



Centro de Tecnologías de Información y Comunicación  
Universidad Nacional de Ingeniería, Lima

# Supercomputación y Grids Computacionales

**Yudith Cardinale y Jesús De Oliveira**  
**Universidad Simón Bolívar**

**Marzo 2009**

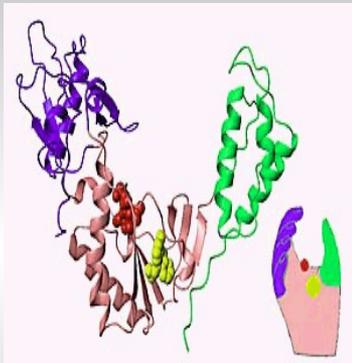


# *Supercomputación*

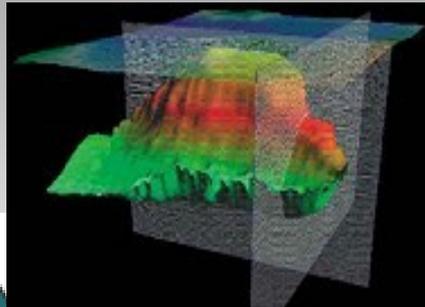
- ◆ *Uso de los computadores más poderosos del momento (supercomputadores) para la resolución de problemas que requieren mucho cómputo y/o análisis de grandes volúmenes de datos*
- ◆ *Computación de Alto Desempeño (High Performance Computing, HPC)*



# Aplicaciones HPC



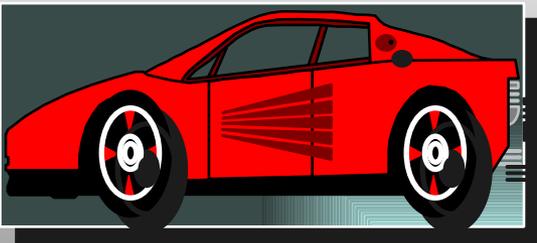
Ciencias de la vida



Petróleo



Internet & e-commerce



CAD/CAM



Biología Digital



Aplicaciones Militares



# Aplicaciones HPC

- ◆ Los datos presentan un gran desafío:
  - Los detectores del Laboratorio Europeo de Partículas Físicas, producirían para el año 2005 varios *petabytes* de datos por año, un millón de veces la capacidad de un computador de escritorio promedio.
  - Analizar estos datos pudiera requerir de alrededor de 20 teraflops/seg de poder computacional. Para el 2004 el supercomputador más rápido ofrecía 280600 Gflops. Hoy en día 1026000 Tflops



# *Arquitecturas para supercomputación*



CRAY 2 (1985-89)

- ◆ Supercomputadores tradicionales se basaban en pocos procesadores, con facilidades vectoriales, con tecnología especializada de punta (30 M\$!!)



# *Computación paralela*

- ◆ Hacer un trabajo  $n$  veces más rápido usando  $n$  procesadores
- ◆ **Programación**: lenguajes paralelos vs compiladores paralelizantes
- ◆ **Herramientas**: depurador, manejador de colas, administración
- ◆ **Inconvenientes**: paralelizaciones imperfectas, dependencias de datos



# *Tendencia de arquitecturas HPC*



BlueGeneL, LLNL,  
280 Tflops, 131K CPU

- ◆ **Clusters:**  
computadoras genéricas unidas por red de alta velocidad y bajo retardo
- ◆ Clusters y MPP dominan la lista TOP500



# *Tendencia de arquitecturas HPC: Clusters*

- ◆ Surgen en 1980
  - ◆ Los más rápidos, en el año 2000, fueron 8000 procesadores RS/6000 (ASCI White System) en el laboratorio nacional Lawrence Livermore.
  - ◆ Ofrecen una mejora considerable en poder de cómputo, pero es un conjunto de computadores dedicados, ubicados en un solo lugar.
  - ◆ En el 2000, el sistema ASCI White costó 110 millones de dolares y necesitó de un nuevo edificio (512 servidores RS 6000 IBM).



# *Ventajas de los clusters*



CUEH, Lab. CAR, USB

- ◆ Mejor relación costo/beneficio
- ◆ Escalabilidad
- ◆ Reutilización
- ◆ Disponibilidad de herramientas Open source (Ej. Gluster, Open MOSIX, etc.)



# *Tendencia de arquitecturas*

## *HPC: Computación en INTERNET*

- Existen alrededor de 400 millones de PC's en el mundo, muchas tan poderosas como los supercomputadores de los 90.
- La mayoría de estos PC's tienen mucho tiempo ocioso.
- La computación en Internet busca explotar los ciclos ociosos de workstations y PC's de modo de crear un sistema poderoso de computación distribuida.
- En 1985 Miron Livny mostró que la mayoría de las estaciones de trabajo permanecían ociosas y propuso el sistema **Condor** para utilizar ciclos libres de CPU. Es efectivo a pequeña escala:  
Universidades.



# *Tendencia de arquitecturas HPC: Computación en INTERNET*

En 1997 apareció **Entropía** (Scott Kurowski). Usa ciclos ociosos de computadores, alrededor del mundo, para resolver problemas de interés científico. En 2 años creció hasta llegar a 30000 computadoras que proveen una rapidez de cómputo por encima de un teraflop/seg. Usando este sistema se identificó el número primo más grande hasta ahora conocido

El próximo paso lo representa David Anderson con su proyecto **SETI@home**. Los ciclos ociosos se utilizan para analizar datos del telescopio de ARECIBO buscando signos que indiquen la presencia de inteligencia extraterrestre. Abarca alrededor del millón de PC's.



# *¡Quiero más!*

- ◆ Limitaciones: consumo de potencia limita agregación. Se habla de Potencia/superficie
- ◆ Solución: agregar recursos distribuidos geográficamente a través de redes de alta velocidad ...

## ***GRIDS COMPUTACIONALES***



# Introducción a grids

Evolución de la Tecnología: *Supercomputadores, Clusters, Computación ubicua*





# Introducción a grids

## Computación Ubicua

- ¿Dónde aplica el concepto de ubicuidad?
  - Los procesos se ejecutan en alguna plataforma elegida por el sistema (el hardware “está en todas partes”)
  - Se ofrece la misma imagen o entorno desde cualquiera de los lugares de trabajo (la misma interfaz “está en todas partes”)
- ¿Se puede hablar de subclases de Sistemas Distribuidos?  
Proponemos dos subclases, **Plataformas Grids** y **Sistemas P2P**:
  - No necesariamente disjuntas
  - Sistemas conformados por sistemas



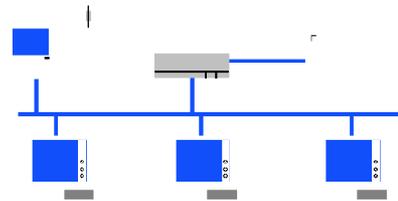
# Introducción a grids

## Computación Ubicua

- Grids Computacionales: se presupone que hay control sobre el acceso a los servidores.
  - ◆ Rendimiento
  - ◆ Transparencia
  - ◆ Tolerancia a fallas
  - Seguridad
- Sistemas P2P: se presupone que los nodos participantes no tienen control sobre el acceso a los mismos.
  - ◆ Cooperación
  - ◆ Independencia de localidad
  - ◆ Rodeo de controles
  - ◆ Anonimato



# Definición de Grids



- Unificación de recursos distribuidos geográficamente



# *Inicios de los Grids*

- Foster y Kesselman, 1998:
  - La computación en Internet es sólo un caso especial de algo mucho más poderoso: la necesidad de distintas organizaciones (ámbito científico) que buscan metas comunes de compartir recursos.
  - El web y el correo electrónico ofrecen mecanismos básicos que permiten a estos grupos trabajar juntos.
  - Lo ideal es agrupar datos, computadores, sensores y otros recursos en un único laboratorio virtual.
  - Los primeros conceptos se exploraron en 1995 (experimento I-WAY): se usaron redes de alta velocidad para conectar 17 sites en Norteamérica.



# Definición de Grids

**GRID** = Un mundo infinito de recursos

- Qué es:
  - Infraestructura de hardware y software que provee alto desempeño y alta disponibilidad
  - Colección de recursos (personas, computadores, instrumentos y bases de datos) conectados por una red de alta velocidad.
  - Mecanismo para que los usuarios puedan usar recursos distribuidos geográficamente de forma transparente, creando la ilusión de un sistema de computación integrado.



# Arquitectura

## Arquitectura de los Grids Computacionales

Aplicación de Servicios

### **Middleware**

- Enmascara la heterogeneidad
- Proporciona un modelo de programación

Sistema de Operación

Computadores y red de  
interconexión



# *Tipos de Grid*



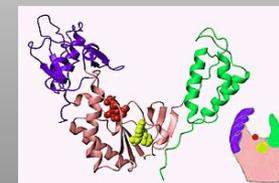


# Componentes de un grid computacional

- **Nodos, Sitios o Computadoras** – PCs, estaciones de trabajo, *clusters*, supercomputadores, laptops, dispositivos móviles, PDA, etc;



- **Software** – *middleware*, aplicaciones de propósitos especiales;





# Componentes de un grid computacional

Se presupone que hay control sobre el acceso a los servidores

- **Datos y Bases de datos** – Acceso transparente a datos clasificados: base de datos del genoma humano, bases de datos ontológicas, etc;
- **Dispositivos e instrumentos especiales** – radio telescopio, instrumentos de medicina;
- **Gente/colaboradores.**



**La integración de estos elementos permiten la creación de “organizaciones virtuales”**



# Organizaciones Virtuales

- La tecnología de GRID y en particular las organizaciones virtuales (VO) cambiaron radicalmente la forma cómo se resuelven los problemas.
- Las VO hacen posible la existencia de **grupos diversos de organizaciones y/o individuos** para **compartir recursos en forma controlada**, de forma tal que los miembros puedan **colaborar para lograr una meta común.**



# *Organizaciones Virtuales*

- El compartir recursos es condicional: cada propietario coloca sus recursos a la disposición de otros, sujeto a restricciones sobre dónde, cuándo y qué se puede hacer.
- El usuario también puede imponer restricciones sobre los recursos que desea.



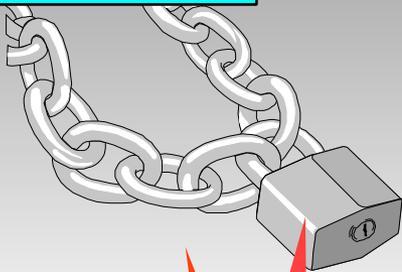
# *Organizaciones Virtuales*

- El compartir puede combinarse y se pueden estar usando a la vez varios recursos pertenecientes a distintas organizaciones.
- El mismo recurso puede estarse usando de distintas formas por distintas organizaciones virtuales.



# Desafíos en la construcción de grids

Seguridad



Acceso uniforme



Economía computacional



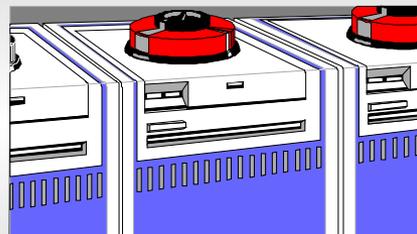
Administración de sistemas



Descubrimiento de recursos



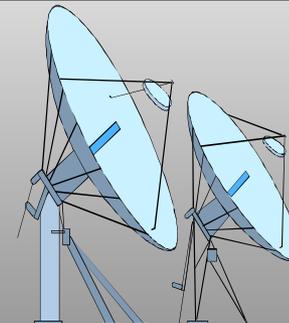
Asignación de recursos  
& Planificación



Localidad de datos



Construcción de aplicaciones



Admin. de redes



# Aspectos que caracterizan los grids

- **Múltiples dominios administrativos y autonomía:** los recursos de un Grid están geográficamente dispersos entre múltiples dominios administrativos y pertenecen a diferentes organizaciones. Es importante respetar la autonomía de los propietarios de los recursos en lo que respecta a la gestión de estos últimos y sus políticas de uso.
- **Heterogeneidad:** Un Grid involucra una gran variedad de recursos que son heterogéneos por naturaleza y abarcan un amplio rango de tecnologías.



# Aspectos que caracterizan los grids

- **Escalabilidad:** Un Grid debe crecer en un rango que va desde pocos recursos, hasta millones. Este crecimiento provoca problemas de desempeño. Como resultado, las aplicaciones que necesitan de un gran número de recursos deben diseñarse para tolerar posibles problemas de latencia y ancho de banda.
- **Dinamismo:** Las características de los recursos (carga, prioridad, disponibilidad, etc.) pueden cambiar en el tiempo. Tanto los administradores de recursos como las aplicaciones, deben ir al paso de este comportamiento dinámico y usar los recursos y servicios disponibles de una forma eficiente y efectiva.



# Aplicaciones para Grids

*Supercomputación distribuida*: usan las GRIDS para agrupar recursos que permitan resolver problemas que no pueden ser resueltos por un solo computador. Ejemplos: simulación interactiva distribuida (entrenamiento y planificación militar), simulación de procesos físicos complejos (cosmología, modelamiento del clima, etc.).

*High-Throughput computing*: la idea es planificar un gran número de tareas independientes o débilmente acopladas (poca sincronización) para aprovechar ciclos de CPUs ociosos. La naturaleza independiente de las tareas conduce a diferentes métodos para resolver el problema. Ejem: Problemas de **criptografía**, se usó durante la etapa crucial en el **diseño de los procesadores AMD K6 y K7, CONDOR.**



# Aplicaciones para Grids

- *Computación por demanda*: satisfacen requerimientos de recursos a corto plazo. Se trata de recursos a los que no se tiene acceso de manera local. Entre los recursos tenemos: BD, sensores, software, cómputo. Ejem: **NEOS y NetSolve**, envían a servidores remotos cálculos que necesitan mucho poder de cómputo o un software especializado.
- *Computación con un manejo intensivo de datos*: sintetizan información a partir de datos en repositorios distribuidos, librerías digitales y bases de datos. El proceso de síntesis es intensivo desde el punto de vista computacional y comunicacional.  
Ejem: **Experimentos Físicos, Fotografías Astronómicas.**

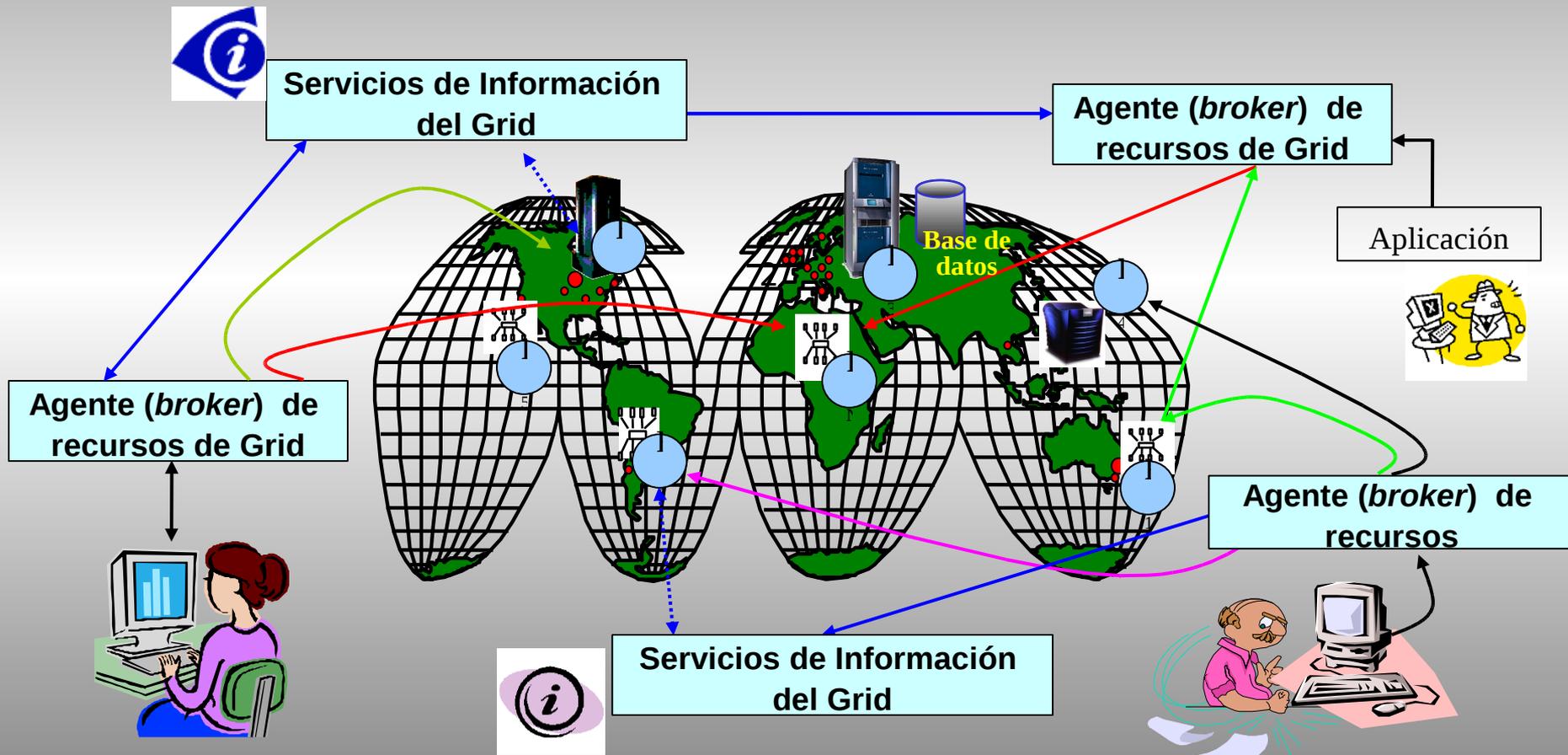


# *Aplicaciones para Grids*

- *Computación colaborativa*: permite interacción hombre-hombre, generalmente las aplicaciones están estructuradas en términos de un espacio virtual compartido.



# Ejemplo de un grid computacional: asignación y agregación de recursos





# Ejemplos de Middlewares

**Globus** (<http://www.globus.org>)



**gLite** (<http://glite.web.cern.ch/glite>)



Worldwide Virtual Computer  
*e pluribus unum: one out of many*

<http://legion.virginia.edu>

<http://www.unicore.org>

**UNICORE**



<http://www.gryds.net/suma>



# *Grids Computacionales en Venezuela*



(<http://www.gryds.net/suma>)

- Proyecto de grid computacional desarrollado en la USB
- *Middleware* para ejecución de aplicaciones en bytecode Java
- Transparentemente ejecuta procesos remotamente con la misma sintaxis del comando “java” pero usando “suma”
- Puede ejecutar programas paralelos, con `mpi_java` (un envoltorio de MPI)



- Otras características resaltantes
  - ejecución de programas interactivamente o fuera de línea (*batch*)
  - manejo de E/S por demanda, no hay que copiar la totalidad de los archivos de entrada al servidor
  - generación de perfiles de rendimiento (*profiles*) tanto para programas secuenciales como paralelos
  - tolerancia a fallas basada en puntos de recuperación (*checkpoints*) usando un algoritmo distribuido de identificación de estados estables



## ● *¿Por qué Java?*

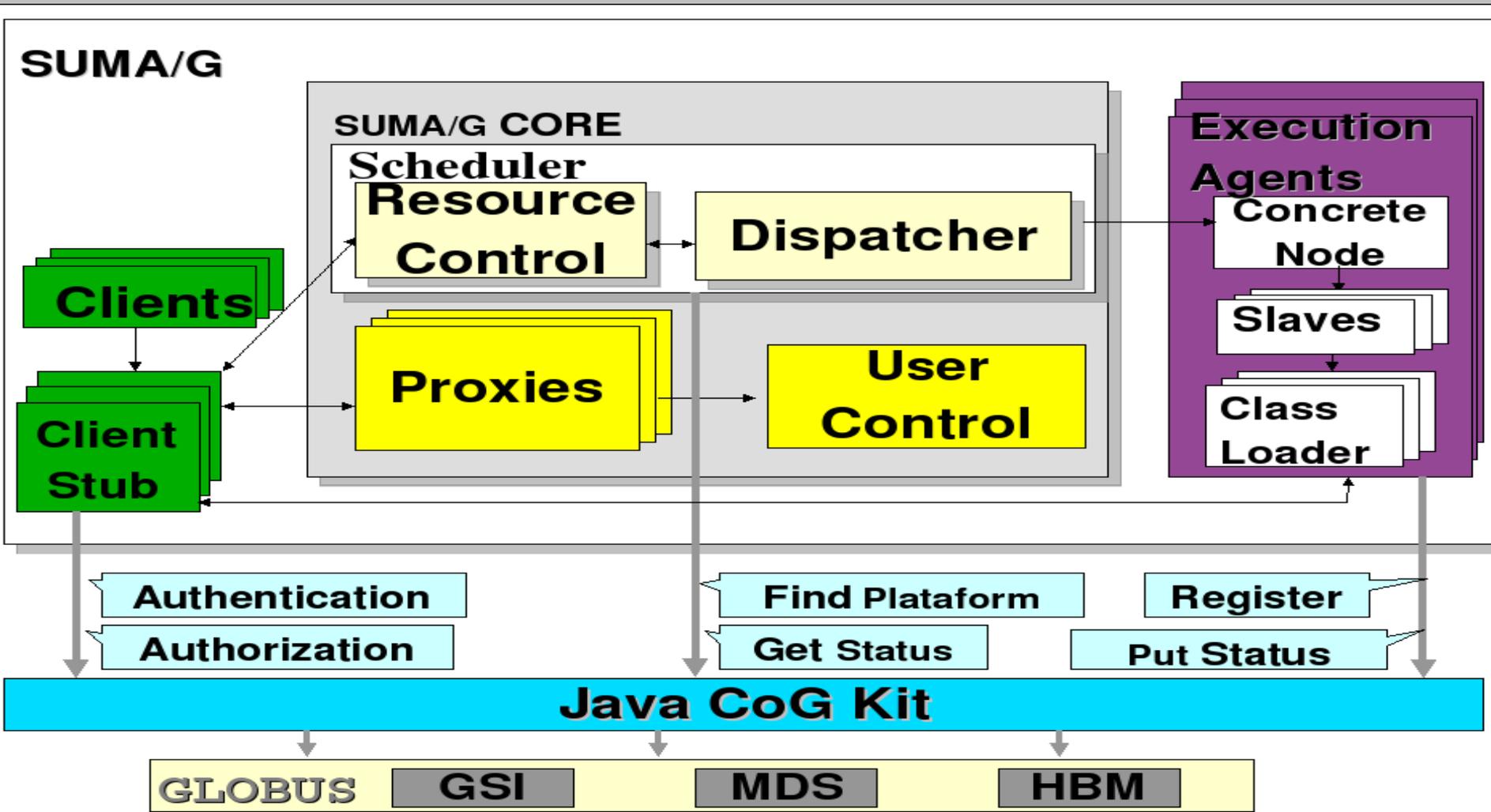
- Portabilidad: resuelve el principal problema de ejecución en máquinas heterogéneas
- Sociales: generaciones enteras de programadores bien entrenados y motivados
- El *Java Virtual Machine* puede ser usado como soporte para otros lenguajes
- *¿Eficiencia?*



- Componentes de SUMA
  - Clientes
    - Comandos de shell o clientes de ventana
    - Pueden generar múltiples procesos (p.e. *rendering*)
  - Sistema de transporte
    - encuentra un servidor de aplicaciones adecuado para la ejecución de cada petición, monitoreando el estado de los servidores y garantizando la ejecución de las aplicaciones
  - Agentes de ejecución, secuenciales o paralelos



- Construido sobre tecnologías estándares, flexibles, portables y bien conocidas.

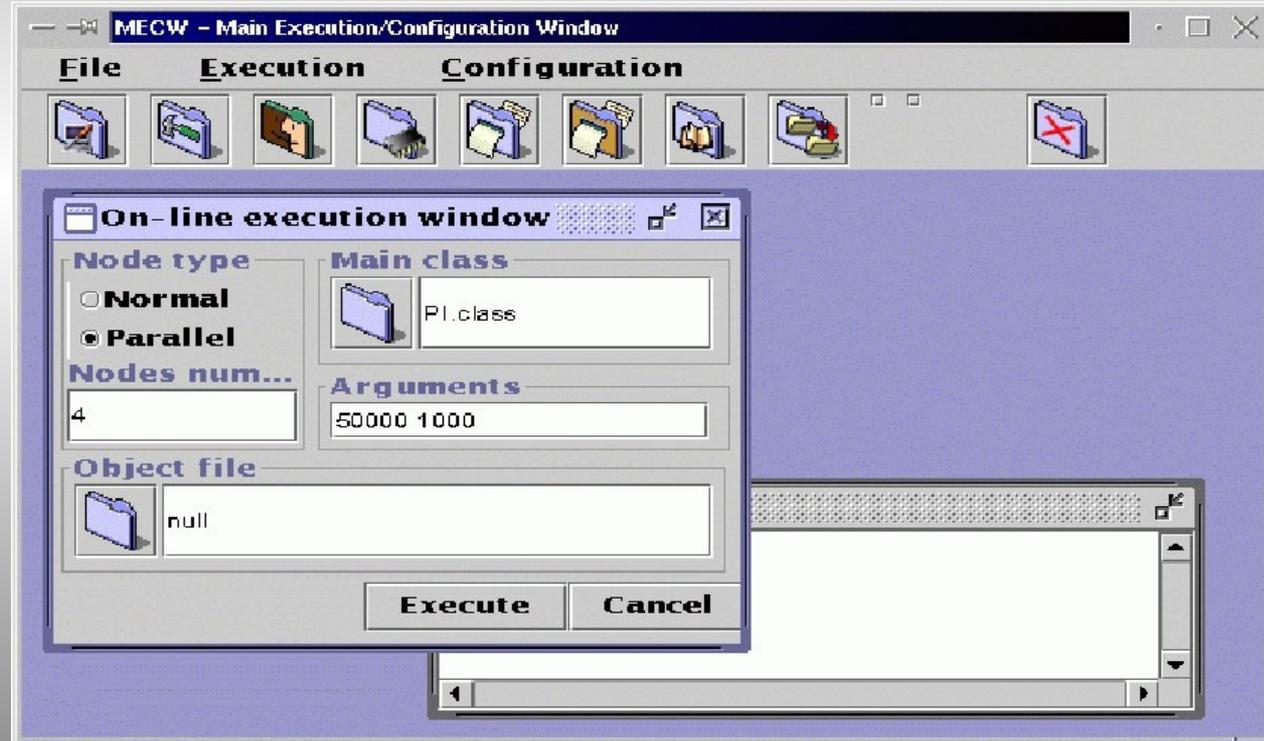


# Interacción con SUMA/G

- Por línea de comando:

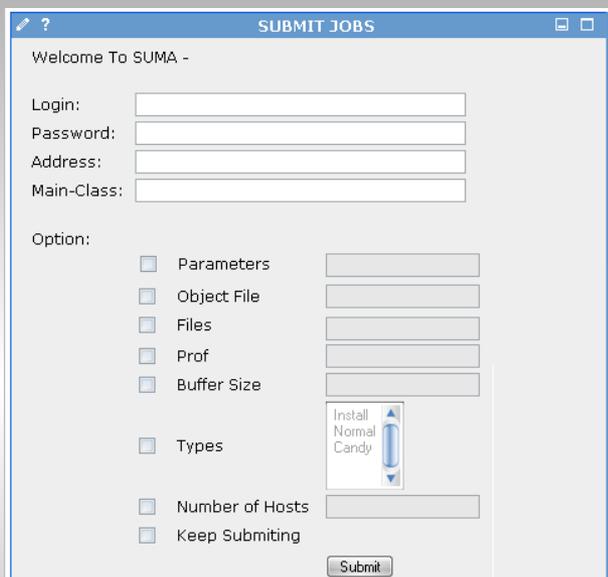
*sumag Execute P1 usuario passwd -t parallel -Num 4 args*

- Gráficamente:



# Interacción con SUMA/G

## ● A través de un Portal



Window title: SUBMIT JOBS

Welcome To SUMA -

Login:

Password:

Address:

Main-Class:

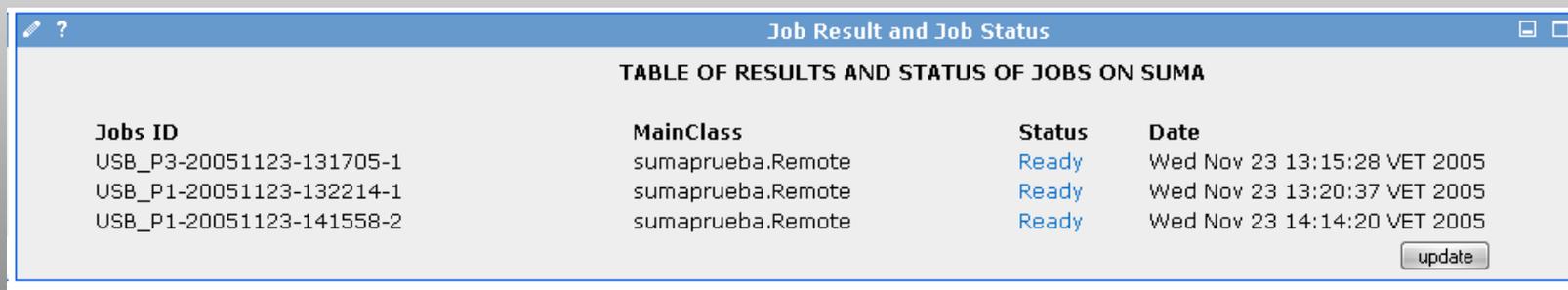
Option:

- Parameters
- Object File
- Files
- Prof
- Buffer Size
- Types
- Number of Hosts
- Keep Submitting

Un Portlet para el acceso y utilización del Servicio de Submit

GridPortlets, portlets de código y licencia abierta para el manejo de credenciales Proxy (usa Java CogKit para comunicarse con MyProxy)

Integración de los servicios JobResult y JobStatus en un único Portlet



Window title: Job Result and Job Status

TABLE OF RESULTS AND STATUS OF JOBS ON SUMA

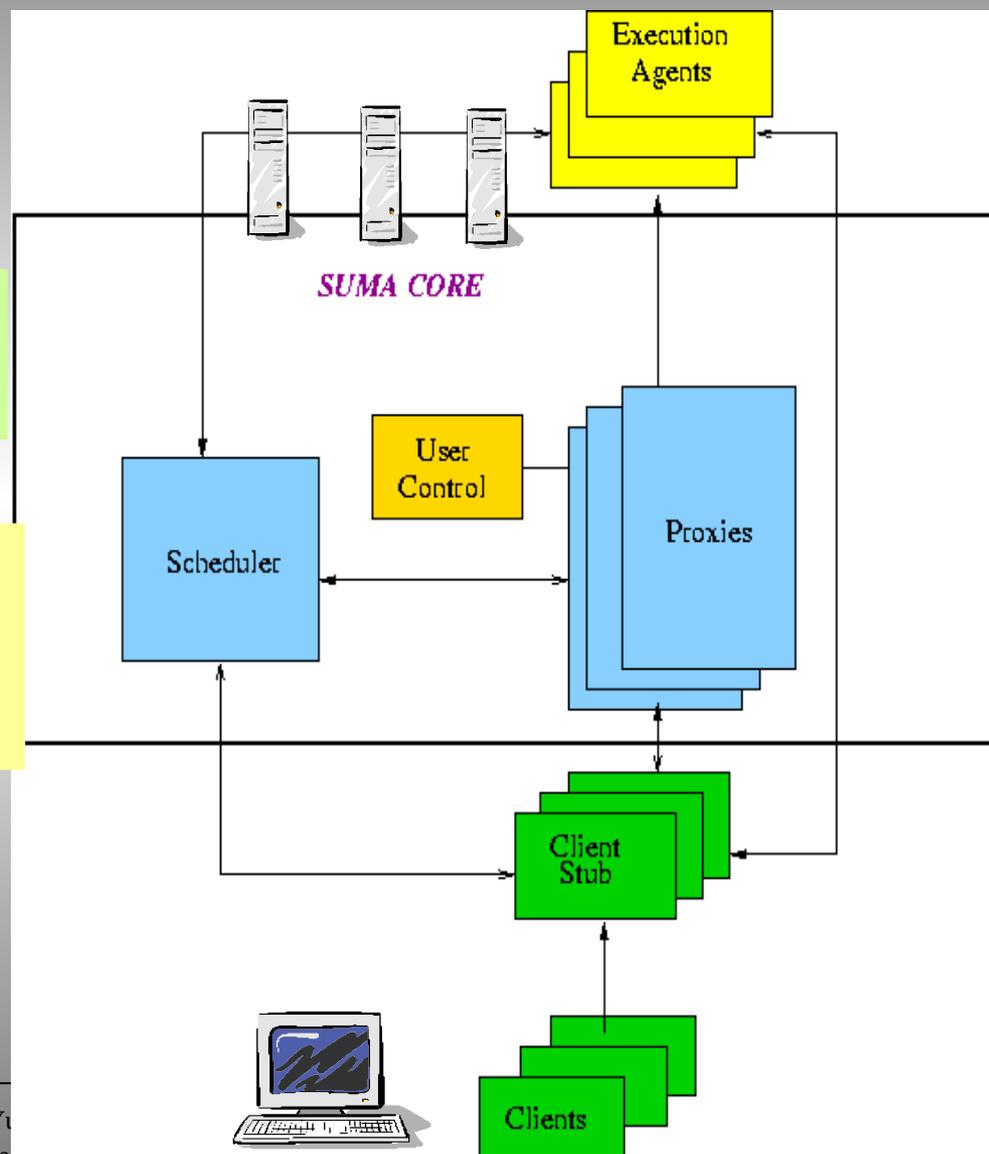
Jobs ID	MainClass	Status	Date
USB_P3-20051123-131705-1	sumaprueba.Remote	Ready	Wed Nov 23 13:15:28 VET 2005
USB_P1-20051123-132214-1	sumaprueba.Remote	Ready	Wed Nov 23 13:20:37 VET 2005
USB_P1-20051123-141558-2	sumaprueba.Remote	Ready	Wed Nov 23 14:14:20 VET 2005

Modelo de tres capas:

El **Ciente** se ejecuta en las máquinas de los usuarios

Los **Agentes de Ejecución** reciben y ejecutan las aplicaciones en los servidores

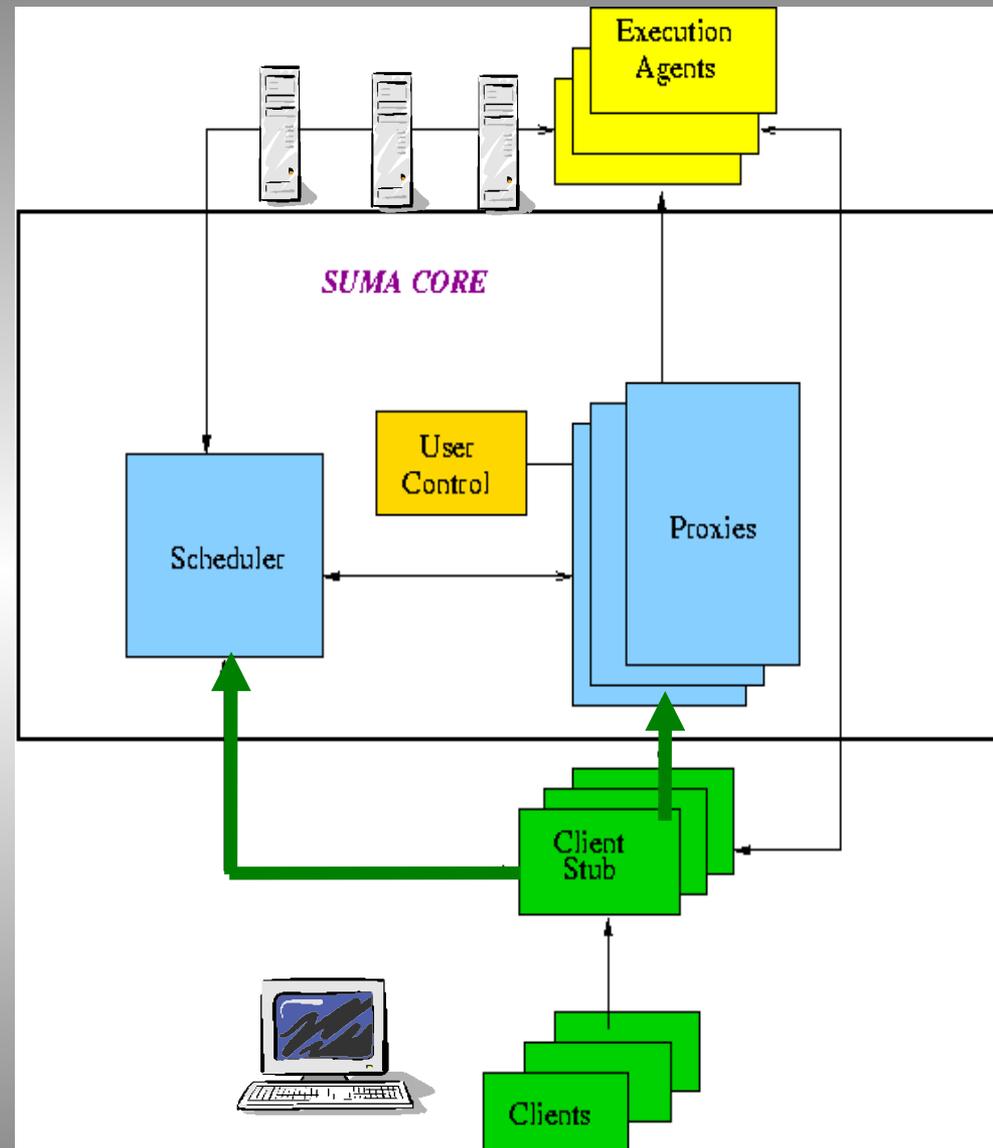
El **Núcleo** contiene componentes de apoyo (ej. el Planificador)





## Modelo de de ejecución:

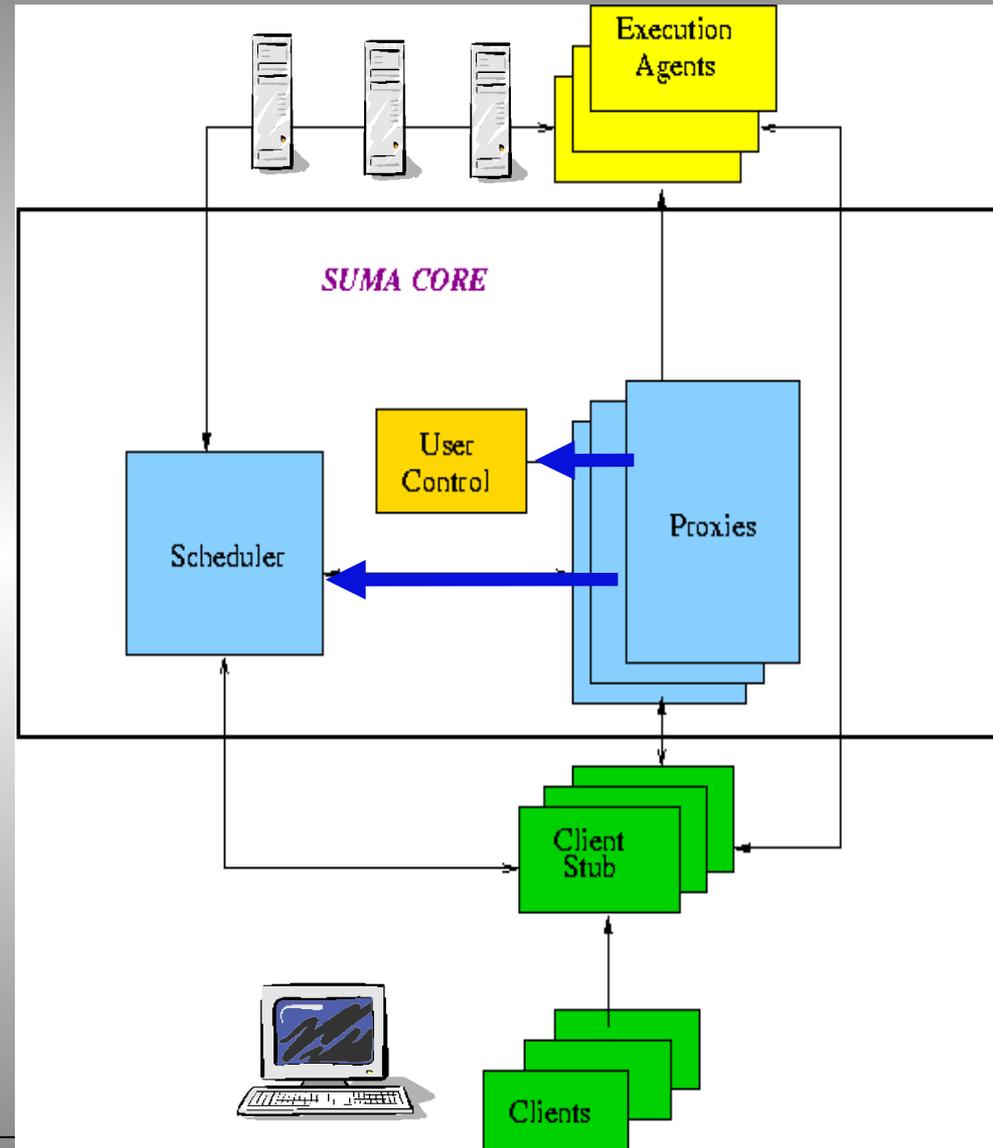
**Paso 1:** El *Client Stub* solicita un *Proxy* al *Scheduler*, luego el Cliente entrega solicitud de ejecución a ese *Proxy*





Modelo de ejecución:

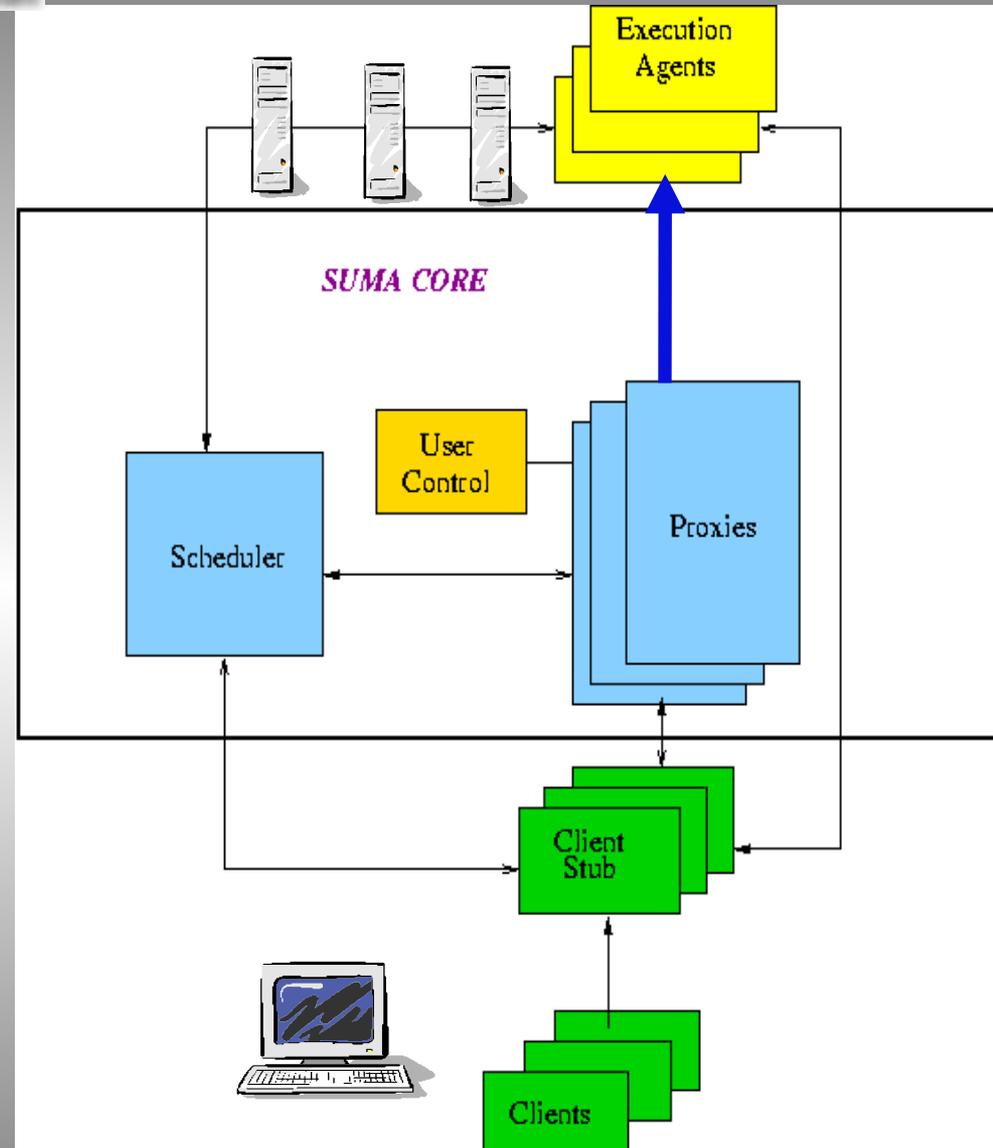
**Paso 2:** El *Proxy* autentica y verifica permisos del usuario, solicita al *Scheduler* un *Execution Agent*





Modelo de ejecución:

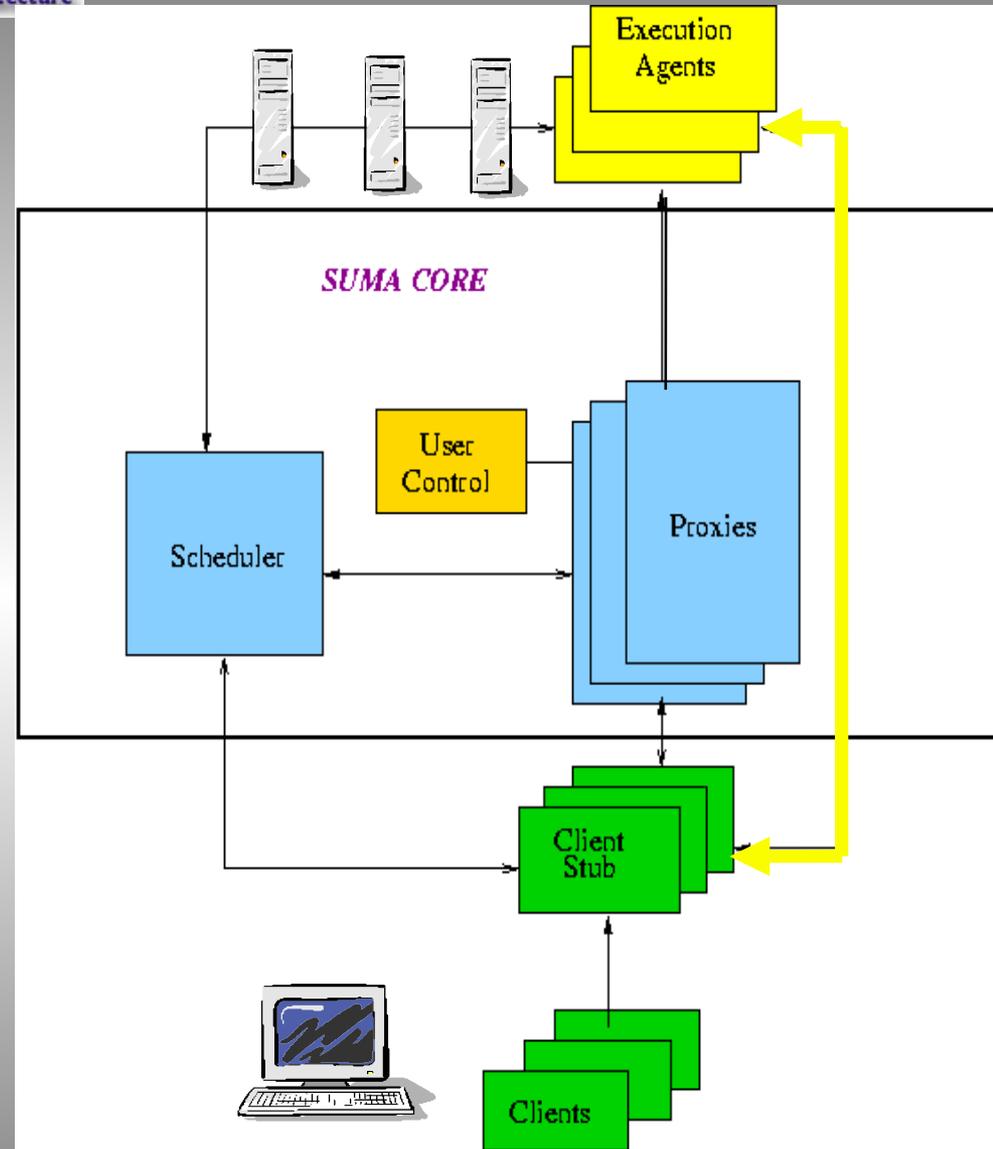
***Paso 3:*** El *Proxy* entrega solicitud al *Execution Agent*





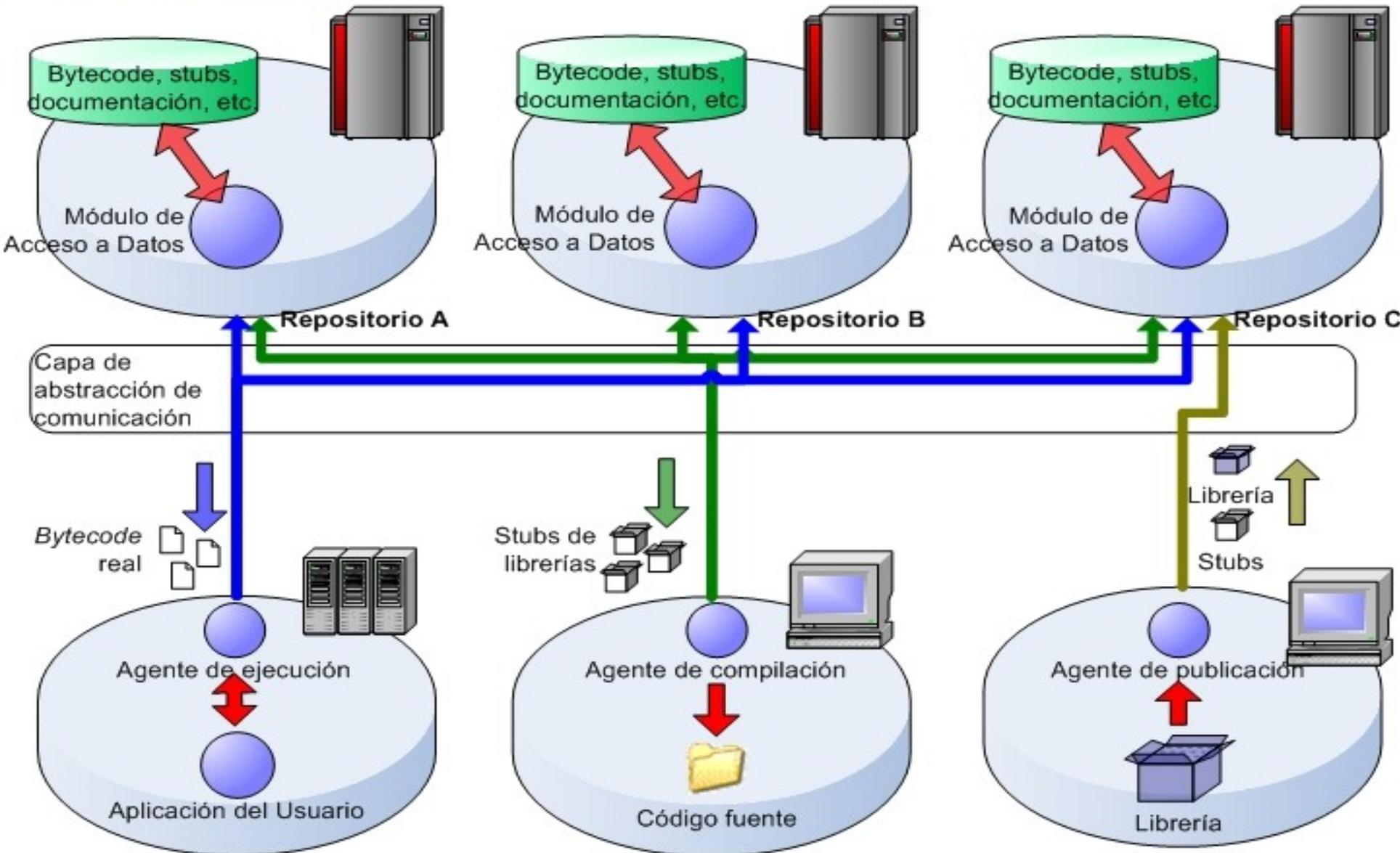
Modelo de ejecución:

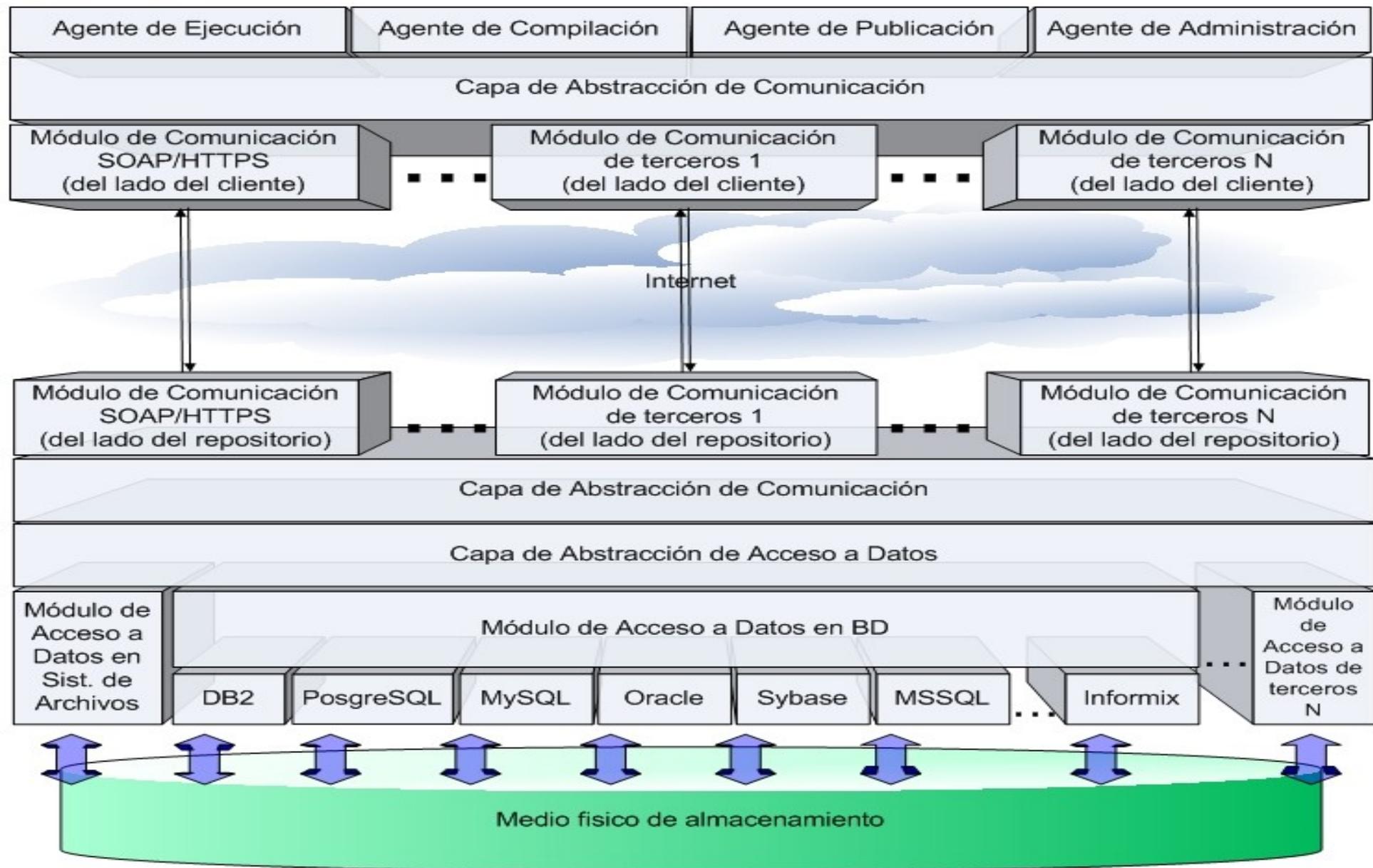
**Paso 4:** El *Proxy* devuelve al *Execution Agent* la referencia del cliente, quien establece conexión. La ejecución comienza, cargando dinámicamente las clases y datos de E/S.





# **Jadima: Administración de Librerías Java Distribuidas**

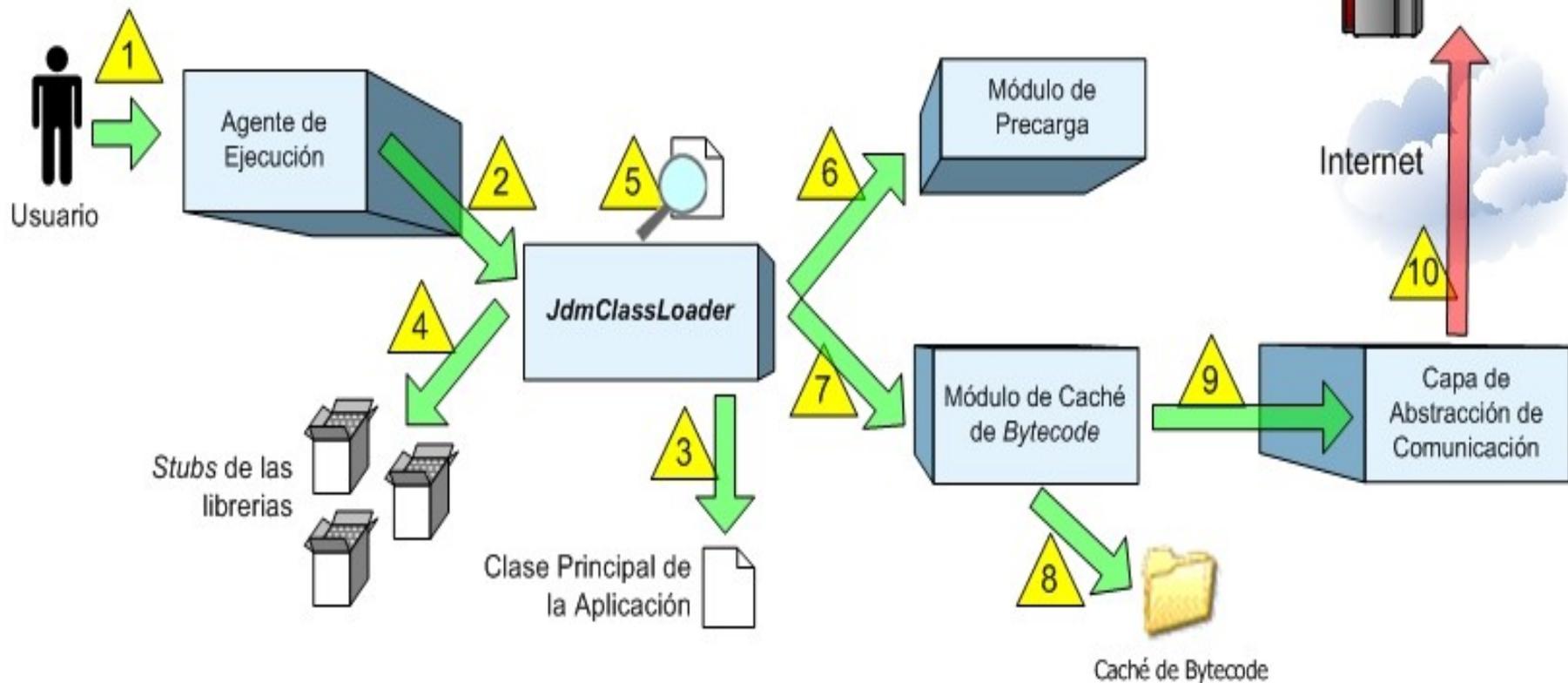






# JaDiMa

java distributed machine

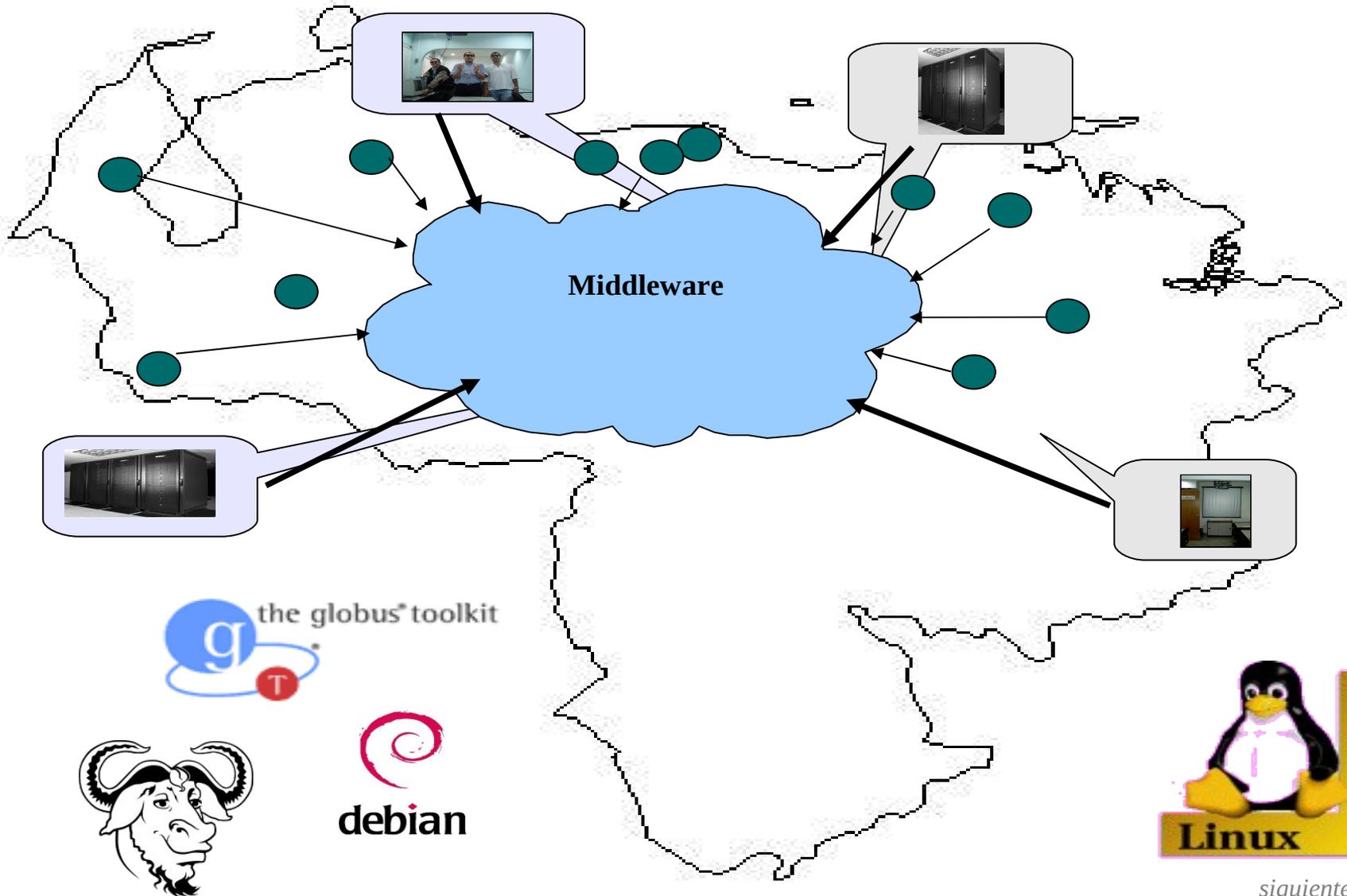




# ***Grid Nacional - Internet2***



# Grid Nacional - Internet2



siguiente



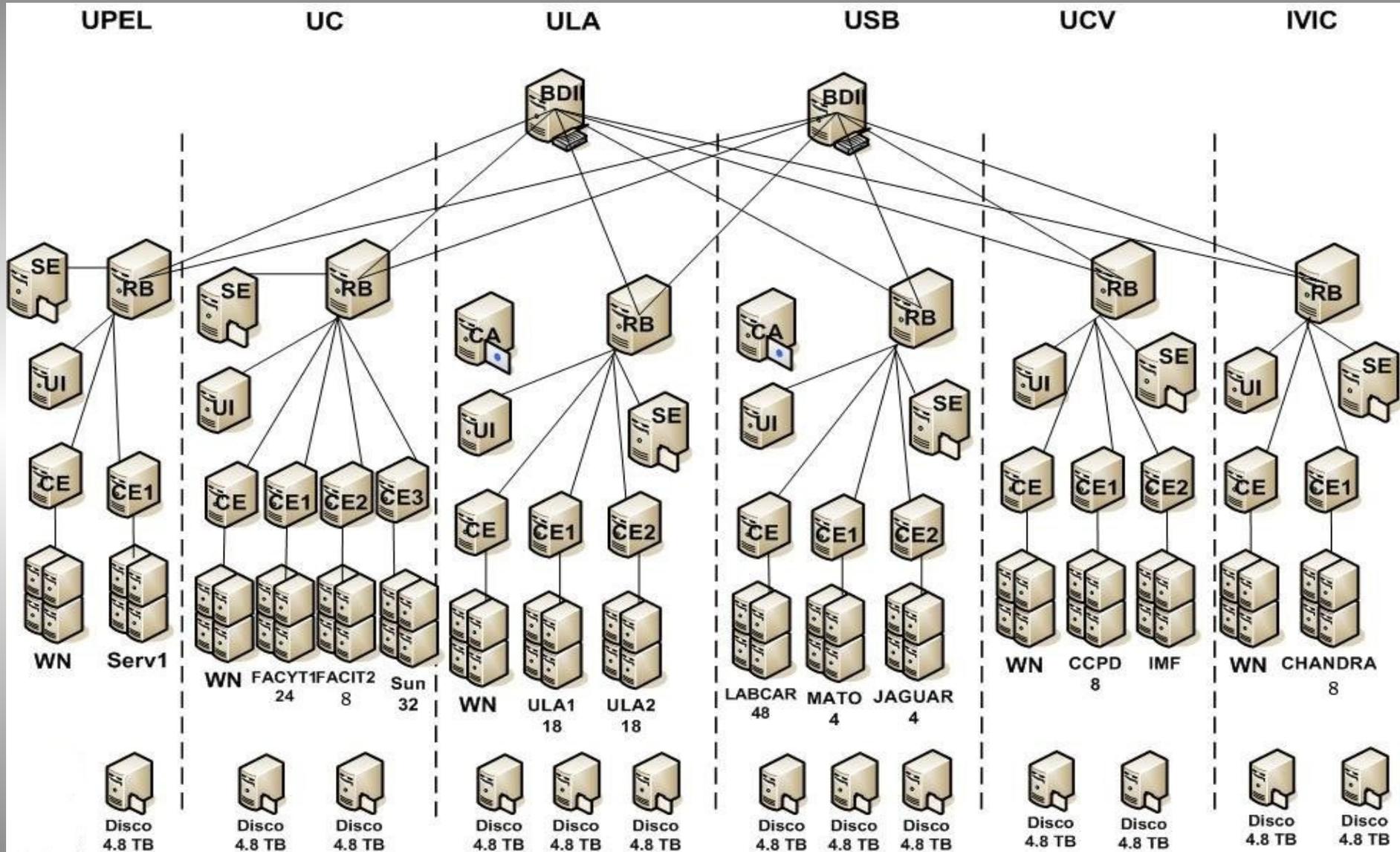


# **Grid Nacional - Internet2**

- *Formado por universidades y centros de investigación nacionales*
- *Tiene como objetivo colocar grandes repositorios de datos en línea, para investigación*
- *Creación de Centros de Acopio de Información en las instituciones participantes*
- *Convenios con otros centros de cómputo / almacenamiento, como TACC, PIC*  
*Utiliza el middleware EGEE / LCG*



# Grid Nacional - Internet2





# Grid Nacional - Internet2

Loggin GUSB - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://159.90.21.233/GUSB/

Red Hat, Inc. Red Hat Network Support Shop Products Training



## Portal GUSB

GUSB es un portal de prueba desarrollado por la [Universidad Simon Bolivar](#) para permitir el uso de los recursos del Grid ULA-USB.

Autenticación de Usuarios

Usuario:

Password:

Ok



Portal **GUSB** es de uso Exclusivo de la [Universidad Simon Bolivar](#) Contacto: [car@labf.usb.ve](mailto:car@labf.usb.ve)

Done

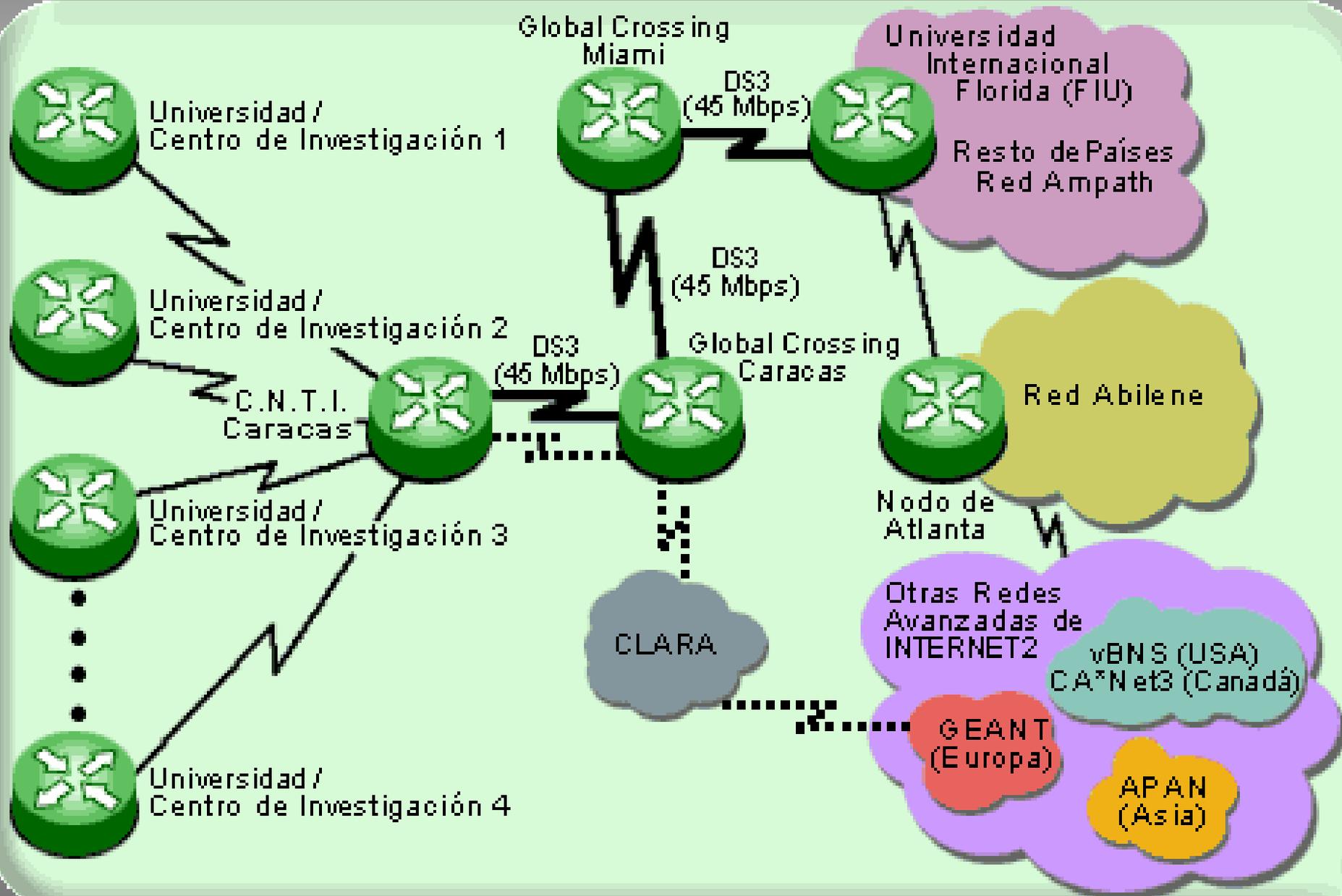


Loggin GUSB - Mozilla Firefox

Mon Mar 20  
4:16 PM



# Grid Nacional - Internet2





# Grid Nacional - Internet2

