

# MOM

(messages oriented middleware)

Samuel Tardieu  
École Nationale Supérieure des Télécommunications  
[sam@enst.fr](mailto:sam@enst.fr)



# Caractéristiques

- ' Un MOM
  - ' Est un système clients/serveurs
  - ' Est un système asynchrone d'échange de messages
    - ' Note: la plupart des implémentations fournissent également une interface synchrone
  - ' Utilise des files de messages
  - ' Possède un « message broker »
  - ' Fournit un API, en général dans plusieurs langages et pour plusieurs systèmes

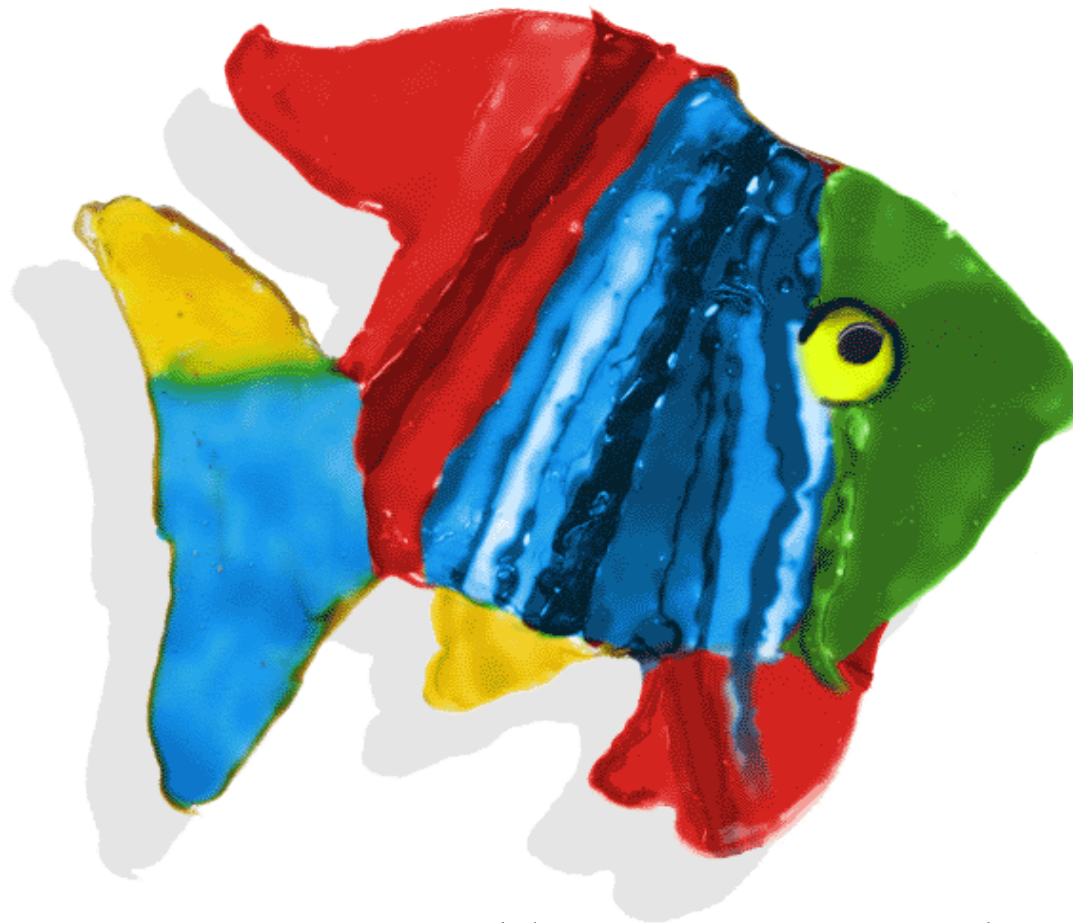


# Utilisations et justifications

- ' Le monde réel est asynchrone
  - ' Chaque programme progresse à son propre rythme
  - ' Les réseaux possèdent des temps de latence importants par rapport au temps de traitement d'une instruction machine
- ' Plusieurs approches existent
  - ' Java Messaging Service (JMS)
  - ' Échange de messages structurés (XML et/ou MIME)



# Le point de vue de xmlBlaster



*Development environments and languages are as heterogeneous as the colors of our rainbow fish.*

<http://www.xmlblaster.org/>

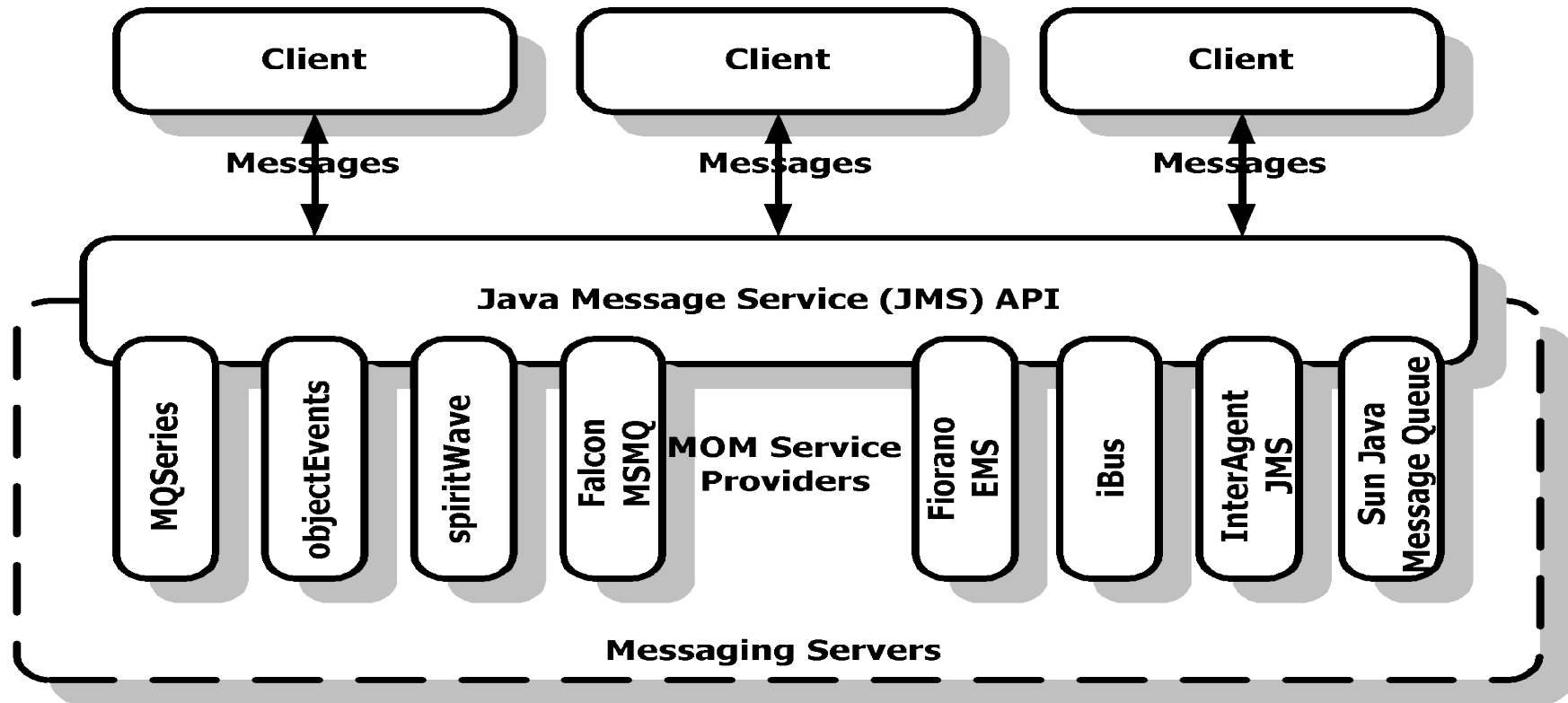


# Techniques utilisées

- ' Peer to Peer (ou Point to Point, PtP)
  - ' Échange direct de messages entre sites distants
- ' Publication/Abonnement (publish/subscribe)
  - ' Chaque site indique à l'avance les catégories de messages qui l'intéressent
- ' Requête/Réponse
  - ' Un site cherchant une information envoie un message, et attend la réponse dans un autre message (RPC)



# Architecture JMS



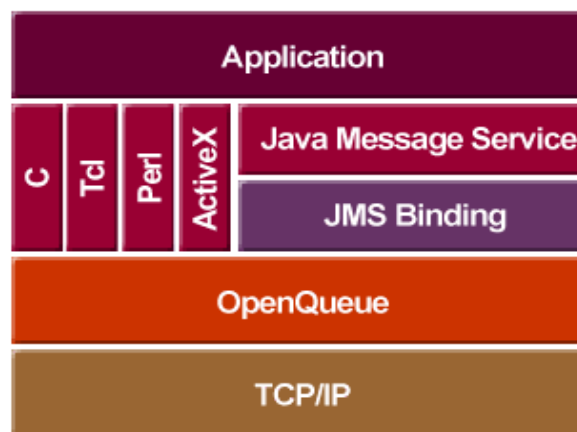
# Quelques MOM (1)

- ' xmlBlaster
  - ' Échange de messages XML
  - ' Basé sur CORBA, RMI, XML/RPC, SOAP, mail
  - ' Filtrage des messages basé sur XPath
  - ' Fonctionne sur des connections HTTP permanentes
  - ' Serveur écrit en Java
  - ' Système de sécurité modulaire (aujourd'hui LDAP et mots de passes chiffrés)



# Quelques MOM (2)

- OpenQueue
  - Écrit en Java
  - Supporte plusieurs langages et paradigmes
  - Échange des informations typées (MIME)





# Quelques MOM (3)

- ' MQseries (IBM)
- ' VCOM (Volvo)
- ' DECmessageQ (Compaq)
- ' ...



# Comparaisons

- ' Par rapport aux sockets « maison »
  - ' Un MOM route les messages
- ' Par rapport aux RPC ou ORB
  - ' Un MOM est asynchrone, un RPC ou un ORB est synchrone
  - ' Chacun peut servir à implémenter l'autre
    - ' Les communications IP sont asynchrones quoi qu'il arrive
    - ' Le service d'événements de CORBA est asynchrone



# Avantages et inconvénients

## ' Avantages

- ' Les requêtes asynchrones ne nécessitent pas d'attendre une réponse
- ' Les files de message permettent d'ajouter des serveurs « au vol » et de faire de la répartition de charge
- ' Les messages ne sont pas liés à un langage ou une architecture

## ' Inconvénients

- ' Les MOM sont plus une boîte à outils qu'un outil de répartition transparent comme CORBA ou l'annexe des systèmes répartis d'Ada



# Conclusion

Not every DAD  
(Distributed Application Development)  
needs a MOM  
But many.....

