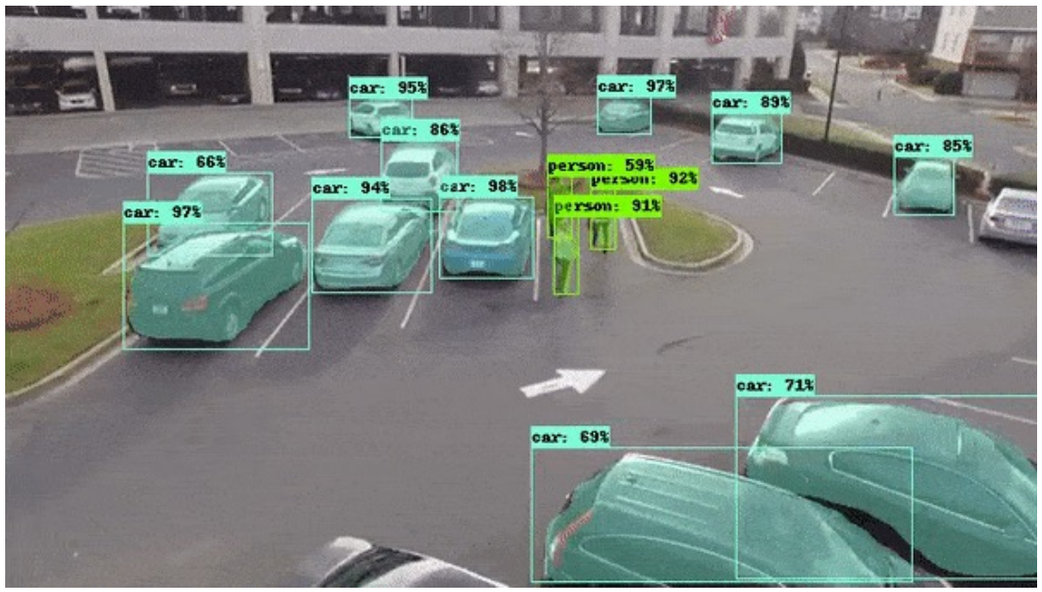
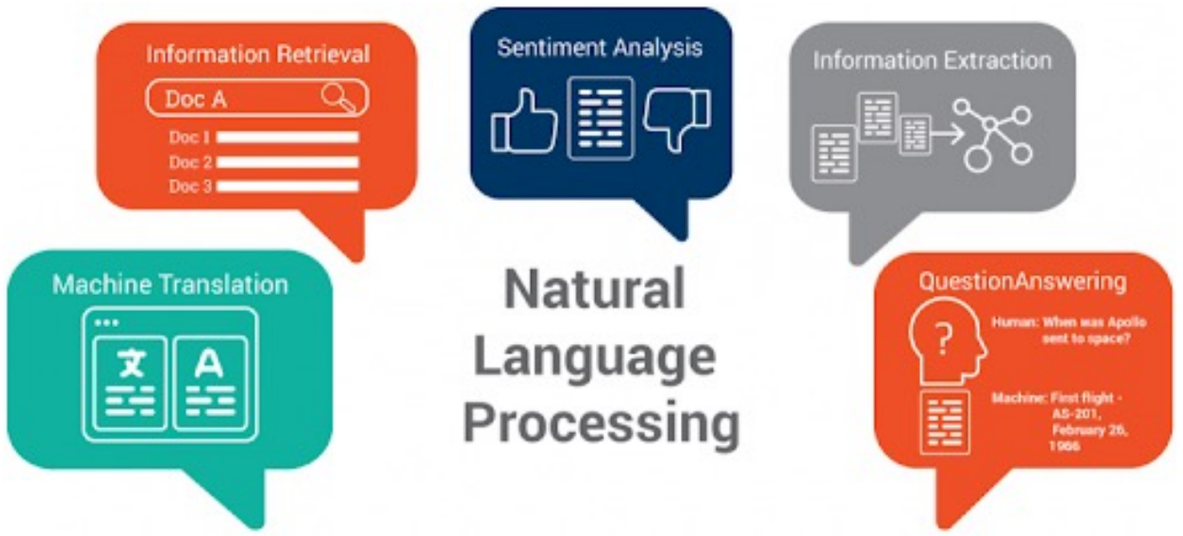
DigitalHouse >

DATA SCIENCE

Big Data e NoSQL

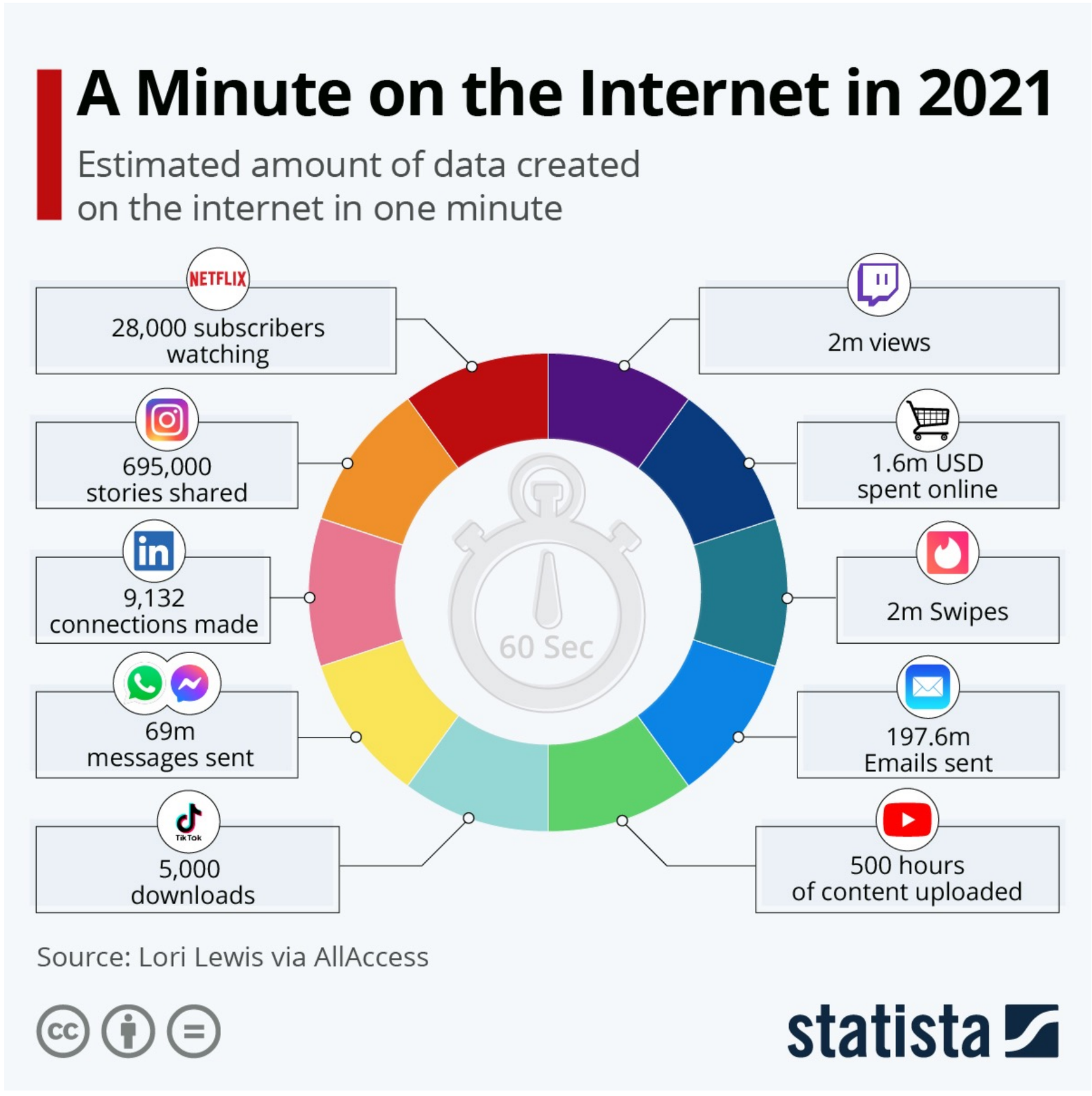
https://www.linkedin.com/in/wagner-mn-santos/





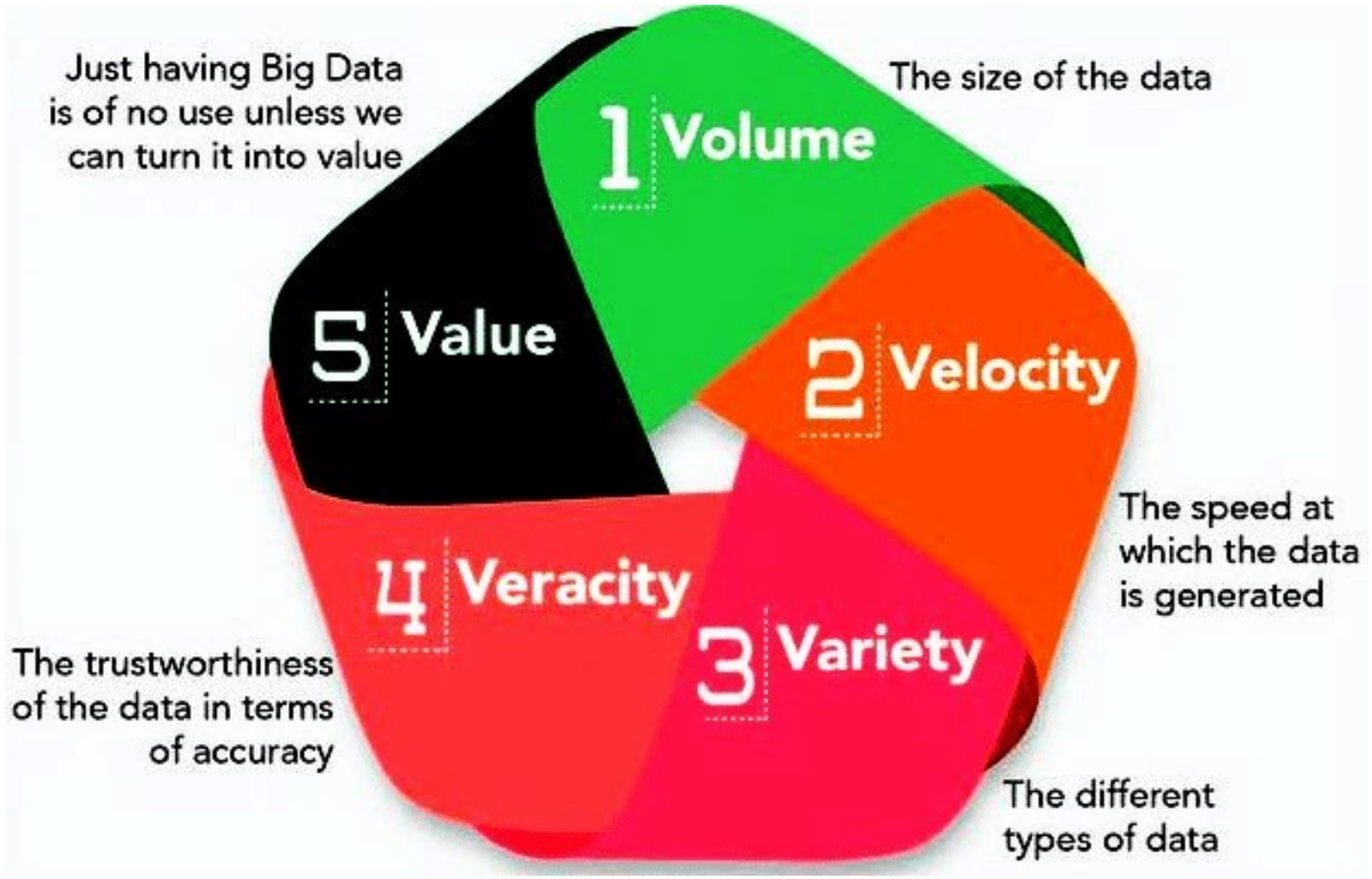
https://www.linkedin.com/in/wagner-mn-santos/

https://apasshow.com.br/blog/index.php/2019/11/21/big-data-em-supermercados-mentalidade-para-conhecer-os-dados/

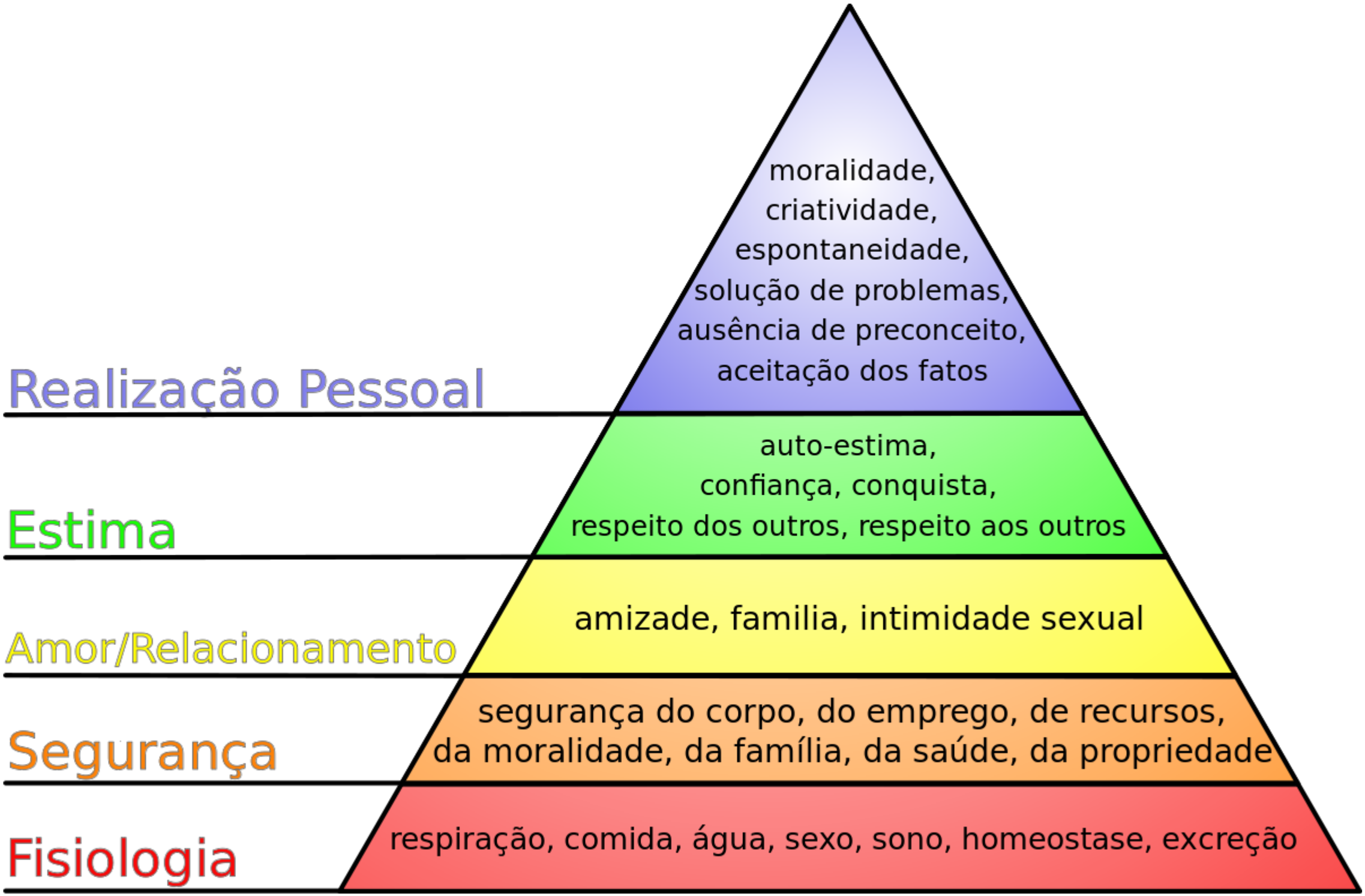

A Minute on the Internet in 2021
Estimated amount of data created on the internet in one minute
NETFLIX
28,000 subscribers watching
2m views
695,000 stories shared
1.6m USD spent online
9,132 connections made
2m Swipes
69m messages sent
197.6m Emails sent
TikTok
5,000 downloads
500 hours of content uploaded
60 Sec
Source: Lori Lewis via AllAccess
statista

https://pt.semrush.com/blog/big-data-conheca-os-5-vs-e-sua-aplicacao-pratica-para-pmes/

5 Vs – BIG DATA



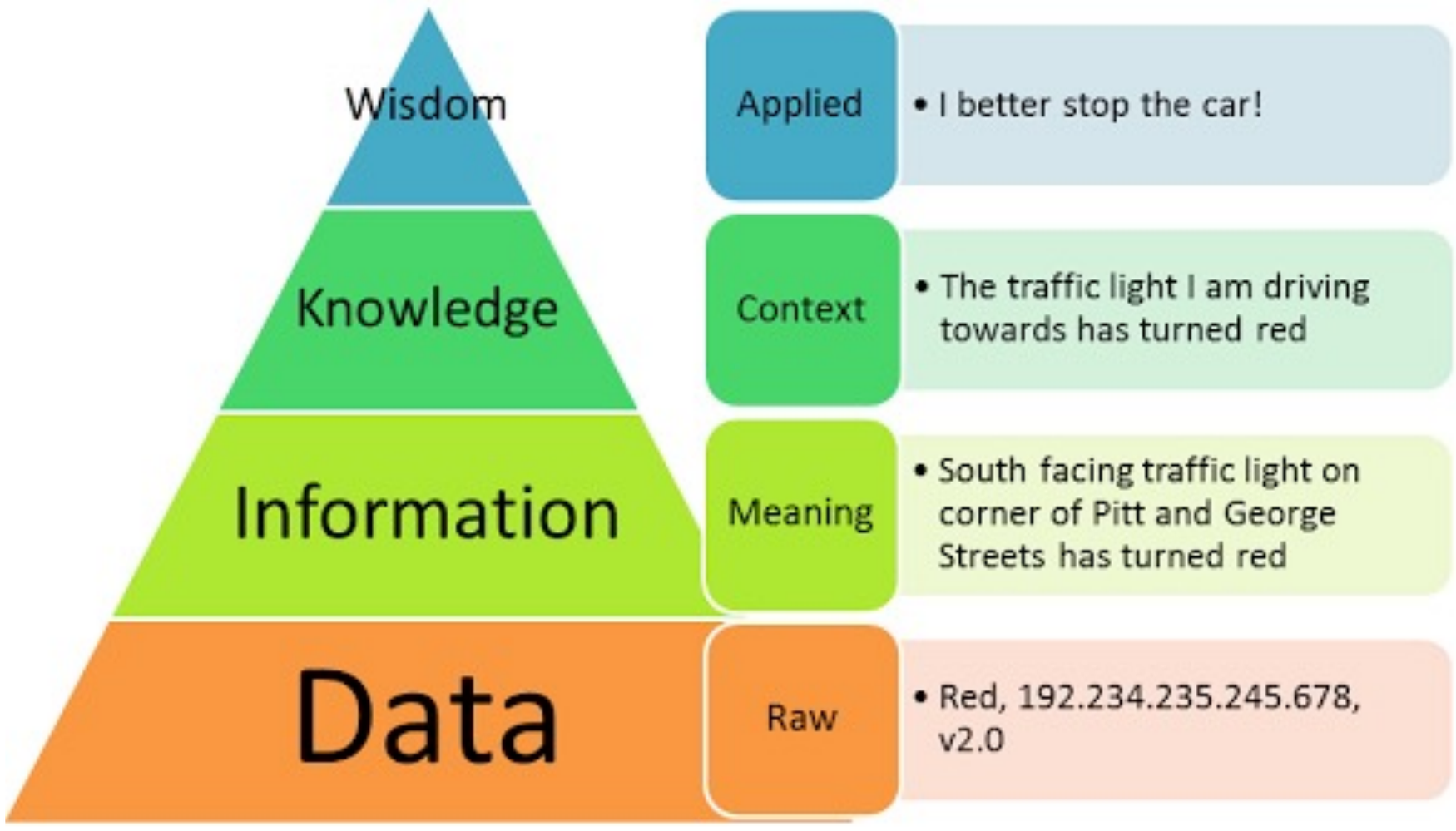
https://www.researchgate.net/figure/The-Five-V-s-of-Big-Data_fig3_315613026


Realização Pessoal
moralidade,
criatividade,
espontaneidade,
solução de problemas,
ausência de preconceito,
aceitação dos fatos
Estima
auto-estima,
confiança, conquista,
respeito dos outros, respeito aos outros
Amor/Relacionamento
amizade, familia, intimidade sexual
Segurança
segurança do corpo, do emprego, de recursos,
da moralidade, da família, da saúde, da propriedade
Fisiologia
respiração, comida, água, sexo, sono, homeostase, excreção

https://pt.wikipedia.org/wiki/Hierarquia_DIKW

A hierarquia **DIKW**, do inglês **D**ata-**I**nformation-**K**nowledge-**W**isdom, também conhecida como Pirâmide do Conhecimento, é uma hierarquia informacional utilizada principalmente nos campos da Ciência da Informação e da Gestão do Conhecimento, onde cada camada acrescenta certos atributos sobre a anterior.

HIEARARQUIA DIKW



DigitalHouse >

ROLES PROJETOS DE DADOS

Data Scientist

also known as Data Managers, statisticians.

A data scientist will be able to take data science projects from end to end. They can help store large amounts of data, create predictive modelling processes and present the findings.

Skills: Mathematics, Programming, Communication

Will use programmes such as:

SQL, Python, R

Data Engineers

also known as database administrators and data architects.

They are versatile generalists who use computer science to help process large datasets. They typically focus on coding, cleaning up data sets, and implementing requests that come from data scientists.

Skills: Programming, Mathematics, Big data

Will use programmes such as:

Hadoop, NoSQL, and Python

Data Analysts

also known as business Analysts.

They typically help people from across the company understand specific queries with charts.

Skills: Statistics, Communication, Business knowledge

Will use programmes such as:

Excel, Tableau, SQL

LEMBRANDO ALGUNS CONCEITOS DE BASES DE DADOS

Não autenticado Discussão Contribuições Criar uma conta Entrar

Artigo Discussão | Ler Editar Editar código-fonte Ver histórico | Pesquisar na Wikipédia

WIKIPÉDIA
A enciclopédia livre

Página principal
Conteúdo destacado
Eventos atuais
Esplanada
Página aleatória
Portais
Informar um erro
Loja da Wikipédia

Colaboração
Boas-vindas
Ajuda
Página de testes
Portal comunitário
Mudanças recentes
Manutenção
Criar página
Páginas novas

Banco de dados

[ocultar]

Origem: Wikipédia, a enciclopédia livre.

Esta página cita fontes confiáveis e independentes, mas que **não cobrem** todo o conteúdo (desde outubro de 2011). Ajude a inserir referências. Conteúdo não verificável poderá ser removido.—
Encontre fontes: Google (notícias, livros e acadêmico)

Esta página ou secção precisa de correção ortográfico-gramatical.
Pode conter incorreções textuais, e ainda pode necessitar de melhoria em termos de vocabulário ou coesão, para atingir um nível de qualidade superior conforme o livro de estilo da Wikipédia. Se tem conhecimentos linguísticos, sinta-se à vontade para ajudar.

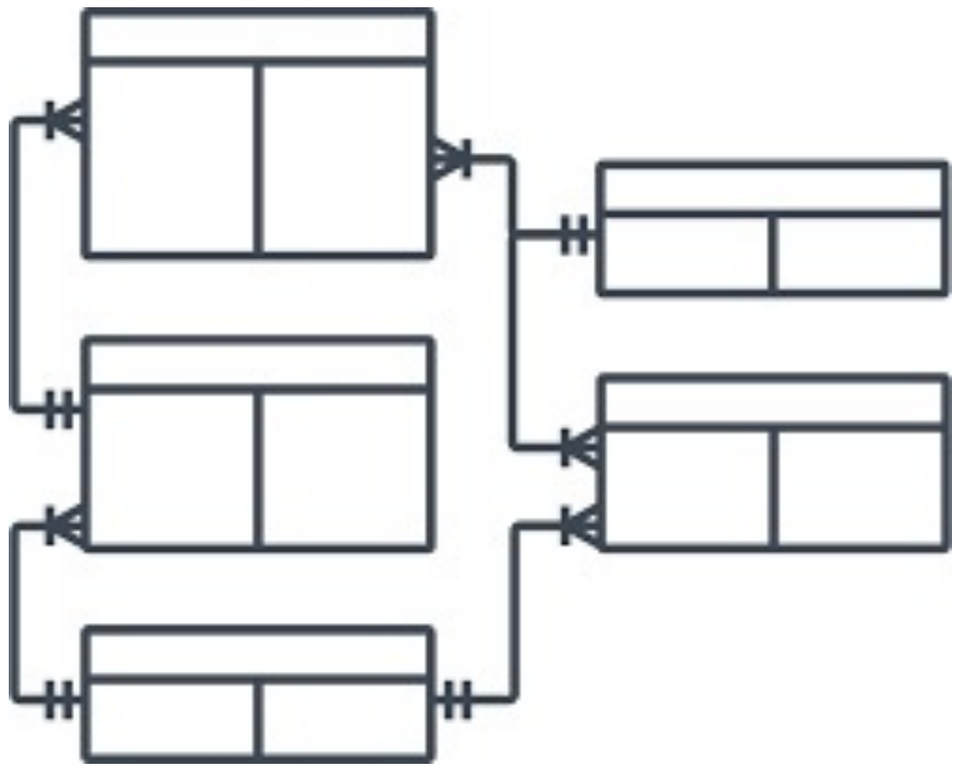
Bancos de dados (pt-BR) ou **bases de dados** (pt) [1] são conjuntos de arquivos relacionados entre si com registros sobre pessoas, lugares ou coisas. São coleções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma pesquisa ou estudo.[2][3][4] São de vital importância para empresas e há duas décadas se tornaram a principal peça dos sistemas de informação. Normalmente existem por vários anos sem alterações em sua estrutura.[5][6]

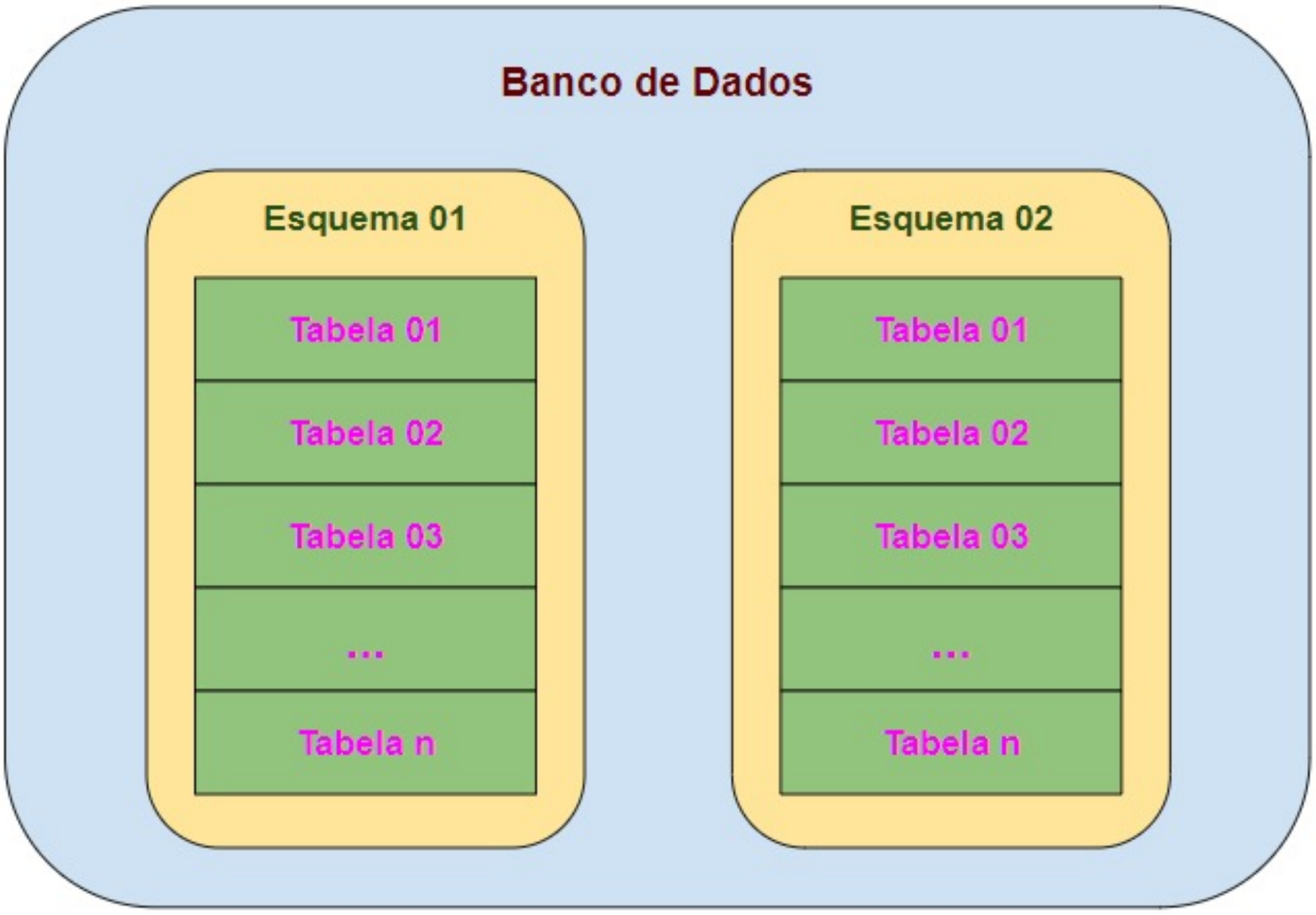
São operados pelos Sistemas Gerenciadores de Bancos de Dados (**SGBD**), que surgiram na década de 70.[7][8] Antes destes, as aplicações usavam sistemas de arquivos do sistema operacional para armazenar suas informações.[9][8] Na década de 80, a tecnologia de SGBD relacional passou a dominar o mercado, e atualmente utiliza-se praticamente apenas ela.[7][8] Outro tipo notável é o SGBD Orientado a Objetos, para quando sua estrutura ou as aplicações que o utilizam mudam constantemente.[5]

A principal aplicação de Banco de Dados é controle de operações empresariais.[10][11][12] Outra aplicação também importante é gerenciamento de informações de estudos, como fazem os Bancos de Dados Geográficos, que unem informações convencionais com espaciais.[2]

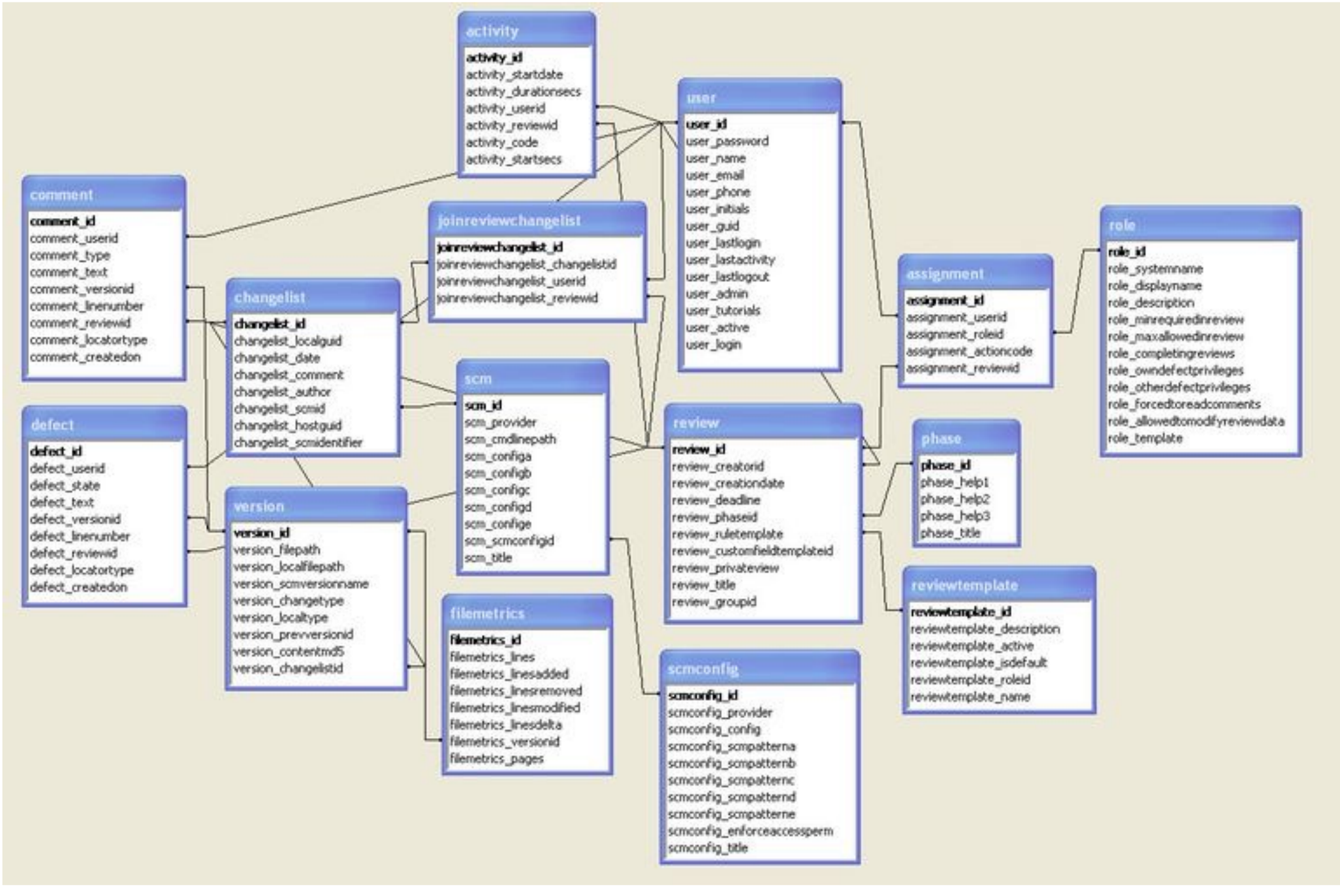
Exemplo de output de uma interrogação SQL a uma base de dados.

Aluno
• Nome • Data Nascimento • Filiação • Endereço • Telefone





http://www.bosontreinamentos.com.br/bancos-de-dados/qual-a-diferenca-entre-esquema-e-banco-de-dados/



https://www.lifewire.com/definition-of-a-schema-in-a-database-1019262

No modelo relacional normalmente as transações devem ser atômicas, consistentes, isoladas e duráveis (**ACID**):

Atomicidade: Todas as modificações de uma transação devem ser executadas ou, caso contrário, nenhuma.

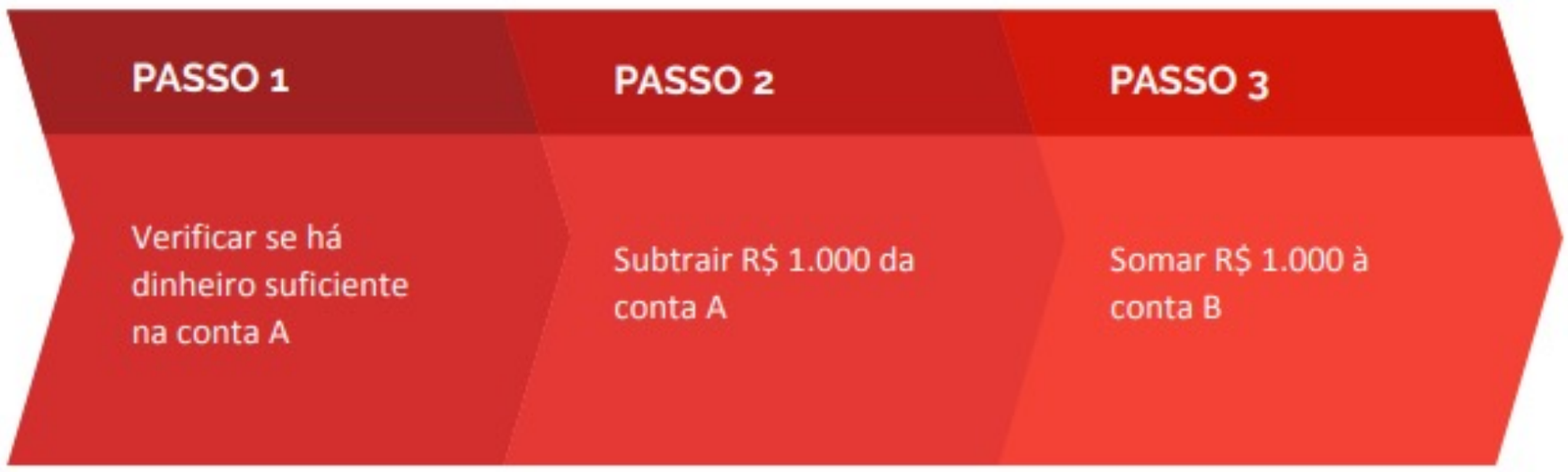
Consistência: A transação começa a ser executada em um estado válido (consistente) e termina em outro estado válido.

Isolamento: Duas ou mais transações simultâneas são executadas sem uma afetar a outra. Se as duas transações precisarem trabalhar sobre o mesmo dado, uma delas deverá esperar que a outra acabe.

Durabilidade: Garante que, uma vez realizada a transação, as mudanças ficarão armazenadas

Uma transação é um conjunto de operações pertencentes a uma tarefa, realizadas sobre um banco de dados e representando uma mudança nos dados.

Exemplo de transação bancária



- O que acontece se o processo falha entre os passos 2 e 3 (ou se apenas os passos 1e 2 são executados)?
- O que uma pessoa recebe quando quer consultar o Saldo entre os passos mencionados?

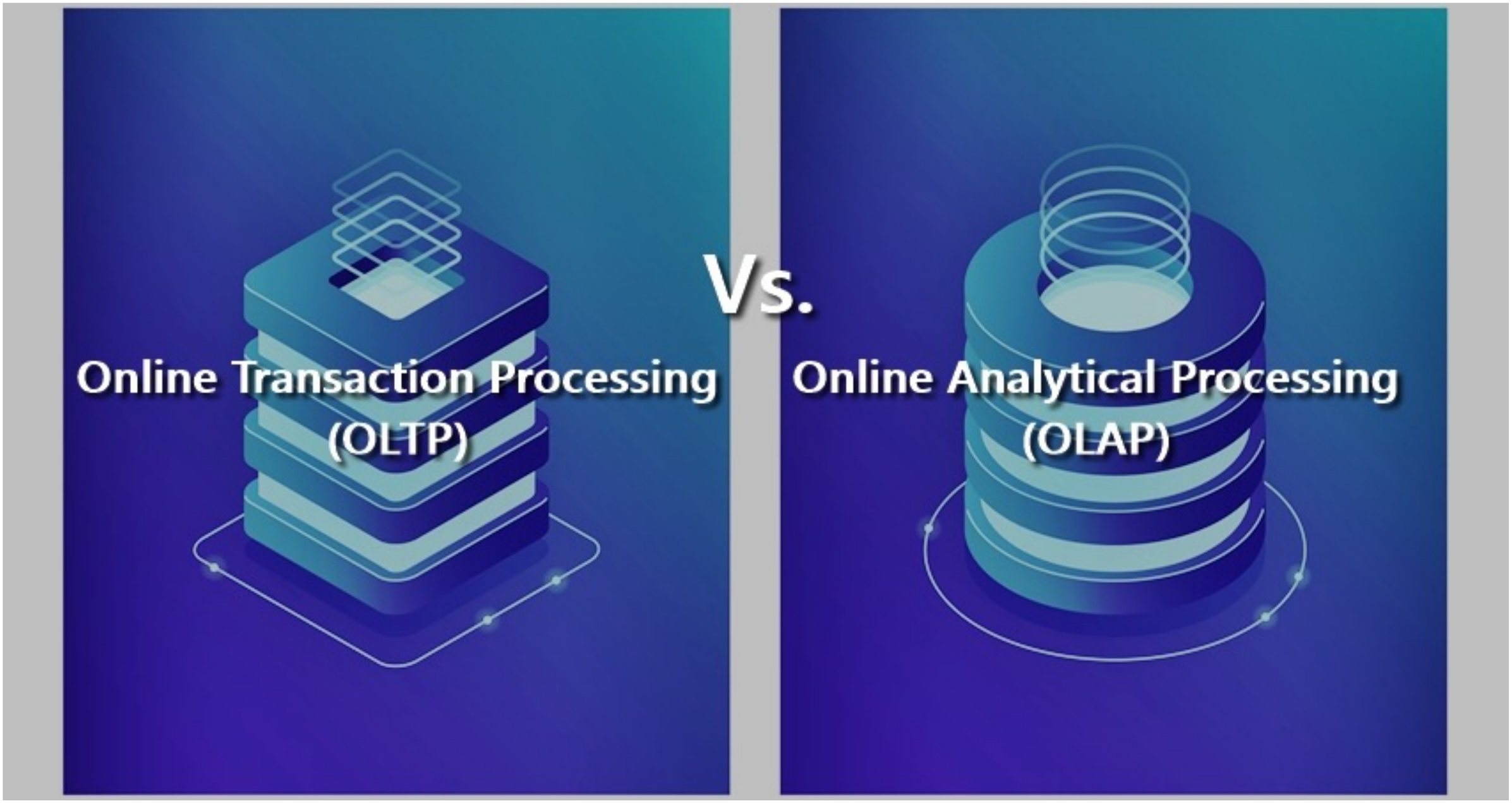

ORACLE
DATABASE

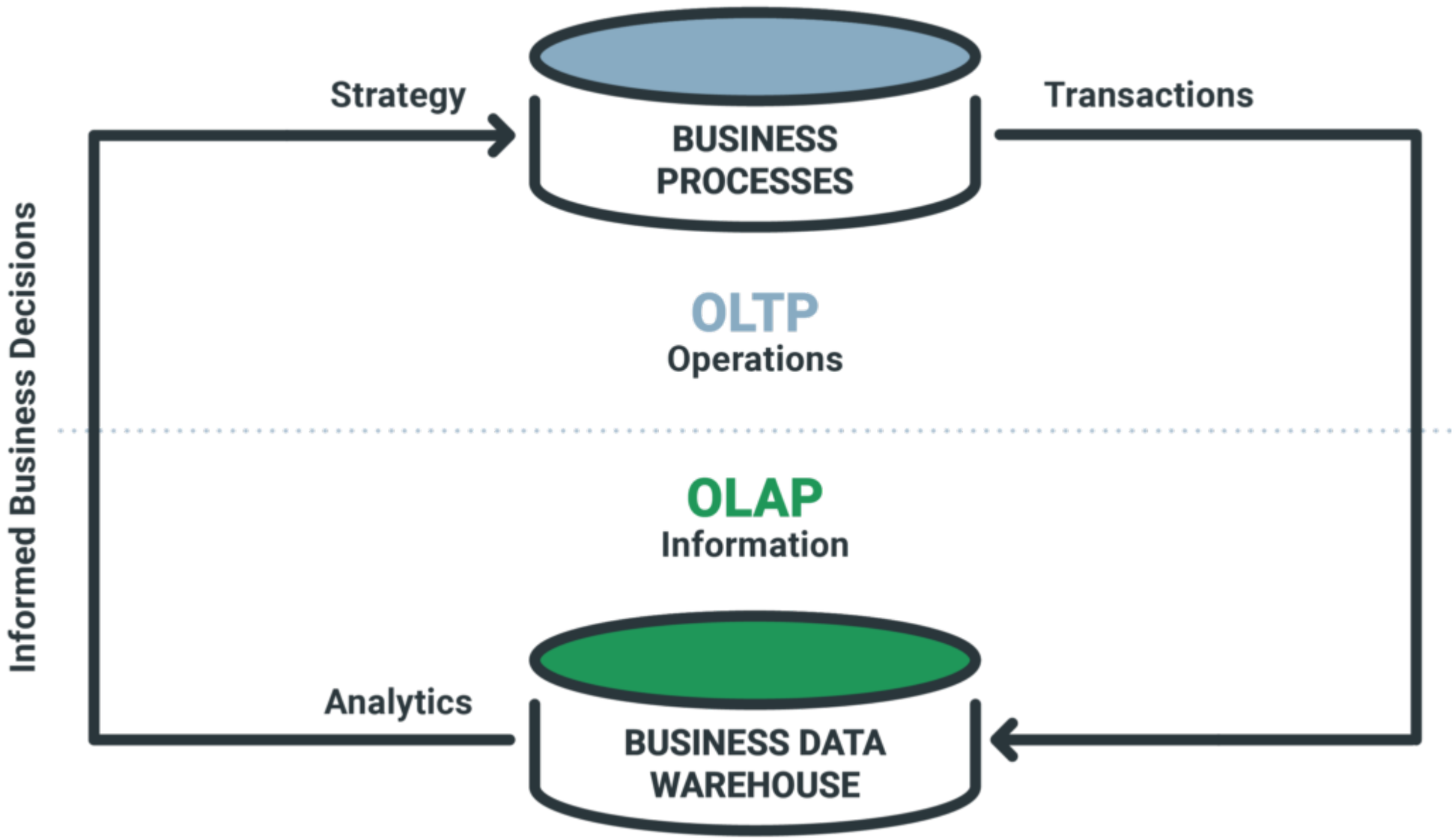

PostgreSQL


MySQL


Microsoft
SQL Server

BASES OLAP E OLTP


Vs.
Online Transaction Processing
(OLTP)
Online Analytical Processing
(OLAP)

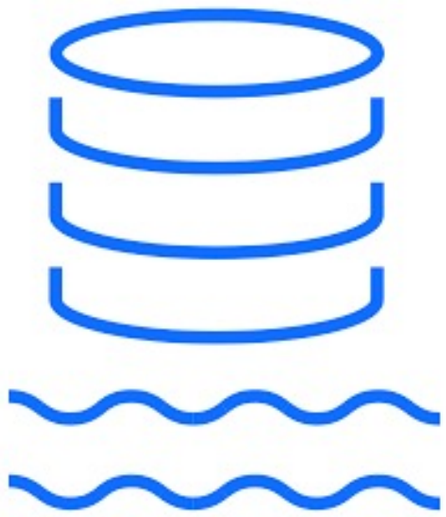

Strategy
BUSINESS PROCESSES
Transactions
OLTP
Operations
Informed Business Decisions
OLAP
Information
Analytics
BUSINESS DATA WAREHOUSE

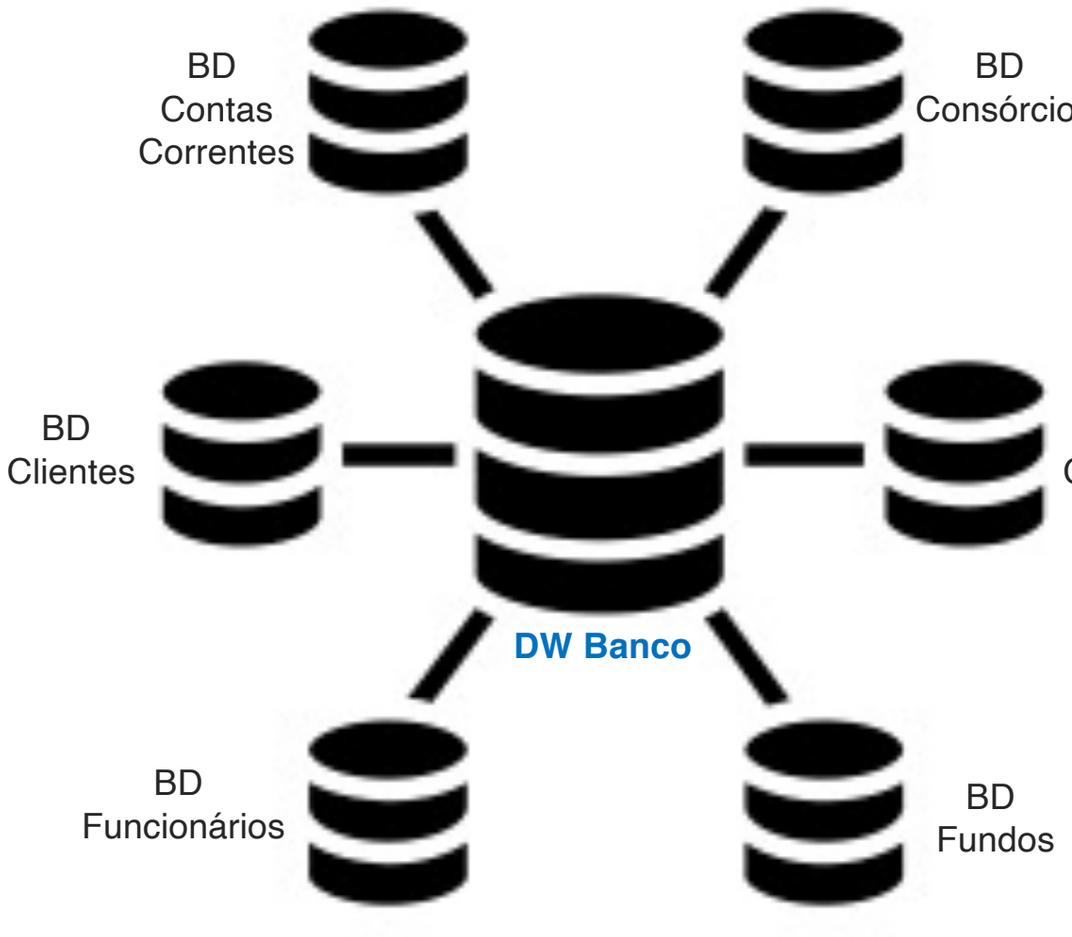
OLAP x OLTP

	OLAP	OLTP
Foco	Foco no nível estratégico da organização. Visa a análise empresarial e tomada de decisão.	Foco no nível operacional da organização. Visa a execução operacional do negócio.
Performance	Otimização para a leitura e geração de análises e relatórios gerenciais.	Alta velocidade na manipulação de dados operacionais, porém ineficiente para geração de análises gerenciais.
Estrutura dos dados	Os dados estão estruturados na modelagem dimensional. Os dados normalmente possuem alto nível de sumarização.	Os dados são normalmente estruturados em um modelo relacional normalizado, otimizado para a utilização transacional. Os dados possuem alto nível de detalhes.
Armazenamento	O armazenamento é feito em estruturas de *Data Warehouse* com otimização no desempenho em grandes volumes de dados.	O armazenamento é feito em sistemas convencionais de banco de dados através dos sistemas de informações da organização.
Abrangência	É utilizado pelos gestores e analistas para a tomada de decisão.	É utilizado por técnicos e analistas e engloba vários usuários da organização.
Frequência de atualização	A atualização das informações é feita no processo de carga dos dados. Frequência baixa, podendo ser diária, semanal, mensal ou anual (ou critério específico).	A atualização dos dados é feita no momento da transação. Frequência muito alta de atualizações.
Volatilidade	Dados históricos e não voláteis. Os dados não sofrem alterações, salvo necessidades específicas (por motivos de erros ou inconsistências de informações).	Dados voláteis, passíveis de modificação e exclusão.
Tipos de permissões nos dados	É permitido apenas a inserção e leitura. Sendo que para o usuário está apenas disponível a leitura.	Podem ser feito leitura, inserção, modificação e exclusão dos dados.

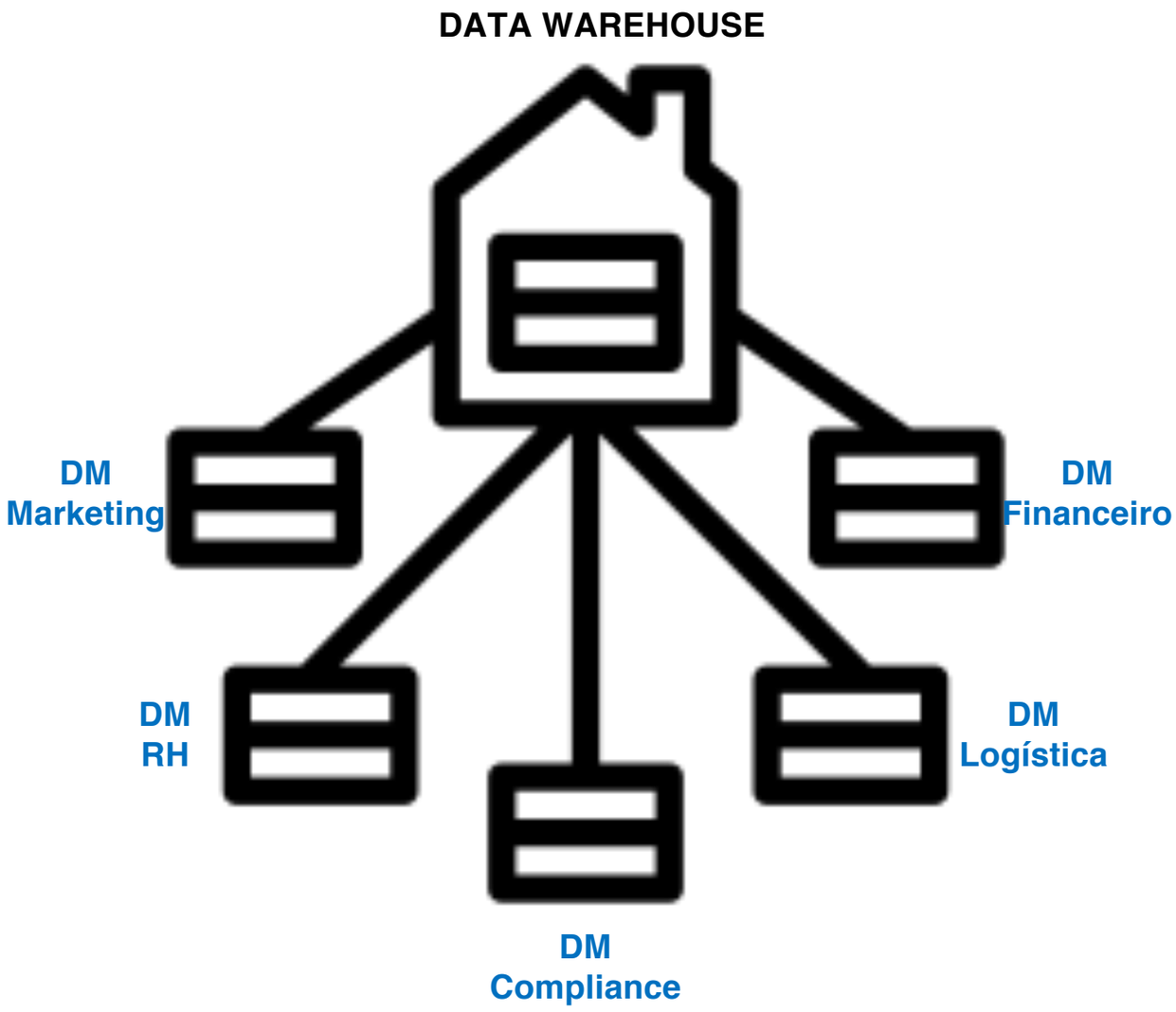
https://canaltech.com.br/business-intelligence/o-que-significa-oltp-e-olap-na-pratica/

QUAIS AS DIFERENÇAS ENTRE DATA BASE, DATA MART, DATAWAREHOUSE E DATA LAKE

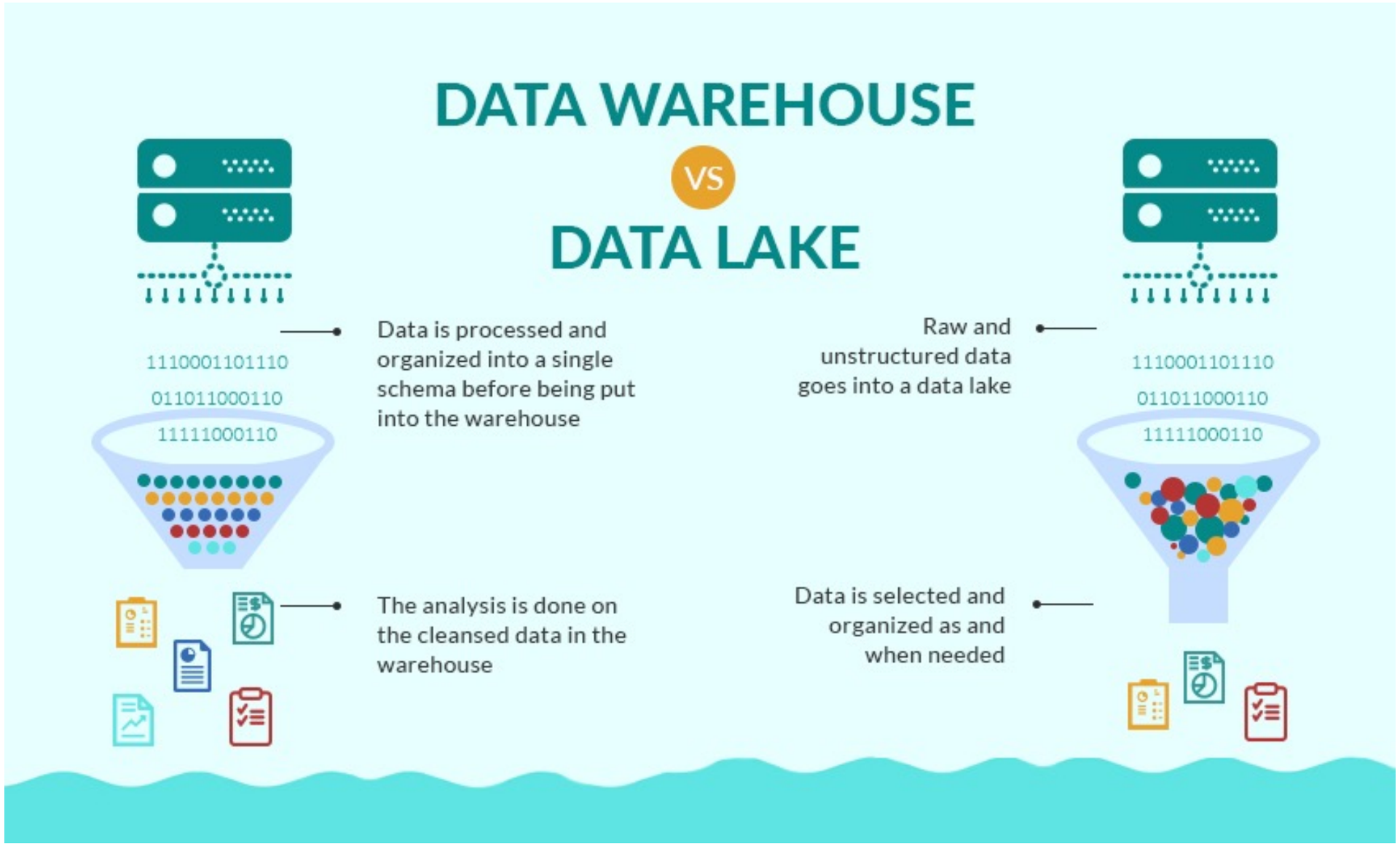




O Data Warehouse (DW) consolida as informações de diversas bases de dados distintas, normalmente sendo utilizado para fins Informacionais (OLAP).



Data Marts (DM) são seleções de dados do Data Warehouse, normalmente gerados para uma área e/ou fim específico.



Data Lakes são utilizados em Big Data e se distinguem de Data Warehouses por não exigir uma estrutura (*schema)* definida no momento da gravação, somente na recuperação. São muito bons para gravar Dados não Estruturados e normalmente possuem uma capacidade muito grande de paralelização das tarefas.

Structured Data

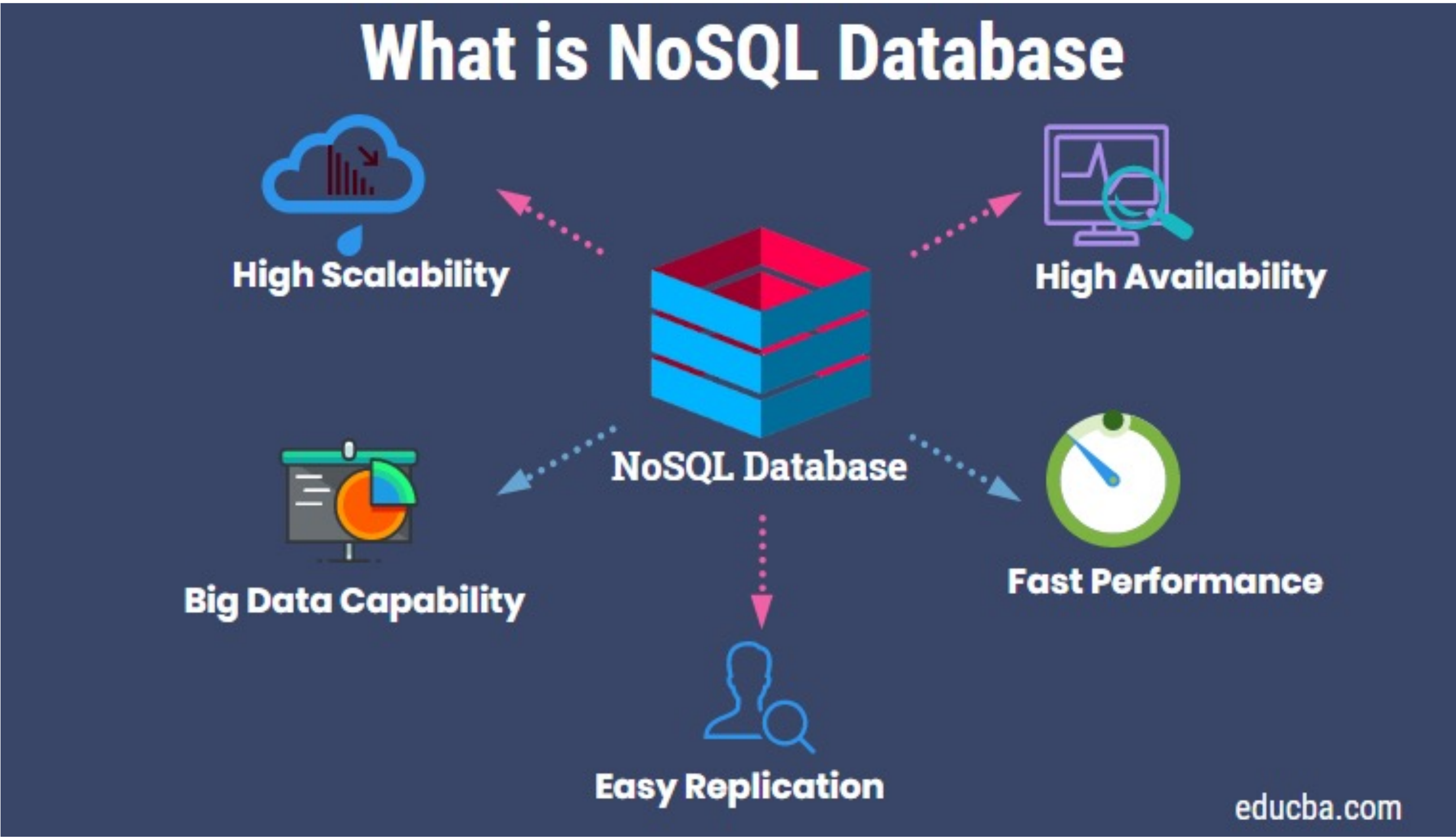
0.103	0.176	0.387	0.300	0.379
0.333	0.384	0.564	0.587	0.857
0.421	0.309	0.654	0.729	0.228
0.266	0.750	1.056	0.936	0.911
0.225	0.326	0.643	0.337	0.721
0.187	0.586	0.529	0.340	0.829
0.153	0.485	0.560	0.428	0.628

Unstructured Data

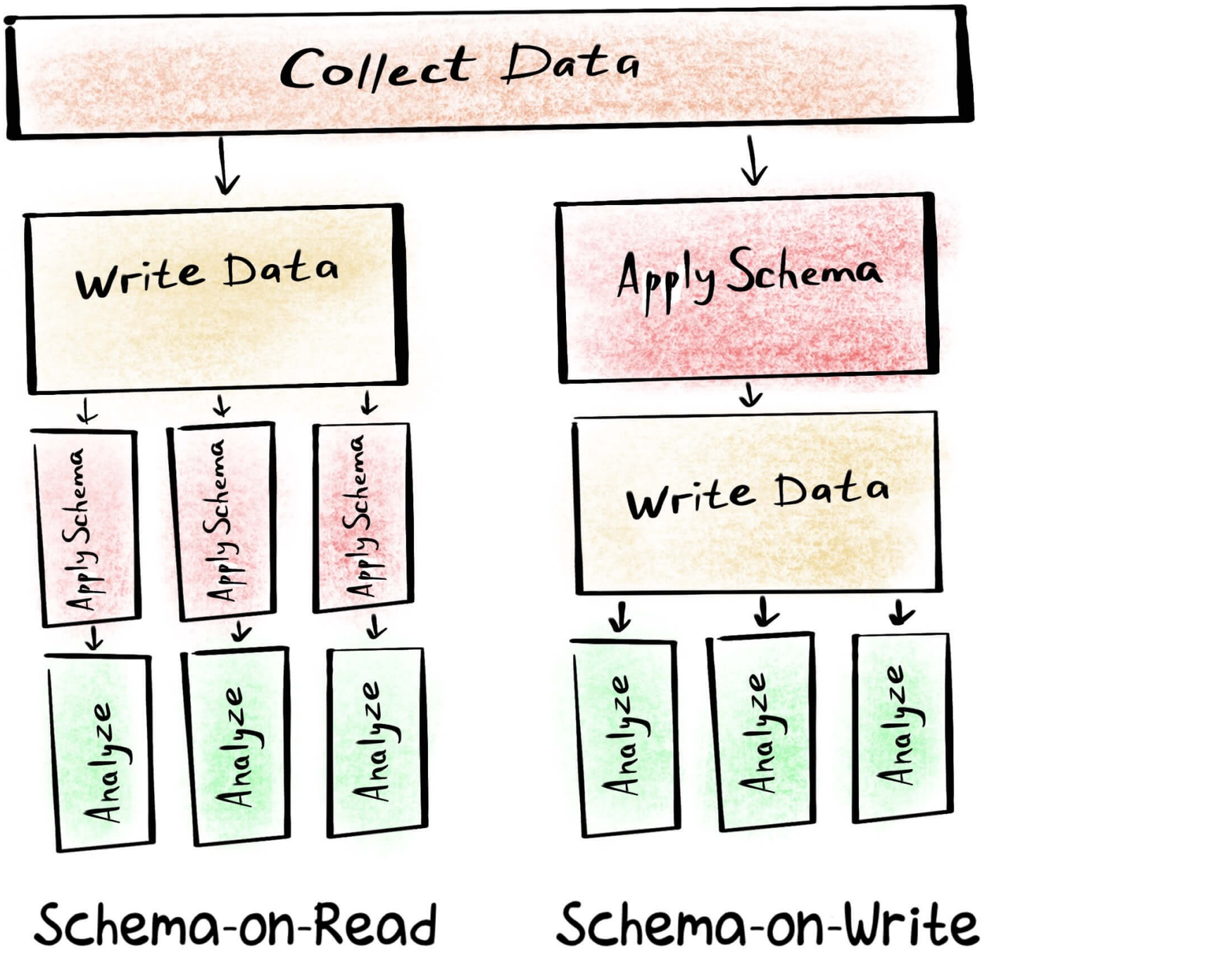

BIG DATA
AI
11001
DATA LAKE

NO-SQL DATABASES


NoSQL
Not Only SQL



https://www.educba.com/what-is-nosql-database/

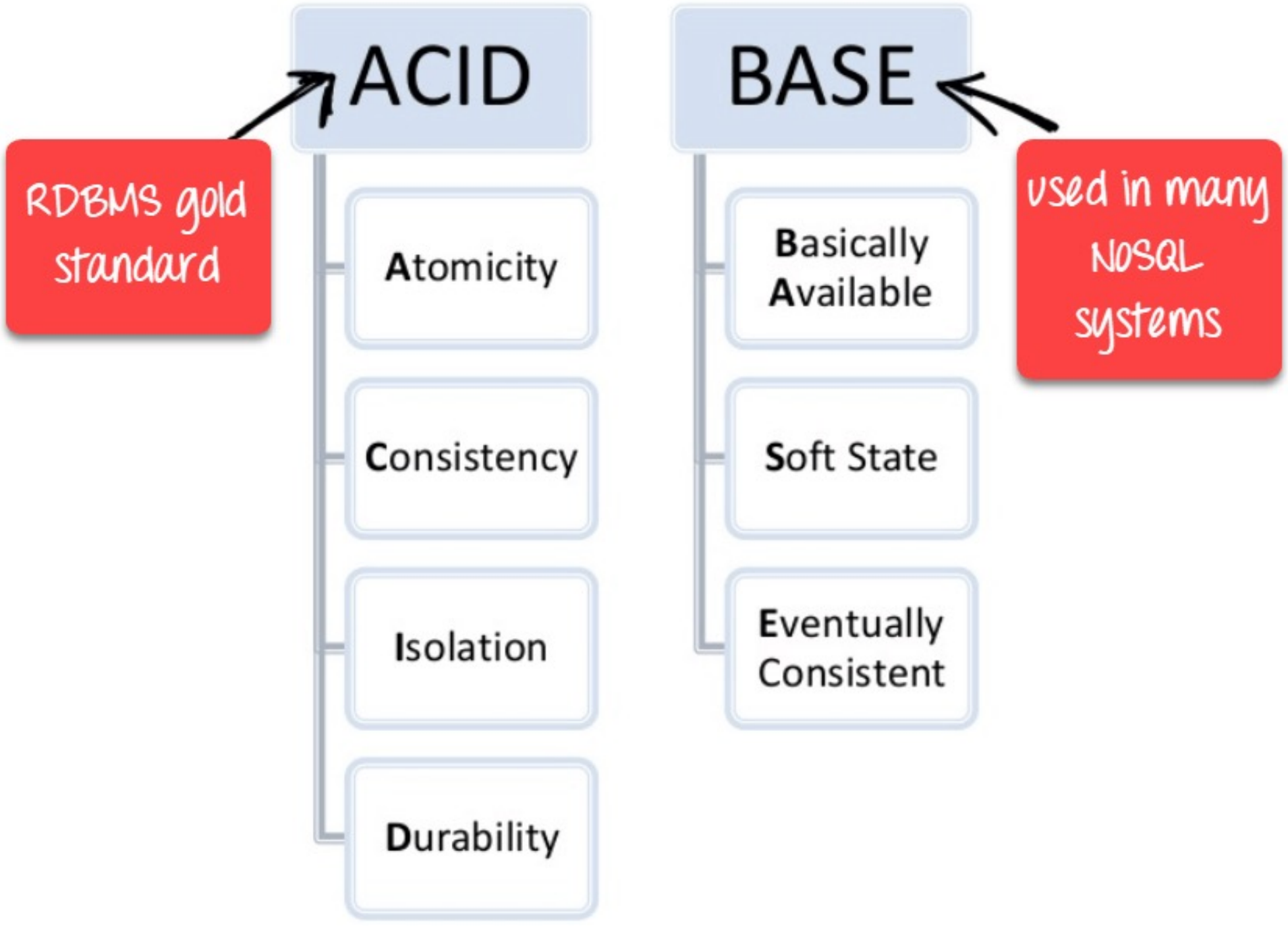


https://luminousmen.com/post/schema-on-read-vs-schema-on-write

Schema-on-Write	Schema-on-Read
- fast reads	- slower reads
- slower loads	- fast loads
- not agile	- very agile
- structured	- structured/unstructured
- fewer errors	- more errors
- SQL	- NoSQL

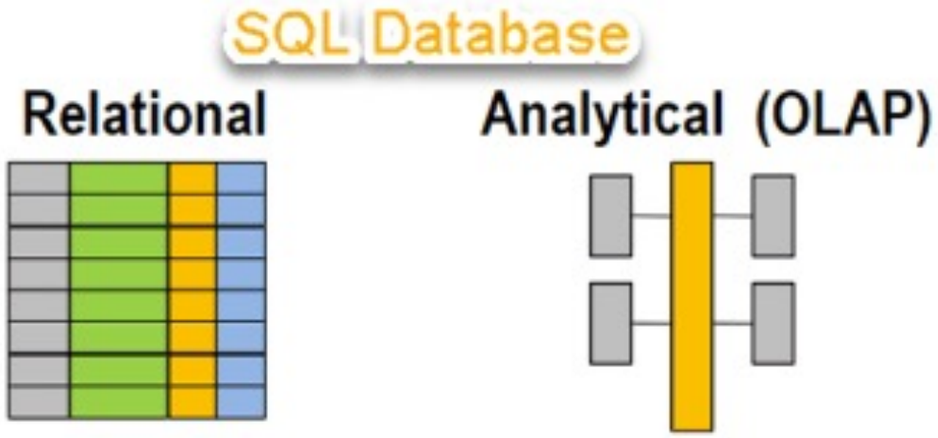
https://luminousmen.com/post/schema-on-read-vs-schema-on-write

@luminousmen.com

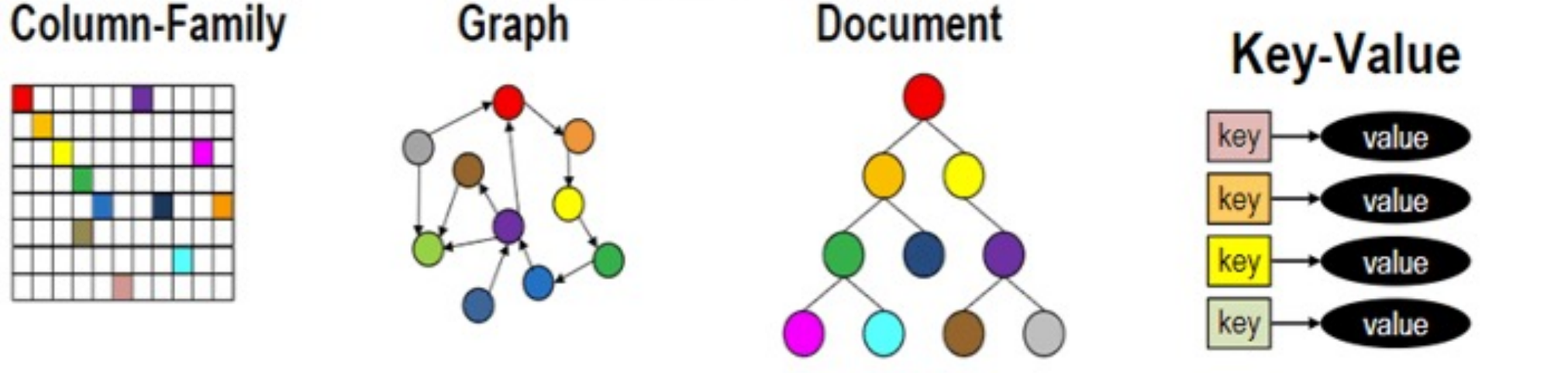


https://www.guru99.com/nosql-tutorial.html

Schema on-Write



Schema on-Read



https://www.sqlindia.com/what-is-nosql-and-how-it-is-different-than-sql/

DB-ENGINES

English
Deutsch

Knowledge Base of Relational and NoSQL Database Management Systems

provided by solid IT

Select a ranking

- Complete ranking
- Relational DBMS
- Key-value stores
- Document stores
- Time Series DBMS
- Graph DBMS
- Object oriented DBMS
- Search engines
- RDF stores
- Wide column stores
- Multivalue DBMS
- Native XML DBMS
- Spatial DBMS
- Event Stores
- Content stores
- Navigational DBMS

Special reports

- Ranking by database model
- Open source vs. commercial

Featured Products

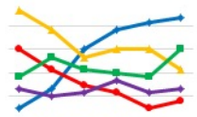
Ranking > Complete Ranking

RSS Feed

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the method of calculating the scores.

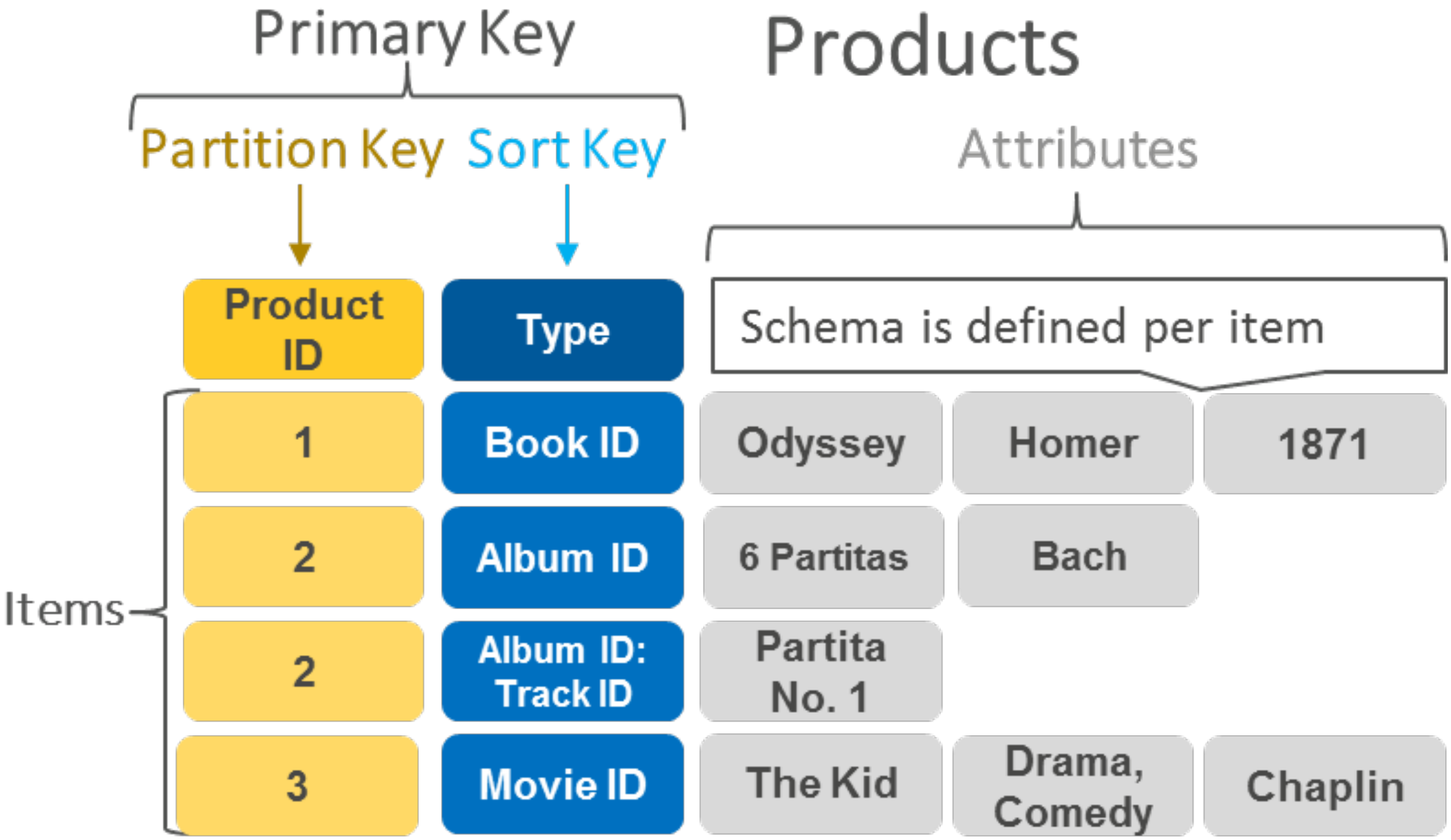


trend chart

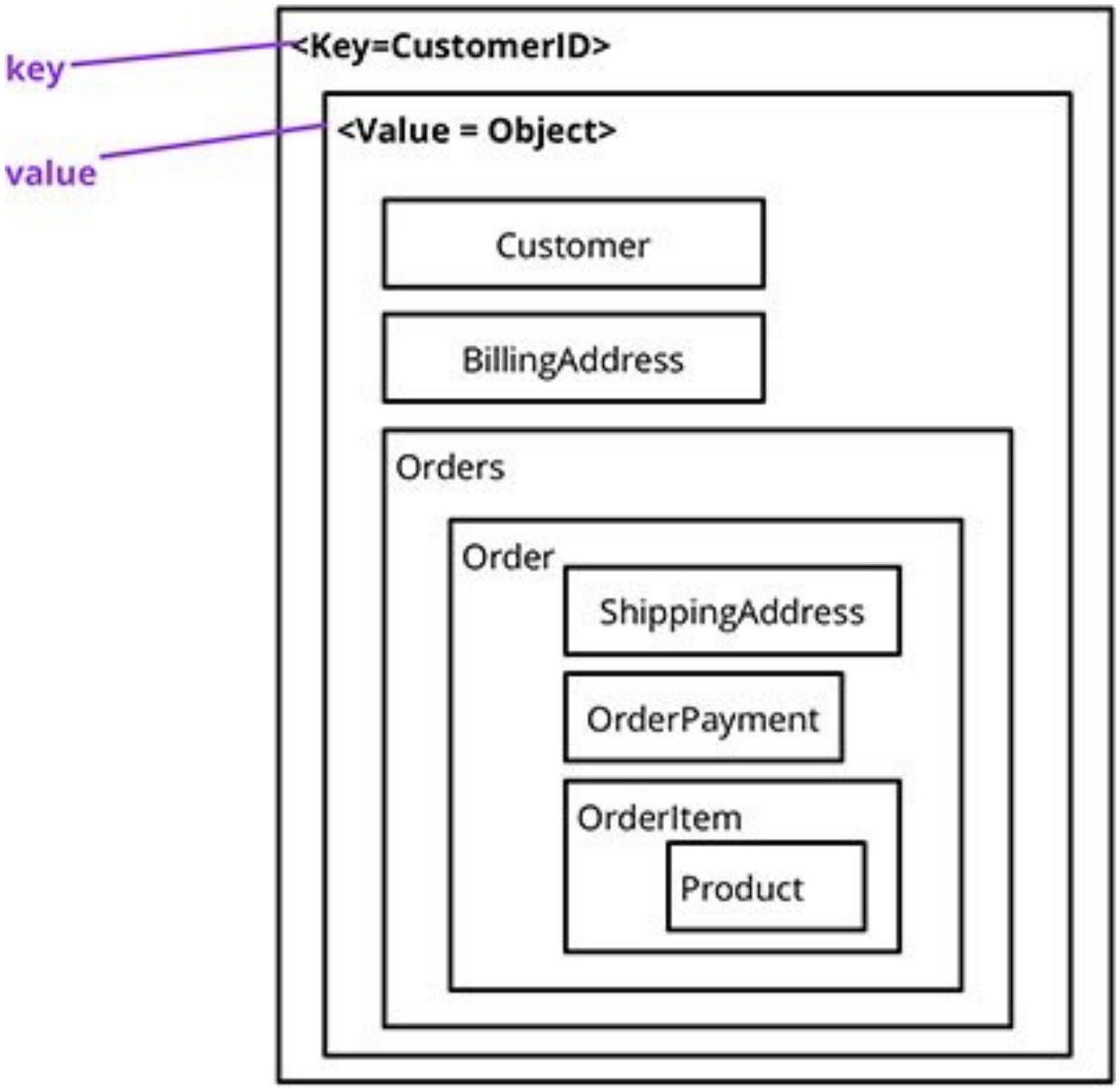
373 systems in ranking, August 2021

Rank Aug 2021	Rank Jul 2021	Rank Aug 2020	DBMS	Database Model	Score Aug 2021	Score Jul 2021	Score Aug 2020
1.	1.	1.	Oracle	Relational, Multi-model	1269.26	+6.59	-85.90
2.	2.	2.	MySQL	Relational, Multi-model	1238.22	+9.84	-23.36
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	973.35	-8.61	-102.53
4.	4.	4.	PostgreSQL	Relational, Multi-model	577.05	-0.10	+40.28
5.	5.	5.	MongoDB	Document, Multi-model	496.54	+0.38	+52.98
6.	6.	↑ 7.	Redis	Key-value, Multi-model	169.88	+1.58	+17.01
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	165.46	+0.31	+3.01
8.	8.	8.	Elasticsearch	Search engine, Multi-model	157.08	+1.32	+4.76
9.	9.	9.	SQLite	Relational	129.81	-0.39	+3.00
10.	↑ 11.	10.	Microsoft Access	Relational	114.84	+1.39	-5.02
11.	↓ 10.	11.	Cassandra	Wide column	113.66	-0.35	-6.18
12.	12.	12.	MariaDB	Relational, Multi-model	98.98	+0.99	+8.06
13.	13.	13.	Splunk	Search engine	90.60	+0.55	+0.69
14.	14.	↑ 15.	Hive	Relational	83.93	+1.26	+8.64
15.	15.	↑ 17.	Microsoft Azure SQL Database	Relational, Multi-model	75.15	-0.06	+18.31
16.	16.	16.	Amazon DynamoDB	Multi-model	74.90	-0.30	+10.15
17.	17.	↓ 14.	Teradata	Relational, Multi-model	68.82	-0.13	-7.96

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



https://aws.amazon.com/pt/nosql/key-value/



http://www.informit.com/articles/article.aspx?p=2266741

Document 1

```
{
  "id": "1",
  “name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  “fullName": "Sarah Jones",
  “isActive": false,
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": "3",
  “fullName":
  {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

https://lennilobel.wordpress.com/2015/06/01/relational-databases-vs-nosql-document-databases/

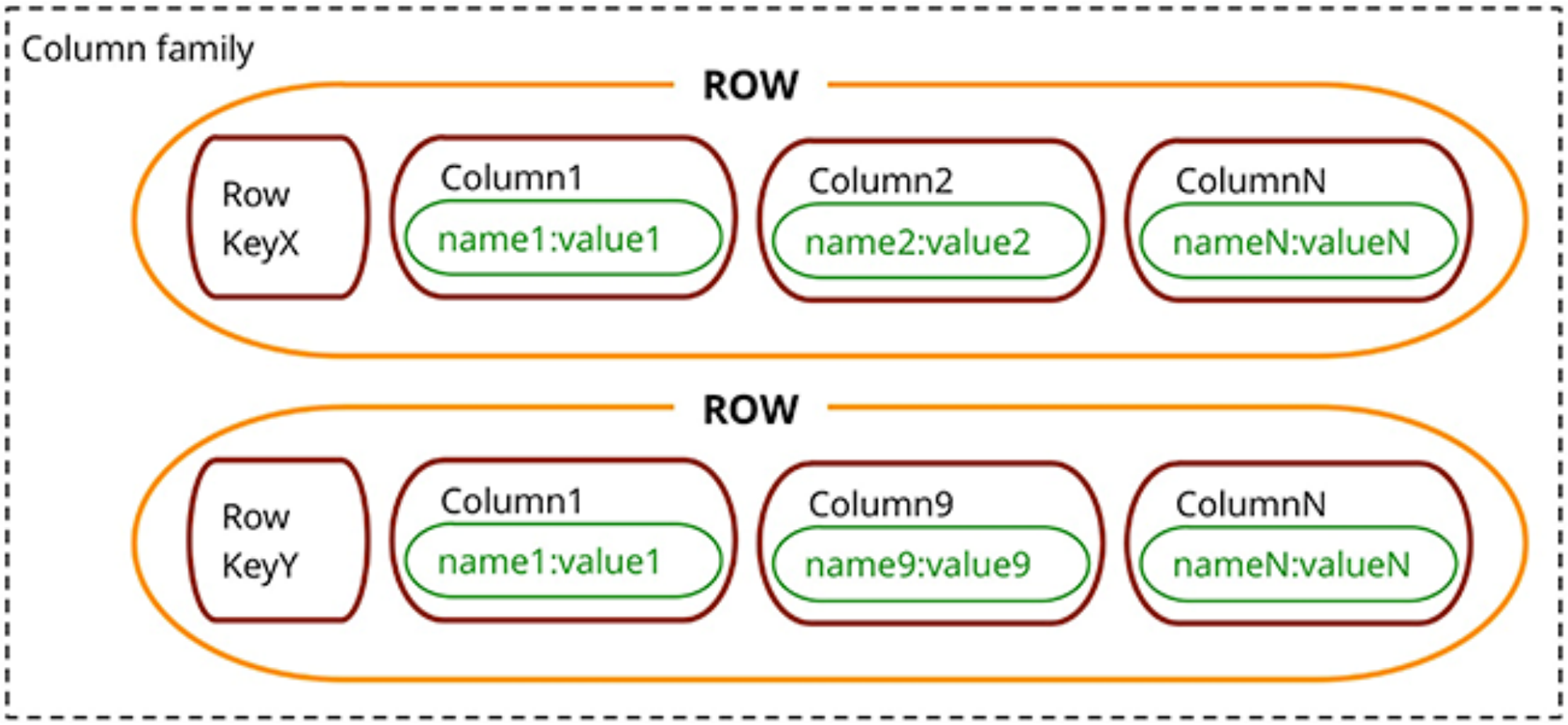
Key

<Document>

```
{
  "customerid": "fc986e48ca6"
  "customer":
  {
  "firstname": "Pramod",
  "lastname": "Sadalage",
  "company": "ThoughtWorks"
  "likes": [ "Biking","Photography" ]
  }
  "billingaddress":
  { "state": "AK",
      "city": "DILLINGHAM",
      "type": "R"
    }
}
```

mongoDB

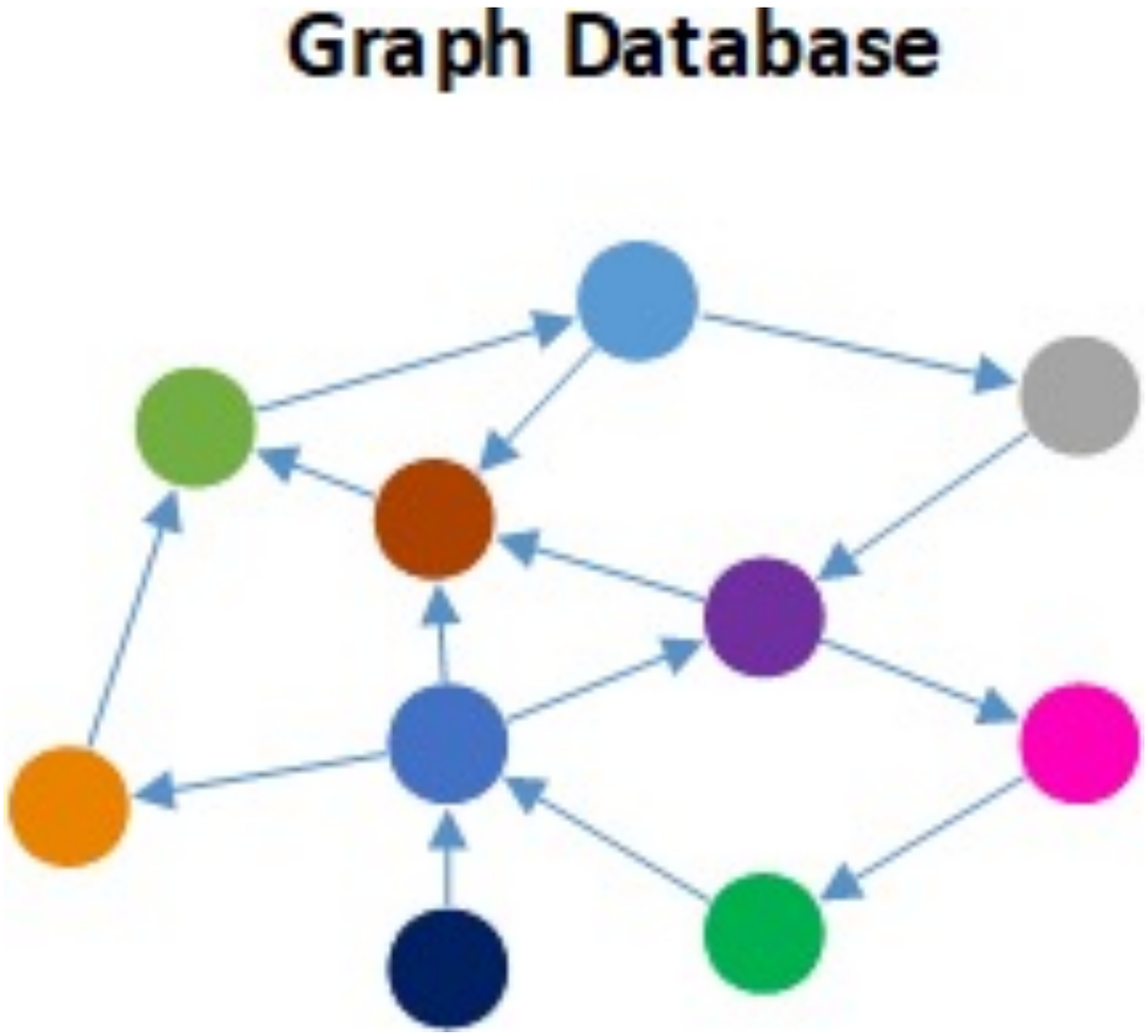
http://www.informit.com/articles/article.aspx?p=2266741

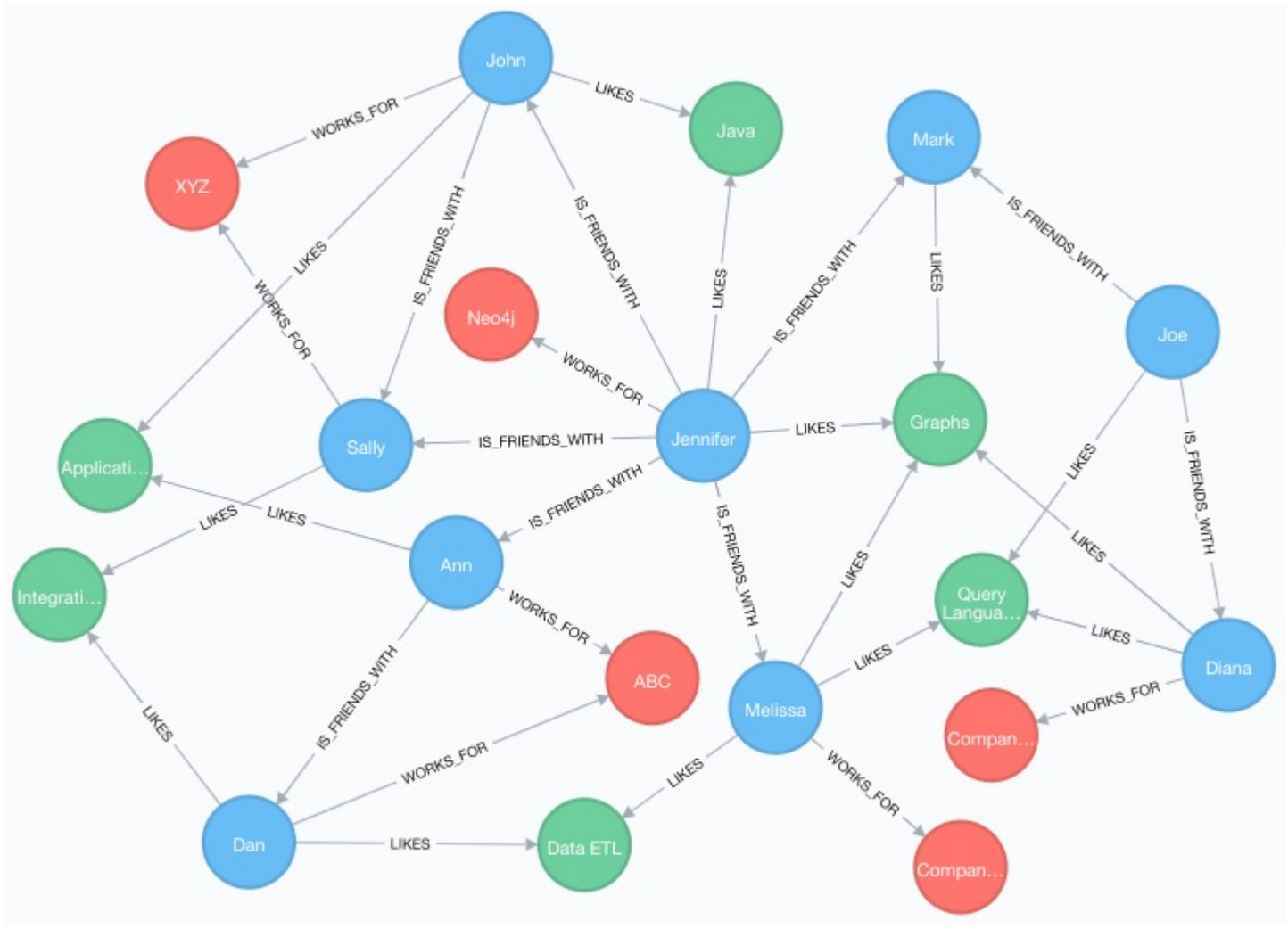


http://www.informit.com/articles/article.aspx?p=2266741

cassandra

APACHE HBASE

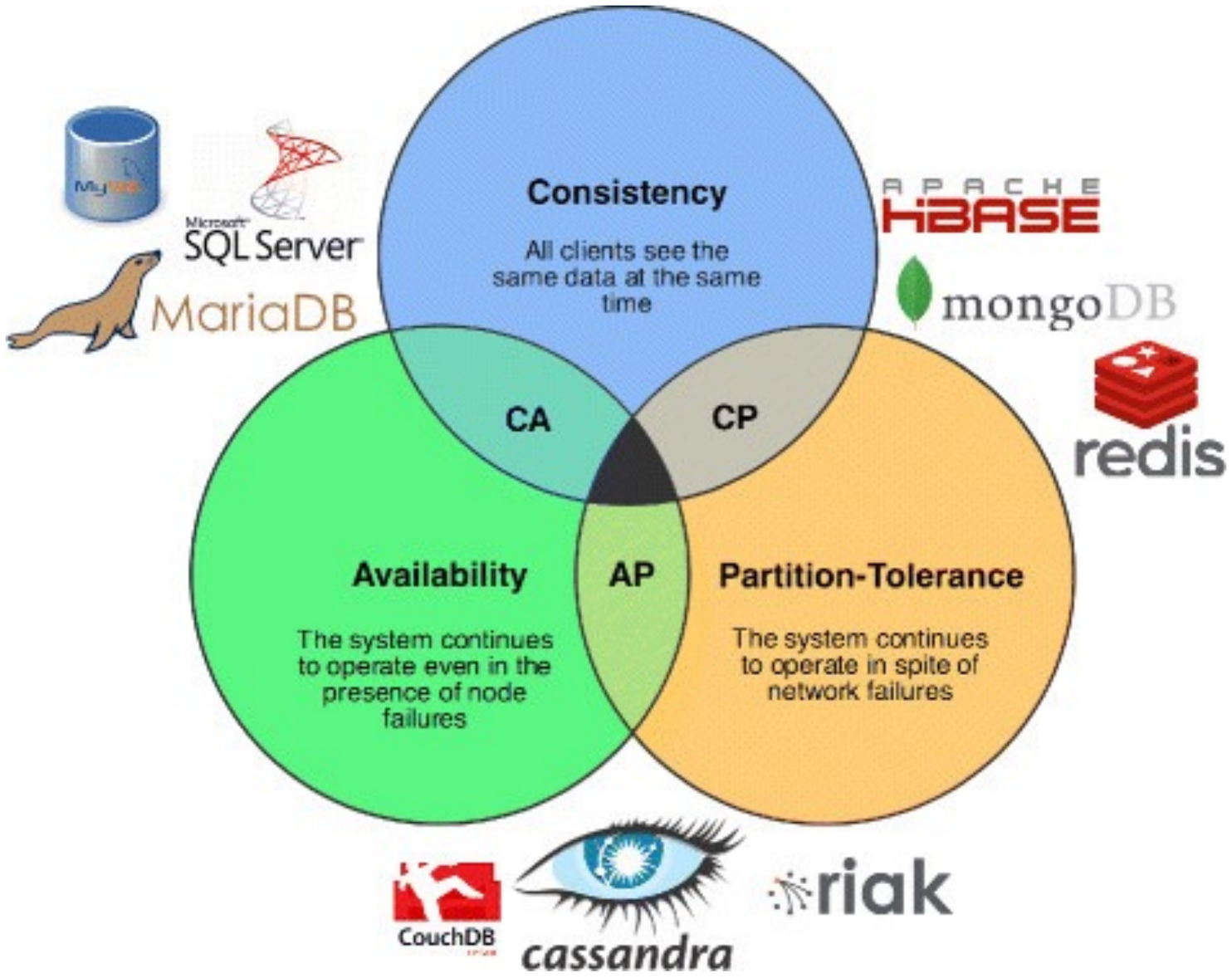

Graph Database



The #1 Database for Connected Data

Obs: o neo4j atende aos requisitos transacionais – ACID.

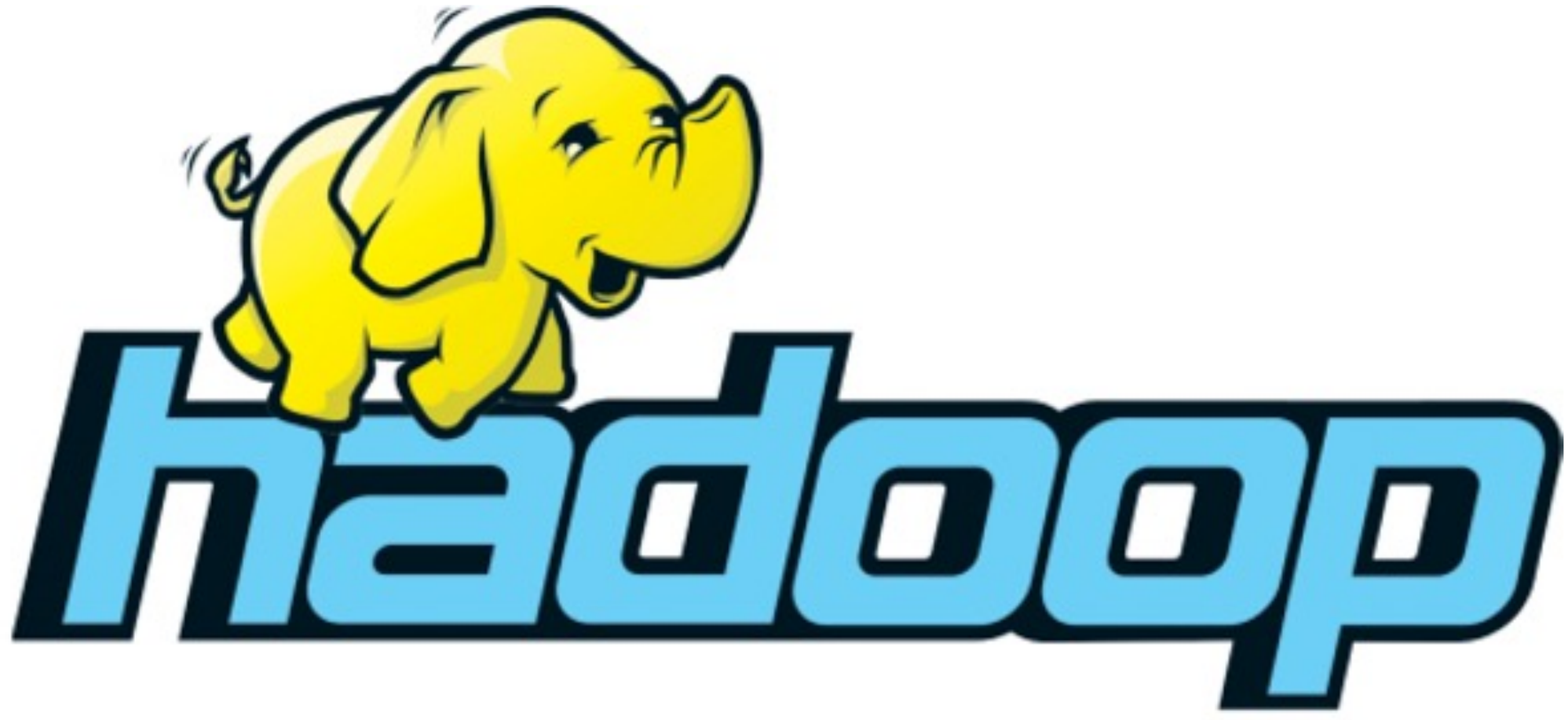
https://youtu.be/QlqylUeqeis



https://medium.com/@lucascodejs/o-teorema-de-cap-utilizado-para-auxiliar-um-servidor-de-sistema-distribu%C3%ADdos-f73d6a4c5d9

https://en.wikipedia.org/wiki/CAP_theorem

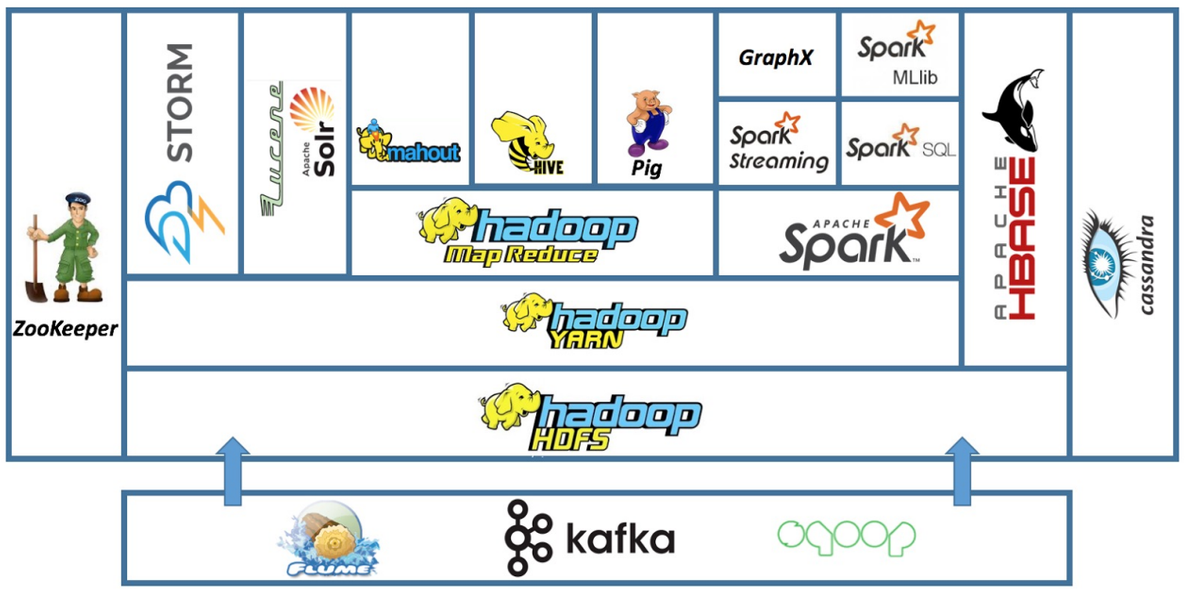
APACHE HADOOP

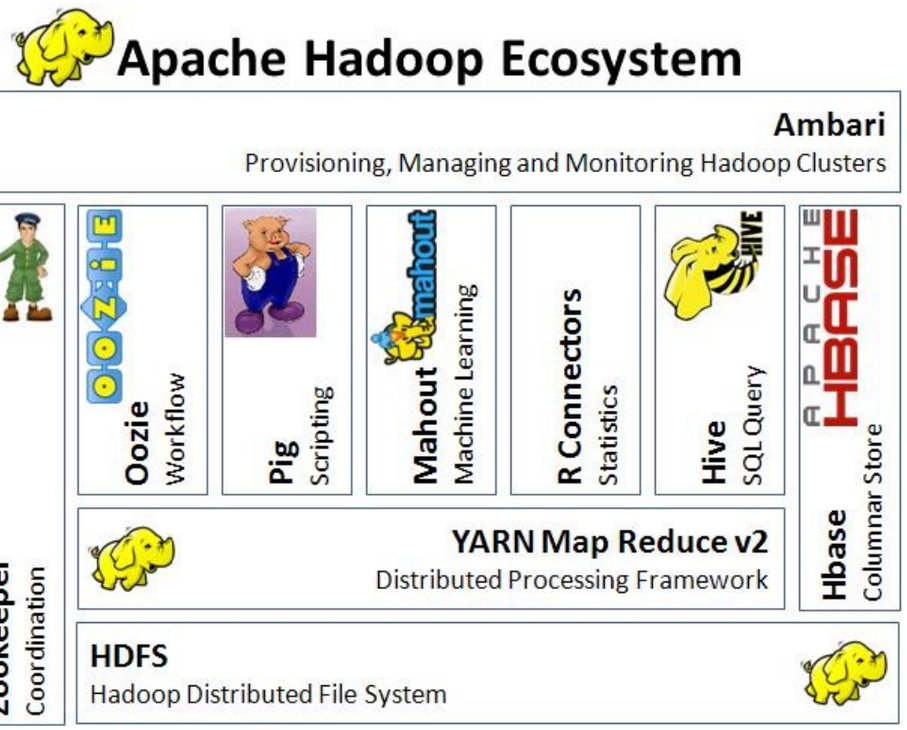


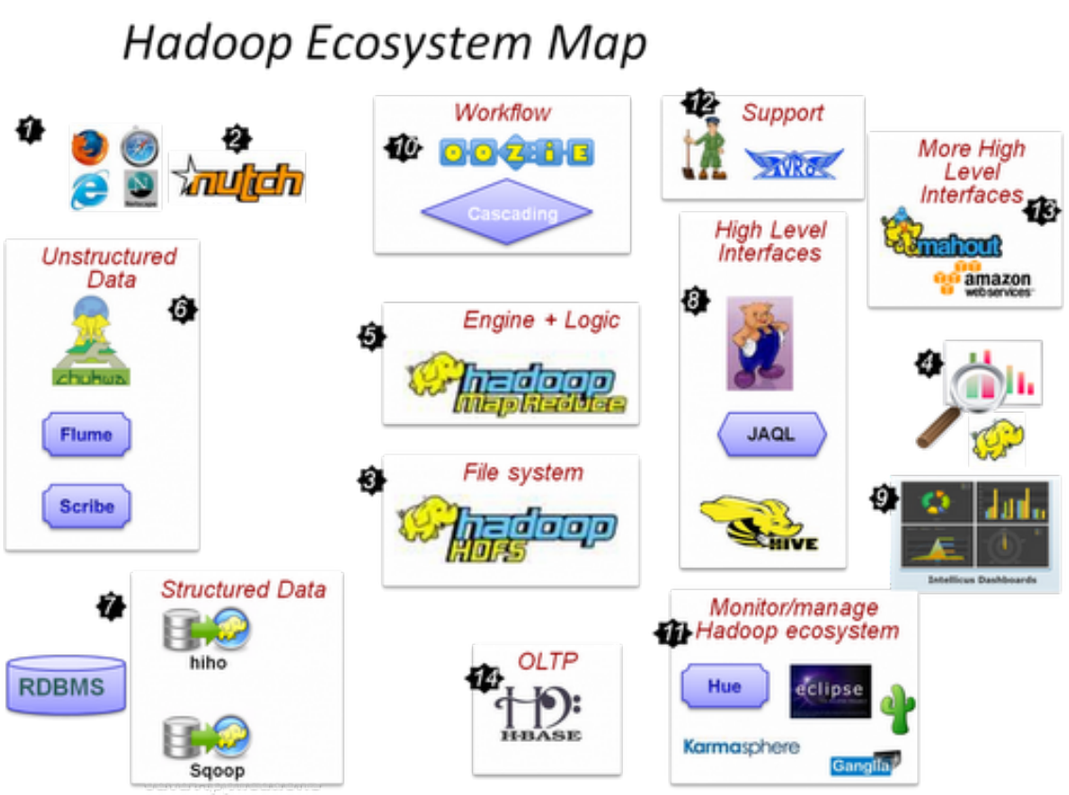
Hadoop é uma plataforma de software, construída em Java, de **computação distribuída** voltada para **clusters** e processamento de **grandes volumes** de dados, com atenção a **tolerância a falhas**. Foi inspirada no **MapReduce** e no **GoogleFS** (GFS).

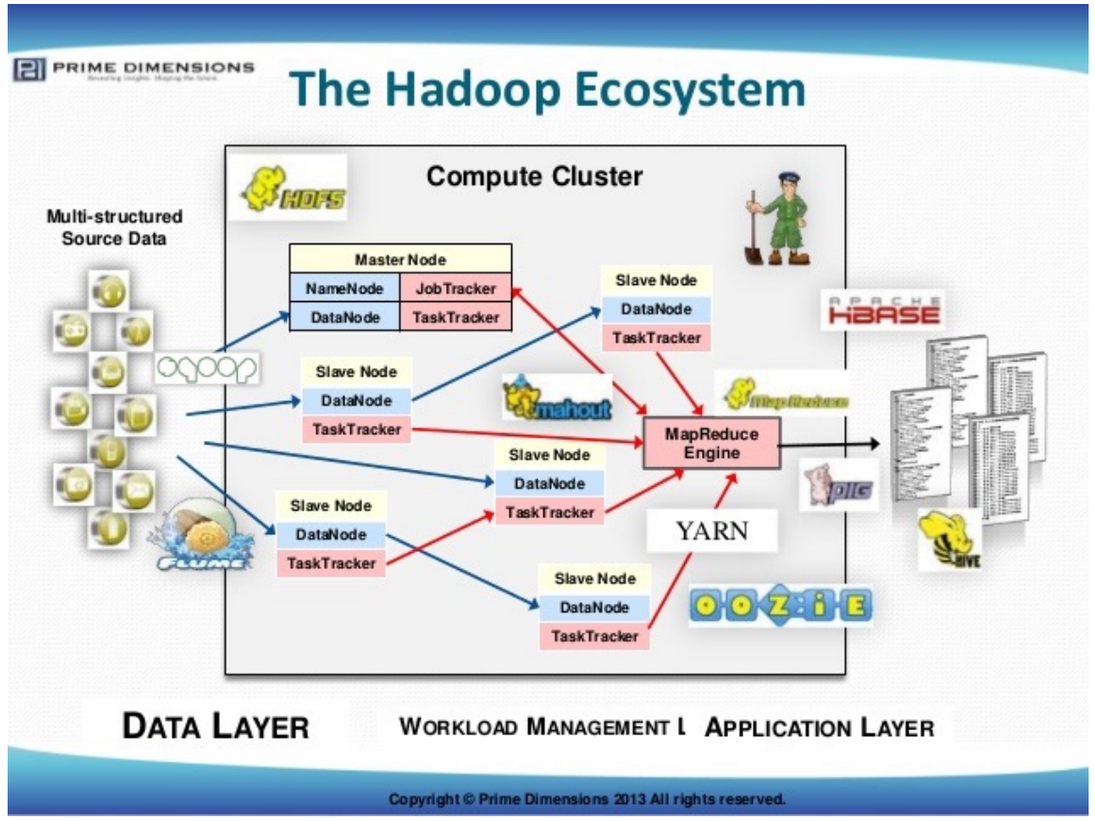
https://pt.wikipedia.org/wiki/Hadoop

HADOOP ECOSYSTEM(??)









HADOOP = GFS + MAP REDUCE

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

ABSTRACT

We have designed and implemented the Google File System, a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients.

While sharing many of the same goals as previous distributed file systems, our design has been driven by observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system assumptions. This has led us to reexamine traditional choices and explore radically different design points.

The file system has successfully met our storage needs. It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets. The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients.

In this paper, we present file system interface extensions designed to support distributed applications, discuss many aspects of our design, and report measurements from both micro-benchmarks and real world use.

Categories and Subject Descriptors

1. INTRODUCTION

We have designed and implemented the Google File System (GFS) to meet the rapidly growing demands of Google's data processing needs. GFS shares many of the same goals as previous distributed file systems such as performance, scalability, reliability, and availability. However, its design has been driven by key observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system design assumptions. We have reexamined traditional choices and explored radically different points in the design space.

First, component failures are the norm rather than the exception. The file system consists of hundreds or even thousands of storage machines built from inexpensive commodity parts and is accessed by a comparable number of client machines. The quantity and quality of the components virtually guarantee that some are not functional at any given time and some will not recover from their current failures. We have seen problems caused by application bugs, operating system bugs, human errors, and the failures of disks, memory, connectors, networking, and power supplies. Therefore, constant monitoring, error detection, fault tolerance, and automatic recovery must be integral to the system.

Second, files are huge by traditional standards. Multi-GB files are common. Each file typically contains many applica-

https://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of MapReduce programs have been implemented and upwards of one thousand MapReduce jobs are executed on Google's clusters every day.

1 Introduction

Over the past five years, the authors and many others at Google have implemented hundreds of special-purpose computations that process large amounts of raw data, such as crawled documents, web request logs, etc., to compute various kinds of derived data, such as inverted indices, various representations of the graph structure of web documents, summaries of the number of pages crawled per host, the set of most frequent queries in a given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

As a reaction to this complexity, we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. Our abstraction is inspired by the *map* and *reduce* primitives present in Lisp and many other functional languages. We realized that most of our computations involved applying a *map* operation to each logical "record" in our input in order to compute a set of intermediate key/value pairs, and then applying a *reduce* operation to all the values that shared the same key, in order to combine the derived data appropriately. Our use of a functional model with user-specified map and reduce operations allows us to parallelize large computations easily and to use re-execution as the primary mechanism for fault tolerance.

The major contributions of this work are a simple and powerful interface that enables automatic parallelization and distribution of large-scale computations, combined with an implementation of this interface that achieves high performance on large clusters of commodity PCs.

Section 2 describes the basic programming model and gives several examples. Section 3 describes an implementation of the MapReduce interface tailored towards our cluster-based computing environment. Section 4 describes several refinements of the programming model that we have found useful. Section 5 has performance measurements of our implementation for a variety of tasks. Section 6 explores the use of MapReduce within Google including our experiences in using it as the basis

https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

HADOOP 1.0 = GFS + MAP REDUCE

Hadoop v1.0

MapReduce
Data Processing
& Resource Management

HDFS
Distributed File Storage

Obs: somente processamento em *Batch*

Hadoop v2.0

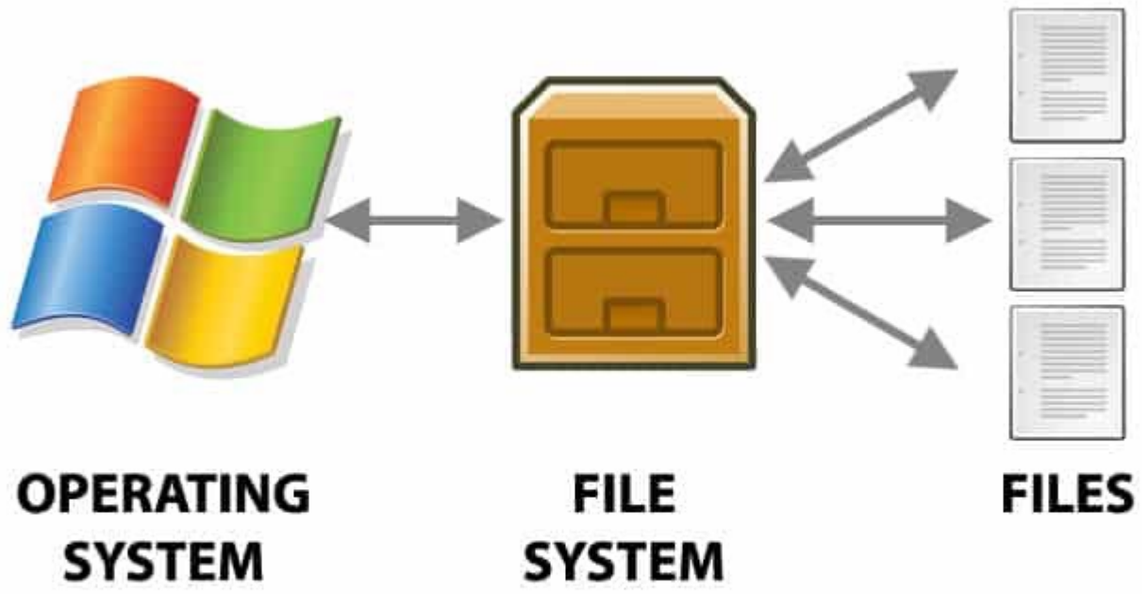
MapReduce

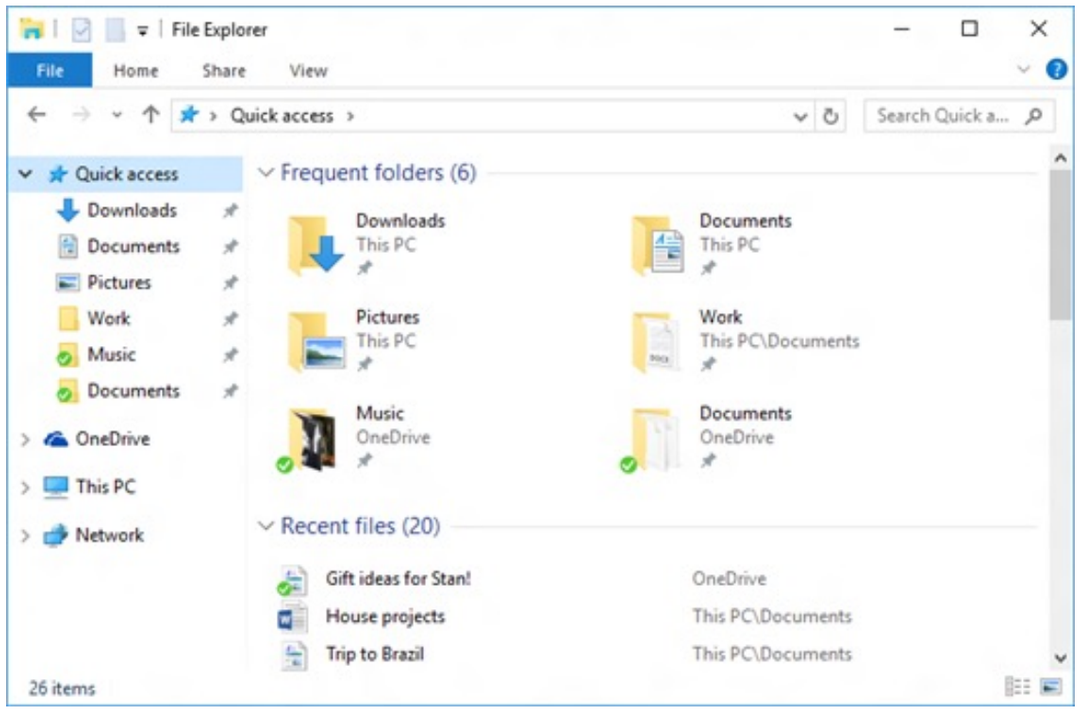
Other Data Processing Frameworks

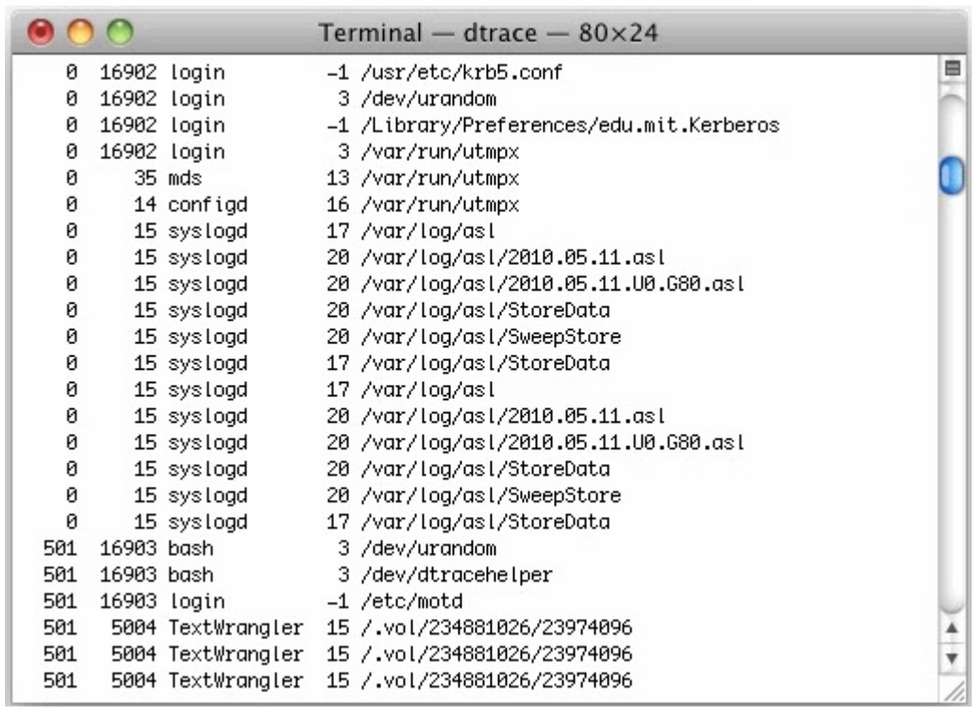
YARN
Resource Management

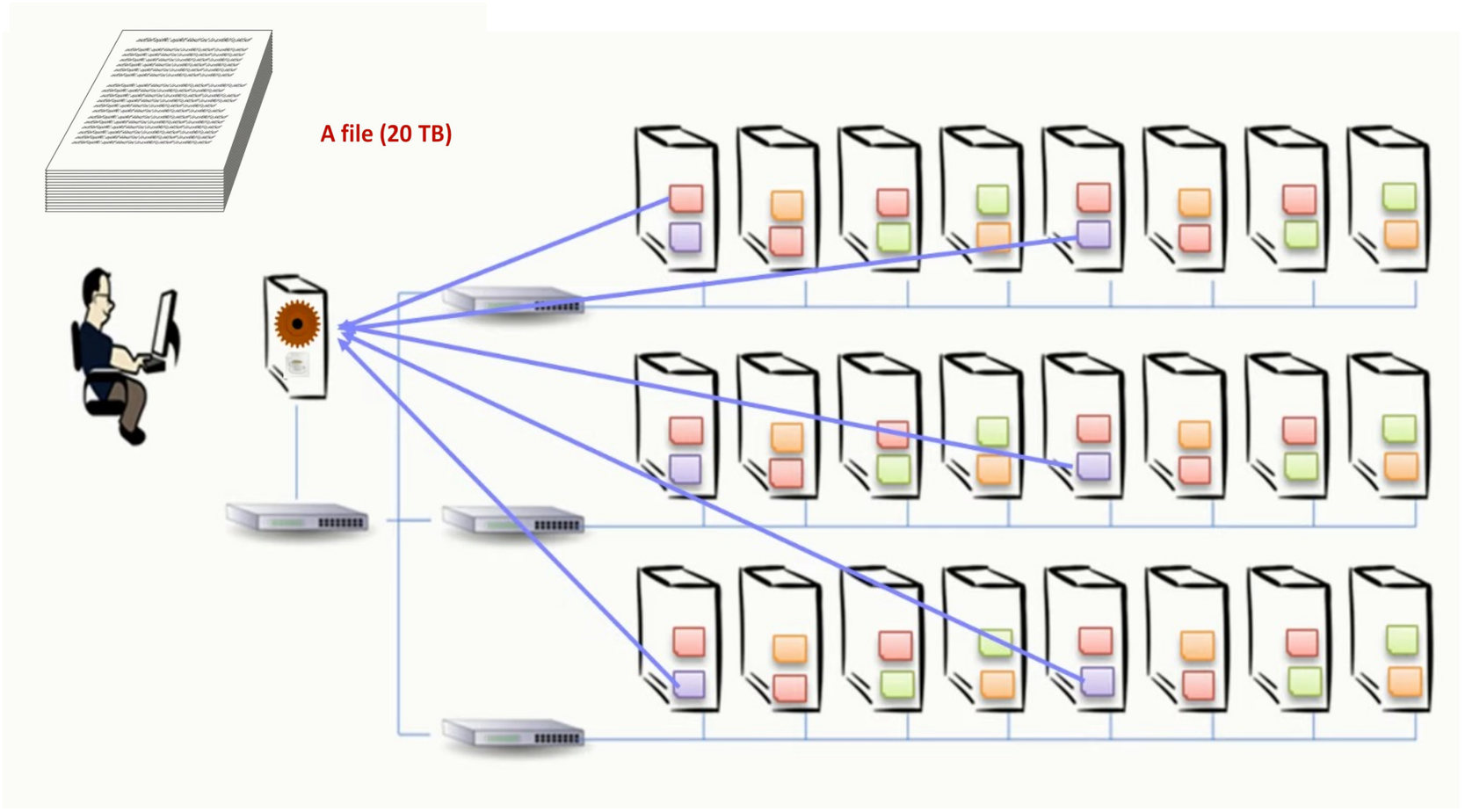
HDFS
Distributed File Storage

https://www.edureka.co/blog/hadoop-yarn-tutorial/


OPERATING SYSTEM
FILE SYSTEM
FILES


File Explorer
File
Home
Share
View
Quick access
Search Quick a...
Quick access
Downloads
Documents
Pictures
Work
Music
Documents
OneDrive
This PC
Network
Frequent folders (6)
Downloads
This PC
Documents
This PC
Pictures
This PC
Work
This PC\Documents
Music
OneDrive
Documents
OneDrive
Recent files (20)
Gift ideas for Stan!
OneDrive
House projects
This PC\Documents
Trip to Brazil
This PC\Documents
26 items

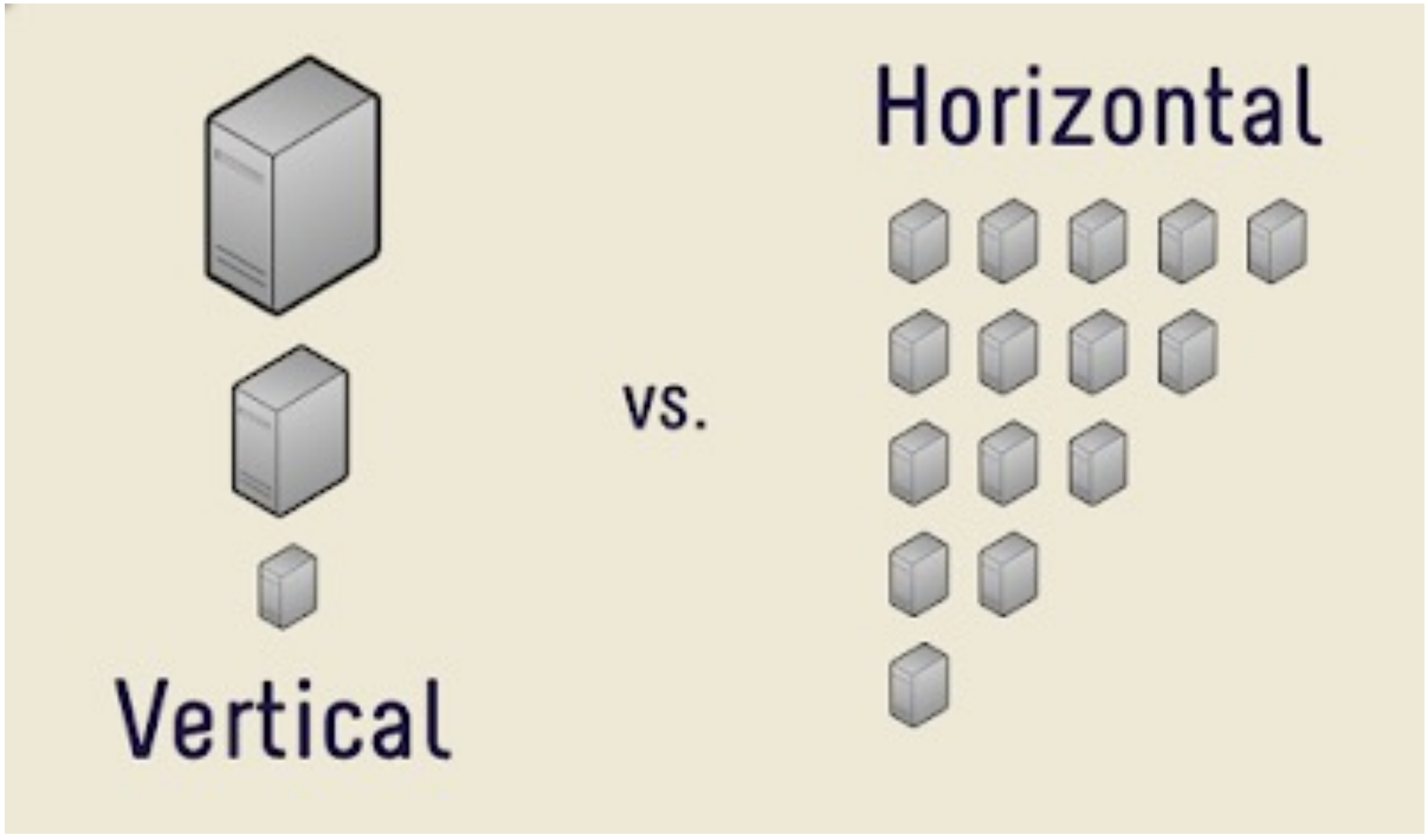

Terminal — dtrace — 80×24
0 16902 login -1 /usr/etc/krb5.conf
0 16902 login 3 /dev/urandom
0 16902 login -1 /Library/Preferences/edu.mit.Kerberos
0 16902 login 3 /var/run/utmpx
0 35 mds 13 /var/run/utmpx
0 14 configd 16 /var/run/utmpx
0 15 syslogd 17 /var/log/asl
0 15 syslogd 20 /var/log/asl/2010.05.11.asl
0 15 syslogd 20 /var/log/asl/2010.05.11.U0.G80.asl
0 15 syslogd 20 /var/log/asl/StoreData
0 15 syslogd 20 /var/log/asl/SweepStore
0 15 syslogd 17 /var/log/asl/StoreData
0 15 syslogd 17 /var/log/asl
0 15 syslogd 20 /var/log/asl/2010.05.11.asl
0 15 syslogd 20 /var/log/asl/2010.05.11.U0.G80.asl
0 15 syslogd 20 /var/log/asl/StoreData
0 15 syslogd 20 /var/log/asl/SweepStore
0 15 syslogd 17 /var/log/asl/StoreData
501 16903 bash 3 /dev/urandom
501 16903 bash 3 /dev/dtracehelper
501 16903 login -1 /etc/motd
501 5004 TextWrangler 15 /.vol/234881026/23974096
501 5004 TextWrangler 15 /.vol/234881026/23974096
501 5004 TextWrangler 15 /.vol/234881026/23974096



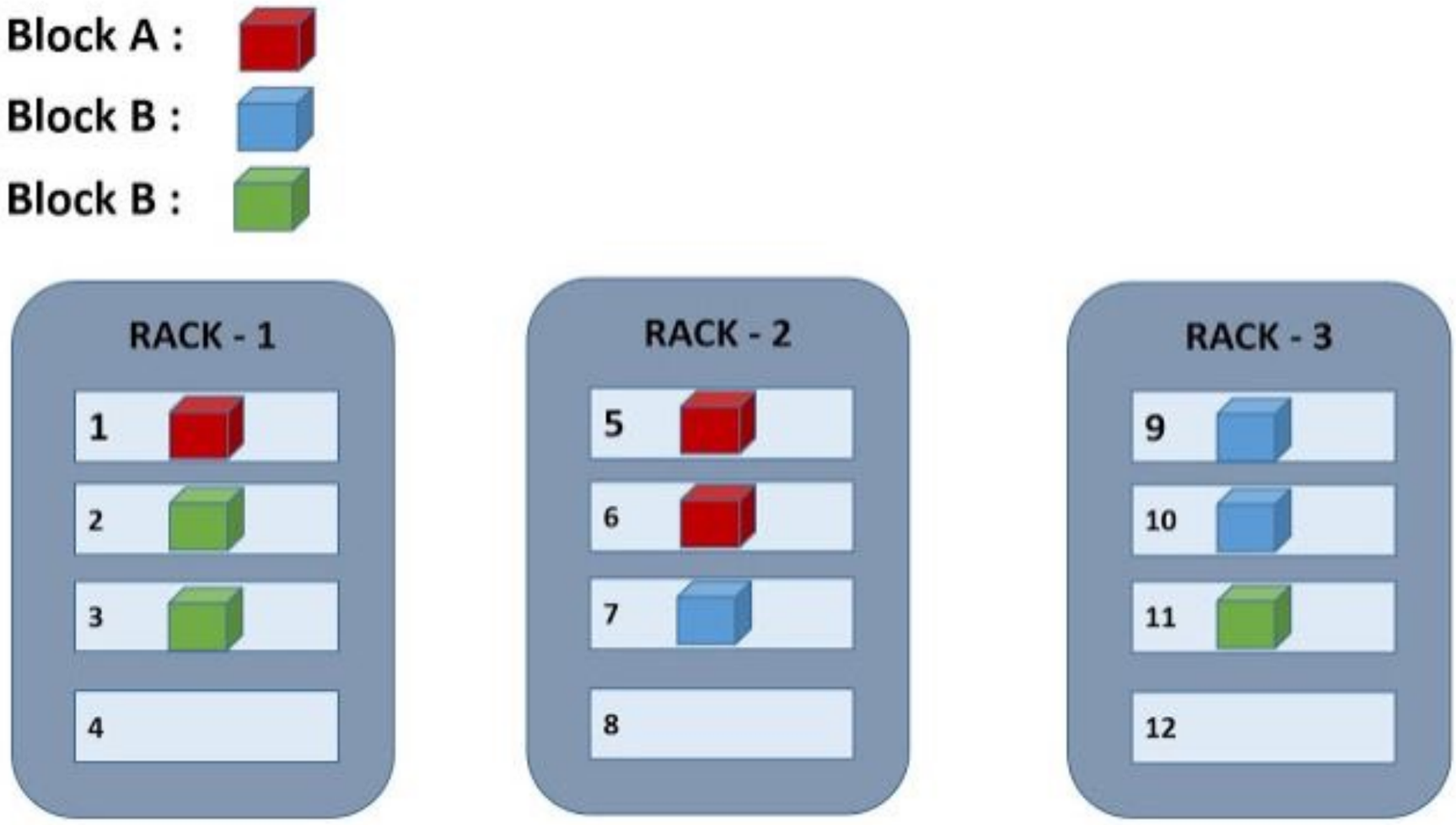
HDFS - Features

1. Distributed
2. Scalable
3. Cost Effective
4. Fault Tolerant
5. High Throughput

O **HDFS** implementa uma maneira **Distribuída** de armazenar arquivos muito grandes. Contempla também uma **tolerância a falhas** com a distribuição de arquivos em diferentes nós da rede. As máquinas podem ser computadores comuns, não precisam ser servidores.

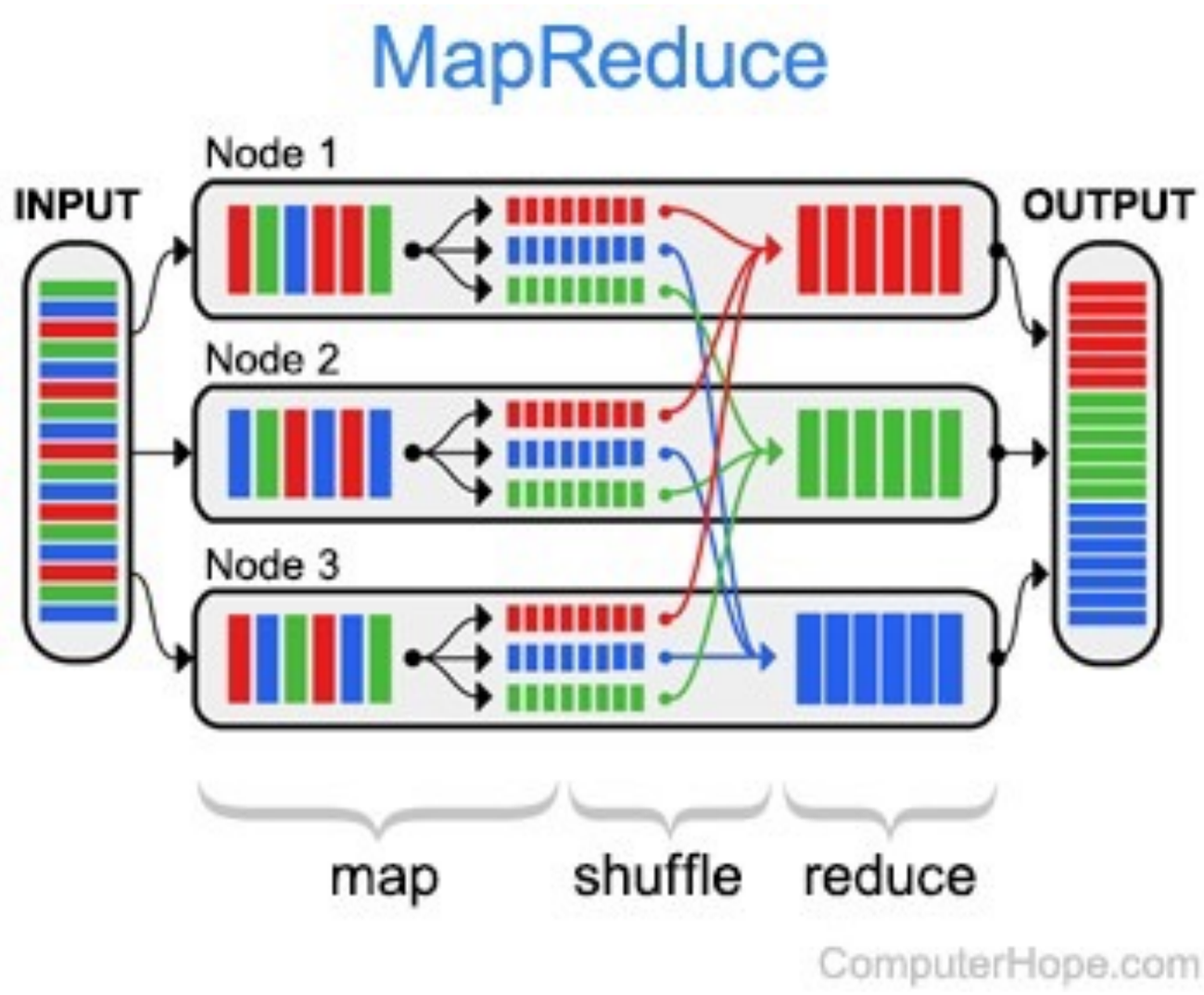


http://pudgylogic.blogspot.com/2016/01/horizontal-vs-vertical-scaling.html



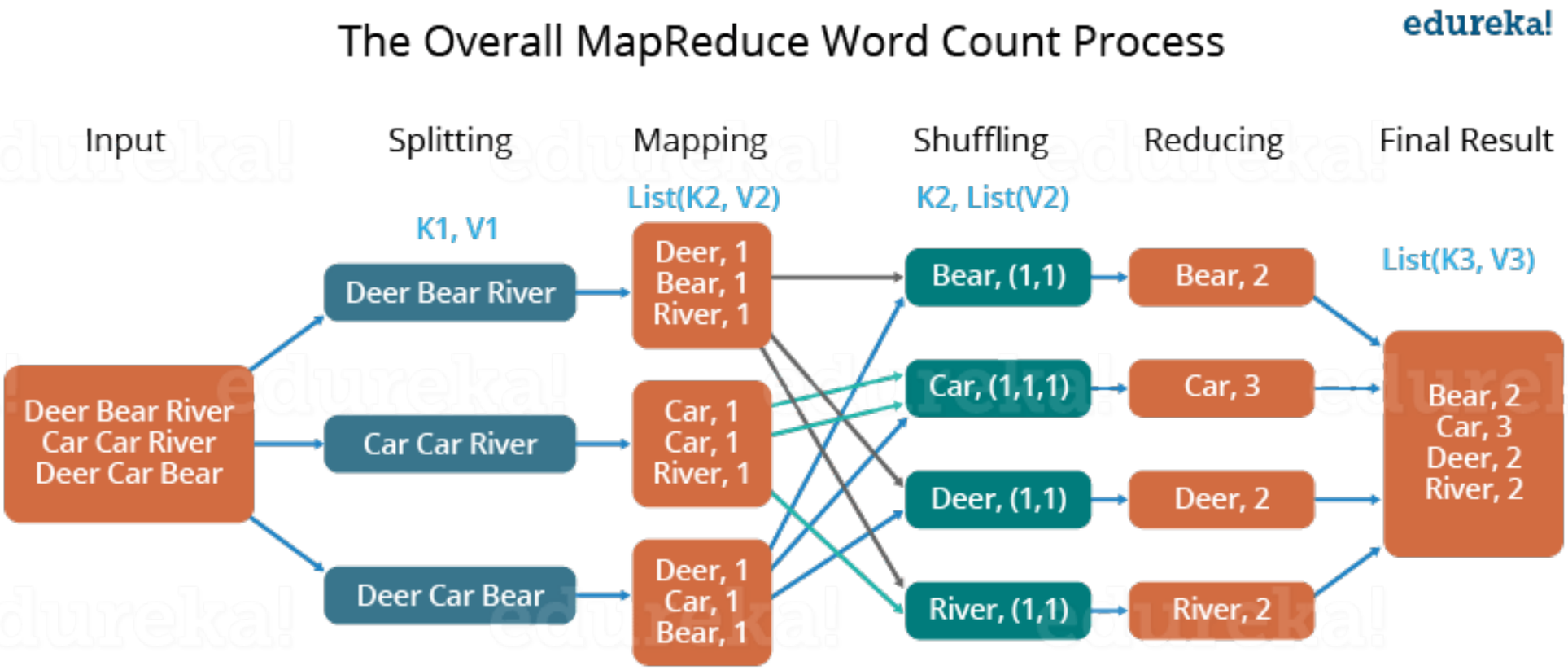
Cada bloco é replicado normalmente em 3 nós distintos. Também é possível indicar que se faça a distribuição dos dados em diferentes racks para aumentar a tolerância a falhas e disponibilidade da aplicação.

Importante lembrar que o HDFS é um **Sistema de Arquivos** – não é uma Base de dados.



https://www.computerhope.com/jargon/m/mapreduce.htm

MapReduce é um modelo de programação desenhado para processar grandes volumes de dados em paralelo, dividindo o trabalho em um conjunto de tarefas independentes

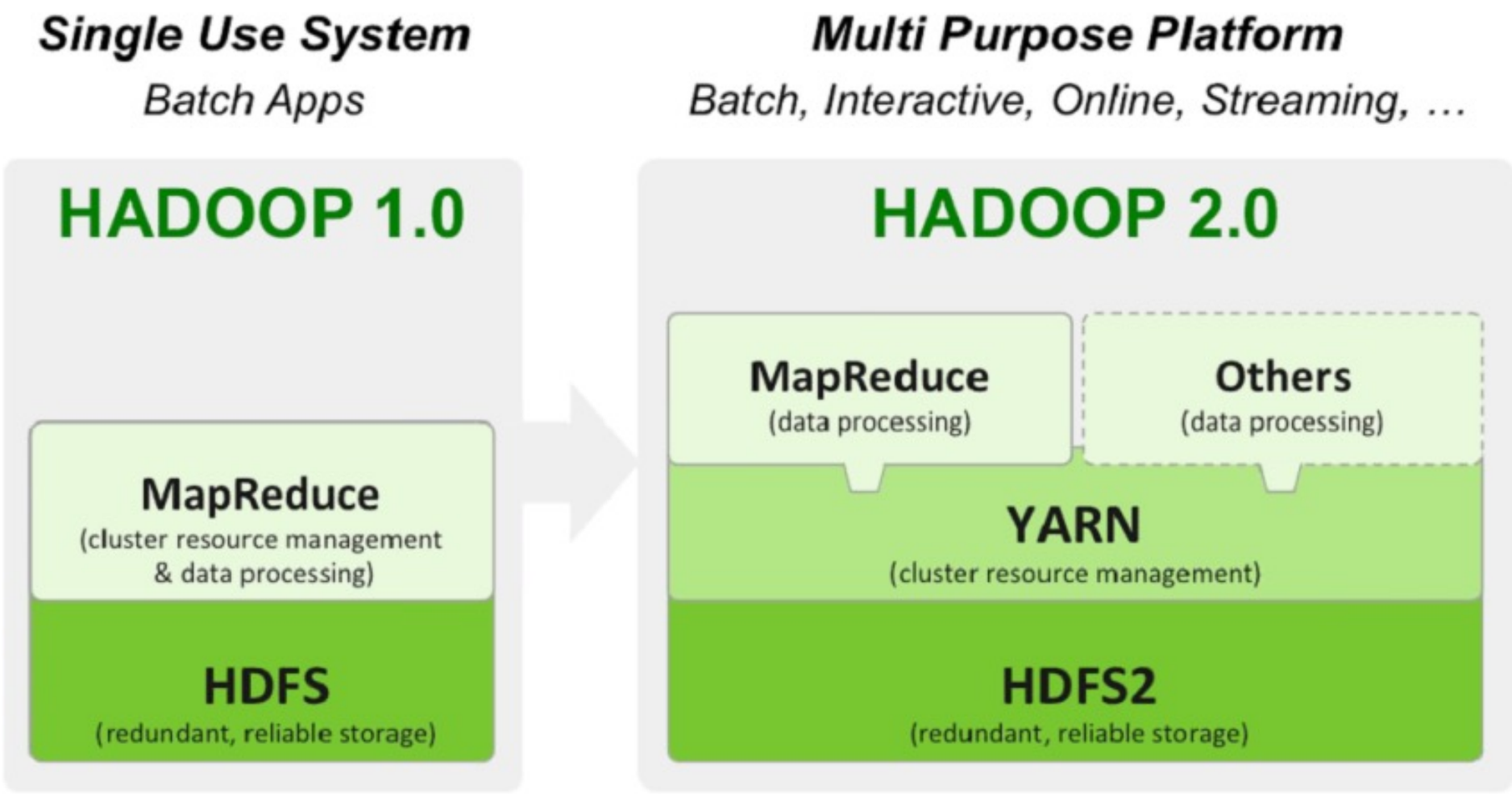


https://www.edureka.co/blog/mapreduce-tutorial/

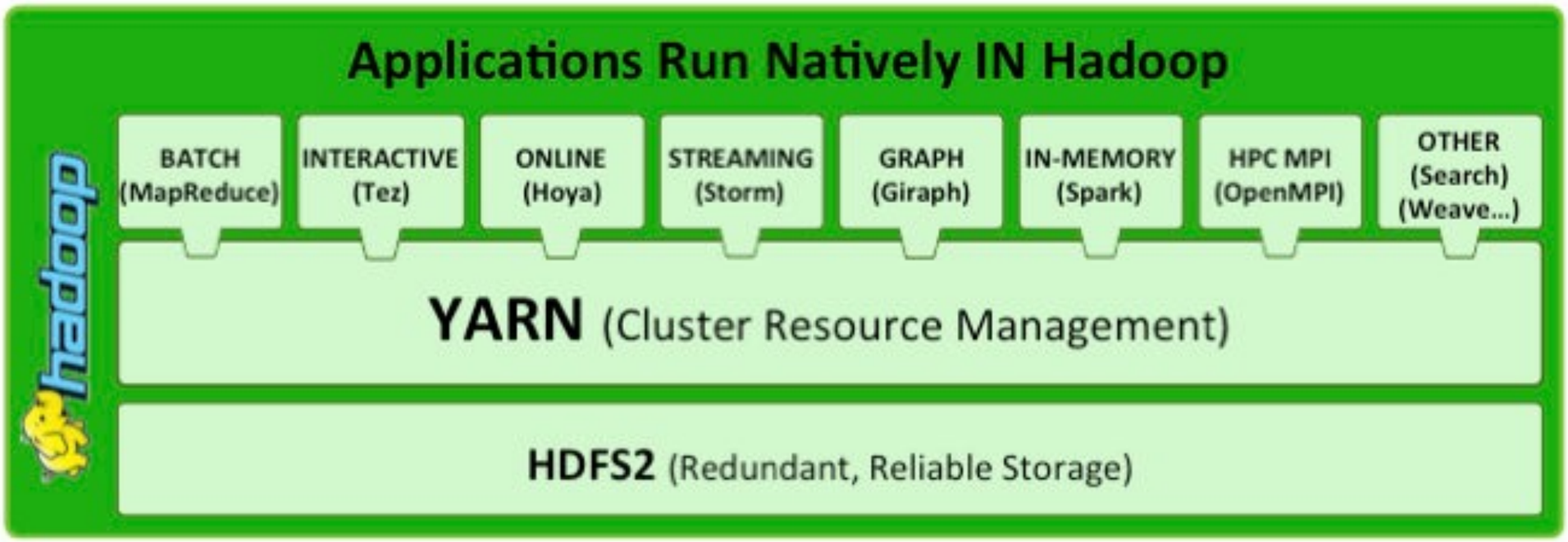
MapReduce é um modelo de programação desenhado para processar grandes volumes de dados em paralelo, dividindo o trabalho em um conjunto de tarefas independentes

https://youtu.be/QSzJSPtmsO8

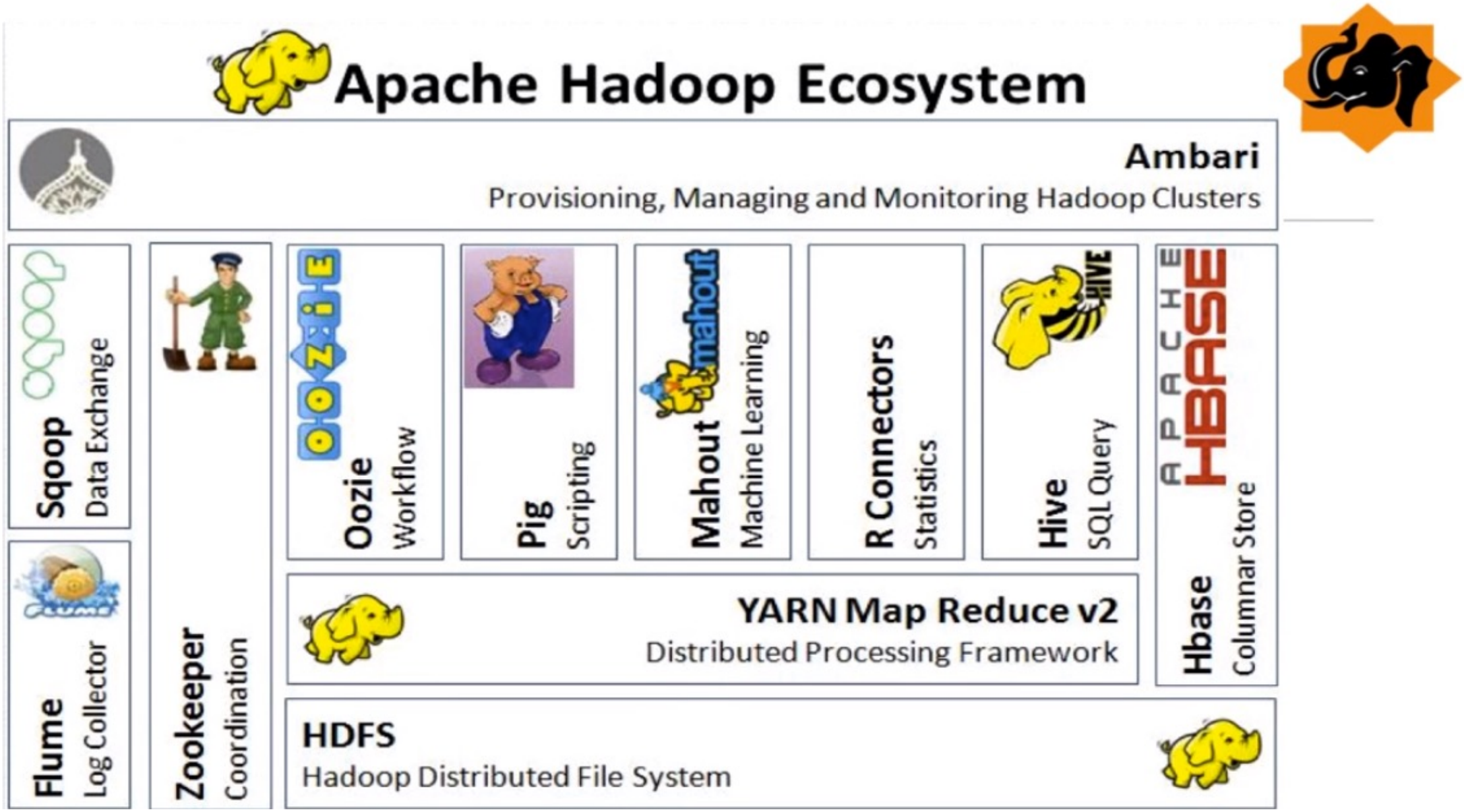
Lembrando que cada nó faz a execução de sua parte em paralelo, permitindo o tratamento de grandes volumes de dados.



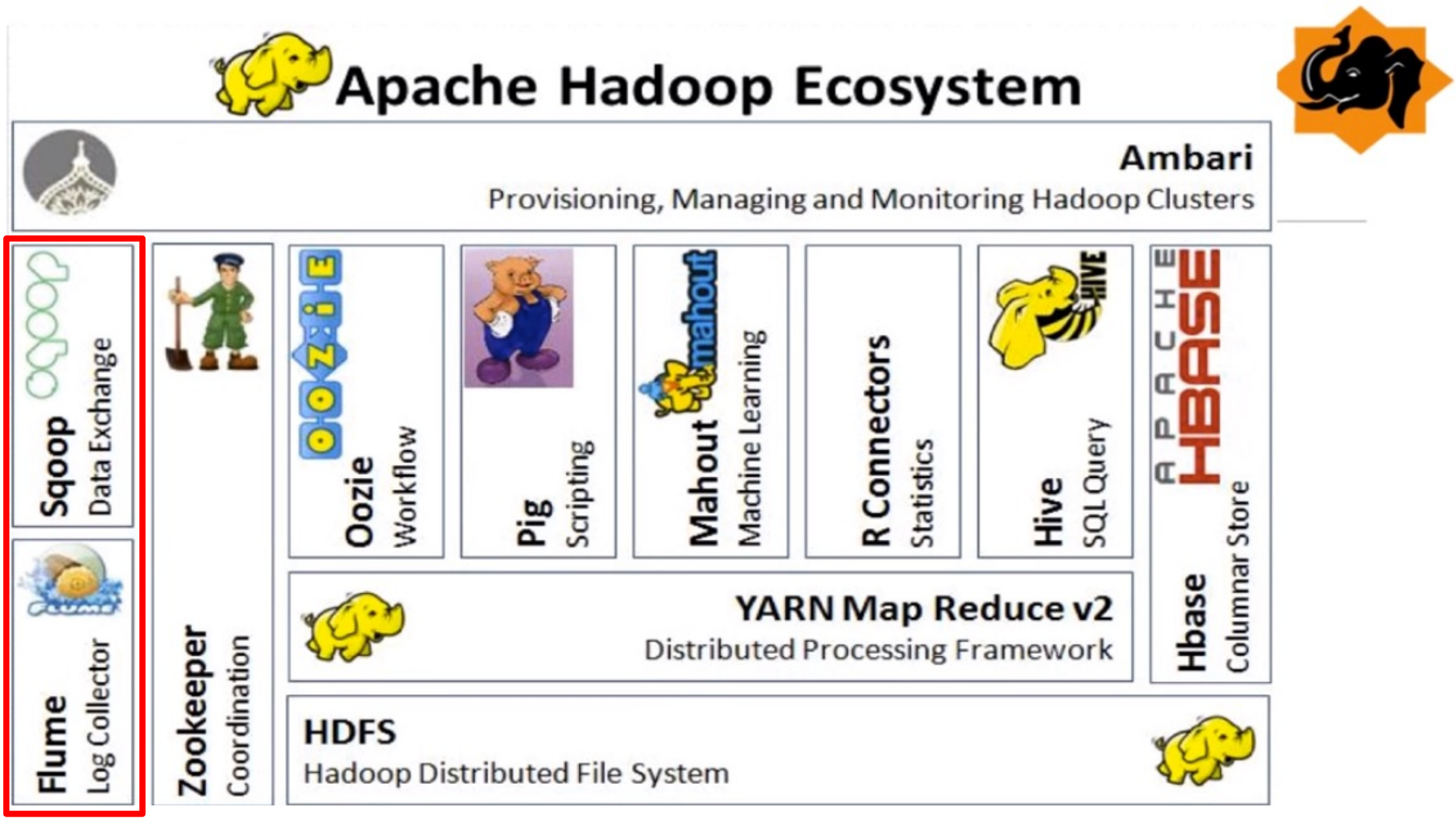
No **Hadoop 2.0** foi introduzido o **YARN** que faz refinou o gerenciamento dos nós na rede.



O **YARN** permitiu a fácil incorporação de diversos outros projetos/ferramentas, estendendo as funcionalidades do Hadoop para outros contextos, além do processamento Batch, como processamento em tempo real.

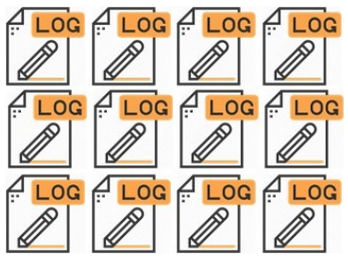


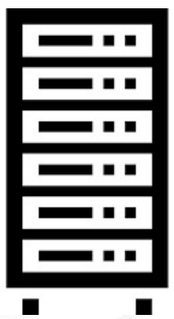
https://youtu.be/1WY63n2XRLM

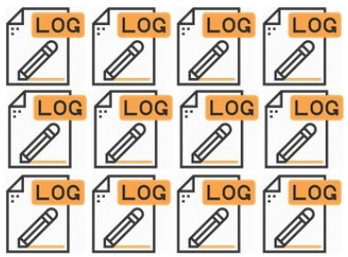


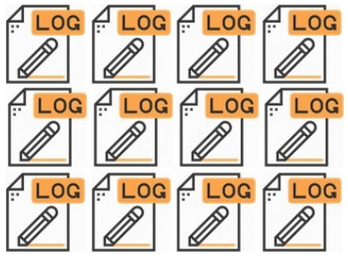
Vamos analisar duas das principais ferramentas para **CARGA DE DADOS** no Hadoop / Big Data. **FLUME** e **SQOOP**.

FLUME – COLETA DE LOGS EM TEMPO REAL

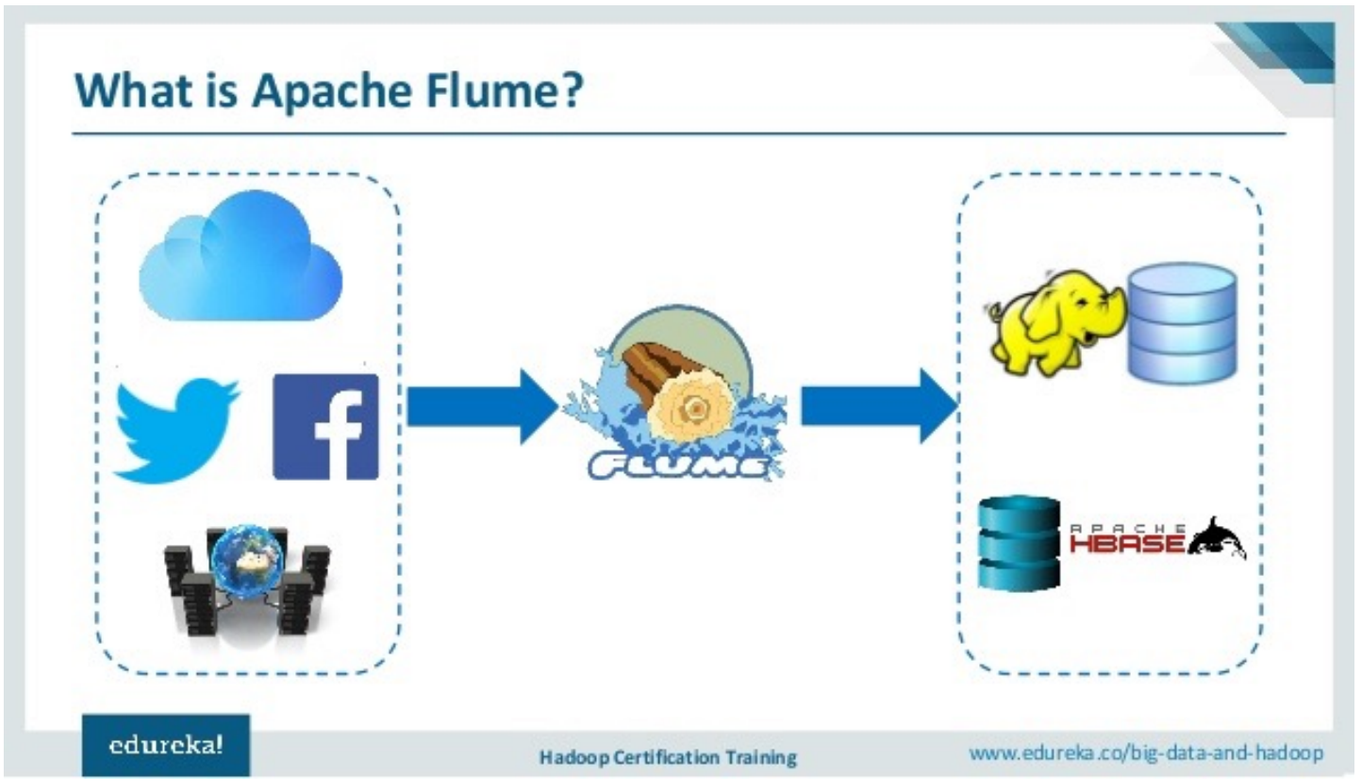








O **FLUME** é utilizado para a carga de **LOGS** de aplicações no Big Data / Hadoop em tempo real.



https://pt.slideshare.net/EdurekaIN/apache-flume-tutorial-twitter-data-streaming-using-flume-hadoop-training-edureka

Os dados podem ser estruturados ou não estrurados.

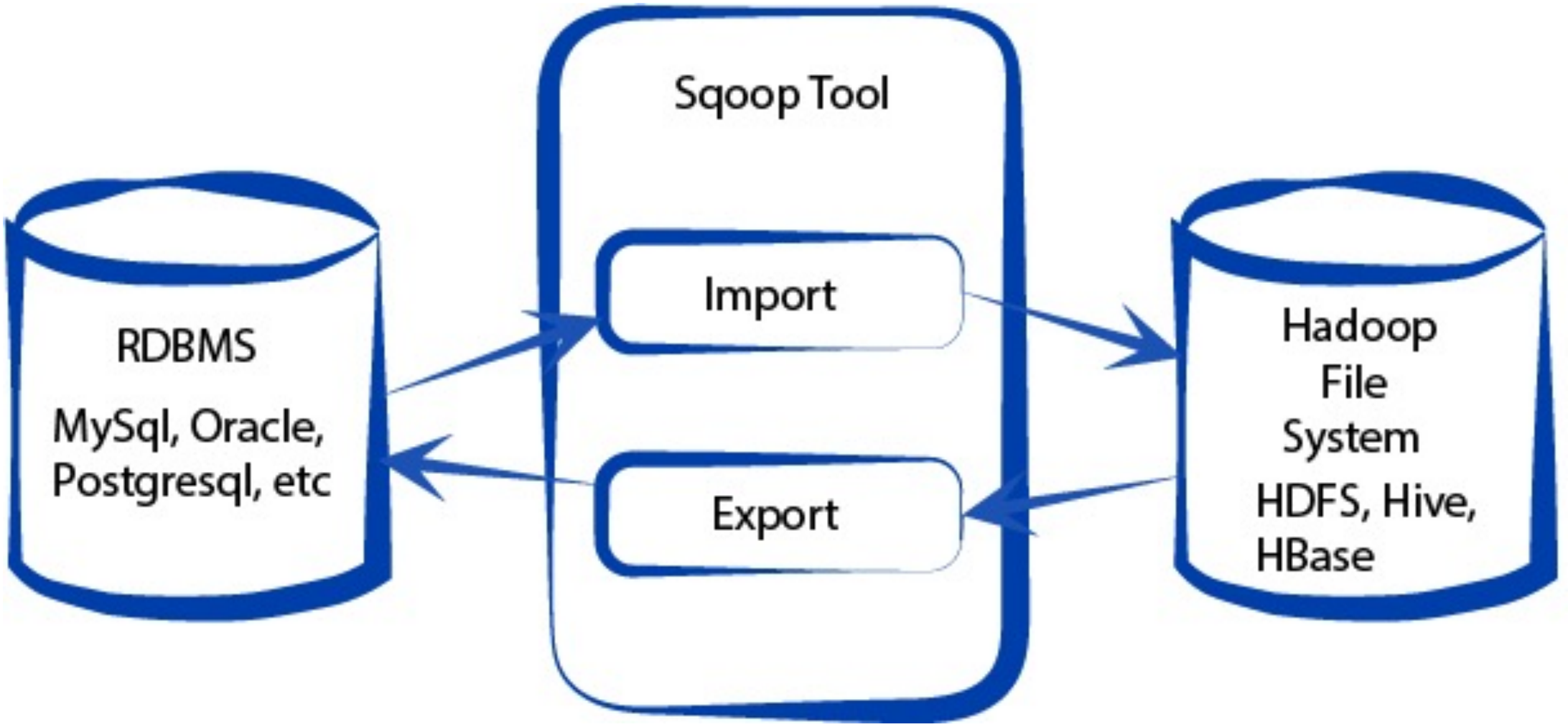
SQOOP = SQL + HADOOP

+

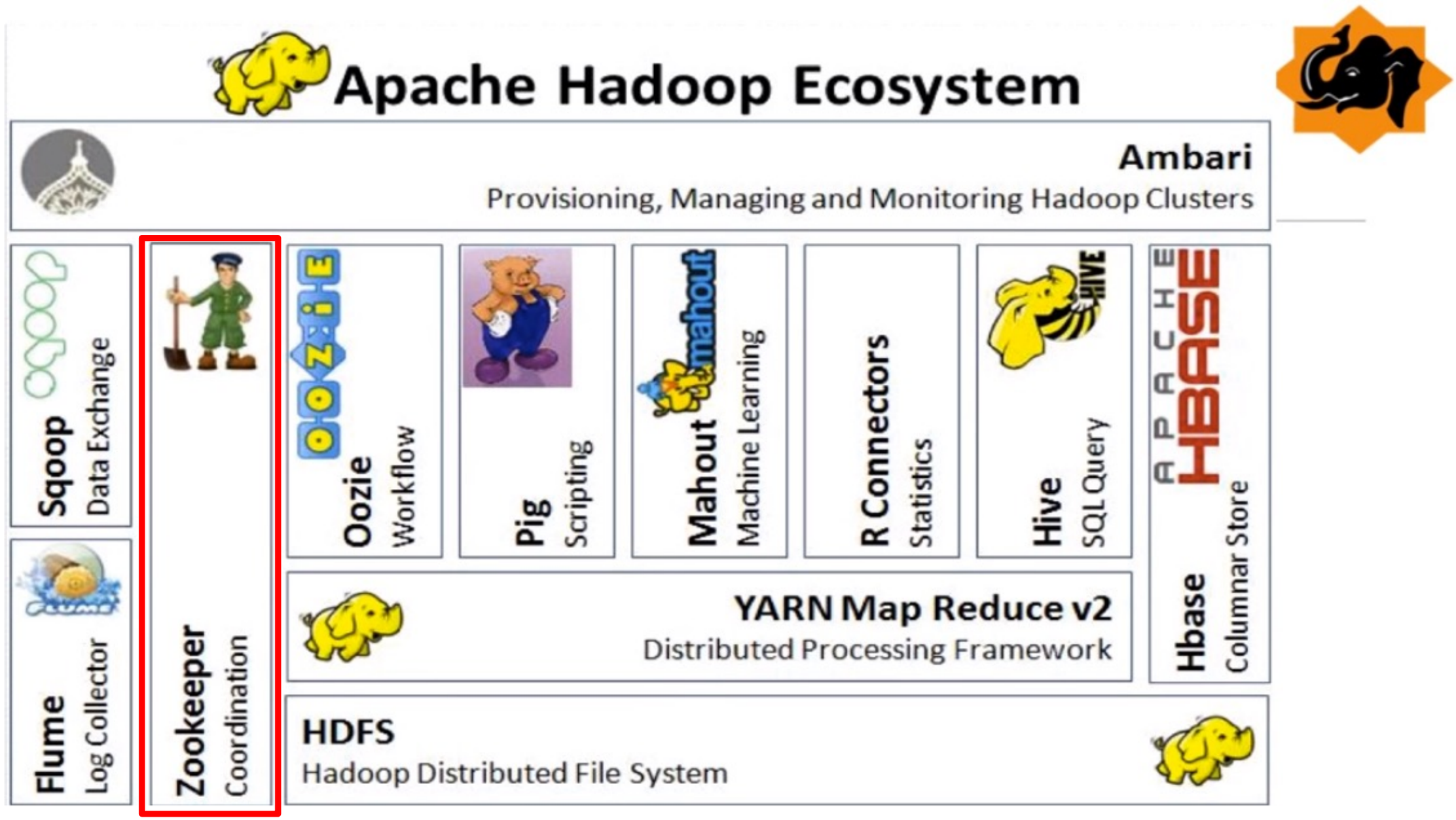
=

SQOOP

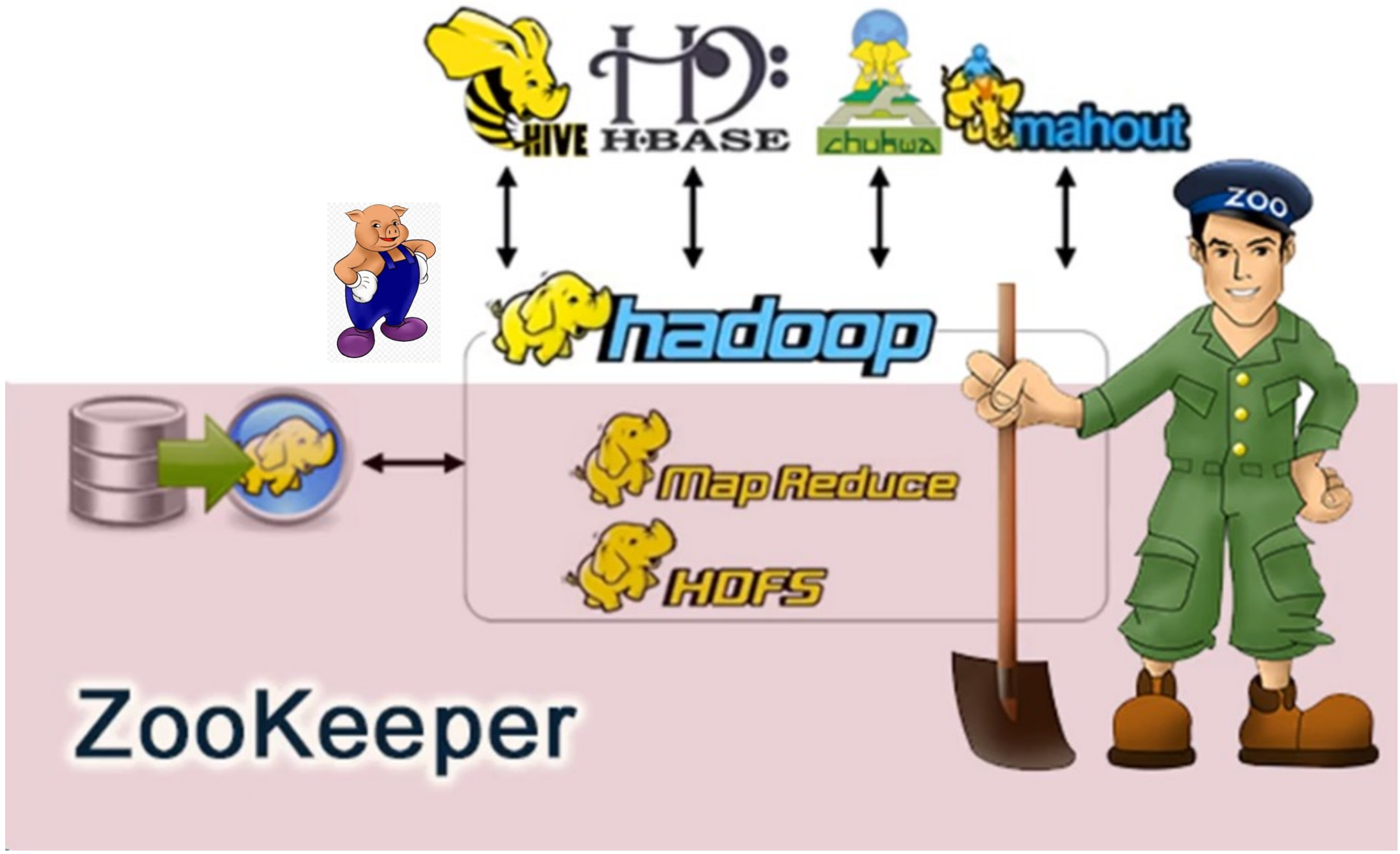
SQOOP = SQL + HADOOP

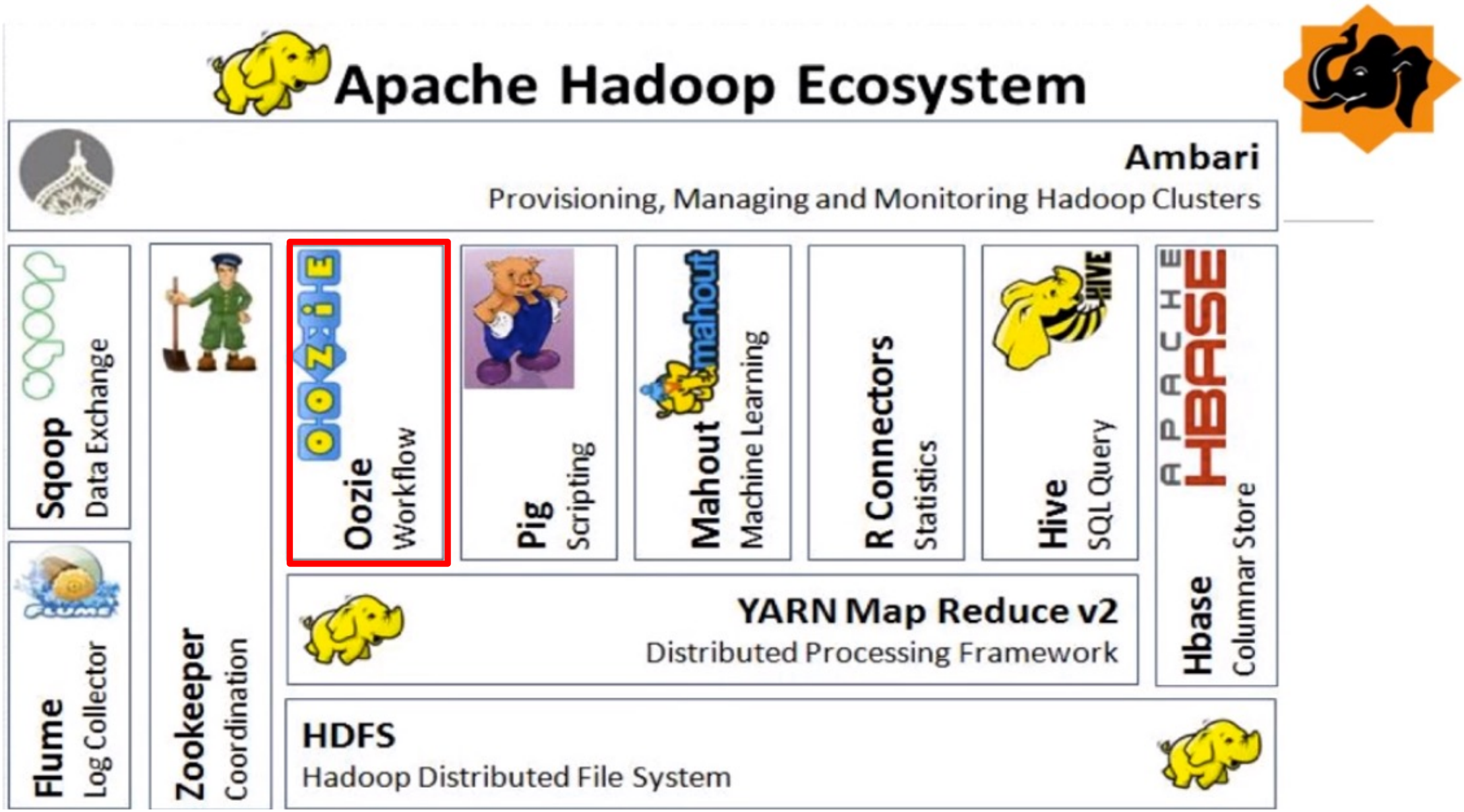


https://www.codigofluente.com.br/aula-13-apache-sqoop-hadoop-sgbdr/

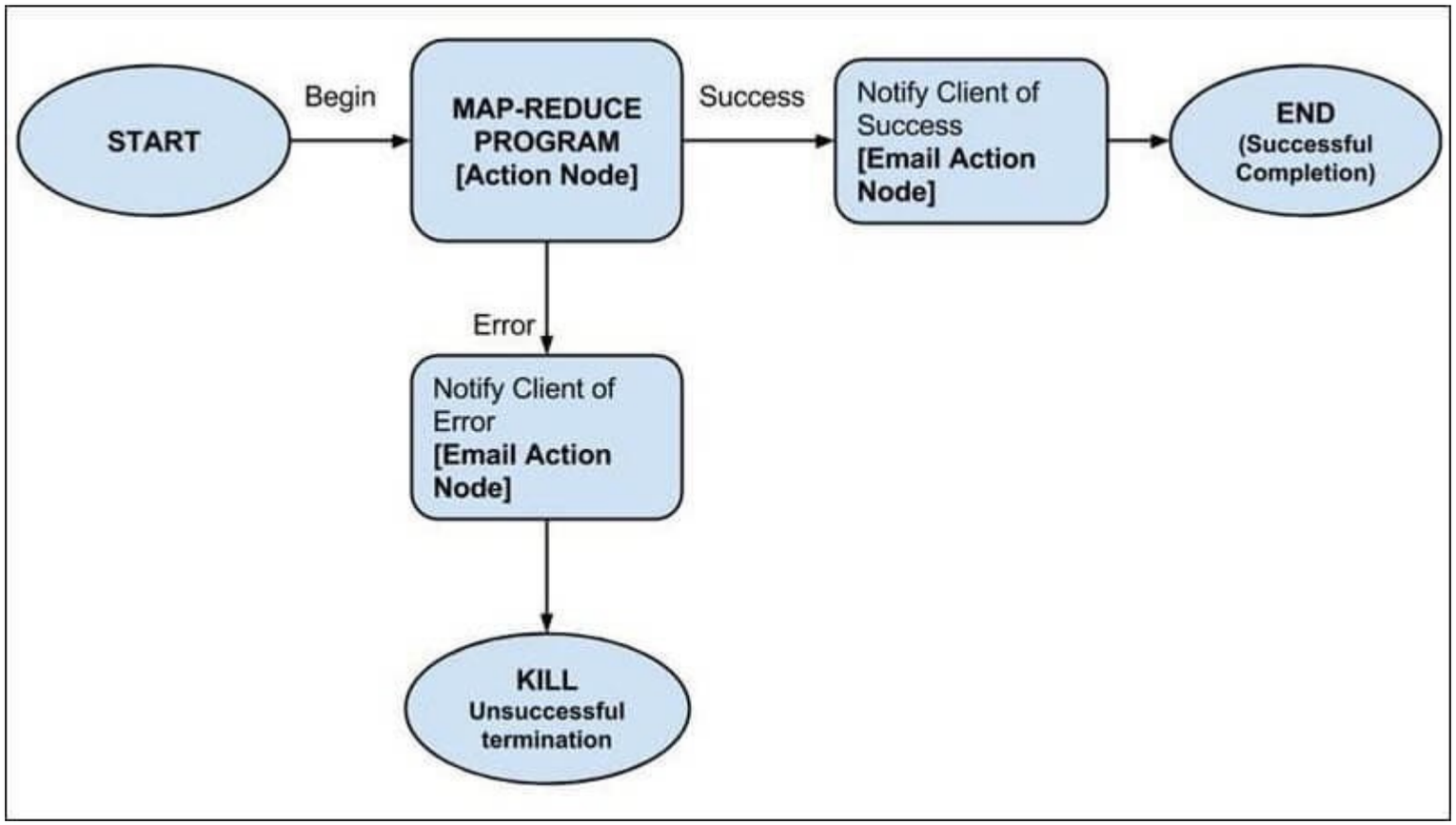


Zookeeper é a camada de coordenação do Hadoop. Permitindo o gerenciamento de servidores e serviços.

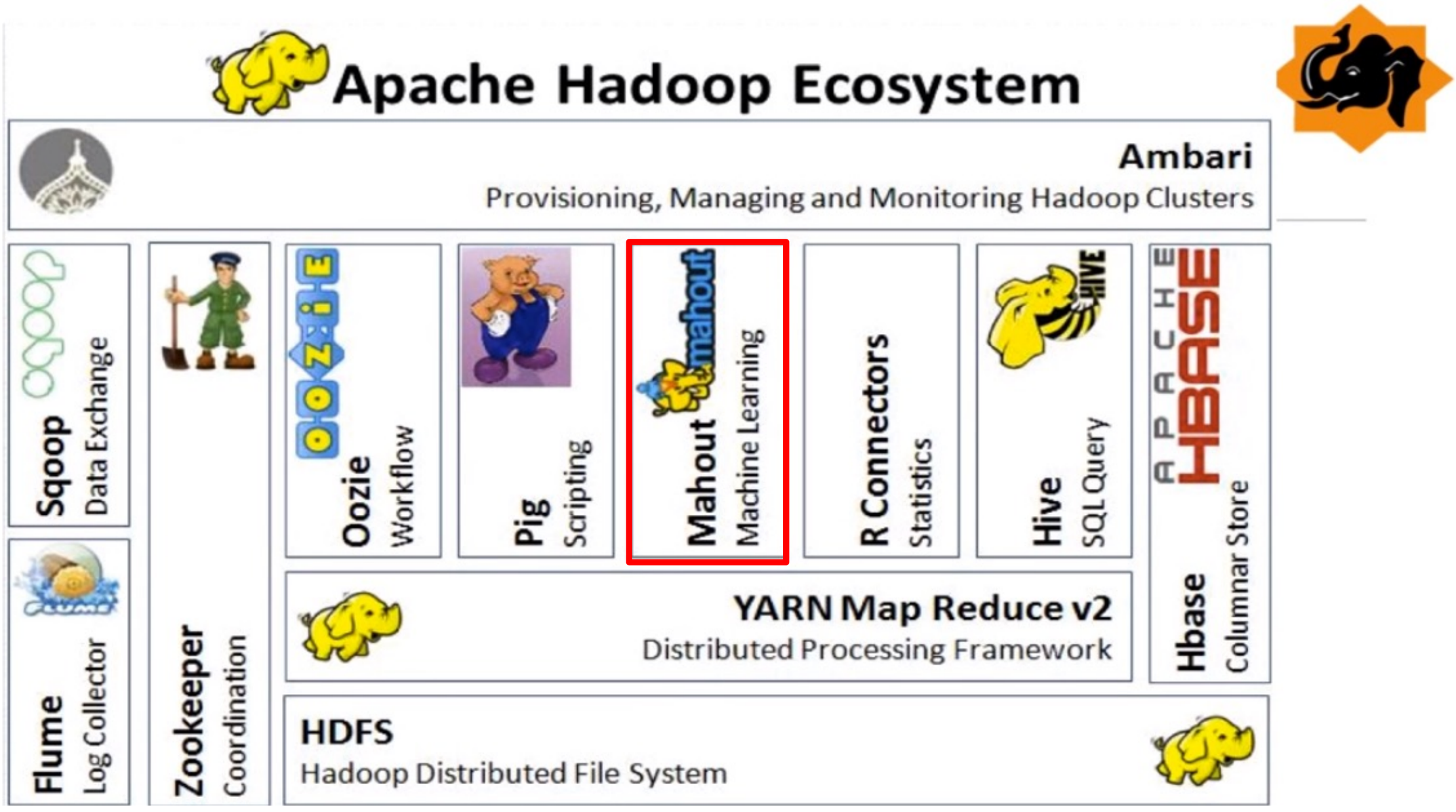

HIVE
HBASE
chukwa
mahout
ZOO
hadoop
Map Reduce
HDFS
ZooKeeper



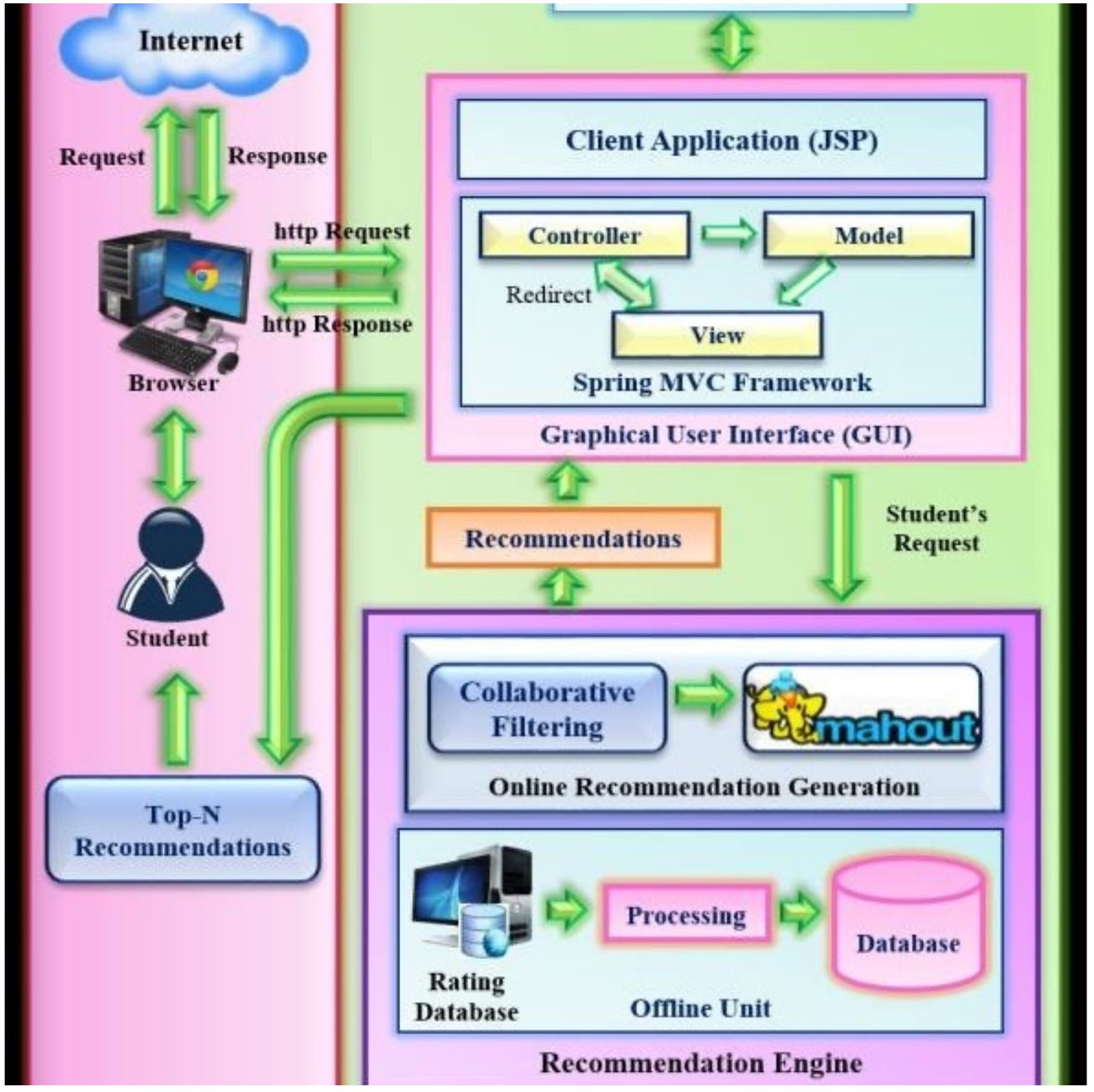
Oozie é uma ferramenta para montar workflows – e programar execuções - de processos no Hadoop.



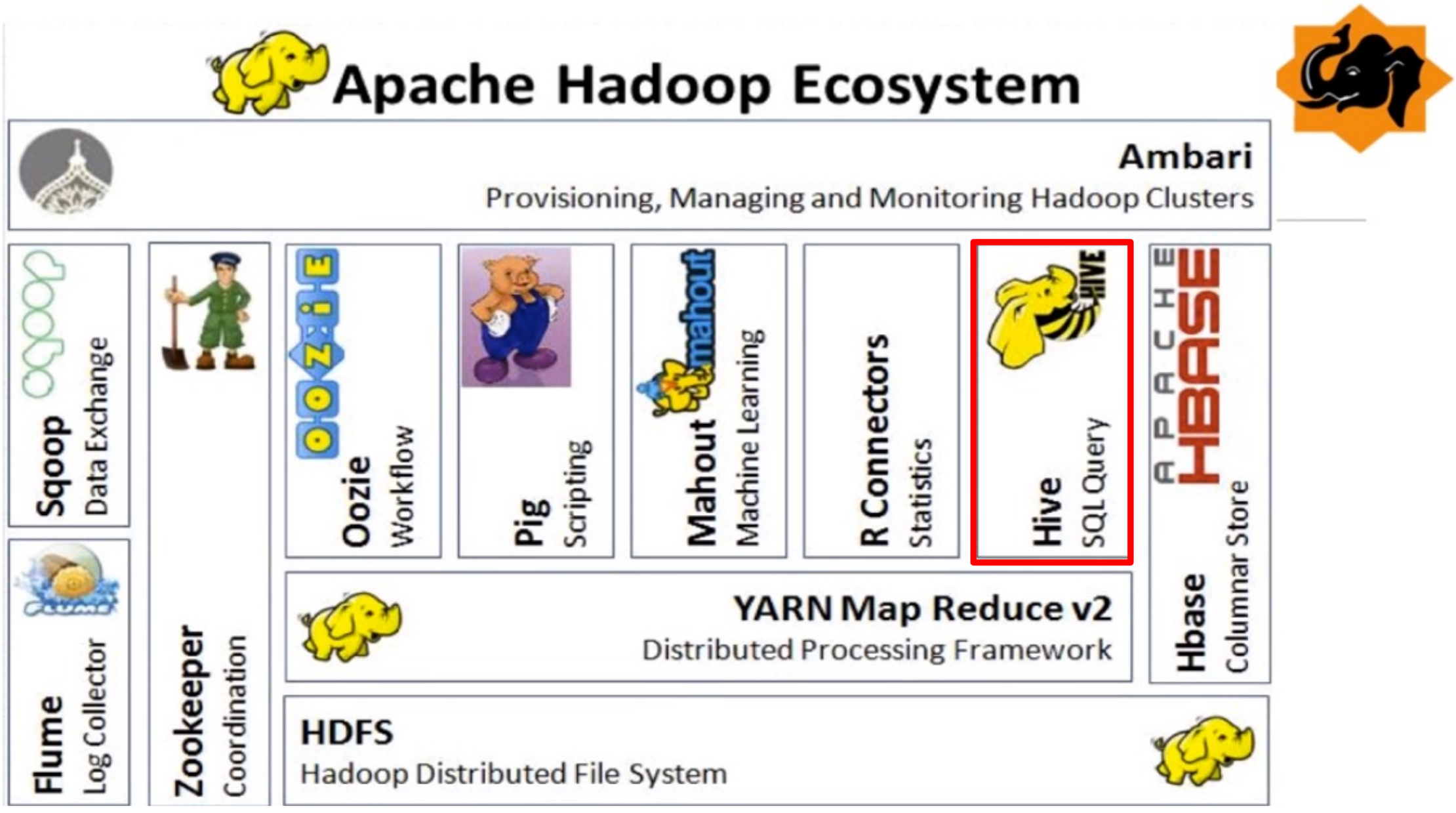
Oozie é uma ferramenta para montar workflows – e programar execuções - de processos no Hadoop.



Mahout é um *framework* de *Machine Learning* que utiliza a linguagem Scala para implementar modelos de ML em sistemas distribuídos.

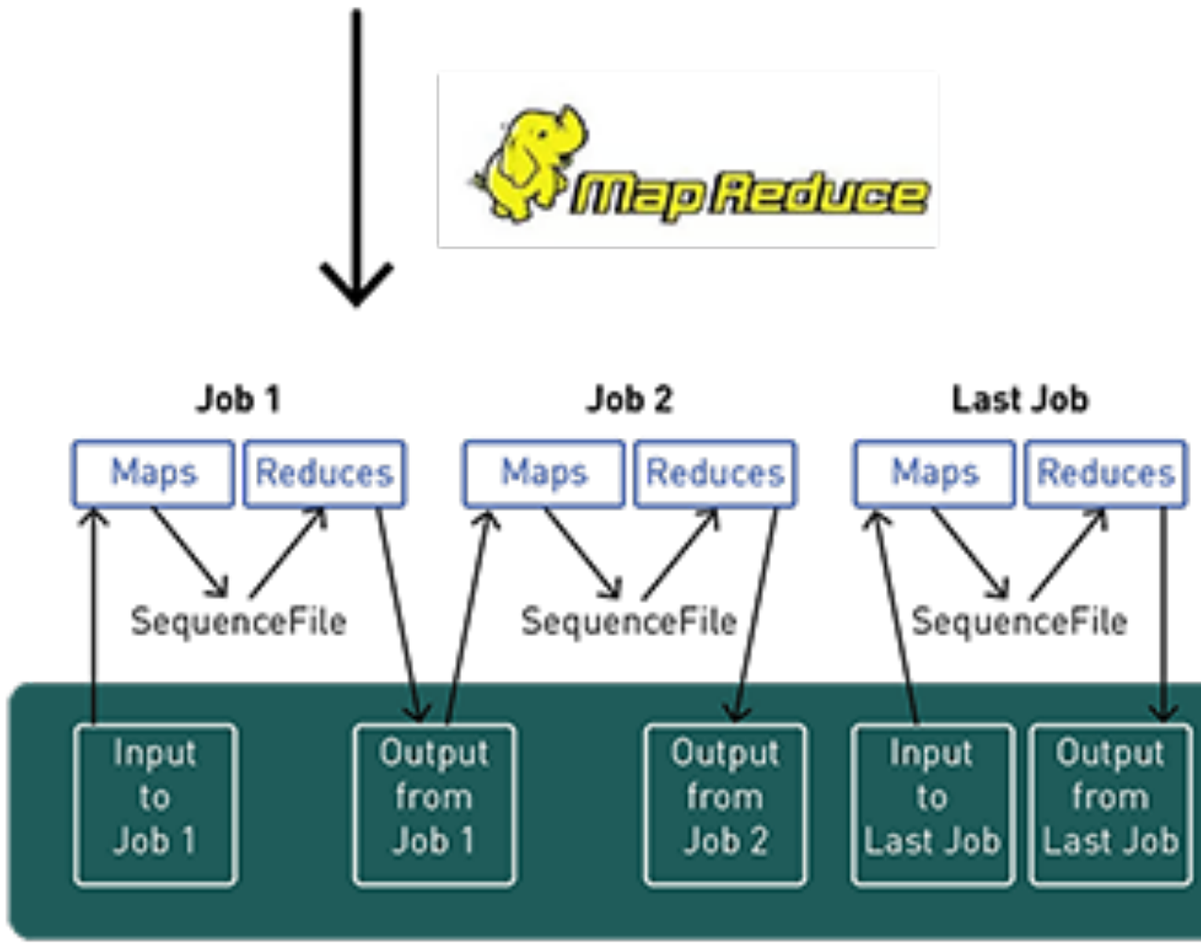


Um caso de uso / funcionalidade utilizada com frequência no Mahout são seus modelos / algoritmos de **Sistemas de Recomendação**.

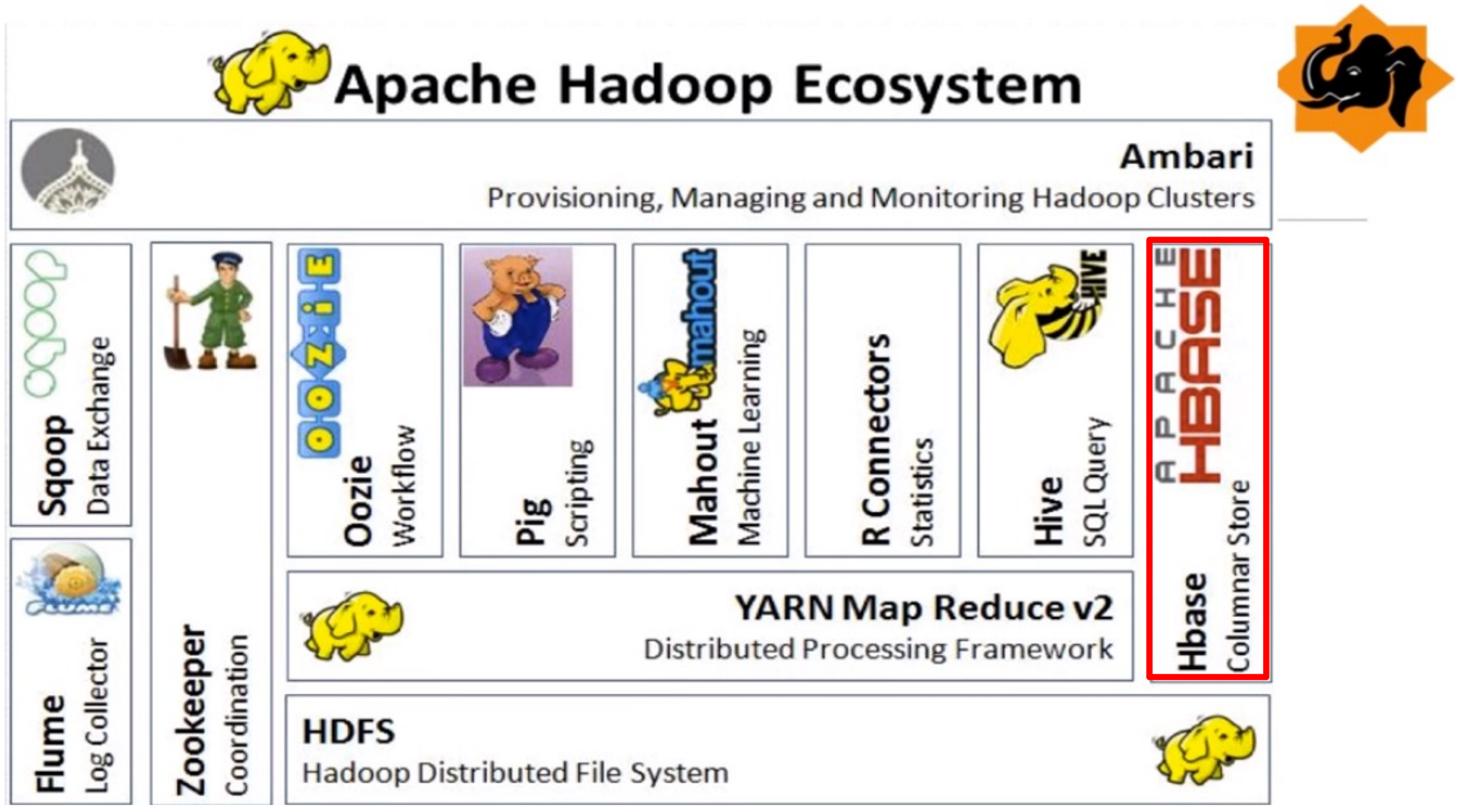


O **Hive** desenvolvido inicialmente pelo Facebook permite a utilização de **Comandos SQL** com o Hadoop.

```
SELECT AVG(price) FROM
trades WHERE id="GOOG"
```

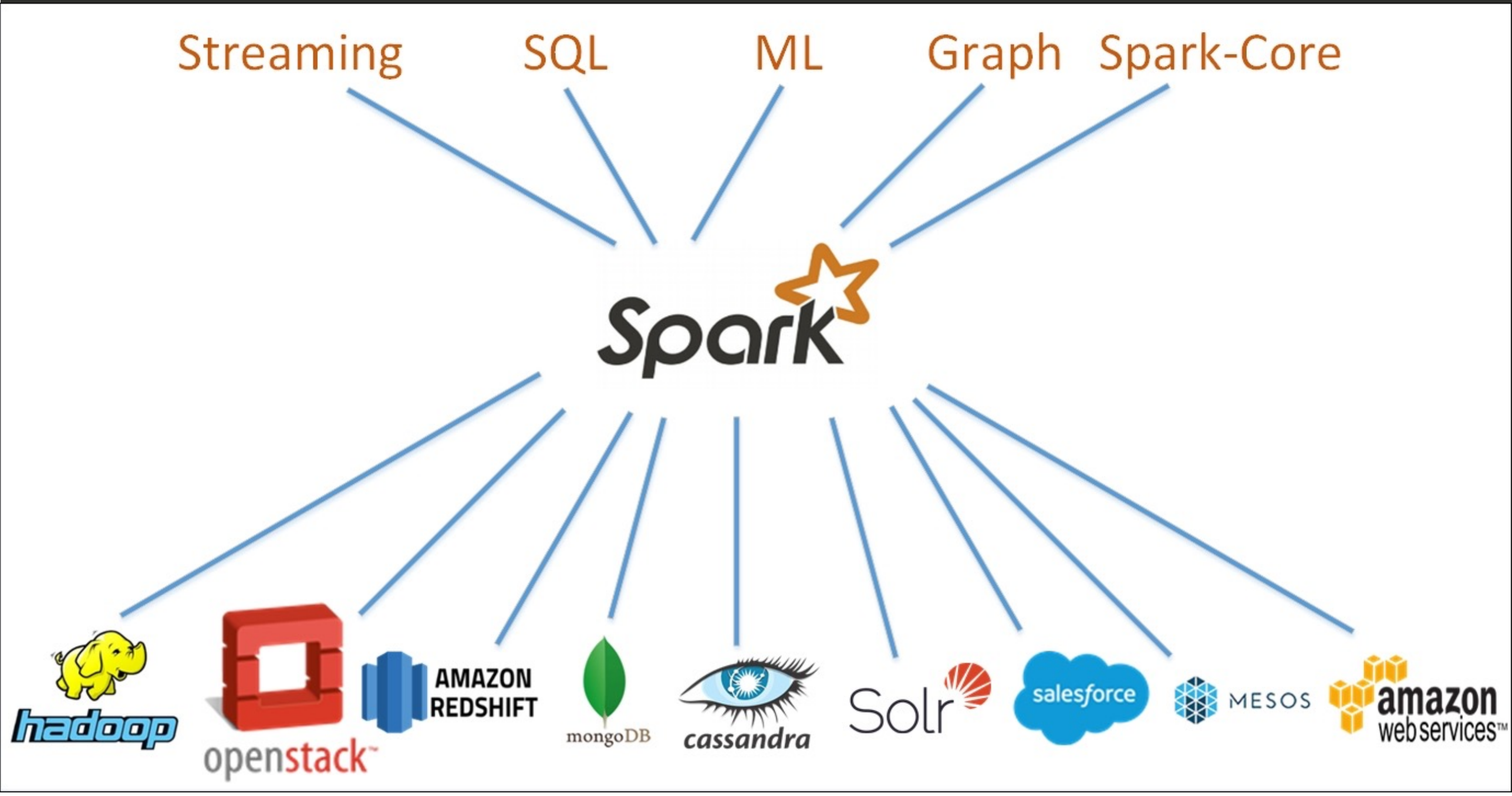


O **Hive** implementa o **HQL** que é muito similar ao SQL e converte os comandos para **Map Reduce**.



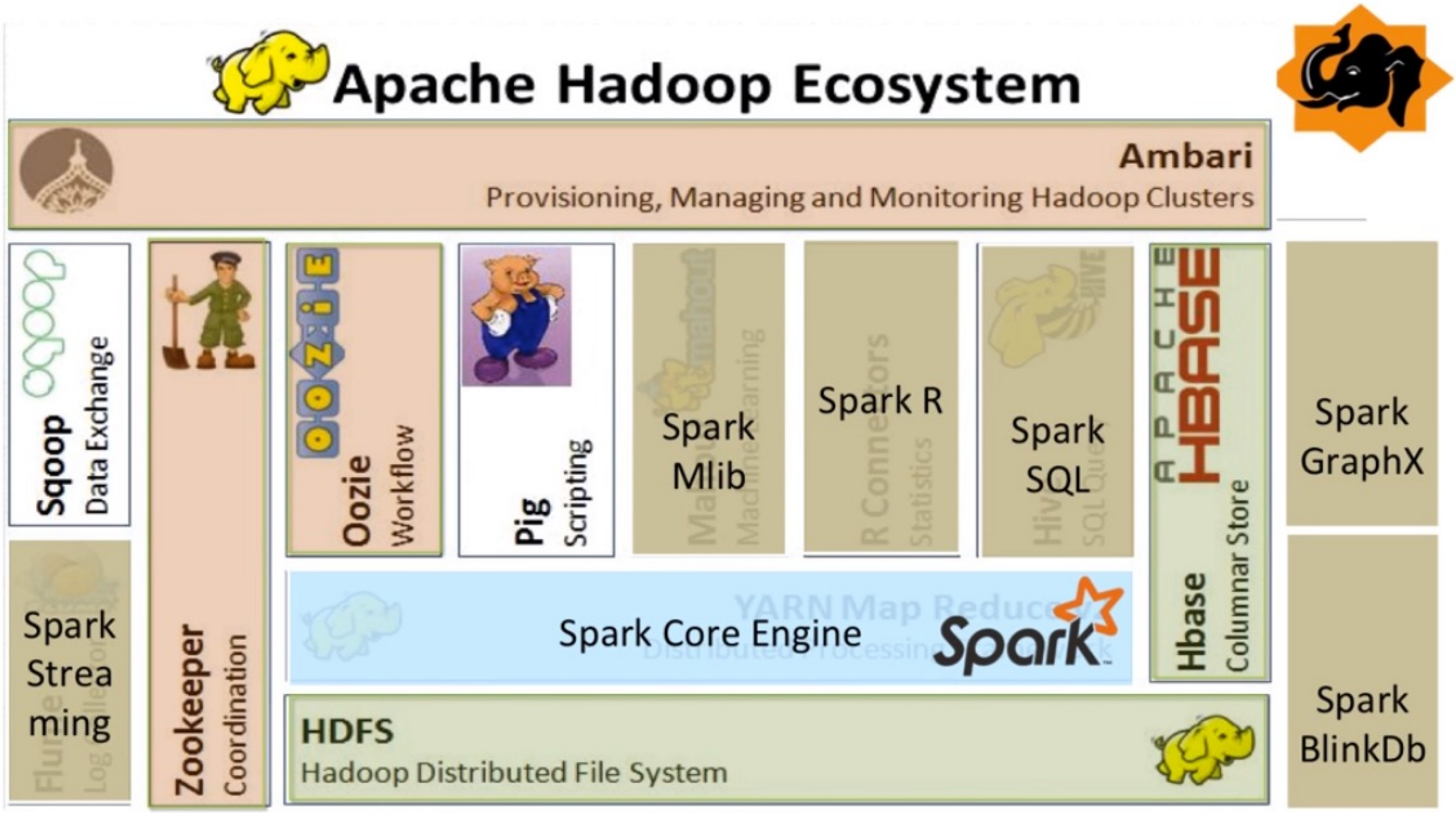
O **HBase** é a base de dados **NOSQL** do Hadoop.

APACHE SPARK

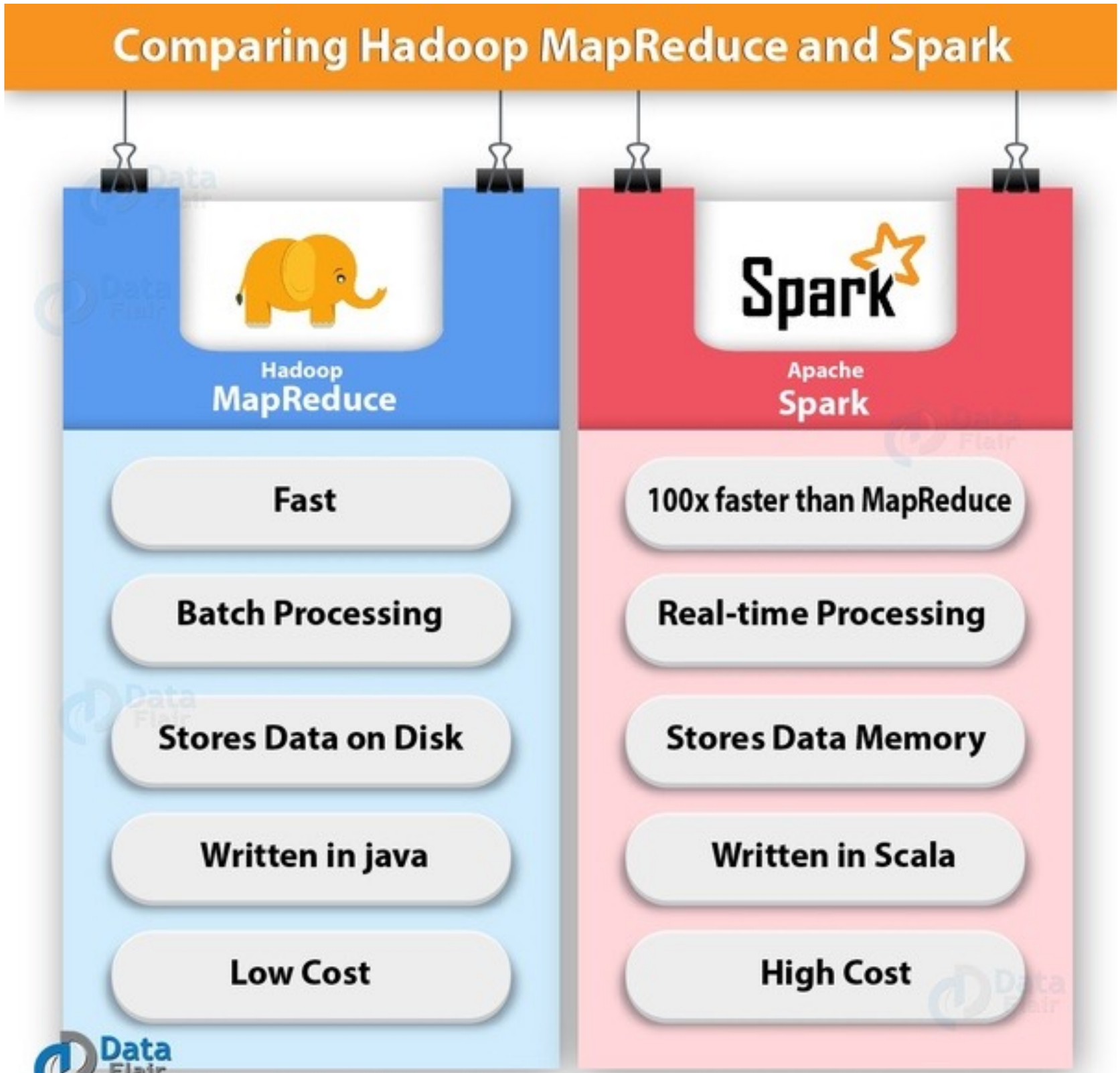


https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781785885136/1/ch01lvl1sec8/apache-spark-architecture-overview

Apache Spark implementa diversas funcionalidades similares a aquelas presentes no Hadoop.



- **GraphX** – permite trabalhar com Grafos de forma eficiente.
- **BlinkDB**- realiza buscas parciais em bases (normalmente NOSQL).



https://www.quora.com/What-is-the-difference-between-Hadoop-and-Spark

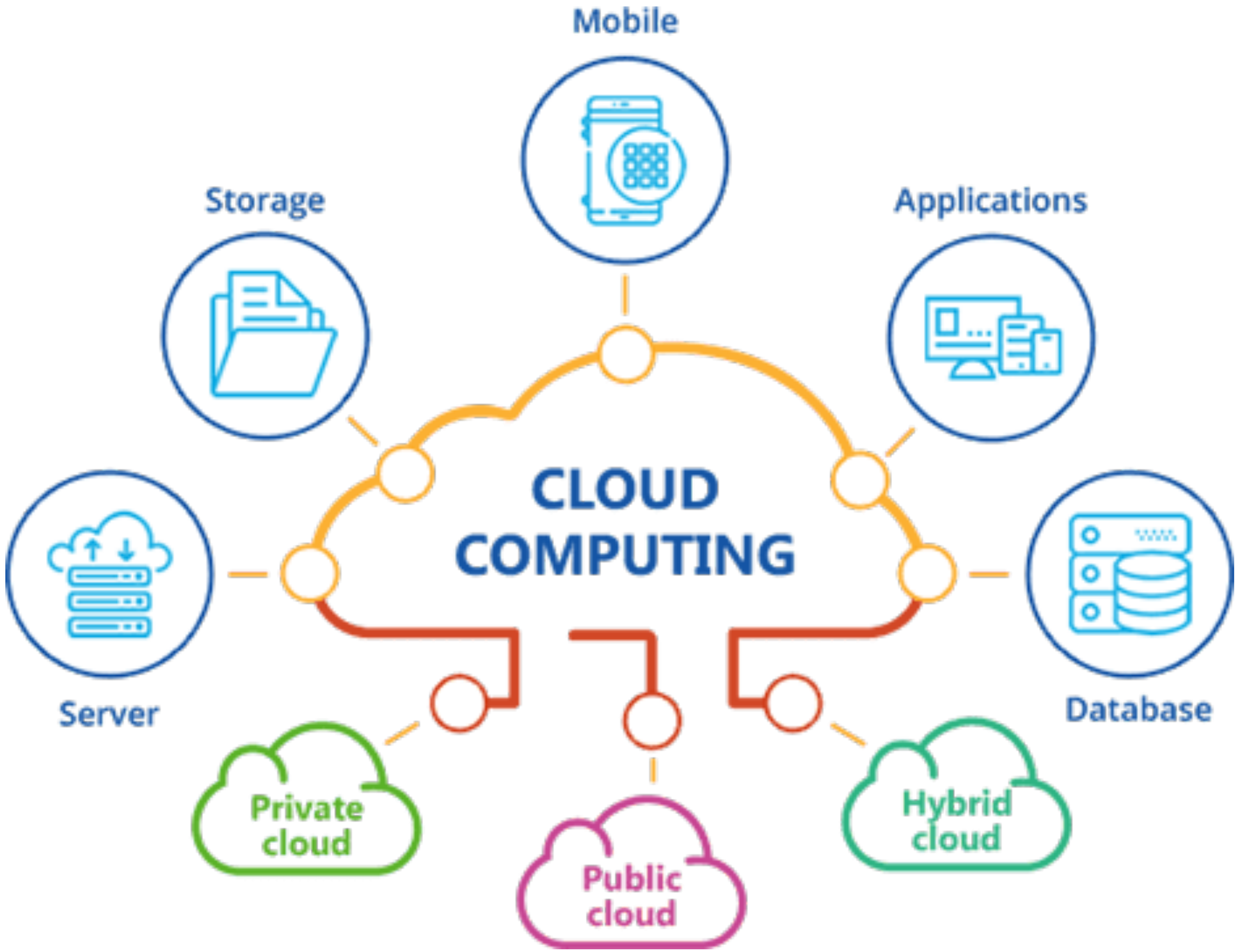
Data Flair

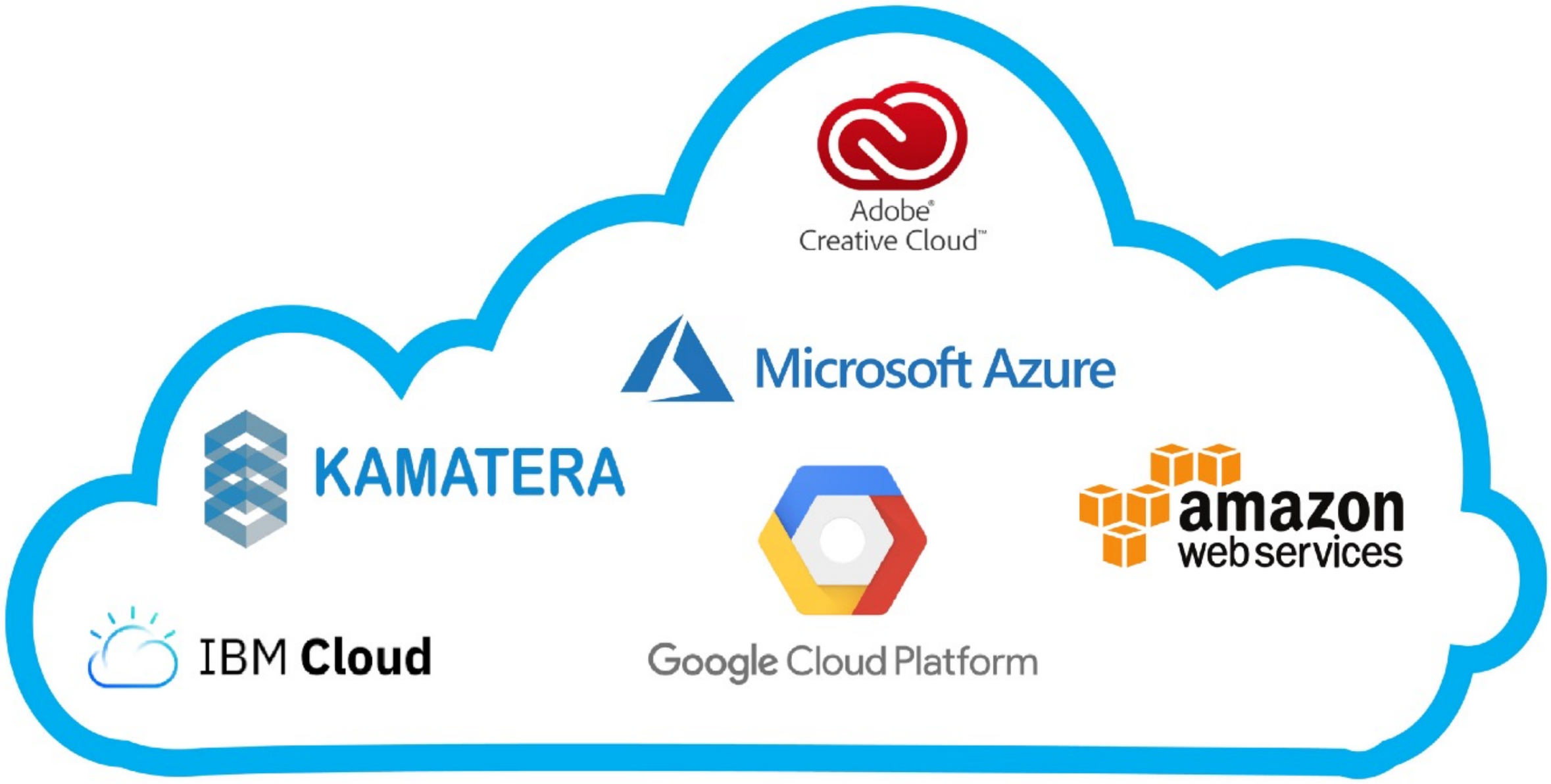
Spark vs Hadoop MapReduce

Factors	Spark	Hadoop MapReduce
Speed	100x times than MapReduce	Faster than traditional system
Written In	Scala	Java
Data Processing	Batch / real-time / iterative / interactive /graph	Batch processing
Ease of Use	Compact & easier than Hadoop	Complex & lengthy
Caching	Caches the data in-memory & enhances the system performance	Doesn't support caching of data

https://www.quora.com/What-is-the-difference-between-Hadoop-and-Spark

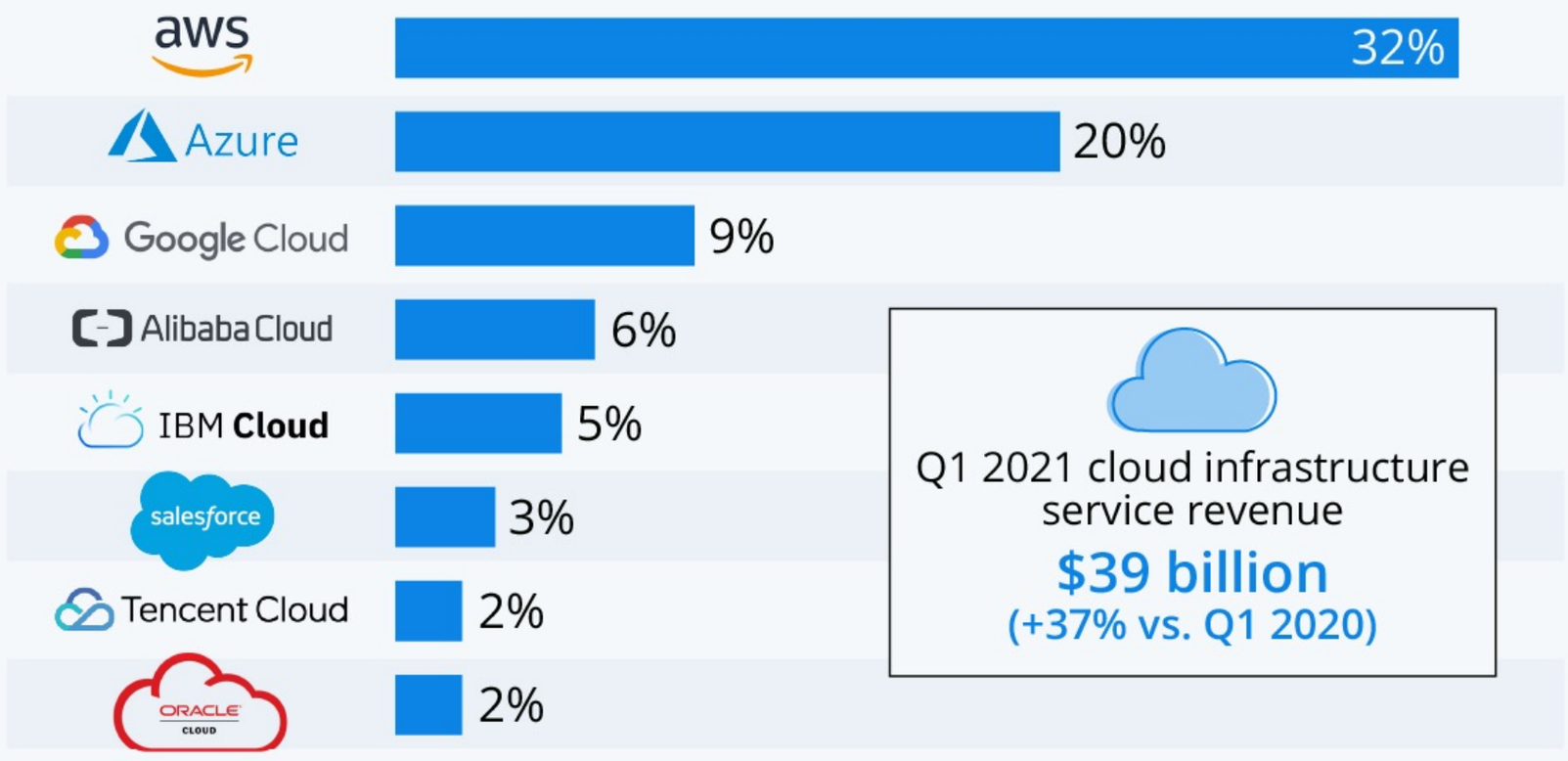
BIG DATA CLOUD COMPUTING


Mobile
Storage
Applications
CLOUD
COMPUTING
Server
Database
Private
cloud
Public
cloud
Hybrid
cloud


Adobe
Creative Cloud
Microsoft Azure
KAMATERA
amazon
web services
IBM Cloud
Google Cloud Platform

Amazon Leads $150-Billion Cloud Market

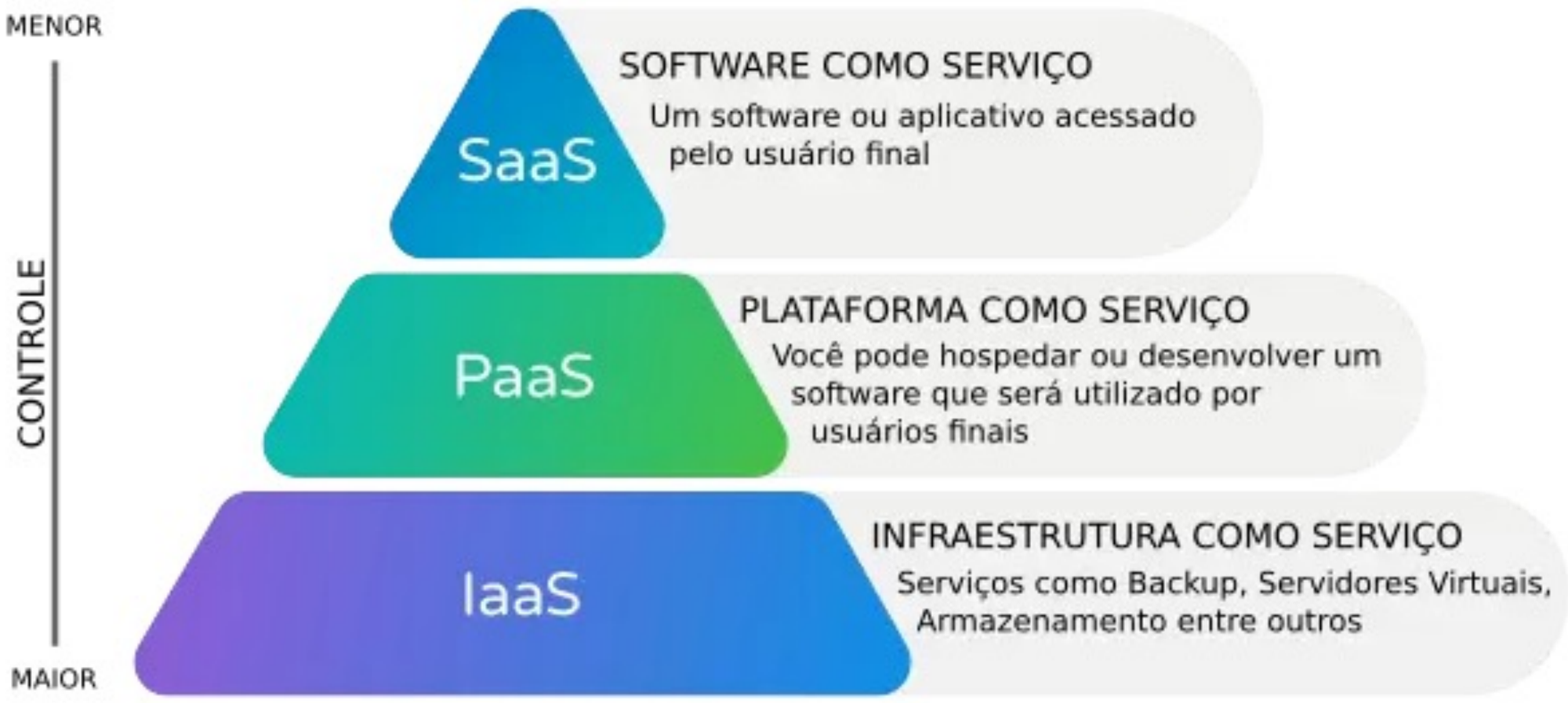
Worldwide market share of leading cloud infrastructure service providers in Q1 2021*



* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group

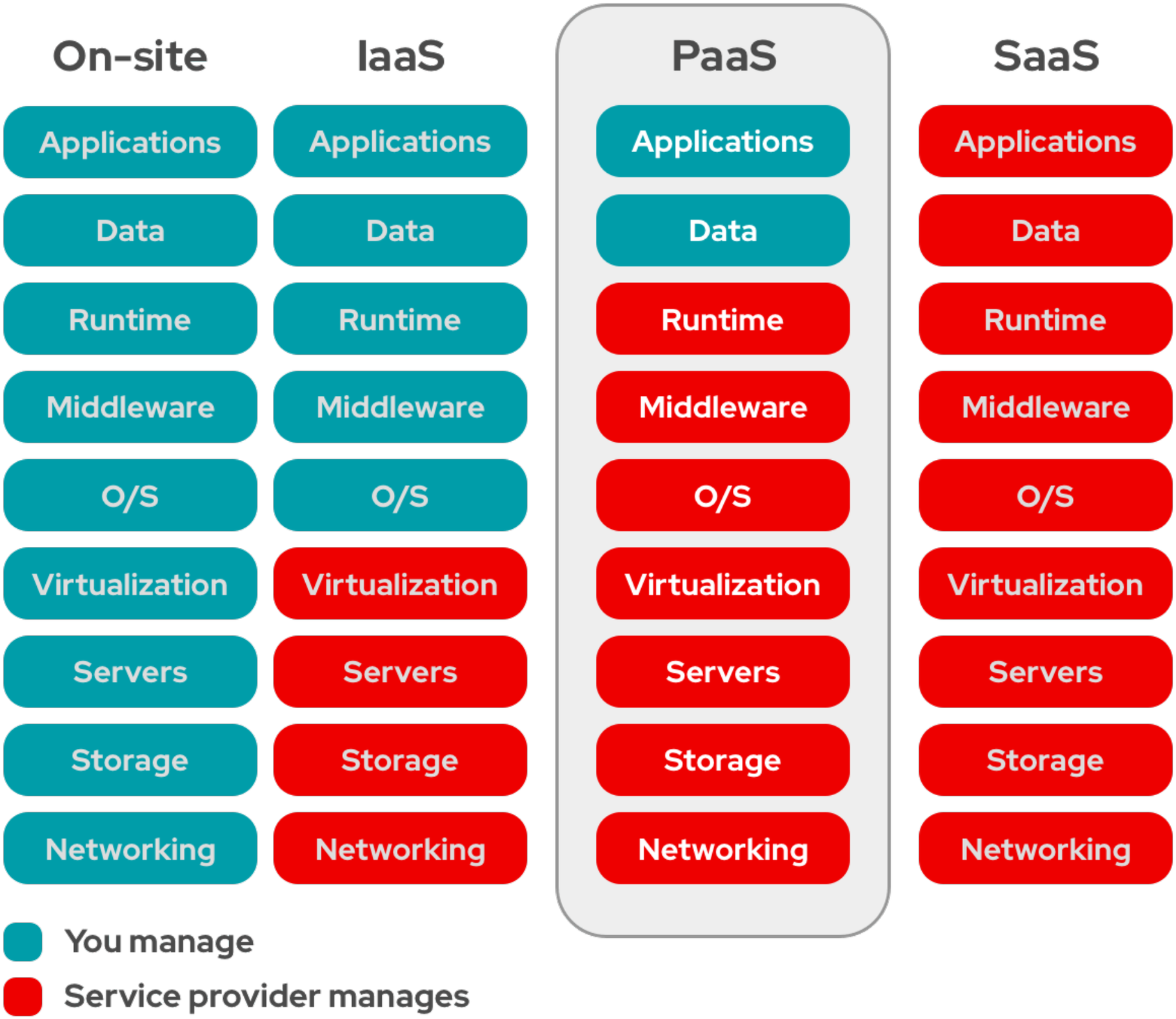
statista



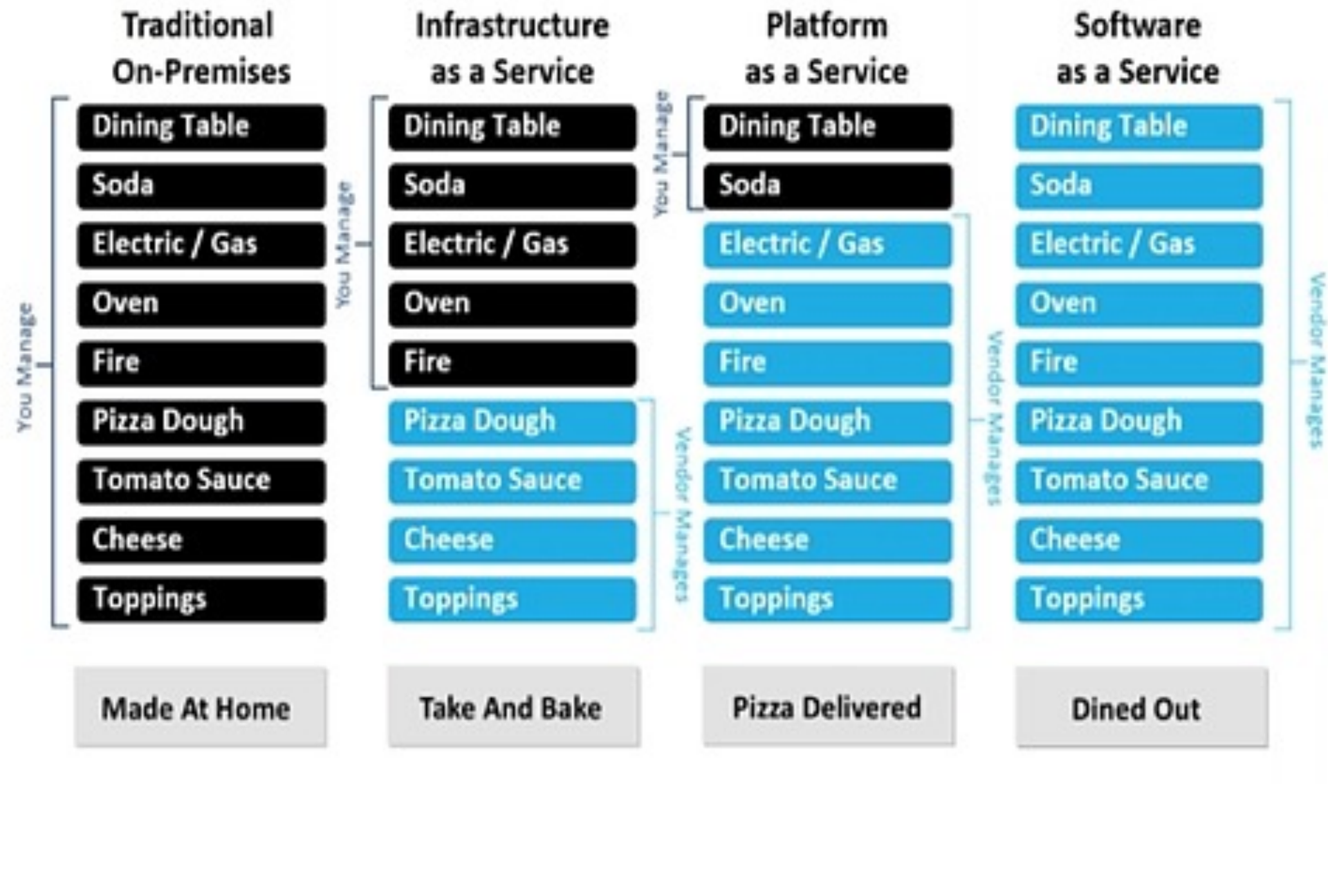
Hierarquia IaaS, PaaS e SaaS

DigitalHouse >

CLOUD – TIPOS SERVIÇOS



The Pizza & The Cloud – How Many Ways?



Top cloud providers in 2021: AWS, Microsoft Azure, and Google Cloud, hybrid, SaaS players

Cloud computing in 2021 has become the de facto choice of IT due to digital transformation shifts accelerated by remote work and the COVID-19 pandemic. Here's a look the cloud leaders stack up, the hybrid market, and the key SaaS players.

By Larry Dignan | April 2, 2021 -- 18:06 GMT (11:06 PDT) | Topic: Managing the Multicloud

Amazon Web Services

The leader in IaaS and branching out

Jump to details

View now at AWS

Microsoft Azure

A strong No. 2, hybrid player and enterprise favorite

Jump to details

View now at Microsoft Azure

Google Cloud Platform

A strong No. 3 with a $11 billion annual revenue run rate, but building out its sales scale and industry approach

Jump to details

View now at Google Cloud

https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/

DigitalHouse >

DATA SCIENCE

Big Data e NoSQL