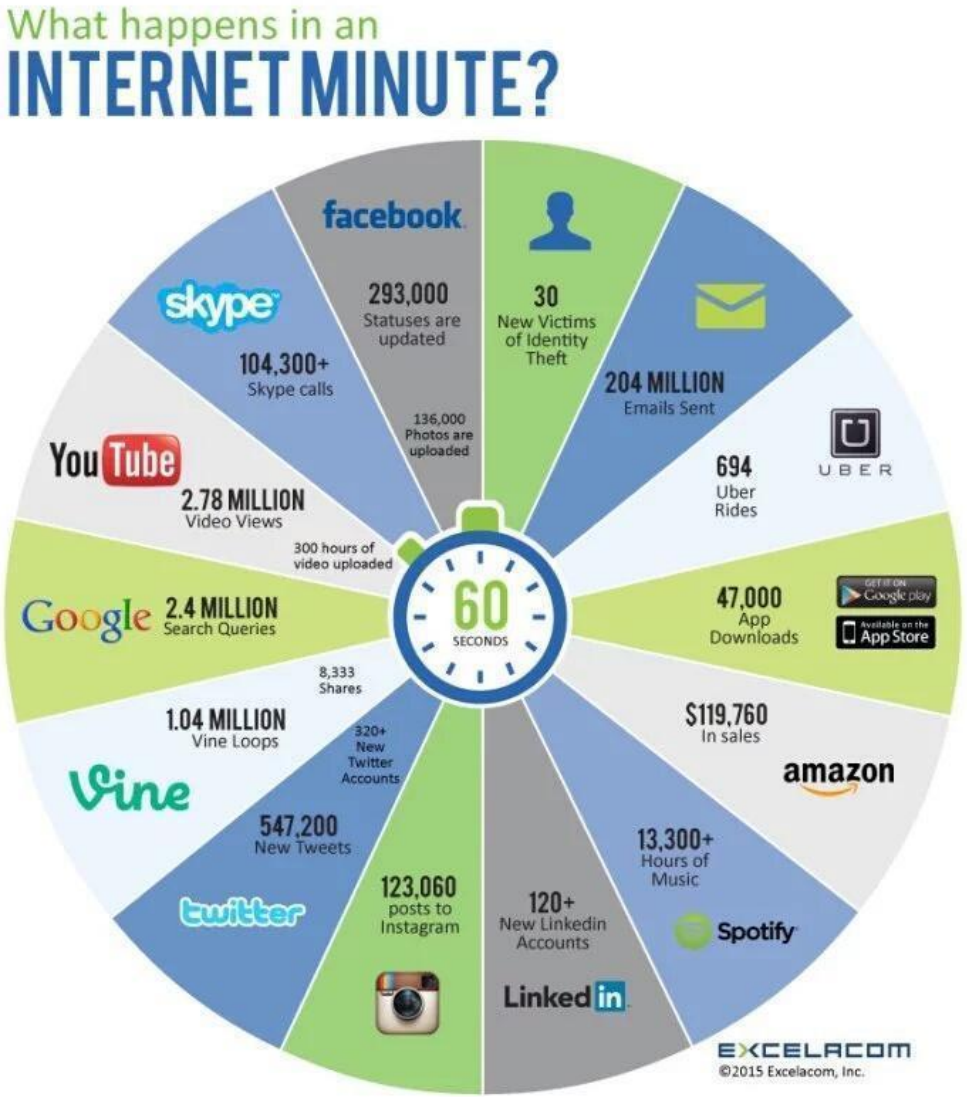
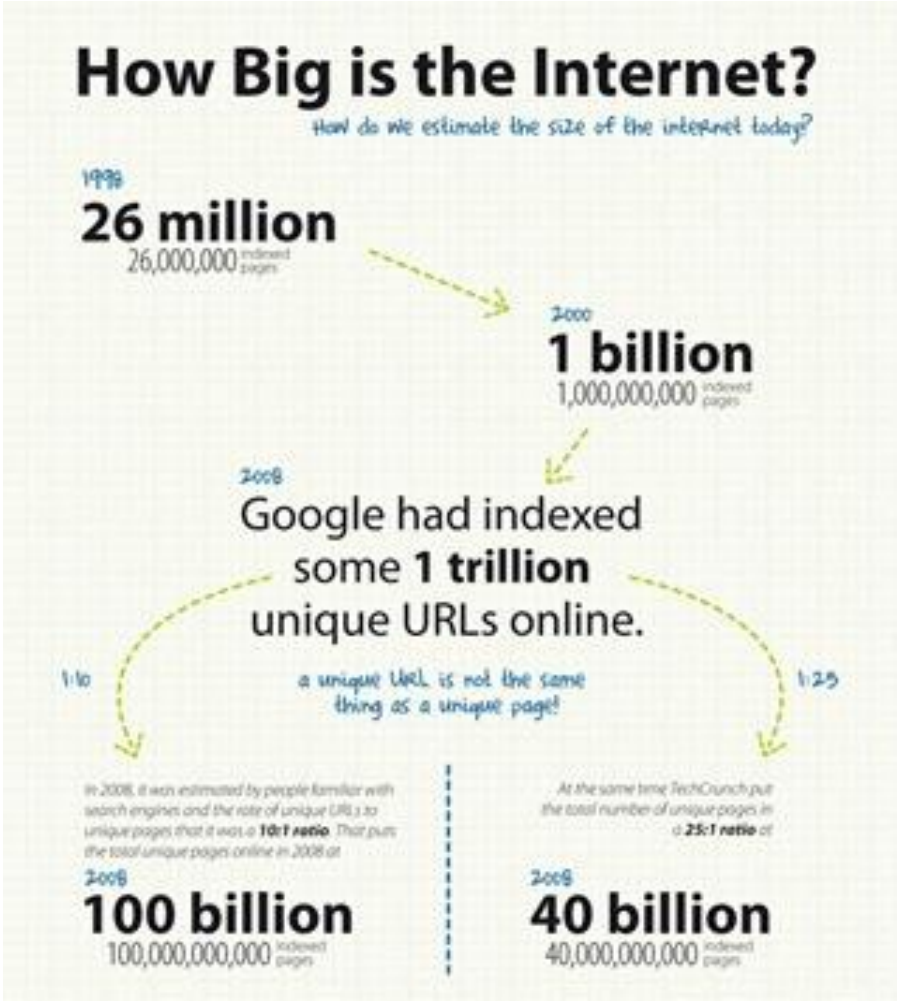
EAE Business School

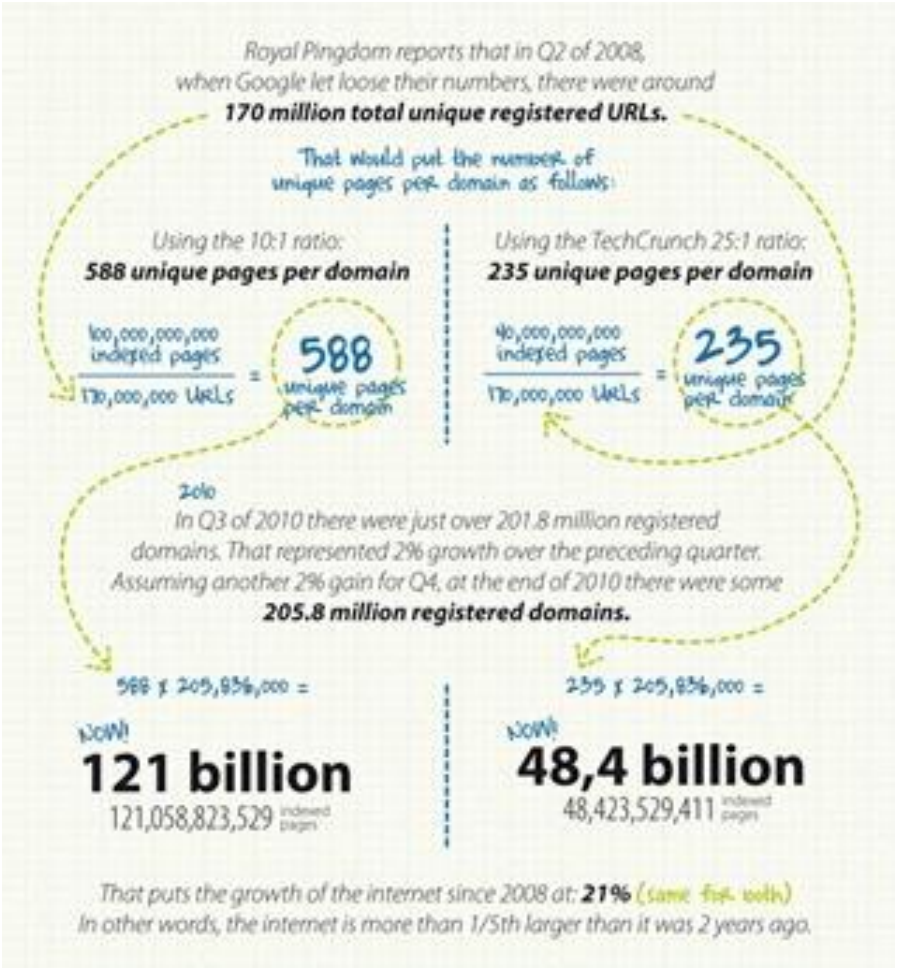
NoSQL & BigData

Máster en Business Intelligence e Innovación Tecnológica

Índice de contenidos


What happens in an
INTERNET MINUTE?
60 SECONDS
facebook
293,000 Statuses are updated
136,000 Photos are uploaded
30 New Victims of Identity Theft
204 MILLION Emails Sent
694 Uber Rides
UBER
47,000 App Downloads
GET IT ON Google play
Available on the App Store
$119,760 In sales
amazon
13,300+ Hours of Music
Spotify
120+ New Linkedin Accounts
Linked in
123,060 posts to Instagram
320+ New Twitter Accounts
547,200 New Tweets
twitter
1.04 MILLION Vine Loops
Vine
8,333 Shares
2.4 MILLION Search Queries
Google
300 hours of video uploaded
2.78 MILLION Video Views
YouTube
104,300+ Skype calls
skype
EXCELACOM
©2015 Excelacom, Inc.


How Big is the Internet?
How do we estimate the size of the internet today?
1998
26 million
26,000,000 indexed pages
2000
1 billion
1,000,000,000 indexed pages
2008
Google had indexed some 1 trillion unique URLs online.
a unique URL is not the same thing as a unique page!
1:10
1:25
In 2008, it was estimated by people familiar with search engines and the rate of unique URLs to unique pages that it was a 10:1 ratio. That puts the total unique pages online in 2008 at
2008
100 billion
100,000,000,000 indexed pages
At the same time TechCrunch put the total number of unique pages in a 25:1 ratio at
2008
40 billion
40,000,000,000 indexed pages


Royal Pingdom reports that in Q2 of 2008, when Google let loose their numbers, there were around
170 million total unique registered URLs.
That would put the number of unique pages per domain as follows:
Using the 10:1 ratio:
588 unique pages per domain
100,000,000,000 indexed pages / 170,000,000 URLs = 588 unique pages per domain
Using the TechCrunch 25:1 ratio:
235 unique pages per domain
40,000,000,000 indexed pages / 170,000,000 URLs = 235 unique pages per domain
2010
In Q3 of 2010 there were just over 201.8 million registered domains. That represented 2% growth over the preceding quarter. Assuming another 2% gain for Q4, at the end of 2010 there were some
205.8 million registered domains.
588 x 205,836,000 =
NOW!
121 billion
121,058,823,529 indexed pages
235 x 205,836,000 =
NOW!
48,4 billion
48,423,529,411 indexed pages
That puts the growth of the internet since 2008 at: 21% (same for both)
In other words, the internet is more than 1/5th larger than it was 2 years ago.

- NoSQL – "not only SQL" – es una categoría general de sistemas de gestión de bases de datos que difiere de los RDBMS en diferente modos:
  - No tienen schemas, no permiten JOINs, no intentan garantizar ACID y escalan horizontalmente
  - Tanto las bases de datos NoSQL como las relacionales son tipos de Almacenamiento Estructurado.
- El término fue acuñado en 1998 por Carlo Strozzi y resucitado en 2009 por Eric Evans
- Evans sugiere mejor referirse a esta familia de BBDD de nueva generación como "Big Data" mientras que Strozzi considera ahora que NoREL es un mejor nombre
- La principal diferencia radica en cómo guardan los datos (por ejemplo, almacenamiento de un recibo):
  - En una RDBMS tendríamos que partir la información en diferentes tablas y luego usar un lenguaje de programación en la parte servidora para transformar estos datos en objetos de la vida real.
  - En NoSQL, simplemente guardas el recibo:
  - NoSQL es libre de schemas, tú no diseñas tus tablas y su estructura por adelantado

Características Principales

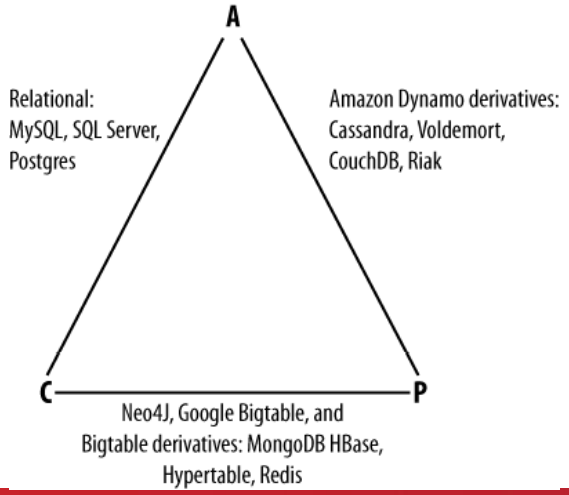
- Fáciles de usar en clústers de balanceo de carga convencionales → facilitan escalabilidad horizontal
- Guardan datos persistentes (no sólo cachés)
- No tienen esquemas fijos y permite la migración del esquema sin tener que ser reiniciadas o paradas
- Suelen tener un sistema de consultas propio en vez de usar un lenguaje de consultas estándar
- Tienen propiedades ACID en un nodo del clúster y son “eventualmente consistentes” en el clúster

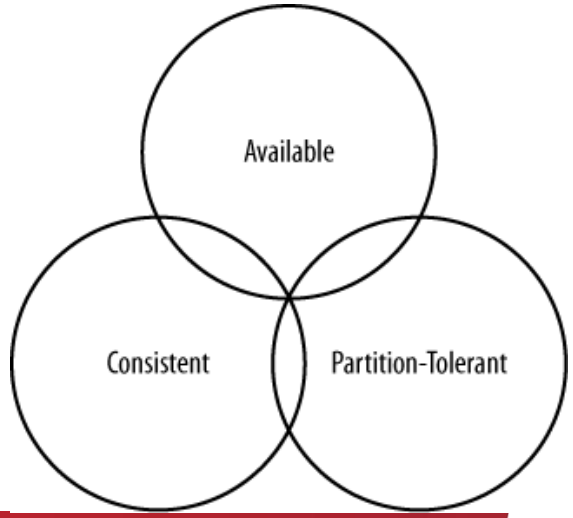
Contexto

- Aplicaciones cloud:
  - Aplicaciones con uso intensivo de datos en miles de servicios de almacenamiento
  - Características básicas:
    - Elasticidad
    - Tolerancia a Fallos
    - Aprovisionamiento automático
- Bases de datos cloud:
  - El escalado tradicional no es posible: necesita mayor inversión, mayor detalle técnico,…
- Principio de arquitectura: escalado horizontal basado en partición de datos y dividir la base de datos entre varios servidores:
  - Manejar el acceso principal en la aplicación.
  - Escalar bien para lecturas y escrituras.
  - No es transparente, la aplicación debe gestionar las particiones.

Introducción al teorema CAP

- Teorema de Brewer: “es imposible para un sistema computacional distribuido ofrecer simultáneamente las siguientes tres garantías”:
  - Consistencia – todos los nodos ven los mismos datos al mismo tiempo
  - Disponibilidad (Availability) – garantiza que cada petición recibe una respuesta acerca de si tuvo éxito o no
  - Tolerancia a la partición (Partition) – el sistema continua funcionando a pesar de la pérdida de mensajes
- Equivalente a:
  - “You can have it good, you can have it fast, you can have it cheap: pick two.”





RDBMS vs NoSQL

- Los RDBMS tradicionales nos permiten definir la estructura de un esquema que demanda reglas rígidas y garantizan ACID
- Las aplicaciones web y sistemas de información modernos presentan desafíos muy distintos a los sistemas empresariales tradicionales (e.j. sistemas bancarios):
  - Datos a escala web
  - Alta frecuencia de lecturas y escrituras
  - Cambios de esquema de datos frecuentes
  - Las aplicaciones sociales (no bancarias) no necesitan el mismo nivel de ACID
- Aparición de soluciones NoSQL
  - Cassandra, MongoDB, Jackrabbit , CouchDB, BigTable, Dynamo o Neo4j
- Consecuencia: relajación de propiedades ACID, especialmente en el contexto Cloud:
  - Disponibilidad no garantizada.
  - Tolerancia a particiones.

ACID vs BASE

- En el mundo relacional estamos familiarizados con las transacciones ACID, que garantizar la consistencia y estabilidad de las operaciones pero requieren lockings sofisticados:
  - ACID = Atomicidad, Consistencia, (Isolation) aislamiento y Durabilidad
- Las BBDD NoSQL son repositorios de almacenamiento más optimistas , siguen el modelo BASE:
  - Basic availability – el almacén funciona la mayoría del tiempo incluso ante fallos gracias al almacenamiento distribuido y replicado
  - Soft-sate – los almacenes no tienen porque ser consistentes ni sus réplicas en todo momento.
    - El programador puede verificar esa consistencia.
  - Eventual consistency – la consistencia se da eventualmente → propagación de datos de forma controlada (p. ej. Cuando no hay actualizaciones)
- BASE es una alternativa flexible a ACID para aquellos almacenes de datos que no requieren un adherencia estricta a las transacciones

Ventajas de NoSQL

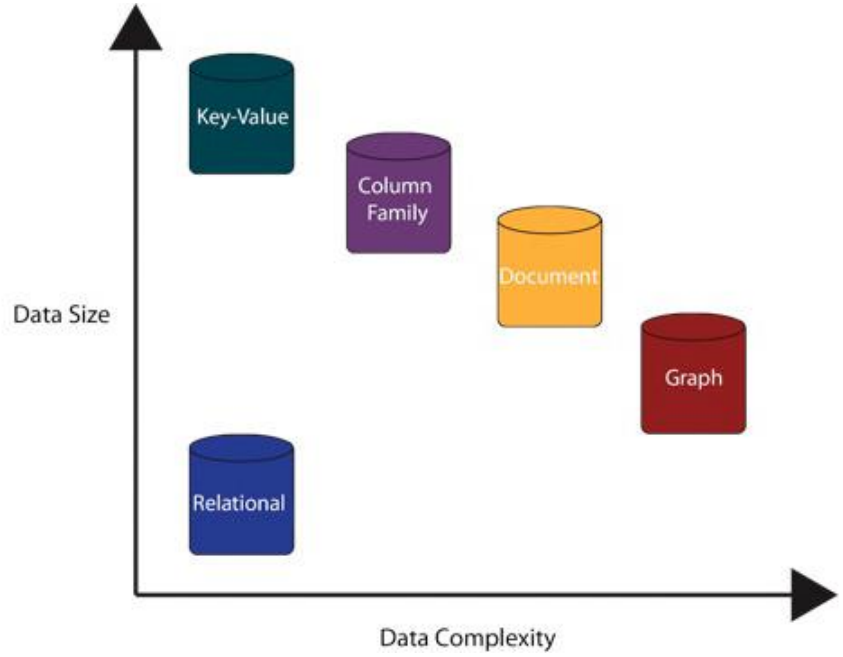
- Desafíos de gestión de información son difíciles de resolver con tecnología de bases de datos relacionales:
  - BBDD no escala a tu tráfico a un coste aceptable
  - El tamaño de tu esquema de datos ha crecido desproporcionalmente.
  - Generas mucho datos temporales que no corresponden al almacén de datos principal (carritos de compra, personalización de portales)
  - BBDD ha sido desnormalizada por razones de rendimiento o por conveniencia para utilizar los datos en una aplicación
  - Dataset tiene grandes cantidades de texto o imágenes, BLOB
  - Consultas contra datos que no implican relaciones jerárquicas sencillas; recomendaciones o consultas de inteligencia de negocio.
  - Ejemplo: "all people in a social network who have not purchased a book this year who are once removed from people who have“
  - Usas transacciones locales que no necesitan ser durables, e.j. Like.
  - Los sites AJAX tienen muchos casos de uso de este estilo.

Conceptos asociados a BDD distribuidas

- **Almacenes basados en columnas y filas**. RDBMS almacenan las filas de modo continuo en disco, mientras que algunas NoSQL guardan columnas de modo continuo
- **Consistencia eventual**: si no se realizan nuevas actualizaciones a un elemento de datos, todos los accesos del elemento devolverán el último valor actualizado
- **Sharding**: es una partición horizontal en una BBDD donde las filas de una tabla se mantienen de modo separado
- **Replicación maestro-maestro** es un método de replicación de BBDD que permite almacenar datos en un grupo de nodos y su actualización por cualquier miembro del grupo
- **Replicación maestro-esclavo** donde un sólo elemento se designa como “maestro” de un datastore y es el único nodo que permite modificar datos
- **Particionado** es la división de una BBDD lógica y sus partes constituyentes en un conjunto de partes independientes.
- **Modelo de consistencia**: contrato entre el programador y el sistema sobre las garantías de consistencia (write & read consistency: one, all, quorum, etc.)

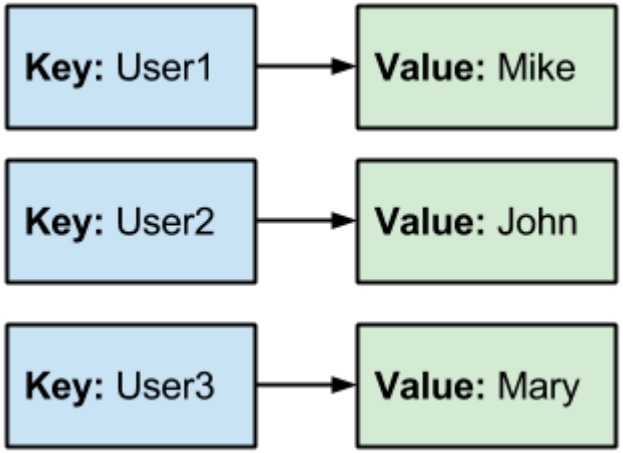
Según tipo

- Los principales tipos de BBDD de acuerdo con su implementación son los siguientes:
  - Almacenes de Clave-Valor
  - Almacenes de Familia de Columnas
  - Almacenes de documentos
  - Grafos



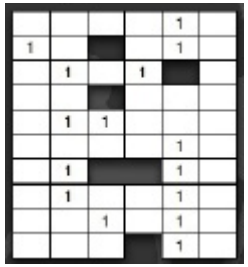
Orientadas a clave - valor

- Su precursor fue Amazon Dynamo
- Basadas en DHT (Distributed Hash Tables)
- Modelo de datos: colección de pares clave/valor
- Ejemplos: Dynomite, Voldemort, Tokyo



Orientadas a columnas

- Su precursor fue Google BigTable
- Modelo de datos: familia de columnas, esto es, un modelo tabular donde cada fila puede tener una configuración diferente de columnas
  - Guardan datos por columna en vez de las BBDD tradicionales que lo hacen por filas
- Ejemplos: HBase, Hypertable, Cassandra, Riak
- Buenas en:
  - Gestión de tamaño
  - Cargas de escrituras masivas orientas al stream
  - Alta disponibilidad
  - MapReduce

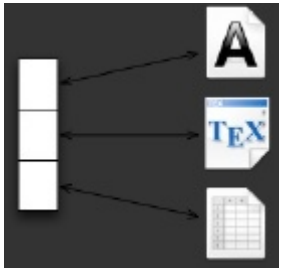


Orientadas a columnas

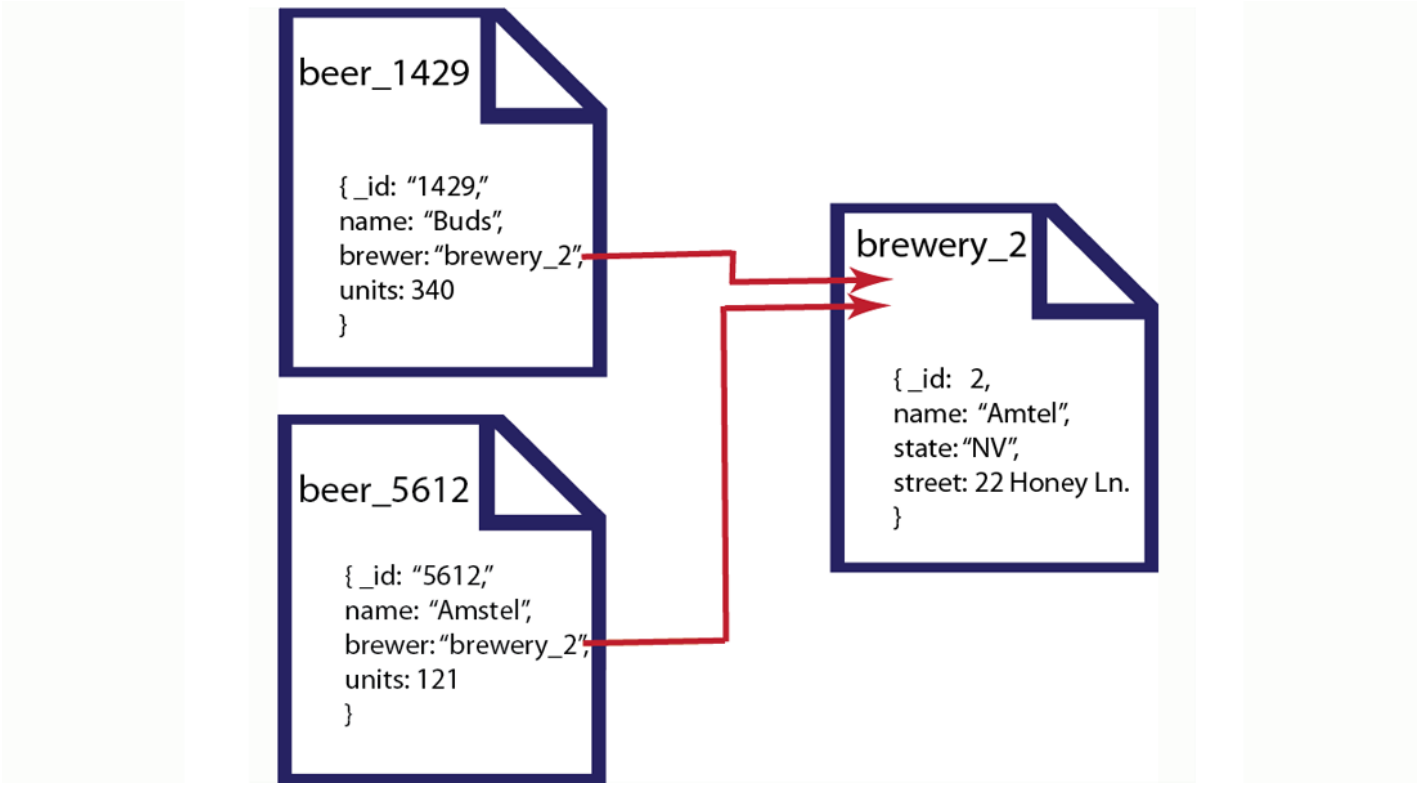
KEY	COLUMN FAMILIES	
ID	CUSTOMERINFO	ADDRESSINFO
1001	FirstName: Tom MiddleName: T LastName: Tester	Address1: 2001 Bayfront Dr. Address2: Suite#813 City: Tampa State: FL Zip: 34637 Country: US
1002	FirstName: Bob MiddleName: B LastName: Builder	Address1: 1234 Sunny Circle City: Beverly Hills State: CA Zip: 90210

Orientadas a documentos

- La precursora fue Lotus Notes
- Modelo de datos: colecciones de documentos (JSON, XML, BSON) que contienen colecciones de claves-valor
- Ejemplos: CouchDB, MongoDB
- Buenas en:
  - Modelado de datos natural
  - Amigables al programador
  - Desarrollo rápido
  - Orientas a la web: CRUD



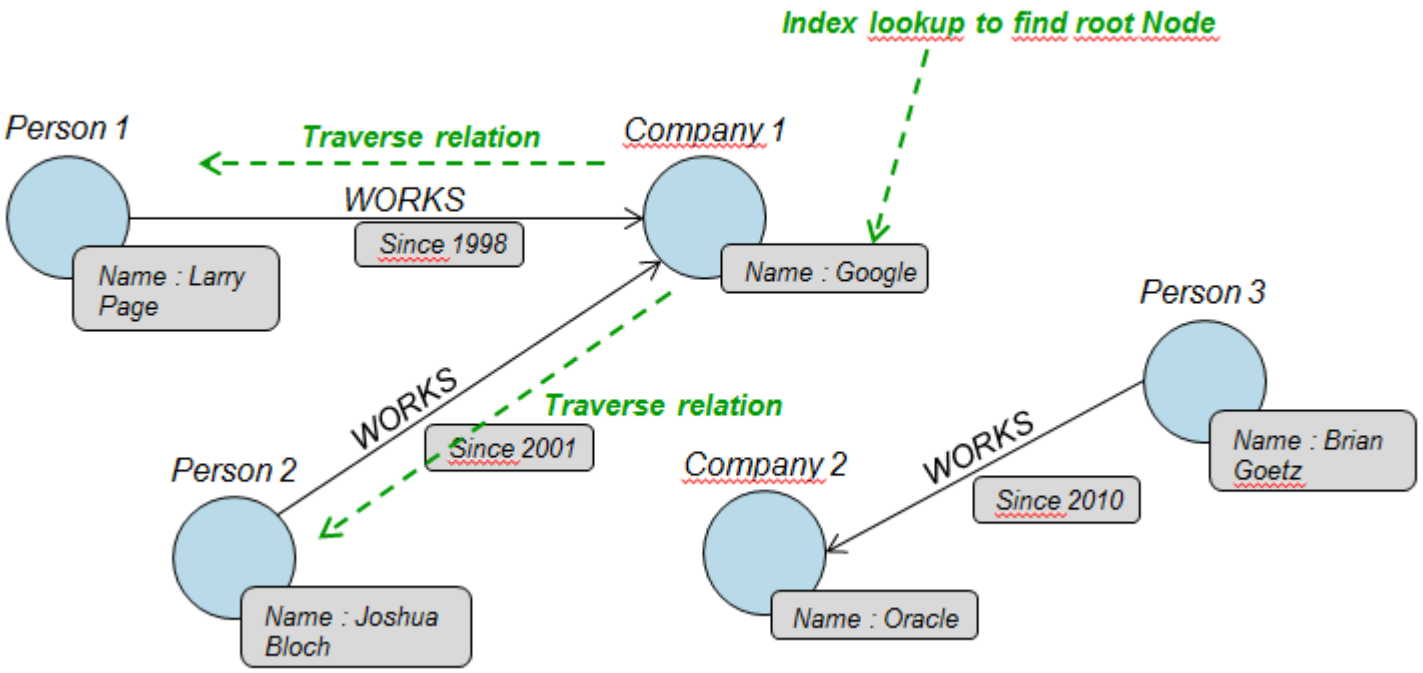
Orientadas a documentos



BBDD orientadas a Grafos

- Inspiradas por Euler y la teoría de grafos
- Modelo de datos: nodos, relaciones con pares clave valor en ambos
- Ejemplos: AllegroGraph, VertexBD, Neo4j
- Buenas en:
  - Modelar directamente un dominio en forma de grafo, una manera común de representar y entender datasets
  - Ofrecer excelente rendimiento cuando los datos están interconectados y no tabulares
  - Realizar operaciones transaccionales que exploten las relaciones entre entidades

BBDD orientadas a Grafos



Principales soluciones clave-valor

Name	Producer	Data model	Querying
SimpleDB	Amazon	set of couples (key, {attribute}), where attribute is a couple (name, value)	restricted SQL; select, delete, GetAttributes, and PutAttributes operations
Redis	Salvatore Sanfilippo	set of couples (key, value), where value is simple typed value, list, ordered (according to ranking) or unordered set, hash value	primitive operations for each value type
Dynamo	Amazon	like SimpleDB	simple get operation and put in a context
Voldemort	LinkeId	like SimpleDB	similar to Dynamo

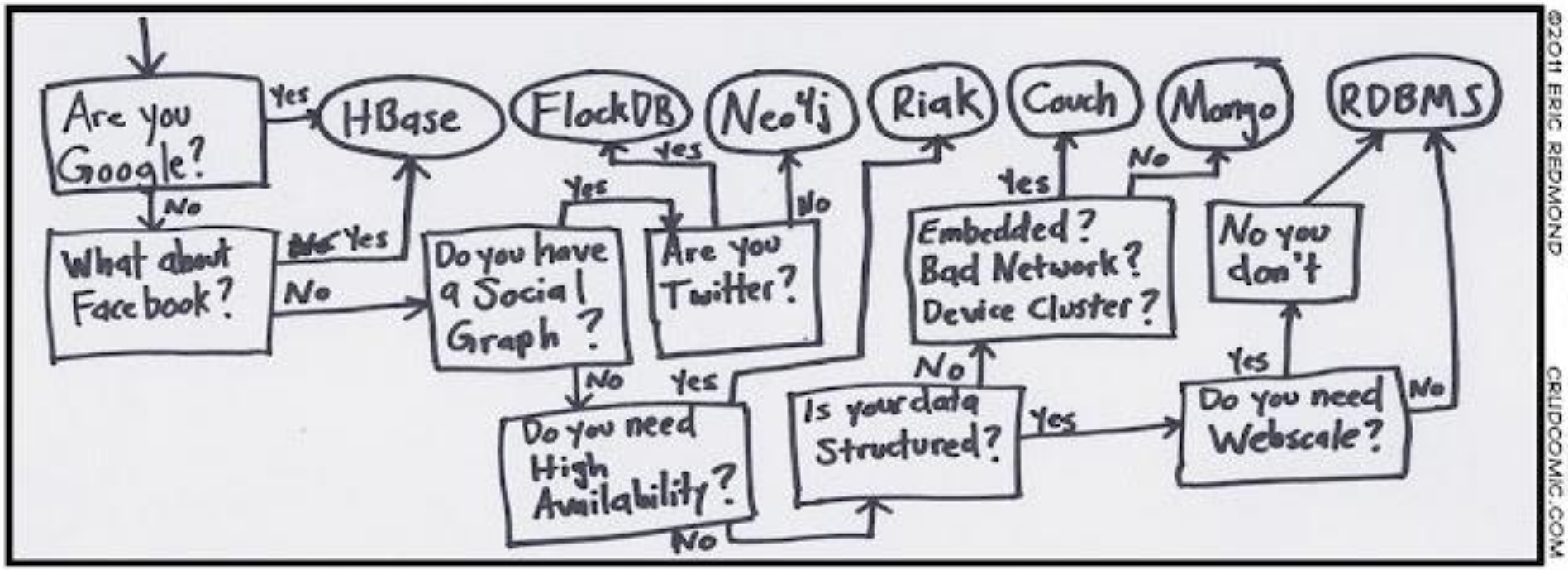
Principales soluciones orientadas a columnas

Name	Producer	Data model	Querying
BigTable	Google	set of couples (key, {value})	selection (by combination of row, column, and time stamp ranges)
HBase	Apache	groups of columns (a BigTable clone)	JRUBY IRB-based shell (similar to SQL)
Hypertable	Hypertable	like BigTable	HQL (Hypertext Query Language)
CASSANDRA	Apache (originally Facebook)	columns, groups of columns corresponding to a key (supercolumns)	simple selections on key, range queries, column or columns ranges
PNUTS	Yahoo	(hashed or ordered) tables, typed arrays, flexible schema	selection and projection from a single table (retrieve an arbitrary single record by primary key, range queries, complex predicates, ordering, top-k)

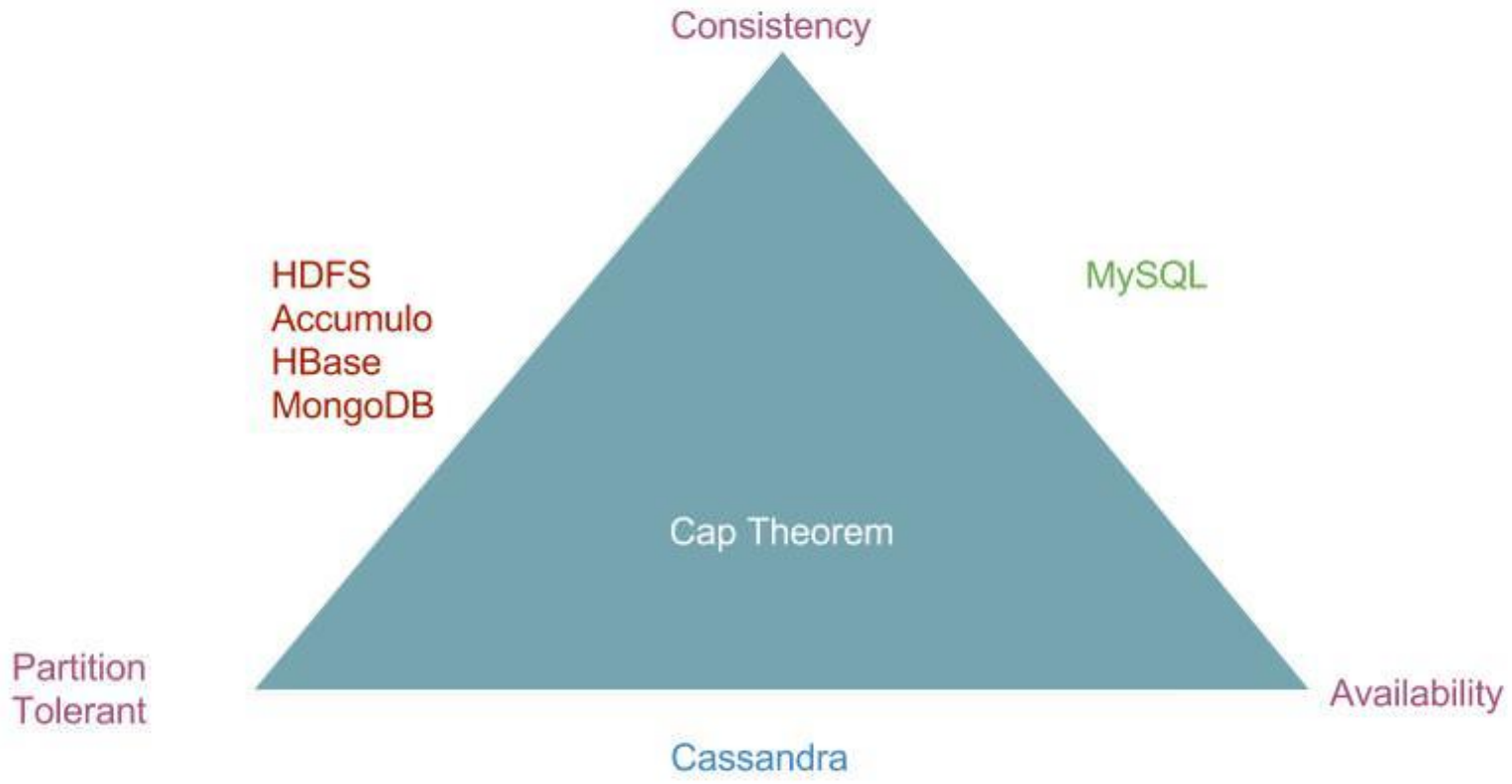
Principales soluciones orientadas a documentos

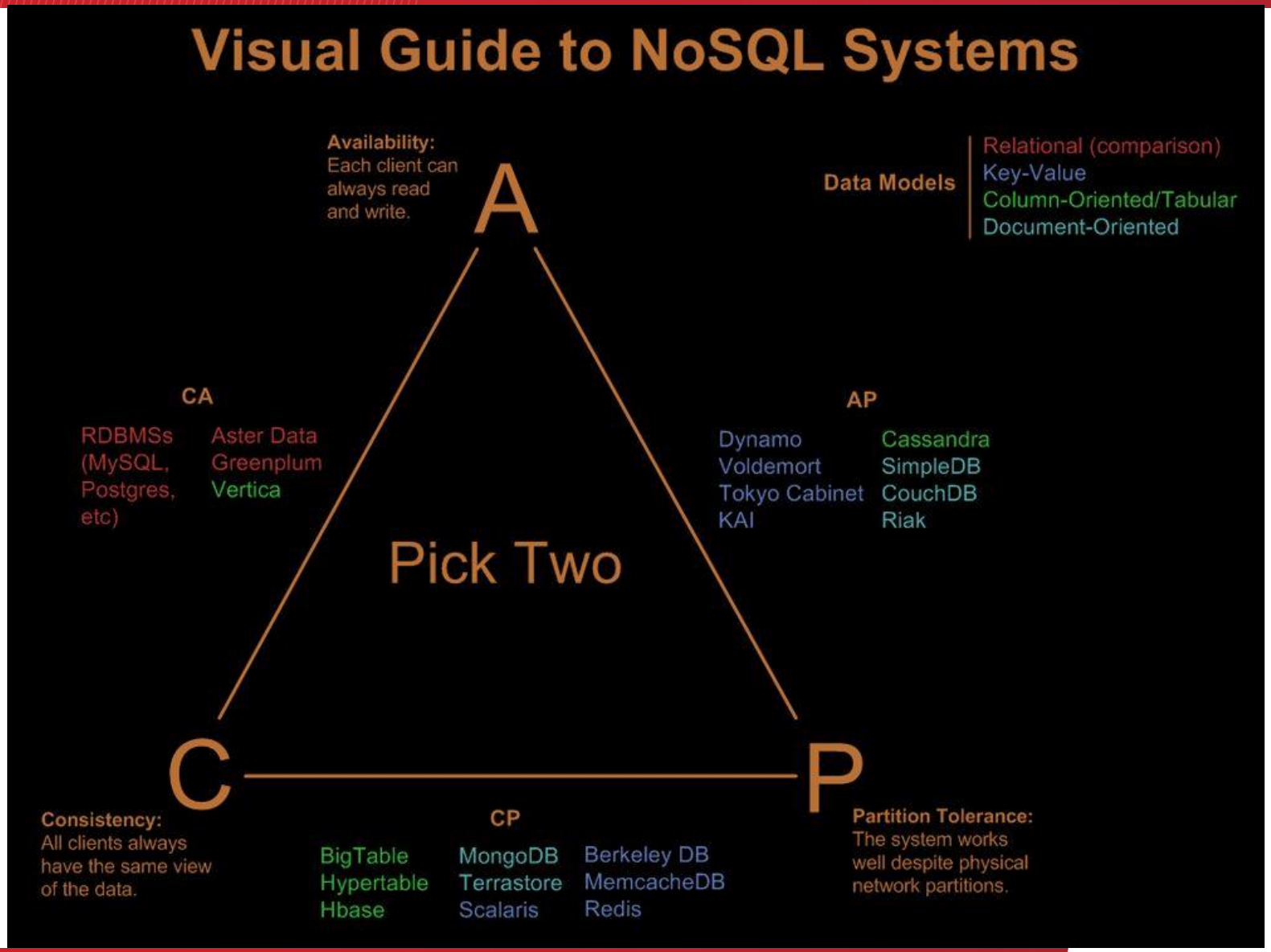
Name	Producer	Data model	Querying
MongoDB	10gen	object-structured documents stored in collections; each object has a primary key called ObjectId	manipulations with objects in collections (find object or objects via simple selections and logical expressions, delete, update,)
Couchbase	Couchbase[1]	document as a list of named (structured) items (JSON document)	by key and key range, views via Javascript and MapReduce

Taxonomía de soluciones NoSQL



Taxonomía de soluciones NoSQL




Visual Guide to NoSQL Systems
Availability:
Each client can always read and write.
A
Data Models
Relational (comparison)
Key-Value
Column-Oriented/Tabular
Document-Oriented
CA
RDBMSs (MySQL, Postgres, etc)
Aster Data
Greenplum
Vertica
AP
Dynamo
Voldemort
Tokyo Cabinet
KAI
Cassandra
SimpleDB
CouchDB
Riak
Pick Two
C
P
Consistency:
All clients always have the same view of the data.
CP
BigTable
Hypertable
Hbase
MongoDB
Terrastore
Scalaris
Berkeley DB
MemcacheDB
Redis
Partition Tolerance:
The system works well despite physical network partitions.

http://db-engines.com/en/ranking

181 systems in ranking, July 2013

Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	Oracle	Relational DBMS	1527.64	+12.74
2.	3.	Microsoft SQL Server	Relational DBMS	1321.99	+35.77
3.	2.	MySQL	Relational DBMS	1305.23	-29.71
4.	4.	PostgreSQL	Relational DBMS	182.37	-17.01
5.	5.	DB2	Relational DBMS	168.43	-8.61
6.	6.	Microsoft Access	Relational DBMS	141.64	-8.02
7.	7.	MongoDB	Document store	140.46	+2.97
8.	8.	Sybase	Relational DBMS	86.20	-2.21
9.	9.	SQLite	Relational DBMS	82.81	-5.01
10.	10.	Teradata	Relational DBMS	56.34	+5.23
11.	11.	Solr	Search engine	45.48	-0.95
12.	12.	Cassandra	Wide column store	42.24	+4.60
13.		FileMaker	Relational DBMS	34.49	
14.	13.	Redis	Key-value store	33.09	-1.13
15.	14.	Memcached	Key-value store	29.45	-1.27
16.	15.	HBase	Wide column store	25.91	+0.13
17.	16.	Informix	Relational DBMS	24.58	-0.15

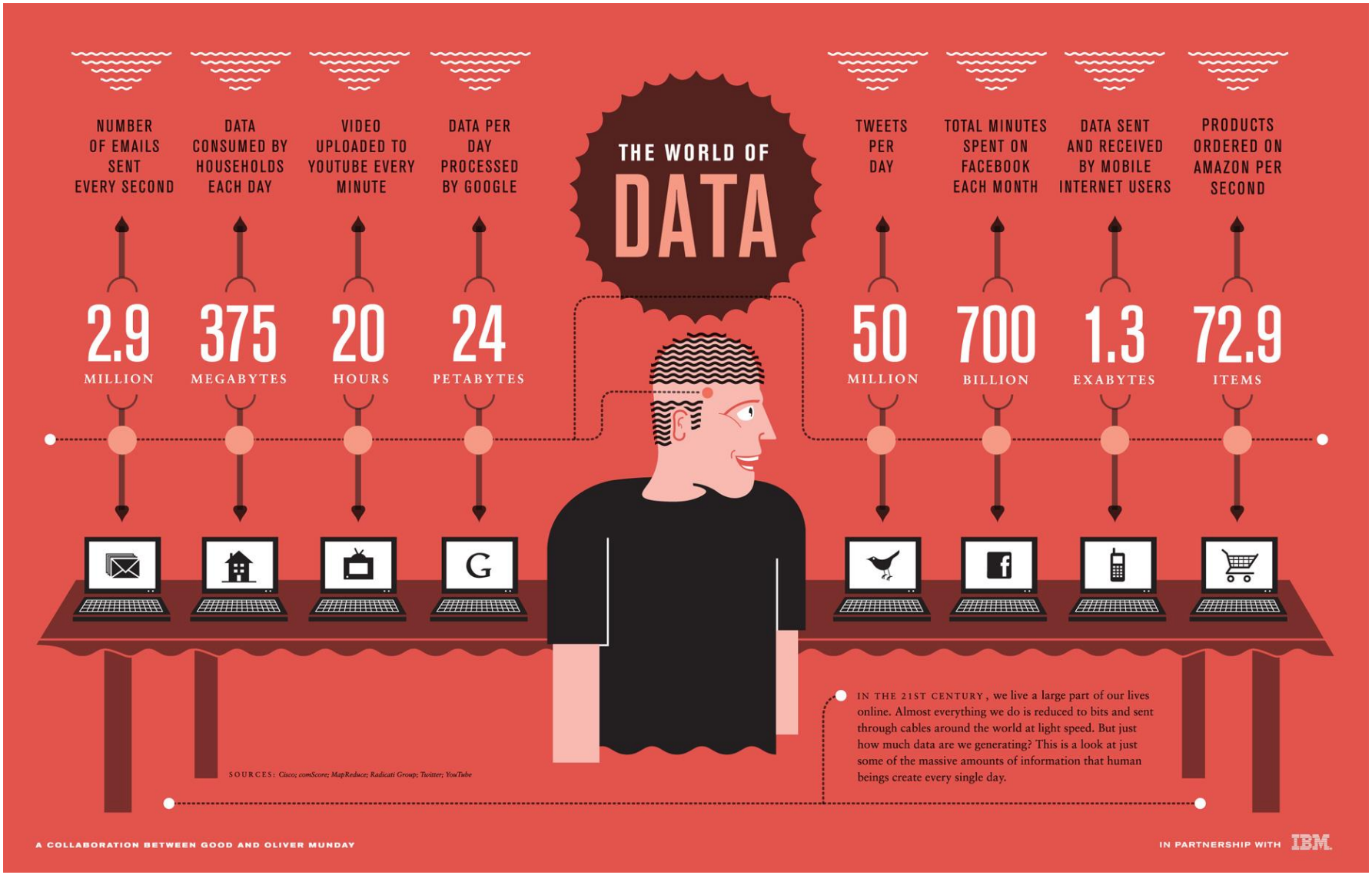
Taxonomía de soluciones NoSQL

315 systems in ranking, January 2017

Rank Jan 2017	Rank Dec 2016	Rank Jan 2016	DBMS	Database Model	Score Jan 2017	Score Dec 2016	Score Jan 2016
1.	1.	1.	Oracle	Relational DBMS	1416.72	+12.32	-79.36
2.	2.	2.	MySQL	Relational DBMS	1366.29	-8.12	+67.03
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1220.95	-5.70	+76.89
4.	↑ 5.	4.	MongoDB	Document store	331.90	+3.22	+25.88
5.	↓ 4.	5.	PostgreSQL	Relational DBMS	330.37	+0.35	+47.97
6.	6.	6.	DB2	Relational DBMS	182.49	-1.85	-13.88
7.	7.	↑ 8.	Cassandra	Wide column store	136.44	+2.16	+5.49
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	127.45	+2.75	-6.59
9.	9.	↑ 10.	Redis	Key-value store	118.70	-1.20	+17.54
10.	10.	↓ 9.	SQLite	Relational DBMS	112.38	+1.54	+8.64
11.	11.	↑ 12.	Elasticsearch	Search engine	106.17	+2.90	+28.96
12.	12.	↑ 14.	Teradata	Relational DBMS	74.17	+0.79	-0.78
13.	13.	↓ 11.	SAP Adaptive Server	Relational DBMS	69.10	-1.32	-14.08
14.	14.	↓ 13.	Solr	Search engine	68.08	-0.92	-7.32
15.	15.	↑ 16.	HBase	Wide column store	59.14	+0.51	+5.77
16.	16.	↑ 18.	Splunk	Search engine	55.49	+0.57	+12.37
17.	17.	17.	FileMaker	Relational DBMS	53.49	-0.63	+4.66

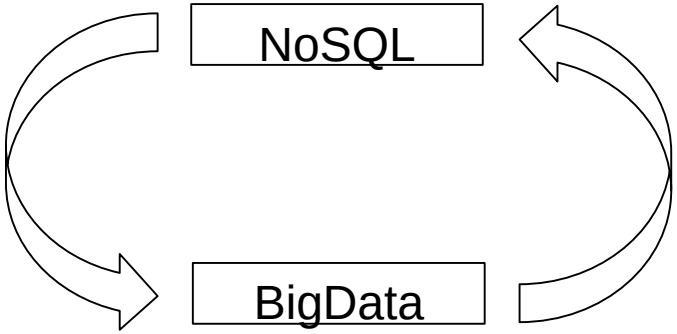
Conclusiones

- NoSQL cubre una parte de las aplicaciones cloud con uso intensivo de datos, principalmente las aplicaciones web.
- Problemas con cloud computing:
  - Aplicaciones Saas: algunas aplicaciones empresariales requieren propiedades ACID, seguridad, etc → NoSQL no puede ser la única opción.
  - Soluciones híbridas: uso combinado de RDBMS y NoSQL en la configuración de un backend. P. ej: voldemort + MySQL.
- Una nueva generación de RDBMS elásticos y escalables aparece: NewSQL.
  - Diseñadas para la escalabilidad horizontal en máquinas distribuidas.
  - Garantizan ACID.
  - Las aplicaciones interactúan mediante SQL.
  - Control de concurrencia libre de bloqueos.
  - Mejor performance que sistemas tradicionales.
  - Ejemplos: MySQL Cluster, VoltDB, Clustrix,…
- Nuevo concepto: SPRAIN
  - Scalability
  - Performance
  - Relaxed consistency
  - Agility
  - Intricacy (complejidad)
  - Necessity


THE WORLD OF
DATA
NUMBER OF EMAILS SENT EVERY SECOND
2.9
MILLION
DATA CONSUMED BY HOUSEHOLDS EACH DAY
375
MEGABYTES
VIDEO UPLOADED TO YOUTUBE EVERY MINUTE
20
HOURS
DATA PER DAY PROCESSED BY GOOGLE
24
PETABYTES
TWEETS PER DAY
50
MILLION
TOTAL MINUTES SPENT ON FACEBOOK EACH MONTH
700
BILLION
DATA SENT AND RECEIVED BY MOBILE INTERNET USERS
1.3
EXABYTES
PRODUCTS ORDERED ON AMAZON PER SECOND
72.9
ITEMS
IN THE 21ST CENTURY, we live a large part of our lives online. Almost everything we do is reduced to bits and sent through cables around the world at light speed. But just how much data are we generating? This is a look at just some of the massive amounts of information that human beings create every single day.
SOURCES: Cisco; comScore; MapReduce; Radicati Group; Twitter; YouTube
A COLLABORATION BETWEEN GOOD AND OLIVER MUNDAY
IN PARTNERSHIP WITH IBM

Introducción

- La explosión de la información produce la aparición de las plataformas NoSQL
- NoSQL facilita el procesamiento de grandes volúmenes de datos, lo que hace aparecer el BigData



Denominamos Big Data a la gestión y análisis de enormes volúmenes de datos que no pueden ser tratados de manera convencional, ya que superan los límites y capacidades de las herramientas de software habitualmente utilizadas para la captura, gestión y procesamiento de datos.

Introducción

Big data Consists of datasets that grow so large that they become awkward to work with using on-hand DB Management tools.

Wikipedia

Big data is when the size of the data itself becomes part of the problem

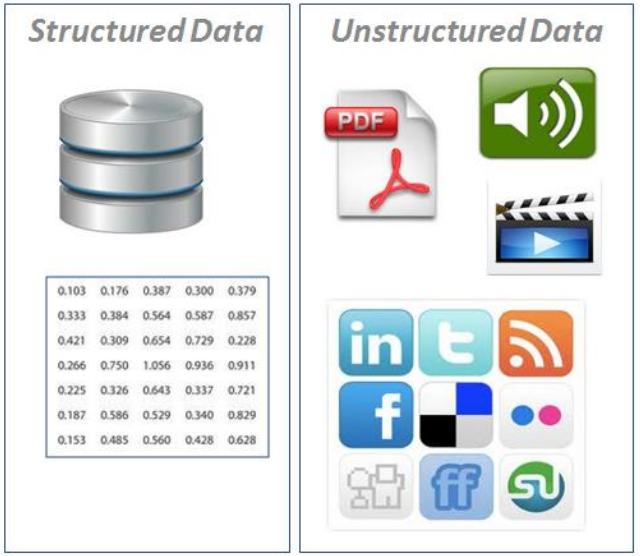
Mike Lukides, O'Reilly Radar

It's not just your "Big Data" problems, it's all about your BIG "data" Problems.

Alexander Stojanovic, Hadoop Manager on Win Azure

Introducción

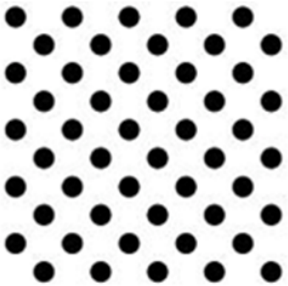
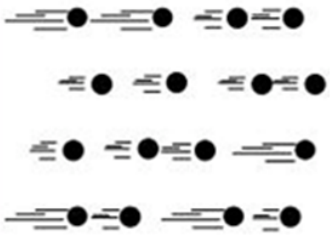
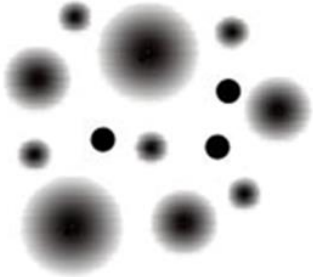
- Big Data, por tanto, hace referencia tanto a la cantidad de datos almacenados como a los procedimientos, herramientas y arquitecturas que van a permitir explotarlos.
- Los conjuntos de datos con los que podemos trabajar son enormes, de forma que su tamaño supera la capacidad de explotarlos con sistemas típicos.
- Otro factor de complejidad a considerar es el formato: los datos pueden presentarse en muchos formatos distintos, estructurados y no estructurados.
- La capacidad de procesar y explotar toda esa información en tiempo real será determinante y constituirá un factor diferencial.
- De esta forma se toma conciencia del valor de la información, más allá de su origen…sistemas transaccionales o ERPs propios, redes sociales, logs aplicativos, medidores de visitas o tráfico, incluso los propios sistemas de BI. Y sobretodo…del valor de su explotación conjunta.



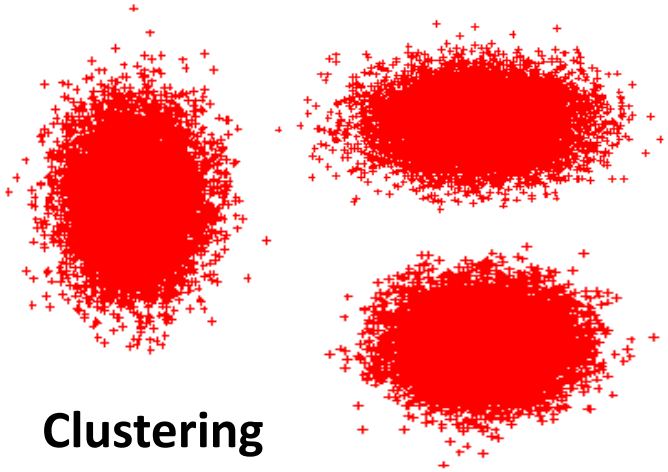
Introducción

Introducción

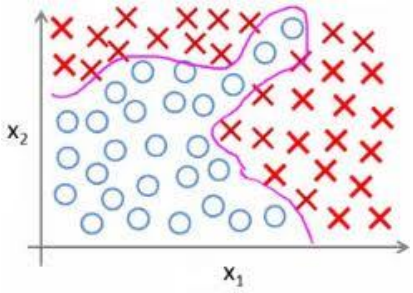
Las 5V

Volume	Velocity	Variety	Veracity	Value
				
Data at Rest	**Data in Motion**	**Data in Many Forms**	**Data in Doubt**	**Data into Money**
Terabytes to Exabytes of existing data to process	Streaming data, requiring milliseconds to seconds to respond	Structured, unstructured, text, multimedia,...	Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations	Business models can be associated to the data

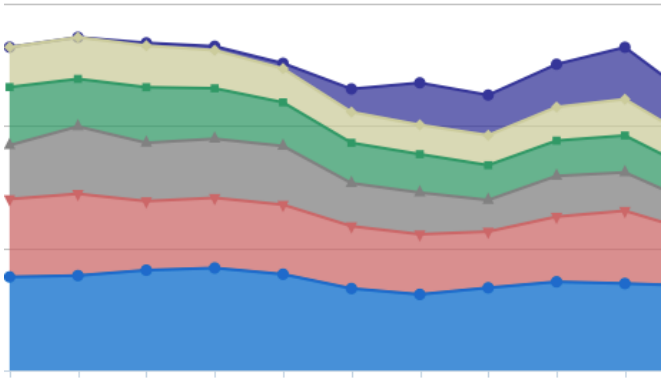
Usos



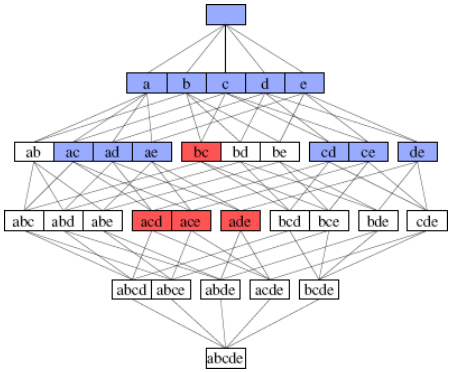
Clustering



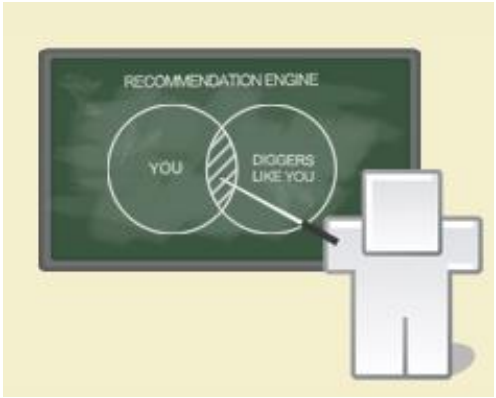
Classification



Real Time Analytics/
Big Data Streams



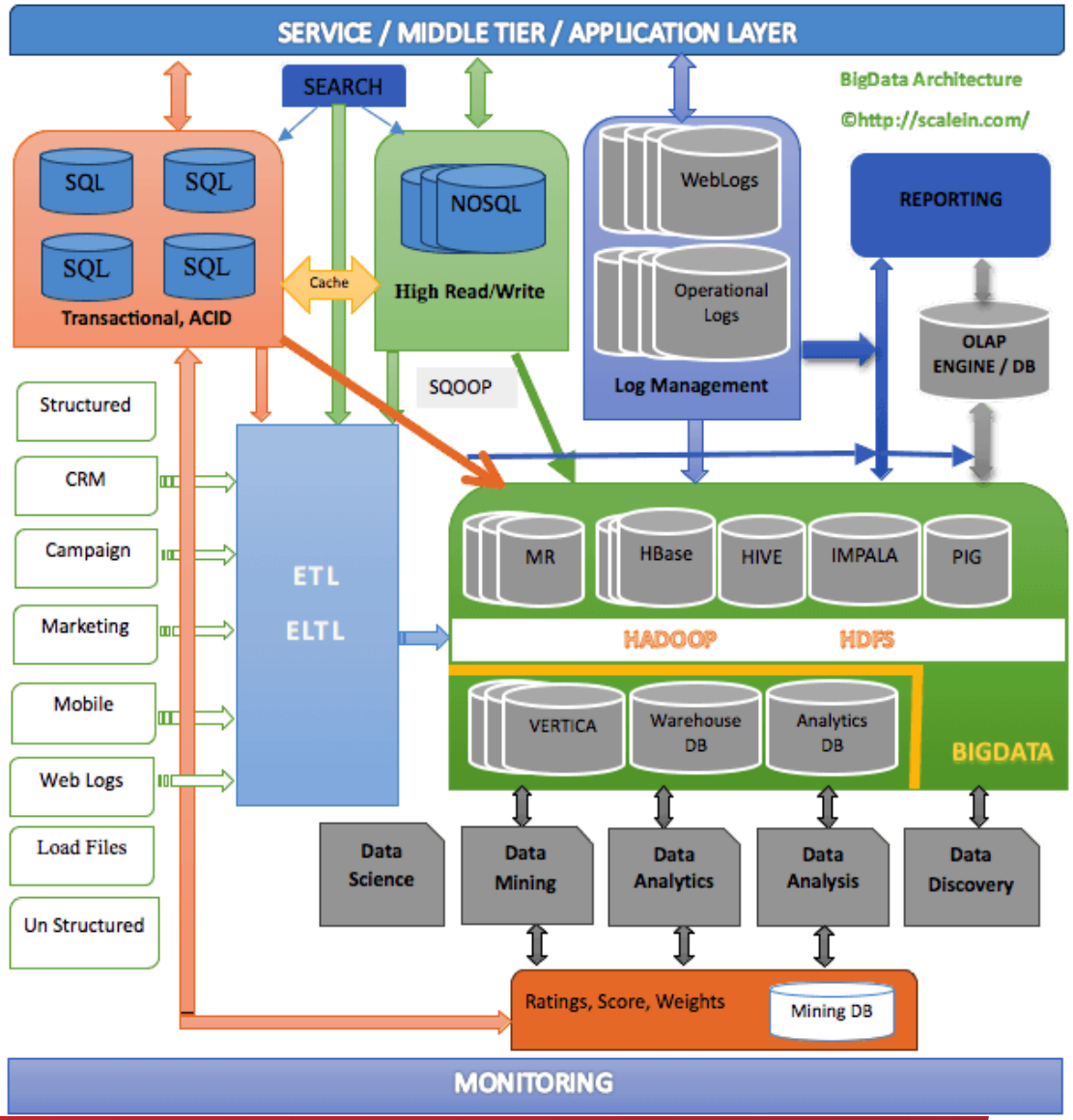
Association



Recommendation
Systems

Social Media Mining
Social Big Data

Arquitectura



Arquitectura

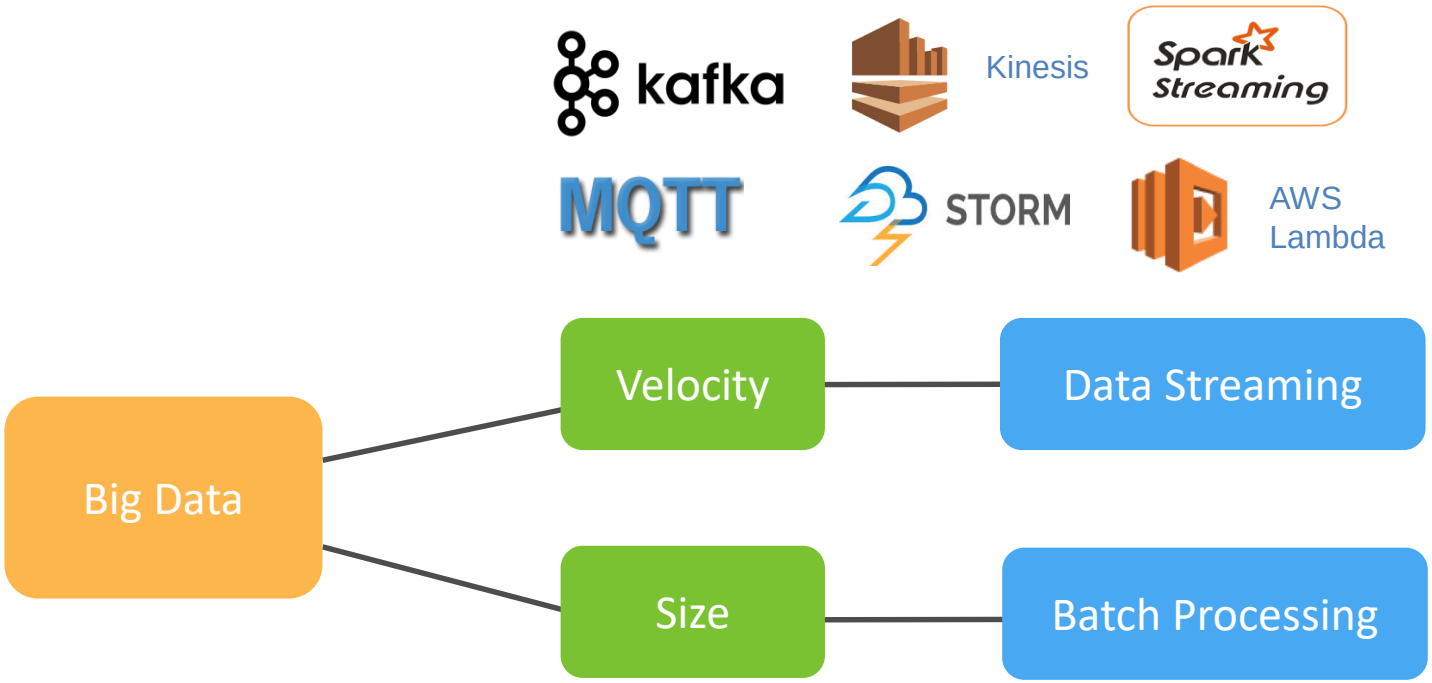
- La variedad de la información, el volumen, la distribución, son factores que provocan una primera decisión estratégica: explotación batch o real time

Arquitectura

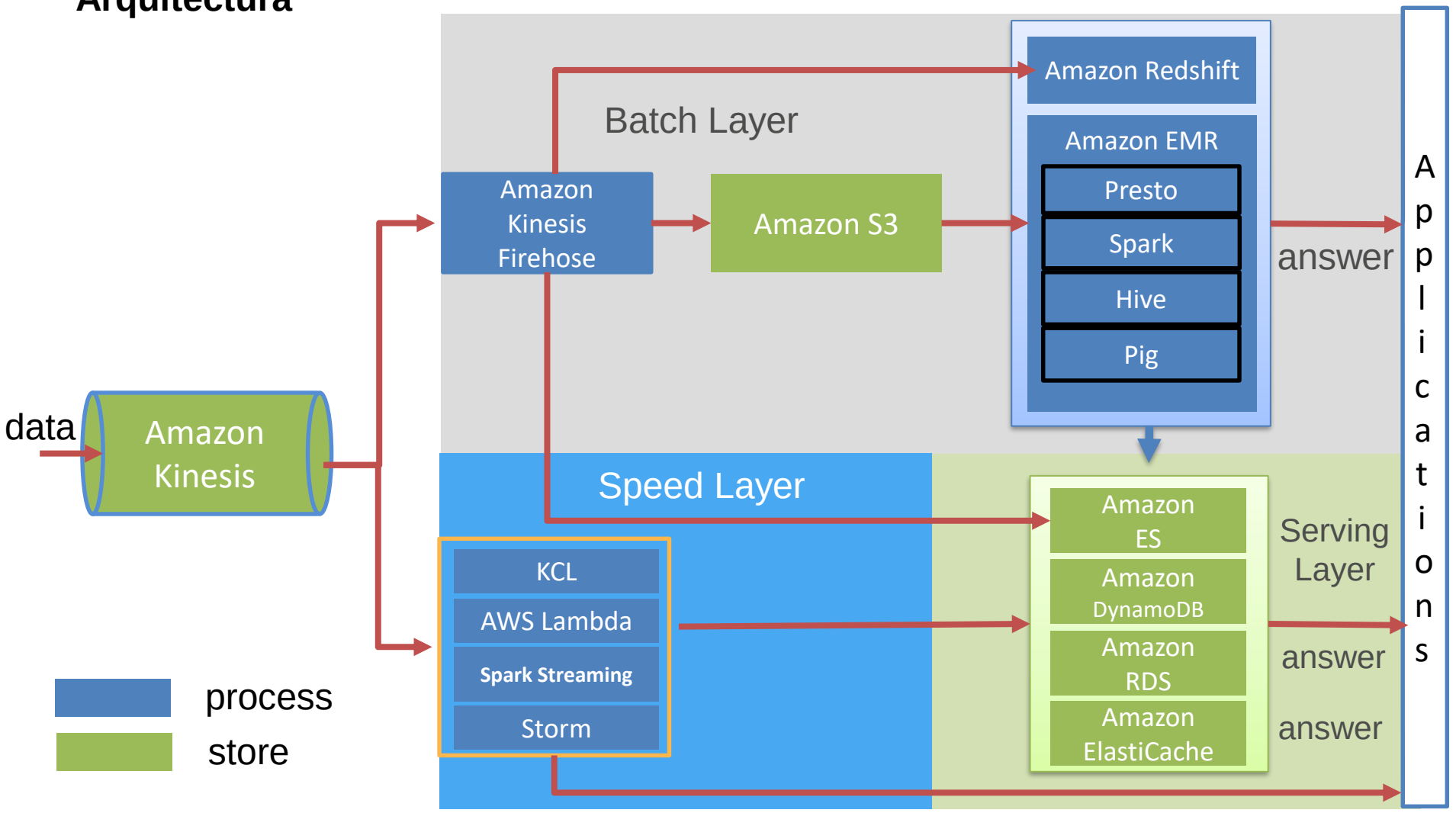
- Cada modo de operación cuenta con unas fortalezas y unas debilidades, por lo que una misma acción no debería ser acometida de las dos formas.
- En cierta manera, esa decisión viene definida por el propio CAP

	Batch Processing (Data Lake)	Real-time Processing (Stream)
Análisis de logs	Que ha ido mal en la última hora	Tomar acciones correctivas
Análisis de facturación	Cuanto ha gastado un cliente de media el ultimo mes?	Notificar a un cliente que su límite de facturación se aproxima
Preferencias de cliente	Qué ads fueron más efectivos?	Qué está viedo el cliente ahora?
Fraude	Auditoría de rastreo/ evidencias forenses	Detección de fraude en tiempo real

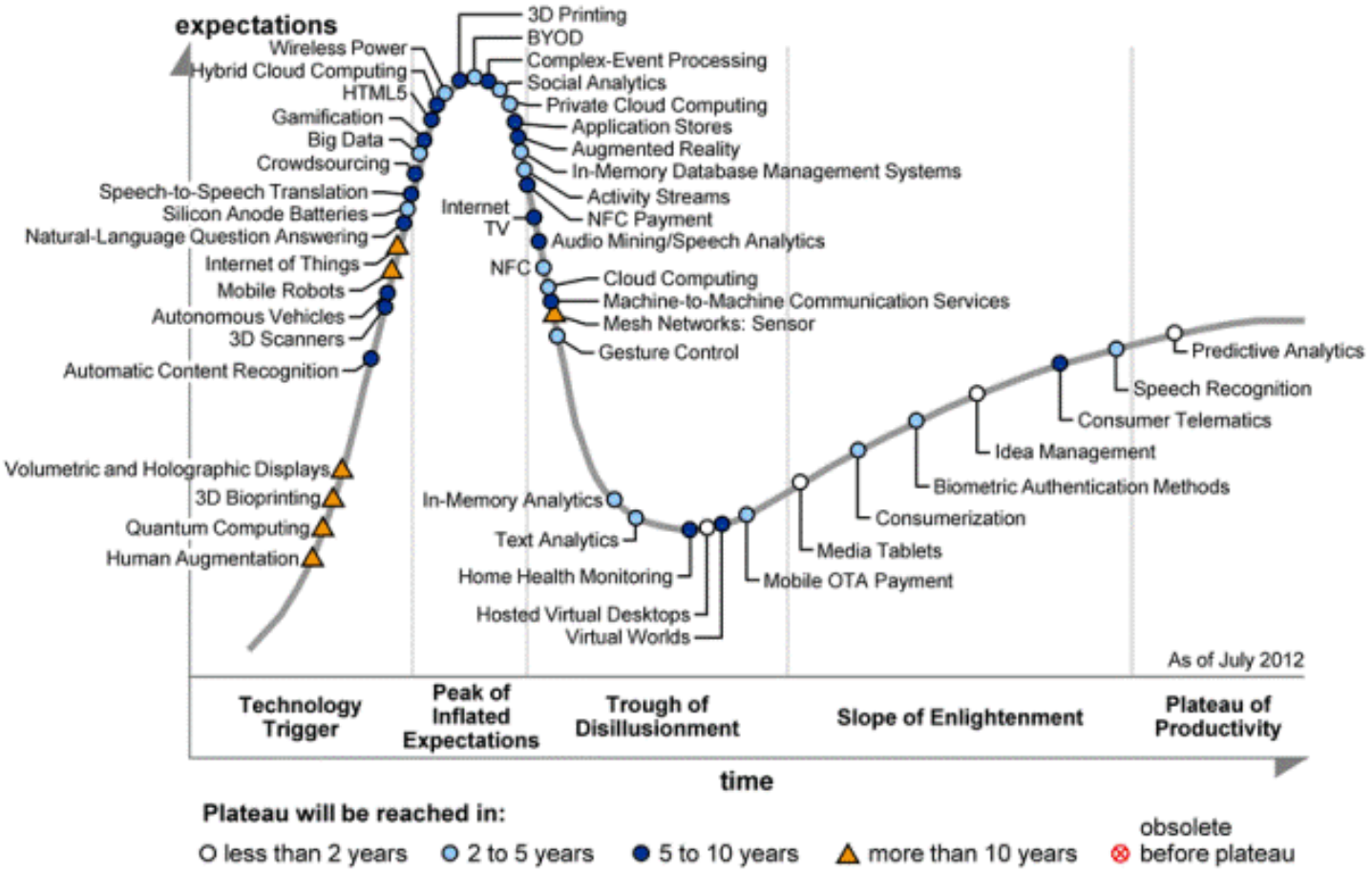
Arquitectura



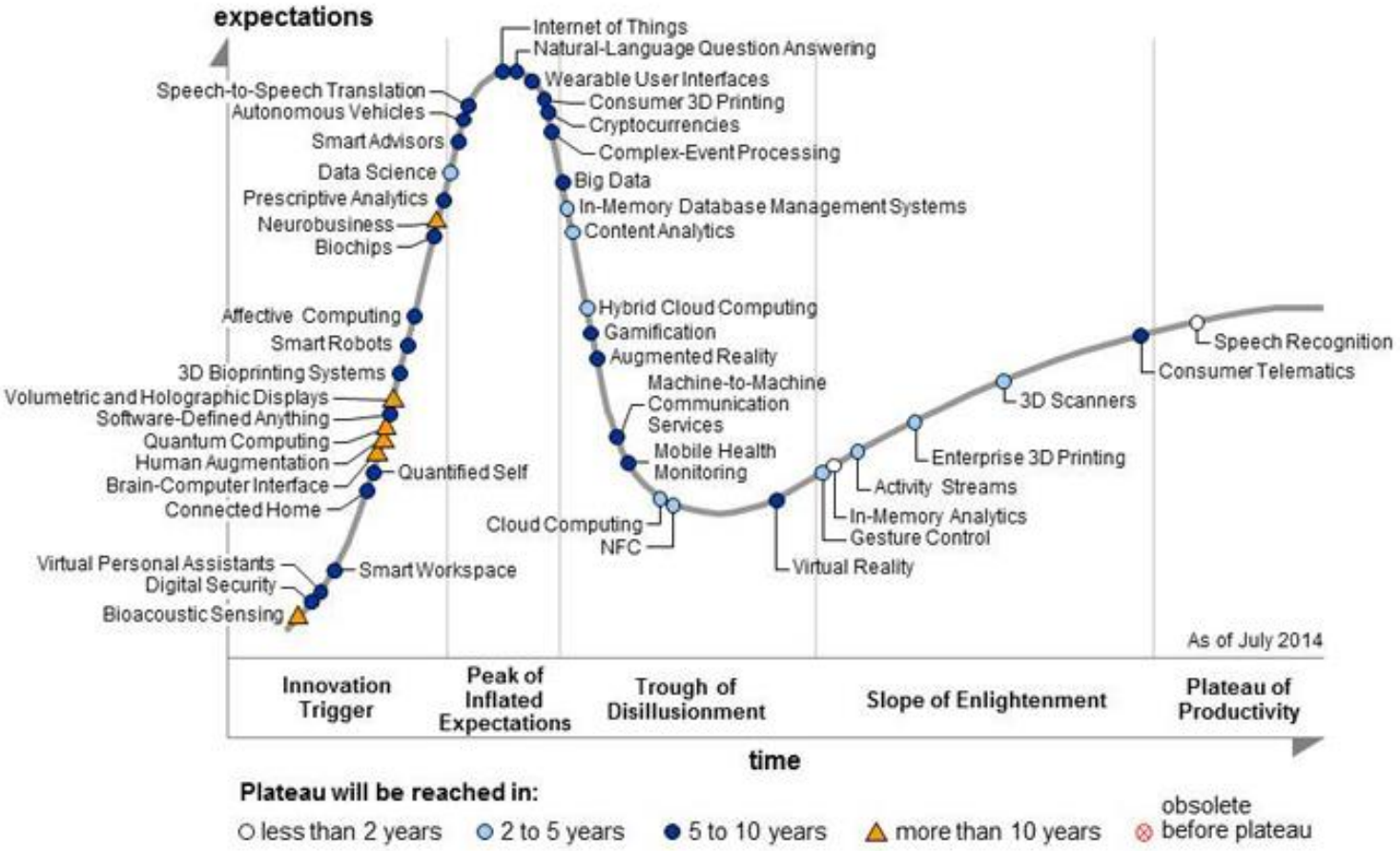
Arquitectura



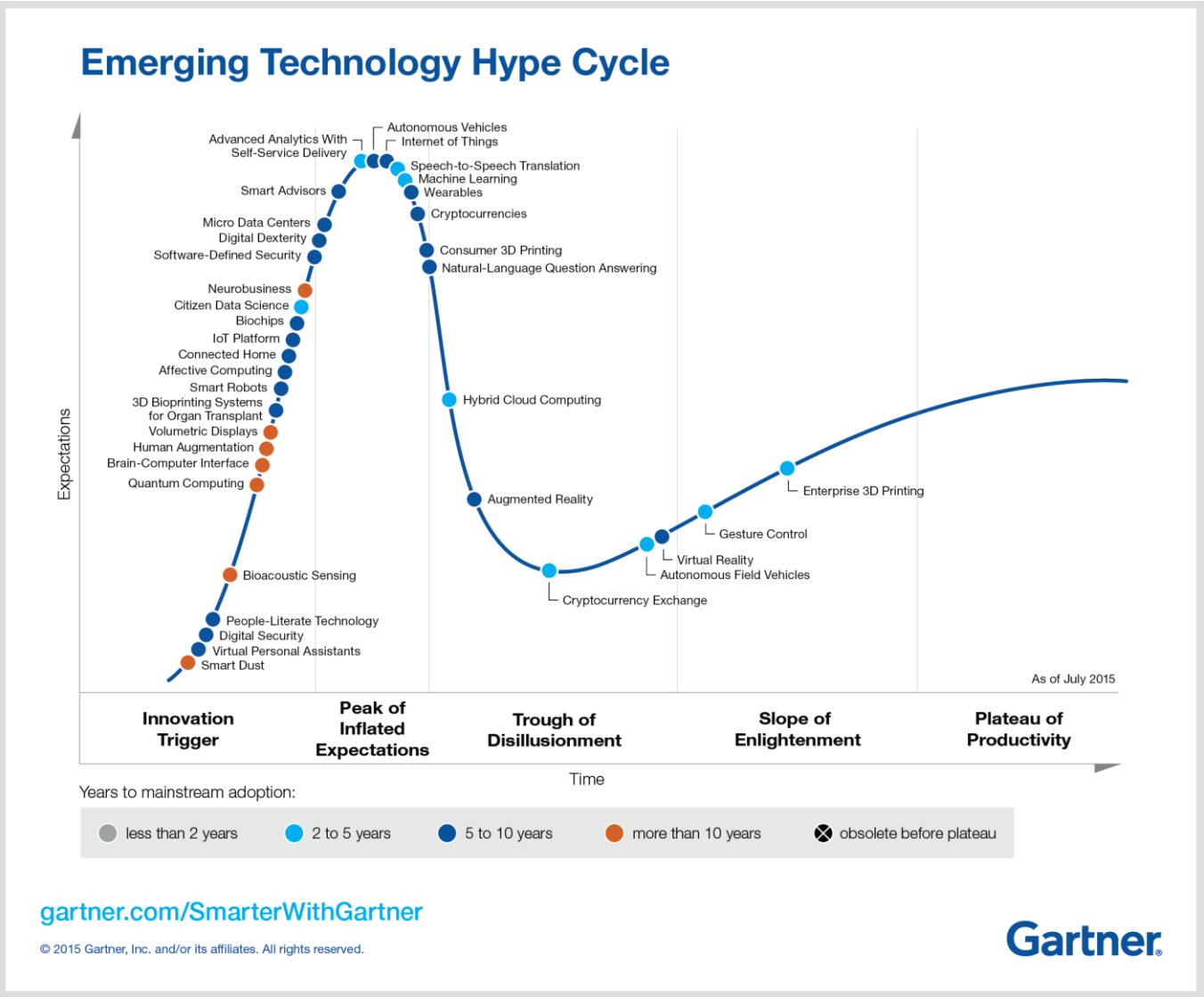
Situación


expectations
Wireless Power
Hybrid Cloud Computing
HTML5
Gamification
Big Data
Crowdsourcing
Speech-to-Speech Translation
Silicon Anode Batteries
Natural-Language Question Answering
Internet of Things
Mobile Robots
Autonomous Vehicles
3D Scanners
Automatic Content Recognition
Volumetric and Holographic Displays
3D Bioprinting
Quantum Computing
Human Augmentation
3D Printing
BYOD
Complex-Event Processing
Social Analytics
Private Cloud Computing
Application Stores
Augmented Reality
In-Memory Database Management Systems
Activity Streams
Internet TV
NFC Payment
Audio Mining/Speech Analytics
NFC
Cloud Computing
Machine-to-Machine Communication Services
Mesh Networks: Sensor
Gesture Control
In-Memory Analytics
Text Analytics
Home Health Monitoring
Hosted Virtual Desktops
Virtual Worlds
Mobile OTA Payment
Media Tablets
Consumerization
Biometric Authentication Methods
Idea Management
Consumer Telematics
Speech Recognition
Predictive Analytics
As of July 2012
Technology Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau

Situación


expectations
Internet of Things
Natural-Language Question Answering
Wearable User Interfaces
Consumer 3D Printing
Cryptocurrencies
Complex-Event Processing
Big Data
In-Memory Database Management Systems
Content Analytics
Hybrid Cloud Computing
Gamification
Augmented Reality
Machine-to-Machine Communication Services
Mobile Health Monitoring
Cloud Computing
NFC
Virtual Reality
In-Memory Analytics
Gesture Control
Activity Streams
Enterprise 3D Printing
3D Scanners
Consumer Telematics
Speech Recognition
Speech-to-Speech Translation
Autonomous Vehicles
Smart Advisors
Data Science
Prescriptive Analytics
Neurobusiness
Biochips
Affective Computing
Smart Robots
3D Bioprinting Systems
Volumetric and Holographic Displays
Software-Defined Anything
Quantum Computing
Human Augmentation
Brain-Computer Interface
Connected Home
Quantified Self
Virtual Personal Assistants
Digital Security
Bioacoustic Sensing
Smart Workspace
As of July 2014
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau

Situación


Emerging Technology Hype Cycle
Expectations
Advanced Analytics With Self-Service Delivery
Autonomous Vehicles
Internet of Things
Speech-to-Speech Translation
Machine Learning
Wearables
Smart Advisors
Cryptocurrencies
Micro Data Centers
Digital Dexterity
Software-Defined Security
Consumer 3D Printing
Natural-Language Question Answering
Neurobusiness
Citizen Data Science
Biochips
IoT Platform
Connected Home
Affective Computing
Smart Robots
3D Bioprinting Systems for Organ Transplant
Volumetric Displays
Human Augmentation
Brain-Computer Interface
Quantum Computing
Hybrid Cloud Computing
Augmented Reality
Enterprise 3D Printing
Gesture Control
Virtual Reality
Autonomous Field Vehicles
Cryptocurrency Exchange
Bioacoustic Sensing
People-Literate Technology
Digital Security
Virtual Personal Assistants
Smart Dust
As of July 2015
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
Time
Years to mainstream adoption:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau
gartner.com/SmarterWithGartner
© 2015 Gartner, Inc. and/or its affiliates. All rights reserved.
Gartner