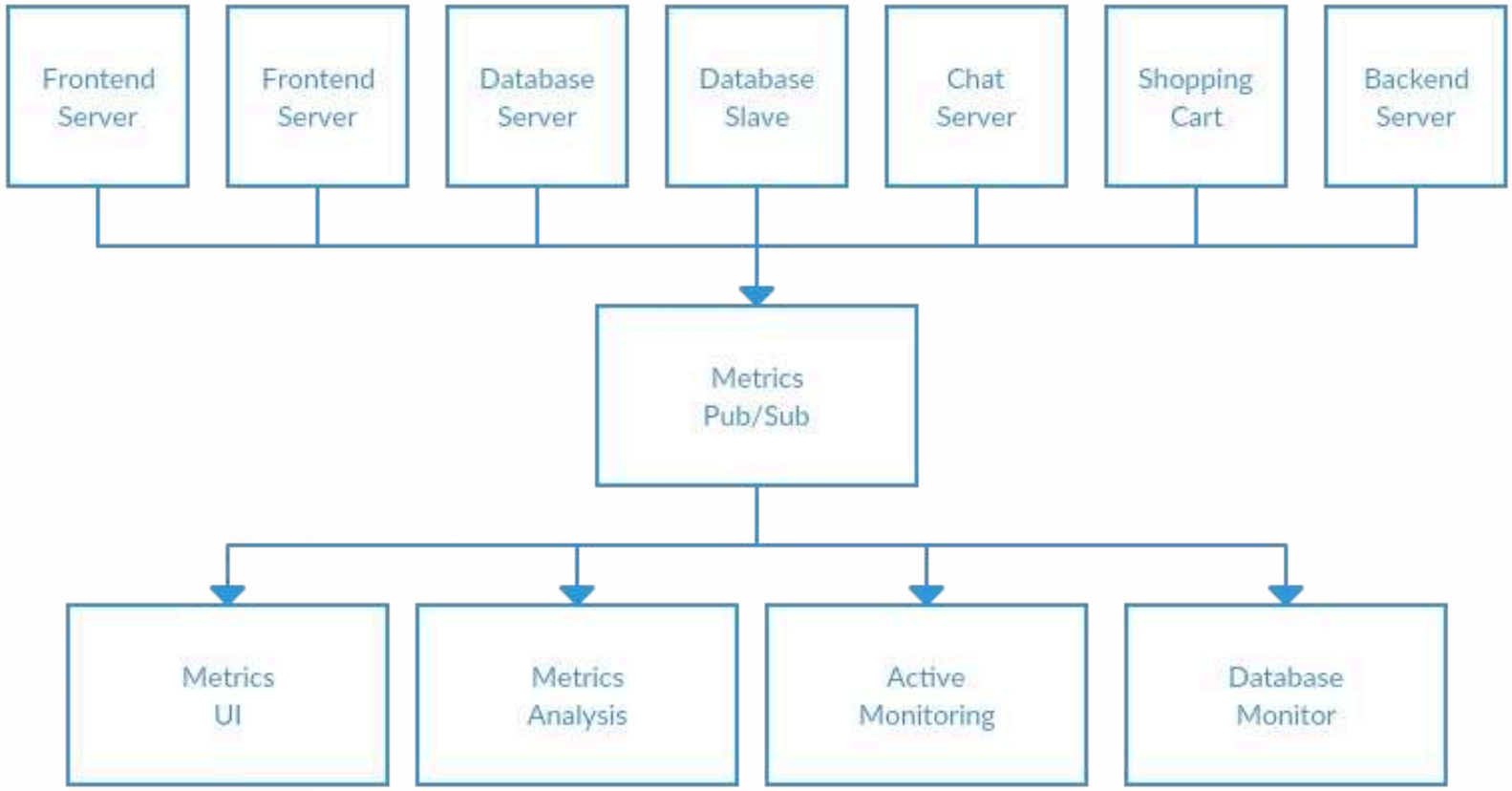


Kafka y la estructura publicador-suscriptor

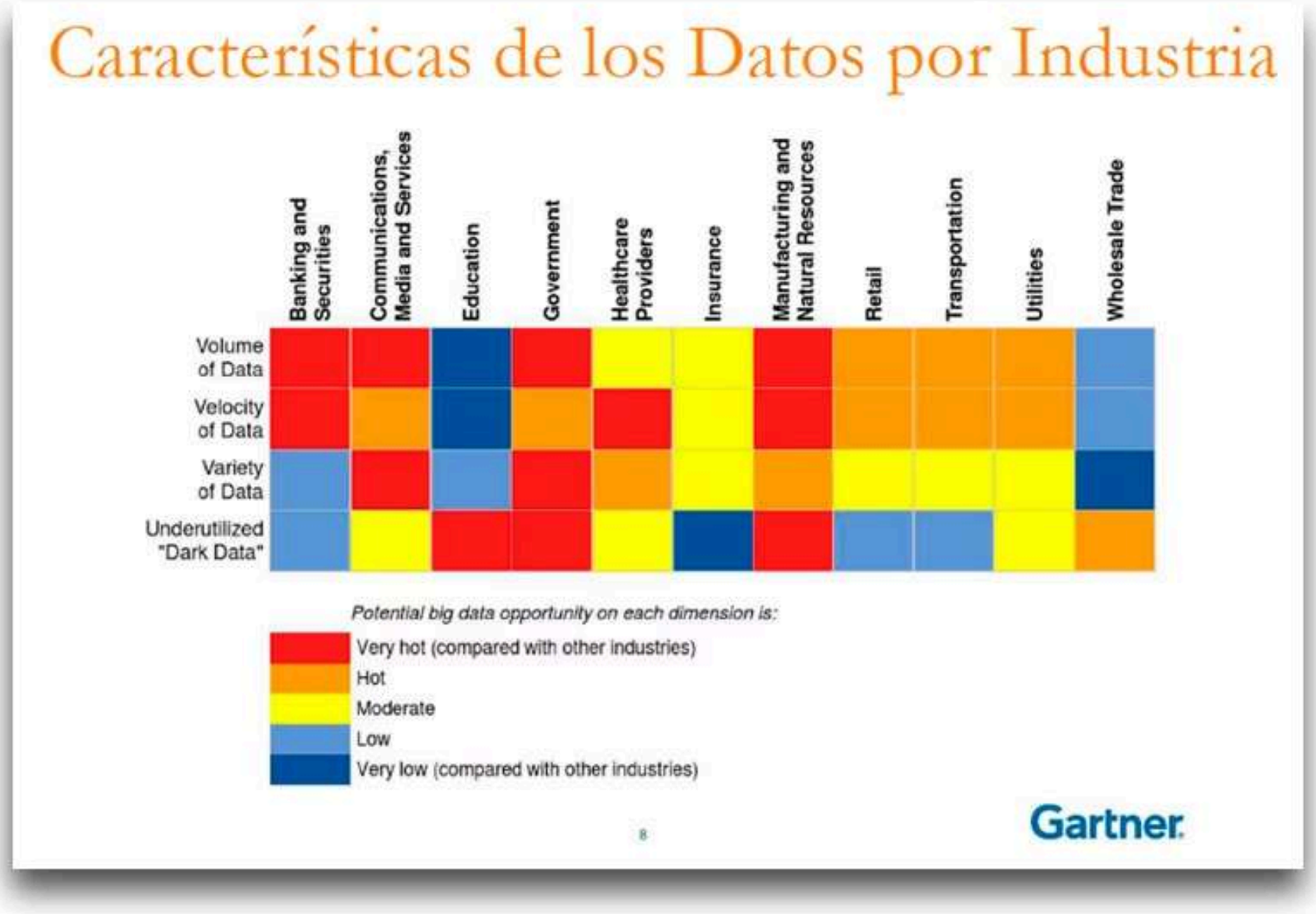
Para responder a este nuevo escenario surge el modelo **publicador–suscriptor (pub/sub)**, en el que:

- los **productores** publican mensajes (eventos) en *topics*
- los **consumidores** se suscriben a esos topics y reciben los datos cuando se generan
- productores y consumidores quedan **desacoplados**, lo que permite escalar y procesar flujos de datos de alta velocidad

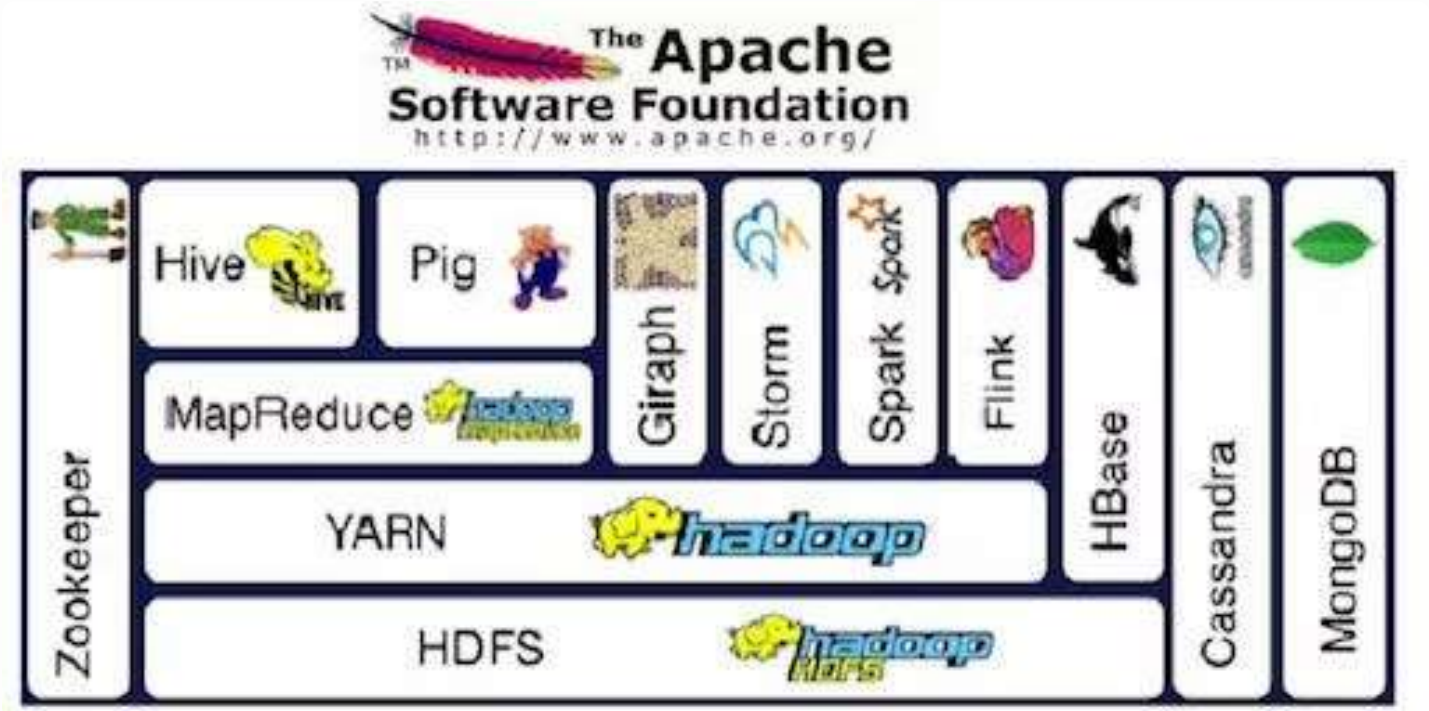
Apache Kafka implementa este modelo de forma distribuida y tolerante a fallos, convirtiéndose en una pieza clave para manejar datos en streaming y arquitecturas modernas de Big Data.



La implementacion en la industria



Arquitectura tipica de Hadoop



Big Data: El valor en los negocios

Big Data se ha convertido en un activo estratégico para las empresas, ya que permite transformar grandes volúmenes de información en decisiones, eficiencia y nuevas oportunidades.

- Mejor toma de decisiones: El análisis de grandes cantidades de datos permite obtener una visión más precisa y completa del negocio, lo que ayuda a tomar decisiones más acertadas y basadas en evidencia.
- Reducción de costos: Big Data ayuda a identificar ineficiencias, procesos lentos o recursos mal utilizados. Esto permite optimizar operaciones y reducir costos operativos.
- Innovación: nuevos productos y servicios. Al analizar patrones y comportamientos, las empresas pueden descubrir necesidades emergentes y crear soluciones, productos y servicios que generen ventaja competitiva.
- Mejor experiencia del cliente: Con un análisis profundo del comportamiento del cliente, las empresas pueden ofrecer servicios más personalizados, mejorar la atención y aumentar la fidelidad.

Dónde entra la Ingeniería de Datos… y cómo se conecta con la Ciencia de Datos?

🔧 Ingeniería de Datos: La base que hace posible Big Data

Para que Big Data genere valor real, las organizaciones necesitan **Ingeniería de Datos**, responsable de:

- Construir las **infraestructuras** que reciben, almacenan y procesan grandes volúmenes de datos.
- Garantizar la **calidad, seguridad y accesibilidad** de los datos.
- Preparar y organizar los datos para que puedan ser usados por analistas y científicos de datos.

Sin Ingeniería de Datos, Big Data no puede transformarse en valor.

🔬 Ciencia de Datos: Transformando datos en inteligencia

La **Ciencia de Datos** toma los datos que la Ingeniería de Datos prepara y los convierte en:

- Predicciones
- Modelos estadísticos
- Inteligencia de negocio
- Insights accionables
- Automatización y *machine learning*