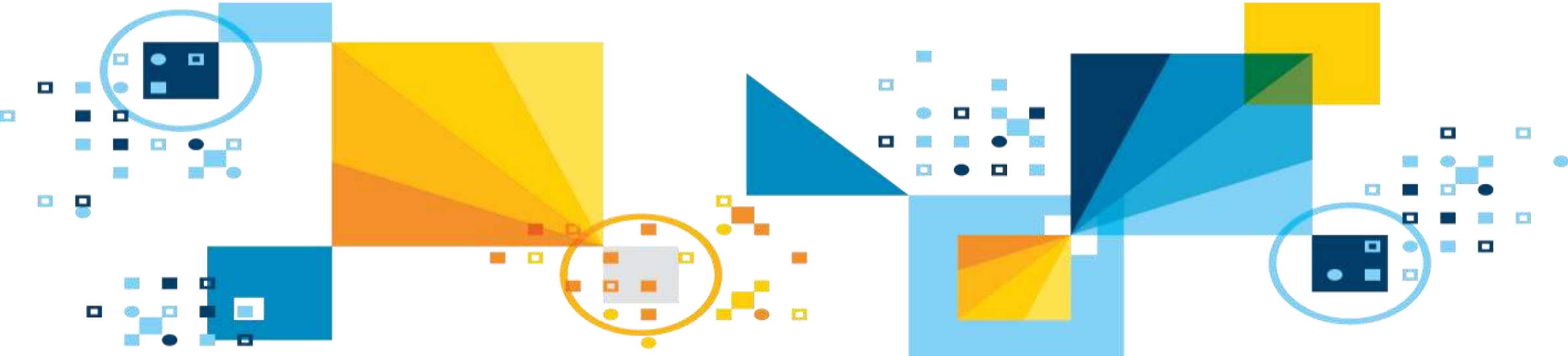

Big Data y su Ecosistema



Temario

- **Conceptos principales.**
- **Desafíos.**
- **Áreas de aplicación.**
- **Procesos involucrados.**
- **Hadoop.**
- **Arquitectura.**
- **Características.**
- **Ecosistema Hadoop.**
- **HDFS.**
- **YARN.**
- **Map Reduce.**
- **Herramientas de alto nivel.**

¿Qué significa BigData?

**Datos de gran volumen y
complejidad que escapan a
soluciones convencionales para
manejarlos de manera eficiente y
económica.**

Big Data - Orígenes de Datos

- **Web & Social Media**



- **Machine Generated Data**

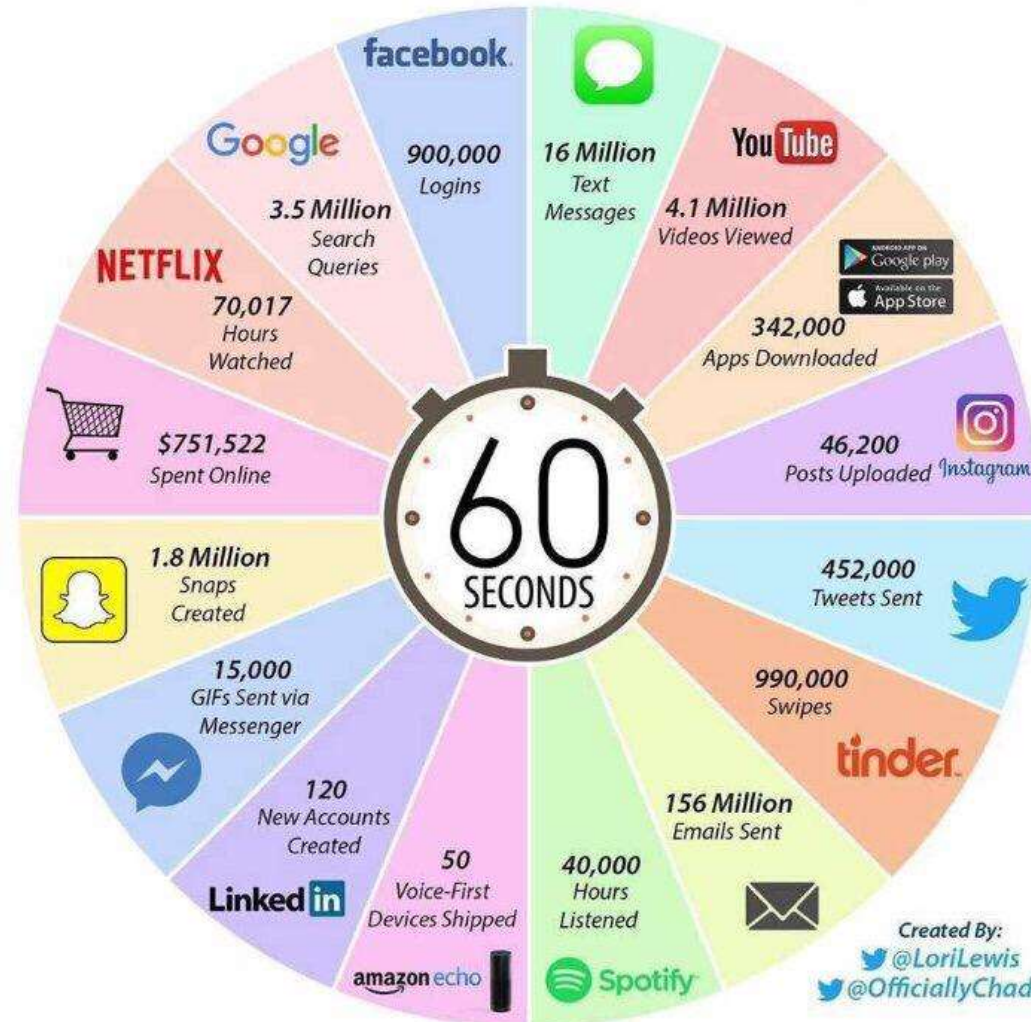


- **Internet of Things**



Big Data - Ejemplos

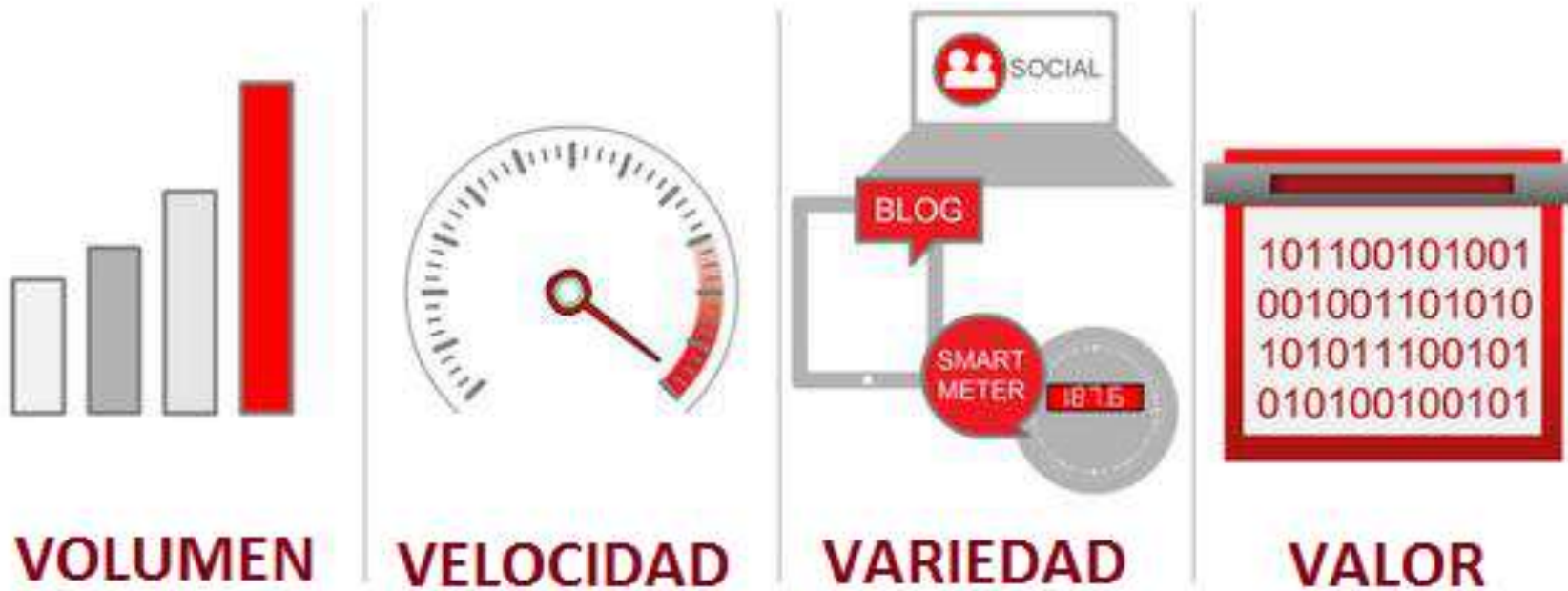
2017 *This Is What Happens In An Internet Minute*



Big Data - Seguridad



Big Data - Propiedades de los datos



Big Data - Propiedades de los datos

- **Volumen:**
 - Grandes volúmenes de datos a procesar (muchos terabytes, posiblemente petabytes, o lo que sea que se escape a lo convencional en la actualidad).
- **Velocidad:**
 - Los datos pueden generarse a una frecuencia muy elevada.
- **Variedad:**
 - Datos estructurados, semiestructurados y no estructurados.
 - Fuentes de datos posiblemente diversas, no uniformes.
 - Generalmente son datos históricos, pero pueden no serlo.
 - Pueden ser generados en tiempo real (depende de la herramienta concreta como está soportado esto).
- **Valor:**
 - Su procesamiento o análisis debe ser de utilidad al negocio.

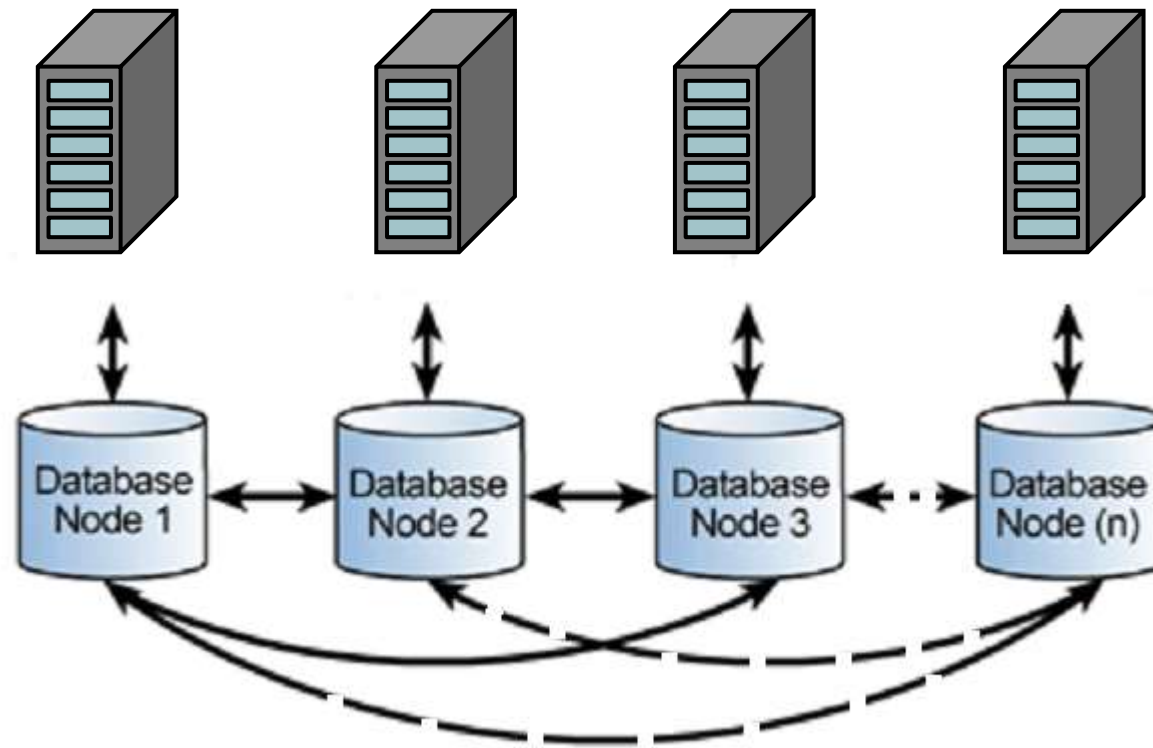
Big Data - Propiedades de los datos

- ▣ **Grandes volúmenes de datos complejos y dinámicos que no pueden almacenarse y tratarse con los métodos tradicionales**

El 90% de todos los datos existentes fue creado en los últimos dos años y este total se duplicará cada dos años.

En 2020 tendremos 50 veces más datos que los que hubo en 2011.

Big Data - Arquitectura Distribuida

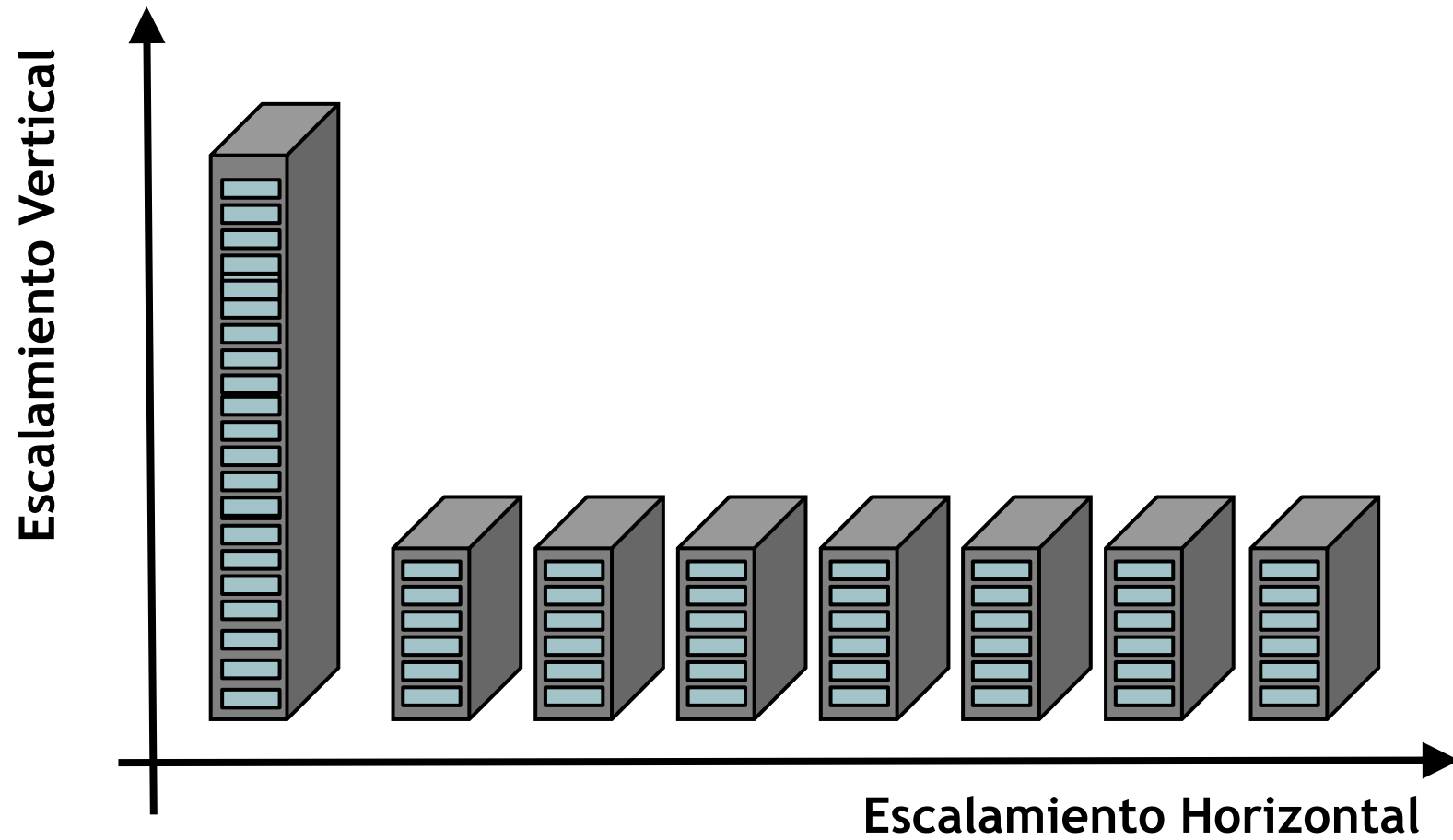


"La falla del hardware es la norma más que la excepción"

Big Data - Arquitectura Distribuida

- La arquitectura de un sistema es el aspecto de diseño más general del mismo y el más importante.
- Cuando hablamos de arquitectura distribuida, nos referimos a un sistema donde sus componentes distribuidos son el principal aspecto de diseño sobre el que se basa el sistema.
- Una arquitectura distribuida tiende a ser más tolerante a fallos, ya que en caso que uno de los nodos quede fuera de servicio, dicha actividad puede ser reemplazada por otro nodo.

Big Data - Arquitectura Distribuida



Big Data - Arquitectura Distribuida

Escalamiento Vertical:

- Consiste en aumentar la capacidad de procesamiento del sistema asignando más recursos a un único nodo.
- Es más simple de comprender y mantener, pero no es tolerante a fallos, ya que todas las tareas recaen en un único equipo.
- También tiene las limitaciones propias de la tecnología que se quiere escalar, por ejemplo existe un límite para la cantidad de memoria RAM que una computadora puede manejar.
- Este tipo de escalamiento es el más utilizado en arquitecturas convencionales.

Big Data - Arquitectura Distribuida

Escalamiento Horizontal:

- Consiste en aumentar la capacidad de procesamiento asignando más nodos a un sistema.
- Dichos equipos no serán capaces de manejar de manera individual un gran volumen de operaciones, pero en su conjunto serán capaces de procesar un mayor número de operaciones.
- Este tipo de arquitectura es más difícil de comprender y mantener, pero es tolerante a fallos ya que la caída de un nodo puede afectar el rendimiento pero no implica una falla en el servicio.
- Este tipo de escalamiento es el utilizado en arquitecturas distribuidas.

5 casos de uso en Big Data



Big Data Exploration

Find, visualize, understand all big data to improve decision making



Enhanced 360° View of the Customer

Extend existing customer views by incorporating additional internal and external data sources



Security/Intelligence Extension

Lower risk, detect fraud and monitor cyber security in real-time



Operations Analysis

Analyze a variety of machine data for improved business results



Data Warehouse Augmentation

Integrate big data and data warehouse capabilities to increase operational efficiency

Big Data y Data Mining



Grandes Volúmenes
de datos



Con el BIG DATA aparecen nuevos factores relacionados con:

Recogida, Almacenamiento y Transmisión de la información

Tiempo de procesamiento

Variabilidad de los datos

Y entonces el proceso de minería de datos pasa a ser una parte del fenómeno de BIG DATA: hacen falta nuevas formas de manipular la información y métodos de data mining que se adapten a grandes volúmenes

¿Qué procesos involucra Big Data?

Tareas

- Almacenar y acceder a grandes volúmenes de datos no-estructurados
- Procesar grandes volúmenes de datos on-line (data streams)
- **Dar utilidad a los datos**
- **Tecnologías para hacer predicciones**



Tecnologías

- Manejo de base de datos
- Modelos de programación paralela y distribuida
- Técnicas de minería de datos**
- Machine learning**

Big Data: Origen de los datos

Internet : Transacciones, reservas de viaje, comentarios, periódicos, redes sociales.

Cámaras web: Videovigilancia, móviles.

Sensores: Wearables, estaciones de control meteorológico, etc.

Big Data: Desafíos

- Así como Henry Ford cambió la forma de fabricar coches transformando al mismo tiempo la forma de trabajo, Big Data emerge como un nuevo paradigma de conocimiento que puede cambiar la forma en la que entendemos las relaciones sociales.

‘Change the instruments, and you will change the entire social theory that goes with them’ (Latour 2009)

¿Es el BIG DATA un fenómeno real o una moda ?

Big Data: Desafíos

□ ¿Será una herramienta útil al alcance de todas las empresas?

ó

□ ¿Llevará a una mayor concentración, en la cual sólo salen beneficiadas las empresas más grandes dueñas de los datos?

Big Data: Desafíos

□ **¿Tener acceso a tantos datos ayudará a tener mejores herramientas, servicios y bienes públicos ?**

ó

□ **¿Llevará a una nueva ola de invasión de la privacidad con ofertas invasivas de marketing ?**

Big Data: Desafíos

¿ Nos ayudará a entender las comunidades digitales y los nuevos movimientos sociales ?

ó

¿Será una herramienta para monitorizar los grupos políticos y suprimir la libertad de expresión?

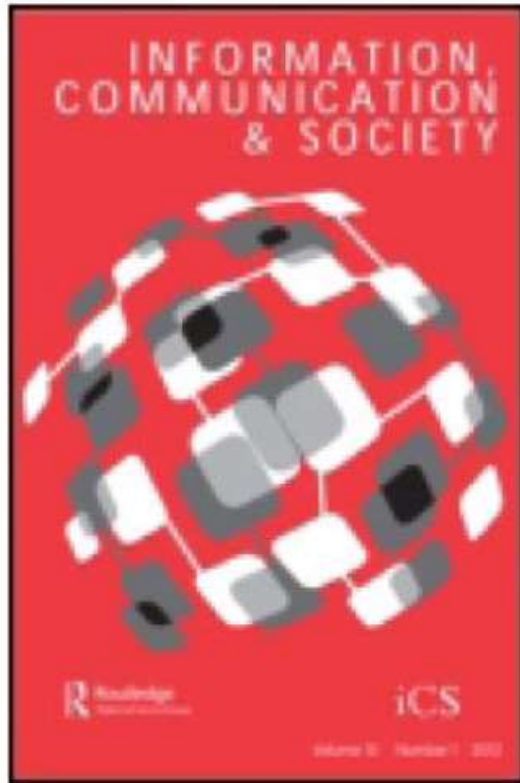
Big Data: Desafíos

□ **¿ Modificará la forma de estudiar la comunicación y conductas humanas?**

ó

□ **¿Estrechará la paleta de opciones de investigación, alterando lo que significa “investigar”?**

Big Data: Desafíos



Information, Communication & Society

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/rics20>

CRITICAL QUESTIONS FOR BIG DATA

danah boyd^a & Kate Crawford^b

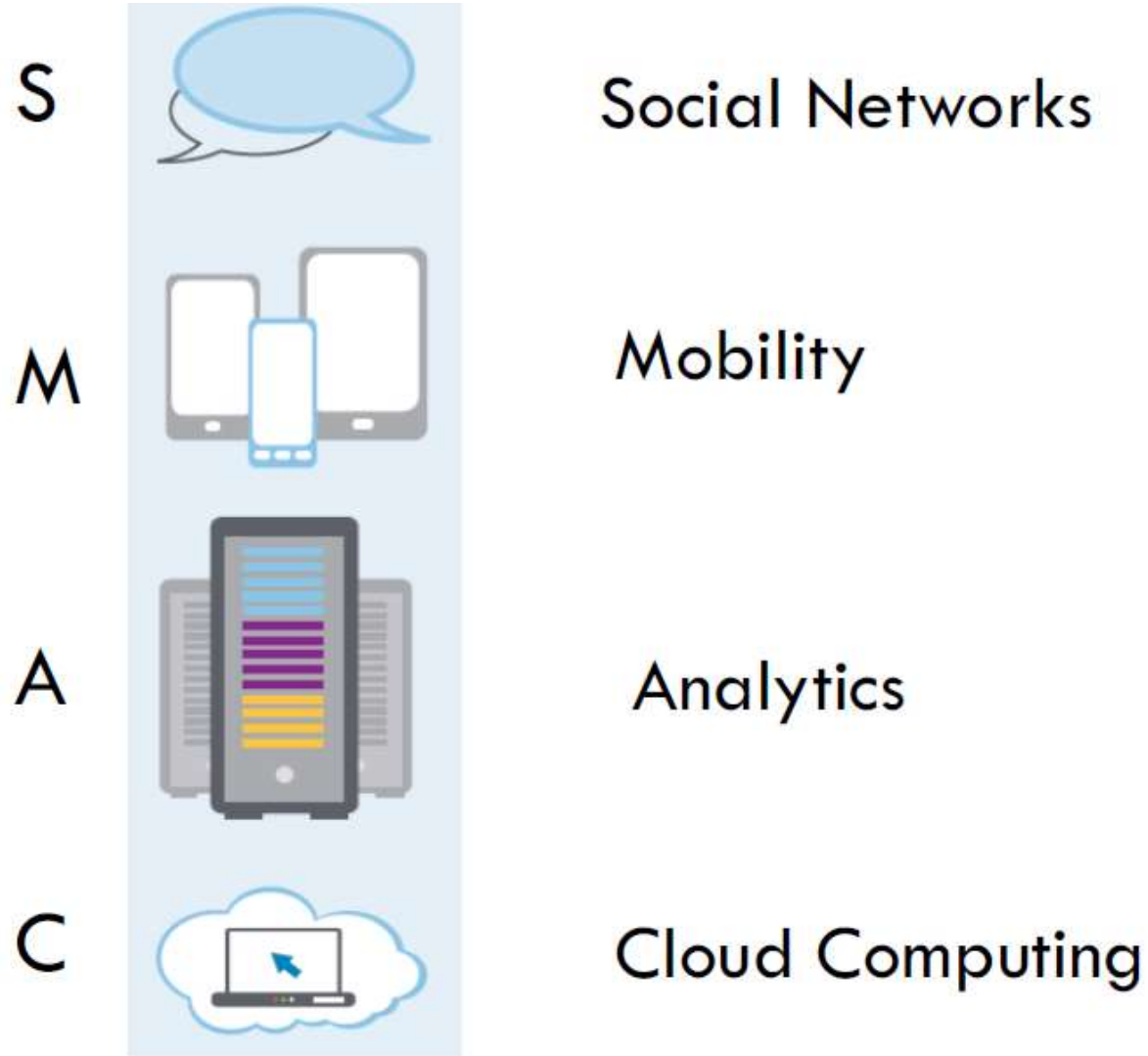
^a Microsoft Research , One Memorial Drive ,
Cambridge , MA , 02142 , USA

^b Microsoft Research , One Memorial Drive ,
Cambridge , MA , 02142 , USA E-mail:

Published online: 10 May 2012.

Essential Reading: *BD-R1-critical-questions.pdf*

Big Data: Nuevas tecnologías para los negocios



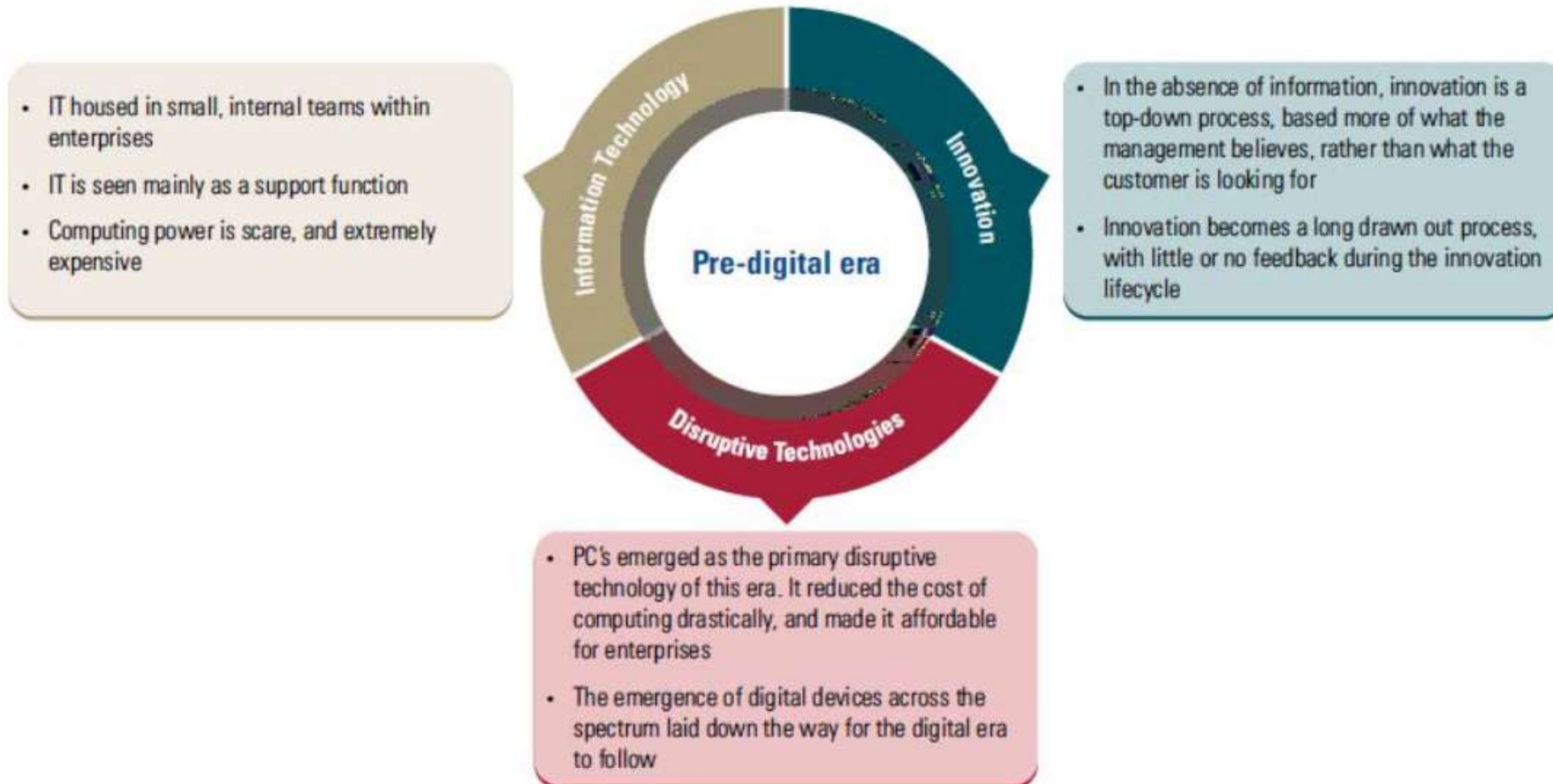
Big Data: Impacto en el marketing

	Pre-digital era	Digital era	Post-digital era
Organization structure	Highly centralized	Semi-decentralized	Highly decentralized
Decision making process	Management driven	Management driven, with customer input	In collaboration with the customer
Customers	Lack information	More informed	Highly informed
Role of technology	Support function	Strategic function	Business enabler
Disruptive technologies	PC's	Internet, mobility, social media	Artificial intelligence, Robotics, Augmented reality, cloud computing, BYOD

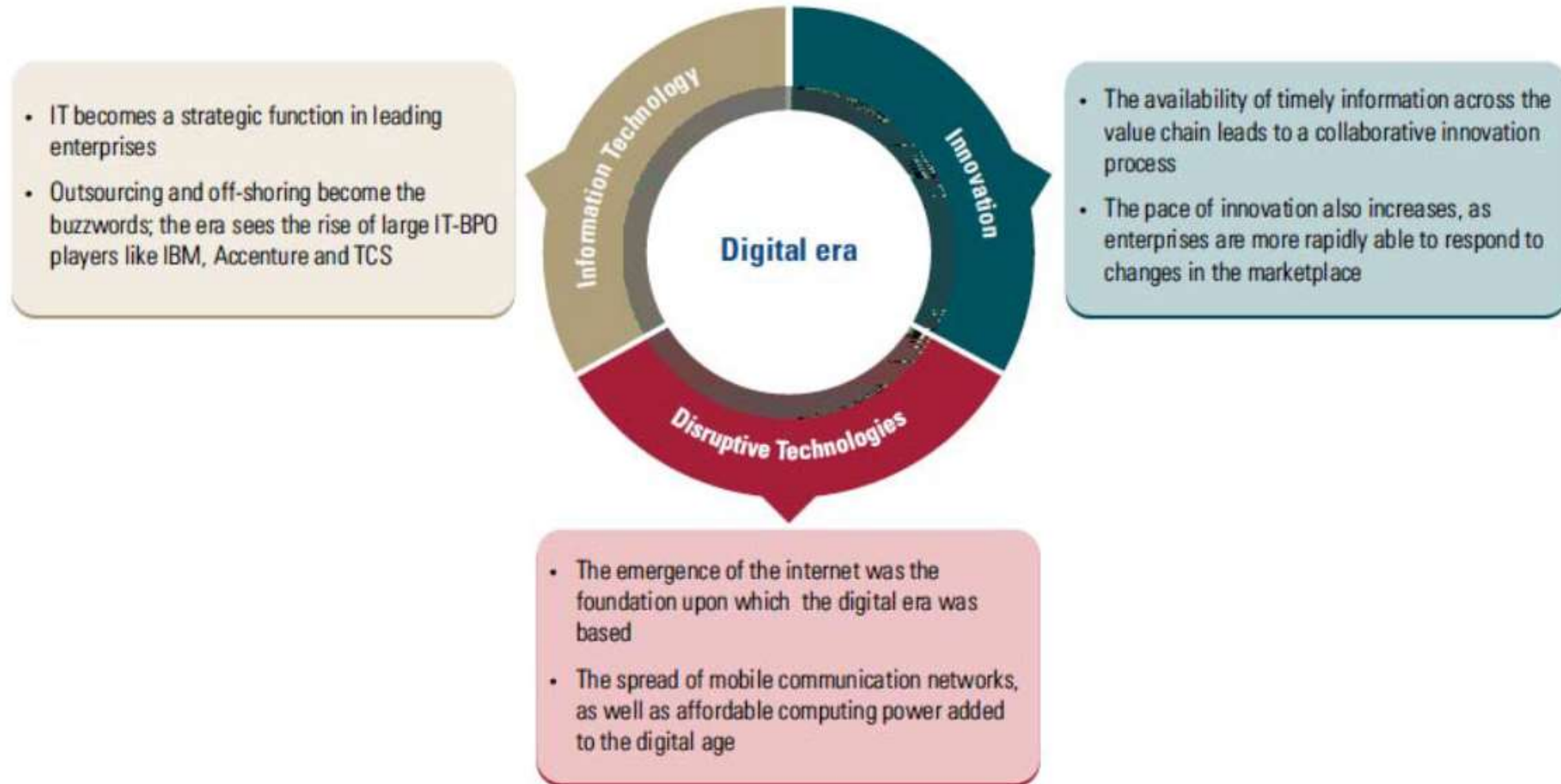
Referencia Esencial
BD-R2-TheSmacCode.pdf



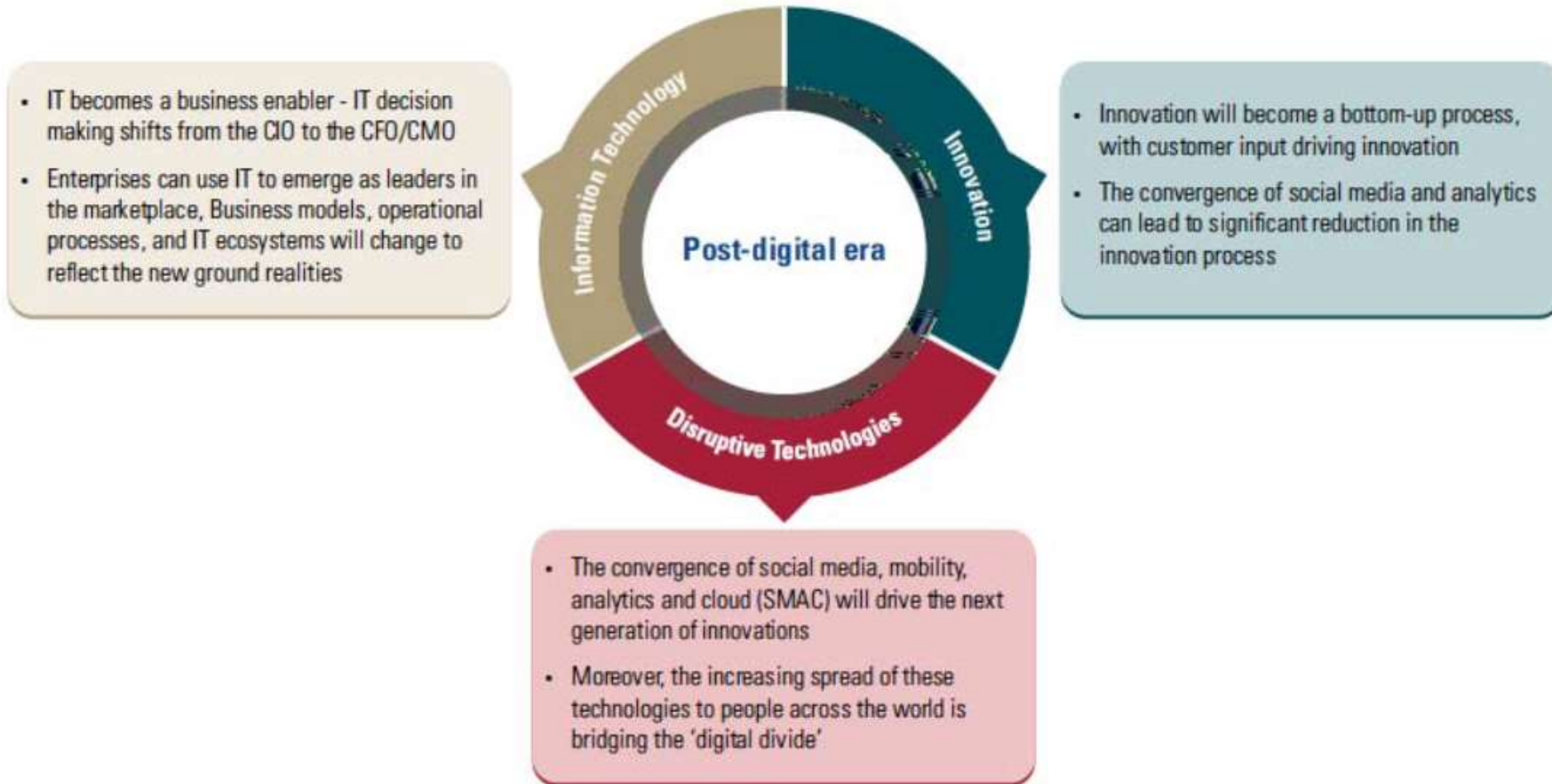
Big Data: Impacto en el marketing



Big Data: Impacto en el marketing



Big Data: Impacto en el marketing



Big Data y marketing digital

El fenómeno de Big Data constituye un nuevo horizonte para el marketing digital.

Es un nuevo reto, una nueva herramienta, que ofrece un potencial enorme, y que habrá que saber explotar de la mejor forma posible. Son dos ámbitos que están condenados a entenderse y a trabajar juntos.

Pasos fundamentales para la explotación del Big Data en Marketing

- ❑ Escuchar y conversar en medios sociales (CRMs u otra fuente).
- ❑ Establecer mecanismos apropiados de recolección de datos, analizando qué mensaje es el más apropiado para cada tipo de cliente.
- ❑ Agrupar los diferentes mensajes por cada tipo de cliente
- ❑ Enviar el mensaje a los clientes en función del dispositivo en el que los recibirá

Big Data y marketing digital

Otros aspectos relevantes el futuro del Big Data y el marketing digital son: **una mejor experiencia de usuario unida a una mayor personalización de los contenidos**, lo que conlleva inevitablemente a un **aumento del grado de relevancia y de valor de los mensajes**.

Entonces podemos ir extrayendo las siguientes conclusiones:

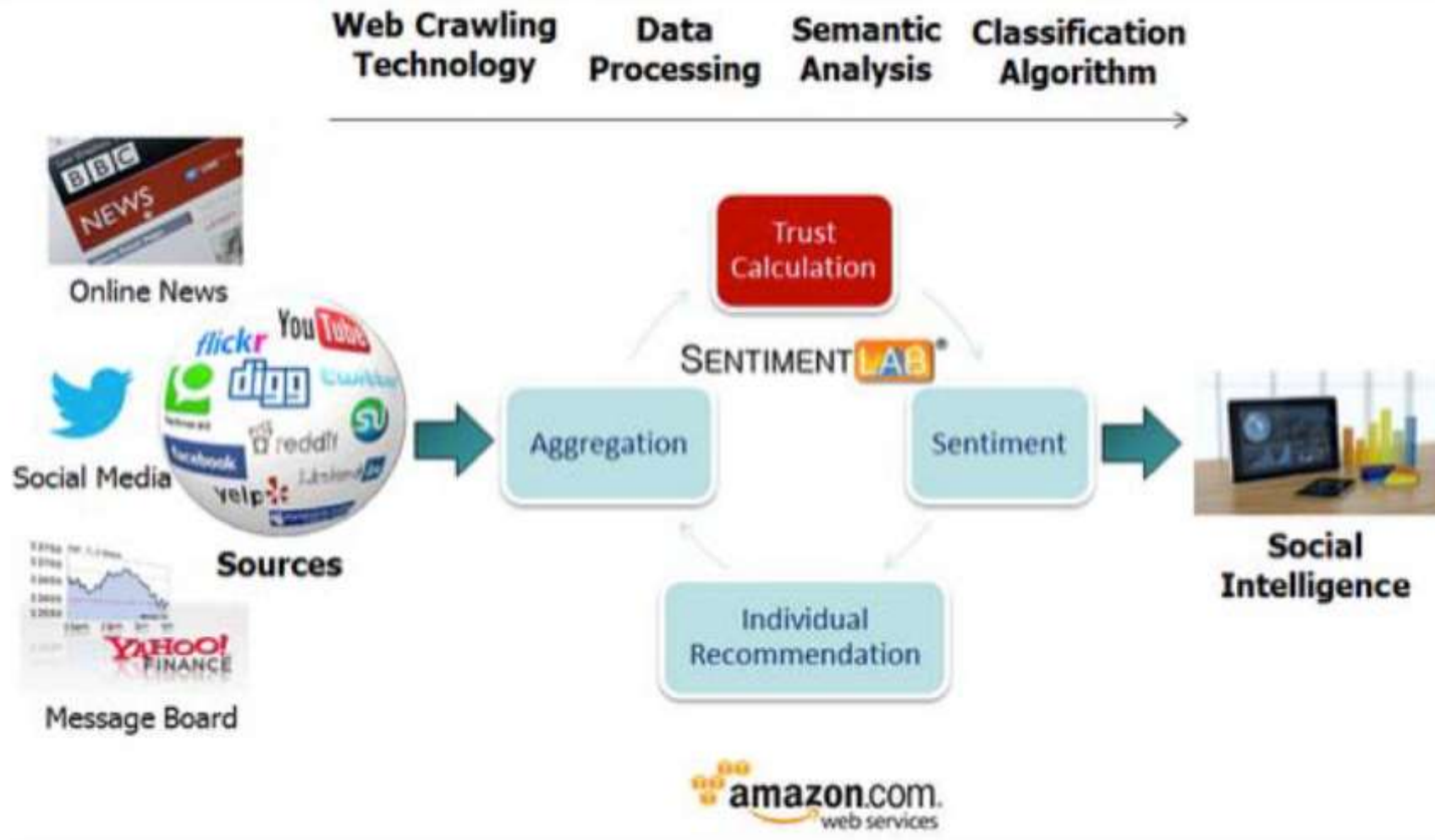
el Big Data es precisamente eso, datos: **Datos en bruto**, que habrá que convertir en información, para posteriormente, **adquirir conocimiento**.

el Big Data proporciona un **marketing en tiempo real**, y **añade la inmediatez como elemento clave** y ampliamente demandado por los usuarios y consumidores finales.

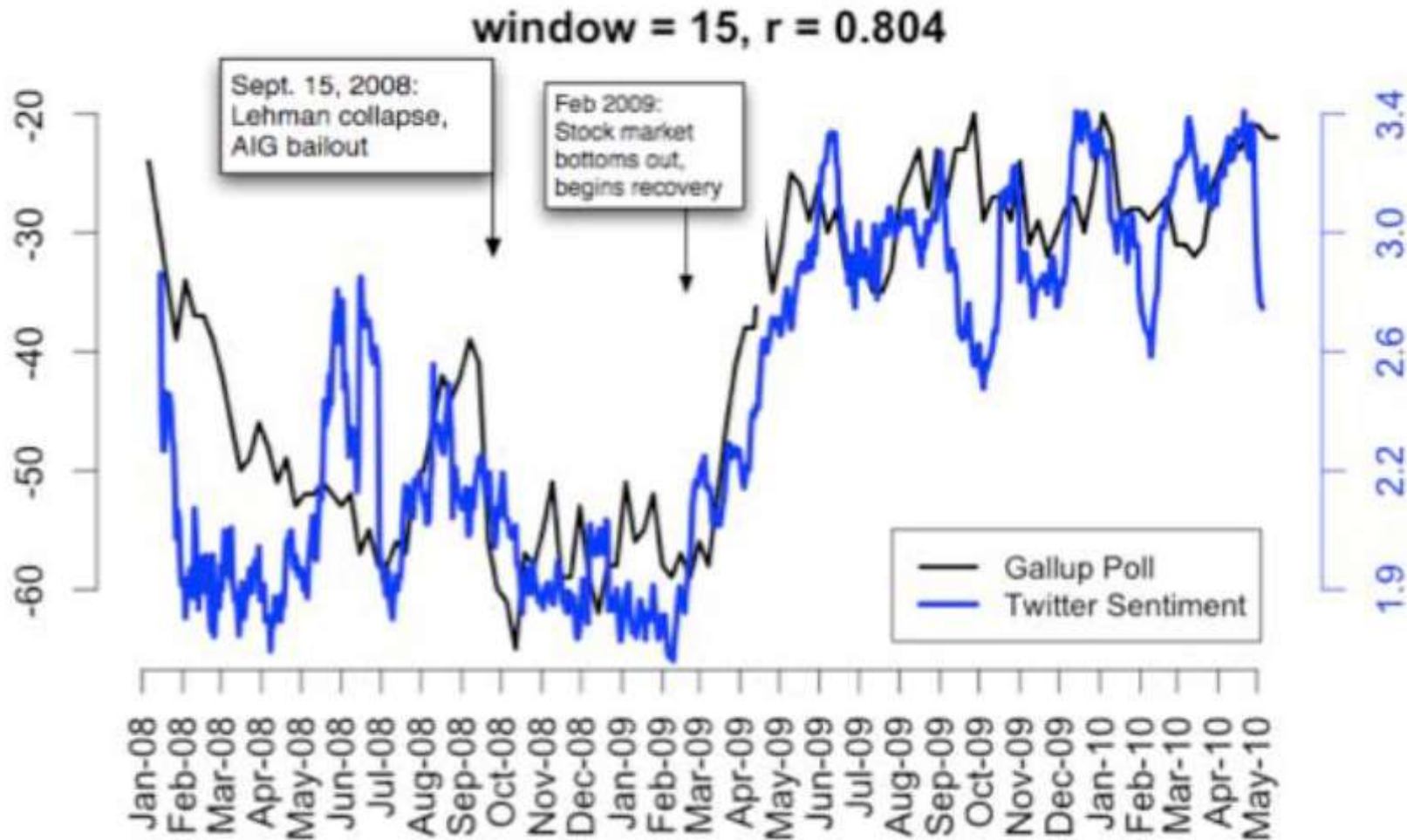
Big Data: Áreas de aplicación

- Sentiment analysis (imagen de políticos, mercados financieros, empresas)
- Recomendaciones on-line: de compra (Amazon, ebay, Netflix, supermercados), de amistades (FB, LinkedIn), publicidad dirigida.
- Optimización de costes ("General Electric", "Smart cities").
- Predicciones (Potenciales clientes, Bolsa, etc.)

Big Data: Métodos involucrados en sentiment analysis



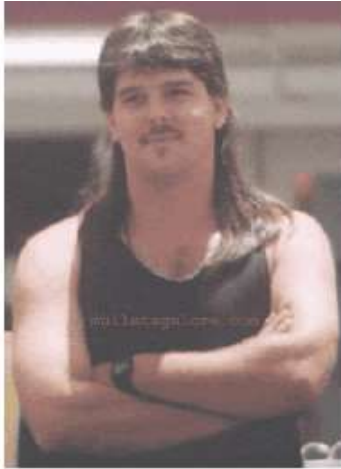
Big Data: Sentiment analysis – Twitter aplicado a medir la Confianza del Consumidor



Big Data: Sentiment analysis – Twitter aplicado a medir la Satisfacción del Consumidor

<u>Hotels</u>	75	73	72	71	71	72	72	71	71	73	72	73	75	71	75	75	75		0.0	0.0
<u>Limited Service Restaurants</u>	69	70	66	68	69	69	70	71	71	74	NM**	76	77	77	78	78	75		-3.8	8.7
<u>Supermarkets</u>	76	75	74	73	73	74	73	75	75	74	73	74	75	76	76	76	75		-1.3	-1.3
<u>Fixed Line Telephone Service</u>	81	80	79	75	74	73	72	70	71	72	71	70	70	70	73	72	75	73	-2.7	-9.9
<u>Internet News & Information</u>									73	74	75	75	73	75	75	74	74		0.0	1.4
<u>Network Cable TV News</u>	77	76	70	62	65	62	64	62	65	68	68	68	69	67	69	71	74	77	-4.1	0.0
<u>Health Insurance</u>								68	69	70	67	68	72	71	73	75	73		-2.7	7.4
<u>Hospitals</u>	74	74	71	67	72	70	69	68	70	73	76	71	74	77	75	77	73	77	5.5	4.1
<u>Wireless Telephone Service</u>											65	63	66	68	68	69	72	71	-1.4	9.2
<u>U.S. Postal Service</u>	61	69	74	69	71	71	72	70	73	72	74	73	71	73	74	74	71		-4.1	16.4
<u>Internet Social Media</u>																	70		N/A	N/A
<u>Gasoline Stations</u>	78	80	77	78	79	76	75	77	76	75	70	69	71	70	74	76	70		-7.9	-10.3
<u>Subscription Television Service</u>								64	61	61	61	61	63	62	64	63	66	66	0.0	3.1
<u>Airlines</u>	72	69	69	67	65	63	63	61	66	67	68	66	65	63	62	64	66		3.1	-8.3
<u>Newspapers</u>	72	68	69	69	66	69	68	68	63	64	68	63	63	66	64	63	65	65	0.0	-9.7
<u>Municipal Utilities</u>																		73	N/A	N/A
<u>Investor-Owned Utilities</u>																		74	N/A	N/A
<u>Cooperative Utilities</u>																		82	N/A	N/A

Big Data: Métodos involucrados en sistemas de recomendación



□ Cliente X

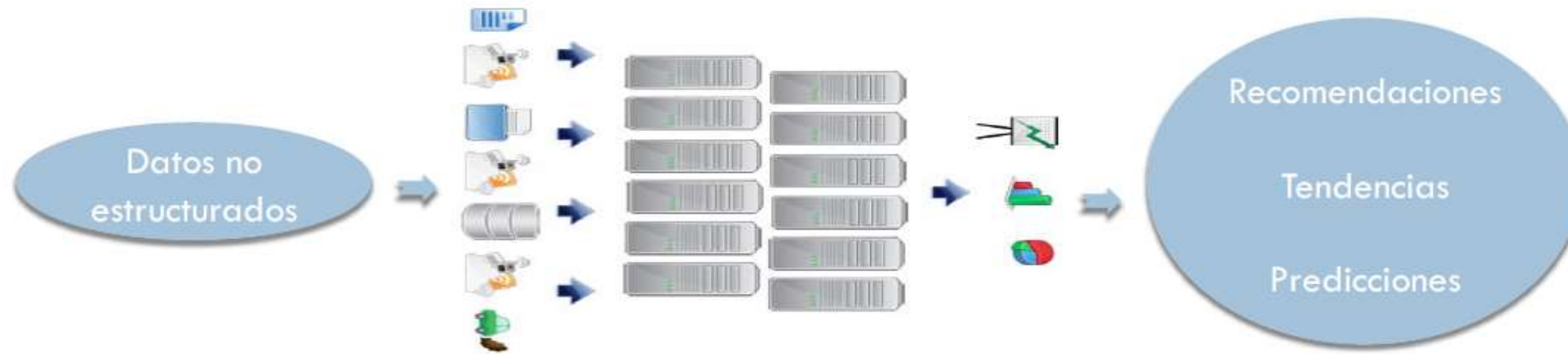
- Compra un CD de Metallica
- Compra un Cd de Megadeth



□ Cliente Y

- Busca sobre Metallica
- El sistema le recomienda Megadeth fde acuerdo a los datos del usuario X

Big Data: Procesos involucrados



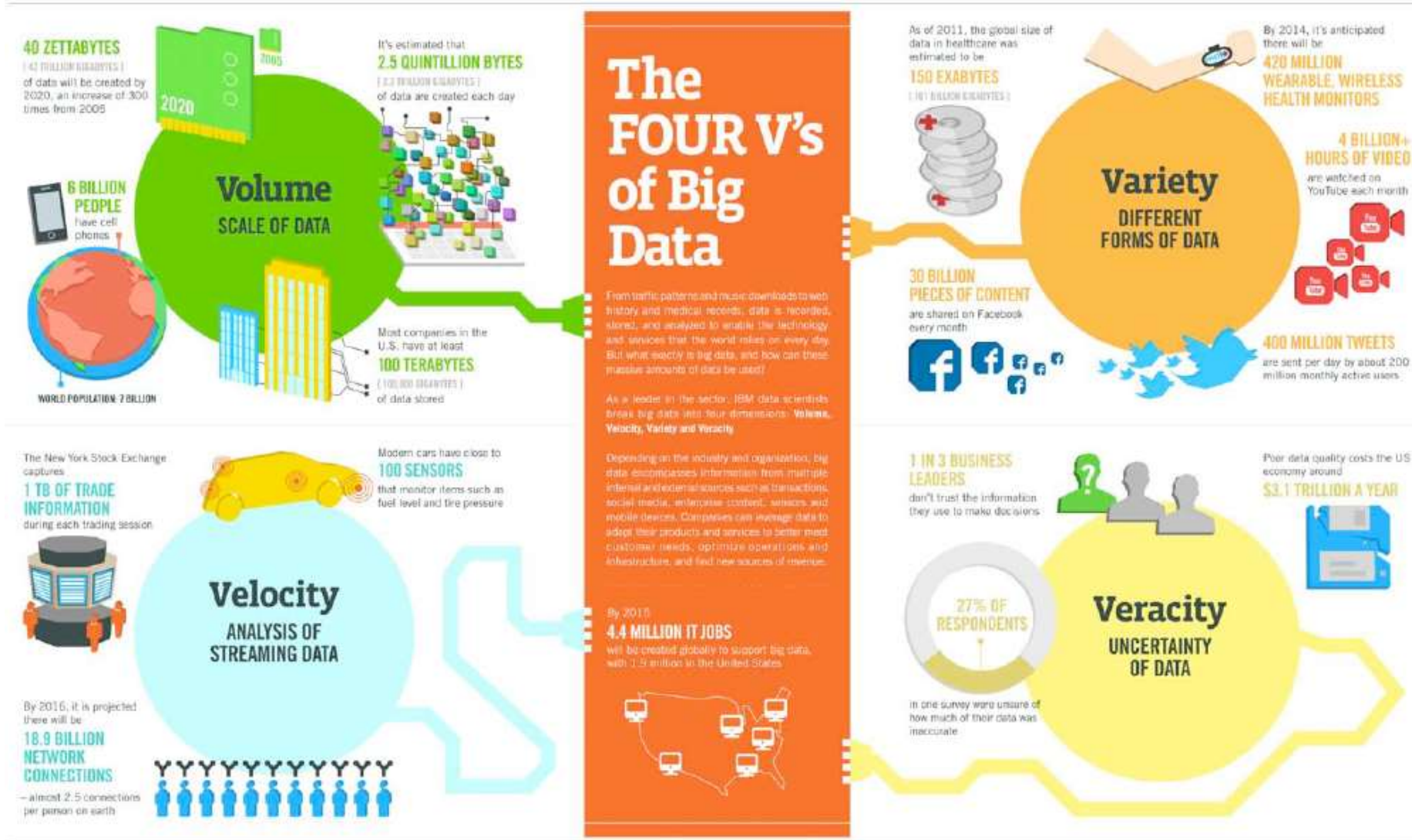
Tareas

- Almacenar y acceder a grandes volúmenes de datos no-estructurados
- Procesar grandes volúmenes de datos on-line (data streams)
- **Dar utilidad a los datos**
- **Tecnologías para hacer predicciones**

Tecnologías

- Manejo de base de datos
- Modelos de programación paralela y distribuida
- Técnicas de minería de datos**
- Machine learning**

Big Data: Las 4 V



Big Data: Volumen

40 ZETTABYTES

[43 TRILLION GIGABYTES]

of data will be created by 2020, an increase of 300 times from 2005

2020

2005

It's estimated that **2.5 QUINTILLION BYTES**

[2.3 TRILLION GIGABYTES]

of data are created each day

A PETABYTE IS A LOT OF DATA

- 1 PETABYTE** = 20 MILLION FOUR-DRAWER FILING CABINETS FILLED WITH TEXT
- 1 PETABYTE** = 13.3 YEARS OF HD-TV VIDEO
- 1.5 PETABYTES** = SIZE OF THE 10 BILLION PHOTOS ON FACEBOOK
- 20 PETABYTES** = THE AMOUNT OF DATA PROCESSED BY GOOGLE PER DAY



6 BILLION PEOPLE have cell phones



WORLD POPULATION: 7 BILLION



~ exabytes



100 PB

Volume SCALE OF DATA

VOLUME



Big Data: Velocidad

The New York Stock Exchange captures

1 TB OF TRADE INFORMATION

during each trading session



Modern cars have close to **100 SENSORS** that monitor items such as fuel level and tire pressure

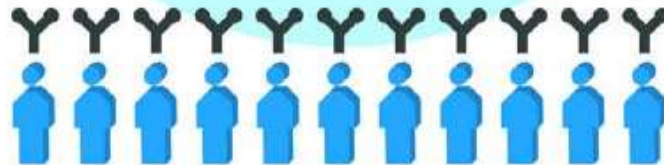
Alibaba vendió 2.000 millones de dólares en una hora (Nov 2014)

Velocity
ANALYSIS OF
STREAMING DATA

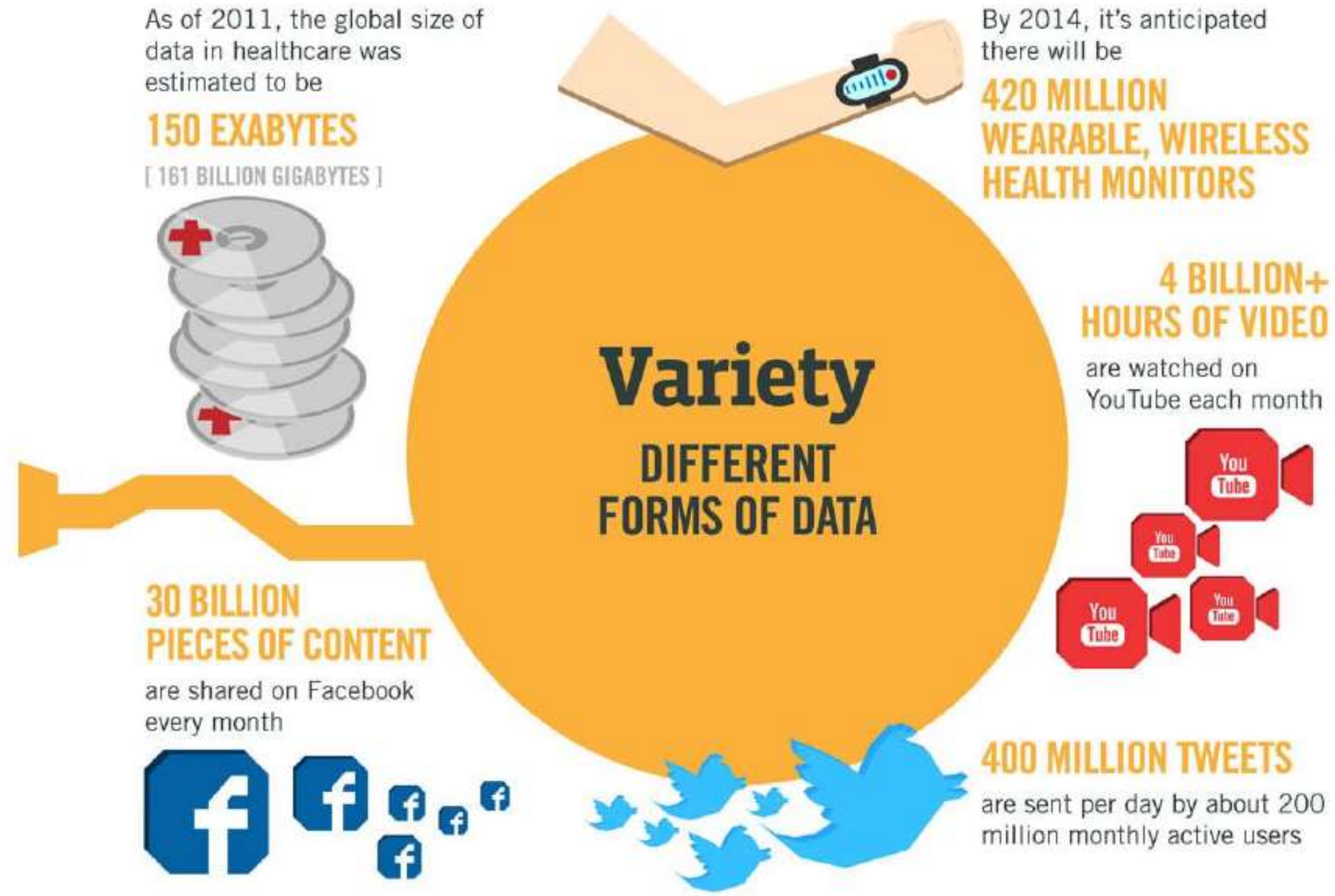
By 2016, it is projected there will be

18.9 BILLION NETWORK CONNECTIONS

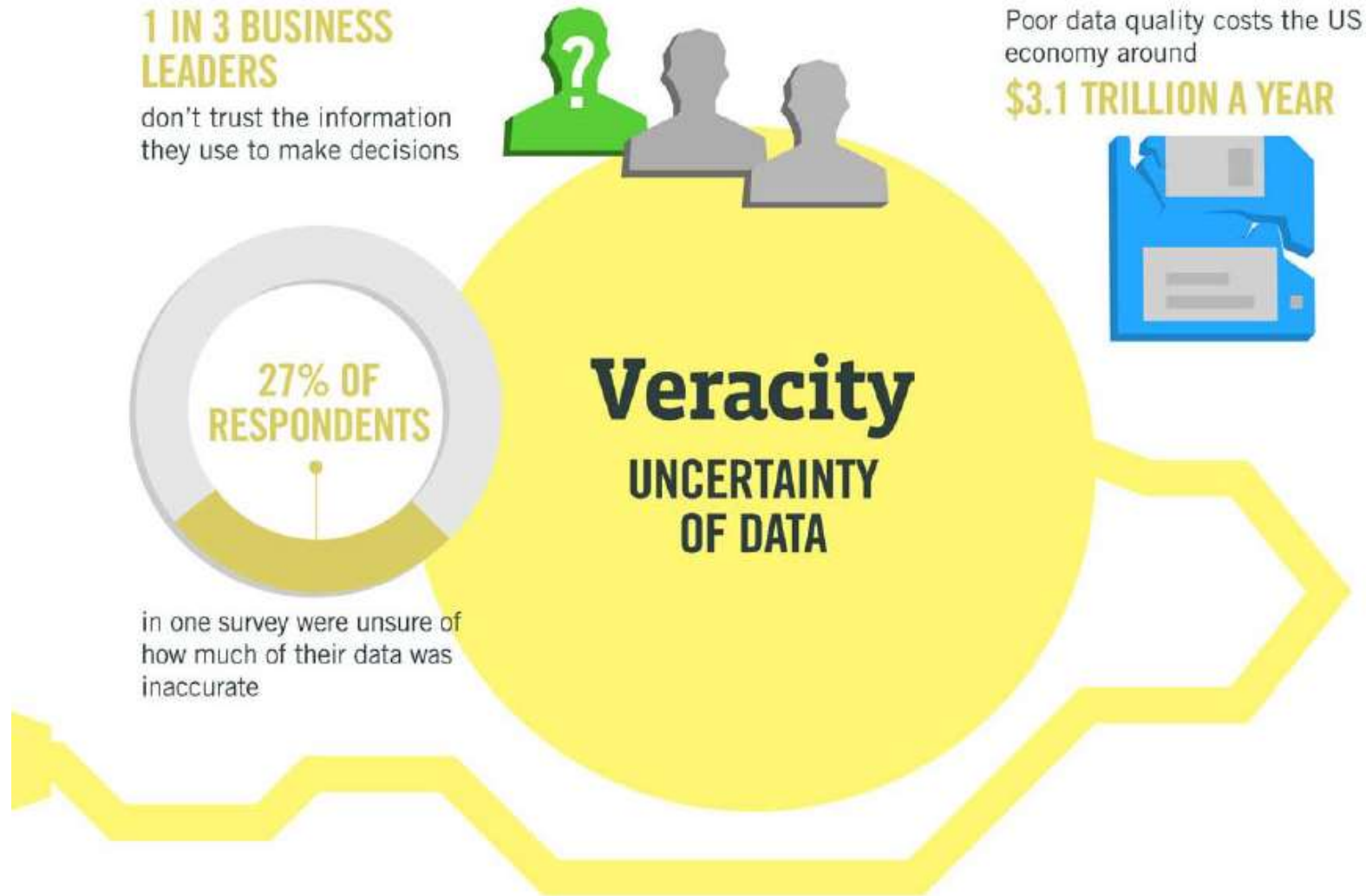
– almost 2.5 connections per person on earth



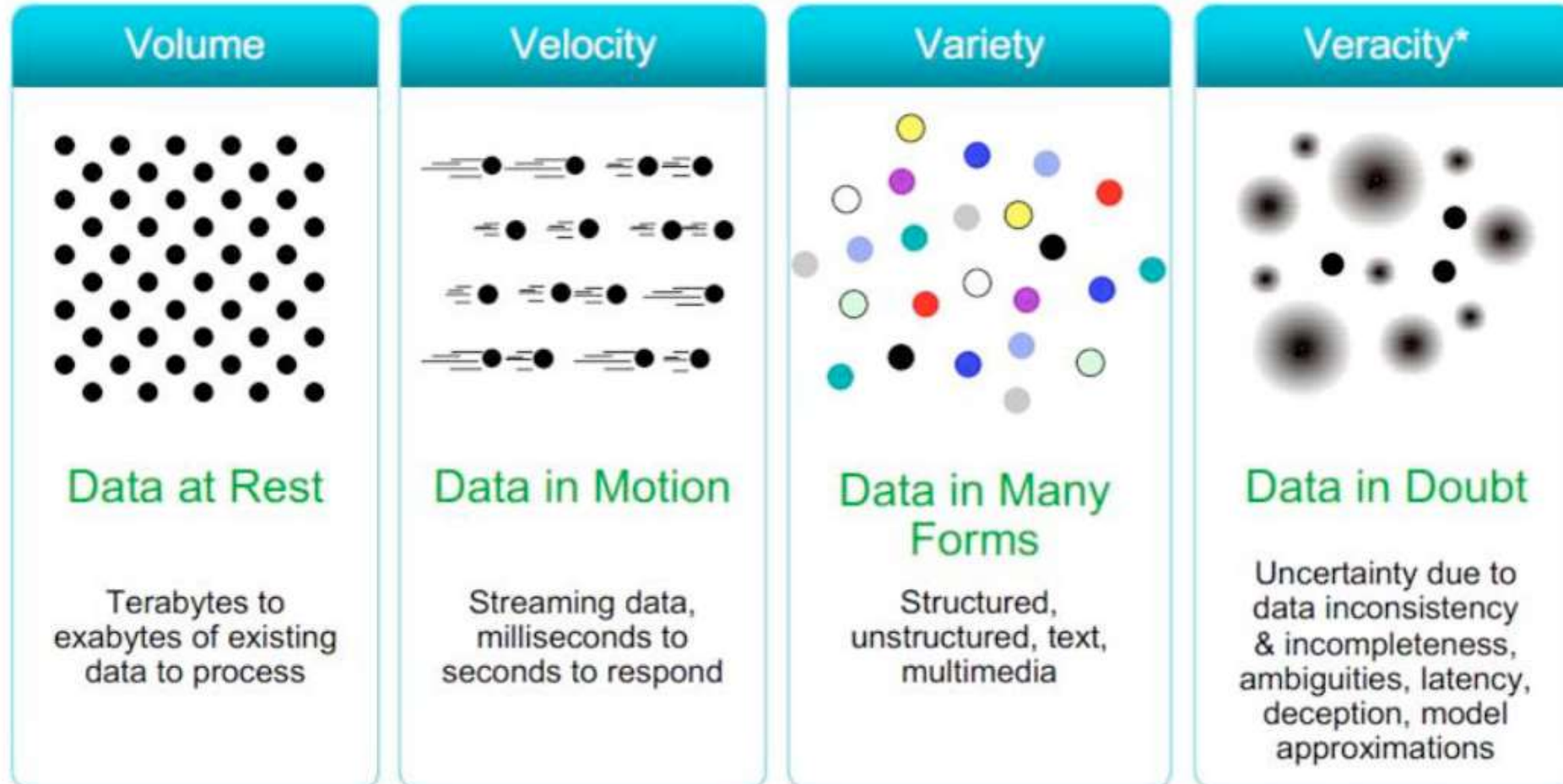
Big Data: Variedad (heterogeneidad de los datos)



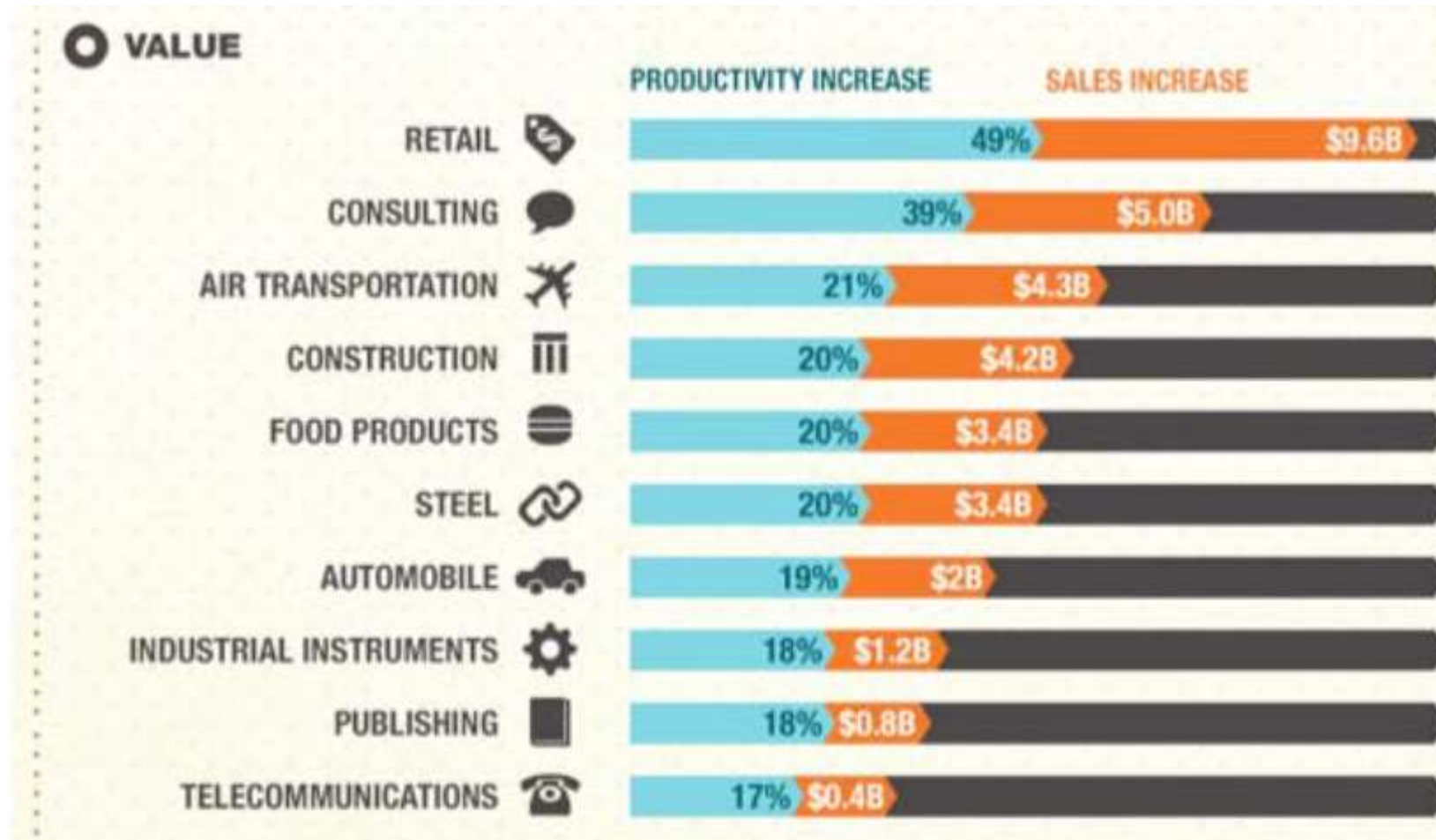
Big Data: Veracidad



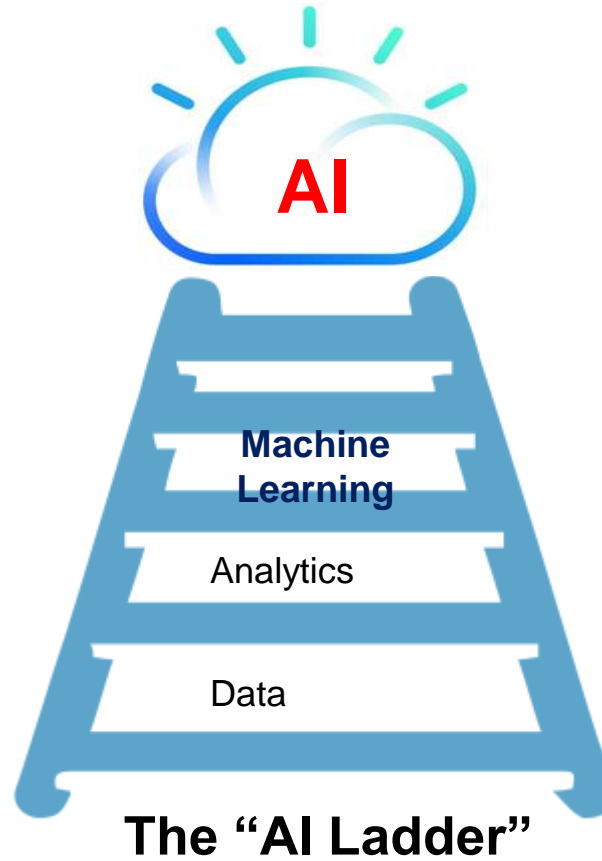
Big Data: Las 4 V



Big Data: Aumento de productividad y ventas previsto por su impacto



Big Data: La escalera hacia la Inteligencia Artificial



Big Data: La escalera hacia la inteligencia artificial

AI requires machine learning.

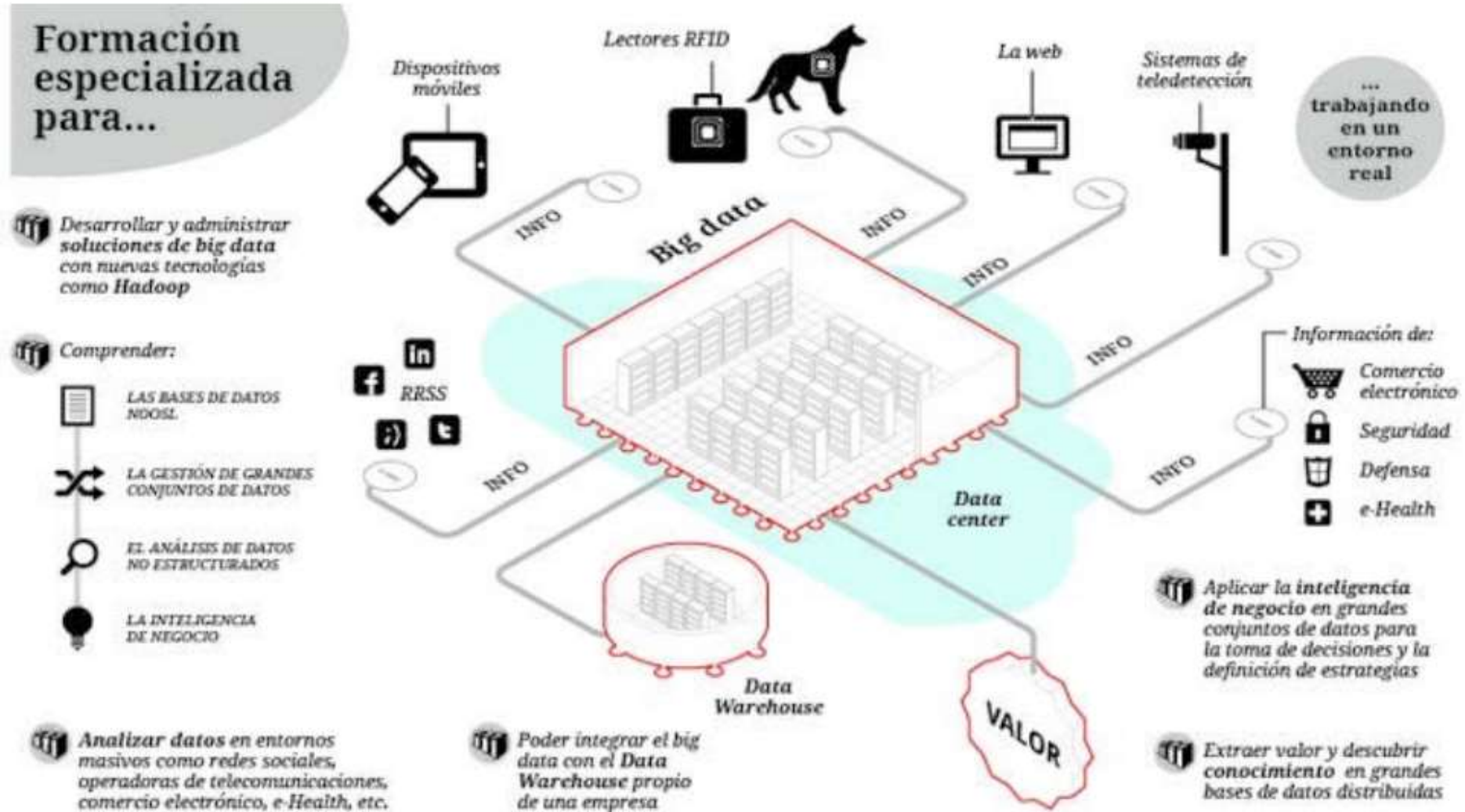
Machine Learning requires analytics.

Analytics requires the right data and information architecture.

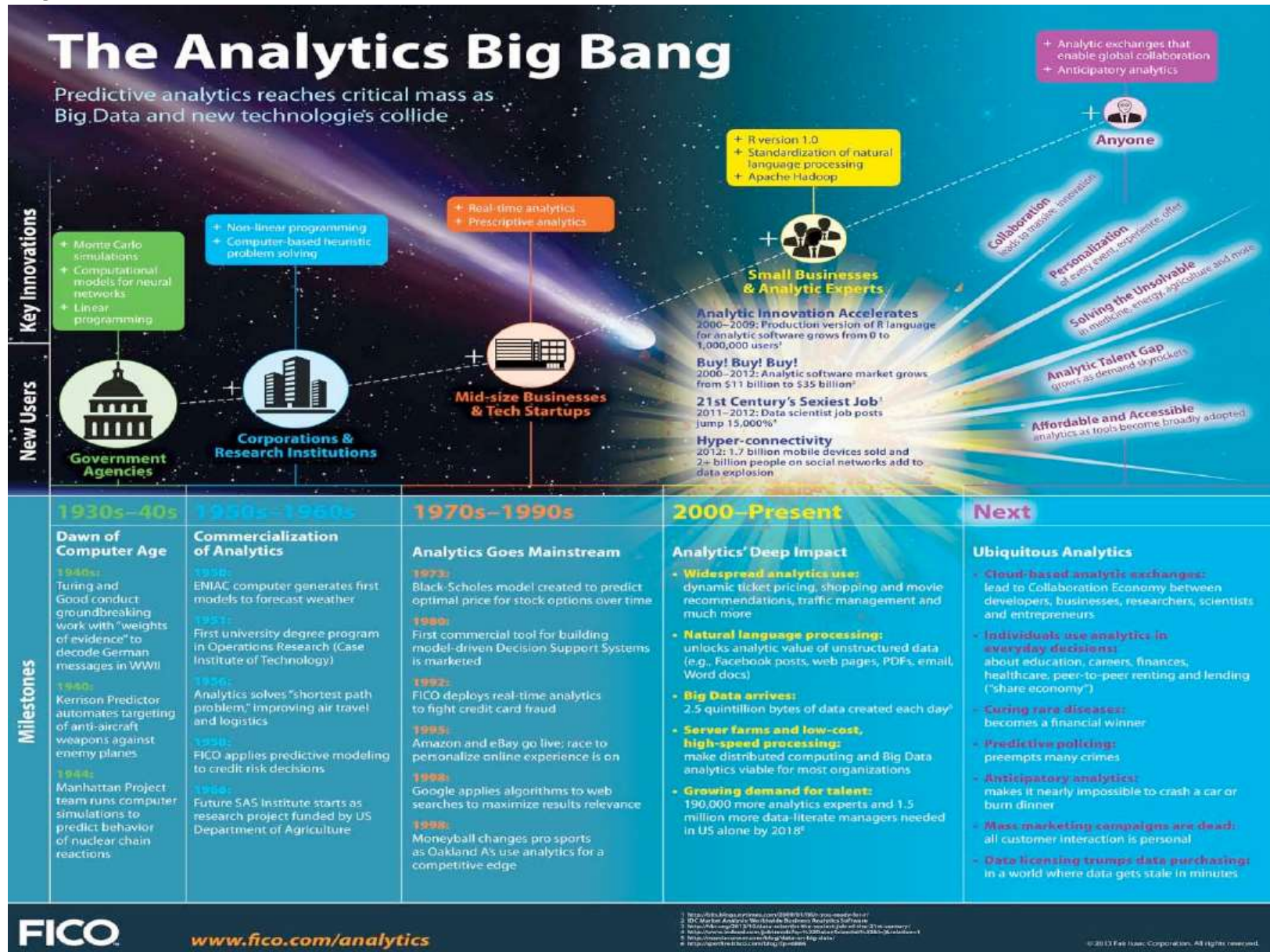
In other words, there is no AI without IA (Information Architecture).

These capabilities form the solid rungs of what we call the AI Ladder – the increasing levels of analytic sophistication that lead to, and buttress, a thriving AI environment.

Big Data: Infografía resumen



Big Data: Analytics



Algunos links útiles

- **cognitiveclass.ai (aka Big Data University):**

- <https://cognitiveclass.ai/>

- **Iniciativa Académica IBM:**

- <https://developer.ibm.com/academic/>

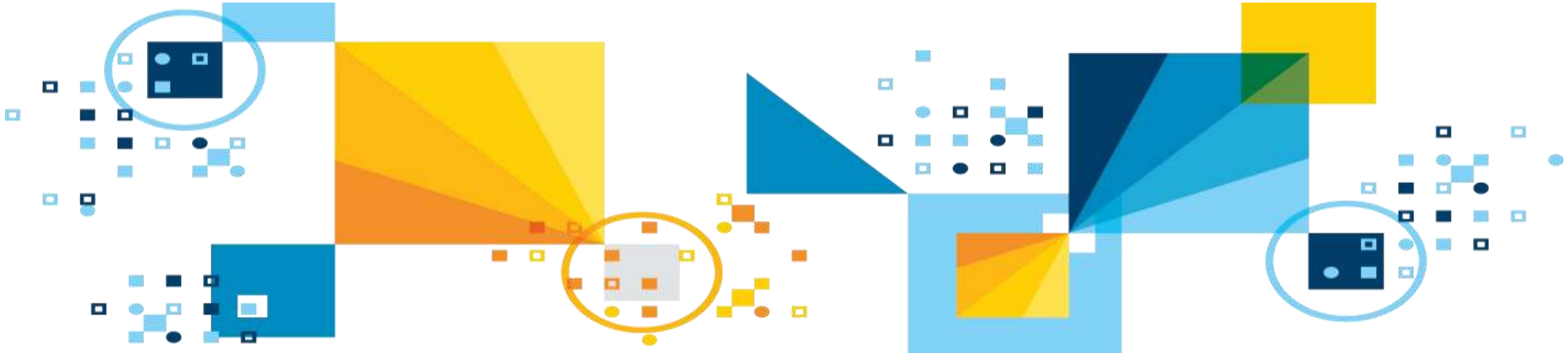
- Constanza Caorsi: caorsico@ar.ibm.com

- **IBM Red Books:**

- <http://www.redbooks.ibm.com/>

AKA: a.k.a.: also know as: también conocido como

Hadoop



¿Qué es Hadoop?



Doug Cutting



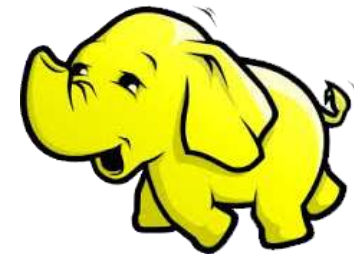
2005: Doug Cutting and Michael J. Cafarella developed Hadoop to support distribution for the [Nutch](#) search engine project.



The project was funded by Yahoo.



2006: Yahoo gave the project to Apache Software Foundation.



¿Qué es Hadoop?

2003

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*



2004

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com

Google, Inc.



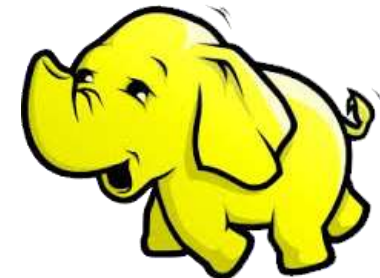
2006

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach,
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
(fay.jeff.sanjay.wilson@kccr.in.%,tushar.fikes.gruber}@google.com)

Google, Inc.

Abstract
Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large number of nodes. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Fi- c. These applications place very different demands on Bigtable, both in terms of data size (from URLs to terabytes of data) and in terms of access patterns. Bigtable achieves scalability and high performance, but Bigtable provides a different interface than such systems. Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of data represented in the underlying storage. Data is accessed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted st-



¿Qué es Hadoop?



- **Apache Hadoop es un framework de código abierto para procesamiento distribuido, confiable y escalable de grandes volúmenes de datos**

- Está pensado para ocultar al usuario la complejidad del cluster
- Está desarrollado completamente en [Java](#)

- **Consiste en 3 subproyectos básicos:**

- MapReduce
- Hadoop Distributed File System (a.k.a. HDFS)
- Hadoop Common

- **Y complementado por varios componentes del ecosistema:**

- Hbase
- Zookeeper
- Pig
- Etc.



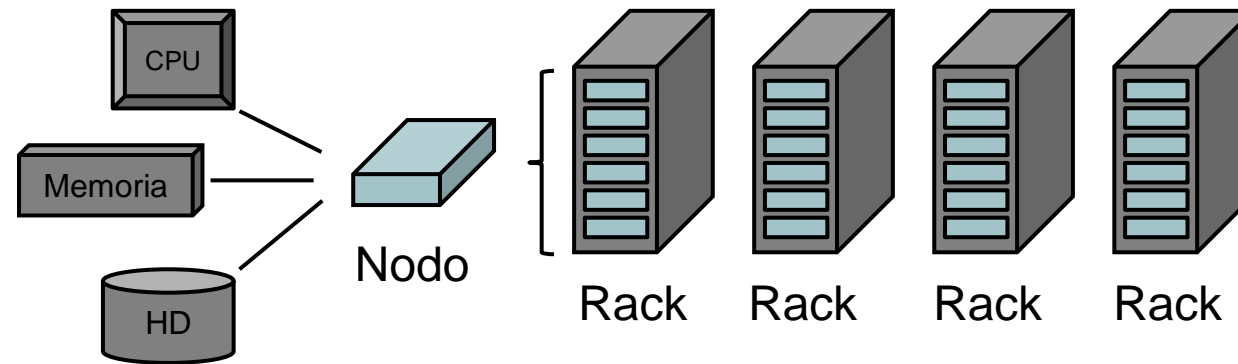
The Apache Software
Foundation

- **Pensado para correr en HW económico (“commodity hardware”)**

Hadoop - Arquitectura

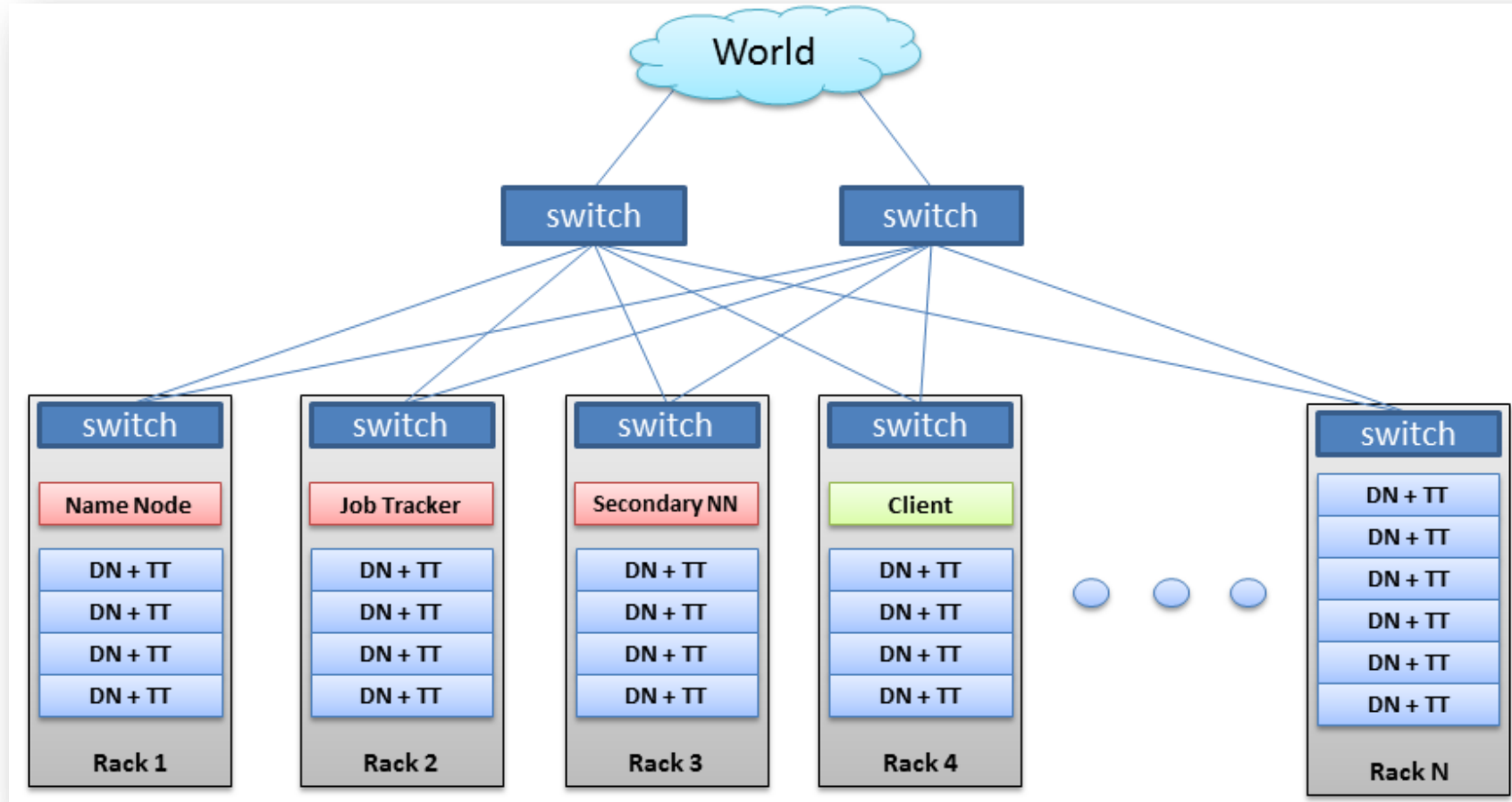


Cluster



- Hardware simple y económico
- Grandes volúmenes de datos, procesados localmente
- Alta velocidad de procesamiento para el almacenamiento

Hadoop - Arquitectura

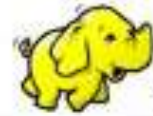


Hadoop - Características

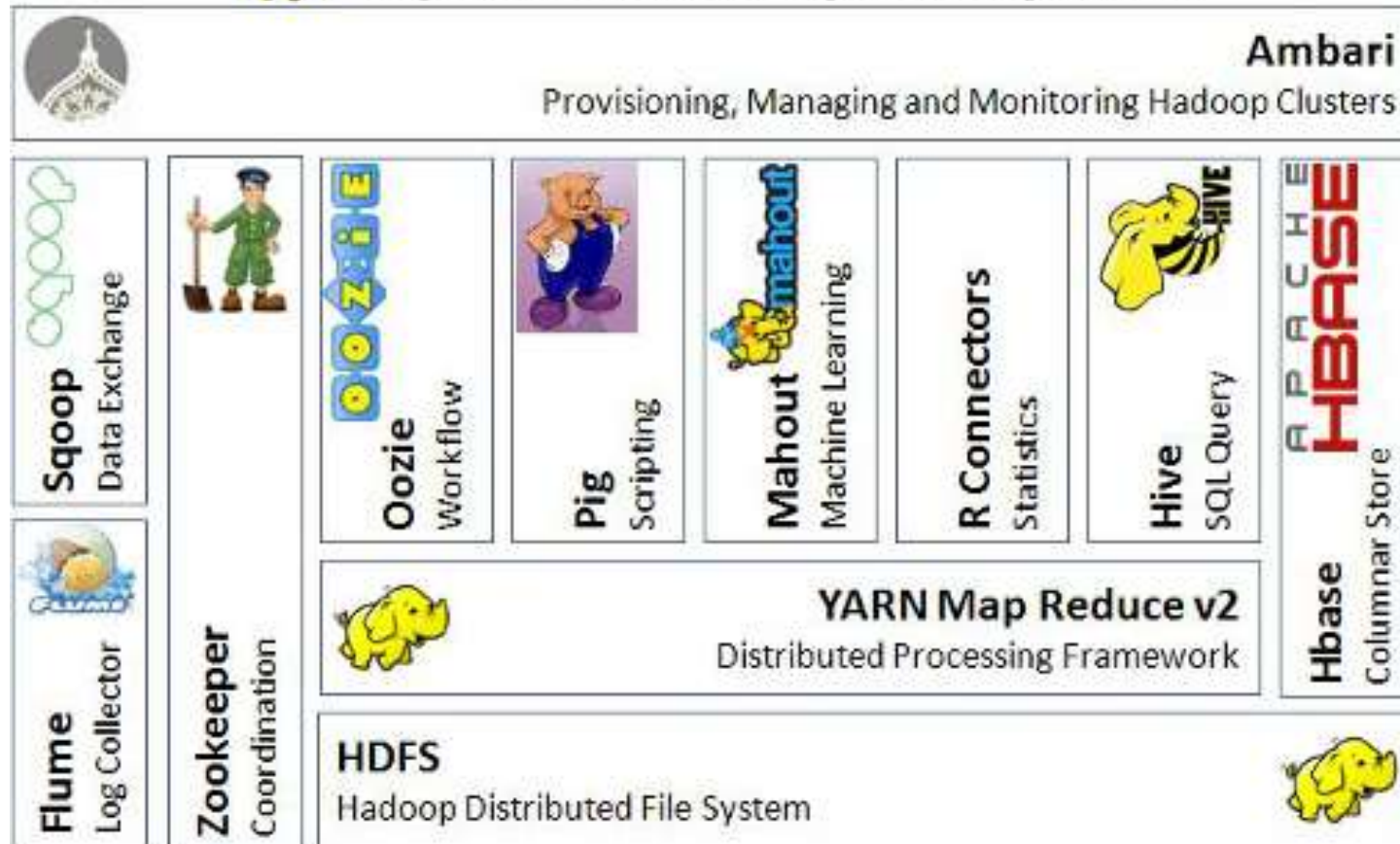


- **Escalabilidad**
 - Facilidad para agregar nuevos nodos
- **Bajo costo**
 - Hardware convencional
- **Flexible**
 - Datos no estructurados
- **Tolerancia a fallos**
 - Al caer un nodo, otros toman su trabajo

Ecosistema Hadoop



Apache Hadoop Ecosystem

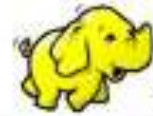


Ecosistema Hadoop

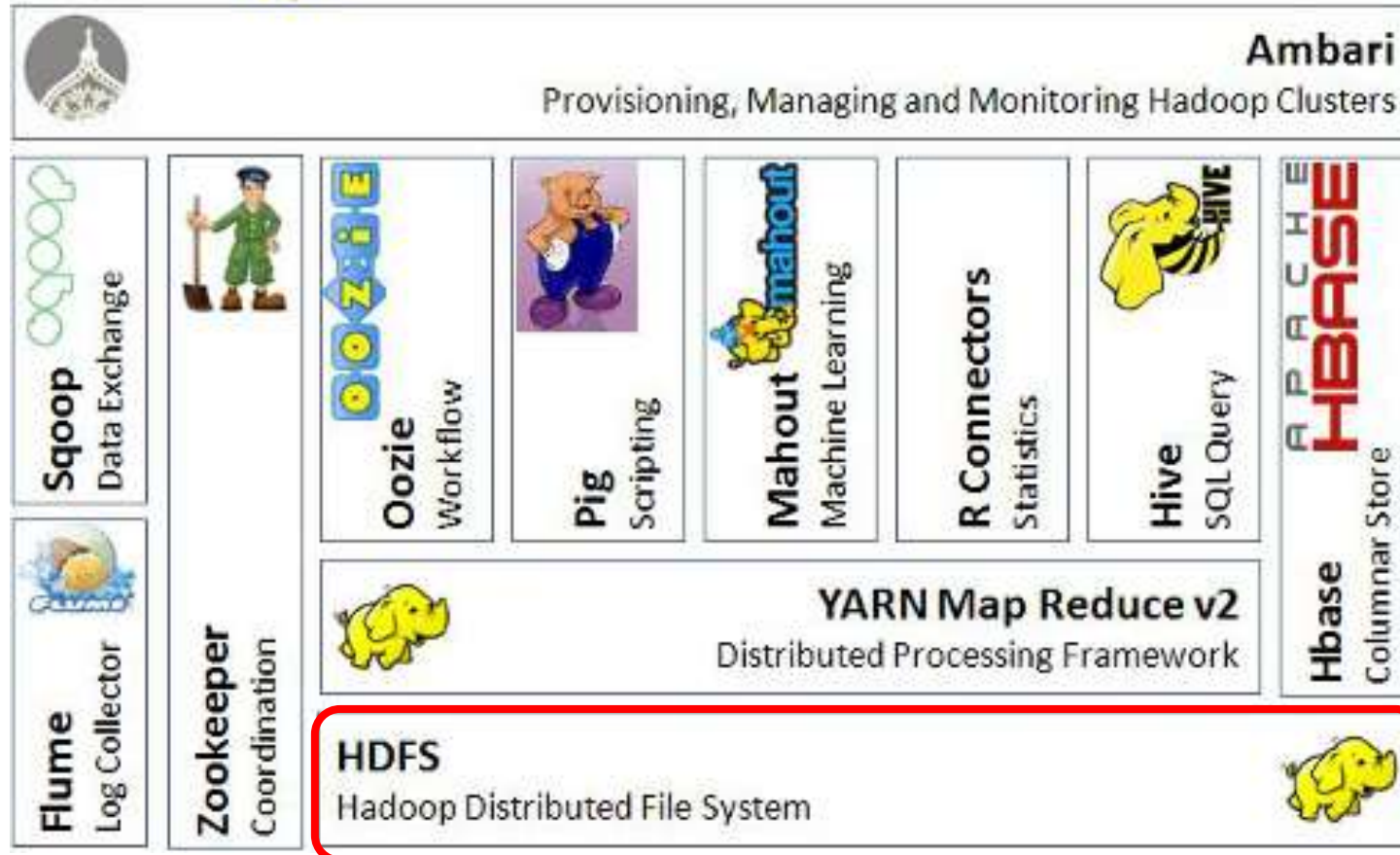


- **HDFS:** Hadoop Distributed File System.
- **YARN Map Reduce:** Plataforma homogénea de procesamiento de datos distribuidos mediante operaciones Map-Reduce.
- **Pig:** Plataforma de alto nivel para ejecutar operaciones sobre datos de forma distribuida, implementando operaciones Map-reduce mediante comandos simples.
- **Hive:** Plataforma de alto nivel que permite ejecutar operaciones simil SQL sobre datos distribuidos. Pensado para Data Warehouse.
- **Mahout:** Componente que se para sobre YARN Map Reduce para ejecutar operaciones de Machine Learning sobre datos distribuidos.
- **Oozie:** Herramienta de scheduling de jobs de Map Reduce, utilizada para coordinar la ejecución de varias operaciones Map Reduce.
- **HBase:** Base de datos NoSQL de tipo columnar, basada en HDFS.
- **Sqoop:** Servicio que actúa como interfaz entre los datos guardados en clúster de Hadoop y el mundo de bases de datos relacional.
- **Flume:** Servicio pensado para la consolidación de información de log en una plataforma de datos distribuida.
- **Zoo Keeper:** Sistema de coordinación para sistemas distribuidos.

HDFS



Apache Hadoop Ecosystem



Hadoop Distributed File System



- **Almacenamiento Distribuído**
- **Ideal para almacenar grandes archivos**
- **Modelo de acceso:**
 - una escritura
 - muchas lecturas
- **Datos replicados**

Hadoop Distributed File System



Falla del hardware

La falla en el hardware es más la norma que la excepción.

Una instancia de HDFS puede consistir en cientos o miles de servidores, cada uno guardando parte de los datos del file system.

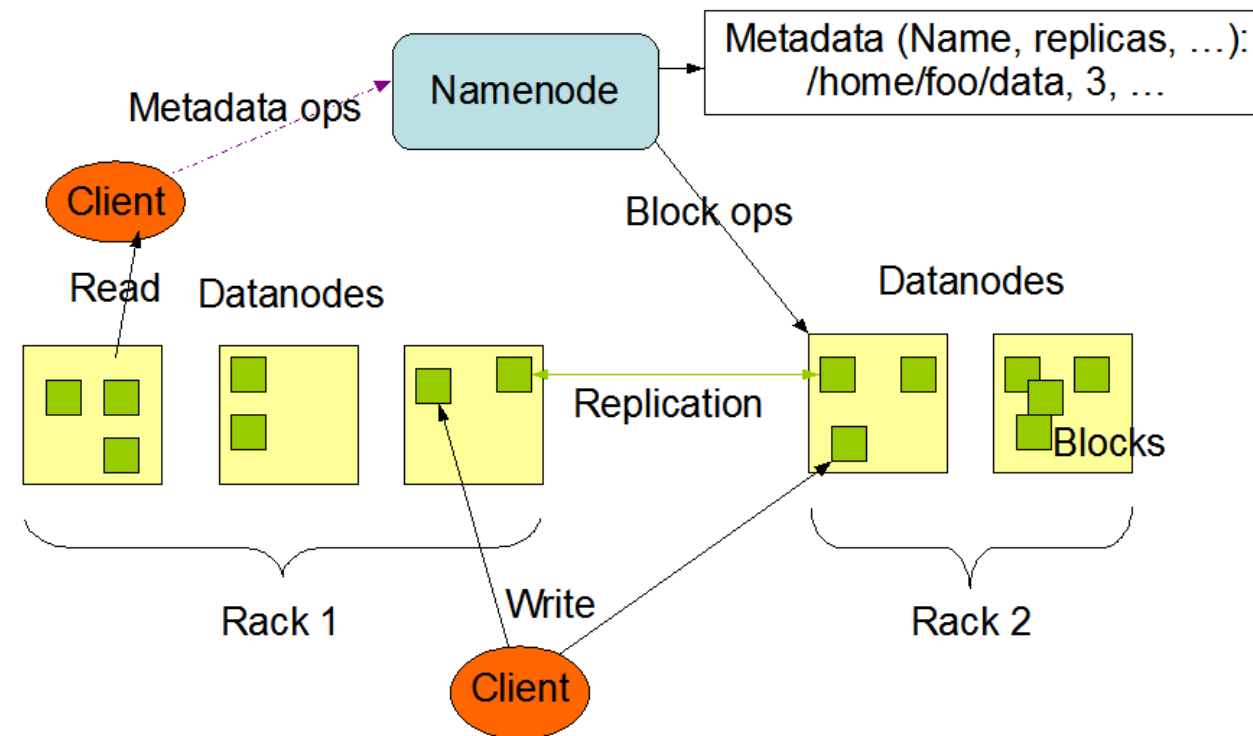
El hecho de que exista un gran número de componentes de hardware propenso a fallas, se traduce en que en algún momento existirá un componente del HDFS fuera de funcionamiento.

Por lo tanto, la detección de fallas y una recuperación rápida y automática es uno de los objetivos principales de HDFS.

Hadoop Distributed File System



HDFS Architecture



“Mover el procesamiento es más conveniente que mover los datos”

Hadoop Distributed File System



- HDFS presenta una arquitectura maestro/esclavo.
- Un cluster de HDFS consiste en un único *Namenode*, server maestro que administra el *namespace* del file system y regula el acceso a los archivos por clientes.
- Además, hay un número de *Datanodes*, en general uno por nodo en el cluster, que administran el almacenamiento de los nodos en los que corren.
- HDFS expone un *namespace* de file system y permite guardar datos de usuario en archivos.
- Internamente los archivos son divididos en uno o más bloques y esos bloques son almacenados en un conjunto de *Datanodes*.

Hadoop Distributed File System



- El *namenode* ejecuta operaciones de *namespace* del file system como apertura, cierre, renombramiento y directorios.
- También determina la asociación de bloques a *Datanodes*.
- Los *Datanodes* son responsables de atender los pedidos de lectura y escrita por parte de los clientes del file system.
- Los *Datanodes* también crean, eliminan y replican bloques a pedido del *Namenode*.

Hadoop Distributed File System



- Los *namenode* y *datanode* son productos de software diseñados para correr en máquinas básicas, usualmente bajo un sistema operativo GNU/Linux.
- HDFS fue desarrollado en Java; cualquier máquina que soporte el lenguaje puede ejecutar el software de *Namenode* o *Datanode*.
- Una implementación típica tiene una máquina dedicada para ejecutar el software de *Namenode* y el resto de máquinas en el cluster corre una instancia del software de *Datanode*.
- La arquitectura no imposibilita ejecutar múltiples *datanodes* en una misma máquina, pero en una implementación real no suele ser visto.
- La arquitectura es simplificada mediante la existencia de un único *namenode*, quien es el gestor y repositorio para toda la metadata del HDFS.
- El sistema se encuentra diseñado de tal forma que los datos de usuario nunca pasan a través del *namenode*.

Hadoop Distributed File System



● **Namenode**

- Server maestro que administra el file system
- Gestiona los directorios y la ubicación de los archivos en Datanodes
- Determina asociación de bloques de archivos a Datanodes y gestiona sus réplicas
- Provee a los clientes la lista de Datanodes que poseen el archivo buscado
- No almacena los datos de los archivos
- Es un punto único de fallo

Hadoop Distributed File System



● Datanode

- Atienden los pedidos lectura y escritura por parte de clientes
- Envía periódicamente un *heartbeat* al Namenode para indicar actividad
- Envía periódicamente un reporte de bloques al Namenode
- Los datanodes pueden comunicarse entre sí

Hadoop Distributed File System



Réplica de datos

- HDFS está diseñado para almacenar de forma confiable un gran número de archivos a través de máquinas de un gran cluster.
- Guarda los archivos como una secuencia de bloques; todos los bloques de un archivo tienen el mismo tamaño con excepción del último.
- Los bloques de archivos son replicados para proveer tolerancia a fallos.
- El tamaño de bloque y el factor de réplica se pueden configurar mediante un archivo. Dicho factor puede especificarse al momento de crear un archivo y ser modificado con posterioridad.
- Los archivos en HDFS son escritos una sola vez y admiten únicamente una escritura a la vez.
- El *namenode* toma todas las decisiones relacionadas con la réplica de bloques y recibe periódicamente de cada *datanode* en el cluster un *heartbeat* (señal que indica actividad) y un reporte de bloques.
- La recepción de un *heartbeat* indica que el *datanode* sigue funcionando correctamente y el reporte de bloques contiene una lista de todos los bloques de un *datanode*.

Hadoop Distributed File System



Punto único de fallos

- El *namenode* en Hadoop 1.x era un punto único de fallo. Desde la versión 2.x incluyen algunas mejoras que permiten configurarlo para reducir el impacto de un fallo.
- Desde la versión 2.x Pueden configurarse dos *namenodes* donde uno es el maestro y el otro esclavo. El esclavo funciona en modo stand by copiando los datos del master. En caso de fallar el nodo maestro, el esclavo puede tomar el rol de maestro. Esto puede realizarse de forma manual o automática.
- La copia de datos del maestro al esclavo se realiza a través de un log compartido de modificaciones. Este log compartido debe estar almacenado en un NFS (Network File System) compartido por ambos nodos. Esto significa que el NFS se convierte en un nuevo punto único de fallos, ya que si este falla el nodo esclavo no puede leer el log y por lo tanto no puede sincronizarse con el nodo maestro.
- En versiones más nuevas de HDFS existe la posibilidad de usar un nuevo componente llamado Quorum Journal Manager, que reemplaza el NFS con un sistema de logeo de transacciones distribuido.

Hadoop Distributed File System



Reporte de bloques

- El reporte de bloques es importante ya que ésta es la única información que tiene el *namenode* para conocer el estado real de los nodos.
- Cuando un cliente quiere escribir datos le pide al *namenode* la lista de *Datanodes* donde deberían escribirse los bloques del archivo.
- Luego los clientes escriben directamente a los *datanodes* pero nunca reportan al *namenode* lo escrito, por esto es importante que los *datanodes* envíen el reporte de bloques periódicamente.
- Los *datanodes* que reciben los bloques se encargan de replicarlos al resto de los nodes de acuerdo al factor de replicación.

Hadoop Distributed File System



● Heartbeat

- Mecanismo por el cual el namenode conoce el estado de los nodos disponibles
- En caso de no recibirse el heartbeat de un nodo, se lo marca como caído

● Réplica

- Una nueva réplica puede ser necesaria si:
 - Un datanode no está disponible
 - Réplica incorrecta / Fallo en disco
 - Se incrementa el factor de réplica

Hadoop Distributed File System



- **Falla de datos en el disco, Heartbeats y Re-Réplica**
 - Cada *datanode* envía periódicamente un mensaje de *heartbeat* al *namenode*.
 - Una falla de red puede causar que un grupo de *datanodes* pierda conectividad con el *namenode*, quien puede detectar esta condición gracias a la ausencia de un mensaje de *heartbeat*.
 - Entonces, el *namenode* marca el *datanode* como caído y no le envía nuevos pedidos de E/S.
 - Todos los datos registrados en el *datanode* caído ya no estará disponible en el HDFS.

Hadoop Distributed File System



- **Falla de datos en el disco, Heartbeats y Re-Réplica**
 - La caída de un *datanode* puede causar que el factor de réplica de un bloque caiga del valor definido.
 - El *namenode* hace un seguimiento constante de los bloques que necesitan ser replicados e inicializa la réplica en cuanto es necesario.
 - La necesidad de una re-réplica puede deberse a varias razones: un *datanode* puede no estar disponible, una réplica puede corromperse, un disco rígido de *datanode* puede fallar o el factor de réplica de un archivo puede haber sido incrementado.

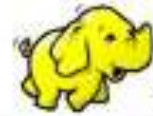
Hadoop Distributed File System



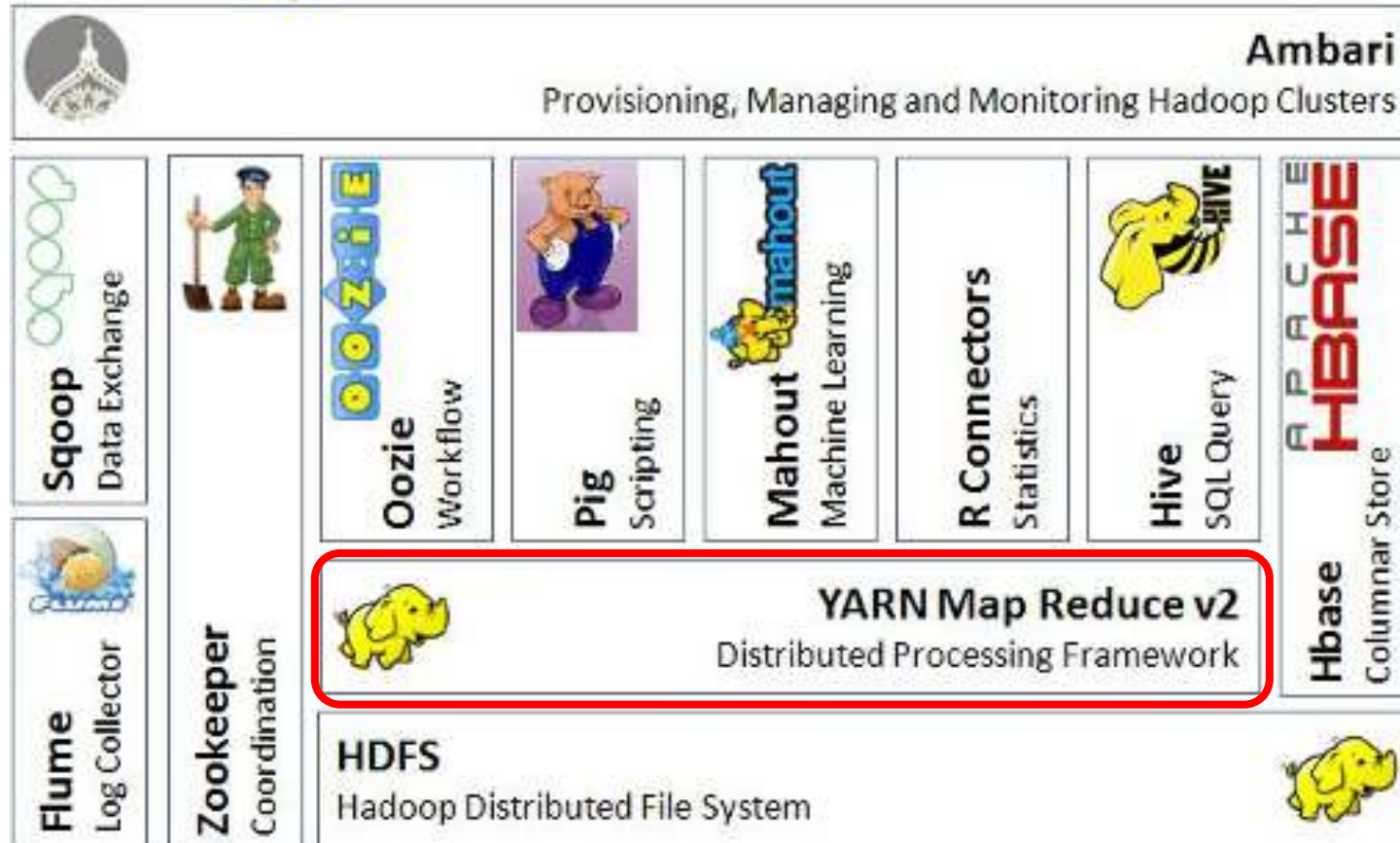
- **Alta y baja de nodos**

- Para dar de alta nodos nuevos es necesario agregarlos en un archivo de configuración utilizado por el *Namenode* (`$HADOOP_PREFIX/etc/hadoop/slaves`). Dependiendo de la versión de hadoop este archivo puede cambiar.
- Cuando se dan de alta nodos nuevos estos automáticamente se empiezan a utilizar para almacenar datos nuevos, sin embargo es posible que estos nodos se estén usando por debajo de su capacidad.
- Una manera de solucionar esto es utilizando el *HDFS Balancer* que se encarga de redistribuir la carga sobre los nodos. Esta situación se puede producir también si se reactivan nodos que estuvieron inactivos durante un tiempo.

YARN + Map Reduce v2



Apache Hadoop Ecosystem



YARN + Map Reduce v2

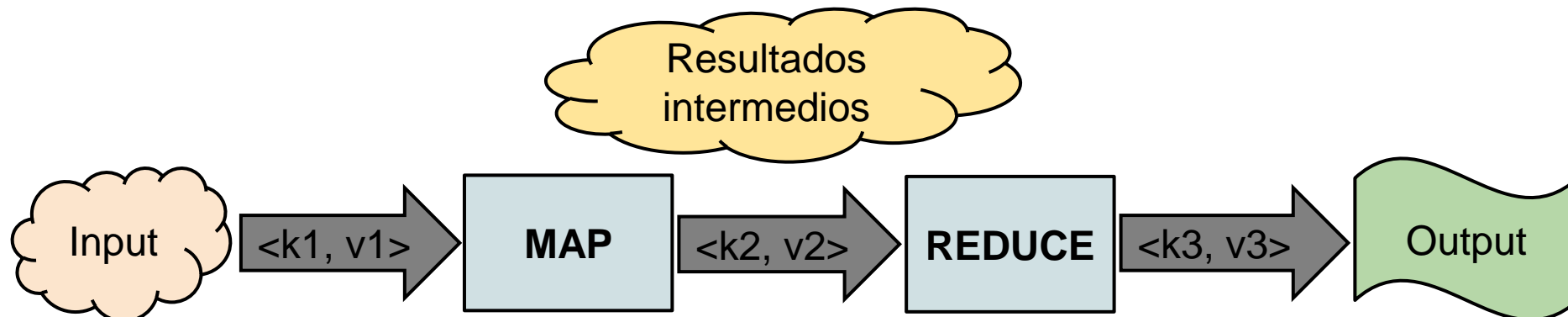


- **Hadoop YARN:** Plataforma de manejo de recursos distribuidos.
- **Hadoop Map Reduce:** Plataforma de procesamiento de datos distribuidos mediante operaciones Map-Reduce.

Hadoop - Map-Reduce



- **Función Map**
 - Procesa pares de clave-valor, generando un grupo de claves-valores intermedios.
- **Función Reduce**
 - Une y combina todos los valores intermedios asociados a la misma clave intermedia.



Hadoop - Map-Reduce



Map-Reduce

- El framework Hadoop Map-Reduce permite desarrollar aplicaciones para procesar grandes volúmenes de datos (>varios TB) en paralelo.
- Este procesamiento, confiable y con tolerancia a fallos, se realiza en grandes clusters (cientos de nodos) compuestos por hardware económico.
- Un trabajo de Map-Reduce generalmente separa los datos de entrada en *chunks* independientes que son procesados por las *tareas Map* en paralelo.
- El framework ordena las salidas del mapa, las cuales se convierten en datos de entrada de la *tarea Reduce*.
- Normalmente tanto la entrada como la salida de un trabajo son almacenadas en un *filesystem*.
- El framework se encarga de planificar tareas, monitorearlas y volver a ejecutar las tareas que fallaron.

Hadoop - Map-Reduce



Map-Reduce

- En general los nodos que almacenan y procesan son los mismos, es decir que el framework Map-Reduce y el *filesystem* distribuido ejecutan en el mismo conjunto de nodos.
- Esta configuración permite al framework planificar efectivamente las tareas en nodos donde se almacena la información, optimizando el ancho de banda a lo largo del cluster.
- Las aplicaciones, como mínimo, especifican la ubicación de las entradas/salidas y proveen funciones *Map* y *Reduce* mediante implementaciones de interfaces y/o clases abstractas.
- Éstas, y otros parámetros de trabajo componen la *configuración de un Job*.

Hadoop - Map-Reduce



Map-Reduce

- Aunque Hadoop está implementado en Java, las aplicaciones Map-Reduce pueden no estar escritas en Java.
- [Hadoop Streaming](#) es una utilidad que permite crear y ejecutar *jobs* con cualquier ejecutable, sean Mapper o Reducer.
- [Hadoop Pipes](#) es una API que permite implementar aplicaciones Map-Reduce ([SWIG](#) / C++).

Hadoop - Map-Reduce



Entradas y Salidas

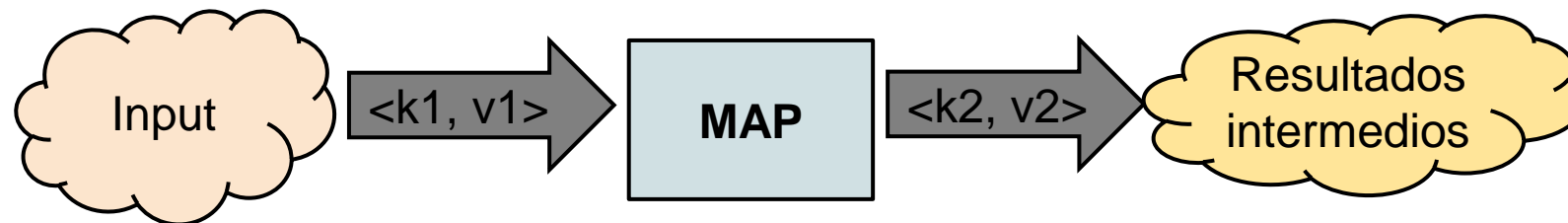
- El framework Map-Reduce opera exclusivamente con pares de clave/valor, es decir, las entradas de los trabajos vienen dadas por un conjunto de <clave, valor> y la salida del trabajo produce conjuntos de <clave, valor>.

Hadoop - Map-Reduce



Mapper

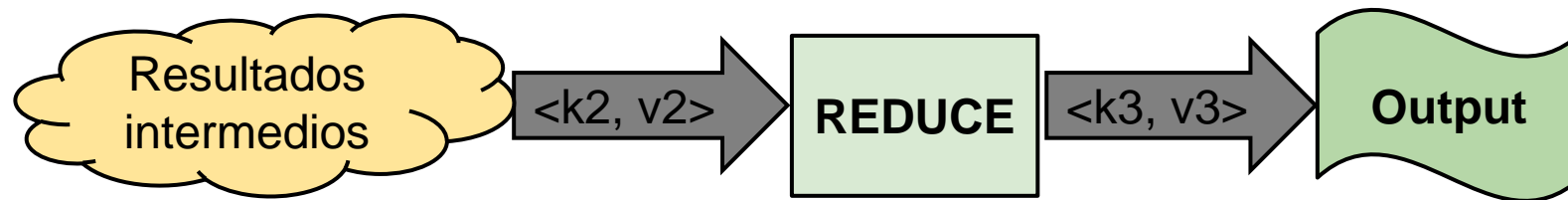
- Nodos {
- Realizan la misma tarea en paralelo
 - Procesan datos del mismo nodo o cercanos
 - Reciben la tarea a realizar y la ubicación del archivo a procesar



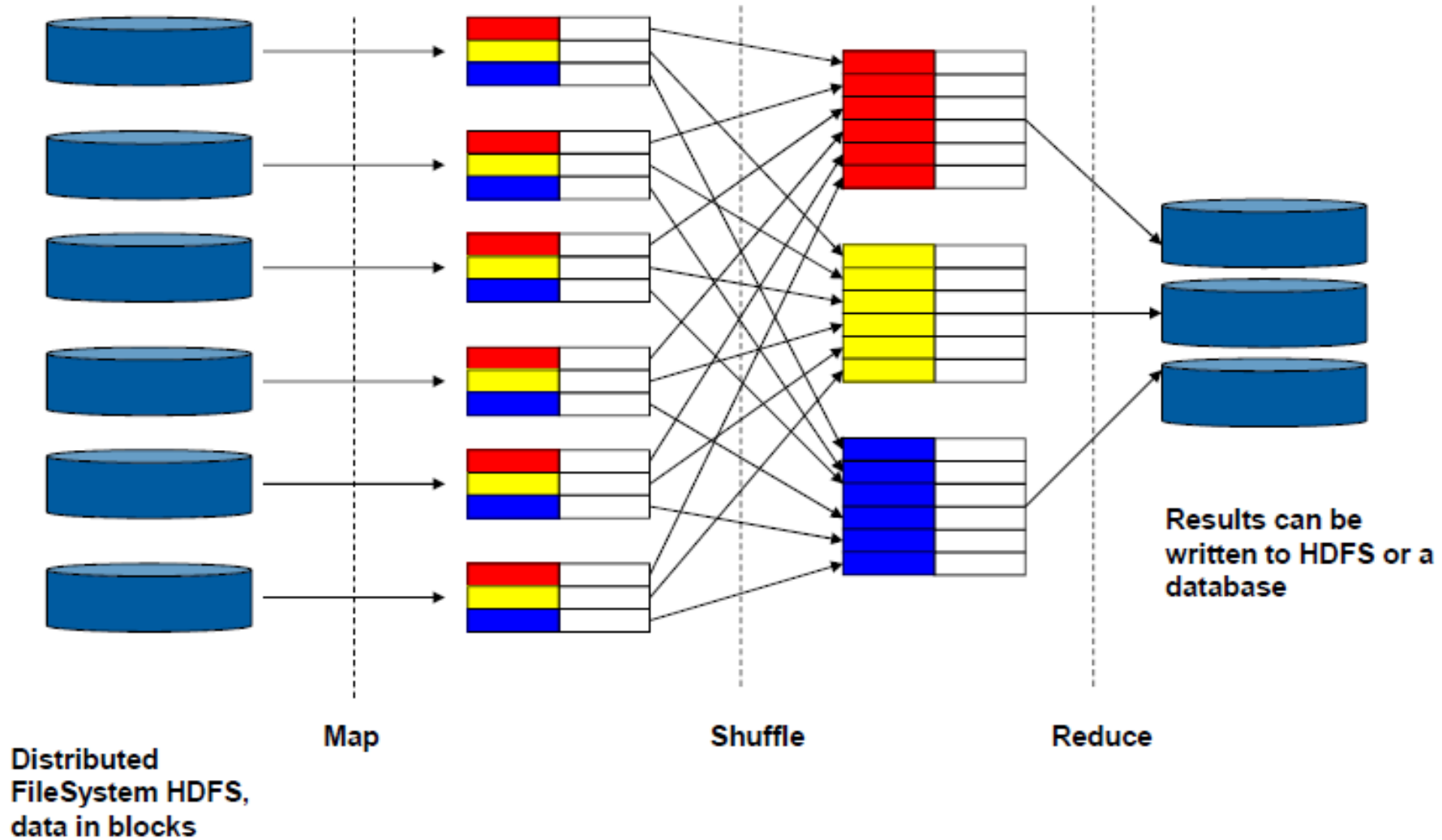
Hadoop - Map-Reduce



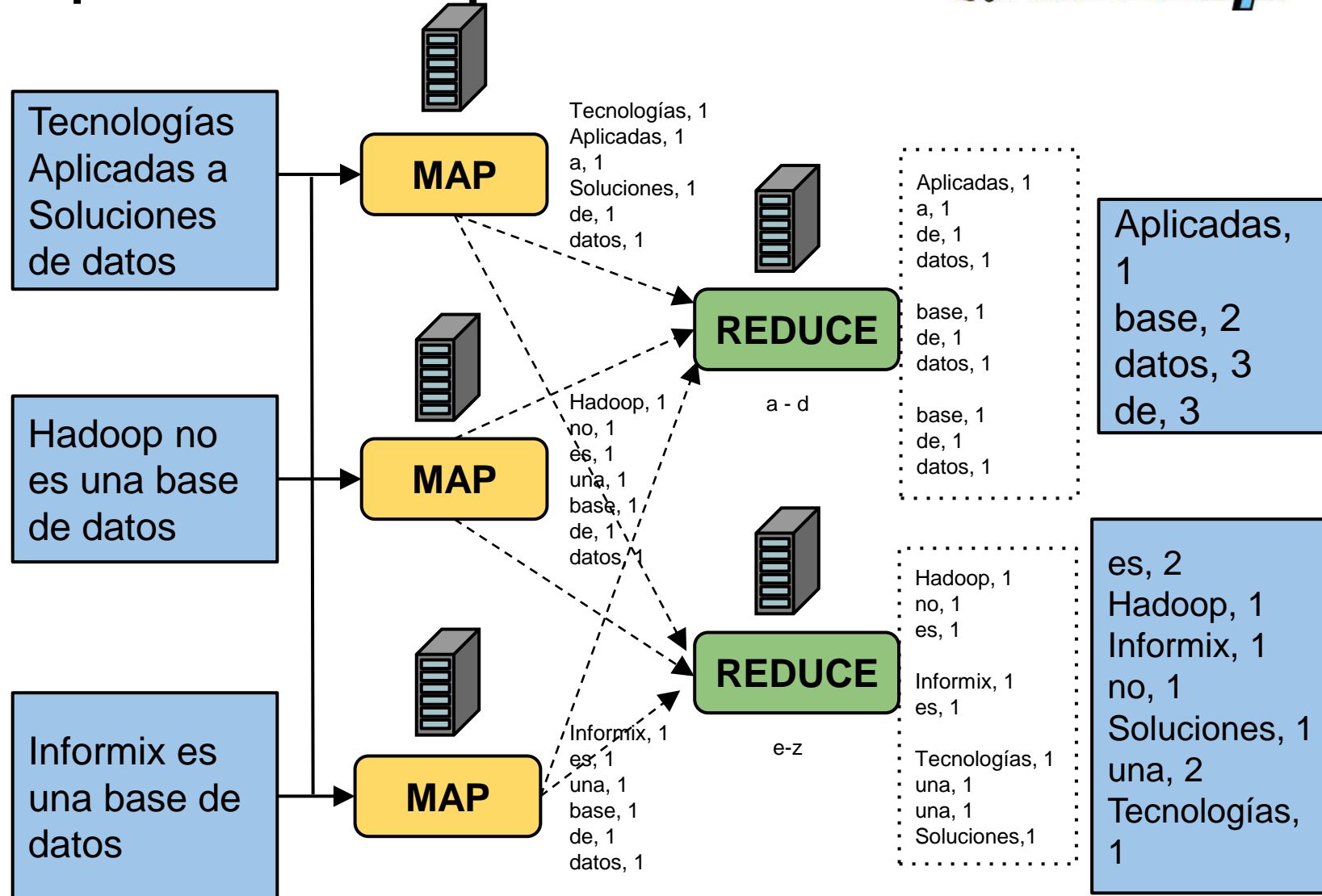
- Reducer
 - Recibe, unifica y ordena los resultados obtenidos por el Mapper
 - Procesa y reduce los resultados obtenidos por el Mapper



Hadoop - Map-Reduce Overview



Hadoop - Contador de palabras



Hadoop - Contador de palabras



Fases:

- **Map:** los mappers generan la salida.
- **Shuffle:** La salida de los mappers es repartida entre los distintos reducers de acuerdo a las claves.
- **Sort:** Los datos recibidos por cada reducer son agrupados por clave <key, [list of values]>.
- **Reduce:** La función de reduce es invocada una vez por cada clave, recibiendo la lista de todos los valores para esa clave.

Hadoop - Contador de palabras



- La función de reduce no puede invocarse hasta que todos los mappers hayan finalizado, ya que se necesita la lista completa de datos para cada clave.
- La salida final de los reducers se almacena en forma de archivos en el HDFS, un archivo por cada reducer.
- Ningún dato intermedio es almacenado en el HDFS, solo se utiliza la memoria y el file system local de cada nodo para manejar el flujo de datos de un dato a otro.

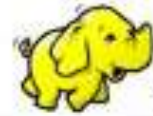
Hadoop - Contador de palabras



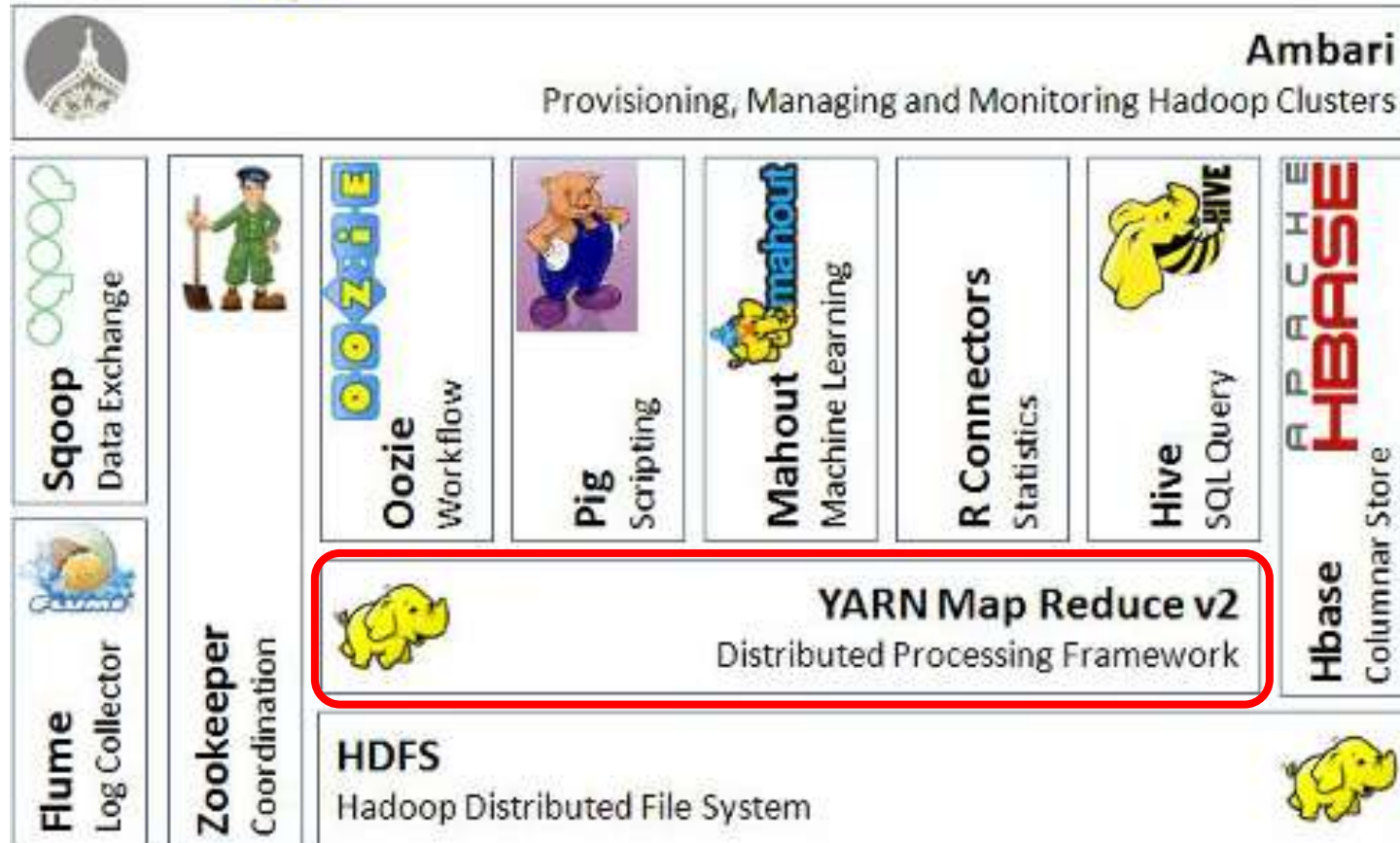
Formatos de entrada y salida:

- Las entradas y salidas de Hadoop se manejan a través de dos interfaces: RecordReader y RecordWriter.
- Estas interfaces permiten leer y escribir pares clave valor. Por default hadoop utiliza dos implementaciones que convierten leen y escriben los pares clave valor en líneas de texto.

YARN + Map Reduce v2



Apache Hadoop Ecosystem

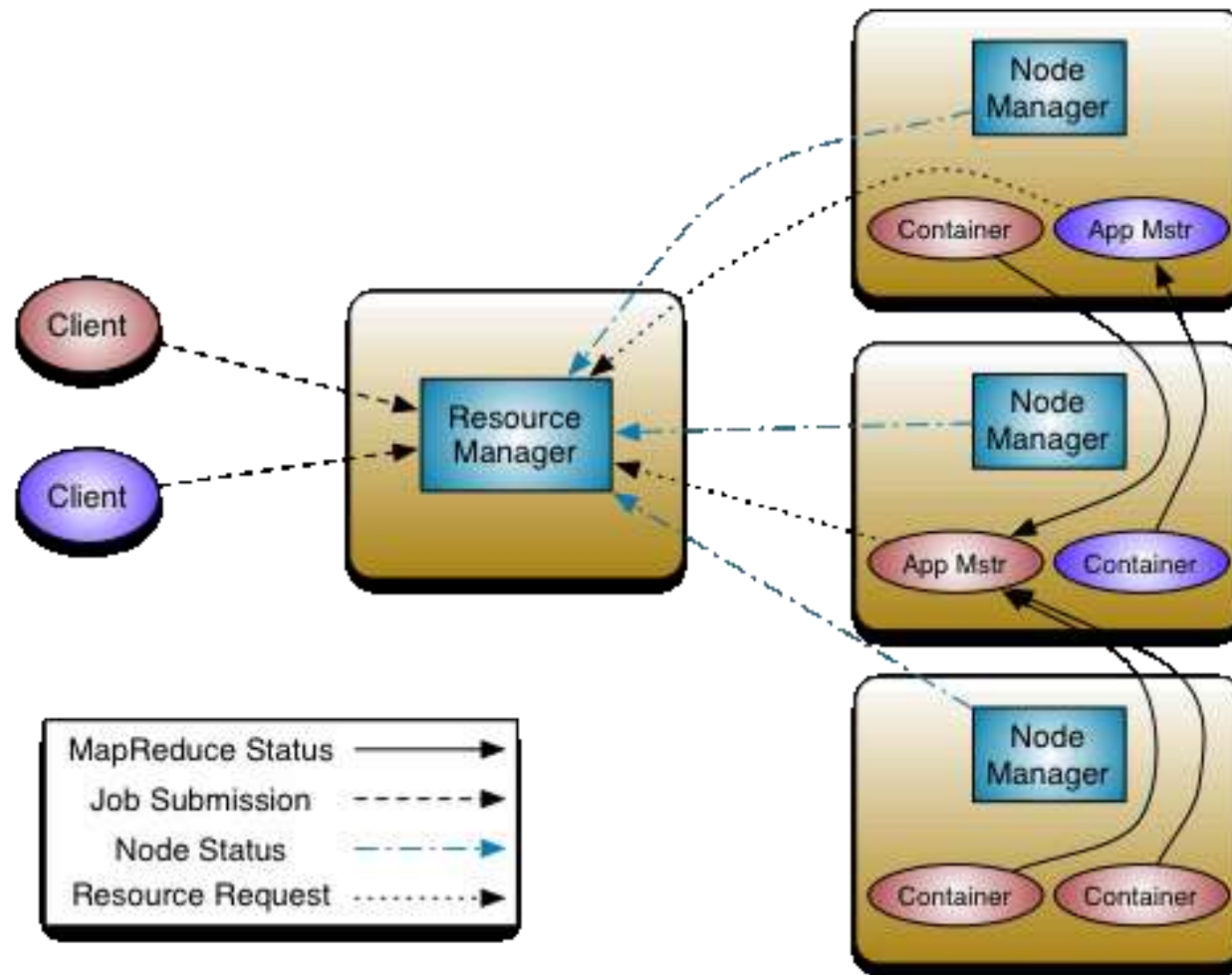


YARN + Map Reduce v2



- **Hadoop YARN:** Plataforma de manejo de recursos distribuidos.
- **Hadoop Map Reduce:** Plataforma de procesamiento de datos distribuidos mediante operaciones Map-Reduce.

Hadoop - YARN



Hadoop - YARN

- **ResourceManager:**
- Se encarga de asignar recursos (nodos) para la ejecución de trabajos.
- Existe un único ResourceManager para todo el cluster.
- Por cada trabajo a realizar asigna un ApplicationMaster y varios Containers a los distintos NodeManagers.

Hadoop - YARN

- **NodeManager:**
- Se encarga de administrar recursos para un nodo a pedido del ResourceManager.
- Existe un único NodeManager por cada nodo.

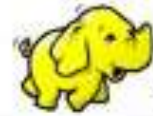
Hadoop - YARN

- **ApplicationMaster:**
- Se encarga de coordinar las tareas a realizar para un trabajo particular y solicitar recursos al ResourceManager.
- Existe un único ApplicationMaster por cada trabajo a realizar.
- Su ciclo de vida comienza y finaliza junto con el del trabajo a realizar.
- Al finalizar notifica de su finalización al ResourceManager el cual a su vez comunica al cliente que el trabajo fue finalizado.

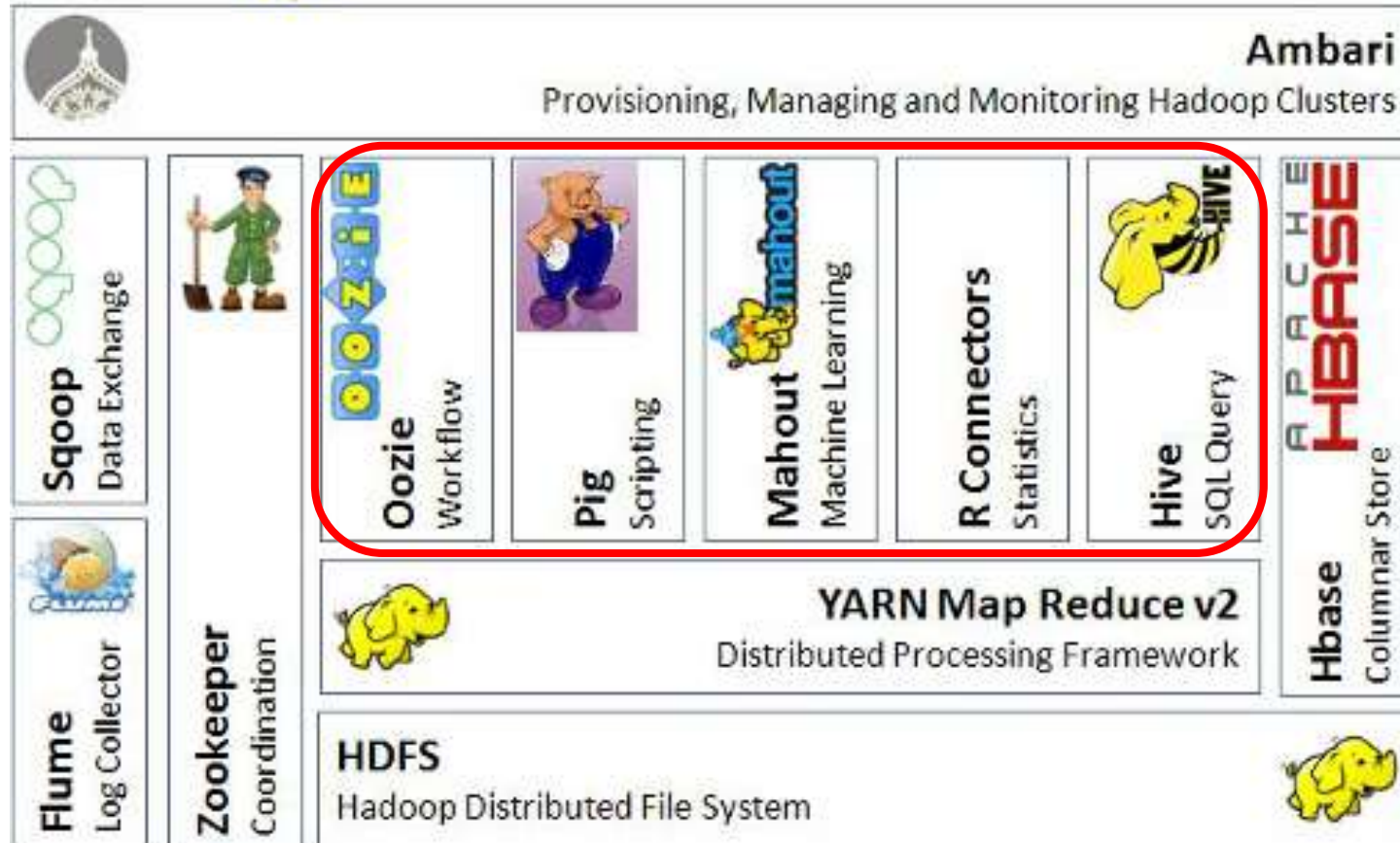
Hadoop - YARN

- **Container:**
- Cada container ejecuta una subtarea del trabajo a realizar.
- Pueden existir varios containers pertenecientes a un mismo trabajo dentro de un mismo nodo.
- Cada instancia de Container representa una asignación de recursos para esa tarea (procesador y memoria).
- Una vez finalizada la tarea notifica a su ApplicationMaster.

Herramientas de Alto Nivel



Apache Hadoop Ecosystem



Herramientas de Alto Nivel



- **Pig:** Plataforma de alto nivel para ejecutar operaciones sobre datos de forma distribuida, implementando operaciones Map-reduce mediante comandos simples.
- **Hive:** Plataforma de alto nivel que permite ejecutar operaciones simil SQL sobre datos distribuidos. Pensado para Data Warehouse.
- **Mahout:** componente que se para sobre YARN Map Reduce para ejecutar operaciones de Machine Learning sobre datos distribuidos.
- **Oozie:** herramienta de scheduling de jobs de Map Reduce, utilizada para coordinar la ejecución de varias operaciones Map Reduce.

Herramientas de Alto Nivel



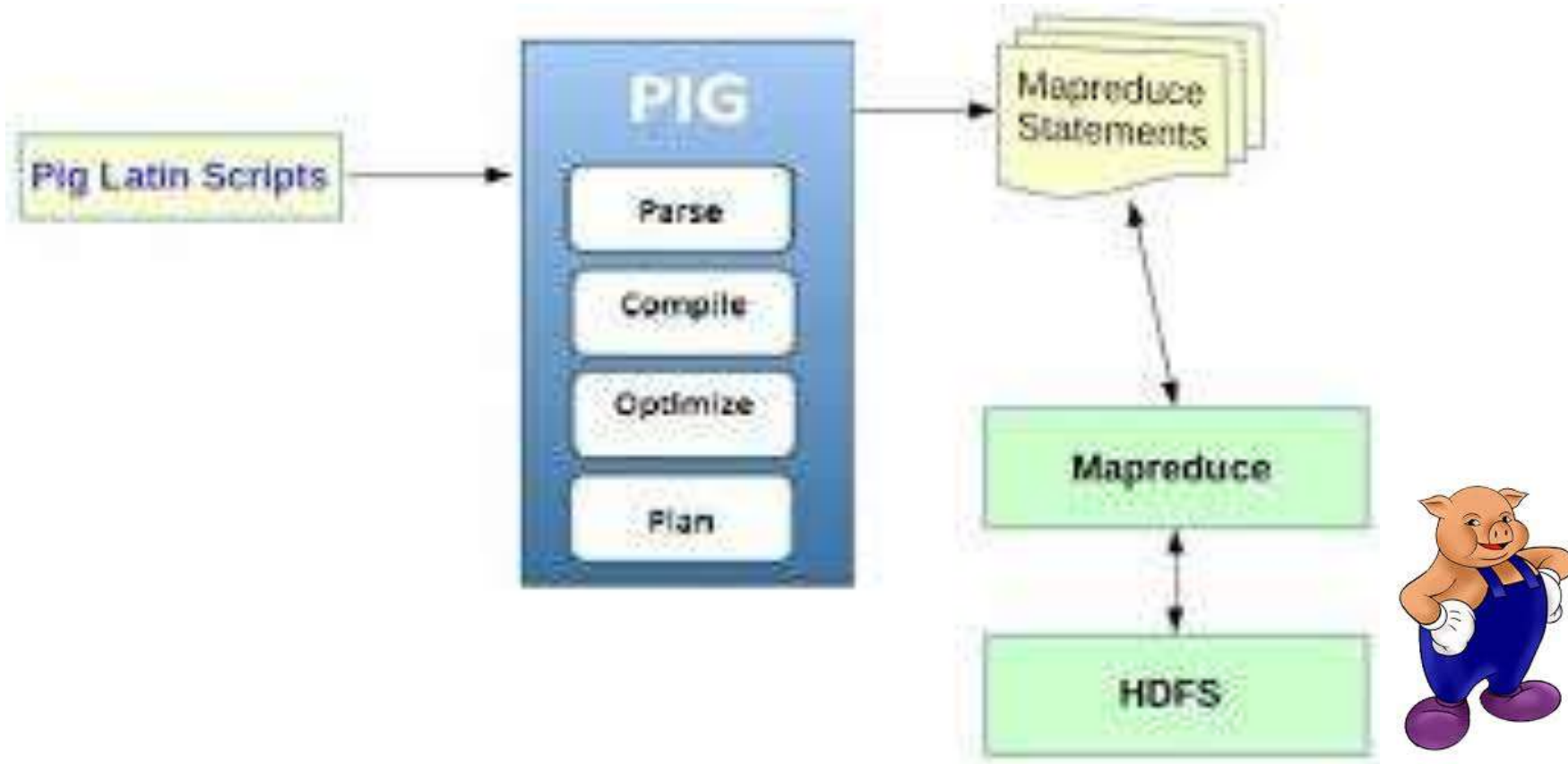
- **Sqoop**: Servicio que actúa como interfaz entre los datos guardados en clúster de Hadoop y el mundo de bases de datos relacional.
- **Flume**: Servicio pensado para la consolidación de información de log en una plataforma de datos distribuida.
- **Zoo Keeper**: sistema de coordinación para sistemas distribuidos.

Apache Pig

- Plataforma de alto nivel para ejecutar operaciones y consultas sobre datos de forma distribuida.
- Simplifica la creación de Jobs Map-Reduce sobre un clúster de Hadoop.
- Utilizar un lenguaje de scripting llamado Pig Latin.



Apache Pig - Flujo de Ejecución



Apache Hive

- Interfaz para realizar consultas tipo SQL sobre un cluster de Hadoop
- Se abstrae del almacenamiento distribuido utilizado y las operaciones Map Reduce
- Pensado para tener un Data Warehouse sobre datos distribuidos (en Hadoop)



Apache Hive - Características

Características de Hive:

- Orientado a Data Warehouse
- Es schema on-read
- Convierte queries simil SQL en jobs para YARN Map/Reduce
- HiveQL como lenguaje de consulta

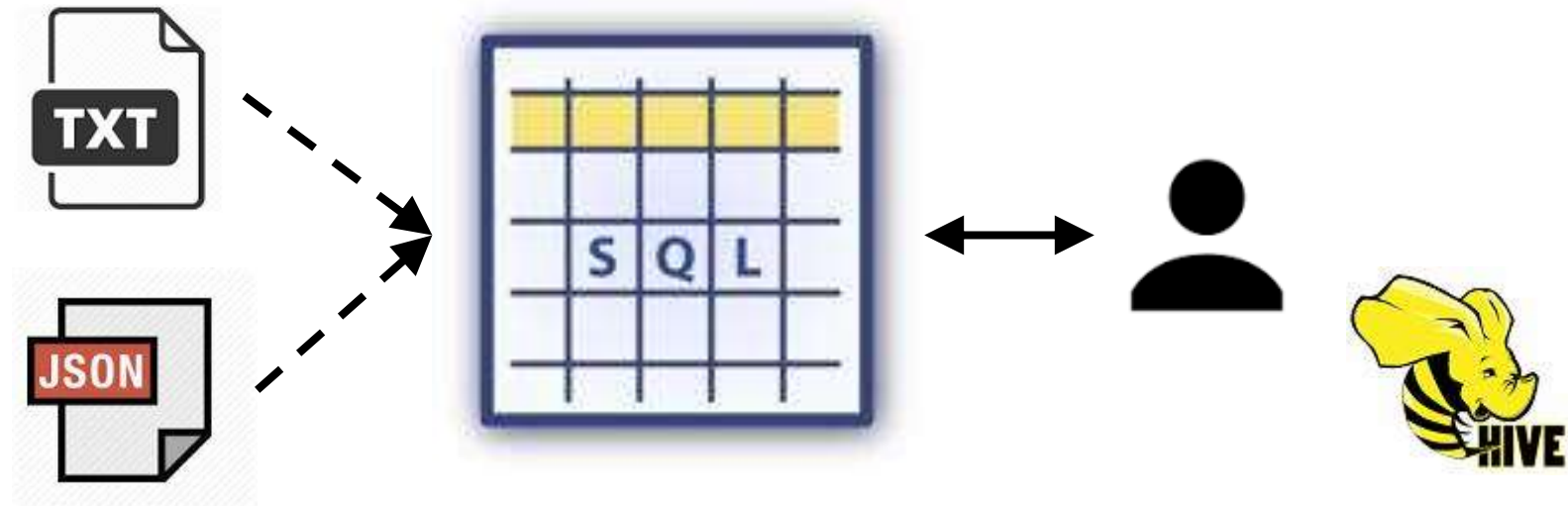
Hive NO está pensado para:

- Sistemas OLTP
- Operaciones en Tiempo Real
- Operaciones a nivel de registro

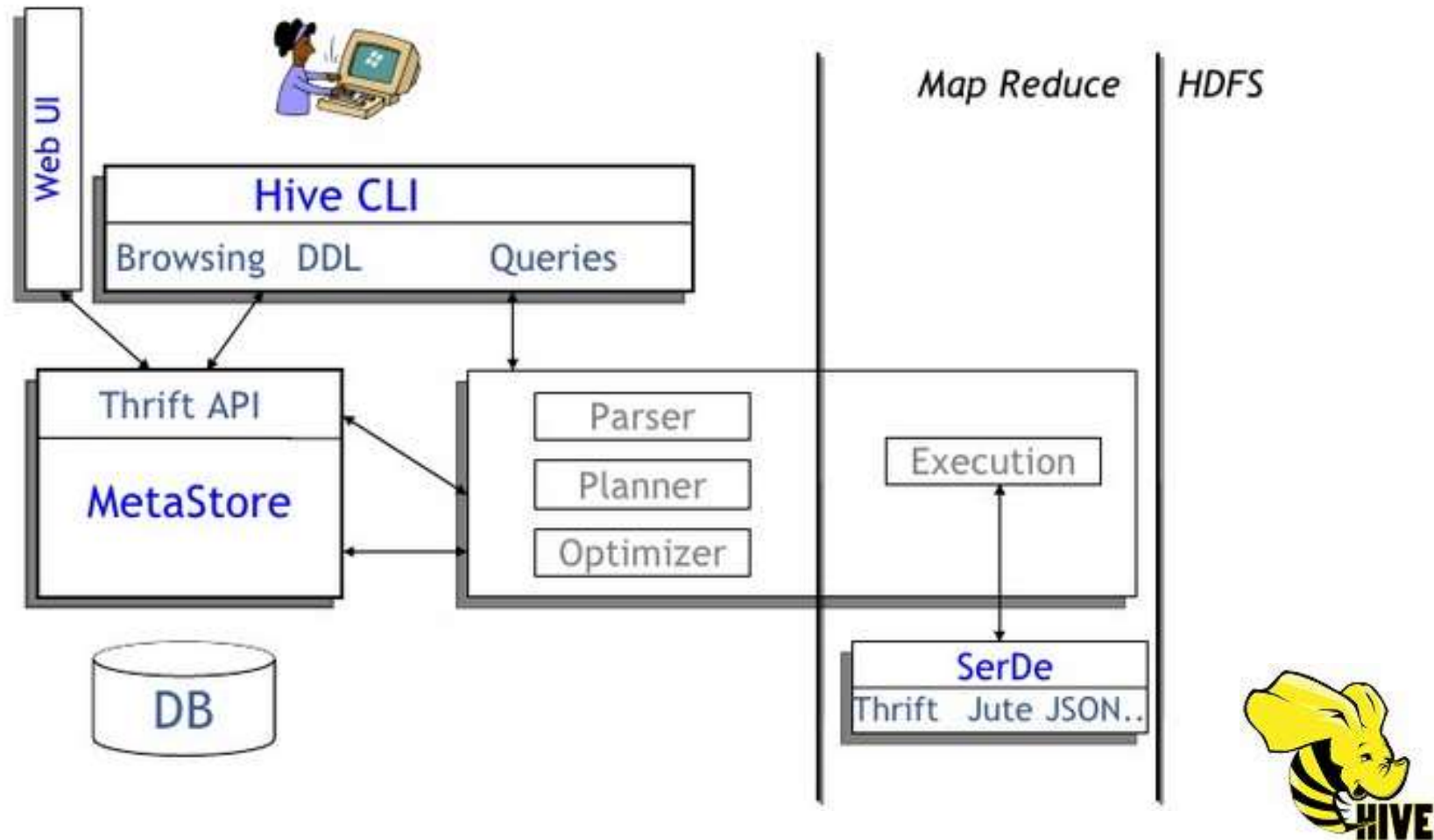


Apache Hive - Schema on Read

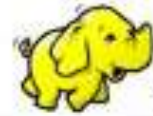
- En Hive, los pasos de Diseño de una tabla son:
 1. Se cargan los datos en HDFS
 2. Se define la estructura de la tabla para acceder a dichos datos
 3. Se leen los datos



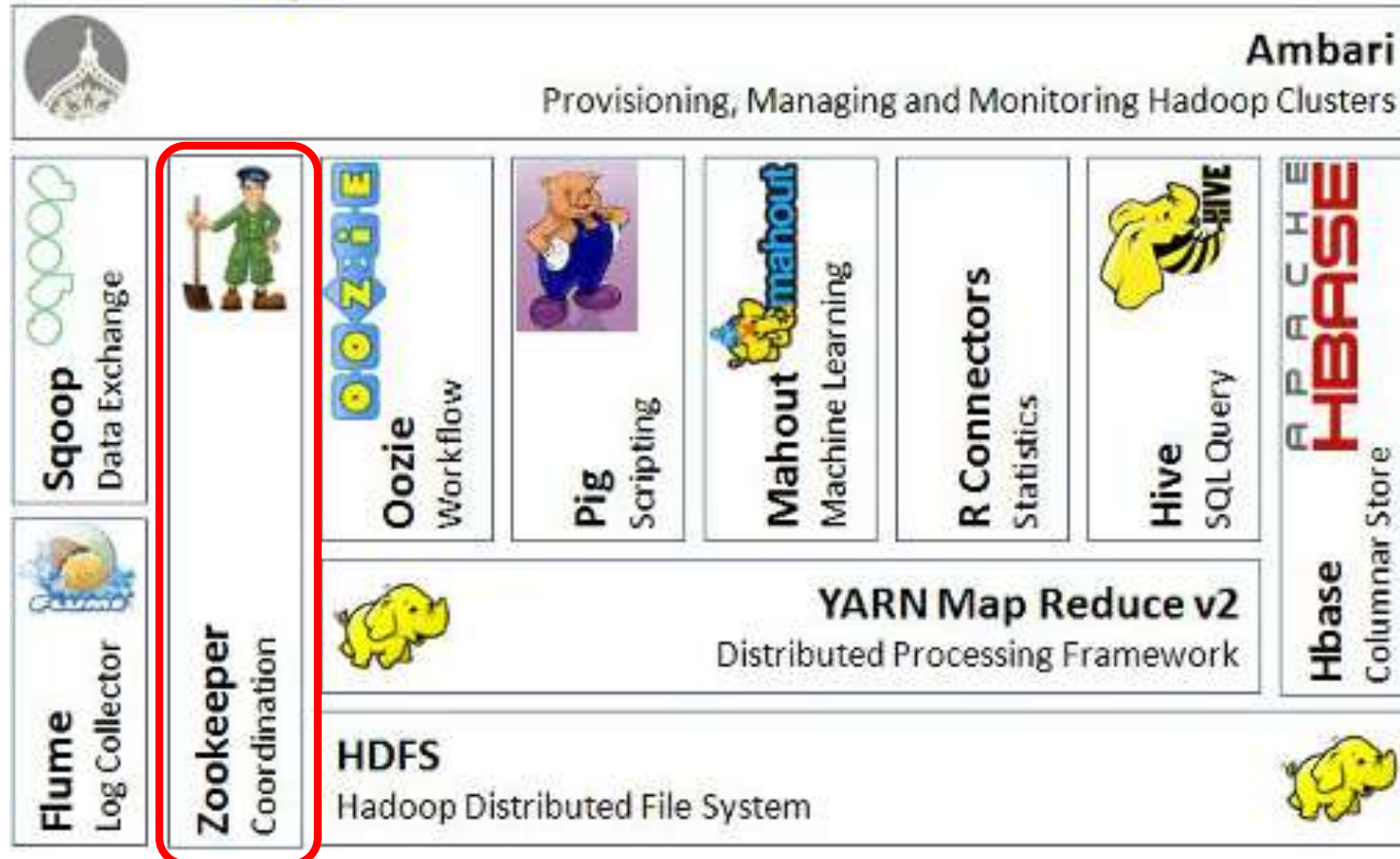
Apache Hive - Arquitectura



Apache Zookeeper



Apache Hadoop Ecosystem

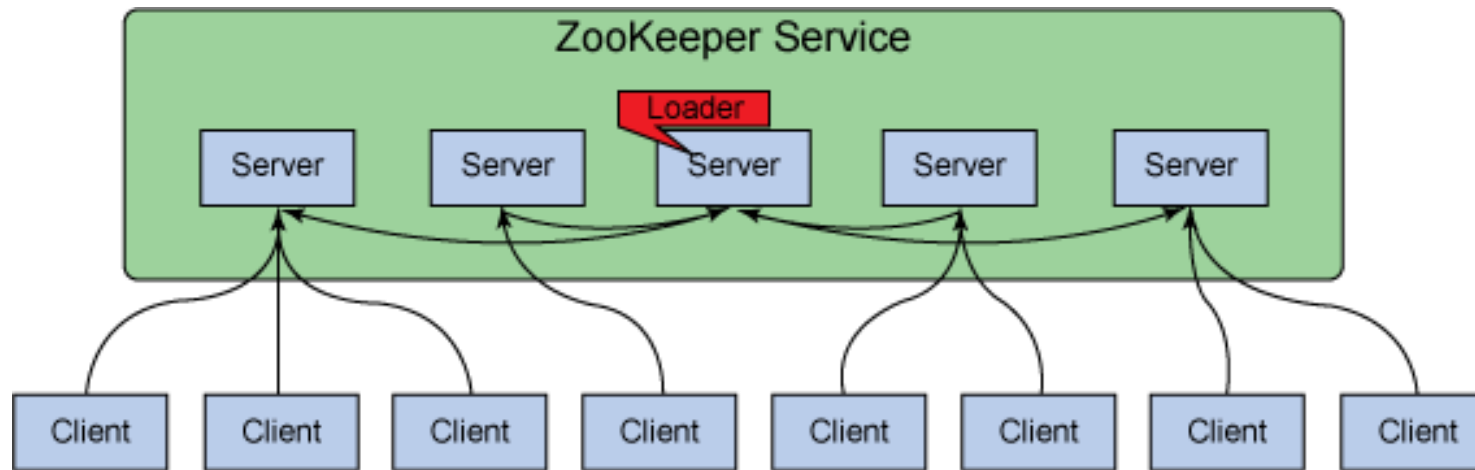


Apache Zookeeper

- Es un sistema de coordinación para sistemas distribuidos
- Ofrece servicios de configuración, nombres, grupos y sincronización
- Permite implementar esquemas de selección de líder, consenso y presencia de nodos en un cluster



Apache Zookeeper - Arquitectura



Apache Zookeeper - Arquitectura

- Un cliente puede conectarse a cualquier servidor de zookeeper.
- Las lecturas siempre se sirven desde el mismo nodo al que el cliente está conectado.
- Las escrituras al servidor son enviadas al líder, el cual las replica al resto de los nodos.
- Las escrituras requieren quorum para poder confirmarse (más de la mitad de los nodos deben haber confirmado la escritura).
- Recién al alcanzar el quorum se le indica al cliente que la escritura fue exitosa.



Apache Zookeeper - Garantías

- Consistencia secuencial
- Atomicidad
- Imágen de sistema única
- Confiabilidad (respuesta satisfactoria)
- Límite temporal (< decenas segundos)



Apache Zookeeper - Garantías

- **Consistencia secuencial:**
 - Las actualizaciones de un mismo cliente se aplican en orden secuencial.
 - Ningún otro cliente va a percibir los cambios en un orden distinto del aplicado por el original.
- **Atomicidad:**
 - Los cambios se aplican satisfactoriamente o fallan, no hay aplicación parcial de los mismos.
- **Imágen de sistema única:**
 - Un cliente va a ver la misma imágen del sistema sin importar el servidor al que se conecte.



Apache Zookeeper - Garantías

- **Confiabilidad:**
 - Si un cliente recibe una respuesta de éxito, el cambio fue aplicado exitosamente, si no recibe la respuesta el cliente no puede saber si el cambio se aplicó o no.
 - Cualquier cambio exitoso visto por un cliente no va a perderse en caso de una falla de un servidor. Nunca se realiza rollback de estos.
- **Límite temporal:**
 - La imagen del sistema vista por un cliente está garantizada en estar actualizada dentro de un límite de tiempo menor a decenas de segundos.
 - En caso de detectarse que es más vieja que este límite el cliente va a recibir un error de conectividad con el servidor.



Apache Zookeeper

¿Quién lo utiliza?

- **YARN** utiliza Zookeeper para soportar HA (Alta Disponibilidad) en su ResourceManager
- **HBase**, base de datos NoSQL basada en HDFS
- **BookKeeper** es un sistema de logging incluido con ZooKeeper



Hadoop - Distribuciones Comerciales

Hortonworks HDP

<https://hortonworks.com/products/data-center/hdp/>



IBM BigInsights

<https://www.ibm.com/analytics/us/en/technology/biginsights/>



Cloudera CDH

<https://www.cloudera.com/developers/inside-cdh.html>



Amazon EMR

<https://aws.amazon.com/emr/>



Alianza IBM + Hortonworks

- IBM, Hortonworks expand partnership to help businesses accelerate data-driven decision making
<https://developer.ibm.com/dwblog/2017/ibm-hortonworks-expand-partnership-help-businesses-accelerate-data-driven-decision-making/>
- Anuncio realizado el 13/Junio/2017
- IBM utilizará la distribución de Hadoop de Hortonworks (HDP) y la complementará con:
 - IBM BigSQL
 - IBM Data Science Experience (DSX)
- BigInsights pasa a End-Of-Support
 - Julio de 2019



Referencias

<http://www.educacionline.com/instituto-de-marketing-online/>

<http://www.puromarketing.com/>

<http://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/>

Diference between big data and data mining: <http://stackoverflow.com/questions/22419958/what-is-the-difference-between-big-data-and-data-mining>

<http://blog.data-miners.com/> G. Linnoff blog

Scaling big data – Twitter

<http://www.kdd.org/sites/default/files/issues/14-2-2012-12/V14-02-02-Lin.pdf>



Facebook de la Asignatura con links a noticias sobre Big Data

Nuevas Tecnologías <https://www.facebook.com/profile.php?id=100009746673220>



Libro: Mining Massive Data Sets

<http://infolab.stanford.edu/~ullman/mmds/book.pdf>

Referencias

<http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>

Las claves del sistema de
recomendación de Netflix

<http://www.bigdatanews.com/profiles/blogs/fast-clustering-algorithms-for-massive-datasets>

<http://www.ibermatica.com/sala-de-prensa/opinion/big-data-el-termino-de-moda-en-el-mundo-de-la-informatica>

