Dauphine | PSL UNIVERSITÉ PARIS | MIDO MATHEMATIQUES ET INFORMATIQUE DE LA DECISION ET DES ORGANISATIONS

Master 2 - Informatique des Organisations Parcours MIAGE IF (Informatique pour la Finance) en Apprentissage

SQL – NoSQL - NewSQL

Maude Manouvrier

Espace Moodle : A5AMIF-100_24/25_SQL, NoSQL et NewSQL_MAUDE MANOUVRIER
Equipe Teams : SQL, NoSQL et NewSQL M2 MIAGE-IF 2024-2025

BIBLIOGRAPHIE (1/2)

Ouvrages et liens utilisés pour le cours

- **P. J. Sadalage et M. Fowler (2013).** ***NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*****. Addison Wesley** (http://bigdata-ir.com/wp-content/uploads/2017/04/NoSQL-Distilled.pdf)

- **G. Harrison (2015).** ***Next Generation Databases NoSQL, NewSQL, and Big Data*****. Apress** (https://link.springer.com/content/pdf/10.1007%2F978-1-4842-1329-2.pdf)

- **N. Travers (2017).** ***NoSQL sur OpenClassrooms*** **:** https://chewbii.com/nosql-sur-openclassrooms/

- **P. Rigaux (2018).** ***Bases de données documentaires et distribuées*** **:** http://b3d.bdpedia.fr/

©Maude Manouvrier - Univ. Paris-Dauphine

BIBLIOGRAPHIE (2/2)

Ouvrages et liens utilisés pour le cours

- https://dbdb.io/
- **P. Senellart,** ***Limites des systèmes classiques de gestion de bases de données*****.** https://www.fun-mooc.fr/c4x/MinesTelecom/04006/asset/Télécom_FBD_S2_Basesdedonnéesrelationnelles_V1.pdf
- **R. Bruchez (2015).** ***Les bases de données NoSQL et le Big Data*****. Eyrolles (Extrait sur** https://static.fnac-static.com/multimedia/editorial/pdf/9782212141559.pdf**)**
- **Cours NoSQL en ligne de R. Bruchez (2019) :** https://www.linkedin.com/learning/l-essentiel-de-nosql/comprendre-les-modeles-de-donnees?u=74606242
- **+ d'autres sites référencés en bas des transparents**

©Maude Manouvrier - Univ. Paris-Dauphine

Introduction

- **SQL**
- **NoSQL**
- **NewSQL ou SQL distribué**
- **Systèmes multi-modèles**
- **Persistance polyglotte**

©Maude Manouvrier - Univ. Paris-Dauphine

Historique des bases de données

Bases de données relationnelles (fin des années 70) – Innovation de rupture

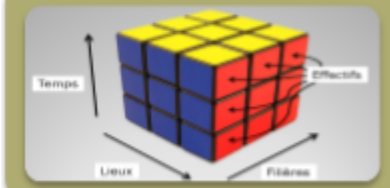
- **Outils de modélisation : E/R, ..**
- **langages de requêtes : SQL, etc.**
- **Optimisation des requêtes**
- **Indexation : Arbres B+, hachage, etc.**
- **Interfaces, outils de report**
- **Gestion des transactions, OLTP**

Bases de données avancées (années 80)

- *modèles* : relationnel étendu, objet, objet-relationnel …
- *applis* : spatial, temporel, multimédia, active, bases de connaissances, ...

Base de données et web (années 90)

- **XML et Bases de Données**
- **Web-Mining**



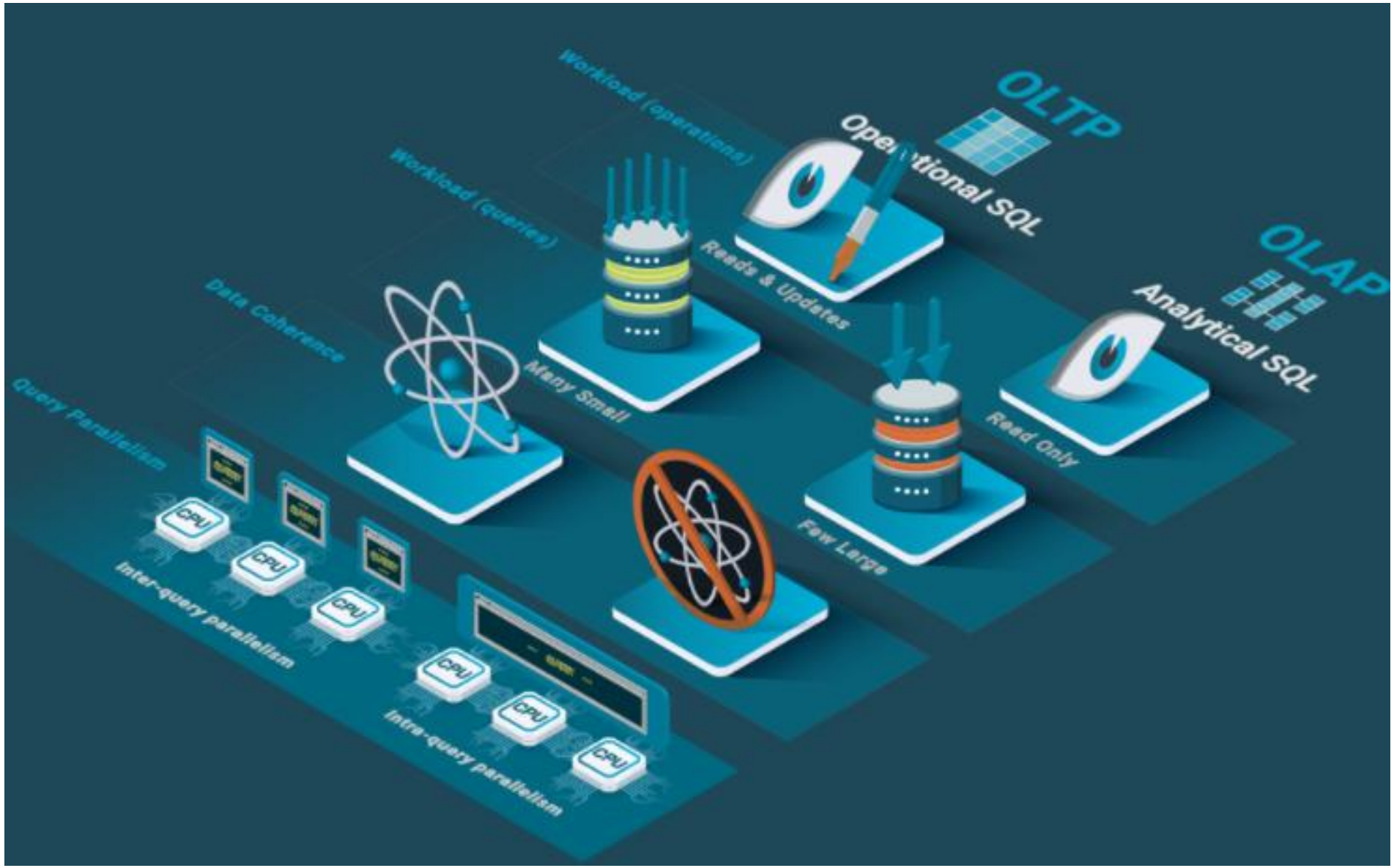
Décisionnel (années 90)

- **Entrepôts et OLAP (*OnLine Analytical Processing*)**
- **KDD et data mining**

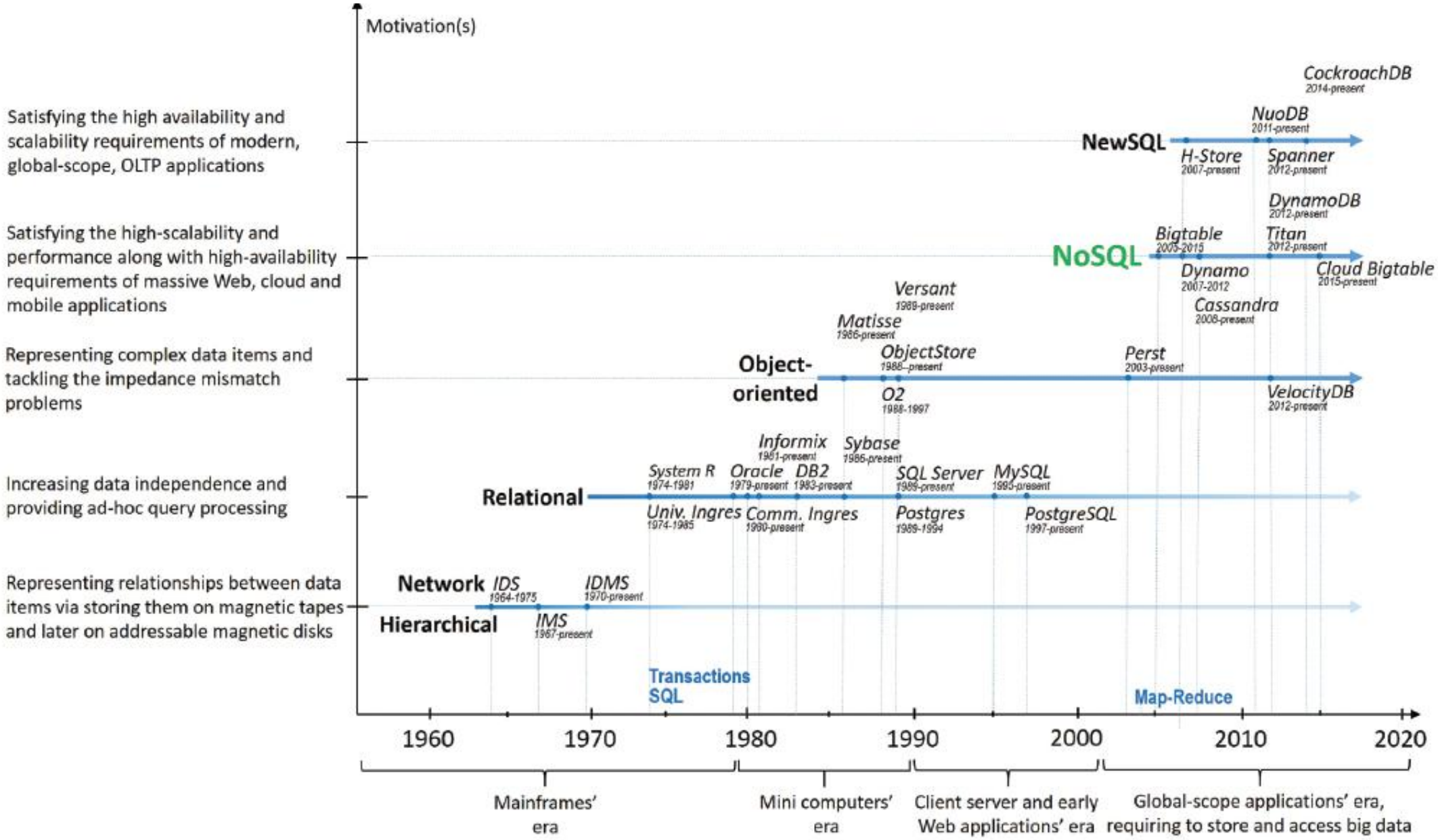
Nouvelles bases de données (fin des années 2000)

- **NoSQL**
- **In Memory**

OLAP (*On Line Analytical Processing*) **vs OLTP** (*On Line Transactional Processing*)



Date de naissance des différents moteurs



©Maude Manouvrier - Univ. Paris Dauphine- Repris de https://www.researchgate.net/publication/324640550_A_survey_on_NoSQL_stores

Classement des moteurs de bases de données

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

424 systems in ranking, February 2025

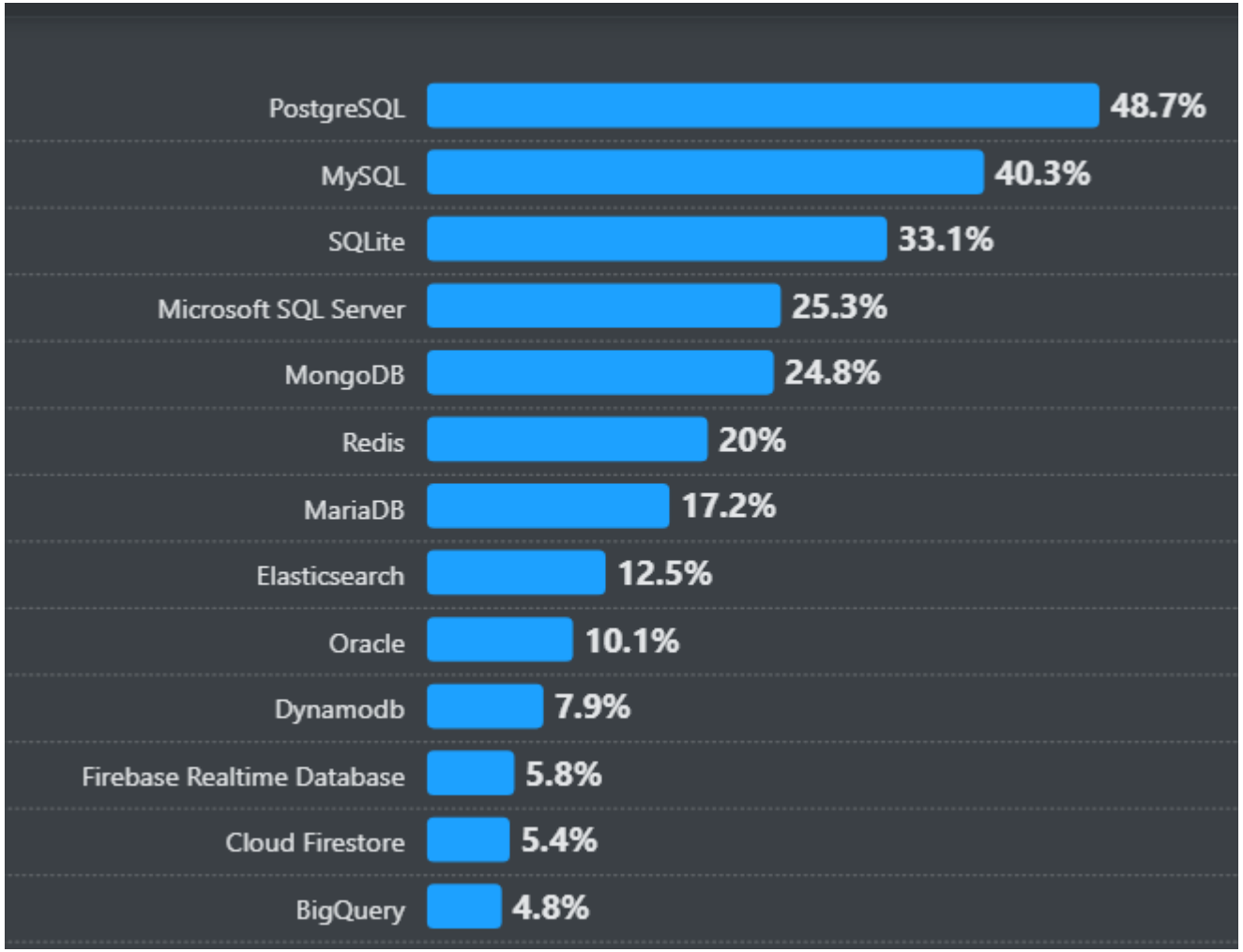
Rank Feb 2025	Rank Jan 2025	Rank Feb 2024	DBMS	Database Model	Score Feb 2025	Score Jan 2025	Score Feb 2024
1.	1.	1.	Oracle	Relational, Multi-model	1254.82	-3.93	+13.38
2.	2.	2.	MySQL	Relational, Multi-model	999.99	+1.84	-106.67
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	786.87	-11.69	-66.70
4.	4.	4.	PostgreSQL	Relational, Multi-model	659.62	-3.79	+30.21
5.	5.	5.	MongoDB	Document, Multi-model	396.63	-5.87	-23.73
6.	↑ 7.	6.	Redis	Key-value, Multi-model	157.91	+4.55	-2.80
7.	↓ 6.	↑ 9.	Snowflake	Relational	155.58	+1.68	+28.13
8.	8.	↓ 7.	Elasticsearch	Multi-model	134.63	-0.29	-1.11
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model	125.44	+2.46	-6.79
10.	10.	10.	SQLite	Relational	113.82	+7.13	-3.47
11.	11.	↑ 12.	Apache Cassandra	Wide column, Multi-model	102.58	+3.39	-6.69
12.	12.	↓ 11.	Microsoft Access	Relational	96.54	+3.84	-16.63
13.	13.	↑ 17.	Databricks	Multi-model	90.03	+2.19	+13.13
14.	14.	↓ 13.	MariaDB	Relational, Multi-model	89.50	+3.92	-7.73
15.	15.	↓ 14.	Splunk	Search engine	80.56	-2.53	-11.09
16.	↑ 17.	↓ 15.	Amazon DynamoDB	Multi-model	75.58	+2.58	-7.31
17.	↓ 16.	↓ 16.	Microsoft Azure SQL Database	Relational, Multi-model	72.76	-1.02	-6.80

©Maude Manouvrier - Univ. Paris Dauphine

Bases de données les plus populaires

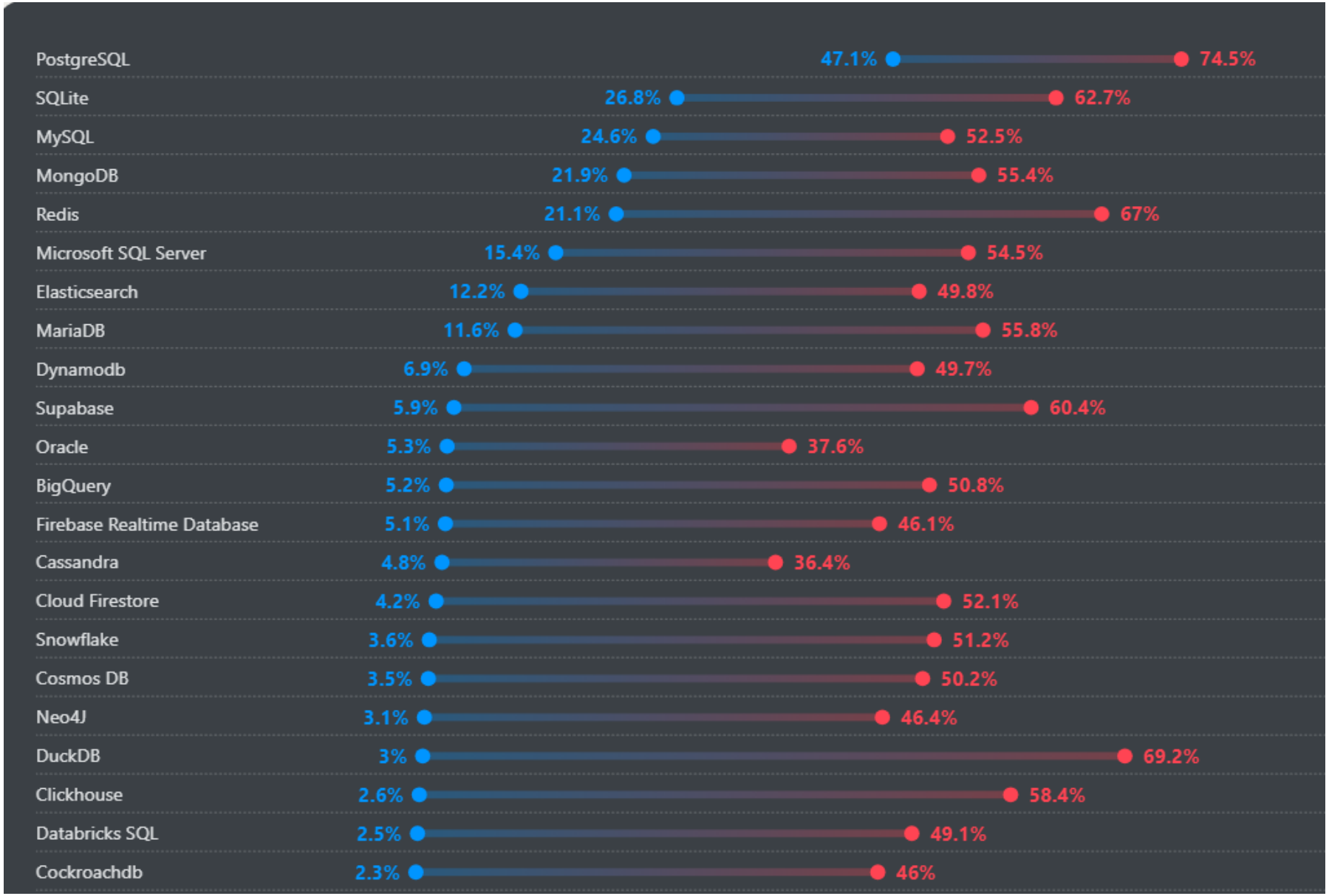
Etude de mai 2024 *stackoverflow* basée sur les réponses de 65 000 réponses de développeurs issues de 185 pays

https://survey.stackoverflow.co/2024/#technology

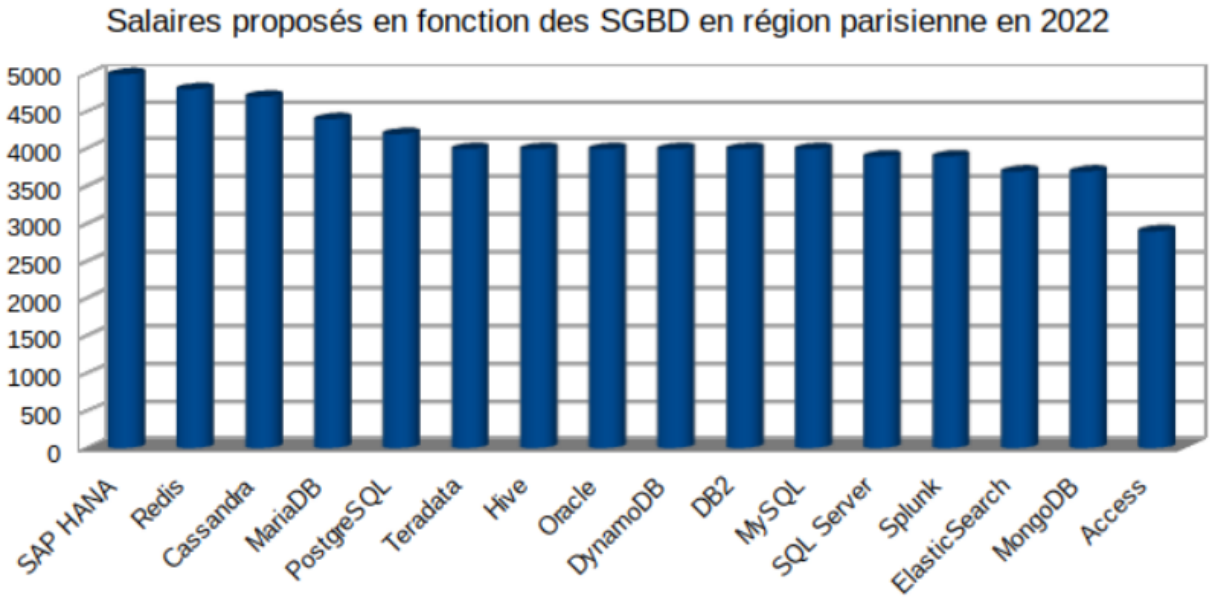


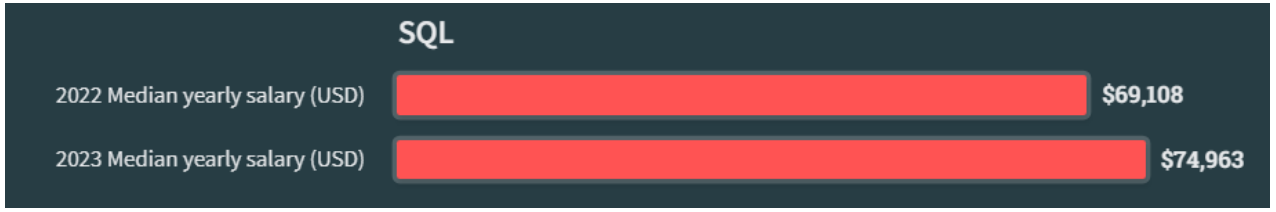
©Maude Manouvrier - Univ. Paris Dauphine

Appréciation/Détestation des bases de données



Profils BD les mieux payés





©Maude Manouvrier - Univ. Paris Dauphine - https://survey.stackoverflow.co/2022#top-paying-technologies-databases et https://emploi.developpez.com/actu/344564/Emploi-informatique-2022-les-SGBD-les-plus-demandes-et-les-mieux-payes/

SQL : différentes versions du standard

- Technologies de bases de données **relationnelles** et **transactionnelles** existantes depuis plus de 50 ans
- **SQL** : langage de requête standardisé en 1986
  - **SQL2/SQL92** : standard adopté en 1992
  - **SQL3/SQL99** : extension de SQL2 avec gestion d' "objets", déclencheurs …
  - **SQL 2003**: auto incrémentation des clés, colonne calculée, prise en compte de XML, …
  - **SQL 2008 et 2011**: correction de certains défauts et manques (fonctions, types, curseurs) …
  - **SQL 2016 :** gestion de documents JSON, …
  - **SQL 2019** : tableaux multidimensionnels (*Multi-Dimensional Arrays* - MDA - 2019), …
  - **SQL 2023** : SQL/PGQ (*Property Graph Queries*) …

SQL : les avantages du relationnel

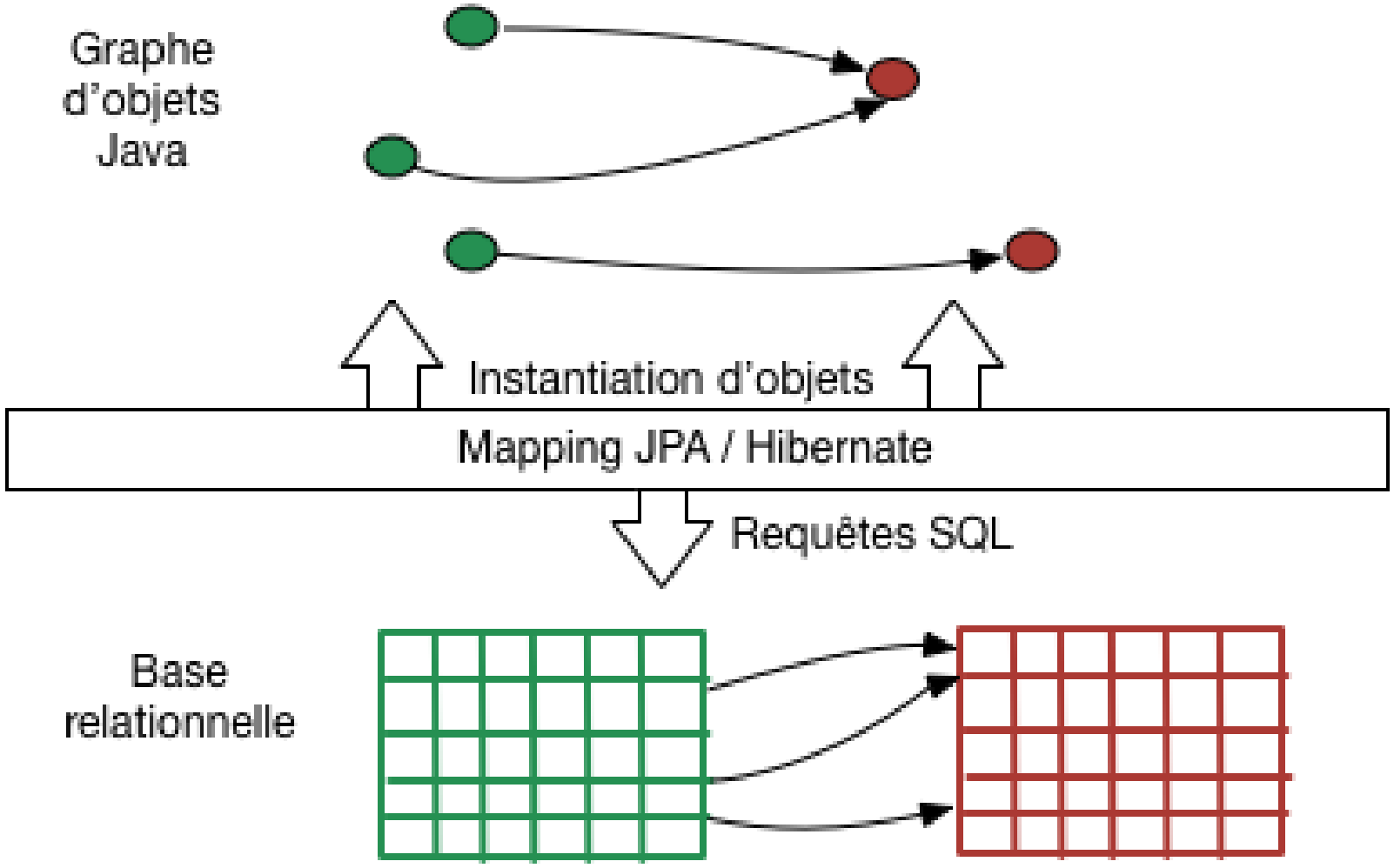
SGBD relationnel (SGBDR) :

- **Base théorique** : modèle relationnel (indépendant de la structure de stockage)
- Gestion de l'**intégrité des données,** de la **concurrence** et de la **reprise sur panne** (ACID)
- **Gestion et optimisation des requêtes** : indépendance entre la déclaration des requêtes et leur exécution, mécanismes d'indexation …
- Outils matures permettant de gérer de grands volumes de données (gigaoctet, voire téraoctet)
- Outils de **mapping objet-relationnel (ORM)** : JPA, Hibernate …

\+ …

©Maude Manouvrier - Univ. Paris Dauphine

Mapping Objet/relationnel



©Maude Manouvrier - Univ. Paris Dauphine – Figure reprise de http://orm.bdpedia.fr/introjpa.html

The Law of the Relational Database

If the only tool you have is a relational database,
everything looks like a table.

A Walk in Graph Databases - 2012

SQL : le coût des jointures

SQL : les inconvénients du relationnel

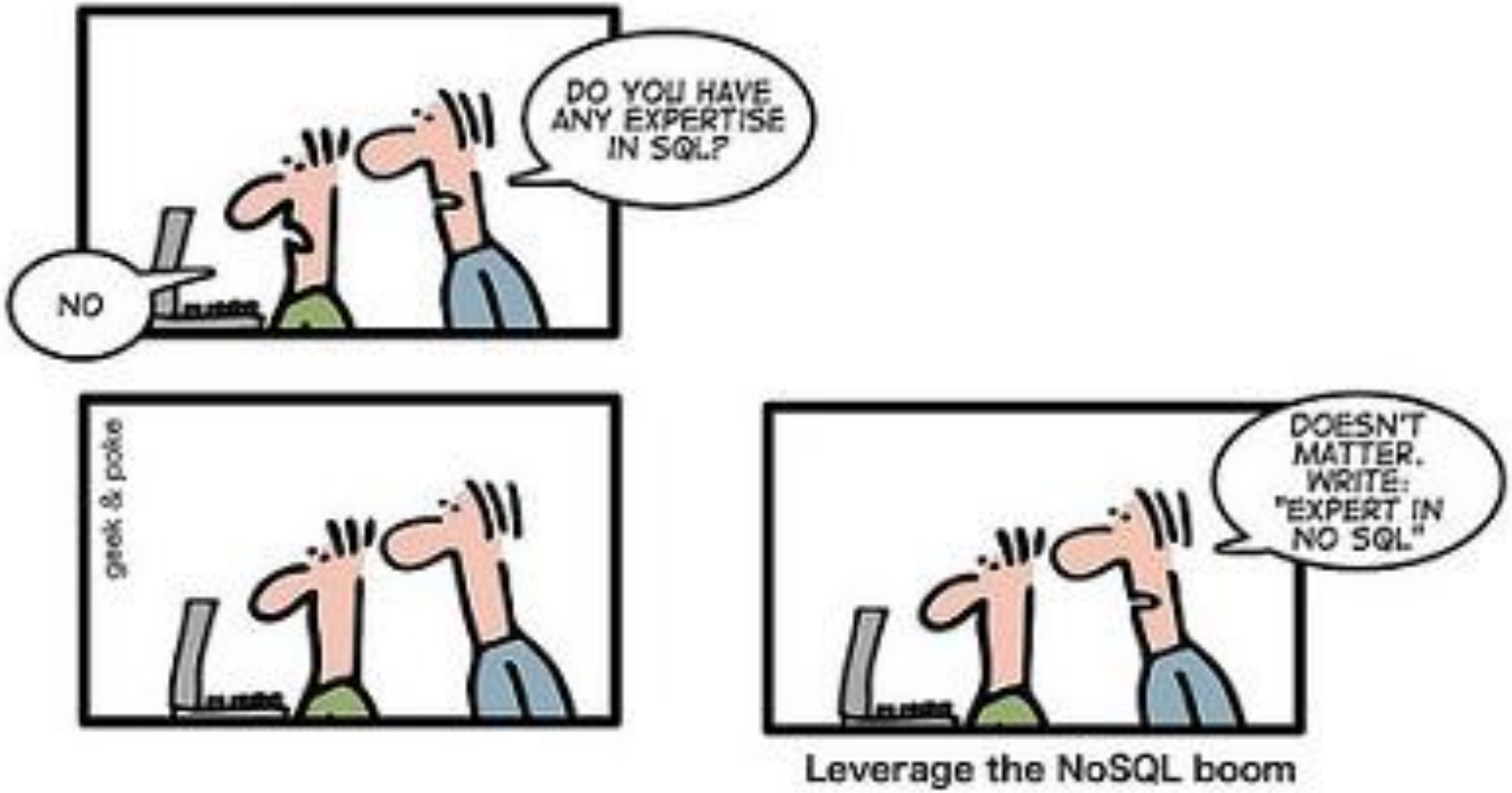
SGBD relationnel (SGBDR) :

- **Modèle relationnel (représentation tabulaire) parfois peu adapté au stockage et à l'interrogation de certains types de données :** hiérarchiques, faiblement structurées, semi-structurées, complexes et hétérogènes
- **Problème d'*Impedance mismatch*** (Défaut d'impédance) **:** Non correspondance des modèles objet et relationnel
- Systèmes généralement centralisés et **non adaptés aux environnements répartis (*distributed*)**
- **Problèmes de gestion de très grands volumes de données (**de l'ordre du pétaoctet) **et de débits extrêmes** (> milliers de requêtes par seconde)
- **Propriétés ACID** : **Surcoût** en latence, accès disques, temps CPU (verrous, journalisation, etc.)
- **Performances limitées par les accès disque**

…

©Maude Manouvrier - Univ. Paris Dauphine

HOW TO WRITE A CV



NoSQL : alternative et complément au relationnel

- ***Not Only SQL*** **= Non relationnel**
- **En contraste avec le *One Size Fits All (« taille unique »)* du SQL**
- Terme (ré)apparu en 2009 vague et incorrect (certains moteurs NoSQL utilisent des variantes du SQL)
- « **NoSQL ne vient pas remplacer les BD relationnelles** mais proposer une alternative ou compléter les fonctionnalités des SGBDR pour donner des solutions plus intéressantes dans certains contextes »
  https://medium.com/nexaby/nosql-5-minutes-pour-comprendre-60dbd579e3e3

©Maude Manouvrier - Univ. Paris Dauphine

NoSQL : pas de table

NoSQL : agrégat

- Unité logique de stockage : **l'agrégat** et non plus relation (table)
- **Agrégat = collection d'objets référencés par une clé** (pas d'autres d'accès que par la clé)
- **Pas de liens entre les agrégats** (contrairement aux relations avec les contraintes d'intégrité référentielles) **- dénormalisation**
- **Indépendance de chaque agrégat d'où une répartition possible des données**
- A utiliser en cas de :
    - **Grand volume de données**
    - **Architecture fortement repartie (*distributed*)**
    - **Requêtes massives**
    - **Schéma évolutif**

©Maude Manouvrier - Univ. Paris Dauphine

NoSQL : avantages et inconvénients

Modèles de données différents et **souplesse/flexibilité au niveau du schéma** : sans schéma ou schéma dynamique, gestion des données éparses (NULL) et semi-structurées …

Passage à l'échelle, haute disponibilité et élasticité : adaptation dynamique à la distribution des données et à la structure du réseau de machines

Langage simple (API)

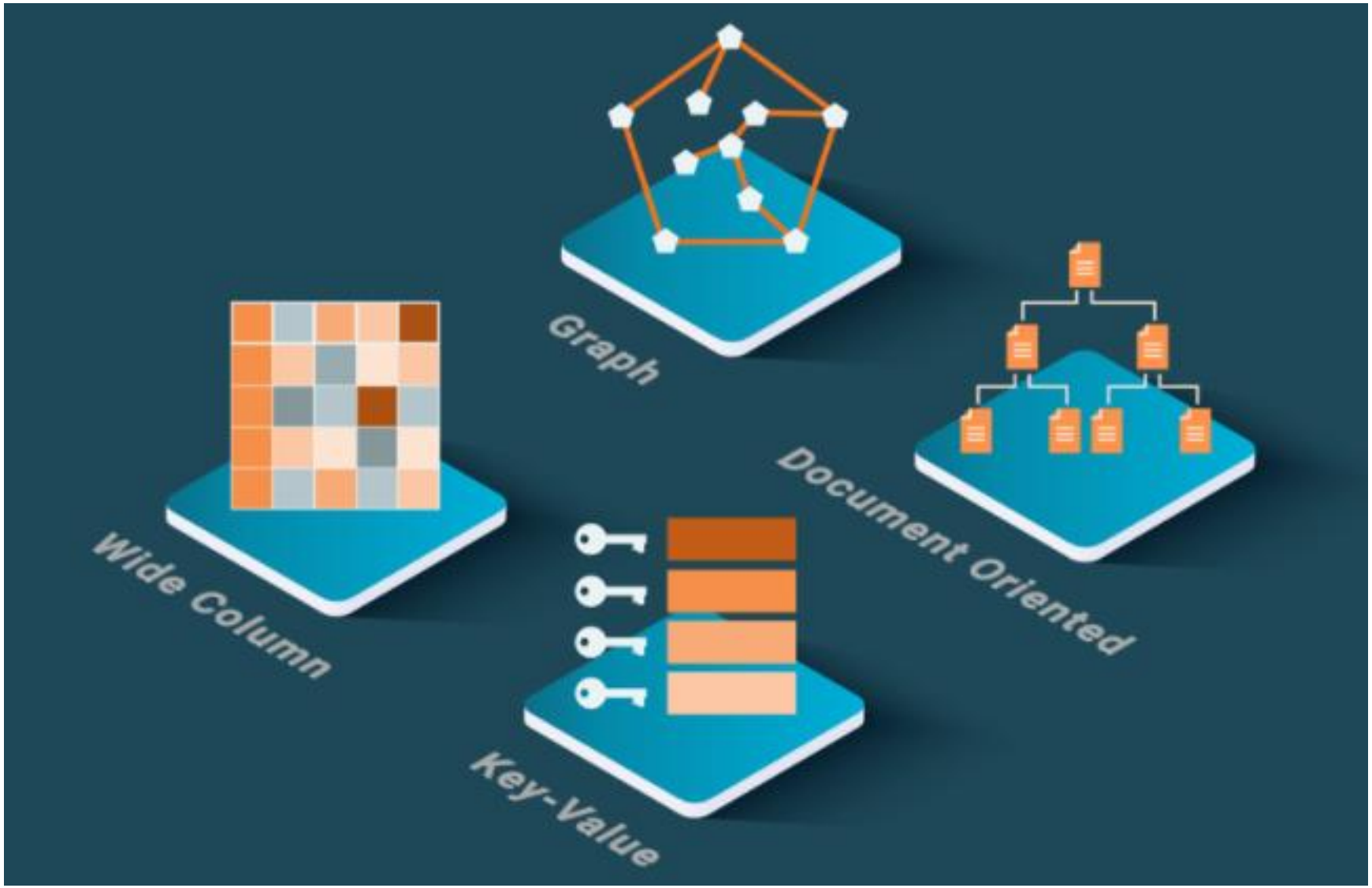
Modèle de données puissant, mais représentations asymétriques des informations - Pas de standards

Problème de cohérence et de contrôle sur les données : Système généralement des systèmes CP (Cohérent et résistant au Partitionnement) ou AP (Disponible et résistant au Partitionnement)

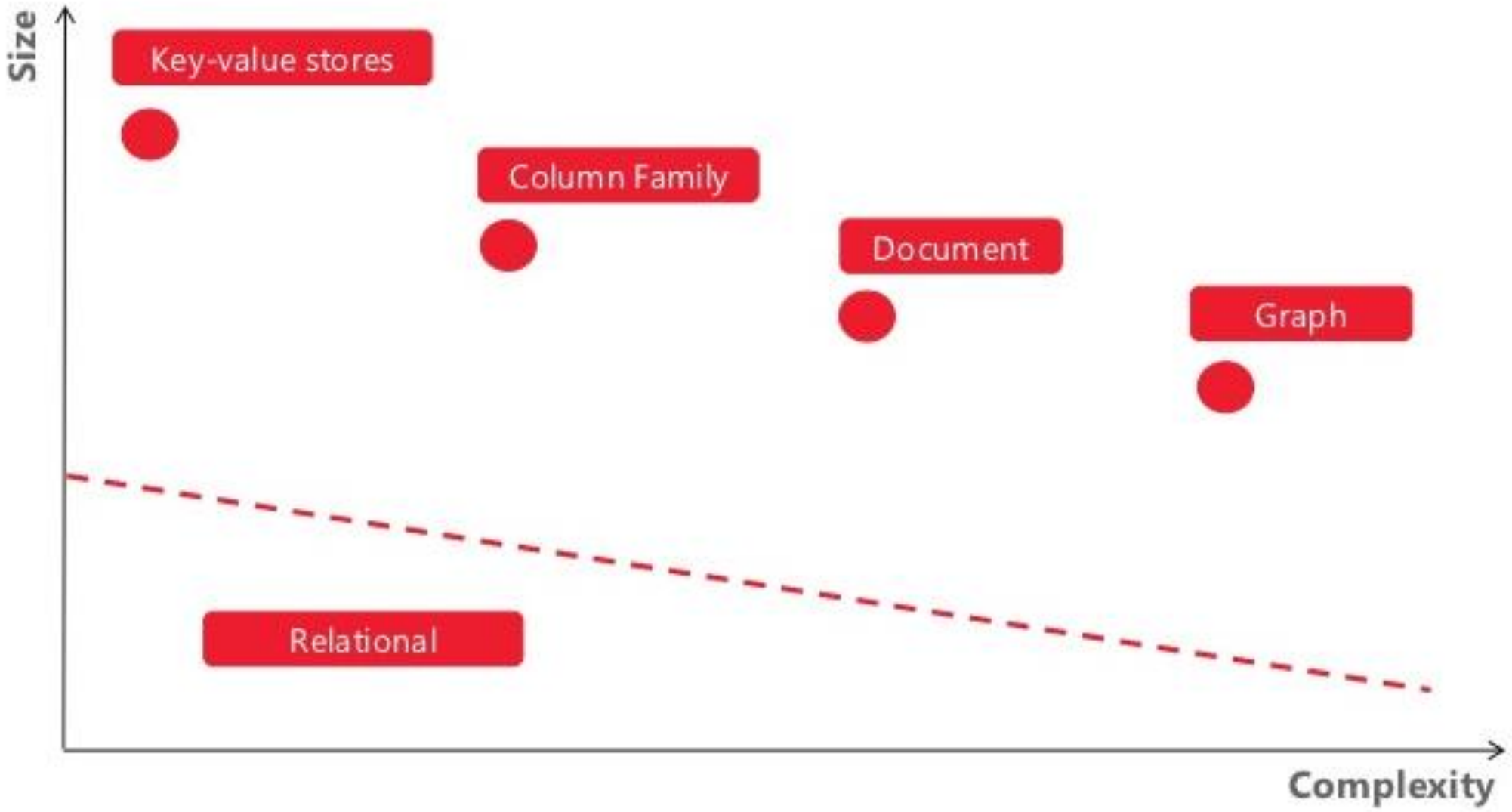
Requêtage limité (pas d'interrogation complexe)

©Maude Manouvrier - Univ. Paris Dauphine

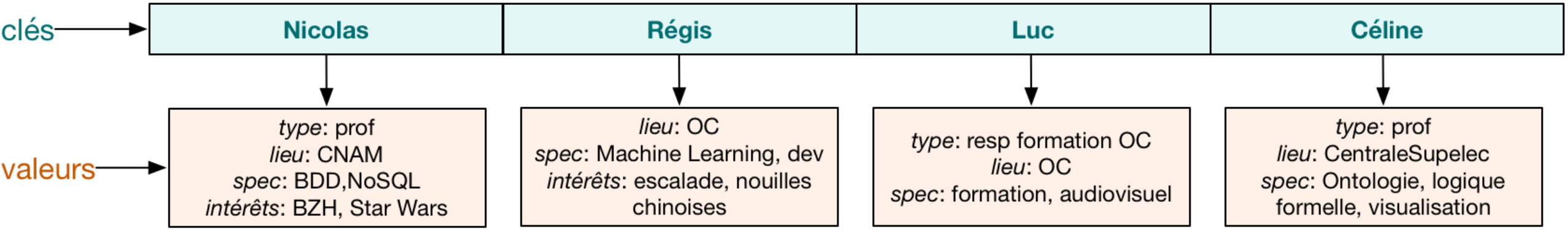
NoSQL : 4 modèles (nom en anglais)



NoSQL : à chaque modèle ses critères



Modèle Clé - Valeur : Exemple



- clé : identifie la donnée de manière unique et permet de la gérer
- valeur : contient n'importe quel type de données

©Maude Manouvrier - Univ. Paris-Dauphine - Figure reprise de https://openclassrooms.com/courses/4462426-maitrisez-les-bases-de-donnees-nosql

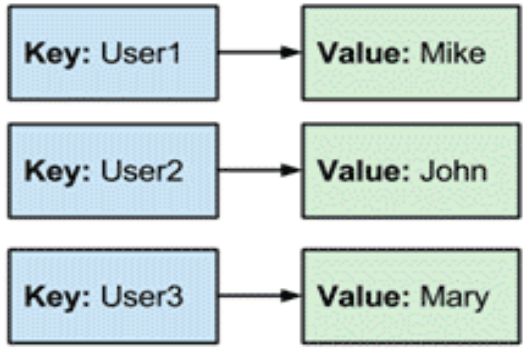
Modèle Clé - Valeur : principes

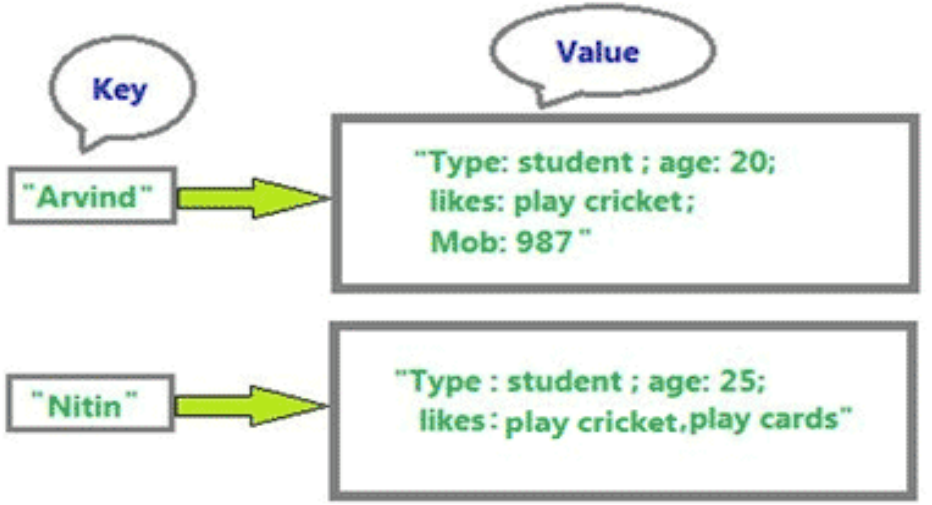
- Principalement fait pour le stockage temporaire et ne permettent que 4 opérations :
  - création : créer un nouveau couple (clé, valeur)
  - lecture : lire un objet (récupérer la valeur) en connaissant sa clé
  - modification : mettre à jour (donner une nouvelle valeur à) l'objet associé à une clé
  - suppression : supprimer un objet connaissant sa clé
- Beaucoup utilisé en tant que cache, pour conserver les sessions d'un site web et plus généralement pour toutes les données que l'on ne souhaite conserver que pendant un certain laps de temps, pouvant aller de quelques secondes à quelques jours

  Exemple :

  - gestion de panier d'achat (Amazon)
  - collecte d'événements (jeu en ligne**)**

Modèle Clé - Valeur : avantages, inconvénients et principaux moteurs





+

Facilement scalable

Temps de réponse en écriture / lecture très bas

−

Mises à jour compliquées

Requêtes rudimentaires

Exemples d'implémentation

Amazon DynamoDB (Beta)

Riak

redis

Classement des moteurs Clé - Valeur

☐ include secondary database models

71 systems in ranking, February 2025

Rank			DBMS	Database Model	Score		
Feb 2025	Jan 2025	Feb 2024			Feb 2025	Jan 2025	Feb 2024
1.	1.	1.	Redis	Key-value, Multi-model	157.91	+4.55	-2.80
2.	2.	2.	Amazon DynamoDB	Multi-model	75.58	+2.58	-7.31
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	22.13	-0.83	-9.86
4.	4.	4.	Memcached	Key-value	16.69	+0.09	-3.72
5.	5.	↑ 6.	etcd	Key-value	7.06	-0.02	-1.08
6.	6.	↓ 5.	Hazelcast	Key-value, Multi-model	5.52	-0.07	-2.66
7.	7.	7.	Aerospike	Multi-model	5.09	+0.05	-2.00
8.	8.	8.	Ehcache	Key-value	4.32	-0.07	-1.58
9.	9.	9.	Riak KV	Key-value	3.33	-0.14	-1.90
10.	10.	↑ 15.	Oracle NoSQL	Multi-model	3.29	+0.05	-0.50
11.	↑ 15.	↓ 10.	RocksDB	Key-value	3.06	+0.33	-1.46
12.	12.	12.	ArangoDB	Multi-model	2.91	-0.01	-1.43
13.	↓ 11.	↓ 11.	Apache Ignite	Multi-model	2.89	-0.05	-1.52
14.	14.	↑ 16.	OrientDB	Multi-model	2.73	-0.06	-0.92
15.	↓ 13.	↓ 13.	Google Cloud Bigtable	Multi-model	2.70	-0.09	-1.46
16.	↑ 17.	↑ 19.	GemFire	Key-value, Multi-model	2.43	-0.01	-0.97

NoSQL : Modèle Colonne

Orientées Colonnes

sont très proches des SGBDR, on y retrouve le principe de « table », mais elles présentent deux grosses différences : les colonnes sont dynamiques et l'historisation de la données se fait à la valeur et non à la ligne.

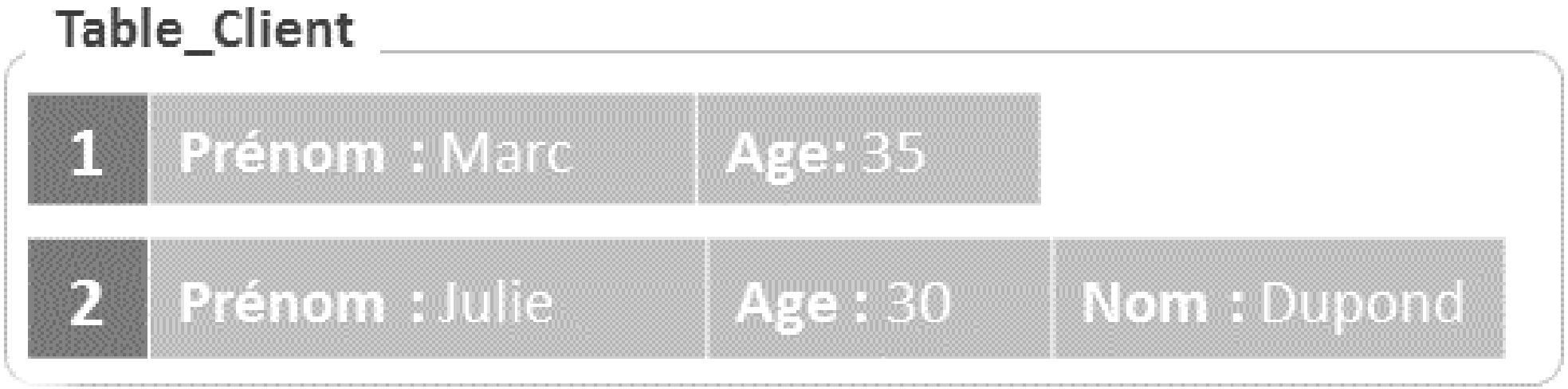


Figure reprise de http://tech.aramisauto.com/common/2016/10/03/nosql-a-quoi-ca-sert.html

Modèle orienté colonne : Exemple

Stockage orienté lignes

id	type	lieu	spec	intérêts
Nicolas	prof	CNAM	BDD, NoSQL	BZH, Star Wars
Régis		OC	Machine Learning, Dev	escalade, nouilles chinoises
Luc	resp formation OC	OC	formation, audiovisuel	
Céline	prof	CentraleSupelec	Ontologie, logique formelle, visualisation	

Stockage orienté colonnes

id	type
Nicolas	prof
Céline	prof
Luc	resp formation OC

id	lieu
Céline	Centrale Supelec
Nicolas	CNAM
Régis	OC
Luc	OC

id	spec
Nicolas	BDD
Nicolas	NoSQL
Régis	Machine Learning
Régis	Dev
Luc	formation
Luc	audiovisuel
Céline	Ontologie
Céline	logique formelle
Céline	visualisation

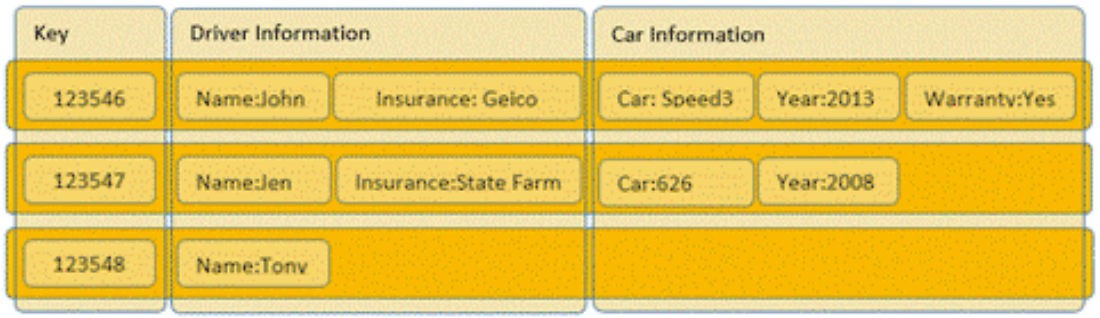
id	intérêts
Nicolas	BZH
Nicolas	Star Wars
Régis	escalade
Régis	nouilles chinoises

©Maude Manouvrier - Univ. Paris-Dauphine - Figure reprise de https://openclassrooms.com/courses/4462426-maitrisez-les-bases-de-donnees-nosql

Modèle orienté colonne : principes

- Non-adapté pour les données non-structurées et la lecture de données spécifiques
- Adapté pour effectuer des traitements sur des colonnes comme les agrégats (comptage, moyennes, cooccurrences...) ou les gros calculs analytiques
- Adapté lorsque l'on doit stocker de très nombreux évènements qui doivent être mis à jour très régulièrement. Comme par exemple :
  - Le suivi de colis (de nombreux évènements dont le statut change : En préparation, en cours de livraison, livré..)
  - La récupération et l'analyse de données en temps réel issues de capteurs, IOT etc.

Modèle orienté colonne : avantages, inconvénients et principaux moteurs



Key	Name	Insurance	Car	Year	Warranty
123456	John	Geico	Speed3	2013	Yes
123457	Jen	State Farm	626	2008	NULL
123458	Tony	NULL	NULL	NULL	NULL

ServiceID	Type	Key
10	Oil	123456
11	Tires	123456
12	Wipers	12345

+

Capacité de stockage accrue

Accès rapide aux données

−

Efficace surtout pour des données de même type et similaires

Requêtage limité

Exemples d'implémentation

APACHE HBASE

cassandra

Classement des moteurs orientés colonne

☐ include secondary database models

13 systems in ranking, February 2025

Rank			DBMS	Database Model	Score		
Feb 2025	Jan 2025	Feb 2024			Feb 2025	Jan 2025	Feb 2024
1.	1.	1.	Apache Cassandra	Wide column, Multi-model	102.58	+3.39	-6.69
2.	2.	2.	Apache HBase	Wide column	24.48	-0.27	-9.47
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	22.13	-0.83	-9.86
4.	4.	4.	Datastax Enterprise	Wide column, Multi-model	4.22	-0.26	-3.43
5.	5.	5.	ScyllaDB	Wide column, Multi-model	3.65	-0.01	-2.51
6.	6.	6.	Microsoft Azure Table Storage	Wide column	2.73	-0.09	-2.92
7.	7.	↑ 8.	Google Cloud Bigtable	Multi-model	2.70	-0.09	-1.46
8.	8.	↓ 7.	Apache Accumulo	Wide column	2.62	-0.17	-1.59
9.	9.	↑ 10.	Amazon Keyspaces	Wide column	0.97	-0.01	+0.12
10.	10.	↓ 9.	HPE Ezmeral Data Fabric	Multi-model	0.76	-0.03	-0.45
11.	11.	11.	Elassandra	Wide column, Multi-model	0.29	0.00	-0.01
12.	12.	12.	Alibaba Cloud Table Store	Wide column	0.21	+0.00	+0.02
13.	13.	13.	SWC-DB	Wide column, Multi-model	0.01	0.00	0.00

NoSQL : Modèle document

Orientées Documents

reposent également sur le paradigme [clé, valeur], mais la valeur, dans ce cas, est un document (JSON ou XML) sur lequel il est possible requêter.

Exemples de données orientées documents

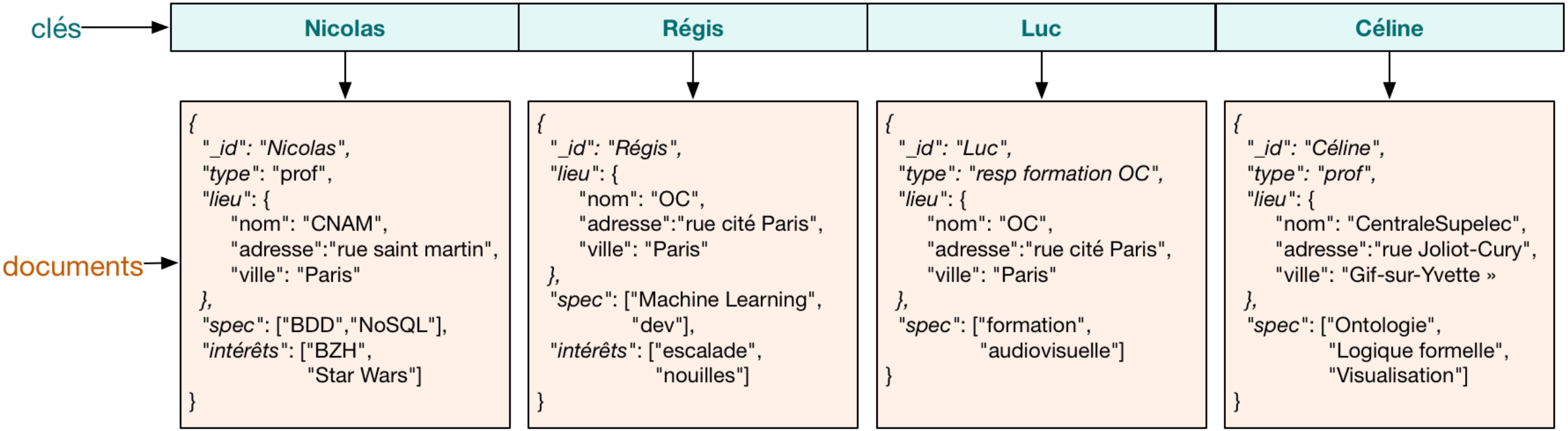


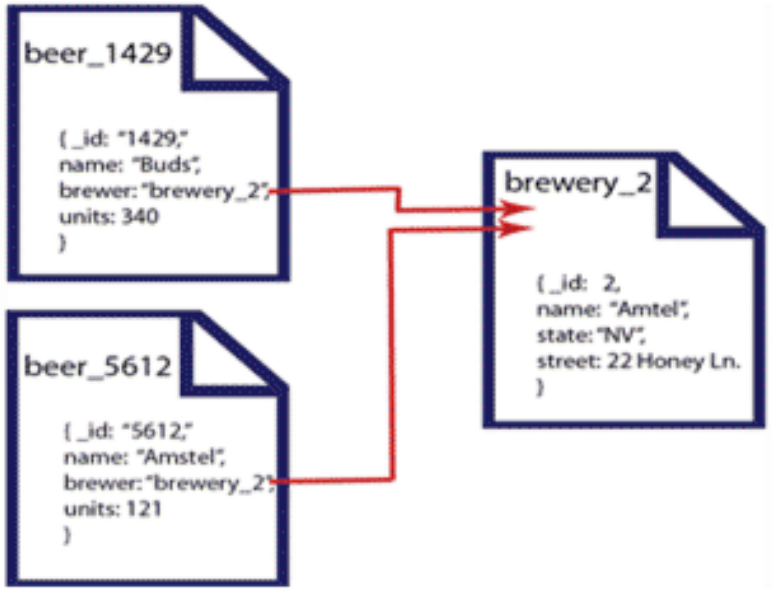
Figure reprise de https://openclassrooms.com/courses/4462426-maitrisez-les-bases-de-donnees-nosql

Modèle orienté document : principes

- Documents structurés mais sans définition de structure préalable nécessaire
- Possibilité de requêter et manipuler ces documents, et notamment récupérer, via une seule clé, un ensemble d'informations structurées de manière hiérarchique (en relationnel, cette opération nécessite plusieurs jointures qui sont très coûteuse en ressources)
- Pas adapté pour les données interconnectées ni pour les données non-structurées
- Exemple d'applications : données clients (stockage de toutes les transactions et information du client (liées à une même clé) au sein d'un même document), gestion de catalogue de produits …

©Maude Manouvrier - Univ. Paris-Dauphine - Repris er adapté de http://www.illustradata.com/bases-nosql-documents-quest-cest/

Modèle orienté document : avantages, inconvénients et principaux moteurs



Exemples d'implémentation

CouchDB relax

mongoDB

Amazon DynamoDB (Beta)

+

Requétage plus complet

Flexibilité

Evolutif au cours du temps

−

Duplication des données

Cohérence pas forcément assurée

Classement des moteurs orientés document

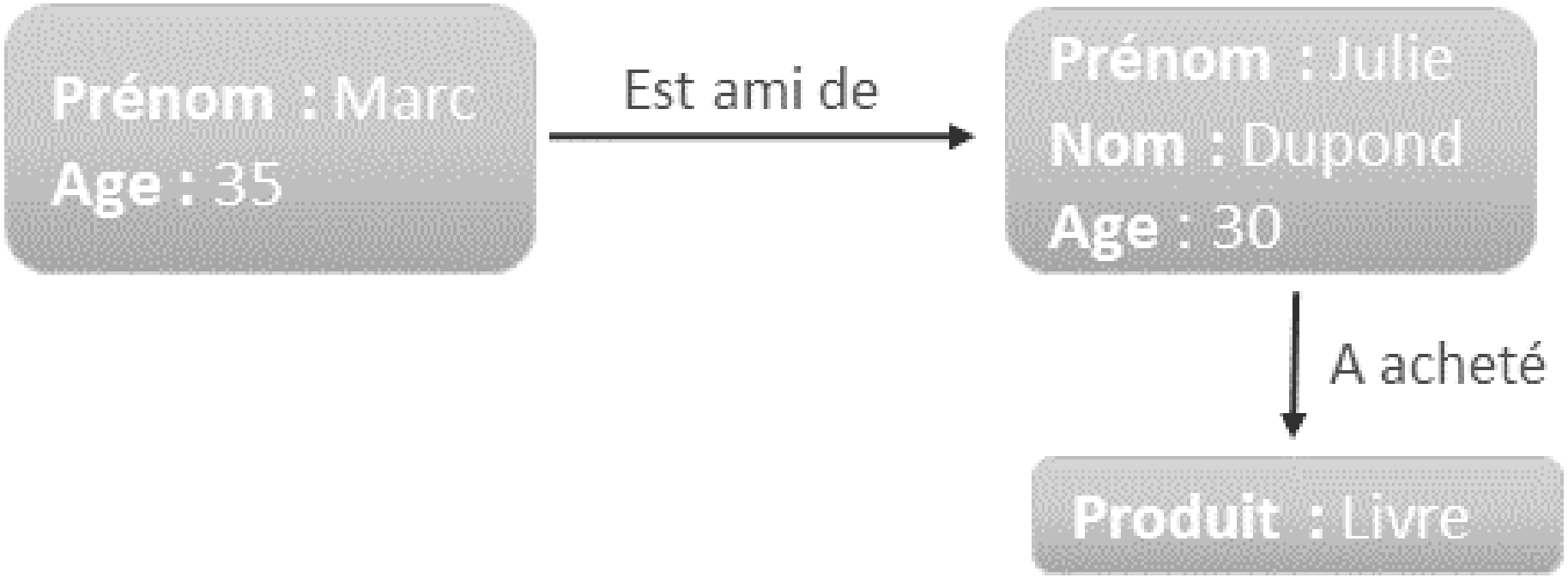
☐ include secondary database models — 58 systems in ranking, February 2025

Rank Feb 2025	Rank Jan 2025	Rank Feb 2024	DBMS	Database Model	Score Feb 2025	Score Jan 2025	Score Feb 2024
1.	1.	1.	MongoDB	Document, Multi-model	396.63	-5.87	-23.73
2.	2.	↑ 3.	Databricks	Multi-model	90.03	+2.19	+13.13
3.	3.	↓ 2.	Amazon DynamoDB	Multi-model	75.58	+2.58	-7.31
4.	4.	4.	Microsoft Azure Cosmos DB	Multi-model	22.13	-0.83	-9.86
5.	5.	5.	Couchbase	Multi-model	15.63	-0.38	-4.85
6.	6.	6.	Firebase Realtime Database	Document	13.32	+0.21	-3.02
7.	7.	7.	CouchDB	Document, Multi-model	7.81	+0.03	-4.86
8.	↑ 9.	8.	Google Cloud Firestore	Document	6.92	+0.19	-3.95
9.	↓ 8.	9.	Realm	Document	6.74	-0.25	-1.22
10.	10.	↑ 11.	Aerospike	Multi-model	5.09	+0.05	-2.00
11.	11.	↓ 10.	MarkLogic	Multi-model	4.16	+0.21	-3.33
12.	12.	↑ 13.	Google Cloud Datastore	Document	3.69	-0.11	-1.60
13.	↑ 14.	↑ 17.	Oracle NoSQL	Multi-model	3.29	+0.05	-0.50
14.	↓ 13.	↓ 12.	Virtuoso	Multi-model	3.14	-0.18	-2.18
15.	15.	↓ 14.	ArangoDB	Multi-model	2.91	-0.01	-1.43
16.	16.	↑ 18.	OrientDB	Multi-model	2.73	-0.06	-0.92

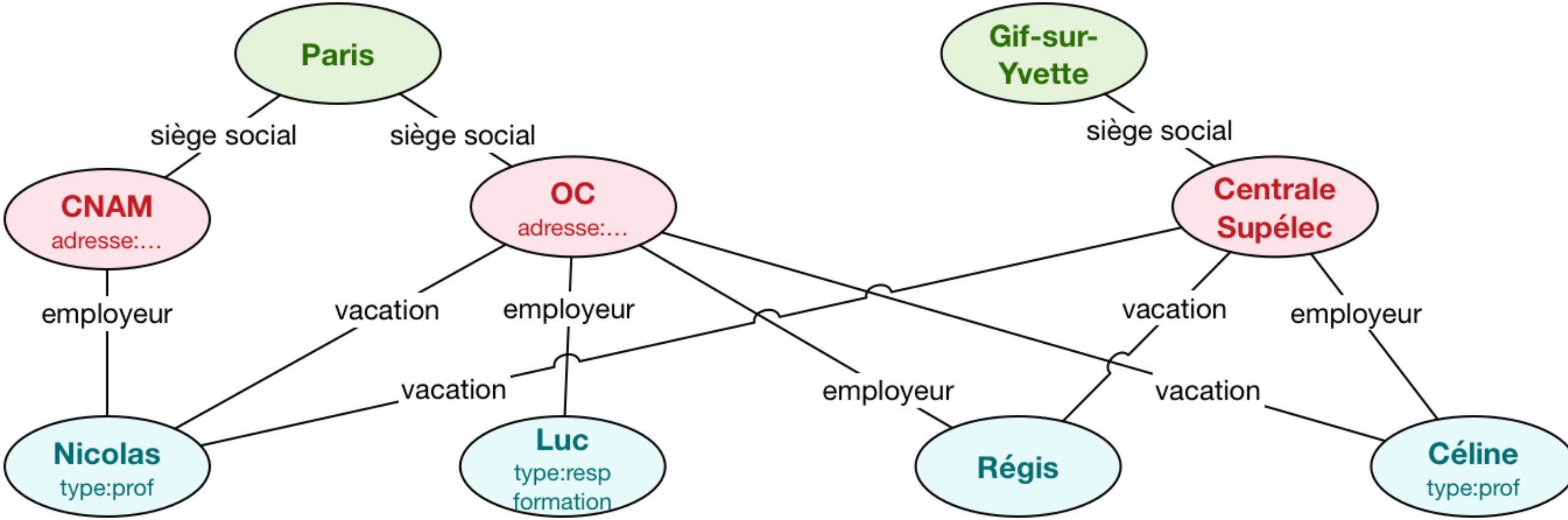
NoSQL : Modèle graphe

Orientées Graphes

stockent les données en se basant sur la théorie des graphes. Elles s'appuient sur la notion de nœuds et de relations.



Données orientées graphe : Exemple

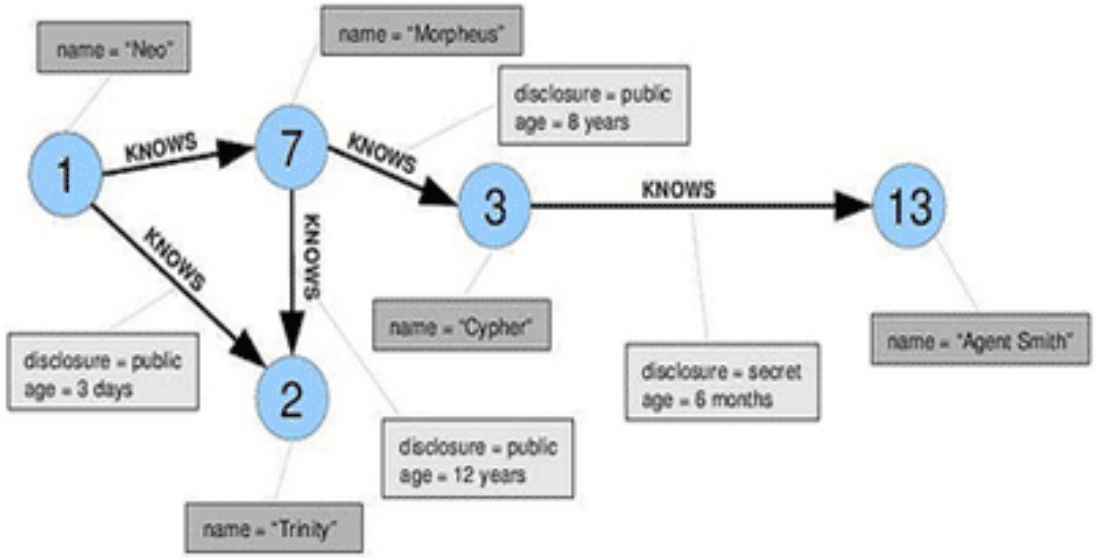


©Maude Manouvrier - Univ. Paris-Dauphine - Figure reprise de https://openclassrooms.com/courses/4462426-maitrisez-les-bases-de-donnees-nosql

Modèle orienté graphe : principes

- Basé sur la théorie des graphes
- Adapté aux objets complexes organisés en réseaux, aux données présentant des dépendances fortes
- Beaucoup plus rapides que les autres systèmes de stockage pour manipuler les données fortement connectées
- Non adapté pour tous les autres contextes que celui des “données fortement connectées”
- Applications : modélisation des données provenant des réseaux sociaux (Twitter, Facebook…), moteur de recommandation, Données géo spatiales (réseaux ferrés etc.) …

Modèle orienté graphe : avantages, inconvénients et principaux moteurs



Adapté à la gestion de données relationnelles

Algorithme de parcours de graphes

Architecture limitée à certains cas

Non adapté aux opération sur les ensembles

Exemple d'implémentation

TITAN

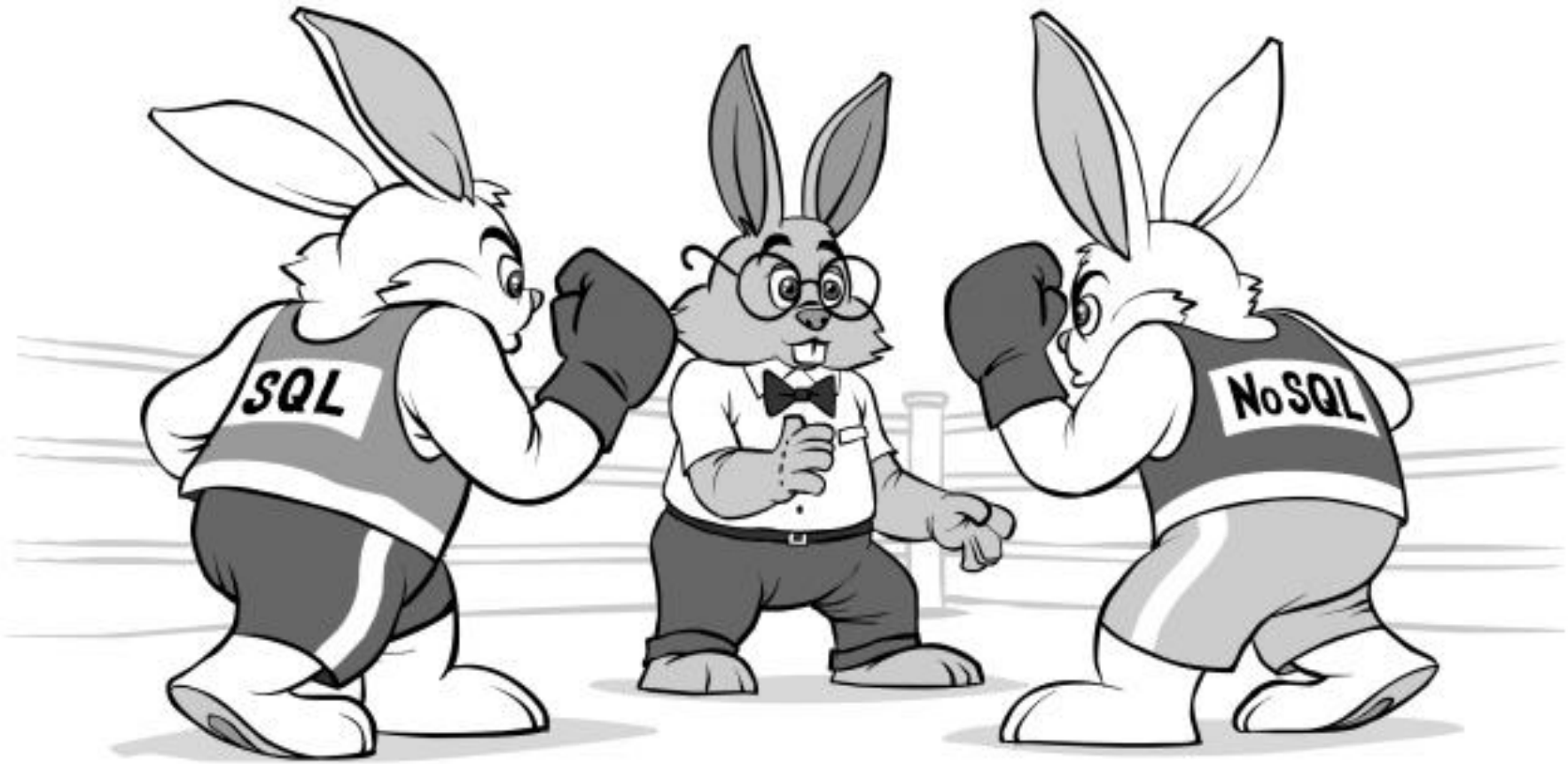
Neo4j

Classement des moteurs orientés graphe

☐ include secondary database models — 43 systems in ranking, February 2025

Rank Feb 2025	Rank Jan 2025	Rank Feb 2024	DBMS	Database Model	Score Feb 2025	Score Jan 2025	Score Feb 2024
1.	1.	1.	Neo4j	Graph	45.34	+1.65	-1.27
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	22.13	-0.83	-9.86
3.	3.	3.	Aerospike	Multi-model	5.09	+0.05	-2.00
4.	4.	4.	Virtuoso	Multi-model	3.14	-0.18	-2.18
5.	5.	5.	ArangoDB	Multi-model	2.91	-0.01	-1.43
6.	6.	6.	OrientDB	Multi-model	2.73	-0.06	-0.92
7.	7.	↑ 9.	GraphDB	Multi-model	2.63	-0.09	-0.12
8.	8.	↓ 7.	Memgraph	Graph	2.54	-0.08	-0.63
9.	9.	↓ 8.	Amazon Neptune	Multi-model	2.12	-0.07	-0.81
10.	10.	↑ 11.	JanusGraph	Graph	1.69	-0.03	-0.61
11.	11.	↑ 12.	Stardog	Multi-model	1.61	-0.07	-0.52
12.	12.	↓ 10.	NebulaGraph	Graph	1.58	-0.08	-0.96
13.	13.	↑ 14.	Fauna	Multi-model	1.49	+0.07	-0.30
14.	14.	↓ 13.	TigerGraph	Graph	1.45	+0.03	-0.52

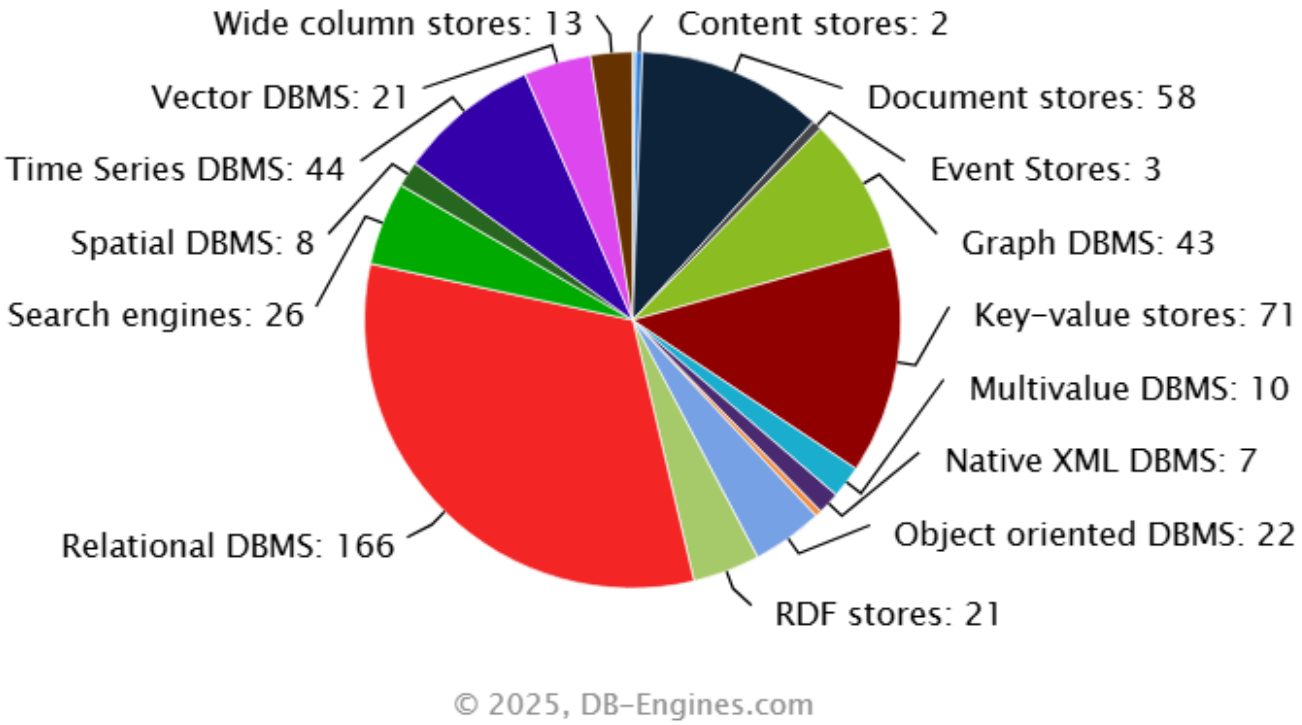
SQL vs NoSQL



SQL vs NoSQL : nombre de moteurs

DBMS popularity broken down by database model

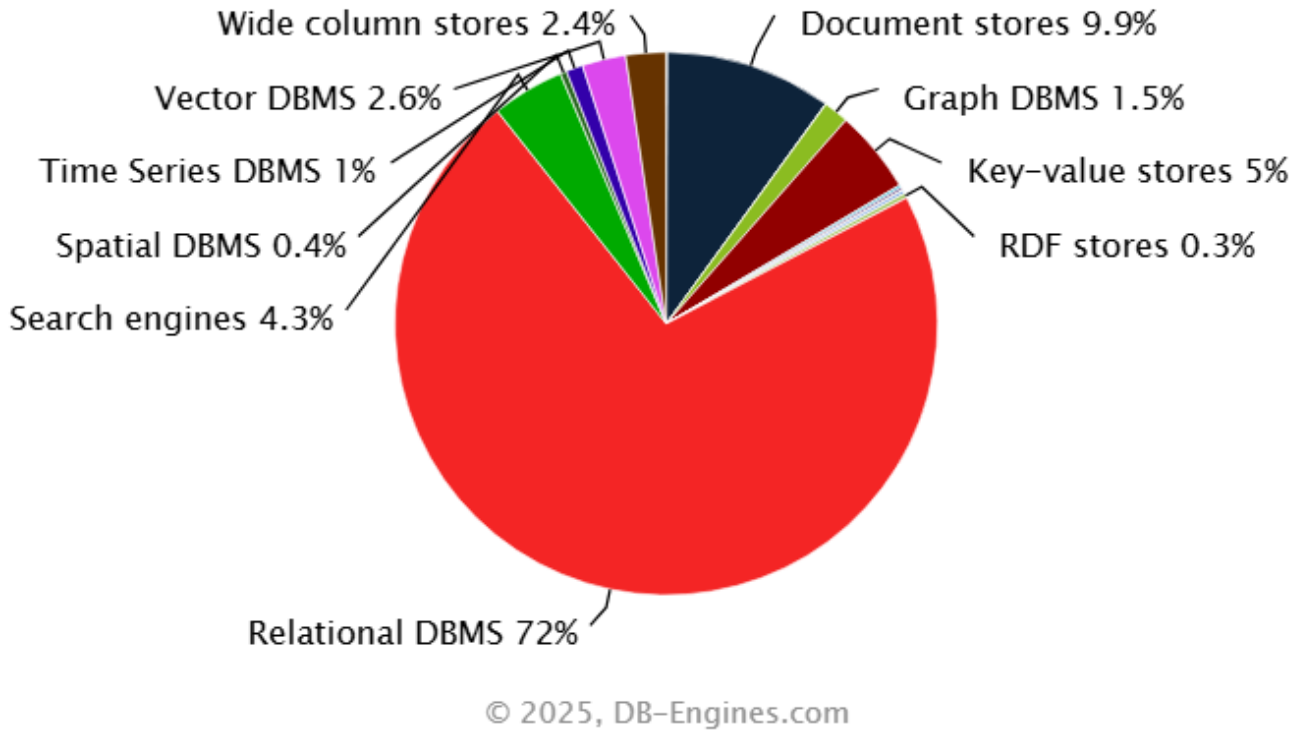
Number of systems per category, February 2025



DB-Engines lists 424 different database management systems, which are classified according to their database model (e.g. relational DBMS, key-value stores etc.).
This pie-chart shows the number of systems in each category. Some of the systems belong to more than one category.

SQL vs NoSQL : popularité

Ranking scores per category in percent, February 2025



This chart shows the popularity of each category. It is calculated with the popularity (i.e. the ranking scores) of all individual systems per category. The sum of all ranking scores is 100%.

SQL vs NoSQL : les points forts de chaque modèle

SQL

- Généraliste
- Logique : schémas, contraintes
- Cohérence forte
- Ressources limitées
- Langage standard: SQL
- Traitements centralisés

Cohérence

NoSQL

- Systèmes sur mesure
- Pas de schéma ni contrainte
- Cohérence « à terme »
- Ressources « illimitées »
- Langages spécialisés, API
- Traitements distribués (MapReduce)

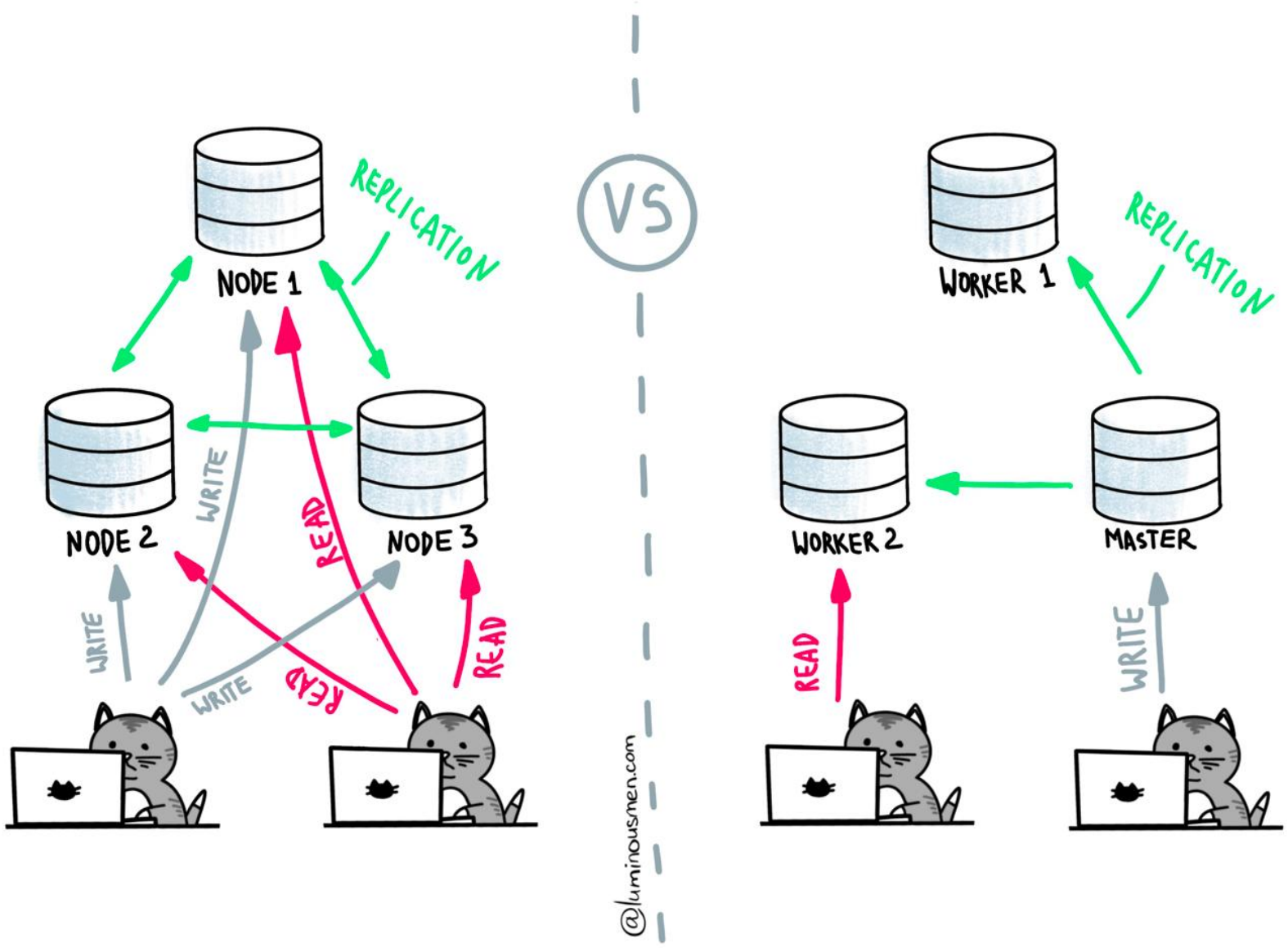
Disponibilité

SQL vs NoSQL : ACID vs BASE

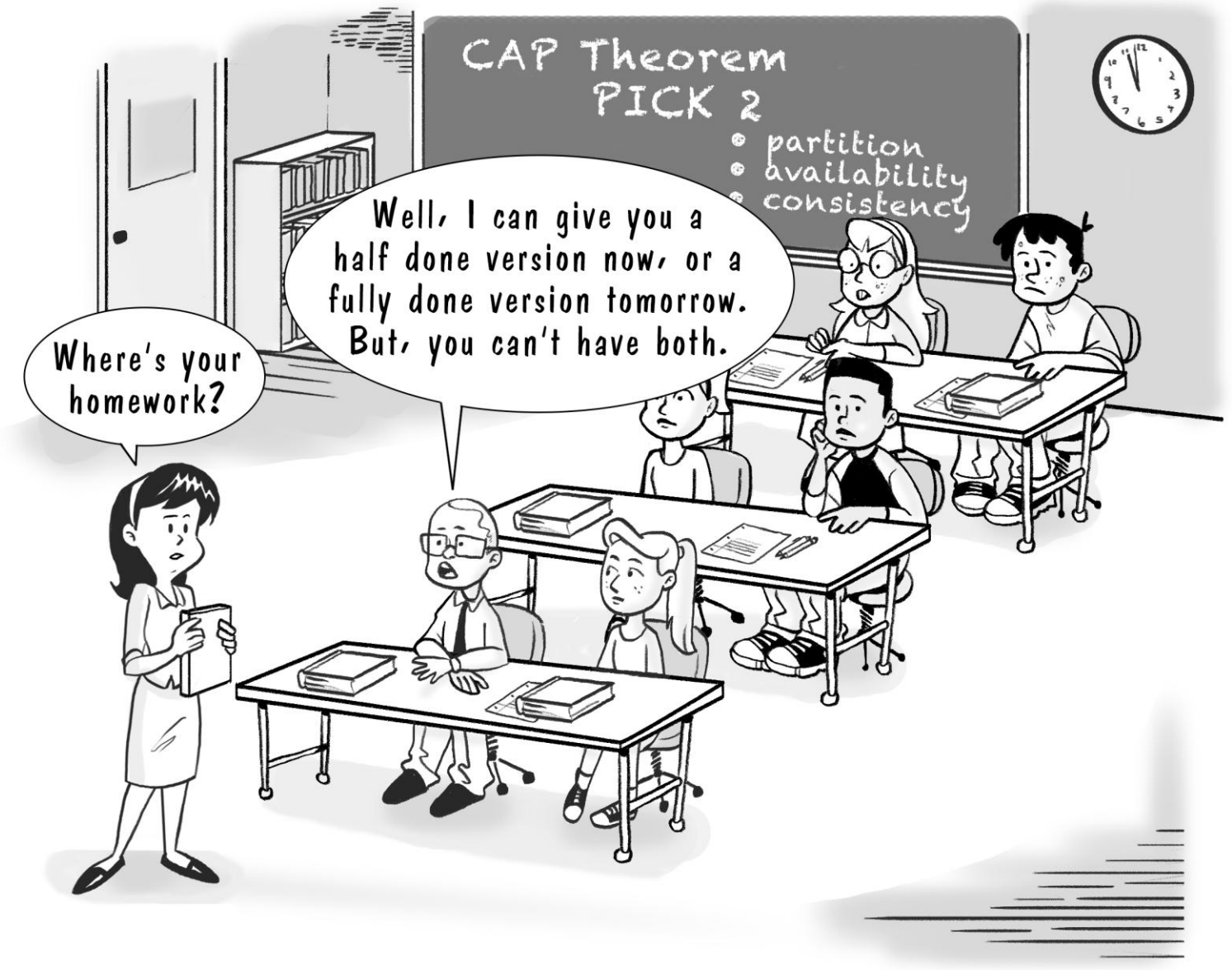
- **A***tomicité* : une transaction s'effectue entièrement ou pas du tout
- **C***ohérence* : le contenu d'une base doit être cohérent au début et à la fin d'une transaction
- **I***solation* : les modifications d'une transaction ne sont visibles/modifiables que quand celle-ci a été validée
- **D***urabilité* : une fois la transaction validée, l'état de la base est permanent (non affecté par les pannes ou autre)

- **B***asically Available* : quelle que soit la charge de la base de données (données/requêtes), le système garantie un taux de disponibilité de la donnée (mais réponse à la requête pouvant contenir des erreurs)
- **S***oft-state* : l'état de la BD n'est pas garanti à un instant donné (les mises à jour ne sont pas immédiates)
- **E***ventually consistent* : la cohérence des données à un instant donné n'est pas primordiale mais à terme, la base atteindra un état cohérent

ACID / BASE


NODE 1
REPLICATION
NODE 2
NODE 3
WRITE
READ
WRITE
READ
WRITE
READ
VS
@luminousmen.com
WORKER 1
REPLICATION
WORKER 2
MASTER
READ
WRITE

Théorème de CAP

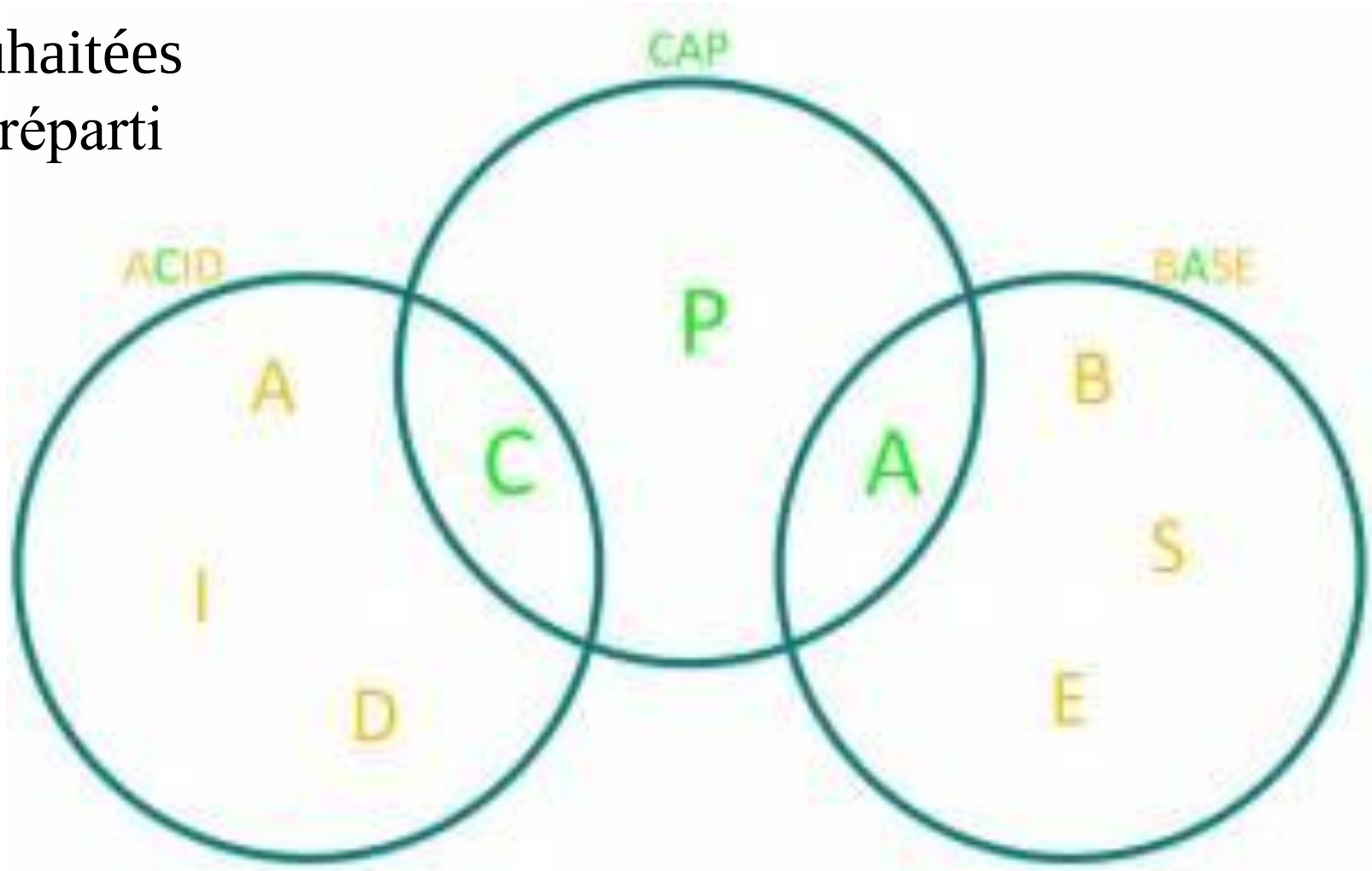


Brewer, Eric A. "Towards robust distributed systems." *PODC*. Vol. 7. No. 10.1145. 2000.

https://sites.cs.ucsb.edu/~rich/class/cs293b-cloud/papers/Brewer_podc_keynote_2000.pdf

Théorème de CAP : Cohérence, Disponibilité et Partitionnement

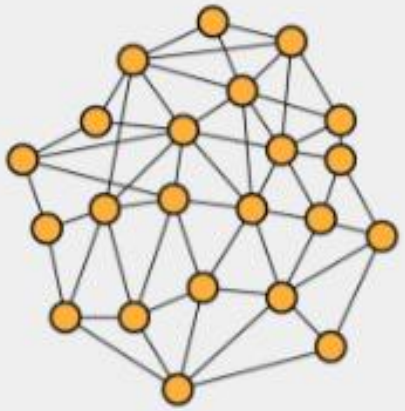
Propriétés souhaitées d'un système réparti *(distributed)*



Théorème de CAP :
Consistency, Availability, Partition Tolerence

C : Consistency

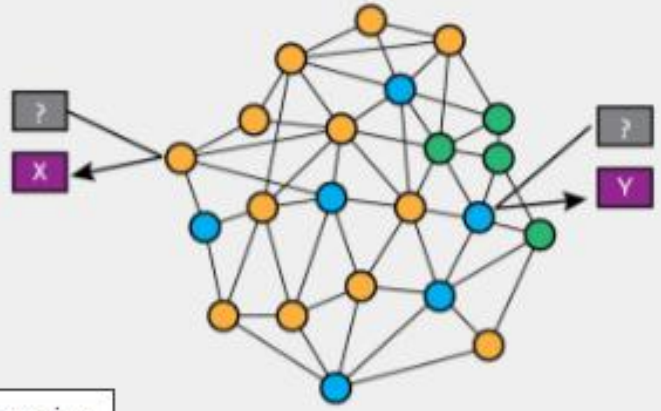
At any given time, all nodes in the network have exactly the same (most recent) value.



= Value: X @ 2018-05-03T08:52:40

A : Availability

Every request to the network receives a response, though without any guarantee that returned data is the most recent.



CAP Theorem - overview

= Value: X @ 2018-05-03T08:52:40

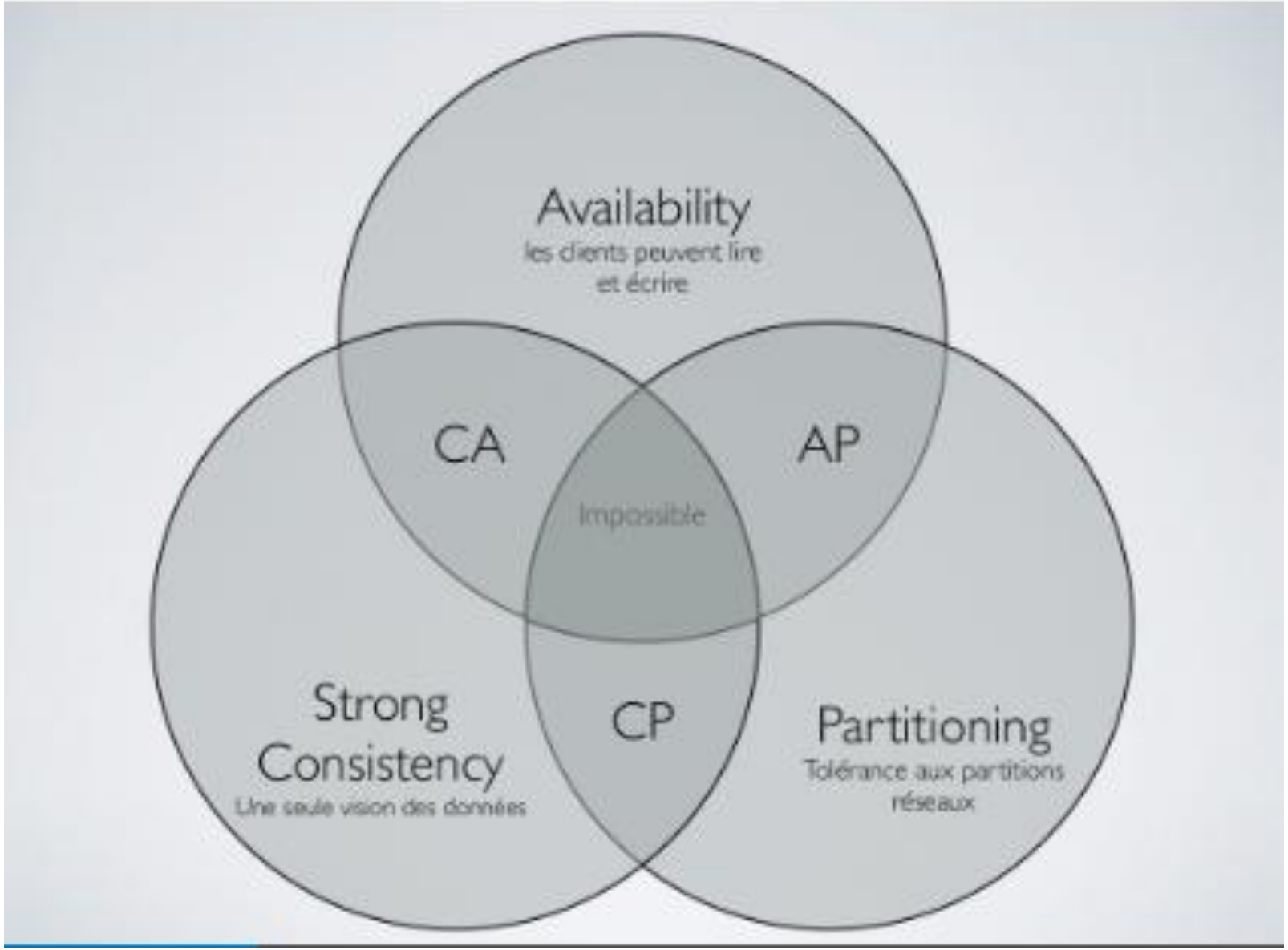
= Value: Z @ 2018-05-03T08:32:58

= Value: Y @ 2018-05-03T07:12:12

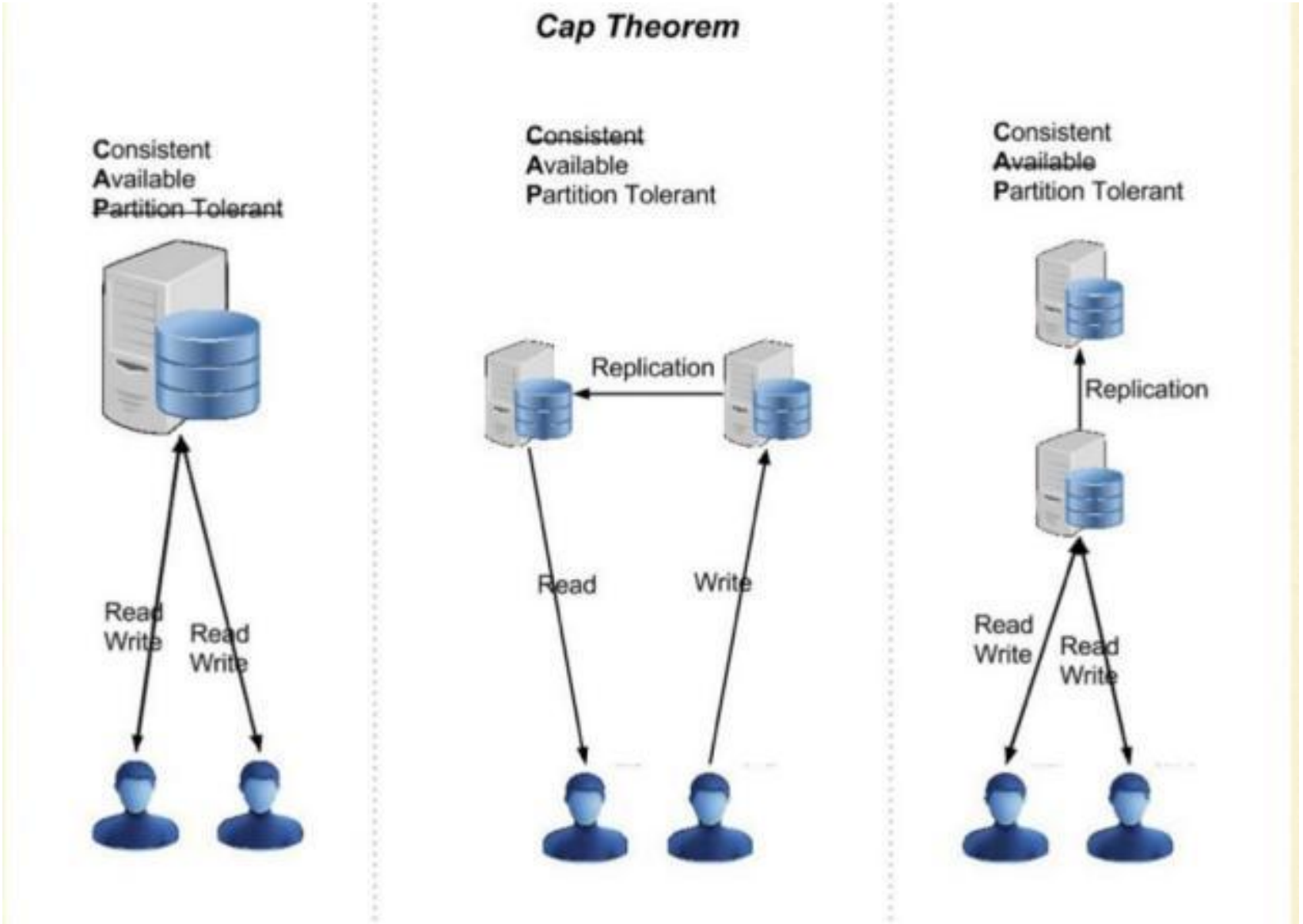
P : Partition tolerance

The network continues to operate, even if an arbitrary number of nodes are failing.

Théorème de CAP : impossibilité d'avoir les 3 propriétés



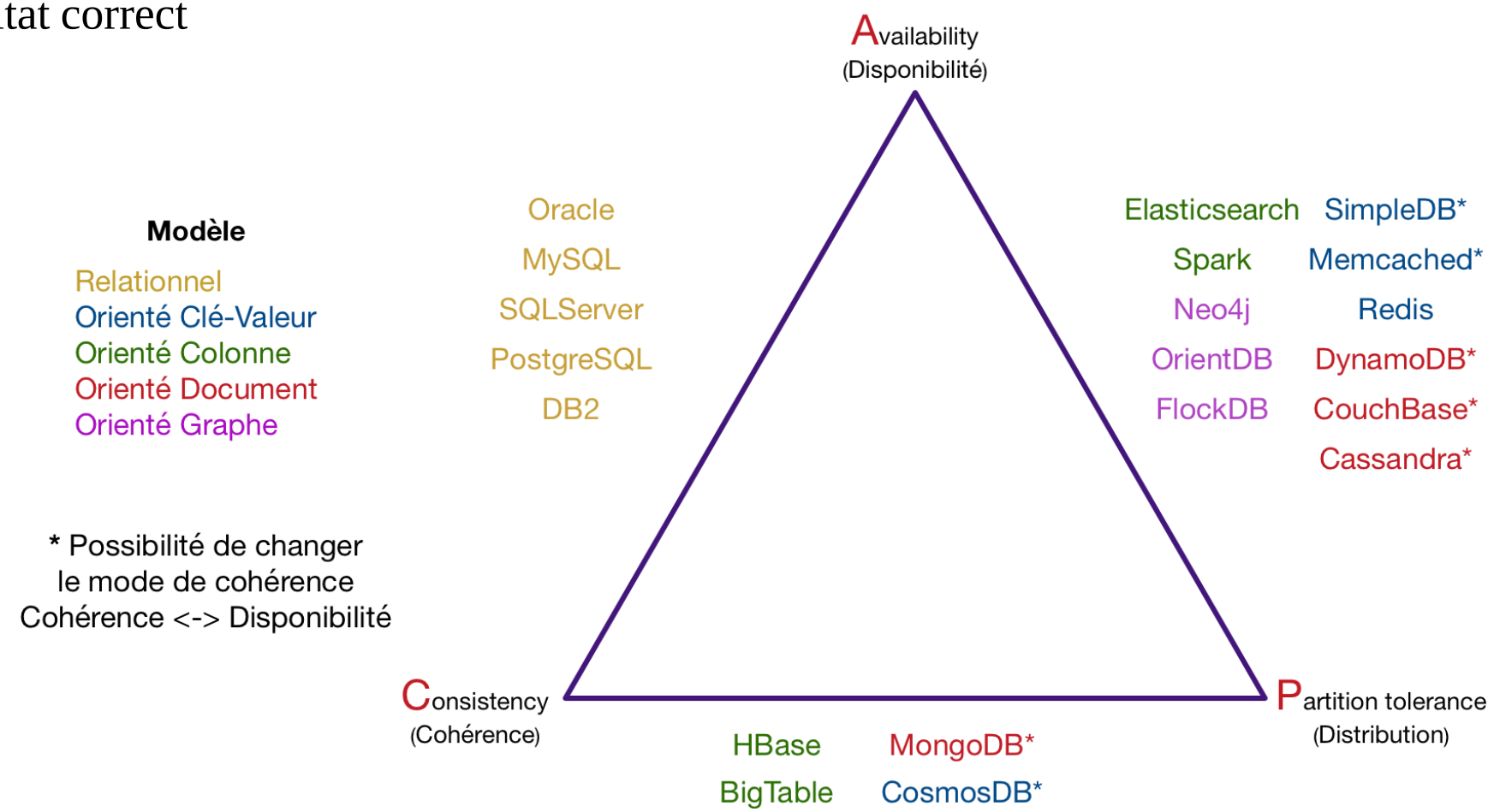
Théorème de CAP : 2 propriétés sur 3



©Maude Manouvrier - Univ. Paris-Dauphine – Repris de https://mytechnetknowhows.wordpress.com/2016/05/31/cap-theorem-consistency-availability-and-partition-tolerance/

Positionnement des moteurs selon CAP

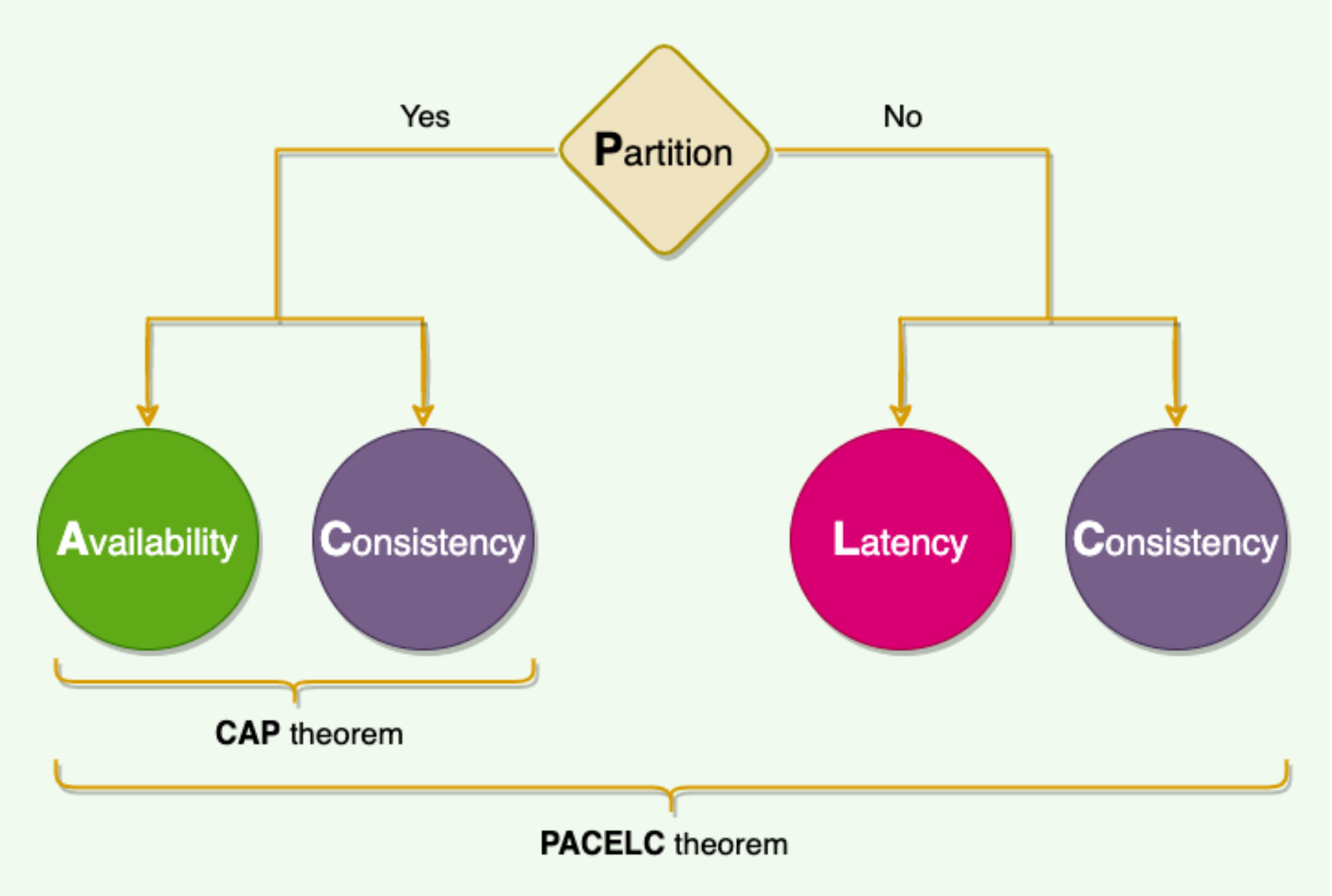
- **C***onsistency* (Cohérence) : une donnée n'a qu'un seul état visible quel que soit le nombre de réplicas
- **A***vailability* (Disponibilité) : tant que le système tourne (distribué ou non), la donnée doit être disponible
- **P***artition Tolerance* (Distribution) : quel que soit le nombre de serveurs, toute requête doit fournir un résultat correct

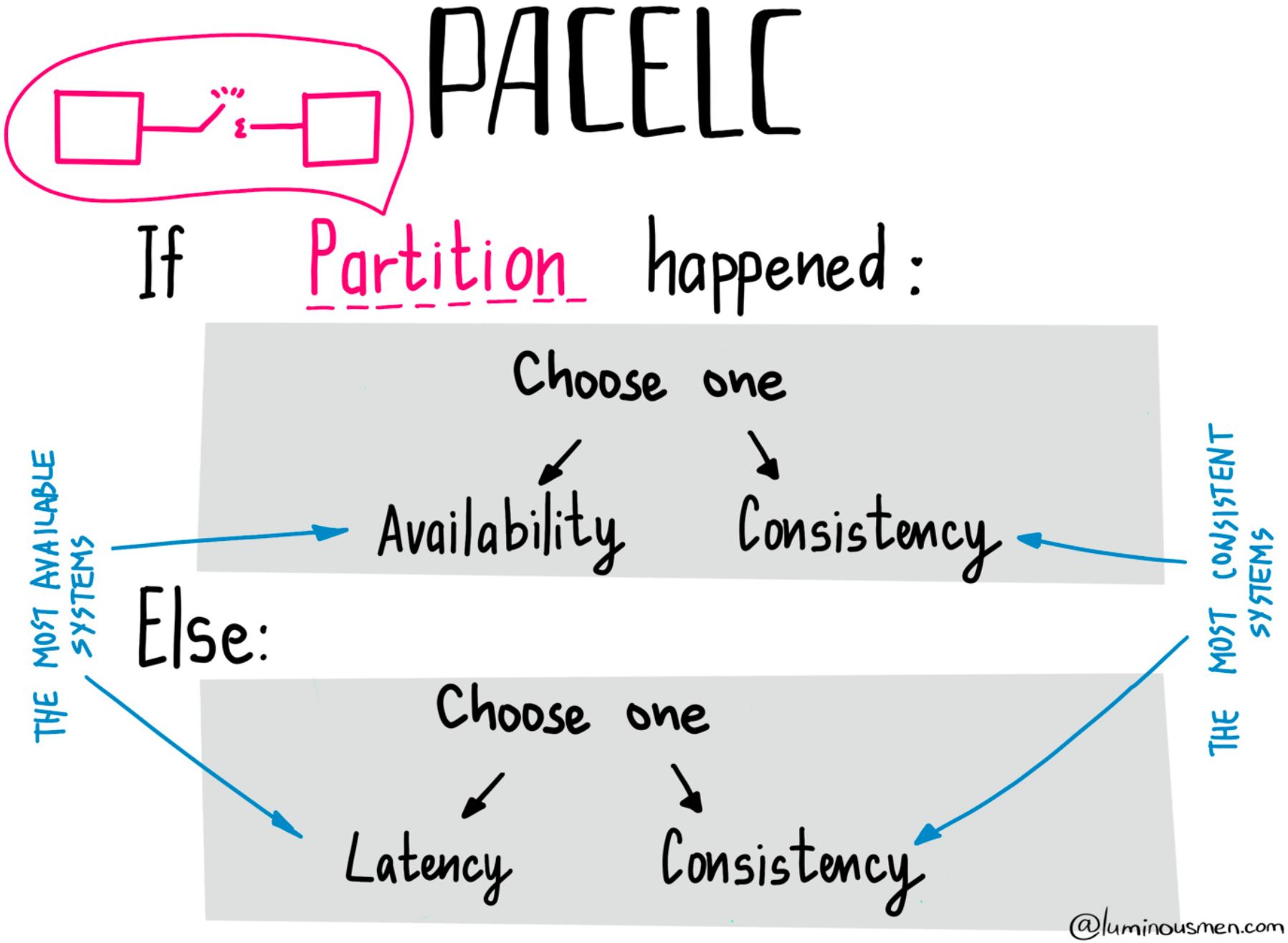


PACELC : alternative basée sur CAP incluant la latence

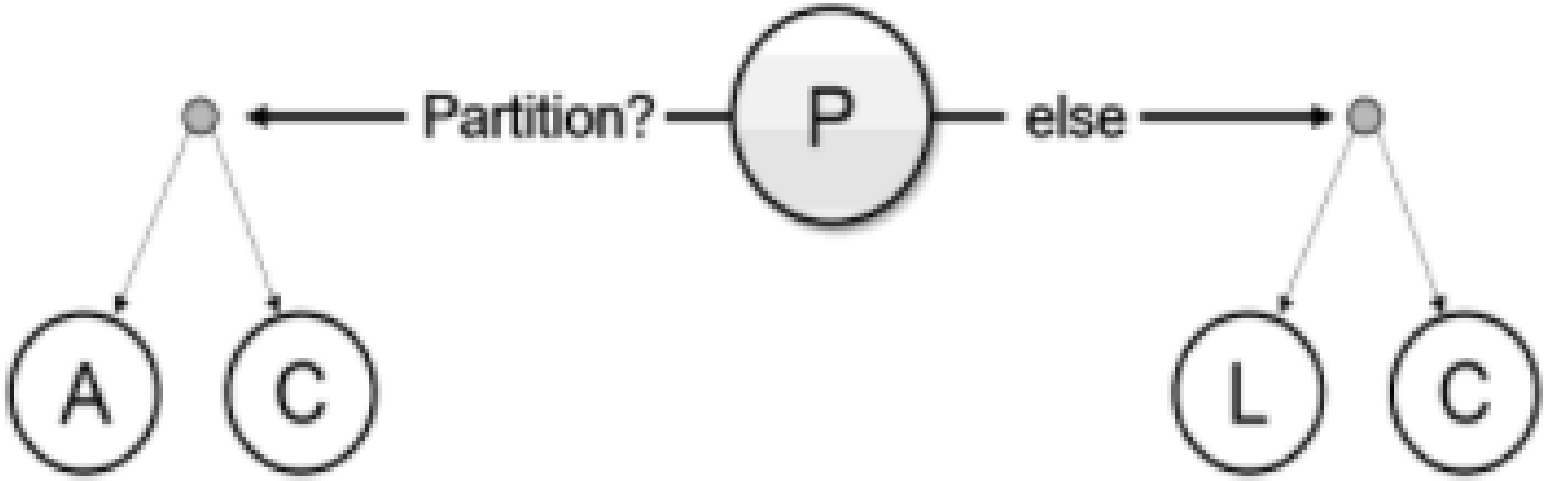
Abadi, Daniel. "*Consistency tradeoffs in modern distributed database system design: CAP is only part of the story*." *Computer* 45.2 (2012): 37-42.

https://www.cs.umd.edu/~abadi/papers/abadi-pacelc.pdf




PACELC
If Partition happened:
Choose one
Availability
Consistency
THE MOST AVAILABLE SYSTEMS
THE MOST CONSISTENT SYSTEMS
Else:
Choose one
Latency
Consistency
@luminousmen.com

PACELC : positionnement des moteurs



PA / EL
DynamoDB, Cassandra, Riak

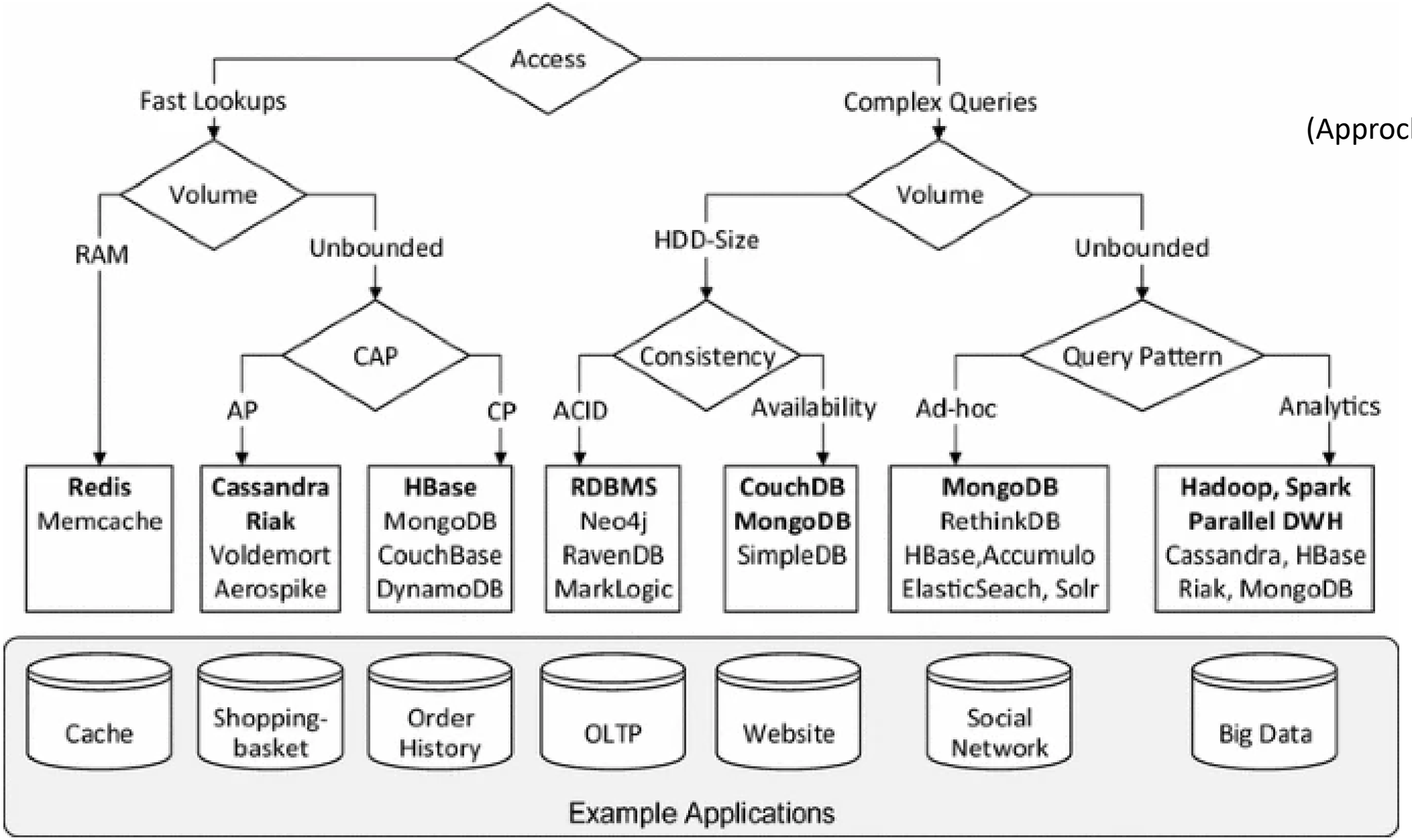
PC / EC
VoldDB, H-Store, Megastore, BigTable, Hbase

PA / EC
MongoDB

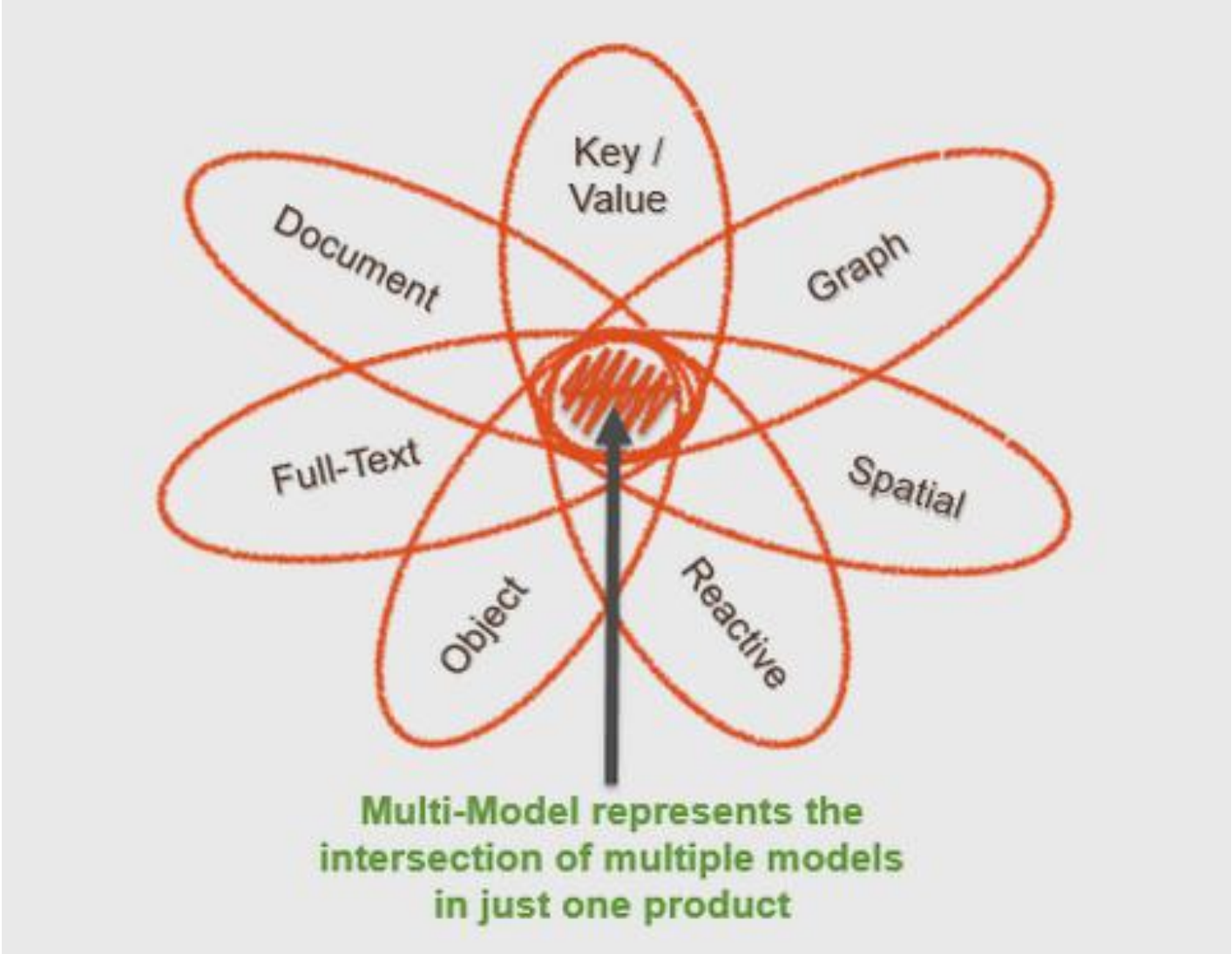
PC / EL
PNUTS

Choisir le bon modèle

(Approche de 2016)



Système multi-modèles



Classement des systèmes multi-modèles

424 systems in ranking, February 2025

Rank Feb 2025	Rank Jan 2025	Rank Feb 2024	DBMS	Database Model	Score Feb 2025	Score Jan 2025	Score Feb 2024
1.	1.	1.	Oracle	Relational, Multi-model			13.38
2.	2.	2.	MySQL	Relational, Multi-model			06.67
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model			66.70
4.	4.	4.	PostgreSQL	Relational, Multi-model			30.21
5.	5.	5.	MongoDB	Document, Multi-model			23.73
6.	↑ 7.	6.	Redis	Key-value, Multi-model			-2.80
7.	↓ 6.	↑ 9.	Snowflake	Relational	155.58	+1.68	+28.13
8.	8.	↓ 7.	Elasticsearch	Multi-model	134.63	-0.29	-1.11
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model	125.44	+2.46	-6.79

Relational DBMS
Document store
Graph DBMS
RDF store
Spatial DBMS
Vector DBMS

Bases de données polyglottes

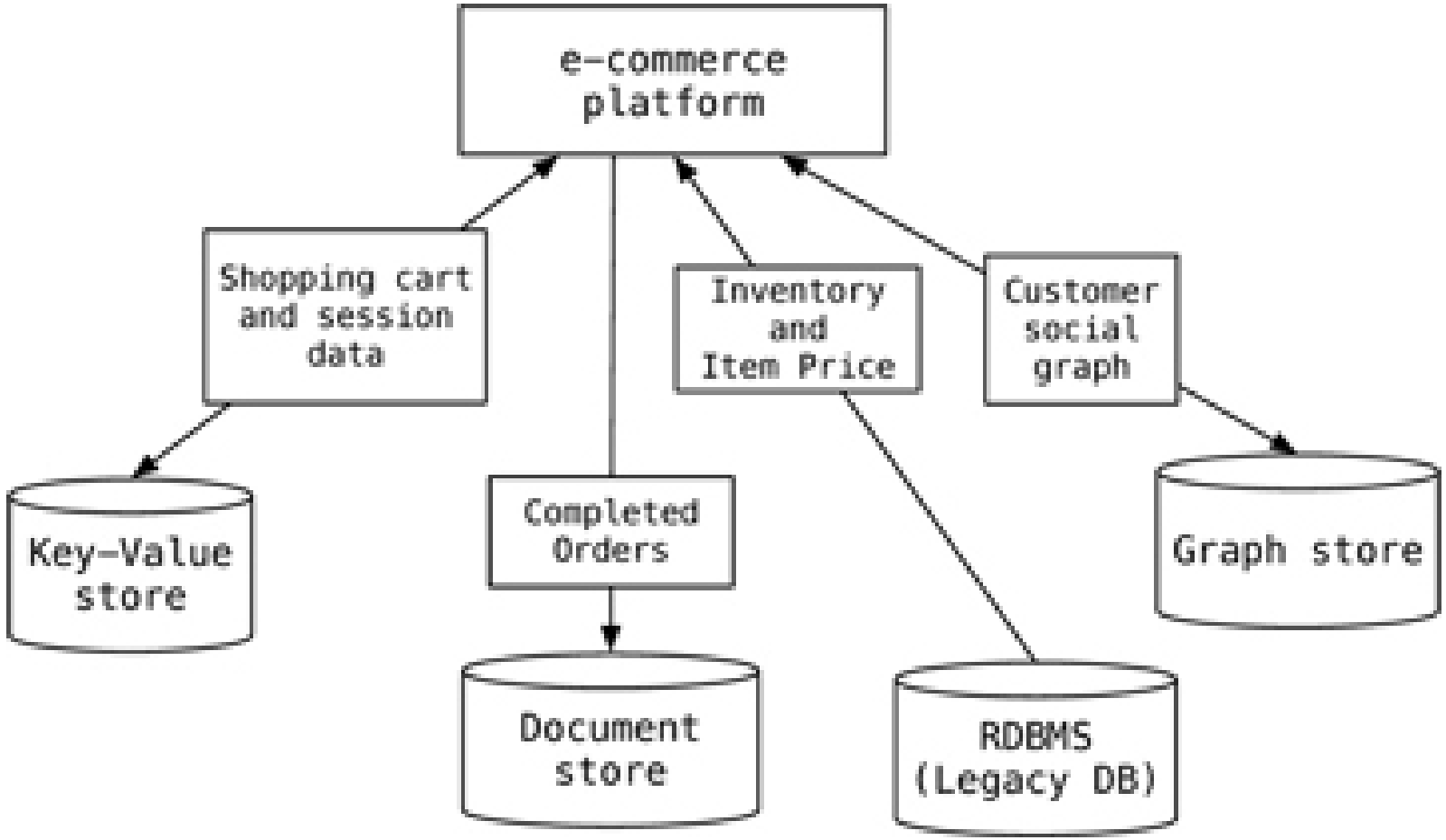
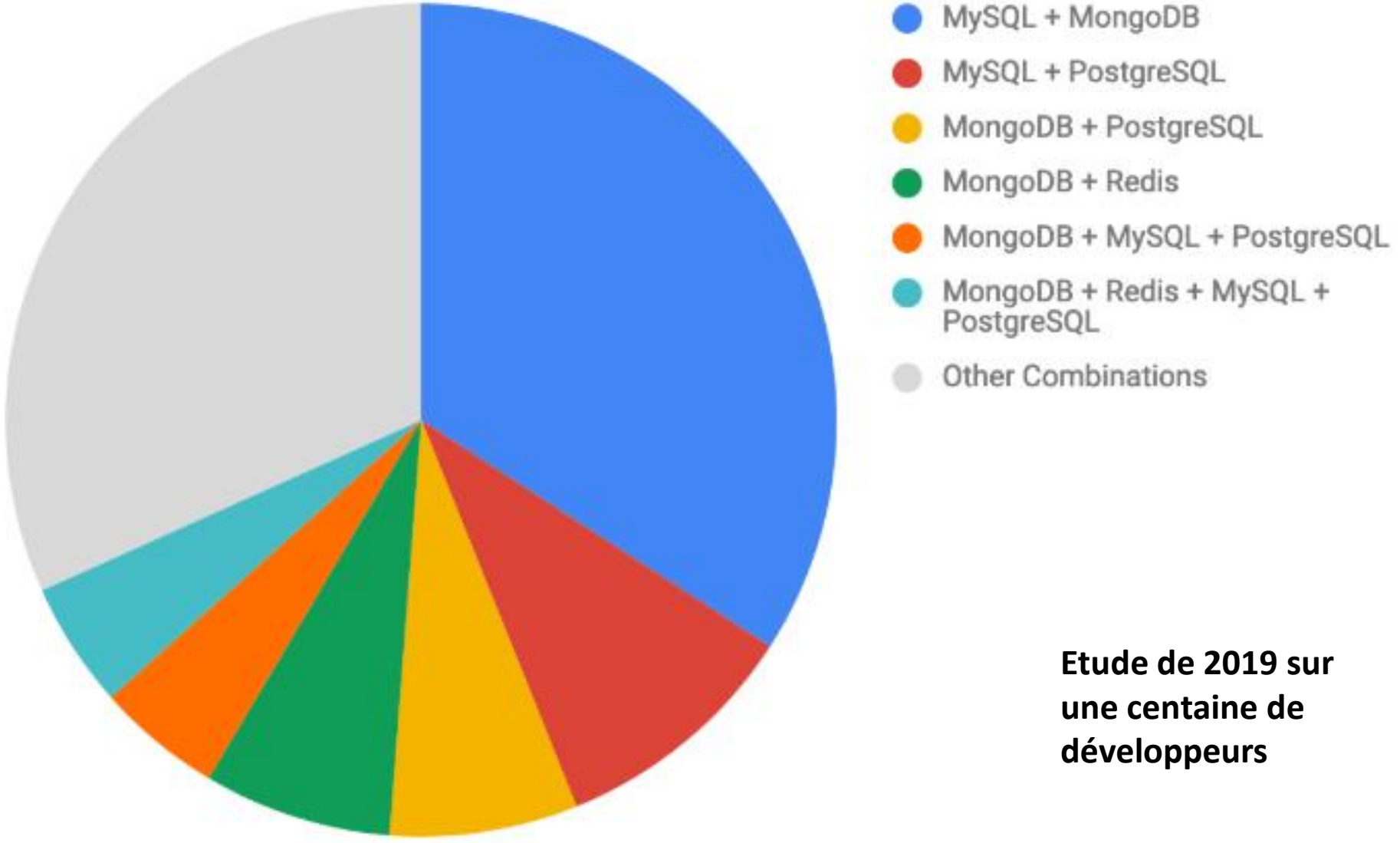


Figure reprise de P. J. Sadalage et M. Fowler (2009). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison Wesley

https://martinfowler.com/bliki/PolyglotPersistence.html

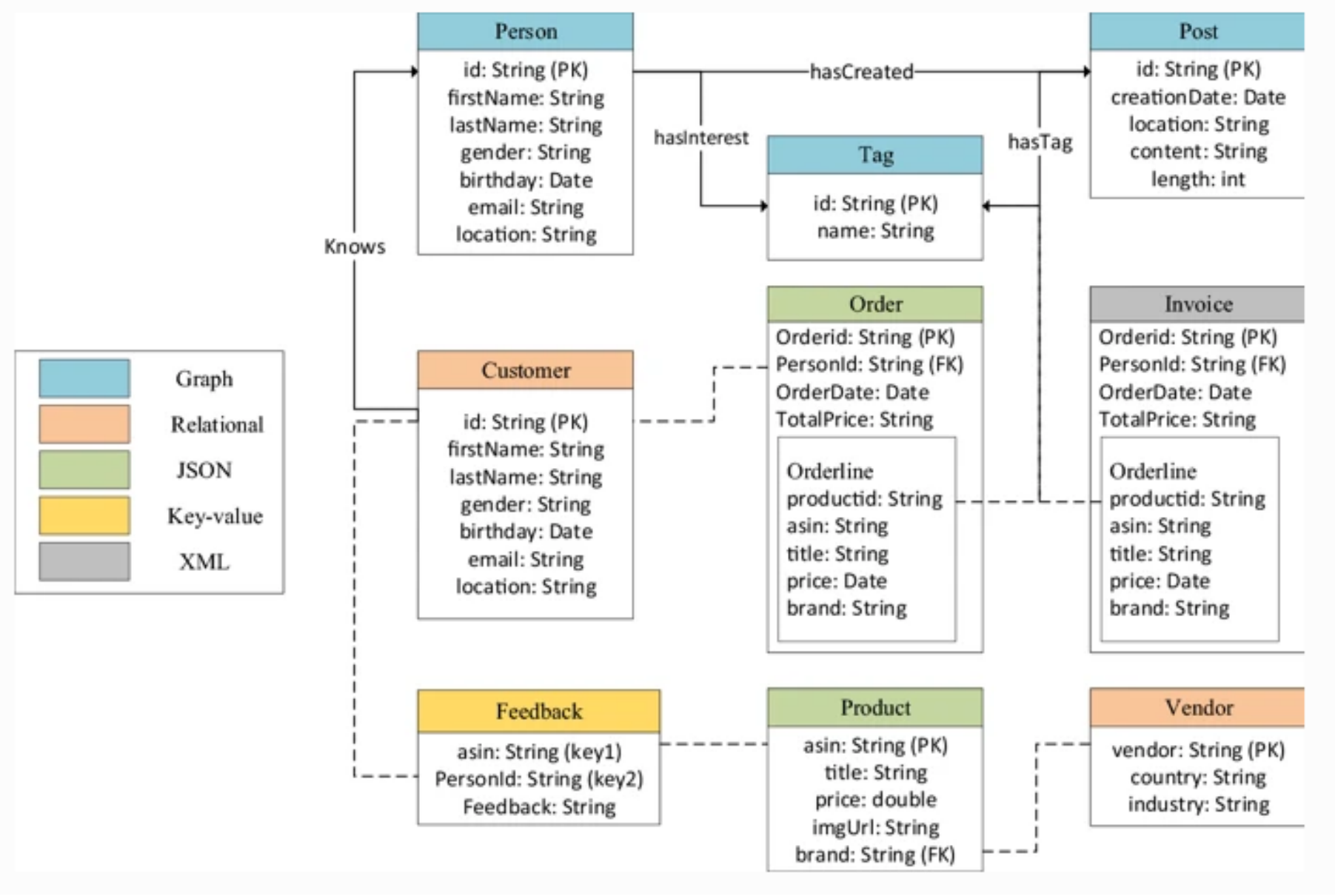
©Maude Manouvrier - Univ. Paris Dauphine

Bases de données polyglottes : tendances



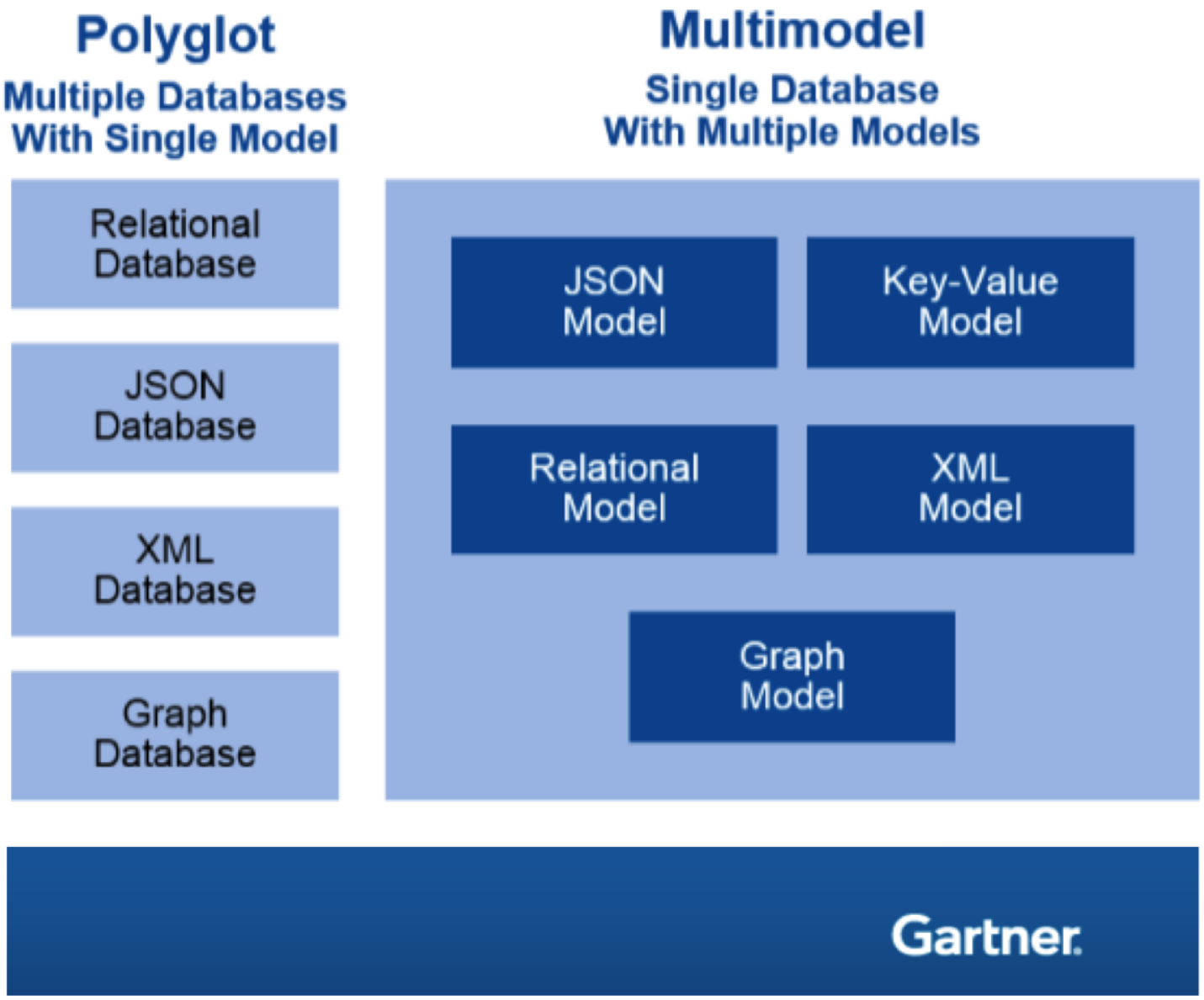
Etude de 2019 sur une centaine de développeurs

Exemple de base polyglotte



©Maude Manouvrier - Univ. Paris Dauphine - https://link.springer.com/article/10.1007/s10619-019-07279-6?shared-article-renderer#Tab1

Bases de données polyglottes versus multi-modèles



©Maude Manouvrier - Univ. Paris Dauphine – Figure reprise de https://avanttic.com/en/blog/polyglot-persistence/

NewSQL

PROPERTIES	TRADITIONAL SQL	NOSQL	NEWSQL
ACID PROPERTY	✓	✗	✓
IN MEMORY DB	✗	✓	✓
BIG DATA	✗	✓	✓
RDBMS	✓	✗	✓

Stonebraker, Michael. "*Newsql: An alternative to nosql and old sql for new oltp apps*." *Communications of the ACM. Retrieved* (2012): 07-06.
https://cacm.acm.org/blogs/blog-cacm/109710-new-sql-an-alternative-to-nosql-and-old-sql-for-new-oltp-apps/fulltext

NewSQL : définition et principaux moteurs

"A class of modern relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing (OLTP) read-write workloads while still maintaining the ACID guarantees of a traditional database system." https://en.wikipedia.org/wiki/NewSQL

- VoltDB

- NuoDB

- Google Cloud Spanner

- CockroachDB

- YugabyteDB

- MariaDB

NewSQL : cas d'utilisation

- En cas de nécessité de :
  - Langages de requêtes riches (jointure, agrégation)
  - Conformité aux propriétés ACID
  - Performances supérieures à celles des SGBDR classiques
- Pour se débarrasser des goulots d'étranglement classiques des SGBDR : verrous, journalisation, gestion des caches
- Besoin de bases de données en mémoire vive, avec copie sur disque asynchrone
- Gestion de concurrence sans verrou (MVCC)
- Architecture distribuée sans partage d'information (*shared nothing*) et avec équilibrage de charge transparent

©Maude Manouvrier - Univ. Paris-Dauphine - Repris de https://www.fun-mooc.fr/c4x/MinesTelecom/04006/asset/Télécom_FBD_S2_Basesdedonnéesrelationnelles_V1.pdf

NewSQL : avantages et inconvénients

Passage à l'échelle, haute disponibilité et élasticité : adaptation dynamique à la distribution des données et à la structure du réseau de machines

Propriétés ACID

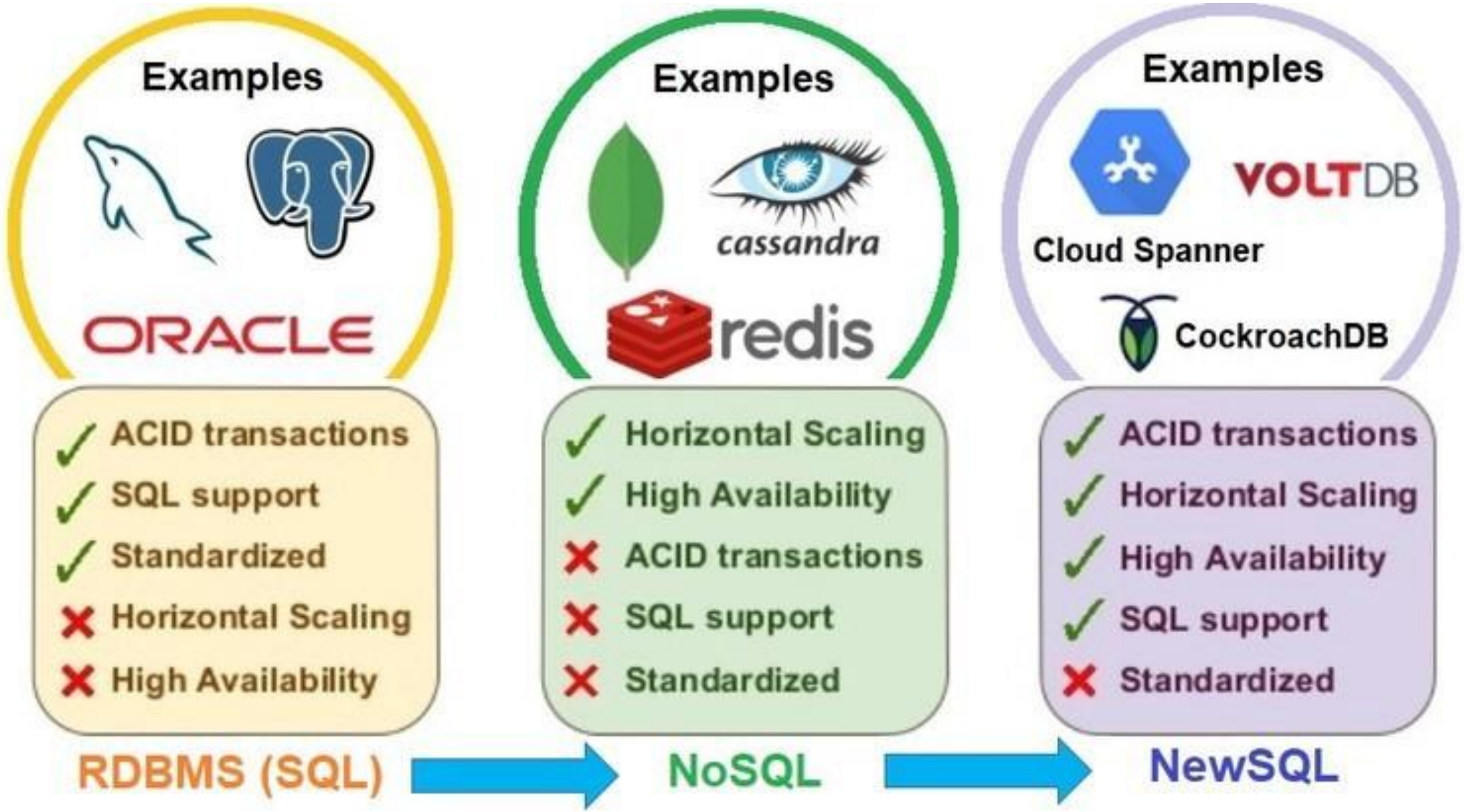
Gestion et optimisation des requêtes SQL

Rigidité du modèle de données

Moins de fonctionnalités que les SGBD SQL ou NoSQL

Systèmes complexes et nouveaux

SQL – NoSQL – NewSQL : avantages et inconvénients



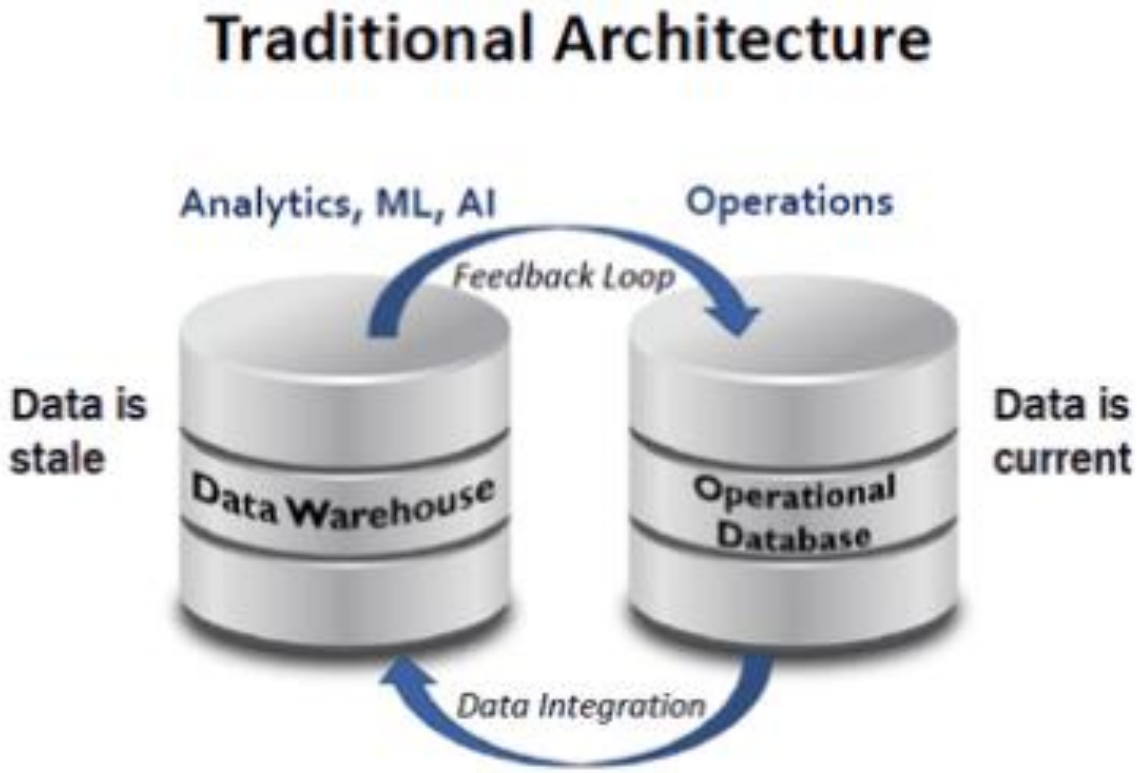
SQL – NoSQL – NewSQL : caractéristiques

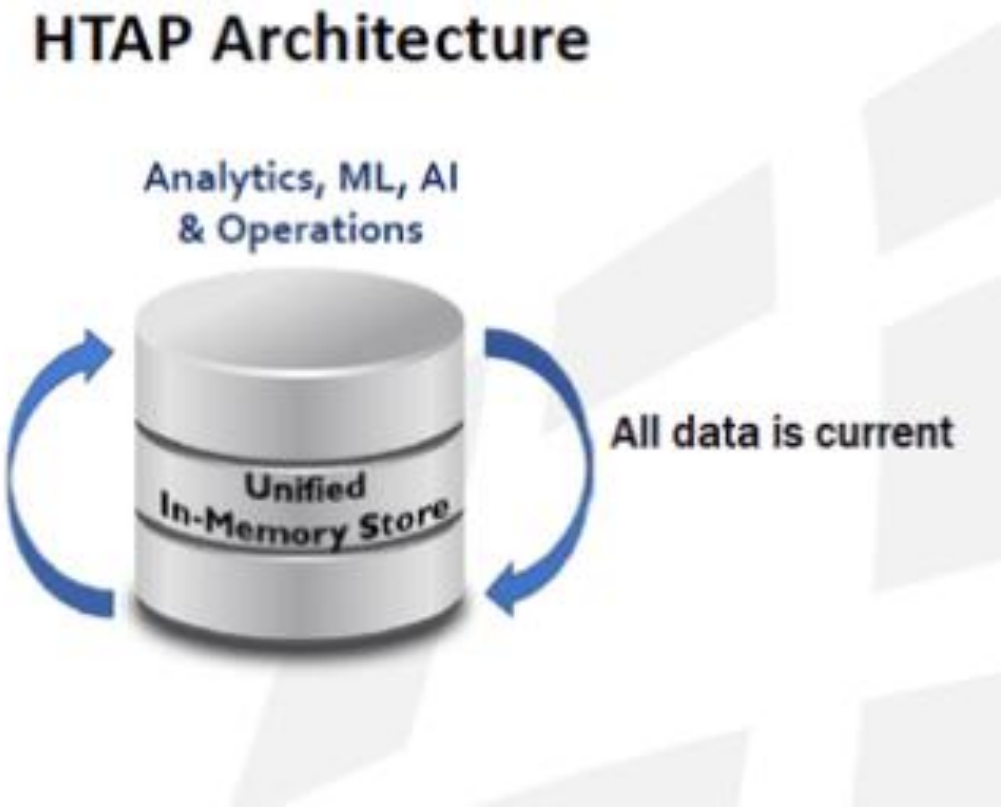
	SQL	NoSQL	NewSQL
Data model	Relational	Specialized: KV, document, graph	Relational
Query language	SQL	Key-value API New query language	SQL Key-value API
Scalability	Only for high-end (Teradata, Exadata)	By design (SN cluster)	By design (SN cluster)
Consistency	Strong	Limited	Strong
Big data ecosystem	External tables (HDFS)	Integration within Hadoop	Integration within Hadoop
Workload	OLAP XOR OLTP	OLTP	OLAP, OLTP, HTAP
Polystore	SQL or HDFS data sources	SQL or HDFS data sources	SQL, NoSQL, HDFS data sources

Activer Windows

©Maude Manouvrier - Univ. Paris-Dauphine - Repris de https://bigdataieee.org/BigData2019/files/Tutorial4.pdf

HTAP
(*Hybrid Transactional/Analytics Processing*)





©Maude Manouvrier - Univ. Paris Dauphine- Repris de https://sapbazar.com/articles/item/1457-in-what-way-in-memory-computing-powers-artificial-intelligence

Exemples de moteurs NewSQL

Vendor	Product	Objective	Comment
Google	Spanner	OLTP	Google cloud distributed database service. Used by F1 for the AdWords app.
LeanXcale	LeanXcale	HTAP	HTAP DBMS with fast insertion, fast aggregation over real-time data and polystore capability
SAP	Hana	HTAP	The HTAP pioneer, based on in-memory, column store
MemSQL Inc.	MemSQL	HTAP	In-memory, column and row store, MySQL compatible
Esgyn	EsgynDB	HTAP	Apache Trafodion for OLTP, Hadoop for OLAP
NuoDB	NuoDB	OLTP	Distributed SQL DBMS with P2P architecture
Splice Machine	Splice Machine	HTAP	HBase as storage engine, Derby as OLTP query engine and SparkQL as OLAP query engine

© P. Valduriez and R. Jimenez-Peris, 2019

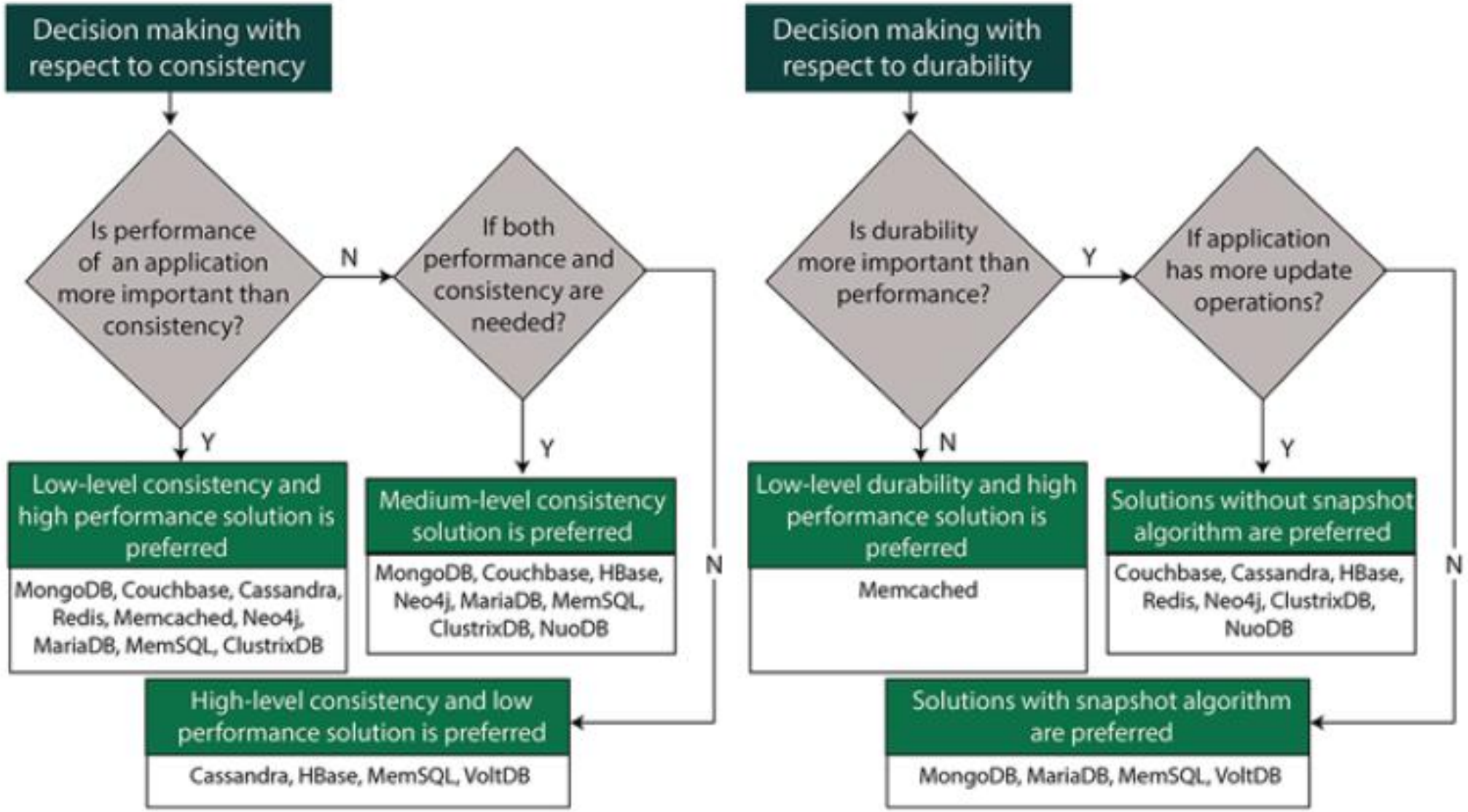
©Maude Manouvrier - Univ. Paris Dauphine – Repris de http://bigdataieee.org/BigData2019/files/Tutorial4.pdf

Quelles données pour quels moteurs NewSQL ?

Database name	Historical	Real-Time	Streaming	Temporal
Google Spanner	no	yes	no	yes
SAP HANA	yes	yes	yes	no
VoltDB	yes	yes	yes	no
TiDB	yes	yes	no	no
Altibase	yes	yes	no	no
Citus	yes	yes	no	yes
CockroachDB	no	yes	no	no
FaunaDB	no	yes	yes	yes
NuoDB	yes	yes	no	no

©Maude Manouvrier - Univ. Paris Dauphine – Repris de https://www.researchgate.net/profile/Sarah-Hahn-5/publication/353364749_Operationalizing_Analytics_with_NewSQL/links/6165712be7993f536cc9d321/Operationalizing-Analytics-with-NewSQL.pdf

Choisir le bon moteur de bases de données



(Extrait d'une approche de 2020)

©Maude Manouvrier - Univ. Paris-Dauphine - Repris https://link.springer.com/article/10.1007/s10619-020-07310-1 et

Exemple d'application polyglotte

MongoDB
Document Store

✓ Product Catalog & Details

Elasticsearch
Search Engine

✓ Product Search
✓ Faceting & Filtering

Redis
Key-Value Store

✓ Social sharing
✓ Ratings/Reviews
✓ Messaging

PostgreSQL
Relational

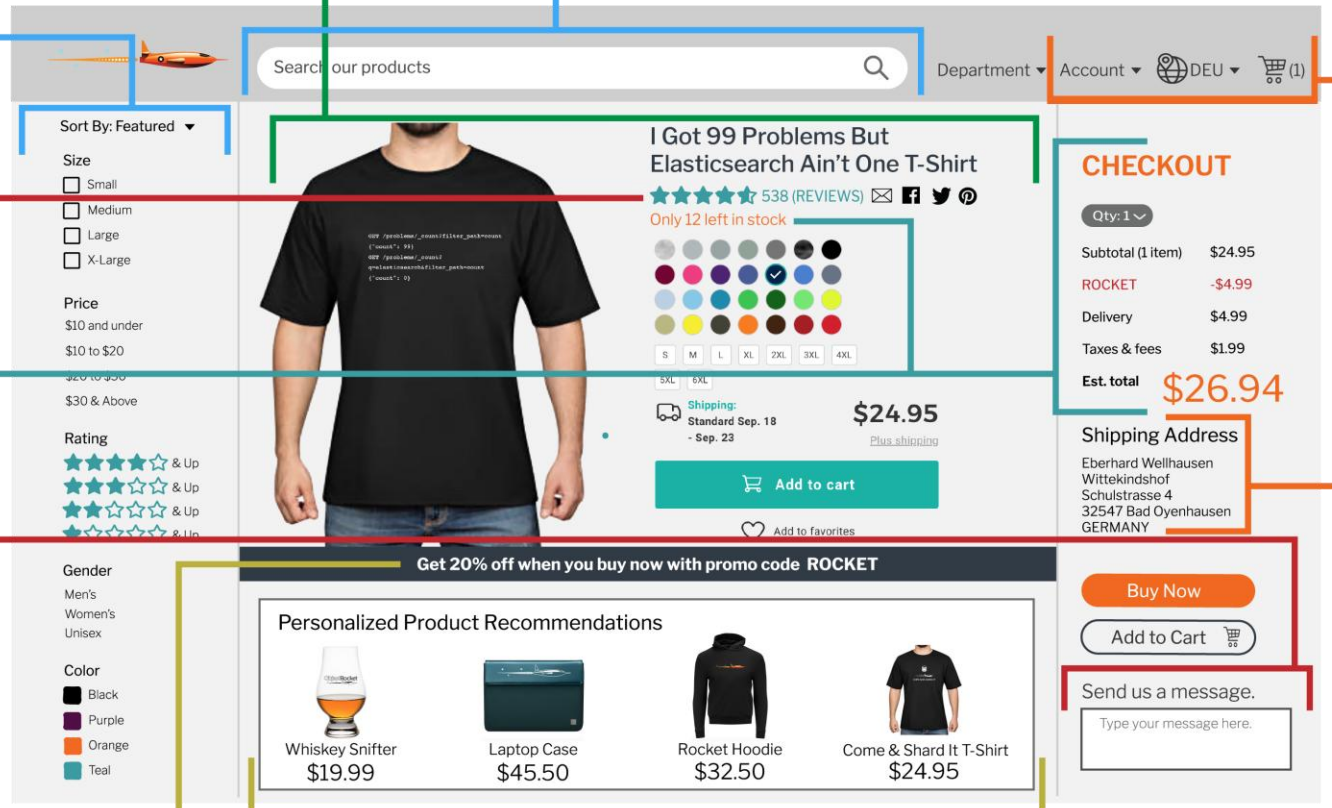
✓ Checkout
✓ Inventory
✓ Order Management

CockroachDB
Distributed SQL

✓ Geo-Partitioning for
Low Latency & Compliance

Hadoop
Wide-Column Store

✓ **Customized** Product Recommendations
✓ Activity-Based Deals/Offers



©Maude Manouvrier - Univ. Paris-Dauphine – Repris https://www.objectrocket.com/polyglot-persistence/