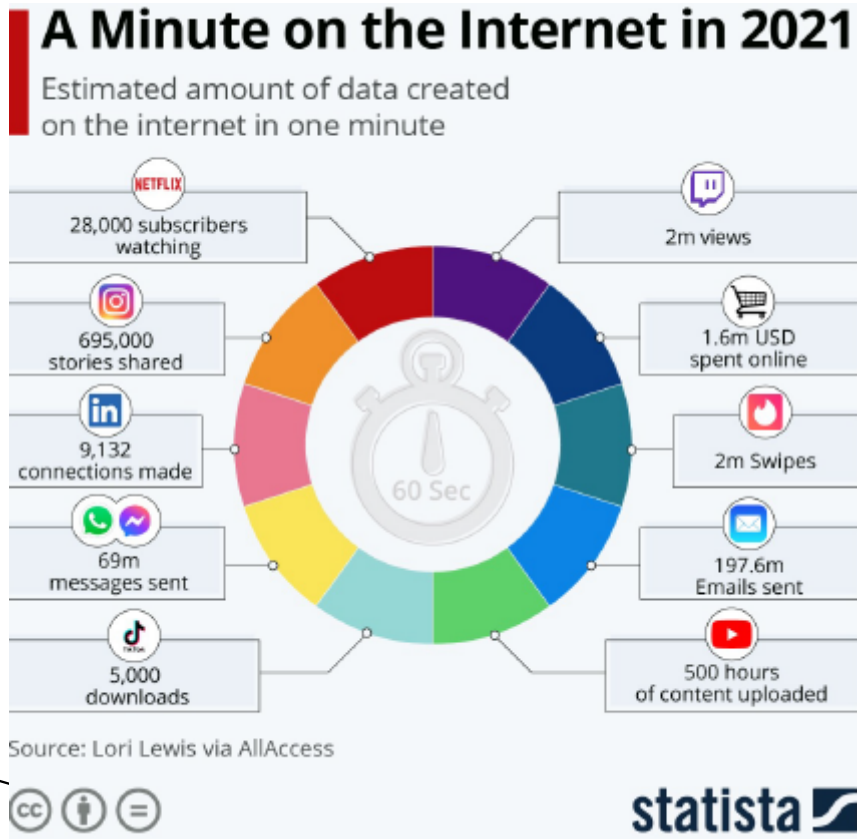
Big Data

Basisprincipes

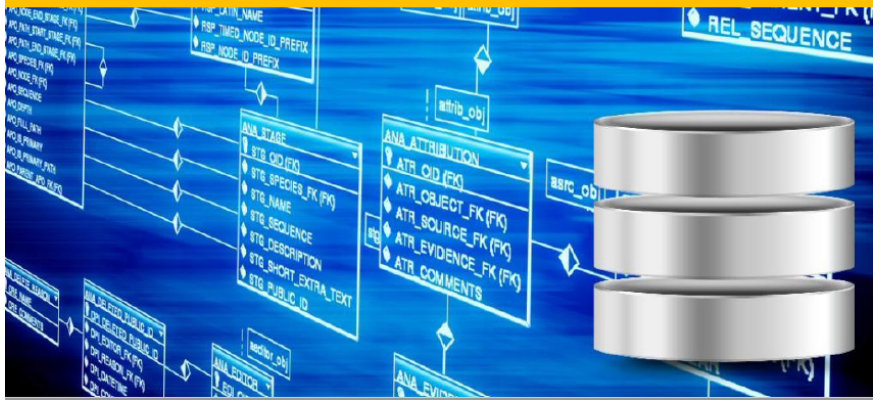
DE HOGESCHOOL MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

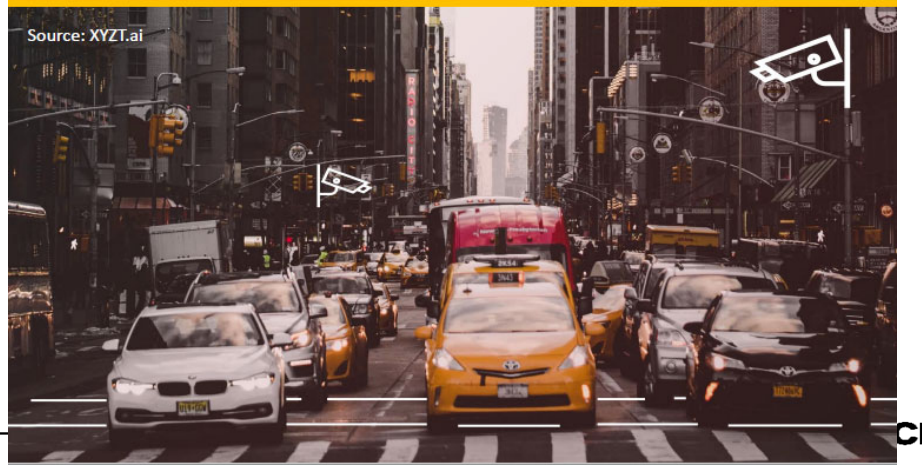
Inleiding



Applications

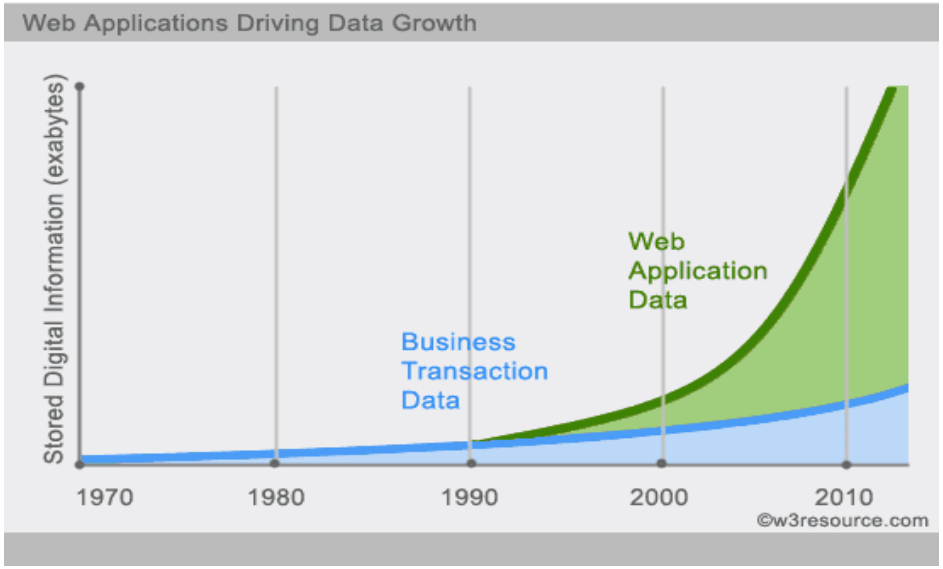


Sensors



CHOOL PXL

Inleiding



- Relationele databanken: Gegevens opslaan door bedrijven
- Massa andere gegevens niet opgeslagen in relationele databank: tweets, facebook, weblogs, feeds, RFID-scans, sensordata, clickstreamdata,…
- Nood aan:
  - Infrastructuur
  - nieuwe programmeeromgeving
  - nieuwe dataomgeving

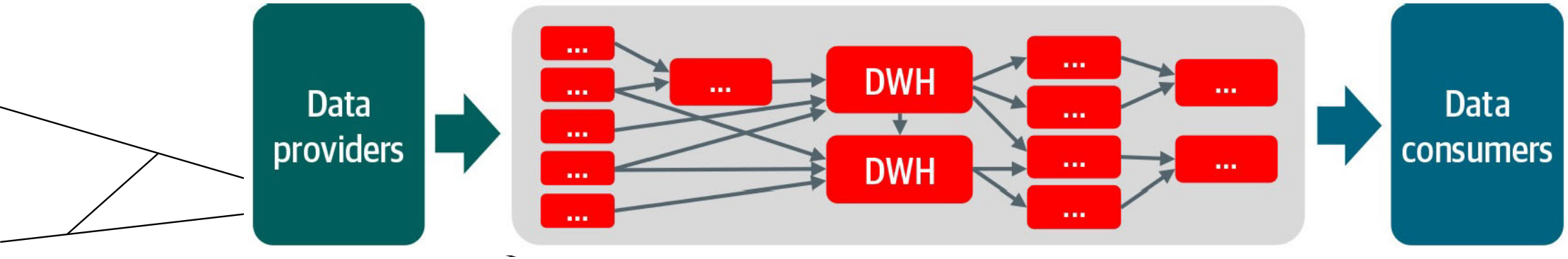
Terminologie

Database: archief voor dataopslag

- Opgeslagen gegevens als zodanig
- Wijze waarop gegevens zijn opgeslagen
- Software waarmee databases worden aangemaakt en benaderd

Datawarehouse

- Gegevensverzameling voor snelle ad-hoc vragen zonder belasting bron
- Nooit rechtstreeks gegevens toegevoegd/gewijzigd/verwijderd
- Gegevens worden gebruikt voor BI-doeleinden
- Voorbeeld controle CV-ketels



Terminologie

Datamining

- Gericht zoeken naar (statistische) verbanden tussen gegevensverzamelingen → patronen:
  - Business Intelligence – BI
  - Artificiële Intelligentie – AI
- Betekenis en inhoud (context) informatie cruciaal
- Snelheid waarmee bruikbare resultaten worden bekomen is in realtime-toepassingen zeer belangrijk bv. monitoren en bijsturen van bedrijfsprocessen
- Doel? Wetenschappelijk, journalistiek, commercieel gebruik

  Vb verband tussen leeftijd klant en type shampoo

Big Data - vroeger

‘Big Data’: al in de jaren ’50

- **Aanvang:** Analyses via wiskunde en/of statistiek(manueel)
- **Later:** gebruik van applicaties o.a. spreadsheets en database-toepassingen(o.a.Access)
- **Doel:** beslissingen nemen voor de toekomst => BI en AI

Big Data – nu hype

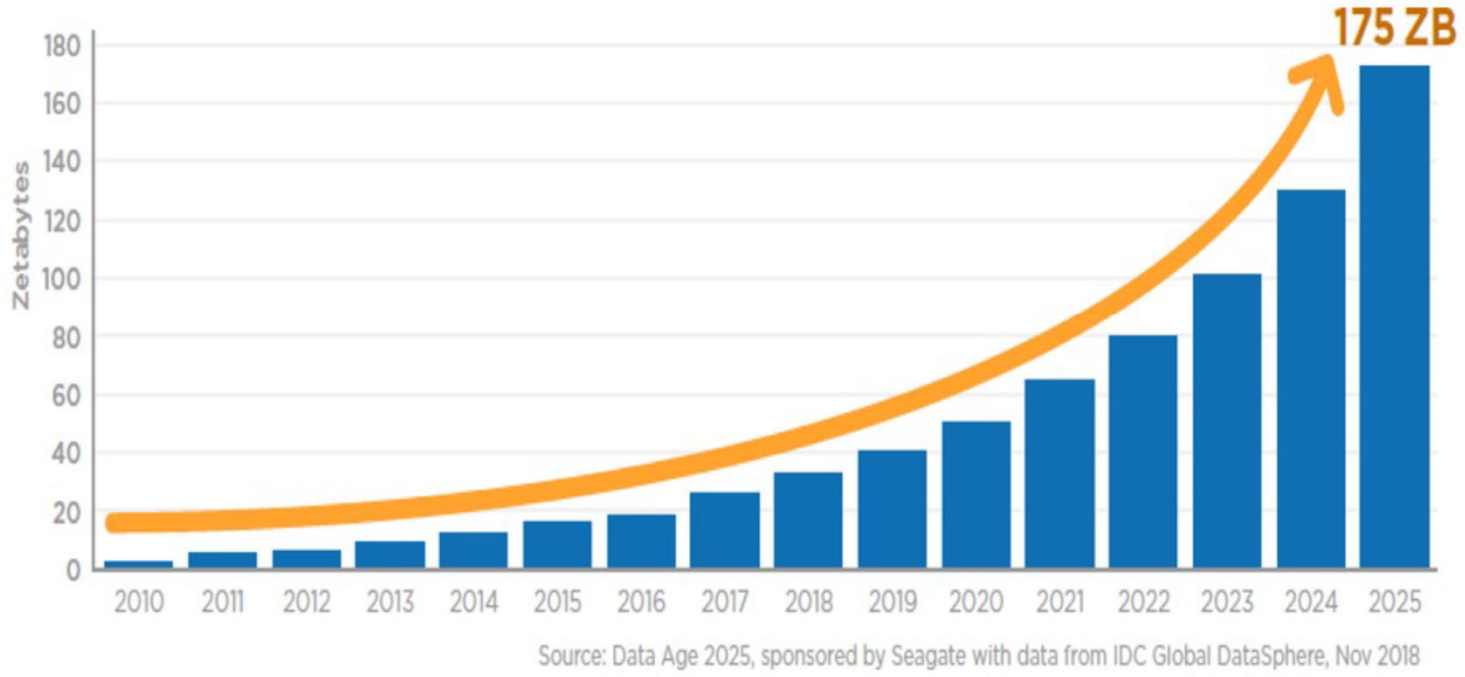
- Voedingsbodem:
  - Hardware mogelijkheden, server
  - Goedkopere en ruimere opslag
  - Mogelijkheden van opensource software
  - Beschikbaarheid massa's gegeven

- Toepassingen:
  - Marketing
  - Politieonderzoek –en opsporing (fraude, cybercrime)
  - Analyses datalekken (bv WikiLeaks, Luxleaks, Panama Papers)
  - Onderzoek gezondheidssector (ziektes, erfelijkheid)
  - Industrie (bv technologie veiligheid auto's)
  - Gaming
  - Bedrijfsbeslissingen

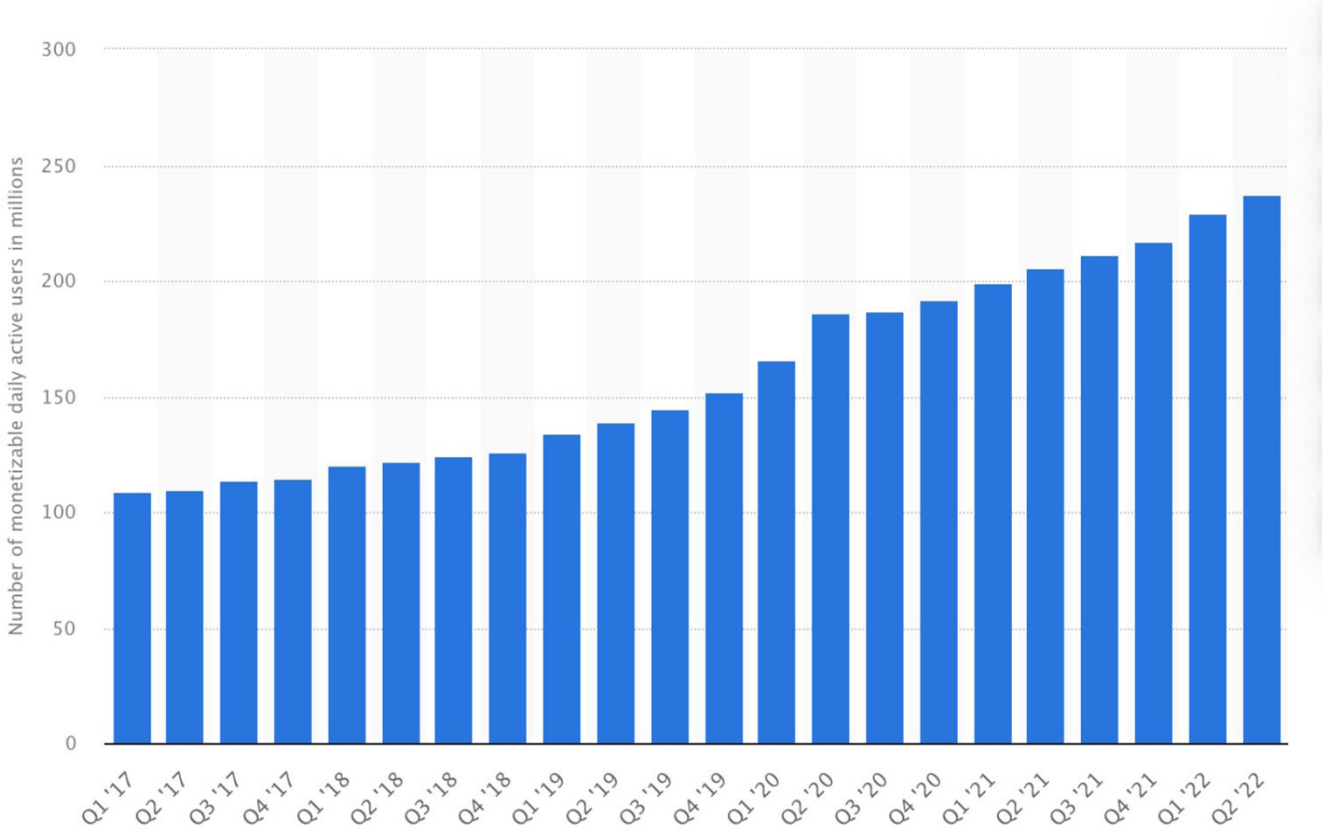
Term Big Data

- Honderden terabytes
- ‘Klassieke’ databank kan gegevens niet aan, alternatief nodig voor niet-relationele gegevens
- 5 V’s:
  - Volume
  - Velocity
  - Variety
  - Veracity
  - Value

Volume – de size of Big Data

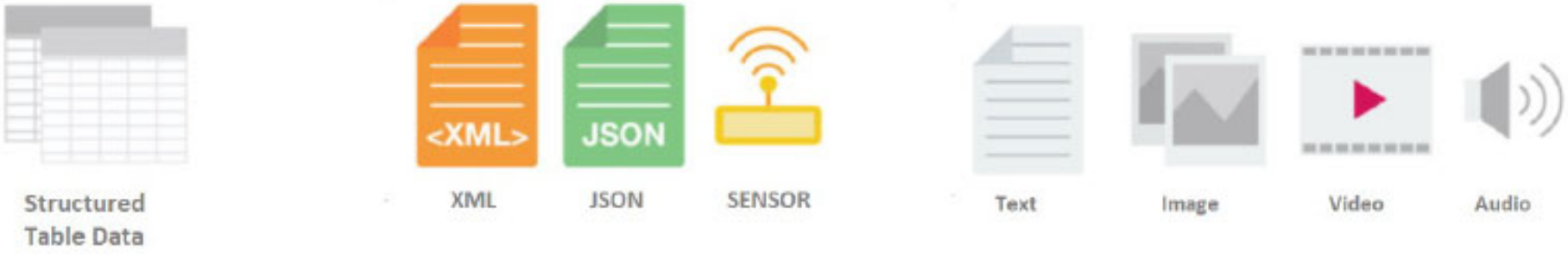


Velocity - the speed at which data is growing

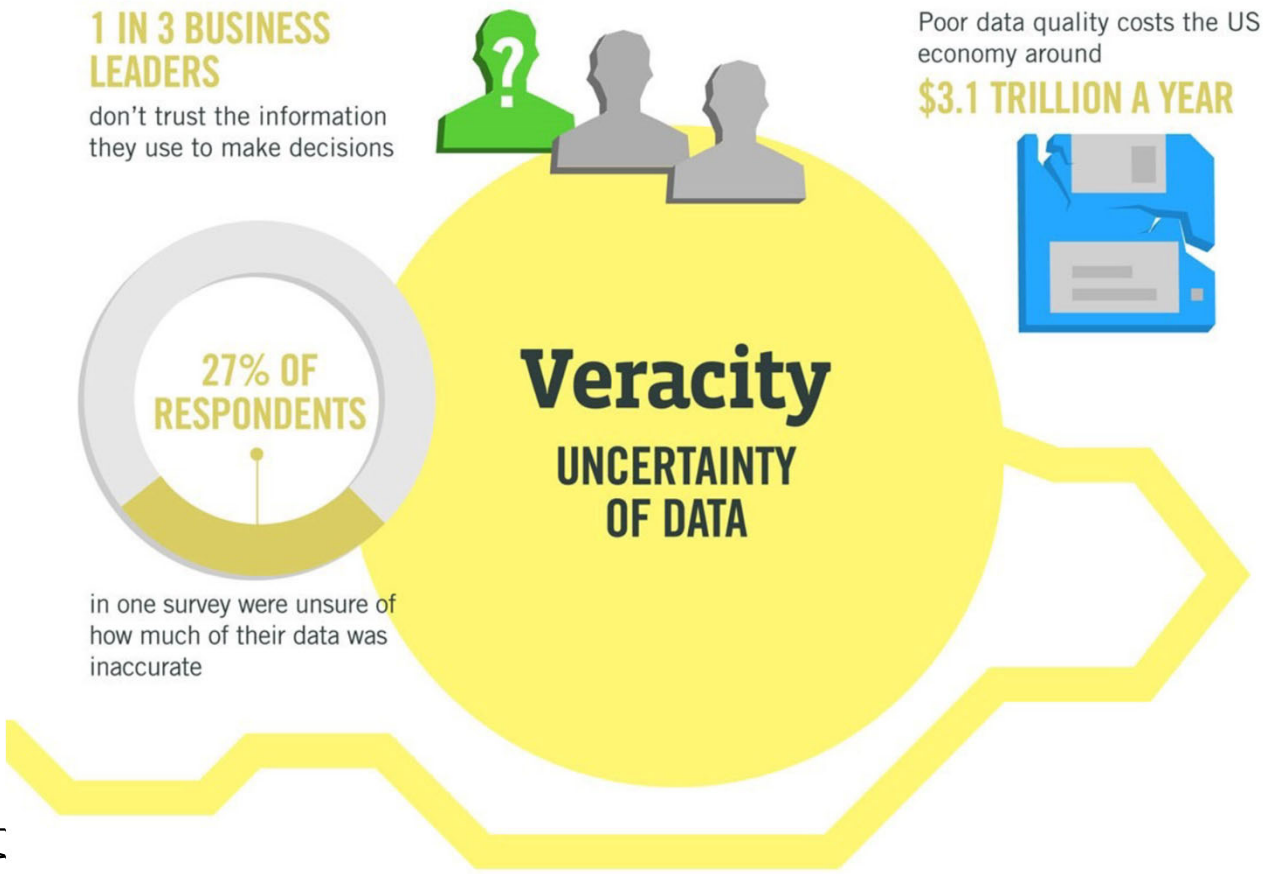


Number of montly active twitter users (in millions) - *statista*

Variety - the different types of data



Veracity – Accuracy or truthfulness of data?



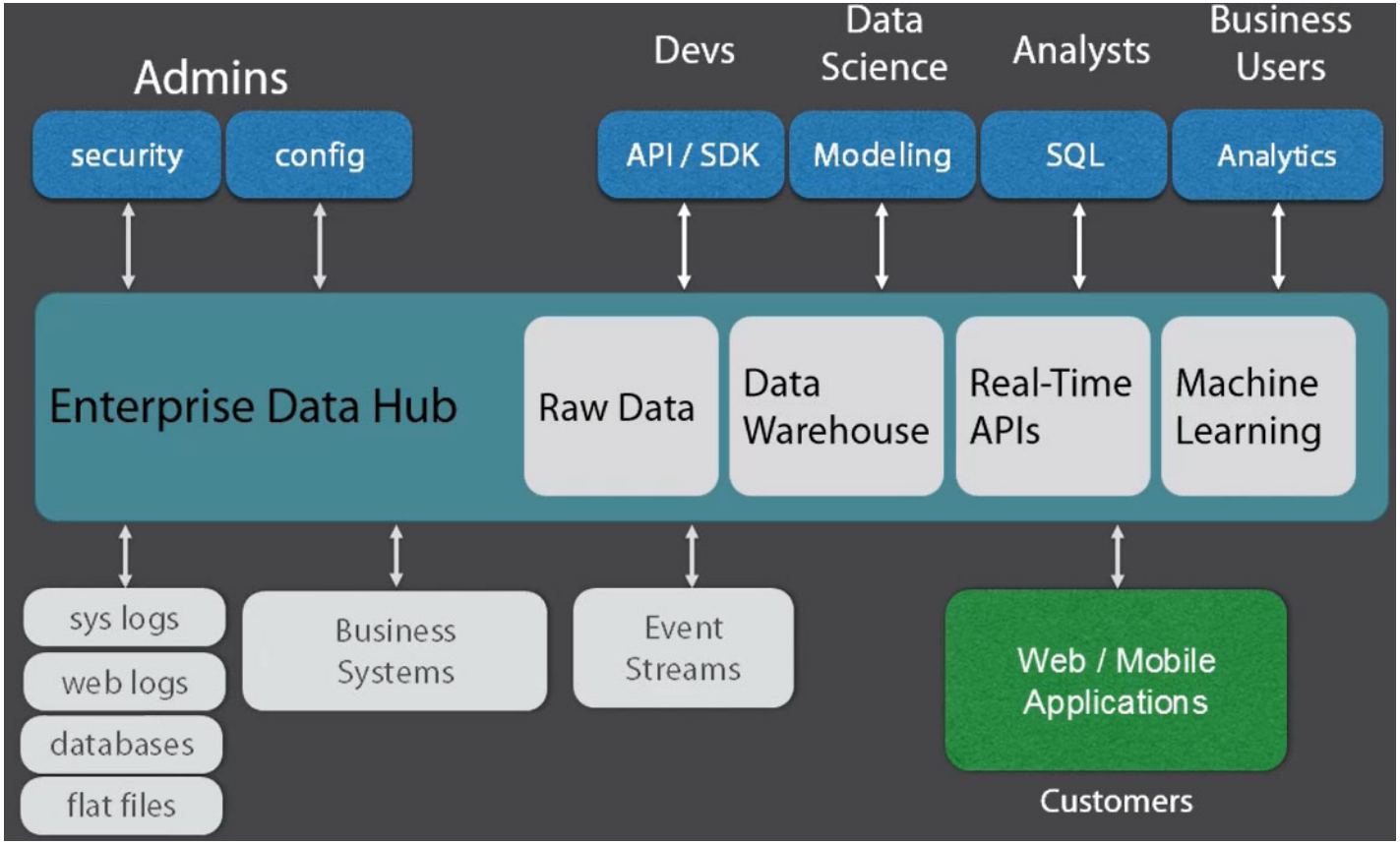
Value – How useful is the data?

Hoe werkt Big Data?

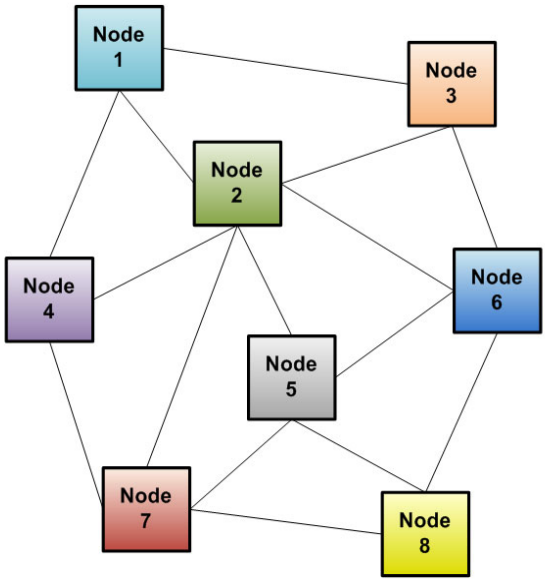
- Architectuur
- Distributed System met nodes
- CAP-theorema

Architectuur

https://app.pluralsight.com/player?author=ben-sullins&name=data-analytics-hands-on-m9&mode=live&clip=3&course=data-analytics-hands-on



Distributed system



- Big data → grote hoeveelheden
  → geen structuur

  Gevolg: verwerkingstijd schaalt met hvh informatie
- Hoe verwerking versnellen?
  → snellere server
  → meer servers
  → optimalisering programma's
- Distributed system
  mainframes, workstations, PC's communiceren via netwerk

https://app.pluralsight.com/player?author=ben-sullins&name=data-analytics-hands-on-m9&mode=live&clip=4&course=data-analytics-hands-on

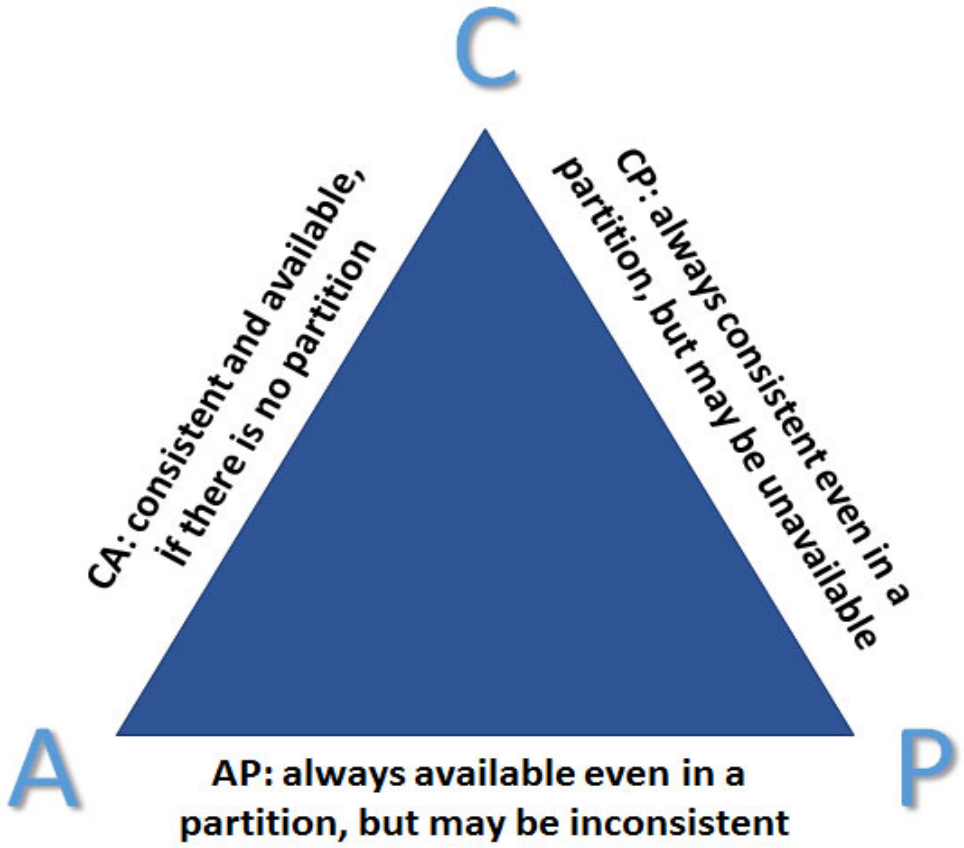
Distributed datastores

- Datastores in een gedistribueerd systeem
  → RDBMS komen hiervoor niet in aanmerking, dus NoSQL

- Voordelen dergelijk systeem:
  - Reliability
  - Scalability
  - Sharing resources
  - Flexibility
  - Speed
  - Open system
  - Performance

CAP-theorema

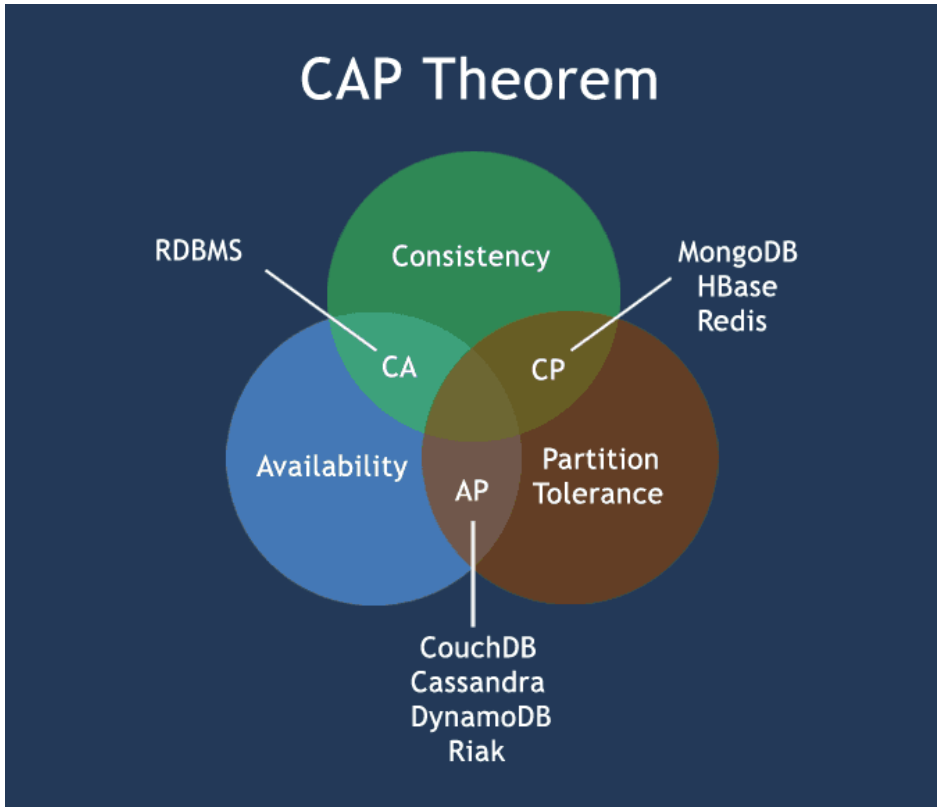
- Consistency
- Availability
- Partition tolerance



Belang van CAP-stelling: https://app.pluralsight.com/player?author=ben-sullins&name=data-analytics-hands-on-m9&mode=live&clip=5&course=data-analytics-hands-on

CAP-stelling – soorten DB

CA: consistentie en beschikbaar
CP: consistentie, alle data niet direct bereikbaar
AP: beschikbaar, niet altijd volledig



Database principes

- ACID
- BASE

Database principes - ACID

- Atomic:
  Elke transactie slaagt volledig, inclusief deelacties, of niet
- Consistent (ACID Consistency):
  Gegevens mogen niet tegenstrijdig worden.
  Referentiële integriteit.
- Isolated:
  Elke transactie wordt los van andere transactie uitgevoerd.
- Durable:
  Transactie is permanent/onomkeerbaar.

Database principes - BASE

- Basic Availability
  Beschikbaarheid van data, zelfs met tijdelijke fouten (spreiding gegevens over meerdere opslagsystemen)
- Soft State
  Consistent zijn ligt bij ontwikkelaar, niet bij databank.
- Eventual Consistency
  Uiteindelijk komen tot consistentie, niet meteen - staat haaks op ACID.

NoSQL DBMS

- niet-relationeel databasemanagement systeem
- distributed data stores met big data
- geen vaste structuren
- vermijdt join-operaties

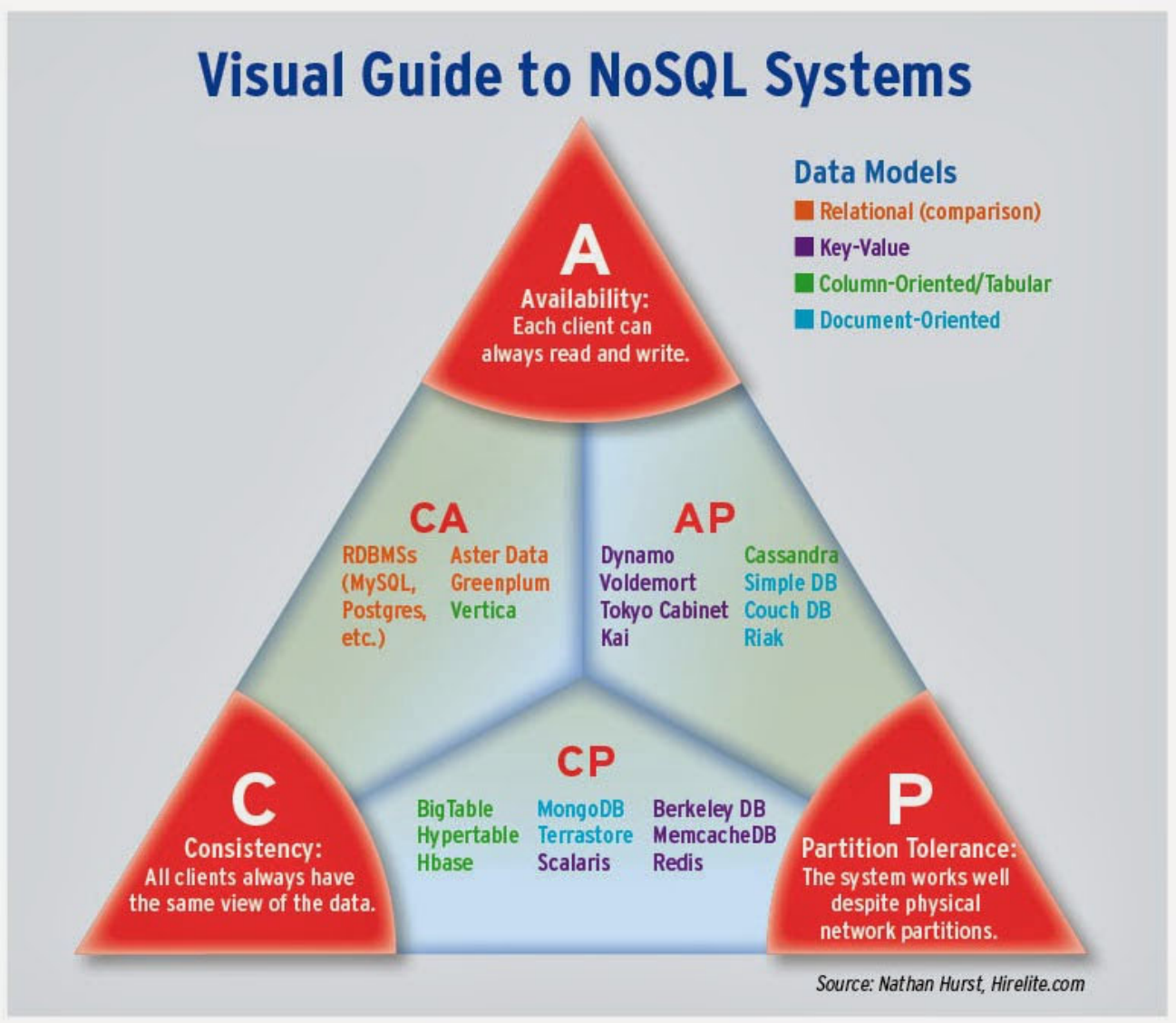
RDBMS ⇔ NoSQL

RDBMS	NoSQL
Gestructureerde data	Not Only SQL – ook ongestructureerde data
SQL – structured query language	Geen standard query language
Data en relaties worden in aparte tabellen opgeslagen	Geen vooraf gedefinieerde structuur
DML – data manipulation language DDL – data definition language	Soms onvoorspelbare data
Altijd data consistency	Eventual consistency maar wel hoge performantie
ACID-transacties	BASE-transacties

Voordelen / nadelen NoSQL

Voordelen NoSQL	**Nadelen NoSQL**
Hoge scalability	Geen standaard
Distributed computing	Beperkte query mogelijkheden
Lagere kost	Eventual consistency is moeilijk programmeerbaar
Flexibiliteit in structuur van data	
Geen gecompliceerde relaties/joins	

NoSQL database types



Database type: Key-value stores

- Meest gebruikte datatype
- Kan vele TB aan gegevens aan
- Laten ongestructueerde gegevens toe
- Makkelijk uitbreidbaar
- Gegevens opgeslagen als hashtable
  elke key uniek, value kan string, JSON-object, BLOB-object,.. zijn
- Key-value pair kan bestaan uit naam gecombineerd met waarde
- Beperking: je kan enkel zoeken via key!

Vergelijking RDBMS - column-oriented store

RDBMS → tabel met rijen en kolommen

Facebook_Friends

Name	City	Age
Matt	Los Angeles	27
Dave	San Francisco	30
Tim	Oakland	33

Opslag data per rij

Matt	Los Angeles	27	Dave	San Francisco	30	Tim	Oakland	33

Toevoegen van een rij:

Matt	Los Angeles	27	Dave	San Francisco	30	Tim	Oakland	33	Jen	Vancouver	30

Vergelijking RDBMS - column-oriented store

RDBMS

Opslag op verschillende harde schijven:

Disk 1		
Name	City	Age
Matt	Los Angeles	27

Disk 2		
Name	City	Age
Dave	San Francisco	30

Disk 3		
Name	City	Age
Tim	Oakland	33

Berekening gemiddelde leeftijd:

- Full Table Scan
- Verspreid over meerdere harde schijven

→ traag

Vergelijking RDBMS - column-oriented store

Column-oriented database

Facebook_Friends

Name	City	Age
Matt	Los Angeles	27
Dave	San Francisco	30
Tim	Oakland	33

Opslag data per kolom

Matt	Dave	Tim	Los Angeles	San Francisco	Oakland	27	30	33

Toevoegen van data:

Matt	Dave	Tim	Jen	Los Angeles	San Francisco	Oakland	Vancouver	27	30	33	30

Vergelijking RDBMS - column-oriented store

Column-oriented database

Opslag op verschillende harde schijven:

Disk 1		
Name		
Matt	Dave	Tim

Disk 2		
City		
Los Angeles	San Francisco	Oakland

Disk 3		
Age		
27	30	33

Berekening gemiddelde leeftijd:

- Geen overtollige data inladen in geheugen
- Data enkel op 1 harde schijven

→ **Veel performanter**

Database type: Column-oriented stores

- Werken met kolommen
- Slaan values kolom aaneengesloten op
- Kolomgegevens in specifieke files (harde schijven)
- Keys verwijzen naar verschillende kolommen
- Queries mogelijk
- Data in kolomfile → zelfde type → gemakkelijke compressie
- Hoge performantie bij gewone queries en groepsqueries → zeer geschikt voor BI en CRM
- Vb: Hbase Cassandra, SimpleDB, SAP HANA

Database type: Documented-oriented stores

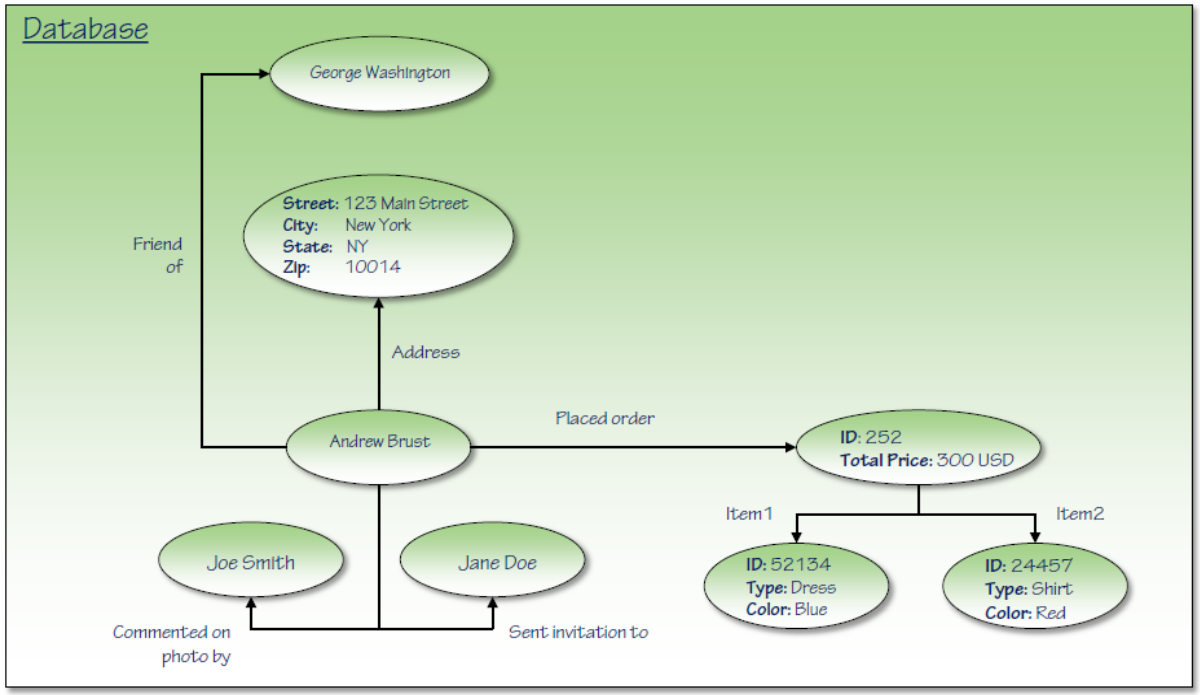
- Verzameling van documenten
- Data in documenten, key geeft toegang
- Niet noodzakelijk vaste structuur
- Documents → collections: groepering data
  - verschillende key-value pairs
  - geneste documenten
- JSON objecten
- Vanuit applicaties verwijzing via URI's
- Queries mogelijk

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

Database type: Graph stores

- Slaan data op in grafiek
- Presentatie zeer toegankelijk
- Verzameling nodes en edges
- Indexen voor opzoeking
- Vb: OrientDB, Neo4J, Apache Giraph

Graph Databases



Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m1-tech-breakdown&clip=4&mode=live&start=77.868176&noteid=fb45d5c9-4e66-4d99-8b73-352b6c0de7e6

NoSQL, relational, or both?

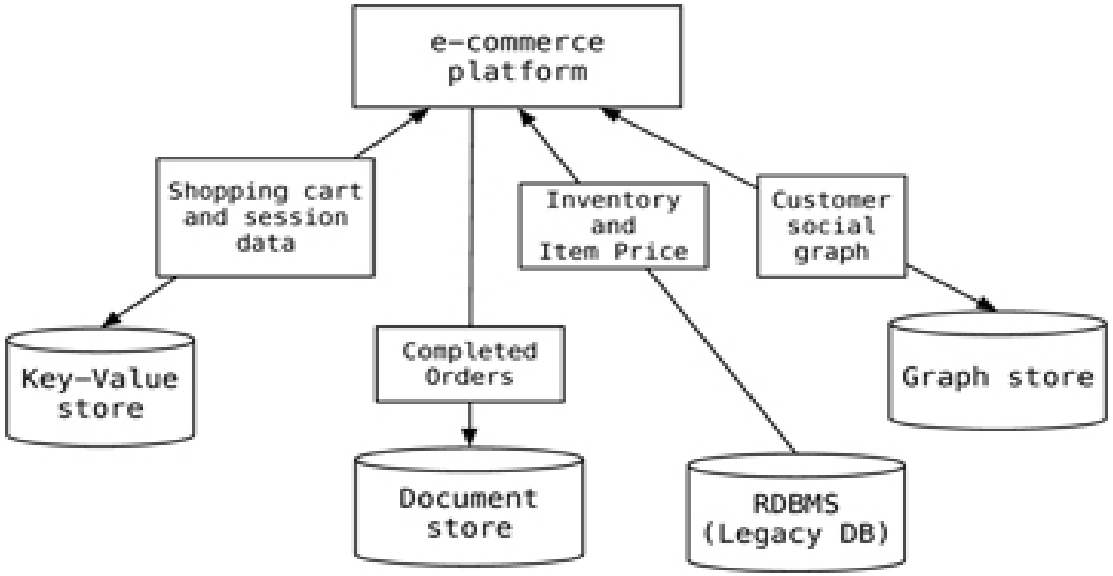


Figure 13.3. *Example implementation of polyglot persistence*

Zie Pluralsight: https://app.pluralsight.com/player?course=understanding-nosql&author=andrew-brust&name=understanding-nosql-m5-both&clip=3&mode=live&start=1.257044&noteid=35a1c93e-d59d-4be8-b130-a9928583f170

Recommendations

- **Large, public, content-centric properties: NoSQL**
- **Internal, LOB supporting business operations: relational**
- **Investment in RDBMS licenses, infrastructure, skills:**
  - Relational
  - Use both (application-dependent)
  - Use hybrid approaches
- **Productivity**
  - Do cost-benefit analysis
    - How much extra dev time/$$?
    - What is cost of less scalable system?
- **It will be tempting to use one for the other**
  - And it very well may work, but that doesn't make it right