A la découverte de CosmosDB

MUG .Net Nantes – 25 Février 2020

Slack Microsoft Tech Community
https://bit.ly/2MngLYh

Scannez moi !

Michel Perfetti

Directeur Technique @Cellenza

@miiitch

- Développeur logiciel depuis 2000
- 7 ans de développement sur Azure

Mais c’est quoi déjà CosmosDB?

“Azure Cosmos DB is a fully managed NoSQL database service for modern app development. Get guaranteed single-digit millisecond response times and 99.999-percent availability, backed by SLAs, automatic and instant scalability, and open-source APIs for MongoDB and Cassandra. Enjoy fast writes and reads anywhere in the world with turnkey multi-master global distribution.”

https://azure.microsoft.com/en-us/services/cosmos-db/

"Azure Cosmos DB is a fully managed NoSQL database service for modern app development. Get guaranteed single-digit millisecond response times and 99.999-percent availability, backed by SLAs, automatic and instant scalability, and open-source APIs for MongoDB and Cassandra. Enjoy fast writes and reads anywhere in the world with turnkey multi-master global distribution."

- Documents (JSON)
- Cassandra
- Graph
- Mongo
- Table

https://azure.microsoft.com/en-us/services/cosmos-db/

“Azure Cosmos DB is a fully managed NoSQL database service for modern app development. Get guaranteed single-digit millisecond response times and 99.999-percent availability, backed by SLAs, automatic and instant scalability, and open-source APIs for MongoDB and Cassandra. Enjoy fast writes and reads anywhere in the world with turnkey multi-master global distribution.”

- Pas d’infra à gérer
- SLA
- Monitoring
- Backup
- Data Explorer

https://azure.microsoft.com/en-us/services/cosmos-db/

“Azure Cosmos DB is a fully managed NoSQL database service for modern app development. Get guaranteed single-digit millisecond response times and 99.999-percent availability, backed by SLAs, automatic and instant scalability, and open-source APIs for MongoDB and Cassandra. Enjoy fast writes and reads anywhere in the world with turnkey multi-master global distribution.”

- Autoscaler
- Via le portail
- Via les API
- Mode « Serverless »

https://azure.microsoft.com/en-us/services/cosmos-db/

“Azure Cosmos DB is a fully managed NoSQL database service for modern app development. Get guaranteed single-digit millisecond response times and 99.999-percent availability, backed by SLAs, automatic and instant scalability, and open-source APIs for MongoDB and Cassandra. Enjoy fast writes and reads anywhere in the world with turnkey multi-master global distribution.”

https://azure.microsoft.com/en-us/services/cosmos-db/

- Multi région
- Au plus près
- Cohérence à la demande
- Multi Write

CosmosDB (NoSQl)

vs

Bases de données (relationnelles)

CosmosDB (NoSQl)

CosmosDB (les +)

- Multi modèles
- Données hiérarchiques (json)
- Scalabilité & Performances
- Cohérence de la donnée

CosmosDB (les -)

- Transactions limitées
- Gestion de l'intégrité
- Cohérence de la donnée

Bases de données (relationnelles)

BDD Relationnelles (les +)

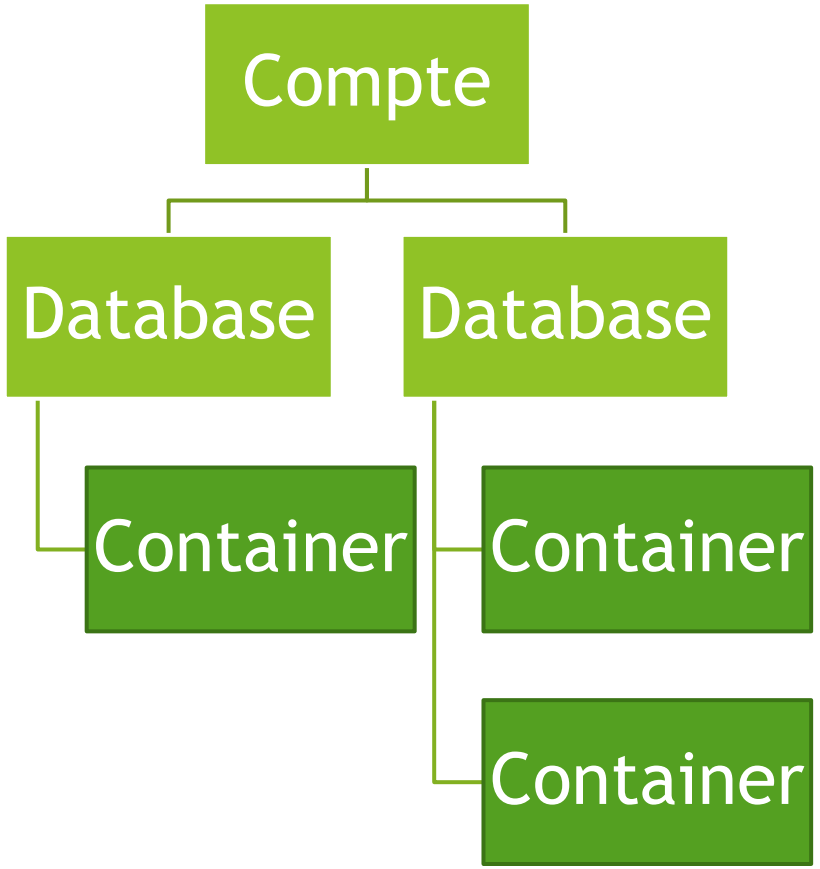
- ACID transactions
- Schéma
- Données cohérentes

BDD Relationnelles (les -)

- Distribution
- Temps de réponse
- Disponibilité

Stockage de la donnée dans Cosmos DB

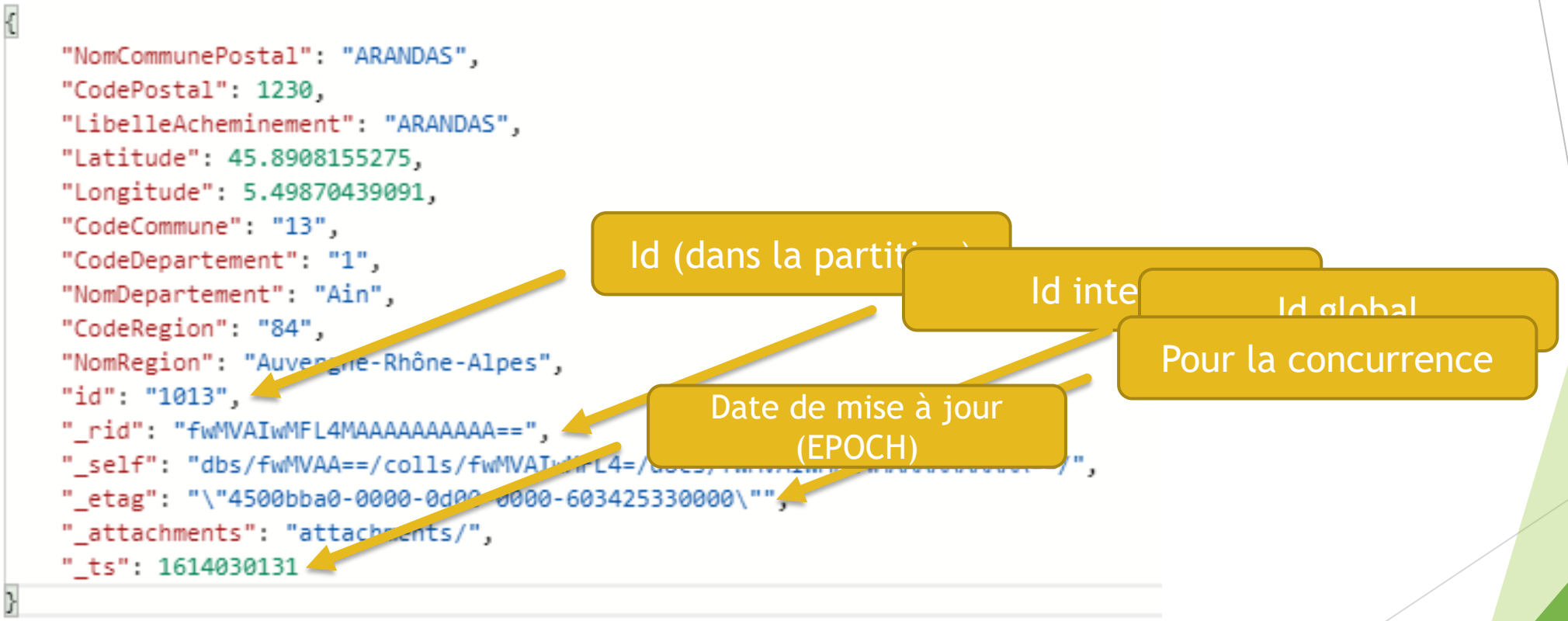
Structure d’une instance CosmosDB



Container: principaux éléments

- Eléments (en fonction du modèle)
- Composants programmables (JS):
  - Procédures stockées
  - Fonctions
  - Déclencheurs (triggers)
  - Merge algos

Structure d’un document



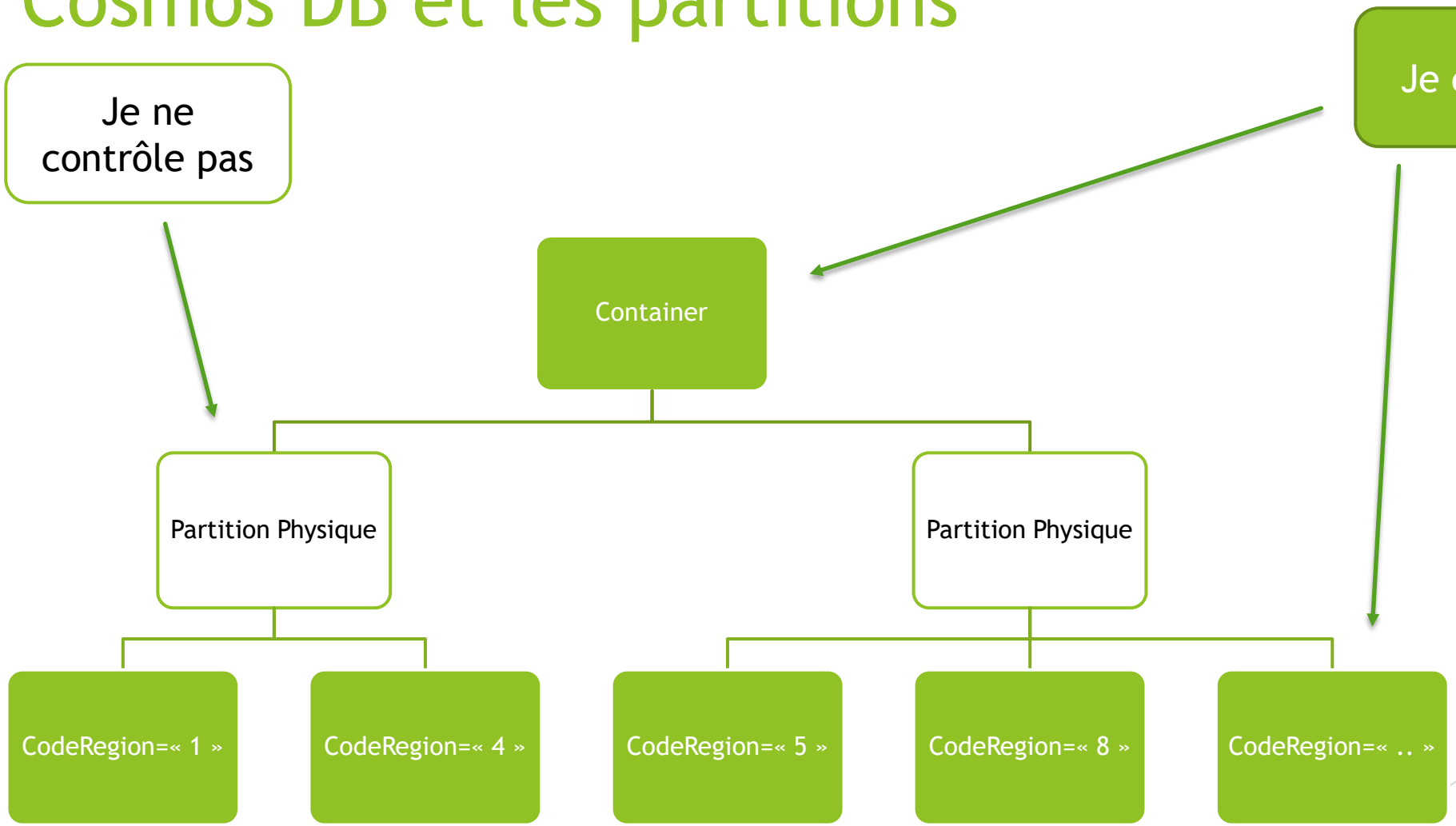
Mais on aurait pu avoir:


```
{
    "NomCommunePostal": "ARANDAS",
    "CodePostal": 1230,
    "LibelleAcheminement": "ARANDAS",
    "Coordonnees":
    {
        "Latitude": 45.8908155275,
        "Longitude": 5.49870439091
    },
    "CodeCommune": "13",
    "Departement":
    {
        "Code": "1",
        "Nom": "Ain"
    },
    "Region" :
    {
        "Code": "84",
        "Nom": "Auvergne-Rhône-Alpes"
    },
    "id": "1013",
    "_rid": "fwMVAIwMFL4MAAAAAAAAAA==",
    "_self": "dbs/fwMVAA==/colls/fwMVAIwMFL4=/docs/fwMVAIwMFL4MAAAAAAAAAA==/",
    "_etag": "\"4500bba0-0000-0d00-0000-603425330000\"",
    "_attachments": "attachments/",
    "_ts": 1614030131
}
```

Mais il manque une information essentielle….

La clé de partition

Cosmos DB et les partitions



Cosmos DB et les partitions

- On contrôle les partitions logiques « via la clé de partition »
  - Eviter les partitions « chaudes »
  - Eviter les requêtes cross partitions
- On ne contrôle pas les partitions physiques
- Le débit (throughtput) est divisé par nombre de partition physique
- On a du monitoring via le portail

Throughtput

- 2 modes:
  - Réservé (avec autoscaling): au niveau de la base ou du container
  - Serverless (paye à la demande)
- Unité: RU/s « Request Unit », 1 RU = lire un fichier de 1Ko
- Quand on dépasse la capacité → throttling

Multi région

- Plusieurs modèles:
  - One Write, Many Read
  - Many Write, Many Read -> réconcilliation
- CosmosDB choisit la région « la plus proche » mais il est aussi possible de la choisir
- Throughput global = Throughput * nb de région

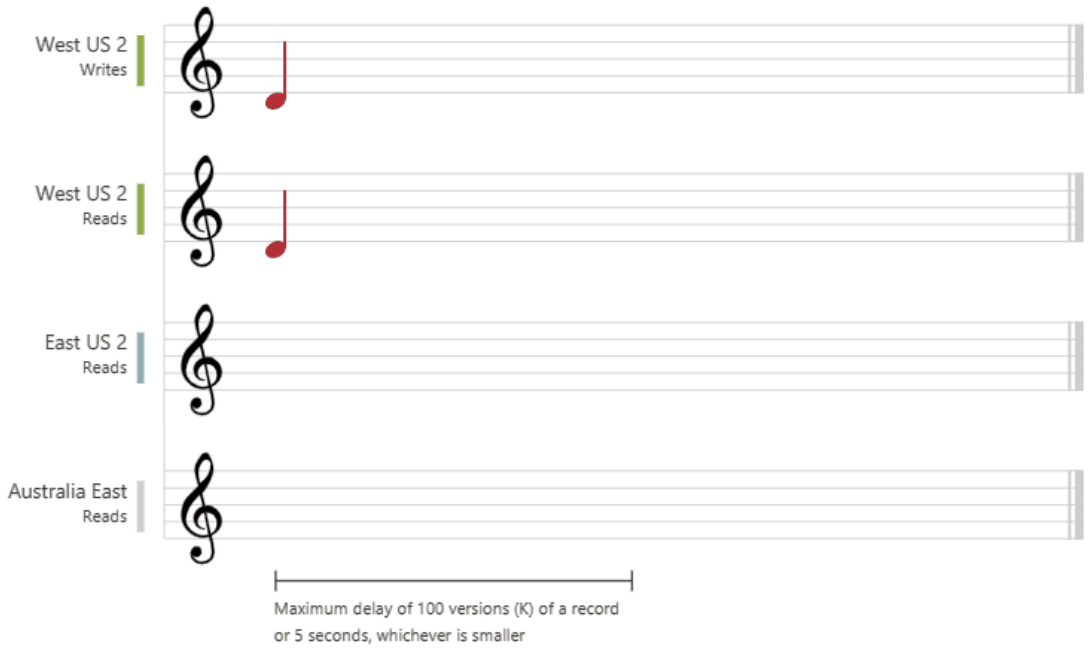
Cohérence

- Plusieurs modèle de cohérence
  - Strong Consistency
  - Bounded Staleness
  - Session
  - Consistent Prefix
  - Eventual
- Cohérence par défaut mais possibilité de choisir par requête.
- Choisir la cohérence en fonction du type de l'application

Strong Consistency

- Toujours la lecture de la version plus récente
- Latence plus élevée

Bounded Staleness

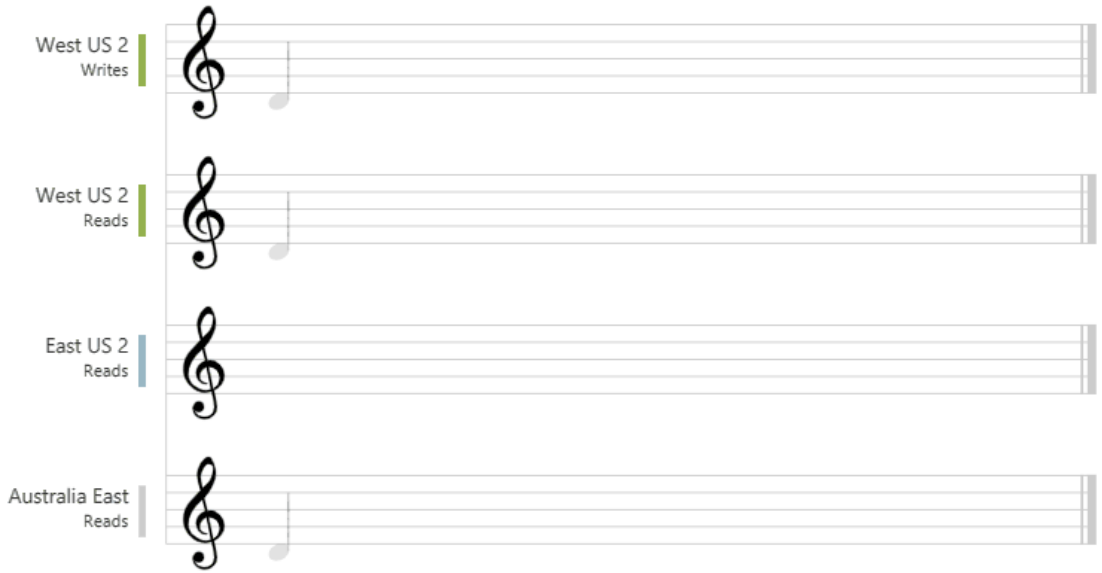


- L’ordre est respectée
- On supporte un decalage (avec un maximum) entre la lecture et la version la plus récente

Session

- Le mode le plus courant
- Chaque lecture dans la même session est la même quelque soit le réplicat

Consistent Prefix



- L’ordre est garanti
- Pas de délai pour recevoir la donnée

Eventual

- Latence la plus faible
- Aucun ordre garanti

Super pouvoir #1: le changelog

- Permet de s’abonner à un changement de donnée
- Permet le développement d’application réactives

Super pouvoir #2: TTL

- Permet de définir quand un document va être automatiquement supprimé
- Valeur par défaut par collection
- Définissable par document