

# Programación Distribuida y Tiempo Real

---

JADE

# Movilidad – Migración de Código

---

Hasta ahora vimos diferentes formas de pasar información/datos en un sistema distribuido.

Movilidad: Transferir programas o código ejecutable

Movilidad de código y Migración de código/procesos se usan como sinónimos, aunque necesariamente no lo son.

# Razones para migrar código

---

- Aumentar la eficiencia: repartir carga computacional y/o disminuir carga de la red de comunicaciones
- Permitir la carga dinámica de código: código no conocido a priori o código por demanda y/o mejorar la distribución/instalación del código en sistemas grandes y/o muy distribuidos
- Últimamente, más relacionado con la eficiencia: mejorar la capacidad o velocidad de respuesta para un usuario

# Proceso en ejecución

---

- Código Binario
- Estado:
  - Estado de ejecución: recursos no compartidos y propios de la ejecución, como la pila, registros del procesador y puntero de programa
  - Espacio de datos: recursos a los que se accede/utiliza (en tiempo de ejecución), como archivos o impresoras

# Modelos de Movilidad

---

- Según lo que transfiere
  - Débil: solamente el código de un proceso. No es un proceso en ejecución sino el código ejecutable
  - Fuerte: código y estado del proceso. Es un proceso en ejecución, es lo que tradicionalmente se llama migración
- Según quien inicia la migración
  - Proactiva: el origen del código inicia la transferencia
  - Reactiva: el destino del código inicia la transferencia

# Movilidad Débil

---

- ¿Dónde se ejecuta el código?
- En el proceso receptor, se transfiere una porción del código. Ejemplo: JavaScript
- En un proceso separado. Ejemplo: Applet

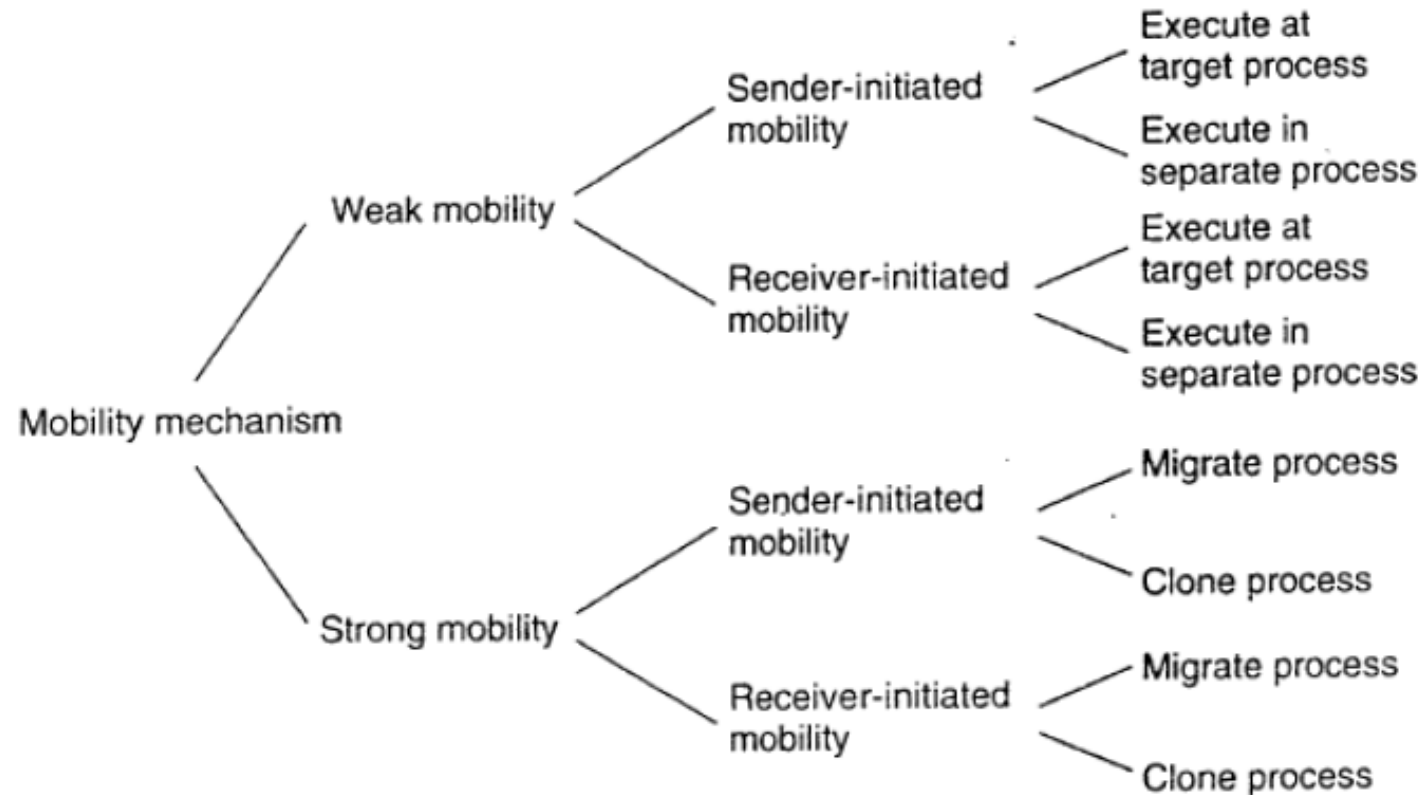
# Movilidad Fuerte

---

- ¿Que se hace con el proceso original?
- Migrar: el proceso literalmente se “mueve” y deja de existir en el sistema inicial/original
- Clonar: se crea una copia exactamente igual en otro sistema y ambos coexisten

# Grafico Clásico (Tanenbaum, 2Ed)

---



**Figure 3-18.** Alternatives for code migration.



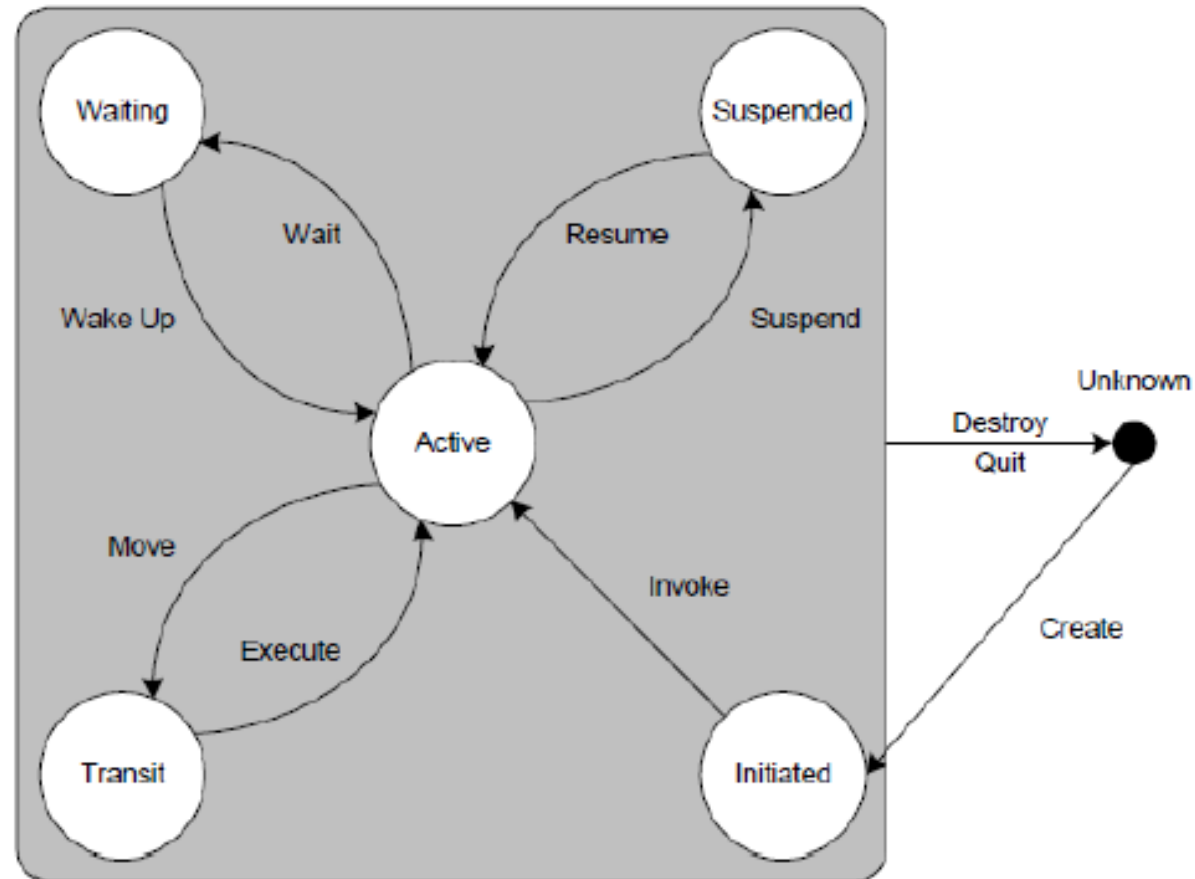
# Agentes y JADE

---

- Agente: Entidad autónoma, con capacidad de decisión y comunicación
- JADE: Java Agent DEvelopment Framework
  - <http://jade.tilab.com/>

# Ciclo de vida de un Agente en JADE

---

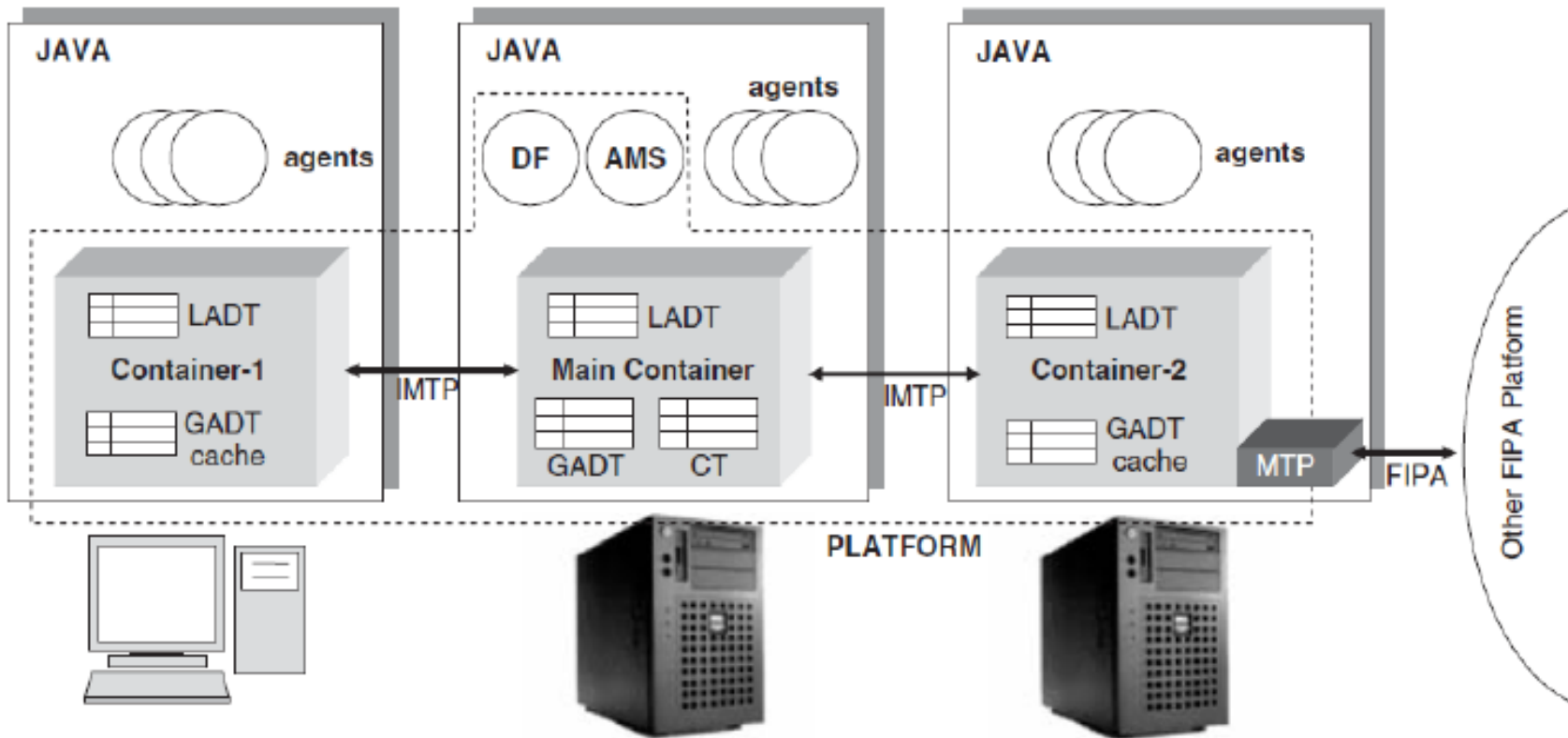


# Contenedores y Plataformas

---

- Contenedor: ambiente de ejecución inmediato del agente.
- Plataforma: conjunto de contenedores

# Contenedores y Plataformas



## LADT: Local Agent Descriptor Table

## GADT: Global Agent Descriptor Table

**DF: Directory Facilitator**

## AMS: Agent Management System

# Instalación y Ejecución

---

- JADE-all-4.3.0.zip
- Prueba:
  - `java -cp lib/jade.jar jade.Boot -gui`
  - `java -cp lib/jade.jar jade.Boot -gui -local-host 127.0.0.1`

# Ejemplo

---

# Ejemplo

---

- `javac -classpath lib/jade.jar -d classes myexamples/AgenteMovil.java`
- `java -cp lib/jade.jar:classes jade.Boot -gui`
- `java -cp lib/jade.jar:classes jade.Boot -gui -container -host localhost  
-agents mol:AgenteMovil`

# Ejemplo Móvil

---

- doMove()
- afterMove()



# Mas...

---

- Comportamientos (Behaviour)
- Comunicación entre agentes