

Base de données

Aperçu des base de données en 2018

Joël Cavat

2018

joel.cavat@hesge.ch

Thèmes abordés dans ce chapitre :

- Context historique
- Les différents paradigmes
- Théorème de CAP
- Cas d'utilisation
- Comment choisir sa base de données ?

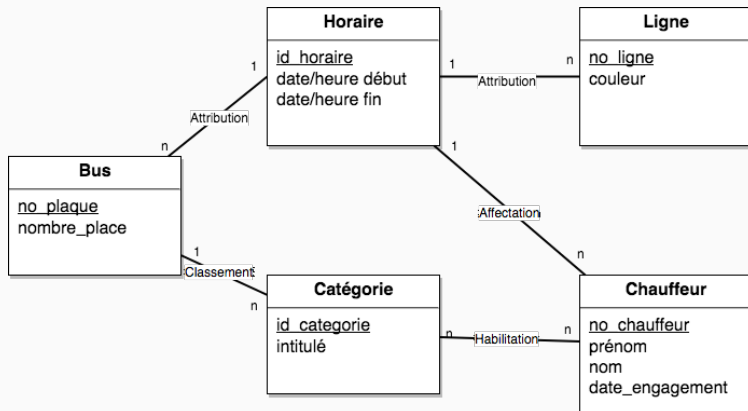
- 1970, Edgar Frank Codd, description du **modèle relationnel** et de la normalisation des données
- 1974, Don Chamberlin, langage SQL pour l'exploitation de bd relationnelles
- **1980**, essor des base de données relationnelles
- 1990, naissance des bd analytiques et objet-relationnel
- 1990 à nos jours, dominance des bd données relationnelles
- 2003, Eric Evans, domain driven design
- 2010, émergence des bd **NoSQL**, rupture de l'approche relationnelle
- 2014, microservices

Définition

Une base de données normalisée est modélisée de manière à éviter des redondances, des incohérences et des anomalies

Base de données

Exemple : Modèle conceptuel (entité-association)

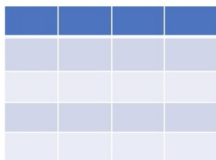


Les différents paradigmes

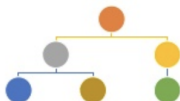
Les différents paradigmes

Databases

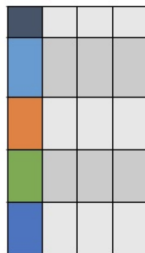
Relational



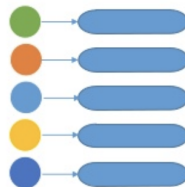
Document



Column-Family



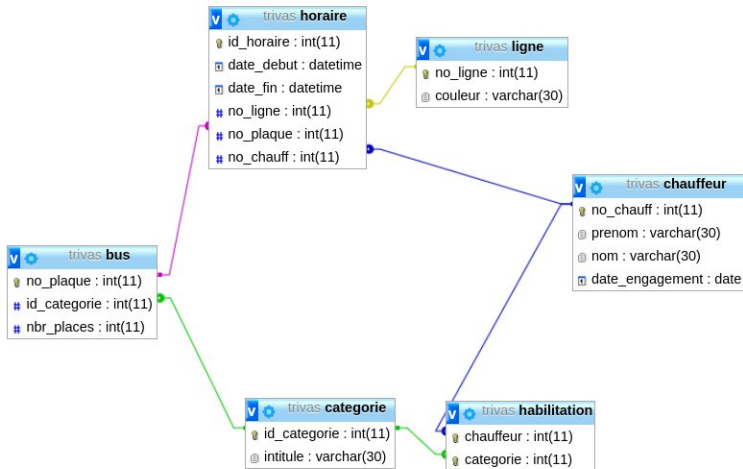
Key-Value



Graph

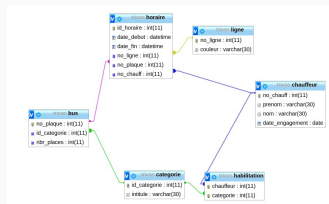


Base de données relationnelles



Caractéristiques :

- Basé sur les relations en théorie des ensembles et l'algèbre relationnelle
- Données **normalisées** et structurées
- Eviter la redondance
- Eviter les anomalies
- **Contraintes** d'intégrités
- Transactions ACID
- Déploiement **centralisé**
- Scalabilité **verticale** pour plus de performance



NoSQL signifie **Not Only SQL**

Motivations

Offrir une meilleure tolérance aux partitionnements et de meilleures performances en proposant une rupture du modèle relationnel classique

Caractéristiques (en règle générale) :

- Données non ou semi-structurées/dénormalisées
- Données redondantes
- Tolérance aux pannes grâce à la replication
- Sensibles aux anomalies
- Scalabilité horizontale pour plus de performance (sharding)
- Performance en lecture/écriture (plus de jointure)
- Déploiement décentralisé ou distribué
- Pas de transactions ACID -> Eventually Consistent
- API simples

Bases de données NoSQL

Popularité :

- Prototyper très vite (Agilité)
- BigData, format des données souvent non structuré
- Scalabilité
- Compétences moindres
- Tolérance aux changements



Image : <https://media.licdn.com>

Bases de données NoSQL

Clés/Valeurs

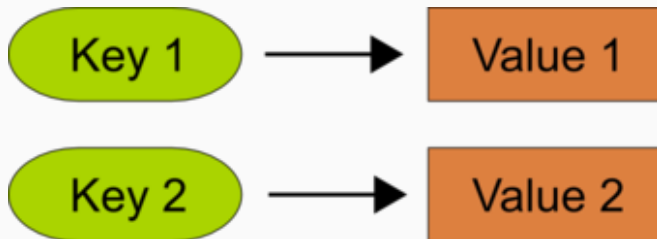
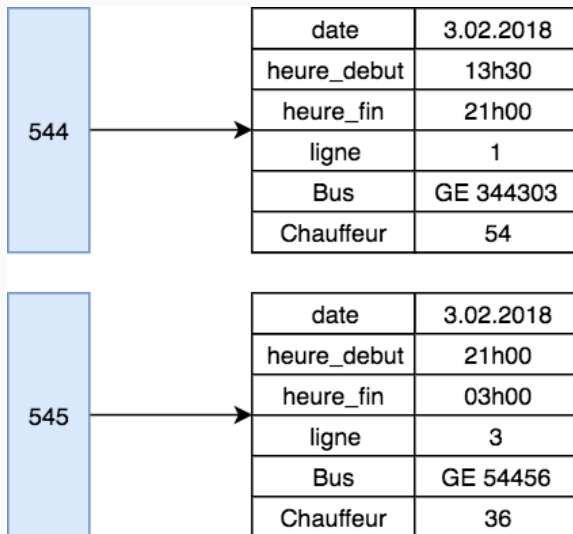


Image : <https://cdn-images-1.medium.com>

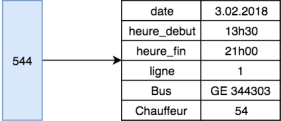
Bases de données NoSQL

Clés/Valeurs



Caractéristiques

- Dictionnaire clé/valeurs
- Simplicité
- Accès rapide aux valeurs
- Pas d'indexation sur les valeurs
- Aucune opération d'agrégations sur les valeurs
- *sharding* + *réplication*



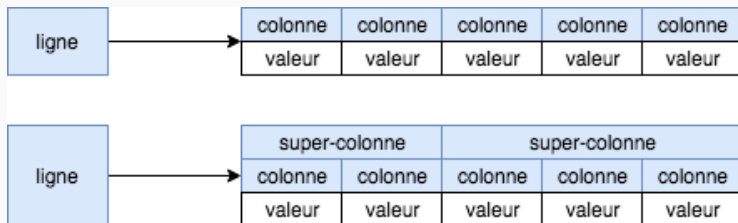
date	3.02.2018
heure_debut	13h30
heure_fin	21h00
ligne	1
Bus	GE 344303
Chauffeur	54

Exemple

- Riak/Redis
- Petits services persistants, Logs/Événements, Message broker
- Vodafone, Atlassian, Trip Advisor, Docker

Bases de données NoSQL

Colonnes



Bases de données NoSQL

Colonnes

ChauffeurInfo			Habilitation	Horaires		
Id	Nom	Email	TrolleyBus	544	654	...
54	Alfred	alfred@tpg.ch	Car Postal	3.02.2018	5.02.2018	
			...			
ChauffeurInfo			Habilitation	Horaires		
Id		Nom	Bus de manège	549	667	...
55	Ludwig	TrolleyBus	4.02.2018	5.02.2018		
		...				
...						

ChauffeurId	Nom	Prénom	AVS	Mail	Adresse
54	Dupont	Alfred	123.32.XX	alfred@tpg.ch	Rue Schaub ...
55	Racloz	Ludwig	123.45.XX		Route de ...
...					

Caractéristiques

- Stockage
par colonnes dynamiques
- Données **dénormalisées**
- Indexation + agrégation
- Très grand volume de données
- Déploiement
décentralisé ou réparti
- **sharding** + **réplication**

ChauffeurInfo			Habilitation		Horaires	
Id	Nom	Email	Trolley/Bus	544	654	
54	Alfred	alfred@tpg.ch	Cer Postal			
			...	3.02.2018	5.02.2018	
			...			
ChauffeurInfo			Habilitation		Horaires	
Id	Nom		Bus de manège	549	657	
55	Ludwig		Trolley/Bus			
			...	4.02.2018	5.02.2018	
			...			
...						

ChauffeurId	Nom	Prénom	AVS	Mail	Adresse
54	Dupont	Alfred	123 32 XX	alfred@bg.ch	Rue Scheub ...
55	Radoz	Ludwig	123 45 XX		Route de ...
...					

Exemple

- Cassandra, HBase, Big-Table
- Catalogues, IOT, Logs/E-vénements temps réel
- eBay, Netflix, Sony, UBS

Bases de données NoSQL

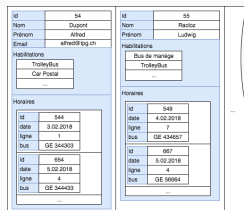
Documents

Id	54								
Nom	Dupont								
Prénom	Alfred								
Email	alfred@tpg.ch								
Habiletations									
TrolleyBus									
Car Postal									
...									
Horaires									
<table><tr><td>Id</td><td>544</td></tr><tr><td>date</td><td>3.02.2018</td></tr><tr><td>ligne</td><td>1</td></tr><tr><td>bus</td><td>GE 344303</td></tr></table>		Id	544	date	3.02.2018	ligne	1	bus	GE 344303
Id	544								
date	3.02.2018								
ligne	1								
bus	GE 344303								
<table><tr><td>Id</td><td>654</td></tr><tr><td>date</td><td>5.02.2018</td></tr><tr><td>ligne</td><td>4</td></tr><tr><td>bus</td><td>GE 344433</td></tr></table>		Id	654	date	5.02.2018	ligne	4	bus	GE 344433
Id	654								
date	5.02.2018								
ligne	4								
bus	GE 344433								
...									

Id	55								
Nom	Racloz								
Prénom	Ludwig								
Habiletations									
Bus de manège									
TrolleyBus									
...									
Horaires									
<table><tr><td>Id</td><td>549</td></tr><tr><td>date</td><td>4.02.2018</td></tr><tr><td>ligne</td><td>7</td></tr><tr><td>bus</td><td>GE 434657</td></tr></table>		Id	549	date	4.02.2018	ligne	7	bus	GE 434657
Id	549								
date	4.02.2018								
ligne	7								
bus	GE 434657								
<table><tr><td>Id</td><td>667</td></tr><tr><td>date</td><td>5.02.2018</td></tr><tr><td>ligne</td><td>4</td></tr><tr><td>bus</td><td>GE 56664</td></tr></table>		Id	667	date	5.02.2018	ligne	4	bus	GE 56664
Id	667								
date	5.02.2018								
ligne	4								
bus	GE 56664								
...									

Caractéristiques

- Stockage de documents non uniformes
- Enregistrements peuvent être imbriqués
- Indexation + agrégation
- Données **dénormalisées**
- Très grand volume de données
- **sharding** + **réplication**
- Déploiement décentralisé

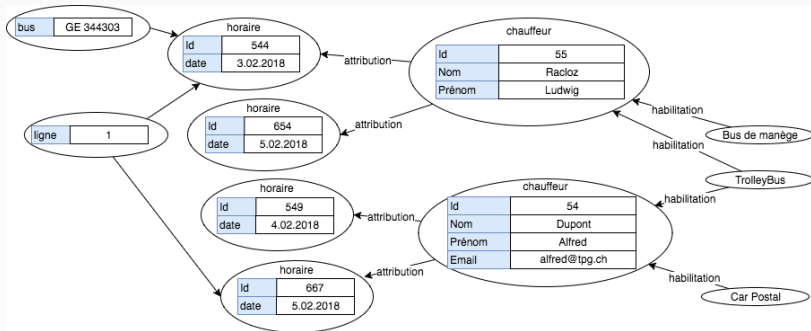


Exemple

- MongoDB, CouchDB
- Catalogues, IOT, statistiques
- Cern, Adobe, Cisco, eBay, EA Games, Facebook, PayPal

Bases de données NoSQL

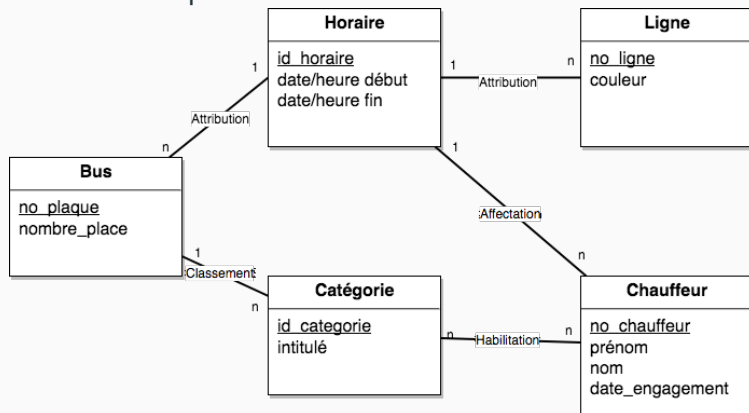
Graphe



Bases de données NoSQL

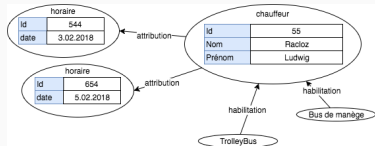
Graphe

Modèle conceptuel



Caractéristiques

- Stockage sous forme de sommets/arcs
- Données peuvent être **normalisées** et **structurées**
- Très grand nombre de relations
- Transactions ACID
- **sharding** + **réplication**
- Déploiement décentralisé
- Modèle conceptuel = Modèle logique



Exemple

- Neo4j, OrientDB, ArangoDB
- Recommandation, réseau social, IOT
- eBay, Cisco, UBS, IBM, Tom-Tom, VERISIGN, Warner, ...

Bases de données NoSQL

Complexité des relations



Image : *DataStax - DBAs guide to NoSQL*

Base de données distribuées

Bases de données NoSQL

Théorème de CAP

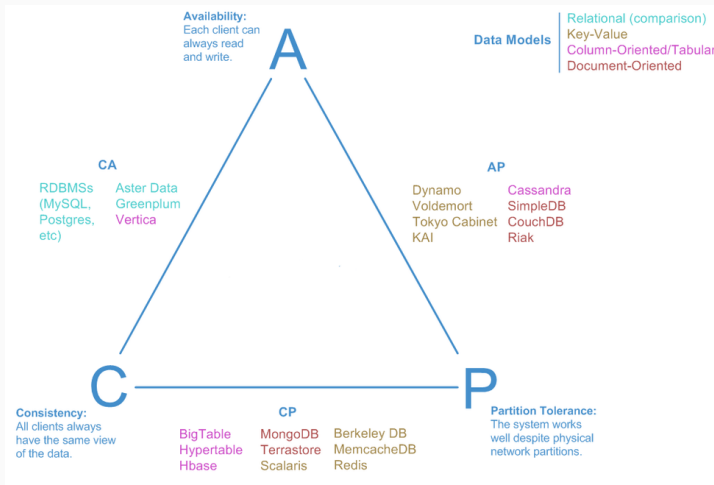
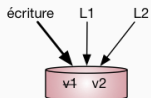


Image : <http://userscontent2.emaze.com/>

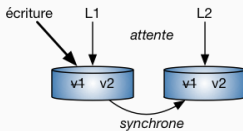
Bases de données NoSQL

Théorème de CAP

CA
Cohérence + Disponibilité



CP
Cohérence + Distribution



AP
Disponibilité + Distribution

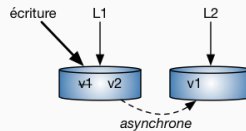
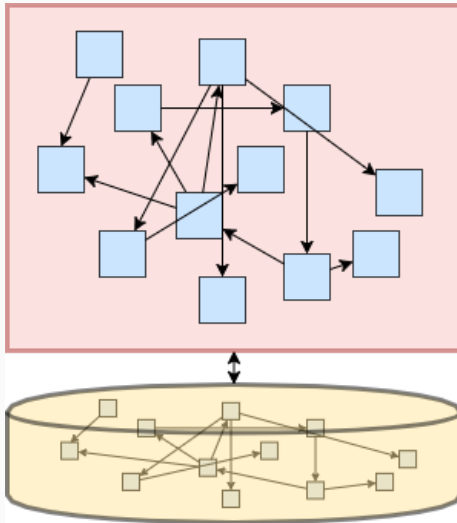


Image : <https://openclassrooms.com>

Base de données et architecture

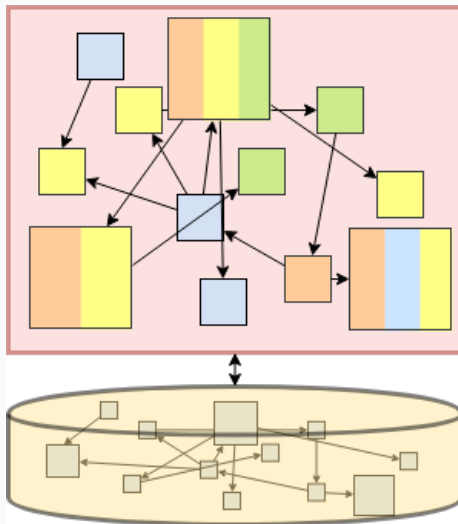
Architecture et bases de données

Système monolithique



Architecture et bases de données

Système monolithique



Architecture et bases de données

Système monolithique

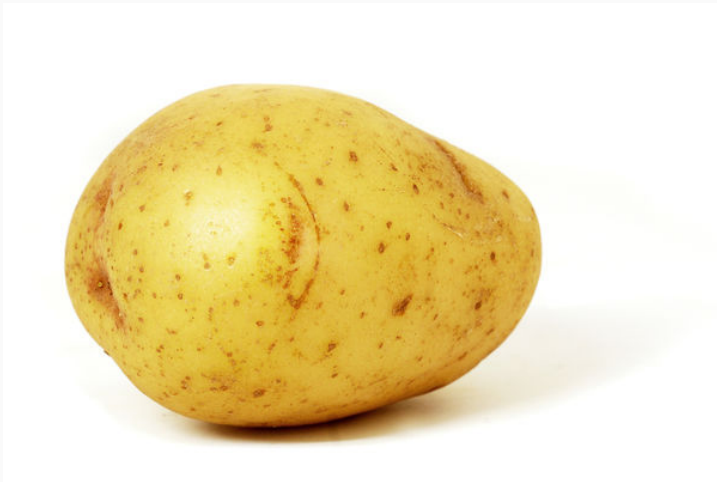


Image : <https://i.imgur.com/>

Architecture et bases de données

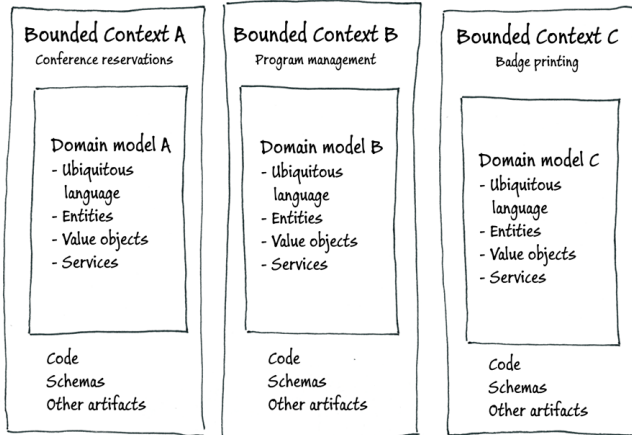
Du système monolithique vers les microservices



Architecture et bases de données

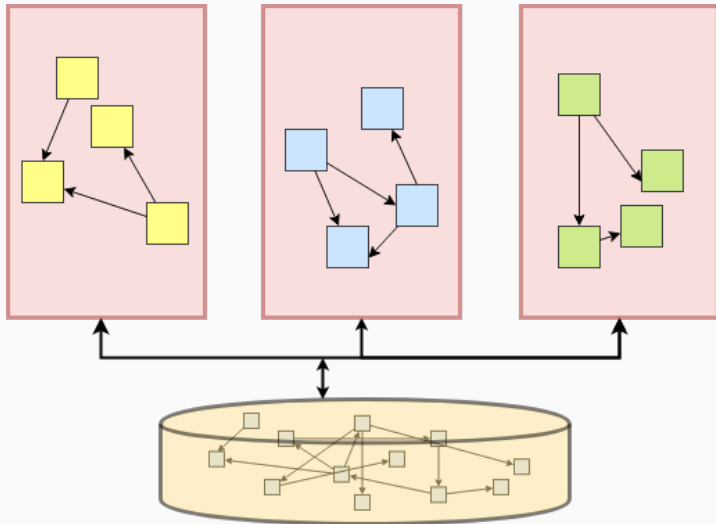
2003 : Domain Driven Design (Eric Evans)

Conference Management System



Architecture et bases de données

2003 : Système orienté services



Architecture et bases de données

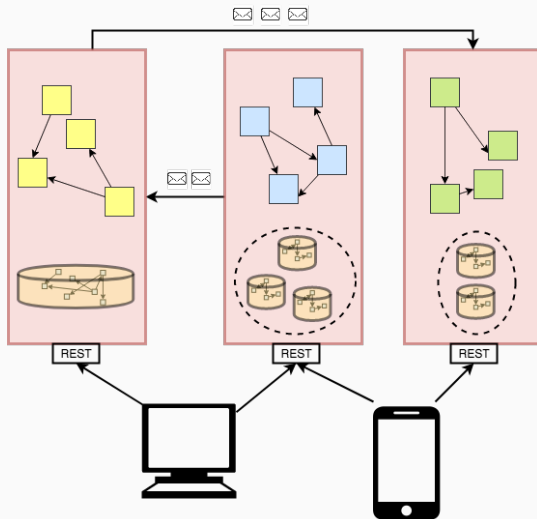
2003 : Système orienté services



Image : *memes.com*

Architecture et bases de données

Dès 2010 : NoSQL, microservices



for ms in microservices :

- domaine compréhensible
- une stack technique
- une équipe
- développement, déploiement, tests rapides et indépendants
- isolé
- maintenance



Idée originale : *Vaughn Vernon*

Image : <http://www.saint-etienne-metropole.fr/>

Quelques considérations :

- **micro**-responsabilité
- microservices
forment un système
- isolation =>
système distribué => :-/



Architecture et bases de données

Programmation réactive

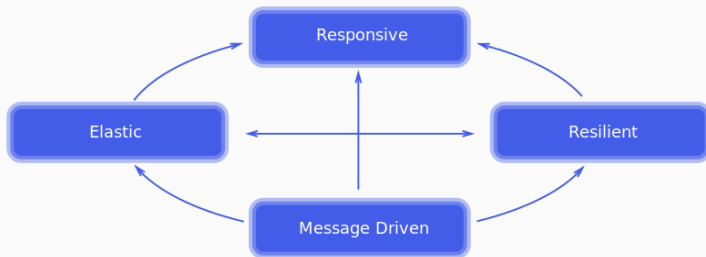


Image : <https://i0.wp.com/springframework.guru>

Quelle base de données choisir ?

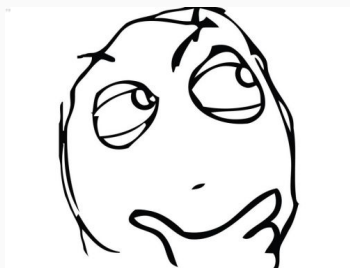
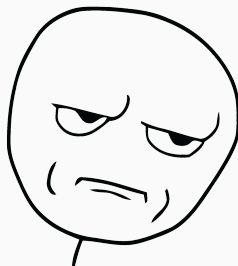


Image : *memes.com*

Quelles**s** bases**s** de données choisir ?

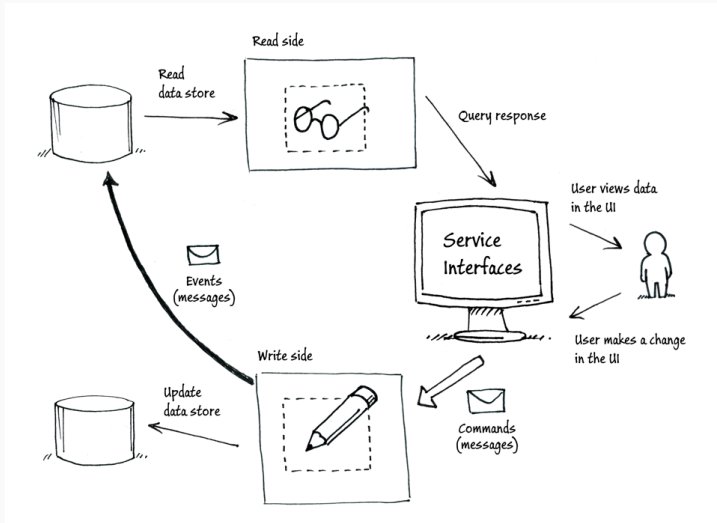


REALLY..????

Image : *memes.com*

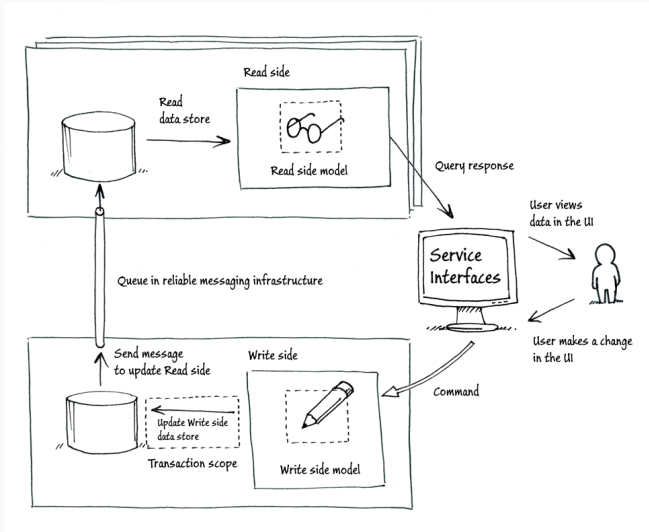
Command and query responsibility segregation (CQRS)

Deux modèles



Command and query responsibility segregation (CQRS)

Deux modèles



Event Sourcing

RDBM != événements

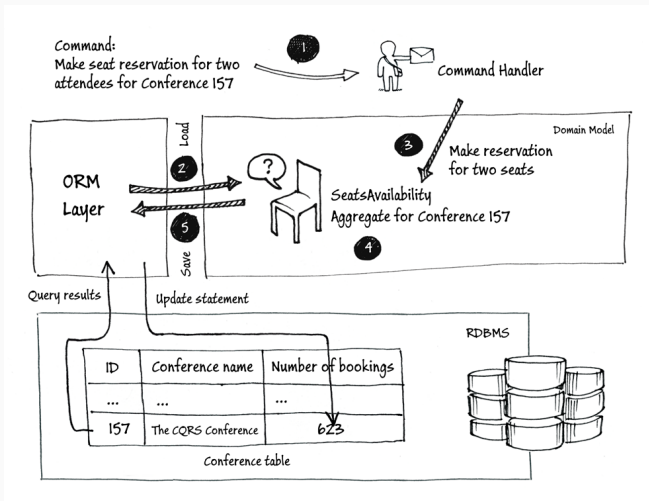


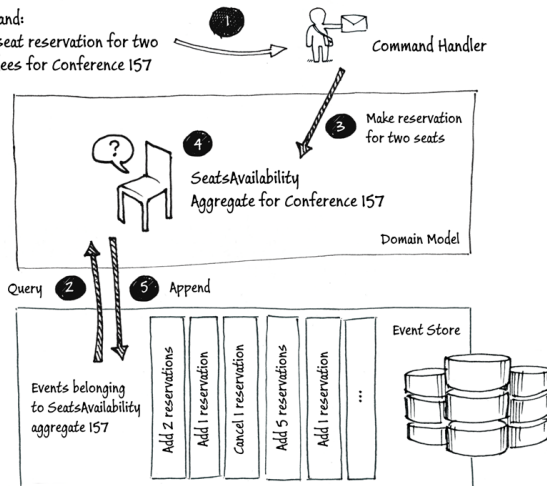
Image : <https://msdn.microsoft.com/>

Event Sourcing

with an Event Store

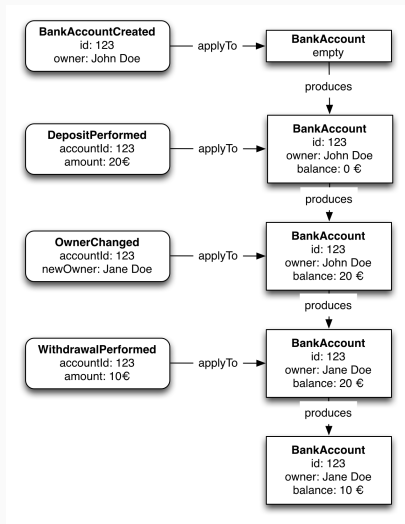
Command:

Make seat reservation for two attendees for Conference 157



Event Sourcing

with an Event Store



2 bases de données + 1 message broker ?

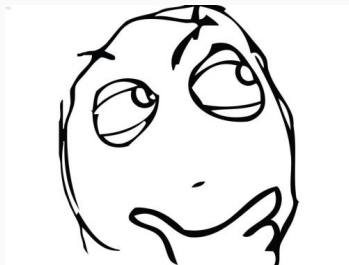


Image : *memes.com*

Architecture et bases de données

Message broker + Event Store



Images : <https://kafka.apache.org> & <http://www.couteauxsuisses.fr/>

Architecture et bases de données

Message broker + Event Store

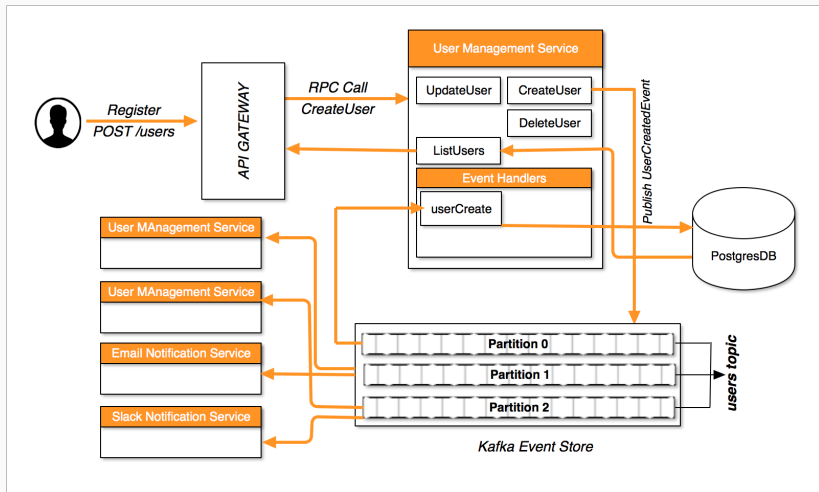


Image : <https://initiate.andela.com>

Conclusions

Quelle(s) base(s) de données choisir en début de projet ? Sur quels critères faut-il se baser ?

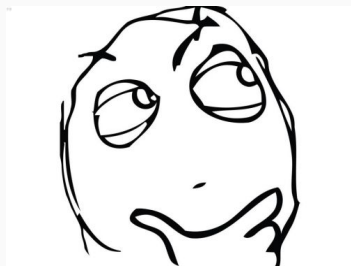
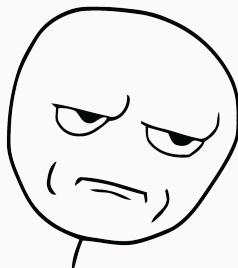


Image : *memes.com*

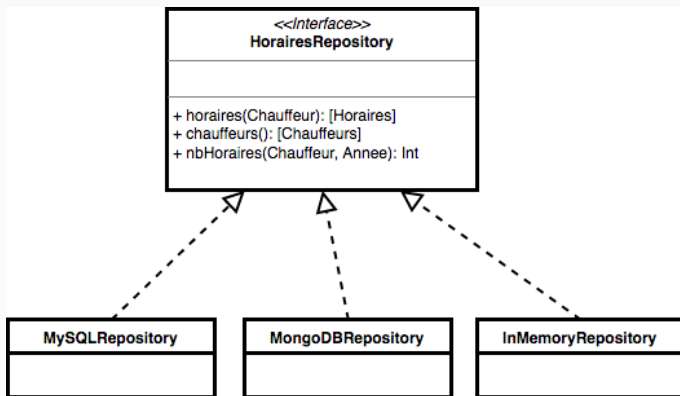
AUCUNE !



REALLY..????

Image : *memes.com*

« A good architecture allows major decisions to be deferred » -Robert Martin, the Clean Architecture



Comment choisir sa base de données ?

- Transactions ?
- Contraintes par le modèle logique ?
- Anomalies ?
- Architecture ?
- Où concentrer les performances ?
- Montée en charge ?
- Quelles requêtes ?
- Relations complexes ?

Simplicité, tolérance aux pannes, montée en charge, performance en lecture => **Dénormalisation** => Efforts sur la gestion des anomalies, Non ACID

Transaction ACID, cohérence et contraintes fortes, éviter la gestion d'anomalies => **Normalisation** => Performance moindre, sharding et réplication difficile