

Interconnect EGEE and CNGRID e-Infrastructures

Giuseppe Andronico

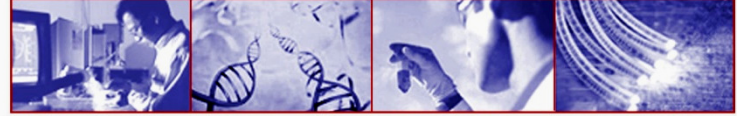
**Interoperability and Interoperation between Europe, India and Asia
Workshop**

Barcelona - Spain, June 2° 2007



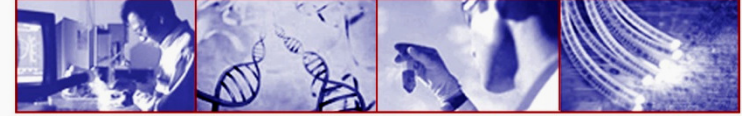
FP6-2004-Infrastructures-6-SSA-026634





Agenda

- *The EUChinaGRID Project*
- *EGEE & CNGRID grid e-infrastructure*
- *gLite-GOS Interoperability & GW approach*
- *Gateway design principles*
- *Batch job level interoperability process*
- *Gilda Testbed in INFN Catania*
- *Data Transfer for small data volumes*
- *Security issues*
- *Future Works*



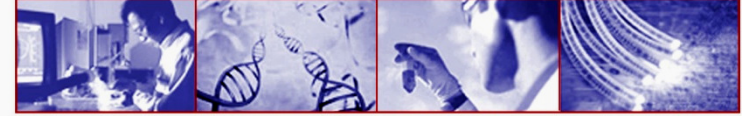
EUChinaGRID Project

- ***Project Banner: Interconnection and Interoperability of Grids between Europe and China***
- ***The EUChinaGRID Project aims to support the interoperability of the Grid infrastructures in Europe and China for the benefit of eScience applications.***
- ***To foster the creation of a intercontinental eScience community:***
 - ***Training people***
 - ***Supporting existing and new applications***
- ***To support interoperable infrastructure for grid operations between Europe (**EGEE**) and China (**CNGRID**)***



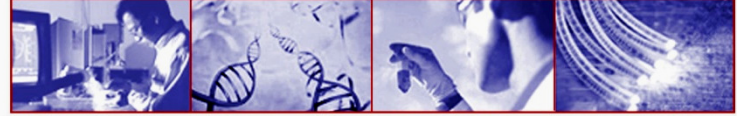
EGEE

- *The Enabling Grids for E-science (EGEE) project draws together over 1000 scientist and engineers from more than 90 institutions in 32 countries.*
- *The **EGEE grid infrastructure** is used at present by applications from many scientific disciplines, as high energy physics, life sciences, geology, computational chemistry and so on.*
- *It is a general purpose infrastructure.*
- *The EGEE grid is composed by over 36000 CPU in addition to 5 Petabytes storage. 30000 concurrent jobs run on the average.*
- ***gLite** is the middleware used in EGEE. It provides a set of service (**SOA architecture**) that allow users to make use of the huge resources of the EGEE infrastructure in a easy way, hiding the complexity.*



CNGRID

- *China National Grid (CNGrid) was developed under the support of China's National Hightech R&D Program.*
- *CNGrid consists of 8 nodes across the country, including 2 major nodes, Computer Network & Information Center of CAS in Beijing and the Shanghai Supercomputer Center in Shanghai.*
- *CNGrid forms an open grid environment for supporting applications in the field of research, resource and environment, service and manufacture.*
- *The aggregated computing capacity of CNGrid exceeds 18 TFlops. The total storage capacity is over 200 TB.*
- *The grid middleware, **CNGrid GOS**, enables the operation of CNGrid and grid applications by supporting resource sharing and cooperative works, which makes CNGrid a testbed for scientific research, technology development, and application demonstration.*



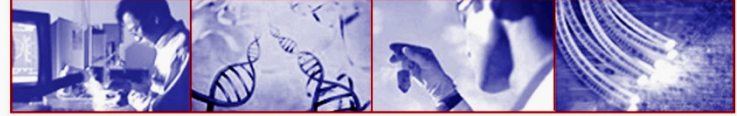
gLite-GOS Interoperability & GW approach

- *gLite and GOS middlewares adopt very different architectures and for this reason the **Gateway approach** to make them interoperable seemed to be the best practicable one.*
- *The role of the Gateway is to **translate requests** coming from one middleware into requests compatible with the other.*
- *Request such as job submission, job control, output retrieval, and data management have to be taken in account.*
- *A method for sharing the knowledge between the 2 middlewares about the status of resource available in their respective infrastructure is the basis for a trivial interoperation prototyping (Information System).*



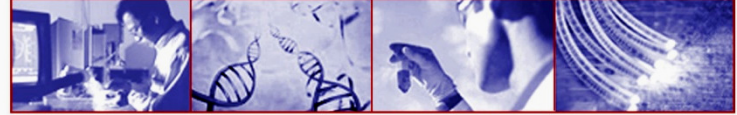
Role of Gateway

- *Gateway is just a logical component, it can be treat as “facade” of underlying grid infrastructure:*
 - *Interface conversion*
 - *Function mapping etc*
- *Gateway should support the following features:*
 - *Transparent to end users of different grid infrastructures*
 - *Easy to extend*
 - *Support massive concurrency and high throughput*
 - *Support standalone deployment or integrated underlying grid middleware etc*
 - *Different instances of the gateway could be configured to connect the infrastructure running GOS and gLite (to guarantee the scalability of the system)*



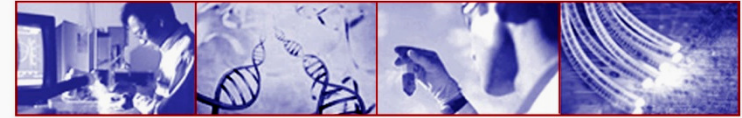
Gateway design principle

- Gateway design depends heavily on **SEDA model** and **IoC model**
- The reasons to choose **SEDA model** and **IoC model** as follows:
 - **SEDA model**
 - **SEDA is short for Staged Event Driven Architecture**
 - **Firstly proposed by Matt Welsh, David Culler, and Eric Brewer, from University of California, Berkeley in their paper **SEDA: An Architecture for Well-Conditioned, Scalable Internet Services****
 - ◆ **Support massive concurrency, high throughput**
 - ◆ **Simplify the construction of well-conditioned Internet services**
 - **In our design, we firstly divide the whole process into several independent basic stages and then compose basic stages into different pipelines for different purpose such as gLite-to-GOS batch job forwarding and so on**
 - **IoC model**
 - **IoC is short for Inversion of Control**
 - ◆ **Provide loose couple among different modules and easy to reuse basic modules**
 - ◆ **Assemble new module easily and quickly**
 - **In our design, HiveMind 1.1 is used as IoC container which is released under LGPL license**

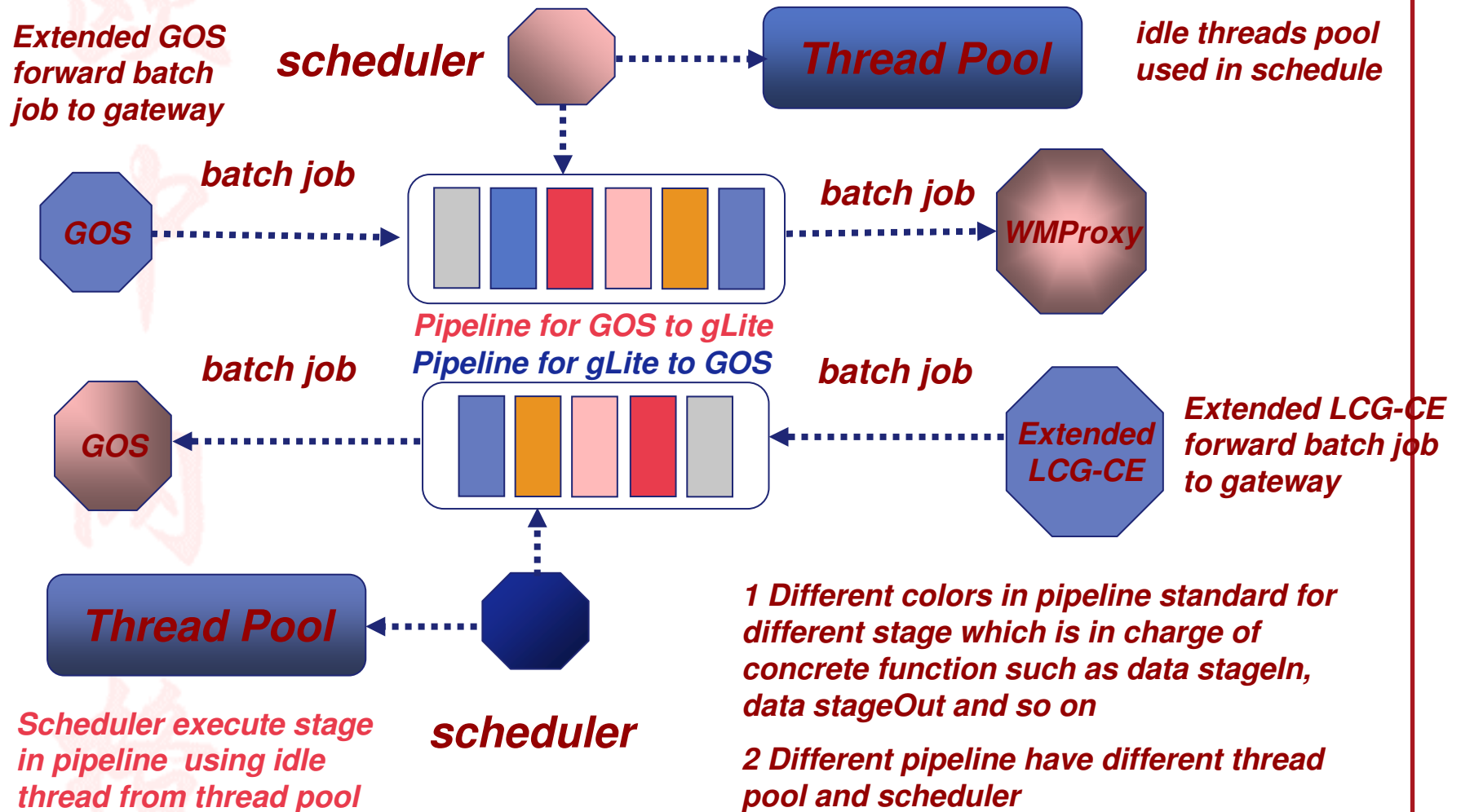


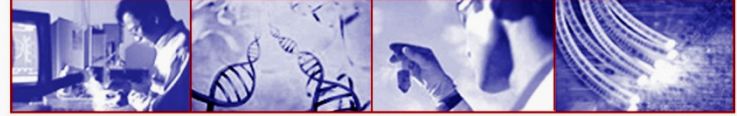
Core components of Gateway

- **Core components of gateway including:**
 - **Pipelines constructed for different purposes**
 - Consists of different basic processing stages
 - Used for different purposes such as forwarding batch job from GOS to gLite and vice versa
 - **Scheduler**
 - Execute processing stages at fixed rate
 - One to one mapping between Pipeline and scheduler
 - **Thread pool**
 - Improves performance
 - One to one mapping between Thread pool and scheduler
- **Different processing stages in same pipeline usually in charge of different concrete functions such as StageIn, StageOut and so on**
- **Detailed description of gateway can be found in the next slide**



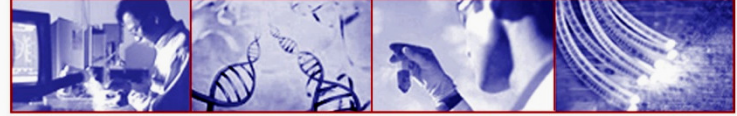
Detailed description of gateway components



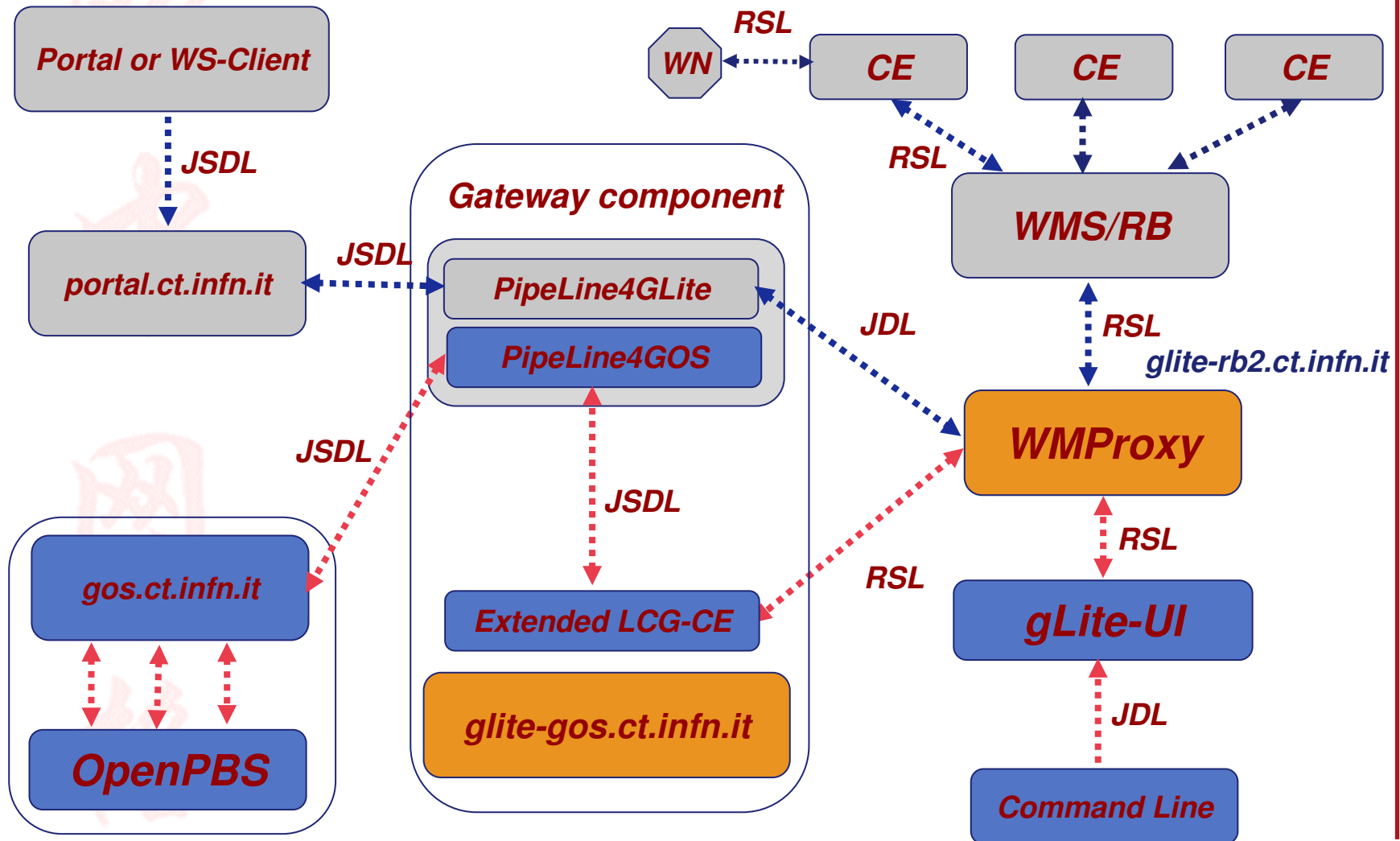


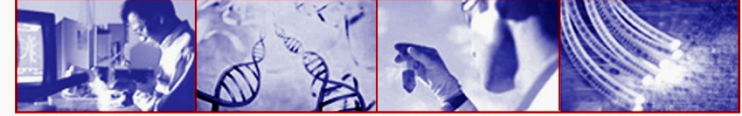
Batch job level interoperability process

- **To implement *batch job level interoperability*, we do the following work:**
 - **The gateway needs to develop a *converter between the Job Description Languages used in gLite (JDL) and GOS (JSDL)***
 - **The common *Web Service interface* can be used to submit jobs**
 - **Extend Job Manager in both gLite and GOS**
 - **Extend gLite LCG-CE Job Manager Framework providing a *special Job Manager for CNGrid* which will forward the batch job to CNGrid for execution**
 - **Provide *Broker plugin for GOS Job Manager* framework (a gLite based RMS System which can interact with the gLite WMPProxy module)**
 - **Sandbox mode data transfer**
 - **A fast approach for cross-domain security scenario**

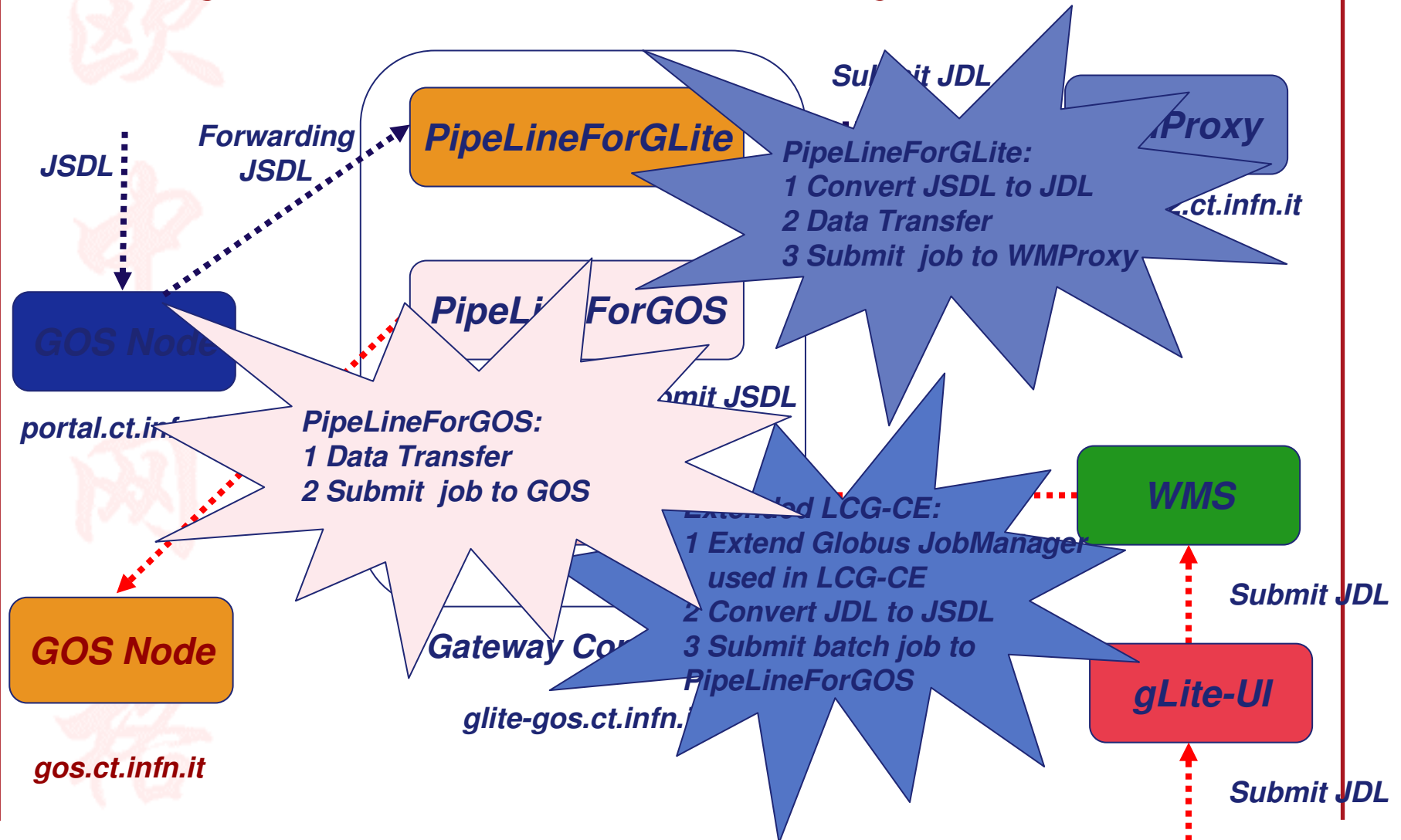


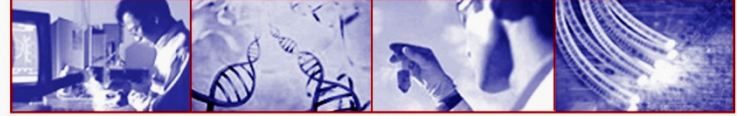
Gilda Testbed in INFN Catania





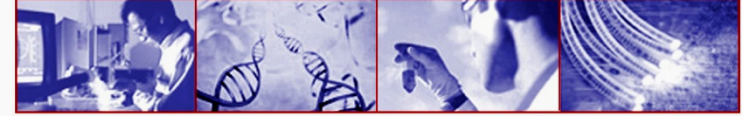
Batch job level interoperability process



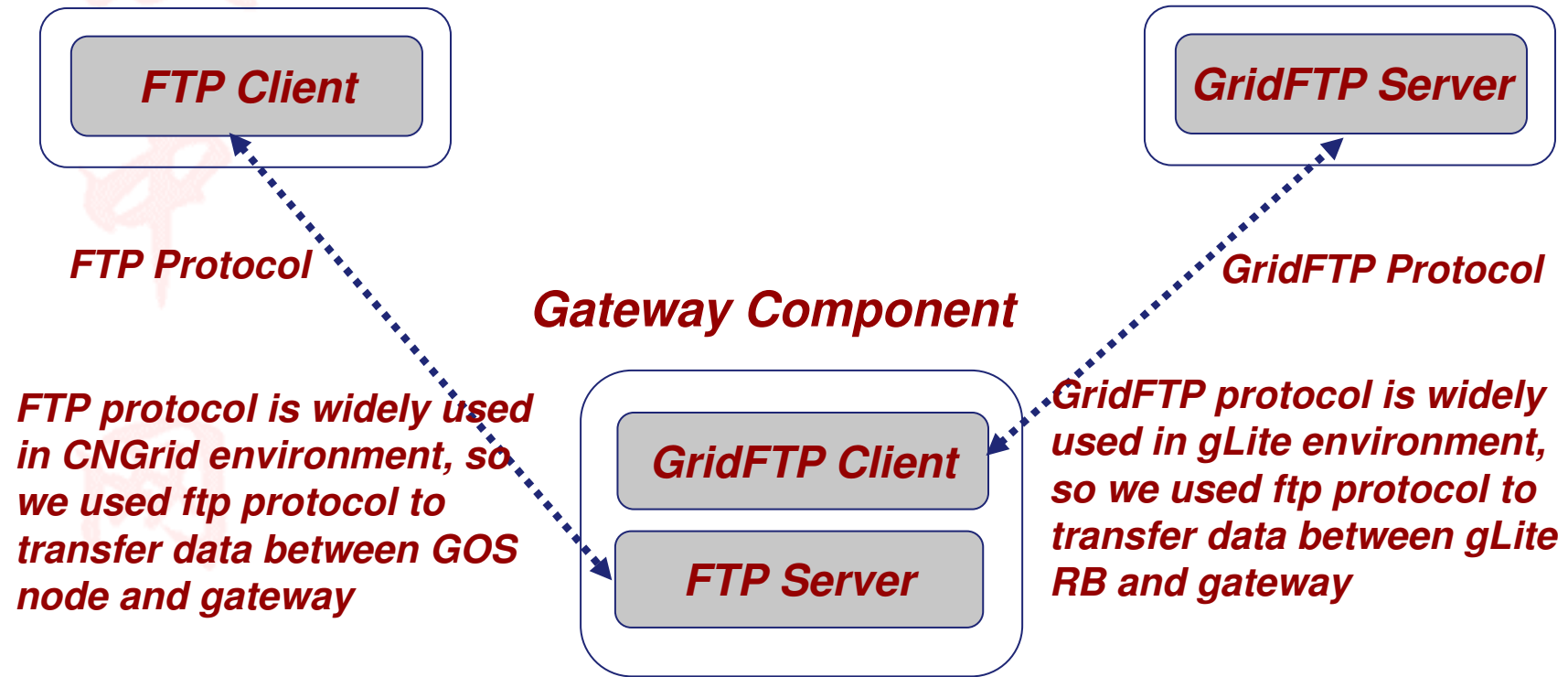


Data Transfer for small data volumes

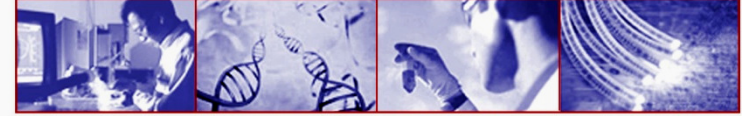
- **Data transfer operations between CNGrid and EGEE uses *sandbox mode***
 - **All the data transfer will pass the gateway**
 - **Suitable for small scale data transfer scenario**
 - **Gateway act as data transfer center and have two different roles at the same time:**
 - **GridFTP client**
 - ◆ **Gateway upload/download necessary data to/from gLite WMS**
 - **FTP server**
 - ◆ **GOS upload/download data to/from gateway component**



Role of gateway component in Data Transfer



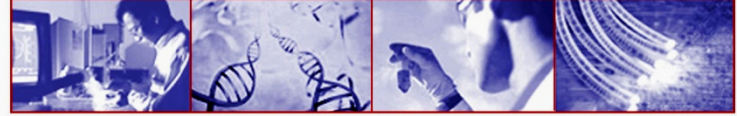
Role of gateway in data transfer scenario



Data Transfer: New development

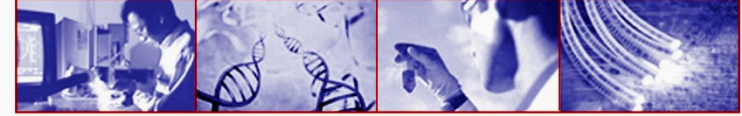
SRM interface deployed on GOS from maintainers

- ▶ Simplify data management interoperability issues
- ▶ A client is available to browse the GOS SE

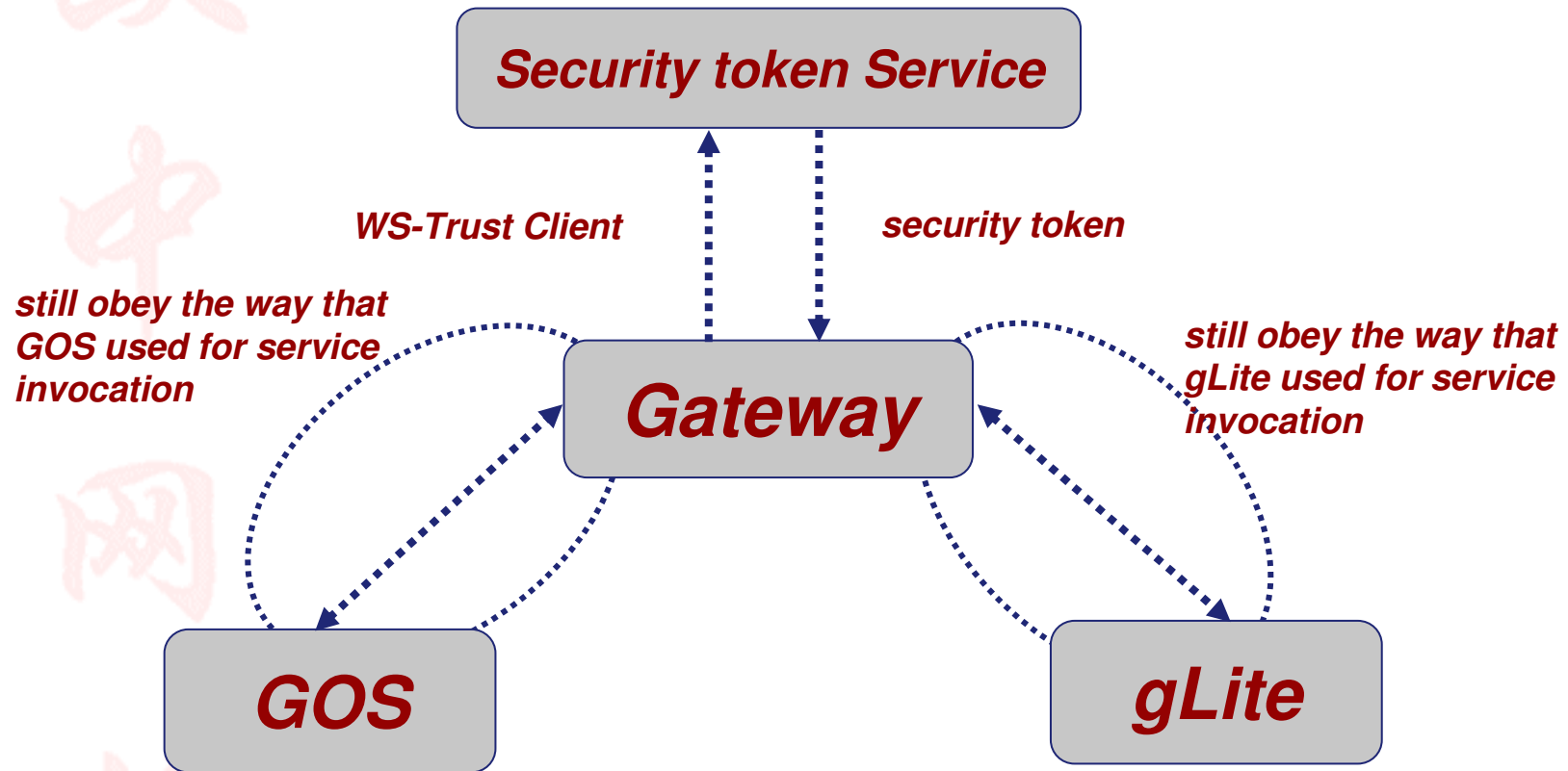


Security issues

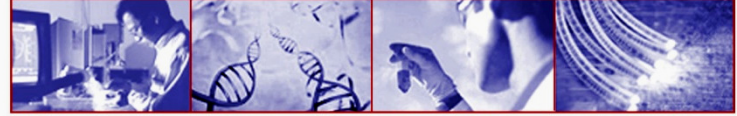
- **Security module used now is just a fast approach for cross-domain security scenario**
 - **Predefine some users in GOS and gLite for interoperability purpose, just a static approach (**Identity Mapping Service**)**
 - **Requests from GOS are submitted to gLite use predefined voms proxy**
 - **Requests from gLite are submitted to GOS use predefined the username**
 - **User Management module is designed to keep mapping relationships**
- **Security token service**
 - **Used to keep, distribute, exchange and verify security token between GOS and gLite and provide dynamical approach**
 - **At present MyProxy Server is setup to store temporary security token, but we plan to replace it with newly developed security token service**



Role of security token service

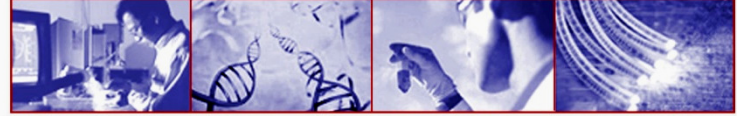


Role of security token service in cross-domain security scenario



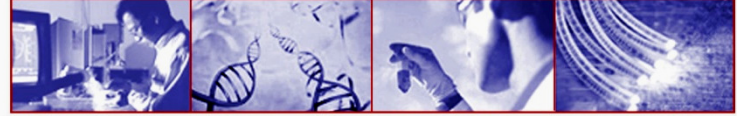
Future Works

- ***Cross-domain Security***
 - ***Security token service based token distribution in cross-domain scenario:***
 - ◆ ***More generic solution for cross-domain security token distribution***
 - ◆ ***Comply with WS-Trust specification***
 - ***Large scale data share and transfer between different grid infrastructures***
 - ***At present we make full use of sandbox mode to support data transfer between GOS and gLite and all the data that transfer between GOS and gLite will pass the gateway, the problems lie in:***
 - ◆ ***Do unnecessary data transfer between the two middlewares.***
 - ◆ ***Cause problems in the case of large scale data transfer (Sandbox volume is less than 20 MB).***
 - ***Adopt SRM Interface in GOS environment to allow large data volumes interoperability.***



Reference materials

- ***SEDA: An Architecture for Well-Conditioned, Scalable Internet Services***, Matt Welsh, David Culler, and Eric Brewer, University of California, Berkeley
- ***Distributed Computation***, technical report of London e-Science Centre (Le-Sc), Imperial College, London
- ***State-of-the-art of Interoperability in EUChinaGrid Project***, EUChinaGrid Project Annual Work Report
- ***HiveMind***, www.apache.org



欧

中

网

格

Thanks for your attentions

Any Questions?