Au-delà des bases de données relationnelles

Les bases de données NoSQL

- Not Only SQL propose de laisser de côté certaines contraintes des bases de données relationnelles. (dénormalisation)
- Dans ce contexte, il est plus intéressant d'avoir un langage de haut niveau pour exploiter les bases de données.
- Contrairement aux BD SQL, qui fonctionnent toutes sous le même principe, il existe plusieurs types de BD No SQL
  - Clé/Valeurs: Redis(VmWare) , SimpleDB (Amazon)
  - Des lignes vers les colonnes: le stockage des données est sous forme de colonne plutôt que de lignes. BigTable(Google), Hbase
  - Gestion de documents: MongoDB, Cassandra.
  - Orienté Graph : Neo4J

Les bases de données NoSQL

Avantages

• Permet de gérer rapidement des tonnes de données (grand volume à une vitesse rapide).

• Schémas dynamiques pour les données non structurées (évolutifs, n'a pas à être connu d'avance).

• Plusieurs façons de stocker des données.

• Moins coûteux (ajout de serveurs).

Inconvénients

• La cohérence des données n'est pas garantie.

• Pas de langage de requête abstrait partagé, donc un travail de programmation spécifique plus
important.

Les bases de données NoSQL

Part du marché des SGBD : Les SGBD relationnels dominent le marché

Quand utilise-t-on le SQL ?

• Les données doivent être structurées. L'organisation est connue (ou pourrait être connue) d'avance.
• L'intégrité des données doit-être respectée
• Les transactions sont importantes. Le principe ACID est important :

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)

Quand utilise t-on le NoSQL ?

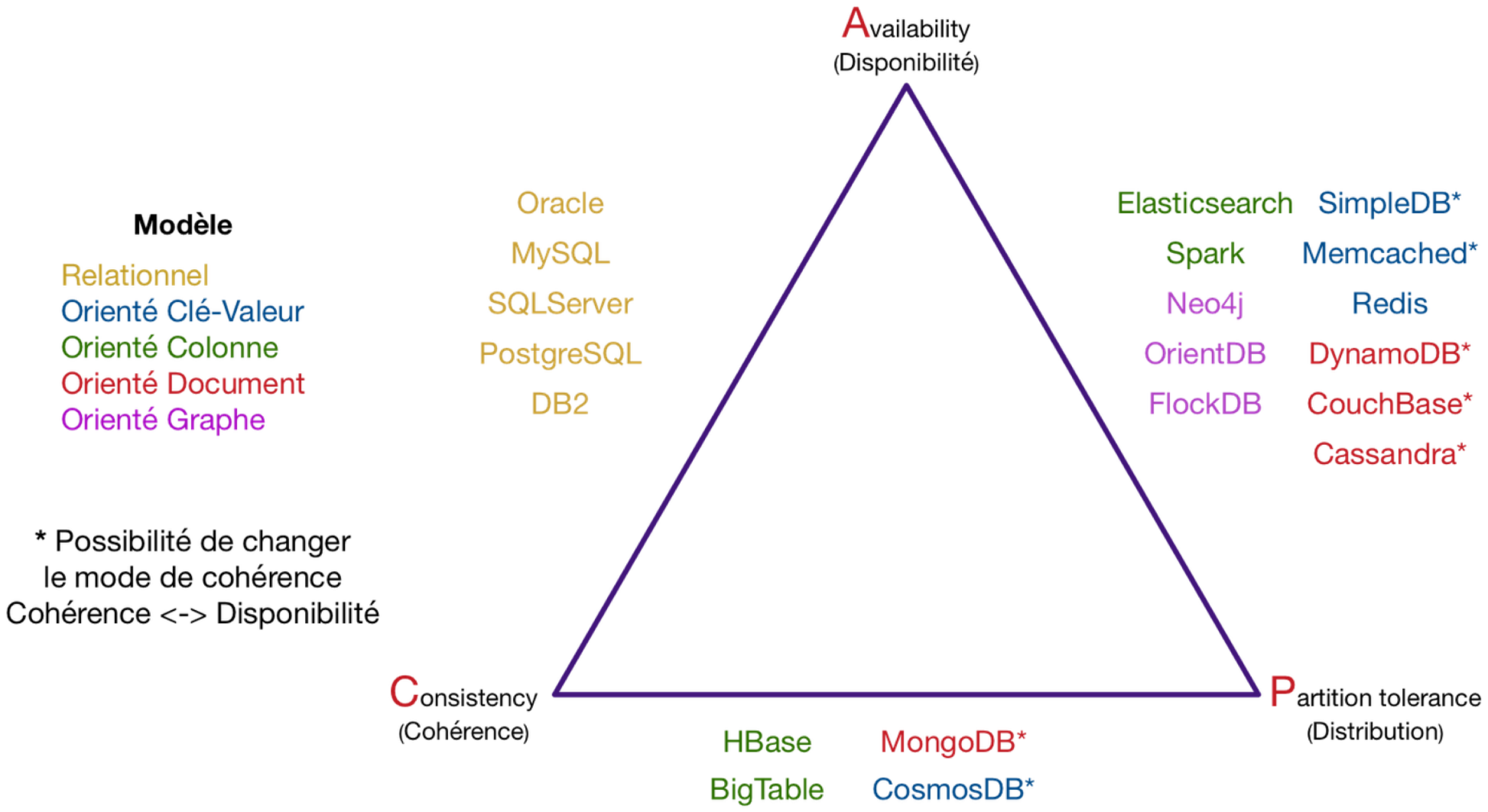
• La structure de données n'est pas importante. Évolutive et pas connue d'avance
• Gestion de beaucoup de données structurées, et non structurée.
• BASE : (contrairement à)
  • Basically Available : quelle que soit la charge de la base de données, le système garantie la disponibilités des données.
  • Soft-state : La base peut changer lors des mises à jour ou lors d'ajout/suppression de serveurs. La base NoSQL n'a pas à être cohérente à tout instant
  • Eventually consistent : À terme, la base atteindra un état cohérent

Théorème de Brewer

Théorème de Brewer dit "théorème de CAP, 2000 :
Indique qu'il est impossible, pour un système distribué, de garantir en même temps les trois
contraintes suivantes:

- **Cohérence (Consistency)**: Tous les noeuds du système voient les mêmes données au
même moment.
- **Disponibilité (Availability)** : Toutes les requêtes reçoivent une réponse.
- **Tolérance au partitionnement (Partition Tolerance)** : Aucune panne ne doit empêcher le
système de répondre correctement (sauf une coupure complète du réseau).

Théorème de Brewer



Triangle de CAP

Source : https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462471-maitrisez-le-theoreme-de-cap

Illustration : les collections (MangoDB) - 1

- Une collection est un ensemble de documents.
- C’est comme une table dans une base de données relationnelle.
- Les collections se trouvent dans une base de données

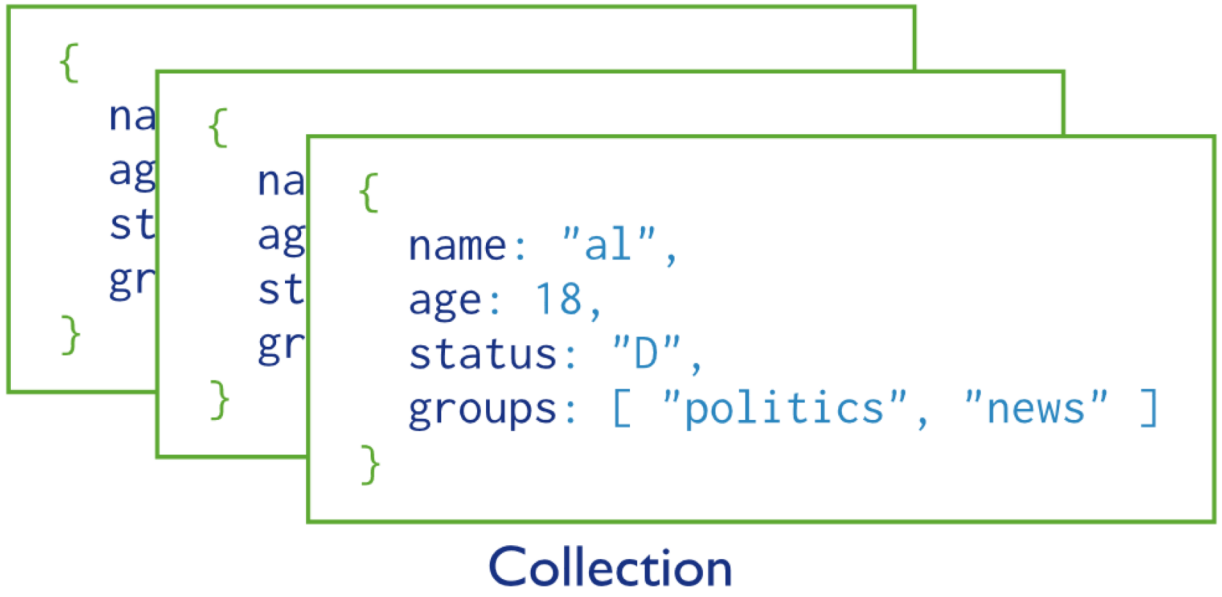


Illustration : les collections (MangoDB) - 2

db.Programmes.find**({"nom":"Patoche"});** va retourner tous les étudiants dont le nom est Patcohe.

db.Programmes.find**({"nom":"Patoche", "prenom":"Alain"});** va retourner les noms des étudiants dont le nom est Patoche et le prenom Alain.

db.Programme.find**({"nom":"Patoche"},{nom:1,prenom:1});** seuls les noms et les prénoms seront affichés

db.Programmes.find({ "etudiant.nom": "Yanick"}); on cherche dans un document imbriqué

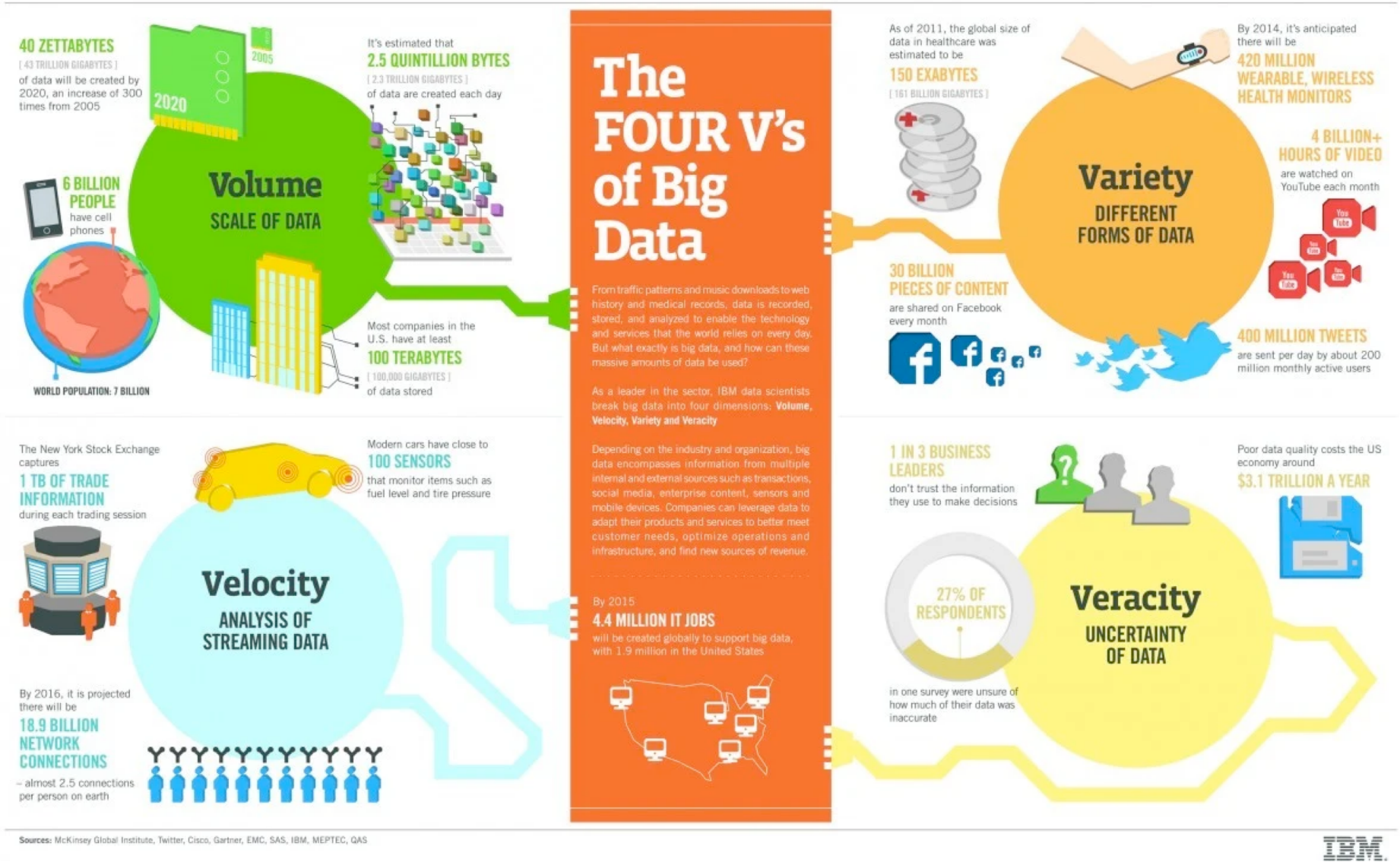
Le document Programmes, contient le document Etudiant

Illustration : les collections (MangoDB) - 3

```
db.contacts.insertOne(
{
   "nom":"Poitras",
   "dep":
            {"code": 420, nom:"info"},
   "cours":"kba"
  }
   );
```

Le champ dep a lui-même deux champs.

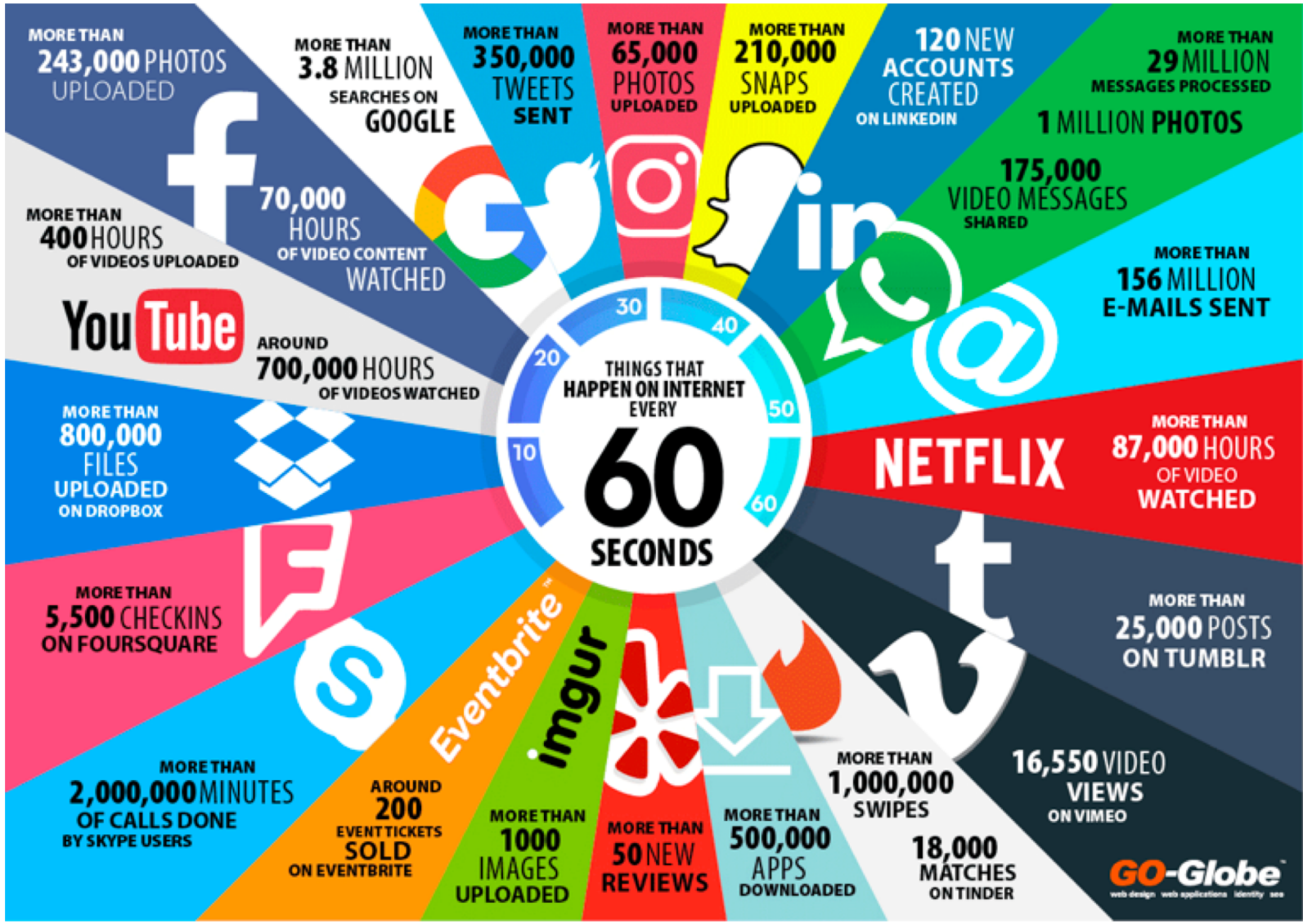
Big data: les 4 V



Source : https://blog.avanci.fr/le-big-data-et-les-4-v-de-la-donnee

Les géants du Web

- Chacun a son outil
- Les outils évoluent aussi en fonction des algorithmes, notamment d'IA



Les géants du Web : exemple

Chaque jour, Facebook gère :

- 2,5 milliards d'objets
- 500 To de nouvelles données
- 2,7 milliards de « Like »
- 300 millions de photos intégrées
- …