

Tecnologías Grid

Globus Toolkit

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



Globus Toolkit

Introducción

□ Globus Toolkit:

▣ Software de código abierto para construir Grids

- Apache License version 2 (tipo BSD)
- Estándar de facto para Grid computing
- El "Grid SDK": base para desarrollar herramientas de Grid

▣ Desarrollado por la Globus Alliance

- Liderado por la Universidad de Chicago

▣ Hitos más importantes

- Comienzo del proyecto: 1996
- Versión 1.0: 1998
- Versión 2.0: 2002
- Versión 4.0: 2005 - Primera basada en servicios web (WS)

Introducción

- Motivación para Globus
 - ▣ Heterogeneidad: distintos sitios, distintas políticas
 - Colas de trabajos, sistemas de monitorización, protocolos de red, etc.
 - ▣ Globus unifica mediante estándares
 - Basado en servicios web (Web Services, WS)
 - WS-RF
 - WS-Notification
 - Interfaces y abstracciones comunes

Introducción

- Enfoque de Globus
 - ▣ Herramientas y servicios para tratar los principales problemas técnicos
 - ▣ Modelo de "bolsa de servicios"
 - No es una solución integrada verticalmente
- Uso
 - ▣ Decenas de Grids nacionales, centenas de aplicaciones...
 - ▣ Para todo tipo de ciencias
 - ▣ Empleado en sistemas reales

Introducción

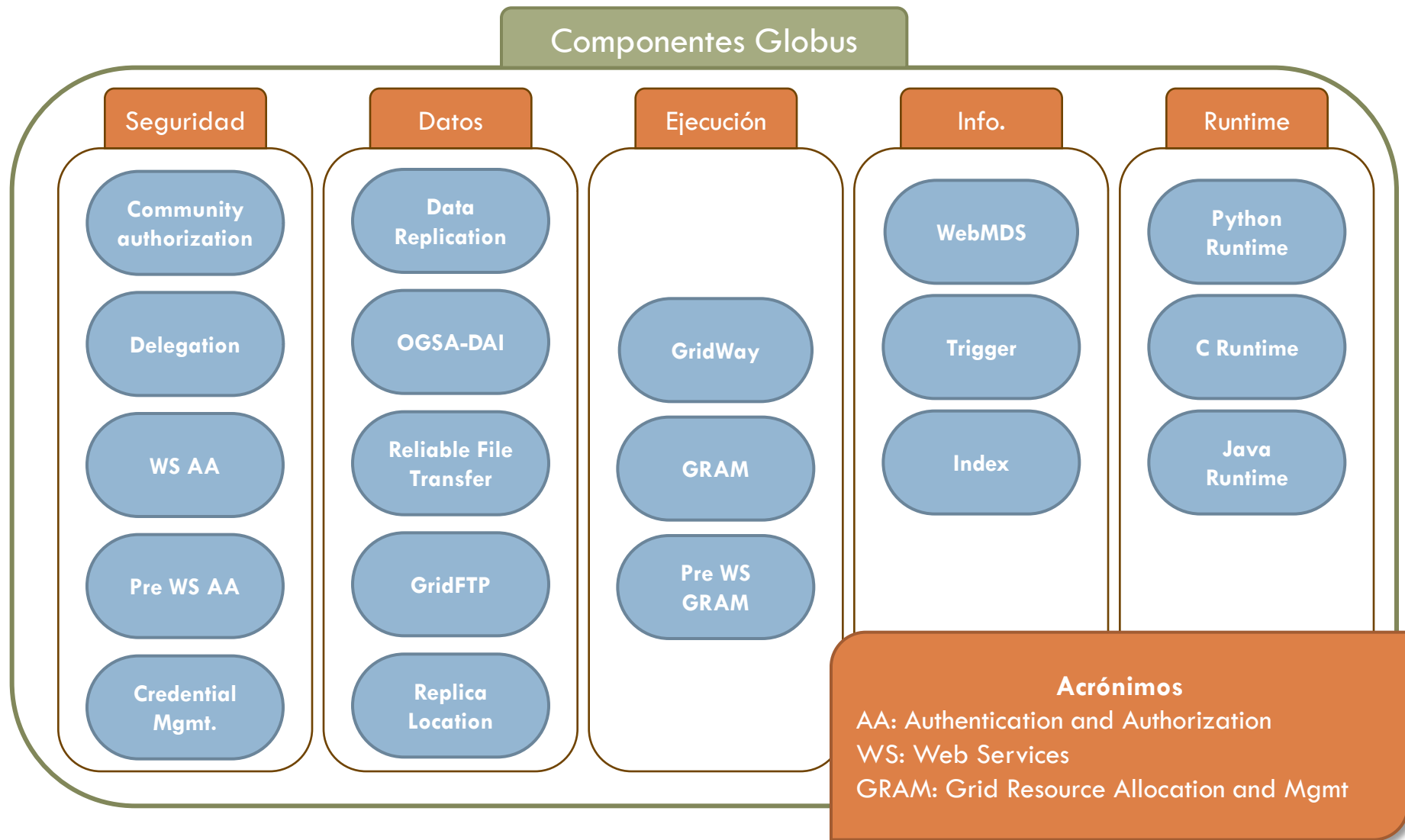
- Desarrollo (I)
 - ▣ dev.globus.org
 - ▣ Modelo de gobierno tipo Apache Jakarta
 - Basado en el consenso
 - ▣ Organización en proyectos
 - Cada proyecto tiene sus responsables
 - Coordinación entre proyectos
 - Interacciones compartidas
 - Reuniones entre responsables
 - Globus Management Comitee
 - Guía general y resolución de conflictos

- Desarrollo (II)
 - ▣ Proyectos no tecnológicos
 - Proyectos de distribución
 - Proyectos de documentación
 - Proyectos en la incubadora
 - Proyecto de gestión de incubadora
 - Proyectos que se quieren unir a Globus

Introducción

- Áreas tecnológicas en Globus
 - ▣ Core runtime
 - Infraestructura para construir nuevos servicios
 - ▣ Seguridad
 - Aplicar políticas uniformes entre distintos sistemas
 - ▣ Gestión de la ejecución
 - Provisión, despliegue y gestión de servicios
 - ▣ Gestión de datos
 - Descubrimiento, transferencia y acceso a grandes datos
 - ▣ Monitorización
 - Descubrimiento y monitorización de servicios dinámicos

Introducción





Globus Toolkit

Entorno de ejecución (runtime)

Entorno de ejecución

- Dos grandes versiones
 - ▣ Pre-WS
 - Basada en protocolos propios
 - ▣ WS (GT4)
 - Basada en servicios web
- ¿Por qué servicios web?
 - ▣ Independientes de la plataforma y del lenguaje
 - ▣ Adecuados para sistemas con bajo acoplamiento
 - Al contrario que CORBA, EJB, etc.
 - ▣ Estándares
 - ▣ Se autodescriben

Entorno de ejecución

□ Definición de servicio web

W3C

Sistema software diseñado para soportar interacciones máquina a máquina sobre una red

■ Típicamente: Servidores y clientes que se comunican por HTTP

■ Servicio web $\neq$ página web

- El usuario es software, no un humano

■ Estándar de descripción: WSDL

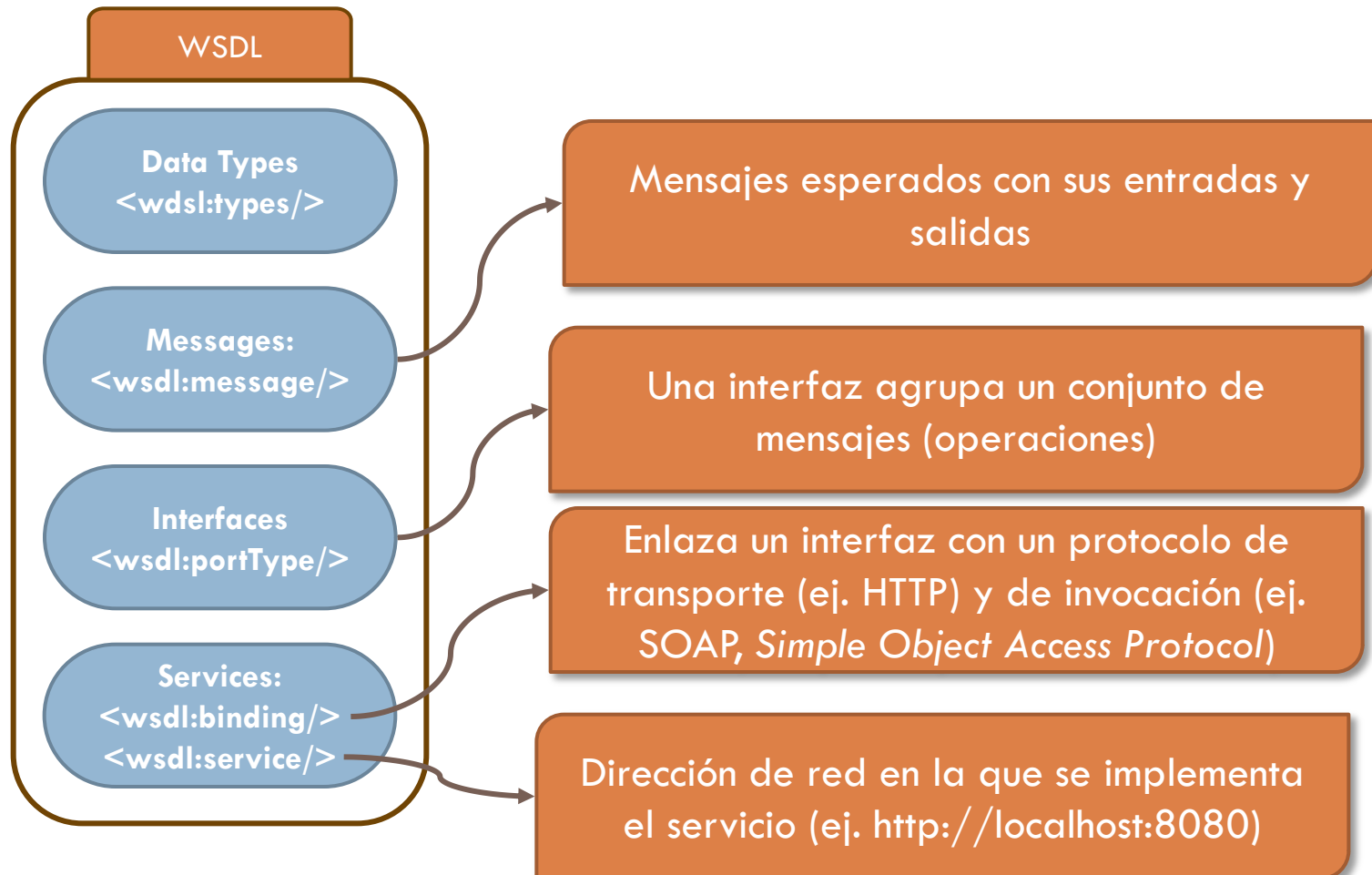
■ Web Services Description Language

■ Basado en XML

■ Servicio = colección de puntos finales de red (puertos)

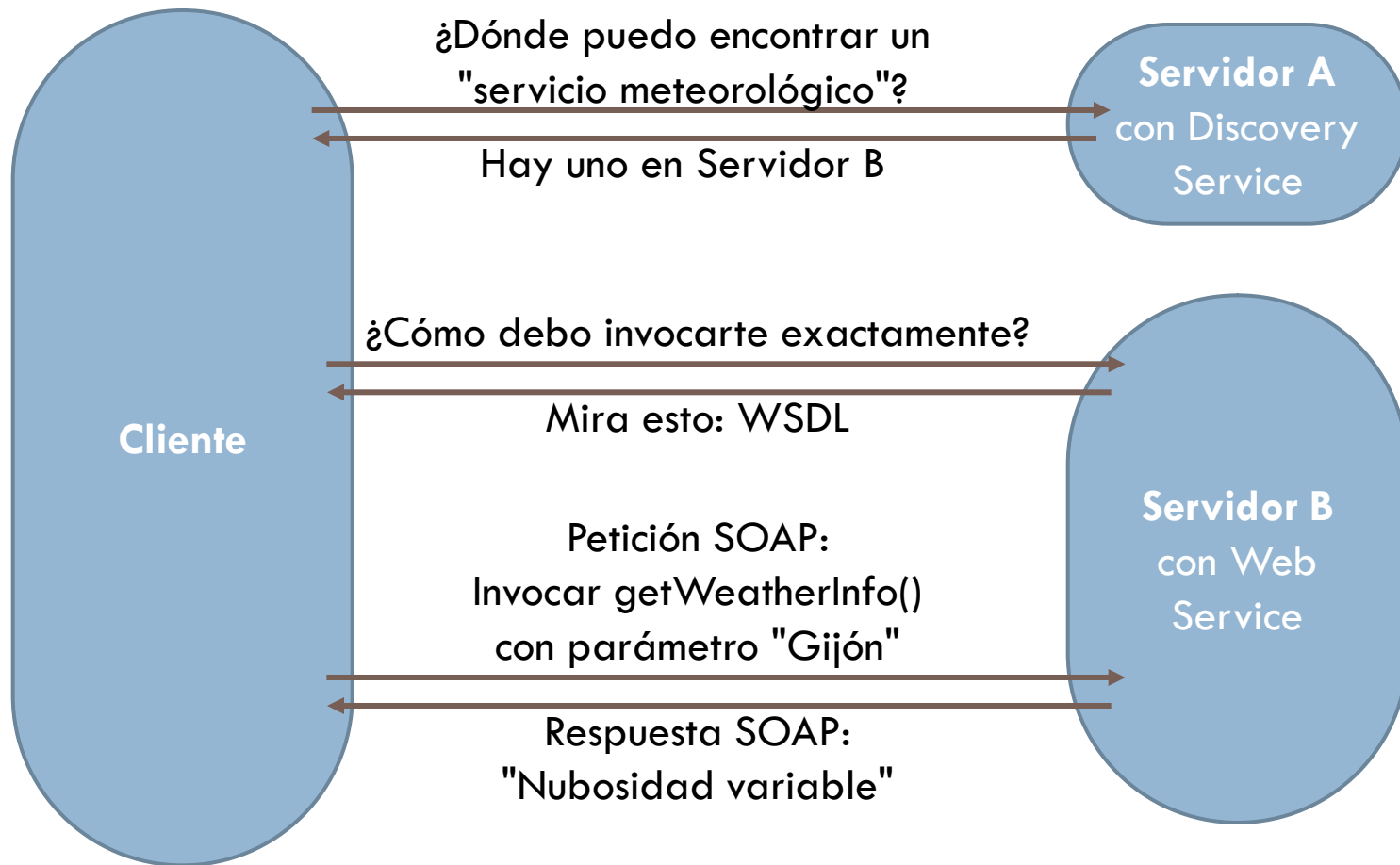
Entorno de ejecución

□ Estructura de un fichero WSDL



Entorno de ejecución

□ Funcionamiento de un servicio web



Entorno de ejecución

□ Contenedor de WS

▣ Servidor HTTP

■ Gestiona mensajes HTTP

■ Ej. Apache

▣ Servidor de aplicaciones

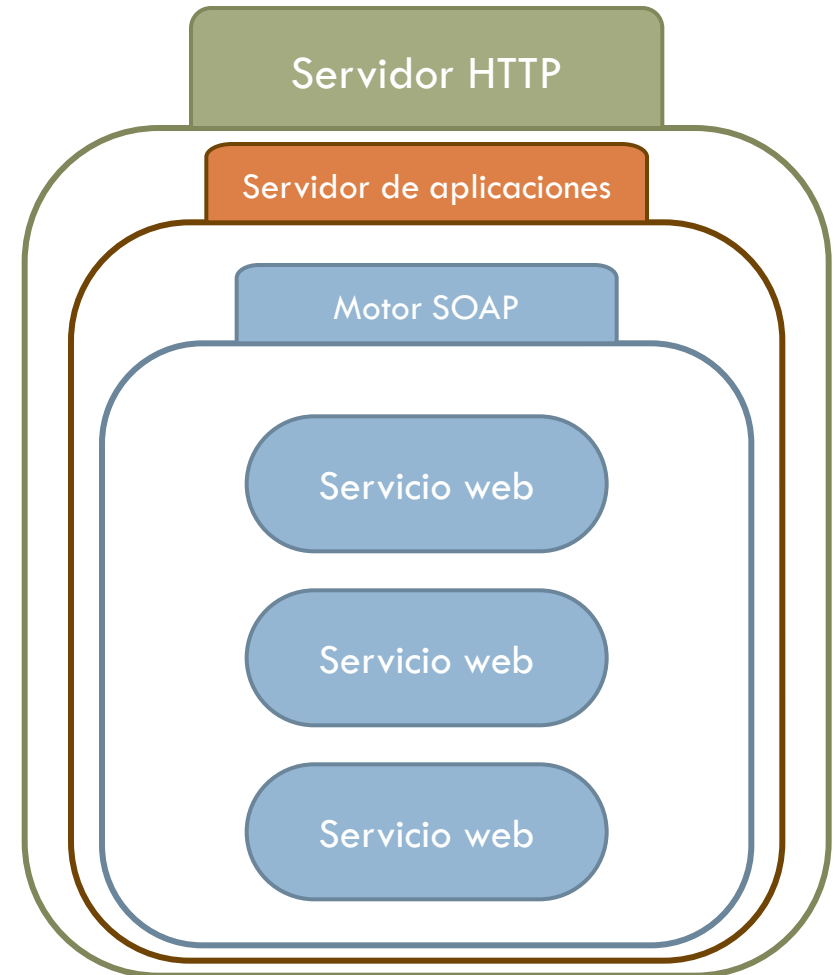
■ Espacio para aplicaciones que deben ser accedidas por distintos clientes

■ Ej. Tomcat

▣ Motor SOAP

■ Gestiona peticiones SOAP

■ Ej. Apache Axis



Entorno de ejecución

□ Servicios con estado

▣ Los servicios no tienen estado

- No guardan información entre invocaciones

▣ Problema: Muchas aplicaciones Grid requieren estado

■ Solución: Guardar el estado en un recurso

- Soluciones ad-hoc: utilizar bases de datos, sesiones en cookies...
- Solución propuesta por Globus Alliance e IBM: WSRF y WSN
 - Estándares OASIS

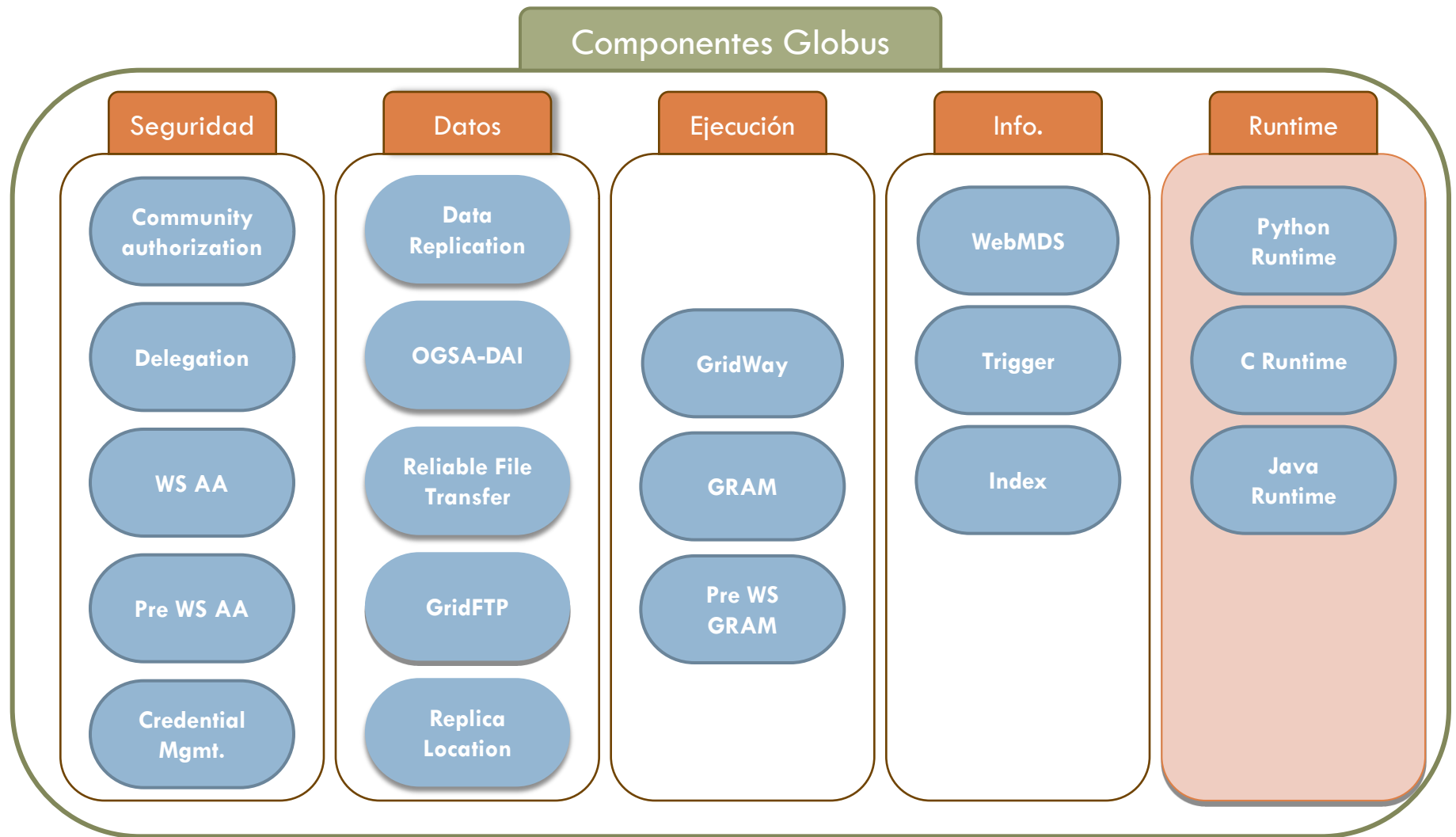
▣ Web Service Resource Framework (WSRF)

- Define interfaces estándar para acceder a WS-Resources

▣ WS-Notification (WSN)

- Permite programación orientada a eventos entre WS

Entorno de ejecución



Entorno de ejecución

- Entorno de ejecución común (*Common Runtime*)
 - ▣ Componentes que proporcionan librerías y herramientas para que los servicios de Globus Toolkit sean independientes de la plataforma
 - C Runtime
 - Capa de abstracción para tipos y estructuras de datos y llamadas a libc
 - C WS core, Java WS core y Python WS core
 - Implementación de WS, WSRF y WSN en C, Java y Python
 - Permiten implementar servicios y clientes web en esos lenguajes
 - Python WS core es una contribución externa a Globus
 - Muy básica



Globus Toolkit

Seguridad

Seguridad

□ Necesidades

- ▣ Comunicación segura entre los componentes de un Grid
- ▣ Seguridad sobre distintas organizaciones
 - No permite un sistema de gestión central de la seguridad
- ▣ Soporte para *Single Sign-On*

□ Solución de Globus

- ▣ Grid Security Infrastructure (GSI)
 - Basada en un conjunto de estándares de la IETF
 - Autenticación basada en certificados X.509
 - Criptografía de clave pública (criptografía asimétrica)

Seguridad

- Certificados X.509
 - ▣ Identifican usuarios, máquinas y servicios
- Autenticación mutua
 - ▣ Las dos partes deben
 - Tener certificados
 - Reconocer las CAs respectivas
 - Deben tener el certificado de la CA de la otra parte
 - Deben confiar en ese certificado
 - ▣ En Globus, se consigue con *Secure Socket Layer (SSL)*
 - También llamado *Transport Layer Security (TLS)*

Seguridad

- Comunicación confidencial e íntegra
 - ▣ Confidencialidad
 - Por defecto, la comunicación entre partes no está cifrada
 - ▣ Integridad
 - Por defecto, se proporcionan mecanismos para asegurar la integridad de la comunicación
 - Cualquiera que escuche puede entender la comunicación pero no modificarla
 - ▣ Se pueden activar o desactivar la confidencialidad y la integridad

Seguridad

- Protección de la clave privada
 - ▣ Responsabilidad de cada usuario
 - ▣ Clave privada almacenada en el ordenador del usuario protegida por una frase de paso
- Delegación y *Single Sign-On*
 - ▣ Objetivo: reducir el nº de veces que el usuario tiene que teclear la frase de paso
 - Una computación en Grid puede requerir acceso a muchos recursos que requieran autenticación
 - ▣ Solución: delegación
 - Usar un *proxy* (poder, representante)

□ Delegación

▣ Proxy = nuevo certificado + nueva clave privada

■ Nuevo certificado

- Incluye una nueva clave pública
- Incluye la identidad del usuario pero indicando que es un proxy
- Firmado por el usuario, no por una CA
- Incluye una fecha de caducidad

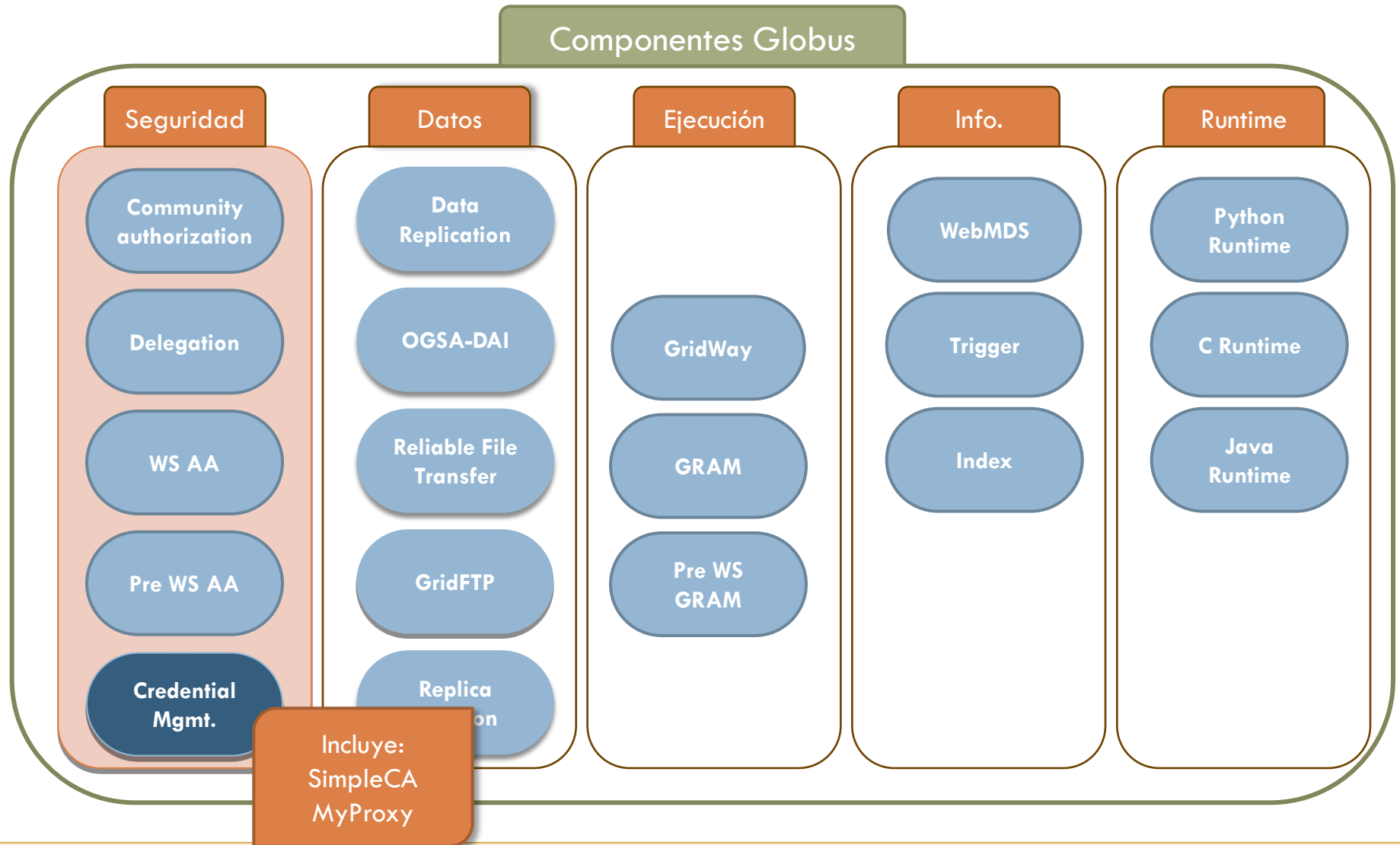
■ Nueva clave privada

- Seguridad menos crítica si tiene validez limitada en el tiempo
 - Se puede almacenar sin cifrar (con permisos de lectura sólo para el usuario)

■ Se puede usar para autenticar al usuario

- Sin necesidad de introducir la frase de paso

Seguridad



□ SimpleCA (I)

- ▣ Paquete que proporciona una autoridad certificadora simple
- ▣ Objetivo
 - Proporcionar credenciales a usuarios y servicios de Globus
 - Para hacer pruebas cuando no se dispone de una autoridad certificadora
 - No pensado para sistemas en producción
 - No es una verdadera CA
 - No revoca ni regenera certificados
 - No verifica la identidad
 - El servicio no es especialmente seguro

□ SimpleCA (II)

▣ Funcionamiento (I)

- En la instalación de una máquina con Globus, ejecutar:

- `setup-simple-ca`
- `setup-gsi -default`

- Pedir certificados de máquina (*host*)

- `grid-cert-request -host 'hostname'`
 - Crea tres ficheros en `/etc/grid-security/`
`hostkey.pem`, `hostcert_request.pem` y `hostcert.pem`

En este paso
`hostcert.pem`
está vacío

- Firmar certificados de máquina

- `grid-ca-sign -in hostcert_request.pem -out hostsigned.pem`
 - Copiar `hostsIGNED.pem` a `/etc/grid-security/hostcert.pem`

□ SimpleCA (III)

▣ Funcionamiento (II)

■ Pedir certificados de usuario

■ grid-cert-request

■ Crea tres ficheros en \$HOME/.globus

userkey.pem, usercert_request.pem y usercert.pem

En este paso
usercert.pem
está vacío

■ Firmar certificados de usuario

■ Enviar a la CA el fichero usercert_request.pem

■ grid-ca-sign -in usercert_request.pem -out signed.pem

■ El responsable de la CA envía signed.pem al usuario

■ El usuario debe copiarlo como \$HOME/.globus/usercert.pem

□ Servicio MyProxy

▣ Servicio de repositorio *on-line* de credenciales

■ Almacena credenciales

- Protegidas por una palabra de paso
- Accesibles a través de la red

■ Elimina la necesidad de copiar claves privadas y certificados entre máquinas

■ Sirve también para autenticarse en portales Grid y renovar credenciales con gestores de trabajo (*job managers*)

■ Almacenar y obtener credenciales de proxy:

- myproxy-init, myproxy-logon

■ Almacenar y obtener credenciales de usuario final:

- myproxy-store, myproxy-retrieve

Seguridad

□ Tipos de credenciales

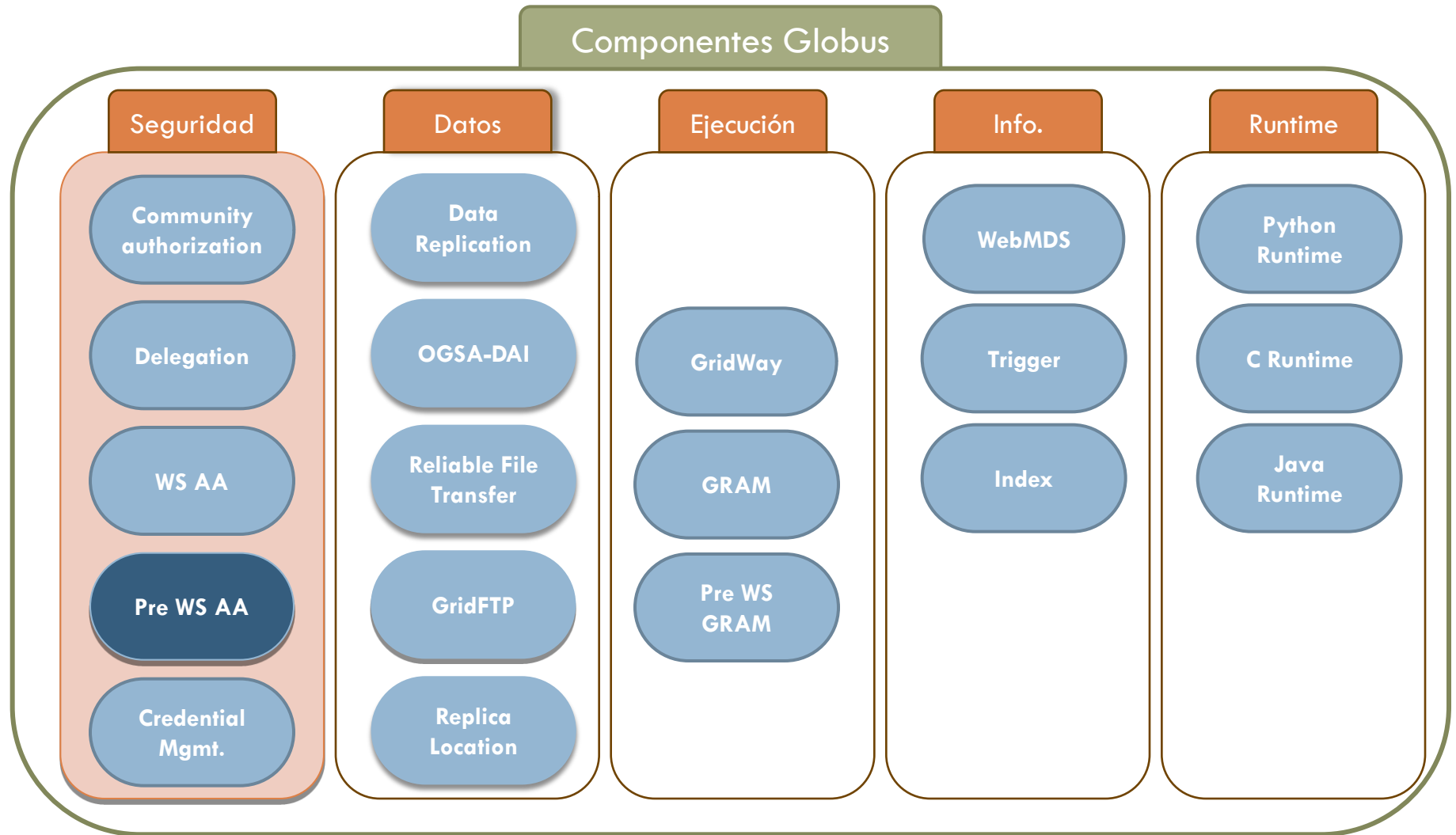
▣ De CA

- Utilizado para verificar la firma de una CA
 - Típicamente en /etc/grid-security/<hash>.0
 - El <hash> es el de la CA

▣ EEC (End Entity Certificate)

- Cualquiera que no sea de una CA
 - De usuario
 - De host
 - De servicio
 - Proxy

Seguridad



- Pre-Web Services Authentication and Authorization
 - ▣ APIs y herramientas para autenticación, autorización y gestión de certificados
 - ▣ Autorización basada en el mapfile
 - Mapea nombres distinguidos (los presentes en los certificados) a usuarios locales
 - Al final, las computaciones de un usuario del Grid tienen que ejecutarse como un usuario del sistema operativo
 - Pueden servir de lista de control de acceso para servicios que funcionen con GSI
 - Típicamente en `/etc/grid-security/grid-mapfile`

□ Pre-Web Services Authentication and Authorization

□ Órdenes:

■ Autenticación

■ Generación y gestión de proxys

- grid-proxy-init, grid-proy-destroy, grid-proxy-info
- El proxy es un fichero en /tmp/x509up_u<uid>

■ Autorización

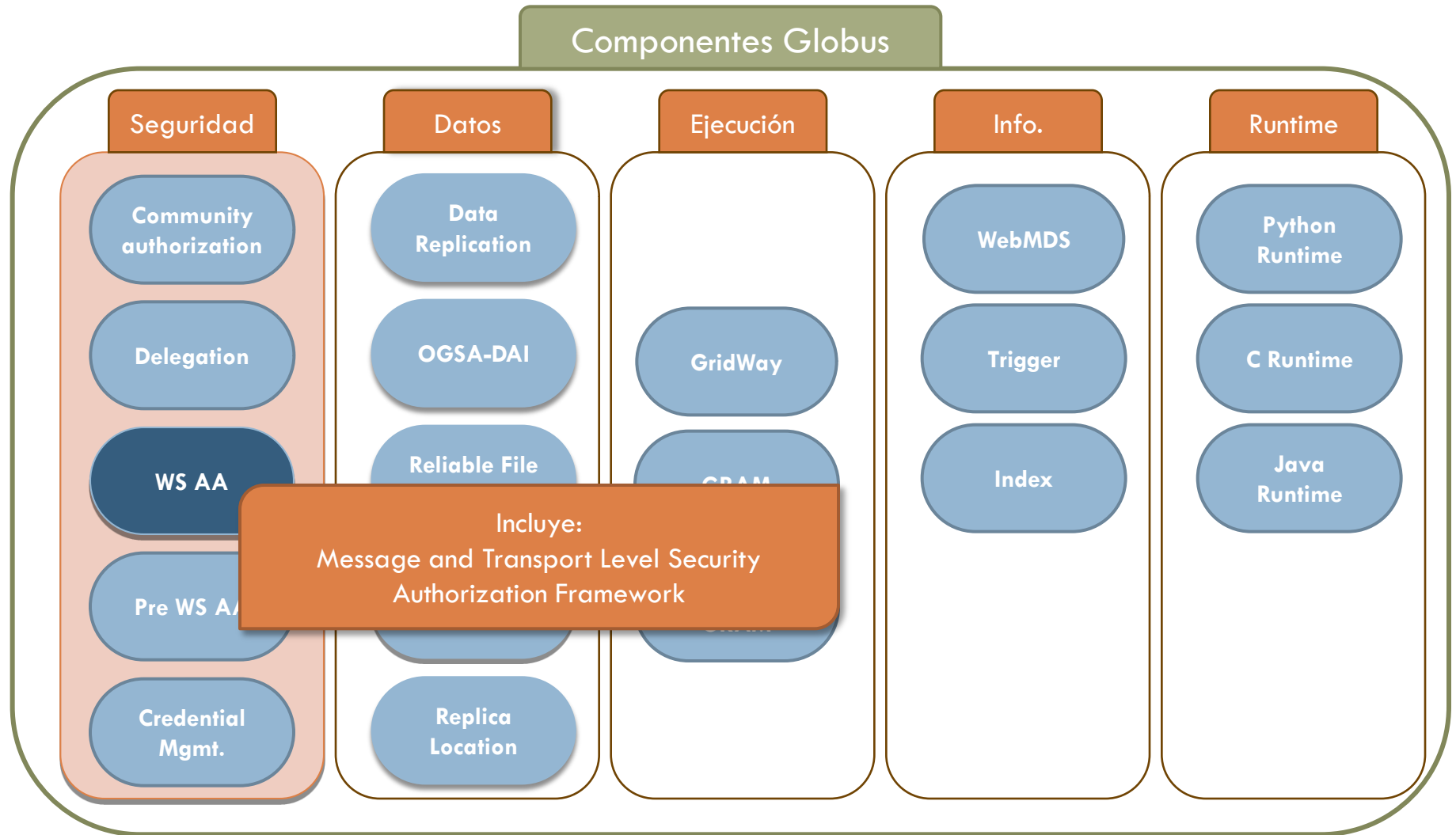
■ Gestión del mapfile

- grid-mapfile-add-entry, grid-mapfile-check-consistency, grid-mapfile-delete-entry

■ Gestión de certificados

- grid-cert-info, grid-cert-request, grid-default-ca, grid-change-passphrase

Seguridad

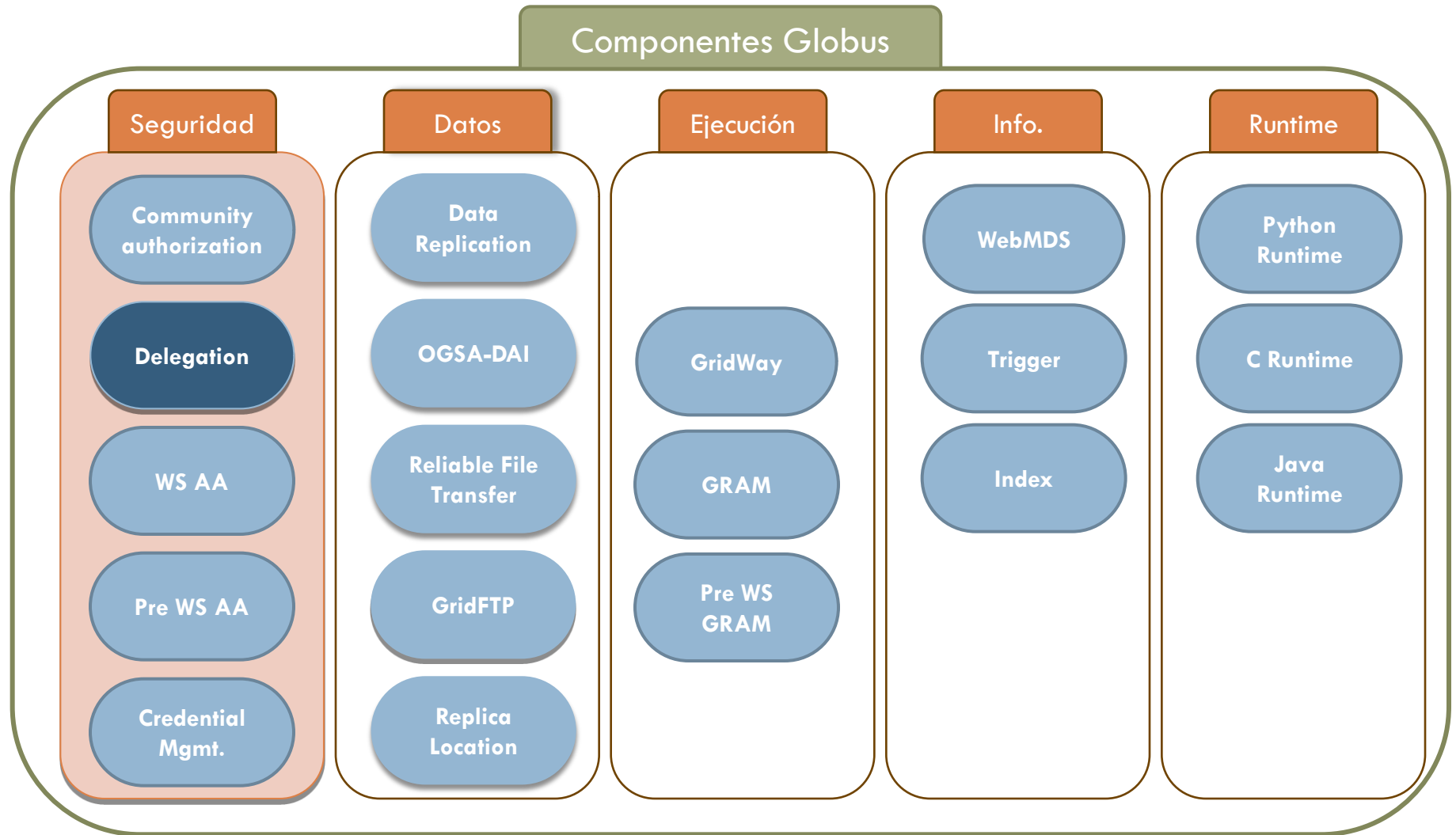


- Message and Transport Level Security
 - ▣ Proporciona protección a los mensajes SOAP
 - Usa HTTPS (HTTP over SSL/TLS)
 - ▣ Implementa los estándares
 - WS-Security
 - WS-SecureConversation
 - ▣ Extiende HTTPS para usar certificados proxy

□ Authorization Framework

- ▣ Proporciona un *framework* para autorización a nivel de contenedor
- ▣ Distintas implementaciones de la autorización
 - none
 - self
 - Sólo se autoriza a servicios de uno mismo
 - gridmap
 - Utiliza el grid-mapfile
 - SAML (Security Assertion Markup Language)
 - Permite pasar la autorización a entidades externas
 - Otros

Seguridad



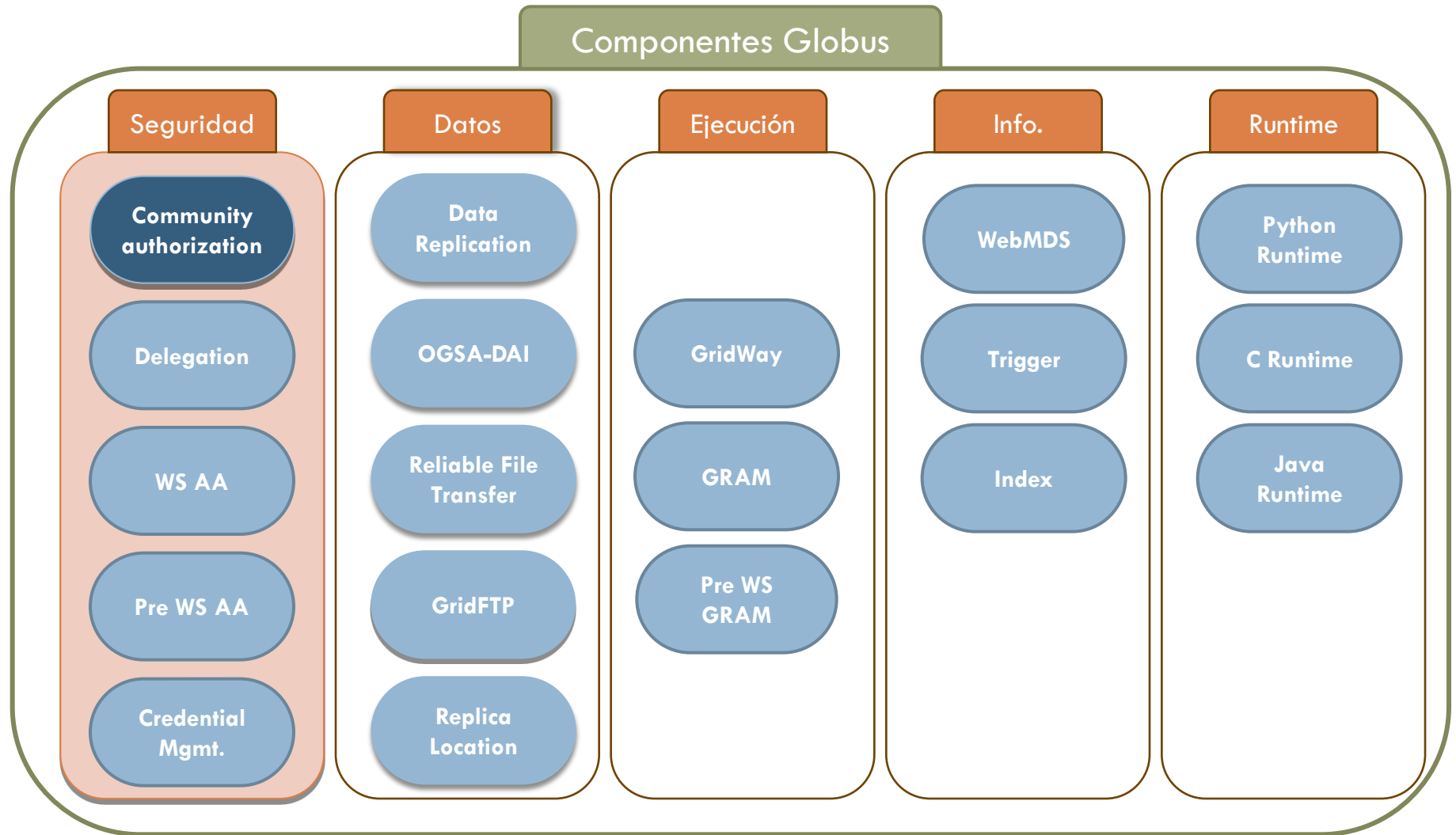
- Servicio de delegación de Globus (I)
 - ▣ Permite delegar derechos a un servicio del mismo contenedor que el servicio de delegación
 - ▣ Acepta una credencial del usuario y proporciona acceso a esa credencial a cualquier servicio autorizado del mismo contenedor
 - ▣ Le da al usuario un *Endpoint Reference (EPR)* que se puede utilizar como identificador de la credencial
 - ▣ El usuario puede refrescar la credencial a través del *EPR*
 - El servicio de delegación lo notifica a cualquier servicio que la esté utilizando

□ Servicio de delegación de Globus (II)

□ Órdenes:

- globus-credential-delegate
 - Permite delegar una credencial
- globus-credential-refresh
 - Permite refrescar una credencial
- globus-delegation-client
 - Cliente de delegación en C. Permite delegar o refrescar
- wsrif-destroy
 - Destruye un recurso (por ejemplo, una credencial)
- wsrif-query
 - Inquiere sobre un documento de propiedades de recurso (por ejemplo, el tiempo de expiración de una credencial)

Seguridad



□ Community Authorization Service (CAS) (I)

□ Objetivo:

- Gestionar la política de accesos de una organización virtual
- Los proveedores de recursos asignan políticas de grado grueso a la comunidad
- La comunidad gestiona las políticas de grado fino

□ Funcionamiento (I)

- Crear un servidor CAS para una comunidad
 - Una persona adquiere una credencial GSI para representar a la comunidad
 - Ejecuta el servidor CAS para esa comunidad con esa credencial de comunidad

□ Community Authorization Service (CAS) (II)

▣ Funcionamiento (II)

- Los proveedores de recursos dan privilegios a esa credencial de la comunidad
 - Usando mapfiles, cuotas de disco, permisos de fichero, etc.
- Los representantes de la comunidad usan el CAS para
 - Gestionar las relaciones de confianza (ej., añadir usuarios y proveedores)
 - Otorgar acceso de grado fino a los usuarios
- Para usar un recurso gestionado por un CAS
 - El usuario hace una petición al servidor CAS
 - Si el servidor da permiso, crea un proxy con el permiso limitado para ese usuario

□ Community Authorization Service (CAS) (III)

▣ Funcionamiento (III)

- El usuario usa la credencial del proxy. El recurso
 - Aplica la política local de acceso a la comunidad
 - La restringe en función de la credencial

▣ Órdenes para el usuario

- `cas-proxy-init -t <tag>`
 - Pide al CAS una credencial proxy del usuario en esa comunidad
- `cas-wrap -t <tag> <programa-grid> <argumentos>`
 - Ejecuta <programa-grid> con sus <argumentos> con el permiso otorgado por las credenciales de ese usuario en esa comunidad
- `cas-wrap -t <tag> grid-proxy-destroy`
 - Destruye la credencial <tag> del usuario en esa comunidad

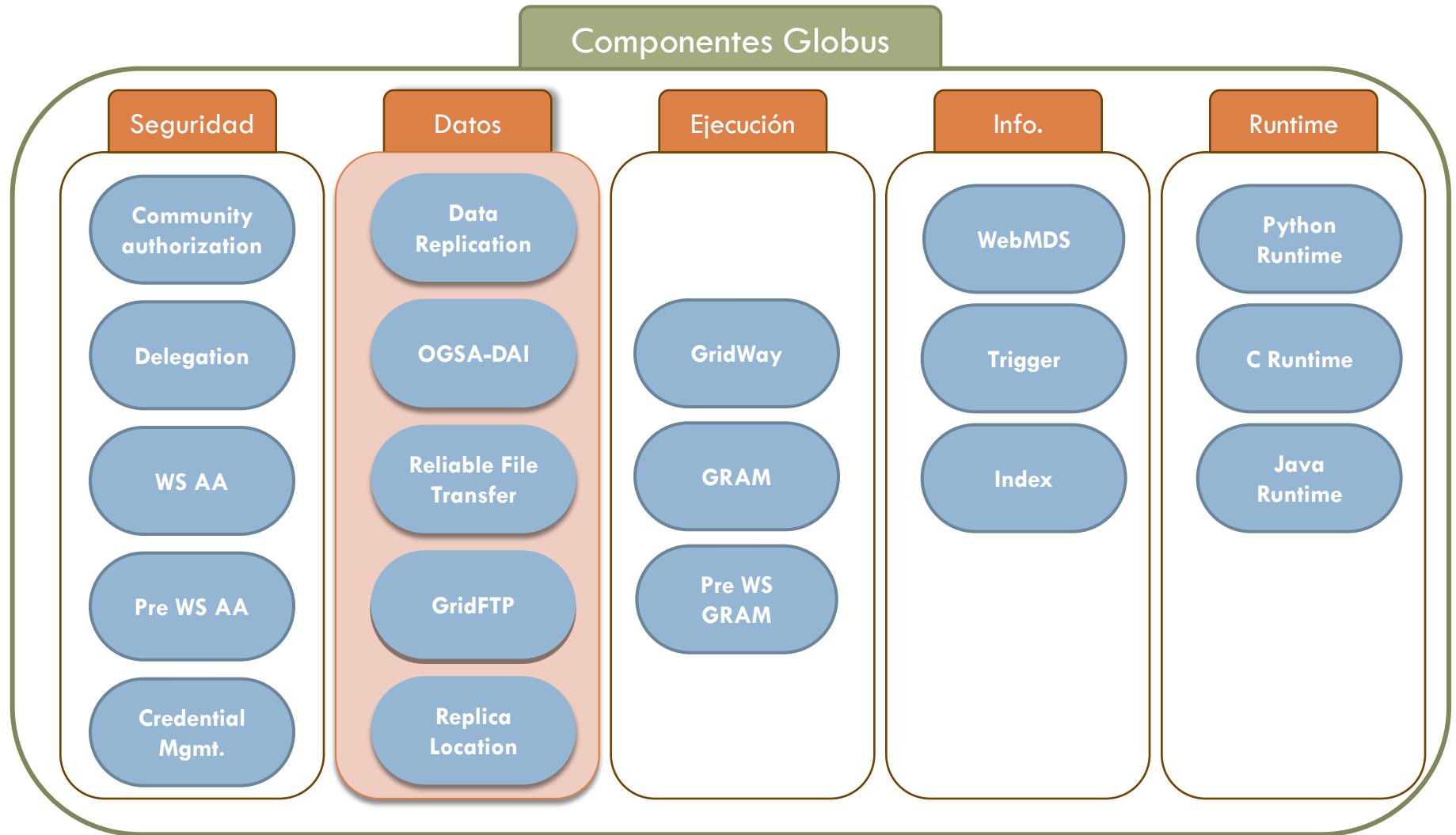
- Community Authorization Service (CAS) (IV)
 - ▣ Órdenes para el administrador
 - cas-enroll, cas-remove
 - Añadir o eliminar un usuario o un recurso a la comunidad
 - cas-action [add | remove]
 - Añadir o quitar acciones a un recurso
 - cas-group-admin
 - Para crear o quitar grupos (de usuarios o recursos) dentro de la comunidad
 - cas-group-add-entry, cas-group-remove-entry
 - Para añadir o quitar usuarios y recursos a grupos
 - cas-rights-admin
 - Para dar o quitar permisos a un recurso



Globus Toolkit

Gestión de datos

Gestión de datos



Gestión de datos

- Servicios de datos proporcionados por Globus (I)
 - ▣ GridFTP: transferencia de información
 - Extiende el protocolo FTP
 - No es un servicio WSRF
 - `globus-url-copy <origen> <destino>`
 - ▣ Reliable File Transfer (RFT): transferencia de información fiable
 - Servicio WSRF
 - Permite crear colas de transferencia fiables
 - Utiliza y extiende GridFTP
 - `rft -file <fichero_EPR> -f <fichero_descr>`

- Servicios de datos proporcionados por Globus (II)
 - ▣ Reliable Location Service (RLS): registro y búsqueda de información replicada
 - ▣ Data replication: herramientas de alto nivel para GridFTP, RFT y RLS
 - ▣ OGSA-DAI (Open Grid Services Architecture - Data Access and Integration): Framework basado en servicios web para flujos de trabajo centrados en datos



Globus Toolkit

Gestión de la ejecución

Gestión de la ejecución

- Gestión de la ejecución
 - ▣ En un Grid se tienen diversos recursos donde ejecutar
 - ▣ Tareas que se deben llevar a cabo para gestionar la ejecución
 - Planificar en qué recurso se ejecuta
 - Ejemplo de planificadores: Unix, Condor, LSF, PBS, SGE...
 - Obtener permisos
 - Tener el ejecutable en el recurso donde se ejecuta
 - Tener acceso a los datos
 - Resolver dependencias entre trabajos
 - Monitorizar
 - Destruir y limpiar

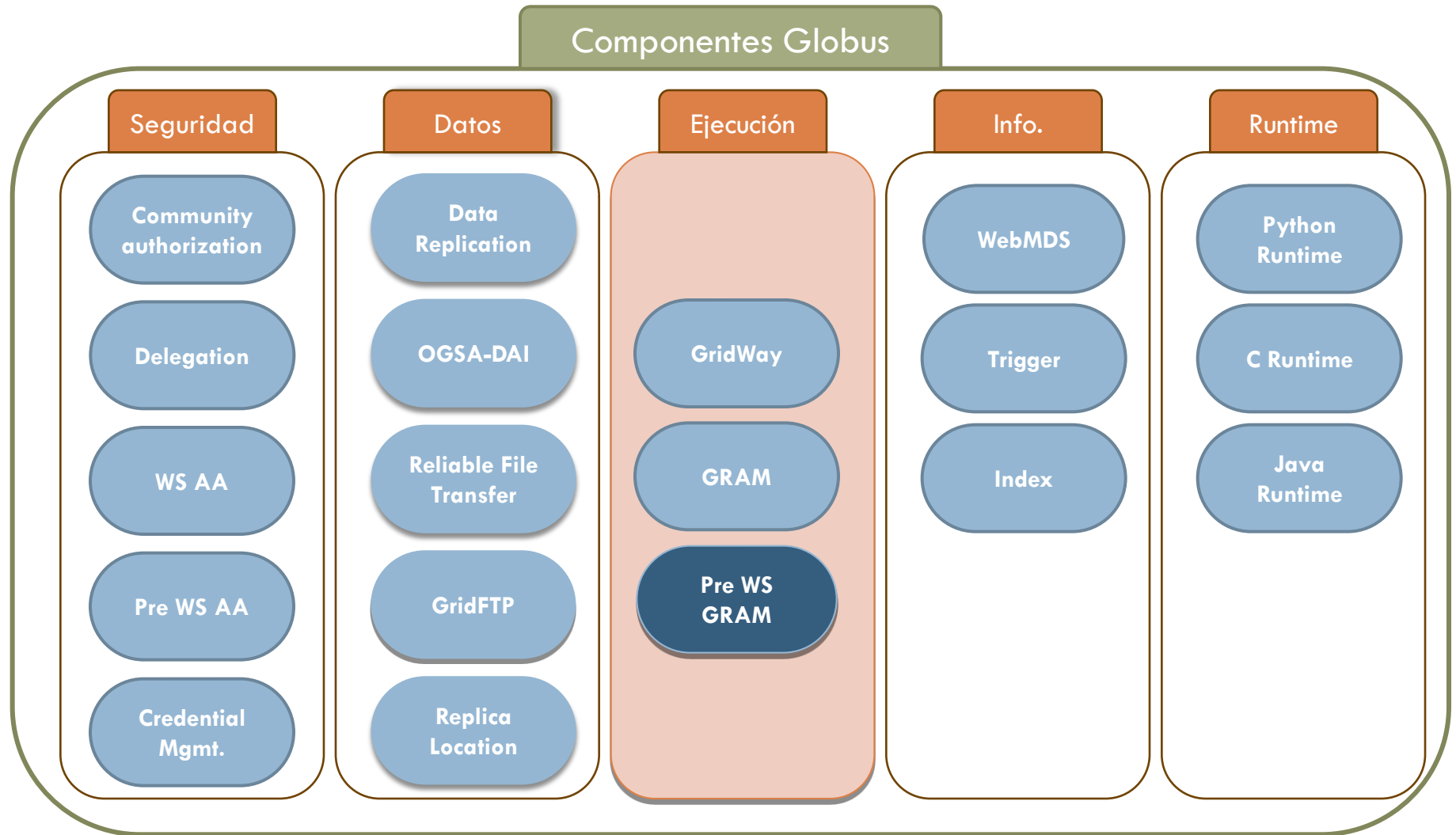
Gestión de la ejecución

- Grid Resource Allocation Manager (GRAM)
 - ▣ Interfaz uniforme para envío y control de trabajos
 - Puesta en escena (*file staging*)
 - Transferencia de ficheros necesarios para la ejecución
 - Fiabilidad
 - Seguridad Grid
 - Disponible en dos versiones:
 - Pre-WS: GRAM2
 - WS: GRAM4
 - ▣ No es un planificador
 - No planifica
 - Es un interfaz hacia los planificadores

Gestión de la ejecución

- GRAM está pensado para trabajos...
 - ▣ Que son programas arbitrarios
 - ▣ Que necesitan monitorización del estado o gestión de credenciales
 - ▣ En los que la organización de ficheros es importante
- Si la aplicación es ligera, con poca entrada/salida, puede ser mejor implementarla como un servicio WSRF

Gestión de la ejecución

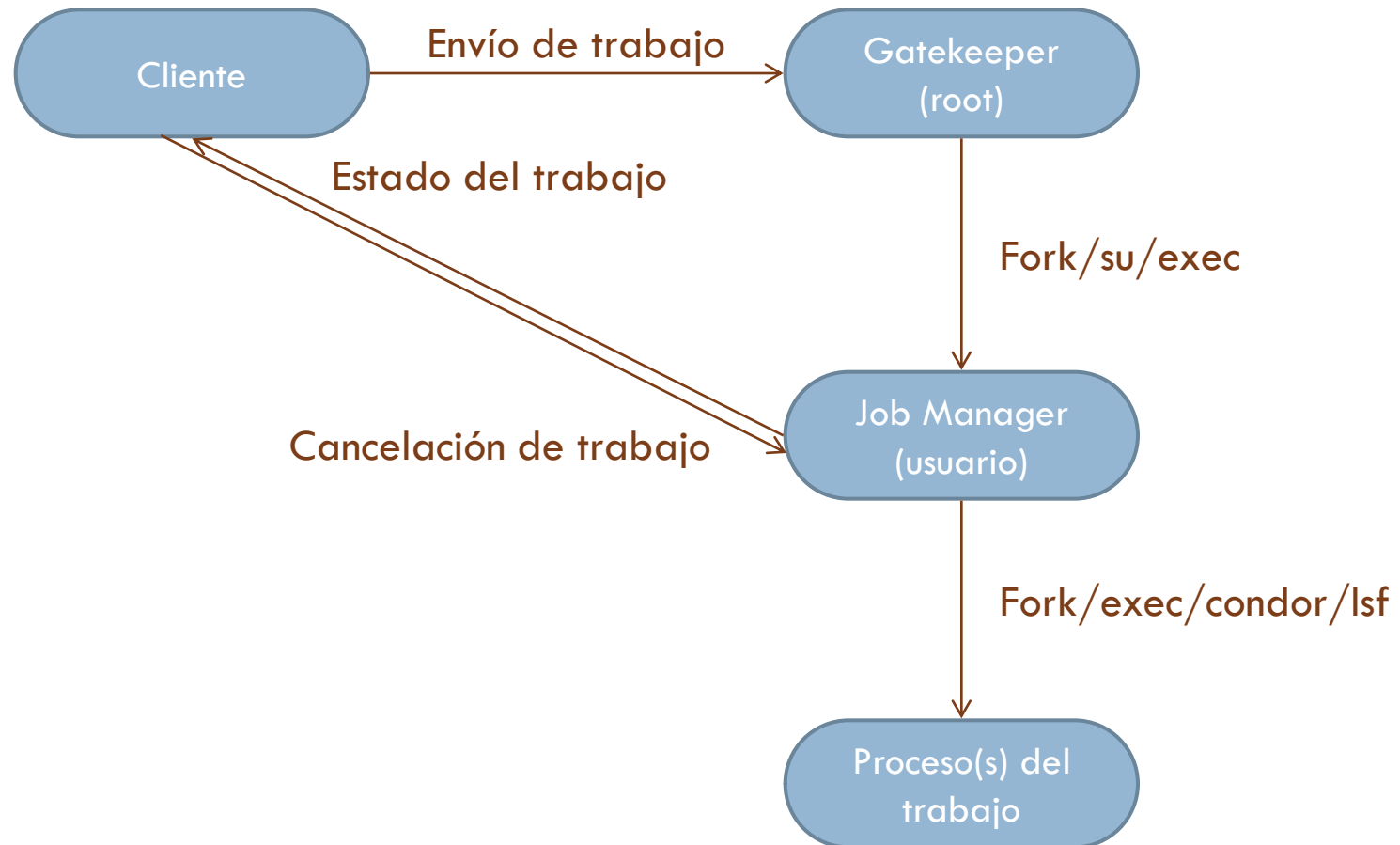


Gestión de la ejecución

- GRAM2 o Pre-WS GRAM
 - ▣ Se incluye en GT4 para dar soporte a sistemas antiguos
 - ▣ La versión de GRAM2 en GT4 añade una característica a la de GT2:
 - Permite escoger con qué usuario ejecutar si una credencial tiene asociados varios usuarios

Gestión de la ejecución

□ Arquitectura de GRAM2 (I)



Gestión de la ejecución

□ Arquitectura de GRAM2 (II)

□ Cliente

- Proceso que usa el API de GRAM

□ Trabajo

- Proceso o conjunto de procesos resultado de una petición de trabajo

□ Petición de trabajo

- Petición con formato RSL que guía:
 - La selección de recursos (cuándo y dónde crear los procesos del trabajo)
 - La creación de procesos de trabajo (qué procesos crear)
 - El control del trabajo (cómo se deberían ejecutar los procesos)

Gestión de la ejecución

□ Arquitectura de GRAM2 (III)

▣ Gatekeeper

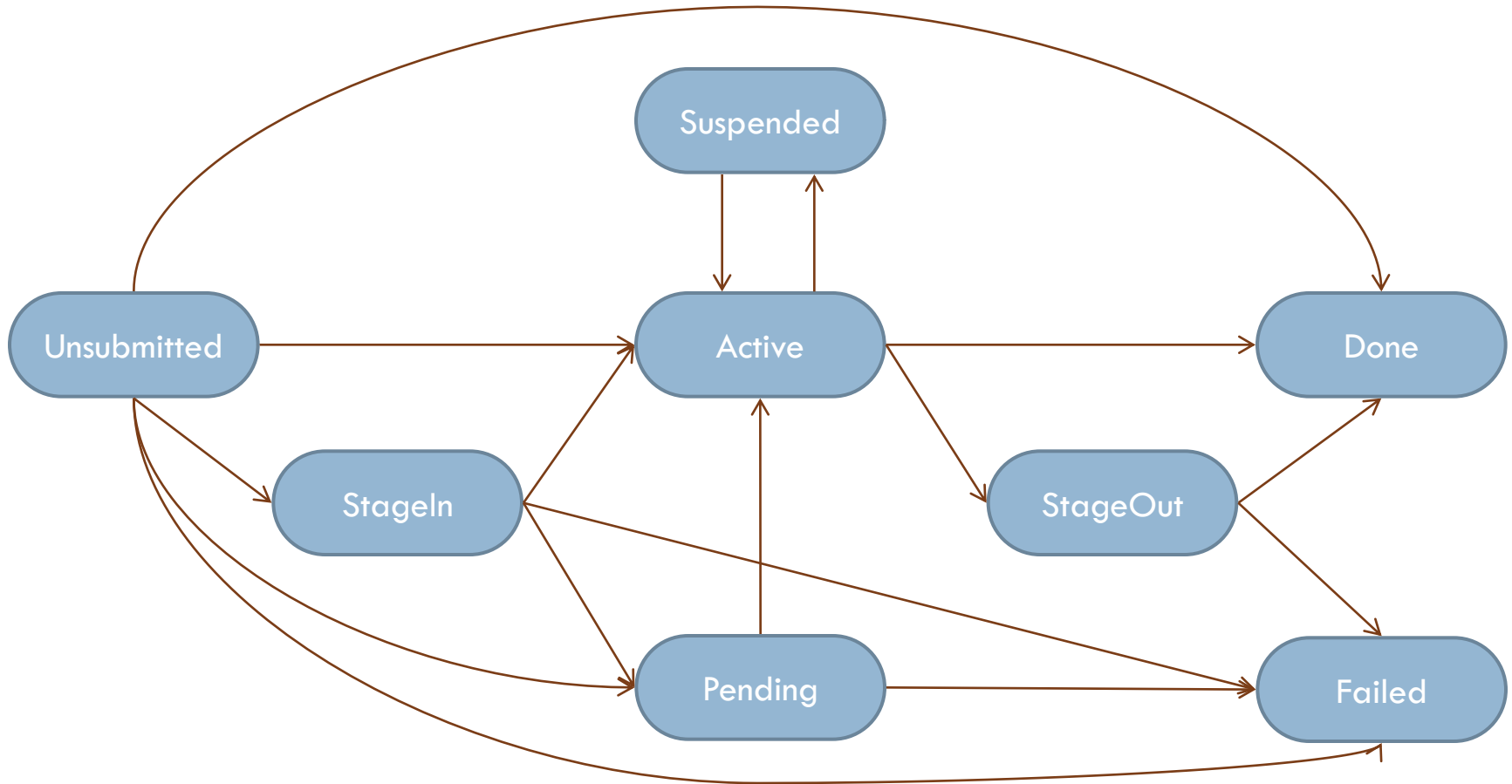
- Servicio del ordenador remoto que comienza la gestión de una petición de trabajo
 - Realiza la autenticación mutua con el cliente
 - Mapea el peticionario a un usuario local
 - Comienza un Job Manager en su máquina
 - Pasa los argumentos al Job Manager

▣ Job Manager

- Hay uno por petición y gestiona la comunicación con el cliente

Gestión de la ejecución

□ Modelo de planificación en GRAM2 (I)



Gestión de la ejecución

- Modelo de planificación en GRAM2 (II)
 - ▣ Unsubmitted: El trabajo todavía no se ha enviado al planificador
 - Se utiliza cuando el Job Manager se para y reinicia antes de haber enviado el trabajo
 - ▣ Stageln: El Job Manager está preparando el fichero ejecutable, la entrada o los datos para el trabajo
 - ▣ Pending: El trabajo ha sido enviado al planificador pero todavía no se le ha asignado un recurso

Gestión de la ejecución

- Modelo de planificación en GRAM2 (III)
 - ▣ Activo: El trabajo tiene todos sus recursos y se está ejecutando
 - ▣ Suspended: El trabajo ha sido detenido temporalmente por el planificador
 - ▣ StageOut: El Job Manager está enviando ficheros de salida de su máquina al almacenamiento remoto
 - ▣ Done: El trabajo se completó con éxito
 - ▣ Failed: El trabajo terminó antes de completarse
 - Por un error
 - Por cancelación del usuario

Gestión de la ejecución

- Resource Specification Language v1.0
 - ▣ Lenguaje de intercambio común para describir recursos
 - ▣ Para GRAM2, no compatible con GRAM4
 - ▣ La sentencia básica es la asignación
 - ▣ Ejemplo:

```
(* esto es un comentario *)  
& (executable = programa )  
  (directory   = /home/nobody )  
  (arguments   = arg1 "arg 2")  
  (count = 1)
```

- GASS (Global Access to Secondary Storage)
 - ▣ Utilizado para la puesta en escena en GRAM2

Gestión de la ejecución

□ Órdenes de GRAM2

▣ globus-job-run

- Ejecuta interactivamente un trabajo

▣ globus-job-submit

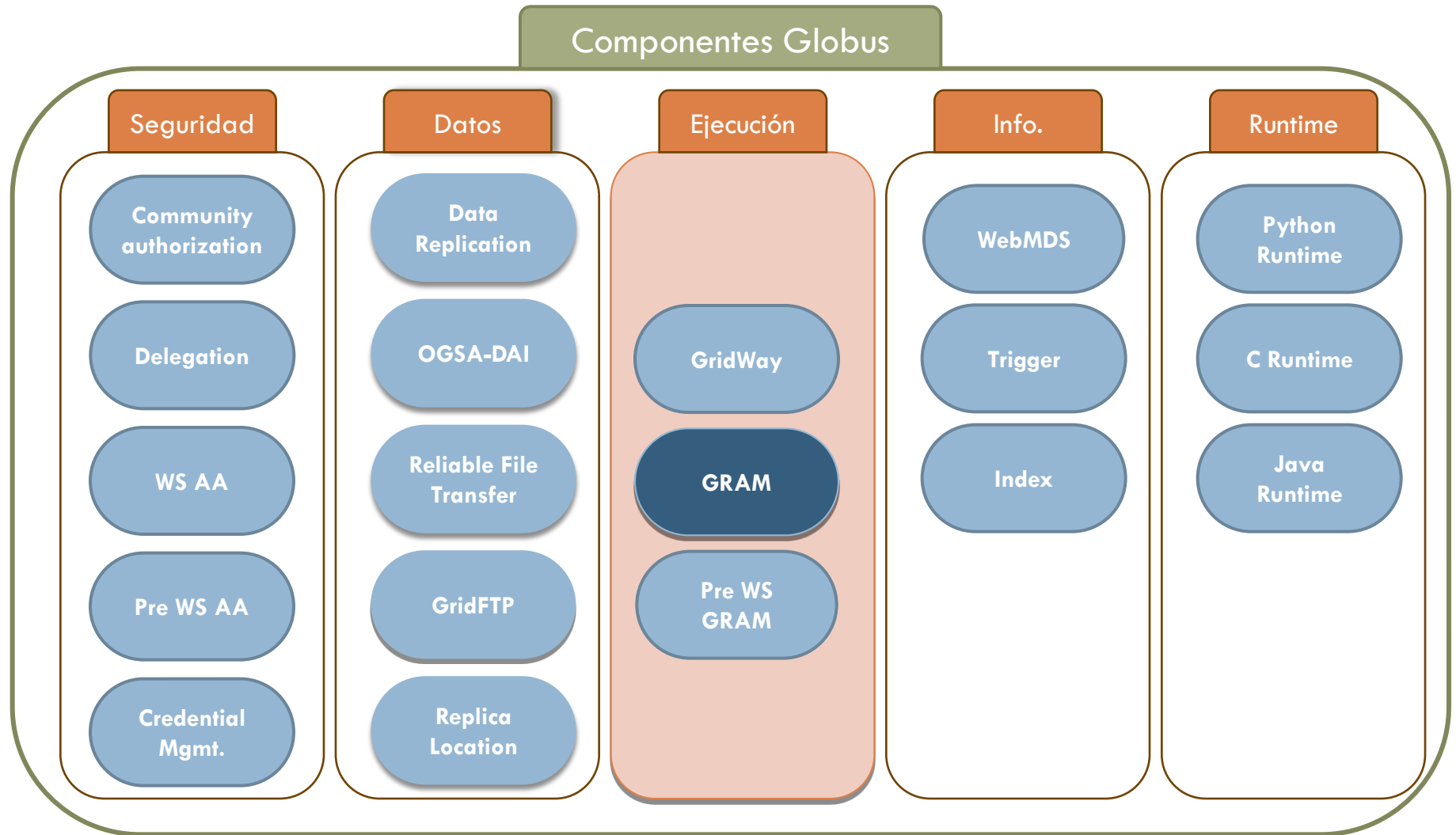
- Ejecuta un trabajo en modo *batch*

▣ globusrun

- Ejecuta trabajos utilizando RSL

- globus-job-run y globus-job-submit son *wrappers* de globusrun

Gestión de la ejecución



Gestión de la ejecución

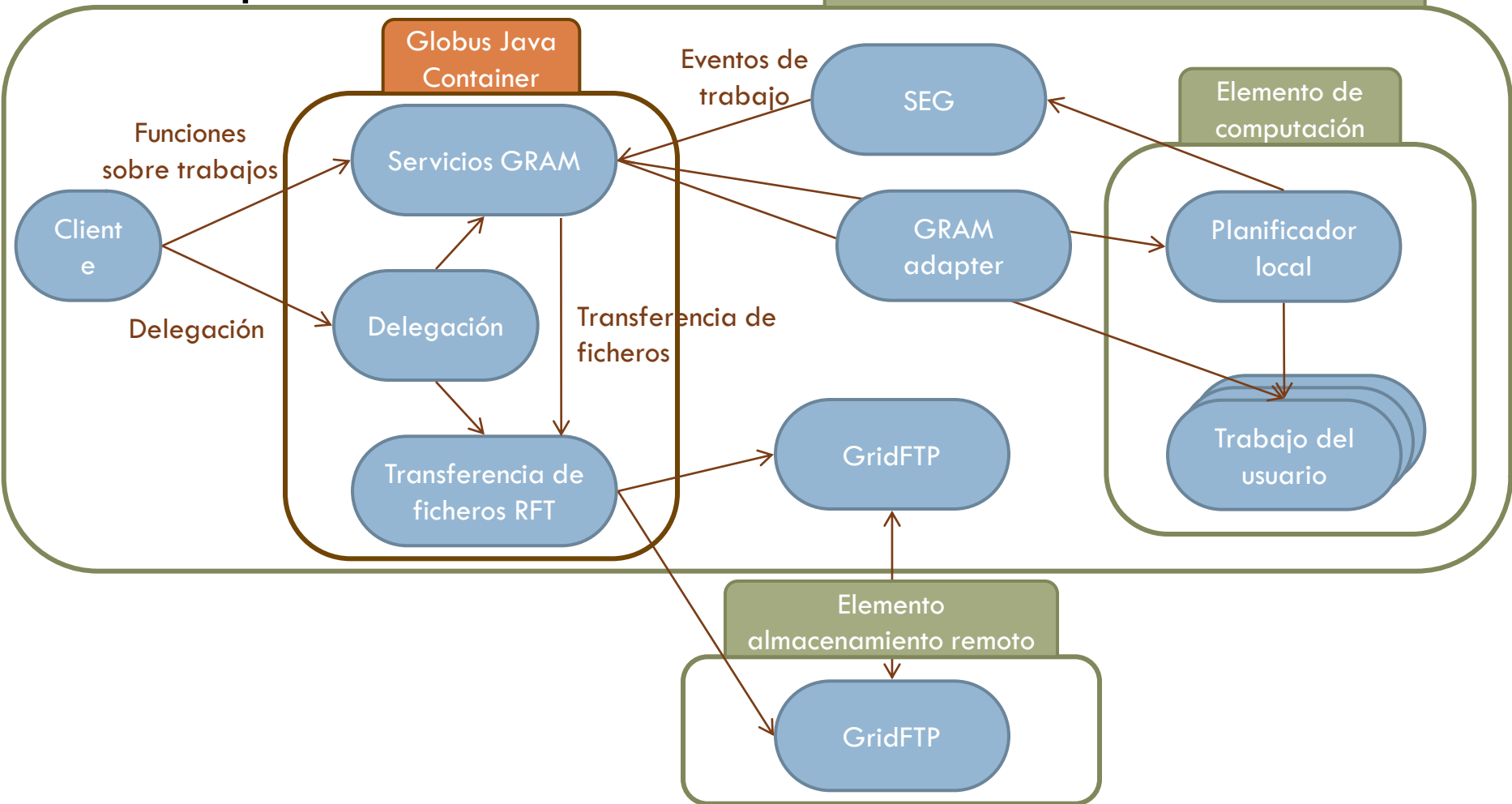
□ GRAM4 o WS GRAM

- ▣ Mejor rendimiento, flexibilidad, estabilidad y escalabilidad
 - GRAM2: ~300 trabajos activos como máximo
 - GRAM4: 32,000 trabajos activos como máximo
- ▣ Mayor sencillez en el camino crítico
 - Usar sólo lo que se necesite en cada caso
- ▣ Gestión de credenciales flexible
 - Caché de credenciales y servicio de delegación
- ▣ Usa GridFTP y RFT para las operaciones de datos
 - Elimina código GASS redundante

Gestión de la ejecución

□ Arquitectura de GRAM4

Máquina(s) de servicio y nodo(s) computación



Gestión de la ejecución

□ Servicios GRAM4

▣ Alojados en el contenedor de WSRF

■ ManagedJob

- Cada trabajo enviado se expone como una instancia de este servicio
- Permite monitorizar y finalizar el trabajo

■ ManagedJobFactory

- Cada elemento de computación se expone como una instancia de este servicio
- Permite crear recursos ManagedJob para ejecutar trabajos

Gestión de la ejecución

- Componentes de GT4 usados por GRAM4
 - ▣ ReliableFileTransfer (RFT)
 - Para realizar la puesta en escena
 - ▣ GridFTP
 - Usado por RFT
 - GRAM4 sólo podrá hacer puesta en escena en nodos que compartan el GridFTP registrado con GRAM4
 - Usado también en la monitorización
 - Permite obtener la salida de cualquier fichero
 - GRAM2 sólo permitía obtener la salida estándar y la de error
 - ▣ Delegation
 - Usado para delegar en los servicios GRAM4 y RFT

Gestión de la ejecución

- Componentes externos usados por GRAM4
 - ▣ Planificador de trabajos local
 - GRAM4 puede ejecutar con `fork()` o llamar a un planificador de trabajos como PBS, LSF, Condor, etc.
 - ▣ Sudo
 - Utilizado para ejecutarse como un usuario local del elemento de computación sin necesidad de ser root
 - En GRAM2, Gatekeeper necesitaba ejecutarse como root
 - Mejora la seguridad

Gestión de la ejecución

- Componentes internos usados por GRAM4
 - ▣ Scheduler Event Generator (SEG)
 - Permite monitorizar trabajos
 - Hay *plug-ins* para distintos planificadores locales
 - ▣ Fork Starter
 - Ejecuta y monitoriza trabajos cuando no hay un planificador local

Gestión de la ejecución

- Visión general del protocolo GRAM4 (I)
 1. Creación
 - Llamando a `ManagedJobFactory::createManagedJob()`
 2. Puesta en escena de credenciales (opcional)
 - Inicializar credenciales para RFT y GridFTP
 3. Credenciales del trabajo (opcional)
 - Se puede crear una credencial para que la use el trabajo durante su ejecución
 4. Refresco de credenciales (opcional)
 - Las credenciales anteriores se pueden refrescar

Gestión de la ejecución

- Visión general del protocolo GRAM4 (II)
 - 5. Mantenimiento de la salida (opcional)
 - Si se desea acceder a los ficheros de salida (que no sean puestos en escena) antes de que se borren
 - 6. Destrucción del trabajo
 - Borrado de ficheros y destrucción del trabajo

Gestión de la ejecución

□ Órdenes de GRAM4

▣ globusrun-ws

■ Parámetros para trabajos sencillos

- -F <maquina>
- -job-command <trabajo>
- -submit
- -streaming (-s)

- Redirige stdout y stderr

■ Para trabajos más complejos, se usa RSL

- -submit -f <ficheroRSL>

Gestión de la ejecución

- Job Description Language (RSL) de GRAM4
 - ▣ Basado en XML
 - ▣ Ejemplo sencillo:

```
<job>
  <executable>/bin/echo</executable>
  <directory>/tmp</directory>
  <argument>12</argument>
  <environment>
    <name>PI</name>
    <value>3.141</value>
  </environment>
  <stdin>/dev/null</stdin>
  <stdout>stdout</stdout>
  <stderr>stderr</stderr>
</job>
```

Gestión de la ejecución

- Resource Specification Language (RSL) de GRAM4
 - ▣ Se pueden utilizar estas variables predefinidas
 - GLOBUS_USER_HOME
 - GLOBUS_USER_NAME
 - GLOBUS_JOB_ID
 - Cada trabajo tiene un UUID
 - GLOBUS_SCRATCH_DIR
 - Directorio alternativo a GLOBUS_USER_HOME, típicamente con más espacio
 - GLOBUS_LOCATION
 - Ruta de la instalación de Globus

Gestión de la ejecución

- Resource Specification Language (RSL) de GRAM4
 - ▣ Para puesta en escena, utiliza etiquetas importadas del esquema de RFT. Ejemplo:

```
<job>
[... ]
<fileStageIn>
  <transfer>
    <sourceUrl>gsiftp://submitting.host:2811/bin/echo</sourceUrl>
    <destinationUrl>file:///${GLOBUS_USER_HOME}/my_echo</destinationUrl>
  </transfer>
</fileStageIn>
[... ]
<fileStageOut>
  <transfer>
    <sourceUrl>file:///${GLOBUS_USER_HOME}/salida</sourceUrl>
    <destinationUrl>gsiftp://submitting.host:2811/tmp/sal</destinationUrl>
  </transfer>
</fileStageOut>
[... ]
</job>
```

Gestión de la ejecución

- Resource Specification Language (RSL) de GRAM4
 - ▣ Limpieza:

```
<job>
[...]  
<fileCleanUp>  
  <deletion>  
    <file>file://${GLOBUS_USER_HOME}/my_echo</file>  
  </deletion>  
</fileCleanUp>  
[...]  
</job>
```

Gestión de la ejecución

- Resource Specification Language (RSL) de GRAM4
 - ▣ Credenciales durante la puesta en escena
 - Los servidores de GridFTP pueden necesitar credenciales distintas a las del servicio GRAM
 - RSL permite especificar credenciales distintas para
 - Ejecución
 - -Jc fich-credencial-trabajo.epr
 - Puesta en escena
 - -Sf fich-credencial-RFT.epr
 - -Tf fich-credencial-GridFTP.epr
 - ▣ Se pueden especificar varios trabajos en un fichero
 - <multijob>

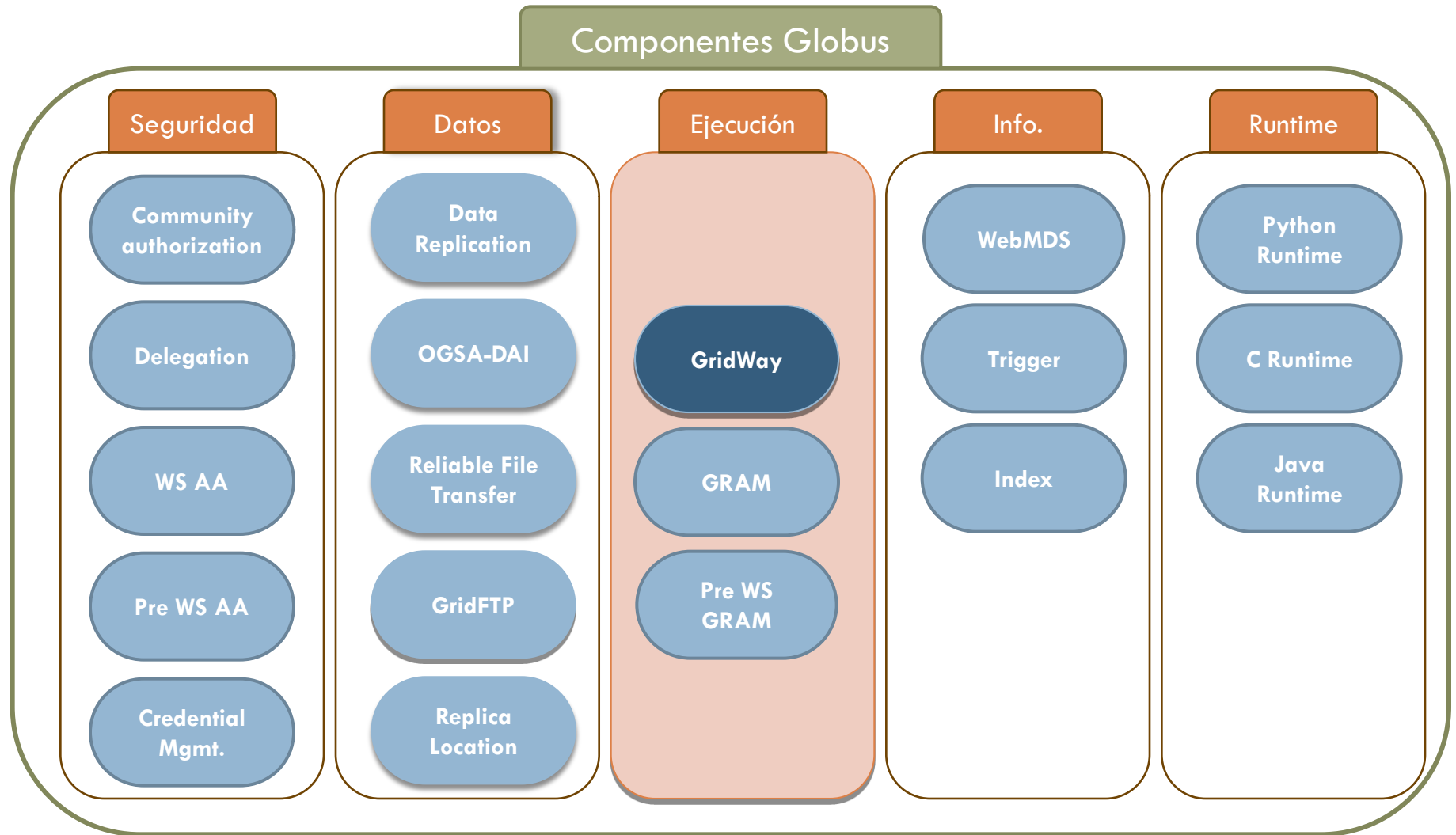
Gestión de la ejecución

- Resource Specification Language (RSL) de GRAM4
 - ▣ Se pueden escoger parámetros del planificador
 - A cuál lanzar
 - A qué proyecto asignar el uso
 - Máximos tiempos de CPU y totales a consumir
 - Mínima y máxima memoria requerida
 - ▣ Se puede escoger bajo qué usuario local ejecutar
 - Si hay varios disponibles en el mapfile

Gestión de la ejecución

- Lanzamientos de trabajo por lotes
 - ▣ Con -batch
 - ▣ El cliente devuelve un EPR del trabajo
 - Se puede redireccionar a fichero con -o
 - ▣ El EPR se puede utilizar para
 - Obtener el estado del trabajo
 - -status
 - -monitor
 - Matar el trabajo
 - -kill

Gestión de la ejecución



Gestión de la ejecución

□ GridWay

▣ Metascheduler

- Planifica entre varias instalaciones de Globus
- Cada instalación puede tener un planificador distinto
 - Planificador = Local Resource Management (LSR)

▣ Desarrollo

- Por la Universidad Complutense de Madrid
- Primer proyecto en pasar de la incubadora a GT4
 - Incluido en octubre de 2007
- Basado en otros servicios de Globus
 - GRAM, MDS, GridFTP, RFT

Gestión de la ejecución

□ Funcionalidades

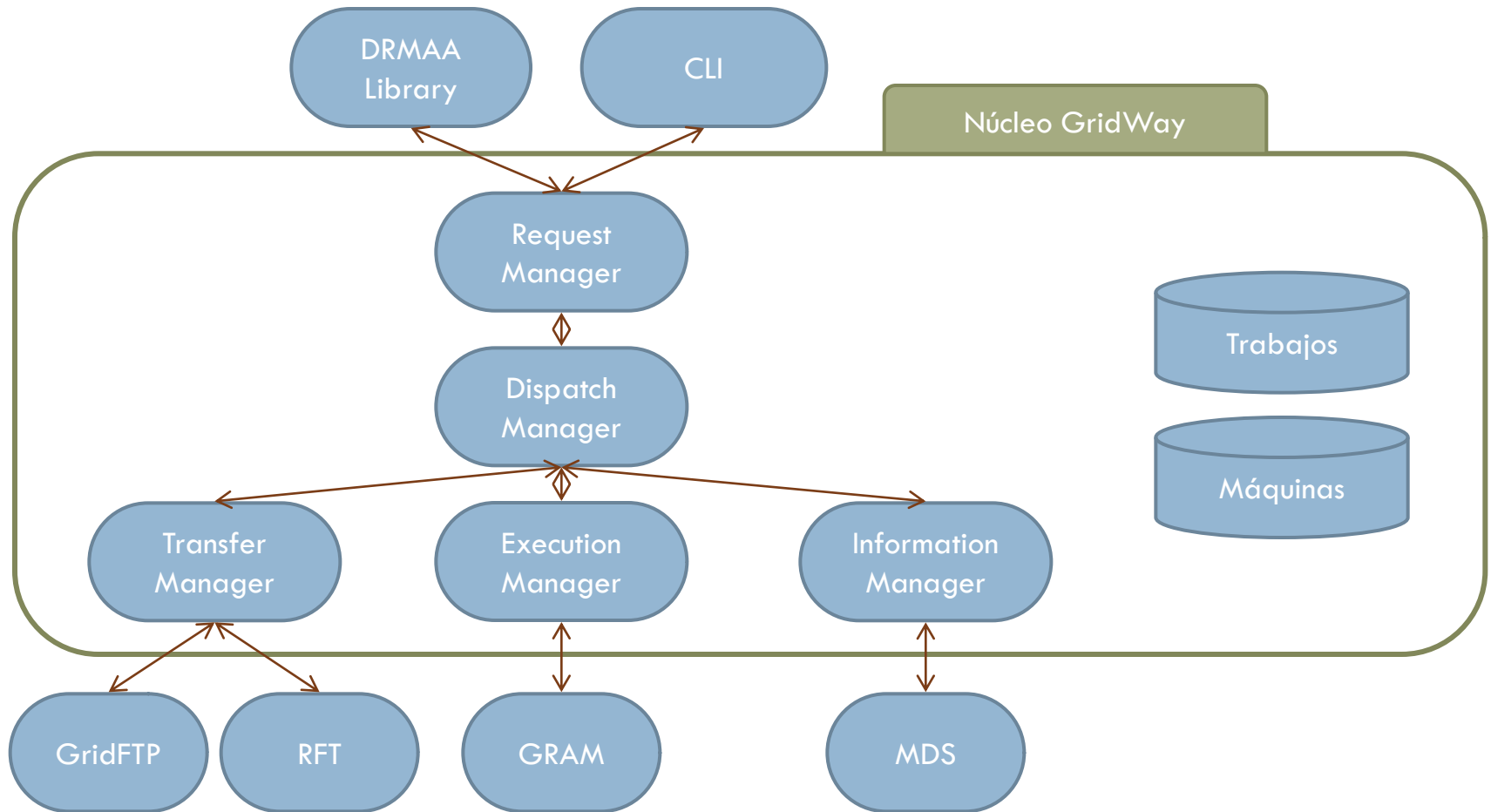
- ▣ Políticas de planificación específicas para Grid
- ▣ Detección de fallos y recuperación
- ▣ Contabilidad
- ▣ Trabajos en array, flujos de trabajo DAG, MPI

□ Interfaz de usuario

- ▣ Estándares del Open Grid Forum (OGF)
 - JSDL (Job Submission Description Language)
 - DRMAA (Distributed Resource Management Application API)
- ▣ Interfaz de línea de comandos

Gestión de la ejecución

□ Arquitectura de GridWay





Globus Toolkit

Servicios de información

Servicios de información

- Monitoring and Discovery System (MDS)
 - ▣ Conjunto de servicios web para monitorizar y descubrir recursos y servicios en Grids
 - La versión Pre WS (MDS2) está obsoleta
 - ▣ Sirve de interfaz estándar de información para otros servicios
 - Monitores de clusters (Ganglia, Nagios...)
 - Servicios (GRAM, RFT, RLS)
 - Planificadores (PBS, LSF, Torque, Condor...)
 - ▣ Interfaces de
 - consulta
 - suscripción

Servicios de información

□ Componentes de MDS4

▣ Proveedores de información

- Todos los servicios WSRF tienen incluida monitorización
- Se pueden usar servicios no WSRF
- Se puede usar cualquier fuente que sepa generar XML

▣ Servicios de alto nivel

- Index Service: agrega datos
- Trigger Service: notifica de cambios en los datos
- Basados en el Aggregator Framework

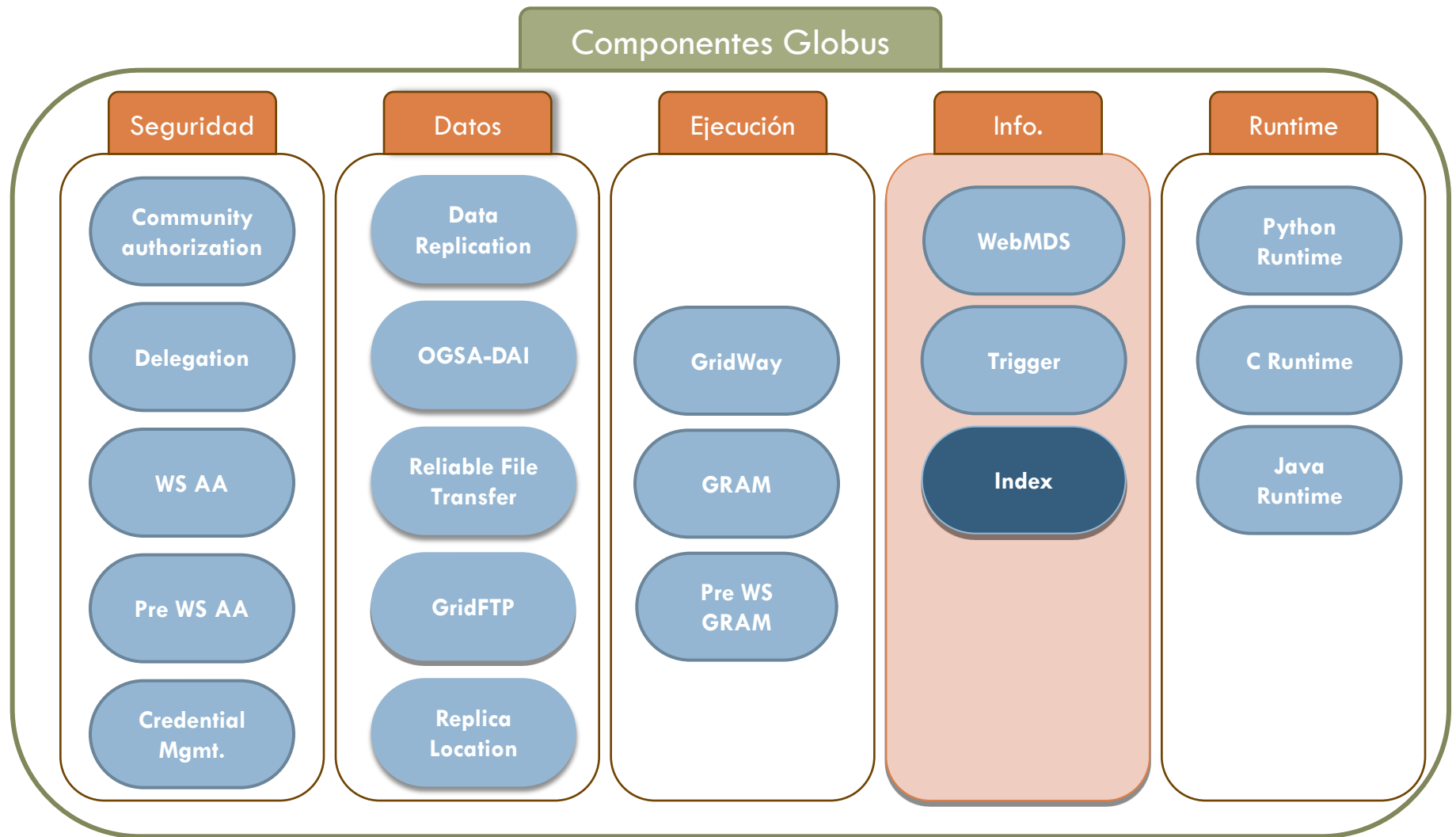
▣ Cliente

- WebMDS

Servicios de información

- Aggregator Framework
 - ▣ Framework para construir servicios de información
 - ▣ Elementos comunes
 - Aggregator source
 - Clase Java que implementa una interfaz para recoger información formateada en XML
 - Mecanismo de configuración
 - Auto-limpieza: un servicio se destruye si después de un tiempo no se refresca
 - ▣ Proveedores de información integrados
 - Haweye (Condor), Ganglia, GRAM4, RFT, CAS

Servicios de información



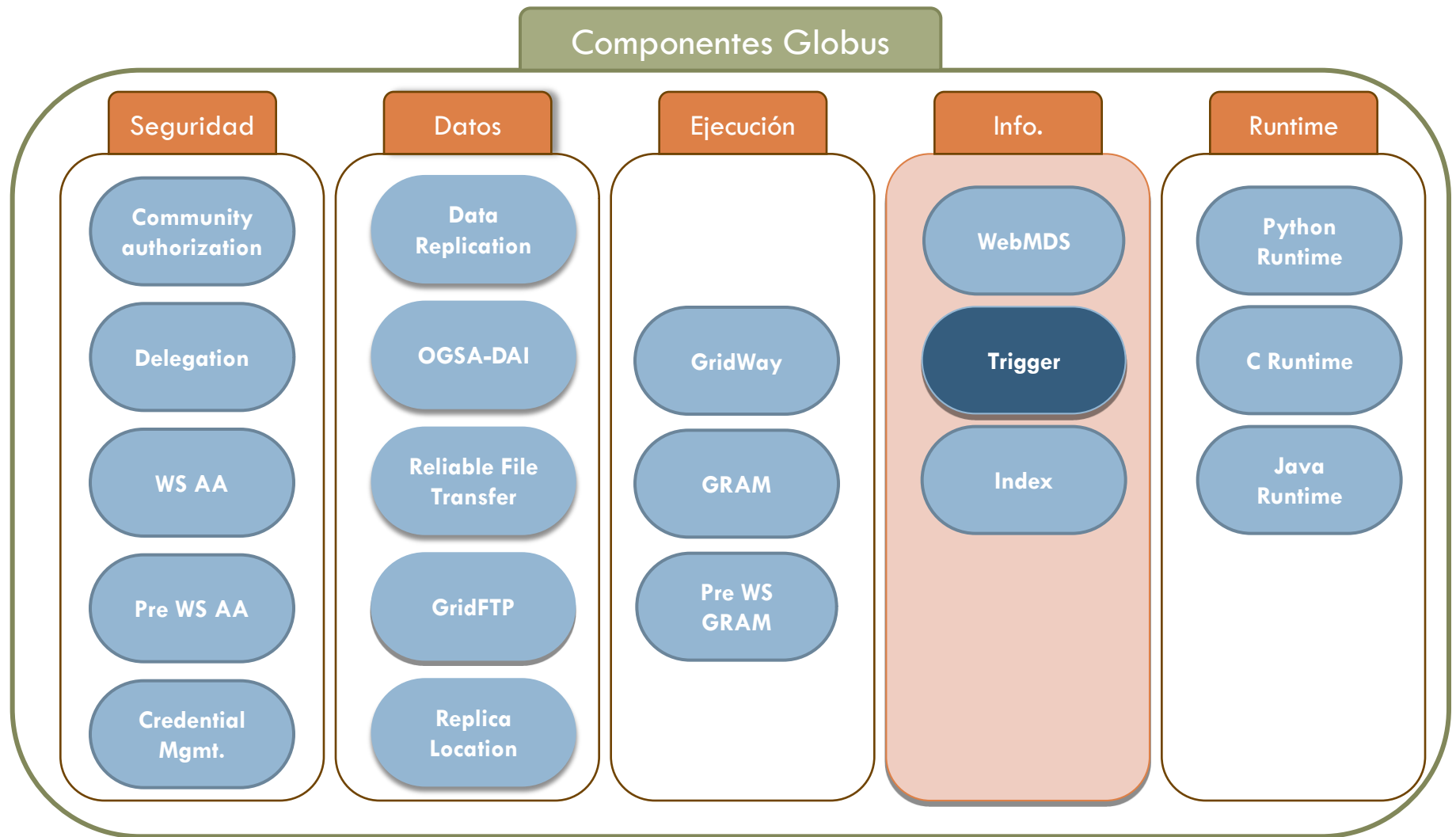
Servicios de información

□ Index Service

- Guarda la información como propiedades de recursos
- Cada contenedor de Globus por defecto tiene un DefaultIndexService que monitoriza los servicios GRAM, RFT y CAS
- Lectura: A través de los interfaces estándar de WSRF
 - Ejemplo desde línea de comandos:

```
wsrf-query -s  
https://localhost:8443/wsrf/services/DefaultIndexService '/*'
```
 - Lo normal es utilizar una interfaz como WebMDS
- Los índices pueden registrarse entre sí
 - Jerarquía multi-raíz de índices para agregar datos

Servicios de información



Servicios de información

□ Trigger Service

▣ Recoge información del Grid

▣ Permite ejecutar programas cuando se cumplen ciertas condiciones

▣ Funcionamiento

■ Seleccionar qué información recoger y cada cuánto

■ Definir *triggers*: condiciones para generar eventos

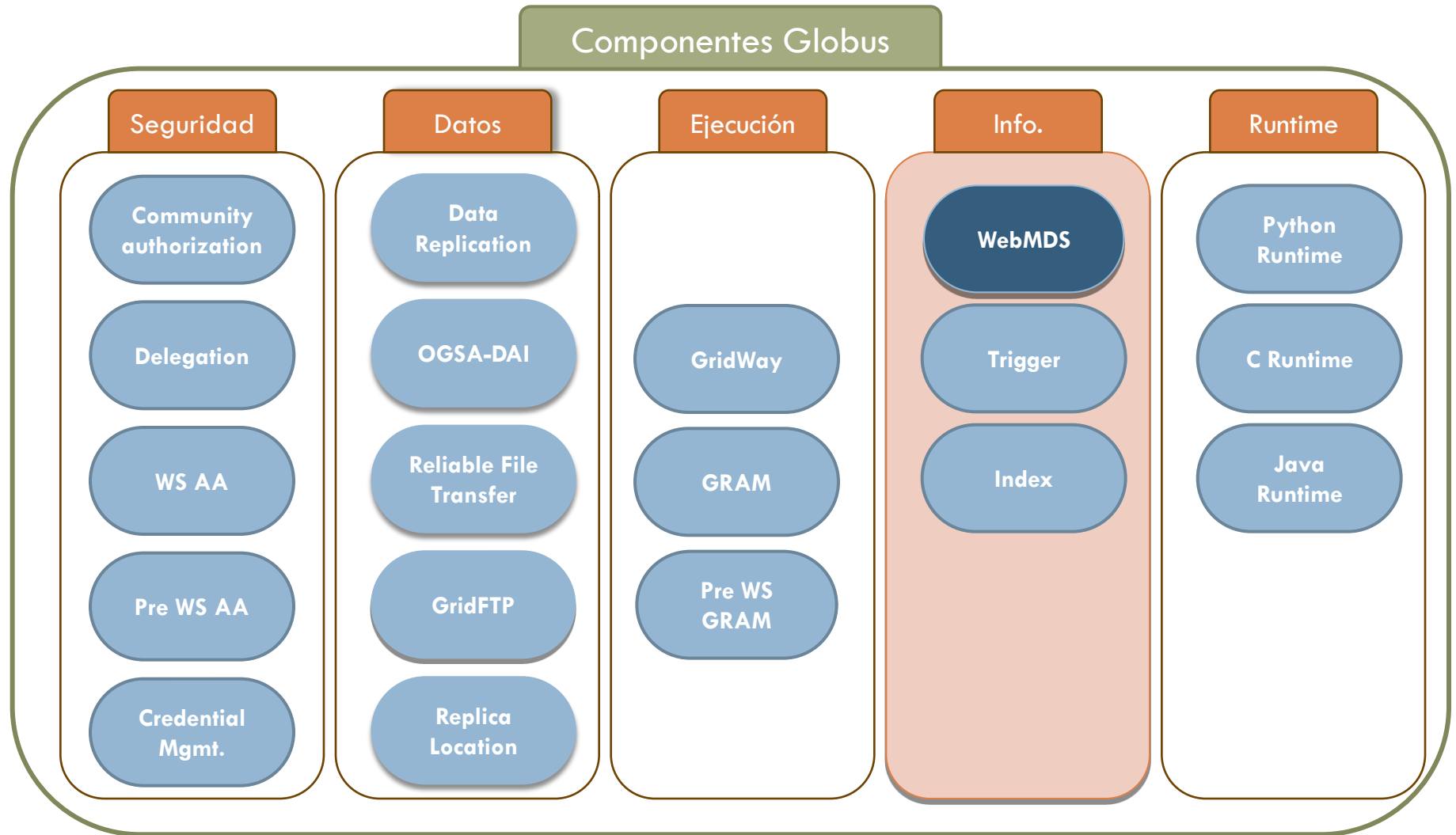
■ Tienen asociada una acción cuando ocurra el evento

■ El servicio recoge la información y ejecuta las acciones si se cumple la condición

▣ Órdenes

■ mds-trigger-create, mds-trigger-view, mds-trigger-edit

Servicios de información



Servicios de información

□ WebMDS

- ▣ Interfaz web para obtener propiedades WSRF
- ▣ *Front-end* amigable para el Index Service
- ▣ Creación de páginas personalizadas usando XSLT

■ Ejemplo:

■ <http://mds.globus.org:8080/webmds>



Globus Toolkit

Instalación y herramientas de alto nivel

Instalación y herramientas de alto nivel

□ Instalación

▣ Plataforma

- Probado en muchos tipos de Linux/Unix
- En Windows, sólo soporte para los componentes Java

▣ Documentación

- <http://www.globus.org/toolkit/docs/>
- Quickstart

▣ Formas de instalación

- Desde los fuentes
- Con VDT

Instalación y herramientas de alto nivel

□ Herramientas de alto nivel

▣ Problema: Globus da una funcionalidad

- muy básica
- y poco amigable para el usuario final

▣ Solución: construir herramientas por encima

■ Ejemplos

- gLite: Desarrolladas en el EGEE
- P-GRADE (Parallel Grid Application and Development Environment): Desarrolladas en Hungarian SuperComputing Grid
- PURSE (Portal-Based User Registration Service): Herramientas de autenticación desarrolladas en el ESG (Earth System Grid)