# 3BIN - Synthèse Big Data

**Intro**

+ datas, smartphones, réseaux sociaux -> traitement des données générées (machine learning, …).

Social business = impliquer parties prenantes dans le produit (crowdfunding, feedbacks,…).

Consom'acteur => influencés/influenceurs. Klout pour analyser poids sociétal des personnes.

Stratégies commerciale, basée sur réseaux sociaux, e-commerce, m-commerce, IoT… combiner datas pour prévoir intérêt consommateurs. Utilisation de clouds publics/privés. Objectif actuel : fluidifier transfert information. Existance open data, problème anonymisation.

**3V**

Volume _ + on a d'infos + on est proche de la réalité

Vélocité _ + les données sont fraiches + elles ont de potentiel

Variété _ + les données sont diverses + on peut les croiser pour atteindre un objectif

Véracité _ + qualité d'infos en évitant les datas influencées (formulaires etc)

Valeur _ + les données sont filtrées

**NoSQL**

sql = axé scalabilité verticale. Not Only Sql :

clef-valeur utilisant la ram pour petites datas (Redis, système d'authentification).

documents stockant large collections. sharding pour découper un peu si besoin. Pas opti pour relations, intérêt dénormalisation. MongoDb.

colonnes via requêtes simplistes et index. gain de place, prévu pour scalab horiz. mémoire prise que si colonne remplie.

graphes via noeuds et arcs, type réseaux sociaux. Neo4j, Orient DB.

**Hadoop**

RAID 0 _ diviser document, meilleure lecture/écriture mais double risque physique.

RAID 1 _ double écriture, réplication parfaite, "backup-auto"

RAID 5 _ raid0 + bloc de parité pour vérifier intégrité. 3DD au lieu de 2 mais + safe. HDFS = Hadoop Distributed File System.

Lie nom fichier à blocs. Permissions et répertoires comme sous unix, mais non lié au noyau. = portable. syst!me virtuel, app externe pour le monter. peut donc être distribué sur +eurs serveurs. grande taille blocs pour économiser place, que grands fichiers. réplication blocs prévue.

Namenode = maître. Connaît état, load balancer. réplication namenode pour éviter single point of failure. Etat chargé au démarrage.

Datanome = travailleurs. Peuvent dialoguer entre eux, évite surcharge namenode. Connectés au namenode au démarrage.

Checkpointing = mise à jour fichier -> rassembler log edition et fichier du fs pour créer le nouveau fichier. Code exemple dans synthèse christopher p12-13.

MapReduce = framework proche données pour séparer travail. jobs, mapReduce, params d'exec.

Map _ clef-valeur les inputs. Combine _ Filtre similaires. Reduce _ Rassemble listes.

**Outils**

Warehouse _ Tout rangé proprement, casté et formaté dans une DB

DataLake _ Tout mis en vrac avec système tags pour accès pertinent, sans cast

**Entreprise**

Modèle disruptif _ big data décisionnel. Approche réactive. Hadoop, mapReduce, machine learning et/ou visualisation.

Modèle évolutif _ post-traitement données. data warehouse/data lake. revente données aux intéressés.

Modèle hybride _ intégration big data / BI. Flux avec warehouses. ! incohérences sources.

Compétences = données internes ou externes, extraction de valeur, gestion idées pour transformation en service/produit.

Changements oranisationnels présents (prévision tendances, décisions + objectives)

Externalisation des données (ventes, échanges, …)