

Les exercices de ce document sont de trois types (A,B,C) correspondant respectivement à des questions de cours, de synthèse, ou une mise en situation. Tous les exercices sont issus des examens des années précédentes, ils ne pourront pas être tous traités en TD, les exercices non traités constituent un entraînement pour l'examen.

Exercice 1 - A — (6pts)

1. Pensez-vous que le *pattern Object Factory* soit applicable avec JMS ?
2. Comparez l'activation et le *pattern Object Factory*.
3. Les transactions distribuées via le protocole de commit à deux phases implanté dans JTA peuvent-elles s'appliquer avec un intergiciel comme JMS ?

Exercice 2 - A — (6pts)

1. Expliquer pourquoi on ne peut pas qualifier CORBA d'intergiciel orienté objet.
2. Quels éléments permettent de distinguer les notions d'objets, de composants et de services ?

Exercice 3 - A — Cours et TP (10pts)

1. Quelles sont les notions développées par le paradigme des objets distribués ?
2. Quelle est la relation entre les serveurs d'applications et les objets distribués ?
3. Quel mécanisme mis en jeu dans les environnements objets distribués permet d'assurer la transparence à la localisation ?
4. Que signifie l'erreur "`port already in use`" au lancement du service RMI-registry ?
5. Quels outils permettent d'assurer une transparence de stockage, c'est-à-dire une indépendance vis-à-vis du SGBD (préciser leur fonctionnement) ?
6. Quels sont les spécificités de CORBA par rapport aux autres technologies objets distribués ?

Exercice 4 - A — (7pts)

1. Quelles sont les limites du service de nommage de Java RMI ?
2. En vous inspirant des idées développées pour les services de CORBA et de la réalisation de l'Object Factory vu en TP, proposer 4 nouvelles orientations du service de nommage de Java RMI. Spécifier l'architecture de ce nouveau service, décrire son interface.
3. On souhaite fédérer plusieurs services de nommage de ce nouveau type. Proposer une solution (architecture, interfaces).

Exercice 5 - B — (8pts)

On souhaite concevoir un méta-intergiciel pour permettre les interactions entre objets distribués Java et CORBA ainsi que l'envoi de messages via JMS. Le système doit être multi-plateformes et multi-langages. Plutôt que d'utiliser un point RMI-IIOP le concepteur préfère implanter la notion de proxy-objet c'est-à-dire un objet qui agit pour le compte d'un autre objet.

1. Dressez la liste de toutes les interactions possibles et donner pour chacune le langage avec lequel vous allez implanter le proxy objet. Faire des schémas.
2. Afin de fonctionner, le méta-intergiciel nécessite un service de nommage. Quelles sont les informations que vous allez stocker dans le service de nommage ?

3. Comment votre service de nommage va interagir avec les services de nommage de CORBA, Java RMI et JMS ?
4. Quel(s) langage(s) utilisez-vous pour spécifier votre service de nommage (donnez le code de spécification de l'interface) ?
5. Quels problèmes restent à résoudre ?

Exercice 6 - B — (10pts)

On souhaite simuler un MOM (*Message Oriented Middleware*) en utilisant Java-RMI.

1. Que proposez vous pour traiter les messages asynchrones ?
2. Comment décrire différentes catégories de messages, comment gérer les abonnements à des canaux de diffusion ?
3. Spécifier uniquement l'interface du serveur en Java-RMI.
4. Écrire le code d'un client en montrant une interaction avec le serveur pour illustrer l'aspect asynchrone.

Exercice 7 - B — (7pts)

Une entreprise de distribution de colis souhaite mettre en place pour ses clients un service objets distribués basé sur Java RMI. L'objectif est de permettre aux sociétés de vente en ligne d'imprimer eux même les étiquettes d'expédition à coller sur les colis. On suppose qu'au début de chaque mois l'entreprise envoie une facture à ses clients comprenant le montant des expéditions faites dans le mois précédent.

1. En supposant que les communications sont sécurisées, spécifier l'interface de ce service.
2. Écrire le serveur et le client.
3. Comment modifier le serveur pour en faire un service web ? Discuter des différentes stratégies.

Exercice 8 - A/B — (7pts)

1. Que signifie l'erreur "`port already in use`" au lancement du service RMI-registry ?
2. On souhaite développer un système comportant un client RMI qui manipule des comptes à travers un objet RMI serveur de compte. Le serveur RMI propose une interface pour créditer ou débiter un compte dont le numéro est passé en paramètre. Il assure la persistance des données au moyen d'une connexion JDBC sur un serveur ORACLE. Peut-on parler d'*object factory* pour désigner le serveur ? Justifier votre réponse et dans le cas d'une réponse négative, proposer une évolution de l'interface du serveur pour en faire un *object factory*, donner son interface Java.
3. Comment Hibernate permet-il d'assurer une transparence de stockage (c'est-à-dire une indépendance vis-à-vis du SGBD) ?
4. Comment le mécanisme d'activation de Java RMI peut-il interagir avec un service de persistance JDBC ou Hibernate ?

Exercice 9 - C — (6pts)

Une structure médicale spécialisée dans le suivi de patients à domicile met en place un système d'information pour collecter des données sur les patients au moyen de matériel médical spécialisé mis en place au domicile du patient (système de dialyse par exemple).

1. En supposant que les patients ont des connexions ADSL permanentes, proposer une architecture et spécifier quels intergiciels sont les plus appropriés ainsi que leur périmètre d'action et leur rôle.
2. Même question en supposant que les connexions des patients ne sont pas permanentes (connexion UMTS/3G par exemple).

3. On suppose maintenant qu'en zone rurale il n'y a pas de connexion utilisable mais que du personnel médical vient visiter le patient à intervalle régulier et collectent les données au moyen d'une clé USB. Proposer une nouvelle architecture et la décrire.

Exercice 10 - C — Mise en situation (7pts)

Une association de type AMAP (Association pour le Maintien d'une Agriculture Paysanne) souhaite offrir un service Internet innovant, à grande échelle. Pour ses adhérents producteurs, elle veut offrir un pack leur permettant d'installer, chez eux, un serveur web, un serveur d'applications connecté sur une ligne ADSL afin de vendre leurs produits en direct sur Internet tout en bénéficiant d'un service externe offert par le système d'information de l'association leur permettant d'expédier des colis.

Pour les adhérents clients, l'AMAP veut mettre en place un service central afin de leur permettre de rechercher, d'acheter des produits chez différents producteurs (éventuellement proche de chez eux) et de se les faire livrer ou de les réserver pour aller les chercher. Pour les produits frais, l'association sous-traite les livraisons à des services externes spécialisés.

Pour les adhérents, qu'ils soient clients ou producteurs, l'AMAP permet de suivre les commandes, les livraisons, les paiements et la facturation.

1. Sous la forme d'un diagramme des cas d'utilisation, définir les grandes fonctionnalités du système.
2. Quelles fonctionnalités vous paraissent difficiles à implanter ?
3. Identifier les problèmes à résoudre (d'un point de vue architecture de système d'information).
4. Proposer une architecture fonctionnelle (faire un schéma et le décrire).
5. Quels intergiciels allez-vous utiliser (justifier clairement votre réponse) ?

Exercice 11 - B — (9pts)

Répondre, en un maximum de 5 lignes par réponse, aux questions suivantes :

1. Quelle caractéristique de CORBA rend la construction de serveurs d'application CORBA difficile ?
2. Comment rendre JEE multi-langages (donnez au moins 2 pistes de solutions) ?
3. Peut-on utiliser JTA avec JMS ?

Exercice 12 - B — (11pts)

On souhaite définir un modèle de composants sur JEE permettant d'ajouter des caractéristiques non fonctionnelles aux composants existants. Par exemple ajouter la traçabilité, la sécurisation de tous les échanges, le cryptage des données, la gestion de version de données, etc.

1. Quelle méthode, technique ou patron de conception (*pattern*) permet, à partir des types existants de composants JEE, d'ajouter des caractéristiques non fonctionnelles.
2. Comment héberger les services non fonctionnels en dehors des serveurs d'applications métier. Faire un schéma de votre architecture logicielle, expliquer son principe en expliquant précisément le rôle des différents intergiciels (*middleware*)
3. Donner un exemple de votre proposition avec un bean de type session.

Exercice 13 - B — (7pts)

On souhaite offrir un mode d'interaction entre objets distribués, autre que l'invocation de méthodes à distance, en s'inspirant des MOM. Les objets doivent être autonomes et capables de répondre lorsqu'ils le souhaitent aux messages. Les messages ne sont pas adressés à des *queues* ou des *topics* comme en JMS mais directement aux objets concernés. L'implémentation doit se faire en utilisant Java RMI.

1. Spécifier l'architecture de votre système et donner les grandes règles de fonctionnement.

2. Proposer une interface pour traiter les messages. Cette interface sera le comportement commun inclus dans tous les objets utilisant cet intergiciel.
3. Comment spécifier le service de nommage de cet intergiciel ? Décrire son fonctionnement et inclure la définition du service dans une interface spécifique. Comment votre service de nommage sera intégré avec le registre RMI, comment sera-t-il utilisé par les objets ?
4. Comment prendre en compte les messages et effectuer leur traitement dans les objets ?
5. Dresser un bilan de votre solution (avantages par rapport aux autres intergiciels, inconvénients, difficultés d'implémentation des fonctionnalités spécifiées).

Exercice 14 - C — (7pts)

Un agence de prévention des risques environnementaux met en place un système d'information pour collecter et traiter des données issues de capteurs divers (vitesse du vent, niveau d'eau, mesure de polluant, etc.).

Certaines collectes sont automatiques lorsque les capteurs sont connectés à un réseau permanent. D'autres collectes font intervenir un technicien qui au moyen d'un réseau Wifi ad-hoc se connecte, sur place, à un réseau de capteurs. Toutes les données sont ensuite stockées dans plusieurs bases de données hébergées sur un site central.

Le traitement des données utilise des serveurs spécifiques ayant un accès direct aux bases de données. Les traitements sont écrits en C, Fortran, Python, Java et effectuent des analyses statistiques et des simulations.

Les résultats sont publiés au moyen d'une application Web. De plus, les usagers peuvent extraire des données au moyen d'une interface de requêtes de l'application Web. Certaines données sont mises à disposition d'entreprises par exemple pour le transport routier.

1. Spécifiez l'architecture du système d'information.
2. Quels intergiciels allez vous utiliser ? Justifier votre réponse pour chaque choix.
3. On suppose maintenant que le système est dupliqué dans chaque région et qu'il est possible d'avoir une vue globale des données produites. Quels sont les impacts sur l'architecture de votre solution ? Identifier les couplages forts qui peuvent freiner les évolutions, proposer des modifications en terme d'architecture, d'intergiciels ou de *patterns* pour diminuer les couplages.

Exercice 15 - A/B — (6pts)

1. Quels éléments ou principes mis en œuvre dans les serveurs d'applications et dans le modèle EJB permettent le découplage du code métier avec les technologies utilisées ?
2. Quelles sont les propriétés non fonctionnelles assurées par un intergiciel (donner des exemples pour chaque intergiciel étudié) ?

Exercice 16 - C — (8pts)

On souhaite développer un service de répartition de charge en Java RMI. La solution doit être *scalable* c'est-à-dire qu'on doit pouvoir ajouter un nouveau nœud simplement en le déclarant. On suppose que chaque nœud du cluster met à disposition un objet distribué **LoadAverage** qui propose trois méthodes pour connaître la charge de la machine (nombre entier) à 1 minute, 5 minutes et 10 minutes, nommées respectivement **Get1min**, **Get5min**, **Get10min**. Ces méthodes font appel à la méthode **System.LoadAvg** qui retourne un tableau de trois entiers (un entier pour chaque valeur de charge). Chaque nœud hébergera des objets distribués. On distingue deux modes de fonctionnement : le mode statique (tous les objets sont connus partout), le mode dynamique (l'objet sur lequel on invoque la méthode est envoyé sur le serveur le moins chargé). Les questions 1 et 4 nécessitent l'écriture de quelques lignes de pseudo-code Java, veillez à être précis.

1. Définir le service implanté par l'objet distribué **LoadAverage**. Comment enregistrer et nommer les objets dans l'annuaire pour respecter le caractère *scalable*. Donner un exemple de code d'utilisation.

2. Quel(s) principe(s) de Java RMI et/ou quel(s) pattern(s) utilisez vous pour implanter la répartition de charge sans avoir à spécifier le serveur sur lequel doit s'exécuter la méthode invoquée. Dans votre réponse, vous devez distinguer les deux modes (statique et dynamique).
3. Faire un schéma de votre architecture pour chacun des modes en montrant une séquence d'exécution (numéroter, légender les flèches qui représentent les interactions entre éléments lors de l'exécution).
4. Donner le pseudo code de votre mécanisme de répartition de charge.
5. Discuter des principes à implanter pour étendre votre solution afin de permettre la migration d'objet (en cours d'exécution) si la charge machine devient trop importante.

Exercice 17 - C — (6pts)

Une groupe industriel équipe ses ingénieurs qualité de tablettes Android. On suppose que les tablettes hébergent un environnement d'exécution Java EE complet et peuvent utiliser une base de données Java native. Chaque usine du groupe possède des lignes de fabrication et des dispositifs d'acquisition de données qui enregistrent, en continu, dans des bases de données ad-hoc les informations de production. Les ingénieurs vont d'usine en usine pour collecter (extraire), analyser les données de fabrication et améliorer les réglages de lignes de productions. Leurs tablettes servent à collecter les données et à envoyer les paramètres de réglages aux lignes de production. L'analyse de données est réalisée au siège du groupe.

1. Faire un schéma pour spécifiez l'architecture globale du système (en incluant plusieurs sites).
2. Quels intergiciels allez-vous utiliser ? Justifier votre réponse pour chaque choix.

Exercice 18 - A — (12pts)

1. Décrire l'injection de dépendance(s) et préciser ses avantages ainsi que les systèmes ou les technologies qui l'implémentent. Donnez un exemple de code.
2. Quel(s) composant(s) de JMS permet(ent) de traiter les environnements mobiles et l'asynchronisme ?
3. Que permettent les annotations des EJB 3 par rapport au modèle des EJB 2 ? Discutez en termes de qualité du logiciel, et notamment de couplage, de la pertinence de l'utilisation des annotations dans un environnement de serveurs d'applications. Donnez des exemples précis pour illustrer vos réponses.

Exercice 19 - B — (8pts)

On souhaite utiliser Java RMI pour effectuer une analyse sur une grande masse de données stockée dans le SGBD Oracle. Le traitement à effectuer est programmé PL/SQL et nécessite une exécution de plusieurs dizaines de minutes. Les clients sont des PC de type bureautique. Le serveur doit effectuer un rappel (*call-back*) au le client pour lui envoyer les données résultant de l'exécution du programme PL/SQL.

1. Faire un schéma de l'architecture logicielle, précisez et argumentez le choix d'autres technologies utilisées en plus de Java RMI. Précisez quels objets distribués seront enregistrés dans l'annuaire.
2. Spécifiez le code des interfaces des différents objets distribués.
3. Comment mettre en place un mécanisme de reprise sur erreur en cas de non disponibilité du client ?

Exercice 20 - A — (6pts)

1. Montrez en quoi le *pattern Object Factory* peut contribuer à la qualité du logiciel en favorisant son évolutivité en réduisant le couplage et l'impact des différentes technologies sur le code métier.

2. Le recours aux MOMs (JMS par exemple) est souvent induit par des considérations techniques (comme par exemple une connexion réseau temporaire ou des délais de réponse très longs), donnez un cas d'utilisation (métier) général impliquant un recours à un MOM. Vous pouvez vous appuyer sur des exemples concrets.
3. Qu'apportent les annotations dans le modèle des EJB 3 par rapport aux EJB 2. Quels en sont les inconvénients ?

Exercice 21 - C — (7pts)

Pour des événements sportifs, une société met en place des capteurs individuels sur les participants pour les positionner et pour récupérer différentes valeurs sur leur état physique à intervalle régulier. On suppose que ces dispositifs utilisent, pour communiquer, une technologie de réseau mobile incluant une couche TCP/IP. On suppose que l'équipement mis en place repose sur des smartphones Android. À partir d'une application Web, le grand public doit pouvoir suivre l'événement et obtenir des informations sur chaque participant.

1. Faire un schéma pour spécifiez l'architecture globale du système.
2. Quels intergiciels allez-vous utiliser ? Justifier votre réponse pour chaque choix.
3. En supposant que l'application Web repose sur des servlets et du code JSP, spécifier : l'architecture des composants EJB ainsi que le rôle de chaque composant et le ou les intergiciels qu'il utilise.

Exercice 22 - A — (10pts)

1. Décrire trois ou quatre caractéristiques propres à chacun des intergiciels (*middleware*) vus en cours.
2. Quel type d'EJB permet de traiter l'asynchronisme ?
3. Expliquer le principe d'activation de Java RMI. Quels problèmes permet-il de résoudre ? Quels sont ses liens avec la persistance des données ?
4. Donner un extrait de code montrant comment réaliser un call-back en Java RMI.

Exercice 23 - B — (10pts)

On souhaite réaliser une plateforme pour la gestion et l'analyse de grandes masses de données (*Big Data*) en utilisant des intergiciels existants.

1. Déterminer les intergiciels nécessaires, justifiez vos choix.
2. Spécifier l'architecture.
3. Donner un exemple d'utilisation.
4. Discuter des avantages et inconvénients des intergiciels étudiés en cours pour ce type de plateforme.

Exercice 24 - A — (6pts)

1. Quel(s) mécanisme(s) permet(tent) de mettre en place la transparence à la localisation dans les intergiciels étudiés (Java RMI, CORBA, JMS).
2. Comparer les intergiciels orientés messages (MOM - message oriented middleware) JMS et Kafka.
3. Comparer JNDI et rmiregistry dans le contexte de développement d'applications de systèmes d'information utilisant abondamment des bases de données.

Exercice 25 - B — (7pts)

On souhaite réaliser un mécanisme de réplication de données. Une base de données relationnelles est considérée comme le maître, c'est la seule à prendre en charge les requêtes d'insertion et de mise à jour. Les autres bases ne sont utilisées que pour exécuter les requêtes de lecture. On souhaite implanter ce mécanisme principalement avec RMI et JDBC. Les requêtes sont envoyées sous la forme de chaînes de caractères aux objets distribués RMI. Pour chaque type de requête,

une méthode est proposée (`send.insert`, `send.update`, `send.select`). Une requête peut être déléguée à un autre objet distribué par le maître. Les réplicats (esclaves) se sont enregistrés vers le maître qui lors d'un nouvel insert ou update leur envoie la requête (qu'ils peuvent exécuter en temps différé). Si un esclave n'est pas joignable, le maître garde la requête de mise à jour dans une file d'attente qui sera envoyée à l'esclave lorsqu'il sera de nouveau disponible.

1. Faire un schéma pour illustrer votre proposition et détailler son fonctionnement au travers d'un exemple d'interactions.
2. Comment se fait l'interaction entre les objets distribués maître et esclaves (nommer le mécanisme) ?
3. Spécifier les interfaces des objets distribués maître et esclaves.
4. Quel peut être l'apport d'un patron de type *Object Factory* dans ce système ?
5. Quel peut être l'apport du mécanisme d'activation dans ce système ?

Exercice 26 - C — (7pts)

Un laboratoire médical pilote une étude sur les mécanismes de propagation des maladies à grande échelle. Pour cela les utilisateurs de smartphone sont invités à installer une application qui enregistre dans une base de données la position de l'utilisateur, ils peuvent ajouter d'autres types de données acquises par différents capteurs ou objets connectés. En analysant les positions successives, il est possible de déterminer l'ensemble des personnes avec lesquelles un utilisateur donné a été en contact (direct ou pas) sur une journée ou plus. Si l'utilisateur est malade, il est invité à décrire sa maladie au moyen de l'application. Le laboratoire analyse les données avec des programmes (modèles de simulations) écrits en C/C++ et publie en continu des données sur son site web. Inversement le système peut envoyer des alertes aux utilisateurs si ils se trouvent dans une région où une maladie contagieuse est particulièrement répandue.

1. Faire un schéma pour spécifiez l'architecture globale du système, le décrire.
2. Quels intergiciels allez-vous utiliser ? Justifier votre réponse pour chaque choix.
3. En supposant que l'application Web repose sur des servlets et du code JSP, spécifier : l'architecture des composants EJB (leur type) ainsi que le rôle (fonctionnel) de chaque composant et le ou les intergiciels qu'il utilise.

Exercice 27 - A — (6pts)

1. Comparer les deux intergiciels orientés messages (MOM - *message oriented middleware*) JMS et Kafka.
2. Que ce soit dans Java RMI, JMS ou dans les modèles de composants EJB, expliquer les apports du *pattern ObjectFactory*. Donner des exemples d'utilisation.
3. Quels sont les composants essentiels à un serveur d'applications, les citer et décrire leur(s) rôle(s).

Exercice 28 - B/C — (7pts)

On souhaite centraliser les logs système de services installés sur des serveurs (apache, mail, firewall, SGBD, etc.) et les traces d'activité des applications, dans une base de données, afin de les exploiter avec des outils de *data-mining*. Une ligne de log contient généralement la date et l'heure, la source (le nom de l'application et adresse IP de la machine), un degré de criticité (warning, info, error, emergency, etc.) et un message détaillé.

L'administrateur des serveurs a écrit un programme en C qui lit les fichiers de log (situés dans le répertoire `/var/log`) et envoie les données sur une machine qui centralise les logs dans des fichiers. Ensuite, des scripts insèrent les données dans les tables d'une base de données Oracle.

On souhaite réécrire ce système en utilisant un intergiciel (*middleware*) et en conservant la base de données Oracle. Le programme qui extrait les données des fichiers de log doit être modifié pour appeler un objet CORBA qui générera des messages répartis entre différents objets RMI selon un critère (source ou criticité par exemple).

1. Faire un schéma pour décrire l'architecture du système et expliquer son fonctionnement au travers d'un exemple d'acheminement d'un message système (faites apparaître clairement les annuaires utilisés, les protocoles et les langages).
2. Spécifier l'interface IDL de l'objet CORBA.
3. Spécifier l'interface de l'objet Java-RMI. Expliquer comment se fera le passage de données vers la base de données.
4. Quels sont les inconvénients en cas de gros volume de données, comment les résoudre ?
5. Pour les logs applicatifs, on suppose que les applications sont hébergées par un serveur d'application JEE. Comment ajouter le mécanisme de log dans les applications sans qu'il ne soit trop intrusif. Comment se fera le dialogue avec le système que vous avez spécifié dans la question 1.
6. Quel peut-être l'apport d'un intergiciel orienté messages comme Kafka dans ce contexte, faire un schéma de la nouvelle architecture et illustrer avec quelques lignes de code l'utilisation de Kafka (inclus dans un des deux objets distribués — question 2 ou 3).

Exercice 29 - C — (7pts)

Un entreprise, installée dans une zone commerciale propose à ses clients un service de courses à la demande. Au moyen d'une application Web et mobile les clients peuvent effectuer une commande. Les salariés de l'entreprise vont ensuite dans les différents magasins pour chercher les produits et informent les clients lorsque leur commande est prête. Les salariés sont aussi chargés de relever les prix des produits dans les différents magasins de la zone commerciale pour alimenter la base de données de l'application. On suppose qu'ils disposent tous d'un téléphone portable Android et d'une communication 4G. On ne souhaite pas installer de base de données sur le téléphone (on gardera les données en mémoire). Chaque action des salariés doit générer un ou plusieurs messages qui sont envoyés aux serveurs de l'entreprise.

1. Faire un schéma pour spécifiez l'architecture globale du système, le décrire.
2. Quels intergiciels allez-vous utiliser ? Justifier votre réponse pour chaque choix.
3. En supposant que l'application Web repose sur des EJB 3, spécifier leur type ainsi que le rôle fonctionnel. Vous devez spécifier au moins 5 EJB différents et montrer leurs interconnexions.
4. L'entreprise est un succès et des conventions avec les magasins sont établies pour fournir un accès plus complet au produits référencés et obtenir des tarifs avantageux sur des achats en grande quantité. Comment modifier l'architecture ?

Exercice 30 - A — (7pts)

1. Quels sont les services proposés par un serveur d'applications (en citer au moins 3 et les décrire) ?
2. Qu'apporte un modèle de composants d'un point de vue technique et d'un point de vue génie logiciel ?
3. Comment simuler un intergiciel orienté messages avec un intergiciel objets distribués. Prendre le cas de Java RMI et de CORBA et montrer comment ils peuvent permettre de réaliser les différents éléments d'un intergiciel orienté messages (annuaire, client producteur, client consommateur et serveur de messagerie).
4. Comparer le mécanisme d'activation de Java RMI et la persistance.

Exercice 31 - B — (6pts)

On souhaite construire un système pour traiter les courriers électroniques (mails reçus par le serveur SMTP) d'un service après vente d'une grande entreprise. L'objectif est de trier les courriers électroniques à l'intérieur du système d'information pour les aiguiller vers des experts ou des applications de traitement automatique.

On souhaite utiliser l'API Java Mail pour accéder au serveur SMTP une seule boîte aux lettres est utilisée (`sav@monentreprise.com`), la fonction `getMail()` retourne un objet mail avec des attributs de type `String` correspondant aux champs `from`, `to`, `subject` et `content`.

Un algorithme de *machine learning* permet de classer les mails qu'il reçoit selon deux critères (topic et urgence). La liste des topics ou thèmes est une liste prédéfinie. Chaque mail obtient un score pour chaque topic. L'urgence est calculée par l'algorithme, c'est un nombre décimal compris entre 1 à 5 (5 pour très urgent). Cet algorithme devra être appelé au moyen de Java RMI.

Votre outil doit envoyer les objets mail à un serveur de messages JMS afin de les trier selon les topics. Les applications viendront ensuite consommer ces messages JMS.

1. Faire un schéma pour décrire l'architecture du système et expliquer son fonctionnement au travers d'un exemple d'acheminement et de traitement d'un message (faites apparaître clairement les annuaires utilisés, les protocoles, les intergiciels, etc.).
2. Spécifier l'interface RMI de l'algorithme de *machine learning*.
3. Donner l'extrait de code pour un appel à l'algorithme de *machine learning*.
4. Spécifier ou décrire précisément le fonctionnement de la production des messages JMS
5. L'ensemble des activités du système doit être tracé, comment inclure dans l'architecture une base de données pour tracer les échanges ? Quels protocoles ou patrons (*patterns*) seront utiles ?
6. Quel pourrait être l'apport d'un intergiciel orienté messages comme Kafka dans ce contexte ? Permet-il de satisfaire aux fonctionnalités attendues ?

Exercice 32 - C — (7pts)

Une entreprise construit des machines pour des applications bio-médicales. Elle possède un bureau d'études effectuant la recherche et le développement des nouvelles machines (spectromètres de masse, spectroscopie RMN, séquenceurs, etc.).

Les machines sont construites à partir de nombreux composants venant de sous-traitants. Le système d'information gère la conception, les tests, la configuration, les commandes, la livraison et la maintenance des machines.

Durant les phases de tests effectuées dans l'entreprise, de nombreuses données sont acquises (plusieurs centaines de Go par machine) et stockées pour être analysées. Lorsque les machines sont en production chez les clients, elles envoient quotidiennement des données (quelques Mo) à l'entreprise afin que celle-ci puisse anticiper leurs périodes de maintenance.

Pour ses clients, l'entreprise propose une application Web pour effectuer des commandes de machines, de consommables etc. Cette même application permet aux clients de suivre l'état de leurs machines et de planifier leur maintenance.

1. Faire un schéma pour spécifiez l'architecture globale du système, le décrire.
2. Quels intergiciels allez-vous utiliser ? Justifier votre réponse pour chaque choix.
3. En supposant que l'application Web repose sur des EJB 3, spécifier leur type ainsi que le rôle fonctionnel. Vous devez spécifier au moins 5 EJB différents des 3 types principaux et montrer leurs interconnexions.
4. L'entreprise utilise des algorithmes de data-mining, coûteux en temps d'exécution, exploitant les données des machines (acquises en phases de tests et production) pour identifier des lots de composants défectueux ou consommables. Les algorithmes sont écrits en Python et en C++. Quelles modifications devez vous faire dans votre architecture. Faire un schéma et préciser les technologies utilisées.

Exercice 33 - C — (6pts)

Une entreprise informatique travaillant pour des chambres d'agriculture régionales propose à des agriculteurs une plateforme de collecte et traitement des données. On suppose que tous les engins agricoles sont équipés de capteurs permettant d'avoir la position, la vitesse, le volume d'éléments récoltés ou déposés (engrais, graines, pesticides, etc.).

Si la zone de culture est couverte en 4G les engins envoient leurs données directement au système d'information de la chambre d'agriculture de leur région sous la forme de n-uplets, position, data, type de mesure, valeur mesurée. Sinon, de retour à l'exploitation, les données sont déversées dans un système qui se connecte au système d'information de la chambre d'agriculture régionale et intègre les données dans un SGBD. Les exploitants agricoles peuvent ensuite visualiser ces données au moyen d'une interface Web.

Les chambres d'agriculture envoient en continu une partie des données au ministère de l'agriculture et effectuent elles-même le calcul d'indicateurs macroscopiques, des prévisions avec des modèles plus ou moins complexes incluant des données météorologiques. Ces modèles sont écrits en Python, Java, Scala, etc.

Les analyses peuvent être visualisées avec l'interface Web de la plateforme, sous la forme de cartes et des recommandations peuvent être faites aux agriculteurs. Ces derniers peuvent aussi extraire des données sous la forme de fichiers qu'ils pourront ensuite manipuler directement. Pour sélectionner les données de ces fichiers, ils remplissent des critères (intervalle de date, type de données, attributs, etc.) depuis l'interface Web et reçoivent ensuite une notification avec un lien pour télécharger les données.

1. Faire un schéma pour spécifiez l'architecture globale du système, la décrire.
2. Quels intergiciels utilisez-vous ? Justifier votre réponse en indiquant pour chaque intergiciel la fonctionnalité qu'il supporte et les propriétés non fonctionnelles qu'il apporte.
3. En supposant que l'application Web repose sur des EJB 2 ou 3, spécifier leur type ainsi que le rôle fonctionnel. Vous devez spécifier au moins 5 EJB différents recouvrant les 3 types principaux et montrer leurs interconnexions/interactions.
4. Pour rentabiliser les développements, l'entreprise informatique propose d'unifier le système à l'échelle de plusieurs régions. Comment modifier l'architecture pour prendre en compte cette nouvelle orientation ? Donner plusieurs pistes et présenter les avantages de celle que vous retenez.

Exercice 34 - A — (6pts)

1. On considère une entreprise de commerce électronique qui gère la partie informatique matérielle, les développements logiciels de sa plateforme Web (*front* et *back office*), ses entrepôts et les expéditions au moyen de partenaires tiers.
L'entreprise est internationale et elle possède plusieurs centaines de milliers de clients et de fournisseurs. Dans ce cadre, comparer l'usage de Kafka et de JMS pour différentes fonctionnalités du système d'information que vous aurez identifiées. Bien mettre en évidence les avantages et inconvénients de ces deux intergiciels pour ces différentes fonctionnalités.
2. Définir les notions de conteneur (*container*) et de modèle de composants proposées dans les technologies de serveurs d'applications. Quels sont leurs apports vis-à-vis de la qualité du logiciel ?
3. Comment ajouter une ou plusieurs librairies (des fichiers `.jar`) dans un fichier de configuration ant (`build.xml`) ? Quelles sont les limites de l'outil ant ?

Exercice 35 - C — (6pts)

Une grande métropole souhaite aider ses habitants à optimiser leurs trajets quotidiens en voiture et éventuellement leur proposer des usages combinés de différents modes de transports. Pour ce faire, chaque usager du système s'inscrit sur la plateforme au travers d'une application Web et installe une application sur son smartphone. Le système enregistre tous les trajets des personnes et leur mode (voiture, à pieds, en bus, tram, etc.) ainsi que les horaires. Après plusieurs semaines d'utilisations, le système peut effectuer des recommandations de trajets et éventuellement re-router dynamiquement (par notification sur l'application mobile) les usagers en fonction des conditions de trafic. Les programmes d'analyses de données sont écrits au moyen de différents outils (langages de programmation Python, Spark/Scala, Java). Les données agrégées anonymisées sont publiées sur la plateforme et visibles à tout abonné. Elles peuvent être visualisées sous la forme de cartes.

1. Faire un schéma pour spécifiez l'architecture globale du système, le décrire.
2. Quels intergiciels allez-vous utiliser ? Justifier votre réponse pour chaque choix.
3. En supposant que l'application Web repose sur des EJB 2 ou 3, spécifier leur type ainsi que le rôle fonctionnel. Vous devez spécifier au moins 5 EJB différents des 3 types principaux et montrer leurs interconnexions.

4. Pour rentabiliser le système, les opérateurs proposent à des commerçants d'afficher des offres promotionnelles en fonction de la position prévue des usagers du système reçues (de 24h à 4h en avance) par notification sur l'application mobile ou bien par courrier électronique. Comment modifier l'architecture pour prendre en compte cette nouvelle fonctionnalité ?

Exercice 36 - B — (7pts)

On souhaite utiliser Java-RMI pour construire un intergiciel orienté messages à haute performance. Les *topics* peuvent être répartis sur plusieurs machines et les messages sont purgés à intervalles de temps réguliers (par exemple toutes les semaines). Si un topic est réparti sur plusieurs machines (nœuds d'un cluster), chaque message peut être répliqué plusieurs fois (*r* fois) pour des questions de tolérance aux pannes, de disponibilité et de performance. Pour réaliser ce système on utilisera des objets distribués Java-RMI. Un des nœuds contient un catalogue c'est-à-dire pour chaque topic le nombre de messages et l'id du dernier message, ainsi que les références aux objets distribués qui gèrent une partie des topics.

Clients producteurs et clients consommateurs se connectent au système mais n'ont pas à savoir qui gère quel topic.

1. Faire un schéma et décrire l'architecture du système. Expliquer son fonctionnement au travers d'un exemple d'acheminement et de traitement d'un message. Mettre en évidence le rôle des annuaires : celui de Java RMI (*rmiregistry*) et un annuaire propre au système.
2. Dans cette architecture, comment utilisez vous le *pattern object factory* ? Spécifier son interface ?
3. Votre système a besoin d'un annuaire spécifique pour gérer les informations sur les topics et les objets distribués qui gèrent les messages. Spécifier au moyen d'une interface RMI votre annuaire.
4. Spécifier l'interface des objets distribués présents sur chaque nœud et gérant un topic particulier.
5. Montrez pour une publication et une consommation de messages l'enchaînement des appels aux différentes méthodes des différentes interfaces. Vous pouvez représenter cela avec un diagramme de séquences.

Exercice 37 - A — (7pts)

1. Expliquer ce qu'est une référence distribuée dans les intergiciels orientés objets.
2. Expliquer la différence entre stub et stub instancié en Java RMI. Comment ces deux éléments sont utilisés dans un appel de méthode à distance ?
3. Qu'apportent les notions de conteneur (*container*) et de modèles de composants pour les serveurs d'applications et en général pour le développement d'applications ?
4. Peut-on réaliser une injection de dépendance avec CORBA ?
5. Même question avec des objets Java RMI et des messages JMS ? La question a-t-elle un sens ? Quels peuvent être les intérêts d'un tel mécanisme ?

Exercice 38 - B — (8pts)

On souhaite utiliser le patron *object factory* combiné avec Java RMI pour gérer de manière séparée : 1) des connexions à des bases de données Oracle et 2) des journaux d'activités sur les actions des utilisateurs.

Les programmes clients peuvent obtenir des objets de type **Connexion** puis soumettre des requêtes. De même, au moyen d'objet **Log** les programmes peuvent garder une trace sur un serveur des activités des utilisateurs c'est-à-dire des appels de méthodes¹. En revanche, les programmes clients ne gèrent pas directement les connexions aux SGBDs, c'est la *factory* qui les met à disposition en retournant une référence à un objet connexion, hébergé sur le serveur, qui acceptera l'envoi de requêtes. Les clients ne se connectent donc pas directement au SGBD. De même, pour le second type d'utilisation, les logs applicatifs sont stockés sur les serveurs.

1. La gestion des connexions aux SGBD et la journalisation sont deux fonctionnalités séparées.

1. Faire un schéma et décrire votre architecture logicielle. Expliquer son fonctionnement au travers d'un exemple. Mettre en évidence le rôle de l'annuaire Java RMI et des *factories*.
2. Donnez au moins trois arguments en faveur de l'utilisation du patron *object factory*.
3. Comment utilisez-vous le patron *object factory*? Spécifier son interface pour les deux cas.
4. Spécifiez l'interface des objets distribués délivrés par les *factories*.
5. Montrez comment utiliser les *object factory* depuis un client, donnez les extraits de code.
6. Comment généraliser l'implémentation du patron *object factory* afin qu'il puisse délivrer tout type d'objet et que les clients fassent une demande d'accès à une instance d'un type particulier.