

IPST-CNAM  
Architecture Logicielles  
**NSY 205**  
Jeudi 28 Juin 2018

Durée : **2 h 30**  
Enseignant : LAFORGUE Jacques

1ère Session NSY 205

## **CORRECTION**

### **1ère PARTIE – SANS DOCUMENT (durée: 1h15)**

#### **1. QCM (35 points)**

##### Mode d'emploi :

Ce sujet est un QCM dont les questions sont de 3 natures :

- **les questions à 2 propositions**: dans ce cas une seule des 2 propositions est bonne.
  - +1 pour la réponse bonne
  - -1 pour la réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est bonne
  - + 1 pour la réponse bonne
  - -1/2 pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
  - + 1/2 pour chaque réponse bonne
  - -1 pour la réponse fausse

Il s'agit de faire une croix dans les cases de droite en face des propositions.

On peut remarquer que cocher toutes les propositions d'une question revient à ne rien cocher du tout (égal à 0).

Si vous devez raturer une croix, faites-le correctement afin qu'il n'y ait aucune ambiguïté.

N'oubliez pas d'inscrire en en-tête du QCM, votre nom et prénom.

Vous avez droit à **4 points** négatifs sans pénalité.

NOM:	PRENOM:
------	---------

La description d'une architecture logiciel peut être un document papier.		Q 1.
1	OUI	X
2	NON	

L'architecture logiciel d'un Système d'Information (SI) est la liste de toutes les spécifications fonctionnelles que le SI doit respecter.		Q 2.
1	OUI	
2	NON	X

La définition d'une architecture permet de préparer et faciliter l'intégration des composants réalisés par les différentes équipes de développement.		Q 3.
1	OUI	X
2	NON	

Dans la démarche d'architecture d'un SI, il existe deux Configurations Architecturales à aborder : la Configuration Architecturale Statique et la Configuration Architecturale Dynamique		Q 4.
1	OUI	X
2	NON	

Il est possible de faire l'Architecture Technique et l'Architecture Physique sans la Configuration Architecturale.		Q 5.
1	OUI	
2	NON	X

L'Architecture Fonctionnelle est la description des exigences dites "non-fonctionnelles" identifiées pendant la conception et la réalisation du Système d'Information. <b>PAS UNIQUEMENT (Question ambiguë =&gt; +1 pour tous)</b>		Q 6.
1	OUI	X
2	NON	X

L'Architecture Technique permet de déterminer le déploiement des composants sur les différentes machines composant le SI.		Q 7.
1	OUI	
2	NON	X

Dans la Configuration Architecturale, le Connecteur est défini comme un composant qui permet de relier deux composants qui se trouve sur des machines différentes.		Q 8.
1	OUI	
2	NON	X

Dans la Configuration Architecturale, le Connecteur précise l'interface requise par un composant et l'interface fournies par un autre composant.		Q 9.
1	OUI	X
2	NON	

Soit les 3 modèles de coopération de composants suivants:

**Modèle 1**

**Modèle 2**

**Modèle 3**

Quel est le mode de contrôle de chacun ces modèles :

1	Modèle 1 = contrôle orienté évènements Modèle 2 = contrôle orienté données Modèle 3 = contrôle orienté procédure	
2	Modèle 1 = contrôle orienté procédure Modèle 2 = contrôle orienté évènements Modèle 3 = contrôle orienté données	<b>X</b>
3	Modèle 1 = contrôle orienté procédure Modèle 2 = contrôle orienté données Modèle 3 = contrôle orienté évènements	

Dans l'Architecture Technique, la conception d'un ORM (Object Relationnal Mapping) permet de réaliser un mapping de programmation efficace entre les données du SI et la Base de Données utilisée.		Q 11.
1	OUI	<b>X</b>
2	NON	

L'utilisation de DAO permet de rendre la couche métier dépendante du support de persistance		Q 12.
1	OUI	
2	NON	<b>X</b>

Il existe différentes approches pour réaliser la persistance des objets dans une base de données. Toutes ces approches nécessitent d'utiliser un progiciel (ou COTS) qui réalise automatiquement le mapping des objets en base de données		Q 13.
1	OUI	
2	NON	<b>X</b>

En JPA, l'annotation @OneToMany permet de traduire le mapping d'une relation *.*		Q 14.
1	OUI	
2	NON	<b>X</b>

L'architecture de type JEE appartient à la famille des architectures suivantes :		Q 15.
1	architectures hiérarchiques et orientées objets	
2	architectures à base de composant	<b>X</b>
3	architectures avec un référentiel de données	

Dans une architecture à base de composant, les différents composants s'exécutent dans différents containers qui assurent notamment la communication entre les composants.		Q 16.
1	OUI	<b>X</b>
2	NON	

Soit le schéma suivant de description d'une architecture type à base de composants :		Q 17.
Ce schéma montre qu'un Client du SI basé sur ce type d'architecture utilise directement les composants du SI.		
1	OUI	
2	NON	X

Le rôle des <b>Conteneurs</b> et des <b>Structure d'Accueils</b> dans une Architecture à Base de Composants est de :		Q 18.
1	gérer le cycle de vie des Composants du SI	X
2	gérer le middleware de communication entre les composants du SI	
3	résoudre les dépendances entre les Composants du SI	X

Le type de la communication entre les composants d'une architecture à Base de Composant est une communication <b>asynchrone</b> .		Q 19.
1	OUI	
2	NON	X

Le principe d'une Architecture à Base de Composants est particulièrement adapté pour réaliser les architectures :		Q 20.
1	distribuées	X
2	N-tiers	X
3	orientées évènements	

Une architecture de type WebServices est basée sur les standards		Q 21.
1	du WEB	X
2	de RMI	

Comme le RMI, l'architecture de type WebServices réalise la communication entre un client et un serveur par l'utilisation d'une amorce (stub) et d'un squelette (Skelton)		Q 22.
1	OUI	X
2	NON	

Dans une architecture Web Services, le client doit être écrit dans le même langage de programmation que celui utilisé pour écrire le serveur contenant l'implémentation des services.		Q 23.
1	OUI	
2	NON	X

Dans une architecture WebServices le rôle du WSDL est :		Q 24.
1	d'enregistrer l'interface publiée par un WebService	
2	définir l'interface des services d'un WebService	X
3	d'encoder et de décoder les messages échangés entre un client et un WebService	

Dans une architecture JEE les EJB Session sont tous des composants sans état interne. Leur état est géré en Session		Q 25.
1	OUI	
2	NON	X

Dans une architecture JEE, un EJB Stateless est une classe contenant des attributs d'objet		Q 26.
1	OUI	X
2	NON	

Un client envoie 2 requêtes consécutives (REQ1, puis REQ2) à un EJB session Stateless et aussi à un EJB session Statefull. Chacun de ces EJB contiennent des attributs.		Q 27.
1	Le EJB <b>Stateless</b> assure que si la requête REQ1 change des attributs de l'EJB alors la requête REQ2 s'exécute avec les valeurs des attributs qui ont été changés par la requête précédente REQ1.	
2	Le EJB <b>Stateful</b> assure que si la requête REQ1 change des attributs de l'EJB alors la requête REQ2 s'exécute avec les valeurs des attributs qui ont été changés par la requête précédente REQ1.	X

Deux EJB Statefull appartenant à deux sessions différentes peuvent s'échanger des informations à travers l'utilisation de Message Driven Beans		Q 28.
1	OUI	X
2	NON	

Dans une architecture REST, les types de requêtes HTTP : GET, PUT, POST, UPDATE et DELETE correspondent à la gestion d'une ressource distante (Create, Read, Update et Delete).		Q 29.
1	OUI	X
2	NON	

Dans une architecture REST, les RESSOURCES sont des composants de type EJB Statefull persistants en Base de Données.		Q 30.
1	OUI	
2	NON	X

Dans une architecture de type MOM, les Composants sont reliés entre eux directement mais leurs liens sont dynamiques et peuvent donc changer en cours d'exécution.		Q 31.
1	OUI	
2	NON	X

<p>En JMS (Java Messaging System), il existe (notamment) deux modes de communication : Queue et Topic.</p> <pre> sequenceDiagram     participant P as Producteur     participant M as Médiateur     participant C1 as Consommateur     participant C2 as consommateur      P-&gt;&gt;M: publish(m1)     P-&gt;&gt;M: publish(m2)     P-&gt;&gt;M: publish(m3)     M-&gt;&gt;C1: send(m1)     M-&gt;&gt;C2: send(m1)     M-&gt;&gt;C1: send(m2)     M-&gt;&gt;C2: send(m2)     M-&gt;&gt;C1: send(m3)     M-&gt;&gt;C2: send(m3)             </pre> <p>Ce diagramme de transition correspond au mode :</p>		Q 32.
1	Topic	X
2	Queue	

<p>Dans un MOM, le mode "Queue" de communication par message assure que tous les consommateurs d'un canal d'évènement reçoivent bien le message.</p>		Q 33.
1	OUI	
2	NON	X

<p>Dans une architecture JEE, les composants qui appartiennent au container WEB sont :</p>		Q 34.
1	les EJB Session	
2	Les Servlets	X
3	Les JSP (Java Server Page)	X

<p>Dans une architecture JEE, une application cliente peut utiliser directement les composants du container EJB</p>		Q 35.
1	OUI	X
2	NON	

*Fin du QCM*

*Suite (Tournez la page)*

## 2. Questions libres (15 points)

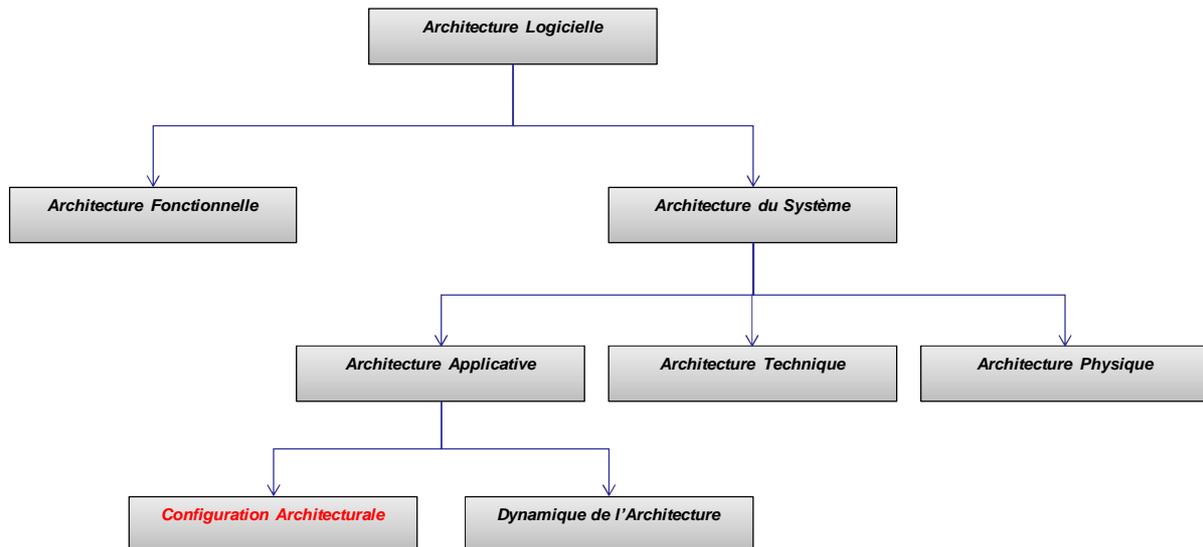
Chaque question est notée sur 5 points.

Vous répondez à ces questions sur une **copie vierge double** en mettant bien le numéro de la question, sans oublier votre nom et prénom.

Vous mettez le QCM dans cette copie vierge double.

### QUESTION NUMERO 1

Soit le schéma suivant de la décomposition de la démarche d'architecture d'un Système d'Information :



Commentez ce schéma en mettant en évidence le rôle primordial de certains de ces composants.

Ce schéma montre qu'une architecture logicielle est structurée en différentes étapes et sous-étapes dont la principale est la "Configuration Architecturale" qui sert de support à la réalisation des Architecture technique et physique. C'est la Configuration Architecturale qui décrit l'architecture statique des composants. Toutes les autres étapes reposent sur cette dernière.

Au fur et à mesure de la démarche, on enrichit l'architecture fonctionnelle avec les exigences non-fonctionnelles. S'il existe un référentiel d'exigences (cahier des charges) alors l'Architecture Fonctionnelle est importante car elle permet de relier les composants identifiés dans la Configuration Architecturale et les exigences fonctionnelles du référentiel.

### QUESTION NUMERO 2

Expliquez en quoi le principe d'une architecture REST est très différent de celui d'une architecture WebServices.. L'architecture REST ne repose pas sur un standard d'infrastructure et de communication particulier (Middleware) car elle repose que sur le protocole http uniquement. Alors que les principes de l'architecture WebServices repose sur un standard rigoureux et normalisé (WSDL, SOA, UDDI).

REST est basé sur un modèle d'accès à des RESSOURCES à travers des adresses URI. Alors que WebServices est une architecture basée sur le modèle RPC (Remote Process Call) mais utilisant quand même que les standards du Web (HTTP, MIME, ...).

### QUESTION NUMERO 3

Citez au moins 5 types de composants de la couche métier d'une architecture JEE.

Précisez en 1 phrase le rôle de chacun de ces composants.

**JSP (Java Server Page)** : page de présentation pour un navigateur dans laquelle il est possible d'exécuter du code Java utilisé pour créer la page.

**Servlet** : code Java exécuter dans la session utilisateur du container Web en fonction d'une URL/URI.

**EJB Session Statefull** : Composant métier de traitement, avec état (permet par exemple de retarder la mise à jour des informations en base de données)

**EJB Session Stateless** : Composant métier de traitement, sans état (ordre immédiat de mise à jour en base de données, générer un évènement dans le cas d'une architecture à base d'évènement)

**Entity** : Composant métier de données (ou Objet métier) (permet d'être persisté en base de données)

**Message Driven** : Composant métier permettant d'être à l'écoute de la génération d'un évènement dans un canal d'évènement)

**EJB Singleton** : Composant métier de traitement, unique dans le SI. Partagé par toutes les sessions utilisateurs.

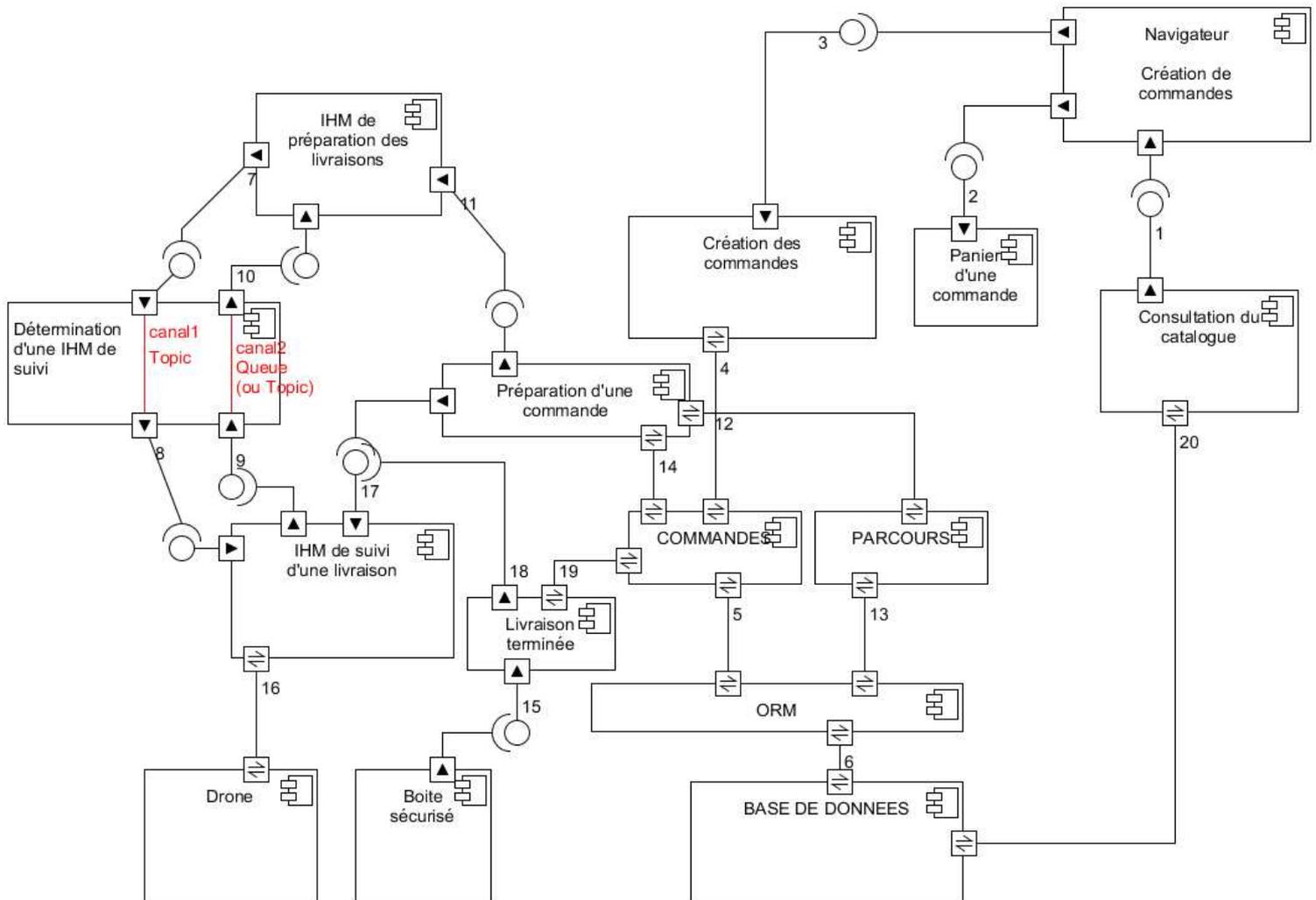
**WebService** : Création d'un webservice basé par exemple sur le protocole SOAP.

*Fin de la 1<sup>ère</sup> partie sans document*

## 2ème PARTIE – AVEC DOCUMENT (durée: 1h15)

## 3. PROBLEME (50 points)

## Question 1 : [30 points]



La création de commande se fait par un *Navigateur* qui utilise le composant de *Consultation du catalogue* qui accède directement à la base de données pour présenter le catalogue via le connecteur 1. Le composant *Consultation du catalogue* via [20] consulte tous les articles pouvant être commandés.

Via le connecteur [1], le composant *Panier d'une commande* stocke en mémoire la commande de l'utilisateur en cours de construction.

Une fois le panier validé, via le connecteur [2] le composant *Création des Commandes* permet de créer les *COMMANDES*.

Le composant *COMMANDES* sont les données de commandes qui sont persistées en base de données via un ORM ([5]).

Le composant *IHM de préparation des livraisons* utilise (via [11]) le composant *Préparation d'une commande* dont le rôle est d'envoyer à l'IHM quelles commandes sont prêtes pour livraison ([14]). Le composant *Préparation d'une commande* accède (via [12]) au *PARCOURS* afin de connaître quel parcours sera utilisé pour livrer la commande (table de correspondance entre l'adresse de livraison et les parcours géré en BD via [13]).

Ensuite l'*IHM de préparation des livraisons* utilise le composant *Détermination d'une IHM de suivi* pour déterminer quelle IHM de suivi est disponible (via [7]). Pour cela il notifie un évènement ("dispo?") ([8]) à toutes

les IHM de suivi. Sur réception de cet évènement, les IHM disponibles envoient un évènement ("dispo #identIhmSuivi ") à l'*IHM de préparation des livraisons* qui prendra, par exemple, la première disponible. L'*IHM de préparation des livraisons* via la *Préparation d'une commande* envoie à l'*IHM de Suivi* concerné le parcours que le drone doit réaliser (via [11] puis [17]).

Le rôle de l'*IHM de suivi d'une livraison* est de suivre le Drone et le piloter si nécessaire (connecteur [16] bi-directionnel).

Une fois que le drone a déposé son paquet dans la *Boîte sécurisée*, ce dernier prévient l'IHM de suivi (via [15]) qui va utiliser le composant *Livraison terminée* (via [18]) pour que ce composant met à jour la commande ([19]) et prévient par mail le client de la commande (l'adresse mail est dans la commande).

**Question 2 : [20 points]**

On va utiliser une architecture JEE pour gérer les composants suivants qui représentent la couche métier :

*Création des commandes*

*Panier d'une commande*

*Préparation d'une commande*

*Détermination d'une IHM de suivi*

*Livraison terminée*

COMMANDES

PARCOURS

ORM

Pour la gestion du catalogue, le composant *Consultation du catalogue* est composé de plusieurs pages JSP qui font des requêtes SQL pour présenter le catalogue à l'utilisateur et choisir les articles.

A chaque choix d'un article, le navigateur via un servlet gère le panier qui est une EJB Session Statefull qui garde en mémoire la liste des articles commandés.

Le navigateur valide le panier via un servlet et l'EJB Session Stateless *Création des commandes* qui met à jour les données de COMMANDES.

Le composant *Préparation d'une commande* est un EJB Stateless pour récupérer les commandes à livrer et pour récupérer le parcours d'une commande.

Le composant *Livraison terminée* est un EJB Stateless qui finit la commande en envoyant un mail et mettre à jour la commande terminée.

Le composant *Détermination d'une IHM de suivi* est le JMS qui composé de 2 canaux d'évènement. Le canal en mode Topic. Chaque *IHM de suivi d'une livraison* à un Driven Message : connecteur [8] pour recevoir l'évènement "dispo?".

Le canal 2 en mode queue. L'*IHM de suivi d'une livraison* pousse un évènement (via [9]) "dispo #ident" s'il est dispo.

Cela veut dire que le composant *IHM de préparation des livraisons* (via un thread) regarde régulièrement quelle IHM est disponible via [10].

Ces composants EJB sont directement utilisés par les IHM de préparation et de suivi car ces IHM sont des applications standalone et peuvent utiliser directement les EJB Session.

Les données COMMANDES et PARCOURS sont des Entity afin de profiter de la persistance automatique de ces informations dans la base de données.

Les boîtes sécurisées réparties sur chacun des bâtiments sont des clients de WebServices créés sur le serveur afin d'avoir une ligne sécurisée http entre ces boîtes et le serveur. Le connecteur [15] est un connecteur WebServices qui utilise le composant *Livraison terminée*.

**Fin du sujet**