



RPC/RMI vs MOM/JMS

Comparison de Styles de Communication

Ada Diaconescu

ada.diaconescu@telecom-paristech.fr





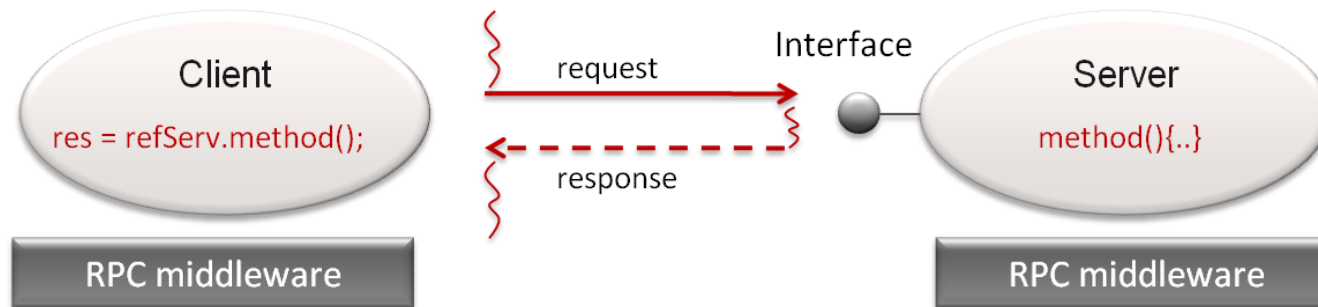
Rappel

RMI & JMS



RPC / RMI – Rappel

■ Architecture et caractéristiques générales

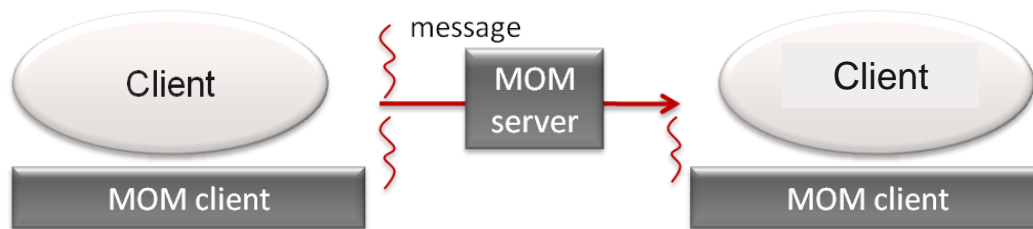


- **Couplage fort**
 - Le client dépend de l'interface du serveur
 - Le client doit « connaître » le serveur (Référence)
- **Dépendance temporelle**
 - Le client et le serveur doivent être simultanément disponibles
- **Communication synchrone/bloquante (habituellement)**
 - Le client est bloqué entre l'envoi de la requête et l'arrivée de la réponse



MOM / JMS – Rappel

■ Architecture et caractéristiques générales

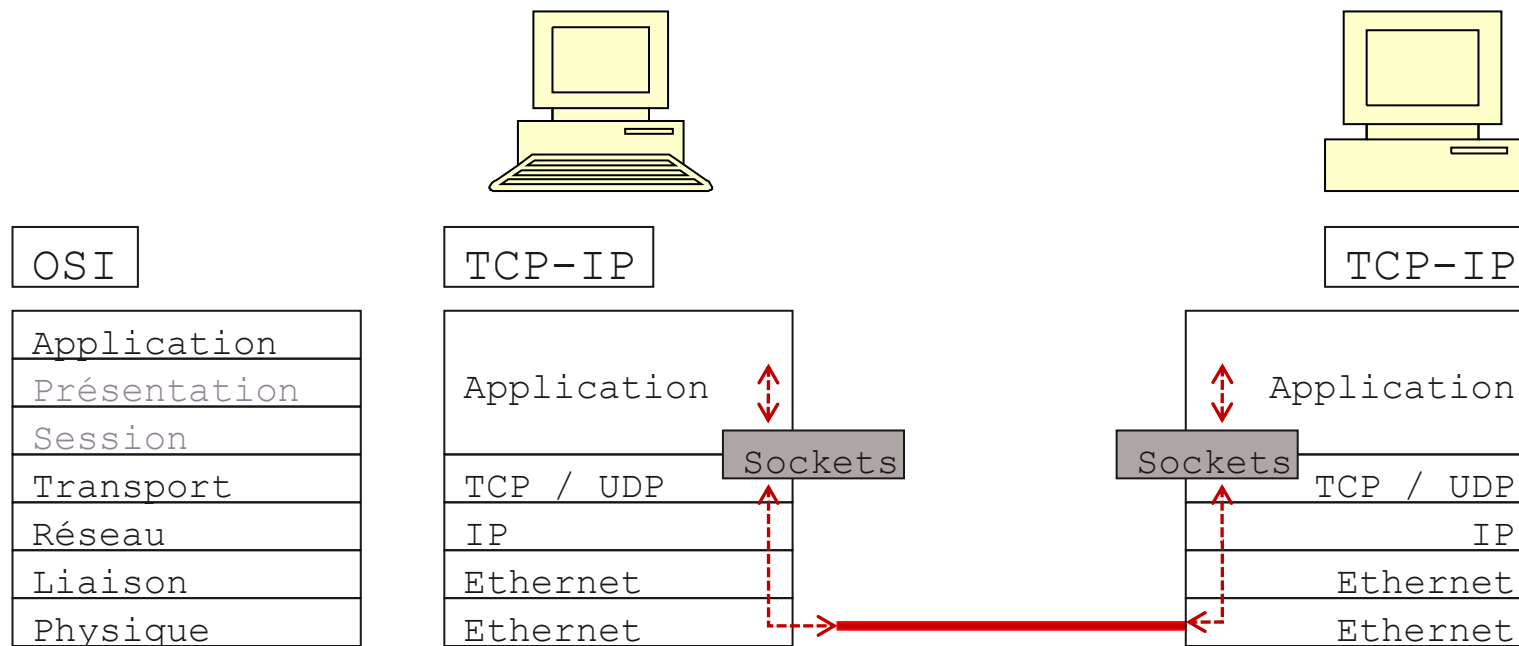


- Couplage faible (possible) => remplacement facile
 - Un composant (client) ne dépend pas de l'interface des autres composants (clients)
 - Un client n'est pas obligé de connaître les autres clients (si utilisation d'un serveur MOM)
- Pas de dépendance temporelle => pas besoins de disponibilité simultanée
 - Les différents composants de l'application ne sont pas toujours disponibles simultanément
 - Sauf communication transitoire (ex : par Topic, et sans suscription durable)
- Possibilité de communication asynchrone / non-bloquante
 - Le producteur n'est pas bloqué après l'envoi de message (sauf attente d'acquiescement)
 - Le consommateur peut recevoir les messages de façon bloquante ou non-bloquante



Communication par message

- Au niveau réseau, toute communication se fait « par message »
- Au niveau application, on peut avoir plusieurs modèles de communication (ex : via l'intergiciel)



Communication par message -- au niveau application --

Vue d'ensemble - caractéristiques

- Synchrone vs. Asynchrone
- Persistant vs. Transitoire
- Unicast vs. Multicast
- Mode « push » vs. « pull »
- Modèles de communication :
Requête/Réponse, Point à Point, Publication / Abonnement, ...
- Systèmes de routage des messages
- Bibliothèque de traitement de messages
- ...

=> Plusieurs modèles de communication par message

- Implantant diverses combinaisons des aspects indiqués auparavant



Communication par message

Communication synchrone vs. asynchrone

■ Synchrone / bloquante

- L'émetteur reste bloqué jusqu'à ce que le destinataire acquitte la réception du message (« Acknowledgement »)

■ Asynchrone / non-bloquante

- L'émetteur continue de s'exécuter après avoir soumis le message pour la transmission



Communication par message

Données persistantes vs. transitoires

■ Communication persistante (« persistent »)

- Le serveur MOM conserve le message jusqu'à ce qu'il soit transmis au récepteur; le message n'est jamais perdu ou effacé
- => pas de dépendance temporelle : l'émetteur et le récepteur *ne* sont *pas* obligés d'être présents en même temps

■ Communication transitoire (« transient »)

- Le serveur MOM conserve le message seulement pendant l'exécution simultanée de l'émetteur et du récepteur
- => dépendance temporelle : l'émetteur et le(s) récepteur(s) doivent être présents en même temps



Communication par message

Unicast vs. multicast

■ « Unicast »

- Le message est envoyé à un seul destinataire

■ « Multicast » - Diffusion (ou group)

- Le message est distribué à plusieurs destinataires



Communication par message

Mode « push » vs. « pull »

■ Mode « push »

- L'émetteur envoie le message au récepteur
- Ex. en appelant une méthode de « callback » du récepteur

■ Mode « pull »

- Le récepteur va chercher le message chez l'émetteur
- Ex. :
 - périodiquement;
 - en restant bloqué jusqu'à ce qu'un message devient disponible;
 - sur notification de l'émetteur (« push » et « pull »)

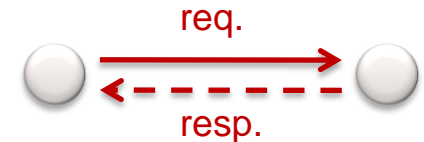


Modèles de communication par message

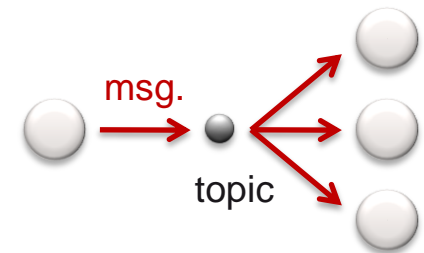
■ Point à Point (« Point to Point »)



■ Requête / Réponse



■ Publication/Abonnement (« Publish/Subscribe »)



■ ...



Modèles de communication par message (ex 1)

■ Point-à-Point (“Point-to-Point” - 1 : 1)

- Chaque message est stocké dans une file (« Queue ») jusqu’à ce que le destinataire le lise
- Chaque message est consommé une seule fois (par un seul destinataire)
- Pas de dépendance temporelle entre émetteur et destinataire du message
- Le destinataire peut acquitter les messages reçus

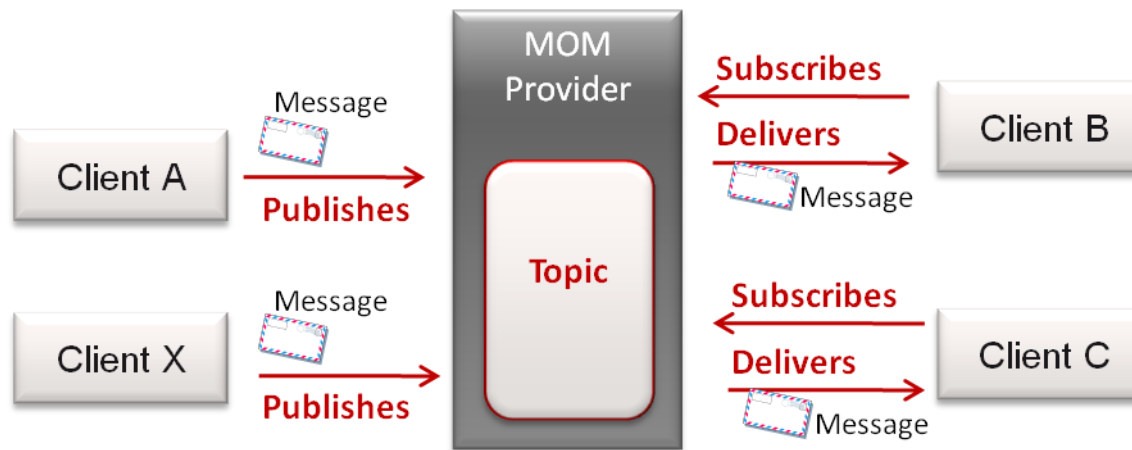




Modèles de communication par message (ex 2)

■ Publication/Abonnement (“Publish/Subscribe” - * : *)

- Le sujet (« Topic ») gère l'envoi de messages pour un ensemble de lecteurs abonnés
- Découplage entre les « publishers » et les « subscribers »
 - Les fournisseurs n'ont pas besoins de connaître les consommateurs
- Dépendance temporelle
 - Un client ne peut lire un message qu'après s'être abonné à un topic,
 - Un client abonné à un topic doit continuer à être actif pour recevoir les message du topic.





Caractéristiques RMI

Communication

■ Synchrone

- le client est bloqué en attendant la réponse du serveur

■ Transitoire

- besoin de disponibilité simultanée du client et du serveur

■ Unicast

- Le client envoie la requête à un seul serveur à la fois

■ Mode « push »

- le client pousse la requête vers le serveur ; le serveur pousse la requête vers le client

■ Modèles de communication :

- requête/réponse



Caractéristiques JMS

Communication

- Producteur asynchrone
- Consommateur synchrone ou asynchrone
- Persistante par les « Queue »
- Transitoire par les « Topic » (sauf « durable » configuration)
- Unicast par les « Queue »
- Multicast par les « Topic »
- Mode « push » entre les producteurs et le MOM
- Mode « push » ou « pull » entre le MOM et les consommateurs
- Modèles de communication : point-to-point ou publish/subscribe



Intergiciels

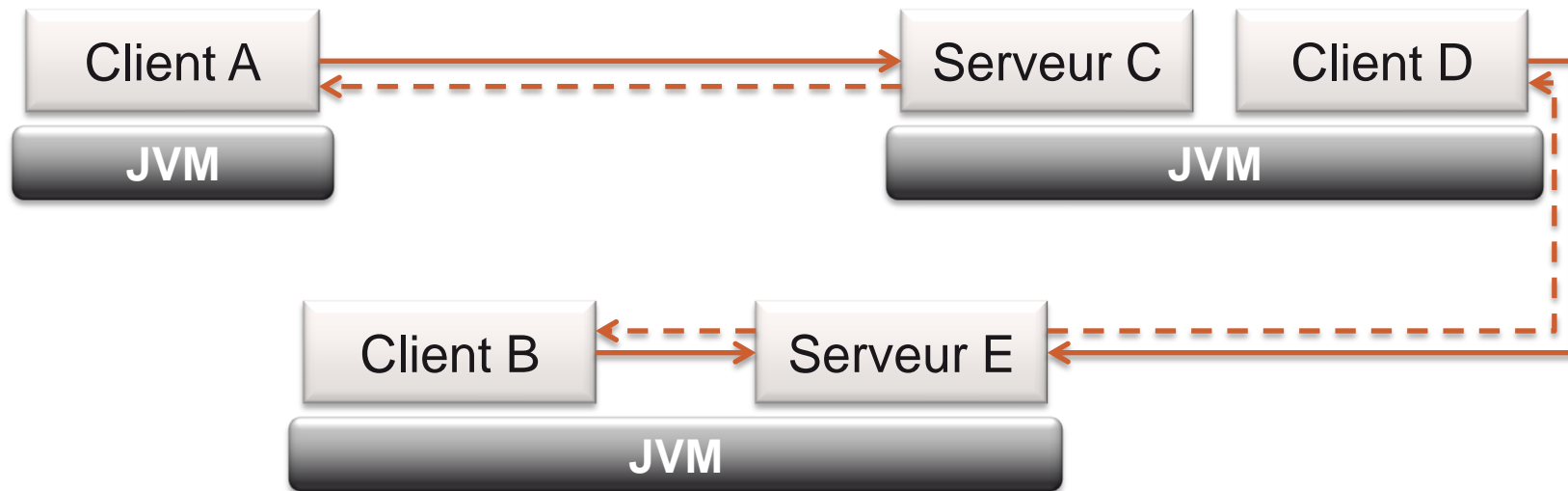
TOPOLOGIES



Topologie RPC / RMI

■ Décentralisée :

- Une instance d'intergiciel sur chaque machine du système repart
- Ex : pour RMI : dans la JVM de chaque application Java
- Ex : Java EE : sur chaque machine, mais serveur JNDI centralisé (possible).





Topologies pour le Fournisseur de MOM

- Différentes topologies possibles pour le Message Broker
 - Centralisée
 - Décentralisée
 - Hybride

- Chaque produit MOM peut utiliser une ou plusieurs topologies



MOM – topologie centralisée (« hub & spoke »)

■ Serveur central de gestion de messages

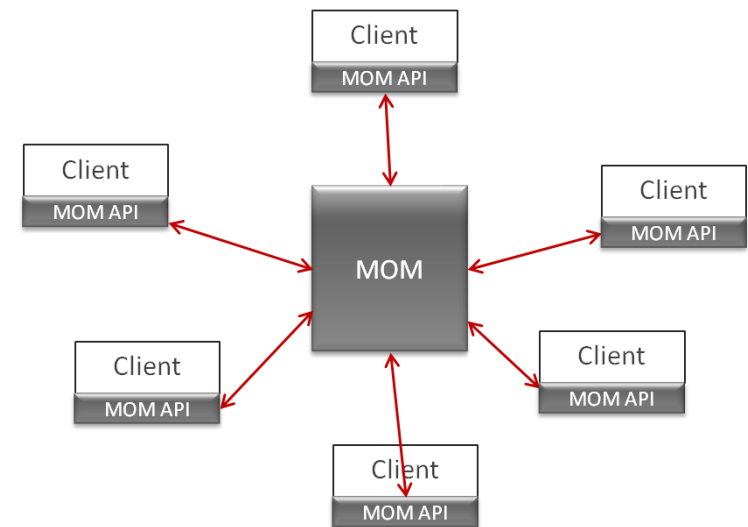
- Routage, stockage, transactions, ..

■ Avantages

- Découple les clients, qui ne voient que le serveur - l'ajout ou l'enlèvement d'un client n'a pas d'impact sur les autres clients

■ Inconvénients

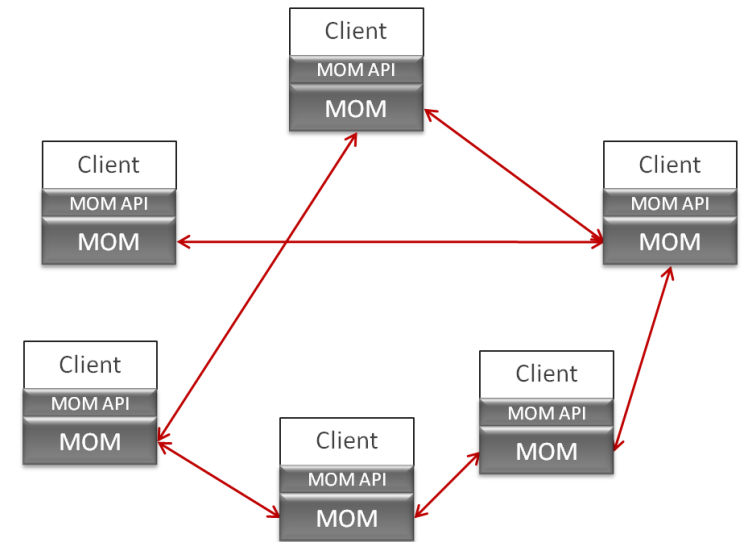
- Engendre un trafic réseau important
- Crée un goulot d'étranglement:
 - Point unique d'échec (« unique point of failure »)
 - Problèmes de passage à l'échelle
 - Possibilité de performance réduite (latence)





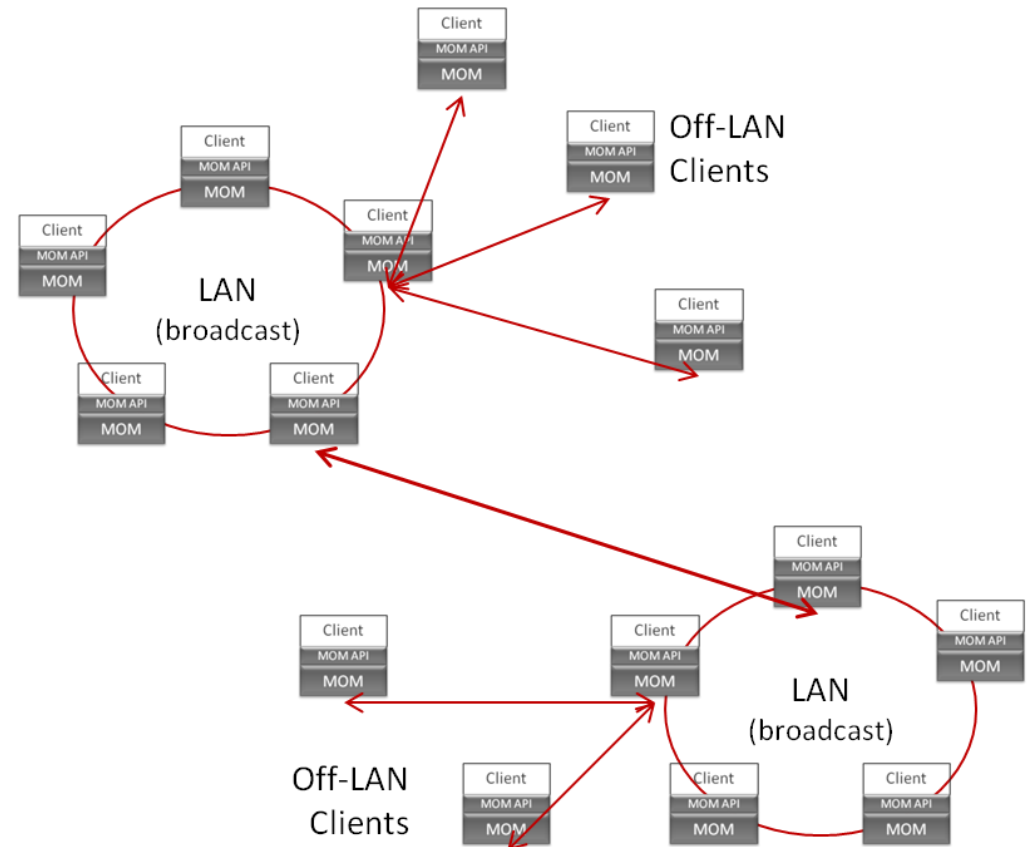
MOM – topologie décentralisée

- Une instance de MOM est installée chez chaque Client - pas de serveur central
 - Stockage, transactions, sécurité, ...
 - Routage de messages
 - Ad-hoc – entre les MOMs
 - Basé sur le protocole réseaux existant - ex. IP-Multicast
- Avantages
 - Distribution des fonctions du MOM entre les serveurs – ex. persistance, sécurité, ...
- Inconvénients
 - Problèmes potentiels d'interopérabilité – ex. différents vendeurs ou versions de MOM
 - Clients plus lourds, duplication des fonctions



MOM – topologie hybride

- Combine les deux architectures précédentes
- Combine les avantages et les inconvénients



MOM – exemple de topologie pour Joram

- Plusieurs Serveurs
- Chaque Client se connecte à un des Serveurs
- Chaque Client peut communiquer avec tout autre Client, quel que soit le Serveur auquel ils se sont connectés

