Nom & Prénom :
Groupe :

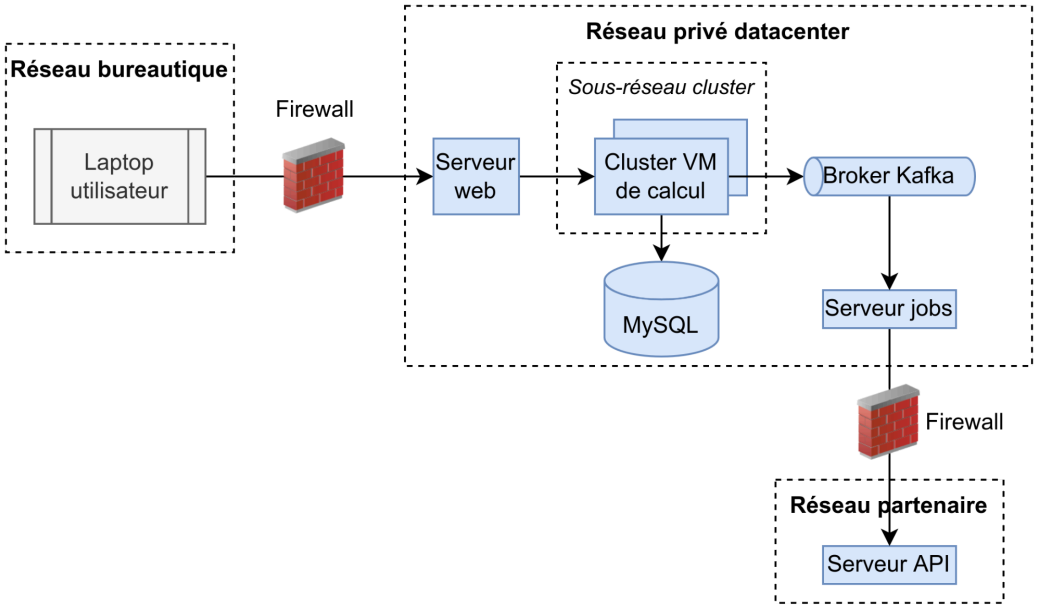
Approfondissement Big Data

R6.VCOD.05 – Evaluation écrite

Les points attribués à chaque question sont des pondérations relatives : le total est ramené à 20 au moment de la notation. Il est tenu compte du niveau général des copies.

Question 1 (1 point)

A quel niveau d'architecture ce schéma correspond-il ?



- ☐ Fonctionelle
- ☐ Applicative
- ☐ Technique

Question 2 (3 points)

On constate que l'interface BI de PomSort tombe régulièrement en panne, empêchant les responsables d'usine de suivre le bon déroulement des aiguillages (même quand ceux-ci fonctionnent bien). Les managers affirment que l'interface est très souvent indisponible, les ingénieurs parlent de quelques incidents isolés par jour ; quant aux directeurs qui vont consulter de temps en temps, ils n'ont pas vu de problèmes.

Pas question de chercher des bugs tant qu'on y verra plus clair. Comment mettre tout le monde d'accord et éviter que les réunions ne se transforment en dialogues de sourds ? Proposez le plan le plus précis possible !

Question 3 (3 points)

Proposer un ADR pour la problématique suivante : *"Les plantages de l'interface BI ont été attribués à un problème de mémoire sur le serveur, lorsque l'utilisateur sélectionne une période d'historique supérieure à 1 an"*.

Rappel : un ADR arrive après une étude technique et entérine une décision de modifier l'architecture. La décision ne peut pas être simplement de mener l'étude en question. Dans ce cas précis, où il n'y a pas eu d'étude réelle, vous êtes libres d'inventer tant que cela reste plausible.

Question 4 (1 point)

Le prototype de PomSort est déployé sur le cloud AWS et donne satisfaction : l'équipe a le budget pour en faire une vraie application. Il faut maintenant la recopier, dans un nouvel espace AWS tout propre. Malheureusement toutes les ressources cloud du prototype avaient été créées à la main dans l'IHM AWS, puis paramétrées avec de nombreux ajustements, et plus personne ne sait comment elles ont été configurées pour faire marcher le prototype.

Quelle pratique l'équipe aurait-elle pu mettre en oeuvre pour éviter cela ?

Question 5 (1 point)

Vous faites partie de l'équipe qui réalise un cycle du produit PomSort : l'optimisation de la production. Concrètement, un moteur d'optimisation (ou "solveur") doit être utilisé, pour aiguiller les pommes intelligemment en fonction des besoins de chaque chaîne, selon les commandes. Se pose la question du choix du solveur, en fonction des règles d'optimisation.

A quelle étape projet cette étude intervient-elle ?

- ☐ Cadrage
- ☐ Développement
- ☐ Déploiement en production

Question 6 (3 points)

Transpomme a des problèmes de production. Elle soupçonne que certaines usines sont parfois engorgées, ce qui engendre des retards en cascade dans les commandes. Difficile d'en être certain cependant, car personne n'a de vue globale sur le devenir des pommes une fois passées les portes de la première usine. Pour analyser cela plus concrètement, elle dispose d'un historique, sur l'année passée, des flux de matière entre les usines (matière : pommes entières, morceaux crus ou cuits, produits transformés divers, additifs, … tout ce qui rentre dans la fabrication des produits de la société).

Comment peut-elle exploiter ces données pour mettre en évidence d'éventuels engorgements ?

Question 7 (2 points)

Les images capturées par PomSort arrivent dans le datalake sous la forme de fichiers RAW à très haute résolution. Il a été décidé de transformer ces données en fichiers JPEG de moyenne résolution, moins gourmands, d'éliminer les images sans intérêt (ex. absence de pomme sur l'image suite à un déclenchement intempestif de la caméra), et d'injecter dans les métadonnées de chaque fichier JPEG (EXIF) le n° de la chaîne de production où la caméra est installée. Cette dernière information provient d'un fichier Excel statique.

Quelles pourraient être les déclinaisons de ces données dans chaque couche d'une architecture en médaillon :

- Couche "Bronze"
- Couche "Silver"
- Couche "Gold"

Question 8 (2 points)

La partie BI de l'application PomSort est composé de 2 parties :

- Un "frontend" qui porte l'interface graphique
- Un "backend" connecté à la base de données et qui fait les requêtes et les calculs pour le compte du frontend. Le backend publie aussi des alertes quand la production décline ; les alertes sont capturées par divers systèmes de surveillance.

Nous avons donc 2 échanges de données : le frontend avec le backend (faisant partie de la même application), et le backend avec des systèmes de surveillance (applications indépendantes).

Pour chacun de ses échanges, préciser quel modèle est le plus approprié : queue ou topic, en expliquant pourquoi.

Question 9 (2 points)

Il y a 2 problèmes avec ce code : lesquels ?

```python
# Librairie d'accès à la BDD
from libs import database

def get_latest_apple_statistics(hours_since: int):
    # Compte les pommes par variétés passées depuis hours_since heures
    sql = f'''
        select
            variety,
            count(*) as apples
        from captures
        where timestamp >= now() - interval {hours_since} hours
    '''

    with database.connect() as session:
        statistics = await session.execute(sql)

    return to_dataframe(statistics)
```

Question 10 (5 points)

Voici la description d'un traitement Spark Streaming, qui tourne en continu :

- Au lancement du traitement, la liste des coopératives avec leurs adresse est chargée depuis la base de données de SAP, l'application qui gère les commandes
- Côté logistique, un système publie, à chaque livraison de lot, un message sur un topic Kafka, avec comme informations : le nom de la coopérative, le n° de lot, et le poids livré
- Les pommes du lot sont réparties dans l'usine, plusieurs chaînes faisant la transformation en compote en parallèle
- En bout de chaîne, une seule unité mélange les compotes produites par les chaînes précédentes pour les mettre en pot. Un système produit un "top" pour chaque pot rempli
- Le traitement Spark doit construire la liste des n° de lots entrant dans la composition de chaque pot. Elle est calculée ainsi : ce sont tous les lots qui sont entrés dans l'usine au plus 1h avant l'émission du "top"
- Cette liste est envoyée sous forme de message à une API HTTP du département logistique, pour l'expédition

Décrire les différents dataframes qui interviennent dans ce traitement. Pour chaque dataframe, indiquer :

- Les sources de données ou autres dataframe dont il dépend
- Si c'est un dataframe Spark "classique" (batch) ou un dataframe de streaming
- Dans le cas d'un dataframe de streaming, l'opération qui correspond à la transformation (ex. jointure, regroupement, …)

Décrire aussi le puits.

NB : il n'est pas demandé d'écrire du code, juste des descriptions sommaires permettant de comprendre la logique du traitement