Sistemas de almacenamiento y procesado distribuido

Tema 1. Introducción

© 2023 Javier Esparza Peidro - jesparza@dsic.upv.es

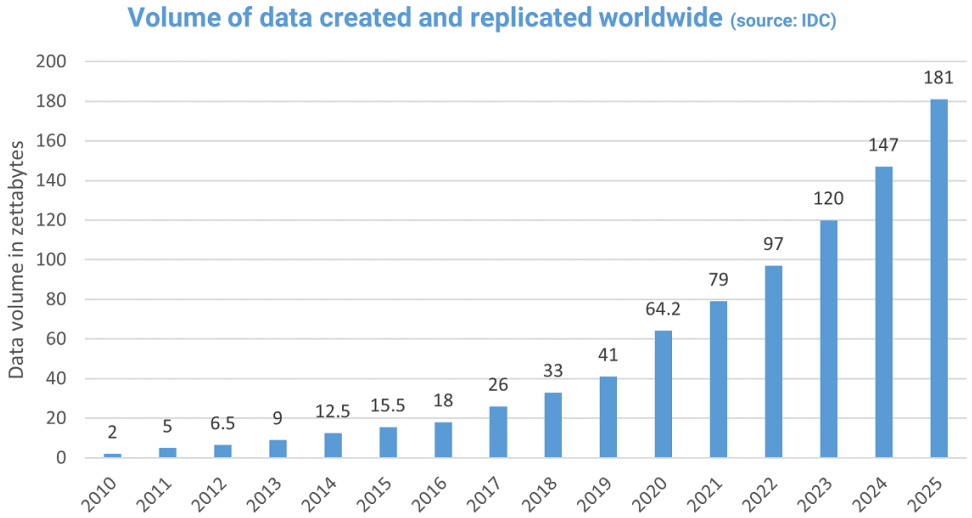
Contenido

- Introducción
- Sistemas de datos
- Ingeniería de datos
- Arquitecturas data-intensive

Introducción

Muchos datos ...

- Hoy en día todo se registra: se generan volúmenes de datos estremecedores
  - En 2020 cada usuario generó > 1.7Mb/s
  - En 2025 cada día se generará 463 exabytes (10^3 pb)
  - En 2025 175 billones de Gb

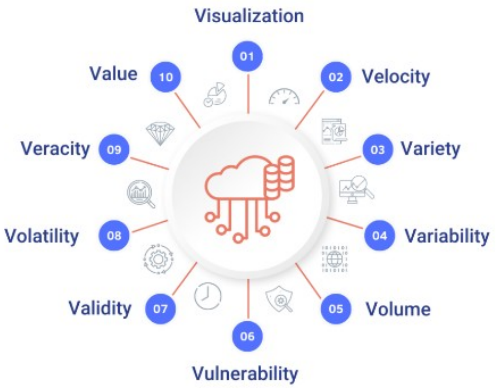


https://www.statista.com/statistics/871513/worldwide-data-created/

Introducción

Big Data

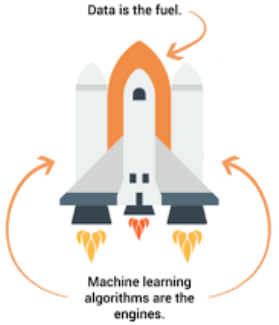
- Cualquier fuente de datos con 3 características (3 Vs)
  - Volumen: imposibe almacenar en una máquina
  - Velocidad: ratio a la que se genera nueva info
  - Variedad: tipos de datos (no/semi)estructurados
- + 10 Vs:
  - Veracidad, variabilidad, valor, validez, vulnerabilidad, volatilidad, visualización



Introducción

La importancia de los datos ...

- Los datos se han convertido en la gasolina de otras técnicas, como Machine Learning



- Cada vez más, las organizaciones adoptan estrategias dirigidas por datos (data-driven)

Creating a Data-Driven Organization. Carl Anderson. O'Reilly. 2015.

Introducción

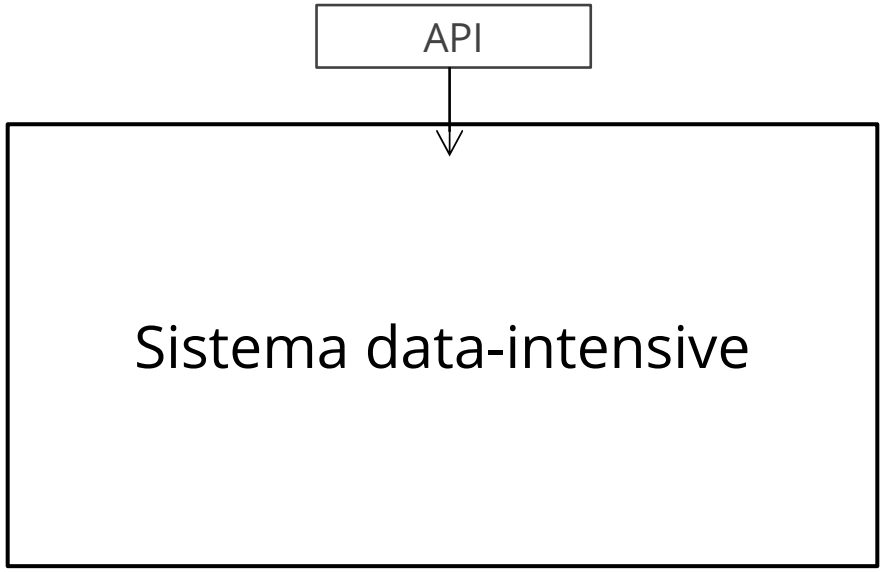
Nuevos métodos ...

- Los sistemas y técnicas de gestión de datos tradicionales no son adecuadas
- La gestión de datos se ha convertido en un problema prioritario
- Es necesario diseñar sistemas, técnicas y herramientas capaces de gestionar la información de manera eficiente
- En general, nos referimos a este tipo de sistemas como sistemas data-intensive

Sistemas de datos

¿Qué es un sistema data-intensive?

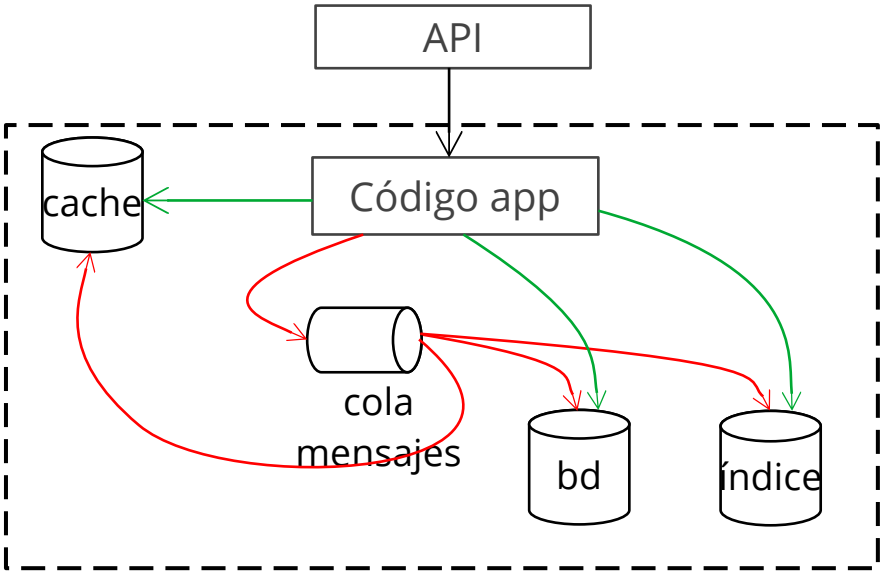
- Sus retos tienen que ver con el volumen de datos, la complejidad de los datos, la velocidad a la que se generan/cambian los datos
- Son sistemas complejos, se componen de piezas más pequeñas, que denominamos sistemas de datos



Sistemas de datos

¿Qué es un sistema data-intensive?

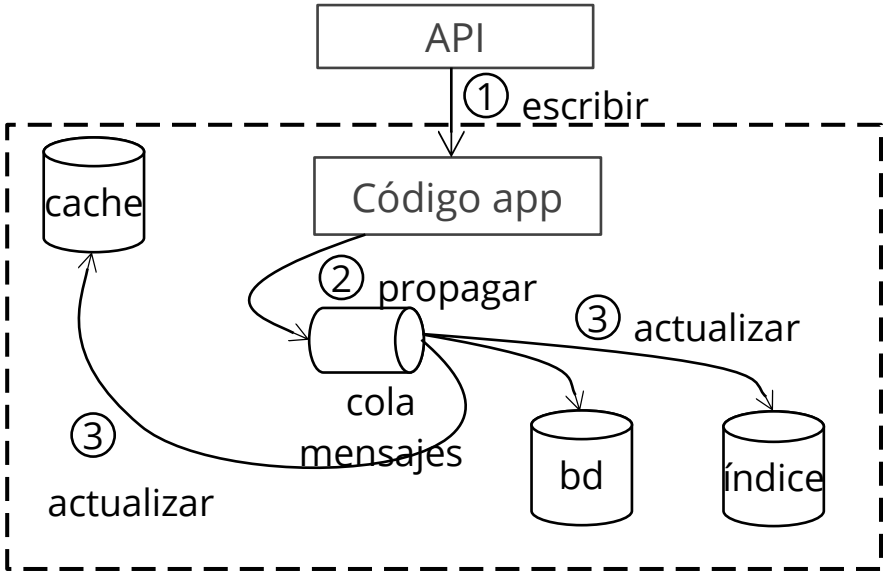
- Sus retos tienen que ver con el volumen de datos, la complejidad de los datos, la velocidad a la que se generan/cambian los datos
- Son sistemas complejos, se componen de piezas más pequeñas, que denominamos sistemas de datos



Sistemas de datos

¿Qué es un sistema data-intensive?

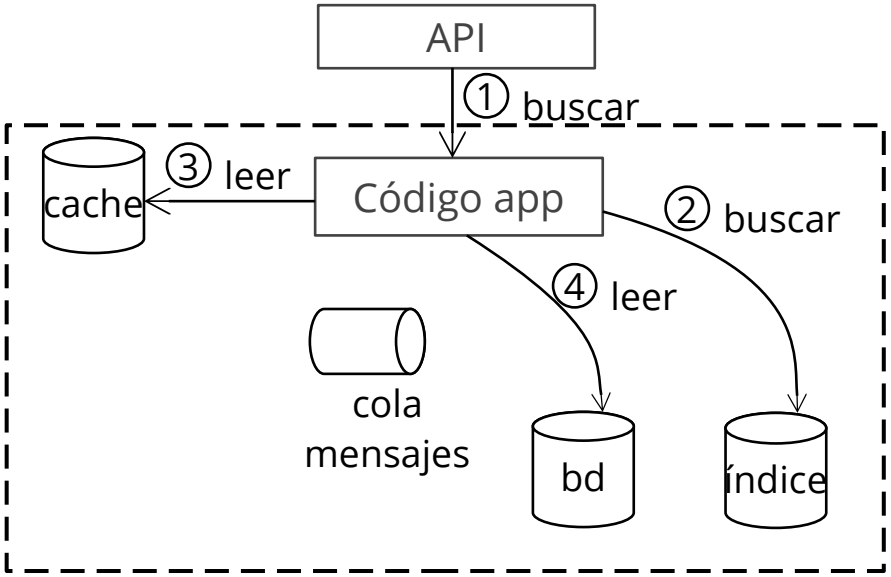
- Sus retos tienen que ver con el volumen de datos, la complejidad de los datos, la velocidad a la que se generan/cambian los datos
- Son sistemas complejos, se componen de piezas más pequeñas, que denominamos sistemas de datos



Sistemas de datos

¿Qué es un sistema data-intensive?

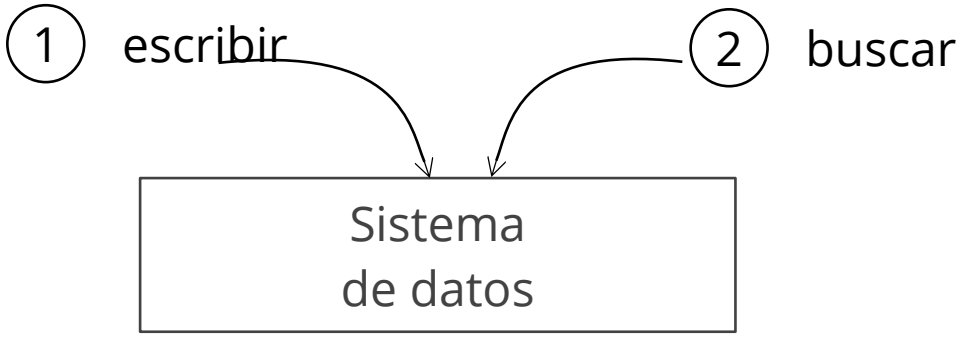
- Sus retos tienen que ver con el volumen de datos, la complejidad de los datos, la velocidad a la que se generan/cambian los datos
- Son sistemas complejos, se componen de piezas más pequeñas, que denominamos sistemas de datos



Sistemas de datos

¿Qué es un sistema de datos?

- Es un sistema con dos funcionalidades básicas:
  1. Almacenar los datos (escribir)
  2. Responder a preguntas sobre los datos (leer/buscar)



- Dependiendo de su propósito, se optimizarán las lecturas, escrituras, o ambas (compromiso)

Sistemas de datos

¿Qué es un sistema de datos?

- Cada sistema data-intensive posee unos requerimientos diferentes
- Hay que identificar estos requerimientos pronto y seleccionar los componentes (sistemas de datos) adecuados
- Para ello, es esencial comprender cómo funciona cada sistema de datos: cómo organiza la información internamente, sus propiedades, ventajas e inconvenientes, etc.

Sistemas de datos

Propiedades

- En un sistema de datos es recomendable maximizar las siguientes propiedades:
  - Fiabilidad: el sistema funciona correctamente incluso si hay fallos
  - Escalabilidad: el sistema puede manejar cargas de trabajo crecientes
  - Mantenibilidad: el sistema es fácil de mantener y de cambiar

Sistemas de datos

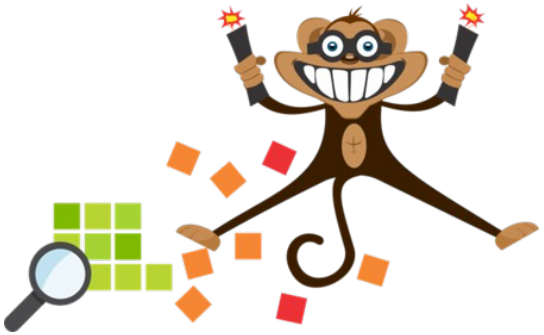
Propiedades > Fiabilidad

- El sistema funciona correctamente incluso si hay fallos
  - El sistema proporciona las funcionalidades esperadas
  - Con el rendimiento deseado
  - Lo hace de manera segura
- El sistema debe ser tolerante a fallos o resiliente
  - Fallos hardware
  - Fallos software
  - Fallos humanos

Sistemas de datos

Propiedades > Fiabilidad

- El fallo ya no es una situación anómala
- El sistema debe normalizar la ocurrencia de fallos
- Ejemplo: Netflix Chaos Monkey



Sistemas de datos

Propiedades > Escalabilidad

- Habilidad del sistema de mantener su rendimiento ante cargas crecientes de trabajo, añadiendo recursos
- Parámetros de carga: peticiones/seg, tasa de lecturas/escrituras, núm sesiones concurrentes, etc.
- Medidas de rendimiento: productividad, tiempo de respuesta, etc.
- Escalado vertical (scale-up) vs horizontal (scale-out)
- Escalado manual vs automático: sistemas elásticos

Sistemas de datos

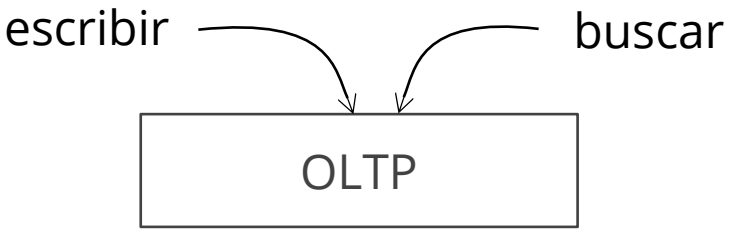
Propiedades > Mantenibilidad

- Sistemas fáciles de mantener y de cambiar
- Limitar al máximo los errores humanos
- Factores que afectan a la mantenibilidad:
  - Operabilidad: facilitar las operaciones del día a día
  - Simplicidad: fácil razonar sobre el sistema. Es recomendable utilizar buenas abstracciones.
  - Evolucionabilidad: facilitar cambios futuros. Es más fácil si el sistema es simple.

Sistemas de datos

OLTP vs OLAP

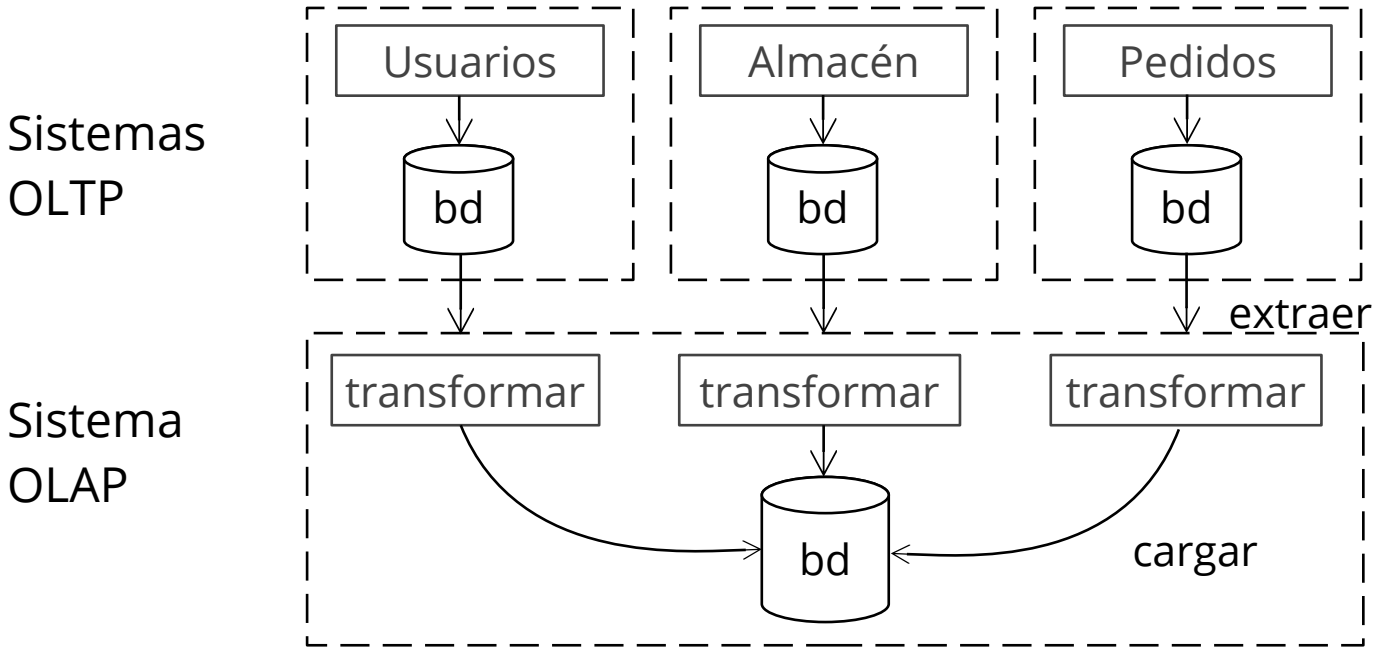
- Online Transaction Processing (OLTP)
- Sistemas interactivos
- Efectúan operaciones “cortas” de lectura/escritura: transacciones
- Volumen de datos “limitado”
- Resultados inmediatos, alta productividad
- Se utilizan índices sobre claves para acceder a los datos



Sistemas de datos

OLTP vs OLAP

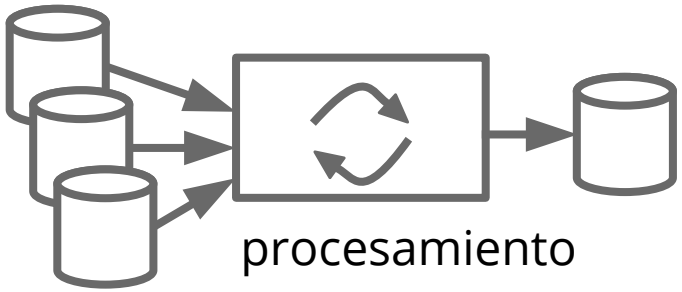
- Online Analytic Processing (OLAP)
- Análisis de datos sobre grandes volúmenes de info
- Obtienen los datos de otros sistemas OLTP (ETL)
- Es habitual que se procesen en diferido (batch)



Sistemas de datos

Sistemas de registro vs sistemas de datos derivados

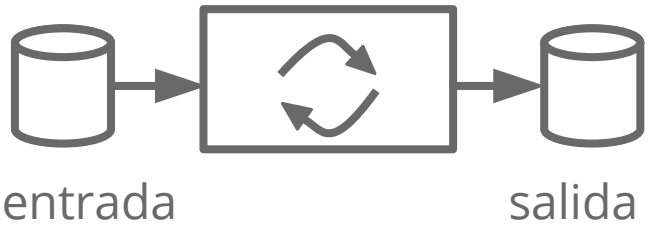
- Sistemas de registro
  - Fuente de verdad
  - Información primaria del sistema
  - Un fallo implica la pérdida de datos
- Sistemas de datos derivados
  - Información derivada de otras fuentes
  - Mecanismos de procesamiento de datos



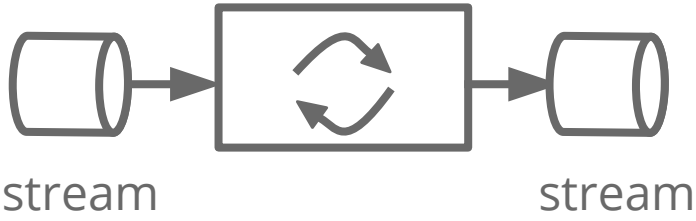
Sistemas de datos

Procesamiento de datos

- Procesamiento en batch
  - Procesa volumen grande de datos
  - Requiere mucho tiempo
  - Se ejecutan periódicamente



- Procesamiento en streaming
  - Procesa eventos dinámicamente
  - Baja latencia
  - Tiempo real



- Las arquitecturas modernas usan una combinación

Ingeniería de datos

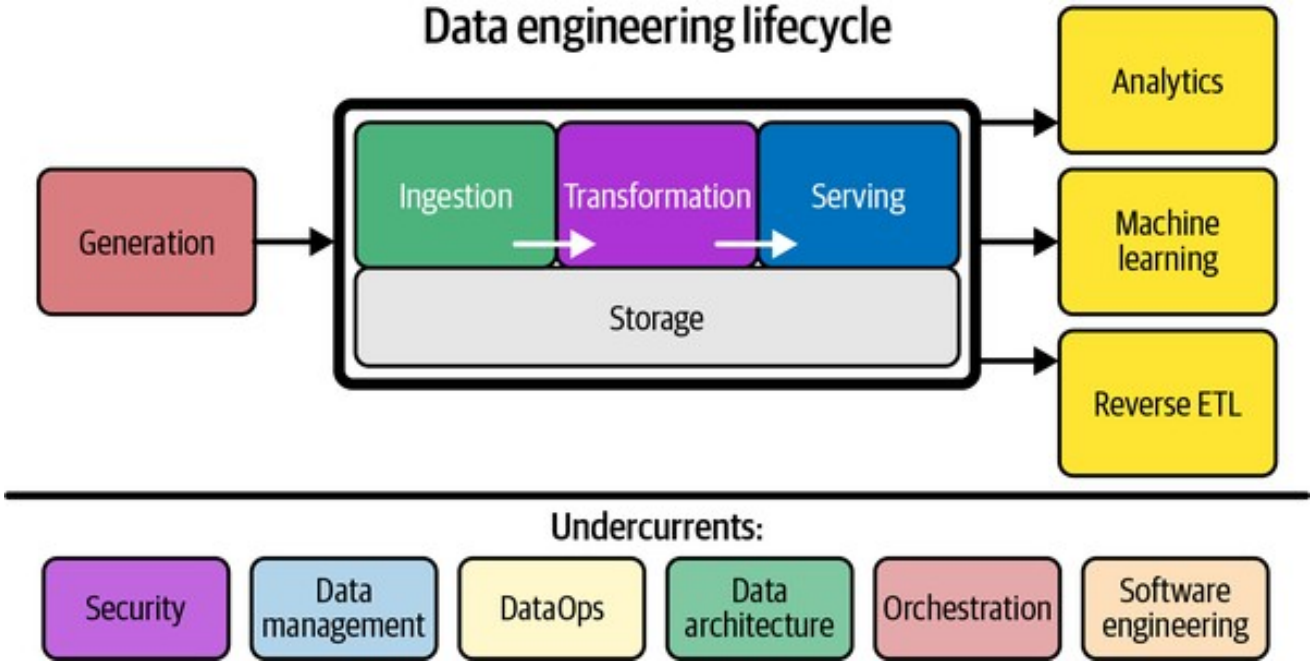
¿Qué es?

- Diseño, implementación y mantenimiento de sistemas y procesos que recogen, almacenan y procesan datos
- Incluye todas las actividades necesarias para generar información de calidad a partir de datos en crudo

Ingeniería de datos

Ciclo de vida

- Etapas: ingestión, transformación, servicio de datos, almacenamiento

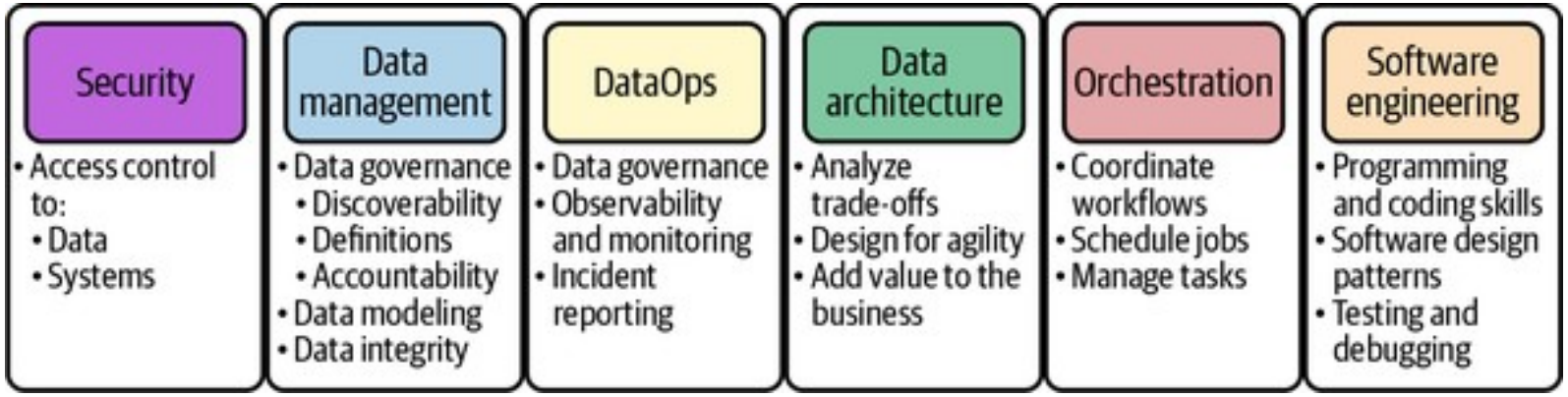


Fundamentals of Data Engineering. Joe Reis & Matt Housley. O'Reilly. 2022

Ingeniería de datos

Ciclo de vida

- Habilidades requeridas: seguridad, gestión de datos, DataOps, arquitectura de datos, orquestación, SE

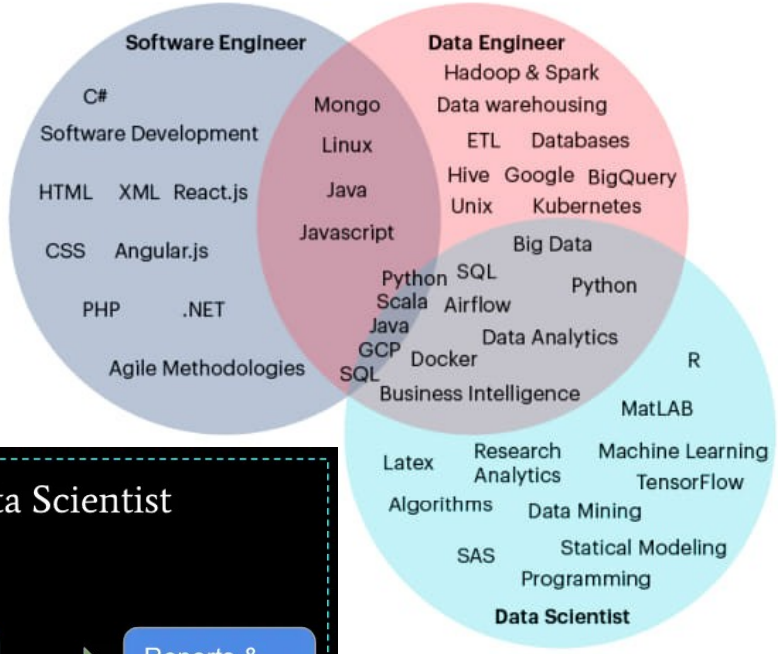


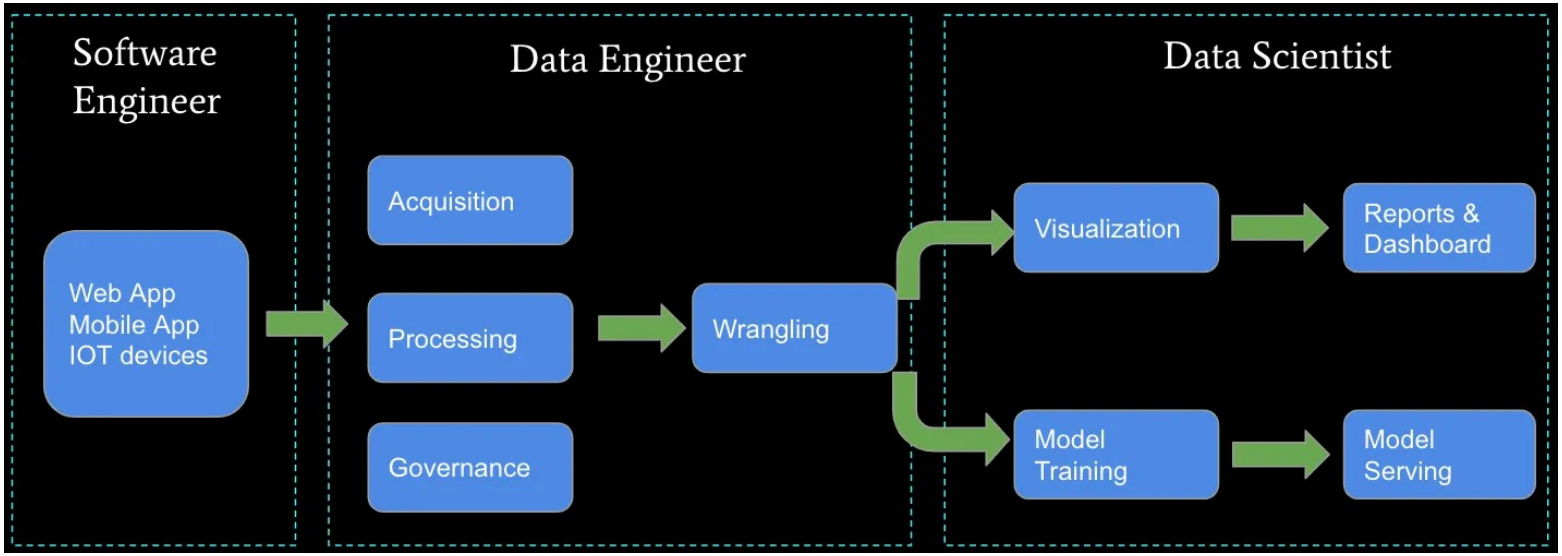
Fundamentals of Data Engineering. Joe Reis & Matt Housley. O'Reilly. 2022

Ingeniería de datos

Fronteras entre perfiles

- Ingeniero de software vs Ingeniero de datos vs Científico de datos



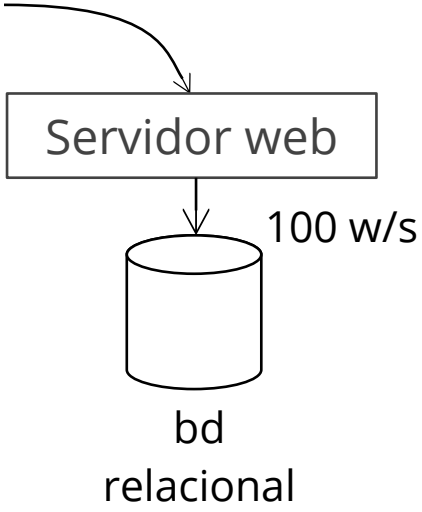


Arquitecturas data-intensive

- Las arquitecturas tradicionales pueden no resultar adecuadas

Arquitecturas data-intensive

- Las arquitecturas tradicionales pueden no resultar adecuadas
- Ejemplo: conteo de visitas a URL con SGBDR

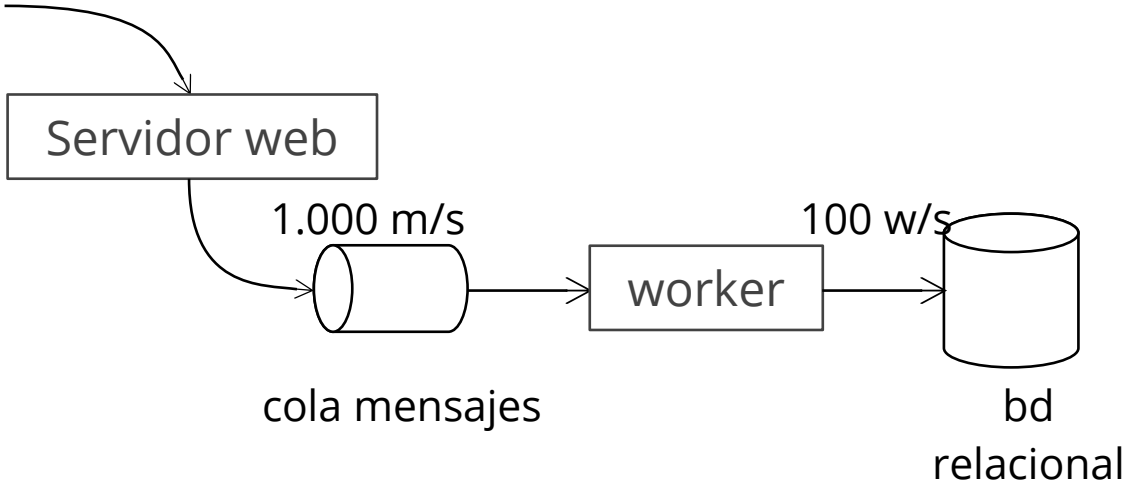


Nombre columna	Tipo
id	int
user_id	int
url	varchar(255)
count	int

- Si la carga de trabajo crece, la bd relacional se satura

Arquitecturas data-intensive

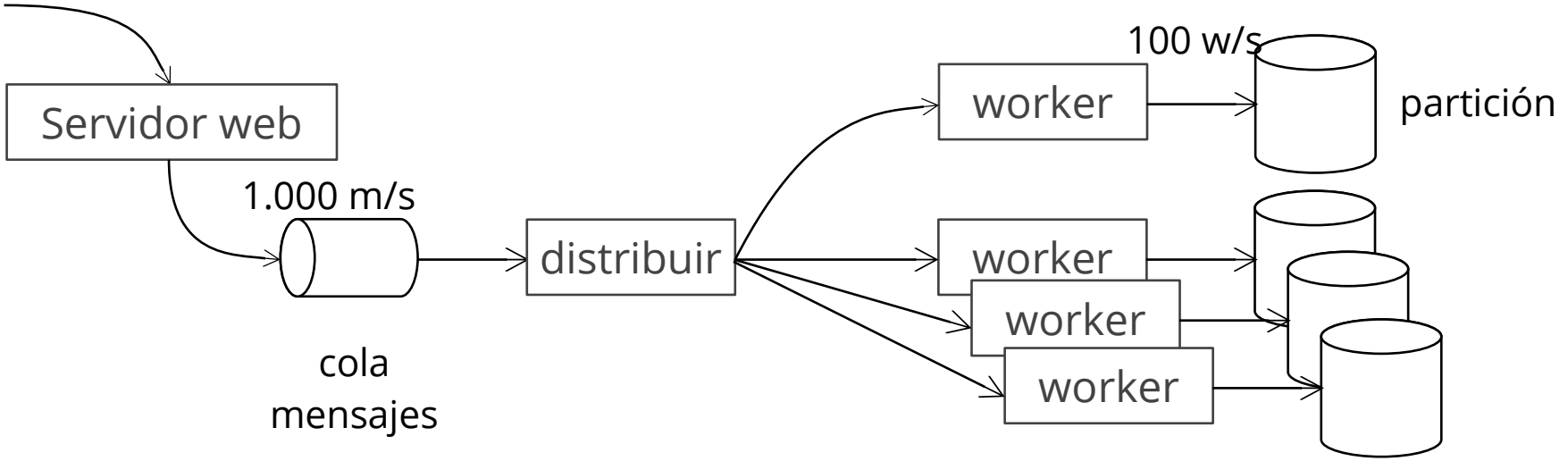
- Las arquitecturas tradicionales pueden no resultar adecuadas
- Ejemplo: conteo de visitas a URL con cola



- Si la carga de trabajo crece, la cola puede quedarse sin memoria

Arquitecturas data-intensive

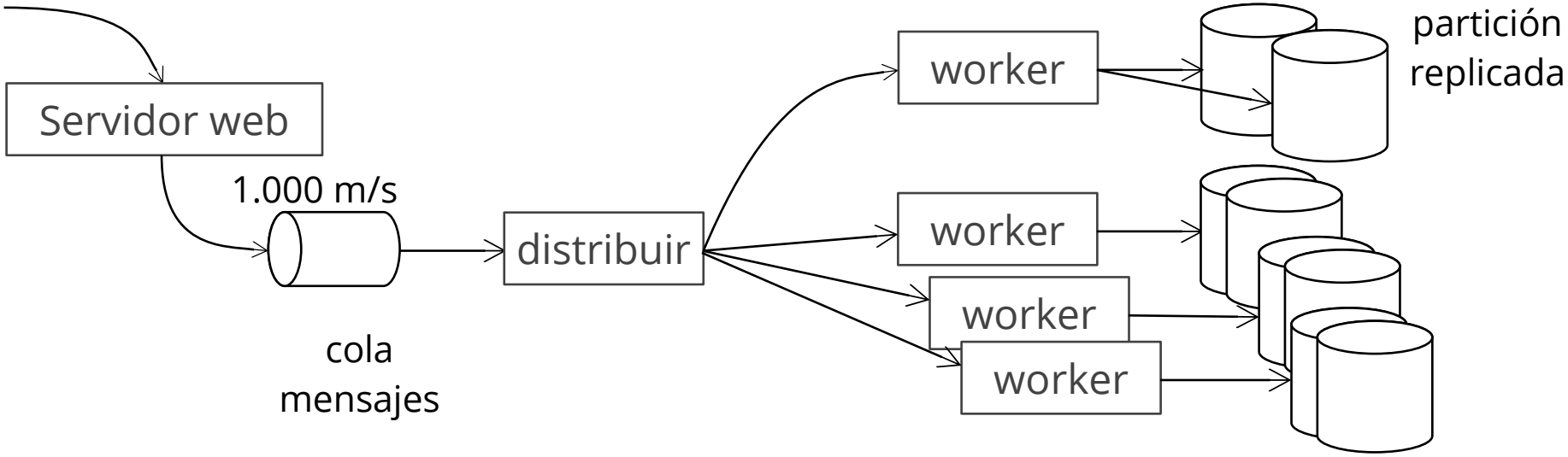
- Las arquitecturas tradicionales pueden no resultar adecuadas
- Ejemplo: conteo de visitas a URL con particiones



- Si la una partición falla, perdemos la información

Arquitecturas data-intensive

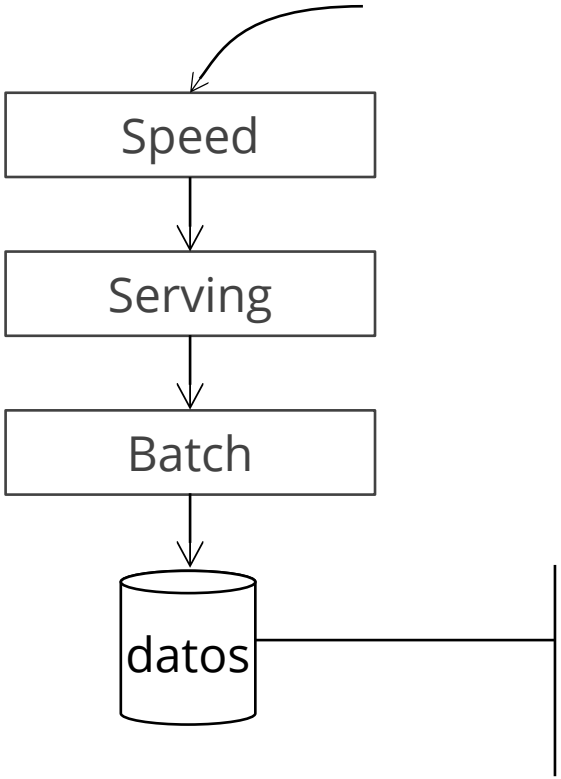
- Las arquitecturas tradicionales pueden no resultar adecuadas
- Ejemplo: conteo de visitas a URL con réplicas



- Wtf??

Arquitecturas data-intensive

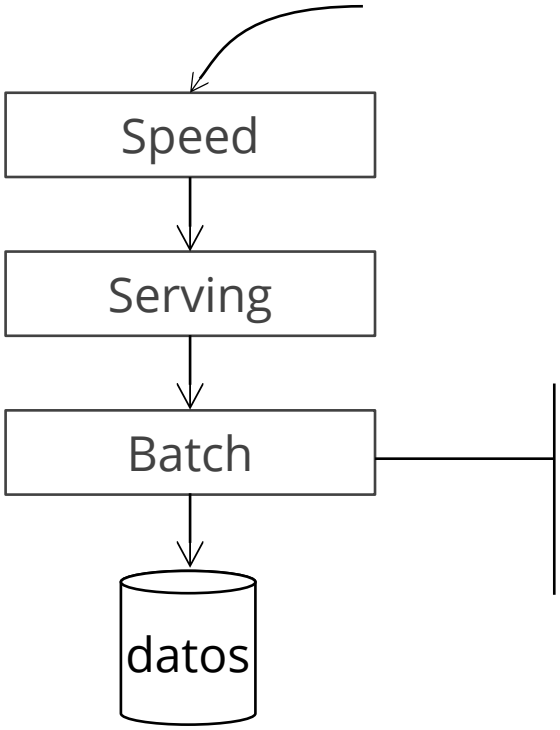
- Existen otras soluciones específicas que gozan de fiabilidad, escalabilidad y mantenibilidad.
- La arquitectura lambda



- Fuente de datos inmutable
- Log de solo añadir
- Baja latencia, consistencia de datos, etc.

Arquitecturas data-intensive

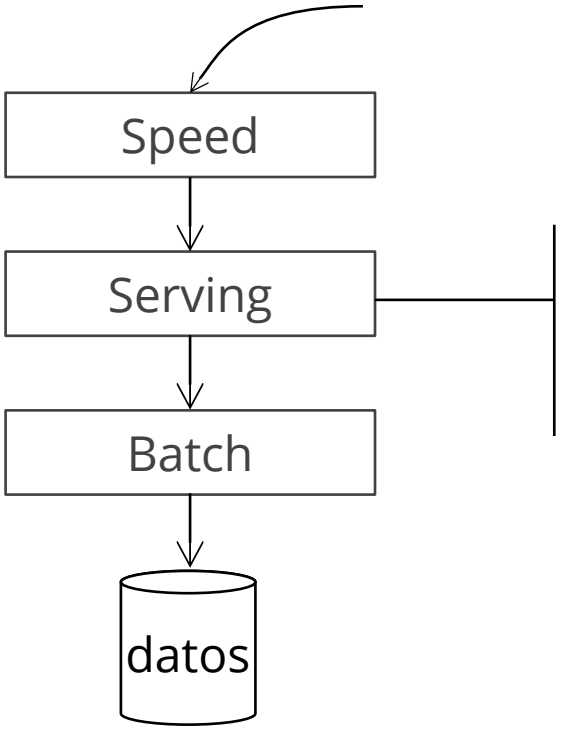
- Existen otras soluciones específicas que gozan de fiabilidad, escalabilidad y mantenibilidad.
- La arquitectura lambda



- Construye vistas derivadas
- Proceso muy costoso
- Las refresca de manera periódica

Arquitecturas data-intensive

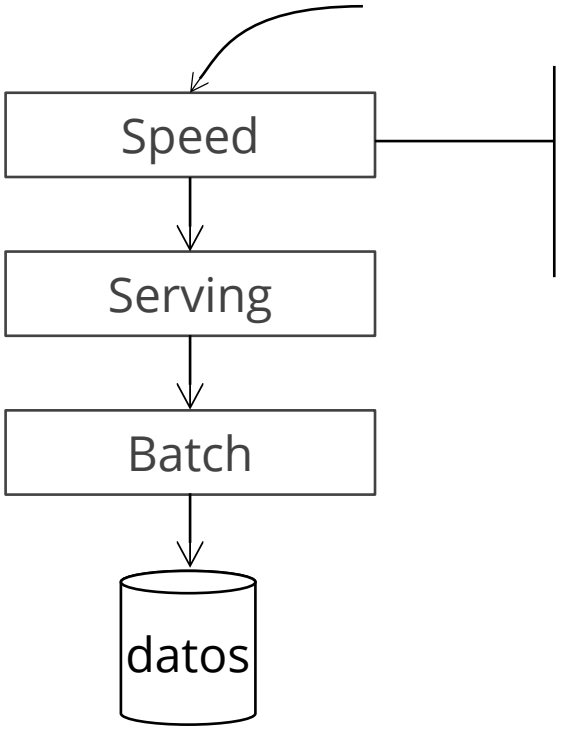
- Existen otras soluciones específicas que gozan de fiabilidad, escalabilidad y mantenibilidad.
- La arquitectura lambda



- Indexa las vistas batch
- El acceso a los datos se produce con baja latencia

Arquitecturas data-intensive

- Existen otras soluciones específicas que gozan de fiabilidad, escalabilidad y mantenibilidad.
- La arquitectura lambda

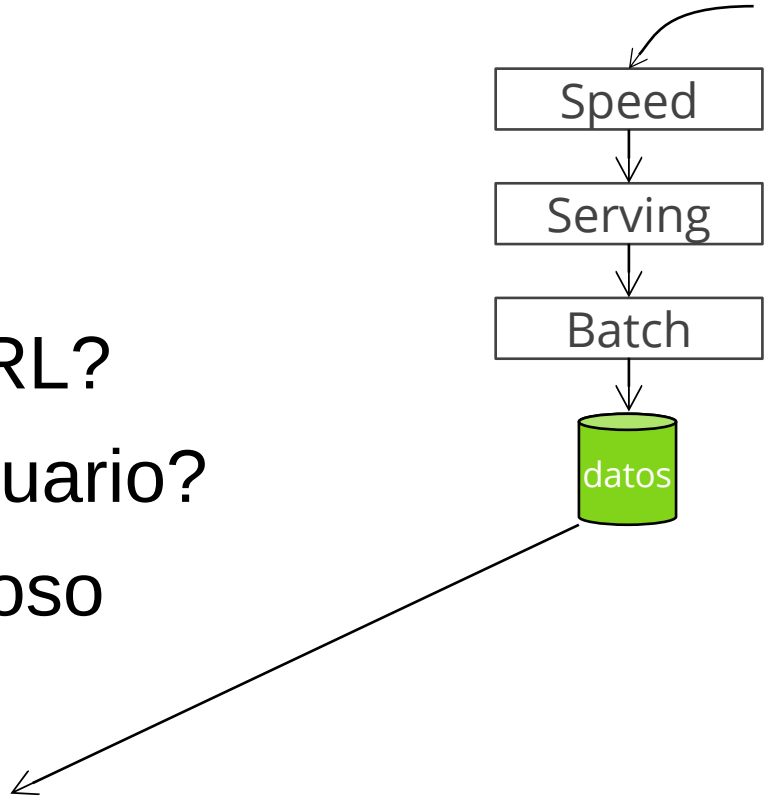


- Complementa las vistas batch
- Proporciona los datos más acualizados con baja latencia
- Sin necesidad de refrescar las vistas batch

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Fuente de datos inmutable
  - Se registra cada visita
  - Almacén de sólo añadir
  - ¿Número de visitas a una URL?
  - ¿Número de visitas de un usuario?
  - Calcular estos datos es costoso

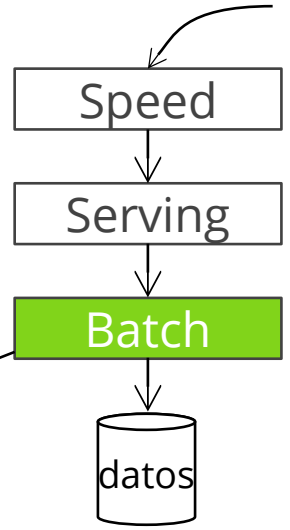


id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Capa batch
  - Vistas derivadas de los datos inmutables
  - Visitas por URL, Visitas por usuario
  - Proceso en batch costoso, se refresca



url	count
www.google.com	2
www.upv.es	1

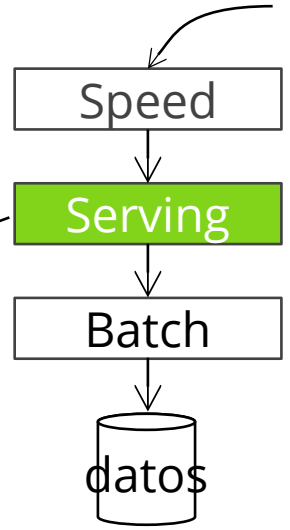
user_id	count
1	2
2	1

id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Capa serving
  - Se indexan las vistas y se publican
  - Baja latencia al acceder a datos



Serving

url	count
www.google.com	2
www.upv.es	1

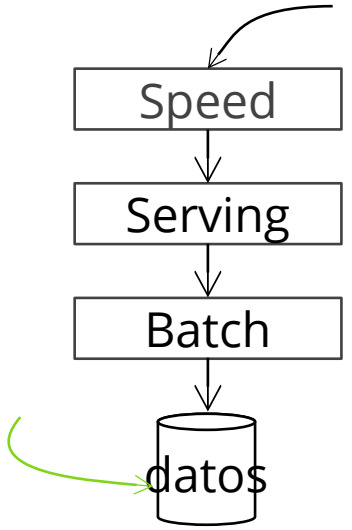
user_id	count
1	2
2	1

id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Actualizaciones??
  - Se añade una nueva entrada



Serving

url	count
www.google.com	2
www.upv.es	1

user_id	count
1	2
2	1

id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

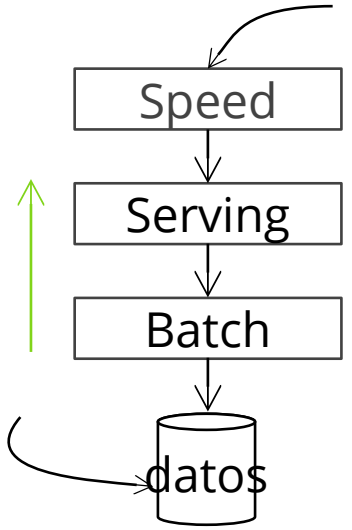
2 www.upv.es

4 2 www.upv.es

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Actualizaciones??
  - Se añade una nueva entrada
  - Debe propagarse por todas las capas



Serving

url	count
www.google.com	2
www.upv.es	2

user_id	count
1	2
2	2

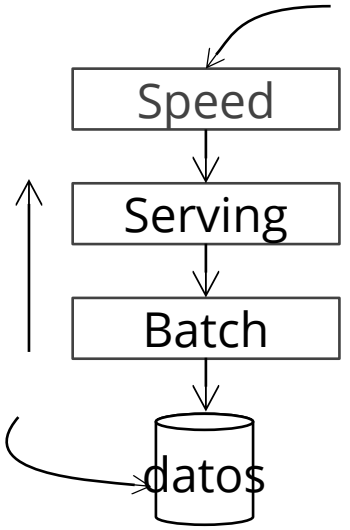
id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

4	2	www.upv.es

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Actualizaciones??
  - Alta latencia en las actualizaciones



Serving

url	count
www.google.com	2
www.upv.es	2

user_id	count
1	2
2	2

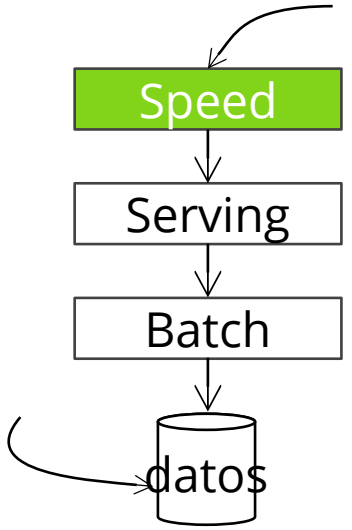
id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

4 2 www.upv.es

Arquitecturas data-intensive

Ejemplo: contador de visitas

- Capa speed
  - Recibe los nuevos datos en streaming
  - Complementa las vistas batch



Serving

url	count
www.google.com	2
www.upv.es	2

user_id	count
1	2
2	2

id	user_id	url
1	1	www.google.com
2	1	www.upv.es
3	2	www.google.com

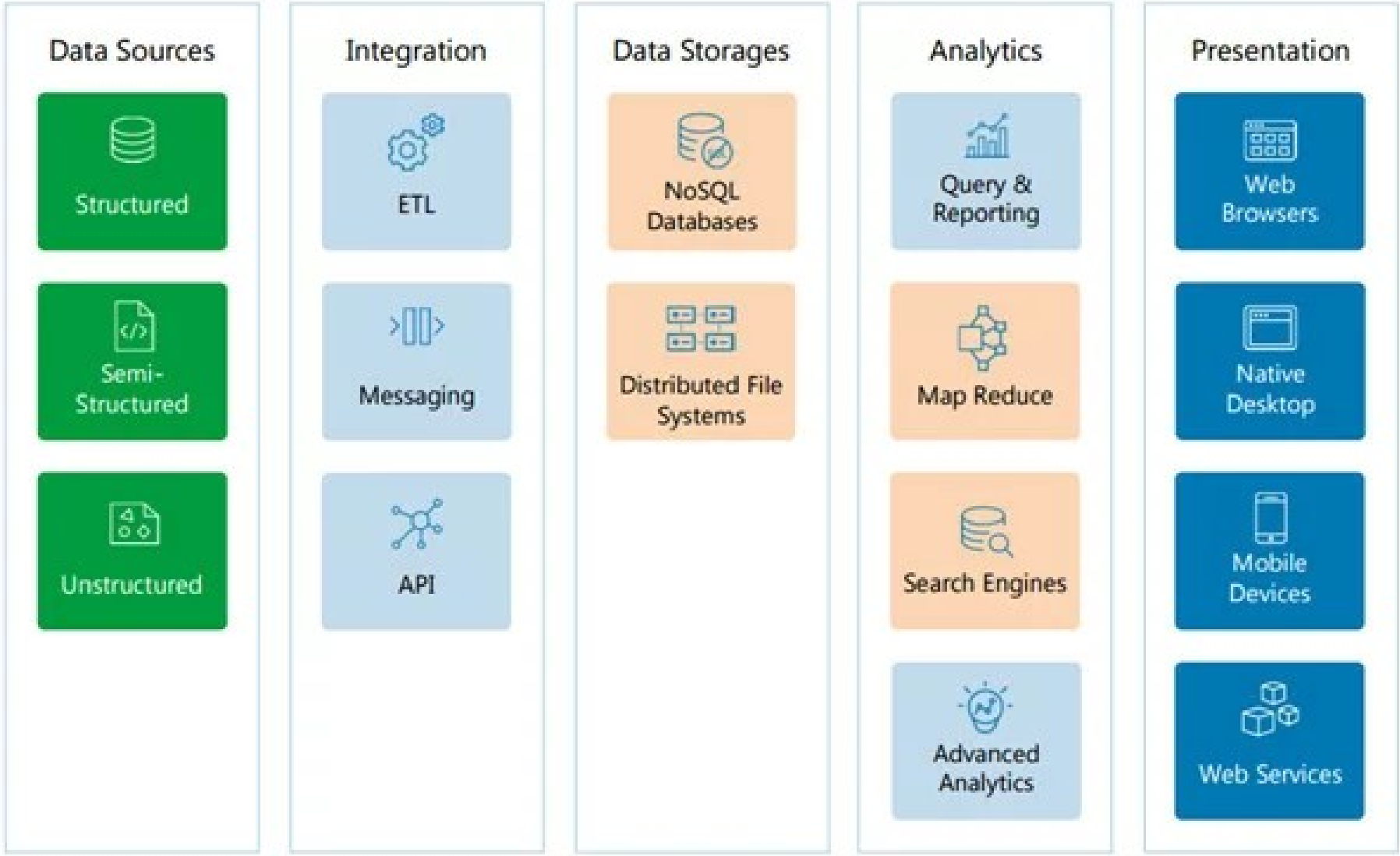
Speed

2 www.upv.es

4 2 www.upv.es

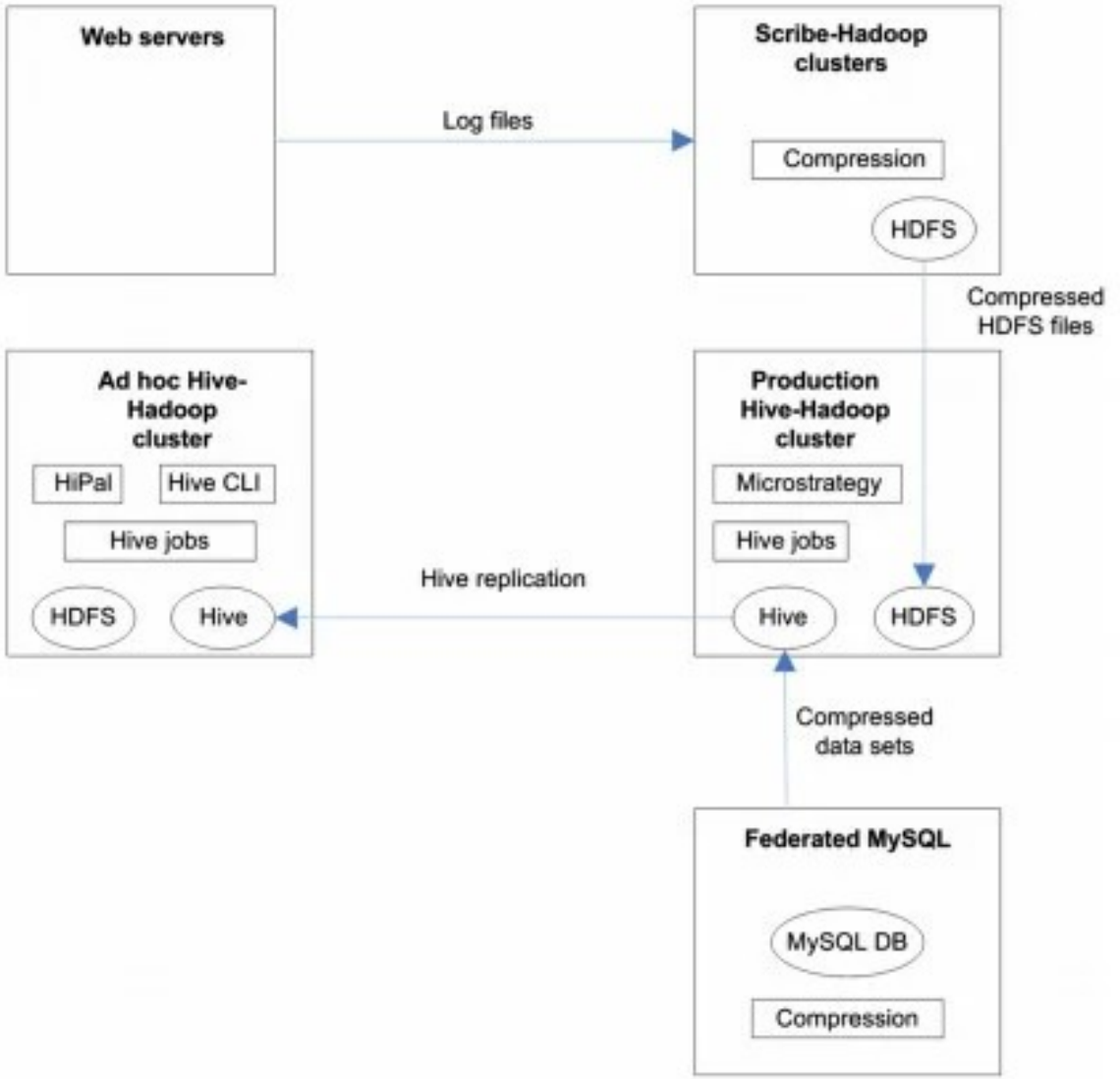
Arquitecturas data-intensive

- Arquitectura de referencia Big Data



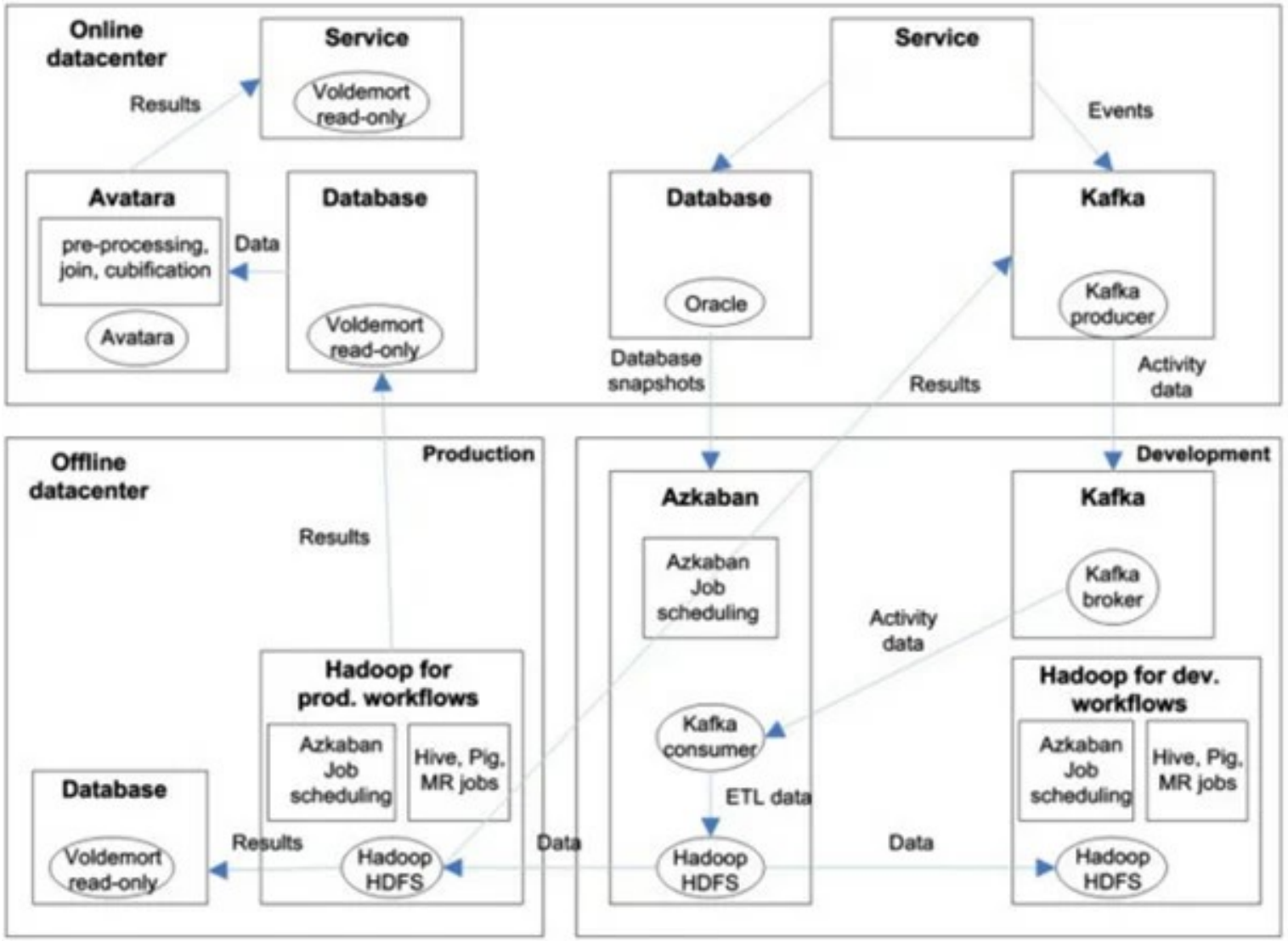
Arquitecturas data-intensive

- Ejemplo Facebook



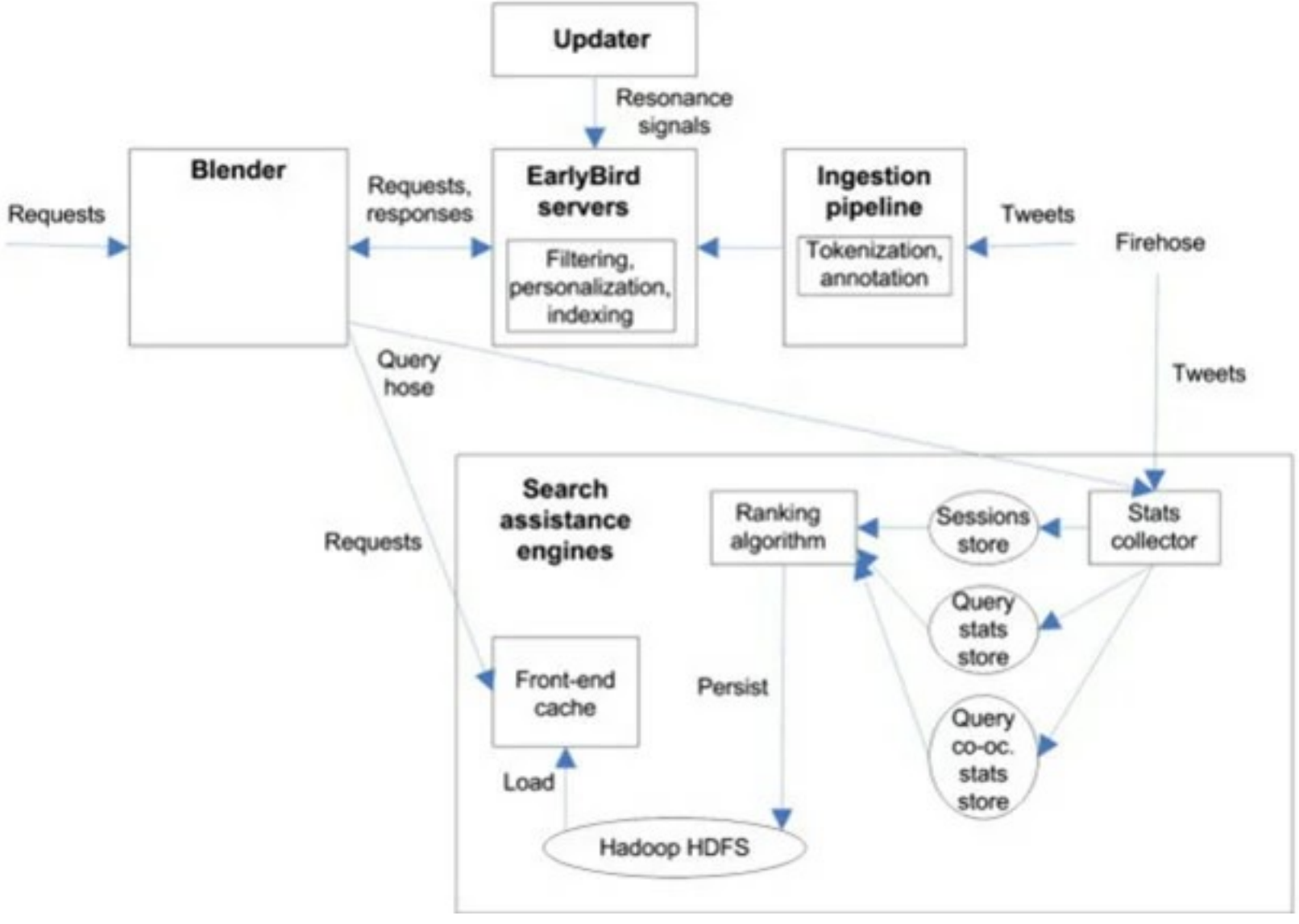
Arquitecturas data-intensive

- Ejemplo LinkedIn



Arquitecturas data-intensive

- Ejemplo Twitter



Conclusiones

- Un sistema data-intensive se compone de múltiples sistemas de datos interconectados
- Es muy importante comprender los principios básicos en los que se asientan estos sistemas de datos
- Existen arquitecturas específicas muy deseables a la hora de gestionar grandes volúmenes de información
- La arquitectura lambda es un ejemplo. La arquitectura kappa es otro ejemplo
- Estas arquitecturas utilizan técnicas de procesamiento de datos en batch y en streaming

Conclusiones

Hoja de ruta

- Tema 2: Almacenamiento de datos
- Tema 3: Procesamiento en batch
- Tema 4: Procesamiento en streaming