

# Big data

---

*PROF. D. HERREMANS*

*DORIENHERREMANS.COM*

50.038 Computational data science

# Big data?

---

- Incremental increase of data
  - IBM estimates that 90% of all data was created in the last two years
  - What is this data?
    - Logs from phone companies (e.g. which towers you connect to), websites
    - Medical data such as X-rays
    - Stock transactions
    - Store order data
    - Social network behaviour
    - ...
  - Growth of data: <https://cloudtweaks.com/2013/11/evolution-of-big-data/>
- We need to be able to store and process it

# What is big data?

---

- First mention:

*“...provides an interesting challenge for computer systems: data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of big data. When data sets do not fit in main memory (in core), or when they do not fit even on local disk, the most common solution is to acquire more resources.”*

In a 1997 paper by NASA scientists Michael Cox and David Ellsworth, when describing a problem they had with visualization

# What is big data?

---

- From around 2008 the term was popularized:

“Data of a very large size, typically to the extent that its manipulation and management present significant logistical challenges.” – Oxford dictionary

“The belief that the more data you have the more insights and answers will rise automatically from the pool of ones and zeros.”

“A new attitude by businesses, non-profits, government agencies, and individuals that combining data from multiple sources could lead to better decisions.”

# What is big data?

---

- ***Too big to be processed on a single machine***
- *Can be structured/unstructured/semi-structured*
- *Lately: refers to predictive analytics*
  
- Challenges: ?
  - Most data is worthless?
  - Data is created fast?
  - Data from different sources in different formats?

# What is big data?

---

- ***Too big to be processed on a single machine***
- *Can be structured/unstructured/semi-structured*
- *Lately: refers to predictive analytics*
  
- Challenges: ?
  - Most data is worthless? 
  - Data is created fast? 
  - Data from different sources in different formats? 

# Applications of big data

---

- Healthcare & research
  - Education
  - Telecom
  - Media
  - IoT
  - Banks & Insurance
  - Retail
  - Manufacturing (automotive production)
  - Energy & Utilities
  - ...
- These system all face challenges (3Vs)

# The 3 V's of big data

---

- Challenges of big data is not just about the size:
  - **Volume:** size of the data
  - **Variety:** data comes from many different sources, in many different formats
  - **Velocity:** speed at which it is being generated and needs to be able to be processed

First defined by [Laney, 2001](#)

These days: fourth V:

- **veracity:** trustworthiness of the data

# 1. Volume

---

- Increasing data amount:
  - Handling Petabytes instead of Megabytes
  - The estimated amount of "**data** in the digital universe" in 2010: 1.2 zettabytes (1.3 trillion gigabytes) – is equal to 75m fully-loaded 16GB iPads. Now, yottabytes.
- What produces data?
  - Internet of Things: mobile devices, cameras, microphones, RFID readers,...
- The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s (Hilbert, 2011)

# 1. Volume

---

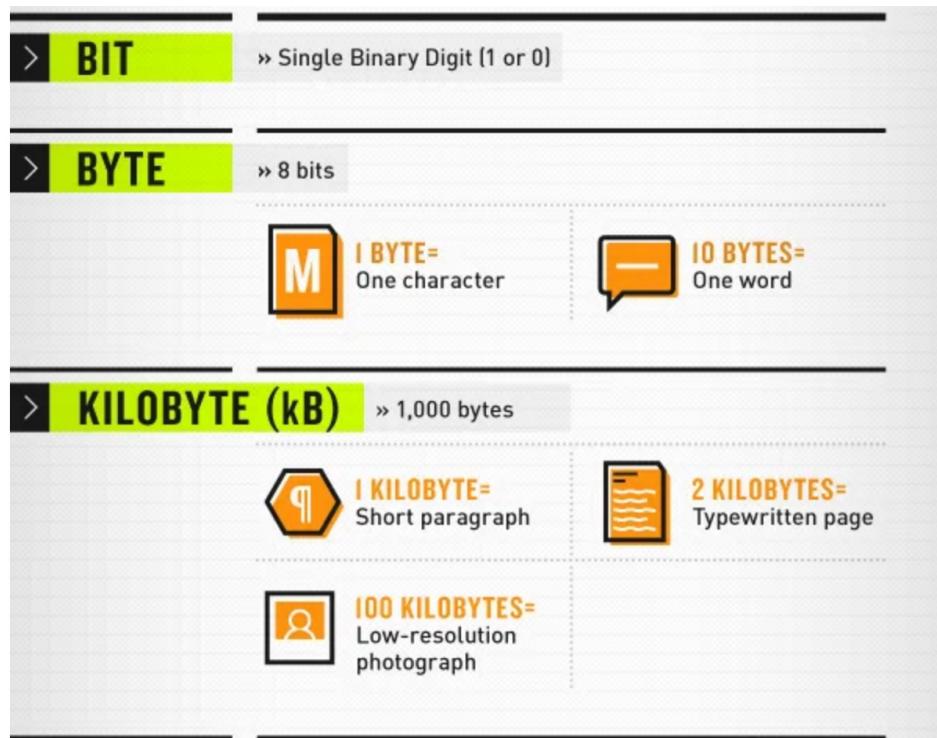
- Cost:
  - 1980: 1 GB → \$100,000 \$
  - 2017: < 0.008 \$

Note: ***reliable*** storage (e.g. Storage Area Network: SAN) is more expensive

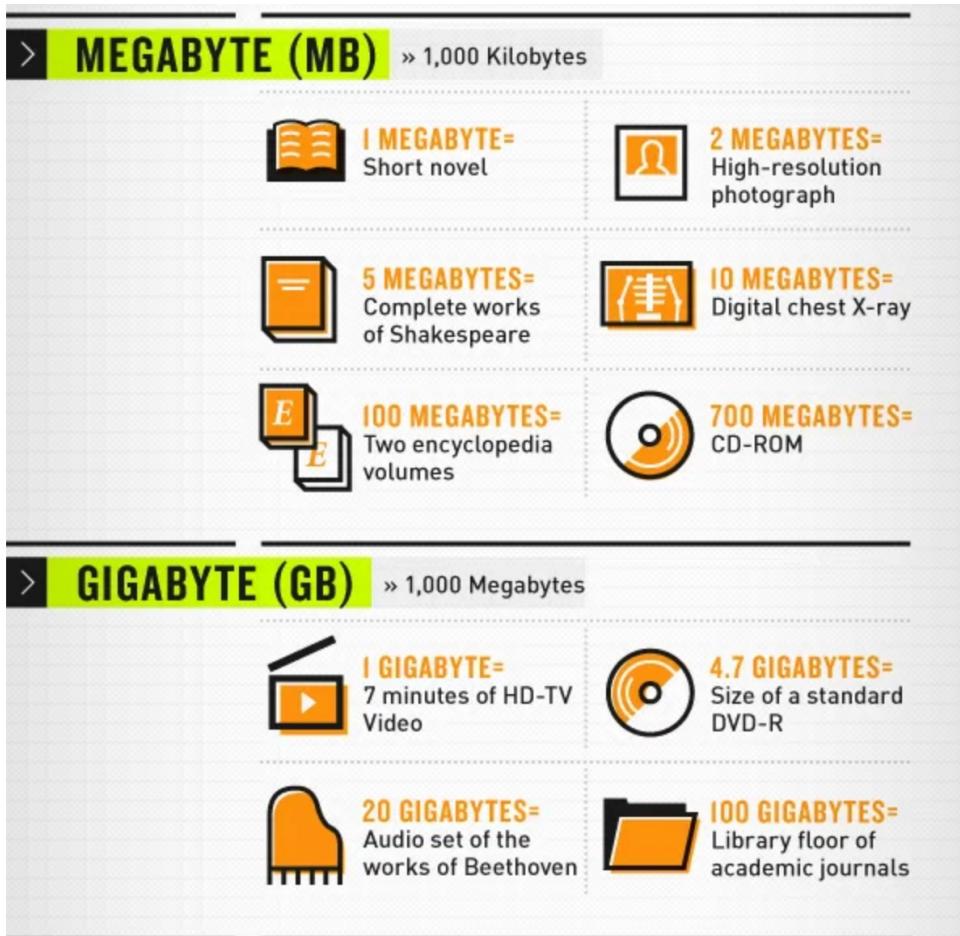
- Most data is useful, so we want to keep as much as possible
- Need a cheap way to ***store, read and process*** data
  - Storing on SAN is easy
  - Streaming this data over network: more difficult

# Data size matters

- Nice infographic on volume of data:
- <https://101.datascience.community/2013/11/15/data-size-matters-infographic/>



# Data size matters



# Data size matters

The infographic illustrates the scale of data storage units by comparing them to real-world examples:

- TERABYTE (TB)** » 1,000 Gigabytes
  - 1 TERABYTE =** 50,000 trees made into paper and printed
  - 10 TERABYTES =** Printed collection of the U. S. Library of Congress
- PETABYTE (PB)** » 1,000 Terabytes
  - 1 PETABYTE =** 20 million four-drawer filing cabinets filled with text
  - 1.5 PETABYTES =** All 10 billion photos on Facebook
  - 20 PETABYTES =** Daily amount of data processed by Google
  - 50 PETABYTES =** Entire written works of mankind, from the beginning of recorded history, in all languages

# Data size matters



## EXABYTE (EB)

» 1,000 Petabytes



### I EXABYTE =

Entire Netflix catalog streamed more than  
3,000 times



### 5 EXABYTE =

All the words ever spoken by mankind



## ZETTABYTE (ZB)

» 1,000 Exabytes



### I ZETTABYTE =

250 billion DVDs



## YOTTABYTE (YB)

» 1,000 Zettabytes



### I YOTTABYTE =

Size of the entire World Wide Web; it would take  
approximately 11 trillion years to download a  
Yottabyte file from the Internet using high-power  
broadband.

# Data size matters

02 HISTORY OF THE HARD DRIVE

Hard drives have increased 50-million-fold in the density of information they can hold since their introduction in 1956:

	<b>1956</b>	IBM 305 RAMAC – the first hard drive.		
	HOLDS 5 MB OF DATA	WEIGHS <b>1 TON</b>	COSTS PER MEGABYTE <b>\$10,000</b>	
	SIZE OF TWO REFRIGERATORS			
	» <b>1963</b>	IBM 1311 – the first removable hard drive.		
	» <b>1980</b>	IBM 3380 – the first gigabyte hard drive.		
	HOLDS <b>1 GB OF DATA</b>	COSTS <b>\$40,000</b>		
	<b>1992</b>	Hewlett-Packard C3013A Kitty Hawk – the first to break 2 GB barrier.		
	HOLDS <b>2.1 GB OF DATA</b>			

# Data size matters

» **1997** IBM Deskstar 16GP Titan – the first drive to use GMR (giant magnetoresistive) heads.  
HOLDS **16.8 GB OF DATA**

» **1998** IBM Microdrive – the smallest-sized hard drive to date.  
HOLDS **340 MB OF DATA**

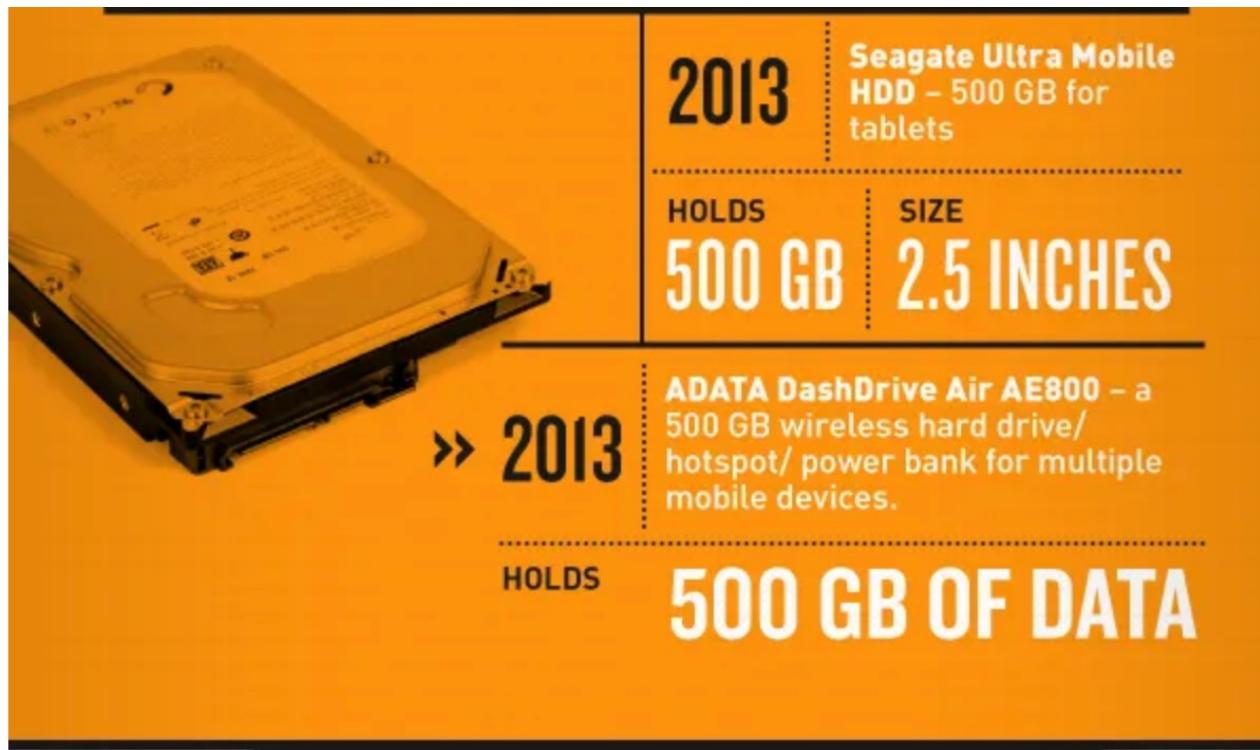
» **2004** Toshiba MK2001MTN – the first 0.85-inch hard drive.  
HOLDS **2 GB OF DATA**

**2006** Seagate Barracuda 7200.10  
HOLDS **750 GB OF DATA**

» **2007** Hitachi GST Deskstar 7K1000 – the first hard drive to break the 1 TB capacity mark.  
HOLDS **1 TB OF DATA**

**2011-2012** All three major hard drive makers – Seagate, Western Digital, and Toshiba – start shipping 4 TB hard drives.

# Data size matters



## > NEW HARD DRIVE TECHNOLOGIES



### HELIUM-FILLED DRIVES

Removes the friction and fluttering of platters as they spin at high speed, allowing drives to fit more platters in a given space.



### SHINGLED MAGNETIC RECORDING (SMR)

The tracks of a drive overlap like shingles on a roof, allowing a hard drive to have more tracks (and thus, more data).



### HEAT-ASSISTED MAGNETIC RECORDING (HAMR)

Allows data to be written more compactly by raising the temperature of the material that can be read by a magnetic field.

» **2013**

**Western Digital** experiments with helium-filled drives, which could offer a capacity of

**5.6 TB**

» **2014**

**Seagate's SMR** technology is predicted to allow hard drives to reach capacities of

**5 TB**

» **2020**

**Seagate's HAMR** technology is predicted to allow hard drives to reach capacities of

**20 TB**

# 2. Variety

---

- What do we want to store?  
Transactions, logs, business, user, sensor, medical, social,...
- SQL: extremely structured/predefined
- BUT: Estimated 90% of data is ‘unstructured’ or semi-structured, e.g.
  - Bank may have scans of receipts that look differently
- We want to store the original data:
  - E.g. transcribing a helpdesk conversation to text may be more efficient, but audio will contain tone of voice → different interpretations

=> This is where **Hadoop** comes in (versus a structured database system):

  - Allows you to store raw data + manipulate and reformat later
  - From SQL blob to Hadoop

# Types of data

---

- **Structured:**
  - Represented in strict format, e.g. SQL DB
  - Necessary for operations, e.g. sales transaction
- **Unstructured:**
  - No predefined data model, not organized in pre-defined manner
  - Often interesting for data mining
  - E.g. text, numbers, files,... can be all mixed. -> Hadoop
- **Semi-structured:**
  - May have a certain structure, but not all data has necessarily identical structure
  - No predefined schema
  - Schema information can be mixed with data values, since each data object may have attributes that are not known in advance, e.g. xml, json -> MongoDB

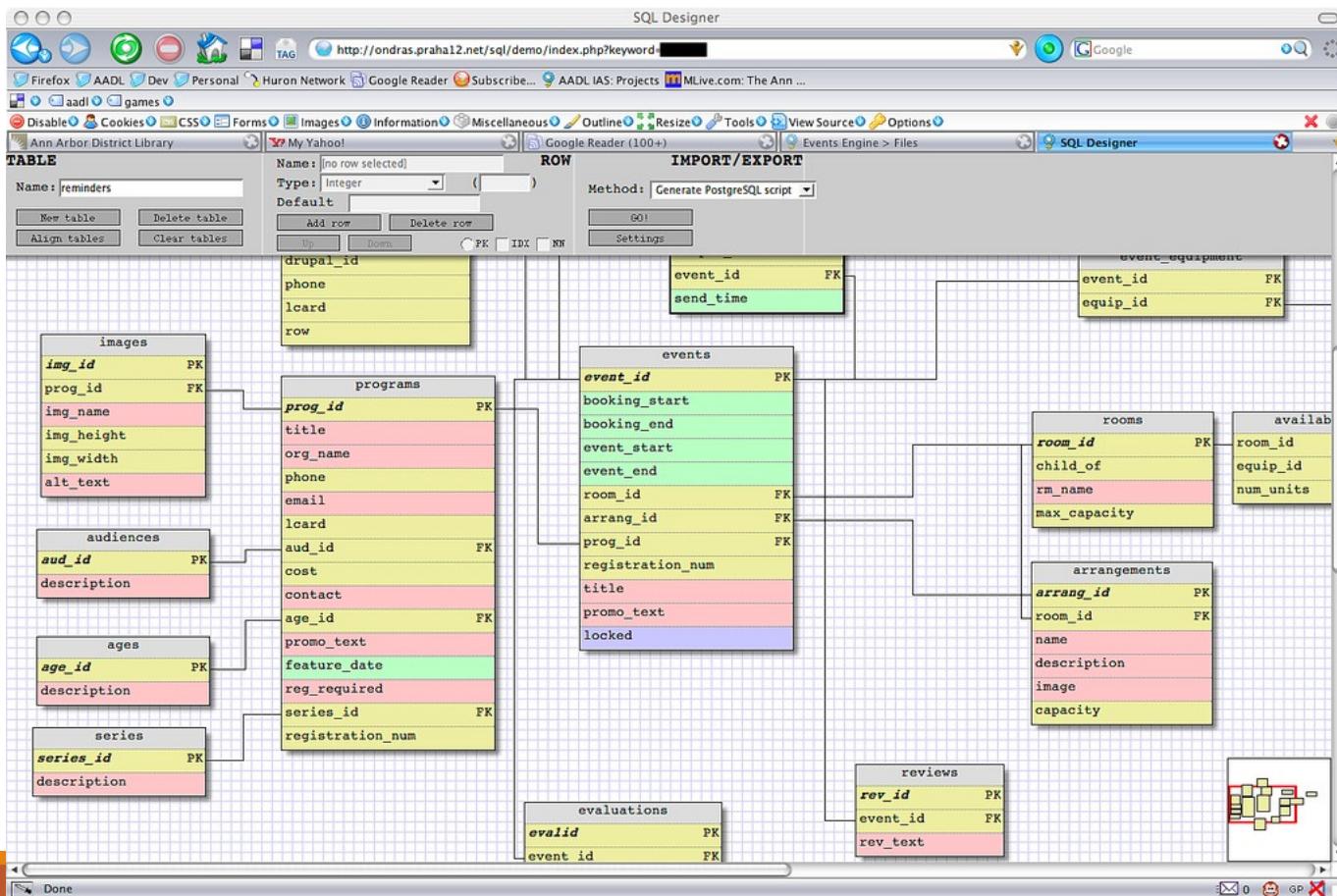
# JSON example

- example from Wikipedia
- no fixed format
- semi-structured, key-value pairs, hierarchical
- “friendly” alternative to XML:
  - <https://aws.amazon.com/compare/the-difference-between-xml-and-json/>
- self-documenting structure

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "isAlive": true,  
  "age": 25,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": "10021-3100"  
  },  
  "phoneNumbers": [  
    {  
      "type": "home",  
      "number": "212 555-1234"  
    },  
    {  
      "type": "office",  
      "number": "646 555-4567"  
    }  
  ],  
  "children": [],  
  "spouse": null  
}
```

# SQL

- Structured, relational database



# Transactional data

---

## Transaction\_Details

ID	Transaction_Header_ID	Gross_Amount
5434125	4526622	-300.00
5420422	4513202	279.72
5415415	4508216	300.00
5413258	4506059	279.72
5434502	4526964	-279.72
5415438	4508239	279.72

## Transaction\_Headers

ID	Payment_Type_Id	GL_Date
4526622	26	2013-10-10 00:00:00.000
4513202	8	2013-10-05 00:00:00.000
4508216	1	2013-12-01 00:00:00.000
4506059	1	2013-01-01 00:00:00.000
4526964	26	2014-12-09 00:00:00.000
4508239	1	2014-12-09 00:00:00.000

## Bill\_Mapping

ID	Trans_ID_Dr	Trans_ID_Cr	Amount
2865991	5420422	5434125	279.72
2865992	5415415	5434125	20.28
2866486	5415438	5434502	279.72

# Twitter data

Tweets	Tweets & replies	Media	Likes
 <b>Dorien Herremans</b> @dorienherremans · Aug 27 Congrats to @ravencheuk for publishing nnAudio in IEEE Access. Anyone interested in a GPU toolbox for on-the-fly spectrogram extraction should check it out #ai #dsp #audio #ismir #music #signalprocessing #PyTorch <a href="http://dorienherremans.com/content/nnaudi...">dorienherremans.com/content/nnaudi...</a>	  1  7 		
 <b>Dorien Herremans</b> @dorienherremans · Aug 12 We have a job opening for a wizard in NLP/sequential models for finance #lstm #pytorch #bert #finance #fintech #singapore <a href="http://dorienherremans.com/content/resear...">dorienherremans.com/content/resear...</a>	   3 		
 <b>Dorien Herremans</b> @dorienherremans · Jul 23 Nice PyTorch walk through for attention #attention #pytorch #ai #nlp <a href="http://nlp.seas.harvard.edu/2018/04/03/att...">nlp.seas.harvard.edu/2018/04/03/att...</a>	  1  		
 <b>Dorien Herremans</b> @dorienherremans · Jul 23 Nice text on going from word2vec to transformer models #ai #nlp #word2vec #bert #machinelearning			

# IP logs

Screenshot of the ELSA (Event Log Search Application) interface showing search results for IP logs.

**Search Query:** srcip:10.124.19.12

**Time Range:** From 2011-11-21 22:05:51 To [empty]

**Fields Summary:** host(4) program(4) class(3) srcip(1) srcport(74) dstip(22) dstport(3) expiration(2) hostname(2) subject(2) proto(2) conn\_bytes(43) o\_int(2) l\_int(2) conn\_duration(17) status\_code(1) content\_length(20) country\_code(3) method(2) site(8) url(23) referer(7) user\_agent(1) domains(8)

**Records:** 100 / 4154 1486 ms

	Timestamp	Logs
Info	Tue Nov 22 08:53:20	1321973538.778549 vfLpkUrpoI6 10.124.19.12 47263 209.85.225.132 443 TLSv10 TLS_ECDHE_RSA_WITH_RC4_128_SHA s2.googleusercontent.com -CN=.googleusercontent.com,O.View,ST=California,C=US 1320932962.000000 1352555962.000000 0ef6837e26d2608700a9e03c863dafe ok host=165.189.226.172 program=bro_ssl class=BRO_SSL srcip=10.124.19.12 srcport=47263 dstip=209.85.225.132 dstport=443 expiration=1352555962 hostname=s2.googleusercontent.com s.View,ST=California,C=US
Info	Tue Nov 22 08:53:20	1321973537.891299 oE6L8vilUv7 10.124.19.12 41018 199.59.149.198 443 TLSv10 TLS_RSA_WITH_RC4_128_SHA twitter.com 970e68f4de429d78cdc280f310267aa67ee8530e8be2e3ec924 Inc.streetAddress=795 Folsom St, Suite 600,L=San Francisco,ST=California,postalCode=94107,C=US,serialNumber=4337446,2.5.4.15=#131450726976617465204F7267616E697A6174696F6E,1.3.6.1.4.1.311.60.2.1.2=#1308446 1310014800 host=165.189.226.172 program=bro_ssl class=BRO_SSL srcip=10.124.19.12 srcport=41018 dstip=199.59.149.198 dstport=443 expiration=1343451599 hostname=twitter.com subject=CN=twitter.com Folsom St, Suite 600,L=San Francisco,ST=California,postalCode=94107,C=US,serialNumber=4337446,2.5.4.15=#131450726976617465204F7267616E697A6174696F6E,1.3.6.1.4.1.311.60.2.1
Info	Tue Nov 22 08:53:25	Teardown UDP connection 144744478313156395 for DET-SEC-124.19.10.12/45091 to OUTSIDE:10.68.15.11/53 duration 0:02:03 bytes 213 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=45091 dstip=10.68.15.11 dstport=53 conn_bytes=213 o_int=DE
Info	Tue Nov 22 08:53:25	Teardown UDP connection 144744478313156396 for DET-SEC-124.19.10.12/52757 to OUTSIDE:10.68.15.11/53 duration 0:02:02 bytes 213 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=52757 dstip=10.68.15.11 dstport=53 conn_bytes=213 o_int=DE
Info	Tue Nov 22 08:53:26	Teardown UDP connection 144744478313156397 for DET-SEC-124.19.10.12/47309 to OUTSIDE:10.68.15.11/53 duration 0:02:03 bytes 217 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=47309 dstip=10.68.15.11 dstport=53 conn_bytes=217 o_int=DE
Info	Tue Nov 22 08:53:26	Teardown UDP connection 144744478313156398 for DET-SEC-124.19.10.12/52485 to OUTSIDE:10.68.15.11/53 duration 0:02:03 bytes 284 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=52485 dstip=10.68.15.11 dstport=53 conn_bytes=284 o_int=DE
Info	Tue Nov 22 08:53:26	Teardown UDP connection 144744478313156399 for DET-SEC-124.19.10.12/57404 to OUTSIDE:10.68.15.11/53 duration 0:02:03 bytes 172 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=57404 dstip=10.68.15.11 dstport=53 conn_bytes=172 o_int=DE
Info	Tue Nov 22 08:54:20	Teardown UDP connection 144744478313156408 for DET-SEC-124.19.10.12/35728 to OUTSIDE:10.68.15.11/53 duration 0:02:03 bytes 221 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=35728 dstip=10.68.15.11 dstport=53 conn_bytes=221 o_int=DE
Info	Tue Nov 22 08:54:20	Teardown UDP connection 144744478313156409 for DET-SEC-124.19.10.12/43103 to OUTSIDE:10.68.15.11/53 duration 0:02:03 bytes 221 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=43103 dstip=10.68.15.11 dstport=53 conn_bytes=221 o_int=DE
Info	Tue Nov 22 08:54:20	Teardown UDP connection 144744478313156410 for DET-SEC-124.19.10.12/51752 to OUTSIDE:10.68.15.11/53 duration 0:02:02 bytes 198 host=165.189.82.68 program=%fwsm-5-302016 class=FIREWALL_CONNECTION_END proto=UDP srcip=10.124.19.12 srcport=51752 dstip=10.68.15.11 dstport=53 conn_bytes=198 o_int=DE

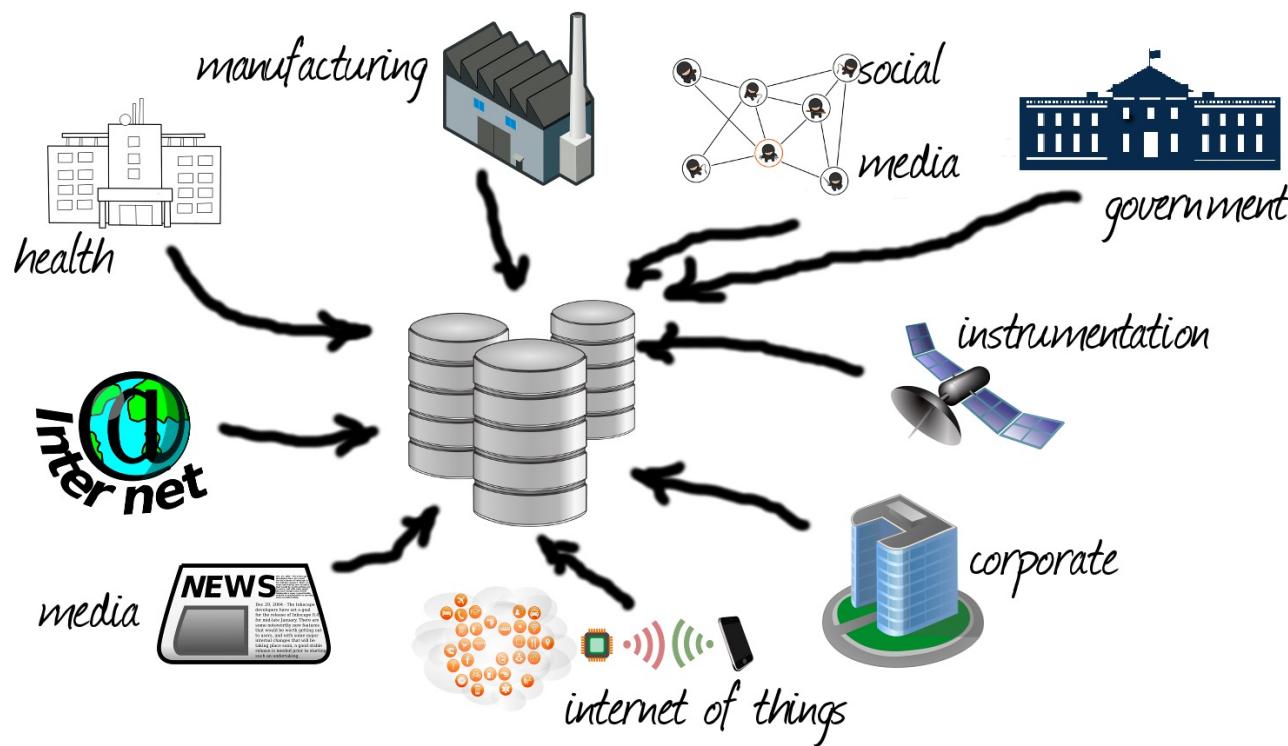
# Example of variety

---

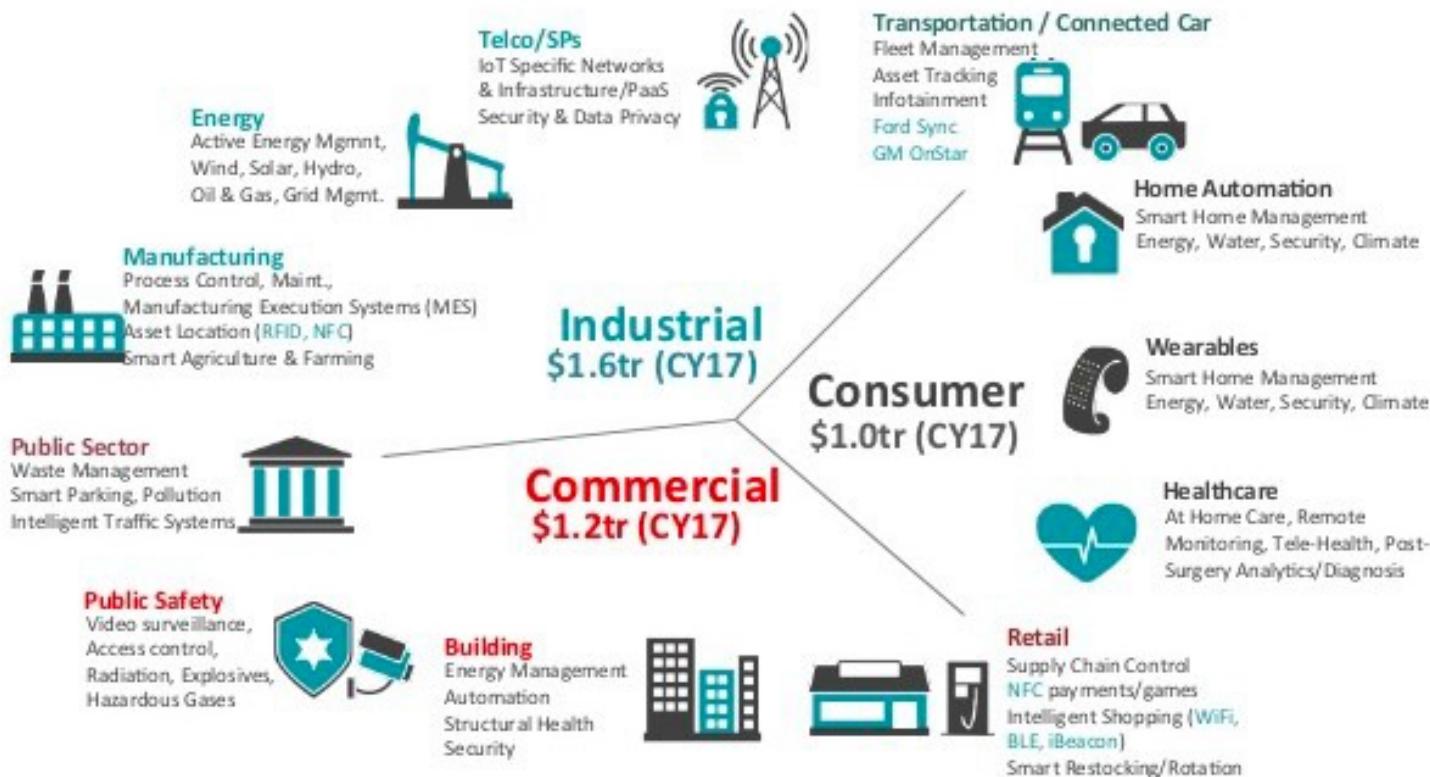
- Delivery company with trucks
- To pick up a new load from a depot, do we just send the closest truck?
  - Current GPS
  - Current map: small roads slower?
  - Traffic data
  - Current load
  - Fuel efficiency
  - Space available in truck (dimensions of other packages)
  - ...

# Sources of data

---



# Internet of things

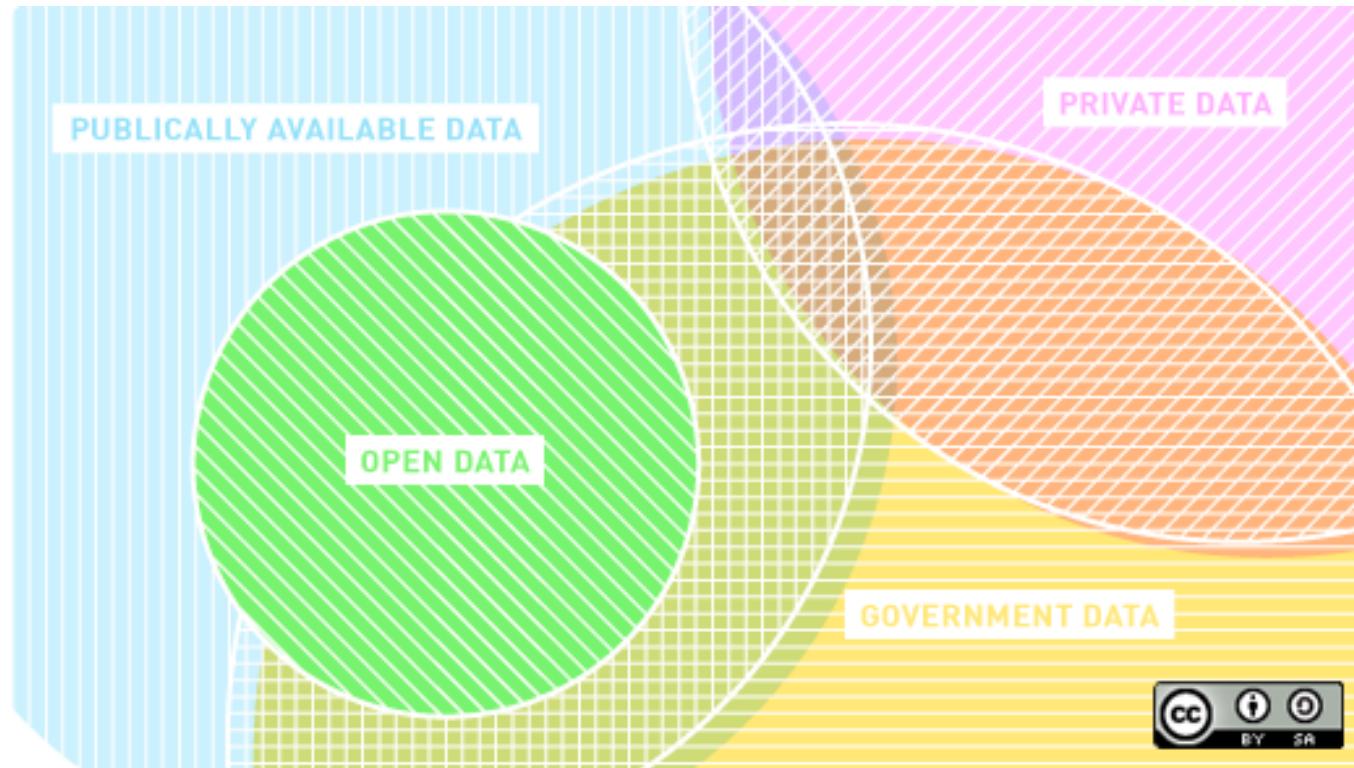


Source: IDC Internet of Things Spending Guide by Vertical Market 2014

# Sources of data

---

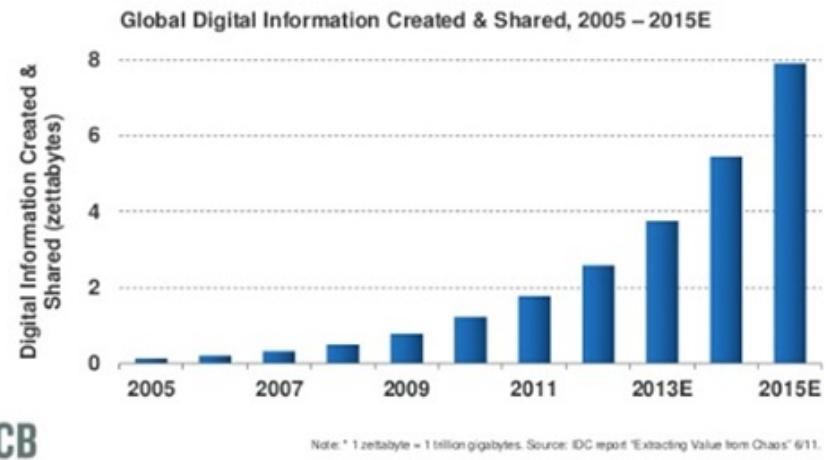
- Some of the best data is open data, e.g. government, Kaggle, etc.



# 3. Velocity

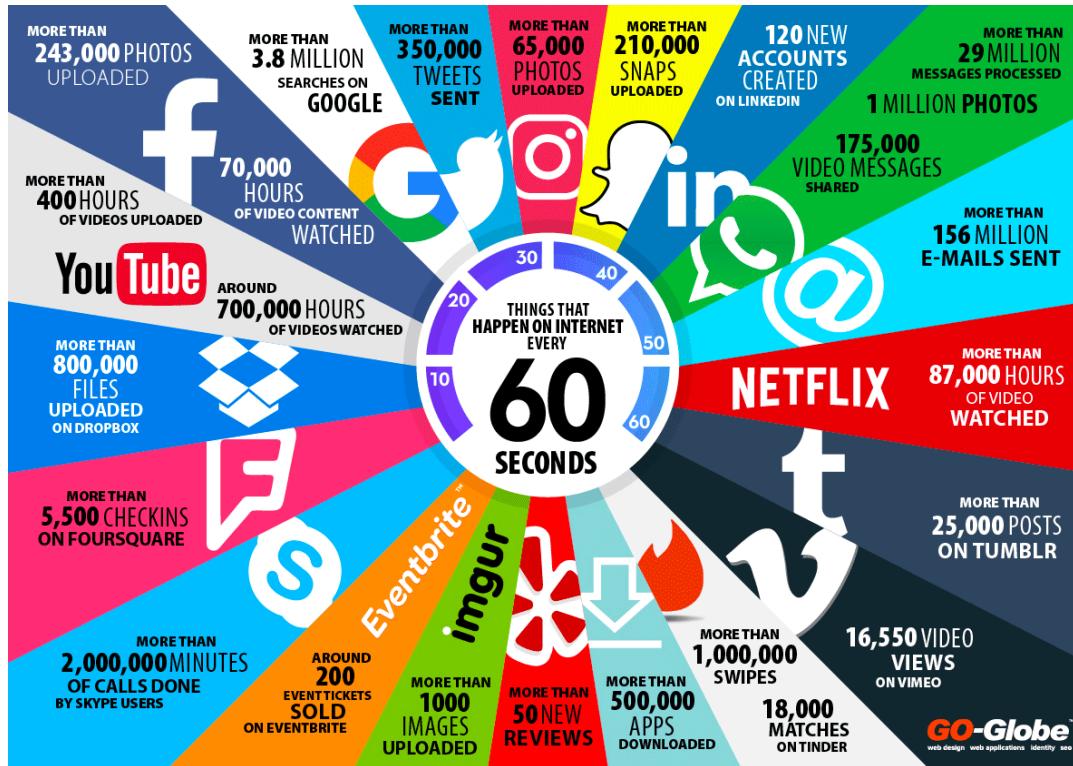
---

- Estimates say each day several exabytes (1 exabyte = 1 billion gigabytes) of data is created every day.
- We need to be able to store data as it arrives
- Why store so much data? E.g. product recommendations



# 60 seconds

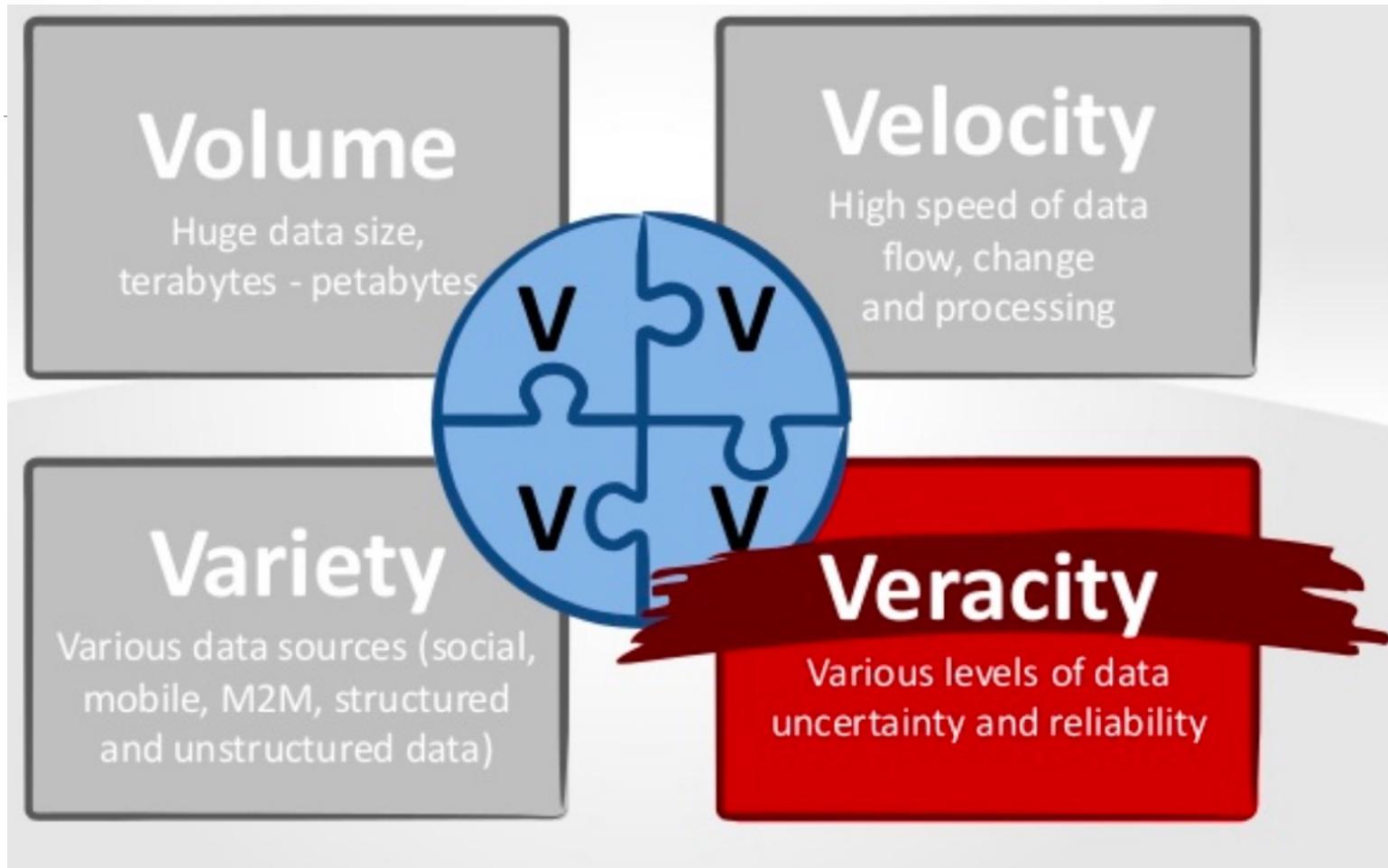
- 2016 infographic – things that happen every 60 seconds
- <https://www.go-globe.com/60-seconds/>



# 4. Veracity

---

- Various data uncertainty and reliability
- Imprecision of data (especially unstructured data), e.g. text message can have double meaning
- Different level of data quality of structured data (certain and precise) versus unstructured data (fuzzy interpreting of images, free text, speech...)
- Technical data quality issues (various formats, source availability, updates)
- → 5<sup>th</sup> V: value...



Source: InfoDiagram

# CAP theorem

---

- Eric Brewer:

*“It’s impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees”:*

**1. Consistency:**

every read receives the most recent write or an error

**2. Availability:**

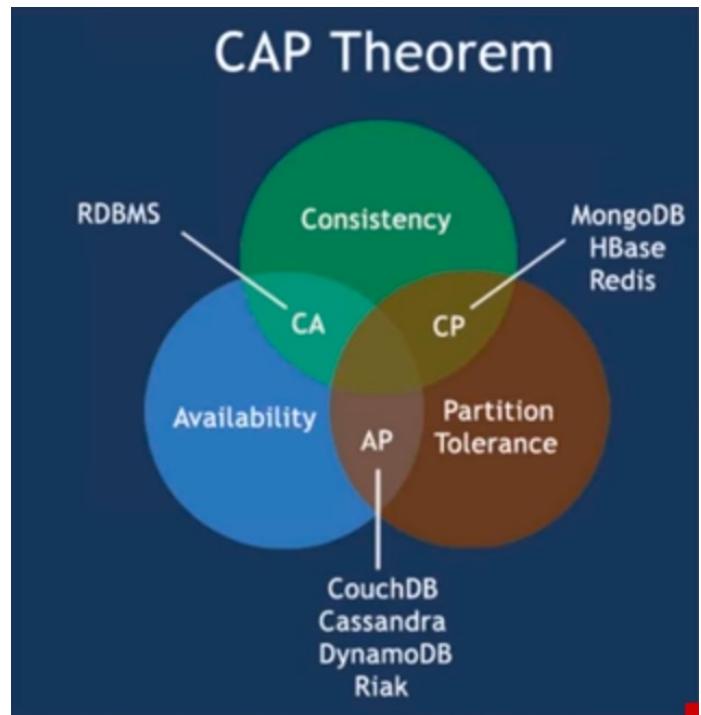
every request receives a non-error response  
(without guarantee that it’s the most recent write)

**3. Partition tolerance:**

The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes. Partition = communications break

# CAP theorem

- CA: Single cluster, all nodes are always in contact. When a partition between nodes should be created the data is out of sync until partition is resolved.
- CP: Some data may not be accessible, but the rest is still consistent/accurate
- AP: System is still available under partitioning, but some of the data returned may be inaccurate. Will resync data once the partition is resolved



# PACELC theorem

---

- Daniel Abadi:

*"if there is a **partition (P)**, how does the system trade off **availability** and **consistency (A and C)**; else (E), when the system is running normally in the absence of partitions, how does the system trade off **latency (L)** and **consistency (C)**?".*

-> *Expansion of the earlier defined CAP theorem*

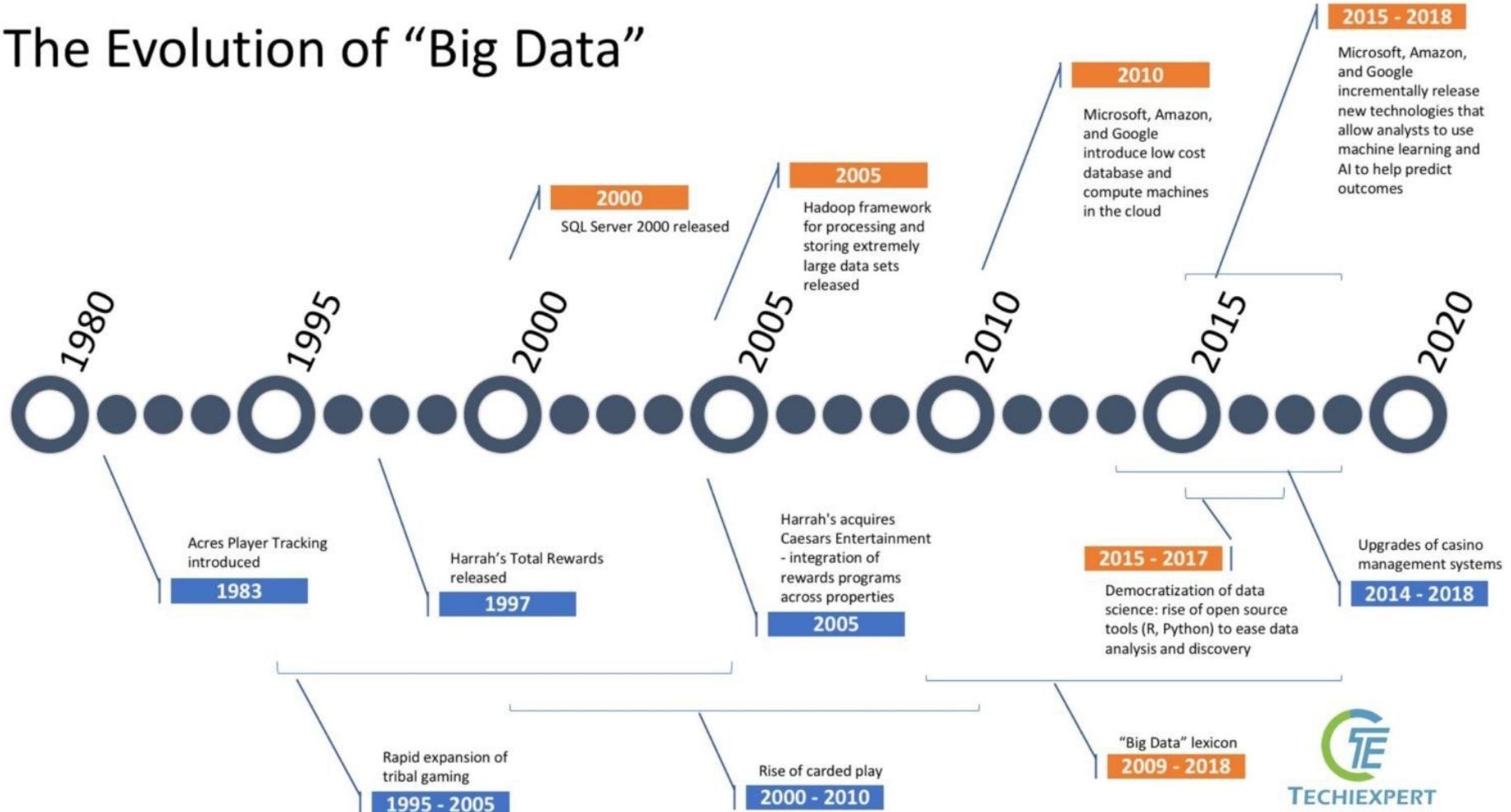
A high availability requirement implies that the system must replicate data. As soon as a **distributed system** replicates data, a tradeoff between consistency and latency arises.

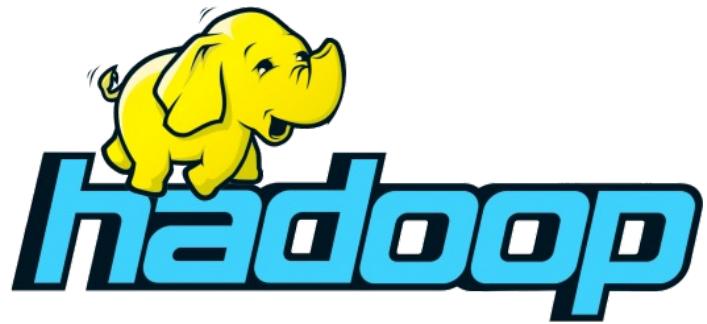
# PACELC

---

DDBS	P+A	P+C	E+L	E+C
DynamoDB	Yes		Yes <sup>[a]</sup>	
Cassandra	Yes		Yes <sup>[a]</sup>	
Cosmos DB	Yes		Yes	
Riak	Yes		Yes <sup>[a]</sup>	
VoltDB/H-Store		Yes		Yes
Megastore		Yes		Yes
BigTable/HBase		Yes		Yes
MongoDB	Yes			Yes
PNUTS		Yes	Yes	
Hazelcast IMDG <sup>[6]</sup>	Yes		Yes	Yes

# The Evolution of “Big Data”





& MAP REDUCE

# Hadoop

---

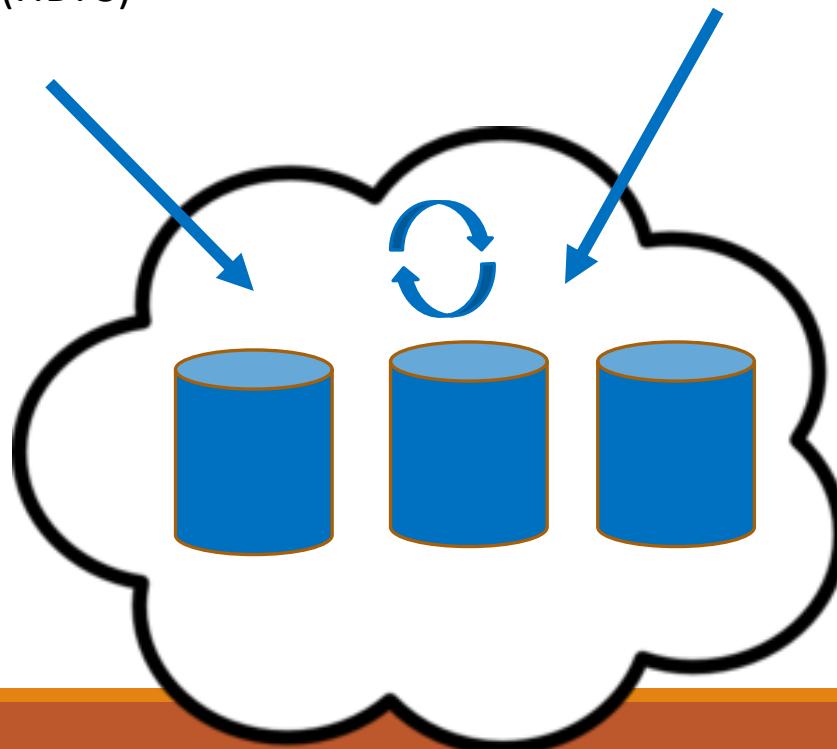
- *"An open-source software framework used for distributed storage and processing of dataset of big data using the MapReduce programming model."*
- 2 parts:
  - Storage: Hadoop Distributed File System (HDFS)
  - Processing: MapReduce  
→ manipulates data where it is stored (data locality principle)
- Inspired by Google File System
- Named after toy elephant of Doug Cutting's son (co-founder together with Mike Cafarella)
- Coded in java

# Core Hadoop

---

**Store**  
In  
Hadoop File  
System (HDFS)

**Process**  
with  
MapReduce



# Hadoop ecosystem

---

- Easier to use:
- Writing MapReduce code isn't very simple, many people know SQL.

Tools that help other applications integrate Hadoop:

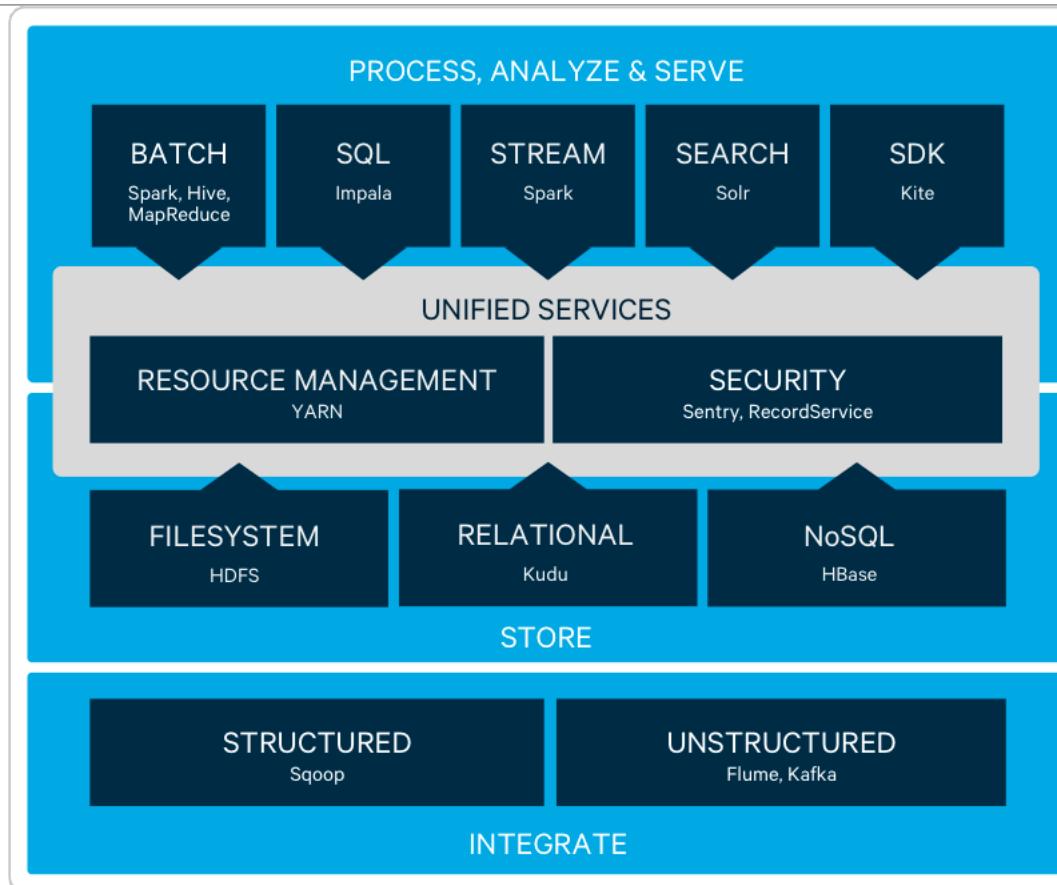
- Hive:
  - Use statements like: "select \* from..."
  - Looks like SQL, but Hive transforms this to Map reduce code
- Pig:
  - Analyze data using a simple scripting language
  - This is then transformed into Map reduce code.

# Hadoop ecosystem

---

- Impala:
    - Query data with SQL directly (no MapReduce)
  - Sqoop:
    - Takes data from SQL DB and puts it into HDFS as delimited files
  - Hue:
    - Graphical interface to the cluster
  - Mahout:
    - Machine Learning library
  - Hbase:
    - Real time database built on top of HDFS
- Cloudera Hadoop Distribution (CDH) integrates all these tools

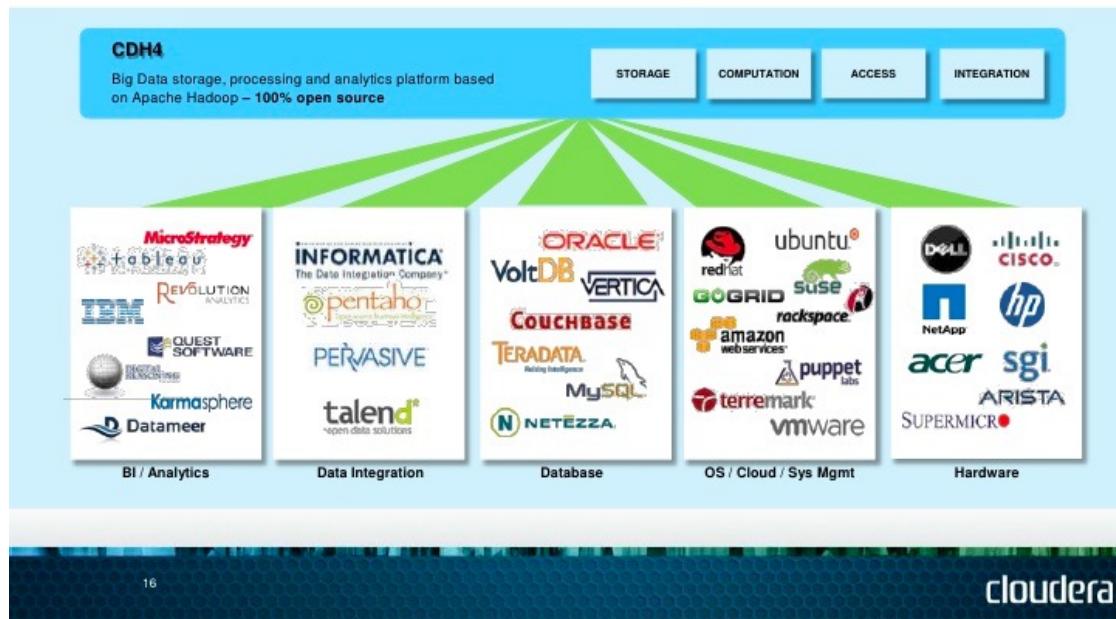
# Cloudera ecosystem



# Cloudera ecosystem

## CLOUDERA'S PARTNER ECOSYSTEM: WIDEST INTEGRATION

All the industry leaders develop on CDH.



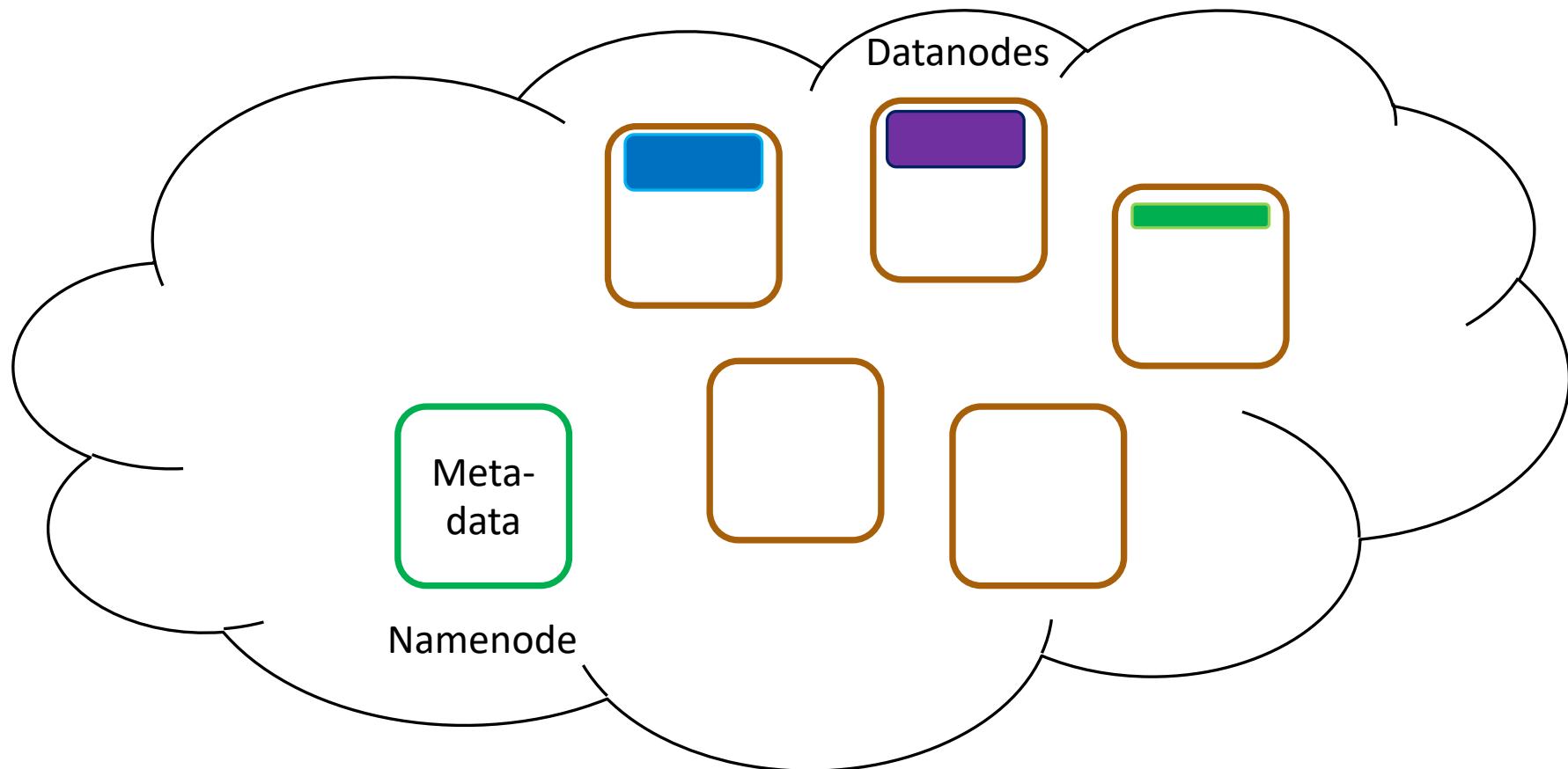
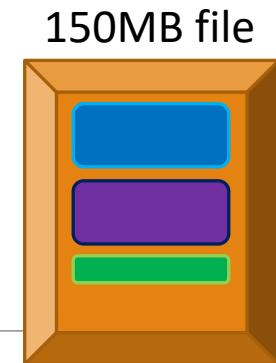
# Hadoop Distributed File System (HDFS)

---

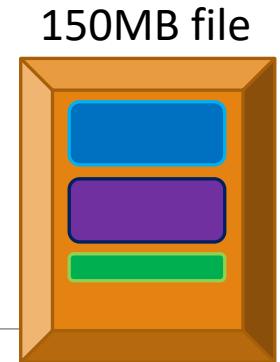
- Looks very much like a normal file system
- Split into blocks (blk\_1, blk\_2...)
- 64MB default block size
- E.g. 150MB file split in blocks of 64 -64 – 22
- Each block stored in a node
- Daemon runs on each of the machines of cluster:
  - Name node: stores metadata (where each block is stored)
  - Data node: actual data blocks

# HDFS

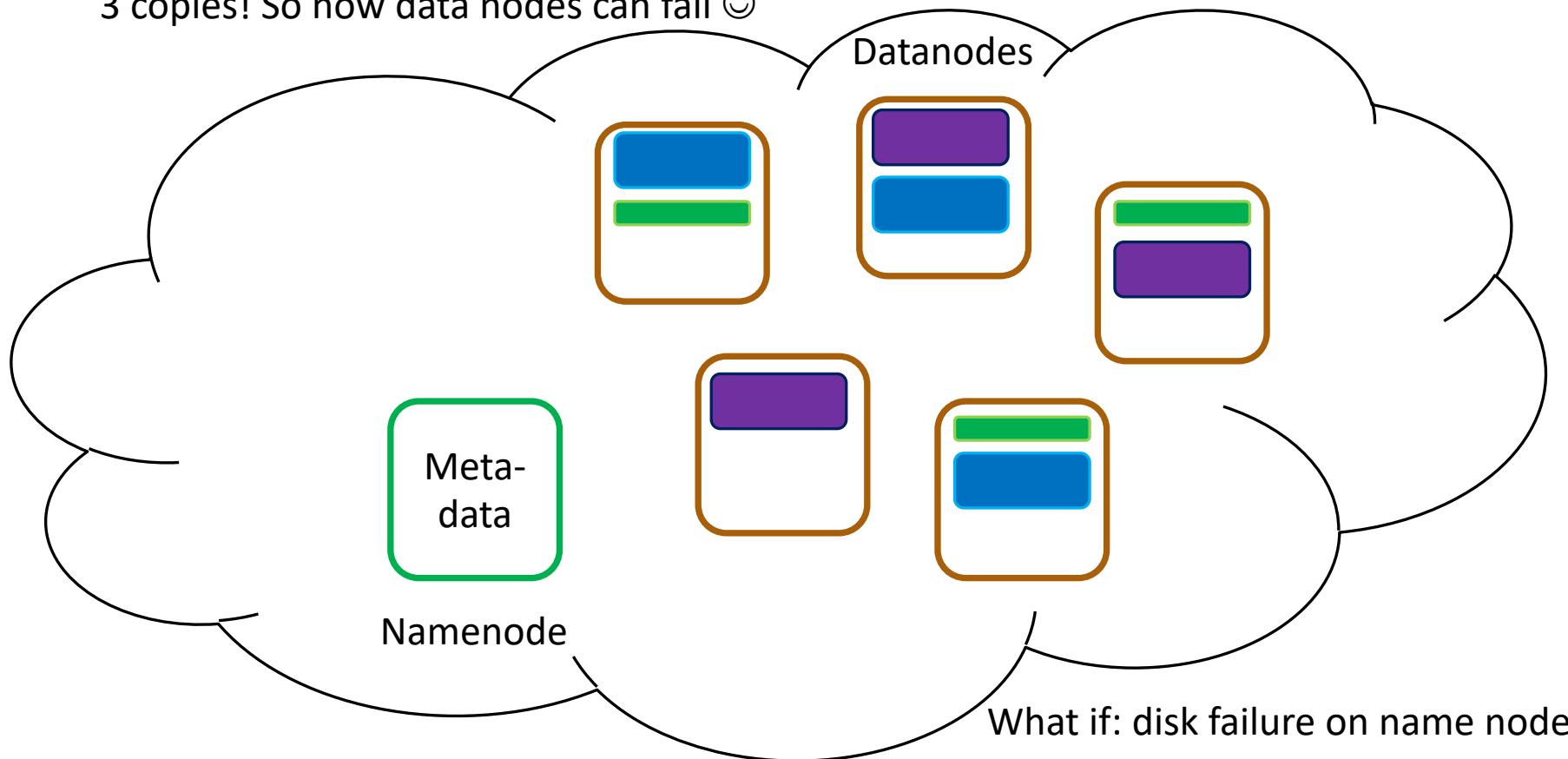
What if: disk failure on a data node?

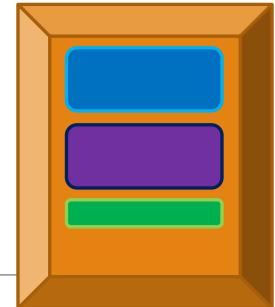


# HDFS: data redundancy!



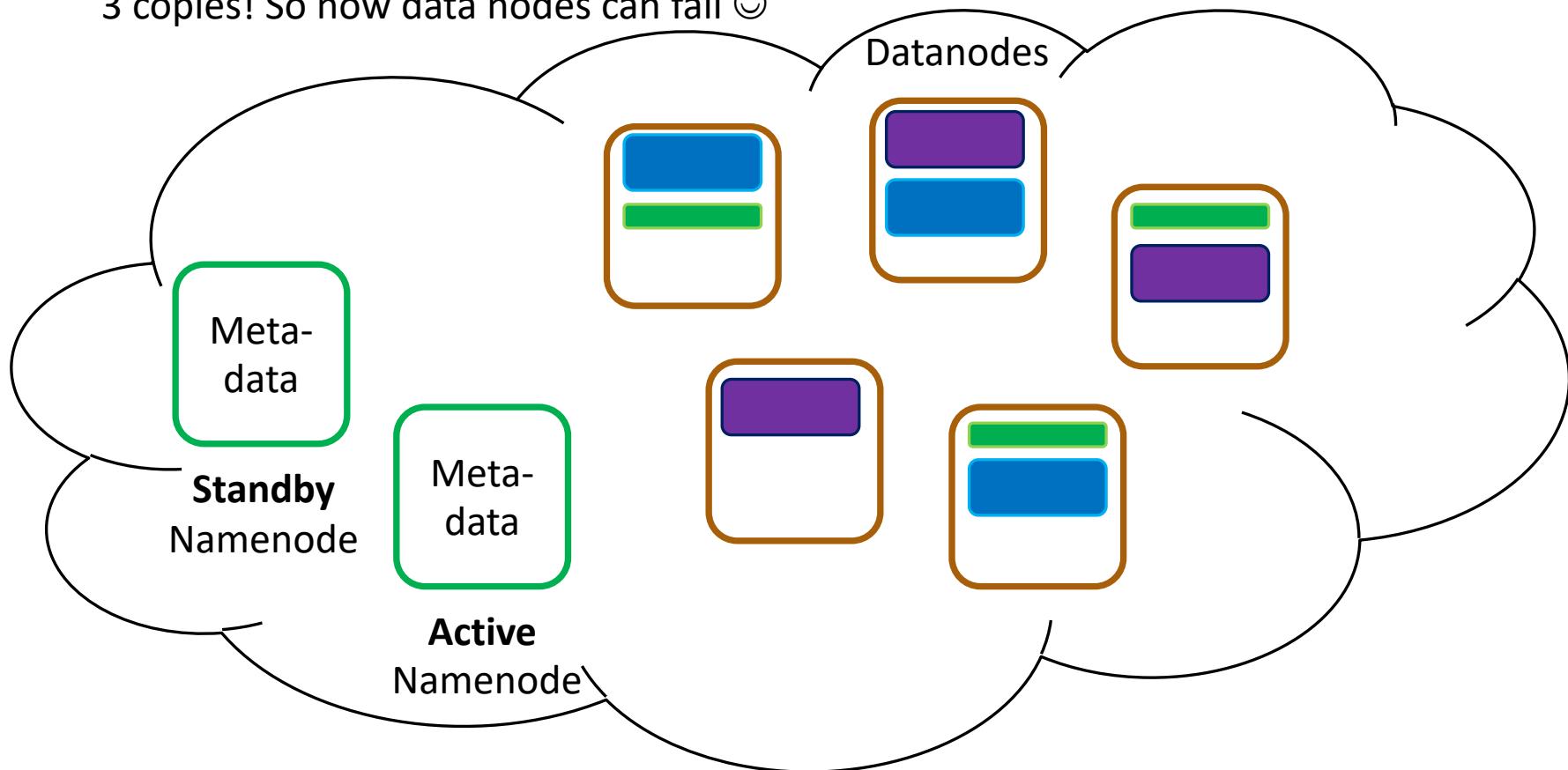
3 copies! So now data nodes can fail 😊





# HDFS: standby Name Node

3 copies! So now data nodes can fail ☺



# Hadoop filesystem (fs) commands

---

List files:

```
hadoop fs -ls
```

Store a local file in hdfs:

```
hadoop fs -put myfile.txt
```

Rename file:

```
hadoop fs -mv myfile.txt newname.txt
```

Remove file:

```
hadoop fs -rm filename.txt
```

Create directory:

```
hadoop fs -mkdir newdir
```

=> very similar to traditional unix commands

# MapReduce (MR): why?

---

- Processing documents top-bottom: slow
- MR: parallel processing
- Retailer example: calculate total sales per store:
  - Traditional: start from top and increment sales for each store as you go along
  - Store: hash table <key, value>
  - Problems running on 1 T of data?
    - It won't work?
    - Out of memory?
    - Long time?
    - Wrong answer?

2012-01-01	London Clothes	25.99
2012-01-01	Miami Music	12.15
2012-01-02	NYC Toys	3.10
2012-01-02	Miami Clothes	50.00

# MapReduce (MR): why?

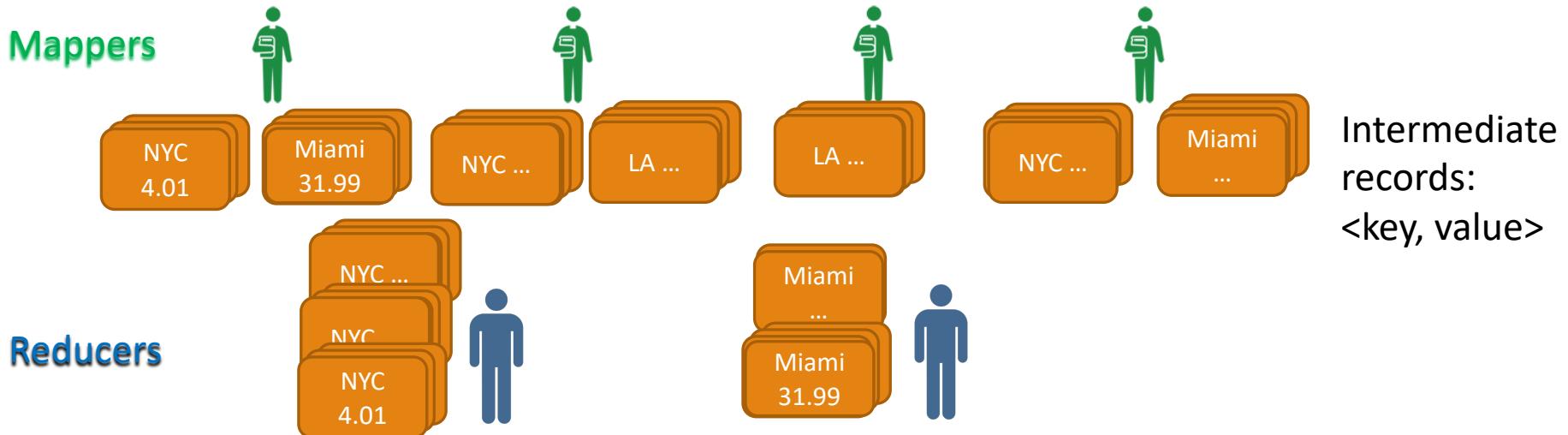
---

- Processing documents top-bottom: slow
- MR: parallel processing
- Retailer example: calculate total sales per store:
  - Traditional: start from top and increment sales for each store as you go along
  - Store: hash table <key, value>
  - Problems running on 1 T of data?
    - It wont' work? X
    - Out of memory? ✓
    - Long time? ✓
    - Wrong answer? X

2012-01-01 London Clothes 25.99  
2012-01-01 Miami Music 12.15  
2012-01-02 NYC Toys 3.10  
2012-01-02 Miami Clothes 50.00

# MapReduce

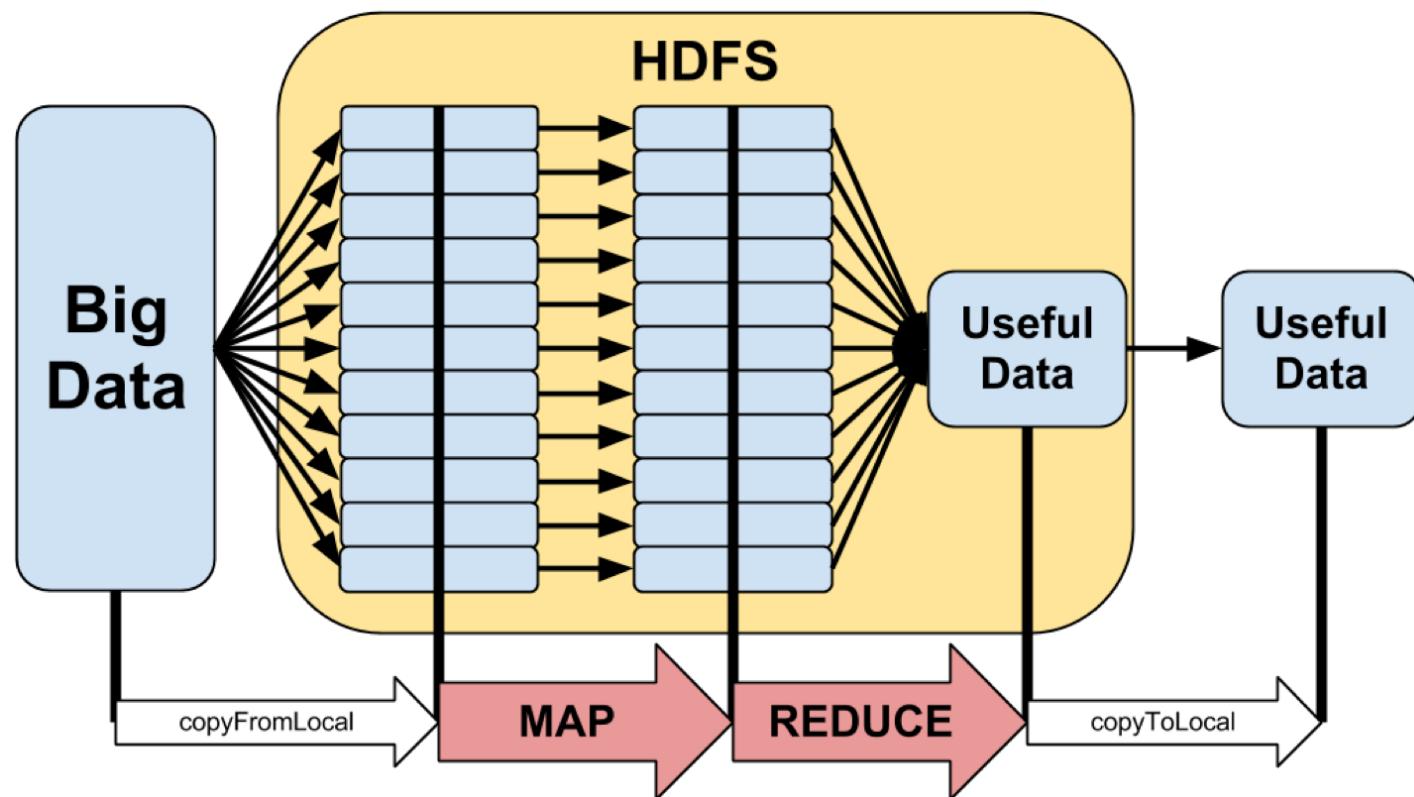
- What if you had more people to help? Mappers and Reducers
- Break ledger into chunks and distribute to mappers



- Reducers get assigned a store
- They ask for the stack of that store at each Mapper: shuffle
- They alphabetically go through their stacks: sort; and process them

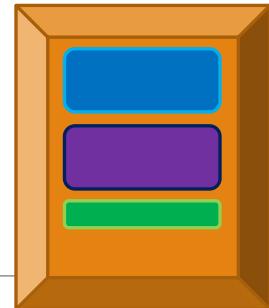
# MapReduce

---

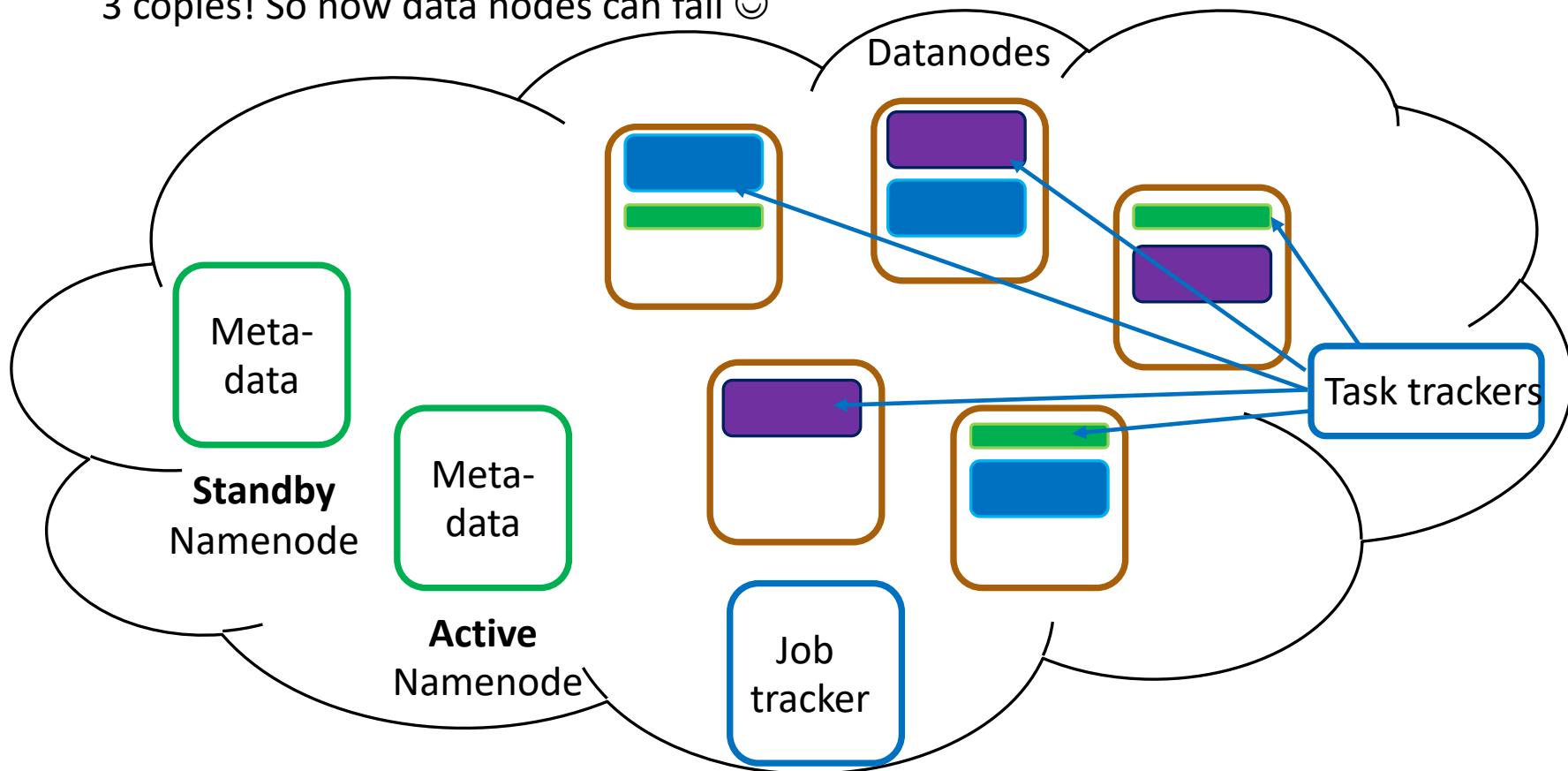


# Daemons of MR

150MB file



3 copies! So now data nodes can fail ☺



# Daemons of MapReduce

---

- MapReduce job submitted to Job tracker
- **Job tracker** splits work to mappers and reducers
- **Task tracker:** runs on each data node, executes the actual mapping and reducing
  - > on the same machine as data node, so saves network traffic!
  - Hadoop will try to make sure a task tracker works on data from the same machine (if not already busy)
- **Mappers** perform filtering and sorting and will pass the intermediate data to reducers
- **Reducers** process this (summary operation) and write final output to HDFS

# Hadoop & MapReduce

---

- In the lab, we will be using a Hadoop emulator. This will make our lives much easier as no installation is required.
  
- This week's lab link:
  
- <https://colab.research.google.com/drive/1gnzq7rElh363MWXGnvQhEQfooW1Bd6U?usp=sharing>

# Did you pay attention?

---

- [Pop quiz:](#)
- <https://forms.gle/FpGKgKRjwBB4wrdK7>