



# BD<sup>2</sup>:

## des Bases de Données à Big Data

**Professeur Serge Miranda**

Département Informatique

Université de Nice Sophia Antipolis

Directeur du Master MBDS ([www.mbds-fr.org](http://www.mbds-fr.org))

- MOOC 2014-2015 sur plateforme FUN : **trailer du cours**

[http://www.canalu.tv/video/universite de nice sophia antipolis/mooc bd 2 des bases de d  
onnees a big data le trailer.15548\)](http://www.canalu.tv/video/universite_de_nice_sophia_antipolis/mooc_bd_2_des_bases_de_donnees_a_big_data_le_trailer.15548)

# BIG DATA ? Couple :

## 1) Gestion de données (*data management*) :

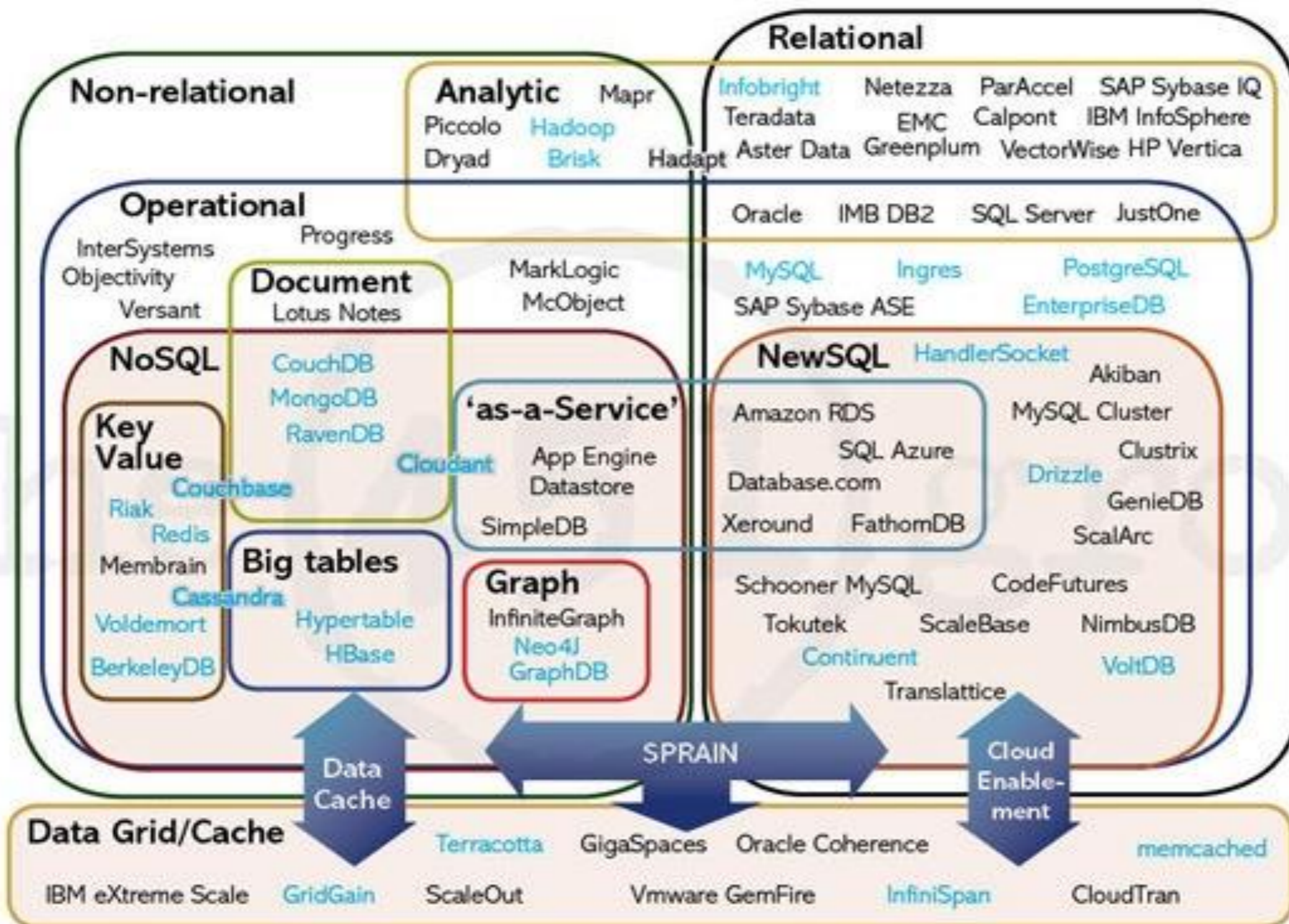
SQL3, OQL, BigquerySQL, NOSQL, CQL, HQL, SPARQL,  
NEWSQL

**NO SQL : REF Open Source : HADOOP/MAP REDUCE**

## 2) Analyse de données (*data Analytics*)

**Mathématiques : Ref OPEN SOURCE : Langage R (> 4000  
packages)**

# Les systèmes de Gestion de Données (DATA Systems) ! (Aslett, 2013)



# DATA SYSTEMS / Systèmes de Gestion des Données *Mobiquitaires* (SGDM) et *paradigmes*

Modèle relationnel de Codd  
Paradigme « Valeur »



SQL3, SQL3/ODMG  
NEW SQL

Modèle « OBJET »



paradigme « POINTEUR-VALEUR »  
(SQL3)  
paradigme « OBJET-VALEUR (ODMG)



SPARQL  
(OWL)

N.O. SQL



paradigme « RDF »  
(Web Sémantique)

paradigme « CLE-VALEUR »  
(Map Reduce)

# Plan

- *Introduction pluridisciplinaire : Une vision stratégique du futur des systèmes d'information avec la mobiquité et BIG DATA < C0 : Séminaire stratégique sur les systèmes de données du futur >*
- Les concepts fondamentaux des bases de données (*Schéma de données, Modèle de Données*) et du BIG DATA avec leurs paradigmes et propriétés en support : TIPS /ACID, RICE, WHAT (et CABS de Google) <C1 : Cours Introduction>
- **Double Approche des Systèmes de données :**
  - **Top down**
    - **Paradigme VALEUR et propriétés TIPS/ACID support du Modèle Relationnel de Codd et SQL2 (Cours 2, Cours 3)**
    - **Paradigmes POINTEUR-VALEUR et propriétés RICE support du Troisième Manifeste de Chris DATE et des standards SQL3/ODMG (Cours 4, Cours 5 et Cours 6)**
    - **Paradigme RDF support des LINKED DATA/Web sémantique (SPARQL,OWL) (Cours 7)**
  - **Bottom Up**
    - **Paradigme CLE-VALEUR et propriétés WHAT support du N.O. SQL (HADOOP/MAP REDUCE) avec le Cours 8**
    - **NEW SQL avec connecteurs Hadoop au SGBD Oracle (Cours 9)**



# Cours 1 : Introduction aux Bases de données et à BIG DATA

Professeur Serge Miranda

Département Informatique

Université de Nice Sophia Antipolis

Directeur du Master MBDS ([www.mbd-s-fr.org](http://www.mbd-s-fr.org))

# « BIG DATA » ? Buzz Word !

Big data is like teenage sex:  
everyone talks about it,  
nobody really knows how to do it,  
everyone thinks everyone else is  
doing it, so everyone claims they  
are doing it....

ÉTUDE RÉALISÉE DU 1<sup>ER</sup> AU 31  
AOÛT 2014 SUR DES DONNÉES  
PUBLIQUES ISSUES DU WEB ET  
DES RÉSEAUX SOCIAUX SUR LE  
THÈME DU BIG DATA.



**6 913**

**messages collectés  
sur le thème du  
BIG DATA**

#### RÉPARTITION DES MESSAGES

	Twitter <b>4 752</b>		Facebook <b>272</b>		Divers / Flux RSS <b>339</b>
	YouTube <b>345</b>		News & Blogs <b>1 033</b>		Forums <b>17</b>
	Instagram <b>158</b>				



# SDSS (Sloan Digital Sky Survey)

**Carte tridimensionnelle (1/3 voute)**

**- 470 Millions d Astres;**

**- 2 M de Galaxies**

**- Projet de 10 ans**

**- Comprendre la VOIE LACTEE ?**

**- découvrir des EXO PLANETES ?**

**→ Image d un PETA PIXEL !**

**→ (besoin de 500 000 Ecrans HD  
pour la visualiser)**

**→ 71 PETA Octets de données**

# Autres Exemples BIG DATA

**Déforestation** : projet PlanetarySKIN (7 tera de données satellites)

**Suivi astronomique en direct** : Projet LSST (30 Tera chaque nuit)

**Micro-organismes marins**: Projet GOS (2 teraoctets)

**Bio Chimie** sur 100 millions de molécules : Projet BSRc

**Cancer du foie** :projet ICGC (200 teraoctets) analyse des BD sur 25000 tumeurs de 50 types de cancers

**Détection épidémies en temps réel** : Projet Healthmap (1 teraoctets) : Suivi progression cholera en Haiti avec 2 semaines d'avance (cholera, grippe, dengue, ebola..)

# **BIG DATA landscape (sky\*) in 2014 ?**

**\* « *The SKY is the limit* » !**

**" LES DATA SONT LE PÉTROLE DU**

**XXI E SIÈCLE » GILLES BABINET** (de *L'Ère*

*numérique, un nouvel âge de l'Humanité*, JANVIER 2014)

***100 milliards d'adresses IP seront utilisées en 2025 (cent fois plus qu'en 2003).***

**MOBIQUITE/ BIG DATA** aussi important que

**Invention ECRIT**

**Invention IMPRIMERIE**

**→ Distribution connaissance, Education, SANTE (NBIC)  
Economie, Etat**

# « DATA » (Donnée) ? vs « Information » ?

## « DATA » (DONNEE)

ENREGISTREMENT DANS UN Code d'un fait ( objet, transaction, observation) du monde réel

## « Information » : Ce que je peux DEDUIRE d'un ensemble de DATA

Ex : Livre de Médecine Chinoise de 1000 pages en Chinois (ensemble de data !)  
« tout le bruit du monde »

Adrian Mc Donough dans *Information economics* définit l'information comme la rencontre d'une donnée (data) et d'un problème

## DATA : « OR GRIS » de ce millénaire !

« Capital immatériel »; Stratégie du « KNOWING YOU » de Google;

« **COMMUNACTEUR** » : acteur d' enrichissement bottom up des COMMONS (EX / Wikipedia, Open Source, Réseaux sociaux,...)

# « DATA » en préfixe ou suffixe!

## 1) DATA en Préfixe

DATA base (19/8/1968 : Ted Codd et Modèle Relationnel), DBMS

DATA bank

DATA warehouse

DATA mart

DATA mining (OLAP, Corrélations, ..), Data Analytics, DATA Pumping (ETL)

DATA Systems

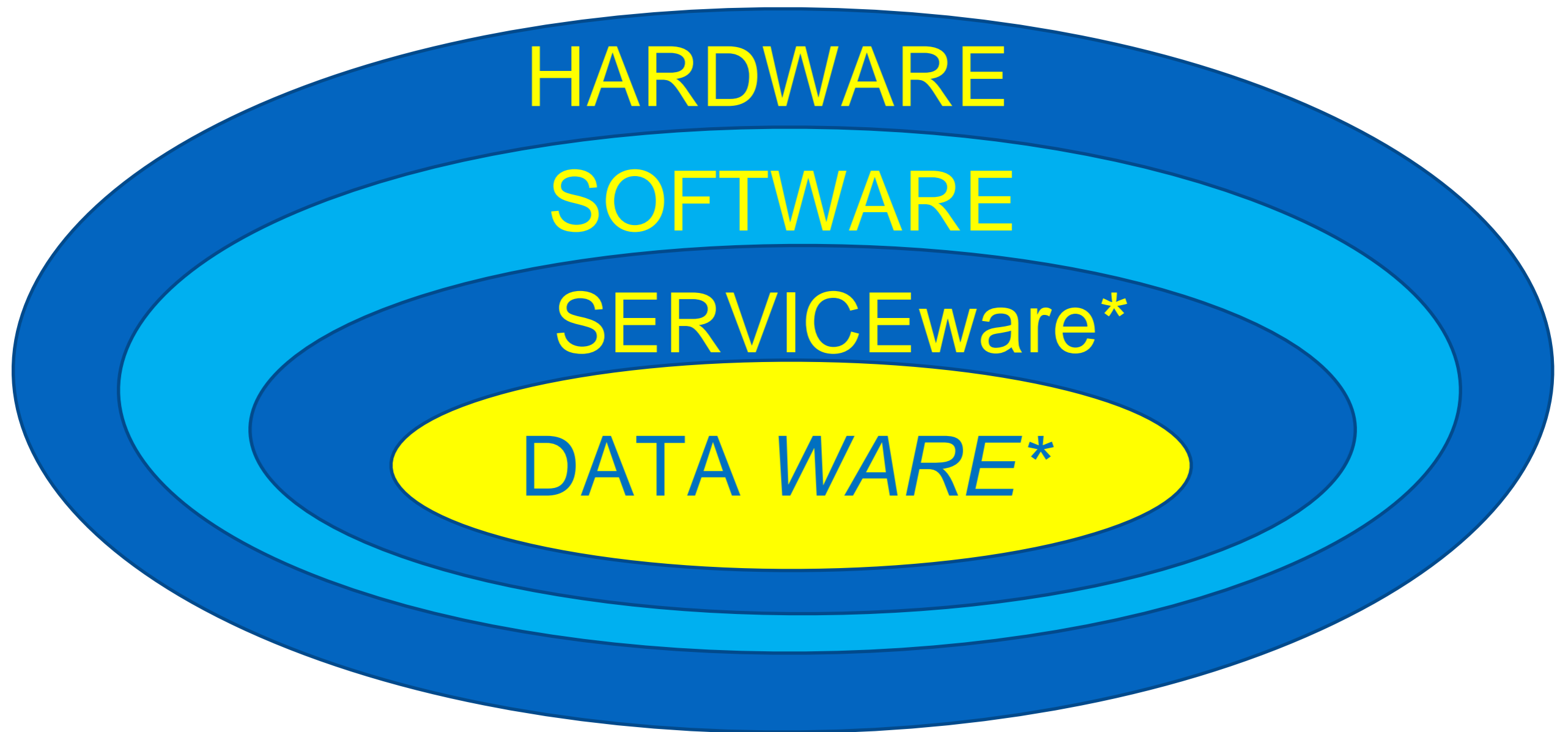
DATA mash up

DATA SCIENCE

## 2) DATA en suffixe :

- Linked DATA, Web DATA (DBpedia, Web Sémantique)
- Meta DATA
- Open DATA
- Smart DATA
- BIG DATA et nouveaux métiers centrés DATA : :
  - CDO « Chief DATA Officer »,
  - « DATA SCIENTIST »,
  - « DATA BROKER »

# Révolution Copernicienne en NTIC



\* *USERWARE* ⑨ « *DATA  
SCIENCE* »

# Ecosystème GESTION de Données du futur



# L' environnement du futur des Bases de données : « CAMS » (IBM 2014)

## « CAMS » :

INFOSTRUCTURES Client Serveur BD avec le Serveur dans le CLOUD

- DaaS/AaaS : « (DATA) ANALYTICS as a service »

- Intégration de gros volumes de données (BIG DATA) avec parallélisme vertical ( *SCALE UP* ) ou horizontal ( *SCALE OUT* )

Intégration des données temps réel des réseaux SOCIAUX, du WEB SEMANTIQUE (RDF), des objets tagués, des capteurs

- Connecteurs Hadoop/SQL

2) Intégration MOBILITE et du transactionnel (NFC)

# Vikram Pandit

(President Citygroup)

Futur = Pile de « **SMAC** »

- **Social**
- **Mobile**
- **APPlications**
- **Cloud**

« Economie du partage » (Uber, AirB&B, Drive, .;)

« *Aucun modèle d'activité ne pourra réussir sans les DATA* »

# CLOUD COMPUTING ?

1) « **INFRASTRUCTURE as a SERVICE** » (IaaS)

2) « **PLATFORM as a SERVICE** » (PaaS)

3) *DATA ?*

- « **DATA as a Service** » d'Oracle (DaaS)

« **ANALYTICS as a SERVICE** » (AaaS) de Google  
Bigquery (Google 2012)

# Parallélisme et Bases de Données

**Lecture Teraoctets ( $10^{12}$ ) par seconde ?**

**Disque Dur type : 100 Mega Octets/sec**

1 Peta Octet ( $10^{15}$ ) par sec ?

→ des milliers de disques dur

3 Solutions :

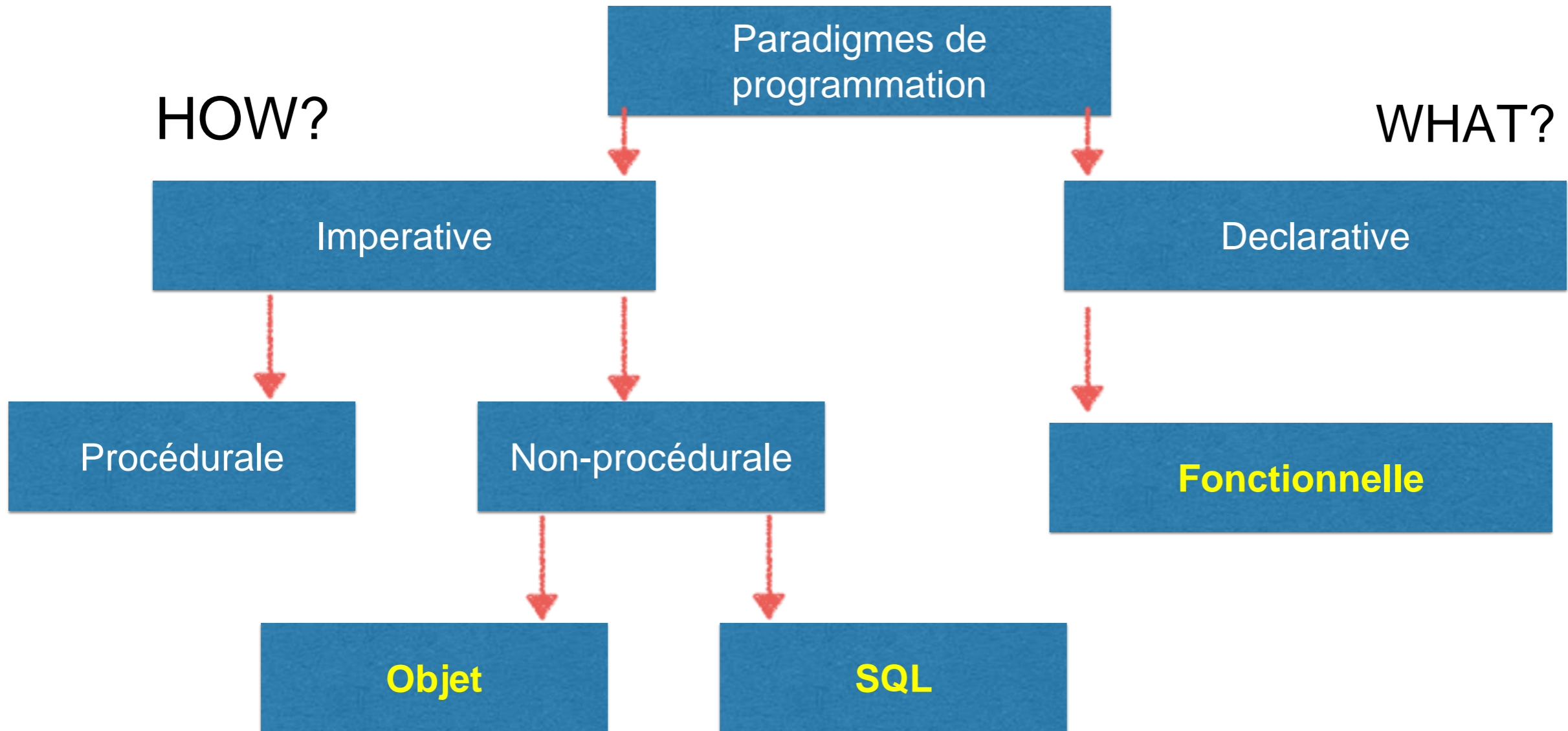
1) Réduire les données

2) SCALE UP : Serveurs parallèles puissants (SMP, MPP)

3) SCALE OUT : Parallélisation sur des milliers de machines

*Exercice : Avant de commencer ce cours, allez sur Internet comprendre la différence entre les types d'architectures du CLOUD (SaaS et PaaS) ainsi que les types de programmation informatique ci dessous*

## Synthèse des « Paradigmes de programmation » [Manning2013]



# Concepts de BASE de GESTION des DONNEES

# *Univers réel, SCHEMA et modèle des données*

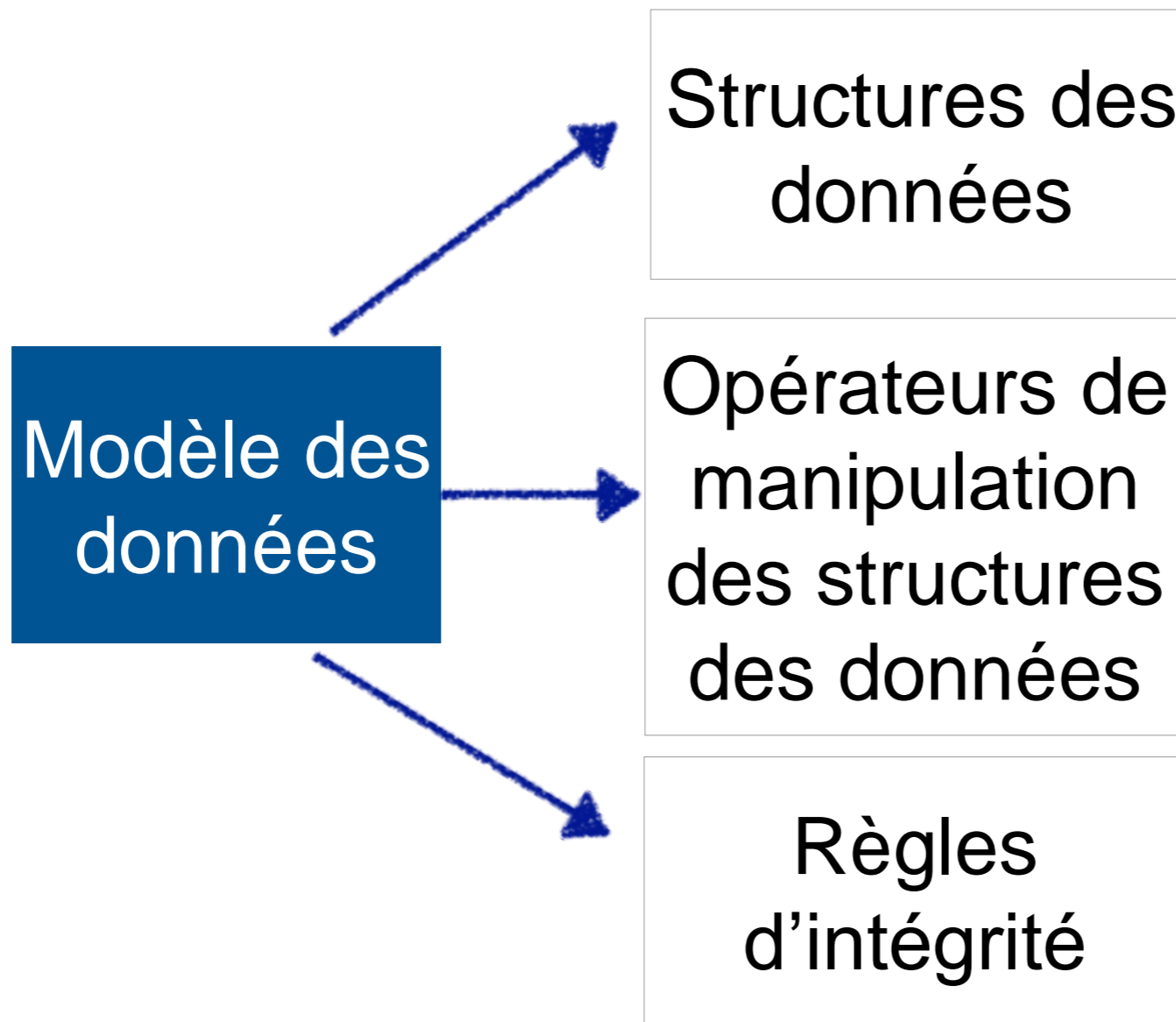
---

Approche de « STRUCTURATION » du monde réel avec un SCHEMA résultant de l'application d'un MODELE de DONNEES (*data model*)



# Modèle des données (*DATA MODEL*)?

---





# Les 3 structures de données du Modèle Relationnel de Codd (19/8/1968)



VALEURS (DATA)



Const. SET

DOMAINES



Const. TUPLE

RELATIONS  
(*"tables" en SQL*)

Domaine = ENSEMBLE de valeurs

*Une RELATION du modèle de Codd est un  
« prédicat à N variables » ou un ENSEMBLE  
(sous ensemble du Produit Cartésien de N Domaines)*

# Représentation d'une RELATION dans le modèle de CODD : Table de valeurs



Domaine  
(« domaine »)

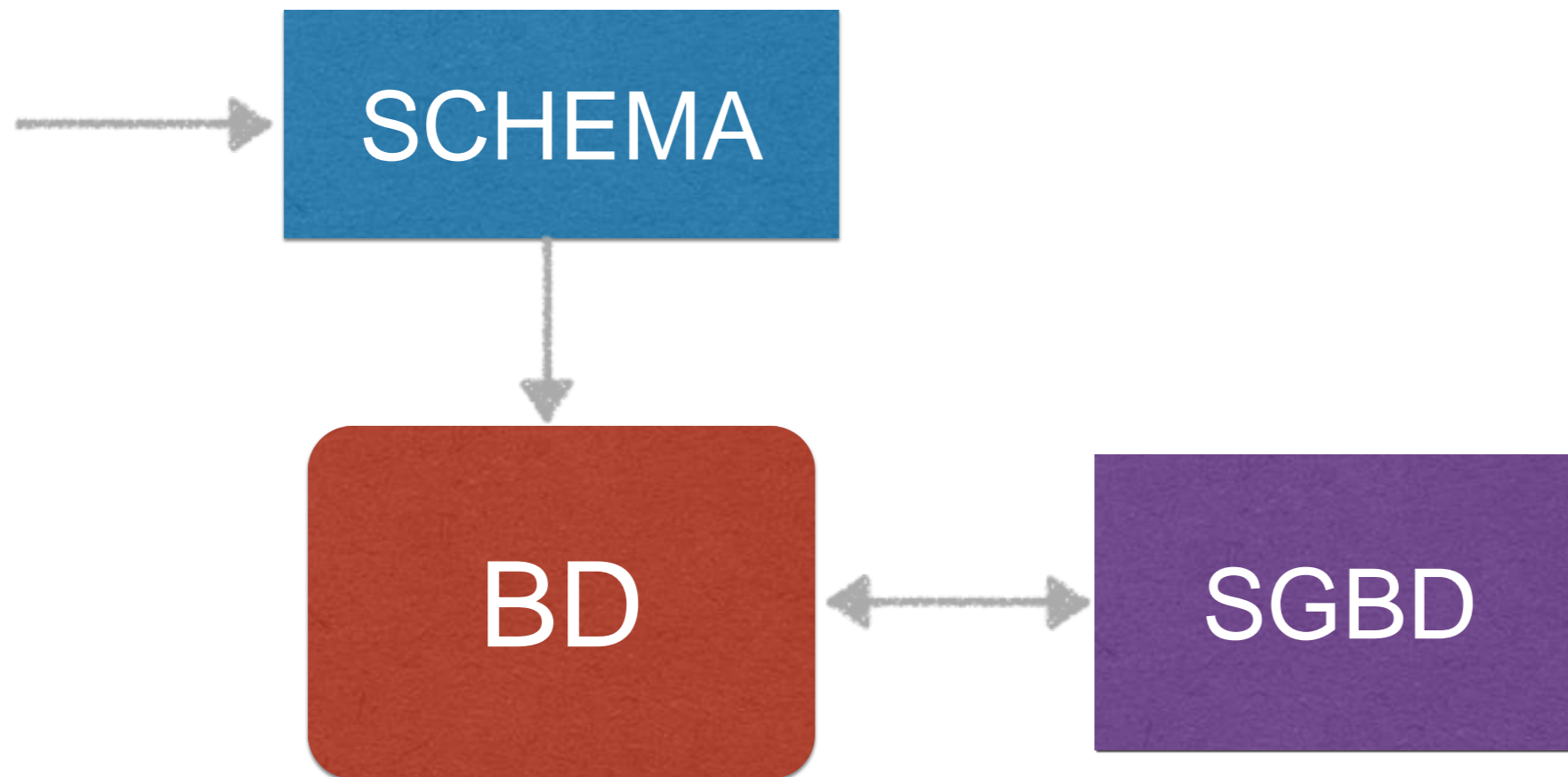
Ville: {Nice, Paris, Lyon, Toulouse}

Pilote	PILNO	PILNOM	ADR
	100	Serge	Nice
	101	John	Paris
	102	Pierre	Toulouse

Ligne= N-UPLET (« *TUPLE* »)  
COLONNE = "ATTRIBUT" (« *ATTRIBUTE* »)

# *Base de Données (data base) et SGBD (DBMS) ?*

---



**SGBD?**

- DEFINITION
- MANIPULATION
- CONTRÔLE  
d'une BD

## Exemple de Schéma relationnel sous forme prédicative (3 prédicats à n variables/ 3 relations)

PILOTE (PL#, PLNOM, ADR)

AVION (AV#, AVNOM, CAP, LOC)

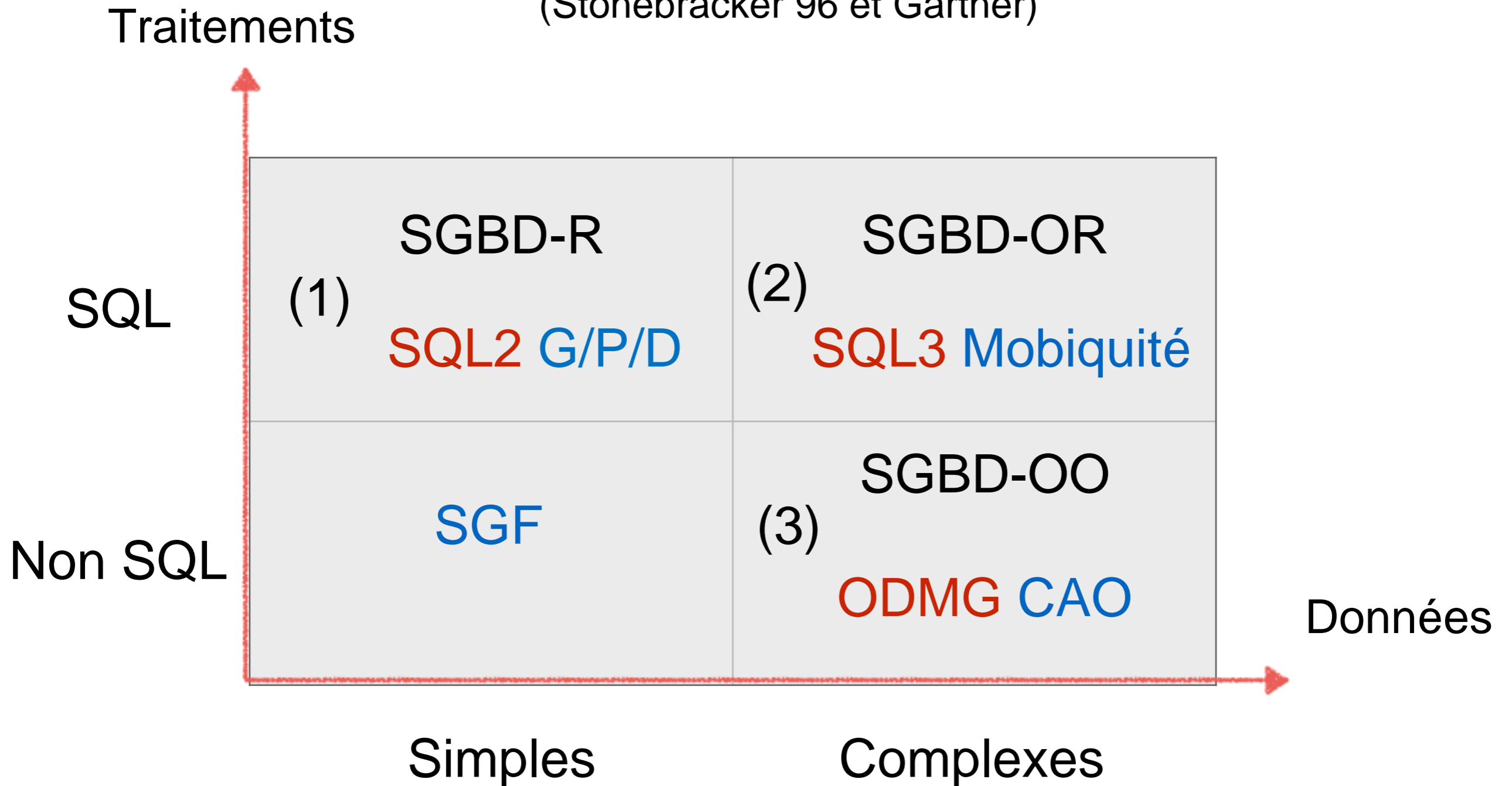
VOL (VOL#, PL#, AV#, VD, VA, HD, HA)

Notes :

- « *Attribut Clé Primaire* » souligné
- Pas de « domaines » dans ce schéma

# Marché BD et standards?

(Stonebracker 96 et Gartner)



(1) 10 G\$ <licences \*> en 2010  
(20 % de croissance, 60 G \$ en 2020)  
(3) : 1/100 de (1) en 2010 et 2020  
(2) : 2x (3) en 2010 ; 2\*(1) en 2020 !  
\* Marché de 27 G dollars avec services et support en 2010

## Approches TOP DOWN des bases de données : standards SQL2, SQL3, ODMG et RDF

Données d'entreprise et standards :

SQL2 et «*paradigme VALEUR*»

SQL3 et «*paradigme POINTEUR-VALEUR*»

ODMG et «*paradigme OBJET-VALEUR*»

WEB DATA et «*paradigme RDF*»



## Rappels programmation : VALEURS vs VARIABLES vs POINTEURS

**VALEUR ?** « constante non modifiable" cf DATA

**VARIABLE ?** Toute variable possède un NOM, une VALEUR et une ADR mémoire ;

**VARIABLE := ( NOM, VALEUR, ADR)**

**POINTEUR ?** type de variable qui contient l' ADR d'une autre variable comme valeur (« indirection »)



# Variables et ses opérateurs de base

Les Variables ont des adresses (pas les valeurs) :

→ . YPE « ADRESSE » avec 2 opérateurs de base :

\* **Référencement/adressage ("referencing")** :  $v \rightarrow \text{adr}$

en C : `ptr = &v;` (avec `char v;` et `char *ptr;`)

```
en PL/1      DECLARE N INTEGER
              DECLARE P POINTER
              P= ADDR (N)
```

\* **Déréférencement ("dereferencing")** :  $\text{adr} \rightarrow v$

en C : `*A` ; en PL/1 : `A--> V`

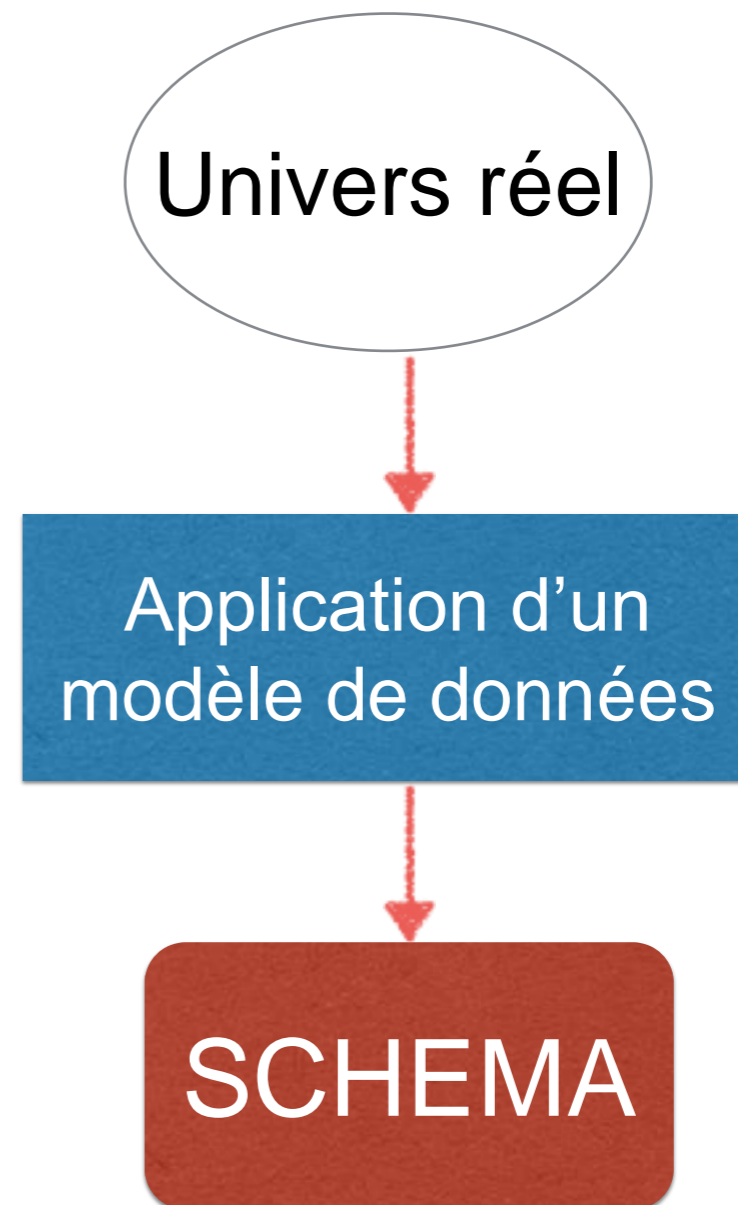


# Approche *Top Down* SQL/ODMG

Approche *top down*  
par  
**STRUCTURATION**  
des données



PRE définition d'un  
**Schéma « fixe »** de  
données



# Apports principaux des *BD Relationnelles* à la communauté informatique : propriétés TIPS

---



« *Transactions* » (avec *propriétés ACID*)



Interface non procédurale (standard SQL)



Persistance (Mémoire paginée)



Structuration (Schéma)

# Apports « TRANSACTIONNELS » des BD

TIPS avec « T » : « TRANSACTIONS »

Propriétés « ACID » des Transactions :

Atomicité,  
Cohérence,  
Isolation,  
Durabilité

OLTP (*On line Transaction Processing*)

Data Warehouse/data Mining (et OLCP : *On Line Complex Processing*)

→ Approche TOP DOWN

# Exercice

Les propriétés TRANSACTIONNELLES « ACID » visent à résoudre 2 problèmes importants dans la cohérence d'une base de données : lesquels ?

# Solution

ACID et 2 PB : PANNE et CONCURRENCE

AC : **Atomicité**- (TOUT ou RIEN des opérations de MAJ concernant une transaction) et **Cohérence** de la BD quelle que soit **la PANNE** qui pourrait se produire

ID (Isolation et Durabilité) : Chaque transaction « bien formée » et écrite de manière **isolée** avec des mécanismes de verrouillage (Verrouillage en lecture/écriture/intention) à deux phases maintiendra **durablement** la cohérence de la BD quelle que soit **la concurrence** (le nombre de transactions qui s'exécutent en parallèle)

# Modèles, Manifestes et standards BD

Modèle Relationnel de CODD (19/8/1968)

→ **standard SQL2**

3 Manifestes sur les "BD du Futur" (Bancilho, Stonebraker et Chris DATE)

3ieme Manifeste de Chris Date

**2ieme Manifeste de M. Stonebraker**

→ **standard SQL3** ("Modèle OR"- Objet Relationnel-)

- **1<sup>ier</sup> Manifeste de Bancilhon**

- **standard ODMG** ("Modèle OO" –Orienté Objet-)

# Apports principaux de l'approche OBJET à la communauté Bases de Données

R

Réutilisabilité (*Héritage ou polymorphisme*)

I

IDENTIFICATION système  
(OID : Object Identifier)

C

Construction d'objets Complexes

E

Encapsulation (Méthodes)

# Les 3 Approches de MODELES de DONNEES OBJETS et OR (Objet Relationnel)





# SQL2 – Relationnel- (Exemple)

Quels sont les pilotes Niçois qui sont en service au départ de Nice ?

```
SELECT pl#, plnom  
FROM pilote, vol  
WHERE pilote.pl#= vol.pl# and pilote.adr= 'Nice' and vol.vd= 'Nice';
```

*Dans l'algèbre de Codd (pas à pas)*

```
V1 = Join Pilote (pl#= pl#)Vol  
V2 = Select V1 (adr= 'Nice' and VD='Nice')  
RES = Project V2 (pl#, plnom)
```

# SQL3 (objet relationnel) - Exemple

Quels sont les pilotes Niçois qui sont en service au départ de Nice ?

```
SELECT REFPIIL -> PL#,PLNOM  
FROM VOL  
WHERE VD= Nice and REFPIIL -> ADR ='Nice';
```

*Note : Avec*

*- REFPIIL attribut de type REF contenant les ROWID (OID) de Pilote et « -> » : Opérateur de déréréfencement*

# OQL (ODMG) -Exemple-

Quels sont les pilotes Niçois qui sont en service au départ de Nice ?

```
SELECT p.PI#, p.PLNOM  
FROM  
  p in pilote  
  v in p.assurevol  
WHERE  
  p.adr= 'Nice ' and v.vd='Nice';
```

*Note : Avec « assurevol », pointeur REF bidirectionnel défini dans le schéma ODMG depuis la classe Pilote vers la classe Vol*

# Approche TOP DOWN de GESTION DES DONNEES *(hors SQL/ODMG)*

- OPEN DATA
- WEB DATA (Web Sémantique)
  - **Paradigme RDF** (Resource Description Framework)

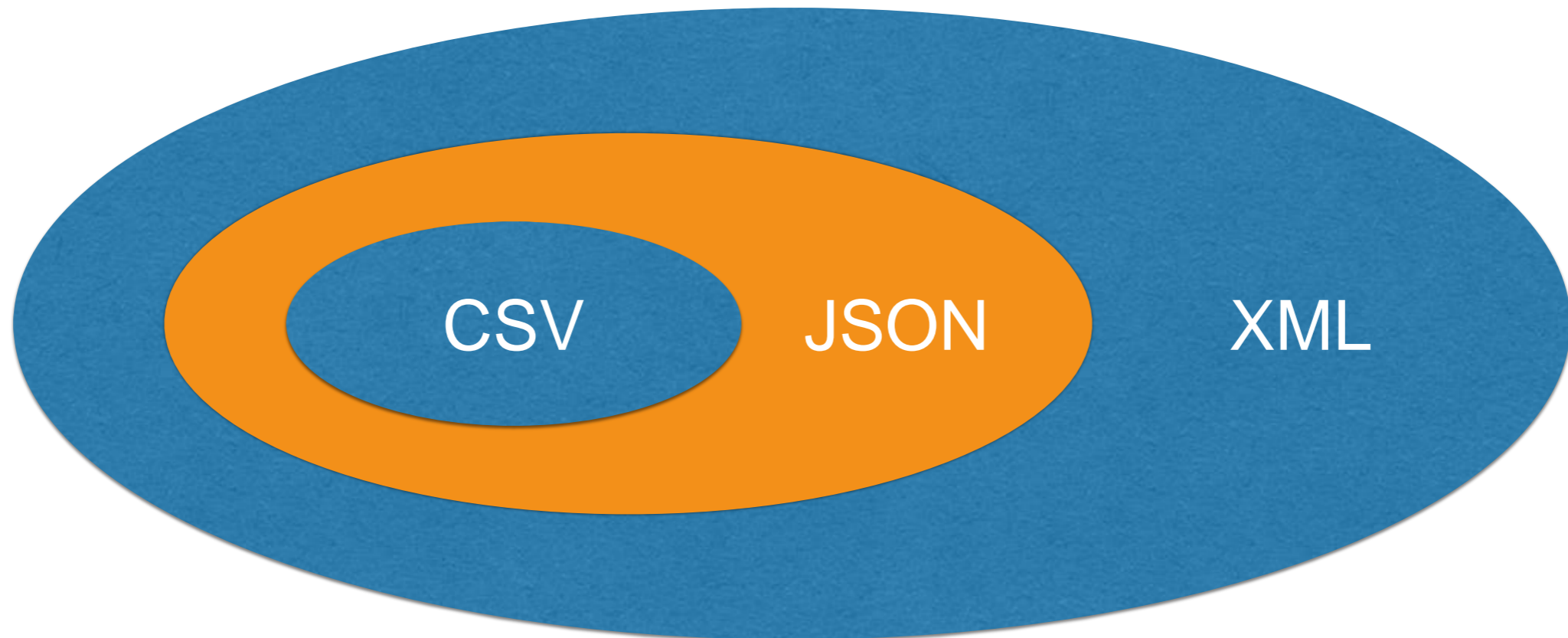
# OPEN DATA

- « *Une **donnée ouverte** (en anglais **open data**) est une information publique brute, qui a vocation à être librement accessible et réutilisable.  
La philosophie pratique de l'open data préconise une libre disponibilité pour tous et chacun, sans restriction de copyright, brevets ou d'autres mécanismes de contrôle. » < Wikipedia >*

# Formats OPEN DATA

- - PDF pour les documents
  - Pour les DATA :
    - CSV (Excel)
    - Standards du Web pour publication, partage et liaison
      - HTML (HTML5), XML, RDF
    - Standards du Web pour syndication
      - RSS, Atom, JSON

# OPEN DATA : CSV, JSON, XML ...



**CSV (Comma Separated Value )** pour fichiers plats (1)

**JSON (Java Script Object Notation)** pour documents hiérarchiques (2)

**XML (eXtensible Markup Language)** pour (1), (2), namespaces,...

# CSV, JSON et XML (Exemples)

## #CSV example

Prenom, nom, titre cours, date  
« Serge », « Miranda », « des BD à Big Data », « 2015 »

---

## // JSON example

{« Prenom »: « Serge », « nom »: "Miranda »,  
« cours »: {« titre »: « Des BD à BIG DATA », « date » : "2015 »}}

---

## <!-- XML example -- >

```
< xml < professeur> serge Miranda</professeur>  
<list>  
<cours> Des BD à BIG DATA </cours>  
<date> 2015 </date>  
</list>  
</xml>
```



# WEB DATA (semantic web)

*“I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web — (the content, links, and transactions between people and computers. A “Semantic Web”, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The “intelligent agents” people have touted for ages will finally materialize”)*

*TIM Berners Lee (2001, Weaving the web”)*

# (Open) Linked DATA / Semantic WEB

Variante *Open Data* issue du *Web Sémantique* : **Open Linked Data**

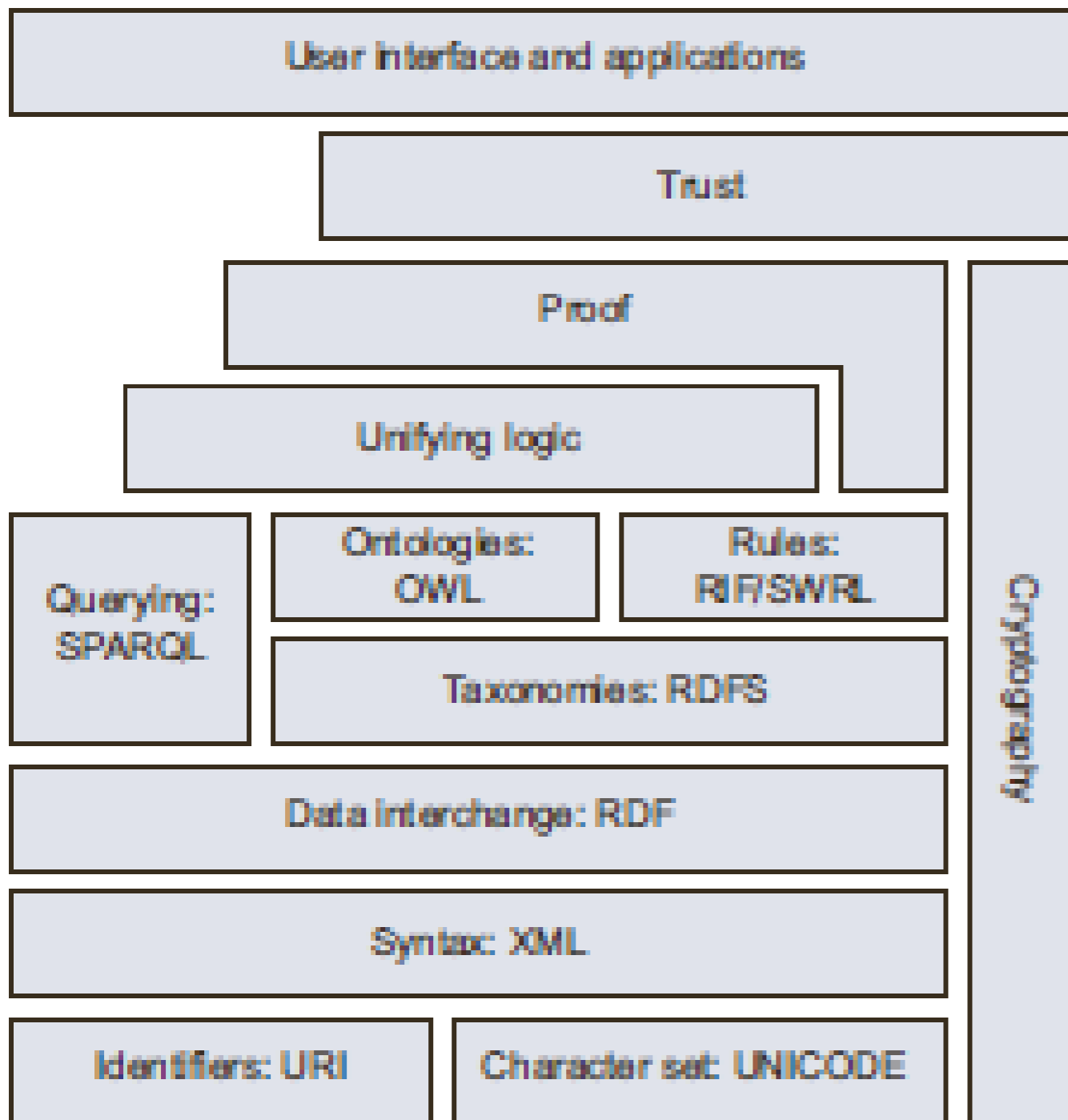
**Le Web sémantique** est un mouvement collaboratif mené par le [World Wide Web Consortium](#) (W3C) <sup>1</sup> qui favorise des méthodes communes pour échanger des données.

Le Web sémantique vise à aider l'émergence de nouvelles connaissances en s'appuyant sur les connaissances déjà présentes sur Internet. **Pour y parvenir, le Web sémantique met en œuvre le [Web des données](#) qui consiste à lier et structurer l'information sur Internet pour accéder simplement à la connaissance qu'elle contient déjà.**

# « 5 star » LINKED OPEN DATA

En 2010, [Tim Berners-Lee](#) a donné une échelle de qualité des données ouvertes qui va de zéro à 5 étoiles.





# SEMANTIC WEB stack

(Manning2013)

**Figure 4.16** A typical semantic web stack with common low-level standards like URI, XML, and RDF at the bottom of the stack. The middle layer includes standards for querying (SPARQL) and standards for rules (RIF/SWRL). At the top of the stack are user interface and application layers above abstract layers of logic, proof, and trust building.

# WEB SEMANTIQUE ?

Un « **Modèle de Données** » !

→ **Des Structures des données(format commun)**

Des identifiants universels de ressources du Web (**URI**)

Un Format unique : **RDF**

Un schéma : **RDFS**

→ **Un langage de Manipulation**

**SPARQL**

**OWL**

# RDF

## *Resource Description Framework*

**Défini par le W3C (January 15th, 2008)**

Héritage de la syntaxe XML

Utilise des **URI** pour identifier les Ressources

- Web page (identifiée par URL)
- Web Service
- fragment d'un document XML
- tout objet, concept, ..

# Example: RDF

< [http://fr.wikipedia.org/wiki/Bill\\_Gates](http://fr.wikipedia.org/wiki/Bill_Gates) > *Sujet*

< <http://www.w3.org/pim/contact#mailbox> > *Prédictat*

« [bill.gates@microsoft.com](mailto:bill.gates@microsoft.com) » *Objet*

# Les DATA en RDF

## Des triplets pour décrire les ressources WEB

(:serge: assureVOL:AF100)

(:Pierre: assureVOL:AF110)

(:AIRBUSA320: est-utilise-dsVOL: AF100)

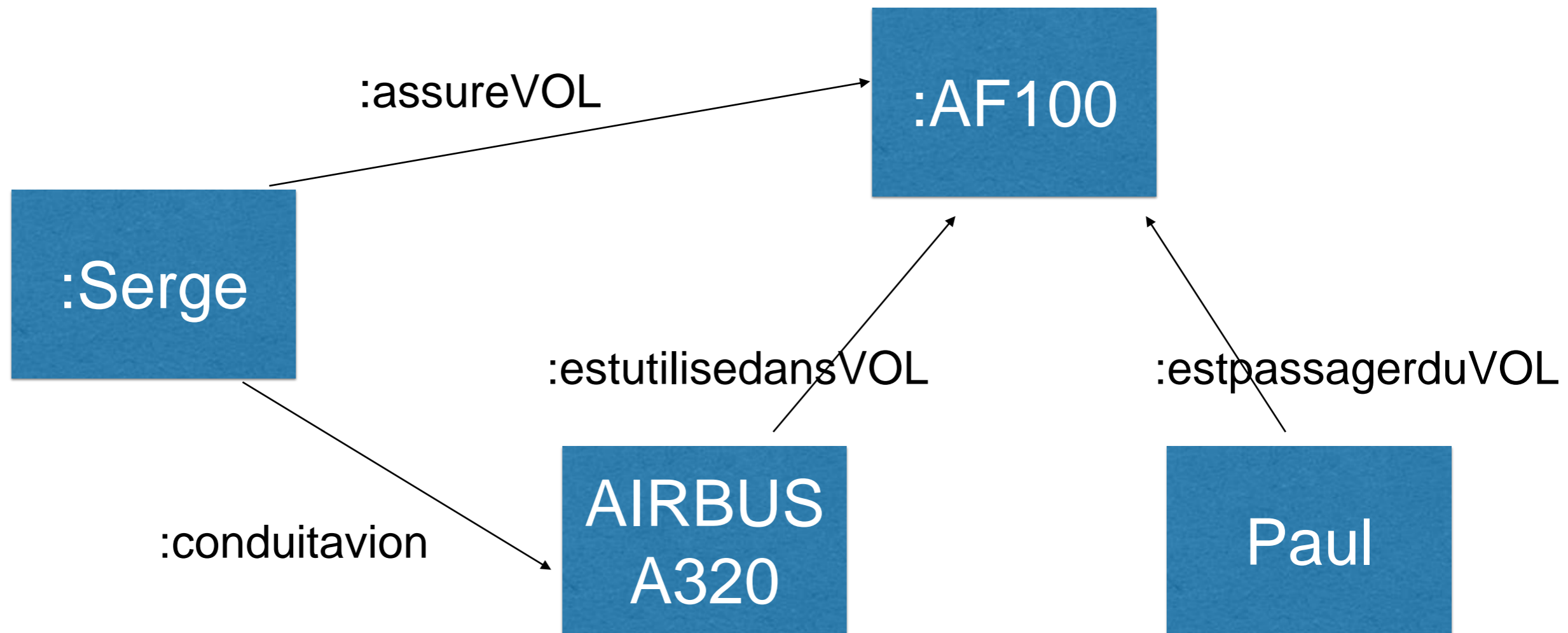
(:Paul: estpassagerduVOL:AF100) ...

**Note : Un triplet RDF <S.P.O> est un FAIT en logique du premier ordre :  $P(S,O)$  avec  $P$ : Prédicat,  $S$  Sujet et  $O$  objet**

Exemple : ASSUREVOL (Serge, AF100)



# Graphe RDF (Exemple)



- **SPARQL (requêtes sur graphe RDF)**
- PREFIX dc: <http://purl.org/dc/elements/1.1/> <URI abrégé> “?” < free variable“ :3 <Data Source>

**SELECT ?X**

**WHERE** { <http://.../.../ > dc:Y ?X } < liste des triplets>

**FROM** Nom du graphe RDF

# SPARQL (Exemple)

Quels sont les pilotes Niçois en service au départ de Nice ?

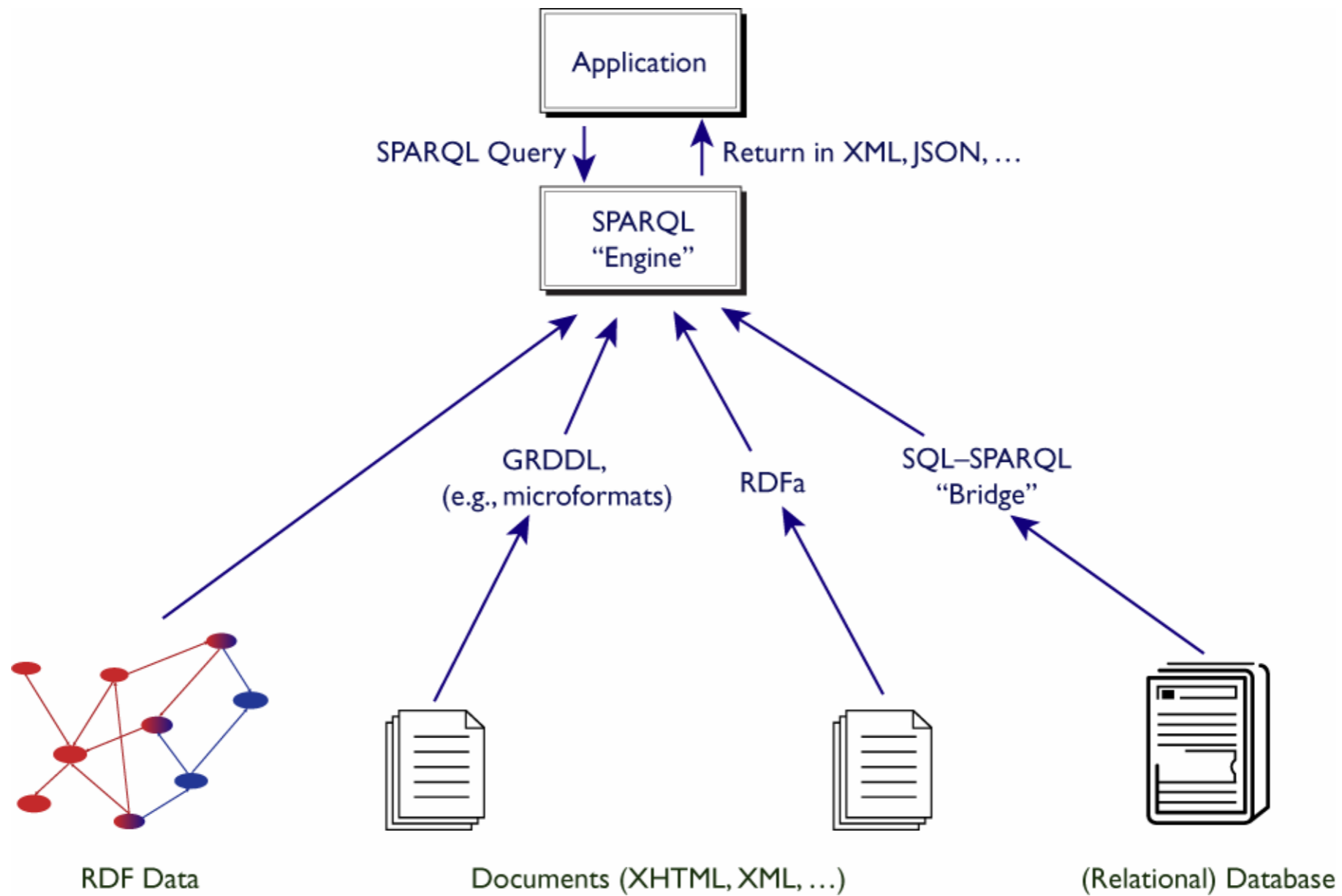
Prefix rdf :<http:// www....>

**SELECT ? Pilote**

**WHERE { GRAPH ?g**

**{ ?pilote rdf :adresse rdf: Nice**

**?vol rdf:villedepart rdf: Nice }}**



Note : **GRDDL (2007)** to get RDF triples out of XML documents

# Recherche : Passerelle SPARQL et SQL\*

```
SELECT ?person ?tel WHERE {
  ?person rdf:type bm:GraduateStudent
  {
    { ?person bm:like ?interest } UNION
    { ?person bm:love ?interest }
  } .
  OPTIONAL { ?person bm:telephone ?tel } .
  ?person bm:age ?age .
  FILTER ( ?age < 25 &&
           REGEX(STR(?interest), "Ball$") )
}
```

\*

ina,

**Fig. 1.** An example of SPARQL query

telephone number of all graduated students with age less than 25 and have an interest of ball sports, in a UOBM [2] ontology (The bm:age is an extended property of the UOBM).

L

# Approches BOTTOM UP de gestion des données du BIG DATA

Deux grands types de systèmes de données **Bottom Up** :

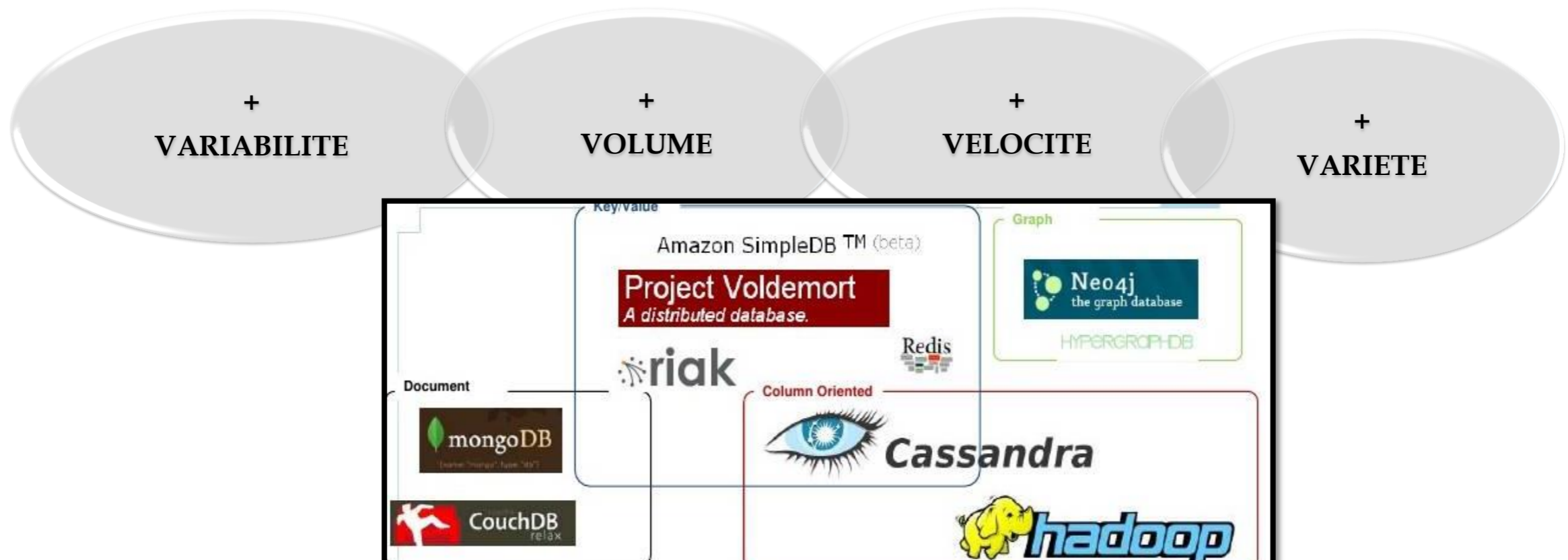
- « N.O. SQL » (Not Only SQL) avec le Paradigme **CLE-VALEUR**
- « NEW SQL » (dont BigQuerySQL de Google) avec le Paradigme **TABLE-CLE/VALEUR**

# NO SQL (Not Only SQL)

(1998)

Approche « *Non Seulement SQL* » (*NOT ONLY SQL*) permettant la gestion de BIG DATA avec les 4 « NO » :

- 1) **NO SCHEMA** (schema free)
- 2) **NO JOIN** (extract data without joins)
- 3) **NO DATA FORMAT**(graph, document, row, column)
- 4) **NO ACID Transactions**



# Propriétés « BASE » transactionnelles et théorème CAP du Big Data

## **BASE :**

**Basically**

**Available**

**Scalability**

**Eventually consistent**

## **CAP Theorem (Brewer, 2000)**

**Consistency,**  $\longrightarrow$  **SQL**

**Availability,**

**Partitioning**  $\longrightarrow$  **NO SQL**



# 4 Grands types de base N.O. SQL

## CLE-VALEUR (Hadoop, ..)

Pour lier les fichiers avec des CLES

Table CLE VALEURS comprenant 3 fonctions simples : PUT, GET, DELETE

## Orienté COLONNES (HBASE, BIG TABLE, ..)

Pour les Matrices vides

## Orienté Documents (MONGO DB, ..)

Pour documents hierarchiques structurés ou non

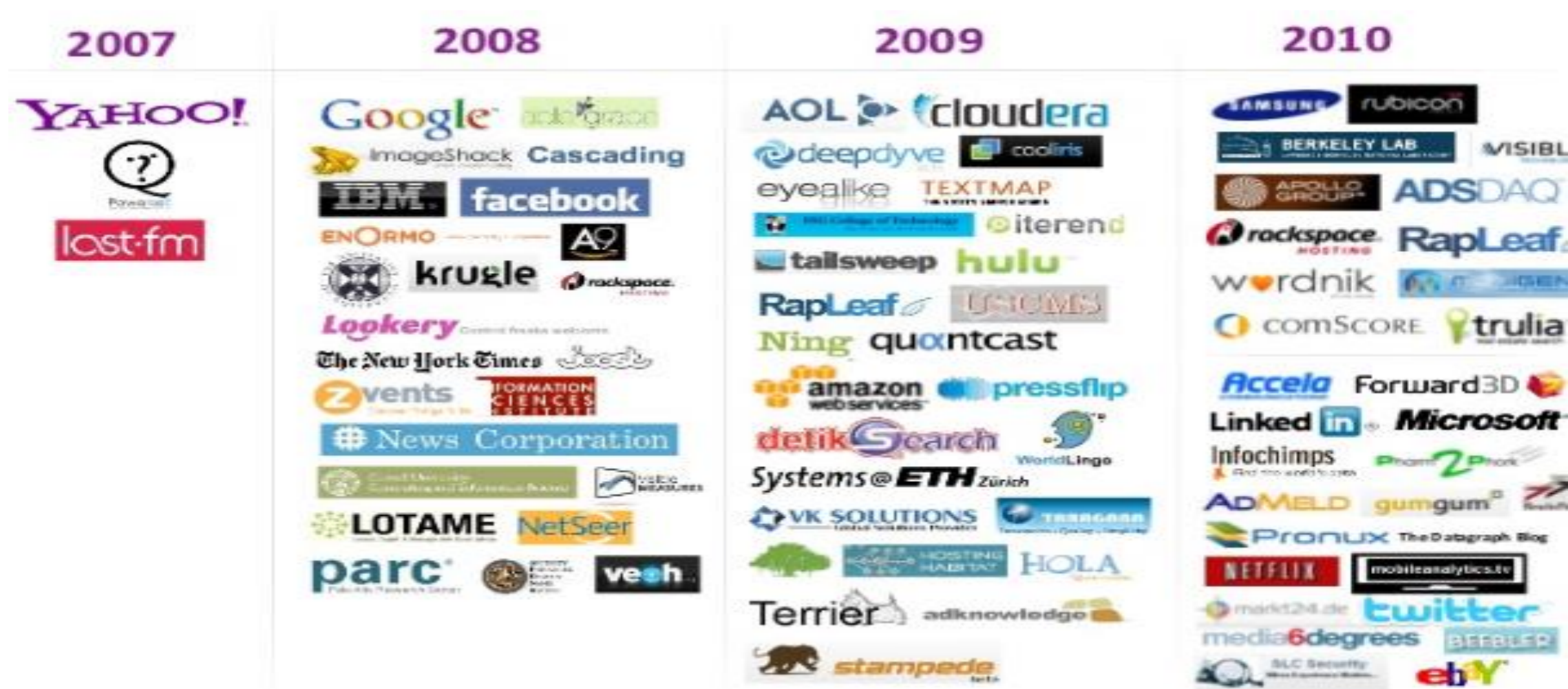
## Orienté GRAPHES (NEO4J,..)

Pour les réseaux sociaux

Traversée graphe indépendante de sa taille

# Introduction à Hadoop

Heck Another Darn Obscure Open-source Project



Un modèle tolérant aux pannes

- Replication des données entre les serveurs du cluster

# Hadoop (Map Reduce) ?

-OPEN SOURCE (Fondation Apache) écrit en JAVA

-Inspiré de publications Google (2004)

‣ [Google Map Reduce](#)

‣ Google Filesystem

- Créé par Doug Cutting, salarié chez Yahoo

**3 distributions HADOOP (initialement Linux)avec des outils**

**1) Cloudera (Impala)**

**2) Hortonworks (version Windows)**

**3) MAPR (performance HDFS)**

# MAP REDUCE

## (Google 2004)

Map Reduce est à l'origine une technique de programmation connue de longue date en programmation fonctionnelle avec 2 grandes Fonctions :

MAP : transformation entrée en couples CLE/VALEUR

REDUCE : agrégation des valeurs pour chaque clé

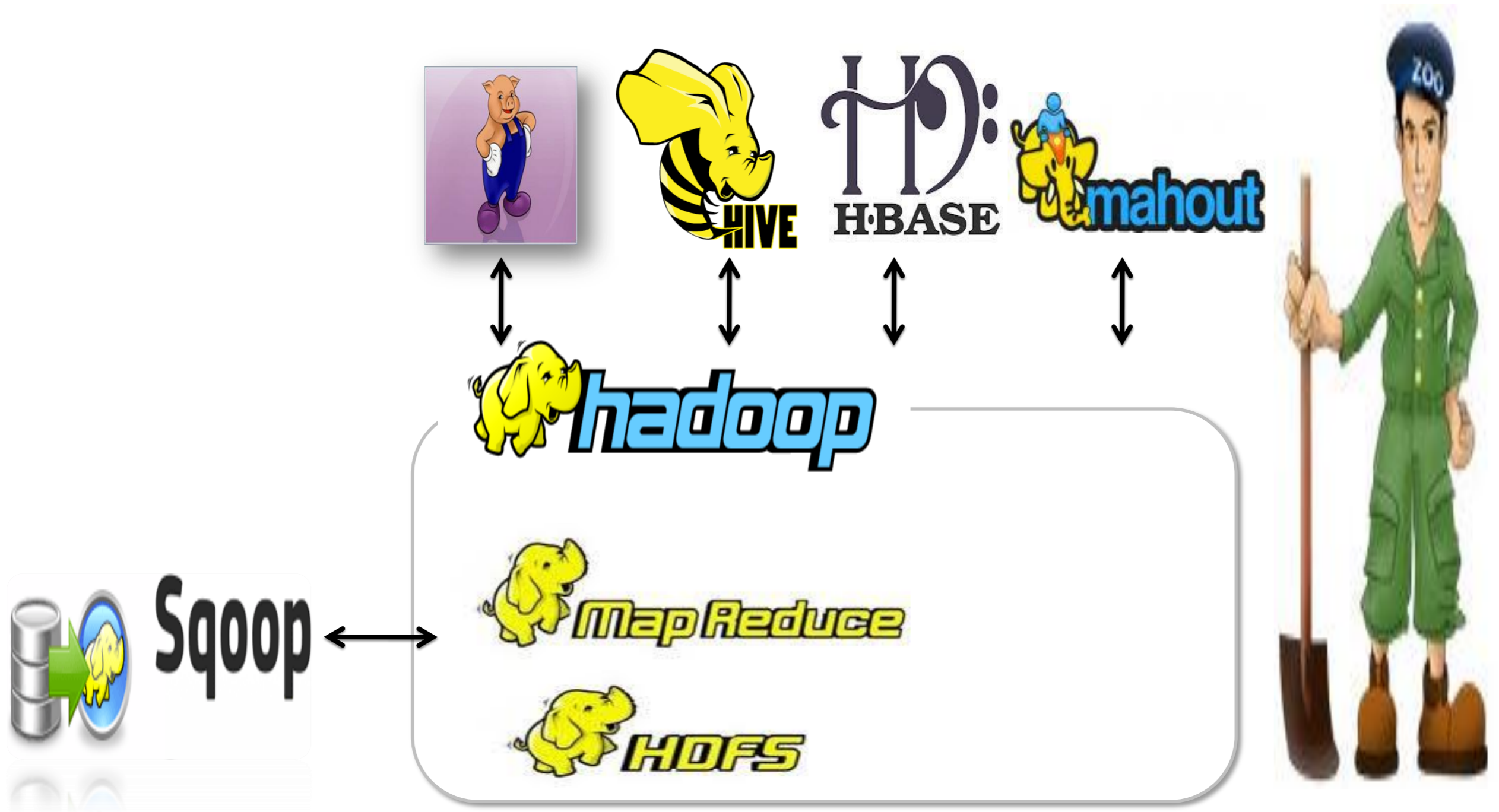
*< Un PROG Complexe peut être décomposé en une succession de taches Map Reduce >*

implantations en open source:

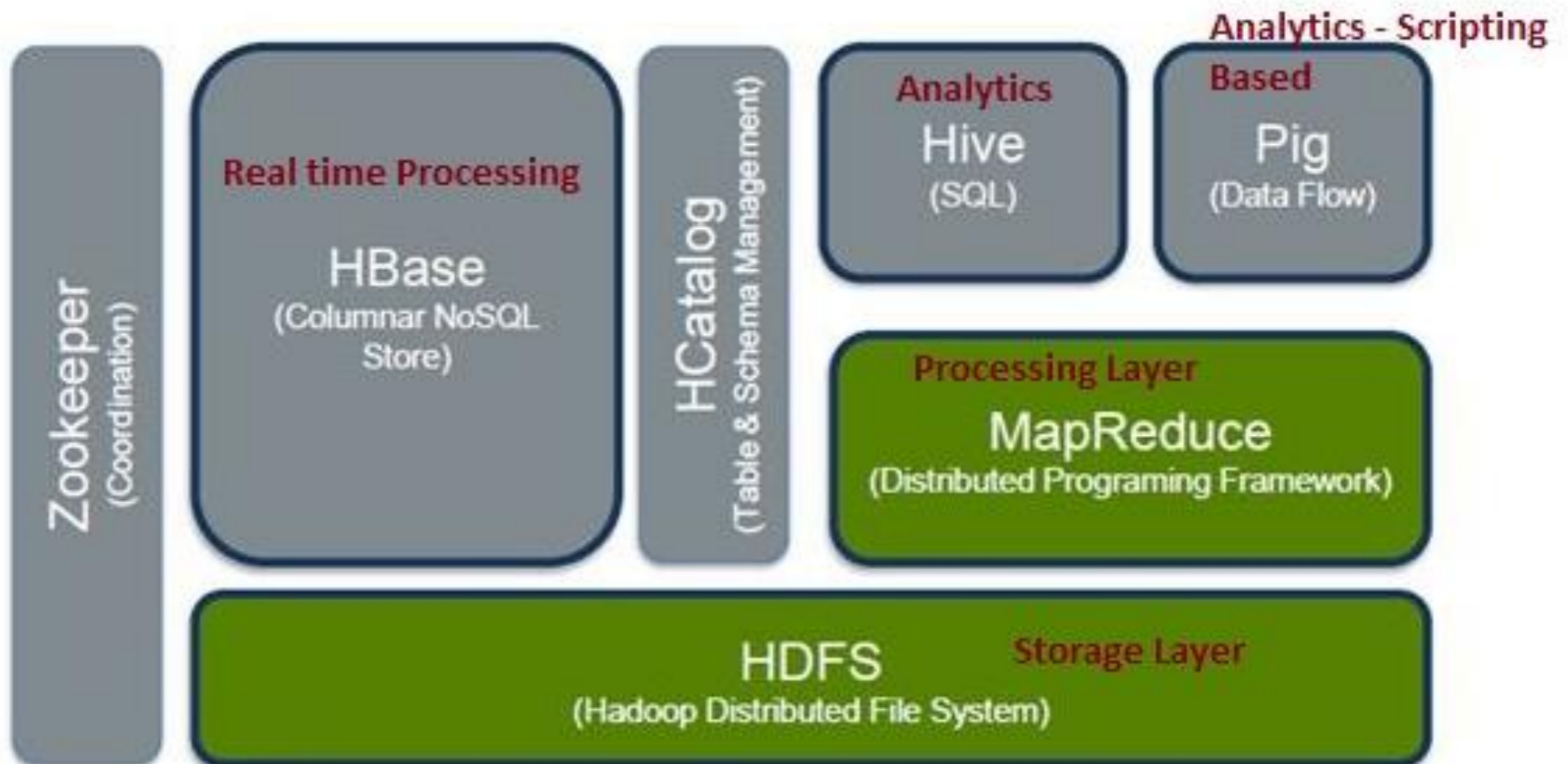
Hadoop (Yahoo! puis Fondation Apache),  
Disco (Nokia), MrJob (Yelp!), etc.

*Autres implantations de MapReduce intégrées dans les bases de données No SQL: CouchDB, MongoDB, Riak, ...*

# L' écosystème Hadoop



# Ecosysteme HADOOP





# *Hadoop MapReduce*

L'architecture MapReduce est composée de :

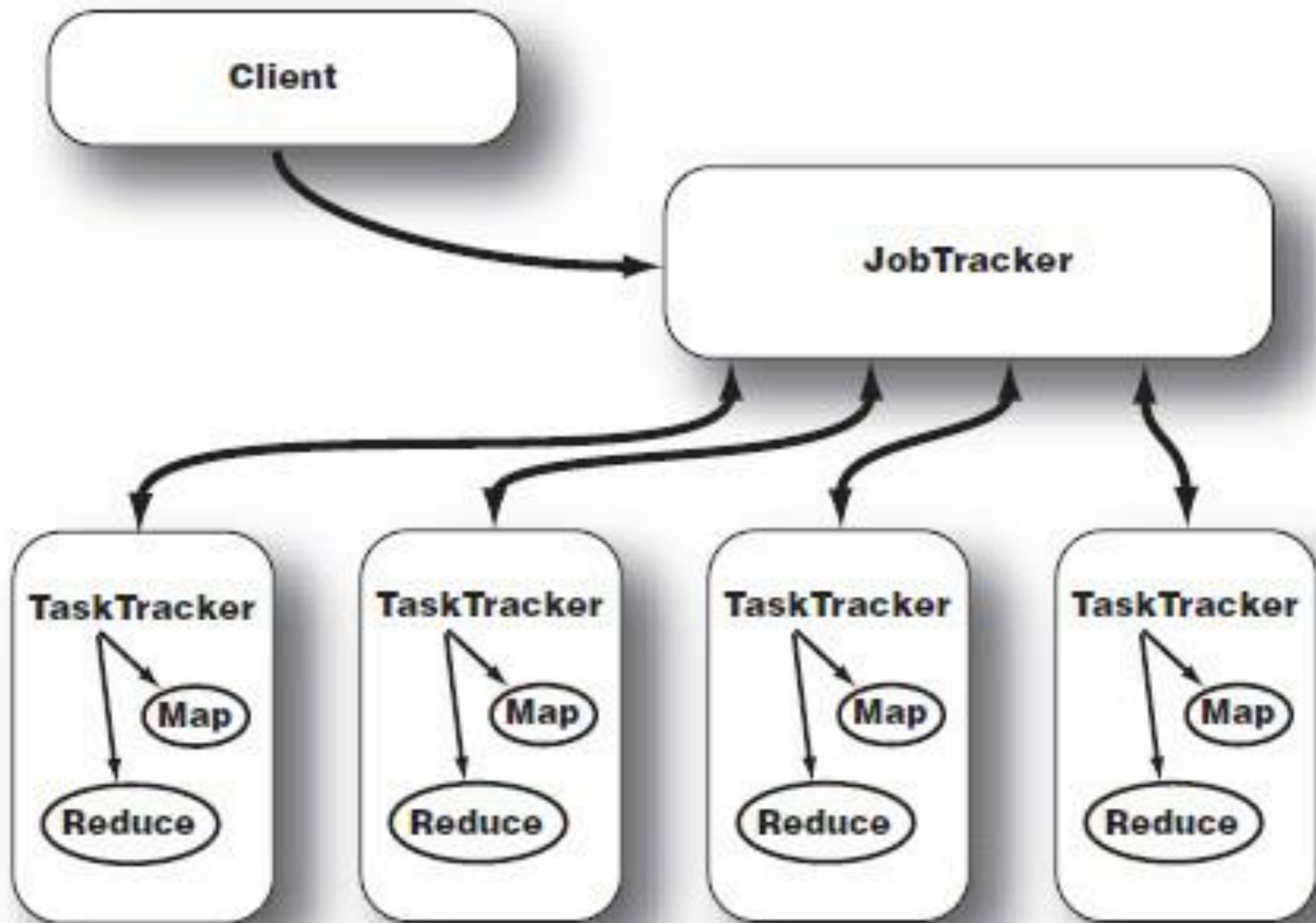
- Un **JobTracker** : centralisateur de tâches
- Des **TaskTracker** qui se chargent d'exécuter les travaux MAP REDUCE demandés.

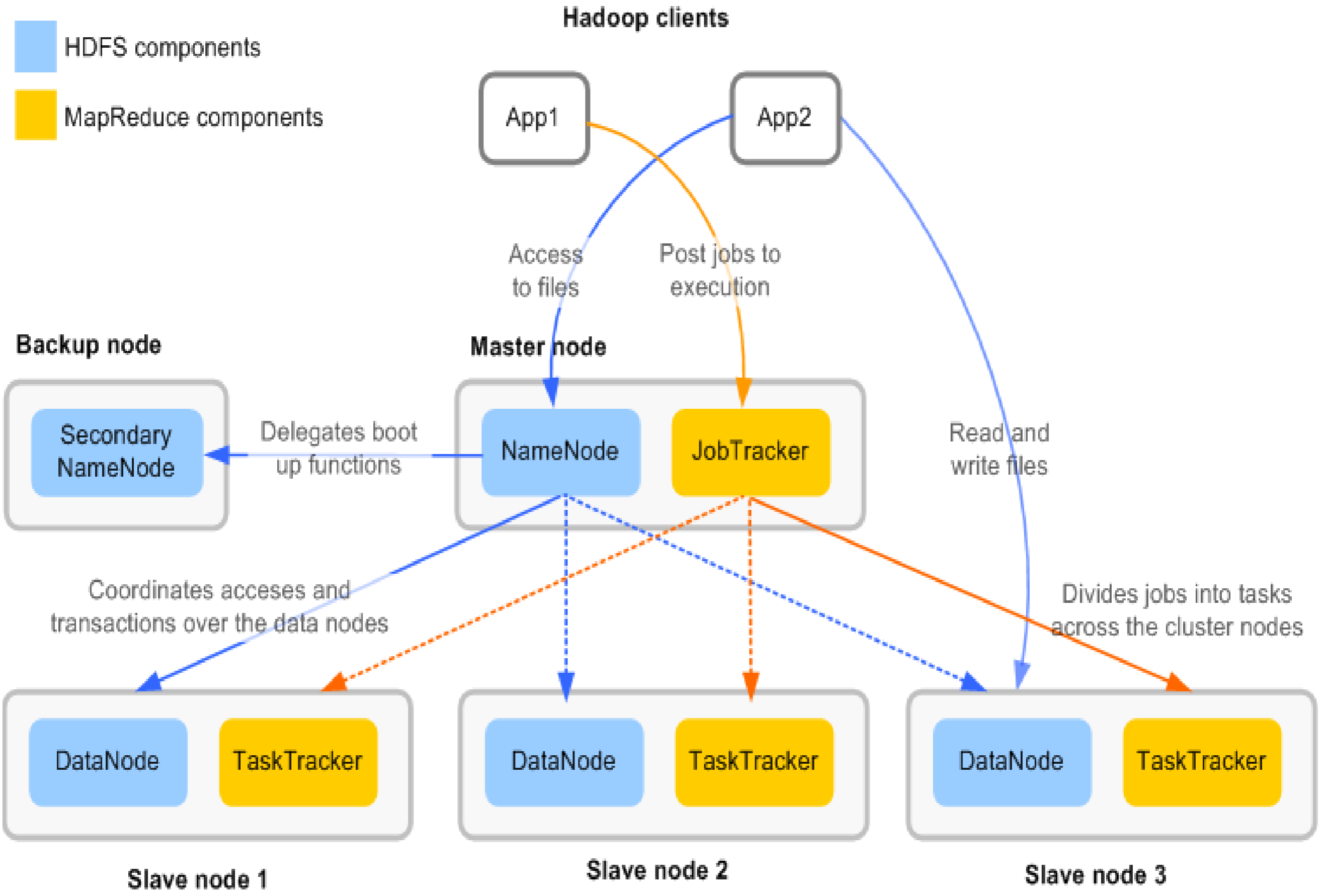
# HDFS (Hadoop)



- Système de fichiers distribué
- Traitement de grands volumes de données
  - Découpage des fichiers par blocs
- Fonctionne sur des serveurs “low cost” (au minimum 3)
- Fault Tolerant et Scalable
  
- **NameNode** : gestion des métadonnées
- **DataNode** : stockage des données







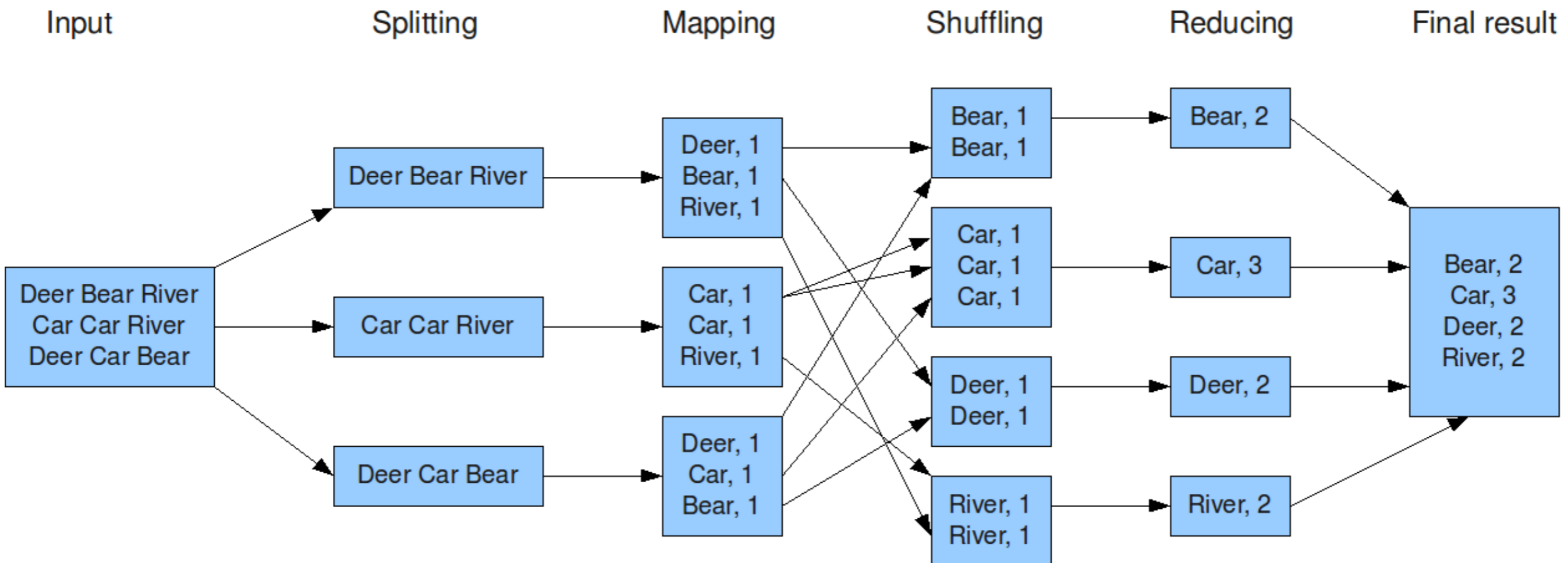
# Etapes d'un traitement MAP REDUCE

Les différentes étapes :

- Découper les données d'entrée (« *splitting* ») en « morceaux » parallélisables
- *Mapper* chacun des « morceaux » pour produire des valeurs associées à des clefs.
- Grouper/TRIER (« *shuffling* ») ces couples clef-valeur par clef.
- *Réduire* (*Reduce*) les groupes indexés par clef en une forme finale, avec une valeur pour chaque clef.

# Exemple MAP REDUCE

(Comptage des mots d'un texte)



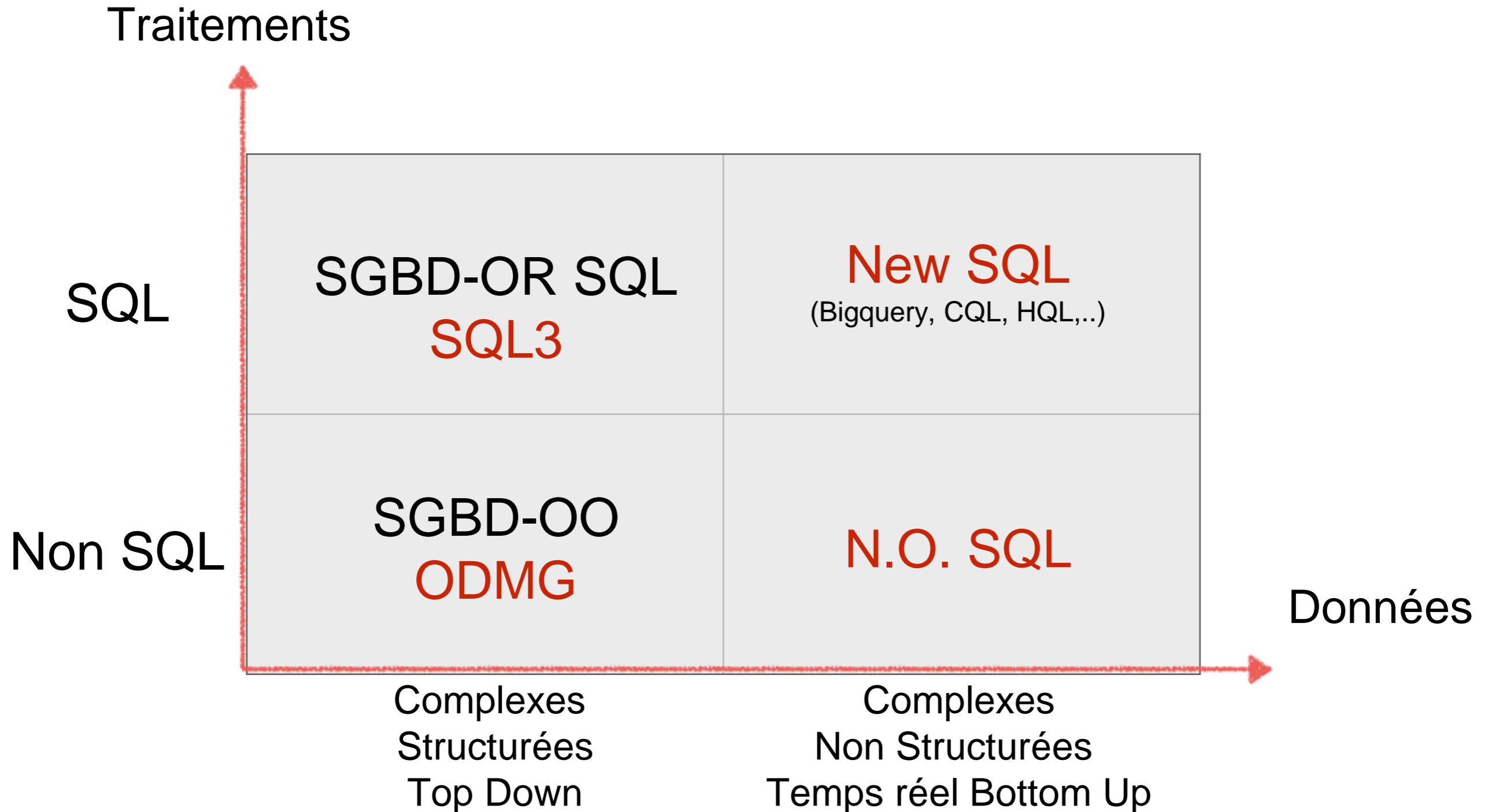
# Les 2 approches (complémentaires) de systèmes BIG DATA

	SQL	NOSQL
<b>type de données</b>	<b>structurées (schéma)</b>	<b>non-structurées (pas de schéma)</b>
Volume	TERA Octet ++	PETA ++ et EXA OCTET ++
Velocity/Variab.	Non	Oui
Transactionnel	Oui (pté ACID)	Non (pté BASE et Th. CAP)
Scalability	Verticale ( <b>Scale UP</b> )	Horizontale ( <b>Scale OUT</b> )
User Interface (JOINS)	Oui (SQL et JOINS)	Non (machine learning,)
Standards (Paradigmes)	SQL3/ODMG (POINTEUR-VALEUR)	(CLE/VALEUR)

# Les 2 approches (complémentaires) de systèmes BIG DATA

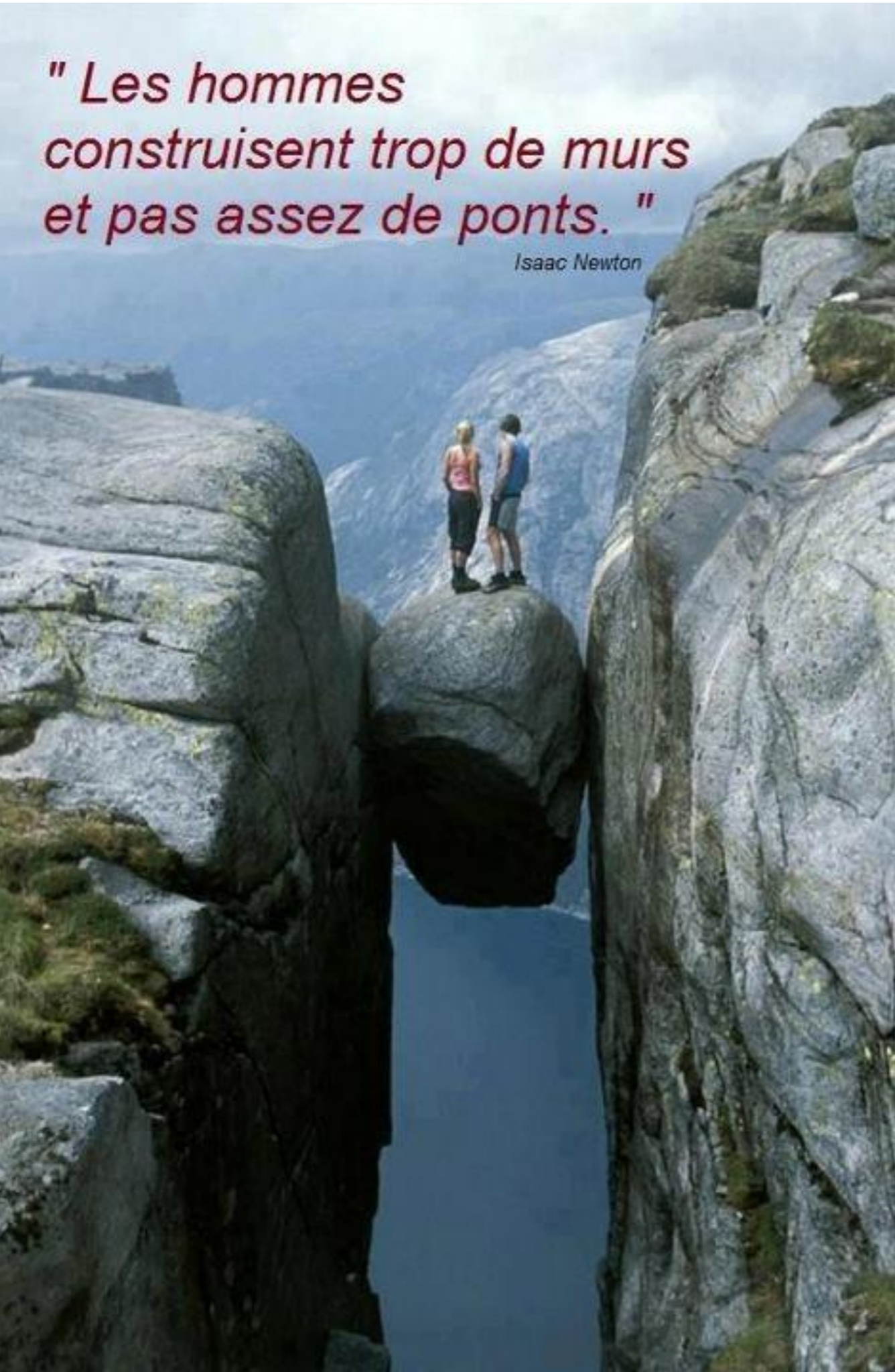
	SQL	NOSQL
AD HOC queries	Oui (SQL)	Non (cf. HIVE ou Bigquery de Google ou CQL de Cassandra)
Admin (DBA)	Oui	Non
Vendor Support.	Oui	Non (Open Source)
Static DATA	Oui	Non (Data flowing)

# Données « COMPLEXES » du BIG DATA : SQL3, NO SQL et NEWSQL? (MIRA2014)



*" Les hommes  
construisent trop de murs  
et pas assez de ponts. "*

*Isaac Newton*



# NEW SQL ?

**PASSERELLES**  
**SQL et NO SQL (Hadoop/ MAP**  
**REDUCE)**  
*(SQL et WEB SEMANTIQUE)*



## « Du N.O. SQL au NEW SQL »

*“Replacing real SQL ACID with either no ACID or “ACID lite” just **pushes consistency problems into the applications where they are far harder to solve.** Second, the absence of SQL makes queries a lot of work”*

M.Stonebraker

***Avec HADOOP, l’administrateur est ...l’utilisateur !***

# Verrous Metissage

*Systemes amphibiens* : Passerelles entre SGBD/Datawarehouse TOP DOWN (SQL) et décisionnel/DATA analytics BOTTOM UP (NOSQL)

- Maintien ACID approche SQL
- Interface interactive SQL++
- Maintien Performances *bottom up* et scalabilité approches NOSQL

→ « NEW SQL »

« *From NO SQL to NEW SQL* »  
[RICH2012] ([STON2011])

**“NEW SQL” (on top of SQL) :**

VoltDB de Stonebraker

**MYSQL**

Scale DB, Clustrix, AKIBAN

NUODB (NimbusDB),

+ TERADATA BIG DATA, Oracle BIG DATA, Microsoft  
BIG DATA, IBM Big Data...

***“Future is polyglot persistence”***

# VOLTDB (Stonebraker 2011)

JUIN 2012 : VOLTDB traite 686 000 Transactions par seconde sur le Cloud Amazon

Exercice : aller sur le site de VOLTDB [www.voldtb.com](http://www.voldtb.com) et identifier 5 differences fondamentales entre VOLTDB/ NEWSQL et une base de données SQL

# Changements OLTP dans les 25 dernières années

## **La plupart des systèmes OLTP peuvent contenir en mémoire centrale**

- 1 BD OLTP d'un TERA Octets dans un simple cluster de 32 noeuds avec 32 gigas/noeud
- 2013 :4 ordres de magnitude plus important en tps (1000 tps en 1985)
  - Oracle 11G (TIMES10) en Juin 2014 : 8,7 million de TPS (débit/crédit)

## *BIG DATA nécessite :*

- *Haut volume/flux de données (velocity)*
- *Analyse de données TEMPS REEL*

# NEW SQL ?

## OBJECTIFS

- Conserver interactivité SQL (Note : Cassandra et Mongo DB intègrent SQL et Bigquery de Google est un dialecte..propriétaire)
- Conserver les propriétés ACID
- Nouvelles architectures SGBD

Architectures traditionnelles des SGBD avec 90% d' overhead pour Big Data

- *Disk based and buffer pool overhead*
  - *Dynamic tuple locking*
  - *Active/passive replication*
  - *WAL*
  - *Multi-threaded*
- 
- *Exercice : Cherchez les définitions des concepts en italique*

# Architecture de VOLTDB

NO BUFFER POOL, NO LOCKING (ordonnancement par timestamp),  
NO WAL (no DATA Log : stored procedure log), No Threading  
overhead

Shared- nothing architecture (cf LAN)

BASE DE DONNEES en MÉMOIRE CENTRALE

*Single threaded*

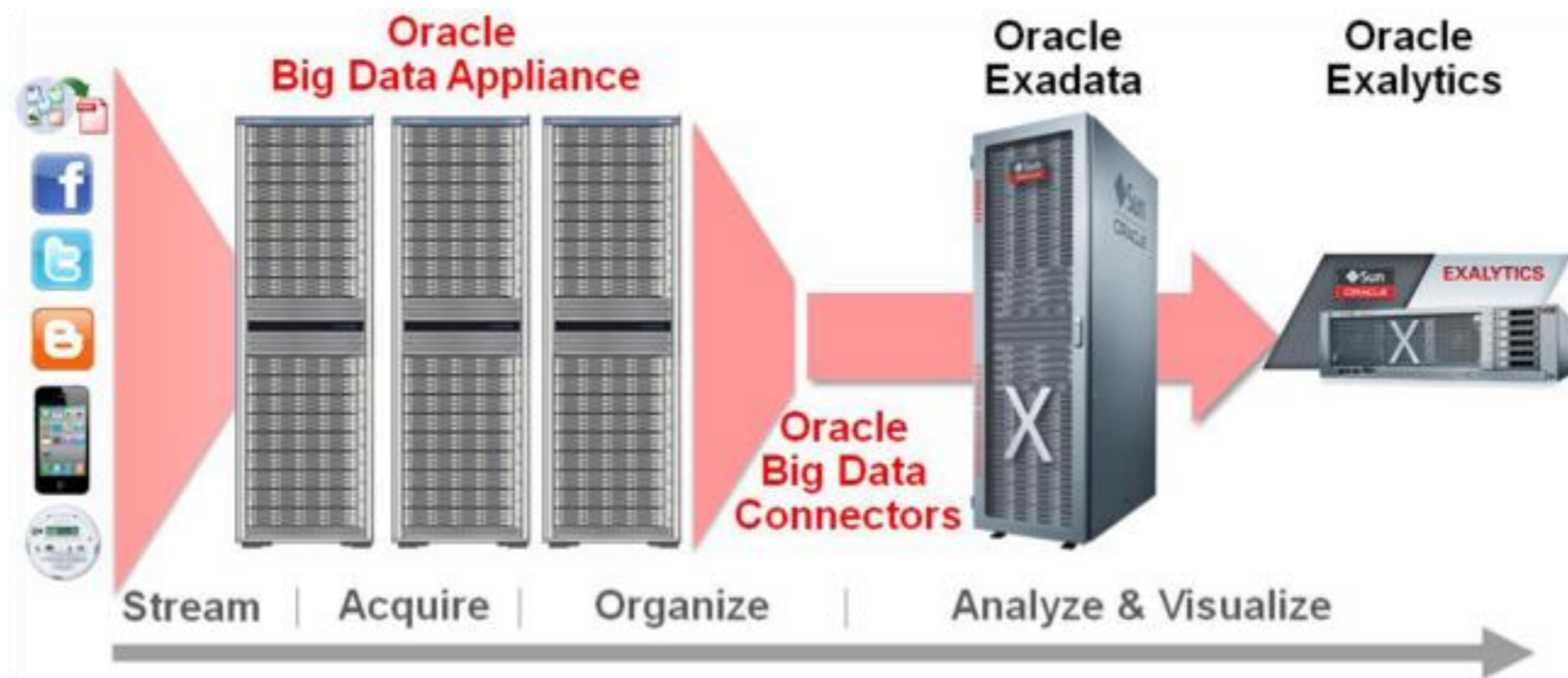
- *No shared data*
- *Main memory divided per core*

*Open Source*

- *Cheaper sales model (100 downloads a day in 2013)*

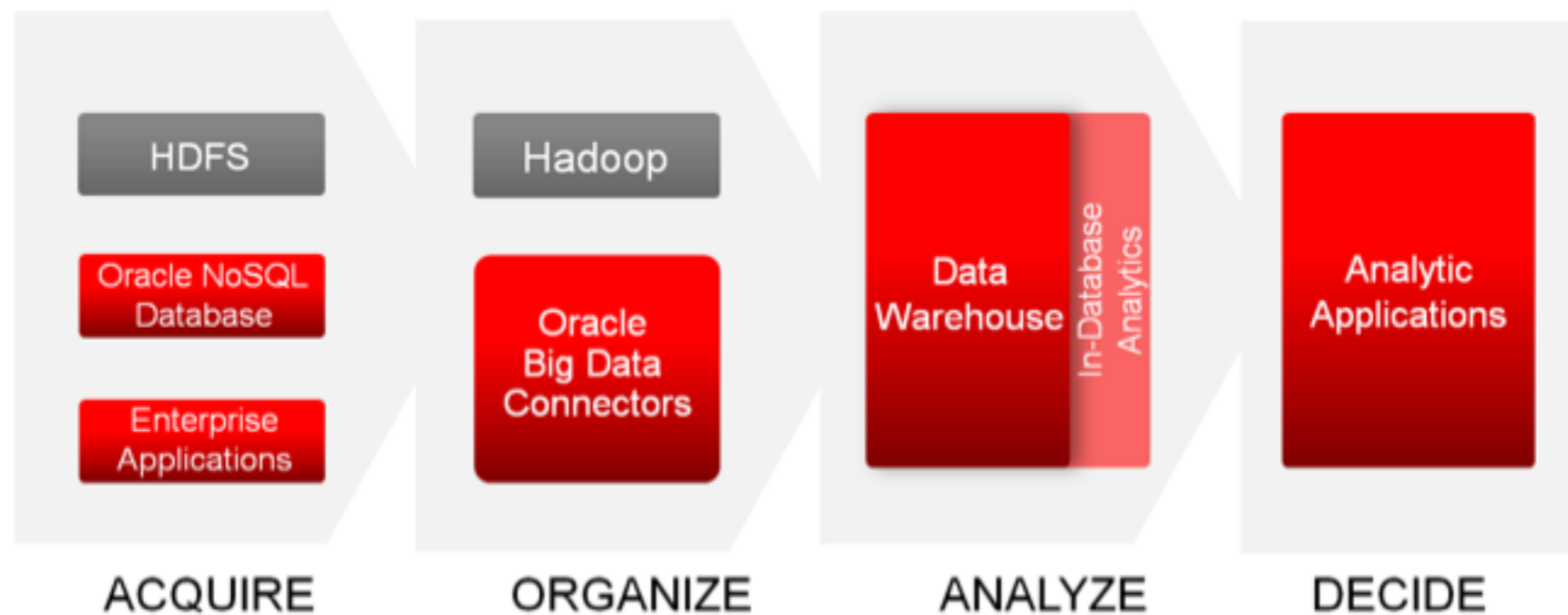
- *Exercice : Cherchez les définitions des concepts en italique*

# BIG DATA d'Oracle pour l'entreprise [ORACLE2012]

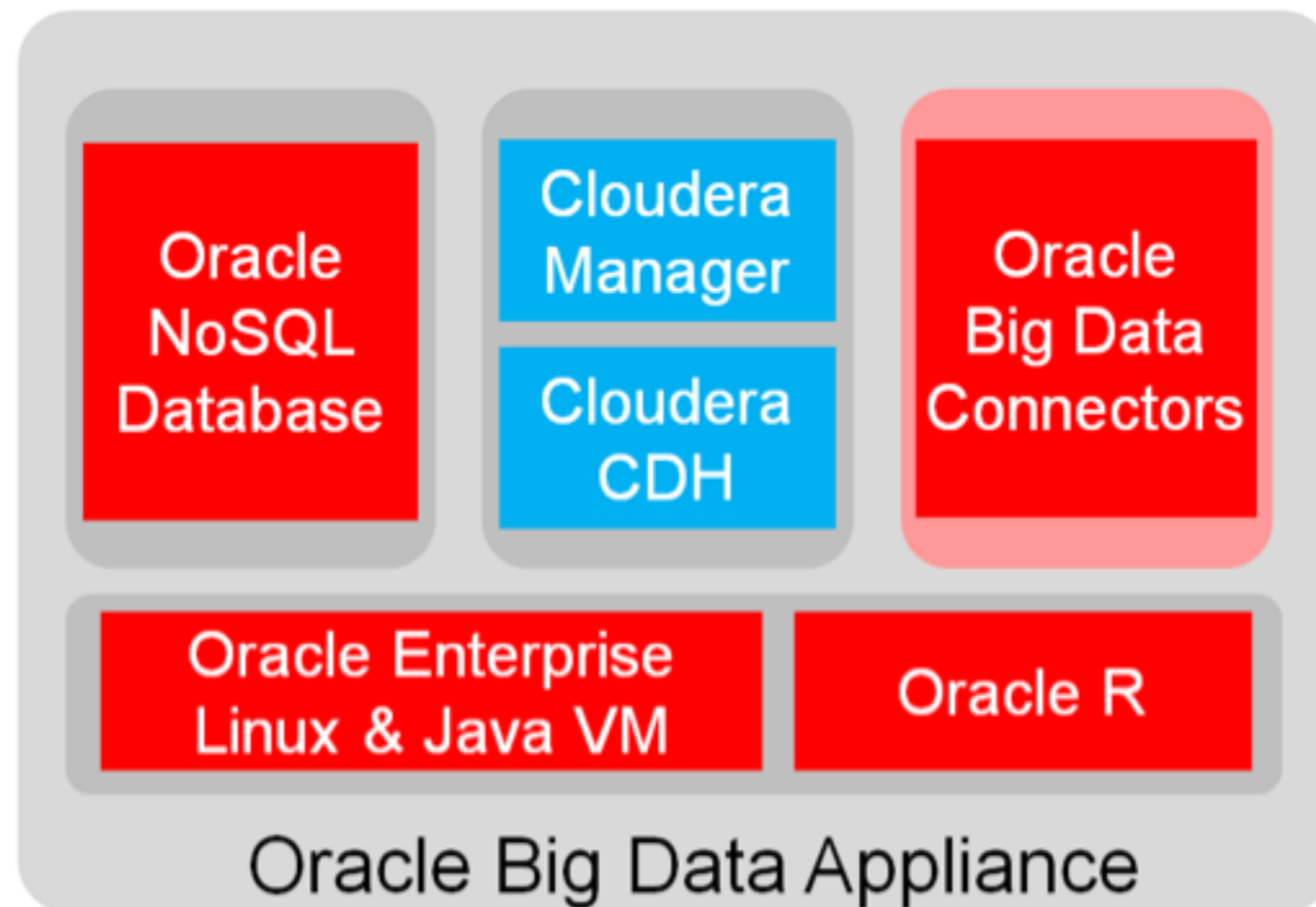




# Solutions BIG DATA d' Oracle : connecteurs avec HADOOP



# Oracle Big Data Appliance (HADOOP/Cloudera)



# CTO de Teradata Stephen Brobst (Oct 2012)

*« Désormais, vous pouvez bénéficier de la puissance de MapReduce et de la facilité d'usage de SQL ,...  
Avant, avec Hadoop, les seules personnes capables d'extraire des données étaient celles qui les avaient placées »*

# Big Data & Teradata

« Unified Data Architecture », avec intégration Hadoop.

– **système de fichiers HDFS** (Hadoop Distributed File System), au moyen du langage de requêtage SQL, un langage très familier dans le monde des bases de données.

– **HCatalog**, un framework de métadonnées Open Source développé par Hortonworks,

et « **SQL-H** », qui permet d'analyser des données stockées sur un filesystem HDFS en utilisant SQL.

- **ASTER**, propriété de Teradata, avait inventé et breveté **SQL-Map Reduce** », qui greffe à SQL des fonctionnalités de Map Reduce.

- L'appliance **Teradata-Aster Big Analytics** plus de 50 applications analytiques pré-intégrées.

• *15 petabytes de données, réparties entre les deux bases, (2012)*

**Aster SQL-H**

**Hadoop  
MapReduce**

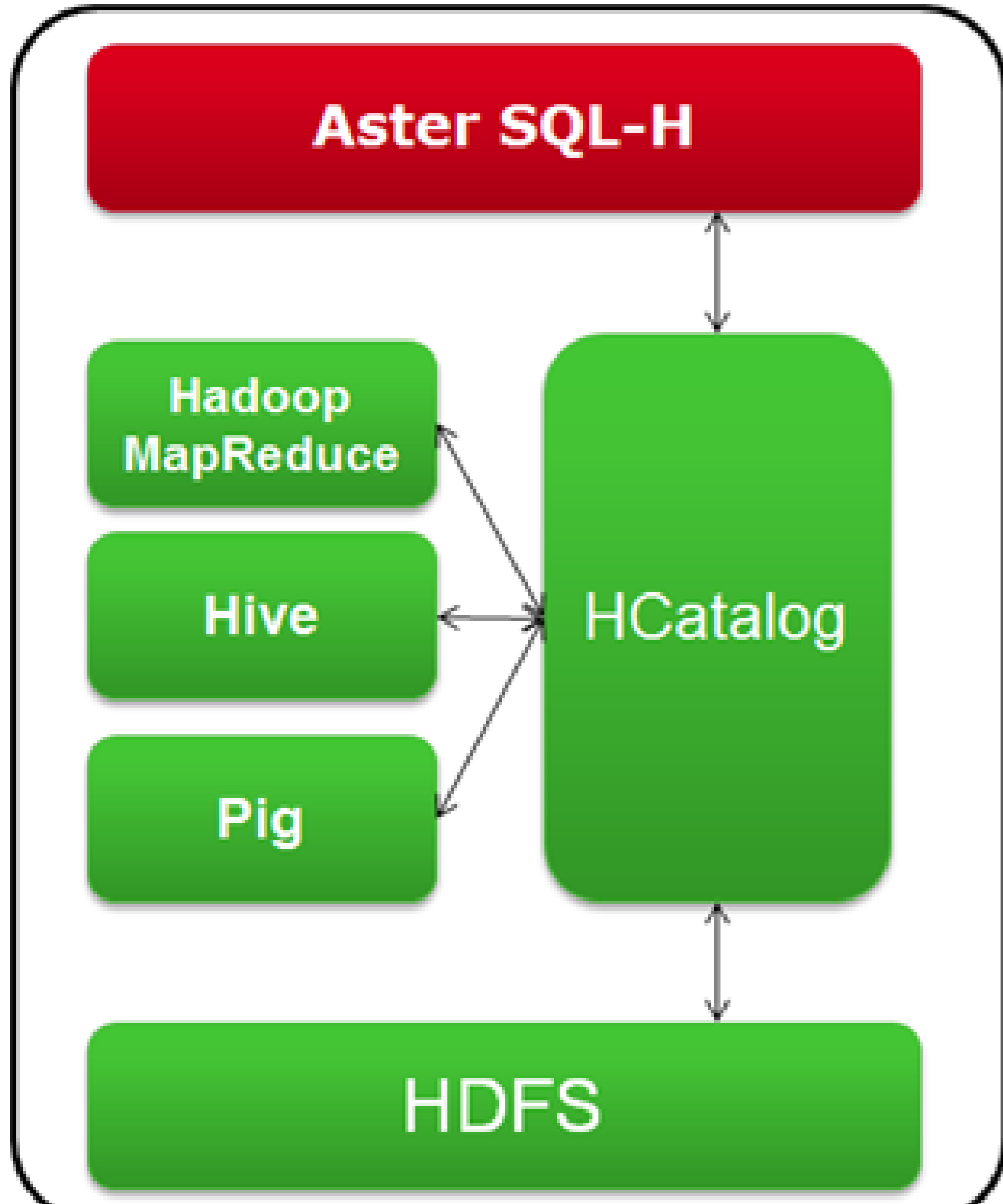
**Hive**

**Pig**

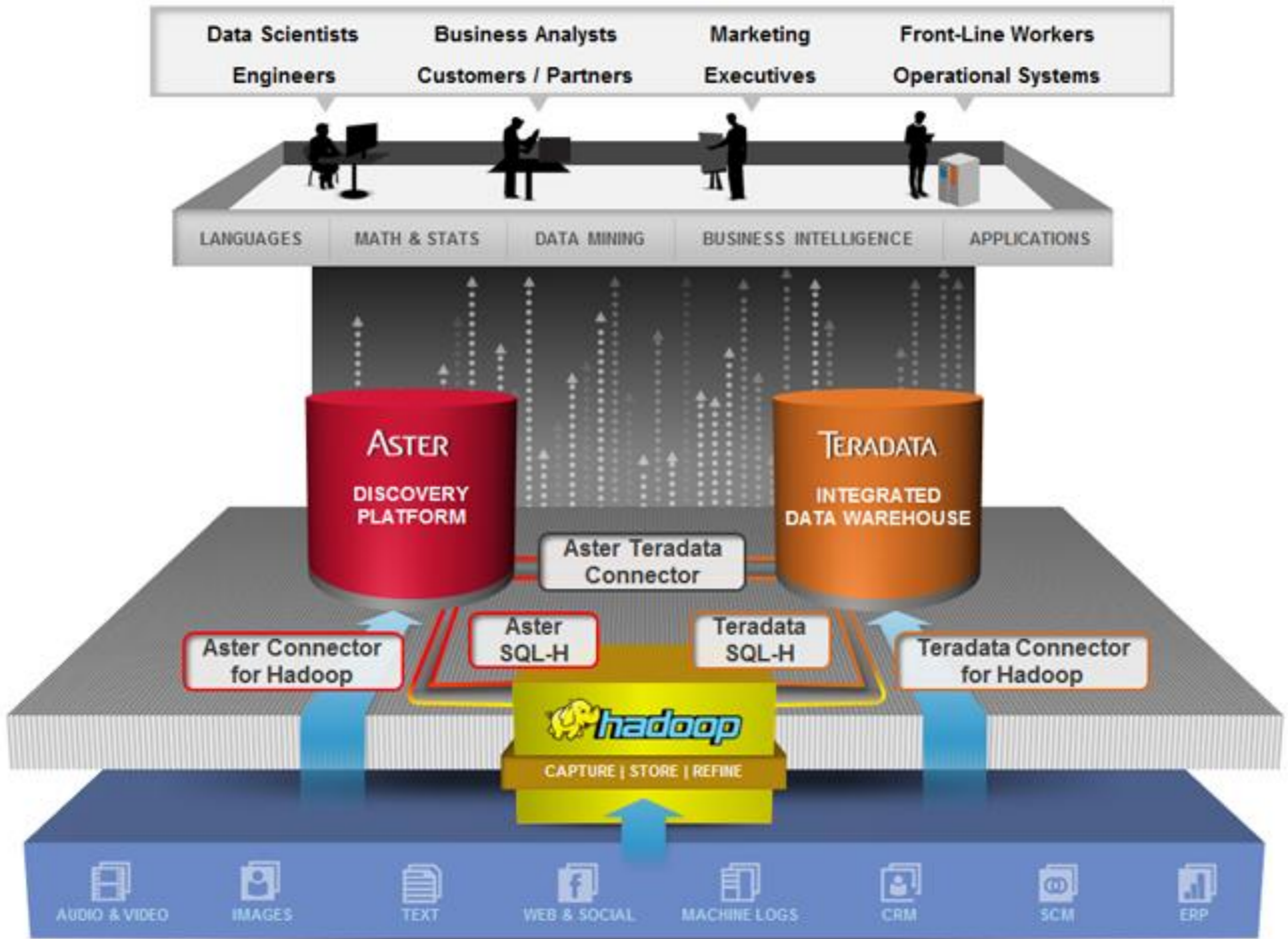
**HCatalog**

**HDFS**

SQL-H de Teradata



# SQL-H Teradata



# TERADATA :

## SQL- MAP REDUCE

```
SELECT ...  
FROM functionname(  
    ON table-or-query  
    [ PARTITION BY expr... ]  
    [ ORDER BY expr... ]  
    [ clausename ( arg... ) ]...  
)
```

...

# Big Data dans SQL SERVER de Microsoft

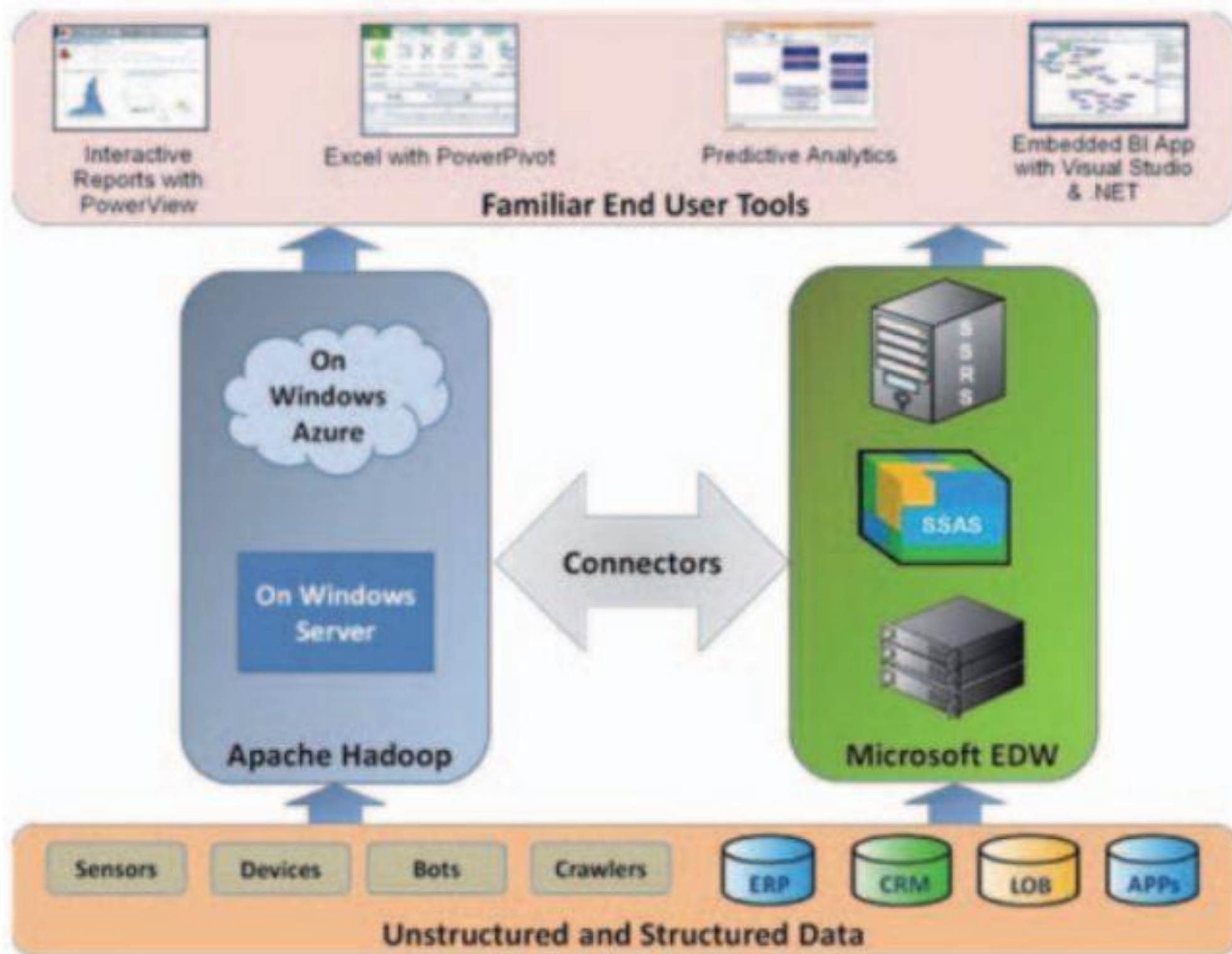
SQL SERVER intègre la composante [Hadoop](#),

## **Interface Excel à Hadoop**

le projet [Apache Sqoop](#),

la mise à disposition de [Mahoot](#) (outils de datamining pour Hadoop)





**Aperçu de la solution Microsoft Big Data**

# Apports du BIG DATA NO SQL (complémentaires SQL):

« **WHAT!** »

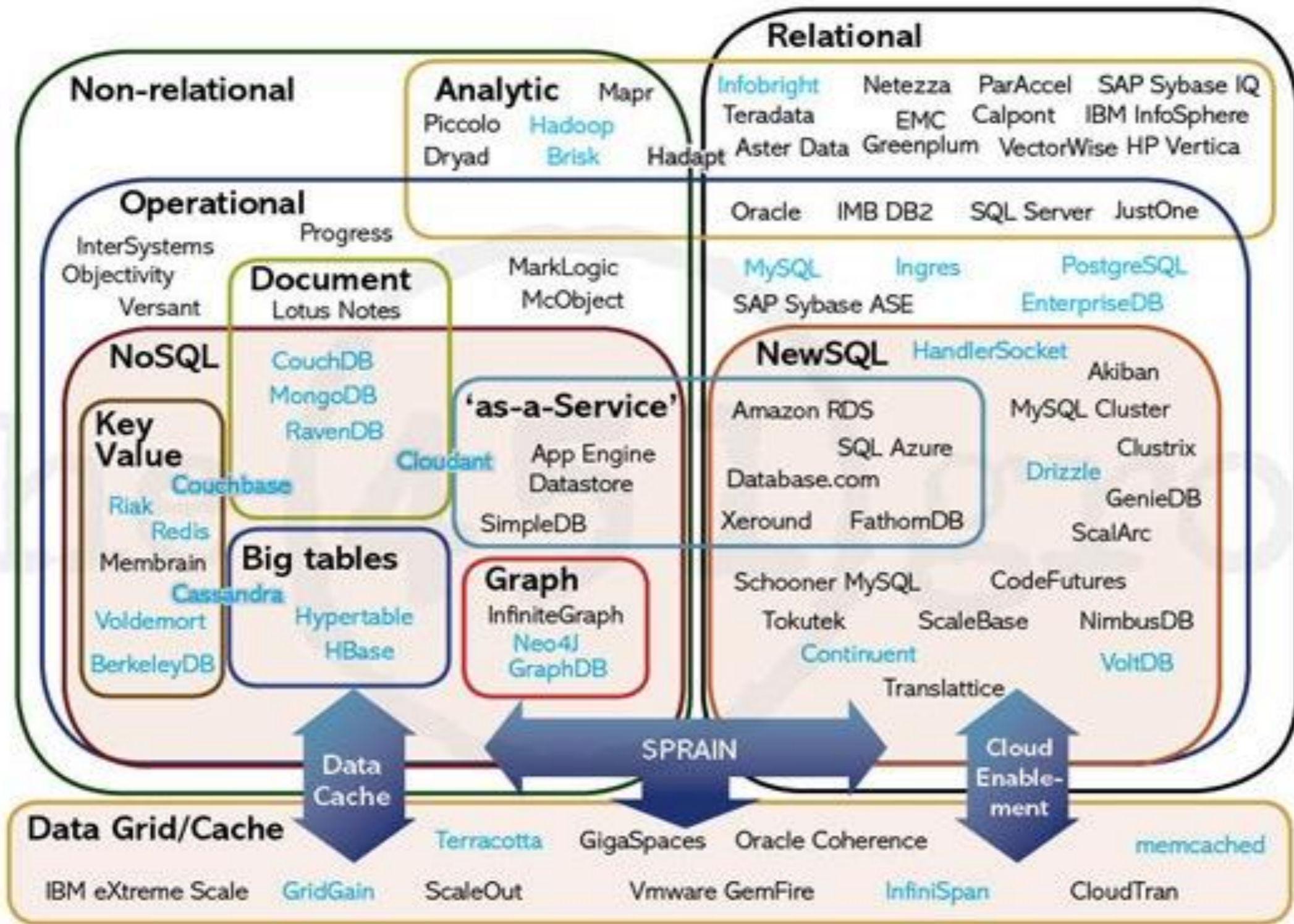
Web DATA : « Open Data », Données des Réseaux sociaux, DATA mobiquitaires, « Linked DATA »/  
« Semantic web » ( [paradigme RDF](#), SparQL, OWL)

Hadoop (Hive, Hbase)/map-reduce ([paradigme CLE  
VALEUR](#))

Analytics (« [couple](#) » *Big Data pour IBM*) / Anti Schema et  
Anti-ACID (BASE, CAP Theorem)

TEMPS REEL, non structuré et bottom up

# Les systèmes de Données (DATA Systems) ! (Aslett, 2013)



# DATA SYSTEMS / Systèmes de Gestion des Données *Massives/mobiquitaires* (SGDM) et *paradigmes associés*

Modèle relationnel de Codd  
Paradigme « Valeur »



SQL3, SQL3/ODMG  
NEW SQL

Modèle « OBJET »



paradigme « POINTEUR-VALEUR »  
(SQL3)  
paradigme « OBJET-VALEUR (ODMG)



SPARQL  
(OWL)

N.O. SQL



paradigme « RDF »  
(Web Sémantique)

paradigme « CLE-VALEUR »  
(Map Reduce)

... et GOOGLE ?

# DREMEL (2006)

Un moteur de requêtes réparti avec parallélisme (SCALE OUT) Utilisé par GOOGLE depuis 2006

BD Orientée COLONNE

Dialecte SQL d'interface depuis 2012 : BIGQUERY (permettant un accès externe par des tiers)

Exécution en parallèle sur des milliers de machines

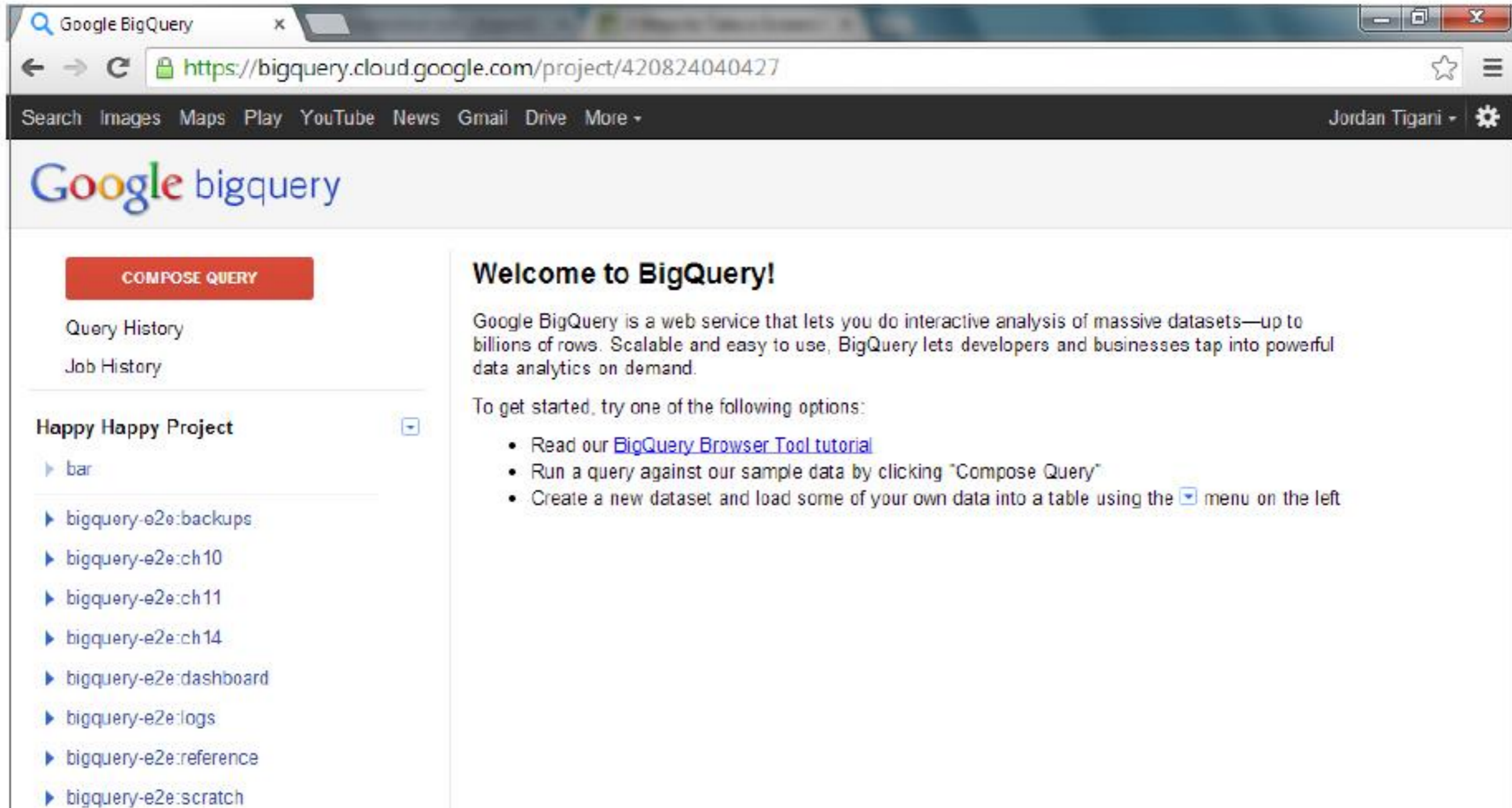
*>n (100 000 disques)*

*> p (10 000 processeurs) < SCALE OUT>*

*50 giga OCTETS/sec avec rep < 5 sec <VIRTUAL CLUSTER unit>*

# Bigquery : un service WEB

Partie de la plateforme CLOUD de Google



The screenshot shows a web browser window with the Google BigQuery interface. The browser's address bar displays the URL `https://bigquery.cloud.google.com/project/420824040427`. The page header includes the Google logo and the text "bigquery". On the left side, there is a navigation menu with a red "COMPOSE QUERY" button at the top, followed by "Query History" and "Job History". Below this, a dropdown menu is open for "Happy Happy Project", showing a list of datasets: "bar", "bigquery-e2e:backups", "bigquery-e2e:ch10", "bigquery-e2e:ch11", "bigquery-e2e:ch14", "bigquery-e2e:dashboard", "bigquery-e2e:logs", "bigquery-e2e:reference", and "bigquery-e2e:scratch". The main content area features a "Welcome to BigQuery!" heading, a paragraph describing the service, and a list of three options to get started: reading a tutorial, running a query, or creating a new dataset.

Google BigQuery

<https://bigquery.cloud.google.com/project/420824040427>

Search Images Maps Play YouTube News Gmail Drive More

Jordan Tigani

## Google bigquery

**COMPOSE QUERY**

Query History  
Job History

### Happy Happy Project

- ▶ bar
- ▶ bigquery-e2e:backups
- ▶ bigquery-e2e:ch10
- ▶ bigquery-e2e:ch11
- ▶ bigquery-e2e:ch14
- ▶ bigquery-e2e:dashboard
- ▶ bigquery-e2e:logs
- ▶ bigquery-e2e:reference
- ▶ bigquery-e2e:scratch

## Welcome to BigQuery!

Google BigQuery is a web service that lets you do interactive analysis of massive datasets—up to billions of rows. Scalable and easy to use, BigQuery lets developers and businesses tap into powerful data analytics on demand.

To get started, try one of the following options:

- Read our [BigQuery Browser Tool tutorial](#)
- Run a query against our sample data by clicking "Compose Query"
- Create a new dataset and load some of your own data into a table using the  menu on the left

# « BIGQUERY SQL »

## NI-NI

- Ni OLTP ni OLAP
- Ni Relationnel Ni NOSQL

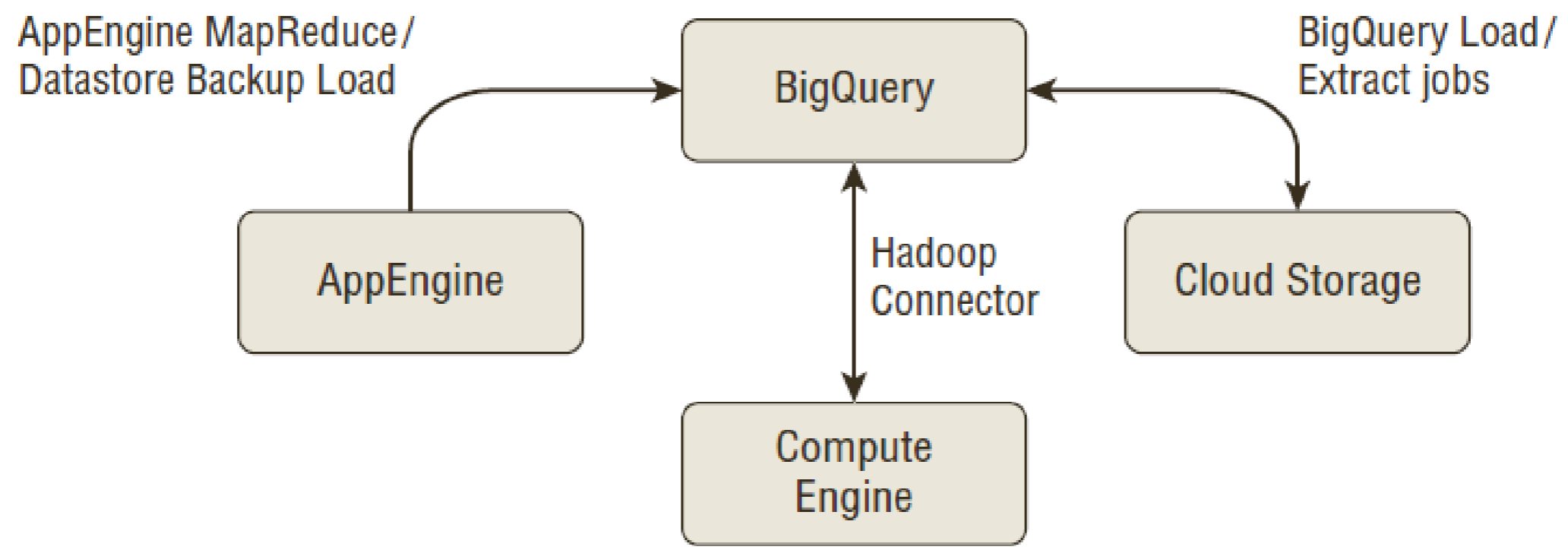
Pas le standard SQL mais un **dialecte propriétaire**

Pas OPEN SOURCE : système propriétaire GOOGLE de stockage orienté COLONNE

 CLOUD Based : Web Browser

+ *GFS (devenu Colossus; distributed File System) + BIGTABLE*





# Benchmark Wikipedia de Bigquery en 2012

2011 : Wikipedia bigquery-sample : wikipedia\_benchmark

*-[Exemples Bigquery SQL tirés de TIGANI2014]*

```
SELECT language SUM (views) AS views  
From [bigquery_samples: wikipedia_benchmark<size> <taille de 1K à ½ Tera>  
WHERE REGEXP_MATCH (title, « G.*O.*O.*G. »)  
GROUP By Language  
ORDER BY Views Desc
```

Compter le mot « KING » dans l'Œuvre de  
SHAKESPEARE  
*(publicdata:samples.shakespeare)<1,6 GB*

```
SELECT LOWER (Word) AS word, word_count AS frequency,  
corpus  
FROM [publicdata:samples.shakespeare]  
WHERE corpus CONTAINS 'king' AND LENGTH (Word)  
> 5  
ORDER BY frequency DESC  
LIMIT 10
```

# NO SQL/HADOOP vs BIG QUERY

## HADOOP :

Orientation BATCH

Pas de Schéma

Pas d Interface interactive SQL (écrire prog en Java)

Open Source

## BIG QUERY

Orientation INTERACTIVE

Schéma

Dialecte SQL Propriétaire

## 4 Apports principaux de GOOGLE à BIG DATA : les propriétés **CABS**

**C**

- **C**LOUD based

**A**

- « **A**nalytics as a service » (AaaS)

**B**

- **B**igquery (sur moteur Dremel)

**S**

- **S**QL (*dialecte SQL*) **I**NTERACTIF



## En conclusion

Approche  
*TOP DOWN*  
des infrastructures  
et  
*BOTTOM UP*

des DATA et des  
SERVICES  
(avec SQL en  
référence)

**CONCLUSION : « *Tribe of IMAGINATION warriors* » (Franketienne, Haïti)**



# Quelques References Big Data

Dan Mc Greary, Ann Kelly « Making sense of NO SQL » Manning 2014

Jordan Tigani, Siddhartha Naidi « Google Bigquery Analytics » WILEY, 2014 (510 pages)

Ian Davis « 30 Minute Guide to RDF and Linked Data” 2009 , Slide Share

[ Mike Stonebraker, “New SQL: An Alternative to NoSQL and Old SQL for New OLTP Apps » ACM, Juin 2011

S. Miranda , « Systèmes d’information Mobiquitaires » Revue RTSI, Sept 2011

W.CHU Editor « Data mining and knowledge Discovery for big data » Springer 2014

F.Provost, T Fawcell « DATA SCIENCE for Business » O’Reilly 2013

ORACLE2012] White Paper Oracle, January 2012 « Oracle BIG DATA for the Enterprise »

An Oracle White Paper, “Oracle NoSQL Database”, September 2011,  
<http://www.oracle.com/technetwork/database/nosqldb/learnmore/nosql-database-498041.pdf>



**Extra slides**

# HADOOP

Les trois principaux acteurs de ce marché sont :

- **Cloudera**, avec la Cloudera Hadoop Distribution, actuellement en version 4 (CDH4), qui package Hadoop 2.0 ;
  - HortonWorks, qui package Hadoop 1.0.3 ;
- MapR, qui propose lui aussi une distribution autour de Hadoop 2.



## Different APIs to write Hadoop programs:

- ▶ A rich **Java** API (main way to write Hadoop programs)
- ▶ A **Streaming** API that can be used to write *map* and *reduce* functions in any programming language (using standard inputs and outputs)
- ▶ A **C++** API (Hadoop Pipes)
- ▶ With a **higher-language level** (e.g., Pig, Hive)

Advanced features only available in the Java API

# **BUSINESS INTELLIGENCE (BI) / DATA ANALYTICS and BIG DATA**

**Real time DATA analysis**  
**Agile Business Intelligence**  
Predictive data Analytics  
Mobile/mobiquitous BI  
« **Social data** » analysis  
***DYNAMIC*** reporting

# « *Agile BI* »\* (Business Intelligence) « *Mobile/Mobiquitous BI* »

« *Ubiquity and mobility are key features of DATA today* » \* < MOBIQUITY ☺ >

« *From STATIC decision-support report to DYNAMIC vizualization* »

**SaaS** : « **Software as a Service** »

**PaaS** : « **Platform as a Service** »

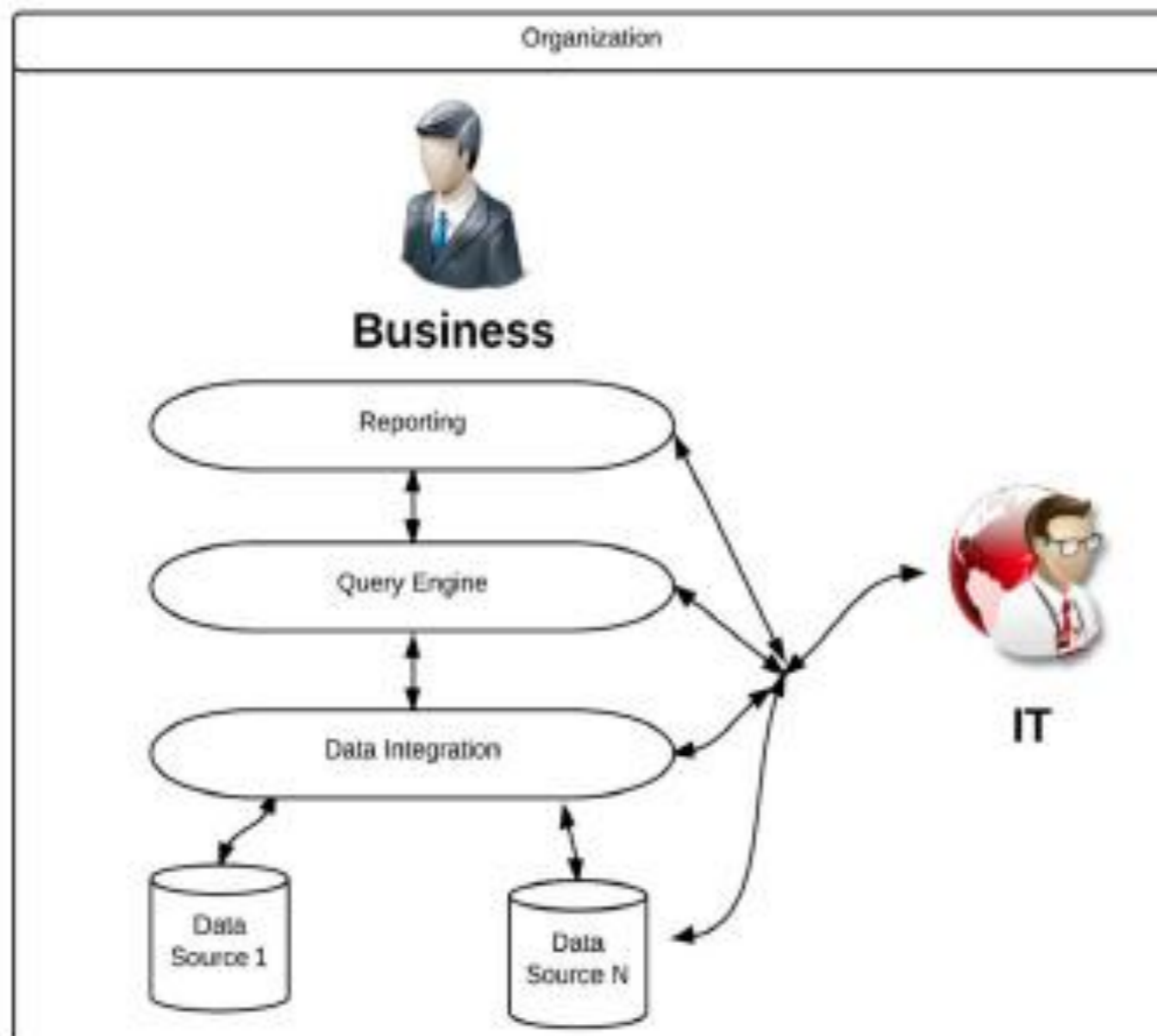
Ex: Azure de Microsoft ou Blue Mix d'IBM

**DATAaaSERVICE** demain ?

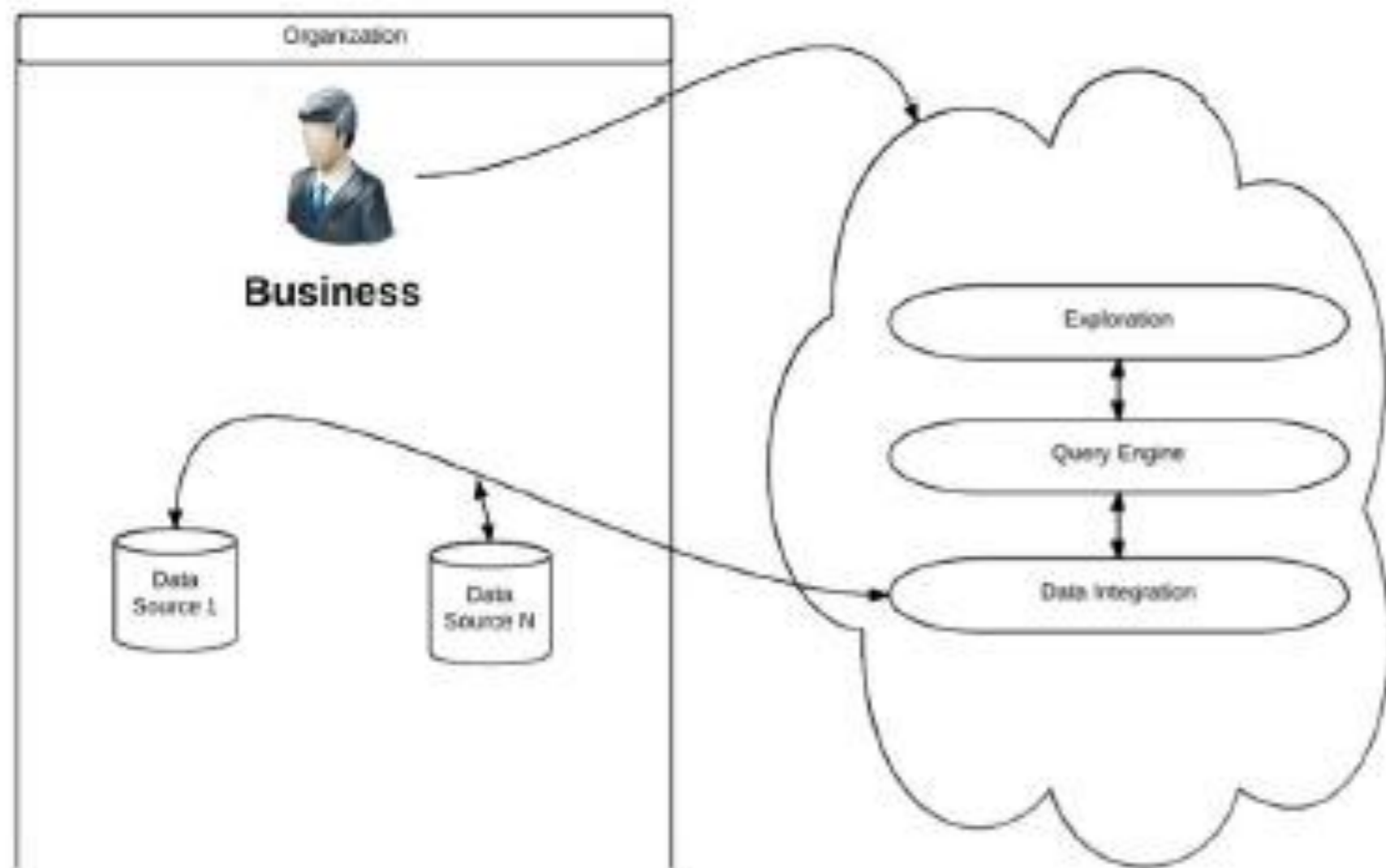
**CLOUD centrics** IT future

Gmail, Dropbox, Goggle Drive, SkyDrive, Facebook, Youtube, Flickr, Instagram, ..

\*« *Agile Business Intelligence : reshaping the landscape* » G. Anadiolis (Oct 2013; GIGAomPRO)

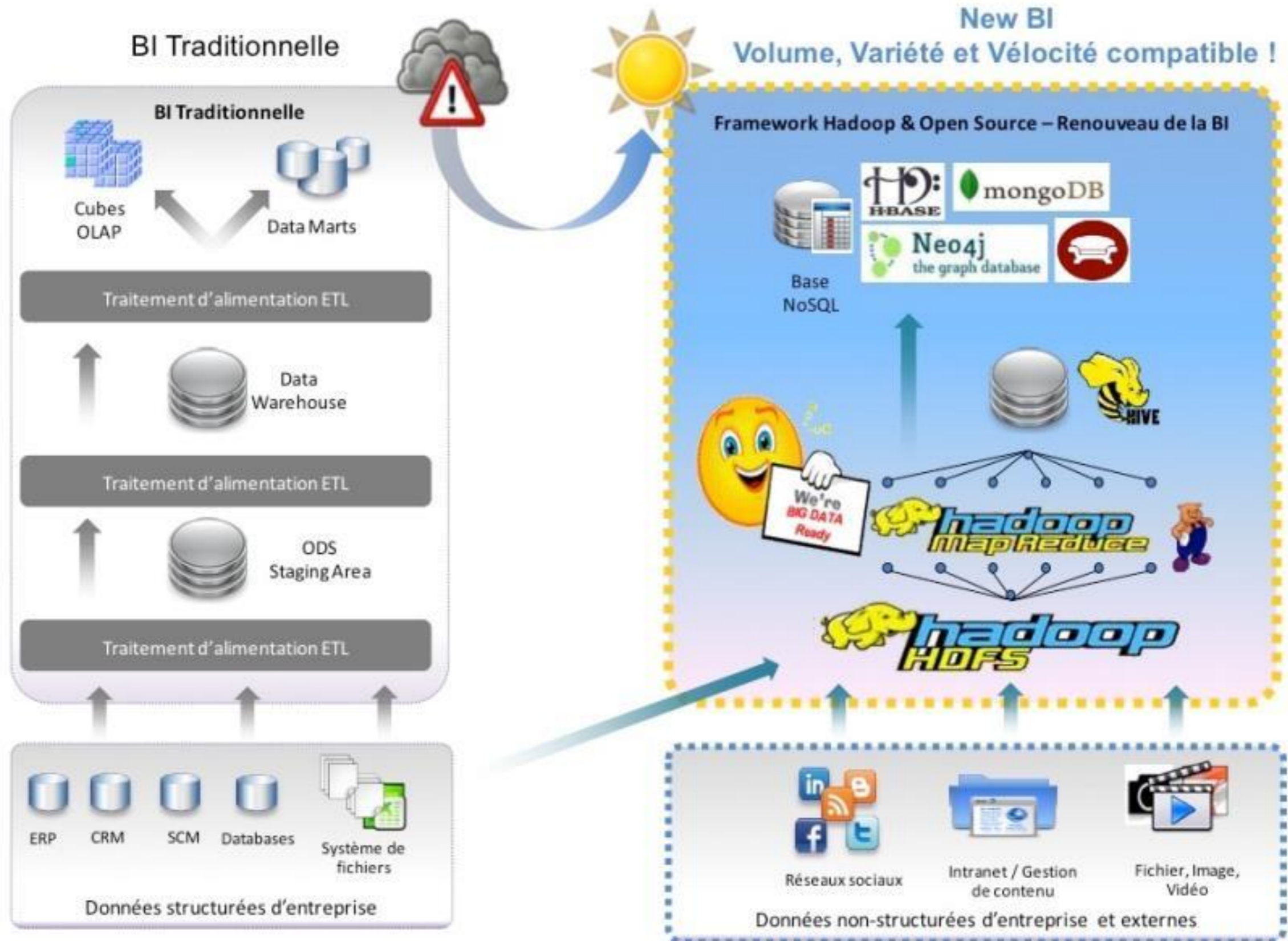


Source: *Linked Data Orchestration, GigaOM Research*



Source: *Linked Data Orchestration*, GigaOM Research

# Une nouvelle architecture Décisionnelle (BI) de BIG DATA 1.0 (Data warehouse) à BIG DATA2.0





## New Decision support pilot tools

*(cf Pr Vincent Blondel, Louvain University, 2013)*

[www.mturk.com](http://www.mturk.com) d' Amazon

Plusieurs milliers de testeurs (coût réduit)

SOCIAL SCIENCES framework

**MOOC**

**Tracking student activity with edX platform**

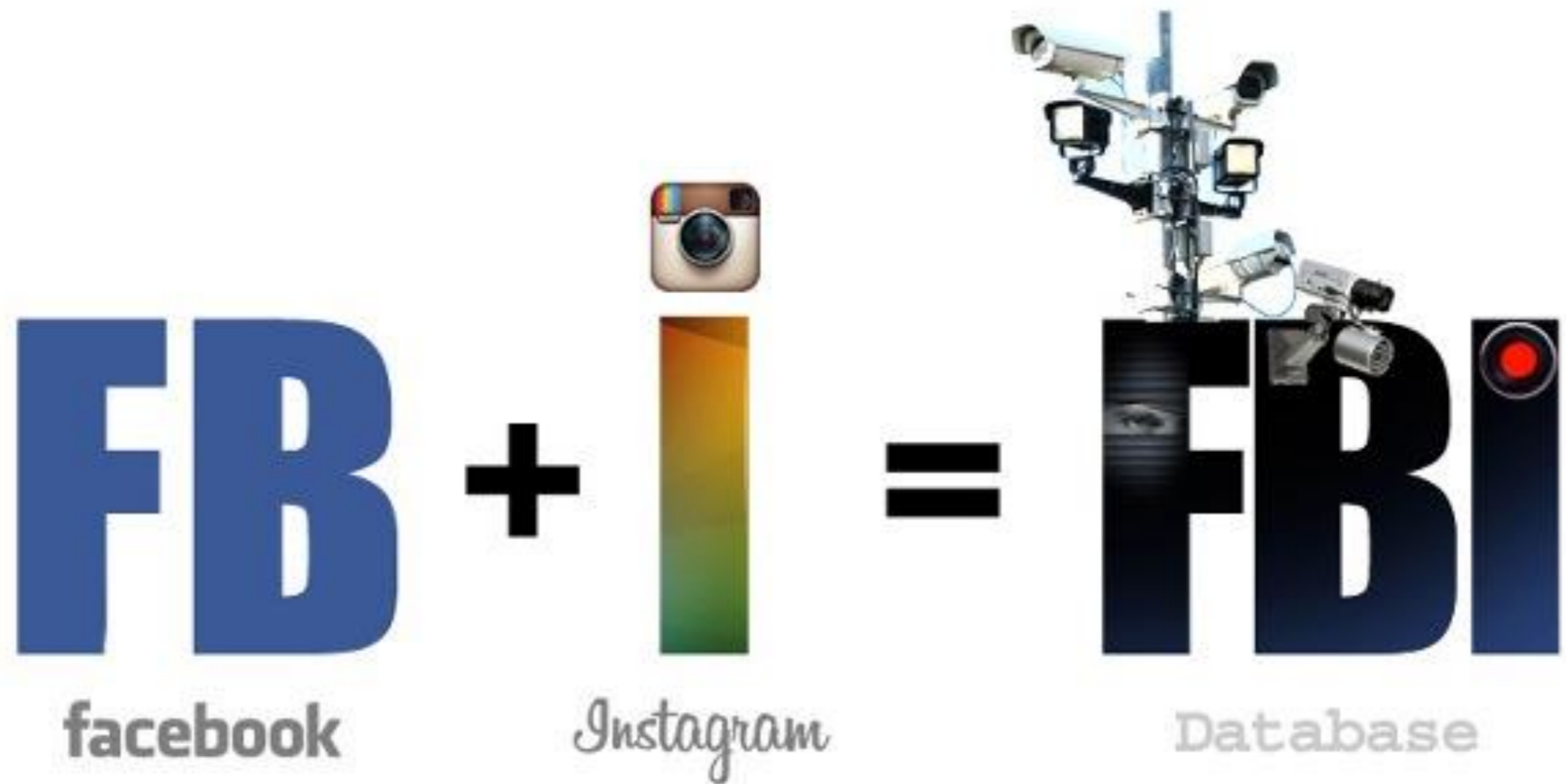
(when ? How long ? Time spent ? Success?

Forum questions?)

→ Learning process !

***Towards Customized personal education***

# WEB 2.0 and ...Big Brother



# SHOWROOMING

THIS PAIR IS SO PERFECT, I  
CAN'T WAIT TO BUY THEM  
CHEAPER ONLINE SOMEWHERE.  
WHAT'S YOUR WIFI PASSWORD?



# « DATA » (Donnée) ? vs « Information » ?

## « DATA » (DONNEE)

ENREGISTREMENT DANS UN Code d'un fait ( objet, transaction, observation) du monde réel

## « Information » : Ce que je peux DEDUIRE d'un ensemble de DATA

Ex : Livre de Médecine Chinoise de 1000 pages en Chinois (ensemble de data !)  
« tout le bruit du monde »

Adrian Mc Donough dans *Information economics* définit l'information comme la rencontre d'une donnée (data) et d'un problème

## DATA : « OR GRIS » de ce millénaire !

« Capital immatériel »; Stratégie du « KNOWING YOU » de Google;

« **COMMUNACTEUR** » : acteur d' enrichissement bottom up des COMMONS (EX / Wikipedia, Open Source, Réseaux sociaux,...)

# « DATA » en préfixe ou suffixe!

## 1) DATA en Préfixe

DATA base (19/8/1968 : Ted Codd et Modèle Relationnel), DBMS

DATA bank

DATA warehouse

DATA mart

DATA mining (OLAP, Corrélations, ..), Data Analytics, DATA Pumping (ETL)

DATA Systems

DATA mash up

DATA SCIENCE

## 2) DATA en suffixe :

- Linked DATA, Web DATA (DBpedia, Web Sémantique)
- Meta DATA
- Open DATA
- Smart DATA
- BIG DATA et nouveaux métiers centrés DATA : :
  - CDO « Chief DATA Officer »,
  - « DATA SCIENTIST »,
  - « DATA BROKER »

# Information (Physique, mathématique) sans ... sémantique

Shannon\* (entropie), Schrödinger\*\* (neguentropie)

La théorie mathématique de l'Information résulte initialement des travaux de Ronald Aylmer Fisher. Celui-ci, statisticien, définit formellement l'information comme égale à la valeur moyenne du carré de la dérivée du logarithme de la loi de probabilité étudiée. < Wikipedia >

$$\mathcal{I}(\theta) = \mathbb{E} \left\{ \left[ \frac{\partial}{\partial \theta} \ln f(X; \theta) \right]^2 \middle| \theta \right\}$$

ENTROPIE (Shannon) : L'entropie permet donc de mesurer la quantité d'information moyenne d'un ensemble d'évènements (en particulier de messages) et de mesurer son incertitude. On la note :

$$H(I) = - \sum_{i \in I} p_i \log_2 p_i$$

La **néguentropie** ou **entropie négative**, est un facteur d'organisation des systèmes physiques, et éventuellement sociaux et humains, qui s'oppose à la tendance naturelle à la désorganisation: l'entropie.

\*Shannon en 1948 : *A Mathematical Theory of Communications*    \*\*Erwin Schrödinger, dans son ouvrage *Qu'est-ce que la vie ?* (1944) pour expliquer la présence de « l'ordre » à l'intérieur des êtres vivants et leur tendance à s'opposer au chaos et à la désorganisation qui régit les systèmes physiques,

# « BASE DE DONNEES » (BD) ? SGBD ?

« *Une base de données (DATA BASE) est un conteneur informatique servant à stocker des données représentant la totalité d'une activité du monde réel* » <WIKIPEDIA>

- Ce stockage se fait en appliquant un « *modèle de données* » (*DATA MODEL*).
  - Le résultat de l'application d'un modèle de données sur un univers réel s'appelle le *SCHEMA de DONNEES (DATA SCHEMA)*
  - Le *modèle relationnel de données défini par Codd* en 1968 régit 90% des Bases de données de 2010
  - Le schéma est le contenant qui va servir à *STRUCTURER les DATA* qui vont être stockées

*le Système de Gestion de base de données –SGBD (DATA BASE MANAGEMENT SYSTEM –DBMS-)* est le logiciel système qui sert à définir, manipuler et contrôler une Base de données

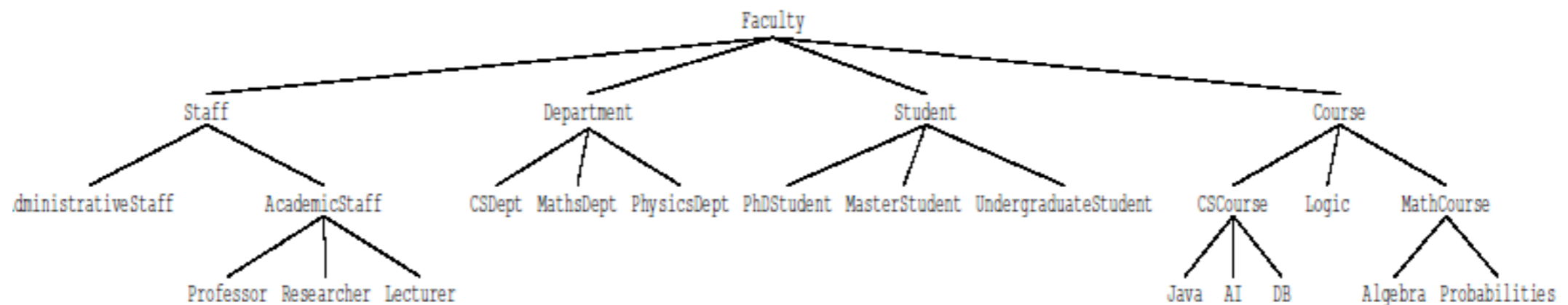
*SQL* (initialement SEQUEL en 1975) est depuis 1980 le standard pour définir, manipuler et contrôler une BASE de DONNEES « relationnelle » (7 Versions en 2014)

# ONTOLOGIE = SCHEMA + INSTANCES

## Définition formelle pour traitement par machines

### CLASSES

- Backbone of the ontology
- AcademicStaff is a **Class**
- (A class will be interpreted as a **set** of objects)
- AcademicStaff **isa** Staff
- (isa is interpreted as set inclusion)





### 3 Langages pour décrire les ontologies du Web

- RDF: a very simple ontology language
- RDFS: Schema for RDF
  - ▶ Can be used to define richer ontologies
- OWL: a much richer ontology language

## Example: SPARQL

Exemple : *Quels sont les Auteurs français nés en 1900 ?*

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?auteur
WHERE {
    GRAPH ?g {
        ?auteur rdf:nationalité rdf:France .
        ?auteur rdf:annéeNaissance rdf:1900
    }
}
```

# Passerelle

## SPARQL et SQL

```
SELECT ?person ?tel WHERE {
  ?person rdf:type bm:GraduateStudent
  {
    { ?person bm:like ?interest } UNION
    { ?person bm:love ?interest }
  } .
  OPTIONAL { ?person bm:telephone ?tel } .
  ?person bm:age ?age .
  FILTER ( ?age < 25 &&
           REGEX(STR(?interest), "Ball$") )
}
```

\*

tria,

Fig. 1. An example of SPARQL query

telephone number of all graduated students with age less than 25 and have an interest of ball sports, in a UOBM [2] ontology (The `bm:age` is an extended property of the UOBM).

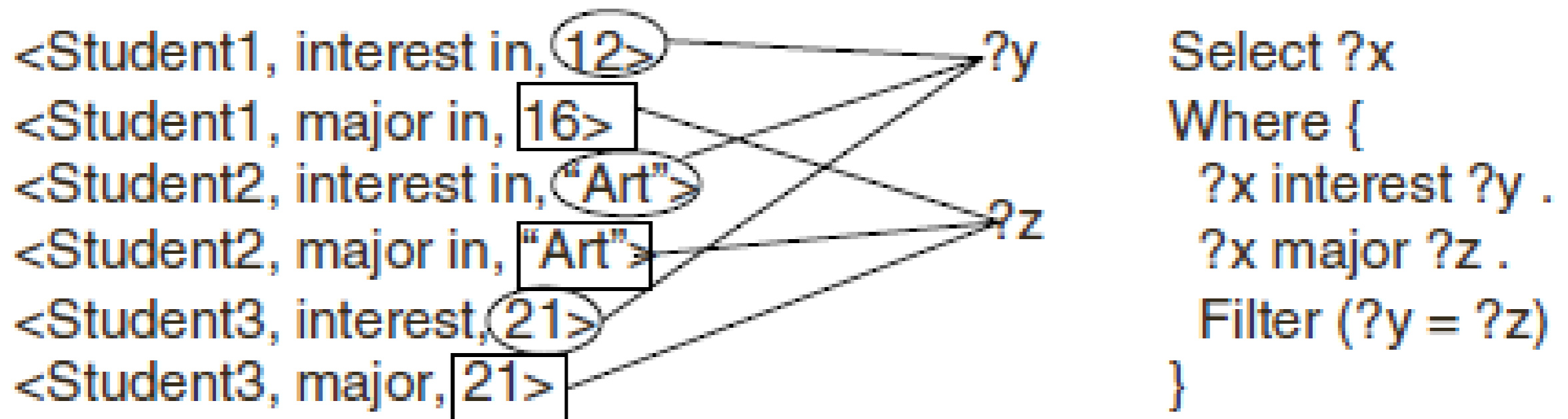
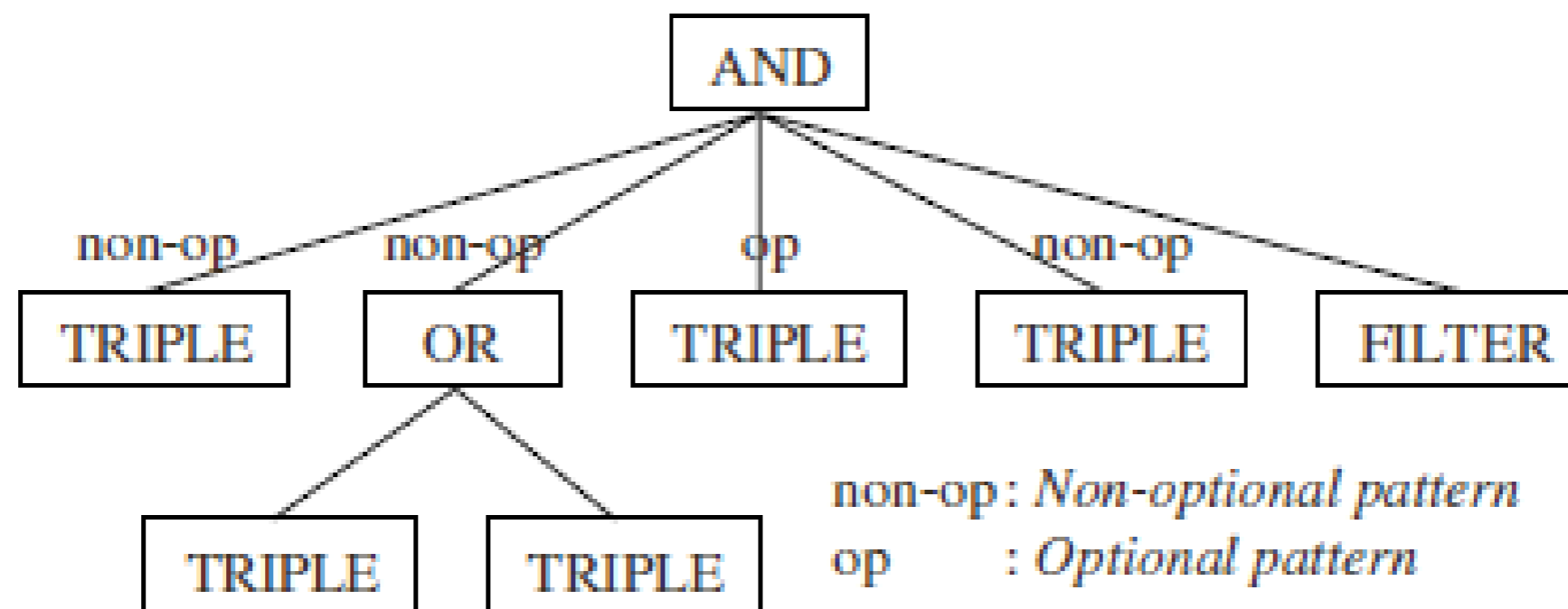


Fig. 2. An example of dynamic data type of RDF resource

The generated SQL may be different for different database schemas. To simplify the discussion, we assume that all the triples are stored in a triple table, in which internal IDs are used instead of IRIs or literal strings. The IRI and literal strings are stored in two separate tables. Most known RDF stores adopt such a schema.



**Fig. 3.** An example SPARQL pattern tree

# Passerelle

```
{?person bm:isFriendOf ?person}
```

is translated into:

```
SELECT subject AS person FROM triple  
  WHERE predicate = pID and subject = object
```

Here, the pID stands for the ID the IRI <bm:isFriendOf>.

- An AND node is translated into a query on consequent joins of the sub-queries from its child pattern nodes.

# OWL (Ontology Web Language)

**Declarative logic-based language** based on  
RDF

Programs (*reasoners*) to  
Verify consistency of knowledge

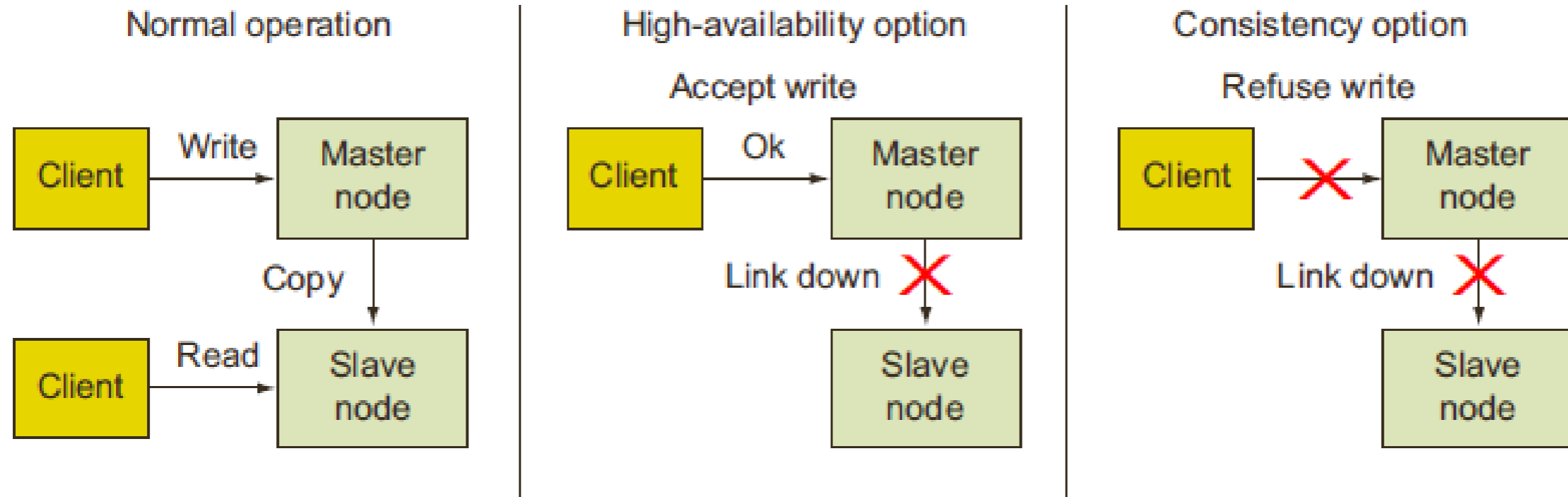
Discover implicit knowledge

OWL Object properties

Semantic inherent approach with reasoning  
capability

*Appropriate for DATA Exchange (protocols)  
Cf [ALIMI2012] to model the SE of Global  
Platform*

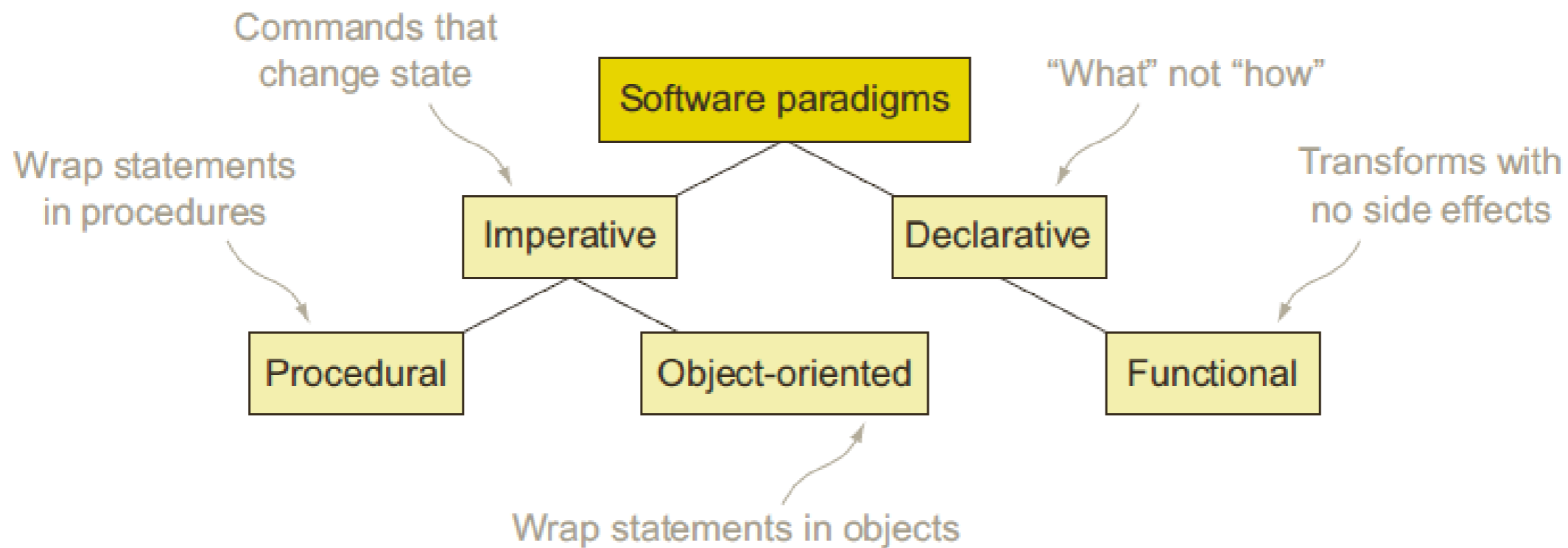
# CAP Theorem



**Figure 2.10** The partition decision. The CAP theorem helps you decide the relative merits of availability versus consistency when a network fails. In the left panel, under normal operation a client write will go to a master and then be replicated over the network to a slave. If the link is down, the client API can decide the relative merits of high availability or consistency. In the middle panel, you accept a write and risk inconsistent reads from the slave. In the right panel, you choose consistency and block the client write until the link between the data centers is restored.



# SOFTWARE PARADIGMS (MANNING 2013)



**Figure 10.1** A high-level taxonomy of software paradigms. In the English language, an imperative sentence is a sentence that expresses a command. “Change that variable now!” is an example of an imperative sentence. In computer science, an imperative programming paradigm contains sequences of commands that focus on updating memory. Procedural paradigms wrap groups of imperative statements in procedures and functions. Declarative paradigms focus on what should be done, but not how. Functional paradigms are considered a subtype of declarative programming because they focus on what data should be