

Tecnologías Grid

Gestión de datos

Master en Sistemas y Servicios Informáticos para Internet
Área de Arquitectura y Tecnología de Computadores
Universidad de Oviedo



Gestion de datos

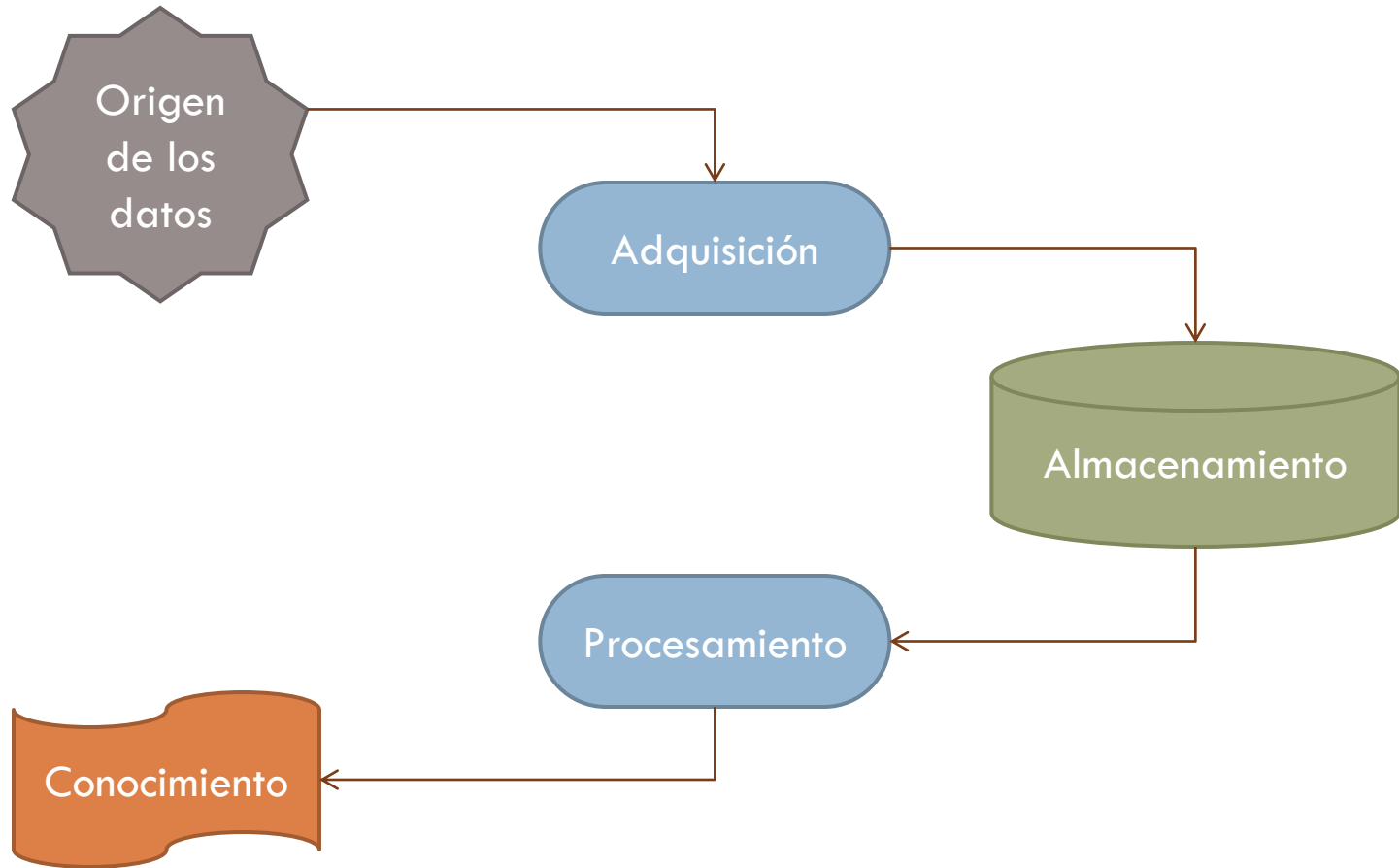
Conceptos previos

Conceptos previos

- Origen de los datos:
 - ▣ Instrumentación
 - Sensores
 - ▣ Simulación
 - Física, química, etc
 - ▣ Imágenes
 - Video, imágenes médicas, renderización
 - ▣ Otros
 - Documentos, ...

Conceptos previos

- Escenario de procesamiento de datos típico:

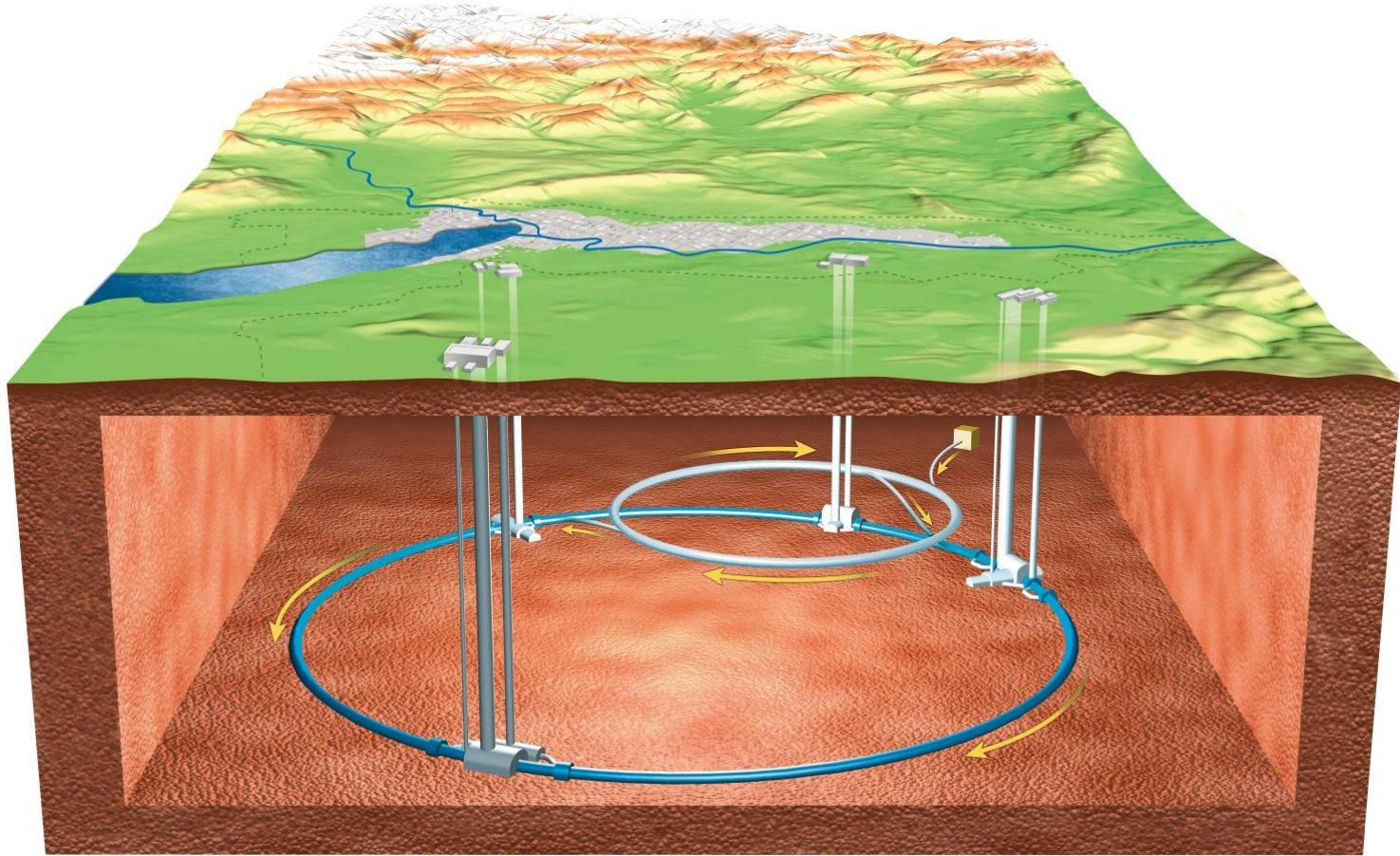


Conceptos previos

- Un ejemplo: Large Hadron Collider (LHC)
 - ▣ Objetivo
 - Explorar los orígenes del universo
 - ▣ Funcionamiento
 - Acelera partículas (haces de hadrones, partículas subatómicas)
 - Las hace colisionar cuando alcanzan el 99% de la velocidad de la luz
 - La colisión produce altísimas energías que permiten simular algunos eventos ocurridos durante o inmediatamente después del big bang

Conceptos previos

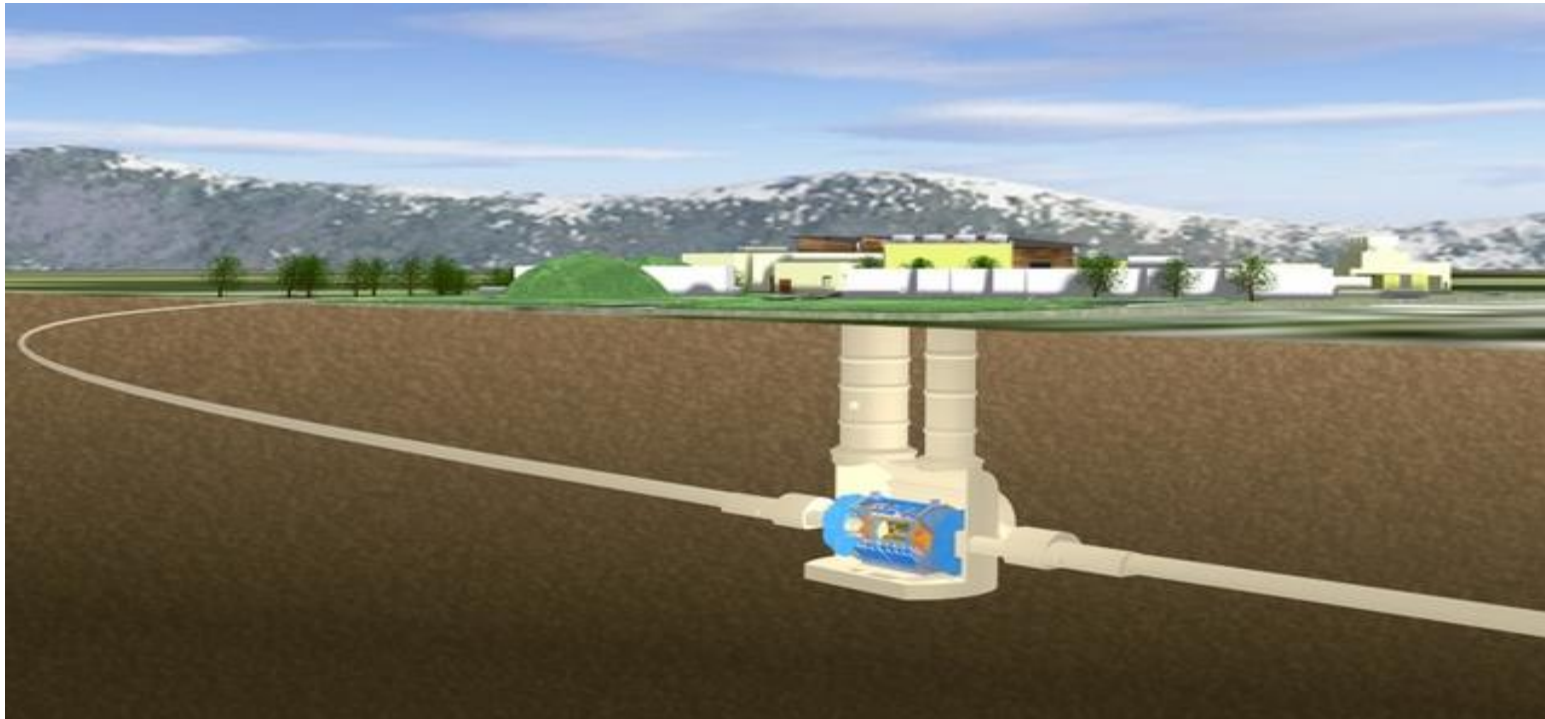
□ Large Hadron Collider (LHC)



<http://lhc.web.cern.ch/>

Conceptos previos

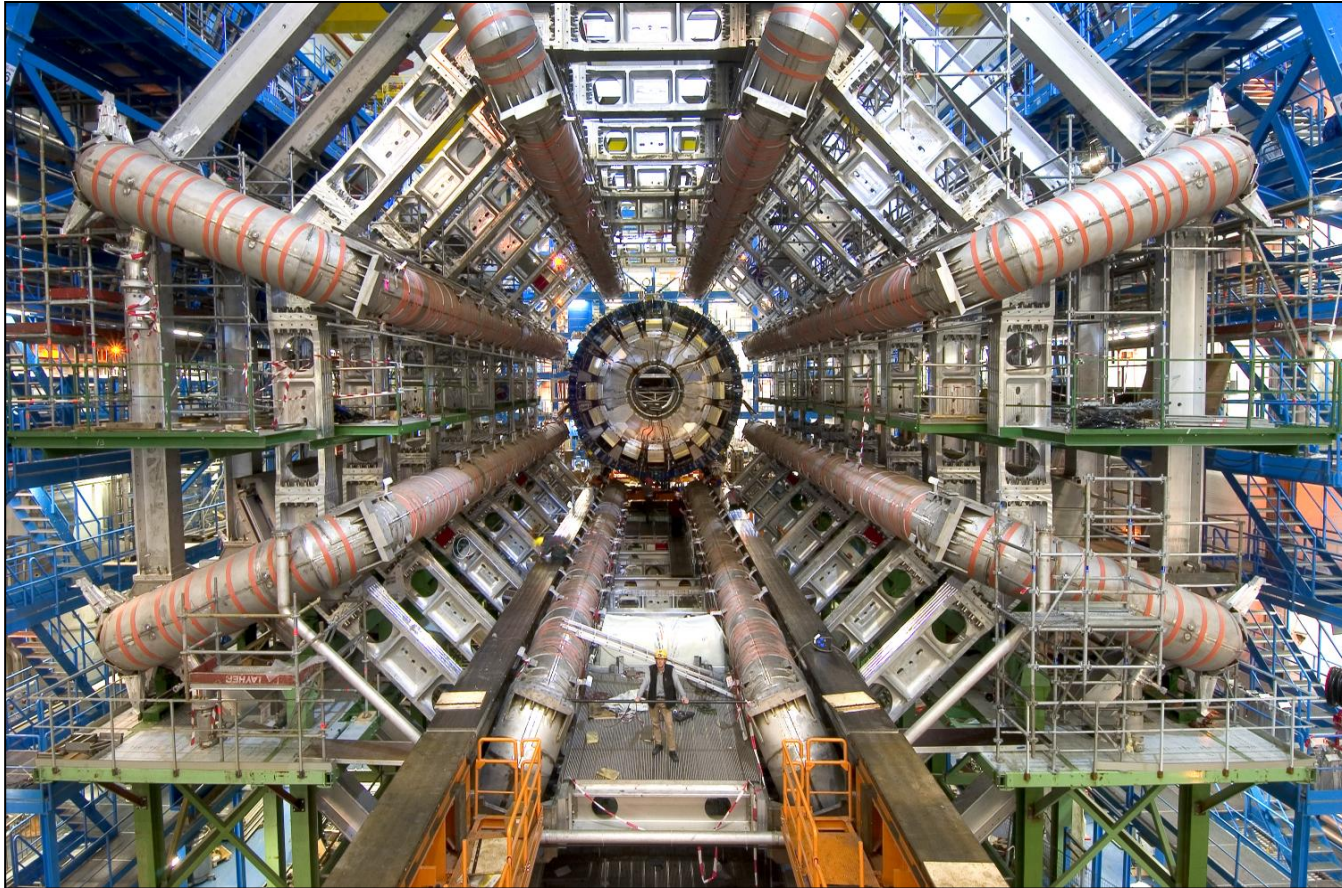
□ Detectores en el LHC



<http://lhc.web.cern.ch/>

Conceptos previos

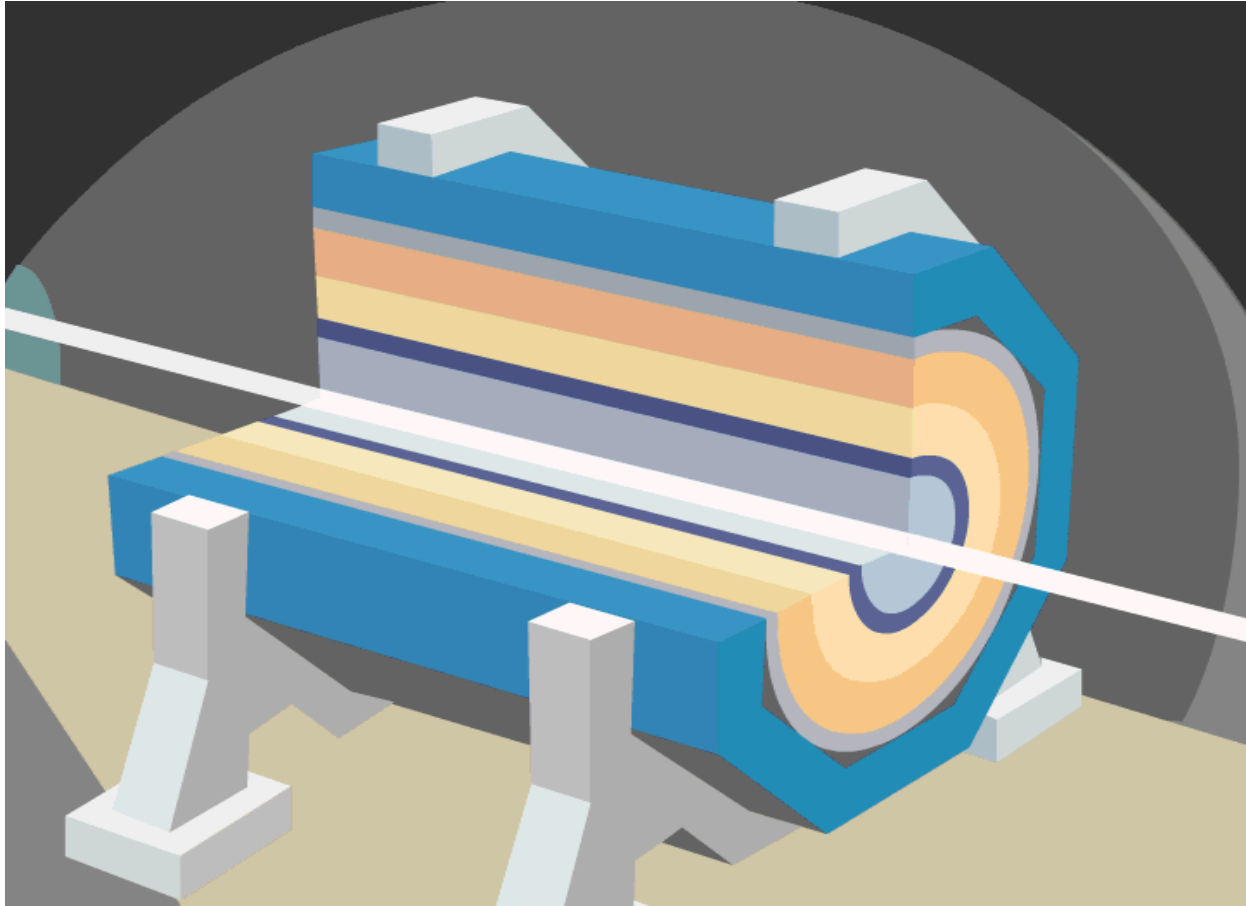
□ Detector del LHC Atlas



<http://lhc.web.cern.ch/>

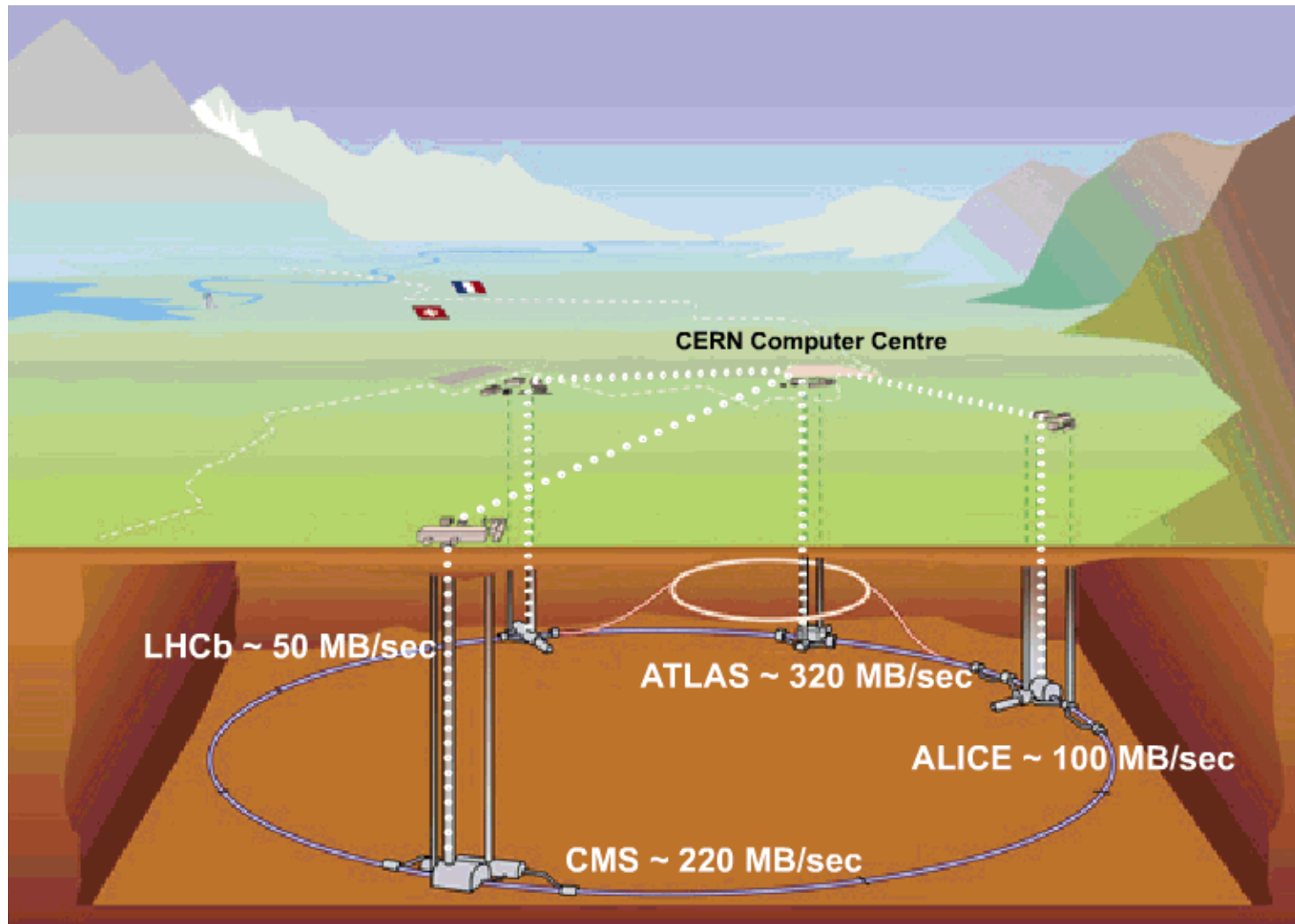
Conceptos previos

□ Colisiones



<http://lhc.web.cern.ch/>

Conceptos previos



<http://lhc.web.cern.ch/>

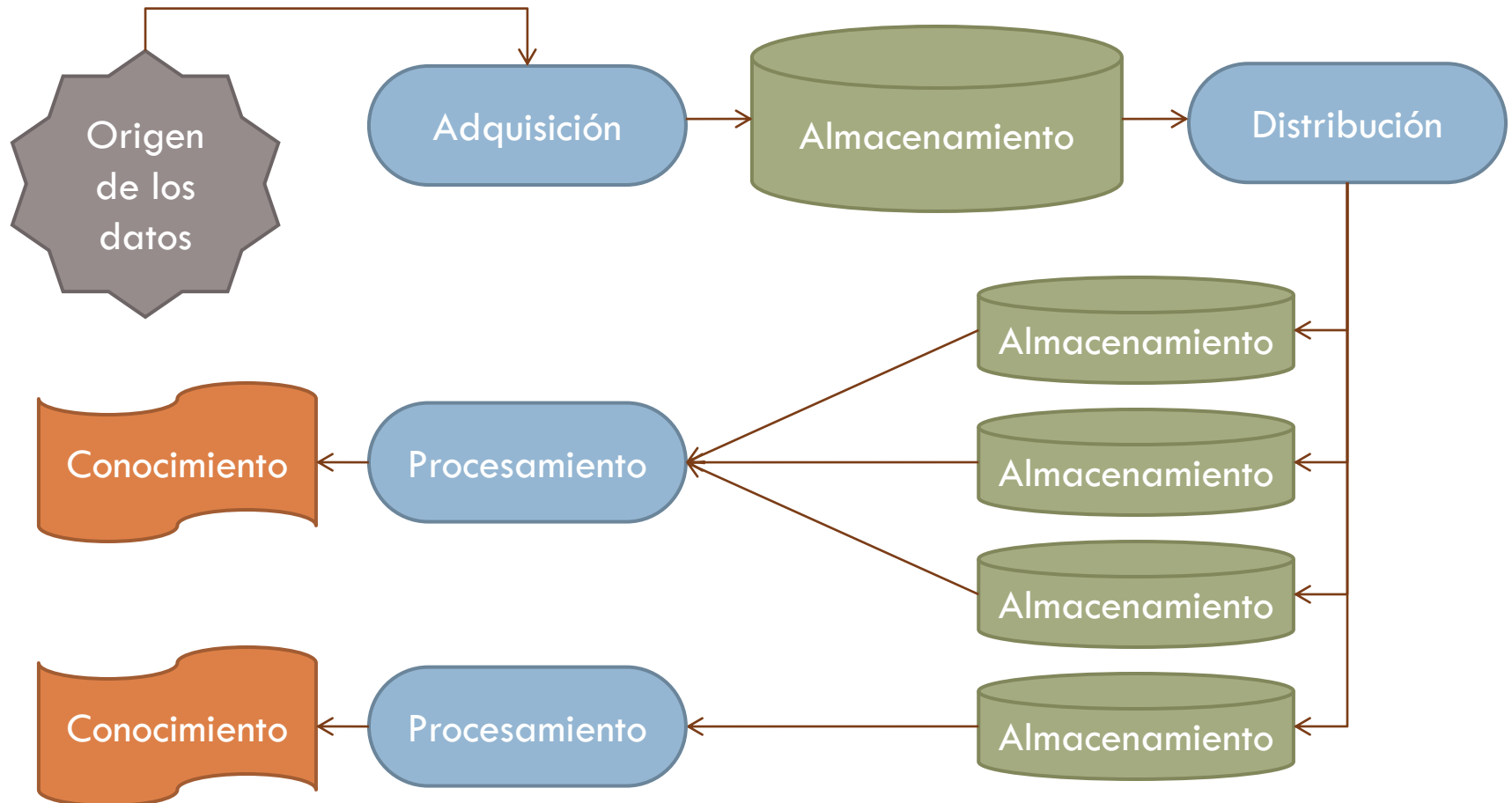
Conceptos previos

- Datos en el LHC:
 - ▣ Varios detectores convierten la física en datos
 - ▣ Se generan aproximadamente 15 petabytes al año
 - 500 megabytes/segundo de media
 - 1000-1600 megabytes/segundo cuando el LHC está funcionando
 - ▣ Miles de científicos alrededor del mundo quieren acceder y analizar los datos

No es posible un escenario de procesamiento de los datos típico

Conceptos previos

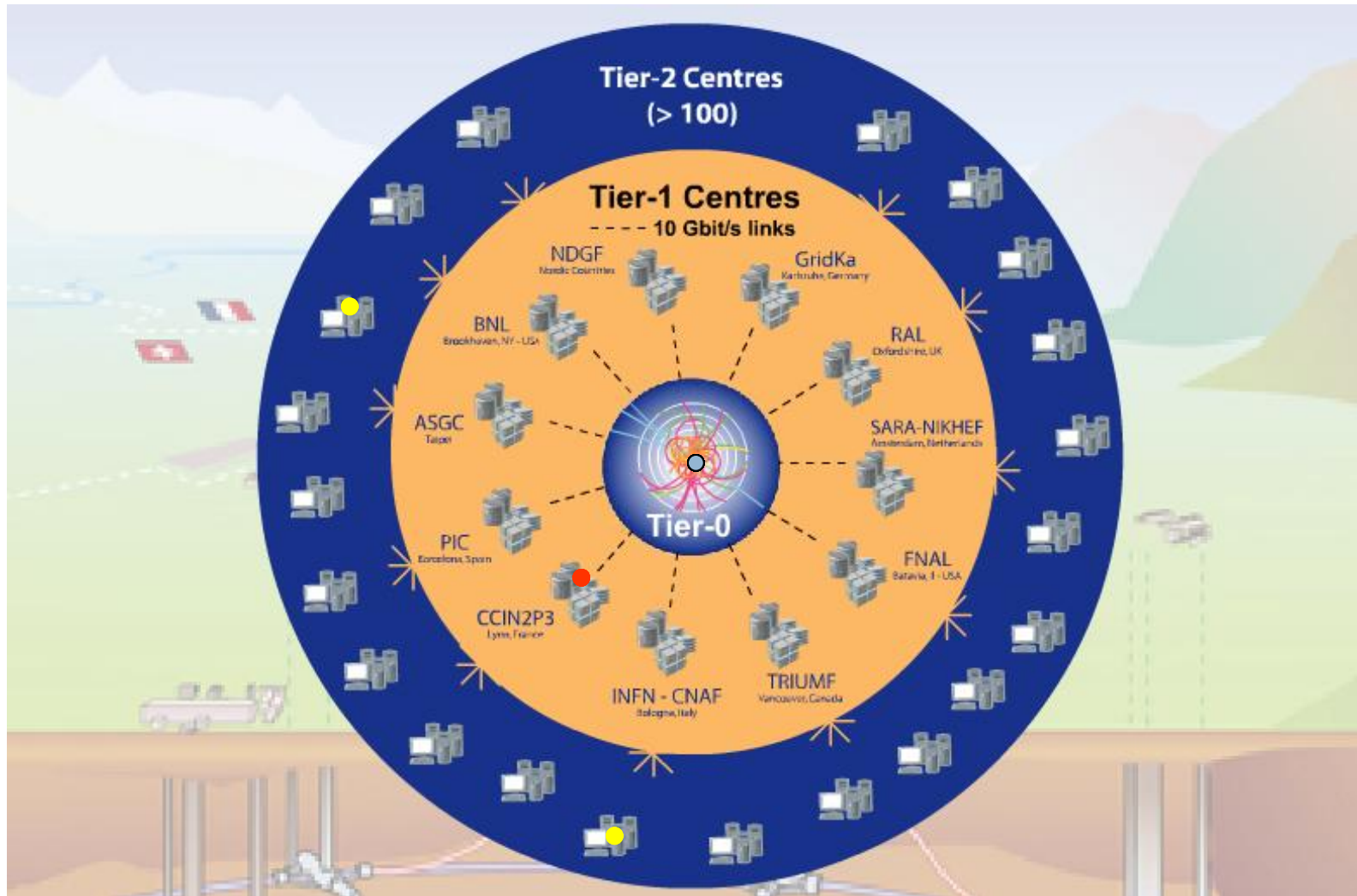
□ Escenario de procesamiento de datos en un grid:



Conceptos previos

- Grid de datos en el LHC:
 - ▣ El almacenamiento primario está en el Tier-0 en el CERN
 - ▣ Del Tier-0 se transfiere a los Tier-1
 - Alemania, Francia, Italia, España, USA, Canada, Países nórdicos, Taipei
 - ▣ Los Tier-1 hacen que los datos estén disponibles a los Tier-2 (centros de investigación)
 - Los científicos pueden acceder a los datos del LHC en un ordenador o cluster cercano

Conceptos previos



<http://lhc.web.cern.ch/>

Conceptos previos

□ Otros ejemplos:

▣ Distributed Aircraft Maintenance Environment (DAME)

- Procesamiento de datos obtenidos de los aviones durante el vuelo

▣ Earth Systems Grid (ESG)

- Procesamiento de datos sobre el clima

▣ Network for Earthquake Engineering Simulation (NEES)

- Procesamiento de datos sobre terremotos

▣ Grid for Ocean Diagnostics, Interactive Visualisation and Analysis (GODIVA)

- Procesamiento de datos sobre el oceano

Conceptos previos

□ Necesidades:

- ▣ Seguridad y control de acceso
- ▣ Descubrimiento de los datos
- ▣ Transferencia de los datos
- ▣ Gestión de replicas
- ▣ Gestión de caches
- ▣ Reserva de recursos

□ Soluciones

- ▣ No hay una solución para todo
- ▣ Diversas tecnologías solucionan problemas concretos



Gestion de datos

Sistemas de ficheros distribuidos

Sistemas de ficheros distribuidos

□ Definición:

- ▣ Sistema que permite compartir información entre múltiples máquinas proporcionando un interfaz de acceso homogéneo
 - La información puede estar
 - Centralizada en una máquina (Network attached storage)
 - Distribuida, incluso de forma redundante, entre muchas máquinas (Cluster filesystems)

□ Características deseadas:

- ▣ Transparencia, Productividad
- ▣ Escalabilidad, Tolerancia a fallos

Distributed File
System (DFS)

Sistemas de ficheros distribuidos

□ Network File System (NFS):

▣ Origen:

- Desarrollado por SUN

- Actualmente por Internet Engineering Task Force (IETF)

▣ Características.

- Permite compartir ficheros a través de la red

- El servidor exporta directorios

- Los clientes montan los directorios

- Los usuarios utilizan los directorios montados como si fueran locales

▣ NFSv4 (influenciado por AFS)

- 4.1 Añade WAN y paralelismo (pNFS)

- Aún no hay implementaciones completas

NFS es similar
a SMB/CIFS
(Samba)

Sistemas de ficheros distribuidos

□ Andrew File System (AFS):

□ Origen:

- Desarrollado por la Carnegie Mellon University en 1983
 - Posteriormente comprado por IBM y liberado como código abierto en 2000 como OpenAFS

□ Características:

- Respecto a NFS, mejora la seguridad y escalabilidad (hasta 5000 máquinas)
- Hace gran uso de cachés
 - Evita tener que acceder al servidor

□ Junto con NFS se considera el ejemplo clásico de DFS

Sistemas de ficheros distribuidos

□ Parallel Virtual File System (PVFS):

□ Origen:

- Proyecto GPL

□ Características:

■ Paralelismo (~RAID)

- Distribuye los ficheros en múltiples servidores
- Gran velocidad de acceso para acceder a grandes cantidades de datos de forma concurrente
- No está diseñado para proporcionar almacenamiento persistente sino para dar soporte a aplicaciones HPC (MPI)

Disponible en
Clusters Rocks

Sistemas de ficheros distribuidos

□ Lustre:

□ Origen:

- Desarrollado por Cluster File Systems Inc.
 - Posteriormente comprado por SUN y liberado con licencia GPL

□ Características:

- Almacenamiento paralelo orientado a objetos
- Escala a decenas de miles de nodos con petabytes de almacenamiento y con una transferencia de gigabytes por segundo
- 15 de los 30 mayores super computadores usan Lustre
- Muy utilizado en Data-Centers

Sistemas de ficheros distribuidos

- General Parallel File System (GPFS):
 - Origen:
 - IBM (de pago)
 - Características:
 - Es el más maduro de los sistemas paralelos (no orientado a objetos)
 - Muy usado en HPC
 - Tolerante a fallos (distribuye los metadatos)
 - Disponible en Unix y Windows (2008)

Sistemas de ficheros distribuidos

□ Hadoop File System (HDFS):

▣ Origen:

- Proyecto de Apache Inspirado en GFS (Google File System)

▣ Características:

- Desarrollado para dar soporte a tareas Map/Reduce
- Pensado para para grandes ficheros cuyo acceso sigue un patrón de una escritura y muchas lecturas
- Tolerante a fallos gracias a la replica de información en los nodos
- No permite montar el sistema de ficheros (si con FUSE)
- Aún en desarrollo (Probado en Yahoo con 10000 nodos)

Sistemas de ficheros distribuidos

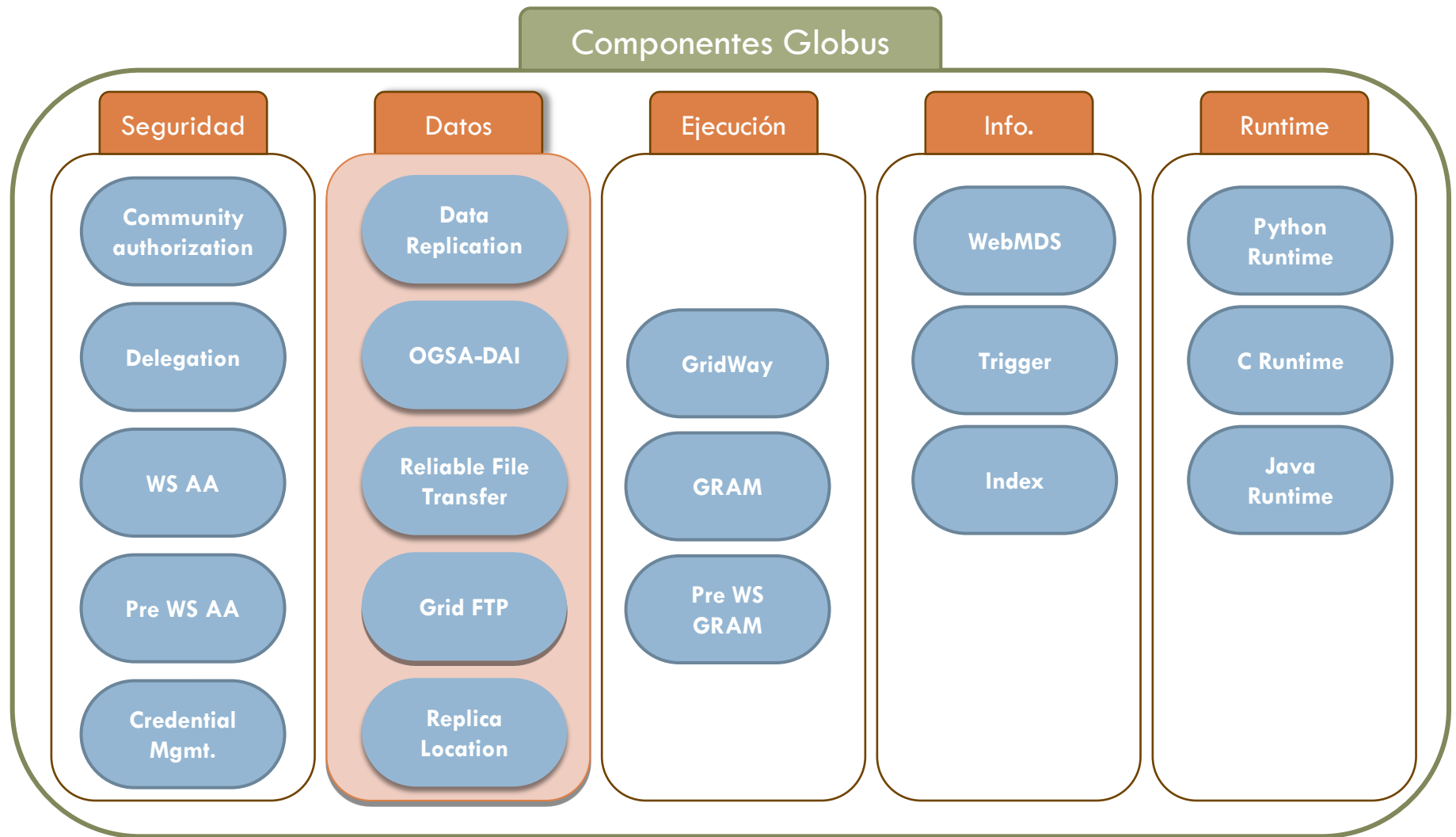
- Problemas de los DFS:
 - ▣ En mayor o menor medida aparecen problemas de escalabilidad
 - ▣ No funcionan de forma adecuada ante hardware heterogeneo
 - ▣ Están diseñados principalmente para ser usados en LAN de alta velocidad, no en WAN
 - ▣ Disponen de una administración centralizada



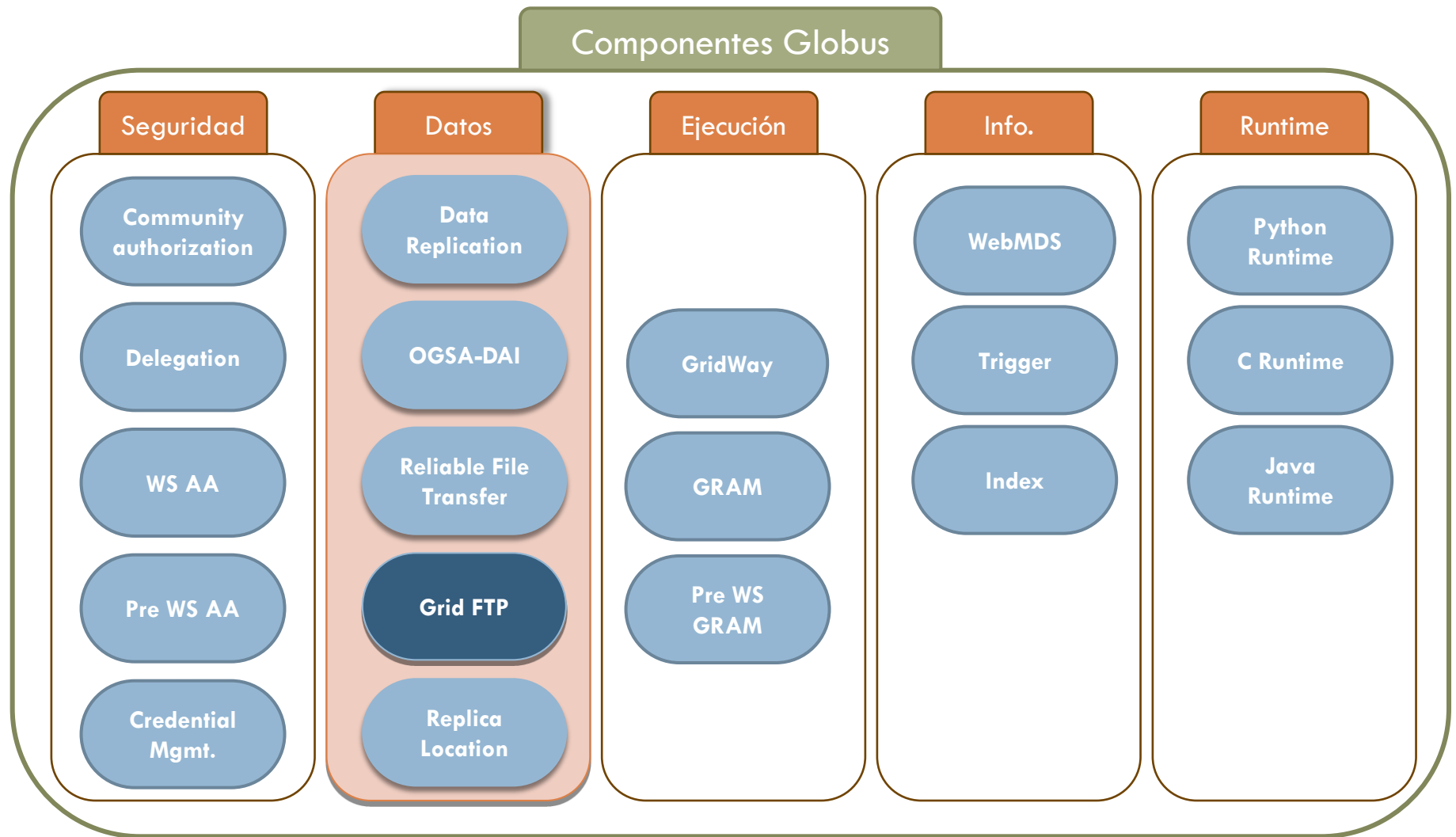
Gestion de datos

Gestión de datos en Globus

Gestión de datos en Globus



Gestión de datos en Globus



Gestión de datos en Globus

□ GridFTP

- ▣ Protocolo de transferencia de información diseñado para transferir gran cantidad de datos a gran velocidad
- ▣ Extiende el protocolo FTP
 - Autenticación y encriptación basada en Globus GSI
 - Permite utilizar múltiples canales de datos
 - Permite ajustar parámetros de la transferencia para mejorar el ancho de banda (Ej: tamaño de bloque TCP)
 - Permite transferir partes de un fichero
 - Permite usar multicasting
- ▣ Al igual que el FTP, no es *firewall friendly*

Gestión de datos en Globus

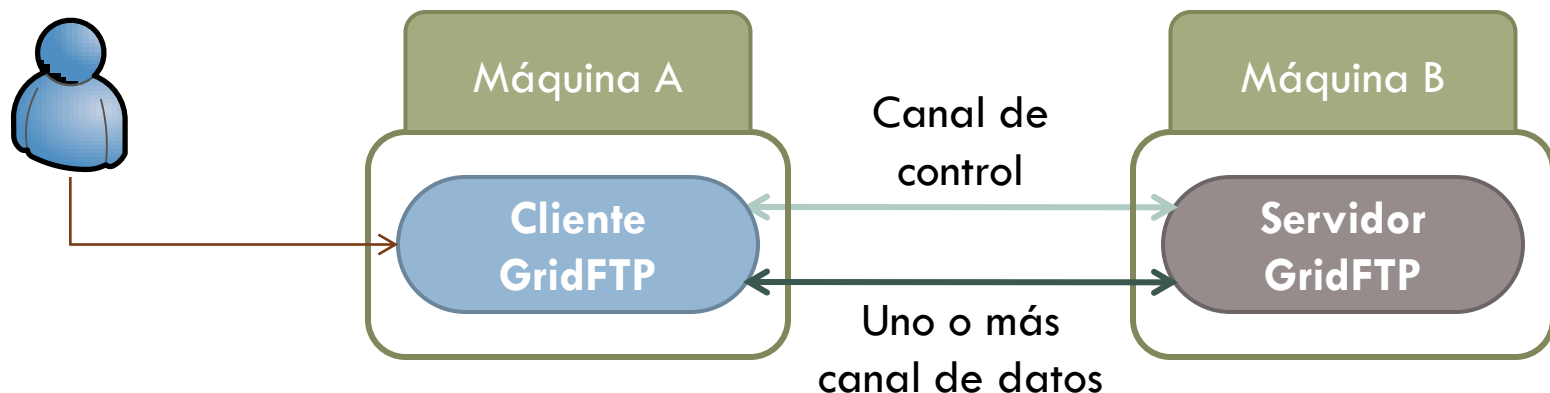
□ GridFTP

▣ Canal de control

- Se intercambian comandos y respuestas

▣ Canal de datos

- Se transfiere la información (ficheros y listados de directorios)

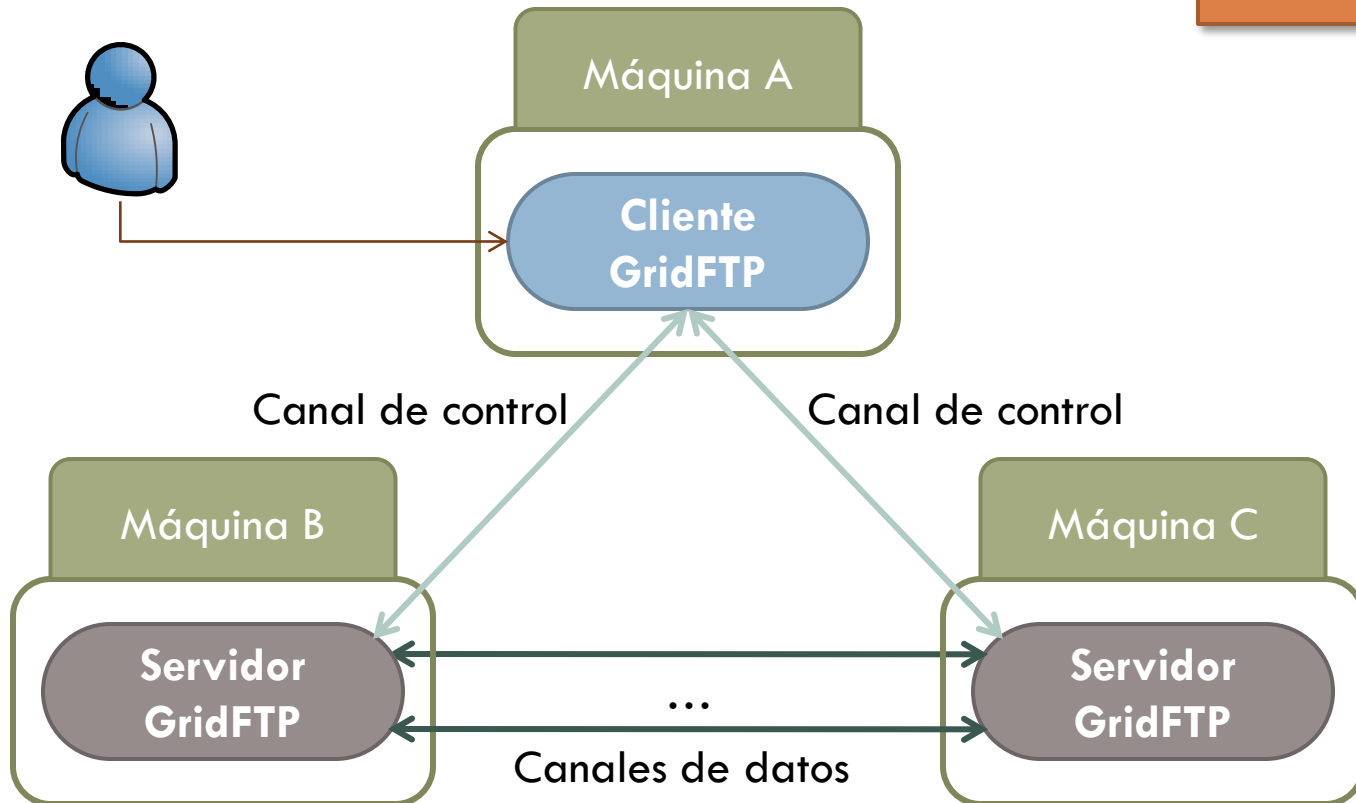


Gestión de datos en Globus

□ GridFTP

▣ Third party transfer

En FTP se denomina FXP



Gestión de datos en Globus

□ GridFTP

▣ Forma de usarlo más habitual:

- `globus-url-copy <fuente> <destino>`

```
$ globus-url-copy gsiftp://remote.host/path/file file:///path/file
```

▣ Otras formas:

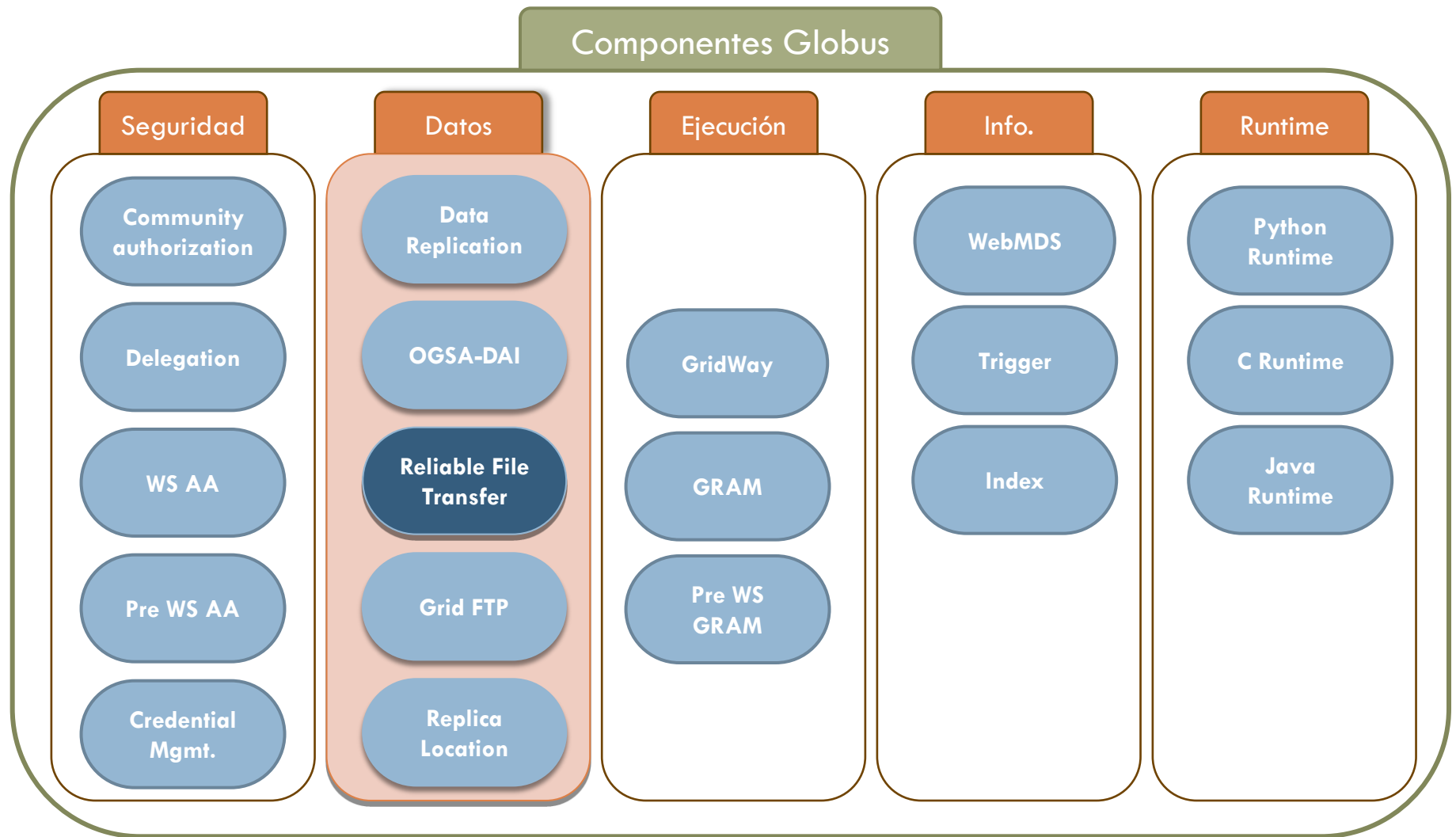
- UberFTP

- Cliente interactivo de consola

- GridFTP GUI

- Cliente en Java con interfaz

Gestión de datos en Globus



Gestión de datos en Globus

□ Reliable File Transfer (RFT)

- ▣ Servicio que proporciona fiabilidad y tolerancia de fallos a la transferencia de información

- Actúa como un cliente de GridFTP

- Convierte a la transferencia en un trabajo

- ▣ Ventajas respecto a usar GridFTP directamente

- Está diseñado como un servicio web WSRF

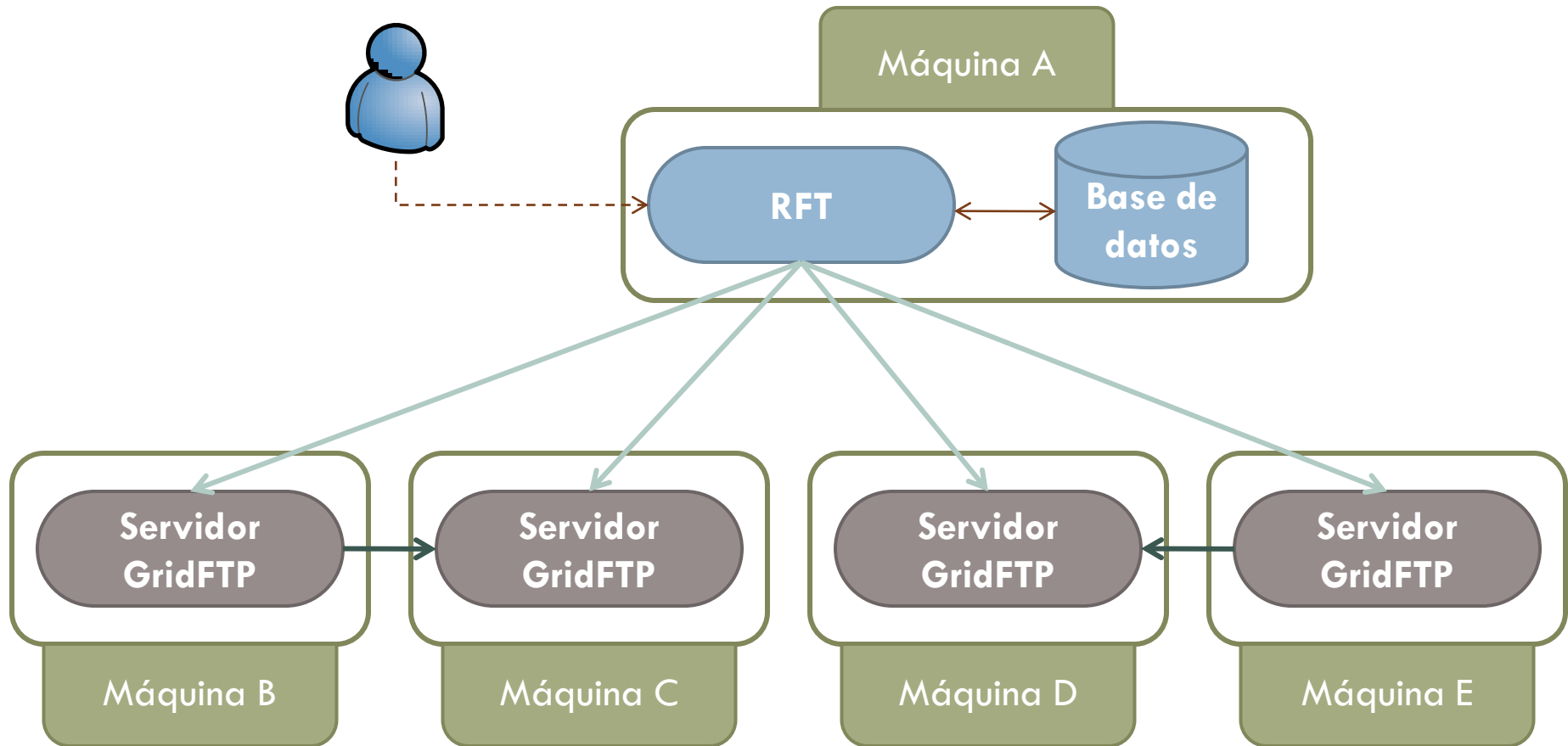
- Otros programas lo pueden utilizar de forma simple

- Se pueden especificar múltiples transferencias

- Evita que el cliente tenga que mantener la conexión de control abierta durante la transferencia

Gestión de datos en Globus

□ Reliable File Transfer (RFT)



Gestión de datos en Globus

□ Reliable File Transfer (RFT)

Diseñado principalmente para ser usado por otros programas

□ Forma de usarlo:

■ `rft -file <fichero_EPR> -f <fichero_descr>`

```
$ rft -file transfer.epr -f transfer.xfr
```

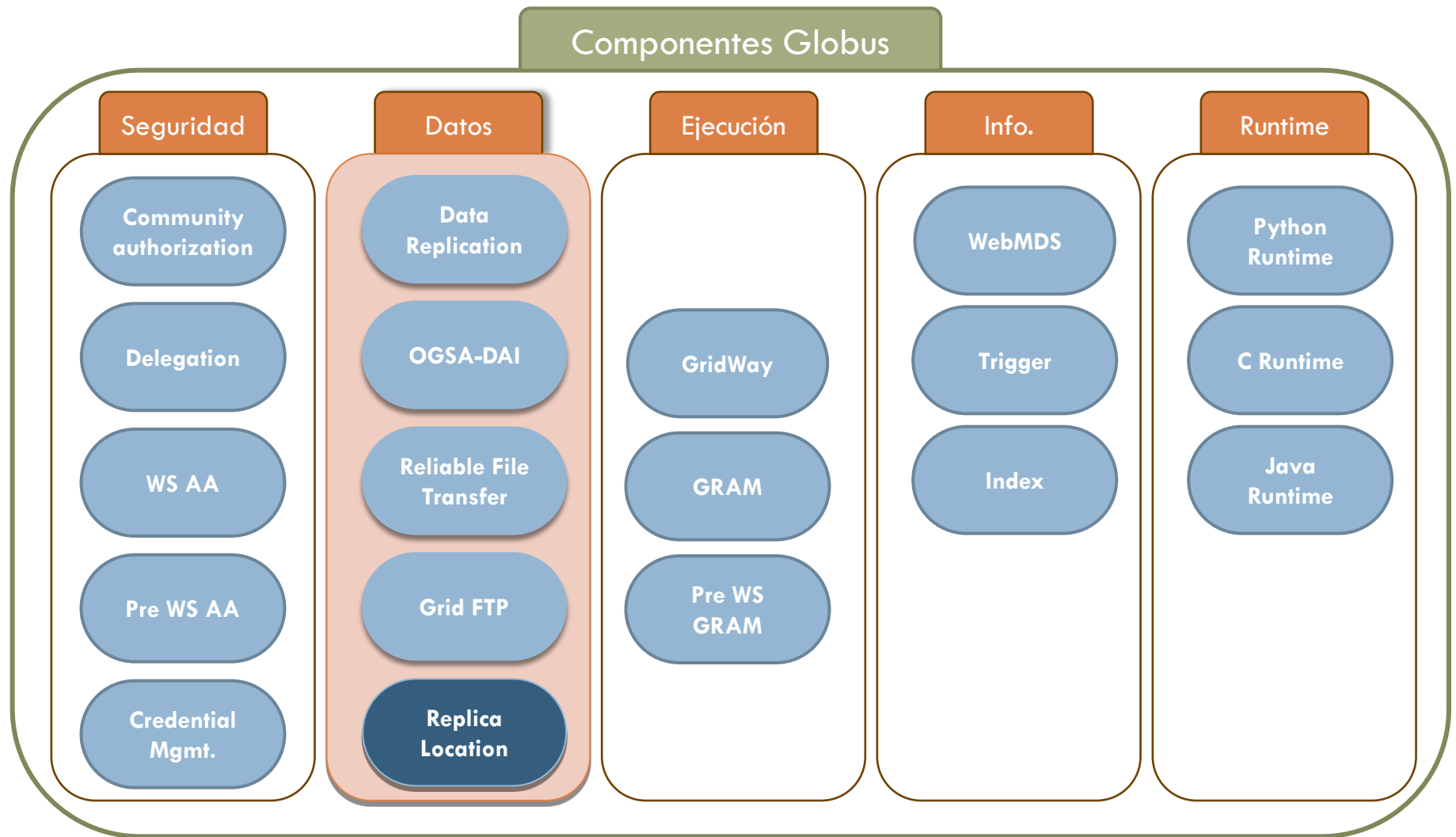
```
#true=binary false=ascii
True
#TCP Buffer size in bytes
16000
#Number of parallel streams
1

...

#Source/Dest URL Pairs
gsiftp://servidor1.atc:2811/data/file
gsiftp://servidor2.atc:2811/backup/file
```

transfer.xfr

Gestión de datos en Globus



Gestión de datos en Globus

- Reliable Locations Service (RLS)
 - ▣ Servicio que gestiona el registro y la búsqueda de información replicada
 - ▣ Elementos del RLS:
 - Local Replica Catalog (LRC)
 - Mantiene un catálogo de información replicada localmente
 - Replica Location Index (RLI)
 - Mantiene un catálogo de información replicada globalmente
 - Es una capa de nivel superior al LRC
 - ▣ Un servidor de RLS puede actuar como LRC, como RLI, o como ambos

Se utilizan replicas por fiabilidad, disponibilidad, escalabilidad y durabilidad

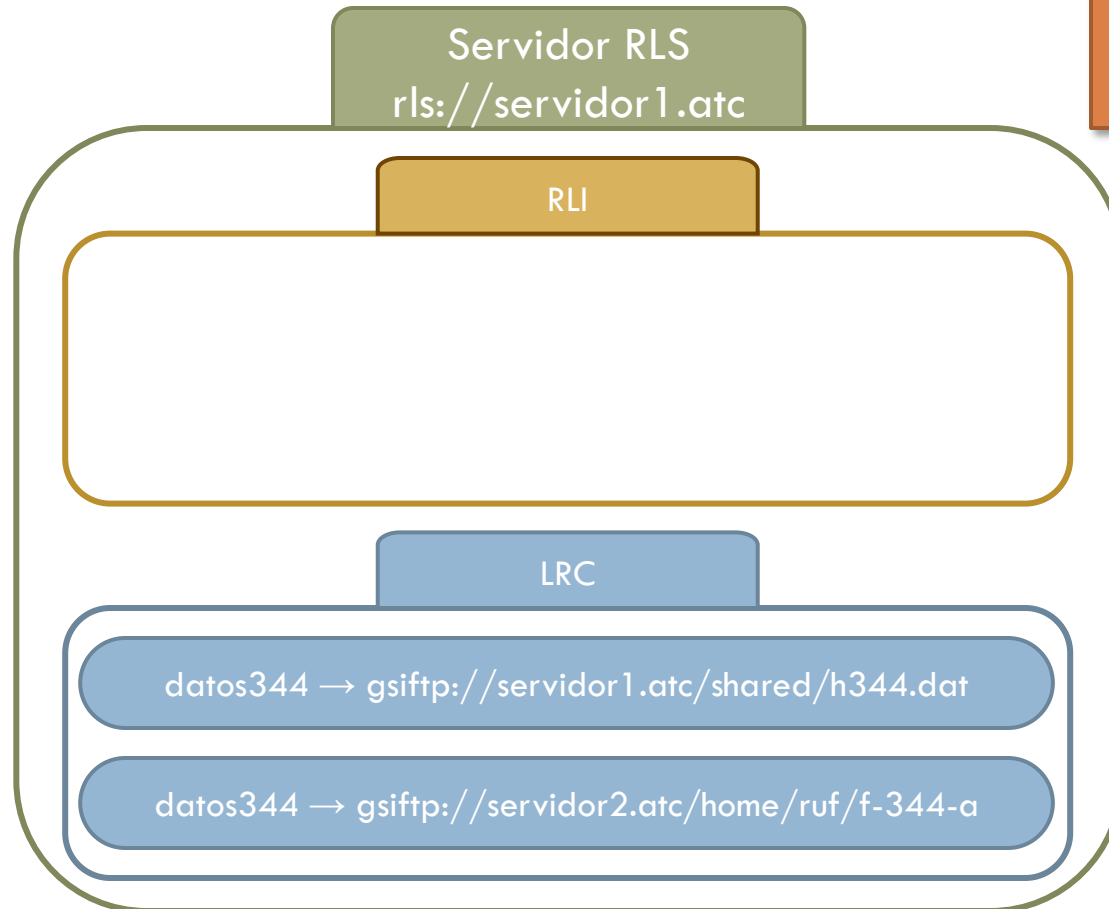
Gestión de datos en Globus

- Local Replica Catalog (LRC)
 - ▣ Mapea nombres de ficheros lógicos (LFNs) a nombre de ficheros físicos (PFNs)
 - ▣ Logical Filenames (LFN)
 - Es el nombre de un fichero
 - No se refiere a un lugar o máquina donde esté almacenado
 - Ej: datos344
 - ▣ Physical Filenames (PFN)
 - Se refiere a la ubicación física de un fichero
 - Ej: gsiftp://servidor1.atc/shared/h344.dat

Gestión de datos en Globus

□ Servidor de RLS actuando de LRC

Es necesario
hacerlo
distribuido

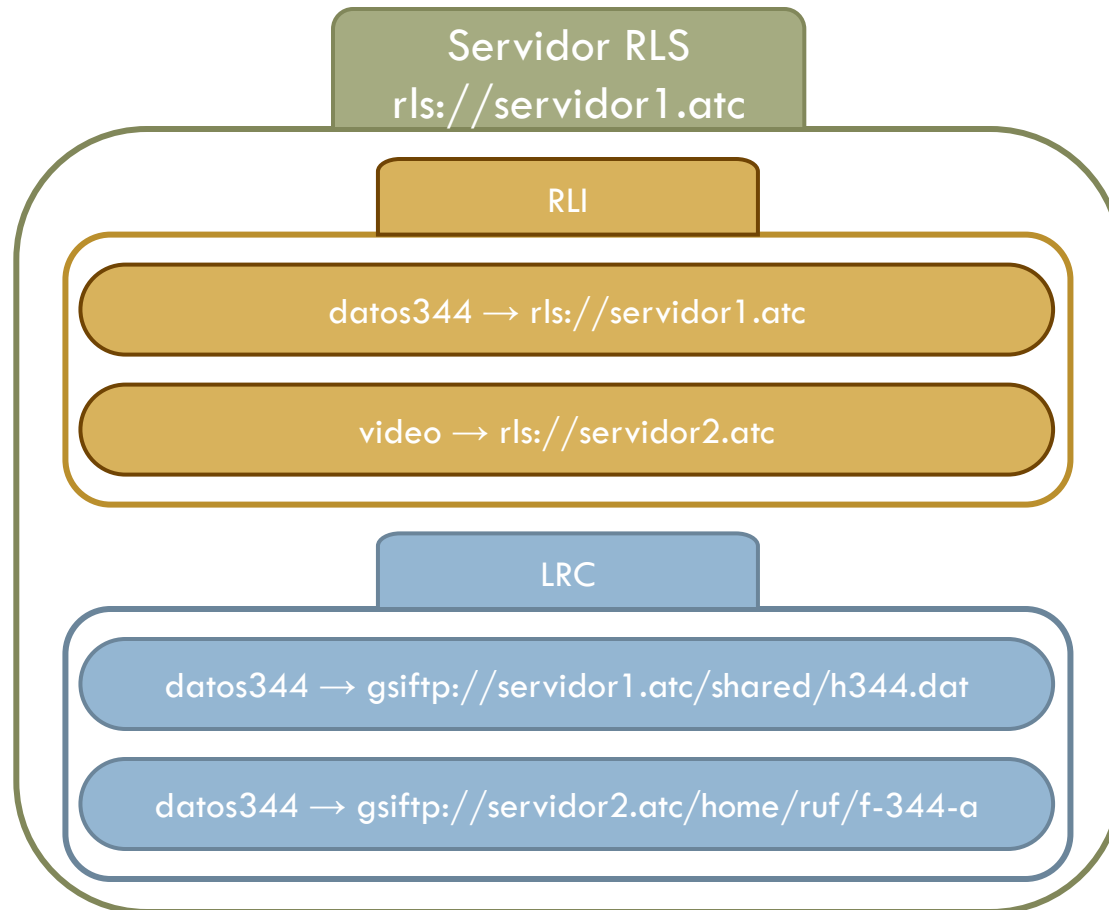


Gestión de datos en Globus

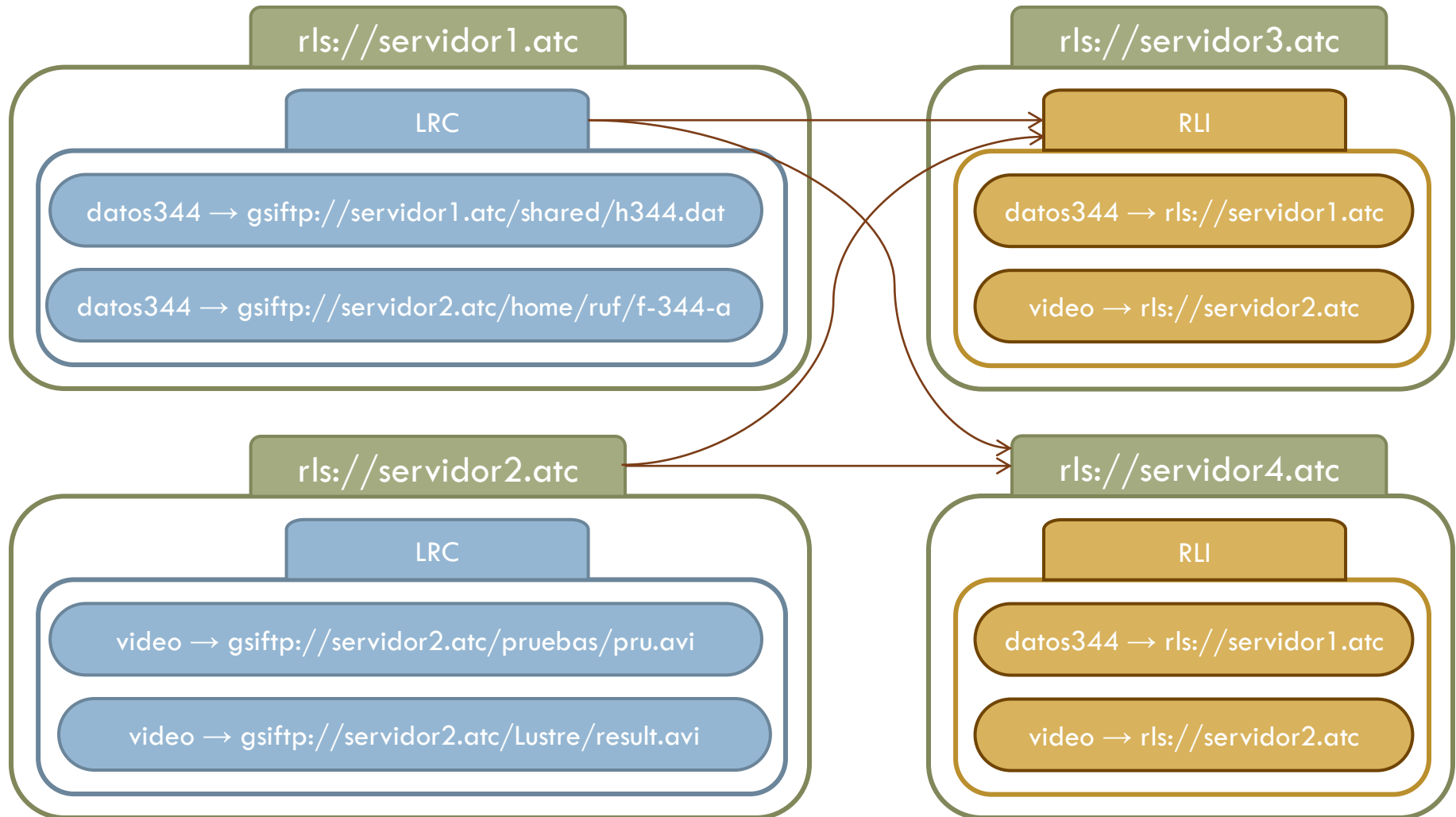
- Replica Location Index (RLI)
 - ▣ Mapea nombres de ficheros lógicos (LFNs) a LRCs que contienen mapeos de esos LFNs a PFNs
 - ▣ Ventajas de usar RLI:
 - Los fallos en el RLI o en el LRC no provocan un fallo global
 - El RLI almacena una información sobre los mapeos reducida y por tanto puede almacenar más
 - Permite centralizar las búsquedas de la información replicada

Gestión de datos en Globus

❑ Servidor de RLS actuando de LRC y RLI



Gestión de datos en Globus

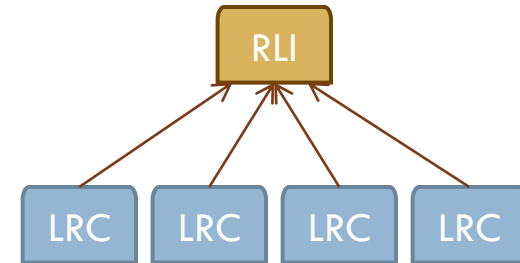


Gestión de datos en Globus

□ Algunas posibles topologías:

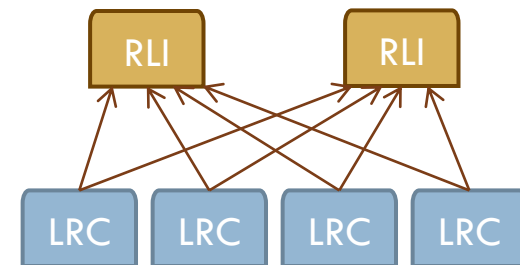
□ Jerarquía simple

- Configuración sencilla



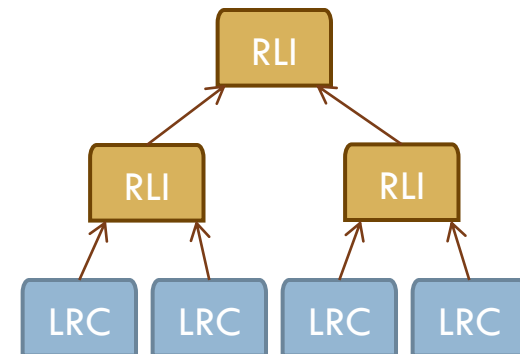
□ Completamente conectado

- Alta disponibilidad



□ Jerarquía en capas

- Para gran cantidad de datos



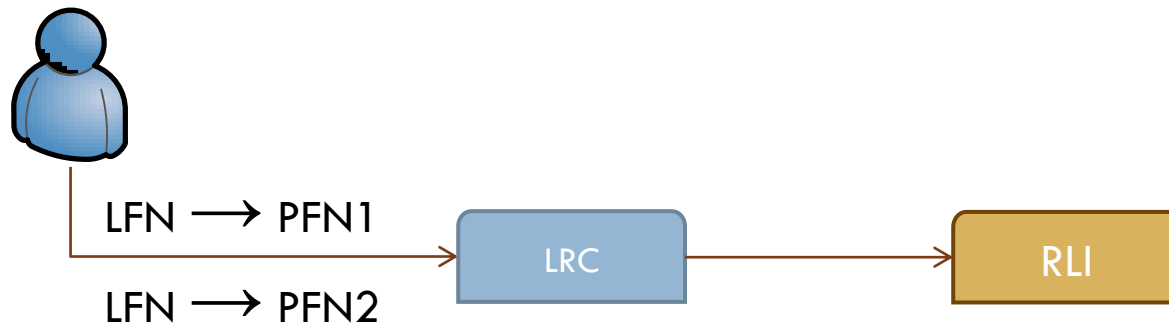
Gestión de datos en Globus

□ Creación de un mapeo

- ▣ Se pide al LRC que cree un mapeo entre un LFN y un PFN1:
 - El LRC informa al RLI del mapeo

□ Adición de un nuevo mapeo

- ▣ Se pide al LRC que cree un nuevo mapeo entre el mismo LFN y PFN2



Gestión de datos en Globus

□ Búsqueda de un LFN

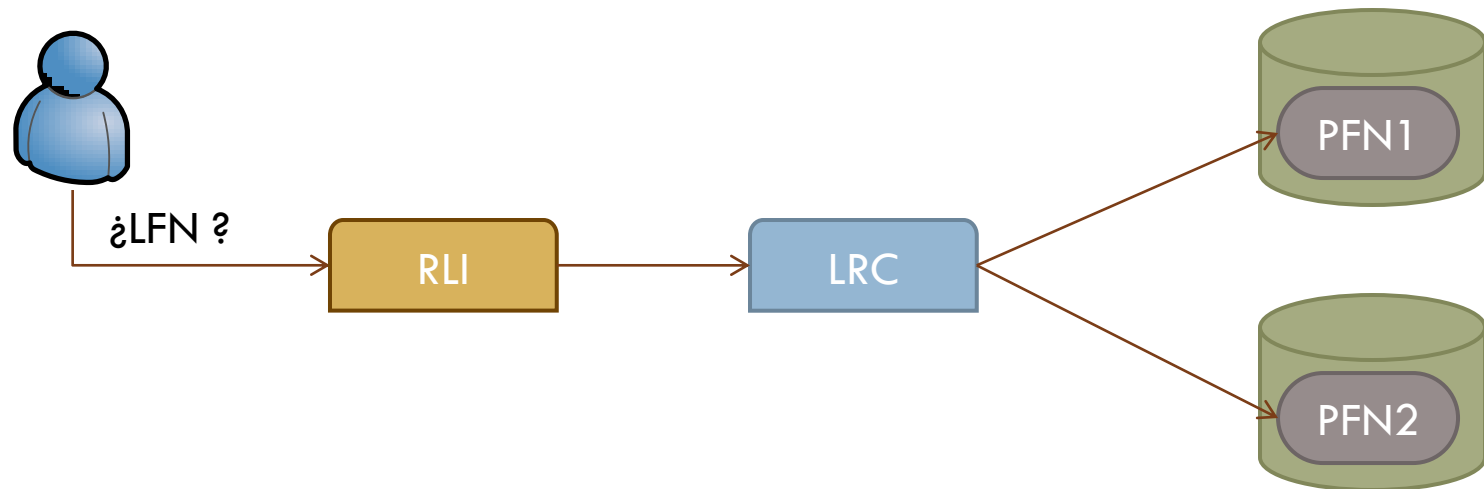
▣ Se pregunta al RLI:

- Responde con una lista de LRCs

▣ Se pregunta a un LRC:

- Responde con una lista de PFNs

Escenario donde los datos no se modifican una vez creados, sólo se leen



Gestión de datos en Globus

□ Utilización:

▣ globus-rls-admin

■ Tareas administrativas

- Ping a un servidor
- Configurar la conexión entre el LRC y el RLI

▣ globus-rls-cli

■ Tareas de usuario

- Pedir información al RLC o al RLI
- Crear y añadir mapeos

Gestión de datos en Globus

□ Ejemplos de utilización:

▣ Crear la conexión entre el LRC y el RLI:

```
$ globus-rls-admin -a rls://lrc-server rls://rli_server
```

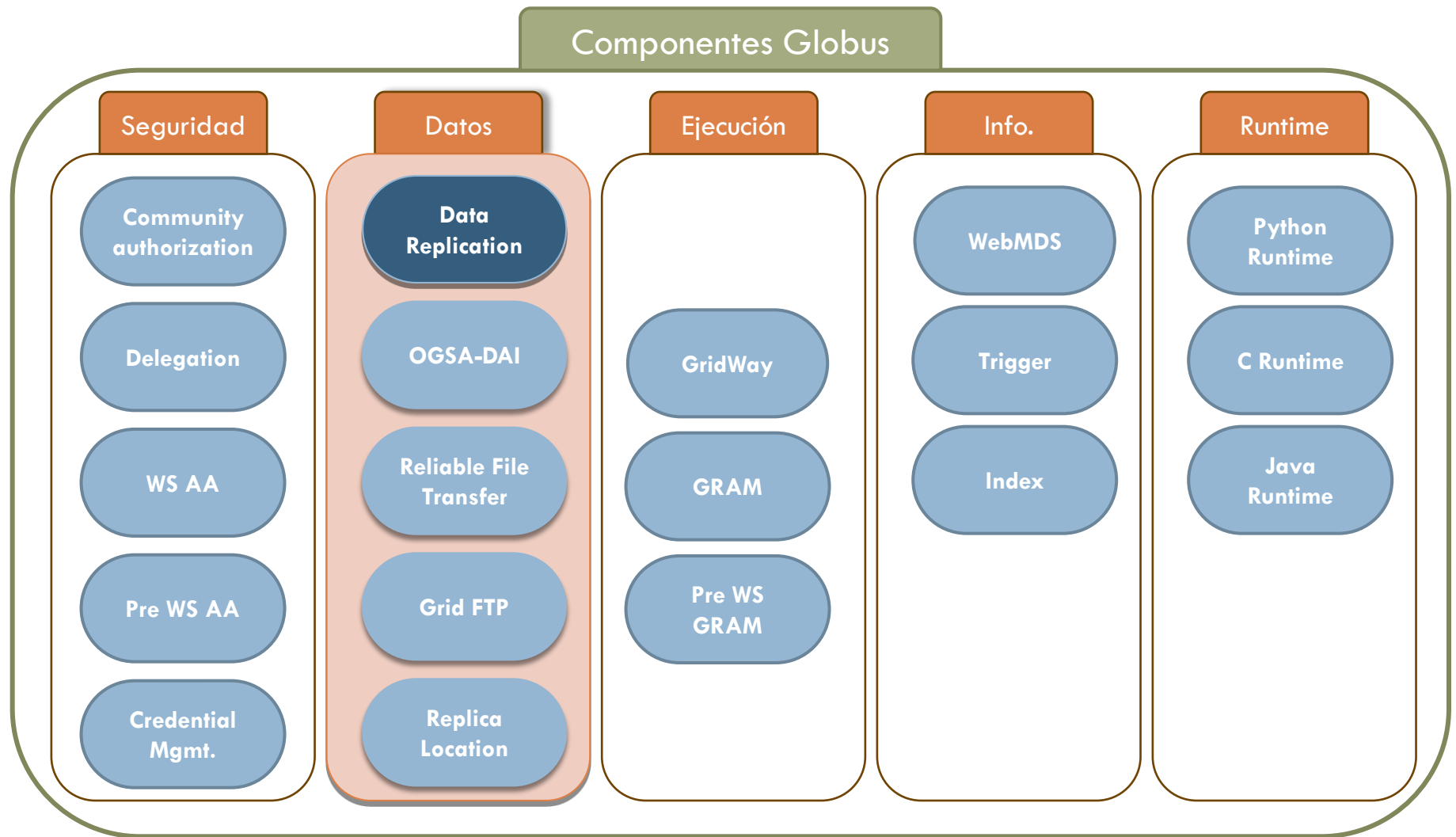
▣ Crear un mapeo:

```
$ globus-rls-cli create datos344 \  
  gsiftp://servidor1.atc/shared/h344.dat rls://lrc_server
```

▣ Buscar información:

```
$ globus-rls-cli query rli lfn datos344 rls://rli_server
```

Gestión de datos en Globus



Gestión de datos en Globus

- Data Replication:
 - ▣ Conjunto de herramientas que permiten interactuar con GridFTP, RFT y RLS a un nivel de abstracción más alto
 - ▣ Dos grupos de herramientas:
 - Replication Client:
 - Permite realizar operaciones sobre replicas (registrar, transferir, etc). Se utilizan los nombres lógicos de ficheros
 - Batch Replicator
 - Proporciona herramientas para localizar replicas, transferir los ficheros usando RFT y para añadir nuevos mapeos una vez que los ficheros han llegado a su destino

Gestión de datos en Globus

□ Replication Client:

□ Transferir un fichero local y registrarlo

```
$ globus-replication-client -r rls://servidor1.atc put \  
./video.avi video gsiftp://servidor2.atc/Lustre/result.avi
```

□ Crear una replica del fichero y registrarla

```
$ globus-replication-client -r rls://servidor1.atc replicate \  
video gsiftp://servidor4.atc/shared/result.avi
```

□ Transferir una replica a un fichero local:

```
$ globus-replication-client -r rls://servidor1.atc get \  
video ./video.avi
```

Gestión de datos en Globus

□ Batch Replicator:

Replication request file

```
video    gsiftp://servidor4.atc/shared/result.avi  
Video    gsiftp://servidor5.atc/mnt/nfs/result.avi
```

```
$ globus-credential-delegate
```

Delegated credential

```
$ globus-replication-create
```

Replication resource

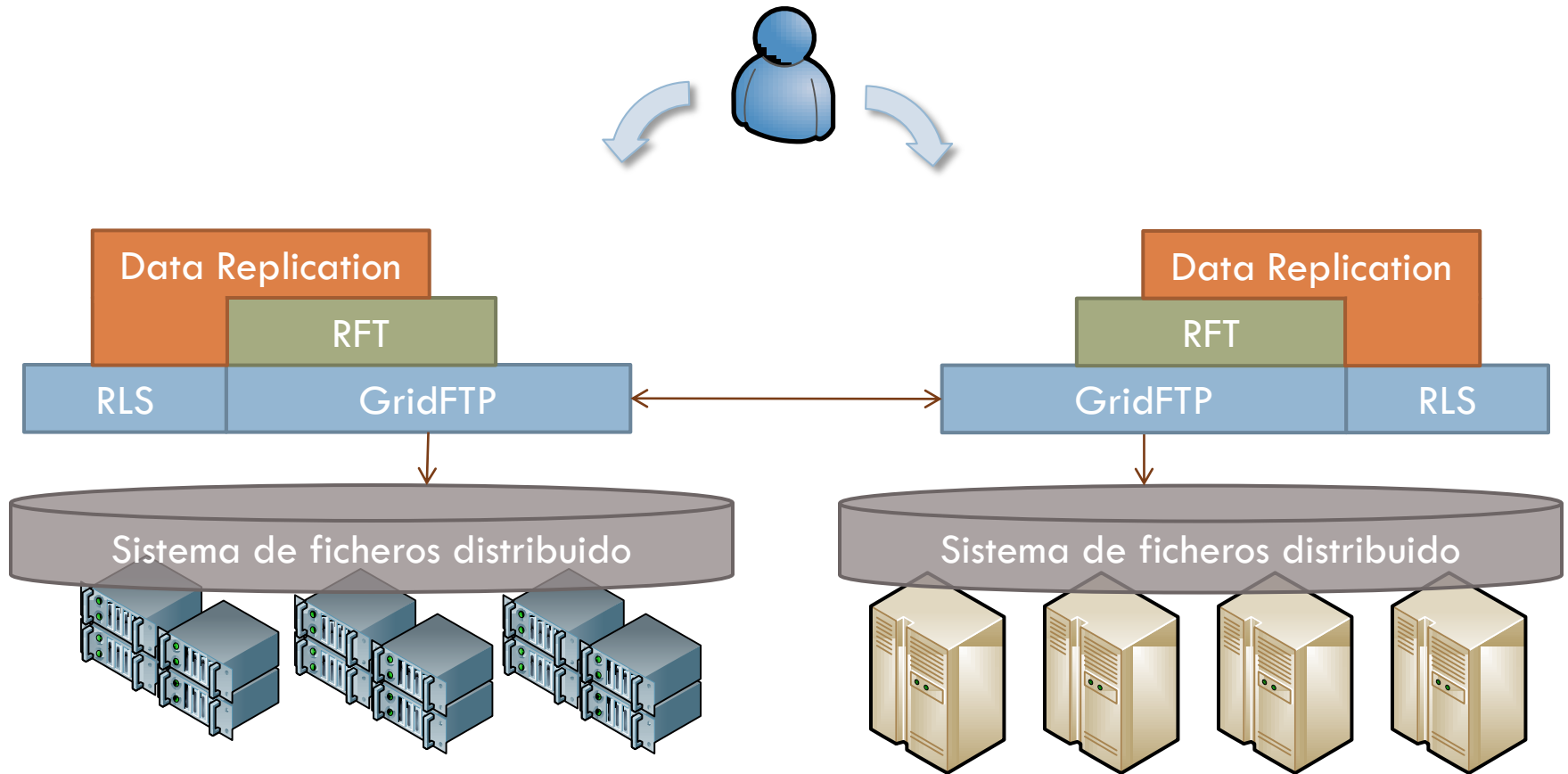
```
$ globus-replication-start
```

RLS

RFT

Gestión de datos en Globus

□ Integración:



Gestion de datos

OGSA-DAI



- Open Grid Services Architecture - Data Access and Integration
 - ▣ Framework basado en servicios web para flujos de trabajo (workflows) centrados en datos
 - ▣ Proporciona un interfaz comun para el acceso y manipulación de datos heterogénos
 - ▣ Sirve como base para servicios de más alto nivel
 - Federación de datos, minería de datos, visualización, ...
- OGSA-DAI es un middleware para "gridizar" bases de datos ya existentes

□ Funciona junto a diversos middleware:

□ Globus



□ Unicore



□ gLite



□ Apache Axis/Tomcat



- OGSA-DAI proporciona:
 - ▣ Acceso a datos
 - Datos estructurados en recursos de almacenamiento heterogéneos
 - ▣ Transformación de los datos
 - Permite transformar la forma en la que los datos están almacenados
 - ▣ Integración de los datos
 - Permite exponer los datos de múltiples bases de datos como si estuvieran almacenados en una sólo base de datos virtual
 - ▣ Envío de los datos
 - Permite enviar los datos a donde se necesite

□ Acceso:

- BD relacional: MySQL, Oracle DB2, SQL Server, Postgres,...
- XML: Xindice, eXist
- Ficheros: CSV, binarios, ...

□ Envío:

- SOAP sobre HTTP
- FTP, GridFTP
- E-mail

□ Transformación:

- XSLT
- ZIP, GZIP

□ Seguridad:

- Basada en certificados X.509

- ¿Qué hace OGSA-DAI?
 - ▣ Ejecuta workflows
 - Conjunto de tareas con dependencias (DAG)
 - ▣ Los workflows contiene actividades
 - Una actividad
 - Recibe unos datos de entrada
 - Procesa los datos
 - Produce unos datos de salida
 - ▣ El workflow se diseña mediante un lenguaje de programación (java) + client toolkit (librería)
 - El programa se conecta al servicio web de OGSA-DAI y le envía el workflow como un documento XML

- Ejemplos de actividades
 - ▣ Ejecutar una query SQL
 - SQLQuery
 - ▣ Comprimir un conjunto de datos en un fichero ZIP
 - ZIPCompression
 - ▣ Listar los ficheros de un directorio
 - ListDirectory
 - ▣ Ejecutar una transformación XSL a un fichero XML
 - XSLTransformation
 - ▣ Enviar datos por correo electrónico
 - DeliverToSMTP

- Recursos de datos
 - ▣ Proporcionan una abstracción de los datos
 - ▣ Se deben dar de alta en el servidor OGSA-DAI
 - Recurso base de datos relacional
 - Recurso base de datos XML
 - Recurso sistema de ficheros
 - Recurso grupo (un conjunto de otros recursos de datos)
 - Recurso OGSA-DAI remoto
 - ▣ Los recursos de datos tienen asociados atributos:
 - Identificador, driver, usuario, contraseña, path, ..

Servidor OGSA-DAI

Recurso de datos

```
dai.resource.id=RecursoMySQL  
dai.db.product=MySQL  
dai.db.vendor=MySQL  
dai.db.version=5  
dai.db.uri=jdbc:mysql://servidor:3306  
dai.db.driver=org.gjt.mm.mysql.Driver  
dai.user=usuarioDAI  
dai.password=secreto1234
```

Recurso de datos

```
dai.resource.id=RecursoFichero  
dai.db.file.path=/shared
```

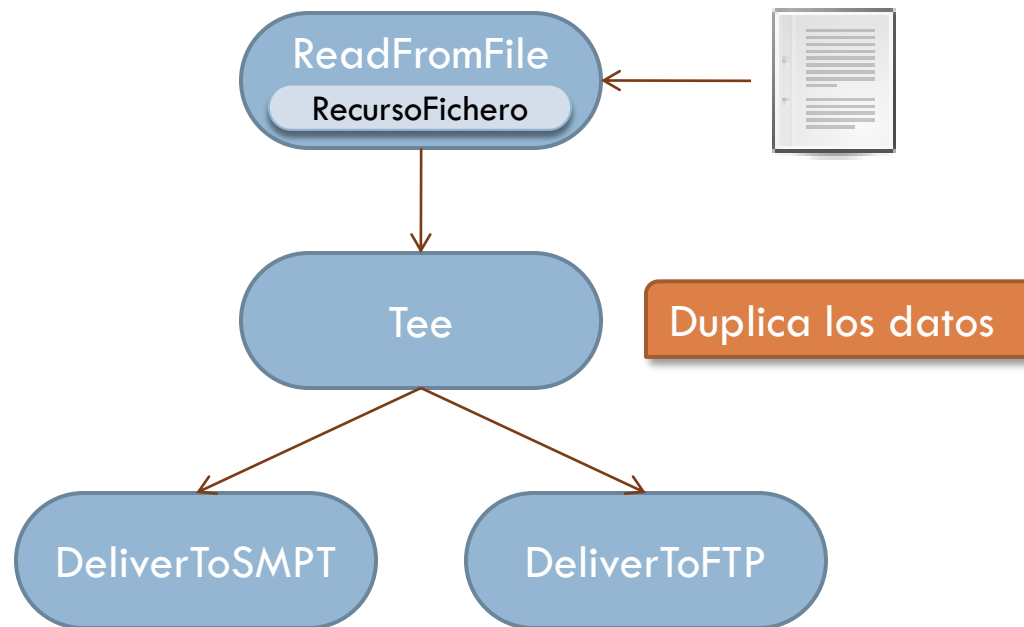
Recurso de datos

```
dai.resource.id=RecursoFichero2  
dai.db.file.path=/home/data
```

Recurso de datos

```
dai.data.resource.id=Recurso_OGSA-DAI_Remoto  
dai.data.resource.uri=http://servidor:8080/wsrf/services/dai/
```

- Ejemplo simple de workflow
 - ▣ Leer un fichero y enviarlo a un servidor de FTP y por correo electrónico

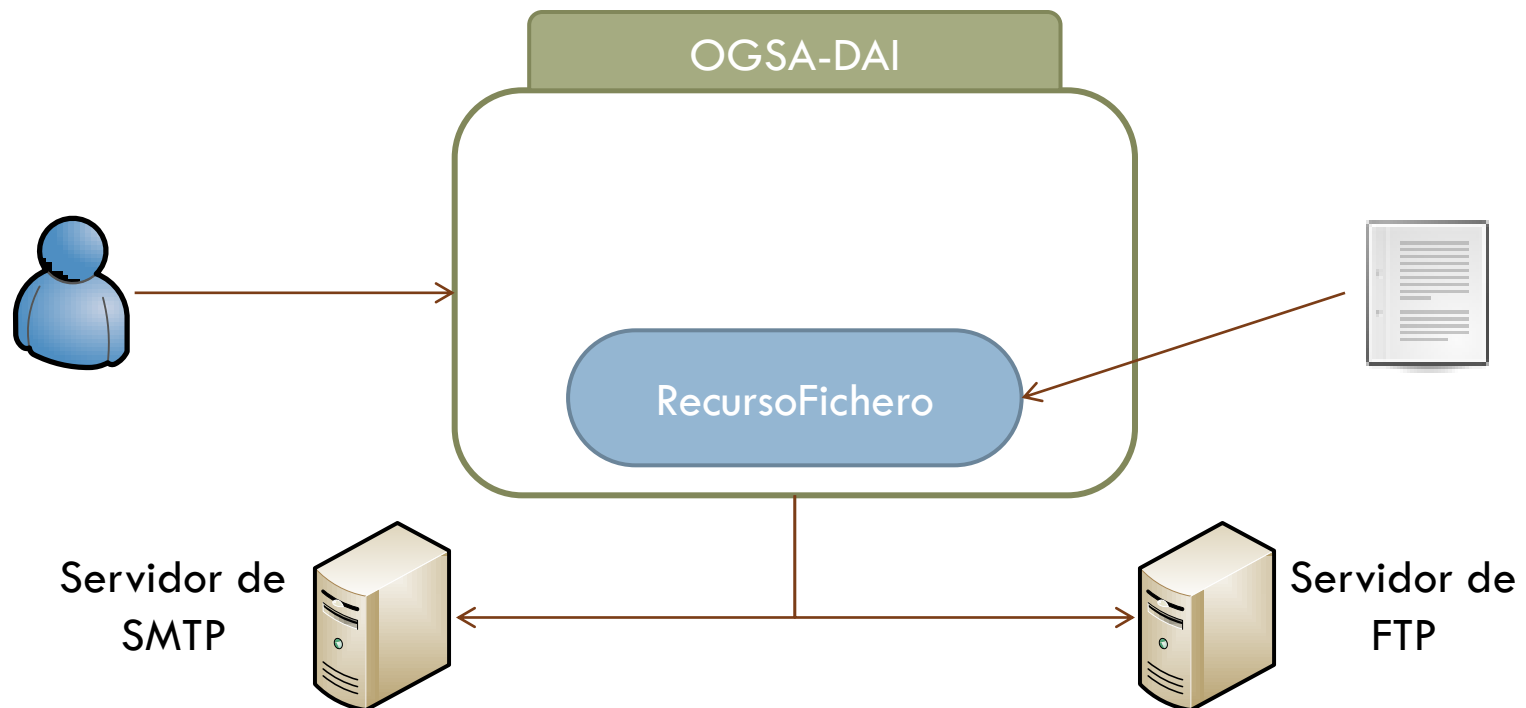


```
...
import uk.org.ogsadai.client.toolkit.activities.delivery.DeliverToFTP;
...
ServerProxy server = new ServerProxy();
server.setDefaultBaseServicesURL(new URL("http://servidor.atc/dai/services/"));
DataRequestExecutionResource drer =
    server.getDataRequestExecutionResource("DataRequestExecutionResource");
...
ReadFromFile readFromFile = new ReadFromFile();
readFromFile.setResourceID("RecursoFichero");
readFromFile.addFile("fichero.txt");
...
Tee tee = new Tee();
tee.connectInput(readFromFile.getDataOutput());
tee.setNumberOfOutputs(2);
...
deliverToFTP.connectDataInput(tee.getOutput(0));
deliverToSMTP.connectDataInput(tee.getOutput(1));
...
PipelineWorkflow pipeline = new PipelineWorkflow();
pipeline.add(readFromFile);
pipeline.add(tee);
...
RequestResource reqRes = drer.execute(pipeline, RequestExecutionType.SYNCHRONOUS);
...
```

Recurso dado de alta
en el servidor

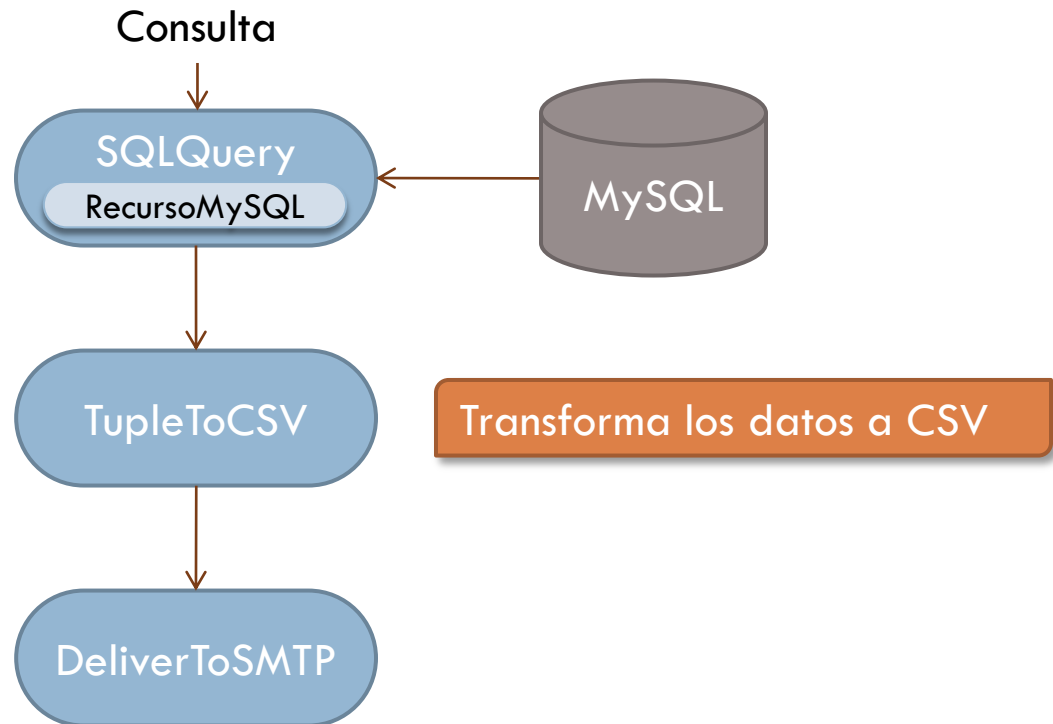
WorkflowSimple.java

□ Diagrama de recursos



□ Otro ejemplo simple

- ▣ Realizar una consulta a una base de datos y enviar el resultado por correo en formato CSV



```
...
String expression = "SELECT * FROM BaseDeDatos";
SQLQuery query = new SQLQuery();
query.setResourceID("RecursoMySQL");
query.addExpression(expression);

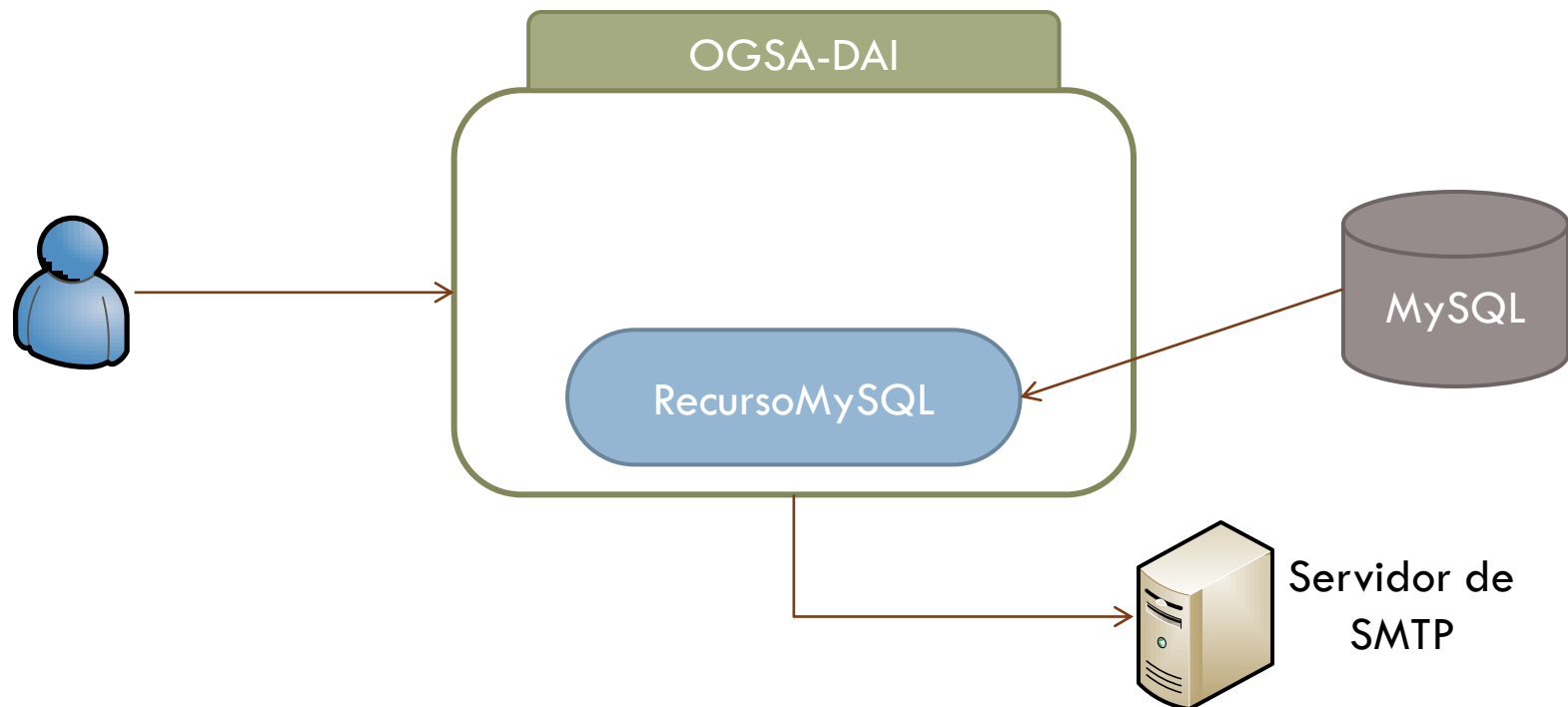
TupleToCSV tupleToCSV = new TupleToCSV();
tupleToCSV.connectDataInput(query.getDataOutput());

DeliverToSMTP deliverToSMTP = new DeliverToSMTP();
deliverToSMTP.connectDataInput(tupleToCSV.getResultOutput());
deliverToSMTP.addFrom("senderName@SMTPserver.com");
deliverToSMTP.addSubject("Resultados de una query SQL");
List to = new ArrayList();
to.add("destinatario1@uniovi.es");
to.add("destinatario2@uniovi.es");
deliverToSMTP.addTo(to.iterator());

PipelineWorkflow pipeline = new PipelineWorkflow();
pipeline.add(query);
pipeline.add(tupleToCSV);
pipeline.add(deliverToSMTP);
...
RequestResource reqRes = drer.execute(pipeline, RequestExecutionType.SYNCHRONOUS);
...
```

WorkflowSimple2.java

□ Diagrama de recursos



□ Un ejemplo práctico

▣ Un médico quiere realizar un estudio

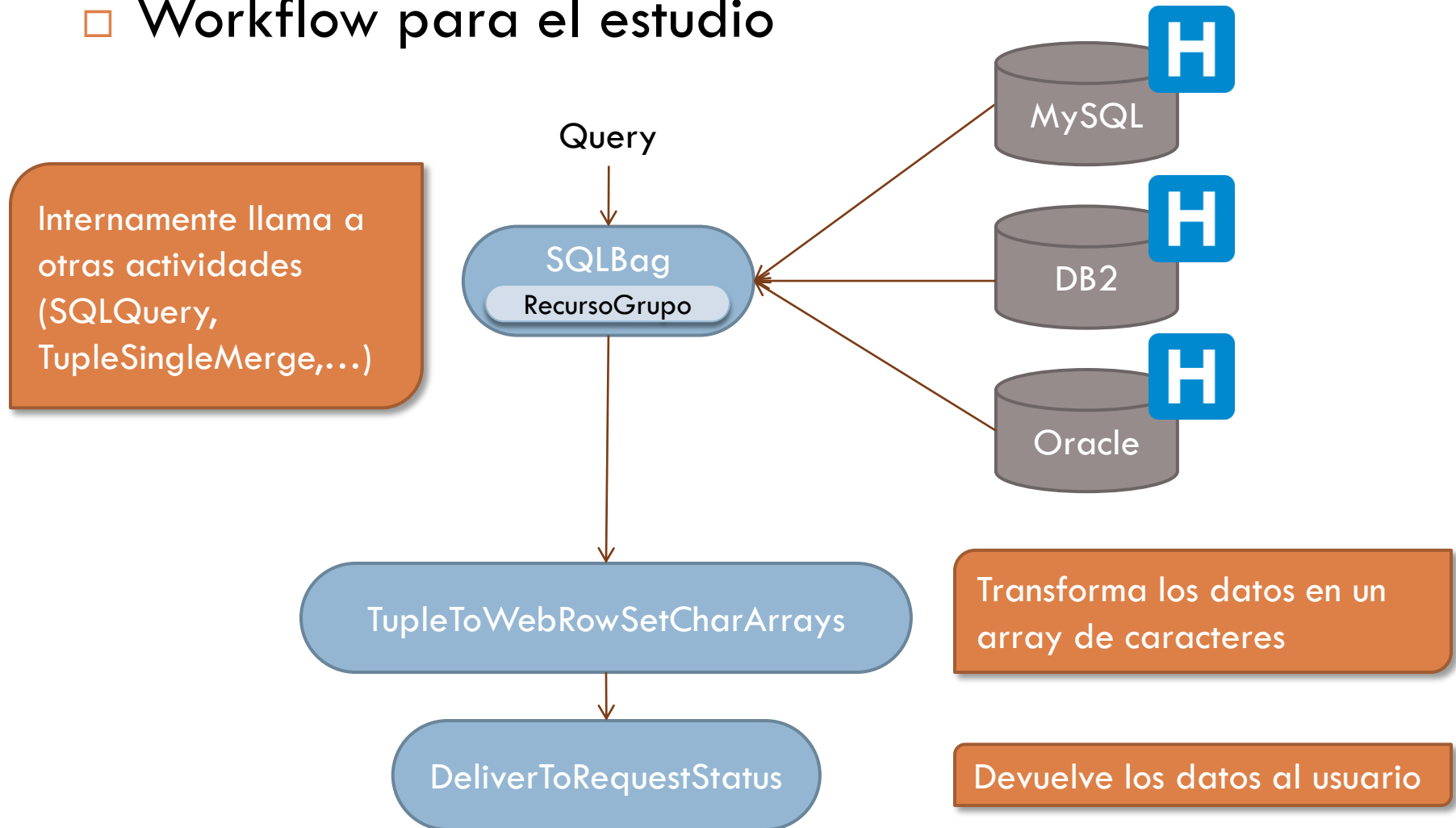
- Determinar la edad a la que los pacientes son diagnosticados con la varicela

```
SELECT edad, count(*) as total  
FROM Pacientes  
WHERE Diagnostico = "Varicela"  
GROUP BY edad  
ORDER BY edad
```

▣ Problema:

- Cada hospital tiene su base de datos
- Cada base de datos utiliza una tecnología diferente

□ Workflow para el estudio



```
...
SQLBag query = new SQLBag();
query.setResourceID("RecursoGrupo");
query.addExpression("SELECT edad, count(*) ...");

TupleToWebRowSetCharArrays tupleToWebRowSet = new TupleToWebRowSetCharArrays();
tupleToWebRowSet.connectDataInput(query.getDataOutput());

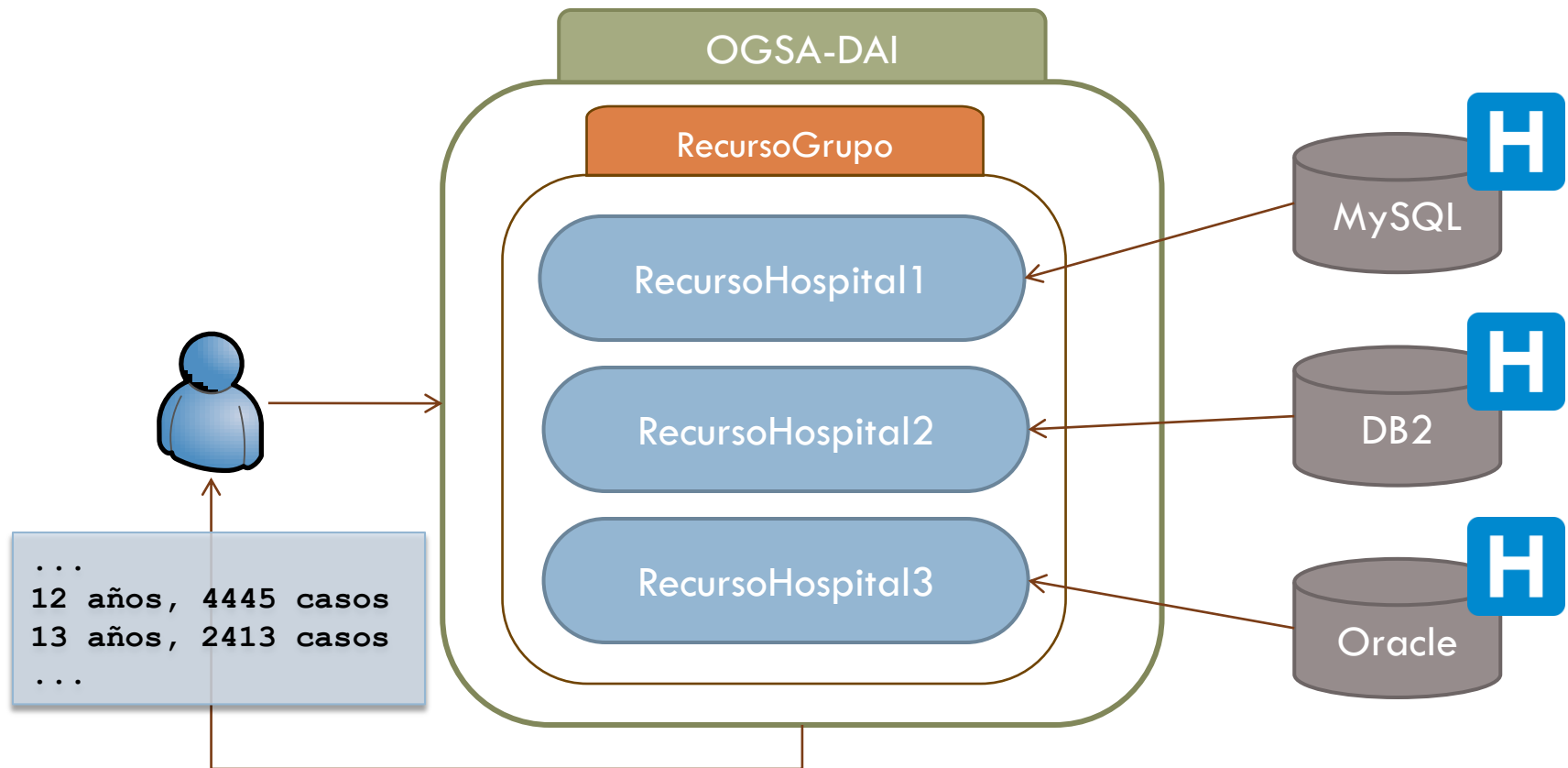
DeliverToRequestStatus deliverToRequestStatus = new DeliverToRequestStatus();
deliverToRequestStatus.connectInput(tupleToWebRowSet.getResultOutput());

PipelineWorkflow pipeline = new PipelineWorkflow();
pipeline.add(query);
pipeline.add(tupleToWebRowSet);
pipeline.add(deliverToRequestStatus);

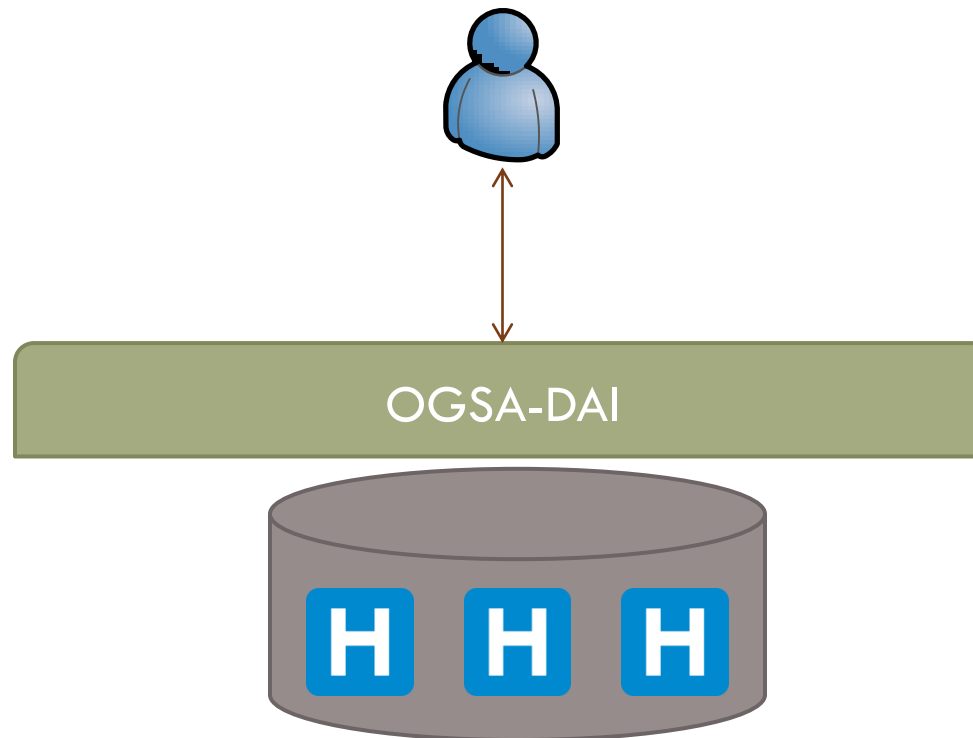
drer.execute(pipeline, RequestExecutionType.SYNCHRONOUS);
while (tupleToWebRowSet.hasNextResult()) {
    ResultSet rs = tupleToWebRowSet.nextResultAsResultSet();
    while (rs.next()) {
        ... // Hacer algo con los resultados
    }
}
...
```

WorkflowVaricela.java

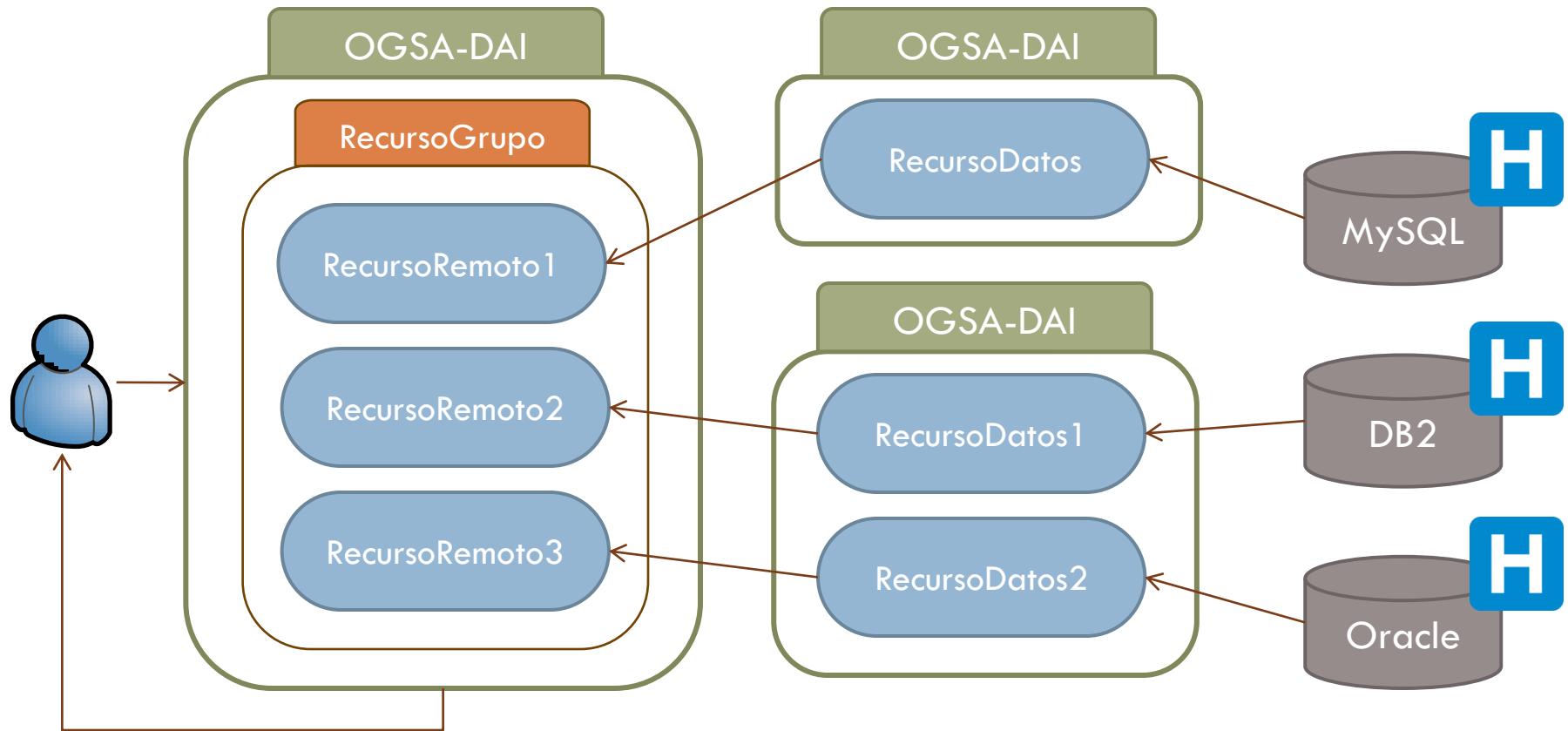
□ Diagrama de recursos



- Desde el punto de vista del usuario



□ Otras posibilidades



- Ventajas de OGSA-DAI respecto al acceso directo a los datos
 - ▣ Proporciona a los cliente un acceso homogéneo a datos heterogéneos
 - ▣ Permite realizar transformaciones de los datos en el servidor
 - Minimiza la comunicación con el cliente
 - ▣ Modelo de seguridad consistente
 - El acceso directo implica la distribución de usuarios y contraseñas para cada recurso
 - ▣ Ofrece sus servicios como un servidor WSRF
 - Su uso es independiente de plataforma y de lenguaje



Gestion de datos

Storage Resource Manager (SRM)

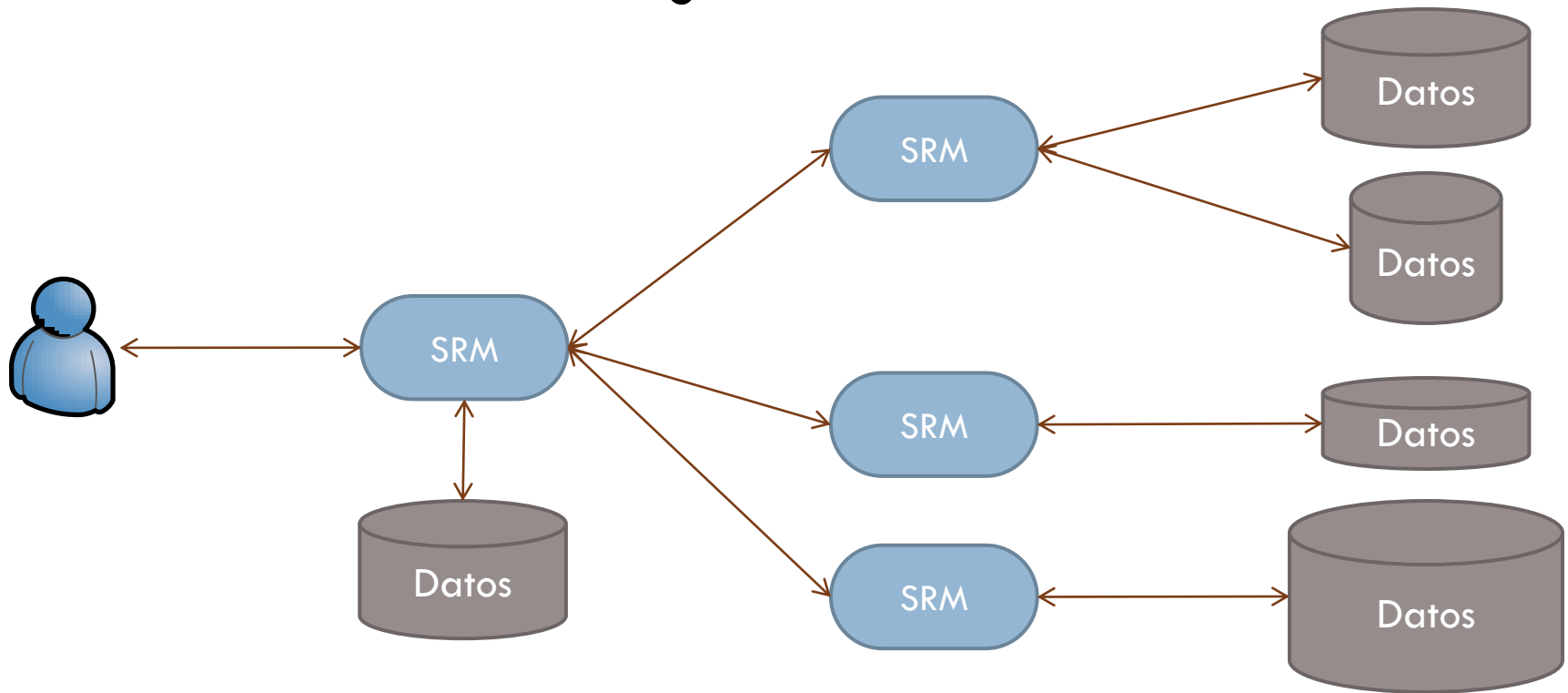
Storage Resource Manager (SRM)

- Estándar establecido por OGF (Open Grid Forum)
 - ▣ Define la interfaz de los recursos de almacenamiento
 - ▣ No define un protocolo de transferencia
- Funciones:
 - ▣ Reserva y liberación de espacio
 - ▣ Transferencia de información y negociación del protocolo utilizado
 - ▣ Gestión de ficheros, directorios y permisos
 - ▣ Interoperabilidad con otros SRM

Storage Resource Manager (SRM)

□ SRM

- ▣ Capa consistente y homogénea sobre una infraestructura heterogénea



Storage Resource Manager (SRM)

□ Necesidades en un posible escenario de ejecución de un trabajo:

1. Reservar espacio
 2. Transferir ficheros
 3. Procesar los ficheros
 4. Transferir los resultados
 5. Liberar el espacio ocupado
-
- The diagram illustrates the mapping of five tasks to system components. Tasks 1 and 2 (Reservar espacio and Transferir ficheros) are mapped to the first SRM component. Task 3 (Procesar los ficheros) is mapped to the GRAM, Condor component. Tasks 4 and 5 (Transferir los resultados and Liberar el espacio ocupado) are mapped to the second SRM component. The components are represented as rounded rectangles: SRM (blue) and GRAM, Condor (green).

SRM gestiona los recursos de almacenamiento en un entorno multiusuario

Storage Resource Manager (SRM)

- Parte del interfaz SRM
 - ▣ Gestión de ficheros y directorios
 - srmLs, srmMkdir, srmMv, srmRm, srmRmdir
 - ▣ Gestión de espacio
 - srmReserveSpace, srmReleaseSpace, srmUpdateSpace, srmGetSpaceTokens
 - ▣ Gestión de transferencias
 - srmPrepareToGet, srmPrepareToPut, srmCopy
 - ▣ Descubrimiento y permisos
 - srmPing, srmGetTransferProtocols, srmCheckPermission, srmSetPermission

Storage Resource Manager (SRM)

□ Ejemplos de uso de SRM

□ Reservar espacio

```
$ srm-sp-reserve -serviceurl servidor.atc -size 1000000 -lifetime 900
```

□ Copiar un fichero al espacio reservado

```
$ srm-copy file:///home/ruf/fich.dat \  
srm://servidor.atc/shared/fich.dat -spac token $SPTOKEN
```

□ Copiar un ficheros entre SRMs

```
$ srm-copy srm://servidor.atc/shared/res.dat  
srm://servidor2.atc/Data/res.dat
```

□ Liberar espacio

```
$ srm-sp-release -serviceurl servidor.atc -spac token $SPTOKEN
```

Storage Resource Manager (SRM)

- Negociación de la transferencia
 - ▣ Los clientes y servidores exponen los protocolos de transferencia de información que soportan y sus preferencias
 - gsiftp, ftp, ssh,...
 - ▣ Se ponen de acuerdo en utilizar uno y lo usan para transferir la información
 - ▣ En caso de que falle se prueba con otro protocolo
 - Tolerancia a fallos

Storage Resource Manager (SRM)

□ Acceso a los ficheros

- SRM proporciona una abstracción para el nombre de un fichero denominada SURL (Site URL)

```
srm://servidor.at/shared/fich.dat
```

- Un cliente debe indicar el SURL cuando quiere acceder a un fichero en un SRM
 - El SURL puede hacer referencia a una o más localizaciones físicas
- El servidor responde con un TURL (Transfer URL) que se utilizará para acceder al fichero de forma temporal

```
gsiftp://mnt/lustre/fich.dat
```

Storage Resource Manager (SRM)

- Algunas herramientas que implementan SRM v2.2:
 - dCache
 - BeStMan (Berkeley Storage Manager)
 - CASTOR (CERN Advanced STORage manager)
 - DPM (Disk Pool Manager)
 - StoRM (Storage Resource Manager)
- SRM-Tester
 - Utilidad para comprobar que las herramientas cumplan con las especificaciones

Storage Resource Manager (SRM)

□ Un posible escenario:

▣ SRM Frontend

- Expone los servicios Web
- Gestiona la seguridad

▣ SRM db

- Almacena las peticiones y el estado
- Almacena información sobre el espacio disponible

▣ SRM Backend

- Ejecuta las operaciones sobre el sistema de ficheros

