

# Computación Distribuida

## Parte II: Computación Grid

---

**Juan Ángel Lorenzo del Castillo**  
**Grupo de Arquitectura de Computadores**  
**Departamento de Electrónica y Computación**  
**Universidad de Santiago de Compostela**

# Índice

---

- Supercomputación clásica
- Grid Computing
- Arquitectura de un Grid
- Middleware para Grid
- Ejemplos de infraestructuras Grid reales

# Índice

---

- Supercomputación clásica
  - Limitaciones de la supercomputación clásica
  - Alternativas a la supercomputación clásica
  - Explotación distribuida de los recursos
  - Problemas actuales en supercomputación
- Grid Computing
- Arquitectura de un Grid
- Middleware para Grid
- Ejemplos de infraestructuras Grid reales

# Limitaciones de la supercomputación clásica

- Aplicaciones que requieren potencia computacional
  - *Alto Rendimiento*: Ejecución de una aplicación en menos tiempo (paralelización previa).
  - *Alta Productividad*: Ejecución de un número mucho mayor de aplicaciones.
  - *Alta Disponibilidad*: Sistema disponible la mayor parte del tiempo posible.

- Solución clásica: Computación centralizada basada en servidor.



# Limitaciones de la supercomputación clásica

---

- Problemas de la supercomputación basada en servidor:
  - Falta de escalabilidad
  - Equipos muy caros
  - Mantenimiento muy caro
  - Las demandas de cálculo podrían ser puntuales
  - Posibles problemas de fiabilidad

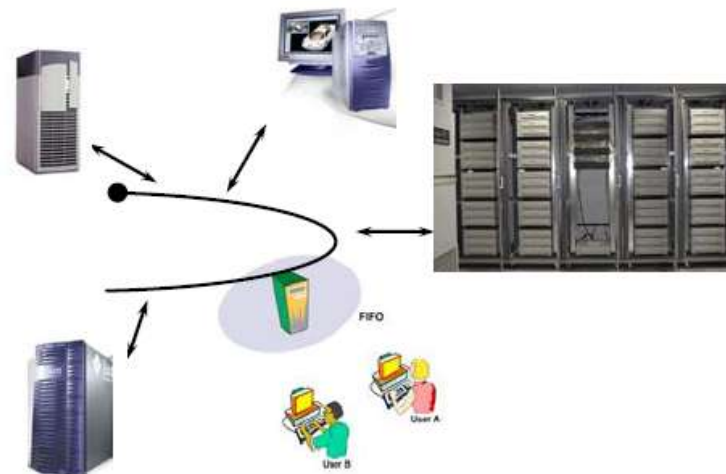
# Alternativas a la supercomputación clásica

- Cluster de estaciones o computadores personales homogéneo dedicado a computación paralela
  - Ventajas:
    - Mejor relación coste/rendimiento (3-10 veces)
  - Inconvenientes:
    - Coste de las comunicaciones muy alto
      - Bus lento
      - Acceso secuencial al bus
      - Sobrecarga TCP/IP
    - Mantenimiento
    - Modelo de programación
  - Conclusión: Buena solución para aplicaciones HTC/HPC



# Explotación distribuida de los recursos

- ¿Cuántas horas al día permanecen desaprovechados mis recursos?
- ¿Si tengo un pico de demanda de CPU puedo utilizar mis equipos distribuidos?
- **Solución:** Utilización de los equipos de una red departamental para ejecutar trabajos secuenciales o paralelos por medio de una herramienta de gestión de carga



# Explotación distribuida de los recursos

---

- Ventajas:
  - Aumentar el aprovechamiento de los recursos informáticos
  - Ciclos de CPU a coste bajo
  - Mejora de la escalabilidad
  - Mejora de fiabilidad
  - Facilidad de administración
  - Facilidad de sustitución de equipos obsoletos
- Software disponible:
  - Sun Grid Engine de Sun Microsystems ([www.sun.com/gridware](http://www.sun.com/gridware))
  - LSF de Platform Computing ([www.platform.com](http://www.platform.com))
  - Condor de la Universidad de Wisconsin ([www.cs.wisc.edu/condor](http://www.cs.wisc.edu/condor))



# Explotación distribuida de los recursos

- Inconvenientes:
  - No pueden gestionar recursos fuera del dominio de administración. Algunas herramientas (Condor, LSF o SGE) permiten colaboración interdepartamental asumiendo la misma estructura administrativa
  - No respetan las políticas de seguridad y de gestión de recursos de las organizaciones
  - Protocolos e interfaces básicos no basados en estándares abiertos
  - El único recurso que gestionan es la CPU. ¿Qué ocurre con los datos compartidos entre organizaciones?
- Por tanto:
  - Escalabilidad limitada a la organización en picos de demanda
  - No puedo amortizar mis recursos cuando están desaprovechados
  - No puedo compartir recursos con otras organizaciones

# Explotación distribuida de los recursos

- Situación planteada:



- Cómo puedo salir de los límites de mi centro o laboratorio?
- ¿Puedo ceder mis recursos a otras organizaciones?
- ¿Puedo compartir recursos con otras organizaciones?

# Problemas actuales en supercomputación

## 1. Necesitamos más potencia de cálculo

- La potencia de los supercomputadores sólo crece linealmente (dos órdenes de magnitud cada 10 años, según [www.top500.org](http://www.top500.org))
- La capacidad de almacenamiento de un ordenador se dobla cada 12 meses. El ancho de banda de red, cada 9 meses. El rendimiento de un procesador, cada 18 meses.

### **Conclusiones:**

Un único sistema no será capaz de analizar los datos que almacenen sus discos.  
Un único centro no podrá analizar el volumen de información generado.  
La red permitirá de forma eficiente usar recursos distribuidos.

# Problemas actuales en supercomputación

2. Los equipos son caros y difíciles de mantener
  - ¿Cómo puedo amortizar la inversión realizada?
3. ¿Qué hago si tengo un pico de demanda de cálculo y no tengo presupuesto para adquirir un supercomputador?
4. Desequilibrio de carga en el tiempo
  - ¿Cómo puedo conseguir una carga más homogénea?

**Solución:**  
**Grid Computing**

# Índice

---

- Supercomputación clásica
- Grid Computing
  - Concepto
  - Definición de Grid
  - Beneficios del Grid
  - Desafíos técnicos y socioeconómicos
  - Estado actual
- Arquitectura de un Grid
- Middleware para Grid
- Ejemplos de infraestructuras Grid reales

# Grid Computing

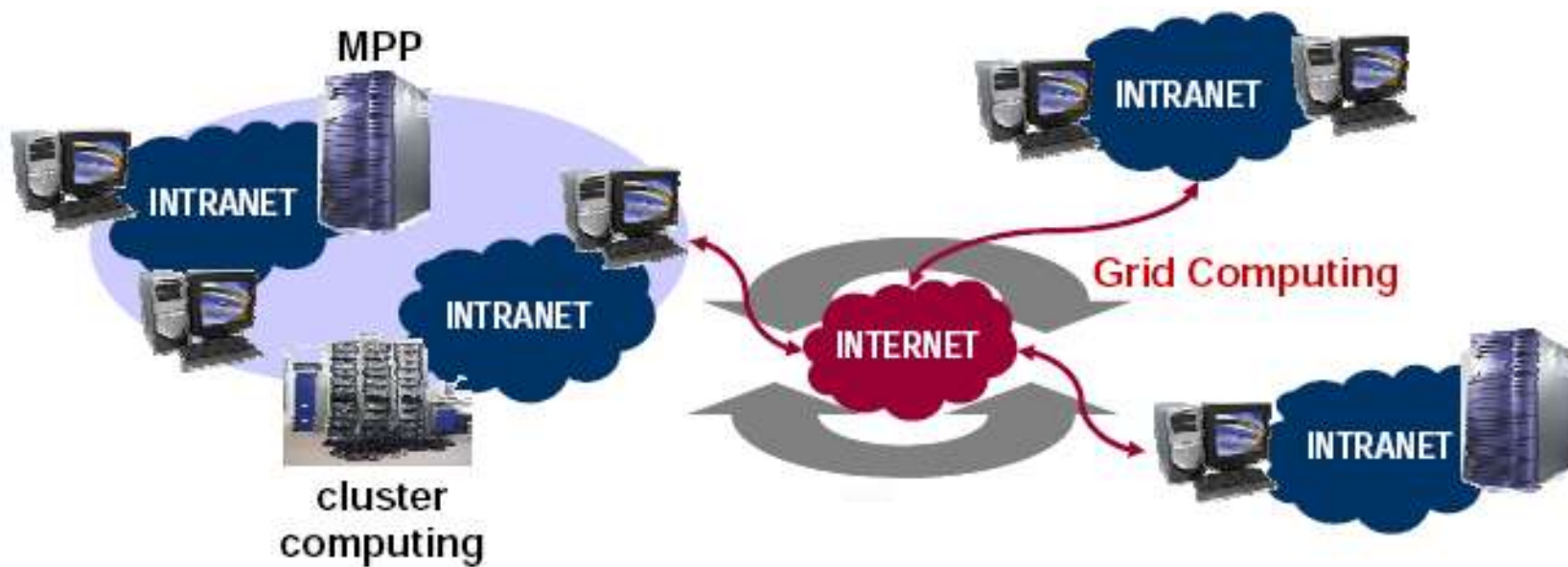
- Nueva tecnología cuyo objetivo es la **compartición de recursos** en Internet de forma uniforme, segura, eficiente y fiable.
- Análoga a las redes de suministro eléctrico (*Power Grid*):
  - Ofrecen un único punto de acceso a un conjunto de recursos distribuidos geográficamente en **diferentes dominios de administración** (supercomputadores, clusters, almacenamiento, fuentes de información, instrumentos, personal, bases de datos...)
- La tecnología Grid es complementaria a las anteriores
  - Permite interconectar recursos en diferentes dominios de administración respetando sus políticas internas de seguridad y su software de gestión de recursos en la Intranet.

# Grid Computing

---

- Definición de Grid
  - Infraestructura hardware y software geográficamente distribuida
  - Gran escala
  - Recursos heterogéneos
  - Múltiples organizaciones administrativas
  - Utiliza protocolos e interfaces estándar, abiertos y de propósito general
  - **Objetivo:** proporcionar soporte computacional a un amplio rango de aplicaciones de forma transparente, de calidad, ubicuo, etc., con calidades de servicio no triviales

# Grid Computing





# Grid Computing

---

- Beneficios del Grid
  - Alquiler de recursos (*Utility Computing*)
  - Amortización de recursos propios
  - Gran potencia de cálculo a precio bajo sin necesidad de adquirir equipamiento
  - Mayor colaboración y compartición de recursos entre varios centros
  - Creación de organizaciones virtuales
  - Negocios basados en proveer recursos

# Grid Computing

---

- Desafíos técnicos
  - Recursos heterogéneos
  - Descubrimiento, selección, reserva, asignación, gestión y monitorización de recursos
  - Desarrollo de aplicaciones
  - Desarrollo de modelos eficientes de uso
  - Comunicación lenta y no uniforme
- Desafíos socioeconómicos
  - Organizativos: Dominios de administración, modelo de explotación y costes, políticas de seguridad...
  - Económicos: Precio de los recursos, oferta/demanda...

# Grid Computing

---

## **Estado Actual:**

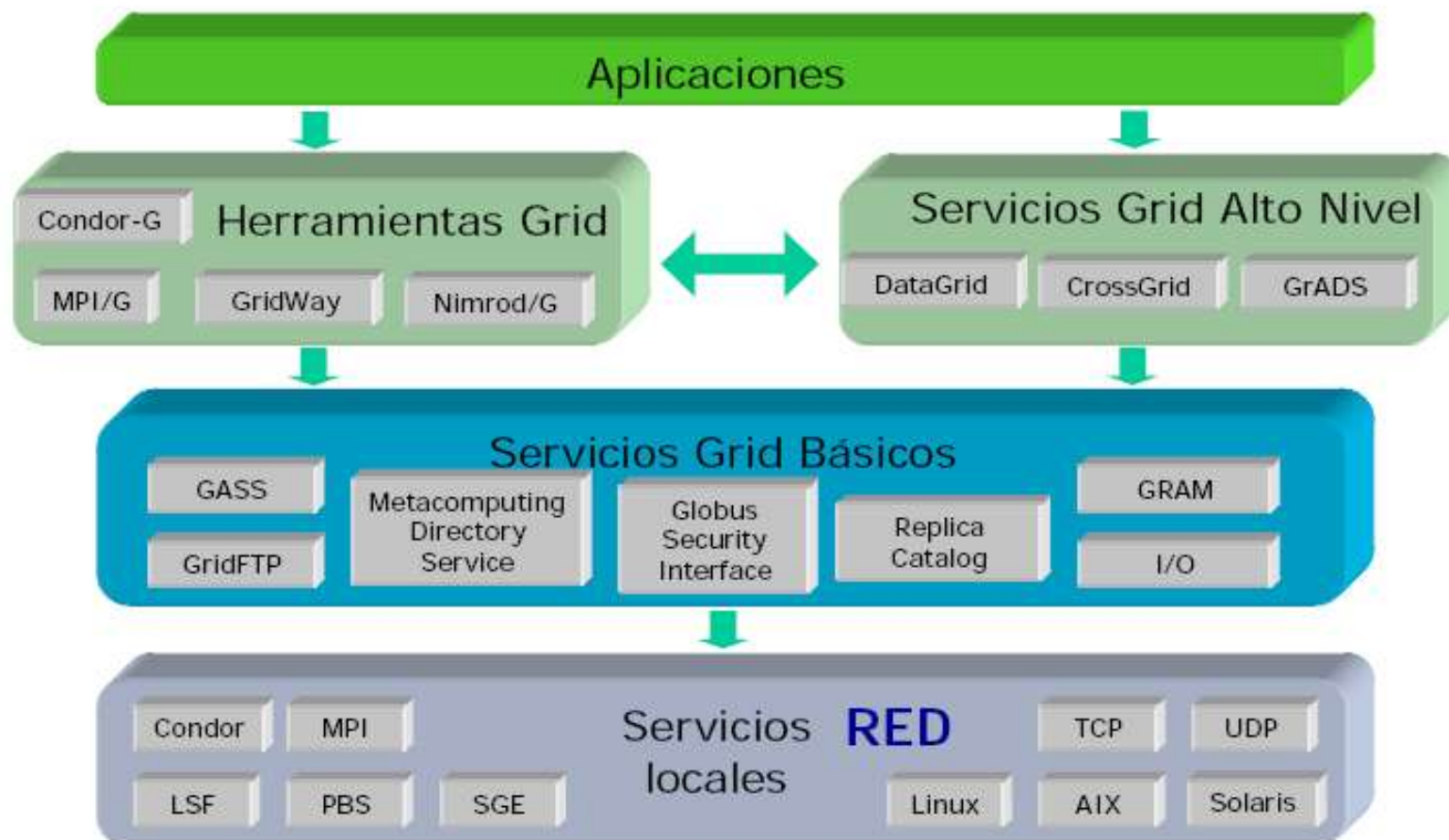
Lejos de satisfacer las expectativas que la comunidad científica y la sociedad han puesto en él

# Índice

---

- Supercomputación clásica
- Grid Computing
- Arquitectura de un Grid
  - Esquema
- Middleware para Grid
- Ejemplos de infraestructuras Grid reales

# Arquitectura de un Grid



# Índice

---

- Supercomputación clásica
- Grid Computing
- Arquitectura de un Grid
- Middleware para Grid
  - Globus Toolkit
  - OGSA
  - WSRF
  - Componentes de Globus Toolkit 4
- Ejemplos de infraestructuras Grid reales

# Middleware Grid

---

- **Globus Toolkit**

- Estándar *de facto* para la implantación de infraestructuras Grid.
- Permite compartir recursos localizados en diferentes dominios de administración, con diferentes políticas de seguridad y gestión de recursos.
- Globus es...
  - Un middleware software
  - Un conjunto de librerías, servicios y APIs
- Globus no es...
  - Una herramienta de usuario o planificador
  - Una aplicación

# Middleware Grid

- Para entender qué es el Globus Toolkit, en primer lugar es necesario especificar el contexto en el que se enmarca:

OGSA + WSRF



# Middleware Grid

---

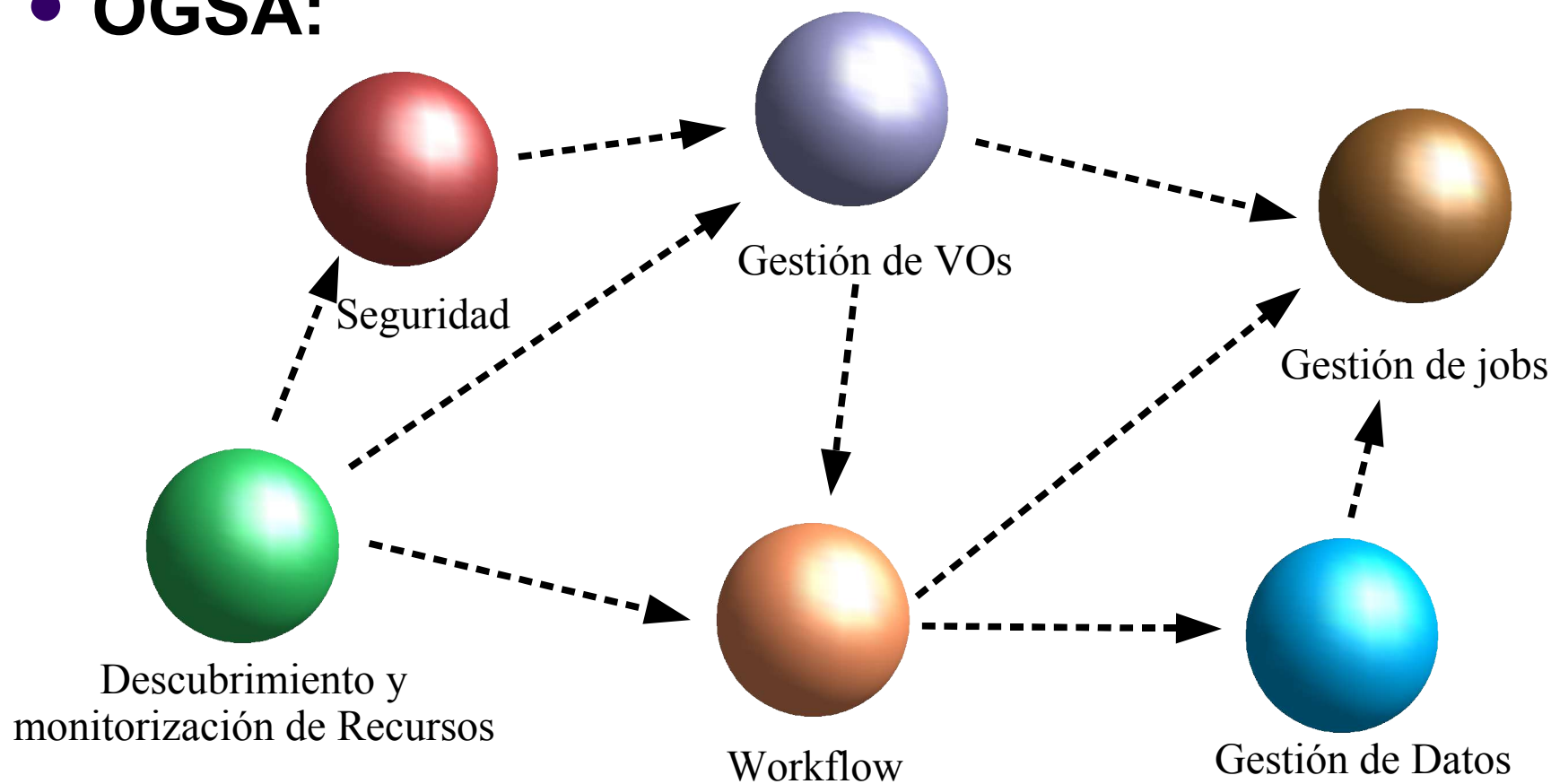
- **OGSA:**

Un sistema Grid está compuesto de varios subsistemas.

Por ejemplo:

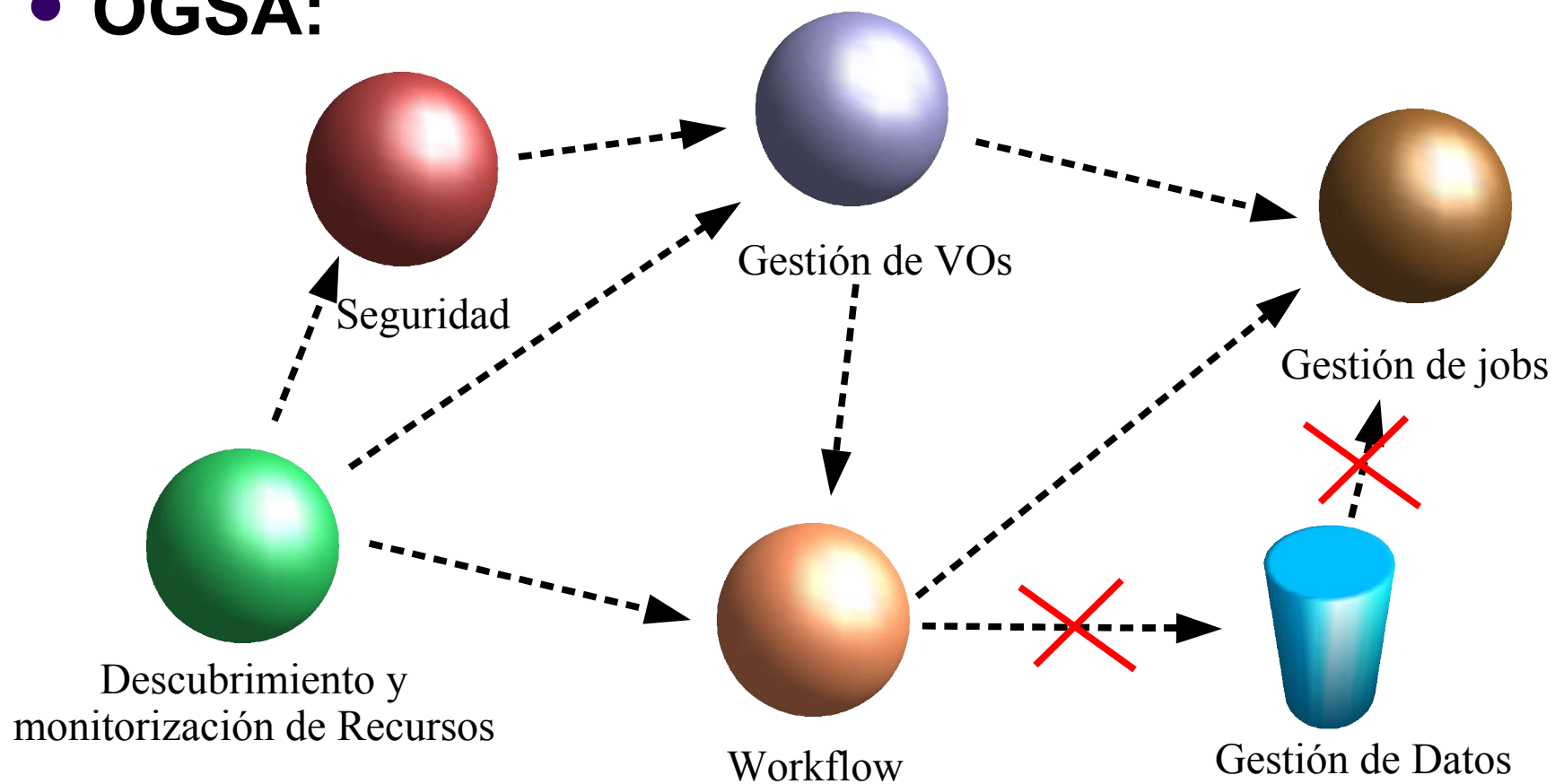
# Middleware Grid

- **OGSA:**



# Middleware Grid

- OGSA:



# Middleware Grid

---

- **OGSA:**

- El *Open Grid Services Architecture* (OGSA) tiene como objetivo estandarizar todos los distintos servicios que se encuentran en una aplicación Grid.
- El proceso de estandarización lo realiza el Global Grid Forum (GGF)
  - <http://www.ggf.org>

# Middleware Grid

- **OGSA:**

- Qué define exactamente OGSA?
  - “Una serie de interfaces y comportamientos básicos que afrontan tareas clave en sistemas Grid”
  - Con esto se consigue que todos los servicios puedan ser accedidos de la misma manera, independientemente de la plataforma, sistema operativo, organización, etc.
- ¿Qué hay “debajo” de OGSA?
  - ¿Cómo se definen los interfaces de los servicios?
  - ¿Cómo se realiza una petición a un servicio?

# Middleware Grid

---

- **OGSA:**

- Teóricamente, OGSA podría basarse en cualquier middleware distribuido (CORBA, RMI, servicios Web...)
- Los Servicios Web son los que mejor cumplen los requisitos de OGSA.
- ¿Y qué es un Servicio Web?

# Middleware Grid

---

- **OGSA:**

- Servicio:

- *“Manifestación lógica de recursos lógicos o físicos (BBDD, programas, dispositivos, personas, etc.) y/o algún tipo de lógica de aplicación que es expuesta a la red.”*
- La interacción entre servicios se realiza mediante intercambio de mensajes
- Las Arquitecturas Orientadas a Servicios (SOA) permiten implementar aplicaciones distribuidas débilmente acopladas entre sí

# Middleware Grid

---

- **OGSA:**

- Servicios Web:

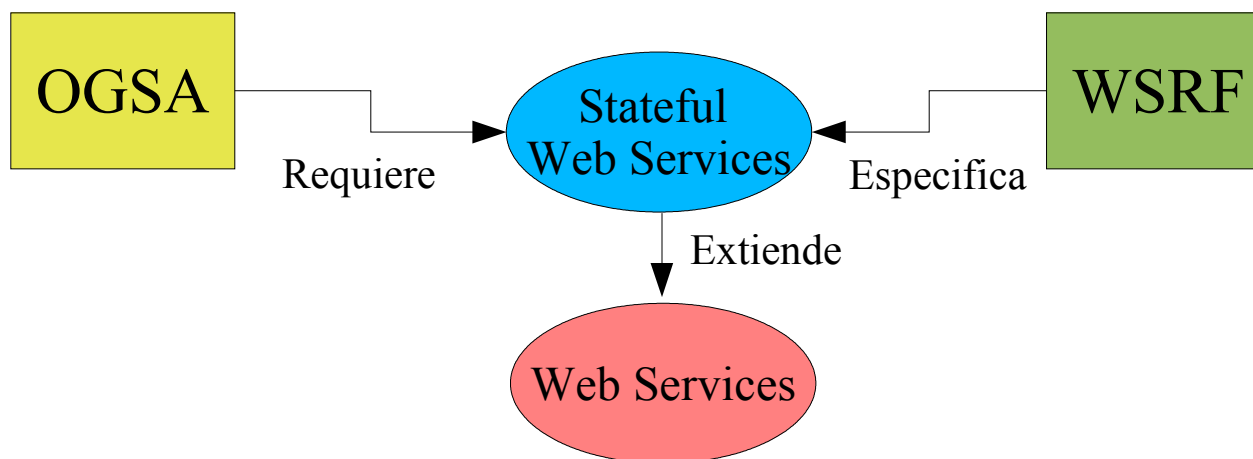
- Basados en principios de SOA, permiten definir bloques para construir aplicaciones distribuidas débilmente acopladas
      - SOAP: Intercambio de mensajes
      - WSDL: Definición de interfaces para describir los contratos relativos al intercambio de mensajes
    - Separación entre el *interfaz* (“lo que hace”) y la *implementación* (“cómo lo hace”). El interfaz está escrito en un lenguaje neutro, independiente del lenguaje de programación y de la plataforma.



# Middleware Grid

- **OGSA:**

- Los Servicios Web tienen una importante carencia: generalmente son *stateless* (no mantienen el estado entre peticiones) y no hay ninguna manera de hacerlos *stateful* (con estado).
- Solución: *Web Services Resource Framework* (WSRF), que extiende los Servicios Web para hacerlos *stateful*, sin romper la compatibilidad con otros Servicios Web.



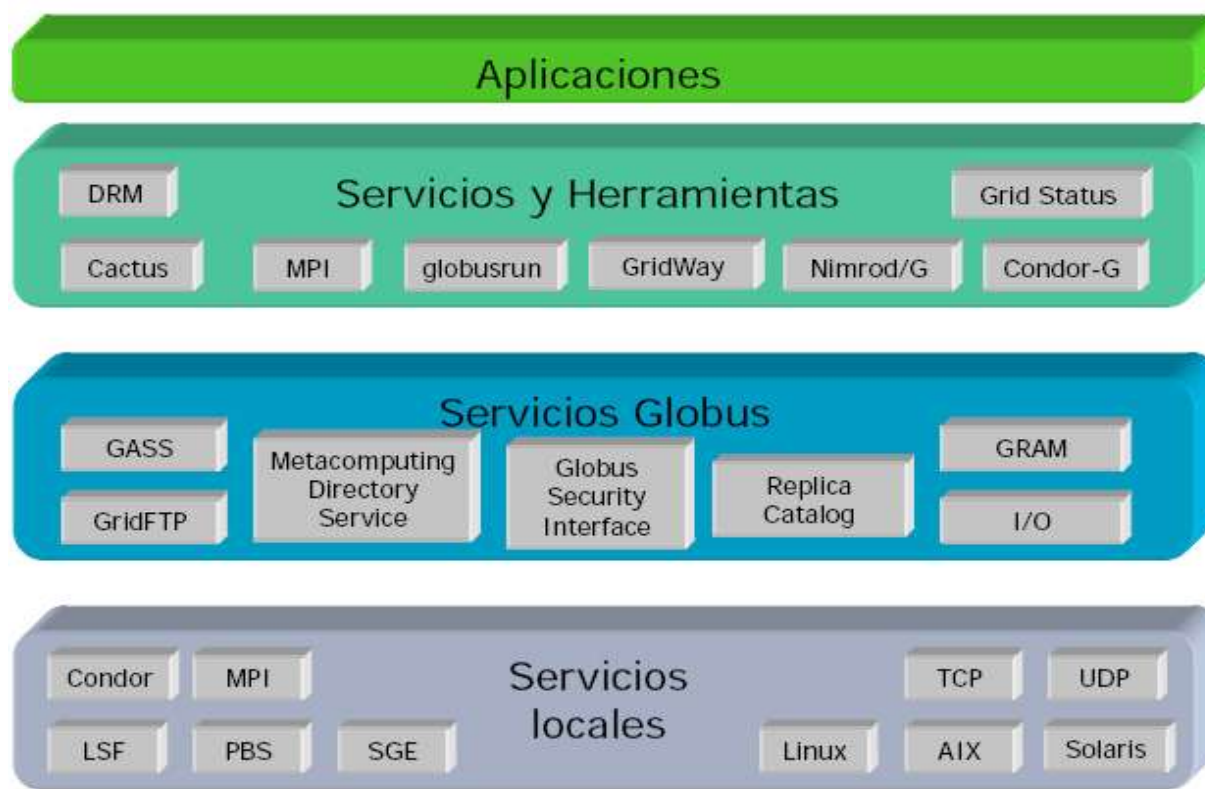
# Middleware Grid

---

- **OGSA + WSRF + GT4:**
  - Globus Toolkit 4 (GT4)
    - Creado por la Alianza Globus ([www.globus.org](http://www.globus.org))
    - Proporciona una serie de componentes que permiten crear sistemas y Servicios Grid.
    - GT4 *no* es una implementación de OGSA.
      - Los estándares del GGF todavía están en fase de desarrollo. Se puede decir que GT4 cumple muchos de los requisitos de OGSA.
    - GT4 incluye una implementación completa de WSRF
      - Existen más implementaciones de WSRF (p.ej: [WSRF.net](http://WSRF.net)).

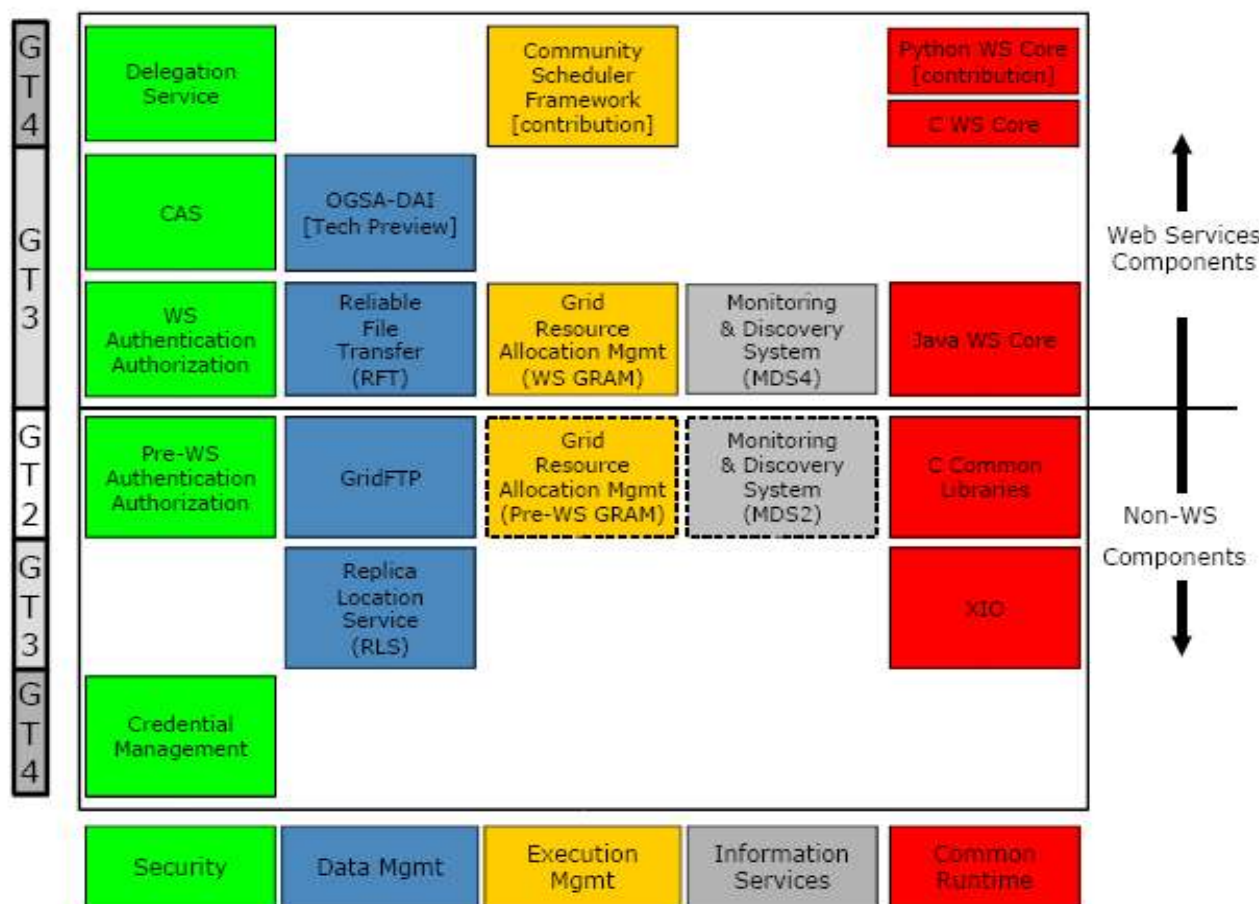
# Middleware Grid

- **GT4: Modelo de capas**



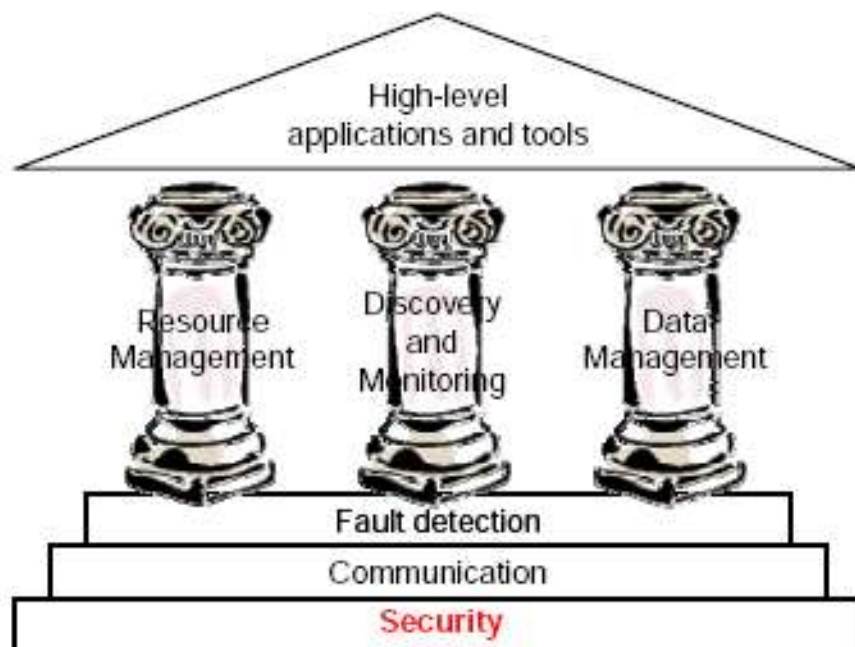
# Middleware Grid

- Globus Toolkit: Componentes



# Middleware Grid

- **Globus Toolkit: Componentes**
  - Seguridad



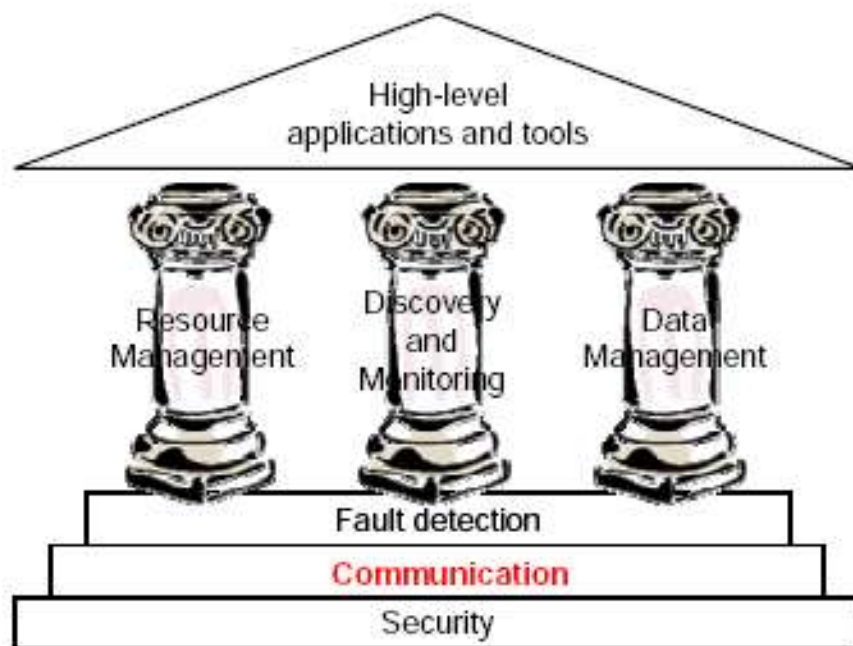
**GSI:** Globus Security Infrastructure

- TLS (OpenSSL)
- Public Key Infrastructure (PKI) + certificados X.509
- Generic Security Services API (GSS-API)

**CAS:** Community Authorization Service

# Middleware Grid

- **Globus Toolkit: Componentes**
  - Comunicación

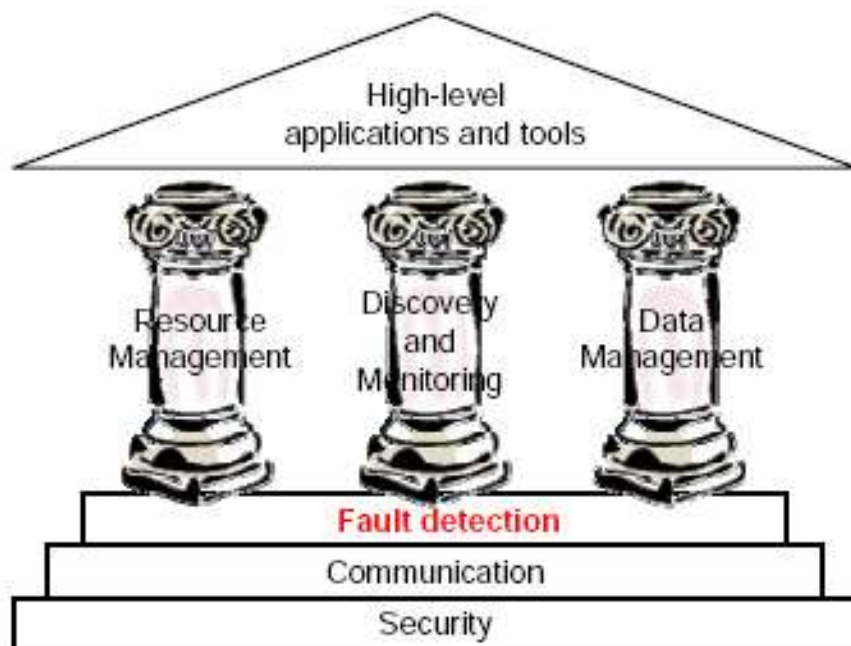


## Globus I/O

- Transport Control Protocol (TCP)
- User Datagram Protocol (UDP)
- File I/O

# Middleware Grid

- **Globus Toolkit: Componentes**
  - Detección de fallos

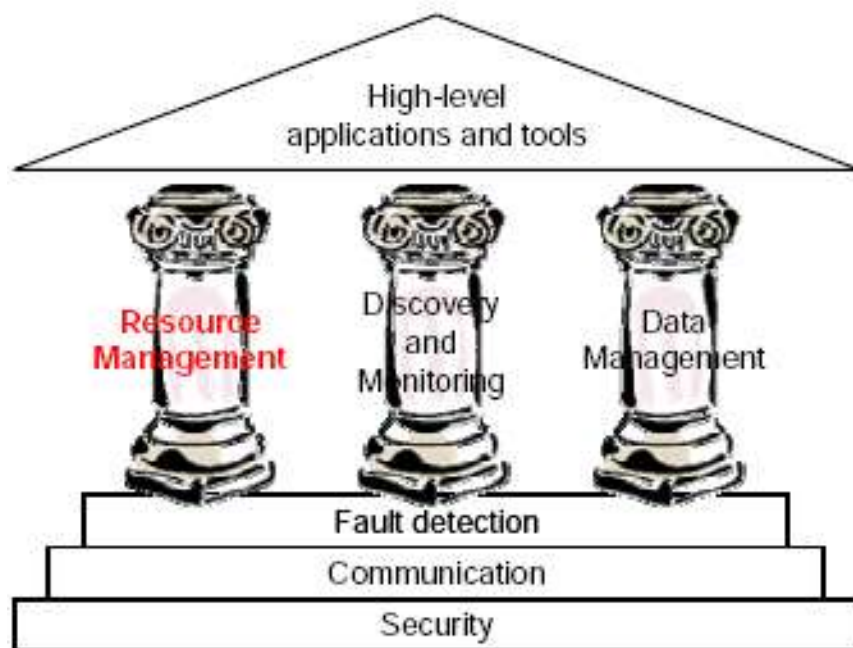


## Globus I/O

- Transport Control Protocol (TCP)
- User Datagram Protocol (UDP)
- File I/O

# Middleware Grid

- **Globus Toolkit: Componentes**
  - Gestión de Recursos

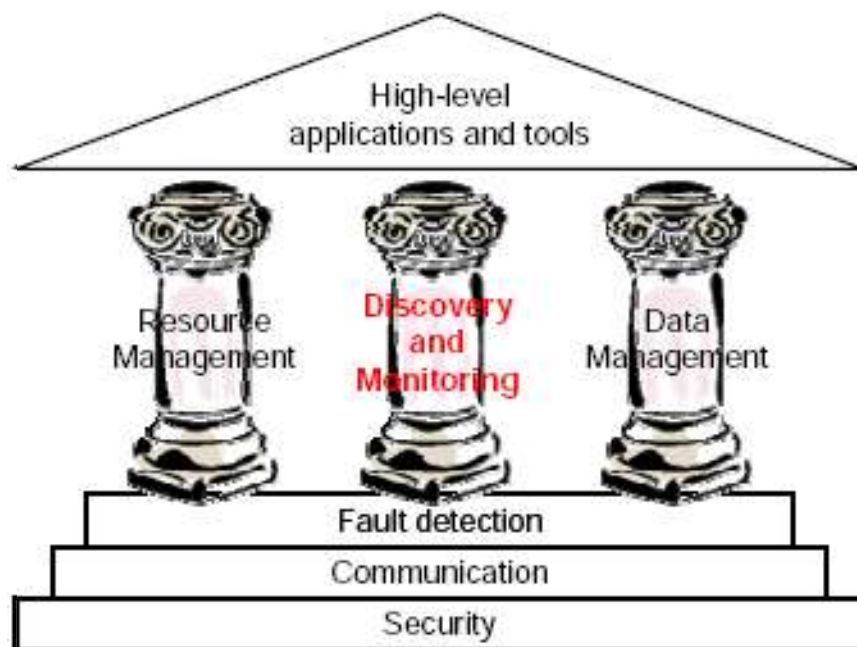


- GRAM:** Globus Resource Allocation Manager
- Gestores Locales (PBS, LoadLeveler,...)
  - RSL (Resource Specification Language)
  - Globus I/O



# Middleware Grid

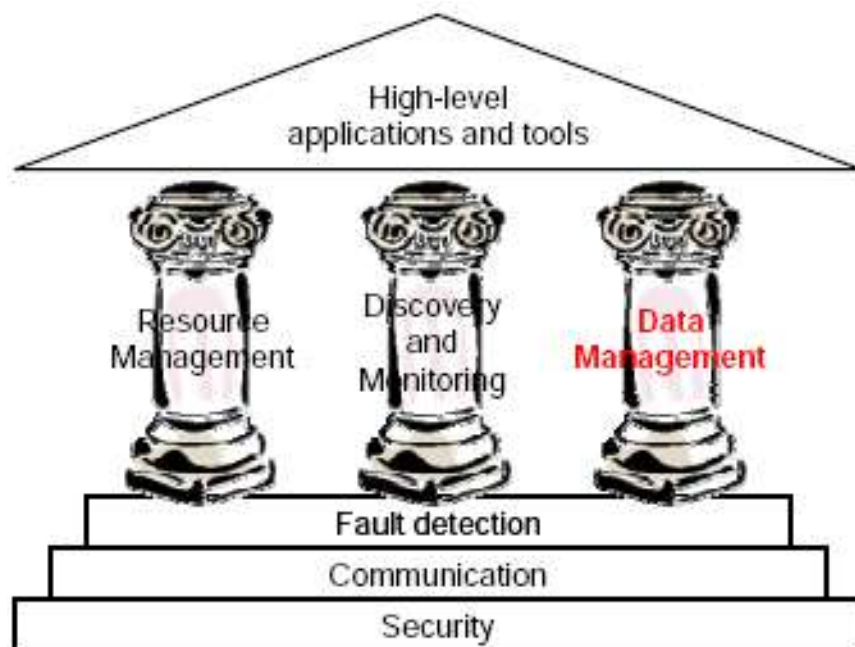
- **Globus Toolkit: Componentes**
  - Descubrimiento y monitorización de Servicios



**MDS:** Monitoring & Discovery System  
**GIIS:** Globus Index Information Service

# Middleware Grid

- **Globus Toolkit: Componentes**
  - Gestión de Datos



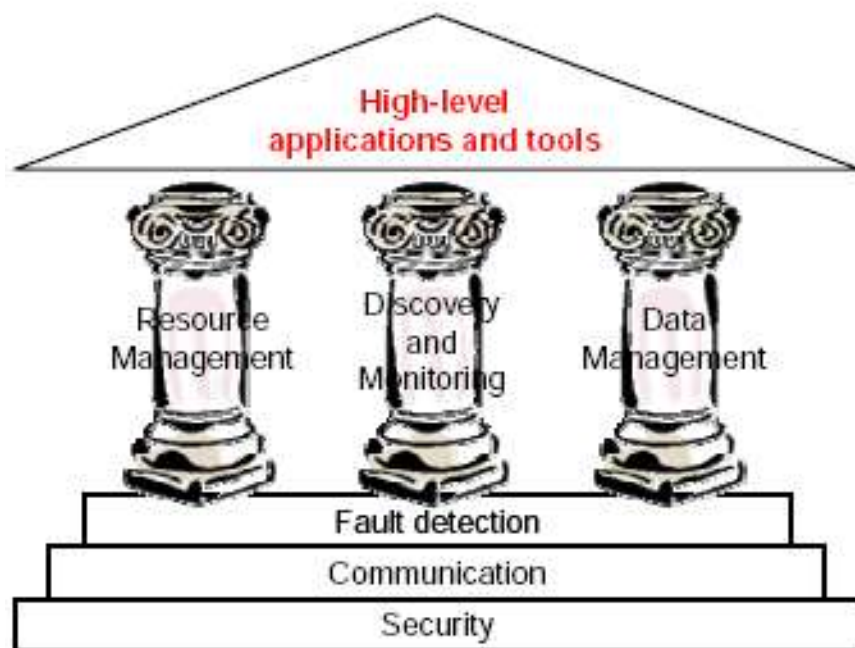
**RLS:** Replica Location Service

**RFT:** Reliable File Transfer

• **GridFTP:** Grid File Transfer Protocol

# Middleware Grid

- **Globus Toolkit: Componentes**
  - Herramientas de alto nivel



## Herramientas de alto nivel:

- Órdenes Globus
- MPICH-G2
- Condor-G
- GridWay

# Índice

---

- Supercomputación clásica
- Grid Computing
- Arquitectura de un Grid
- Middleware para Grid
- Ejemplos de infraestructuras Grid reales
  - Introducción
  - IrisGrid
  - El Proyecto EGEE

# Infraestructuras Grid reales

- Pero todo esto... ¿lo utiliza alguien?

**CrossGrid** 

<http://www.crossgrid.org>

**EU-DataGrid**



<http://www.eu-datagrid.org>

**FlowGrid**



<http://www.unizar.es/flowgrid>

**Damien**



<http://www.hlr.de/organization/pds/projects/damien/>

**iAstro: Cost Action**

 iAstro

<http://main.cs.qub.ac.uk/~fmurtagh/iastro/>

**CESGA-CESCA Grid**

[http://www.cesga.es/Novas/defaultL.html?2003/2003\\_07\\_28.html&2/](http://www.cesga.es/Novas/defaultL.html?2003/2003_07_28.html&2/)

# Infraestructuras Grid reales

- IrisGrid ([www.rediris.es/irisgrid](http://www.rediris.es/irisgrid))



- Dos objetivos:

- Coordinar a nivel académico y científico a los grupos de investigación interesados en esta tecnología, tanto en su desarrollo, como implantación y aplicaciones.
- **Crear la infraestructura Grid nacional** que permita el uso de esta tecnología tanto a nivel de aplicabilidad en diferentes ámbitos, como a nivel de desarrollo e innovación en este campo.

- Pretende:

- Aportar los estándares, protocolos, procedimientos y guías de “buenas prácticas” necesarios para construir un Grid nacional de investigación coordinando a los diferentes grupos interesados.

# Infraestructuras Grid reales

- EGEE ([www.eu-egee.org](http://www.eu-egee.org))

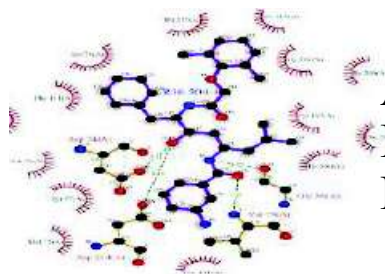


- El problema:

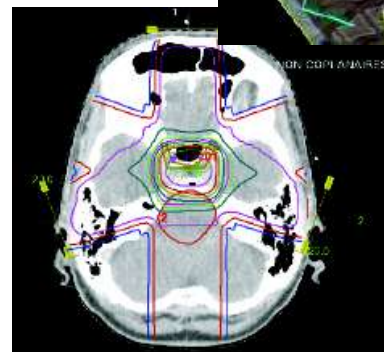
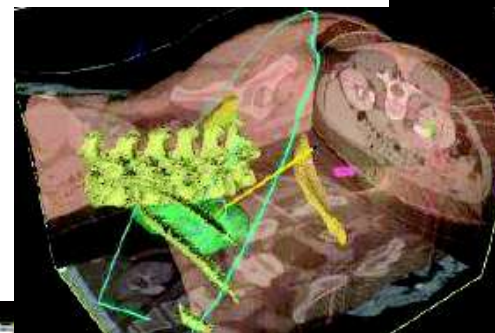
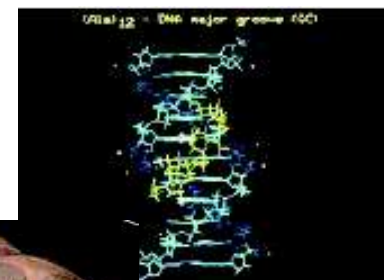
- La ciencia se está “digitalizando” cada vez más y necesita tratar con cantidades cada vez mayores de datos.
- Las simulaciones son cada vez más detalladas.
- La ciencia experimental utiliza sensores muy sofisticados para realizar medidas muy precisas, lo que produce grandes cantidades de datos.

# Infraestructuras Grid reales

- EGEE ([www.eu-egee.org](http://www.eu-egee.org))
  - Ejemplos de aplicación:



Análisis de la Estructura Molecular



Aplicaciones Biomédicas



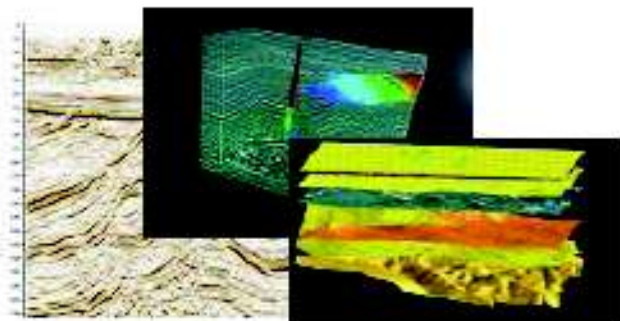
Física de Altas Energías



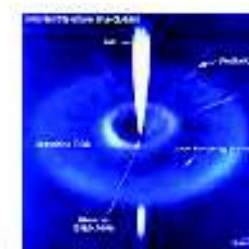
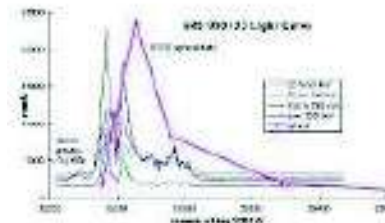


# Infraestructuras Grid reales

- EGEE ([www.eu-egee.org](http://www.eu-egee.org))
  - Ejemplos de aplicación:



Geología

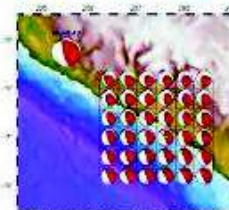


Astronomía

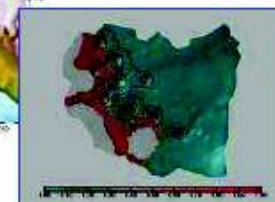
Observaciones de la Tierra por satélite



Estudio de terremotos



Hidrología



# Infraestructuras Grid reales

- EGEE ([www.eu-egee.org](http://www.eu-egee.org))



- Objetivos:

- Crear y mantener una infraestructura Grid segura y robusta.
- Mejorar y mantener su middleware
- Atraer recursos y usuarios de la ciencia y la industria

- Estructura:

- 140 instituciones en 33 países, federados en Grids regionales.
- 14000 CPUs
- 5 PB de almacenamiento disponible

# Infraestructuras Grid reales

- EGEE ([www.eu-egee.org](http://www.eu-egee.org))
  - Middleware de EGEE: gLite
    - Interoperabilidad y coexistencia con la infraestructura existente
    - Robusto y tolerante a fallos
    - Aproximación orientada a servicios
    - Licencia Open Source



# Infraestructuras Grid reales

- EGEE ([www.eu-egee.org](http://www.eu-egee.org))



- Inconvenientes:

- Centrado en física de altas energías
- Poca heterogeneidad y autonomía
- Establece el middleware de usuario, middleware básico y recursos

# ¿Preguntas?

---

