



# Conteneurs Adaptable Dynamiquement pour la plateforme CORBA Component Model Model

Journée de l'action spécifique CNRS-GET  
Systèmes répartis et réseaux adaptatifs au contexte (“Context-Aware”)

Laboratoire Informatique de PARIS6 – LIP6

Assia HACHICHI, Gaël THOMAS, Bertil FOLLIOT, Cyril MARTIN



# Agenda

---

- **Introduction**
- **Solution: Container Virtual Machine**
  - **Concept**
  - **Machine Virtuelle Virtuelle**
  - **Architecture**
- **Prototype: Container Virtual Machine**
- **Exemple**
- **Conclusion et Perspectives**

# Introduction

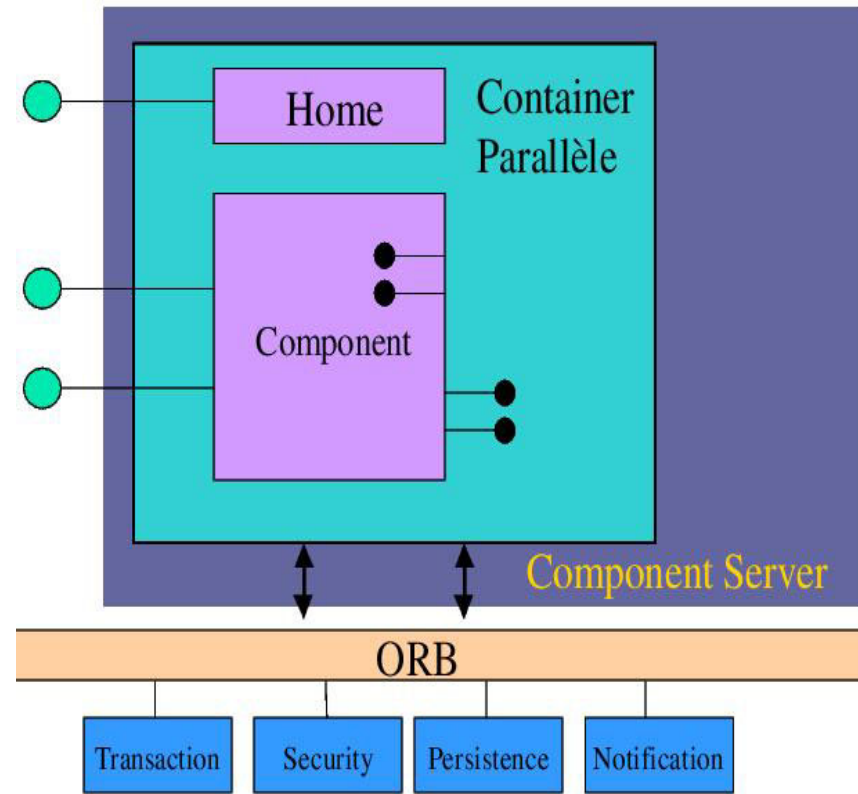
---

- Utilisation des applications orientées composants dans le domaine des Systèmes répartis.  
Apparition des middlewares : Couche de logiciel (réparti) destinée à:
  - Masquer la répartition des traitements et des données
  - Fournir une interface commode aux applications
  - Séparer le code système et le code métier
- Nouveaux services
  - *Configuration* statique au déploiement
- En cours d'exécution (*dynamique*)
  - Màj sans interruption de service
  - Qualité de service
- Exemple: Domaine des télécommunications

**Objectif:** Offrir un mécanisme permettant de rajouter et adapter les services à la volée.

# Introduction

- Technologies distribuée modélisant Les composants :
  - EJB (Entreprise Java Beans) de Sun
  - COM/DCOM
  - Architecture .NET de Microsoft
  - CCM (CORBA Component Model) de IOMG
- OpenCCM
  - Plateforme ouverte pour composants CORBA
  - Norme CCM de l'OMG
  - Lifi - Lille



# Agenda

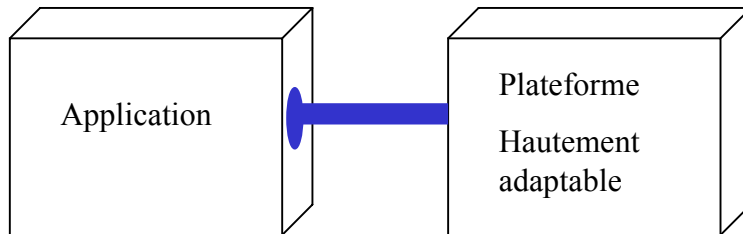
---

- ❑ Introduction
- ❑ **Solution: Container Virtual Machine**
  - ❑ Concept
  - ❑ Virtuelle Virtuelle Machine
  - ❑ Architecture
- ❑ **Prototype: Container Virtual Machine**
- ❑ Exemple
- ❑ Conclusion et Perspectives

# Concept

---

- Idée: Rajouter un point d'entrée transparent.
  - Effectuer les adaptations dynamiques.
  - Injecter du code à la volée.
- Besoin d'une plateforme hautement adaptable.
  - Injecter du code à la volée.
  - Injecter du code.
  - Quelques soit la plateforme de l'application.

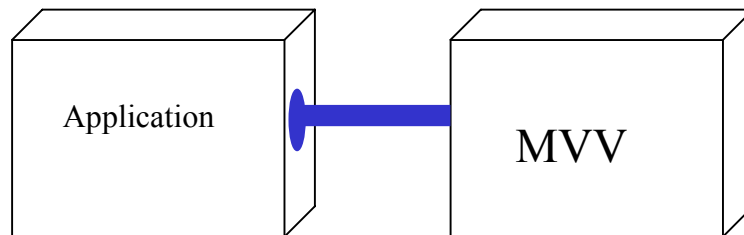


# Machine Virtuelle Virtuelle

## Comment adapter dynamiquement?

---

- La MVV est une plateforme hautement adaptable.
  - Construire un environnement d'exécution personnalisé
- Les objectifs principaux de la MVV
  - Permettre l'adaptation du langage et du système
  - Permettre de réaliser l'interopérabilité entre différents langages.



# Adaptation dynamiquement

---

- Adaptation des composants orientés système
  - Rajouter (adapter) un composant.
  - Implémentation du code service système.
  - Adapter l'implémentation
- Package - Service
  - Charger du code de service
  - Activer le service
- Adaptation service en C
  - Écriture du code en C
  - Possibilité d'effectuer des appels systèmes

# Agenda

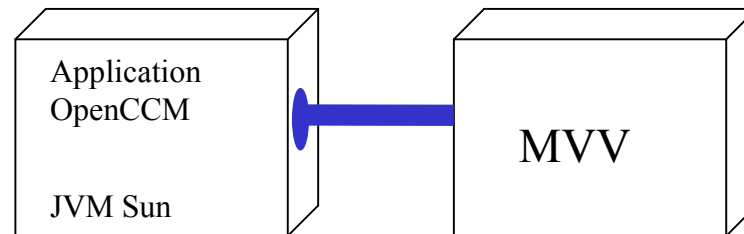
---

- ❑ Introduction
- ❑ Solution: Container Virtual Machine
  - ❑ Concept
  - ❑ Virtuelle Virtuelle Machine
  - ❑ Architecture
- ❑ **Prototype: Container Virtual Machine**
- ❑ Exemple
- ❑ Conclusion et Perspectives

# Prototype

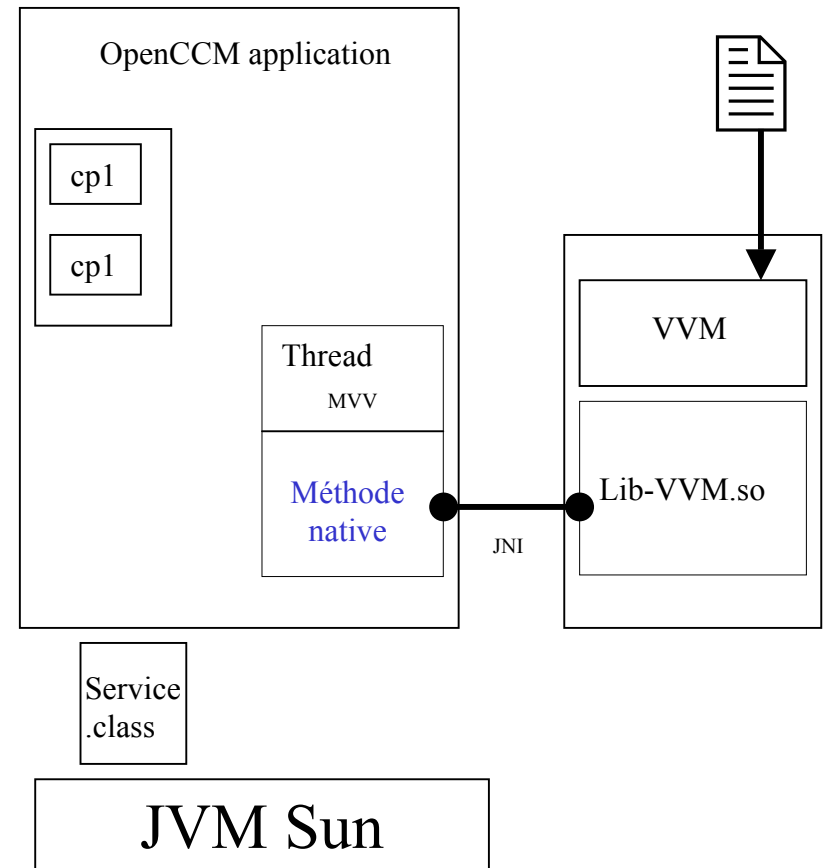
---

- OpenCCM
- La Machine Virtuelle Java de Sun



# Container Virtual Machine

- **Principe:** Rajout de point d'entrée
- Utilisation de méthode *native*
- VVM manipule les mêmes symboles et méthodes
- **JNI:** permet au code java de fonctionner avec des applications et bibliothèques écrites en d'autres langages.



# Agenda

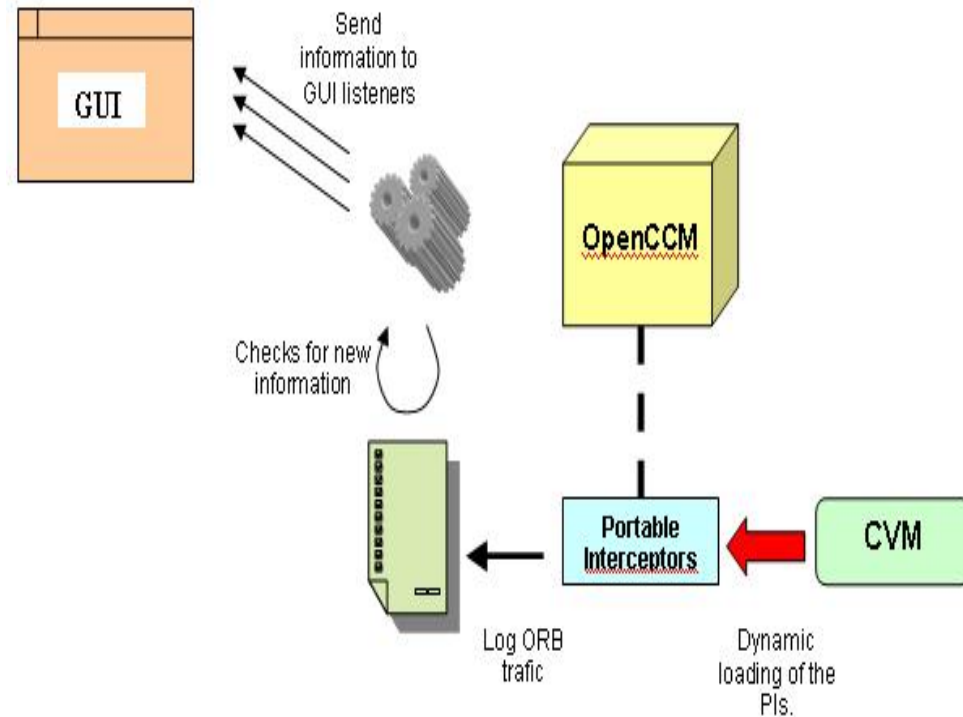
---

- ❑ Introduction
- ❑ Solution: Container Virtual Machine
  - ❑ Concept
  - ❑ Virtuelle Virtuelle Machine
  - ❑ Architecture
- ❑ Prototype: Container Virtual Machine
- ❑ Exemple
- ❑ Conclusion et Perspectives

# Portable Interceptor Monitoring

Crochets insérés à l'intérieur de l'ORB.

- Invoquer à chaque opération
- L'information logger:
  - Timestamps
  - Identifications de Transaction
  - Opérations
  - Côté d'invocation
  - Temps d'exécution



# Agenda

---

- ❑ **Introduction**
- ❑ **Solution: Container Virtual Machine**
  - ❑ **Concept**
  - ❑ **Virtuelle Virtuelle Machine**
  - ❑ **Architecture**
- ❑ **Prototype: Container Virtual Machine**
- ❑ **Exemple**
- ❑ **Conclusion et Perspectives**

# Avantages

---

- CVM offre plusieurs avantages :
  - Permet à l'administrateur de spécifier les propriétés systèmes à la volée.
  - Réduit le coût du déploiement.
  - Spécifications CVM valides pour toutes les applications CCM

# Perspectives

---

- Généraliser la CVM
  - Implémentation CCM
  - Middleware orientées composant
- Adapter les plateformes actuelles aux domaines des télécommunications
  - Tolérance aux fautes
  - Qualité de services
  - Sécurité
- Transformation de modèle
  - Automatiser l'adaptation