



# M2M

## Middleware To Middleware

---

- Modèles et mécanismes de répartition
- Intergiciel et fonctionnalités
- Intergiciels configurables
- Généricité et personnalisation
- Intergiciels génériques
- Interopérabilité et passerelles
- Intergiciels schizophrènes



# Définition d'un système réparti

---

Un système réparti se compose de:

- plusieurs systèmes calculatoires autonomes...
- sans mémoire physique commune...
- communiquant par l'intermédiaire d'un réseau

Le terme « distribué » provient de la traduction de « distributed »



# Modèle de répartition

---

- Aide à définir et concevoir un système réparti
- Se fonde sur des mécanismes de répartition
  - Passage de messages
  - Sous-programme distant
  - Objet réparti
  - Objet partagé (mémoire répartie)
  - Transaction
- Fournit un contrôle sur ces mécanismes
  - Langages de description ou de programmation
  - Interfaces de bibliothèques ou de services
- S'accompagne d'un intergiciel de mise en oeuvre



# Mécanismes et variantes

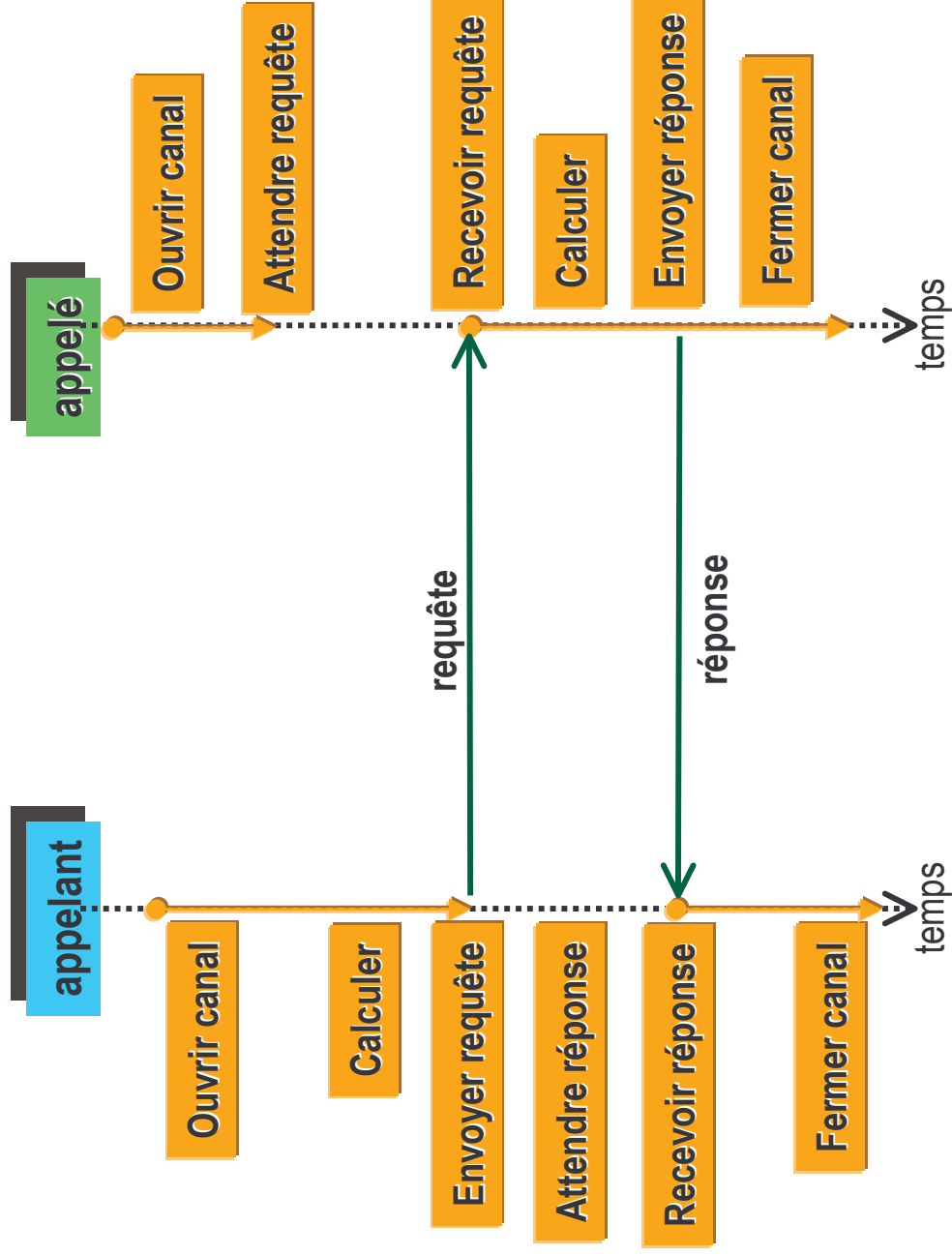
## Passage de messages

---

- Type de communication
  - Unicast (1 -> 1)
  - Multicast (1 -> N)
- Protocole de communication
  - Non-fiable (UDP)
  - Fiable (TCP)
- Nommage
  - Symétrique
  - Asymétrique
- Synchronisme
  - Synchrone
  - Asynchrone
- Réception par
  - Émetteur
  - Taille
- Localisation
  - Charge de l'émetteur
  - Service bien connu
- Hétérogénéité
  - Endianess, 32/64 bits
  - Dictionnaire (XDR, CDR)
- Emballage / Déballage
  - Structures dynamiques

# Scénario classique

## Passage de messages





# Exemples

## Passage de messages

---

- Interface réseau
  - Interface bas niveau - UDP, TCP, ATM
  - IPv6, multicast, SSL
- PVM/MPI
  - Calcul massivement parallèle
  - Communication de groupe
- Java Messaging Service
  - Interface haut niveau
  - Message oriented middlewares
  - Point à point
  - Publish / Subscribe



# Mécanismes et variantes

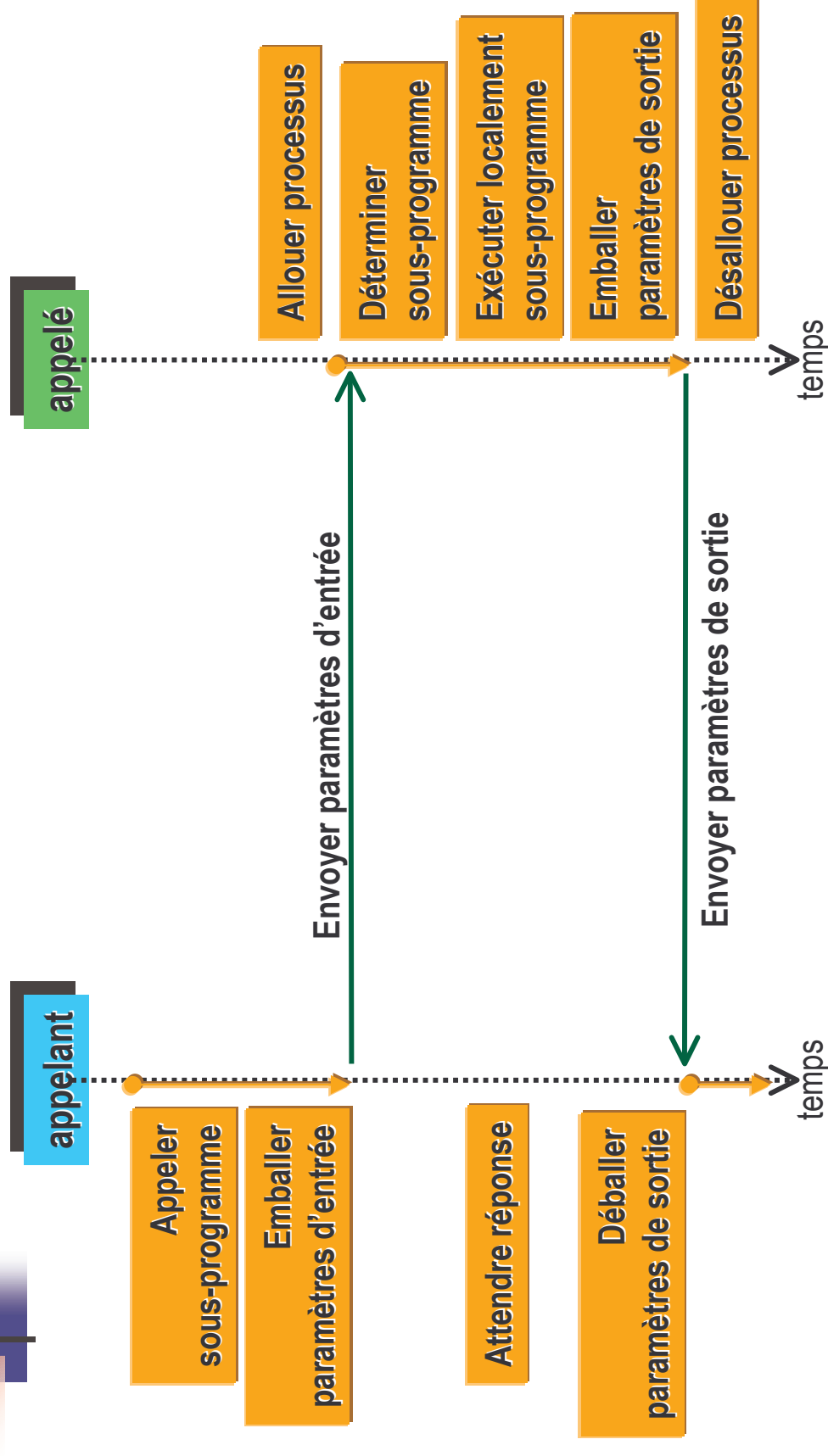
## Sous-programmes distants

---

- Généralisation de la programmation procédurale
- Paramètres empilés dans le réseau et non sur la pile
- Tâche temporaire qui exécute le sous-programme
- Pas de synchronisation avec le serveur
- Problème des structures dynamiques
- Production automatique du code de requête/réponse
- Résolution « statique » de la localisation du serveur
- Avortement de l'appel sur le client et sur le serveur
- Référence sur sous-programmes distants

# Scénario classique

## Sous-programme distant





# Exemples

## Sous-programmes distants

---

- RPC sun
  - Service de nommage ou de localisation (portmapper)
  - Compilateur **rpcgen** qui produit souche/squelette
  - Intégré dans DCE (*Distributed Computing Environment*)
- DSA (annexe des systèmes répartis d'Ada95)
  - Réduit la frontière entre réparti/monolithique
  - Services de nommage et de localisation internes
  - Annulation d'appel de sous-programmes distants
  - Référence sur des sous-programmes distants
- SOAP (Simple Object Access Protocol)
  - RPC à base de HTTP + XML (protocole uniquement)
  - Solution concurrente à CORBA de MicroSoft

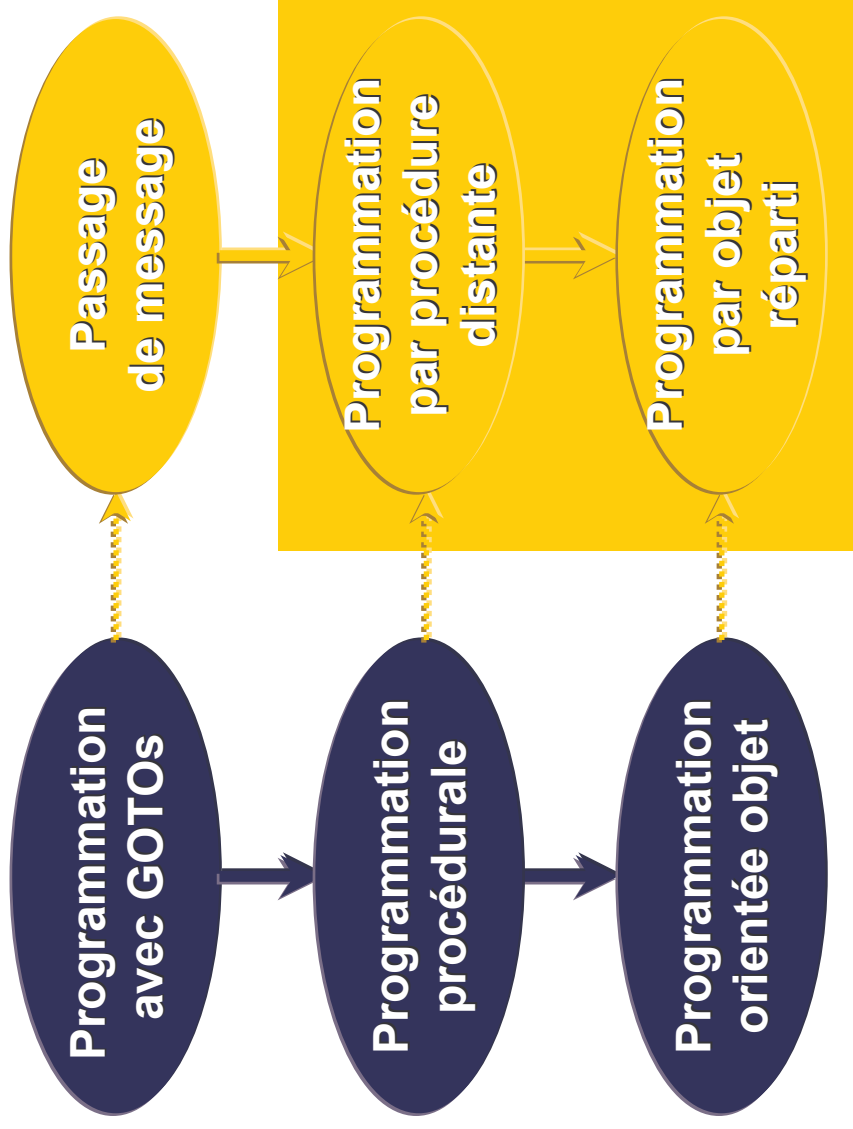


# Mécanismes et variantes

## Appel de méthode à distance

- Généralisation de l'orienté objet
- Similitude avec des RPCs dynamiques
- Accès protégé aux attributs des objets
- Définition de la notion de référence distante
- Extension simplifiée par ajouts de service
- Migration éventuelle

# Analogie entre programmation monolithique et répartie





# Exemples

## Appel de méthode à distance

---

- Modula-3
  - Extension au langage Modula-2
- RMI (remote method invocation)
  - Extension au langage Java
- CORBA
  - Langage de définition d'interface
- DCOM
  - Langage de définition d'interface
- DSA (annexe des systèmes répartis d'Ada95)
  - Extension au langage Ada95



# Mécanismes et variantes

## Appel de méthode à distance

---

- Généralisation de la mémoire partagée
- Accès direct aux attributs des objets
- Principe de localité et organisation possible en page
- Cohérence forte ou faible et mise en œuvre de cache
- Atomicité par verrou en mémoire
- Divers supports
  - Passage de message
  - Sous-programme distant
  - Objet réparti
  - Fichier partagé
  - Base de donnée



# Exemples

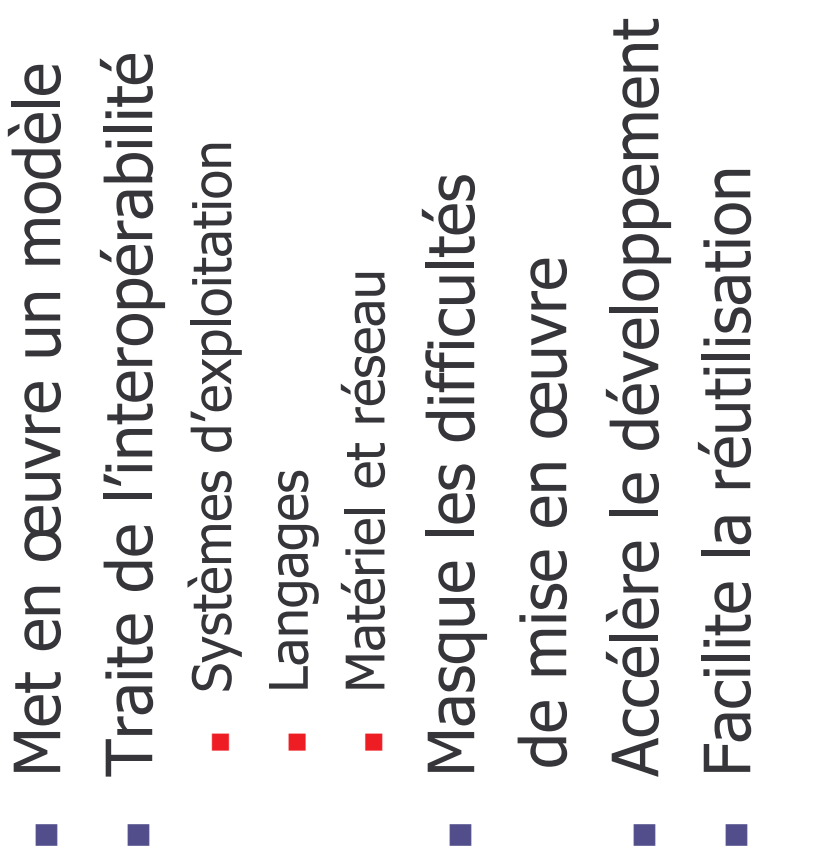
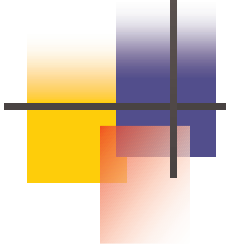
## Objets partagés

---

- Ivy
  - Interface de programmation (Li et Hudak)
  - Cohérence forte
- ThreadMarks
  - Interface de programmation (Keleher)
  - Cohérence forte et faible
- Gothic
  - Projet INRIA dans la lignée de Guide, Chorus, Cool
- DSA (annexe des systèmes répartis d'Ada95)
  - Extension au langage Ada
  - Atomicité par objets protégés

# Intergiciel

## Définition



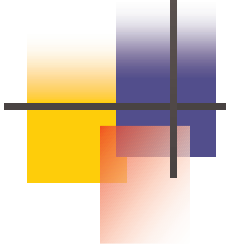


# Intergiciel

## Fonctionnalités

---

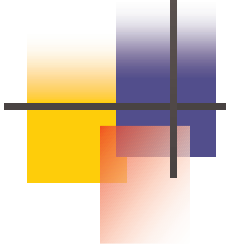
- Protocole de transport (TCP, UDP, ...)
- Hétérogénéité (XDR, CDR, ...)
- Protocole de requête (Groupe, asynchrone, ...)
- Activation (Persistence, adaptateur, ...)
- Fiabilité (*at-most-once*, réplication, ...)
- Transparence (Localisation, nommage, ...)
- Interopérabilité (Langage, Système, Matériel)



# Intergiciels configurables

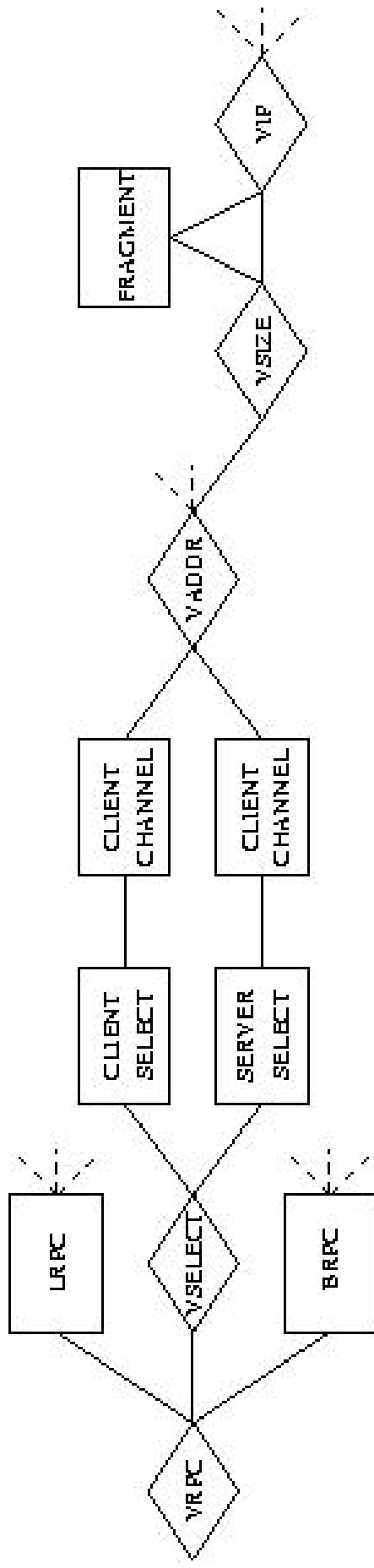
---

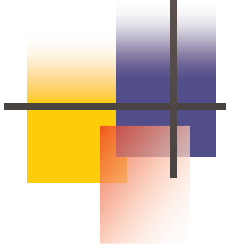
- Un intergiciel fournit un ensemble de services
- Une application utilise certains services
- L'intergiciel se configure selon les besoins
  
- Un intergiciel doit pouvoir s'adapter en fonction des besoins et des ressources
  - Protocoles de transport
  - Concurrence des requêtes
  - ...



# X-Kernel

- Conçu pour l'assemblage de protocoles
- Protocole concret
  - Fonctionnel (Passage de message, sous-programme distant)
  - Algorithmique (Compression, sécurité, ...)
- Protocole virtuel
  - Délégation auprès d'autres protocoles selon le contexte

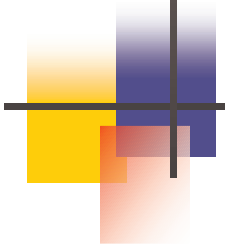




# TAO

<http://www.cs.wustl.edu/~schmidt/TAO.html>

- ACE (Adaptive Communication Environment)
  - Services de concurrence et de communication
- TAO (The ACE ORB)
  - Environnement CORBA libre et configurable
  - Configuration par **gabarits de conception**
  - Compilateur configurable (optimisation, ...)
  - ORB configurable
    - Protocoles de transport
    - Sélection de processus légers



# GLADE

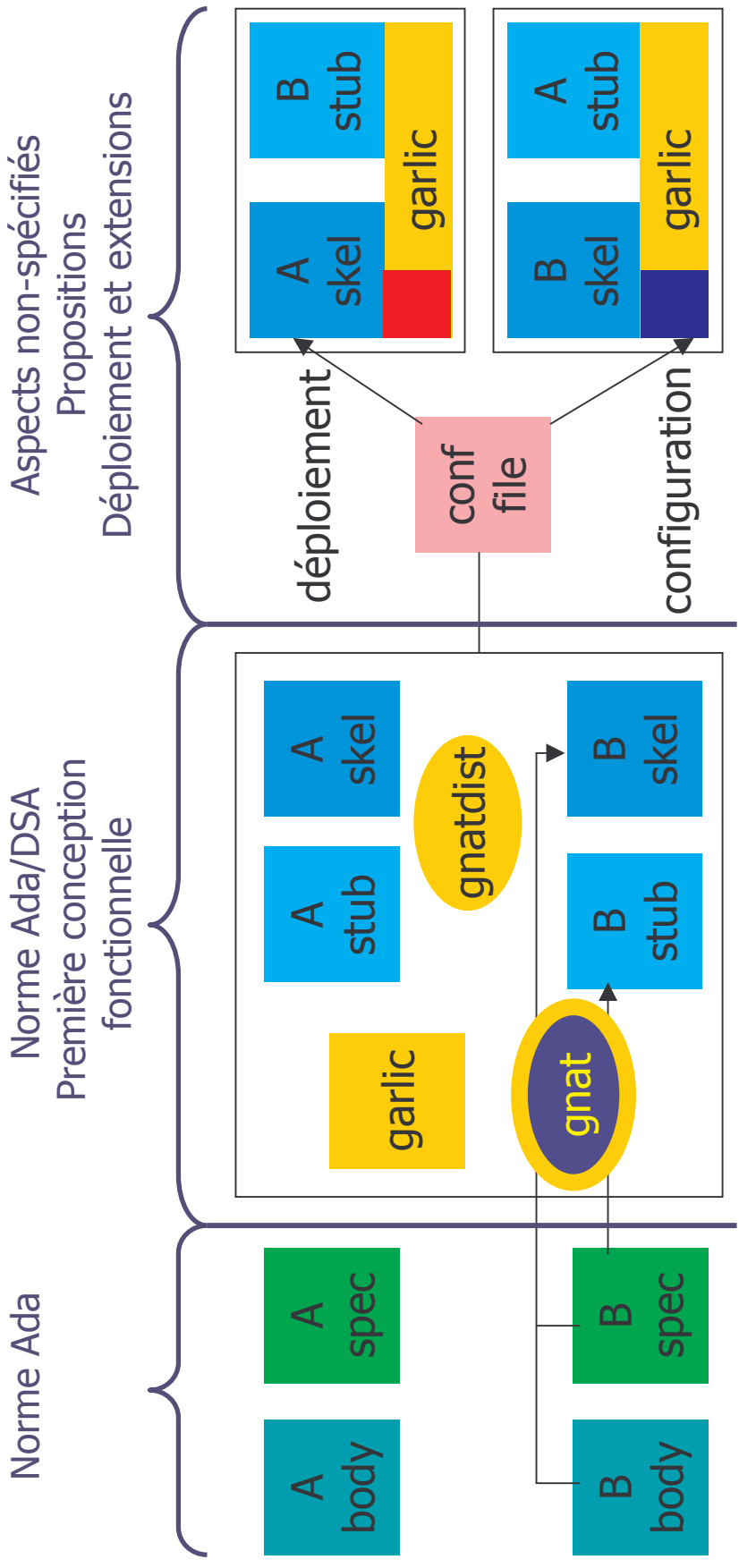
<http://www.gnat.com>

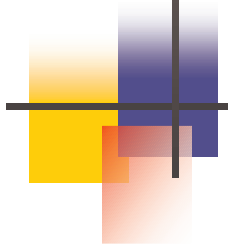
---

- Configuration après conception
  - Assemblage des unités réparties en partition
  - Adaptation de l'ORB en fonction des besoins
  - Fondé sur des **gabaris de conception**
- Éléments configurables
  - Protocoles de communication
  - Supports de stockage
  - Filtres de communication
  - Sys. de communication léger
  - Terminaison
  - Activation et Concurrence
  - Priorité



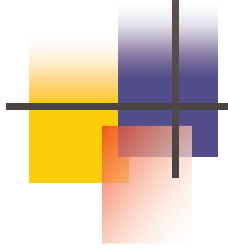
# Architecture de GLADE





# Généricité et personnalisation

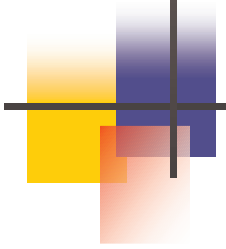
- Similitudes d'implémentations des intergiciels
- Similitudes d'architectures et de composants
- Architecture et composants **génériques**
  - Instanciation en fonction du modèle de répartition
  - Production d'une **personnalité** (instance)
  - Concrétisation de certains composants
  - Remplacement de certains composants
  - Suppression de certains composants



# Intergiciels génériques

---

- Extension de la notion de configurabilité
- Configuration selon le modèle de répartition
  - Adapter le modèle de répartition selon les besoins
  - Produire rapidement un intergiciel sur mesure
- Effort de conception important
  - Architecture
  - Composants
- Qualité des intergiciels (forte réutilisation)



# QuarterWare

---

- Intergiciel générique développé en C++
  - Personnalité CORBA
  - Personnalité RMI (adaptée pour C++)
  - Personnalité MPI
- Conçu à l'aide de gabarits de conception
- Proche de mécanismes de réflexivité
- Inspiré du projet TAO
- Faible factorisation des composants
- Travail de personnalisation important

# Jonathan

<http://www.objectweb.org>

- Intergiciel générique libre développé en Java
  - Personnalité CORBA (David)
  - Personnalité RMI (Jérémie)
  - Personnalité MOM (Joram)
- Fondé autour des liaisons ODP
  - Liaisons implicites
  - Liaisons explicites
- Partie du projet ObjectWeb



# Paradoxe de l'intergiciel

---

Si les intergiciels traitent de l'interopérabilité entre architectures matérielles et systèmes d'exploitation, ils sont à l'origine d'une nouvelle forme d'hétérogénéité en raison du modèle de répartition

- Concept de « **middleware to middleware** »
- Interopérabilité entre modèles de répartition
- Production de passerelles statiques ou dynamiques

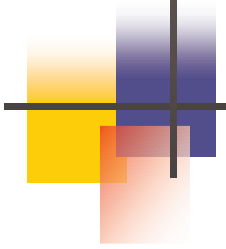


# CorbaWeb

<http://corbaweb.lifl.fr>

---

- Passerelles génériques entre Web et CORBA
- Relie un client Web à un serveur CORBA
- Produit fichiers HTML et formulaires CGI
- Transforme une requête HTTP en une IIOP
- Développé à partir du langage CORBAScript
- Fondé sur
  - l'invocation dynamique (DII)
  - le répertoire d'interfaces (IR)



# Schizophrénie

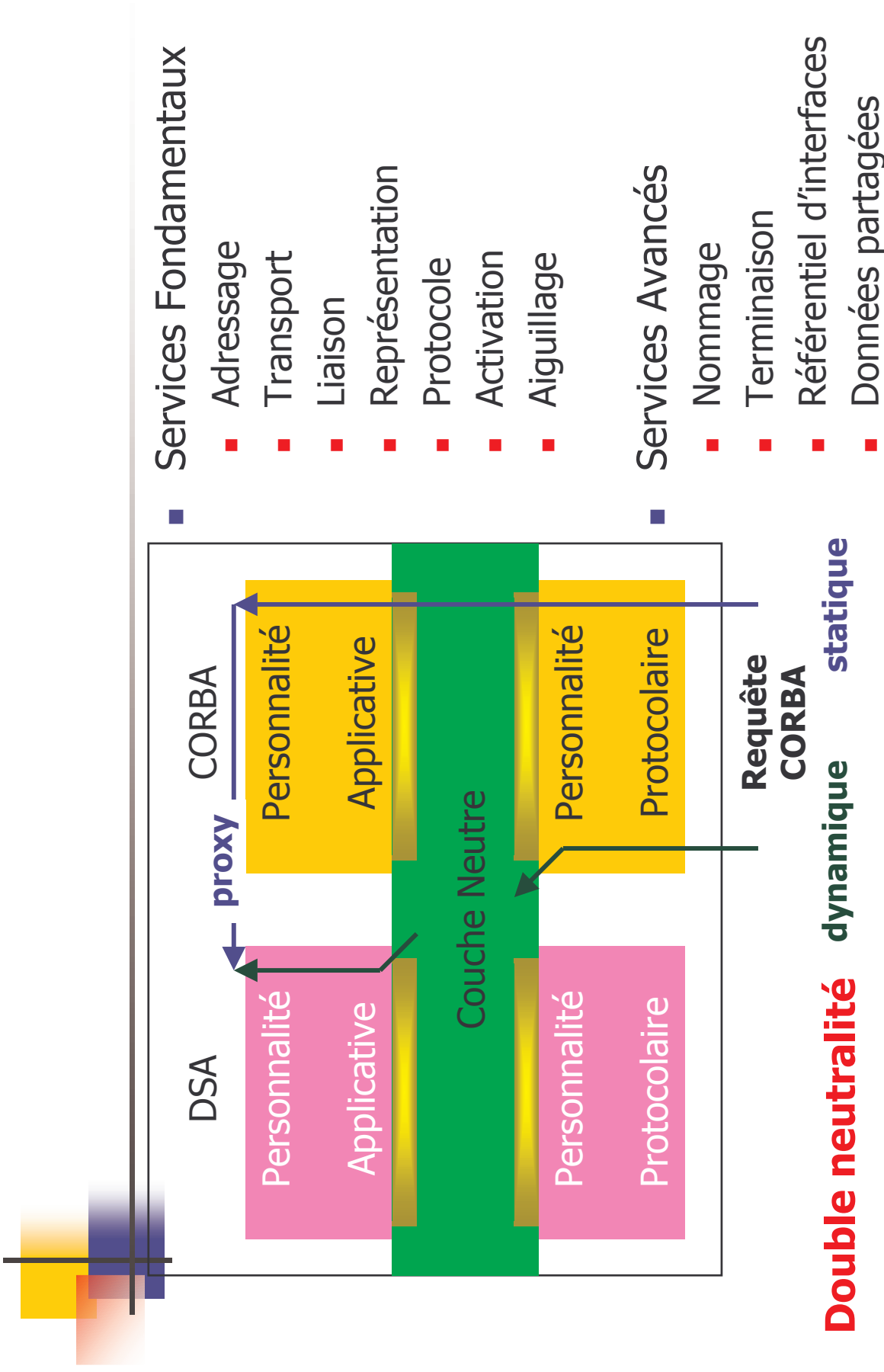
---

La schizophrénie caractérise chez un intergiciel sa capacité à disposer, simultanément, de plusieurs personnalités afin de les faire interagir

Solution pour la mise en œuvre de passerelles dynamiques entre modèles

- Cohabitation de personnalités multiples
- Partage code / données entre personnalités
- Découplage entre applicatifs et protocoles

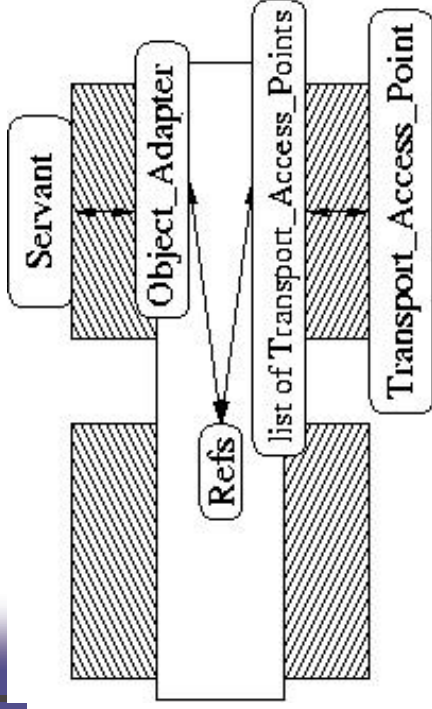
# Architecture et Services



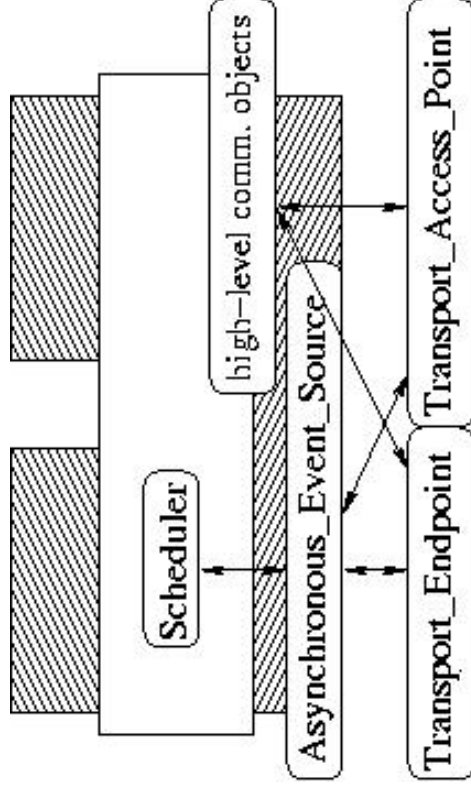
**Double neutralité** dynamique statique



# Adressage et Transport



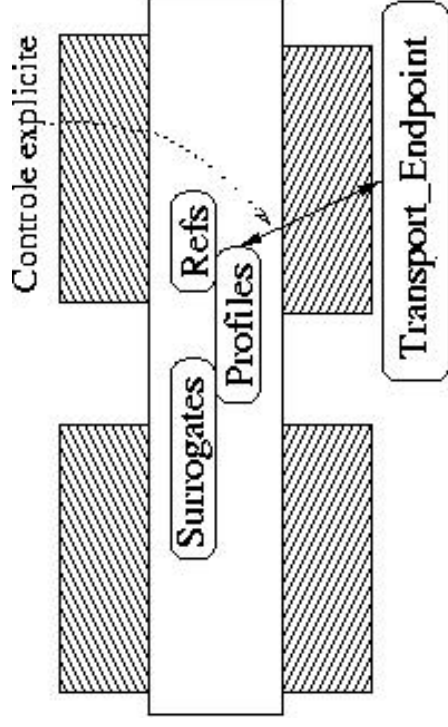
- Addressage
  - Enregistre un « servant »
  - Associe une « socket »
  - Produit une référence
  - Traduit la référence



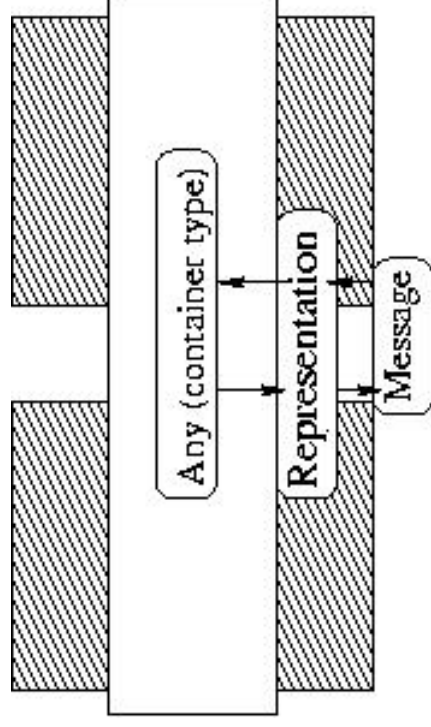
- Transport
  - Crée un point d'écoute
  - Crée un point d'échange
  - Traite l'asynchronisme



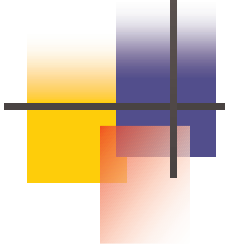
# Liaison et Représentation



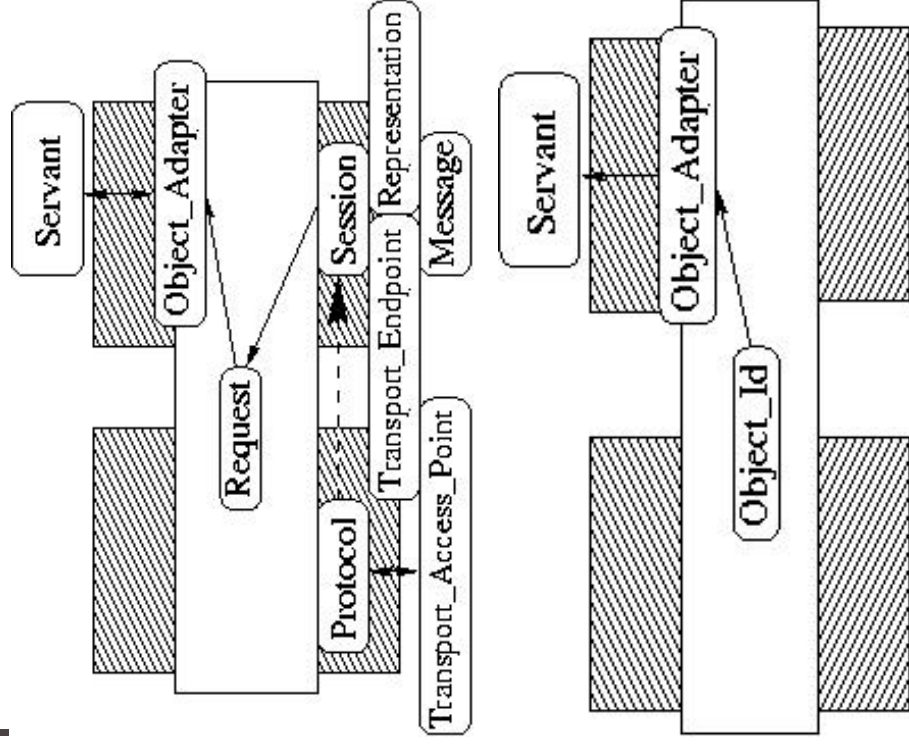
- Liaison
  - Méthodes identiques entre objet et surrogé
  - Surrogé lié (« socket »)
  - Surrogé non-lié (« ref »)
  - Proxy pour passerelle



- Représentation
  - Emballage
  - Déballage
  - Type autodescriptif



# Protocole et Activation

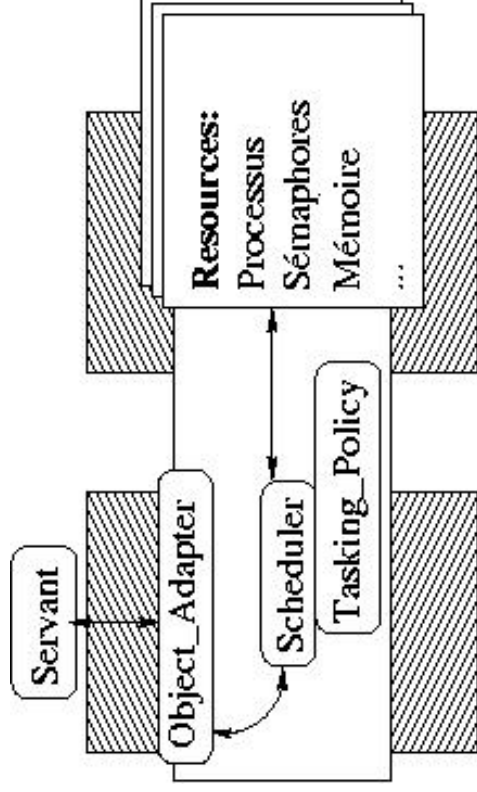


- Protocole
  - Crée requêtes/réponses
  - Proche de représentation
  - Transmet à l'activation
- Activation
  - Décode la référence
  - Associe
    - Un objet physique
    - Un objet logique
  - Persistance, etc



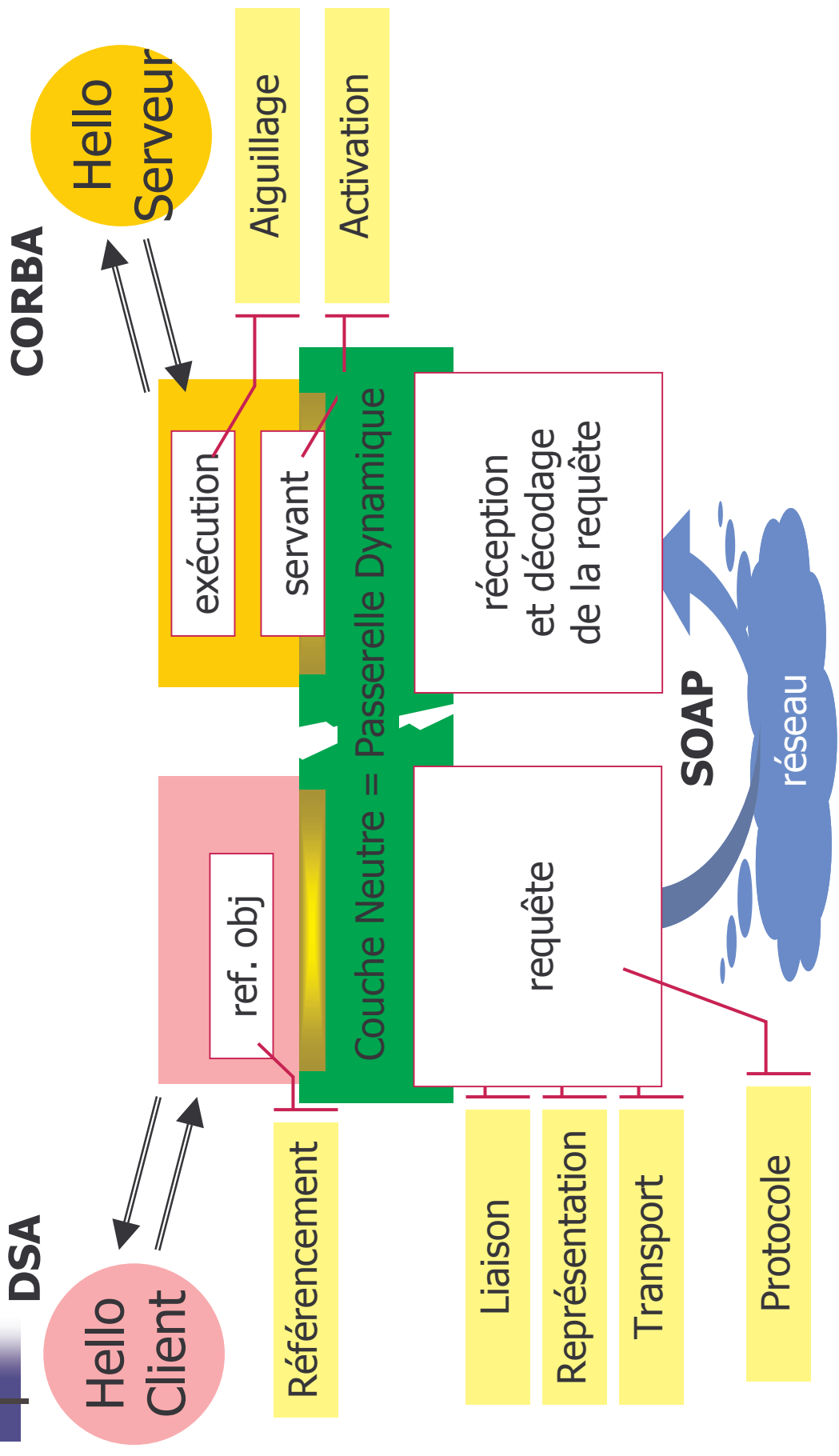
# Aiguillage

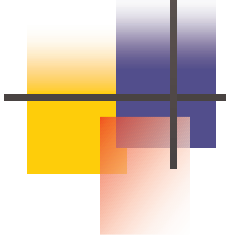
- Aiguillage
  - Crée un processus léger
  - Alloue les ressources
  - Exécute la requête





# Scénario





# PolyORB

<http://libre.act-europe.fr/polyorb>

---

- Premier intergiciel schizophrène
  - Fondé sur des gabarits de conception
  - Logiciel libre du projet GNU (GNAT/GCC)
  - Guide méthodologique
- Personnalités
  - CORBA, applicative et protocolaire
  - SOAP, protocolaire
  - MOM, protocolaire
  - DSA, en cours