

# Introduction to Software Architecture

---

- Date : 21 mai 2021
- Durée : 09h00 – 11h30 (2 heures 30)
- Aucun document autorisé ; barème donné à titre indicatif.

Lisez tout le sujet avant de commencer à répondre aux questions ; les questions sont identifiées en gras dans le texte ; les différentes parties sont indépendantes.

Vous devez répondre aux questions posées dans le sujet sur la copie d'examen fournie, avec d'éventuelles feuilles intercalaires. Vous ne pouvez pas sortir de la salle d'examen durant la première heure (avant 10h00), ni durant le dernier quart d'heure (après 11h15). **Toute fraude identifiée sera systématiquement transmise au conseil de discipline de l'Université.**

## **Partie #1 : Principes de base /8 (0h30)**

Durant votre projet « CastexSki », vous avez été confronté au développement concret d'une architecture 3-tiers. En utilisant les connaissances acquises durant votre formation dans le département (vous pouvez/devez faire référence à d'autres cours qu'ISA) et votre expérience en projet ISA, **répondez à chaque question de manière synthétique (~150 mots, 200 max) :**

1. Quelle est la différence entre un objet métier et un composant métier ? et entre "objets métiers" et "objets tout court" ?
2. Pourquoi introduire des cascades dans les relations entre objets métiers ?
3. Quel est l'impact d'être "Stateful" pour un composant ? Quels sont les avantages des composants "Stateless" ?
4. Quel est le rôle des intercepteurs dans une architecture ?

## **Partie #2 : Étude de cas /12 (2h00)**

Castexski a été mis en service avec succès. Après un an d'utilisation, des évolutions sont demandées par la société gestionnaire des stations de ski, et le briefing suivant vient d'être reçu.

### **Partie 2.1 : Utilisation de caméras (/4)**

Des caméras sont placées en bas et en haut de toutes les remontées mécaniques. Elles permettent de donner une vue d'ensemble de la station sur le Web, mais les gestionnaires veulent les exploiter pour affiner les informations sur la charge en temps-réel des remontées (et donc aussi le rééquilibrage de trafic) en faisant tourner des algorithmes de comptage de personnes par analyse vidéo pour estimer la charge des personnes en attente devant la remontée à partir des caméras en bas de piste (on considère que seules les caméras bien positionnées pour faire l'analyse vidéo sont accessibles).

Du point de vue du système de gestion de station, le système d'analyse vidéo est considéré comme un ensemble de services externes qui remontent uniquement les informations nécessaires avec :

- un service d'identification des caméras implémenté en REST. il permet de récupérer (GET) l'ensemble des ids de caméras, et pour chaque caméra, le nom de la remontée filmée, un booléen qui indique si l'analyse du flux est bien effective sur la caméra ;



- un autre service qui permet d'obtenir le comptage du nombre de personnes en attente devant la remontée pour une caméra à partir de son id, le comptage étant fourni avec un timestamp correspondant à l'instant filmé par la caméra.

A partir de ces informations, le système Castexski devra calculer le temps d'attente en bas de chaque remontée et son évolution à court terme (la file d'attente est-elle en train de s'allonger ou de se raccourcir ?). Le rééquilibrage de trafic pourra se faire si le temps d'attente dépasse ou redescend sous un certain seuil, mais aussi bien avant, si l'évolution à court terme montre que le file s'allonge ou se raccourcit très vite.

## Partie 2.1 : Géolocalisation des pisteurs (/8)

Afin d'optimiser le traitement des accidents, mais aussi afin de répartir plus de pisteurs sur les pistes à fort trafic, chaque pisteur sera équipé d'une balise de géolocalisation en temps-réel (les batteries des équipements durent une journée de travail) et les fonctionnalités suivantes doivent être implémentées :

- affichage sur les écrans de contrôle du centre station de la position en temps-réel des pisteurs à partir des positions obtenues toutes les minutes ;
- envoi de message texte à un ou tous les pisteurs (le message reçu par le pisteur renvoie un accusé de réception au système / le message lu par le pisteur renvoie un second accusé de réception) ;
- ordre de déplacement à un point précis pour un pisteur (avec le même double accusé de réception que pour le point précédent).

*Pour cette partie, vous êtes libre d'utiliser le protocole qui vous semble le plus approprié avec les interfaces que vous définirez pour effectuer la liaison entre chaque pisteur et le système CastexSki.*

### Points à aborder

Sur la base des informations contenue dans ce briefing, vous devez proposer une architecture 3-tiers y répondant, tout en prenant en compte le fait que l'on souhaite avoir uniquement des composants **stateless**.

Seuls les composants et objets de CastexSki nécessaires au traitement des nouveaux scénarios doivent être décrits avec les deux extensions demandées. Faites bien attention à identifier dans vos réponses l'impact de vos modifications par rapport à l'existant, en explicitant les différences et leurs conséquences sur l'architecture.

Les points suivants vous permettront de cadrer votre étude de cas :

1. Identifiez les objets métiers au sein de votre architecture ;
  - Décrivez ces objets sous la forme d'un diagramme de classes ;
  - Expliquez le *mapping* Objet-Relationnel quand c'est pertinent.
2. Identifiez les différents composants à mettre en jeu dans votre architecture ;
  - **Décrivez l'assemblage** sous la forme d'un diagramme de composants ;
  - **Décrivez les interfaces** de chaque composant (signatures typées des interfaces en pseudo-code) et des services externes que vous *mockez*
3. **Justifiez vos choix de conception** pour cette architecture, en identifiant quels éléments appartiennent à quel tiers de l'application, et pourquoi ils sont nécessaires dans le système
4. Expliquez comment cette architecture s'implémente dans l'écosystème J2E.

N'oubliez pas que les justifications ont une très grande importance, des propositions bien justifiées rapporteront des points même si elles sont incomplètes.