

Examen SOA1 : Services & Intégration

- Date : 17 novembre 2014
- Durée : 13h30 – 17h30 (4 heures)
- **Aucun document autorisé; barème donné à titre indicatif.**

Deleted: →

Lisez tout le sujet avant de commencer à répondre aux questions;
Les questions sont identifiées en gras dans le texte du sujet.

Les parties 1 et 2 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 14h30), ni durant le dernier quart d'heure (après 17h15). **Toute fraude identifiée sera systématiquement transmise au conseil de discipline de l'UNS.**

Partie #1 : Étude d'une API existante

/ 10

(2h)

Question	1.1	1.2	1.3	1.4
Barème	/1	/2	/3	/4

Vous trouverez en annexe de ce sujet l'API orientée ressource officielle des magasins *Walmart* (première chaîne de distribution en Amérique du Nord). Vous êtes employé par la DSI de *Walmart*, en charge d'un projet pilote pour migrer l'API existante vers une technologie basée sur SOAP.

1. **Décrivez en pseudo-code** comment calculer la moyenne des évaluations associées aux 10 produits suggérés (recommandés) lors de l'achat d'un téléphone portable « *Samsung Galaxy Note IV* » en utilisant l'API de *Walmart*.
2. **Analysez l'API définie par Walmart.** Est-elle respectueuse de ce qui est attendu dans une implémentation orientée ressource (une phrase maximum) ? Quelles sont ses forces ? Pouvez vous identifier des faiblesses, par exemple au niveau des adresses de ressources exposées et/ou des objets métiers retournés ?
3. **Identifiez les objets métiers** que vous souhaiteriez exposer aux consommateurs de votre API. Proposez un diagramme de classe (justifié) modélisant ces objets métiers.
4. **Identifiez le ou les services** que vous souhaiteriez exposer au dessus de SOAP (paradigme RPC ou DOC, au choix). Pour chaque service, décrivez en une phrase son objectif puis identifiez clairement son interface : opérations exposées, signature de chaque opération (paramètres d'entrée et message de retour).

Partie #2 : Étude de cas « BD-* »

/ 10

(2h)

Question	2.1	2.2
Barème	/4	/6

On considère ici une structure académique fictive, *Multitech'Nice*. Contrairement à d'autres écoles de la région, la vie associative étudiante au sein de *Multitech'Nice* est florissante : le *bureau des étudiants* (BDE) gère la vie étudiante au quotidien, le *bureau de la musique* (BDM) organise répétitions et concerts sur le campus, le *bureau des sports* (BDS) propose des activités sportives et gère des tournois inter-promos, le *bureau des jeux* (BDJ) organise des après-midi jeux de sociétés et des LAN-party, et pour finir le *bureau des anciens* (BDA) organise la remise des diplômes annuelle et anime un cycle de conférences pour maintenir des liens entre *Multitech'niçois* actuels et anciens, le tout en bonne intelligence et avec le soutien de l'administration de l'école. Chaque association est indépendante des autres (déclarée en préfecture comme association 1901).

Sortant grandie de cette synergie entre les différentes associations étudiantes qui y sont hébergées, l'administration propose le financement d'un projet permettant de fédérer les systèmes d'informations des différentes associations pour rendre plus efficace la communication entre association et renforcer la synergie existante.

Exercice #2.1. Définition des interfaces de services

On s'intéresse ici aux associations suivantes : BDE, BDM et BDS. Leurs spécifications métiers sont définies ci-après.

- Le BDE gère la vie quotidienne des étudiants. Les étudiants peuvent utiliser un système de *Foire aux Questions* (FAQ) pour tout savoir de la vie quotidienne sur le campus. Ils peuvent envoyer de nouvelles questions aux membres du bureau ([ils ne passent pas par le mail](#)), et ceux-ci peuvent y répondre et éventuellement enrichir la FAQ. [Une question posée par un étudiant \(et sa réponse éventuelle\) n'est pas visible d'autres étudiants](#). Le BDE organise aussi des soirées, à un tarif préférentiel pour ses adhérents, [les entrées peuvent être réservée en ligne \(en attente du paiement\)](#);
- Le BDM gère la dimension musicale de la vie étudiante. Il décrit les différents concerts donnés sur le campus, et permet de réserver des places de concerts (sous réserve du paiement en cas d'évènement payant, potentiellement à tarif préférentiel) ;
- Le BDS gère les tournois sportifs. Les étudiants peuvent former des équipes pour s'inscrire aux différents tournois proposés (sous réserve du paiement en cas d'évènement payant), et voir les résultats obtenus par les différentes équipes. [Les responsables du BDS peuvent créer de nouveaux tournois](#).
- Le BDS et le BDM sont inclus dans le BDE : [tout adhérent au BDE est automatiquement adhérent \(gratuitement\) aux BDS et BDM](#) (mais la réciprocité n'est pas vraie).

Chacune des associations dispose d'un service de gestion de ses adhérents, qu'on considère identique (le service est dupliqué chez chaque association). Le service doit permettre d'enregistrer et de lister les adhérents de l'association (*e.g.*, nom, prénoms, email, année), leur historique de cotisation, [de connaître le montant de l'adhésion à l'association et d'obtenir une copie pdf de la carte d'un adhérent](#).

Définissez les interfaces (*i.e.*, opérations exposées et objets métiers) nécessaires à la mise en œuvre d'une architecture orientée service pour ces associations. Vous

utilisez au moins 2 des 3 paradigmes de définition d'interface vue en cours (Resource, RPC ou DOC, au choix), en justifiant votre choix.

Exercice #2.2. Intégration de services

Multitech'Nice a acheté quatre services de gestion : (i) le système de paiement par carte bancaire du *CréditGénéral*, (ii) le système de gestion d'impression *Xenon*, (iii) le système de gestion d'offres d'emploi *Jobster* et pour finir (iv) le service d'envoi d'email *Publmail*.

Leurs interfaces sont données ci-dessous :

- *CréditGénéral* :
 - process : Payer x CardNumber x Amount x Recipient -> Boolean
 - getPayments : Recipient x AccessCode -> ListOf<(Payer, Amount, Date)>
- *Xenon* :
 - getQuotaLevel : UserName -> Integer
 - print : UserName x AccessCode x PDF x NumberOfCopy -> Boolean
- *Jobster* :
 - postOffer : Offer -> Boolean
- *Publmail*
 - send : Message x Addressee -> Boolean
- Le détail du contenu de chaque objet métier (e.g., Offer, Payer) est laissé à votre [discrétion](#).

Le service informatique souhaite utiliser un ESB pour intégrer les services des diverses associations avec les services achetés par Multitech'Nice afin de gérer les scénarios suivants.

- Scénario #1 : **Gestion des tournois de Poker par le BDJ.**
Les tournois de poker sont à cheval entre les prérogatives du BDJ et du BDS. Le BDJ exploite donc le mécanisme de tournoi fourni par le BDS pour gérer ses tournois de poker. Une opération de création de tournoi de poker est exposée sur le bus, qui crée un tournoi coté BDS, puis informe par email les adhérents des deux associations de la mise à disposition de ce nouvel évènement.
- Scénario #2 : **Publication des offres d'emploi par le BDA.**
Le BDA expose une opération permettant de poster une offre d'emploi. Cette opération est réservée aux adhérents de l'association. Si l'offre est postée par un adhérent, elle est créée dans Jobster, et les adhérents des associations en dernière année sont prévenus par email de la parution de l'offre.
- Scénario #3 : **Unification du paiement des cotisations.**
Le système expose un service d'inscription mutualisé, permettant aux étudiants de s'inscrire à une ou plusieurs associations simultanément. Le paiement est pris en charge par carte bleue, un email de confirmation est envoyé à l'étudiant ainsi qu'aux trésoriers des associations concernées, et les cartes d'adhérents associées sont envoyées à l'impression, si le quota d'impression est encore « large ».

En utilisant le paradigme des flots de données (i.e., « pipe and filter »), proposez des flots d'intégration permettant de mettre en œuvre les 3 scénarios précédents. Identifiez (et justifiez) les objets métiers échangés sur le bus.