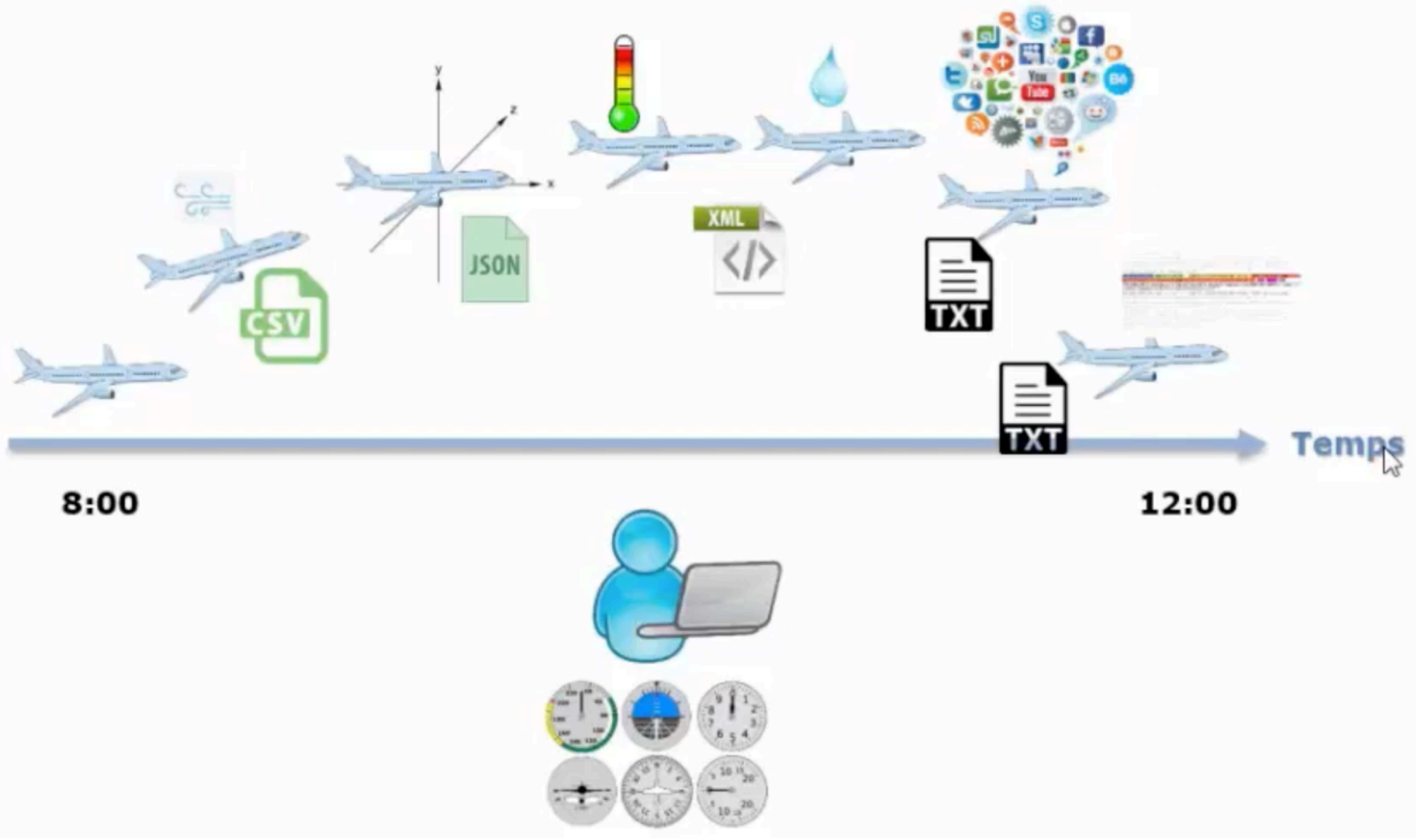

CSV
JSON
XML
TXT
TXT
Temps
8:00
12:00

# LA DÉMESURE DES OCTETS

**UNE SUITE DE HUIT CHIFFRES (0 OU 1)**

**1o**

**Octet**
o

**UNE PAGE DE TEXTE**

**30 ko**

**Kilooctet**
ko
1 000 octets

**UN MORCEAU DE MUSIQUE**

**5 Mo**

**Mégaoctet**
Mo
1 000 ko

**UN FILM DE DEUX HEURES**

**1 Go**

**Gigaoctet**
Go
1 000 Mo

**6 MILLIONS DE LIVRES**
(PRESQUE LA MOITIÉ DU CATALOGUE DE LA BIBLIOTHÈQUE NATIONALE DE FRANCE)

**1 To**

**Téraoctet**
To
1 000 Go

**2 MILLIARDS DE PHOTOS NUMÉRIQUES DE RÉSOLUTION MOYENNE**

**1 Po**

**Pétaoctet**
Po
1 000 To

**TOUTES LES INFORMATIONS PRODUITES JUSQU'À 2003**

**5 Eo**

2003

**Exaoctet**
Eo
1 000 Po

**LA TOTALITÉ DES DONNÉES ENREGISTRÉES EN 2011**

**1,8 Zo**

2011

**Zettaoctet**
Zo
1 000 Eo

**EN 2013, LE PLUS GRAND DATA CENTER AU MONDE SERA CAPABLE DE TRAITER SIMULTANÉMENT 1 Yo DE DONNÉES**

**1 Yo**

**Yottaoctet**
Yo
1 000 Zo

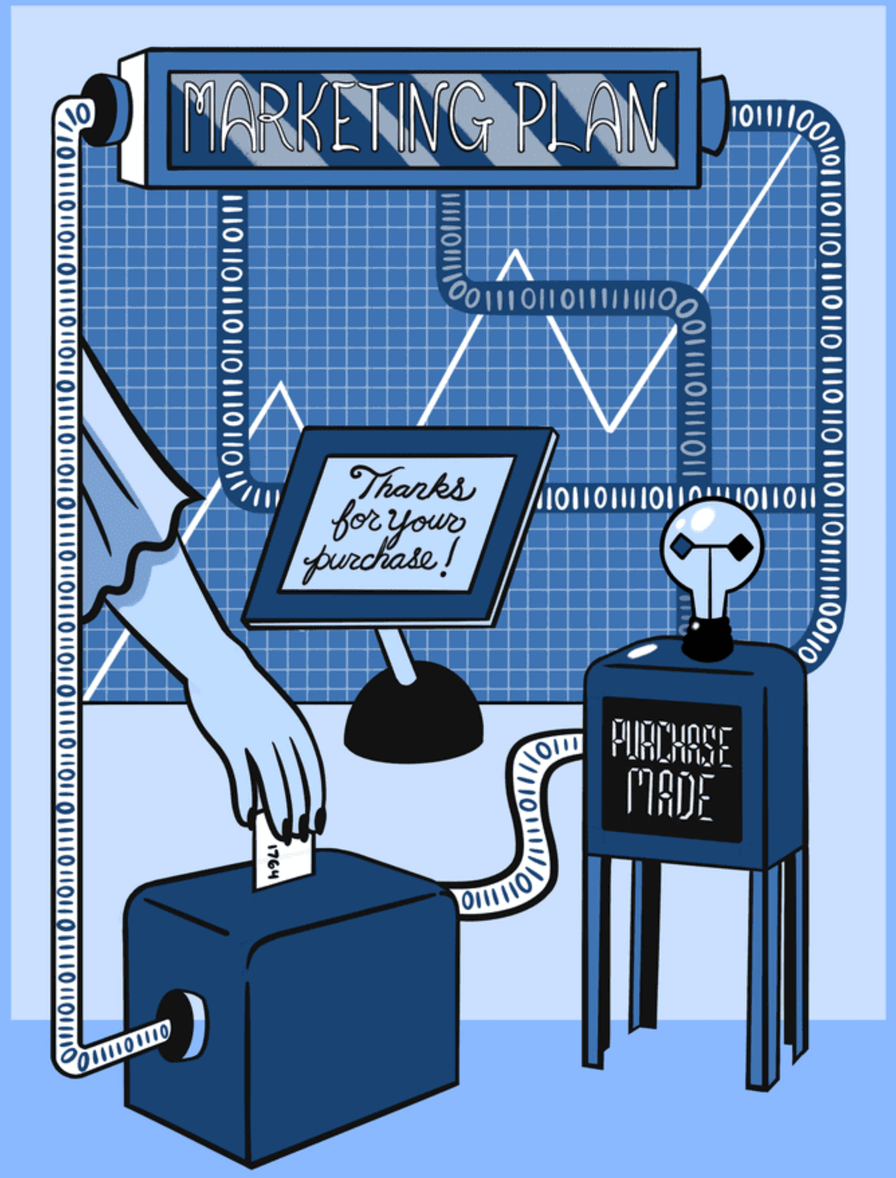
“L’humanité produit autant d’informations en deux jours qu’elle ne l’a fait en deux millions d’années.”

(…)

Les chiffres donnent le tournis : chaque minute, environ 350 000 tweets, 15 millions de SMS et 200 millions de mails sont envoyés au niveau mondial ; pendant le même laps de temps, des dizaines d'heures de vidéos sont mises en ligne sur YouTube, des centaines de milliers de nouveaux fichiers sont archivés sur les serveurs de Facebook”

Et ça, c’était en 2012…


SAY BIG DATA
ONE MORE TIME

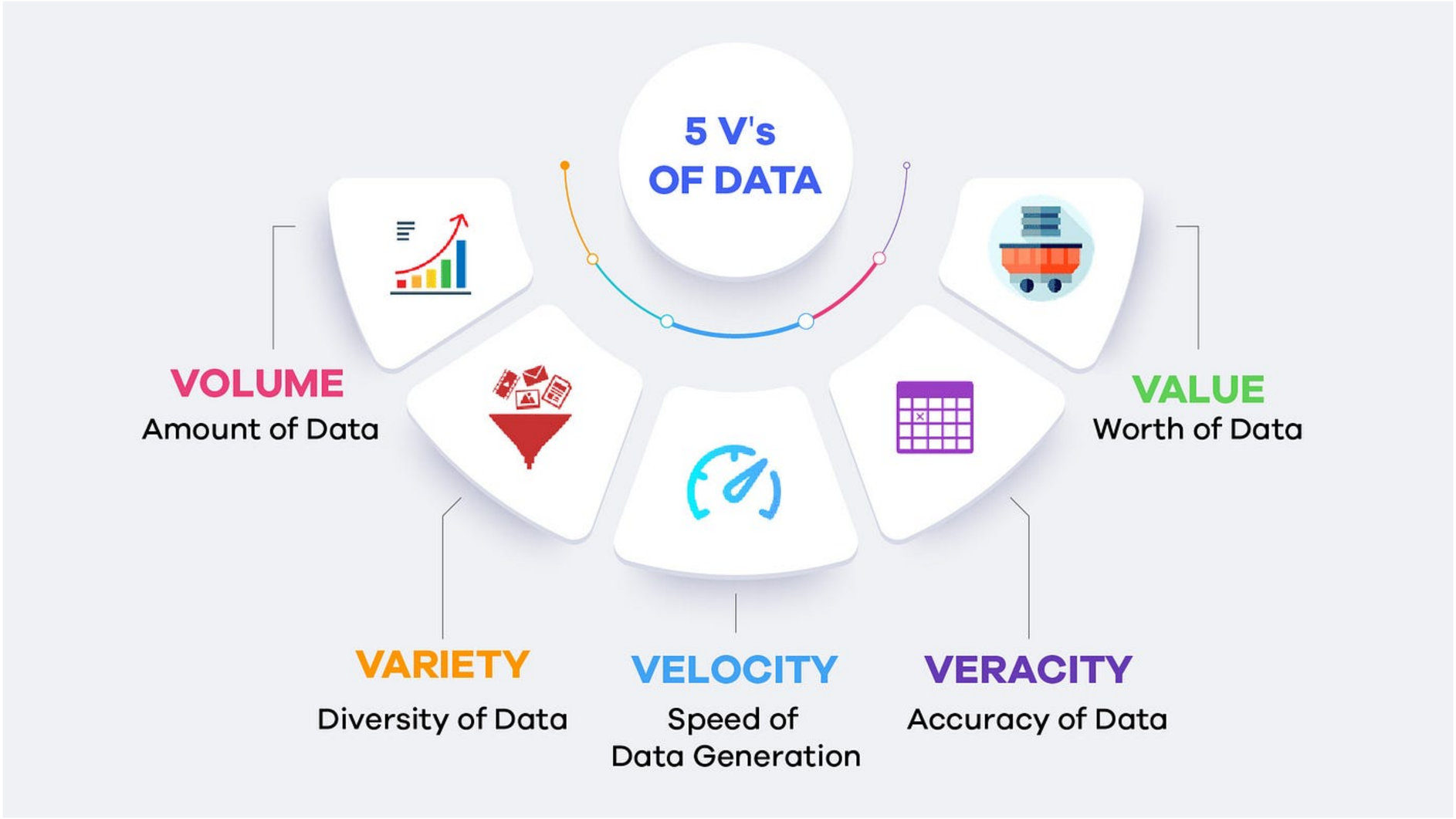



# Big Data

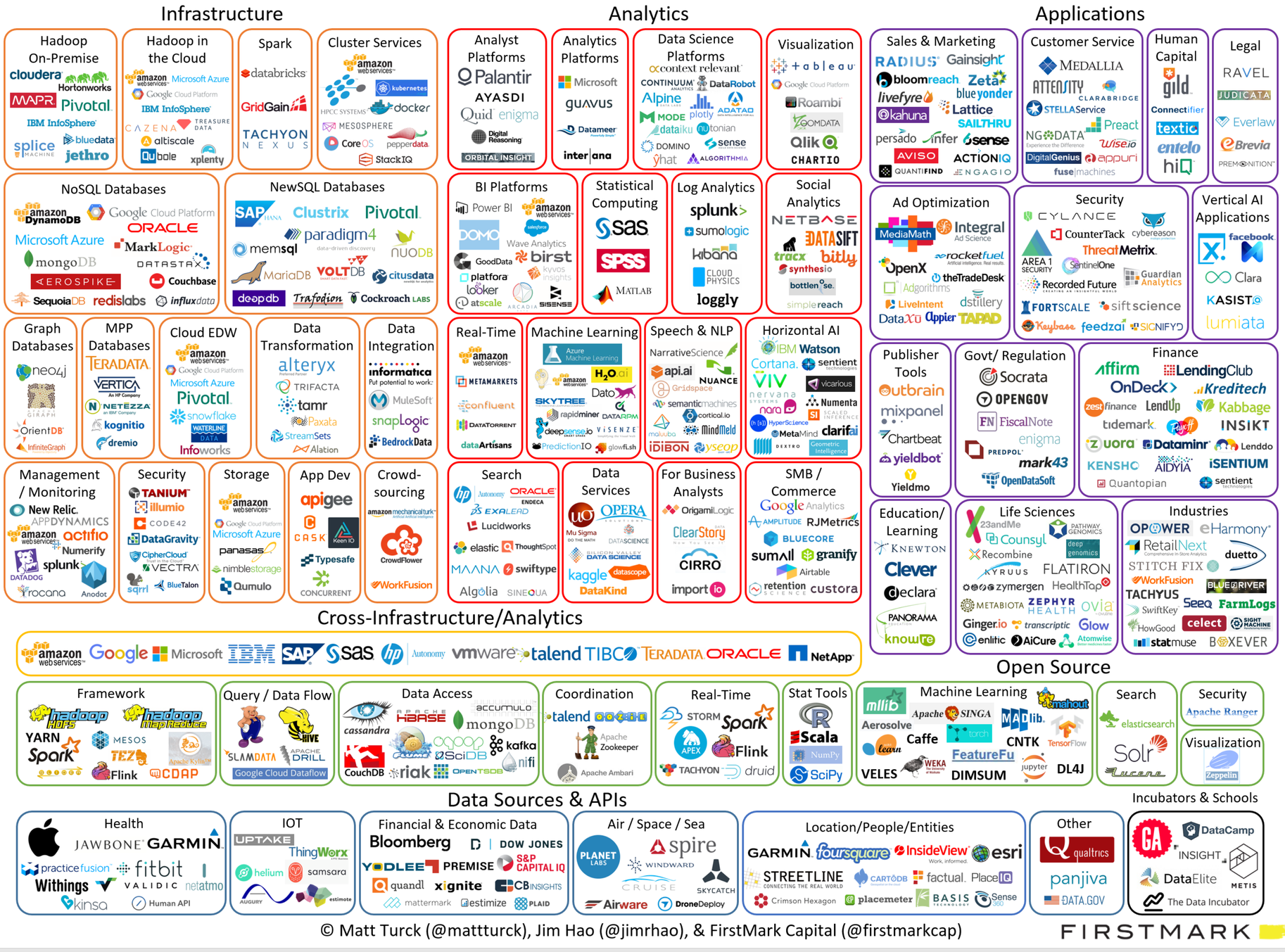
*['big 'dā-tə]*

Large, diverse sets of information that grow at ever-increasing rates.

Investopedia


5 V's
OF DATA
VOLUME
Amount of Data
VARIETY
Diversity of Data
VELOCITY
Speed of
Data Generation
VERACITY
Accuracy of Data
VALUE
Worth of Data

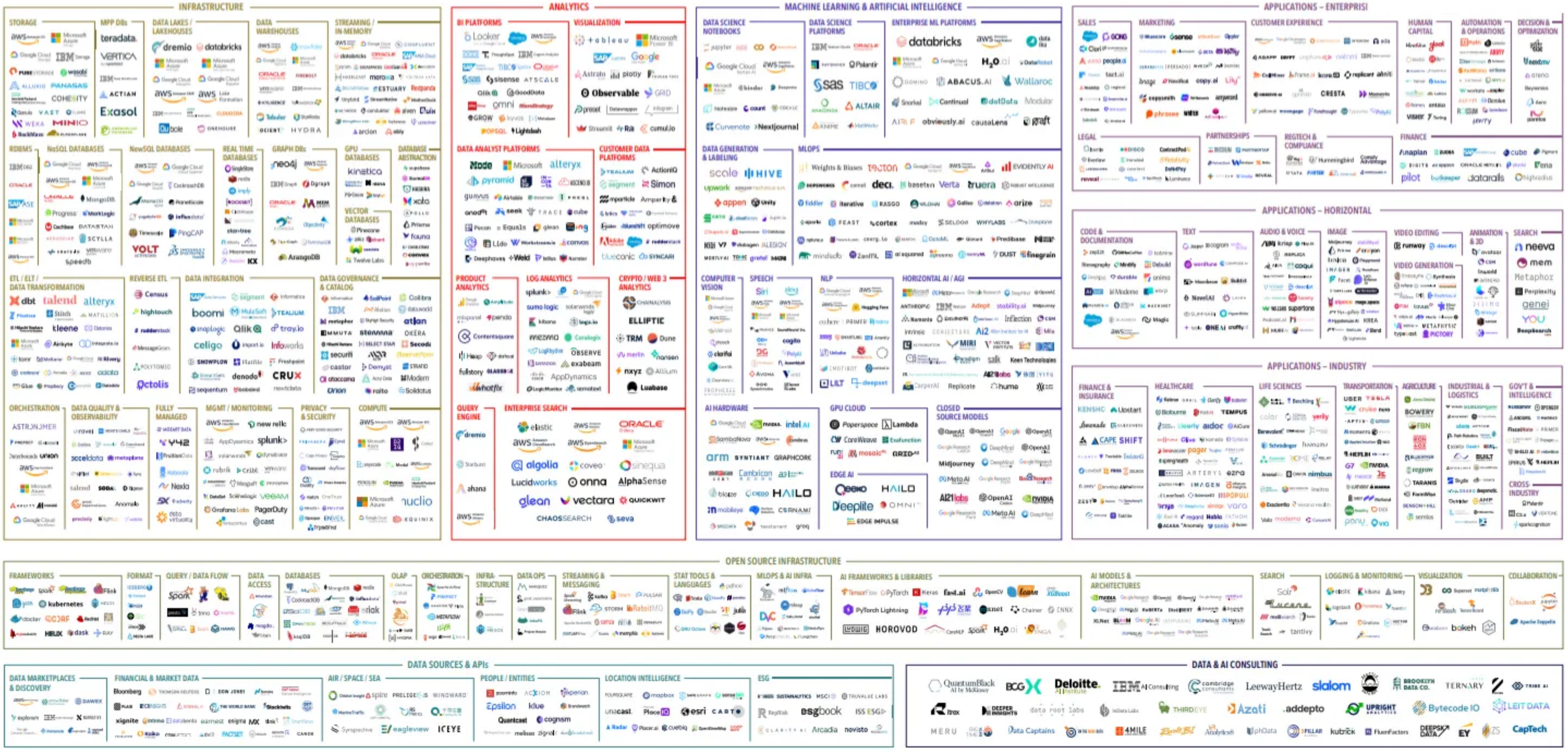
# Big Data Landscape 2016






FIRSTMARK

# THE 2023 MAD (MACHINE LEARNING, ARTIFICIAL INTELLIGENCE & DATA) LANDSCAPE

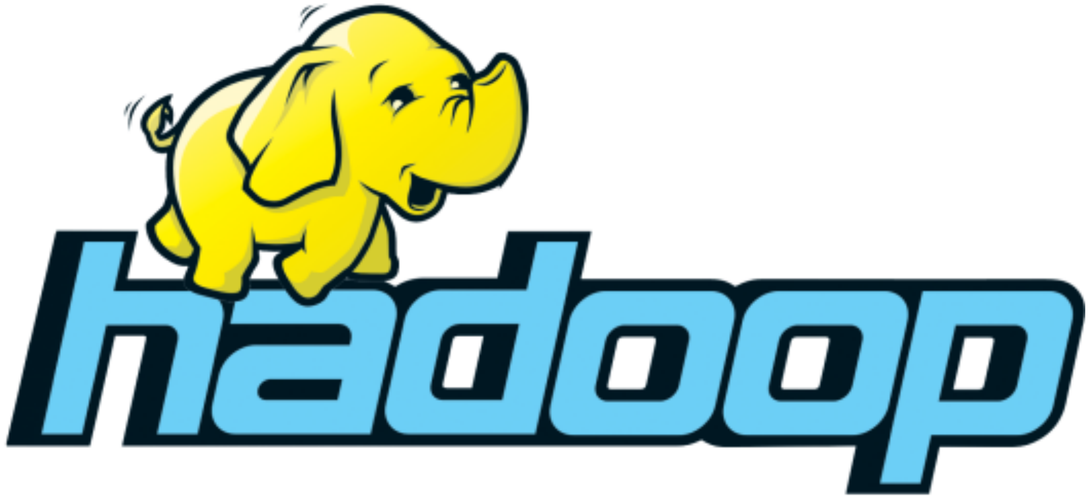


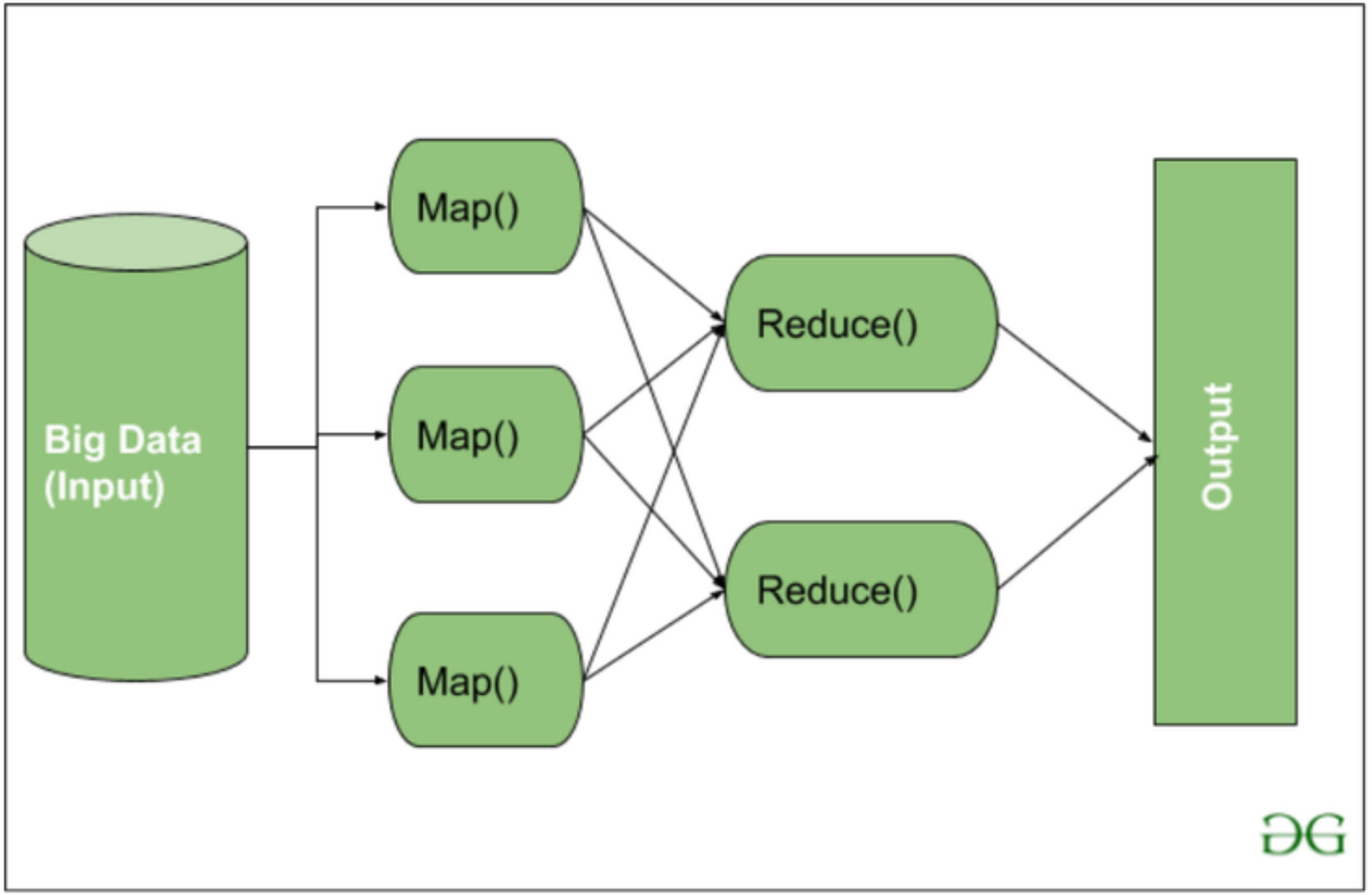

Version 1.0 - Feb 2023

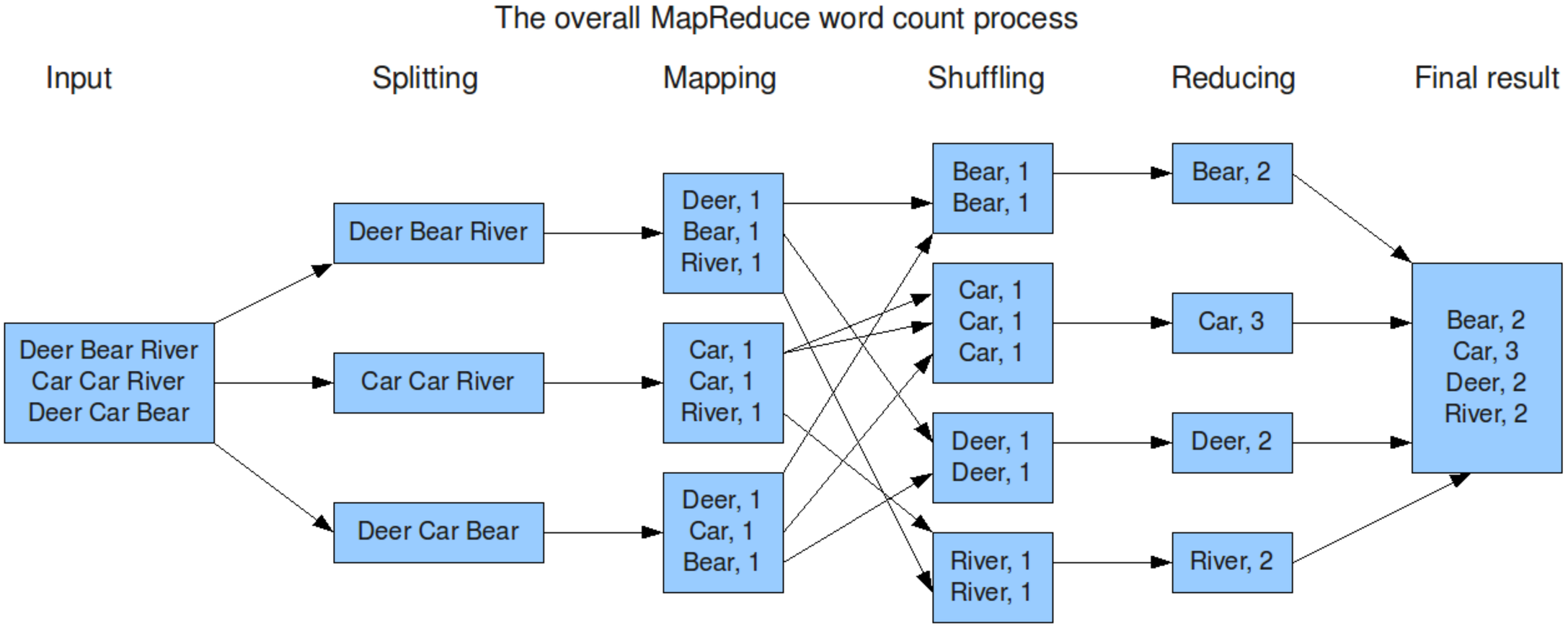
Blog post: mattturck.com/MAD2023
Interactive version: MAD.firstmarkcap.com
Comments? Email MAD2023@firstmarkcap.com
FIRSTMARK
EARLY STAGE VENTURE CAPITAL

10,000,000,000,000,000,
000,000,000,000,000,000,
000,000,000,000,000,000,
000,000,000,000,000,000,
000,000,000,000,000,000,
000,000,000,000 = 1 googol

=>


hadoop


Big Data
(Input)
Map()
Map()
Map()
Reduce()
Reduce()
Output


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2



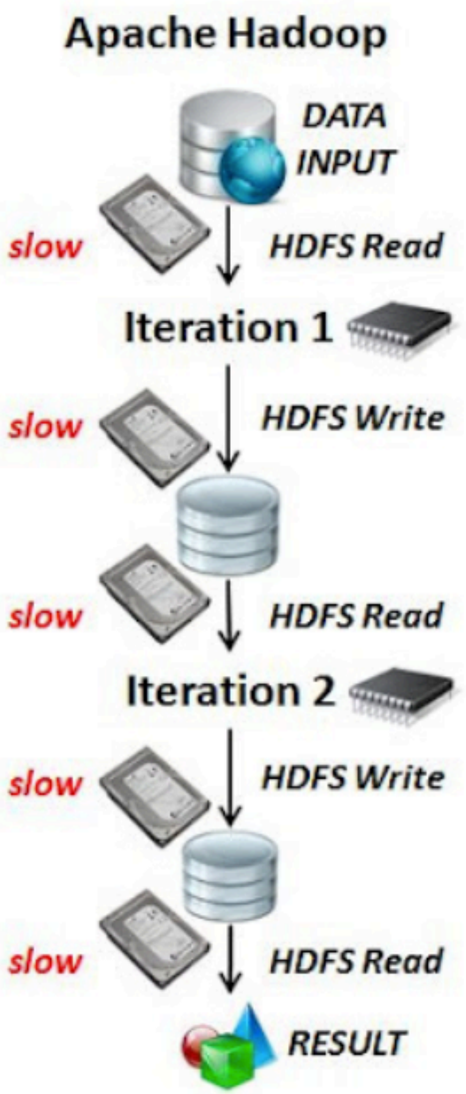
CLÉMENT LESAFFRE • 08h25, le 22 mars 2021

**Le 10 mars, un important incendie ravageait les datacenters strasbourgeois d'OVH, leader français de l'hébergement de sites Internet. Depuis, une centaine de techniciens travaillent jour et nuit pour sauver ce qui peut l'être et redémarrer les serveurs. Mais des entreprises ont perdu des données capitales et l'image d'OVH risque d'en pâtir.**

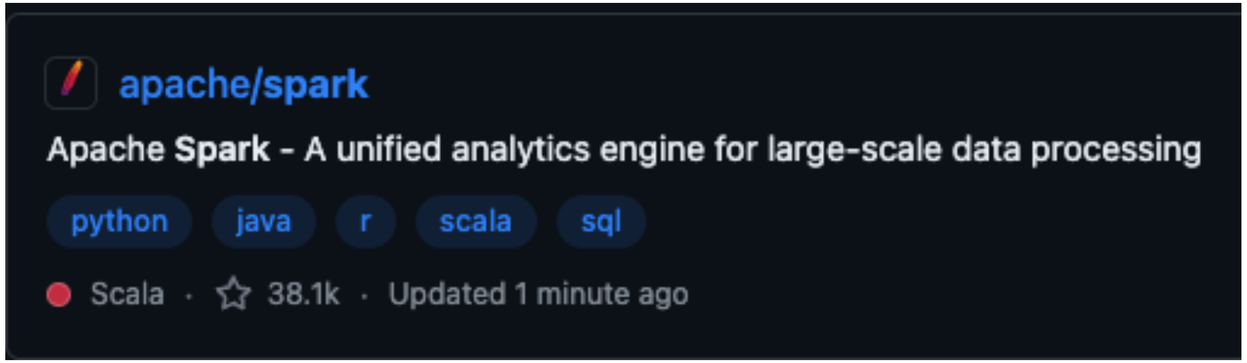
Stack Overflow
https://stackoverflow.com › why-is...

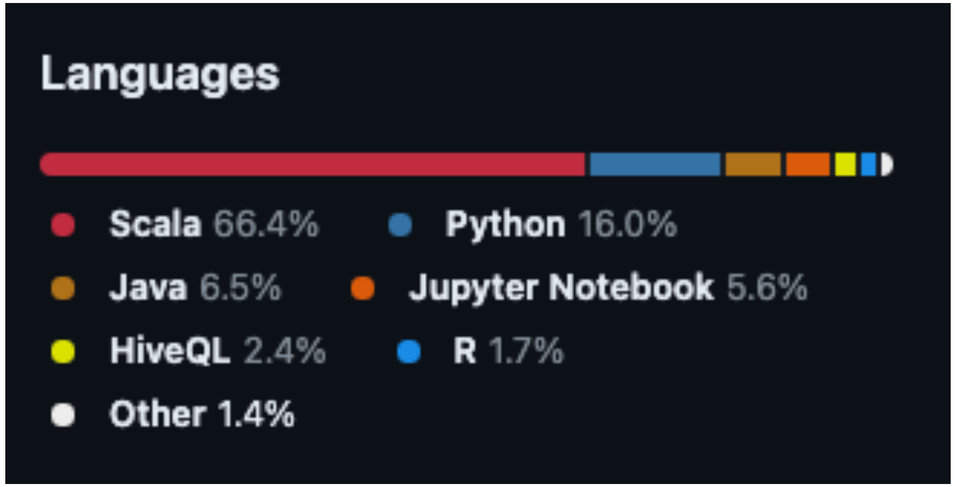
## Why is hadoop slow for a simple hello world job

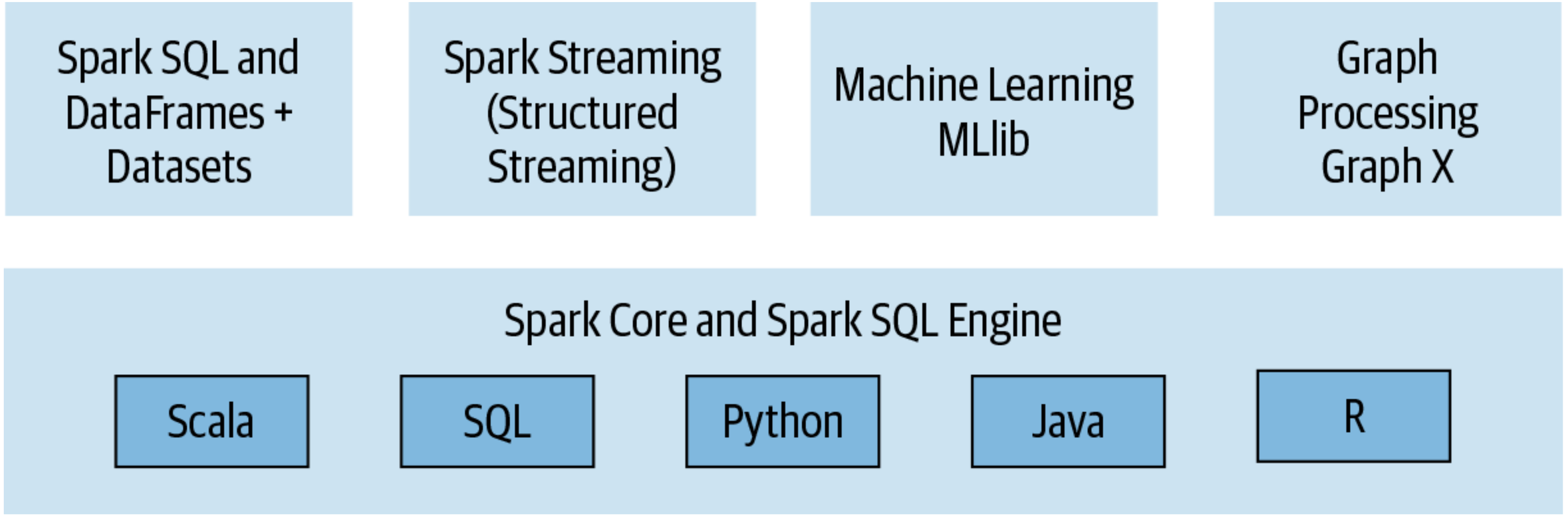
Mar 1, 2019 — I am following the tutorial on the hadoop website: https://hadoop.apache.org/docs/r3.1.2/hadoop-project-dist/hadoop-common/SingleCluster.html. I ...

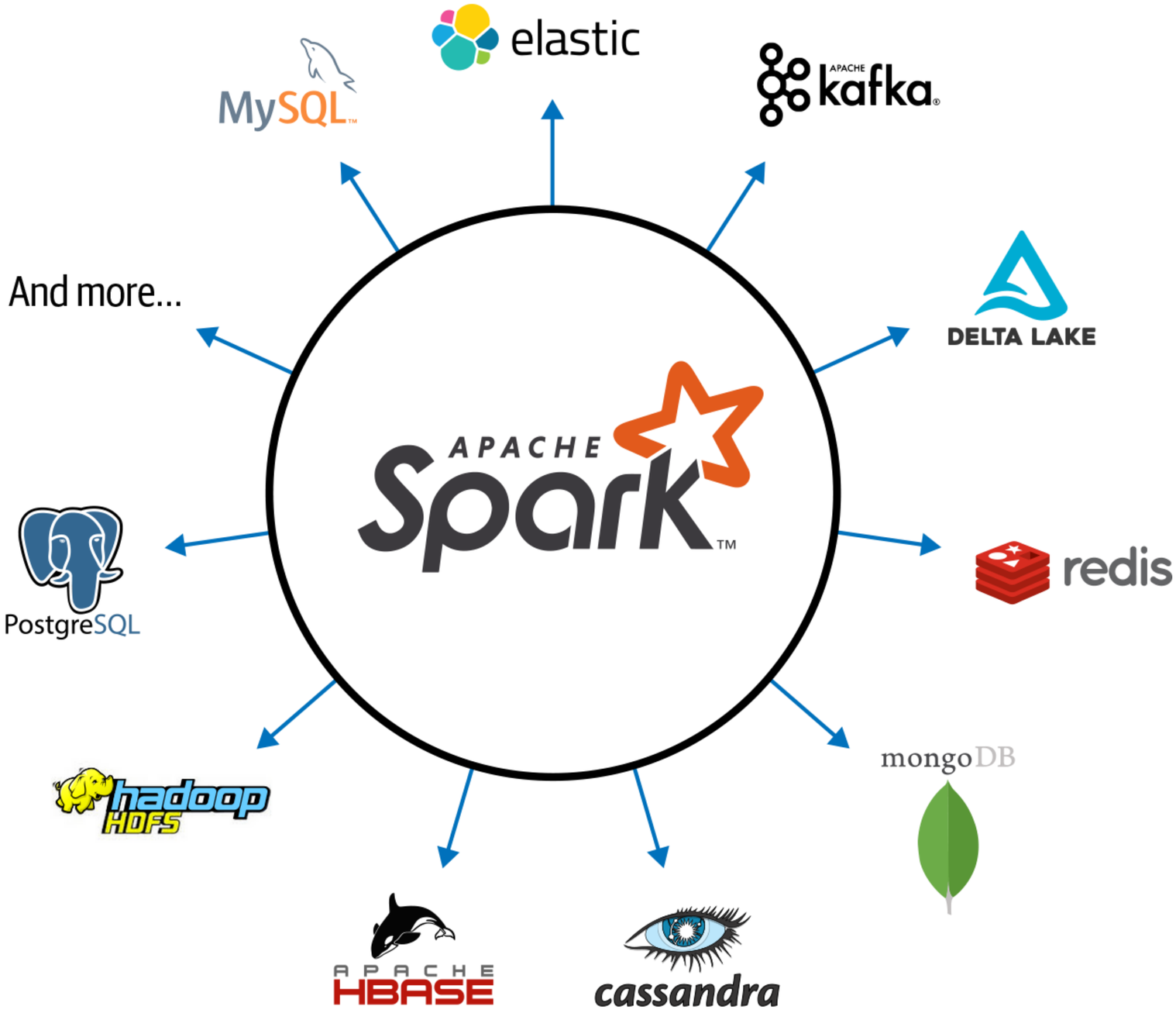

Apache Hadoop
DATA
INPUT
slow
HDFS Read
Iteration 1
slow
HDFS Write
slow
HDFS Read
Iteration 2
slow
HDFS Write
slow
HDFS Read
RESULT


Spark

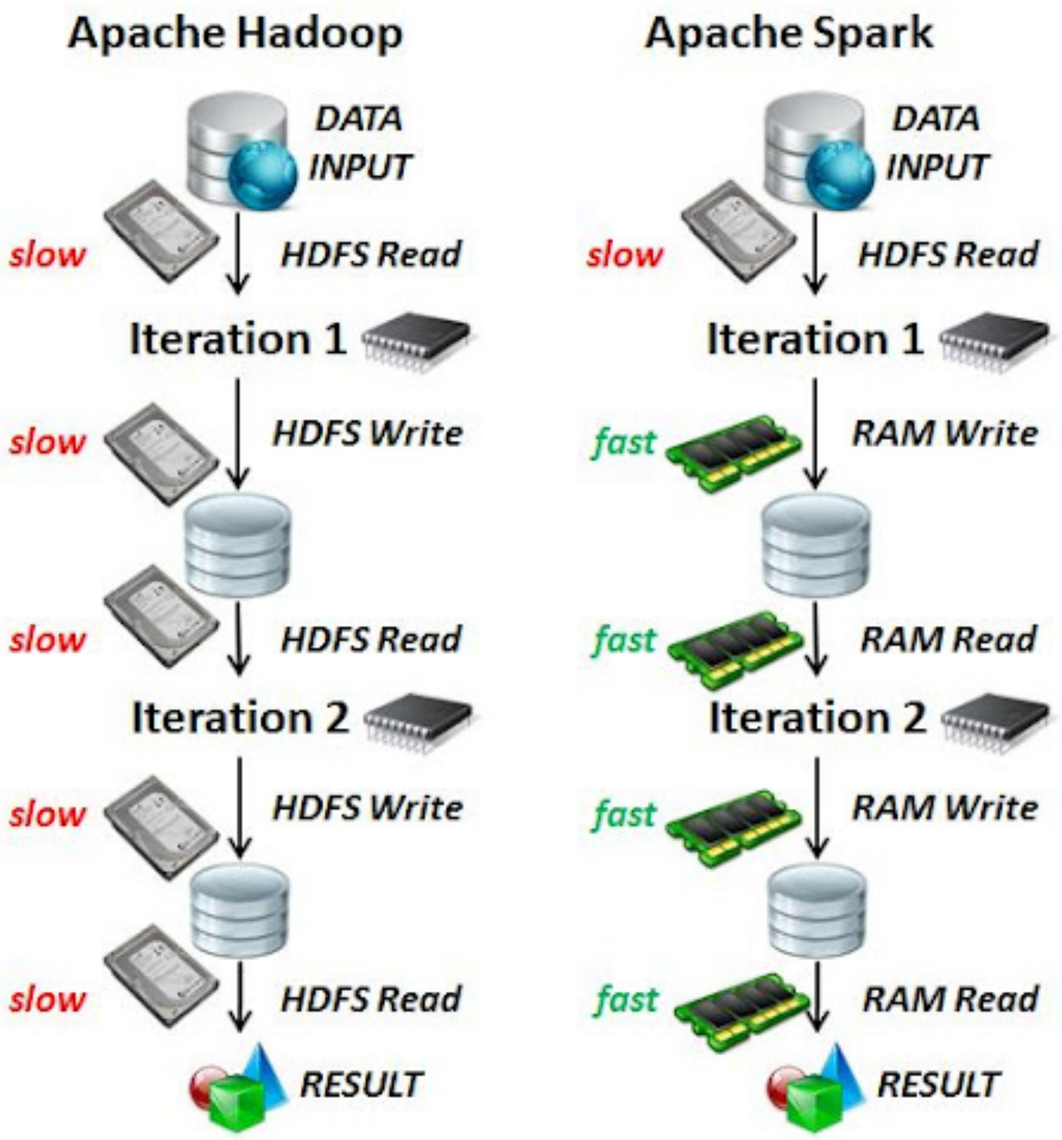

apache/spark
Apache Spark - A unified analytics engine for large-scale data processing
python
java
r
scala
sql
Scala · 38.1k · Updated 1 minute ago

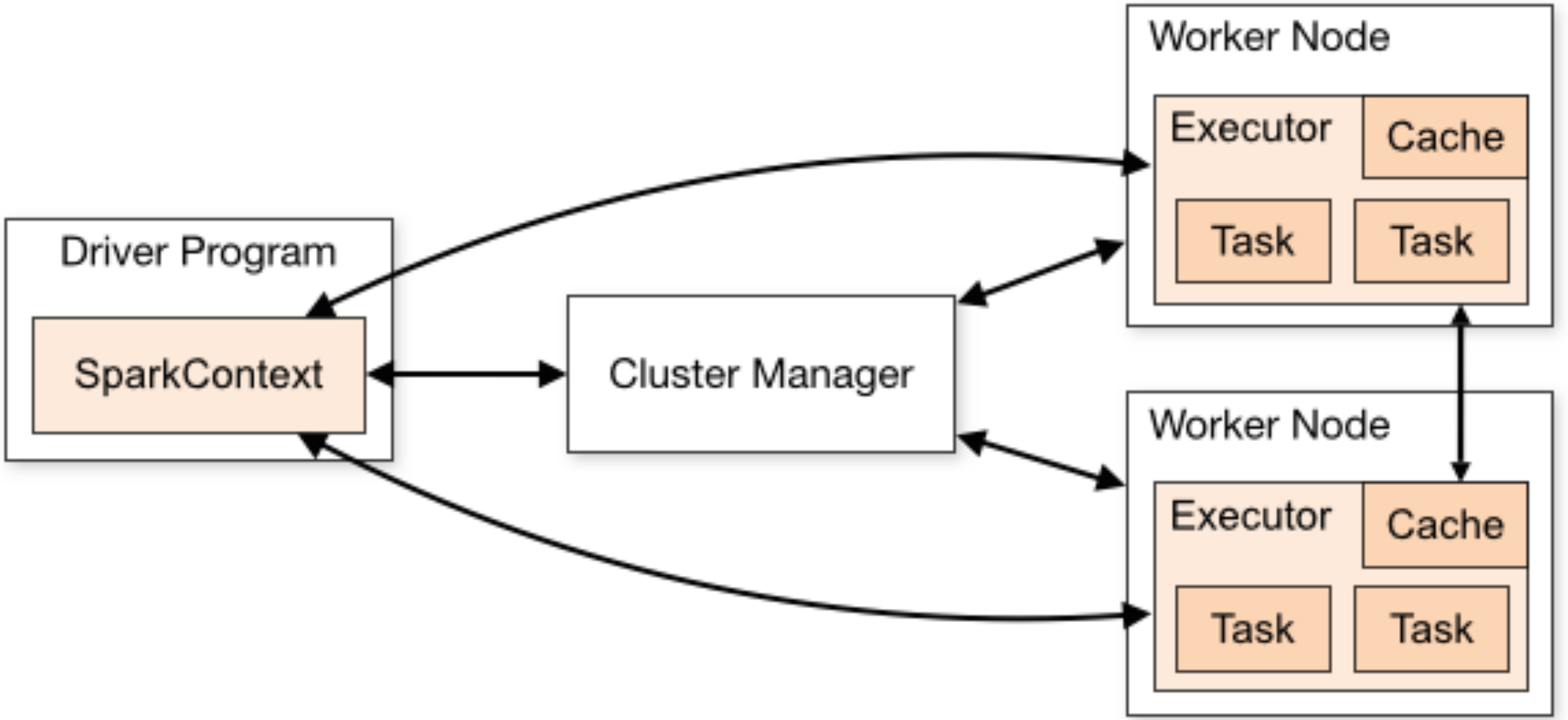

Languages
Scala 66.4%
Python 16.0%
Java 6.5%
Jupyter Notebook 5.6%
HiveQL 2.4%
R 1.7%
Other 1.4%


Spark SQL and DataFrames + Datasets
Spark Streaming (Structured Streaming)
Machine Learning MLlib
Graph Processing Graph X
Spark Core and Spark SQL Engine
Scala
SQL
Python
Java
R


elastic
MySQL
APACHE kafka
And more...
DELTA LAKE
APACHE Spark™
PostgreSQL
redis
mongoDB
hadoop HDFS
APACHE HBASE
cassandra


Ew, I stepped in shit.
imgflip.com


Apache Hadoop
DATA INPUT
slow
HDFS Read
Iteration 1
slow
HDFS Write
slow
HDFS Read
Iteration 2
slow
HDFS Write
slow
HDFS Read
RESULT
Apache Spark
DATA INPUT
slow
HDFS Read
Iteration 1
fast
RAM Write
fast
RAM Read
Iteration 2
fast
RAM Write
fast
RAM Read
RESULT


Worker Node
Executor
Cache
Task
Task
Driver Program
SparkContext
Cluster Manager
Worker Node
Executor
Cache
Task
Task

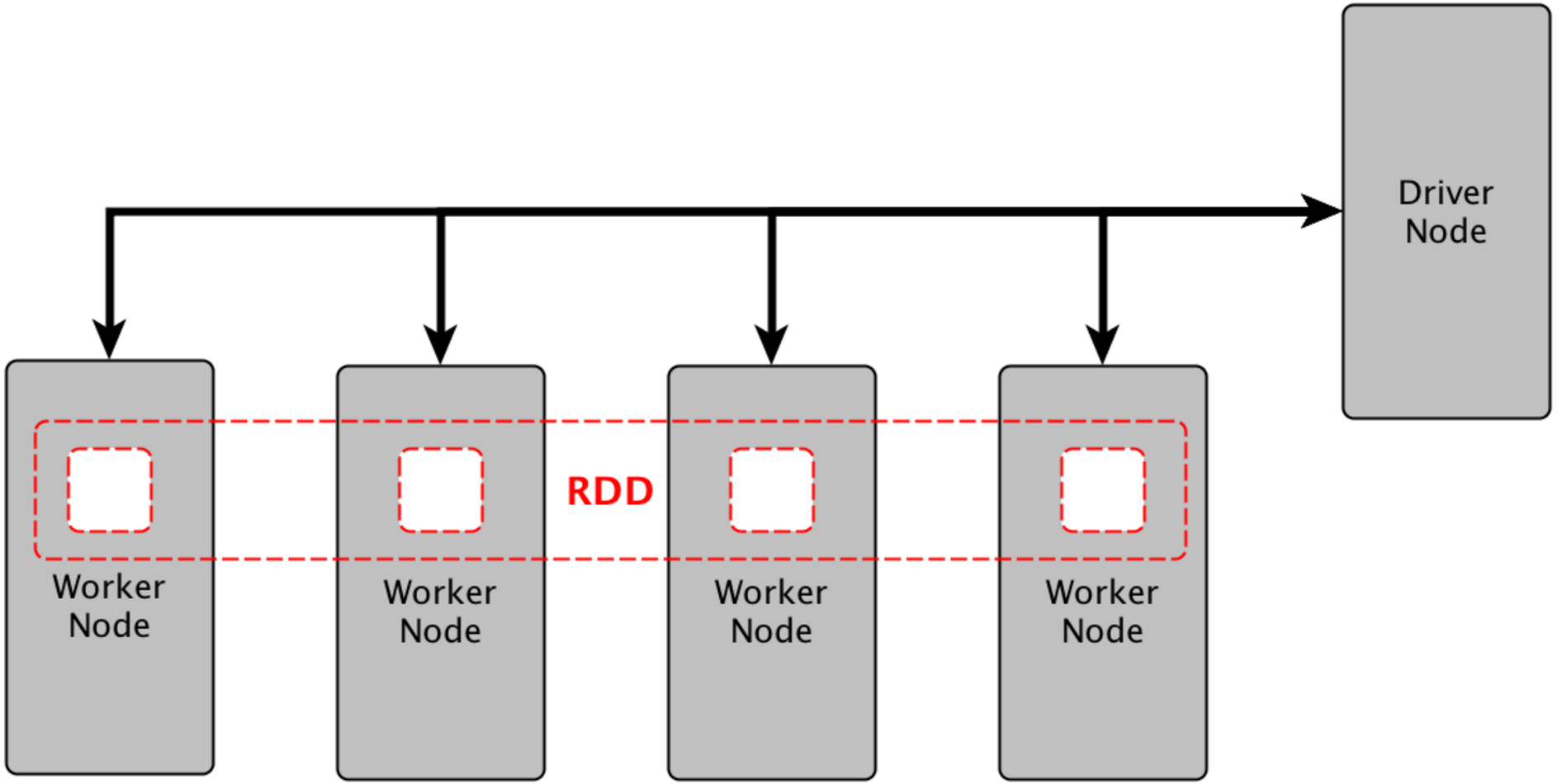
| Mode | Spark driver | Spark executor | Cluster manager |
| --- | --- | --- | --- |
| Local | Runs on a single JVM, like a laptop or single node | Runs on the same JVM as the driver | Runs on the same host |
| Standalone | Can run on any node in the cluster | Each node in the cluster will launch its own executor JVM | Can be allocated arbitrarily to any host in the cluster |
| YARN (client) | Runs on a client, not part of the cluster | YARN's NodeManager's container | YARN's Resource Manager works with YARN's Application Master to allocate the containers on NodeManagers for executors |
| YARN (cluster) | Runs with the YARN Application Master | Same as YARN client mode | Same as YARN client mode |
| Kubernetes | Runs in a Kubernetes pod | Each worker runs within its own pod | Kubernetes Master |

# RDD

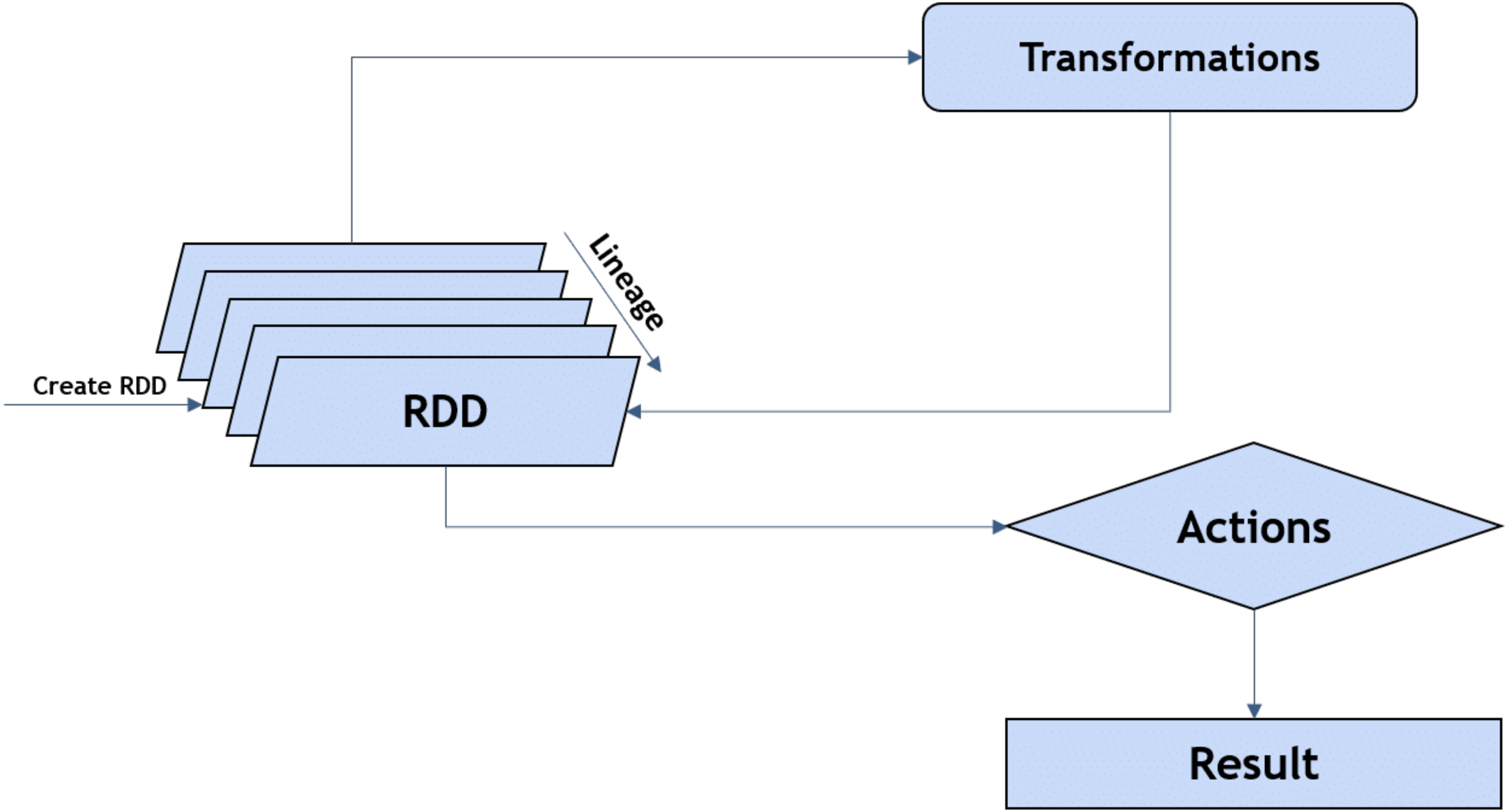
Resilient Distributed Dataset

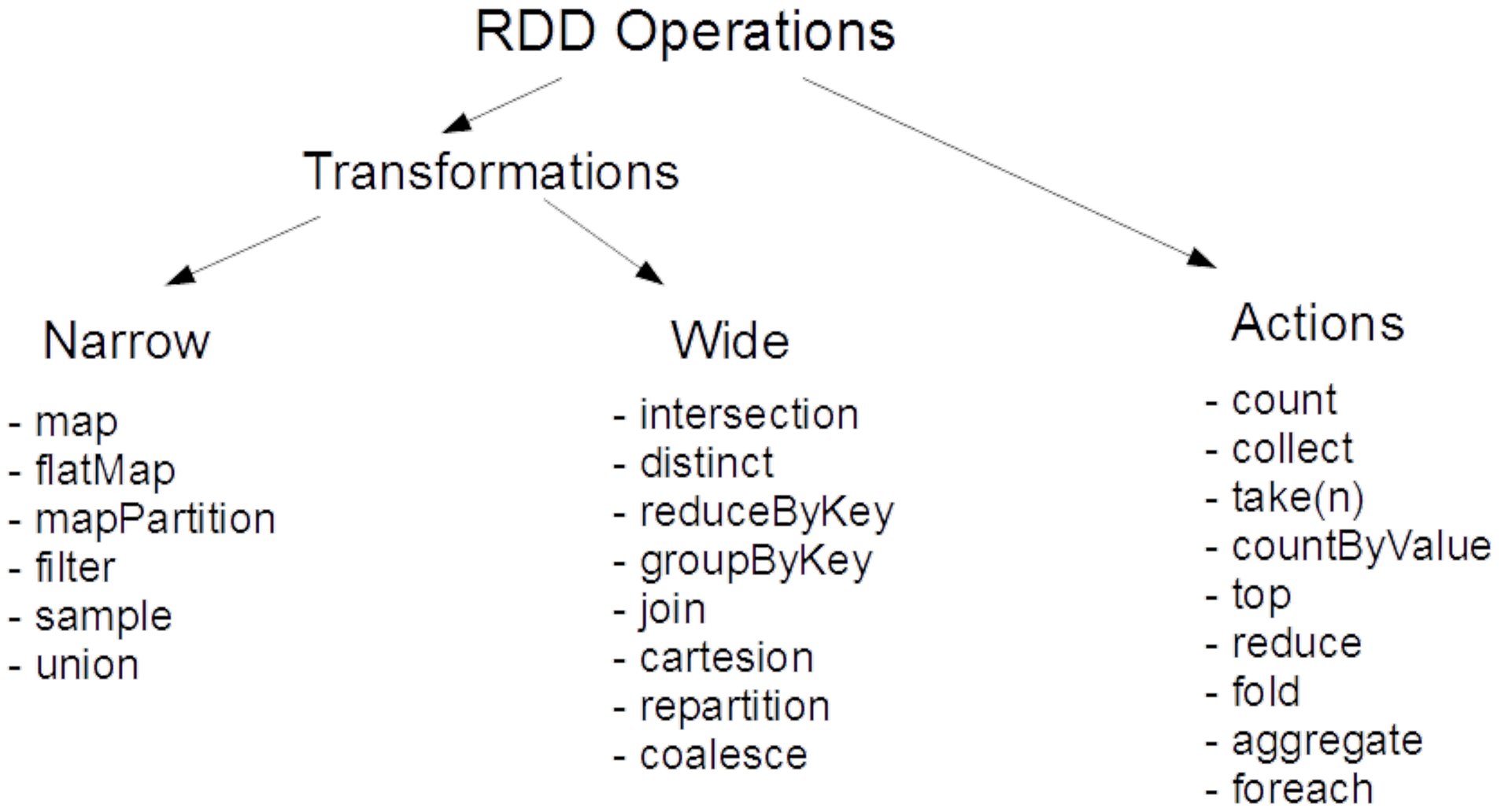
# Lazy Evaluation

Lazy evaluation (or call-by-name) is an evaluation strategy which delays the evaluation of an expression until its value is needed


Driver
Node
RDD
Worker
Node
Worker
Node
Worker
Node
Worker
Node

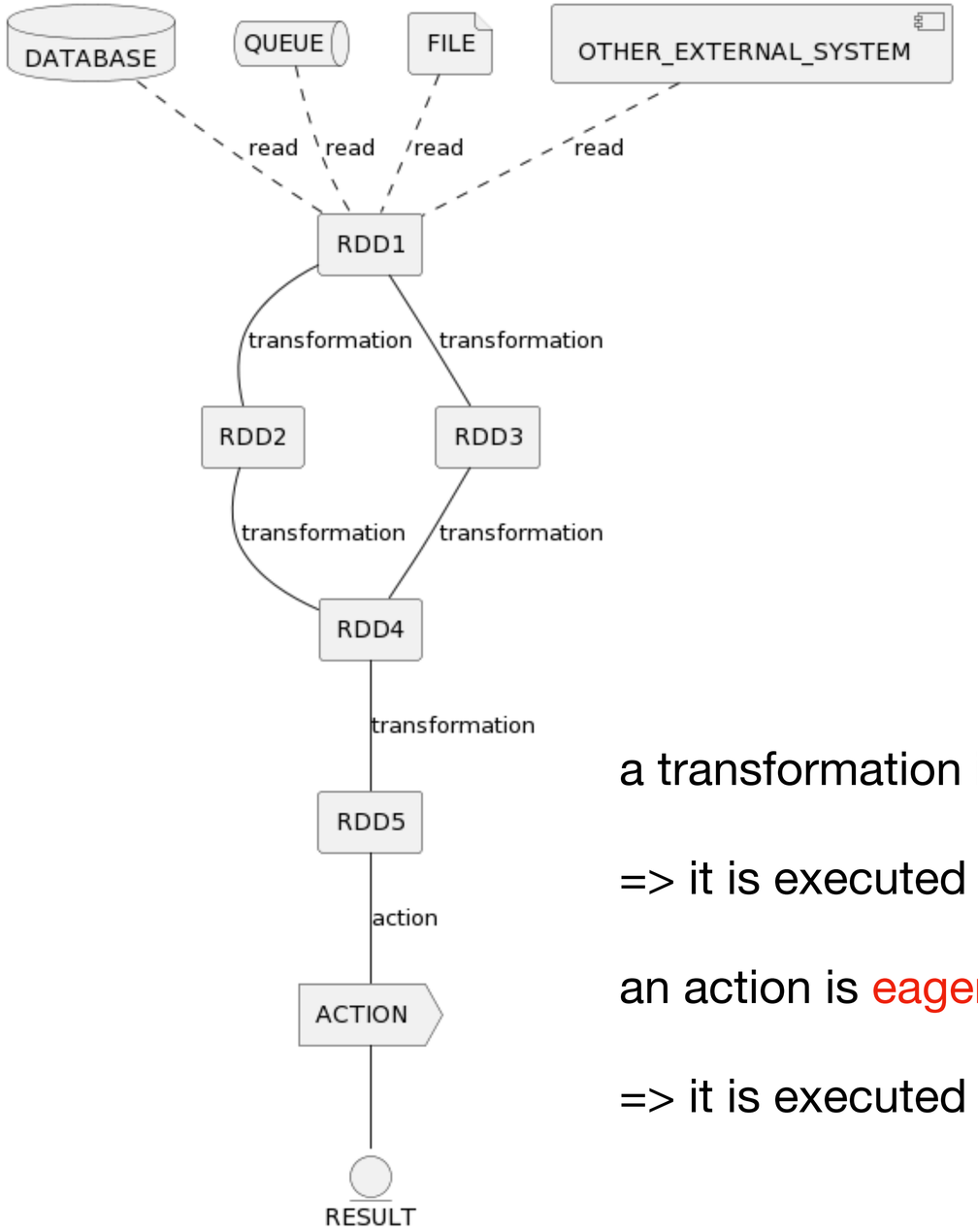
Spark Operations = TRANSFORMATIONS +  ACTIONS


Transformations
Lineage
Create RDD
RDD
Actions
Result


RDD Operations
Transformations
Narrow
- map
- flatMap
- mapPartition
- filter
- sample
- union
Wide
- intersection
- distinct
- reduceByKey
- groupByKey
- join
- cartesion
- repartition
- coalesce
Actions
- count
- collect
- take(n)
- countByValue
- top
- reduce
- fold
- aggregate
- foreach

Spark execution plan is a

Direct Acyclic Graph (DAG)

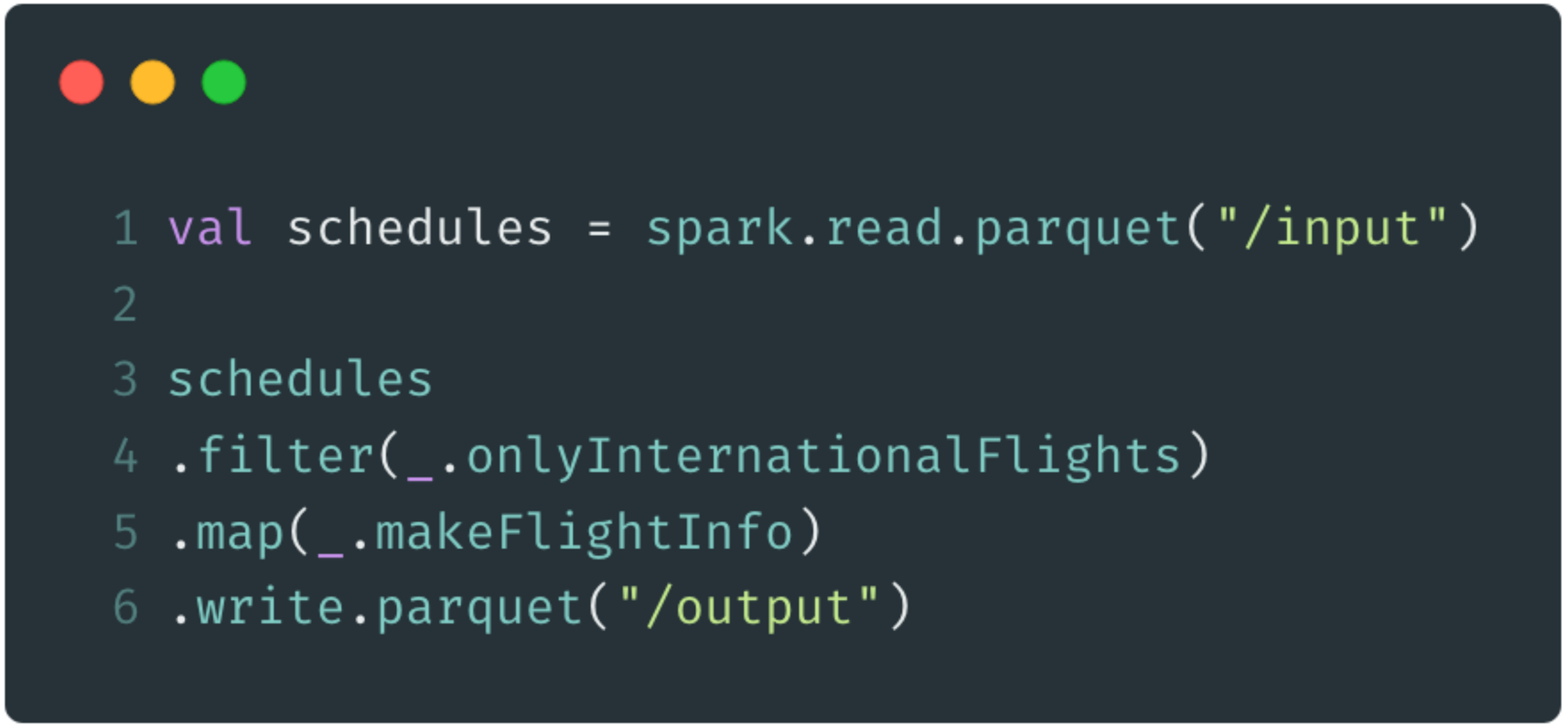



a transformation is lazily evaluated

=> it is executed when an action using it is called

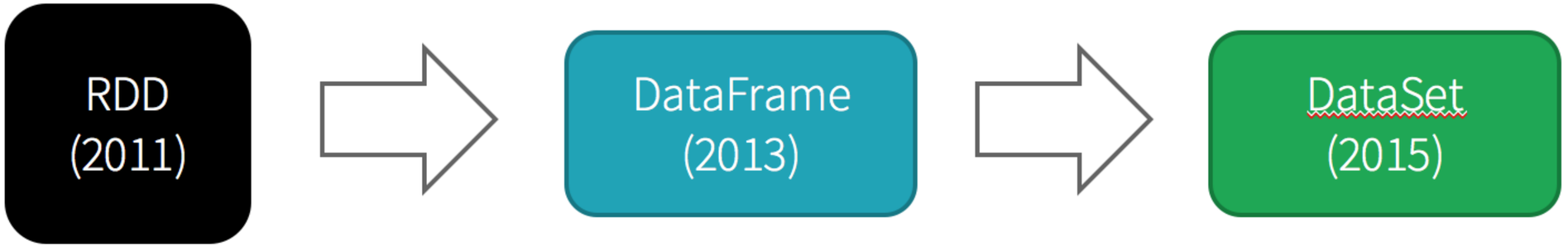
an action is eagerly evaluated

=> it is executed instantly



```
1 val schedules = spark.read.parquet("/input")
2
3 schedules
4 .filter(_.onlyInternationalFlights)
5 .map(_.makeFlightInfo)
6 .write.parquet("/output")
```

# History of Spark APIs




Distribute collection of JVM objects

Functional Operators (map, filter, etc.)

Distribute collection of Row objects

Expression-based operations and UDFs

Logical plans and optimizer

Fast/efficient internal representations

Internally rows, externally JVM objects

Almost the "Best of both worlds": **type safe + fast**

But slower than DF
Not as good for interactive analysis, especially Python


when was php released
ALL
IMAGES
NEWS
VIDEOS
MAPS
PHP / Initial release date
June 8, 1995
php
How long does trash live?
ALL
IMAGES
VIDEOS
NEWS
MAPS
Trash / Lifespan
1000 years
Lower classifications and overview

| Feature | Spark RDD | Spark DataFrame | Spark Dataset |
| --- | --- | --- | --- |
| Data representation | Immutable distributed collection of data | Structured data organized into named columns | Distributed collection of data with optional schema |
| Data processing | Fine-grained control | High-level abstraction | Ease of use and performance |
| Suitability | Developers who require precise control | Data analysts and SQL experts | Data professionals who need a balance of control and convenience |
| Key distinctions | Offers more control, but more complex | Offers more convenience, but less control | Offers a balance of control and convenience |