

**Master 2 : Option Architecture Distribuée**  
**Module : Architecture Orientée Service (SOA)**

**Exercice 1 : (7p)**

Choisissez la/les bonne(s) réponse(s) :

**1. En SOA, qu'est-ce qu'un service ?**

- a) Une base de données partagée
- b) Une unité fonctionnelle autonome accessible via une interface
- c) Un composant matériel
- d) Un programme exécuté localement uniquement

**2. Quel principe SOA permet de modifier un service sans impacter les autres ?**

- a) Couplage fort
- b) Orchestration
- c) Couplage faible
- d) Centralisation

**3. Quel rôle joue le contrat de service ?**

- a) Définir l'implémentation interne du service
- b) Décrire l'interface et les règles d'utilisation du service
- c) Gérer les performances réseau
- d) Stocker les données du service

**4. Dans une architecture REST, une ressource est identifiée par :**

- a) Un port réseau
- b) Un identifiant binaire
- c) Une URI
- d) Un fichier XML

**5. Quelle méthode HTTP est généralement utilisée pour créer une ressource ?**

- a) GET
- b) PUT
- c) POST
- d) DELETE

**6. En SOA, l'hétérogénéité fait référence au fait que :**

- a) Tous les services utilisent la même technologie
- b) Les services peuvent être développés avec des langages, plateformes et systèmes différents
- c) Les services sont exécutés uniquement sur un même serveur
- d) Les services ne peuvent communiquer qu'en mode synchrone

**7. Quel est l'avantage principal de SOAP par rapport à REST ?**

- a) Simplicité de mise en œuvre
- b) Support natif de standards avancés (sécurité, transactions)
- c) Utilisation exclusive de JSON
- d) Absence de surcharge

**8. Quel élément WSDL associe une opération à un protocole concret ?**

- a) PortType
- b) Message
- c) Binding
- d) Types

**9. Dans SOAP, le rôle de l'enveloppe (Envelope) est de :**

- a) Définir la base de données

- b) Encapsuler le message échangé
- c) Identifier les services dans UDDI
- d) Décrire l'interface graphique

**10. Une architecture orientée événements (EDA) se base principalement sur :**

- a) Des appels synchrones
- b) Des échanges par événements asynchrones
- c) Des transactions distribuées
- d) Des bases de données centralisées

**11. Quel est l'intérêt principal de combiner SOA et EDA ?**

- a) Réduire le nombre de services
- b) Améliorer la scalabilité et la réactivité
- c) Supprimer les contrats de service
- d) Remplacer SOAP

**12. Dans le Web sémantique, une ontologie permet de :**

- a) Décrire la structure XML
- b) Donner un sens partagé aux données
- c) Accélérer le réseau
- d) Générer du code

**13. Quel langage est utilisé pour représenter les relations sémantiques entre ressources ?**

- a) HTML
- b) RDF
- c) WSDL
- d) SOAP

**14. Quel est l'objectif final des Web Services sémantiques ?**

- a) Améliorer l'interface utilisateur
- b) Automatiser la découverte et la composition des services
- c) Remplacer les services Web classiques
- d) Réduire la consommation mémoire

**Exercice 2 : (6p)**

On considère un système distribué basé sur une architecture orientée service (SOA), dans lequel les services communiquent principalement par des appels directs synchrones.

Avec l'augmentation du nombre d'utilisateurs, il devient nécessaire d'intégrer un nouveau service de supervision chargé de collecter et traiter les événements produits par les autres services.

1. Quels problèmes d'évolution et de performance peuvent apparaître lors de l'intégration de ce service de supervision dans une SOA classique ?
2. Expliquez comment une architecture orientée événements (EDA) peut être utilisée pour améliorer la communication entre les services existants et le nouveau service.
3. Donnez trois bénéfices concrets de l'utilisation d'une EDA dans ce contexte.

**Exercice 3 : (7p)**

Soit le code WSDL suivant, décrivant un service Web de gestion des étudiants.

1. Décrivez les différentes parties de ce code WSDL.
2. Quelles sont les opérations exposées par ce service Web ?
3. Ajoutez une nouvelle opération dans ce code WSDL permettant de supprimer un étudiant à partir de son identifiant (ID).

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.example.com/StudentService/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="StudentService"
  targetNamespace="http://www.example.com/StudentService/">

  <wsdl:documentation>
    Definition of a web service for managing students.
  </wsdl:documentation>

  <wsdl:types>
    <xsd:schema
      targetNamespace="http://www.example.com/StudentService/">

      <xsd:element name="Student">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="ID" type="xsd:string"/>
            <xsd:element name="Name" type="xsd:string"/>
            <xsd:element name="Level" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="GetStudent">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="ID" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="GetStudentResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="tns:Student" minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="AddStudent">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="tns:Student"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="AddStudentResponse">
        <xsd:complexType/>
      </xsd:element>

    </xsd:schema>
  </wsdl:types>

  <wsdl:message name="GetStudentRequest">
    <wsdl:part name="parameters" element="tns:GetStudent"/>
  </wsdl:message>

  <wsdl:message name="GetStudentResponse">
    <wsdl:part name="parameters"
      element="tns:GetStudentResponse"/>
  </wsdl:message>

  <wsdl:message name="AddStudentRequest">
    <wsdl:part name="parameters" element="tns:AddStudent"/>
  </wsdl:message>

```

```

<wsdl:message name="AddStudentResponse">
  <wsdl:part name="parameters" element="tns:AddStudentResponse"/>
</wsdl:message>

<wsdl:portType name="StudentService">
  <wsdl:operation name="GetStudent">
    <wsdl:input message="tns:GetStudentRequest"/>
    <wsdl:output message="tns:GetStudentResponse"/>
  </wsdl:operation>

  <wsdl:operation name="AddStudent">
    <wsdl:input message="tns:AddStudentRequest"/>
    <wsdl:output message="tns:AddStudentResponse"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="StudentServiceSOAP" type="tns:StudentService">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="GetStudent">
    <soap:operation
      soapAction="http://www.example.com/StudentService/GetStudent"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="AddStudent">
    <soap:operation
      soapAction="http://www.example.com/StudentService/AddStudent"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

<wsdl:service name="StudentService">
  <wsdl:port name="StudentServiceSOAP">
    <binding="tns:StudentServiceSOAP">
      <soap:address location="http://www.example.org/StudentService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

*Bon courage*