



# Middleware et calcul haute performance à grande échelle

---

**Jean-Marc.Geib@lfl.fr**

Laboratoire d'Informatique Fondamentale de Lille

UMR CNRS 8022

Equipe GOAL & action INRIA Jacquar

**Forum ORAP**

27 Mars 2003



# Calcul Haute Performance à Grande Echelle

---

## | Hétérogénéité et interopérabilité

- ? En réseaux
- ? En points de calcul
- ? En services systèmes
- ? En paradigmes de programmation
- ? En paradigmes de calcul et de communication

## | Comment

- ? Mixer les plates-formes techniques
- ? Les approches de programmation et de communication
- ? Réutiliser des services systèmes performants
- ? Déployer, assembler, configurer et administrer les éléments

## | Tout en préservant les performances



# Des approches techniques...

---

## | Communications

- ? Fichiers
- ? Sockets
- ? Remote Procedure Call (RPC)
- ? Remote Method Invocation et Objets répartis (RMI-Corba)
- ? Web Services (SOAP-WSDL)
- ? Composants distribués (EJB,CCM, .Net)

## | Programmation

- ? Orientée Programme
- ? Orientée Objets + wrapping de programme => serveurs d'objets
- ? Orientée Composants + wrapping de serveurs d'objets

## | Middleware = bus et services systèmes

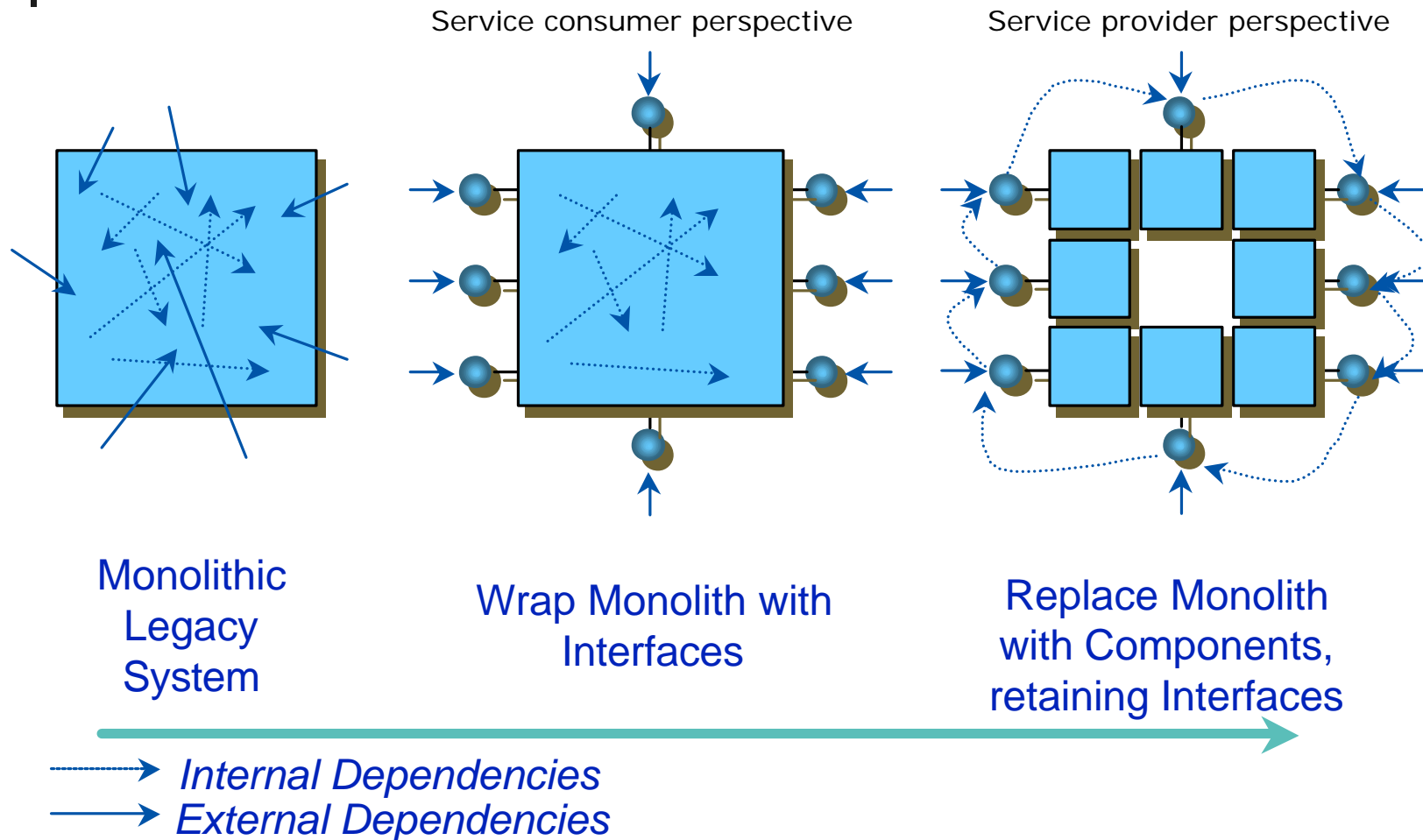


# Des défis techniques pour le middleware

---

- | Adaptation du middleware à un environnement de calcul haute performance
  - ? Savoir gérer différents types de ressources
    - Architectures (grappes, grilles...), Réseaux, Codes séquentiels/parallèles et/ou synchrones/asynchrones (Ex Asynchronous Messaging Invocation)
    - Bibliothèques et connecteurs spécifiques? (Ex parallel Corba)
  - ? Savoir mixer différents types de ressources
    - Choix configurables ou dynamiques de l'utilisation des ressources
    - Middleware multi-protocoles et adaptatif (Pluggable Transport) ?
  - ? Savoir mixer plusieurs middlewares sur l'architecture
    - Un cœur de middleware commun + des personnalités ?
  - ? Savoir déployer et configurer le middleware et les ressources
    - Retrouver le métier d'architecte
- | Mais aussi des défis de haut niveau pour la réalisation d'applications réparties à grande échelle
  - ? Généraliser l'utilisation de composants

# La transition vers les composants



Adapted from the Butler Group

Adaptée aux environnements de calcul haute performance!



# From DOC Middleware ...

---

- | **Distributed Object Computing Middleware**

- ? Exemples: CORBA – DCOM - Java RMI – SOAP
- ? Une vision structurante d'objets répartis
- ? Longue expérience académique et industrielle

- | Pas de notion d'architecture des applications

- ? Les liens entre objets sont cachés dans le code
- ? Pas manipulables ni configurables par un architecte

- | Pas de séparation claire entre les aspects fonctionnels et les aspects non-fonctionnels de l'applications

- ? Tous les aspects sont mixés dans le code et programmés explicitement
- ? Grande difficulté pour les experts du domaine

- | Pas de facilités de packaging et de déploiement

- ? Seulement des solutions *ad hoc*



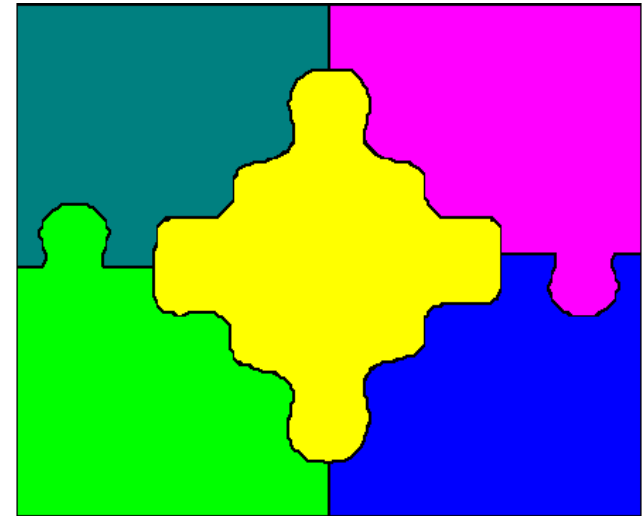
## ... To DCC Middleware

---

- | Distributed Component Computing Middleware
  - ? Évolution du DOC middleware
  - ? Exemples CCM - .NET - EJB
  - ? Couvre plus complètement le software process ( packaging, assembling and deploying)
  
- | Des notions d'architecture
  - ? Les liens entre composants sont explicites et configurables
  - ? Les composants sont des entités de premier ordre
  
- | Les aspects fonctionnels et non fonctionnels sont clairement séparés
  - ? Via les composants et conteneurs de composants
  - ? Via des descripteurs XML plutôt que par programmation
  
- | Le packaging est prévu pour du déploiement automatique
  - ? Format standard pour les paquetages et les binaires
  - ? API de déploiement

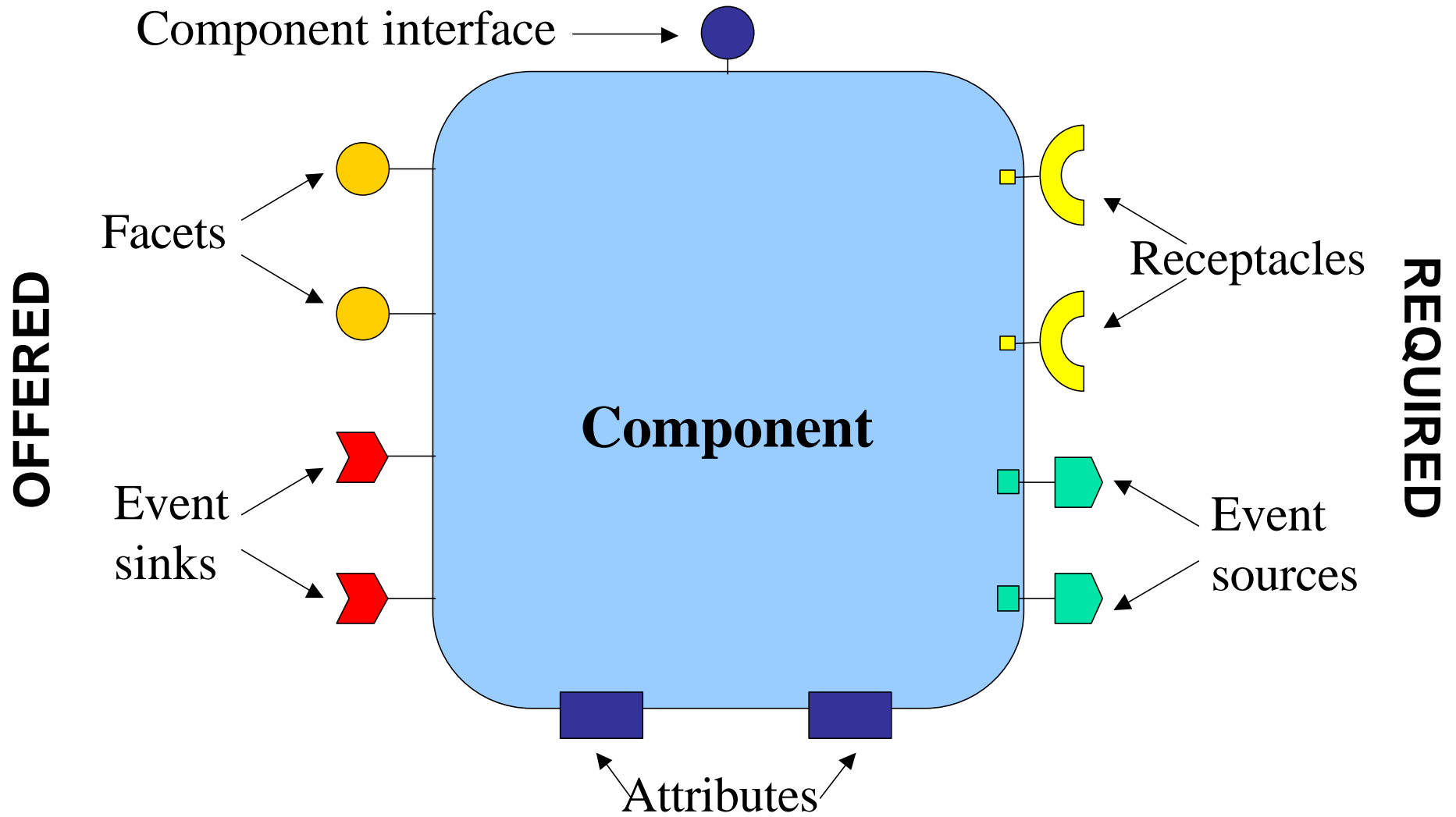
# Qu'est ce qu'un composant?

- ⋮ Self contained unit
  - ? Encapsulating data and logic (like objects)
- ⋮ Composable unit
  - ? Well defined external interfaces
  - ? No communication path is hidden
- ⋮ Configurable unit
  - ? Reused in various technical & business contexts
- ⋮ Packaged and deployable unit

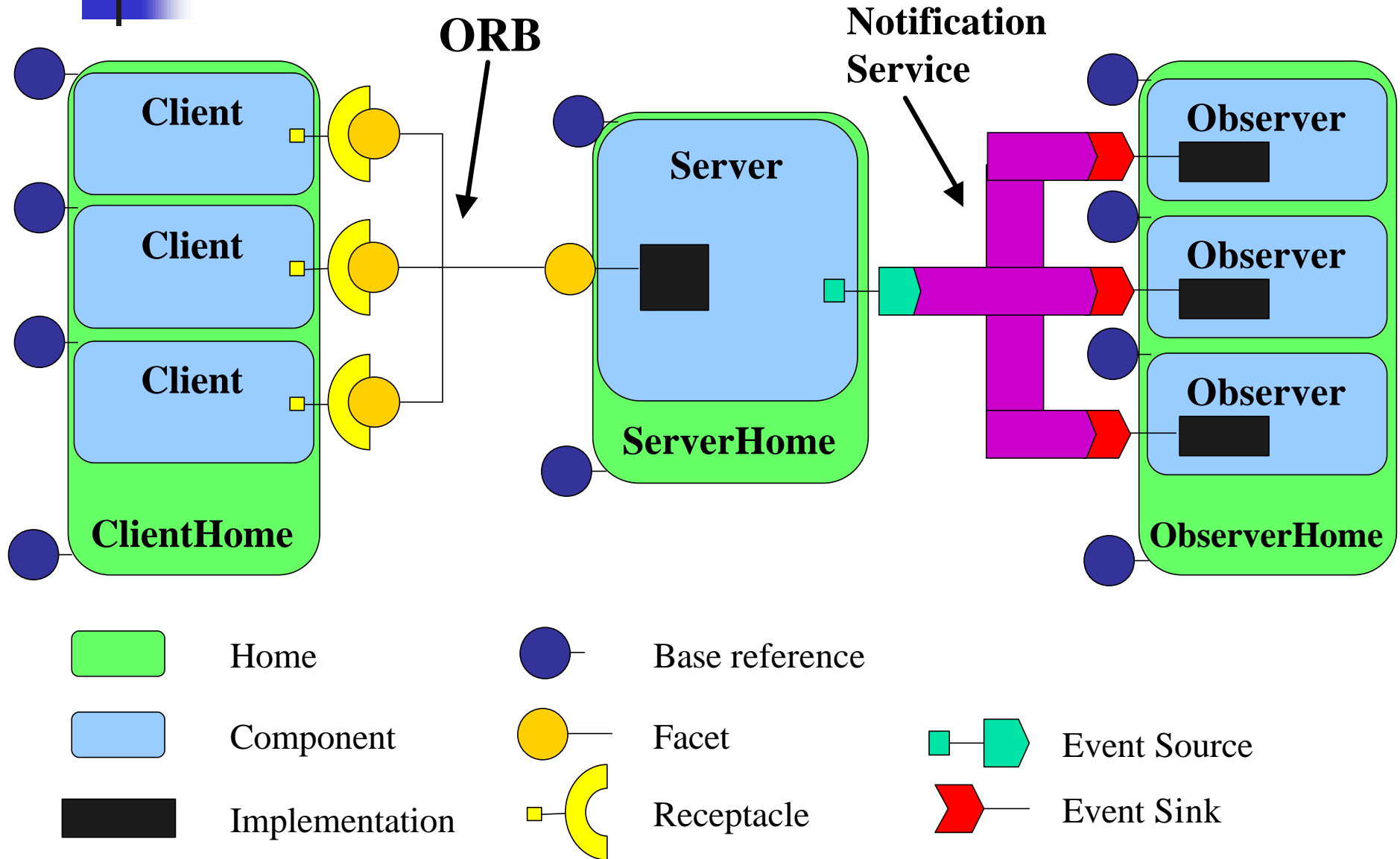




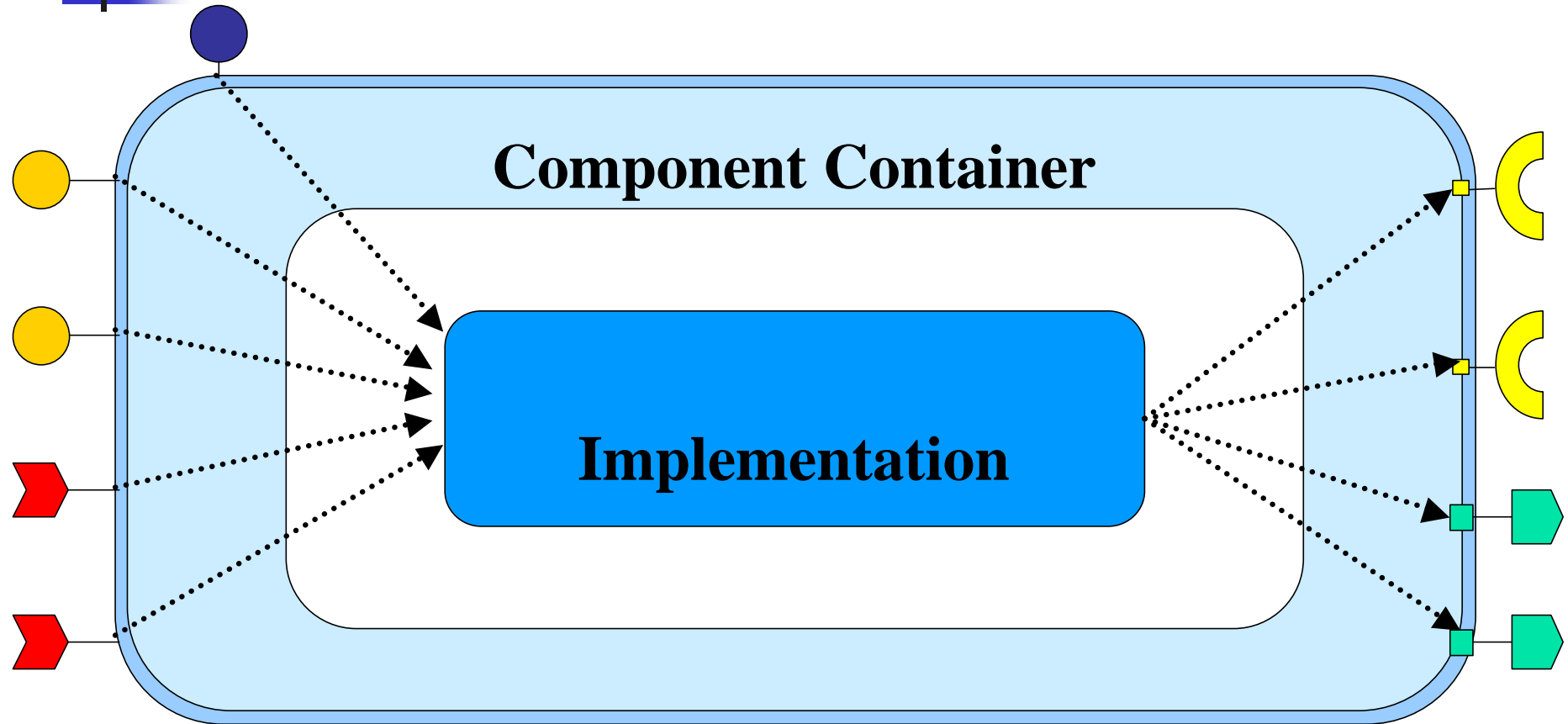
# Le modèle de composant



# Assemblage de composants



# Composant et conteneur



- ⋮ Le code métier est dans l'implémentation
- ⋮ Les propriétés non-fonctionnelles et les connecteurs sont dans le conteneur

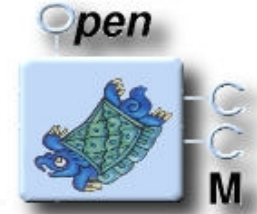


# An Example CCM : CORBA Component Model

---

- | The Industry's First Multi Language Component Standard
  - ? Component-based software engineering process
  - ? Advanced component model
  - ? Server-side container framework
  - ? Packaging and distributed deployment
  - ? EJB interworking
  - ? Component meta models
  - ? Ready for Model Driven Architecture (MDA)

# OpenCCM



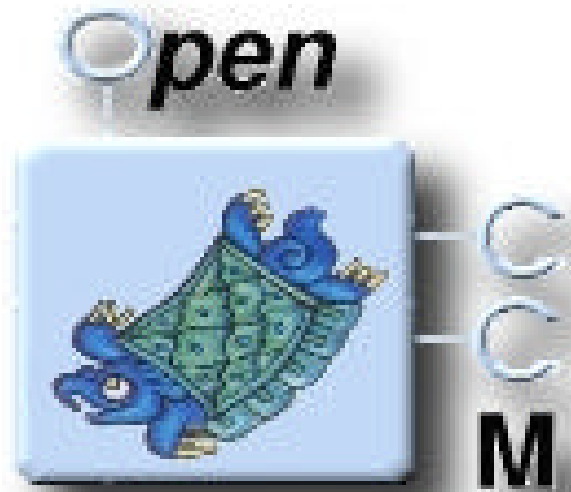
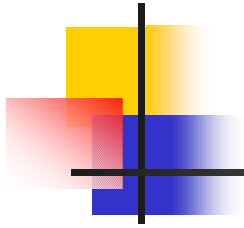
- ! A LIFL and ObjectWeb project

- ? <http://www.objectweb.org/openccm>
- ? Open source under LGPL



- ! A flexible and open platform to design, develop, package, assemble, deploy and administrate CORBA Component applications

- ? ORB independent
- ? Reusable, flexible, adaptable, customizable, ...
- ? A évaluer pour le calcul haute performance
  - En cours dans Grid-RMI
    - Conteneurs pour haute performance
    - Chaîne de production de composants parallèle



<http://www.objectweb.org/openccm>