

Examen GMI 3

Construction d'Applications Réparties

Le 22 janvier 2002 – 14h-17h

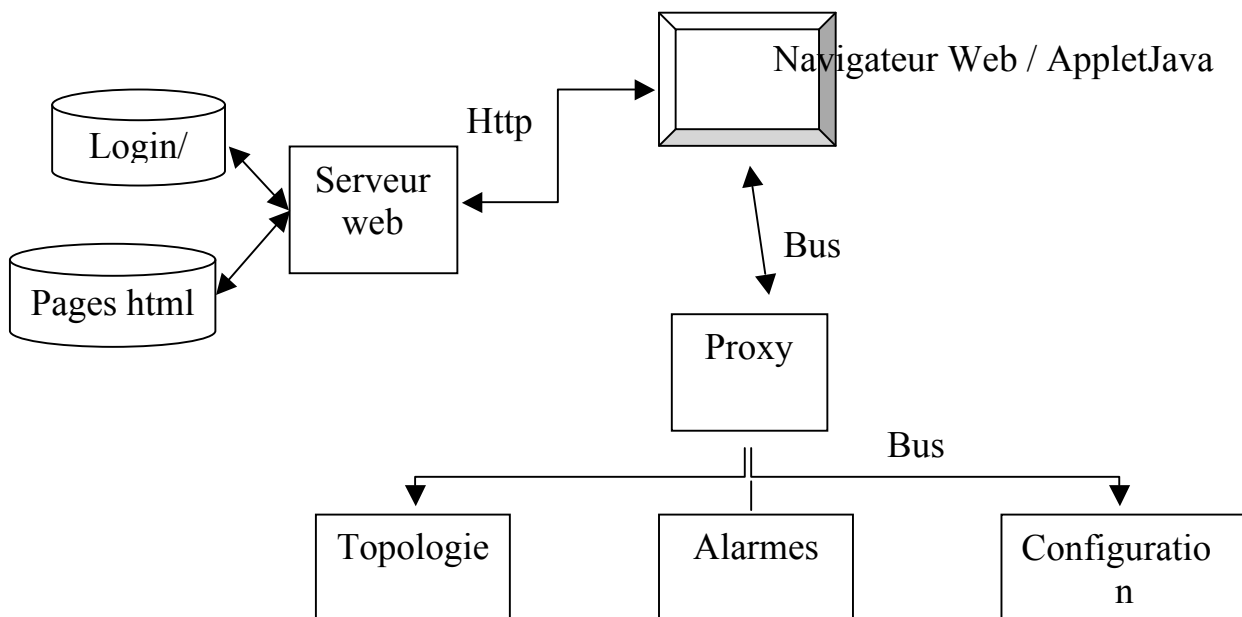
Documents autorisés

Lire le sujet entièrement avant de commencer

Justifier vos réponses

Une application d'administration réseau est accessible depuis l'extérieur de l'entreprise par les ingénieurs qui sont de garde chez eux le week-end ou le soir. Cet accès externe permet la surveillance du réseau ainsi quelques interventions de façon à maintenir le réseau dans un état de bon fonctionnement.

Le fonctionnement de l'application suit un modèle client-serveur. L'ingénieur est identifié par un serveur web lors de la demande de chargement d'une page de garde. Une fois l'ingénieur identifié, la page de garde est chargée sur son navigateur et permet le chargement et le lancement d'une applet Java (un programme en byte-code exécuté sur la machine virtuelle Java du client). L'applet Java gère par la suite le dialogue avec les différents éléments du réseau administré. Le dialogue entre l'applet Java et les différents éléments du réseau fonctionne au-dessus d'un bus CORBA. Pour protéger l'accès aux différents éléments du réseau, un proxy ou mandataire est installé à l'entrée du réseau. Ce dernier permet l'interrogation des différents modules de l'application d'administration par l'applet Java.



La première opération réalisée par l'applet Java installée sur le client consiste à récupérer sur le serveur Web la référence IOR permettant l'accès au proxy via CORBA.

Q1. Rappeler ce qu'est une référence IOR ? (1 point)

Q2. Habituellement, ces références, associées aux noms des services rendus (ici l'application d'administration) sont gérées par des serveurs de noms. On interroge le serveur de noms pour connaître la référence IOR d'un service proposé sur le bus CORBA. A votre avis, pourquoi l'IOR est-elle ici détenue par le serveur Web ? Est-ce pour des raisons de sécurité ou de robustesse de l'application ? justifiez votre réponse. (2 points)

Les objets CORBA Topologie, Alarme et Configuration proposent un ensemble de services accessibles via le proxy. L'interface IDL complète du proxy contient les définitions de ces trois objets.

Q3. Pour l'objet `Topologie`, il s'agit d'un ensemble de méthodes permettant de découvrir le réseau. 3 méthodes dans cet objet sont proposées :

- La méthode de consultation des nœuds qui rend la liste des nœuds du réseau appelée `consult_liste_noeud`. Un nœud est désigné par son nom (ex « Asimov » ou « Pouchan »).
- La méthode de consultation des caractéristiques d'un nœud appelée `consult_noeud`. Cette méthode, à partir du nom du nœud donnée en paramètre d'entrée sous forme de chaîne de caractères (ex « Routeur Asimov »), rend le numéro IP du nœud (« 163.173.128.60 »), ainsi que le type de machine (« Routeur CISCO »). Ces deux paramètres sont décrits sous forme de chaînes de caractères. Si le nœud n'existe pas, une exception est levée (appelée `NœudInexistant`).
- La méthode de structuration du réseau permet de connaître l'ensemble des machines se trouvant sur un sous-réseau. La méthode appelée `structure_reseau` donne à partir d'une adresse IP de sous réseau (ex « 163.173.128.0 »), la liste des machines se trouvant sur ce sous réseau. Si l'adresse IP donnée en paramètre d'entrée ne correspond pas à une adresse du sous-réseau, une exception est levée (appelée `SousRéseauxInexistant`).

Donner la description de l'interface de cet objet CORBA `Topologie` en IDL. (2 points)

Q4. L'objet `Configuration` permet d'ajouter des éléments dans le réseau et modifier les informations concernant les nœuds. Les trois méthodes de cet objet sont :

- La méthode `ajout_noeud` qui a pour paramètres en entrée le nom du nœud, son adresse IP et le type de matériel. Si le nom existe déjà ou l'adresse IP est déjà utilisée, une exception est levée (appelée `NoeudExistant`)
- La méthode `modif_noeud` qui a pour paramètres en entrée le nom du nœud, ainsi que les nouvelles valeurs des adresses IP et du type de matériel. Si le noeud n'existe pas, une exception est levée (appelée `NoeudExistant`).
- La méthode `retire_noeud` permet de retirer un nœud de la configuration. Un paramètre en entrée contient le nom du nœud à retirer. Si le noeud n'existe pas, une exception est levée (appelée `NoeudExistant`).

Donner la description de l'interface de cet objet CORBA `Configuration` en IDL. (2 points)

Q5. L'objet `Alarme` permet de récupérer la liste des alarmes levées et de les acquitter. Les deux méthodes sont :

- La méthode `liste_alarme` rend la liste des alarmes levées, ainsi que leur niveau.

- La méthode `acquitte_alarme` permet d'acquitter une liste d'alarmes. La liste d'alarmes à acquitter est passée en paramètre d'entrée.

Donner la description de l'interface de cet objet CORBA `Alarme` en IDL. (2 points)

Q6. De manière à ce que les alarmes remontent plus rapidement que lors d'une demande explicite de la part de l'ingénieur réseau, une instance d'un objet `AlarmeRapide` est insérée dans l'applet Java installée sur le navigateur Web. Son IOR est enregistrée auprès du proxy (qui se chargera de l'appel) lors de son instanciation.

Lors d'une levée d'alarme, une méthode `levée_alarme` sur cet objet est appelée. Il s'agit d'une méthode asynchrone qui a pour paramètre d'entrée l'alarme levée ainsi que son niveau. Elle permet de prévenir l'ingénieur de manière interactive d'un problème sur le réseau. Donner l'IDL de cet objet. (2 points)

Q7. Pour cet objet `AlarmeRapide`, quelle est l'entité ou le module qui joue le rôle de client, même question pour le rôle de serveur. Justifiez votre réponse. (1 point)

Q8. L'interface IDL CORBA du proxy, visible par les clients, correspond à l'interface des trois objets serveurs (`Alarme`, `Configuration`, `Topologie`). Lorsqu'un client souhaite accéder à un des trois objets, il s'adresse au proxy dont il connaît l'IOR. Celui-ci sert de point d'entrée vers les différents objets et retransmet la requête vers l'objet concerné. C'est en fait le proxy qui sert de serveurs de noms dans cette architecture. Si l'on ajoute un nouveau service, le proxy doit être capable d'enregistrer cet objet. De même il doit le retirer de sa liste lorsque celui-ci est interrompu (par exemple pour des opérations de maintenance). On ne considère pas les pannes. Donner l'interface IDL CORBA du proxy permettant de gérer les serveurs? (2 points)

Q9. Quel est le point faible de cette architecture client/proxy/modules d'administration en termes de sûreté de fonctionnement? (2 points)

Q10. L'intérêt de ce proxy réside dans le fait qu'il constitue un point de passage obligé dans les communication entre le client et les différents modules. Ce point de passage va permettre d'ajouter des fonctionnalités de sécurité telles que le cryptage des messages par une couche SSL, mais aussi l'enregistrement des suites d'échanges entre client et modules dans des journaux ou encore le contrôle des séquences des opérations appelées par le client. En cas de séquence non autorisée, le proxy pourra arrêter le dialogue entre le client et le module. A votre avis, pour mettre en place ces fonctionnalités, faut-il changer l'interface du proxy ? Justifier votre réponse (1 point)

Q11. On souhaite enregistrer la suite des échanges, c'est-à-dire gérer un journal sur le proxy. On y enregistre donc chaque appel de méthode avec ses paramètres et son résultat. Ces enregistrements sont datés. Donner l'organisation du code à mettre en place pour réaliser la journalisation dans le proxy (3 points).

- Comment est organisé l'enregistrement de ces informations ?
- Quelles sont les méthodes nécessaires ?
- Quel est le code de ces méthodes ?
- Où ces méthodes sont-elles utilisées ? Donner un exemple d'utilisation de ces méthodes.