

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

SQL – NoSQL - NEWSQL

Maude Manouvrier

Espace Moodle : A5AMIF-100_22/23_SQL, NoSQL et NewSQL_Maude MANOUVRIER

Equipe Teams : SQL, NoSQL et NewSQL M2 MIAGE-IF (code 6u7htre)

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/>
<https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql>
- P. Rigaux (2018). *Bases de données documentaires et distribuées* :
<http://b3d.bdpedia.fr/>

BIBLIOGRAPHIE (2/2)

Ouvrages et liens utilisés pour le cours

- <https://nosql-database.org/> et <https://dbdb.io/>
- **A. Foucret**, « Livre blanc sur NoSQL ». Smile
<https://www.smile.eu/fr/livres-blancs/livres-blancs/nosql>
- **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>)
- **Formation NoSQL CNRS** - supports de cours :
<http://rbdd.cnrs.fr/spip.php?article315>
- + d'autres sites référencés en bas des transparents

Introduction

- **SQL**
- **NoSQL**
- **NewSQL ou SQL distribué**
- **Systemes multi-modèles**
- **Persistence polyglotte**

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



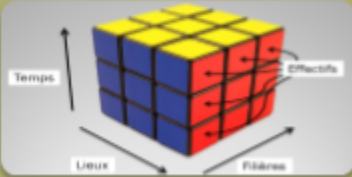
Bases de données avancées (années 80)

- *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

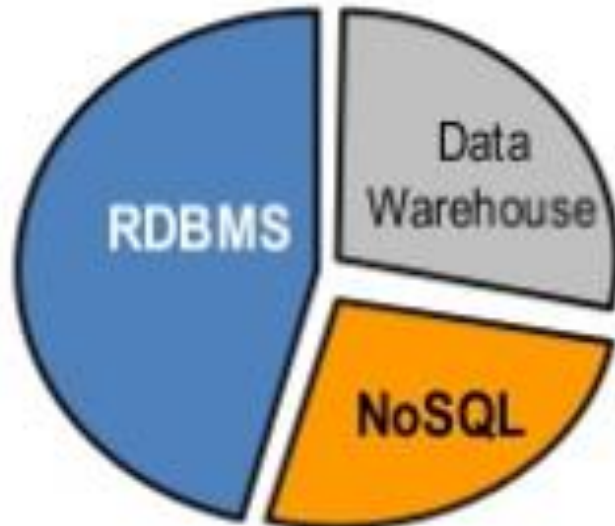
Répartition des modèles dans le temps



1985-1995

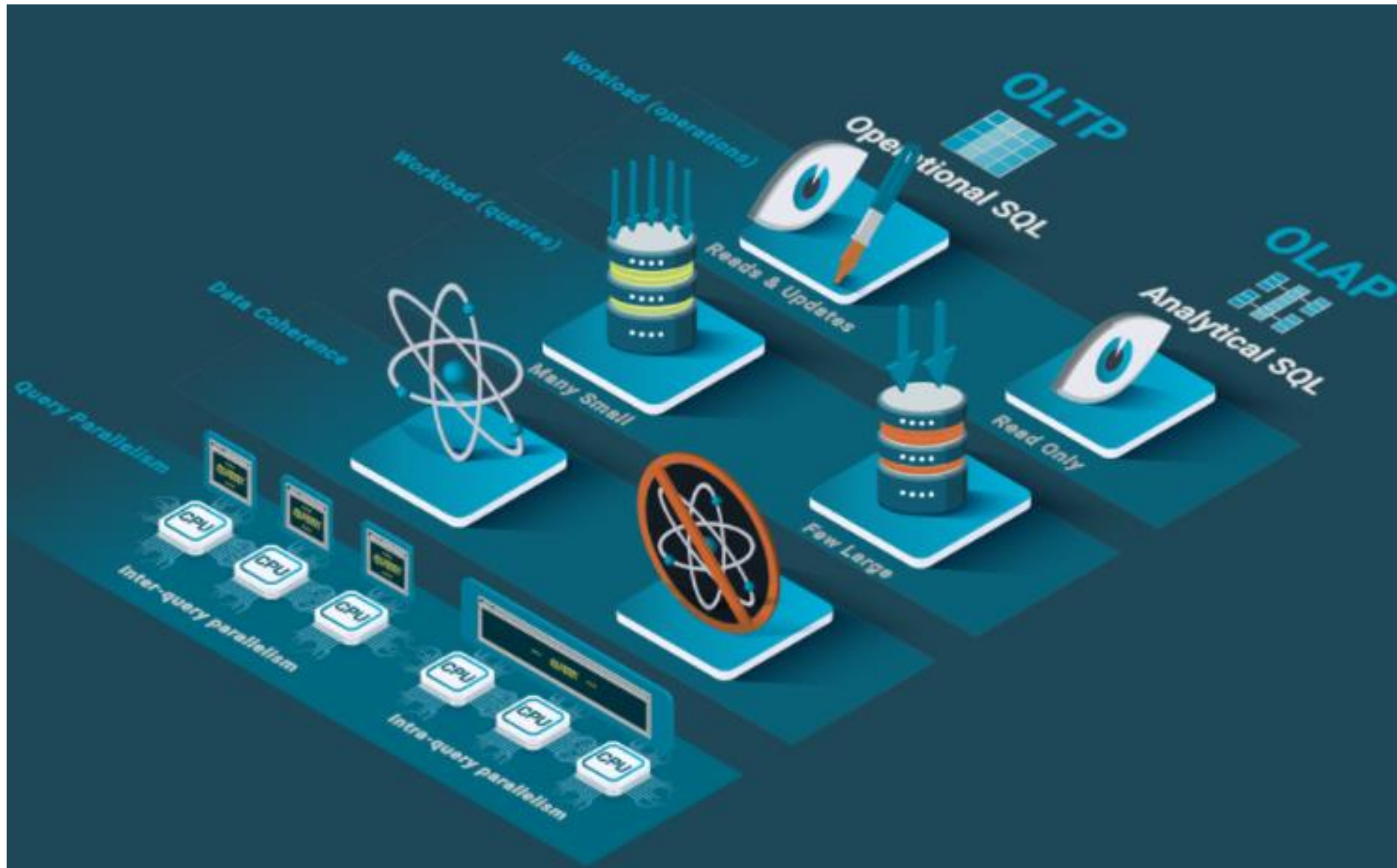


1995-2010



2010-Now

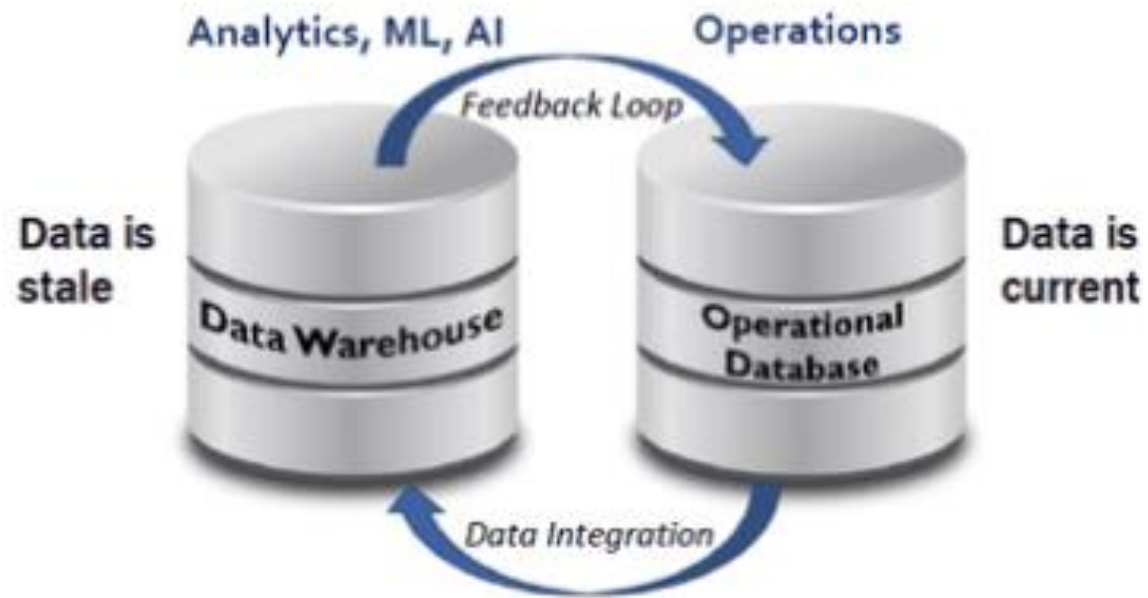
OLAP (On Line Analytical Processing) vs OLTP (On Line Transactional Processing)



HTAP

(Hybrid Transactional/Analytics Processing)

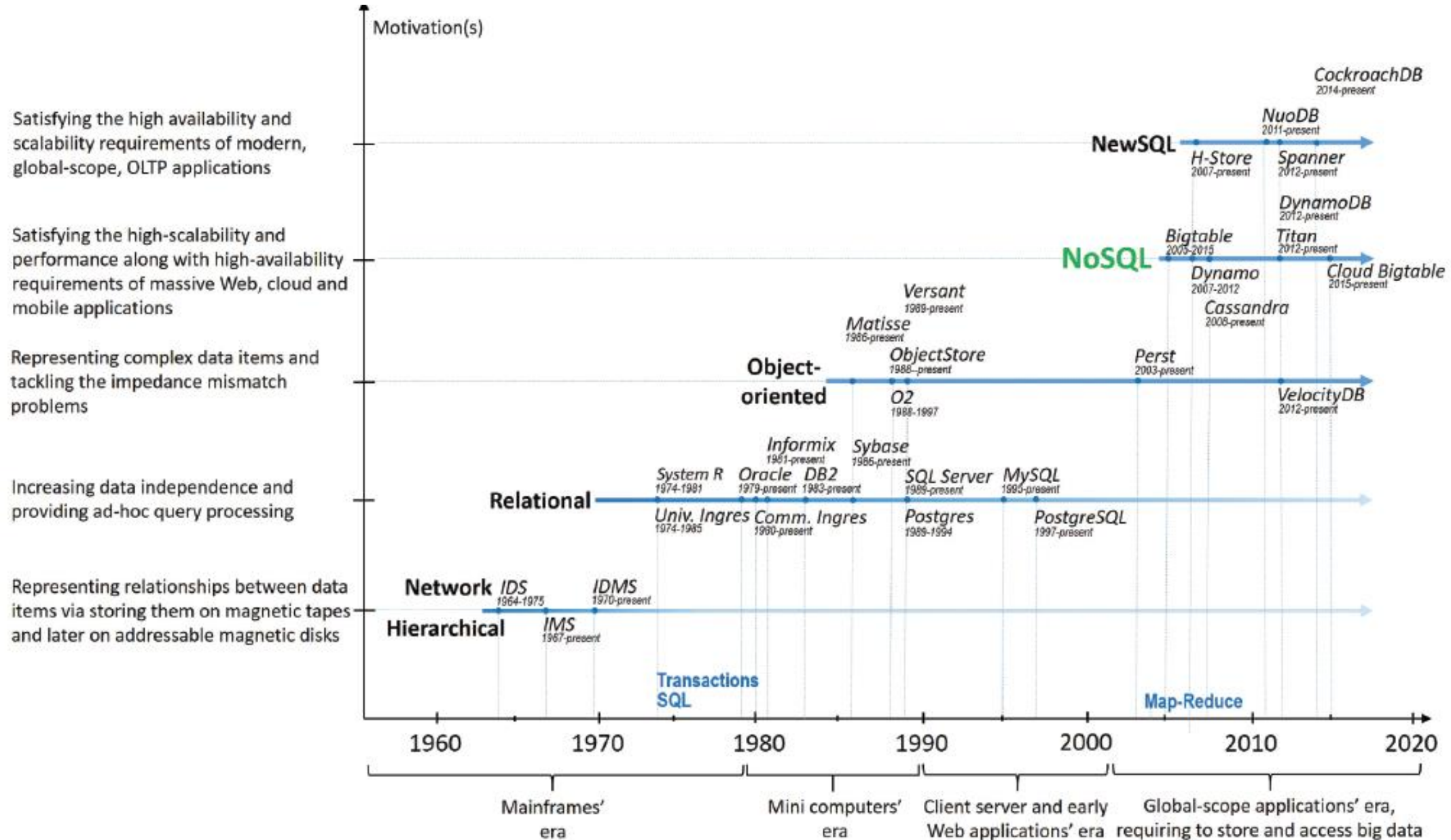
Traditional Architecture



HTAP Architecture



Date de naissance des différents moteurs



Classement des moteurs de bases de données

410 systems in ranking, February 2023

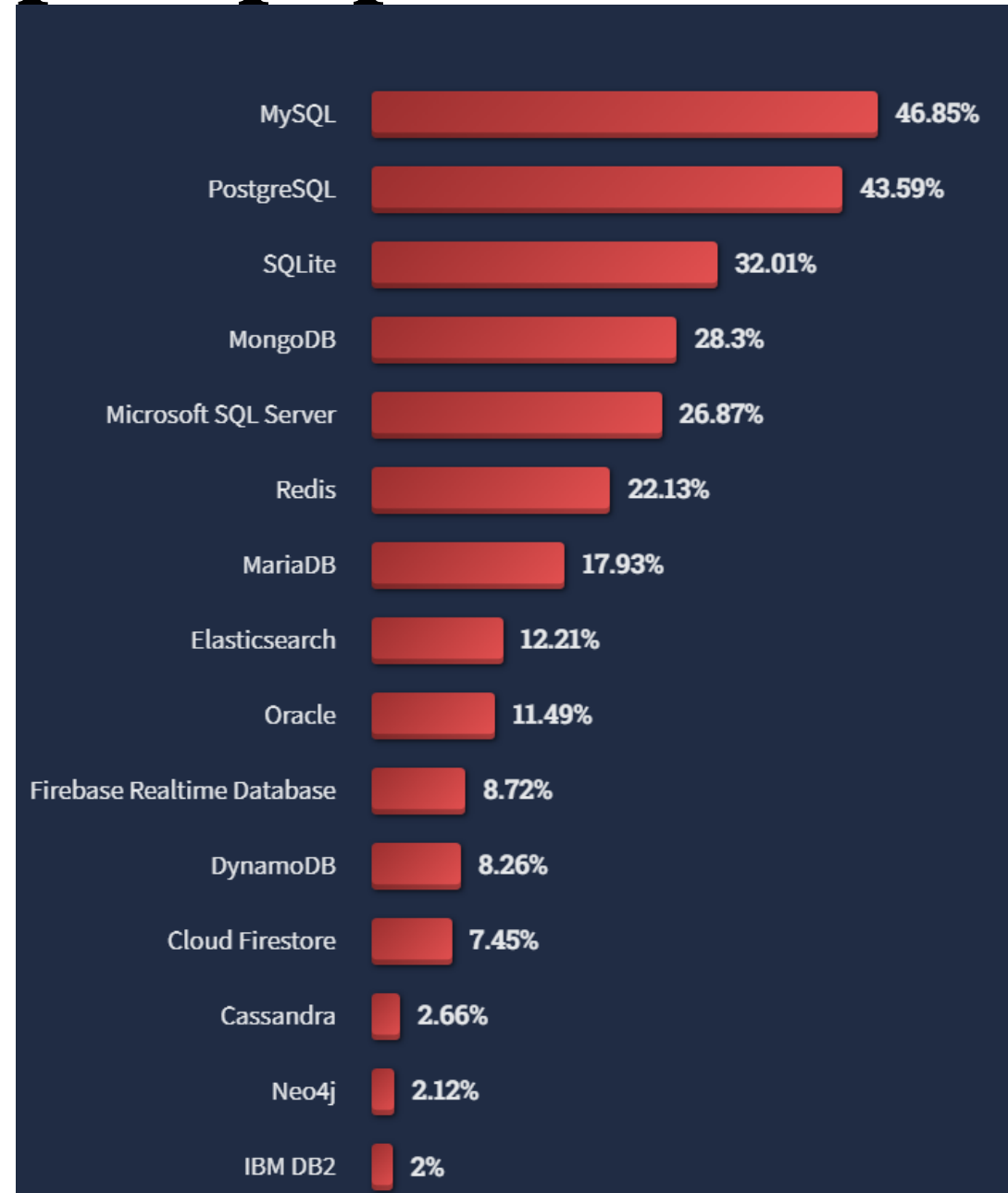
Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle	Relational, Multi-model	1247.52	+2.35	-9.31
2.	2.	2.	MySQL	Relational, Multi-model	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL	Relational, Multi-model	616.50	+1.65	+7.12
5.	5.	5.	MongoDB	Document, Multi-model	452.77	-2.42	-35.88
6.	6.	6.	Redis	Key-value, Multi-model	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model	138.60	-2.56	-23.70
9.	10.	10.	SQLite	Relational	132.67	+1.17	+4.30
10.	9.	9.	Microsoft Access	Relational	131.03	-2.33	-0.23
11.	12.	11.	Cassandra	Wide column	116.22	-0.09	-7.76

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

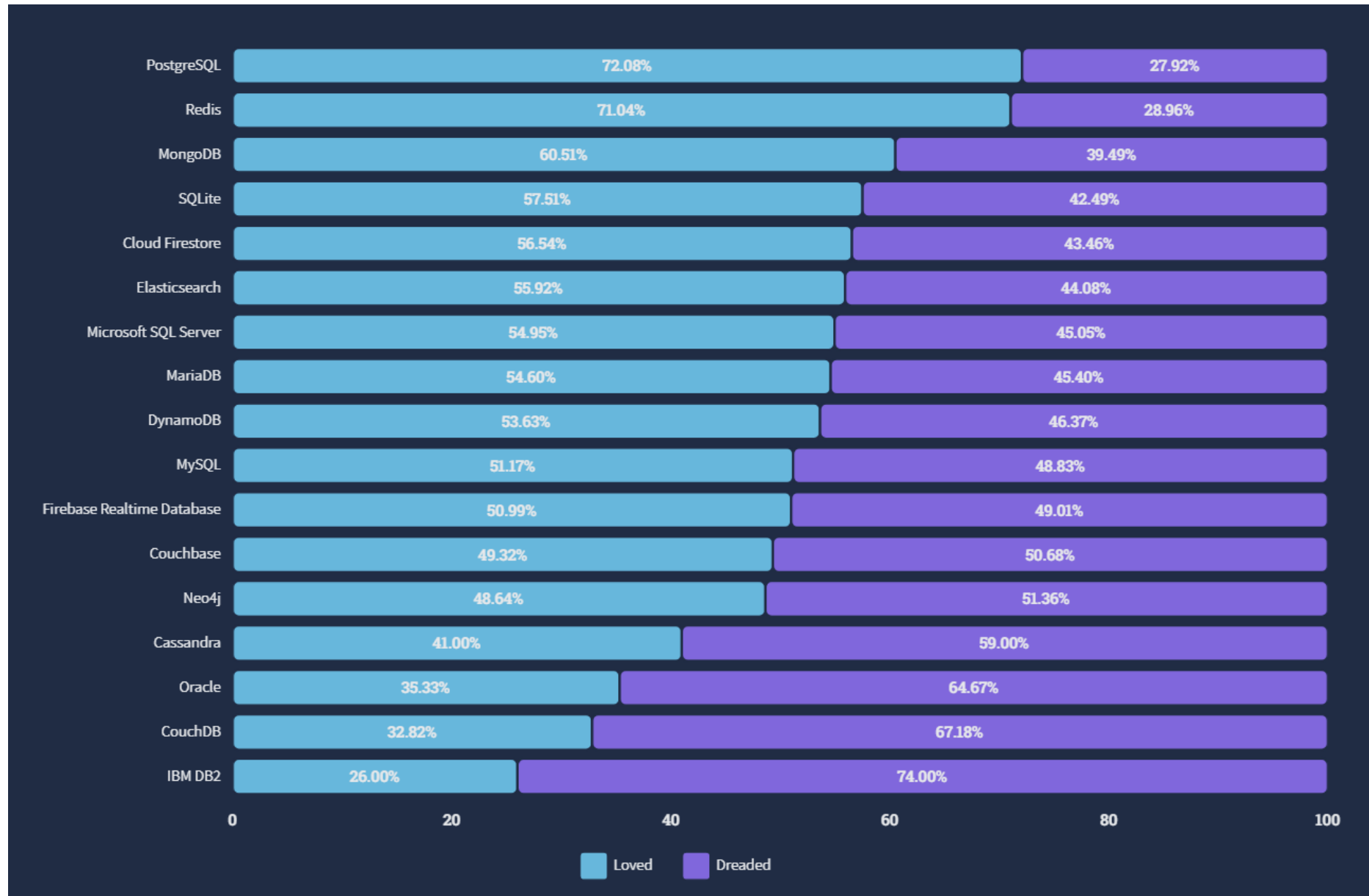
Bases de données les plus populaires en 2022

Etude *stackoverflow* basée sur les réponses de 70 000 réponses de développeurs issues de 180 pays

<https://survey.stackoverflow.co/2022/#technology>



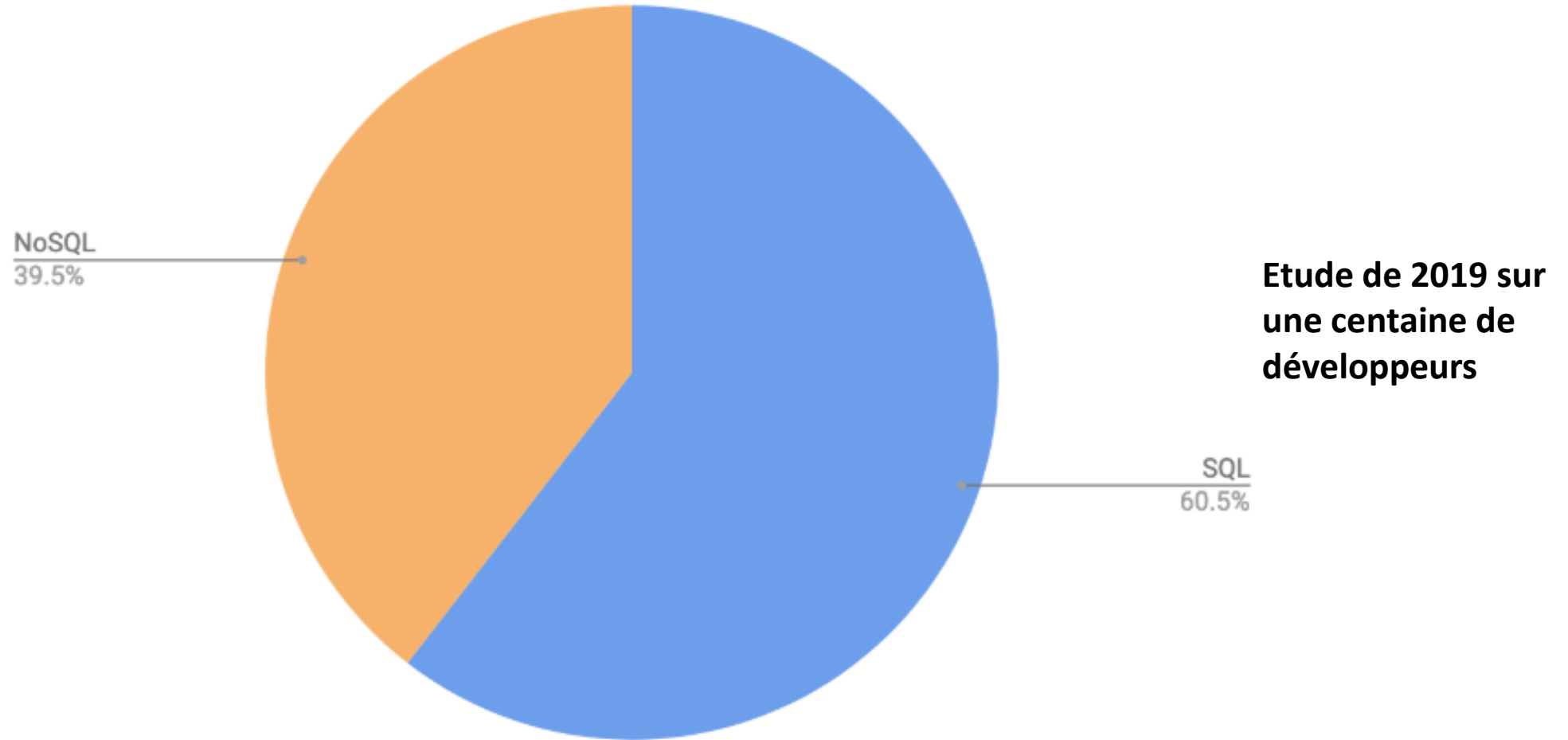
Appréciation/Détestation des bases de données en 2022



Tendances d'utilisation du SQL et du NoSQL

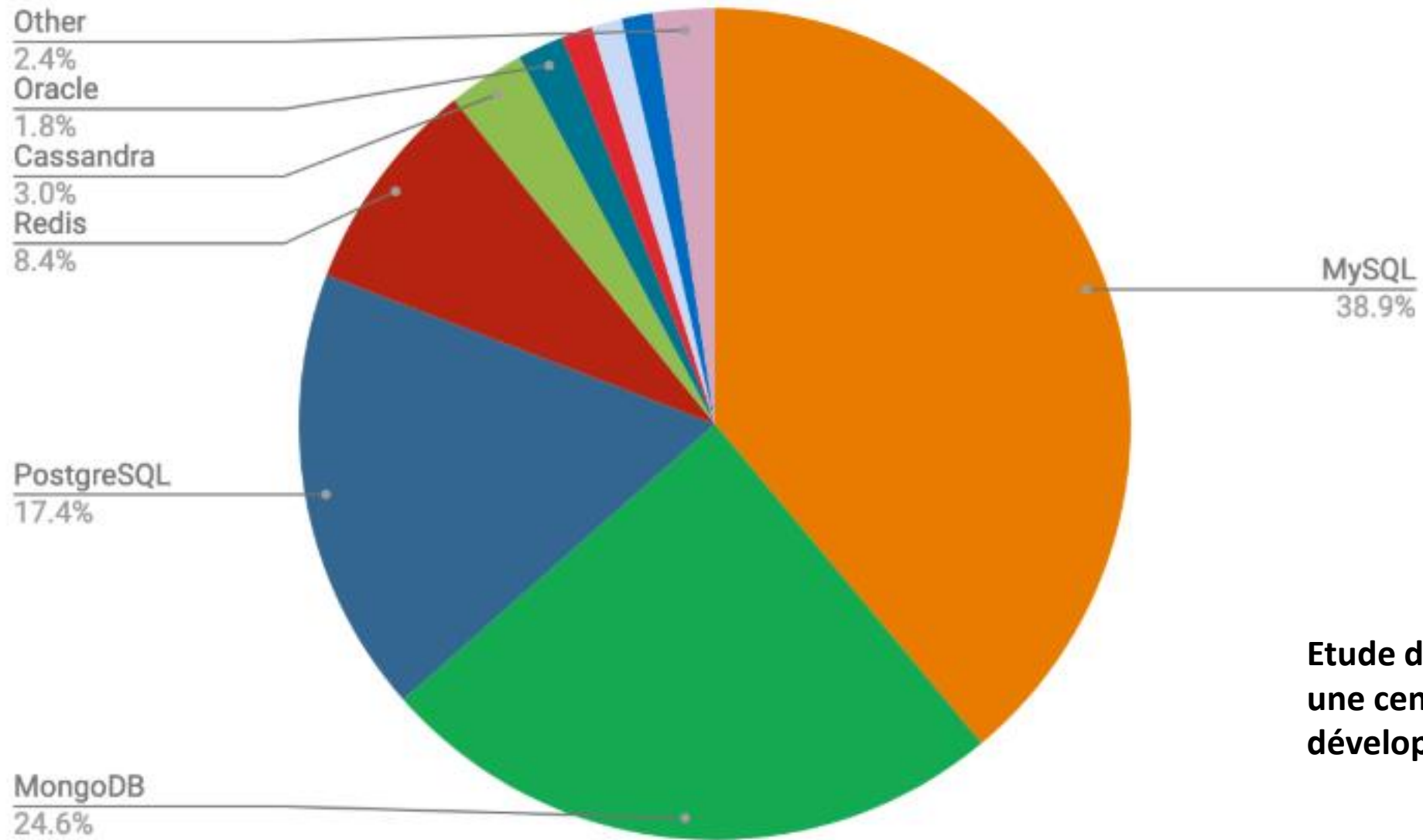
SQL Database Use: 60.48%

NoSQL Database Use: 39.52%



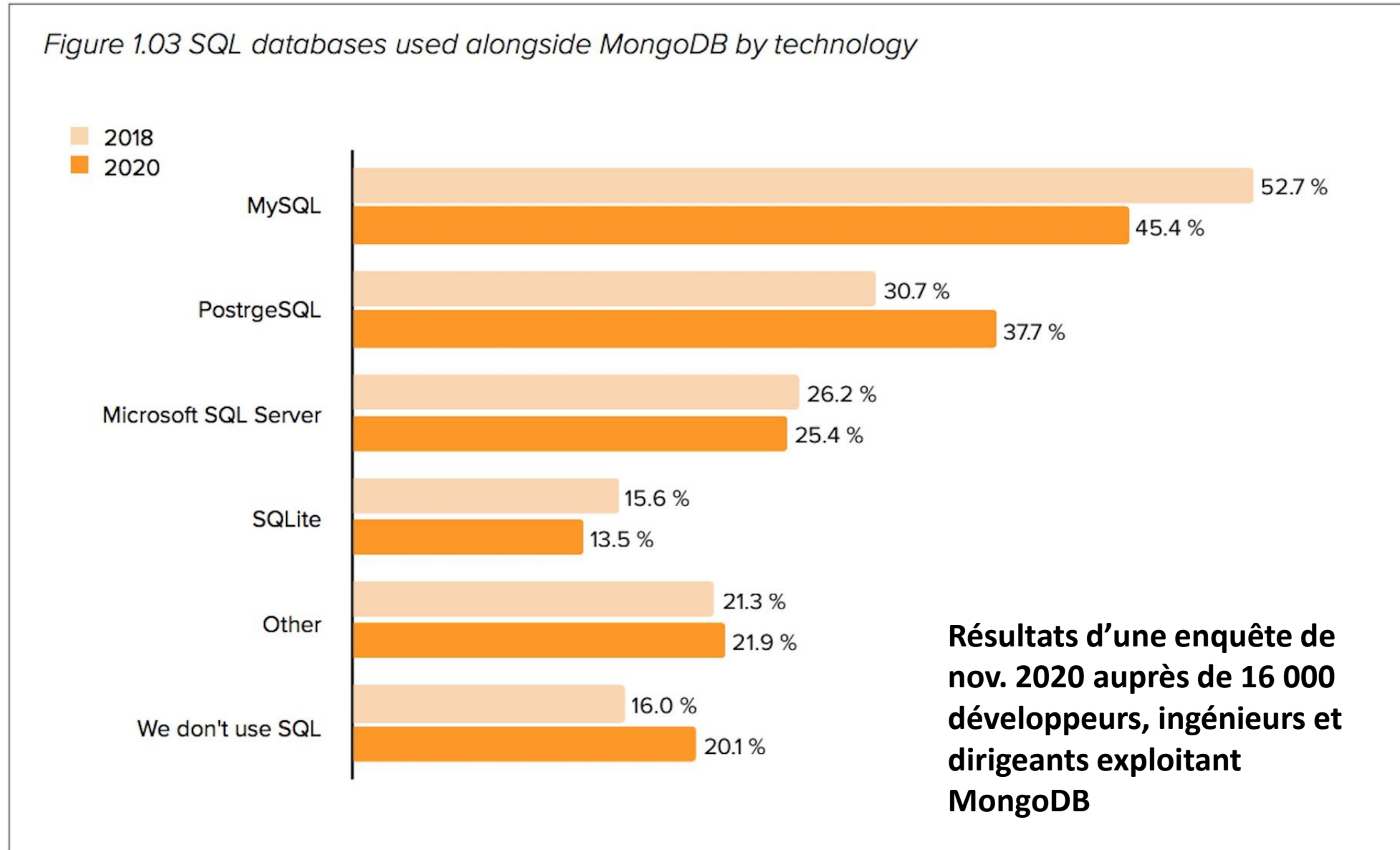
Etude de 2019 sur
une centaine de
développeurs

Tendances d'utilisation des différents moteurs



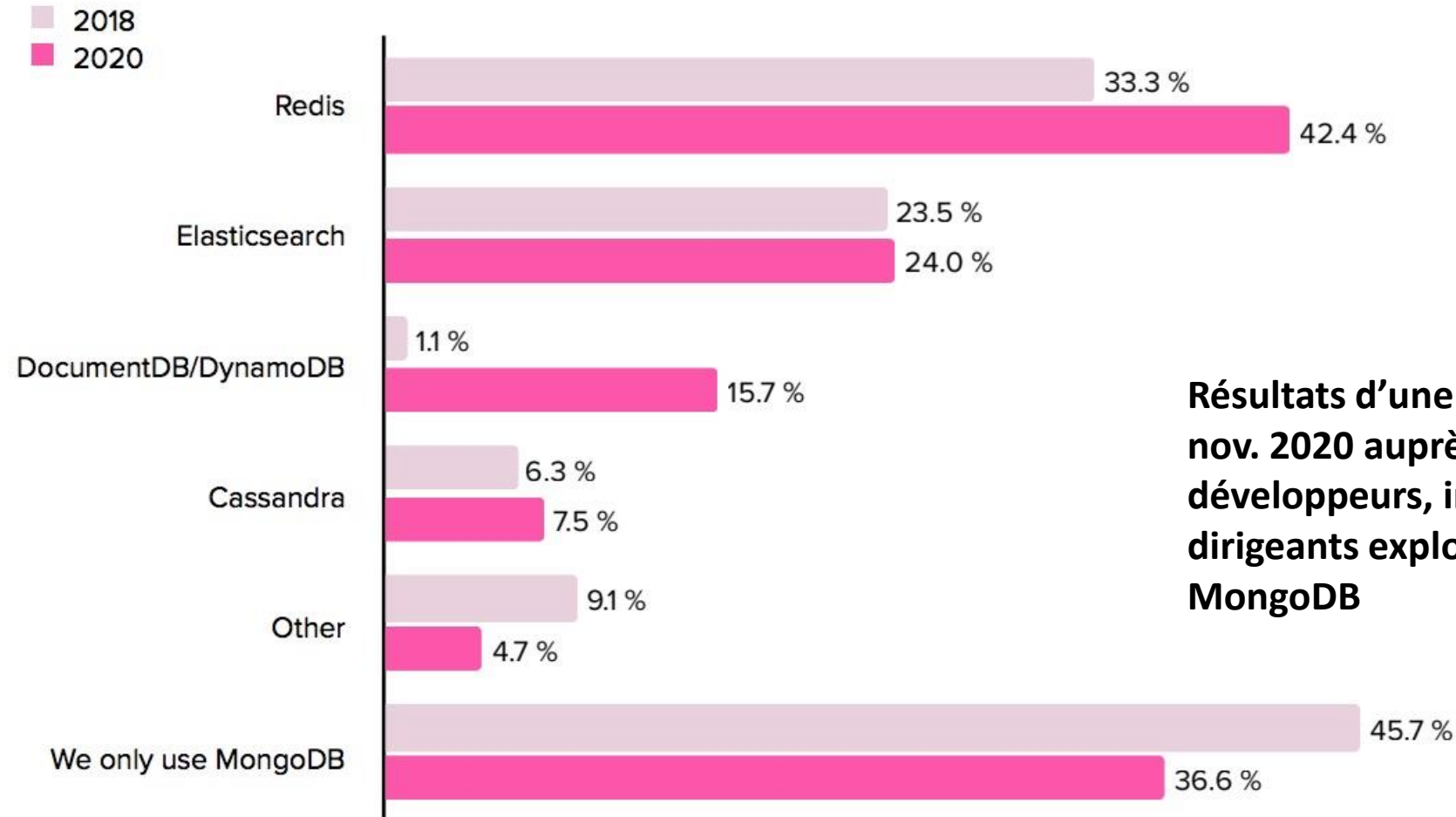
Etude de 2019 sur
une centaine de
développeurs

Tendances d'utilisation conjointe SQL et NoSQL



Tendances d'utilisation de plusieurs moteurs NoSQL

Figure 1.04 NoSQL databases used alongside MongoDB by technology

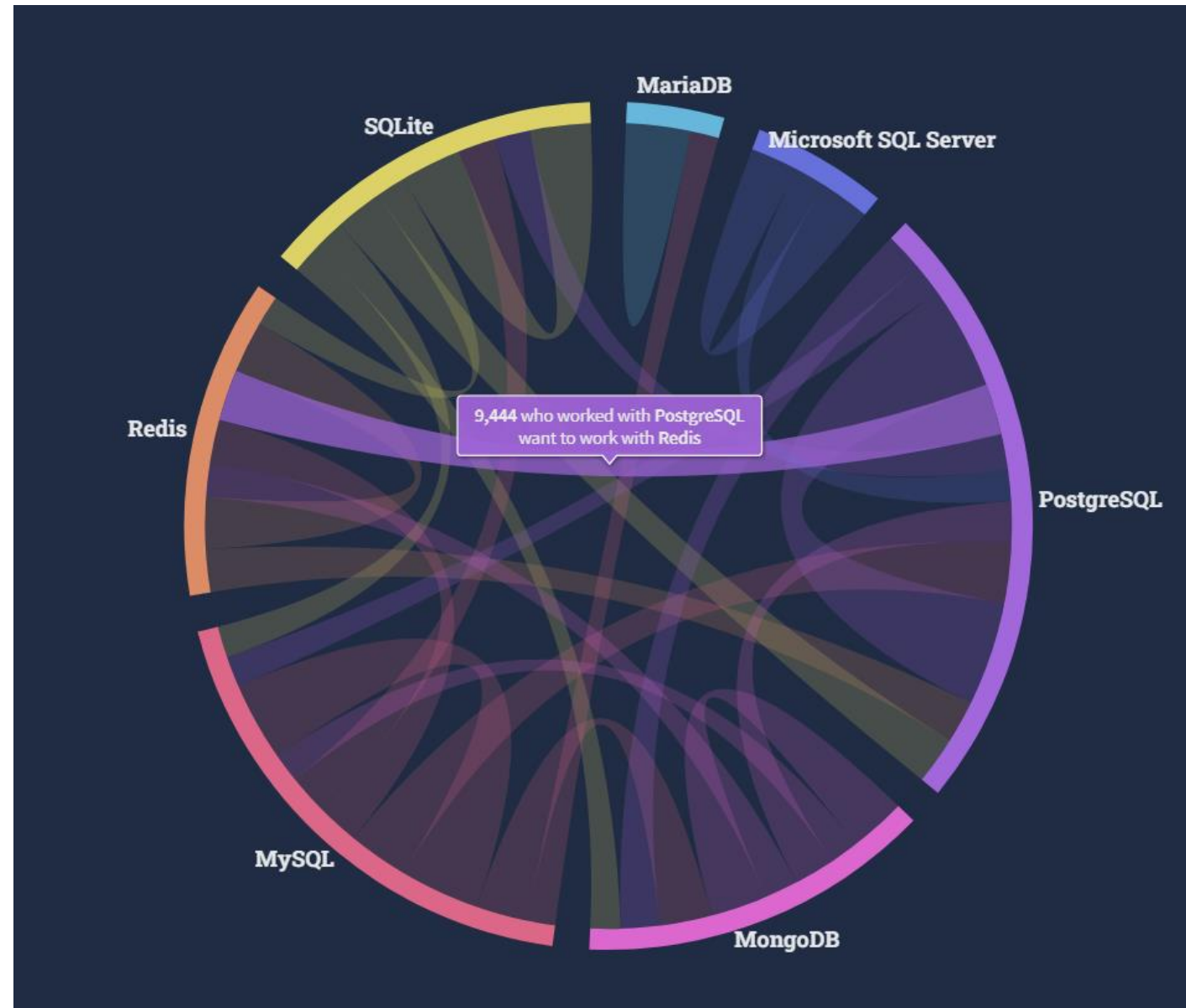


Résultats d'une enquête de nov. 2020 auprès de 16 000 développeurs, ingénieurs et dirigeants exploitant MongoDB

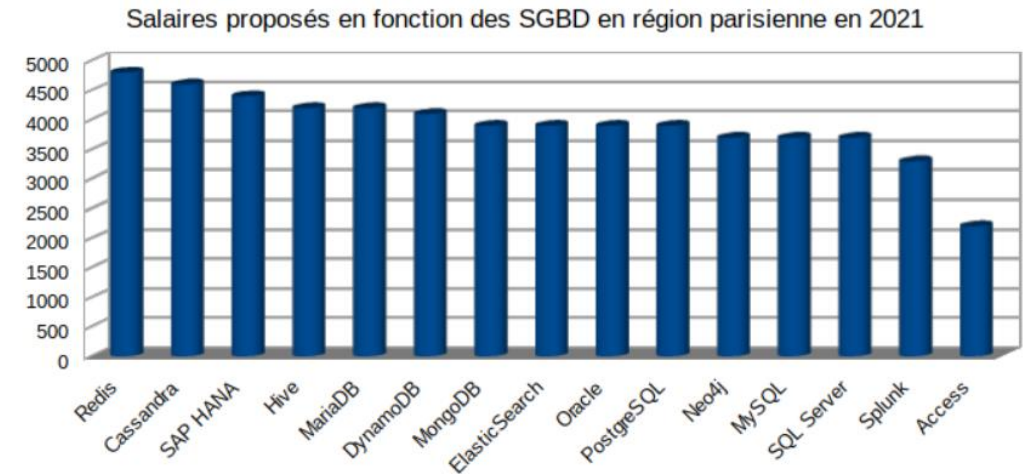
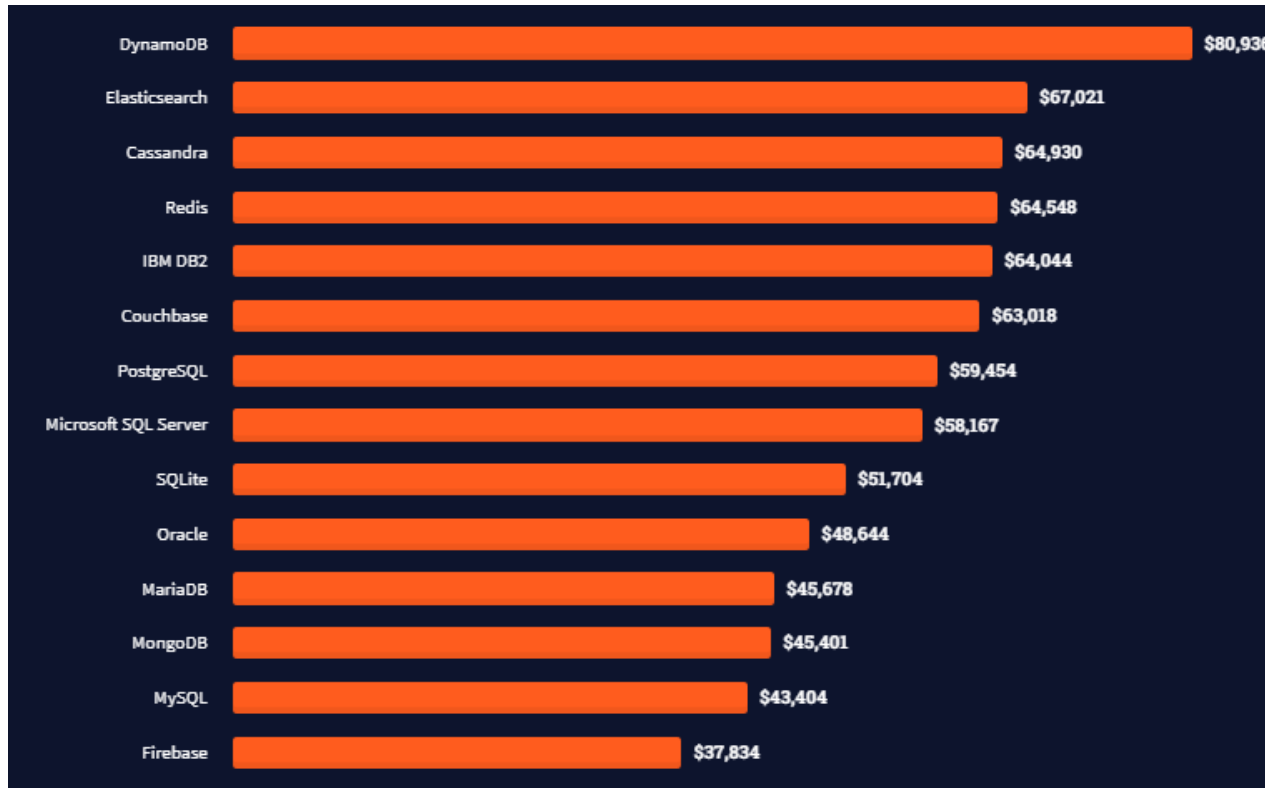
Tendances d'utilisation de plusieurs moteurs NoSQL

Etude *stackoverflow* basée sur les réponses de 70 000 réponses de développeurs issues de 180 pays

<https://survey.stackoverflow.co/2022/#technology>



Profils BD les mieux payés en 2021



Position du SQL parmi les technologies les plus populaires en 2022



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 2020+** : SQL/PGQ (*property graph queries* – en cours) ...

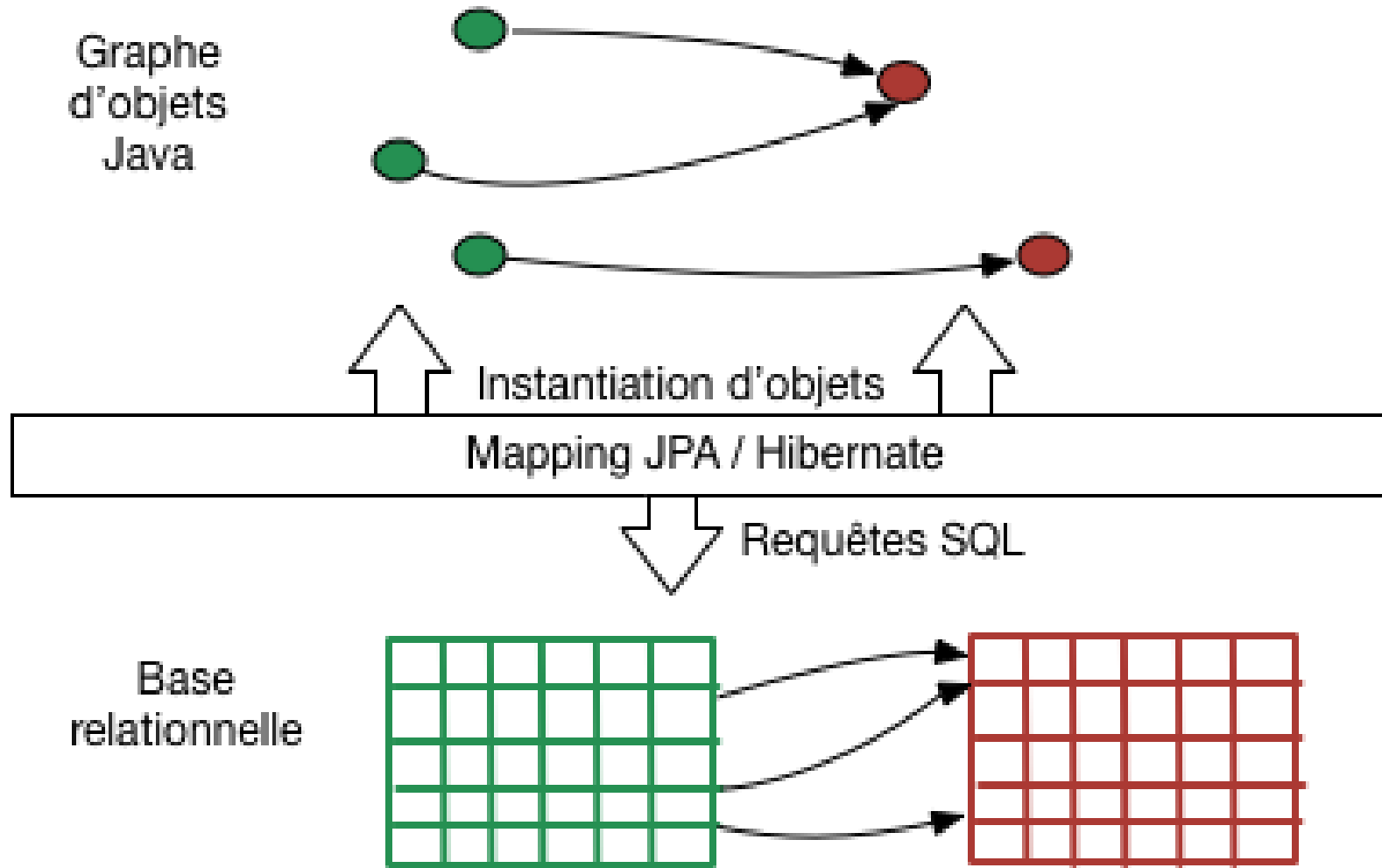
SQL : les avantages du relationnel

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 ...

+ ...

Mapping Objet/relationnel



The Law of the Relational Database



By HkingArtist.com

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
-  **Systemes 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**

...

HOW TO WRITE A CV



Leverage the NoSQL boom

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://blog.nexaby.com/2011/04/02/nosql-5-minutes-pour-comprendre/>







NoSQL : pas de table



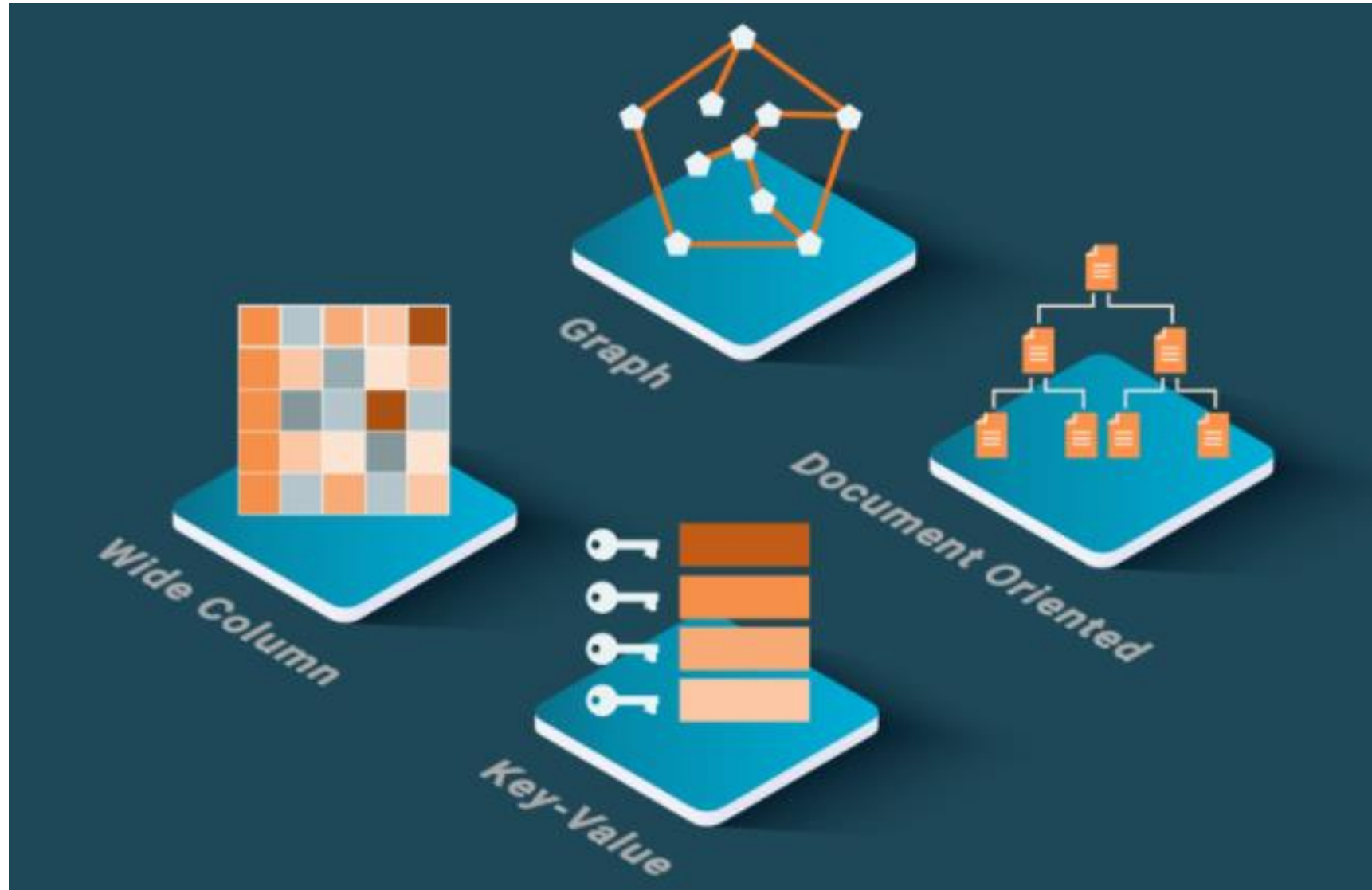
NoSQL : agrégat

- 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 répartie (*distributed*)**
 - **Requêtes massives**
 - **Schéma évolutif**

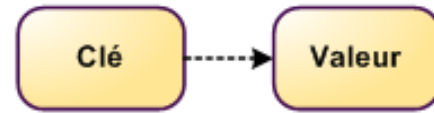
NoSQL : avantages et inconvénients

-  **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)**

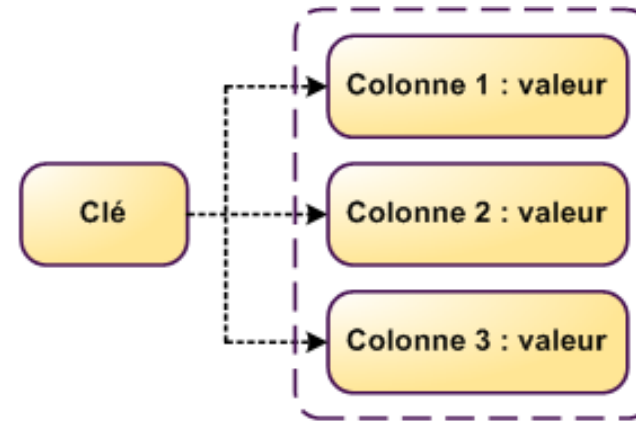
NoSQL : 4 modèles (nom en anglais)



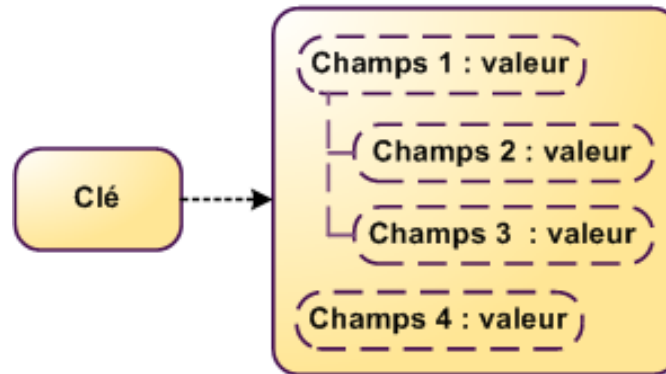
NoSQL : 4 modèles (nom en français)



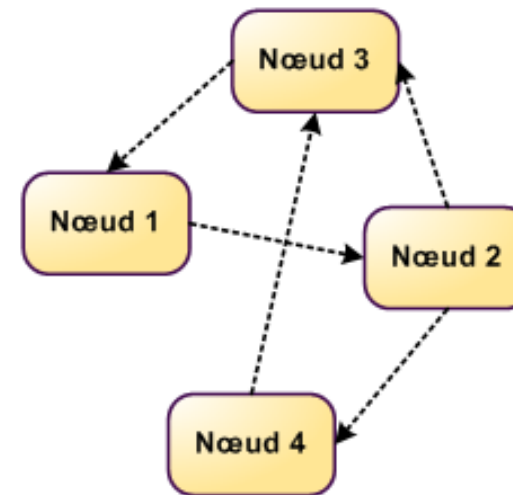
BDD Clé-Valeur



BDD Orientée colonnes

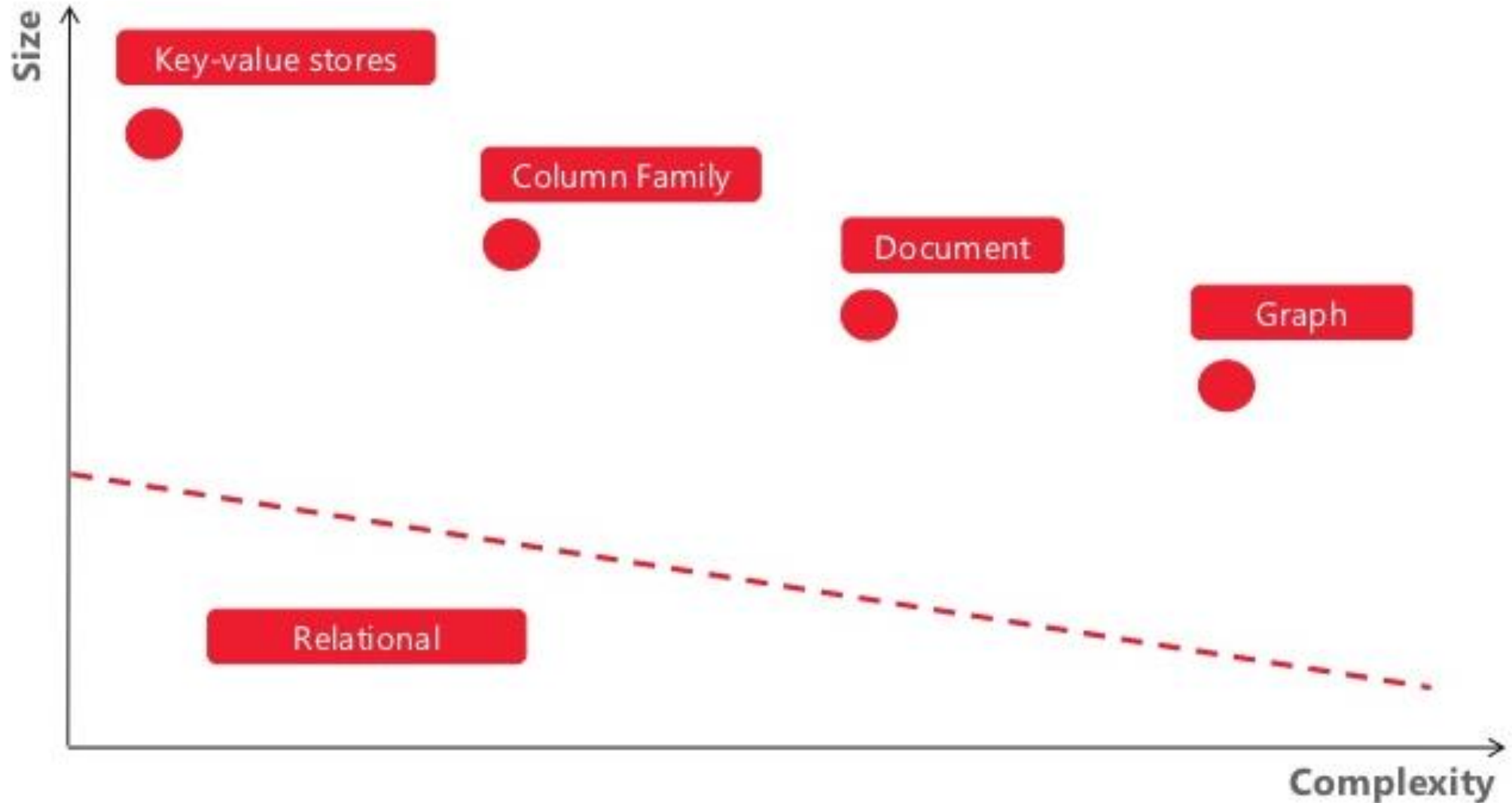


BDD Orientée document

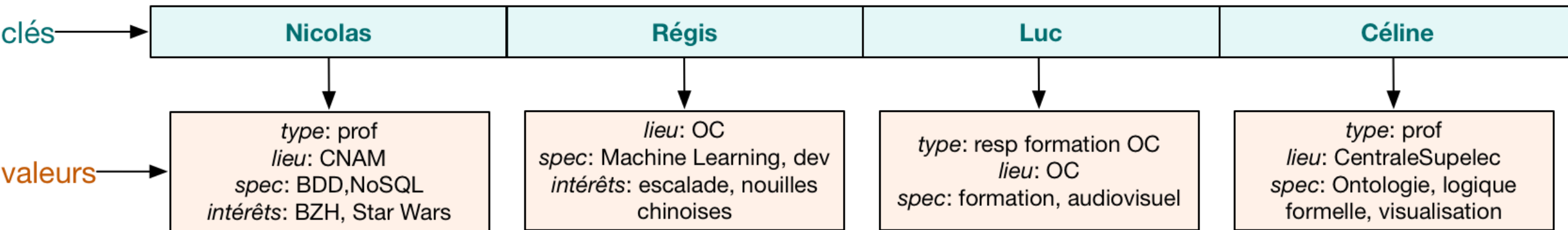


BDD Orientée graphe

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

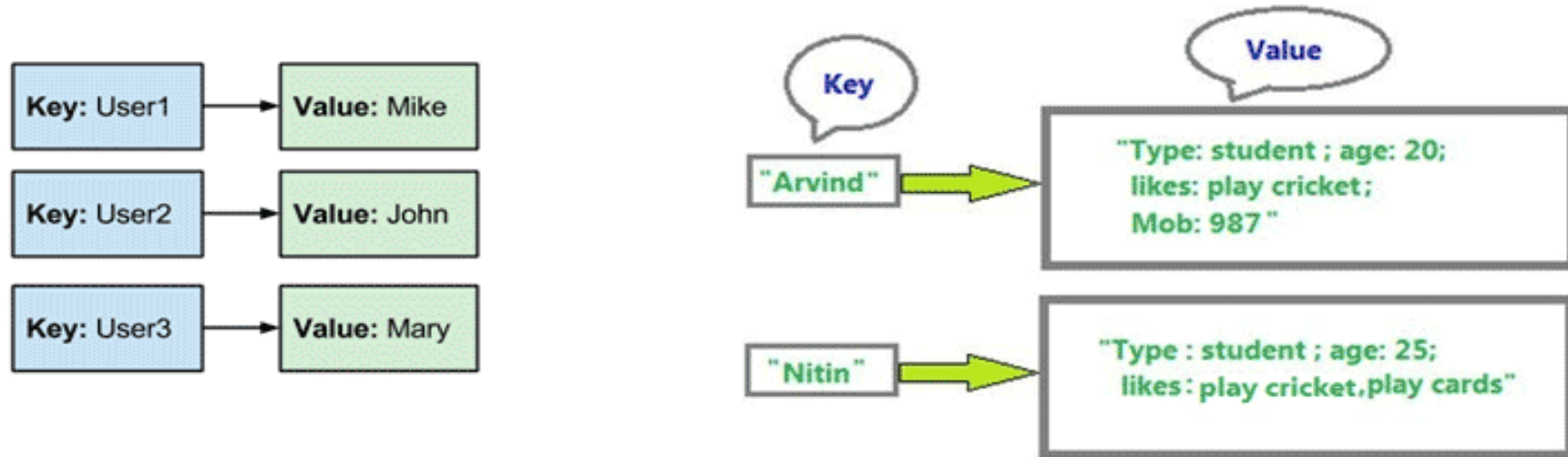
Modèle Clé - Valeur : principes

- 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)



Classement des moteurs Clé - Valeur

include secondary database models

68 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Redis	Key-value, Multi-model	173.83	-3.72	-1.96
2.	2.	2.	Amazon DynamoDB	Multi-model	79.69	-1.87	-0.67
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	36.51	-1.45	-3.45
4.	4.	4.	Memcached	Key-value	23.17	-0.58	-2.60
5.	5.	6.	Hazelcast	Key-value, Multi-model	9.11	-0.23	-0.39
6.	6.	5.	etcd	Key-value	8.51	+0.01	-3.19
7.	7.	10.	Aerospike	Multi-model	6.56	+0.09	+0.95
8.	8.	9.	Ehcache	Key-value	6.04	+0.04	-0.34
9.	10.	14.	Google Cloud Bigtable	Multi-model	5.92	+0.45	+1.75
10.	9.	8.	Ignite	Multi-model	5.50	-0.40	-1.00
11.	12.	11.	ArangoDB	Multi-model	5.29	+0.22	-0.11
12.	11.	7.	Riak KV	Key-value	5.14	-0.10	-1.78
13.	14.	12.	OrientDB	Multi-model	4.54	+0.06	-0.49

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.

Table_Client

1	Prénom : Marc	Age: 35	
2	Prénom : Julie	Age : 30	Nom : Dupond

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

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	Driver Information	Car Information
123546	Name:John Insurance: Geico	Car: Speed3 Year:2013 Warranty:Yes
123547	Name:Jen Insurance:State Farm	Car:626 Year:2008
123548	Name:Tony	

Key	Car Information
123546	Car: Speed3 Year:2013 Warranty:Yes ServiceID:10 Type:Oil ServiceID:11 Type:Tires ServiceID:12 Type:Wipers

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



Classement des moteurs orientés colonne

include secondary database models

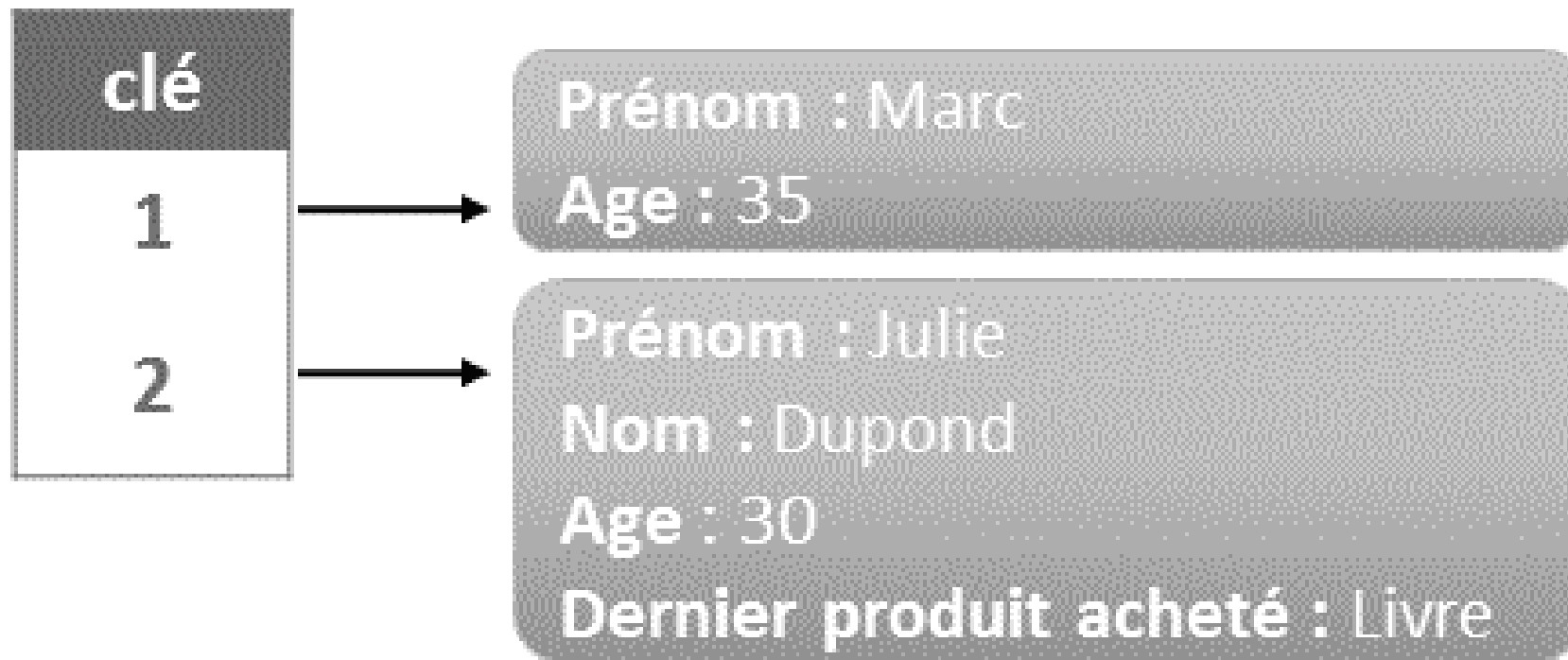
13 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Cassandra	Wide column	116.22	-0.09	-7.76
2.	2.	2.	HBase	Wide column	38.41	-0.93	-5.21
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	36.51	-1.45	-3.45
4.	4.	4.	Datastax Enterprise	Wide column, Multi-model	8.20	+0.51	-2.54
5.	5.	5.	Microsoft Azure Table Storage	Wide column	5.93	+0.11	+0.28
6.	7.	6.	Google Cloud Bigtable	Multi-model	5.92	+0.45	+1.75
7.	6.	7.	Accumulo	Wide column	5.75	+0.05	+1.82
8.	8.	8.	ScyllaDB	Wide column, Multi-model	5.63	+0.30	+1.75
9.	9.	9.	HPE Ezmeral Data Fabric	Multi-model	1.29	+0.04	+0.45
10.	10.	10.	Amazon Keyspaces	Wide column	0.91	+0.06	+0.35
11.	11.	11.	Elassandra	Wide column, Multi-model	0.63	+0.11	+0.12
12.	12.	12.	Alibaba Cloud Table Store	Wide column	0.23	-0.05	-0.23
13.	13.	13.	SWC-DB	Wide column, Multi-model	0.02	-0.01	-0.03

NoSQL : Modèle document

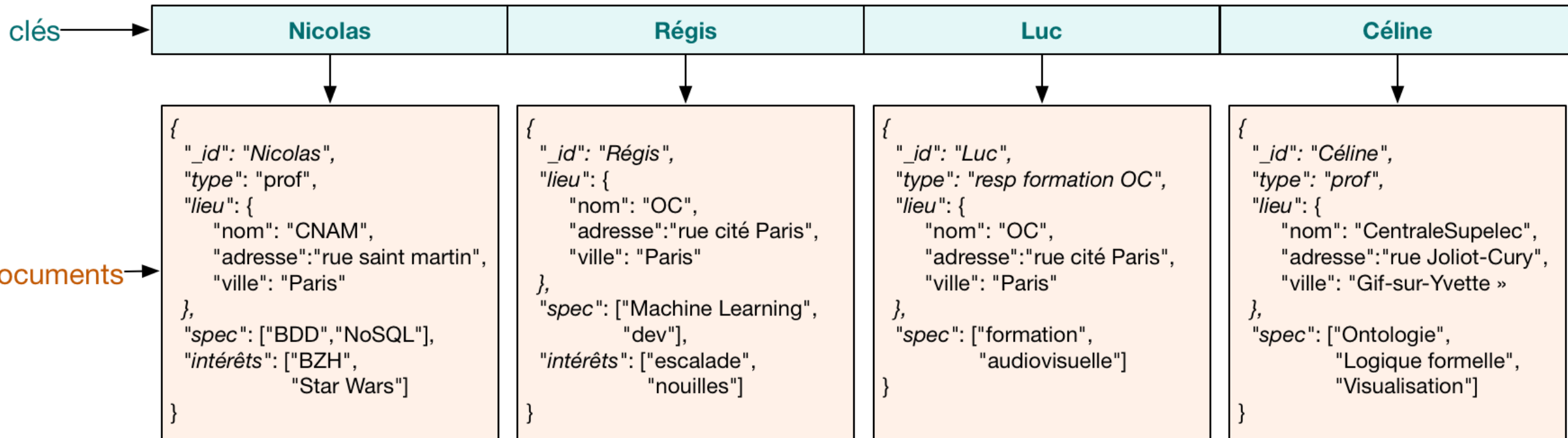
Orientées Documents

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.



illustra

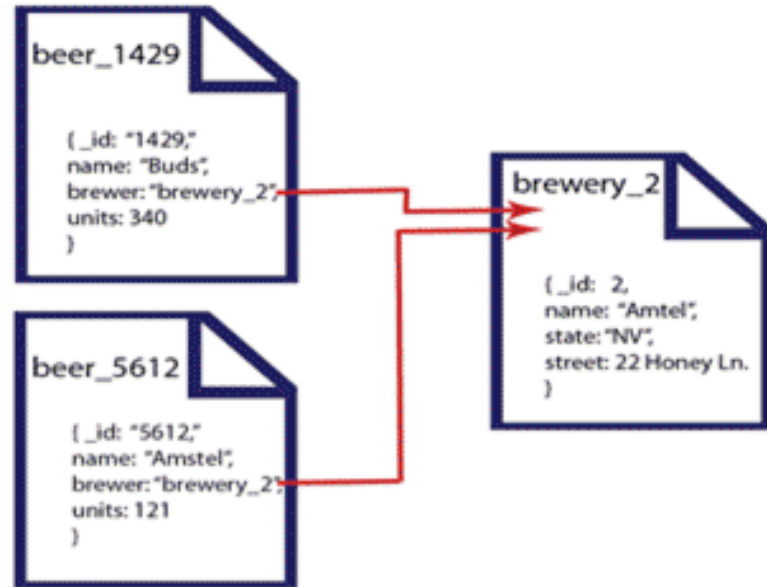
Exemples de données orientées documents



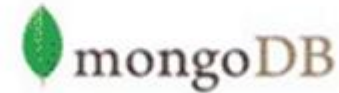
Modèle orienté document : principes

- 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 ...

Modèle orienté document : avantages, inconvénients et principaux moteurs



Exemples d'implémentation



Amazon
DynamoDB (Beta)



Requêtage plus complet

Flexibilité

Evolutif au cours du temps



Duplication des données

Cohérence pas forcément assurée

Classement des moteurs orientés document

include secondary database models

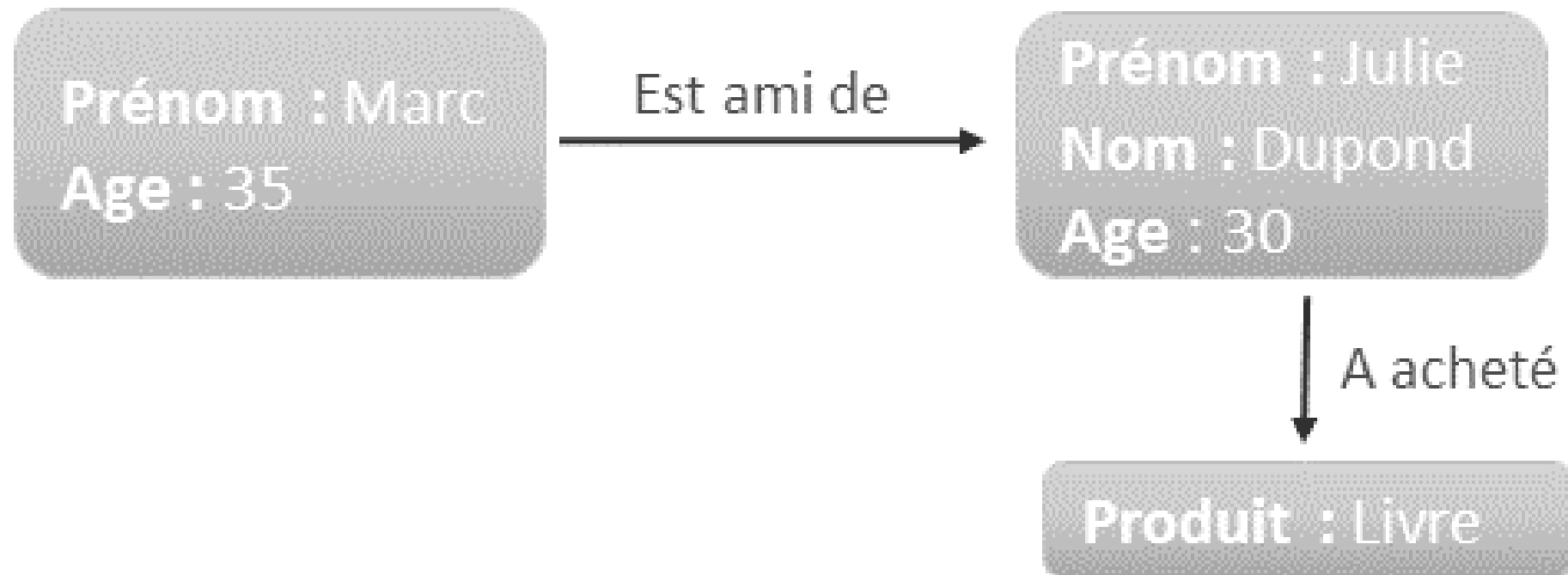
56 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	MongoDB	Document, Multi-model	452.77	-2.42	-35.88
2.	2.	2.	Amazon DynamoDB	Multi-model	79.69	-1.87	-0.67
3.	3.		Databricks	Multi-model	60.33	-0.49	
4.	4.	3.	Microsoft Azure Cosmos DB	Multi-model	36.51	-1.45	-3.45
5.	5.	4.	Couchbase	Document, Multi-model	24.86	-0.44	-5.21
6.	6.	5.	Firebase Realtime Database	Document	18.49	-0.27	-0.66
7.	7.	6.	CouchDB	Document, Multi-model	14.45	-0.44	-3.01
8.	8.	9.	Google Cloud Firestore	Document	11.51	+0.41	+2.45
9.	9.	7.	MarkLogic	Multi-model	8.84	-0.01	-0.61
10.	10.	8.	Realm	Document	8.24	-0.09	-1.17
11.	11.	13.	Google Cloud Datastore	Document	6.89	+0.22	+1.64
12.	12.	10.	Aerospike	Multi-model	6.56	+0.09	+0.95
13.	13.	12.	Virtuoso	Multi-model	6.11	+0.23	+0.72
14.	14.	11.	ArangoDB	Multi-model	5.29	+0.22	-0.11
15.	15.	14.	OrientDB	Multi-model	4.54	+0.06	-0.49

NoSQL : Modèle graphe

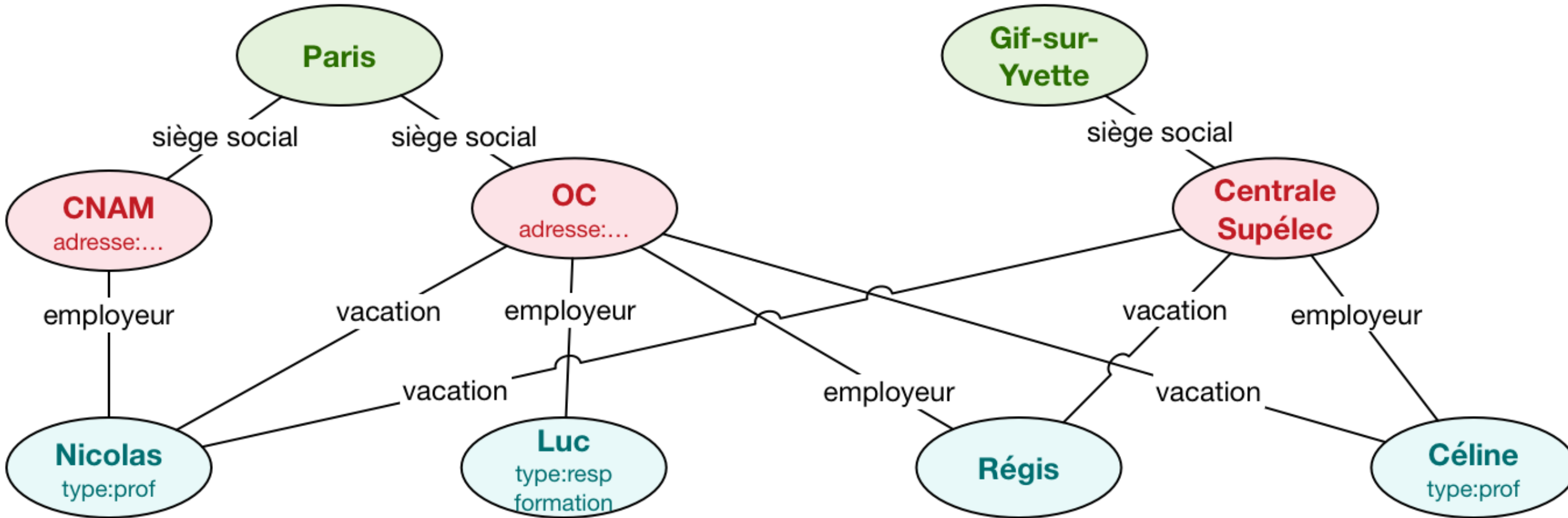
Orientées Graphes

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



g.com

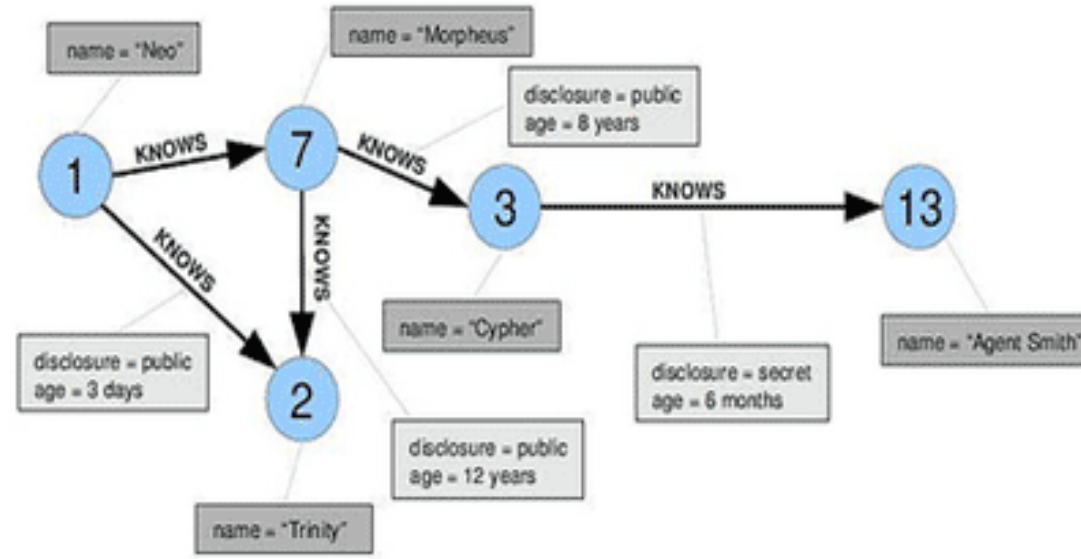
Données orientées graphe : Exemple



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 opérations sur les ensembles

Exemple d'implémentation



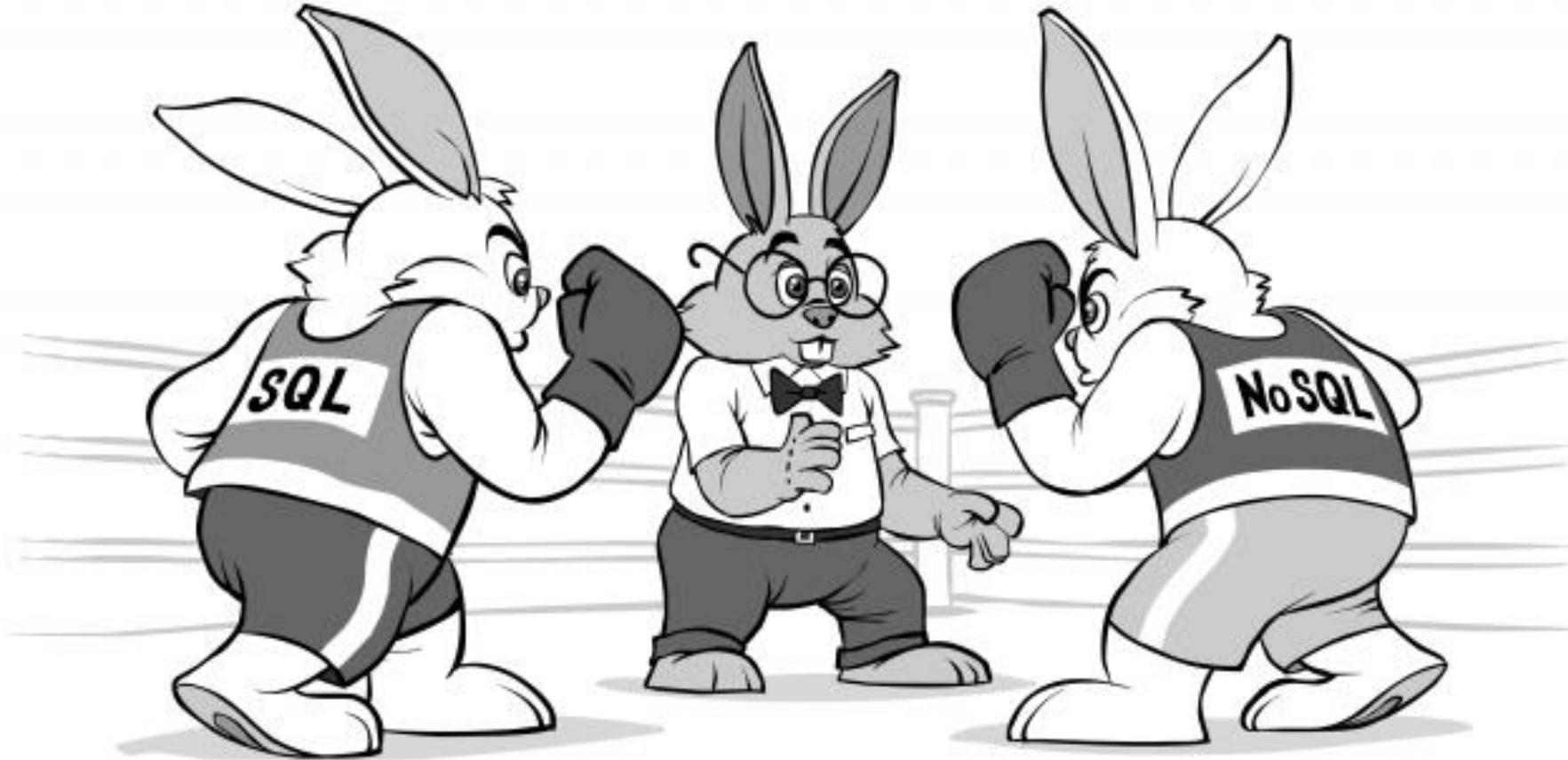
Classement des moteurs orientés graphe

include secondary database models

39 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Neo4j	Graph	55.43	-0.41	-2.81
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	36.51	-1.45	-3.45
3.	3.	4.	Virtuoso	Multi-model	6.11	+0.23	+0.72
4.	4.	3.	ArangoDB	Multi-model	5.29	+0.22	-0.11
5.	5.	5.	OrientDB	Multi-model	4.54	+0.06	-0.49
6.	7.	8.	JanusGraph	Graph	2.79	+0.16	+0.44
7.	6.	6.	Amazon Neptune	Multi-model	2.73	-0.08	-0.26
8.	8.	7.	GraphDB	Multi-model	2.44	-0.09	-0.49
9.	9.	9.	TigerGraph	Graph	2.16	-0.04	-0.08
10.	11.	12.	Fauna	Multi-model	1.89	+0.12	+0.57
11.	10.	11.	Dgraph	Graph	1.86	+0.06	+0.14
12.	14.	15.	NebulaGraph	Graph	1.79	+0.28	+0.63
13.	17.	20.	Memgraph	Graph	1.68	+0.36	+1.30
14.	12.	10.	Stardog	Multi-model	1.63	+0.01	-0.35
15.	13.	13.	Giraph	Graph	1.59	+0.06	+0.28

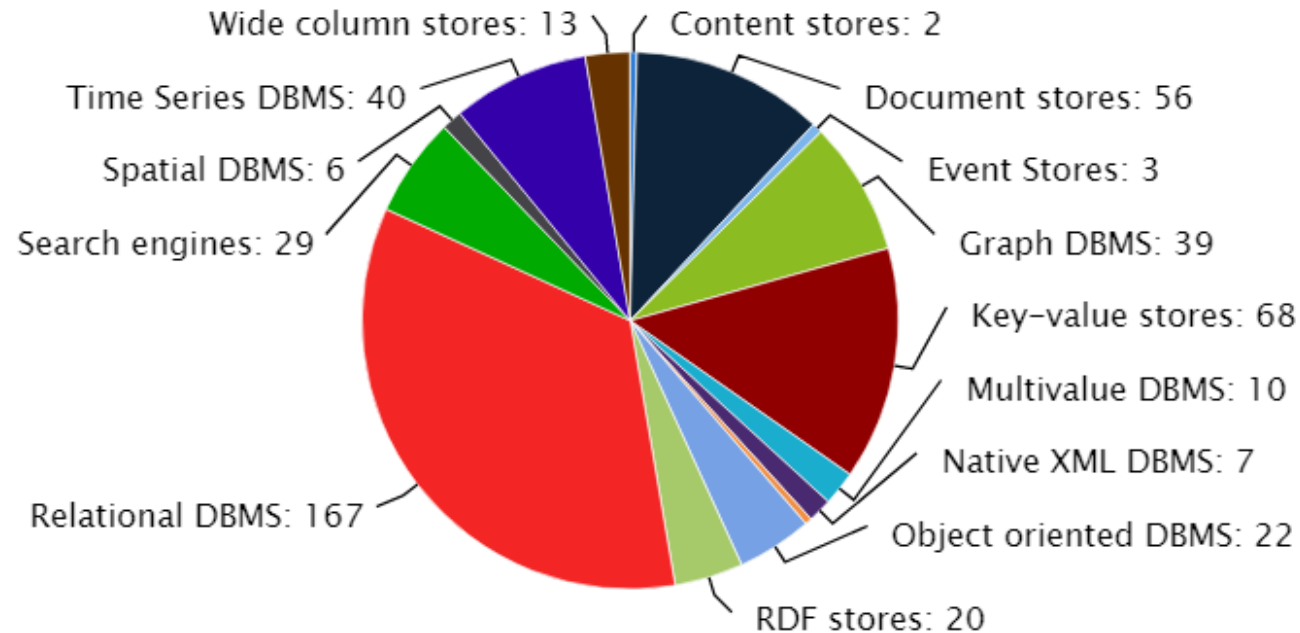
SQL vs NoSQL



SQL vs NoSQL : nombre de moteurs

DBMS popularity broken down by database model

Number of systems per category, February 2023



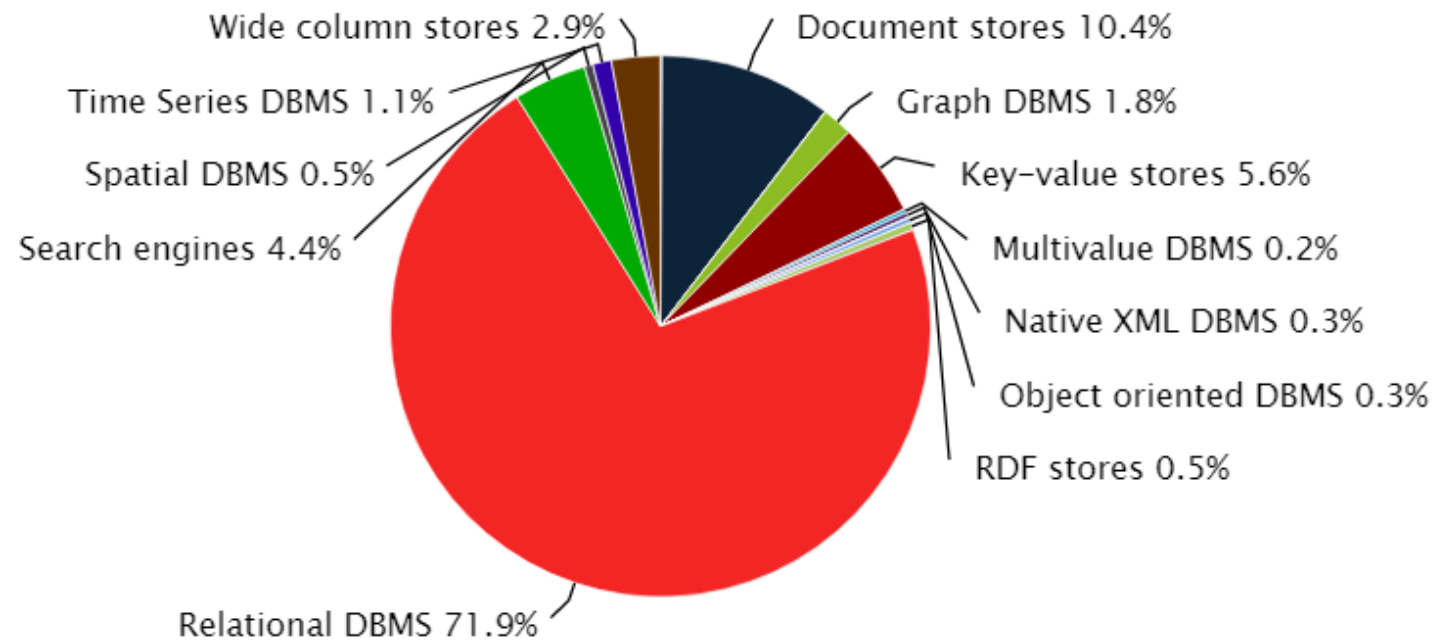
© 2023, DB-Engines.com

DB-Engines lists 410 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 : pourcentage d'utilisation

Ranking scores per category in percent, February 2023



© 2023, DB-Engines.com

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

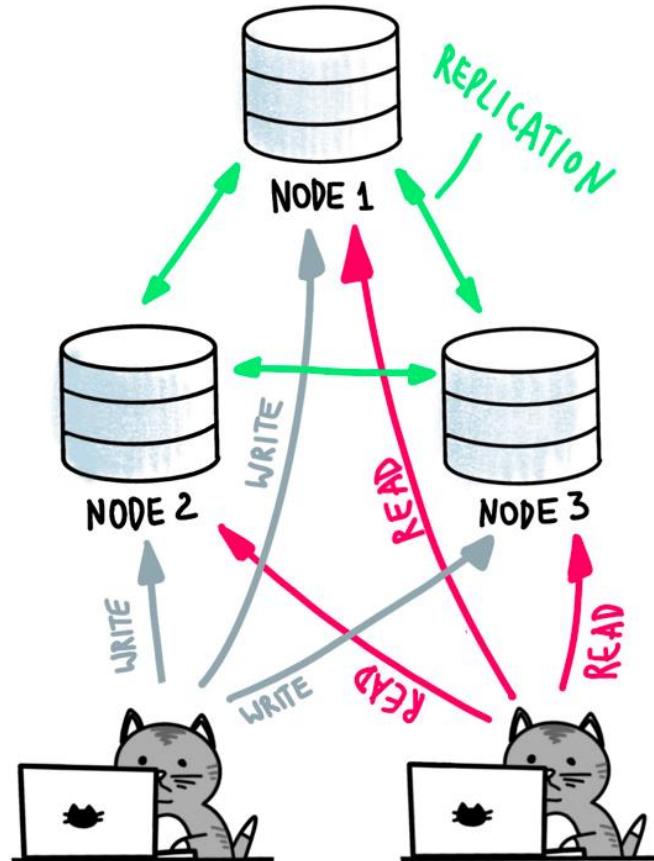
SQL vs NoSQL : ACID vs BASE

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

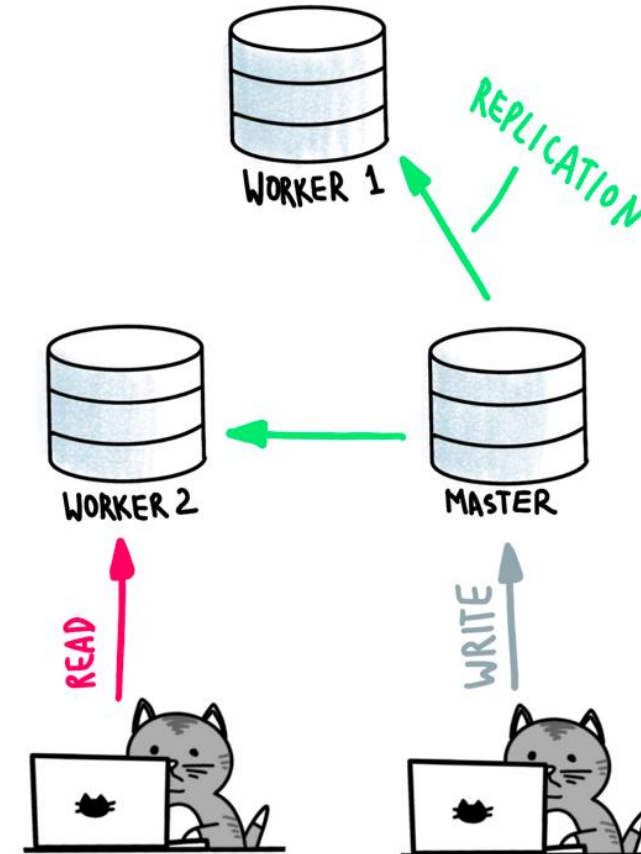


- *Basically Available* : quelle que soit la charge de la base de données (données/requêtes), le système garantit un taux de disponibilité de la donnée (mais réponse à la requête pouvant contenir des erreurs)
- *Soft-state* : l'état de la BD n'est pas garanti à un instant donné (les mises à jour ne sont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

ACID / BASE

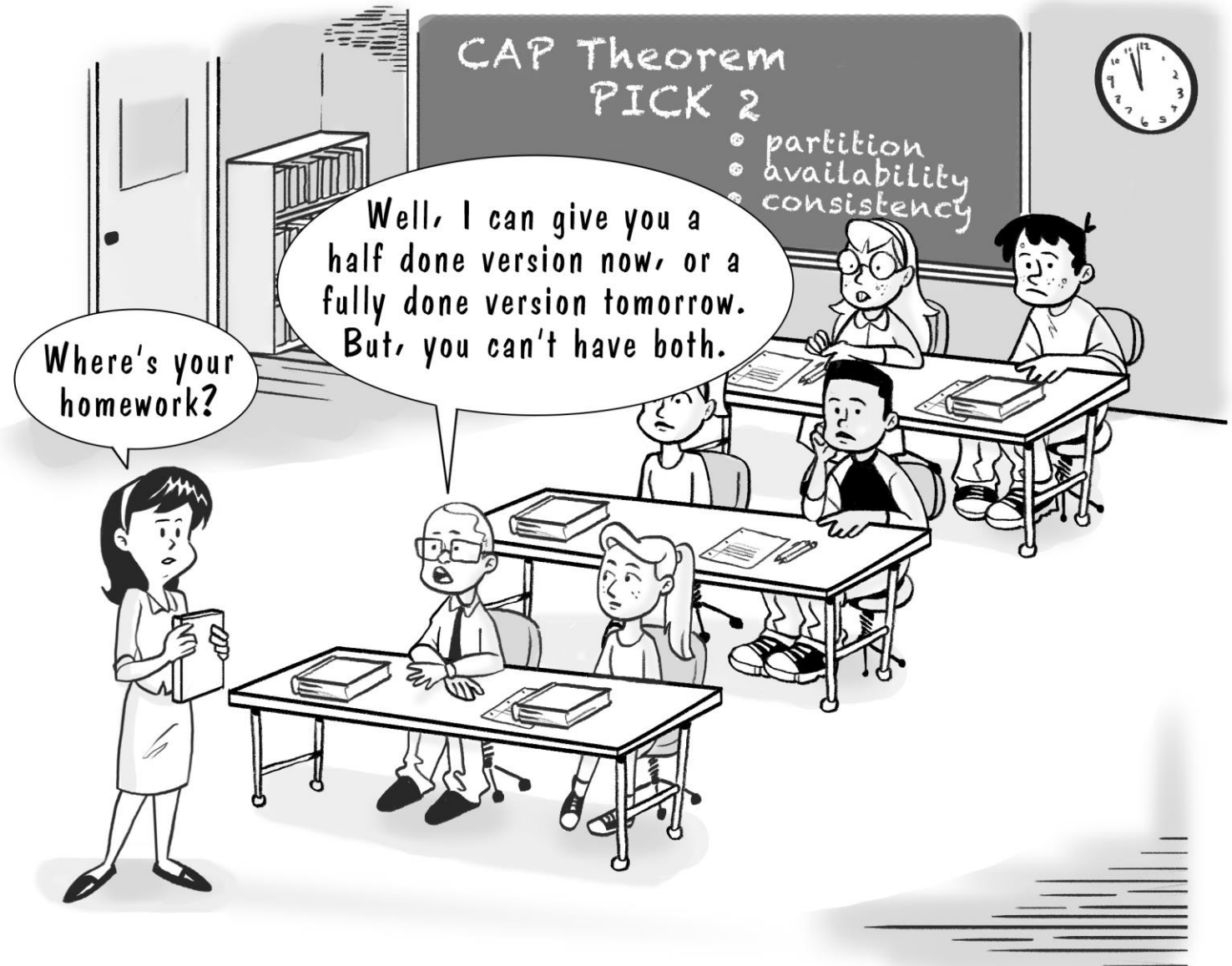


VS



@luminousmen.com

Théorème de CAP



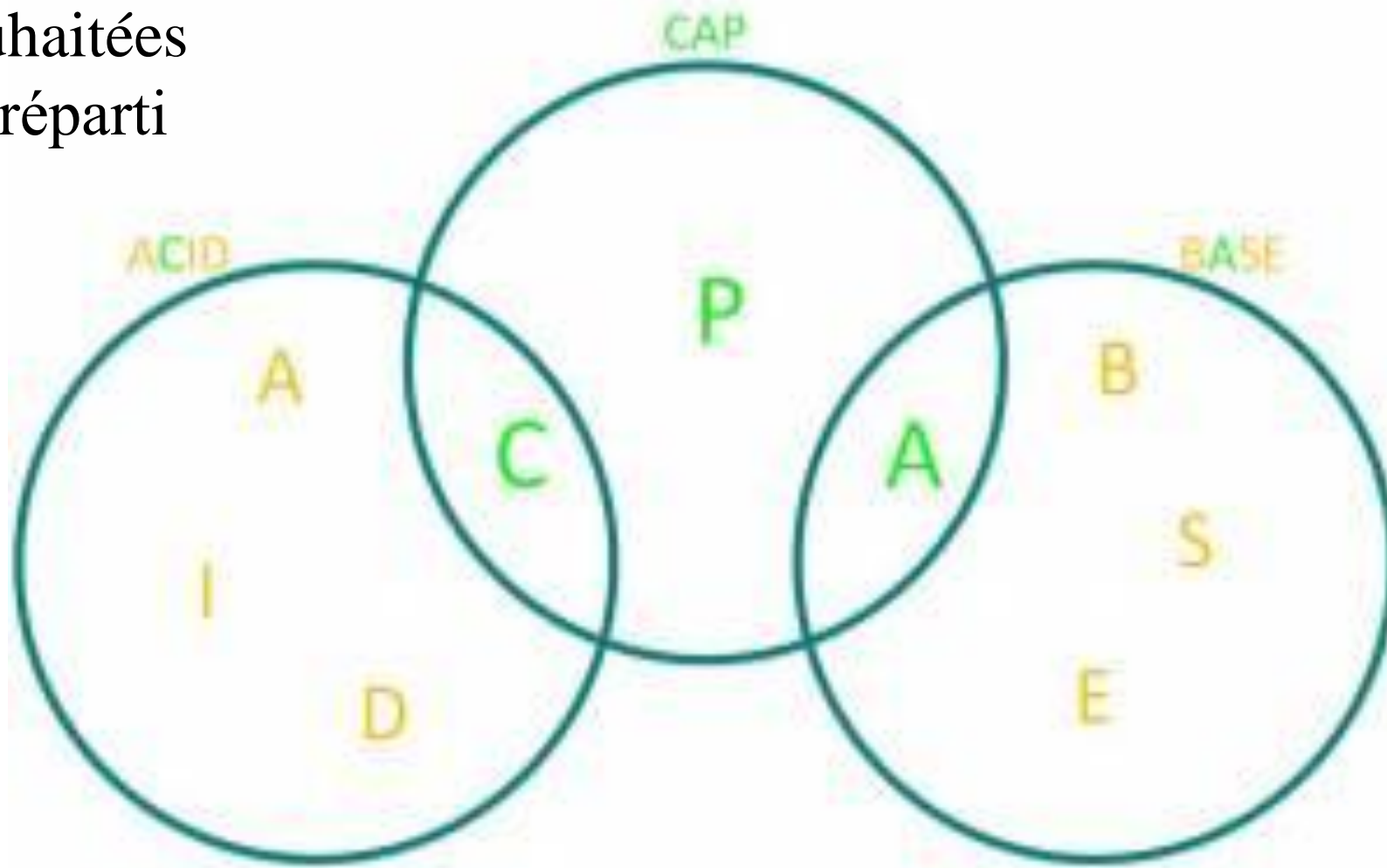
Brewer, Eric A. "Towards robust distributed systems." *PODC*. Vol. 7. No. 10.1145. 2000.

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 Tolerance

C : Consistency

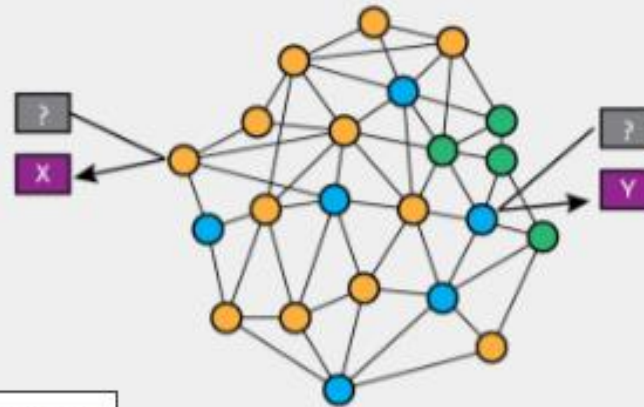
At any given time, all nodes in the network have exactly the same (most recent) **value**.



● = 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

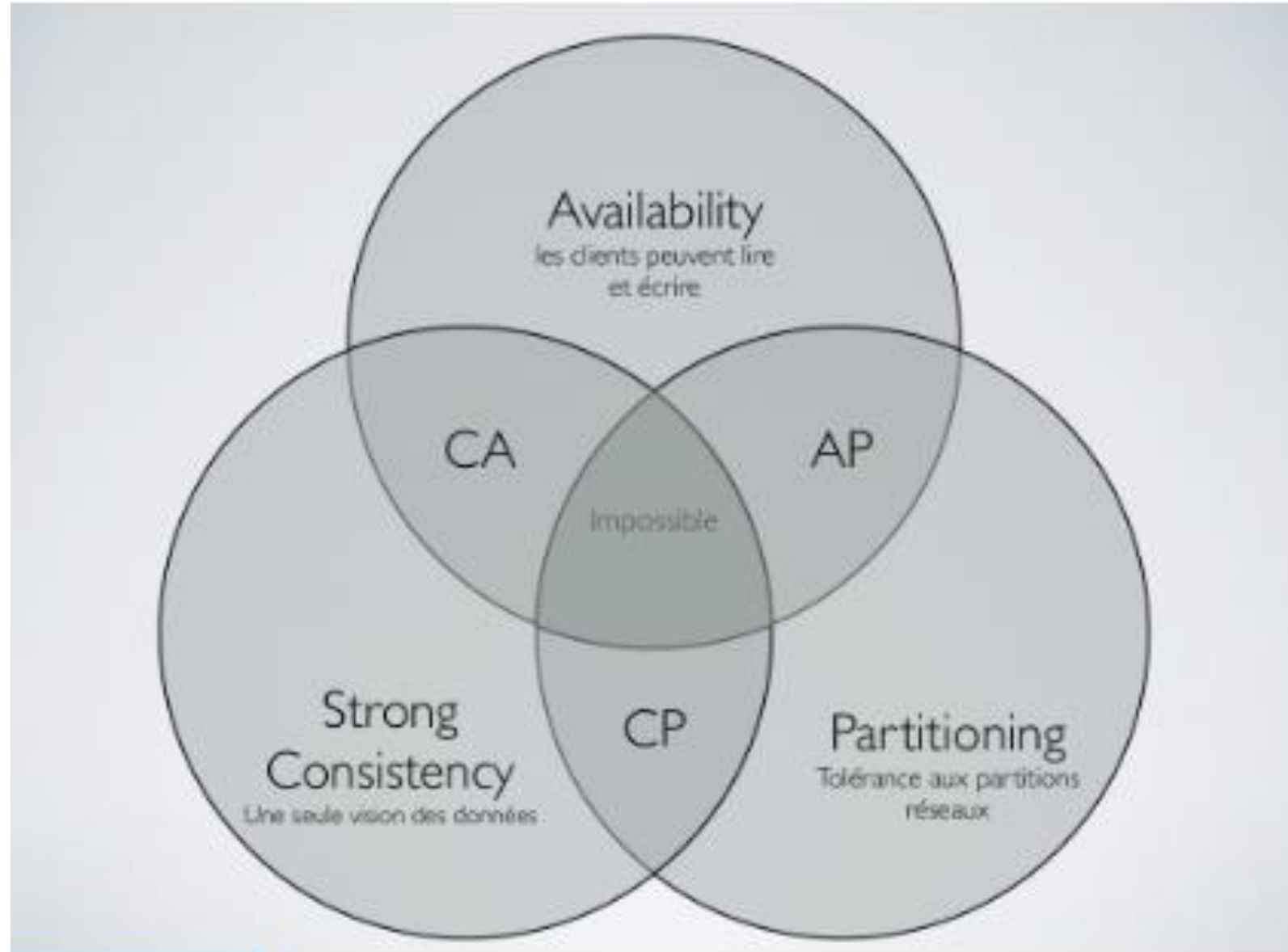
● = Value: Y @ 2018-05-03T07:12:12

P : Partition tolerance

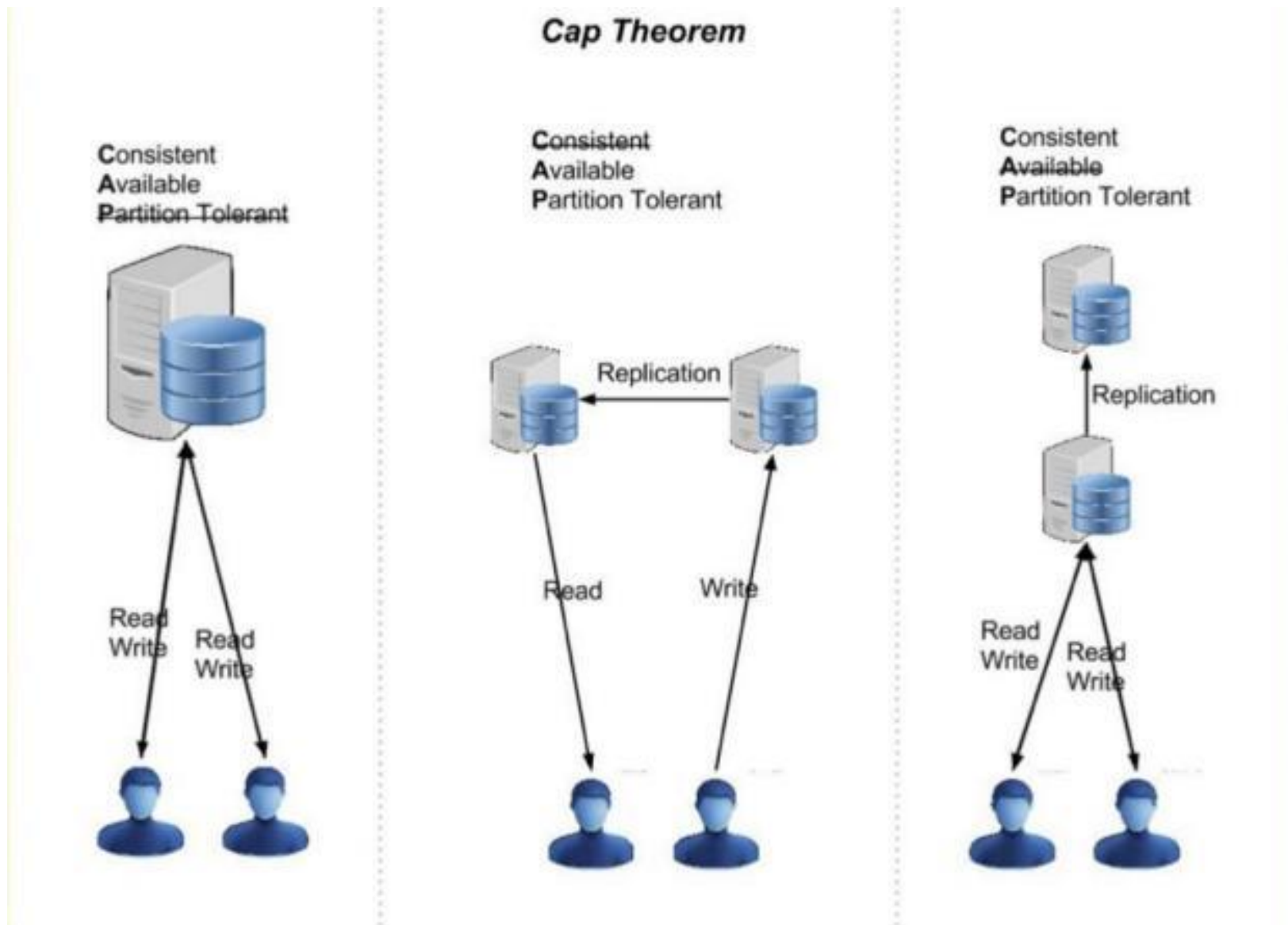
The network continues to operate, even if an arbitrary number of nodes are **failing**.



Théorème de CAP : impossibilité d'avoir les 3 propriétés

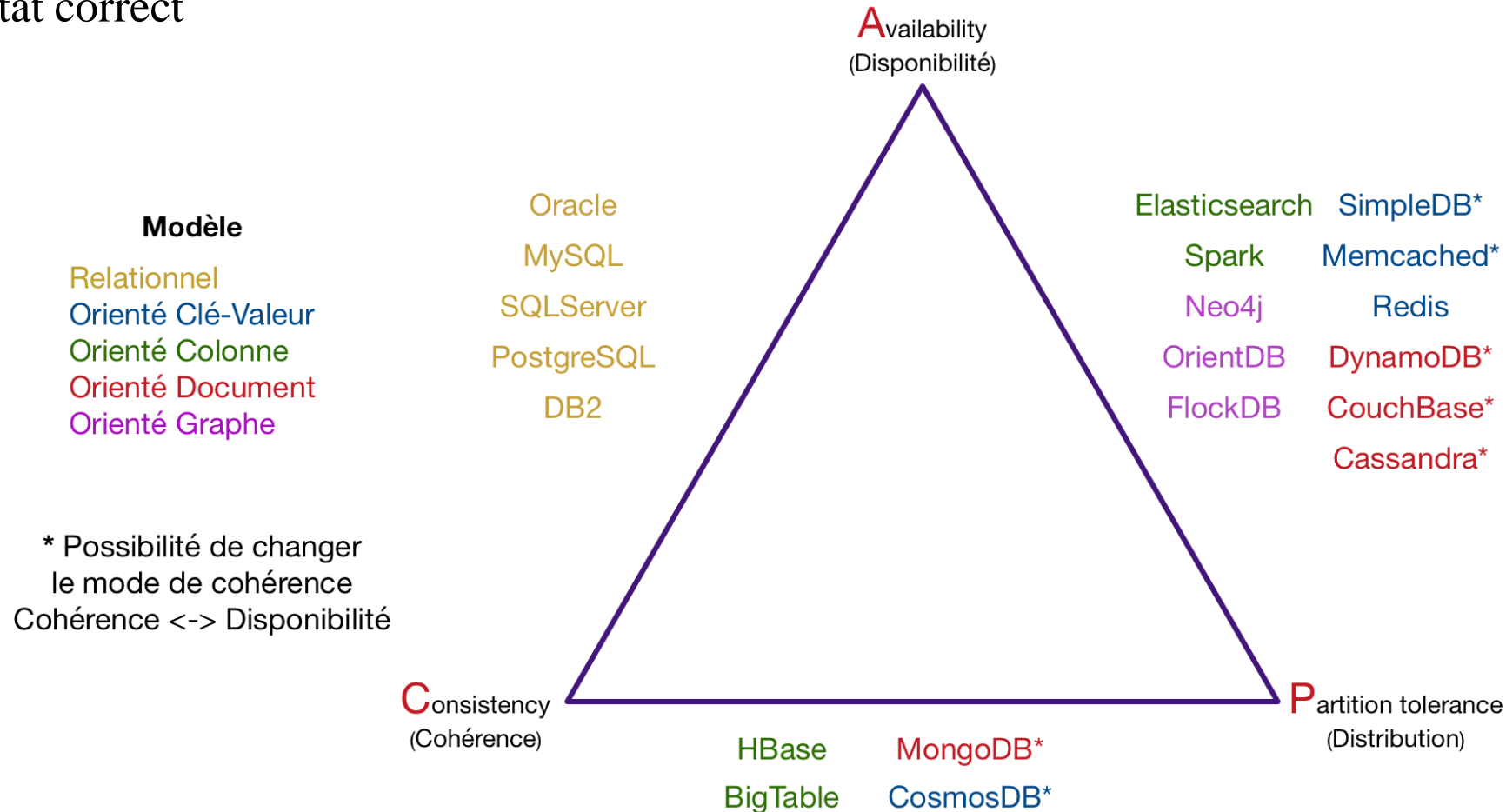


Théorème de CAP : 2 propriétés sur 3



Positionnement des moteurs selon CAP

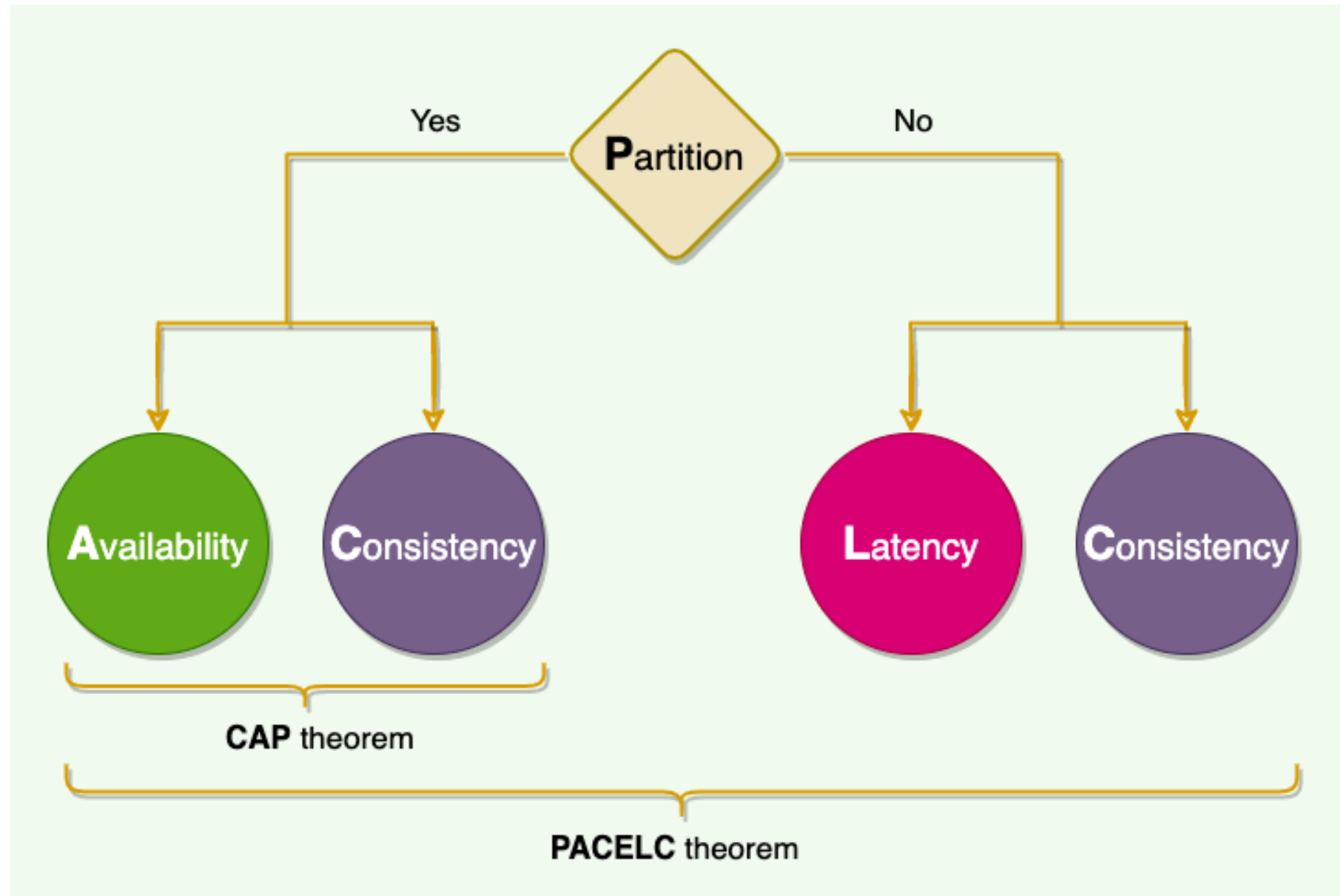
- **Consistency** (Cohérence) : une donnée n'a qu'un seul état visible quel que soit le nombre de réplicas
- **Availability** (Disponibilité) : tant que le système tourne (distribué ou non), la donnée doit être disponible
- **Partition Tolerance** (Distribution) : quel que soit le nombre de serveurs,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

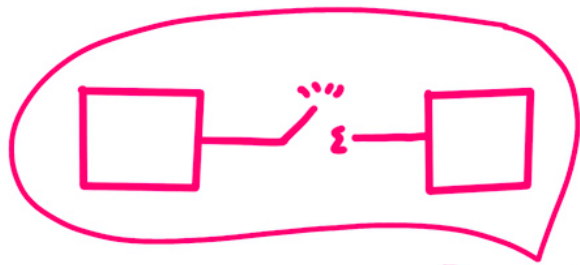


PACELC : alternative basée sur CAP incluant la latence

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

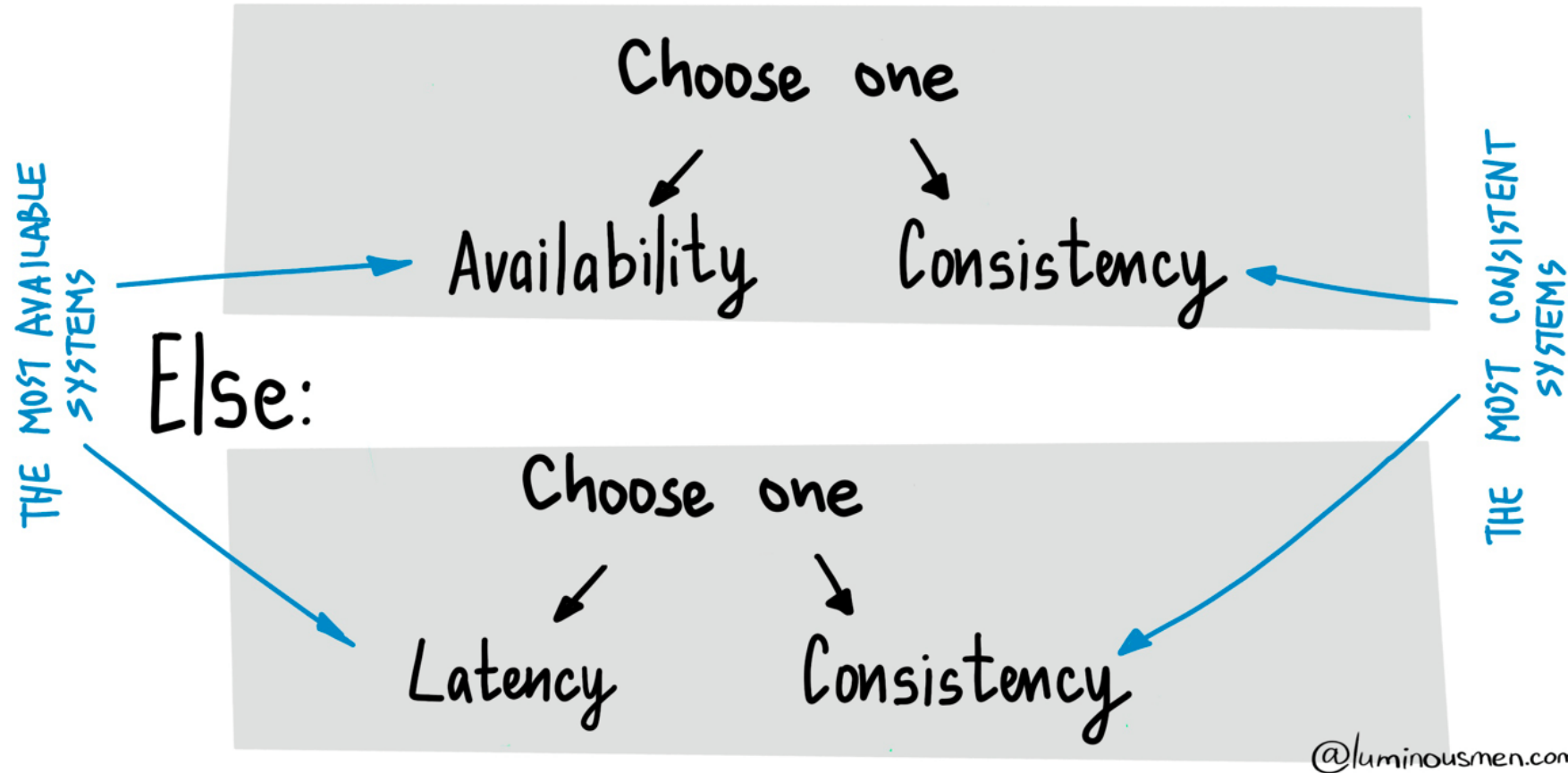
<https://www.cs.umd.edu/~abadi/papers/abadi-pacelc.pdf>





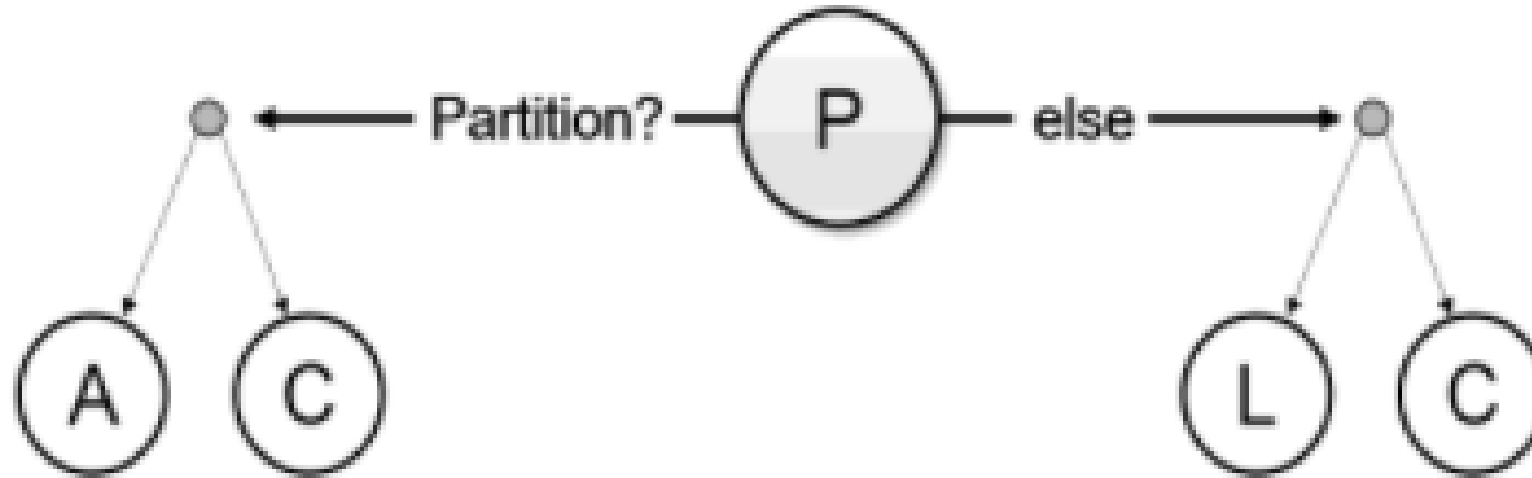
PACELC

If Partition happened:



@luminousmen.com

PACELC : positionnement des moteurs



PA / EL

DynamoDB, Cassandra, Riak

PG / EC

VoldDB, H-Store, Megastore, BigTable, Hbase

PA / EC

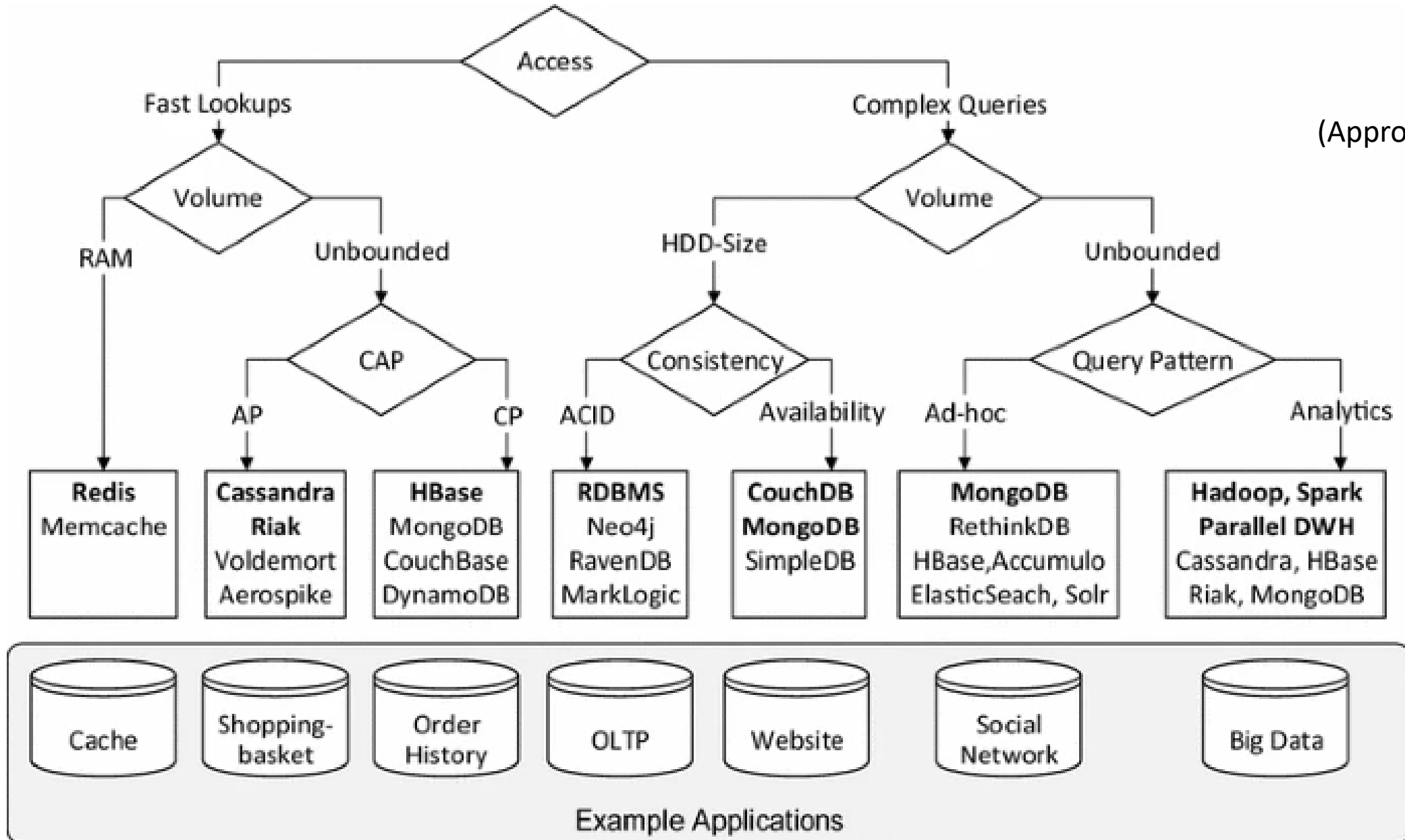
MongoDB

PG / EL

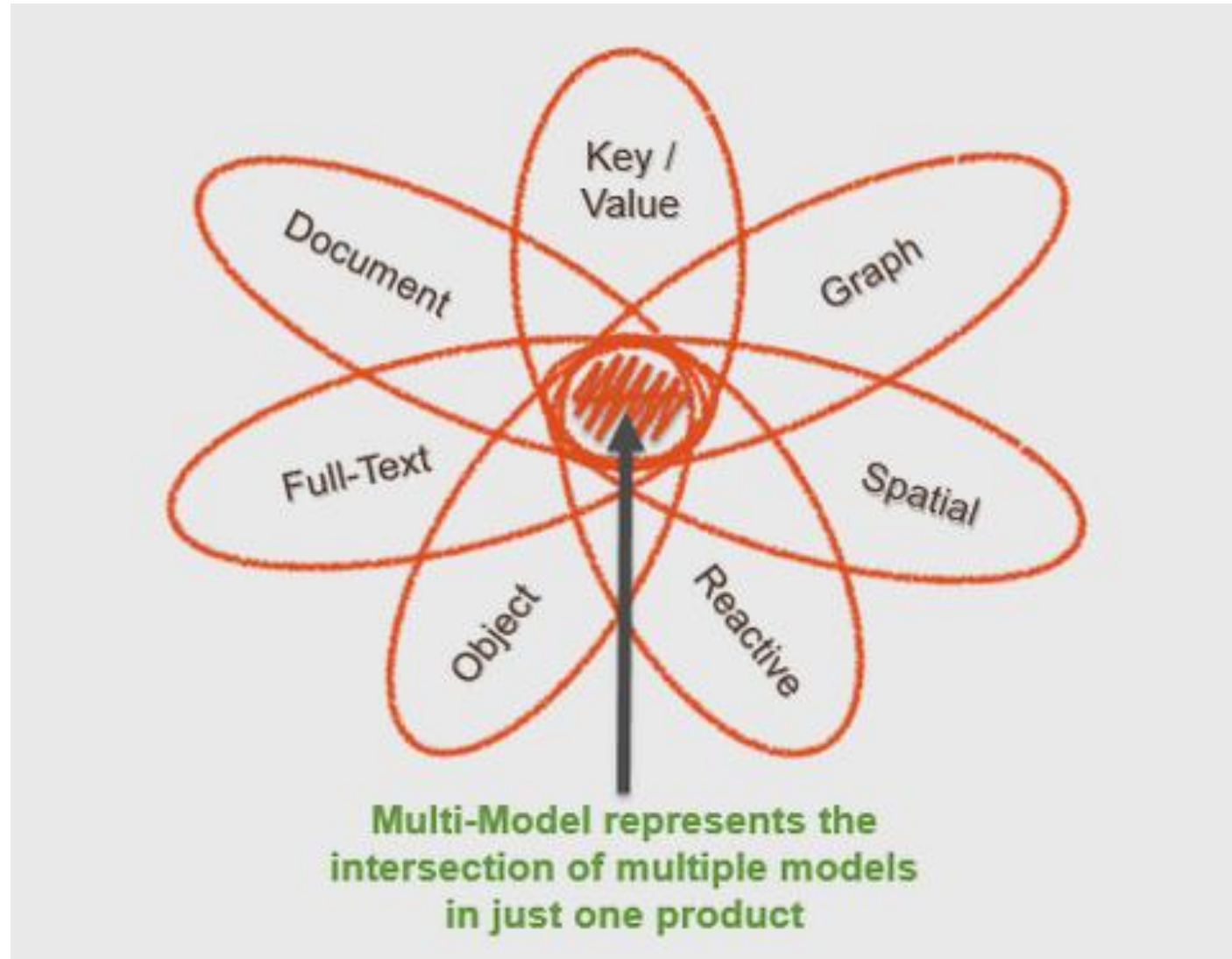
PNUTS

Choisir le bon modèle

(Approche de 2016)



Systeme multi-modeles



Classement des systèmes multi-modèles

410 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle	Relational, Multi-model	187.50	-9.31	-9.31
2.	2.	2.	MySQL	Relational, Multi-model	187.50	19.23	19.23
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	187.50	19.96	19.96
4.	4.	4.	PostgreSQL	Relational, Multi-model	187.50	+7.12	+7.12
5.	5.	5.	MongoDB	Document, Multi-model	187.50	35.88	35.88
6.	6.	6.	Redis	Key-value, Multi-model	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model	138.60	-2.56	-23.70
9.	10.	10.	SQLite	Relational	132.67	+1.17	+4.30
10.	9.	9.	Microsoft Access	Relational	131.03	-2.33	-0.23

Relational DBMS,
Document store,
Graph DBMS,
RDF store,
Spatial DBMS

Exemples de système multi-modèles

From: [Holistic evaluation in multi-model databases benchmarking](#) 2019

System	Query language	Primary model	Secondary model	Storage strategy
AgensGraph	OpenCypher, SQL	Relational	Graph, JSON	One engine
ArangoDB	AQL	JSON	Graph, Key-value	One engine
OrientDB	SQL-like	Graph	JSON, Key-value	One engine
PostgreSQL	SQL-extension	Relational	ALL but graph	One engine
Marklogic	Xpath	XML	JSON, RDF	One engine
MongoDB	API	JSON	Graph	One engine
Redis	API	Key-value	Graph, JSON	One engine
Spark SQL	SQL-like	DataFrame	Graph, JSON, Key-value	One engine
Datastax	CQL	Column	JSON, Graph	Multiple engines
DynamoDB	API, SQL	–	JSON, Graph, Key-value	Multiple engines
CosmosDB	API, SQL	–	ALL but XML	Multiple engines
Oracle 12c	SQL-extension	Relational	ALL	Both

Bases de données polyglottes

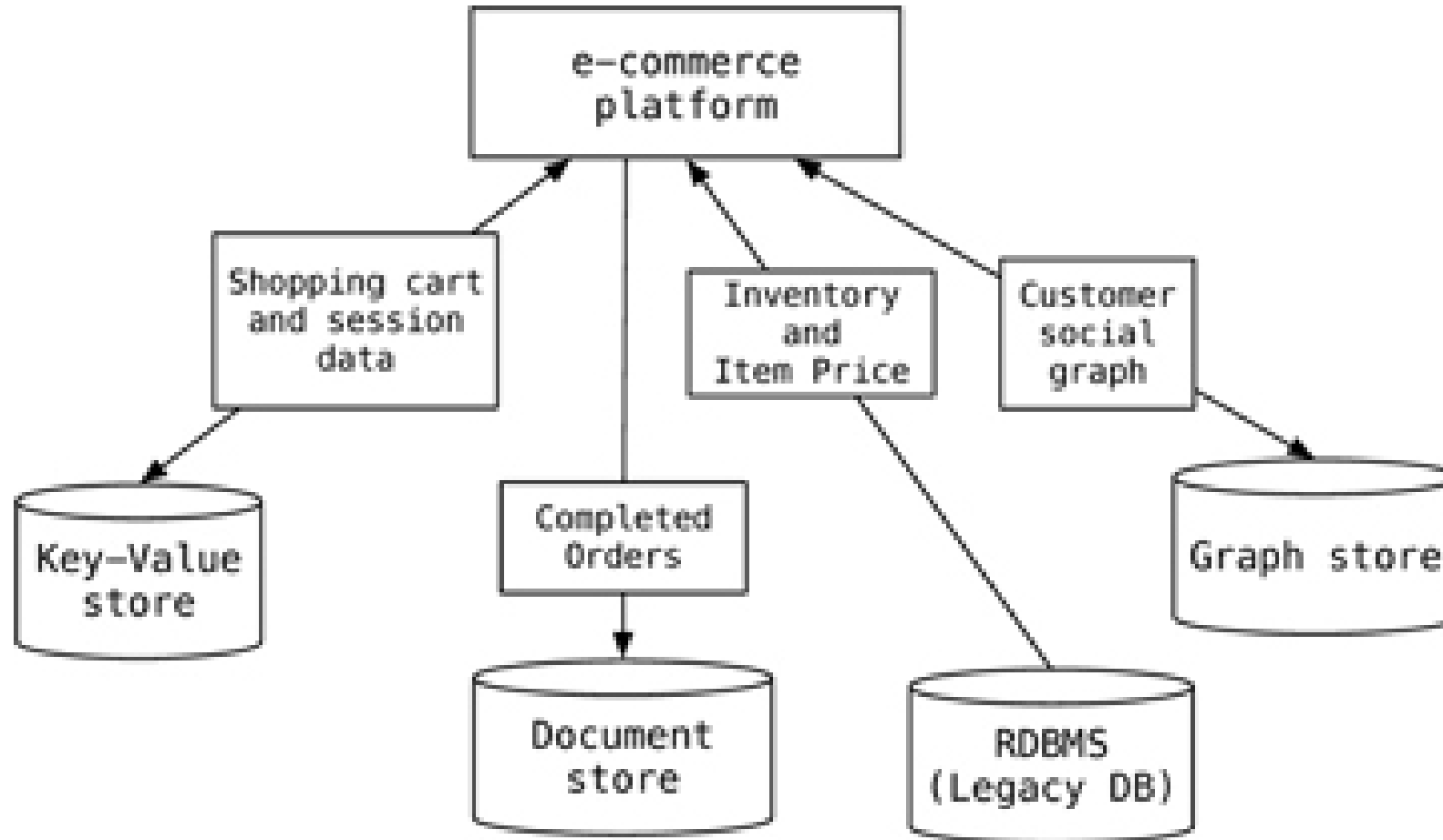
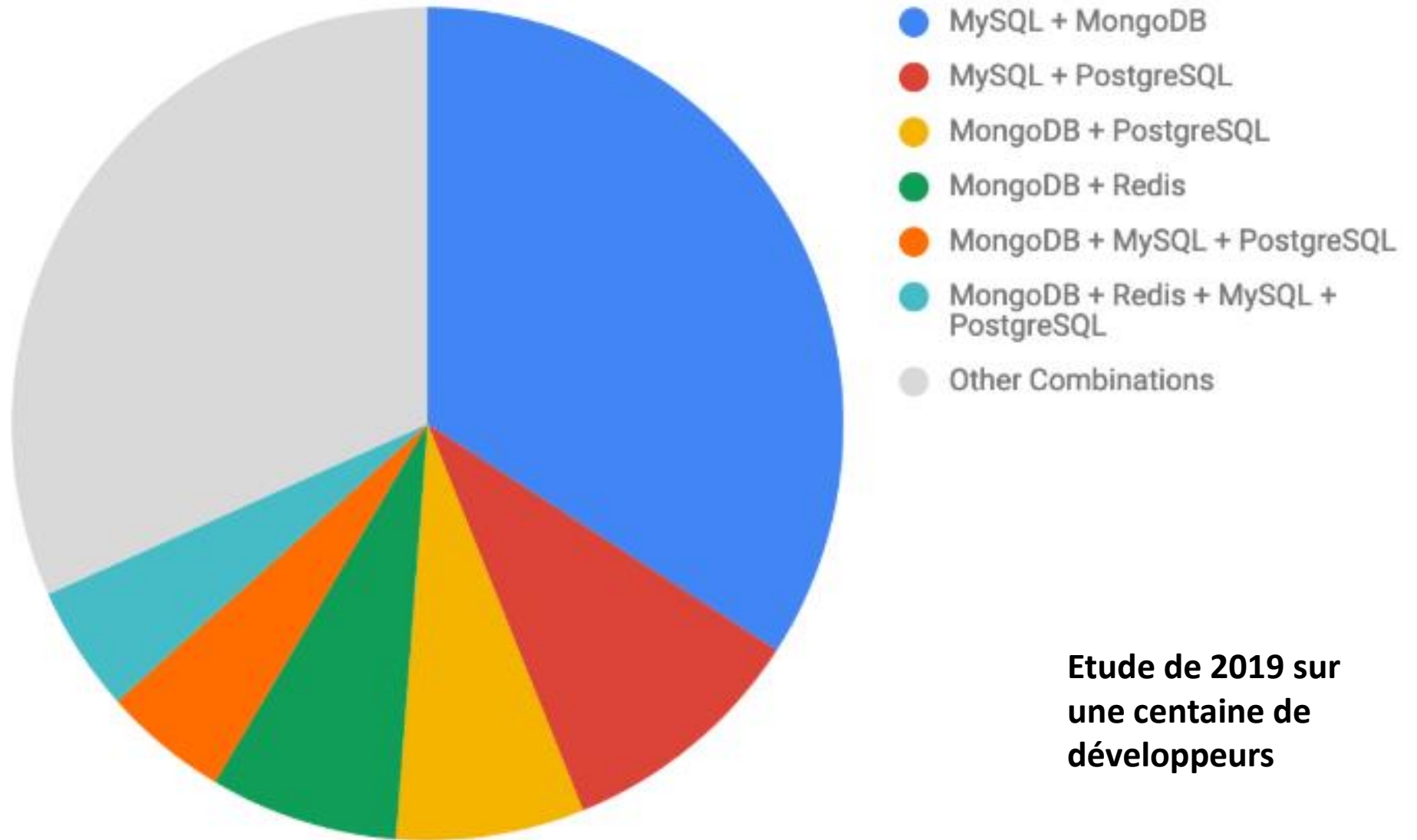


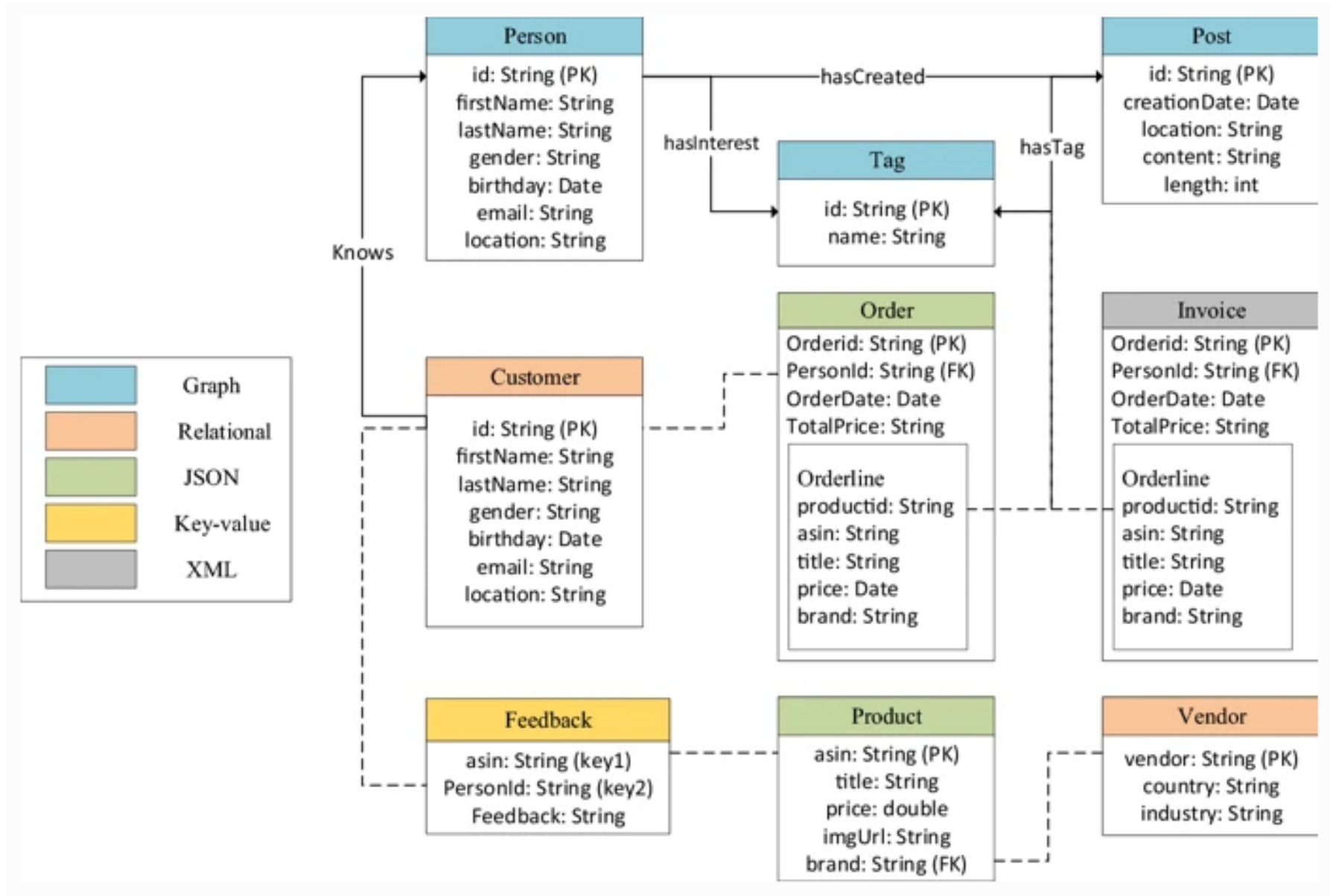
Figure reprise de P. J. Sadalage et M. Fowler (2009). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison Wesley

Bases de données polyglottes : tendances

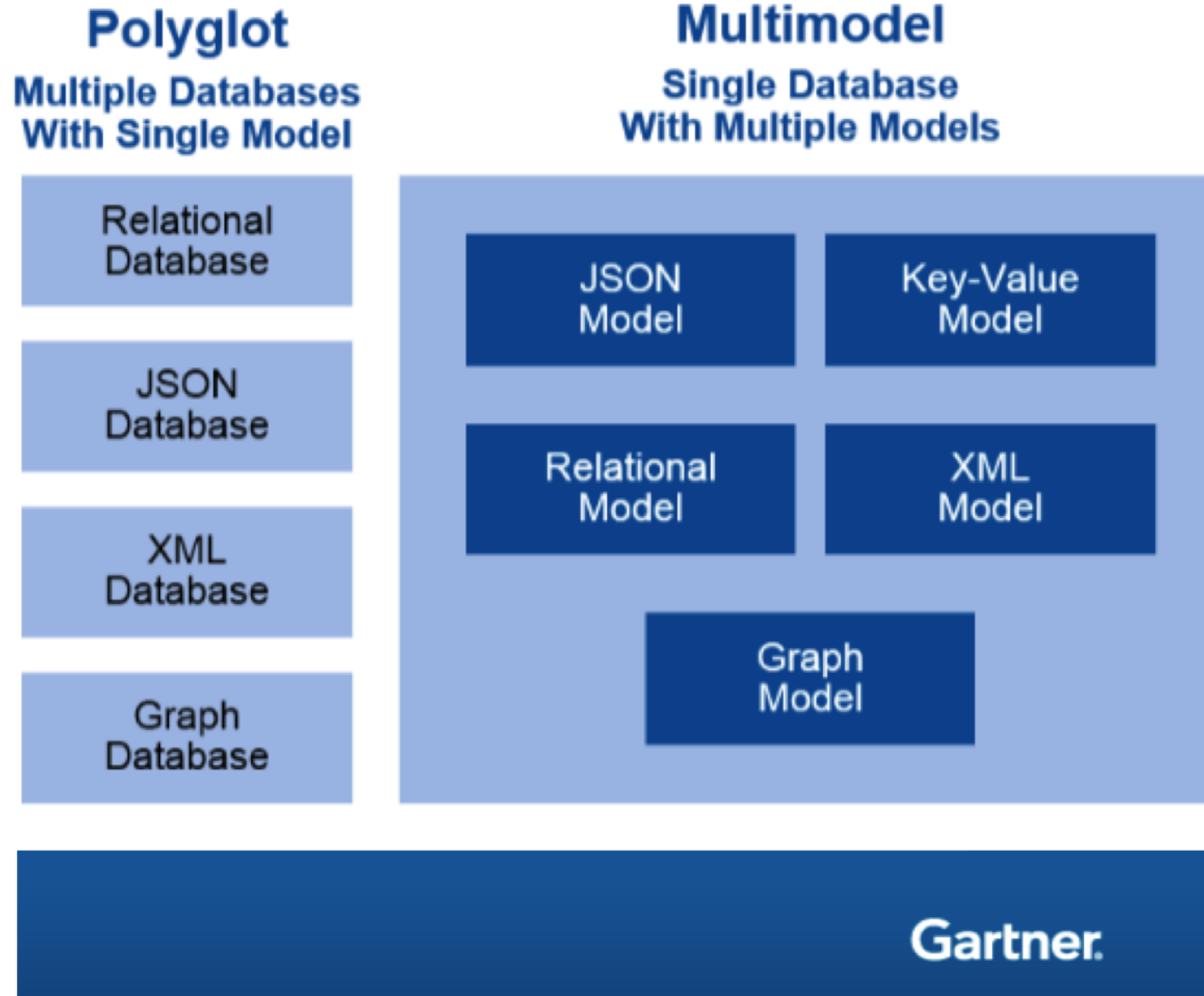


**Etude de 2019 sur
une centaine de
développeurs**

Exemple de base polyglotte



Bases de données polyglottes versus multi-modèles



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	134. ↓ 127. ↑ 138.	VoltDB	Relational	2.29	-0.11	+0.60
▪ NuoDB	186. ↑ 189. ↓ 176.	NuoDB	Relational	1.34	+0.15	+0.34
▪ Google Cloud Spanner	92. ↓ 91. 92.	Google Cloud Spanner	Relational	4.38	-0.02	+0.69
▪ CockroachDB	57. 57. ↑ 58.	CockroachDB	Relational	7.27	+0.08	-0.20
▪ YugabyteDB	93. ↓ 86. ↑ 117.	YugabyteDB	Relational, Multi-model	4.38	-0.13	+2.04

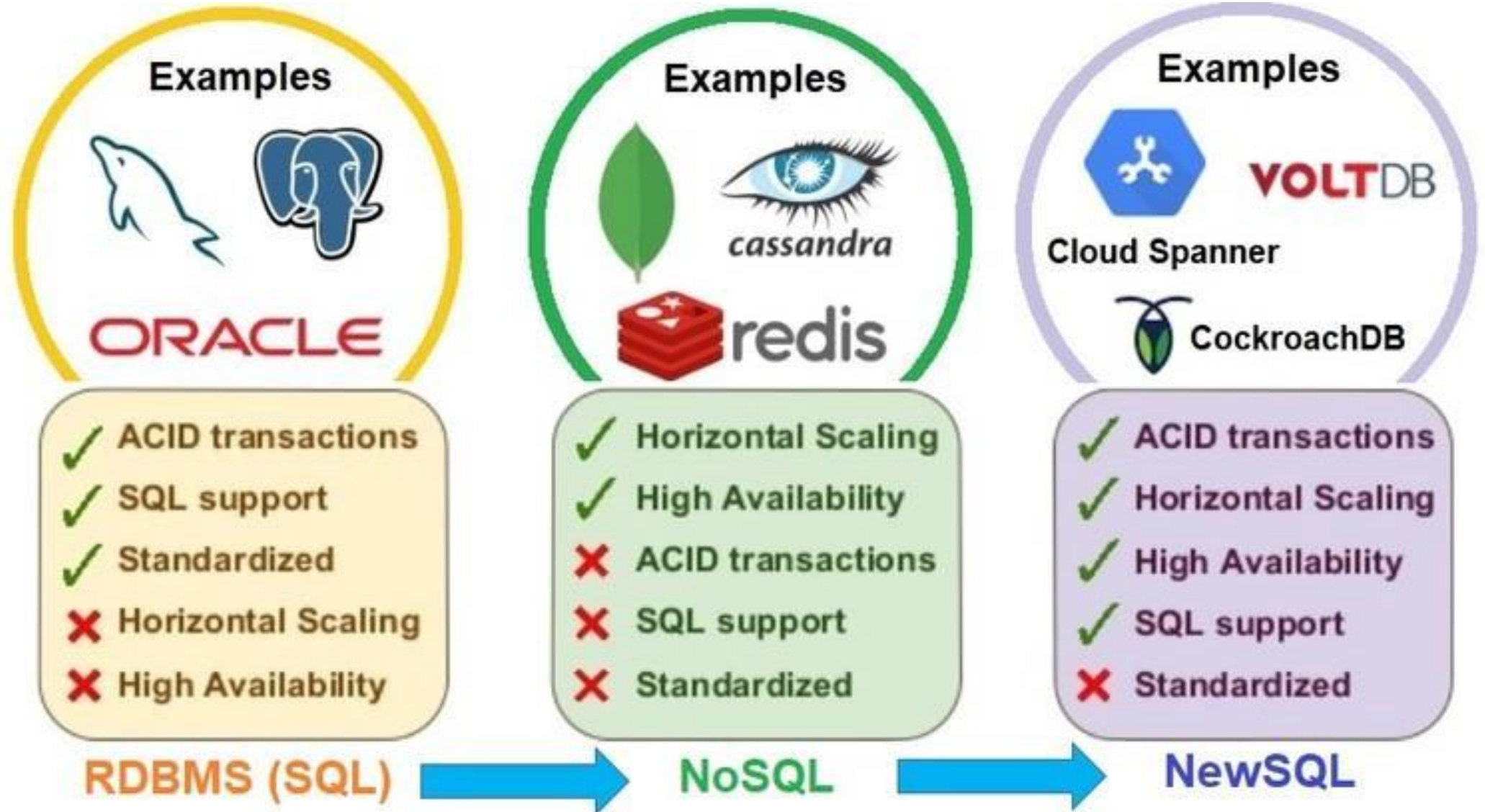
NewSQL : cas d'utilisation

- 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

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

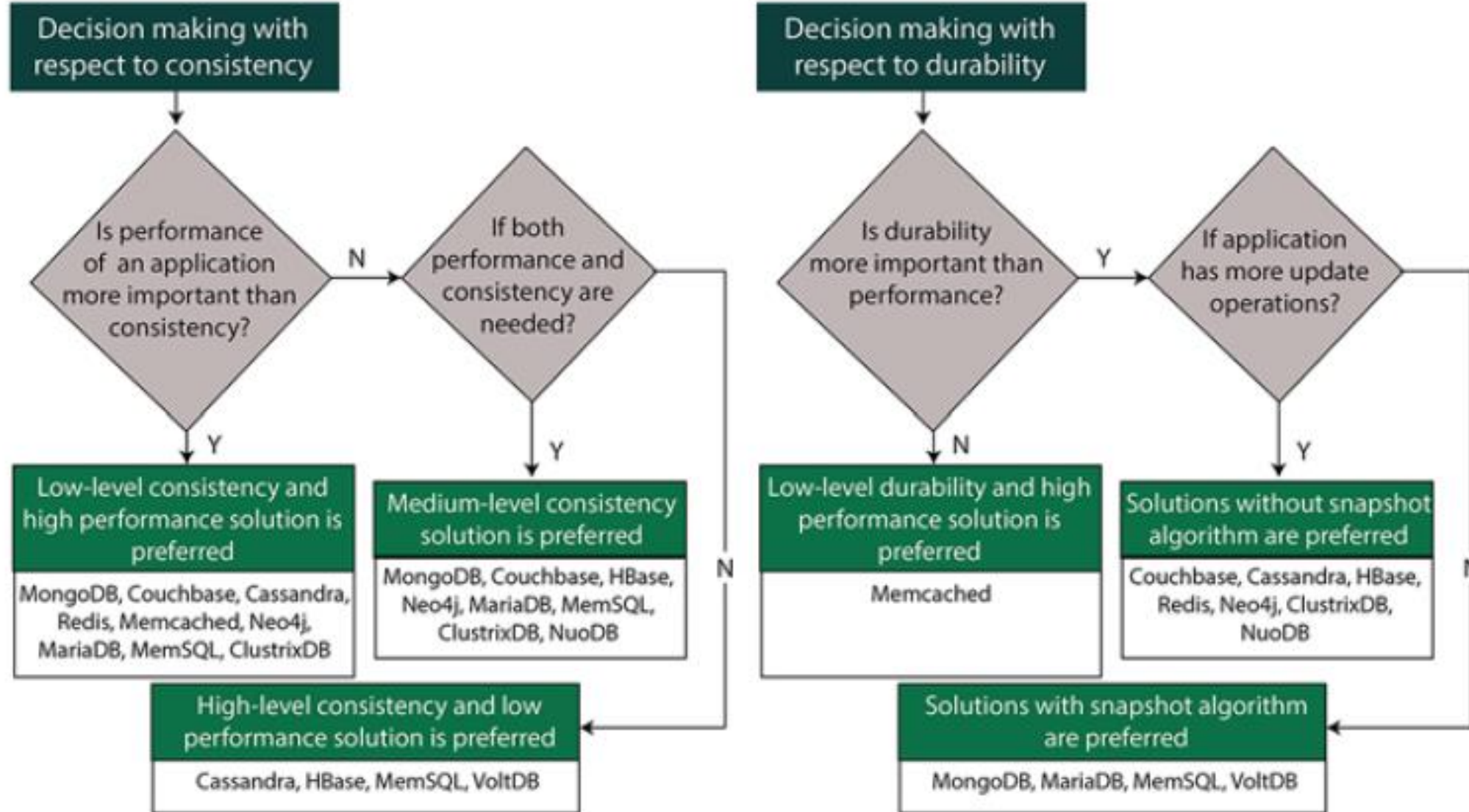
Exemples de moteurs NewSQL

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

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


Choisir le bon moteur de bases de données




(Extrait d'une approche de 2020)

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

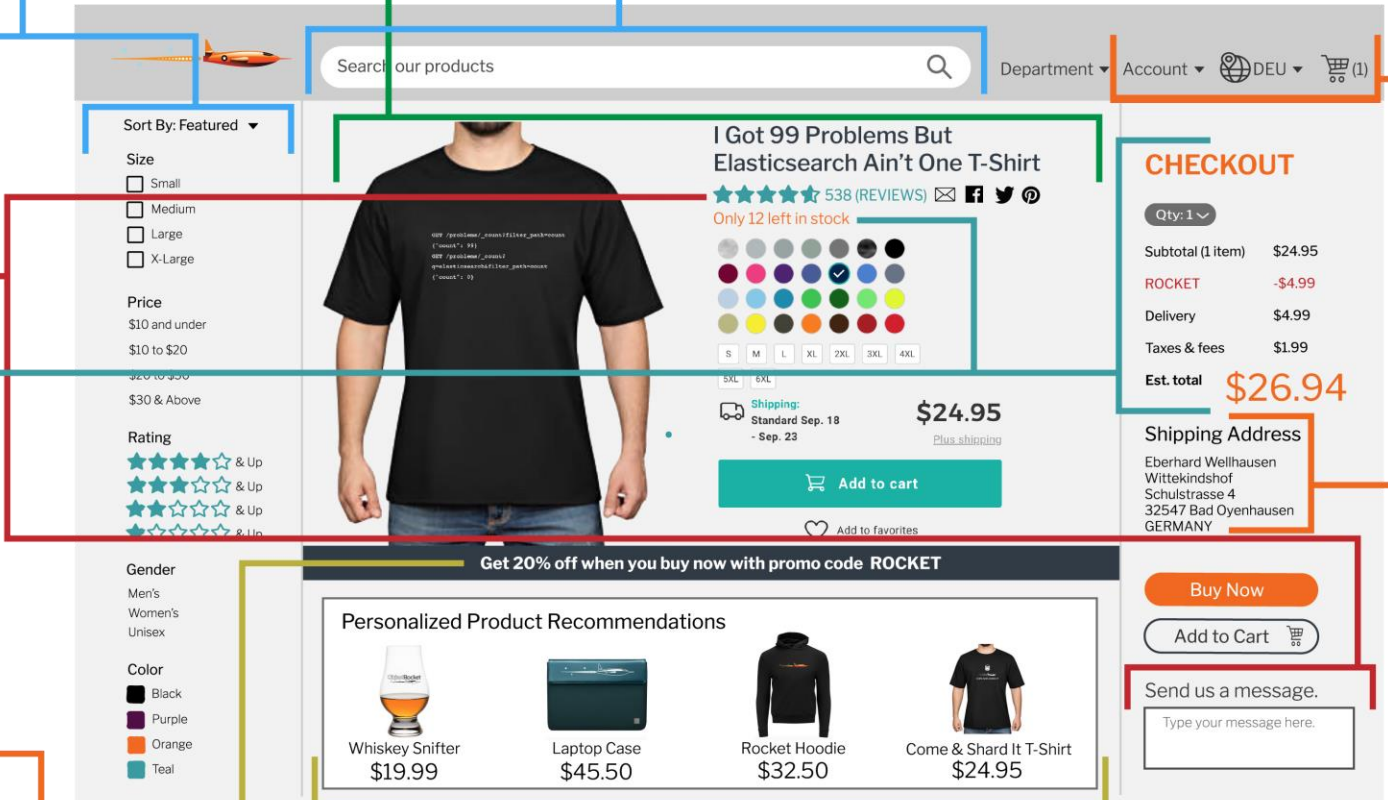
 ✓ Checkout
✓ Inventory
✓ Order Management

 CockroachDB
Distributed SQL

 ✓ Geo-Partitioning for
Low Latency & Compliance

 Hadoop
Wide-Column Store

 ✓ Customized Product Recommendations
✓ Activity-Based Deals/Offers



The screenshot shows a product page for a t-shirt titled "I Got 99 Problems But Elasticsearch Ain't One T-Shirt". The page includes a search bar, a navigation menu, a product image, a size selection dropdown, a price range filter, a rating section, a gender selection dropdown, a color selection dropdown, and a personalized product recommendations section. The checkout section shows a subtotal of \$24.95, a discount of \$4.99, a delivery fee of \$4.99, and taxes & fees of \$1.99, resulting in an estimated total of \$26.94. The shipping address is Eberhard Wellhausen, Wittekinds Hof, Schulstrasse 4, 32547 Bad Oeyenhausen, GERMANY. A promotional code "ROCKET" is applied, offering 20% off. The personalized product recommendations section shows four items: Whiskey Snifter (\$19.99), Laptop Case (\$45.50), Rocket Hoodie (\$32.50), and Come & Share It T-Shirt (\$24.95).