

# Globus et sa mise en œuvre dans le cadre de Grid 5000

Christophe Demarey, INRIA Futurs  
Le 30 octobre 2006

INSTITUT NATIONAL  
DE RECHERCHE  
EN INFORMATIQUE  
ET EN AUTOMATIQUE



## Content

- Introduction to Globus
- Deploying a Globus Grid into Grid 5000
- Demonstration with ParadisEO-G

# I. Introduction to Globus

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



## What is Globus?

The Globus Toolkit is a collection of solutions (tools) to problems that frequently come up when trying to build collaborative distributed applications.

### “Connectivity Layer” Solutions

- Service Management (WS Core)
- Monitoring / Discovery (WS Core)
- Security (GSI and WS-Security)
- Communication (XIO)

### “Resource Layer” Solutions

- Computing / Processing Power (GRAM)
- Data Access / Movement (GridFTP, OGSA-DAI)

### “Collective Layer” Solutions

- Data Management (RLS, DRS, RFT, OGSA-DAI)
- Monitoring / Discovery (Index, Trigger, Archiver services)
- Security (CAS, MyProxy)





# GSI (Grid Security Infrastructure)

The primary motivations behind GSI are:

- the need for secure communication (authenticated and perhaps confidential) between elements of a computational Grid.
- the need to support security across organizational boundaries, thus prohibiting a centrally-managed security system.
- the need to support "single sign-on" for users of the Grid, including delegation of credentials for computations that involve multiple resources and/or sites.

GSI relies on Public-key-based authentication (X.509 certificates).  
Simple Certification Authority included.



## GSI Overview

	Message-level Security w/X.509 Credentials	Message-level Security w/Usernames and Passwords	Transport-level Security w/X.509 Credentials
Authorization	SAML and grid-mapfile	grid-mapfile	SAML and grid-mapfile
Delegation	X.509 Proxy Certificates/ WS-Trust		X.509 Proxy Certificates/ WS-Trust
Authentication	X.509 End Entity Certificates	Username/ Password	X.509 End Entity Certificates
Message Protection	WS-Security WS-SecureConversation	WS-Security	TLS
Message format	SOAP	SOAP	SOAP
	Supported, but slow	Supported, but insecure	Fastest, so default



# RFT (Reliable File Transfer)

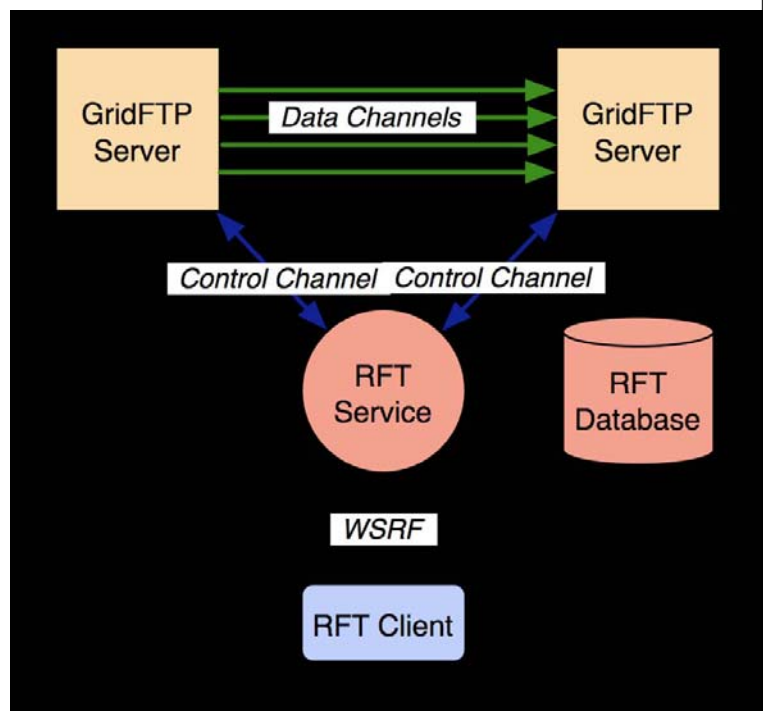
A WSRF service for queuing file transfer requests

- Server-to-server transfers
- Checkpointing for restarts
- Database back-end for failovers

Allows clients to request transfers and then “disappear”

- No need to manage the transfer
- Status monitoring available if desired

Use of GridFTP servers



# GRAM (Globus Resource Allocation Manager) : Execution management

Common WS interface to schedulers

- Unix, Condor, LSF, PBS, SGE, ...

GRAM is *not* a scheduler.

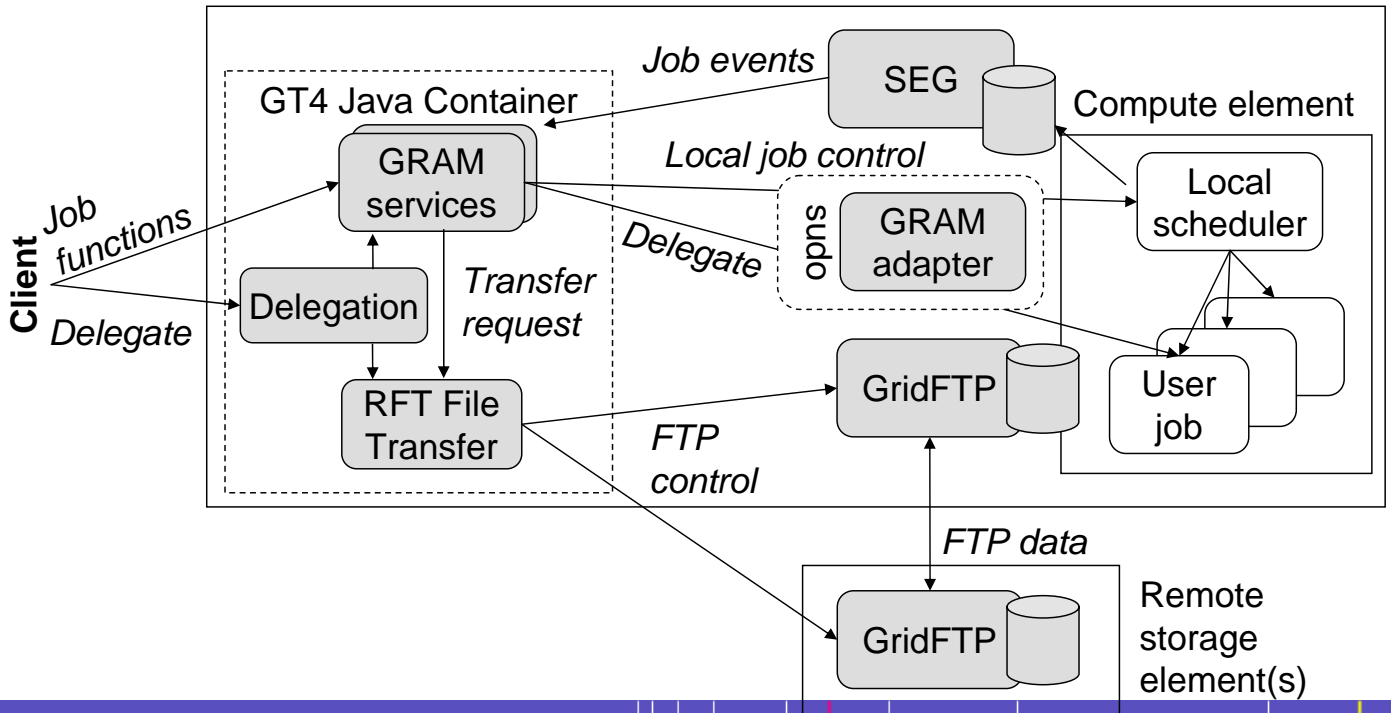
- No scheduling
- No metascheduling/brokering
- Often used as a front-end to schedulers, and often used to simplify metaschedulers/brokers

More generally: interface for process execution management

- Lay down execution environment
- Stage data (GridFTP & RFT used for data operations)
- Monitor & manage lifecycle
- Kill it, clean up

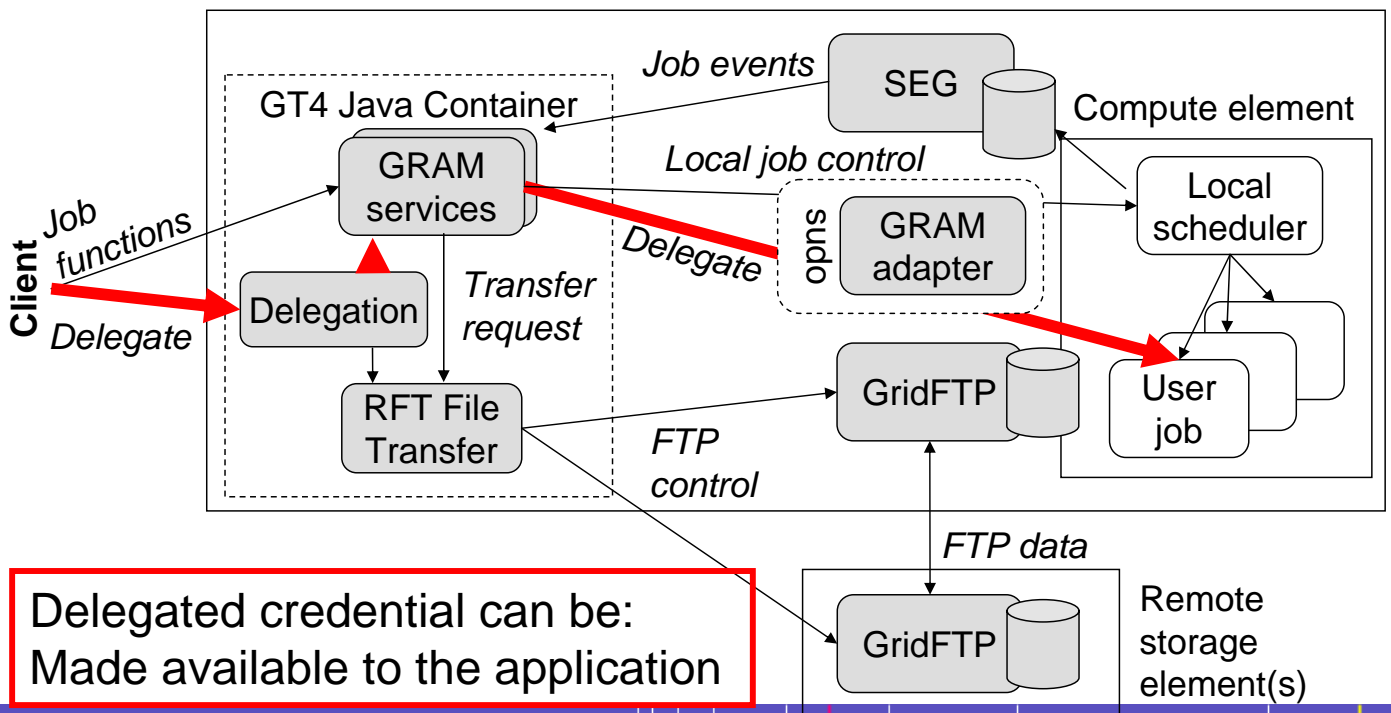
# GT4 WS GRAM Architecture

Service host(s) and compute element(s)



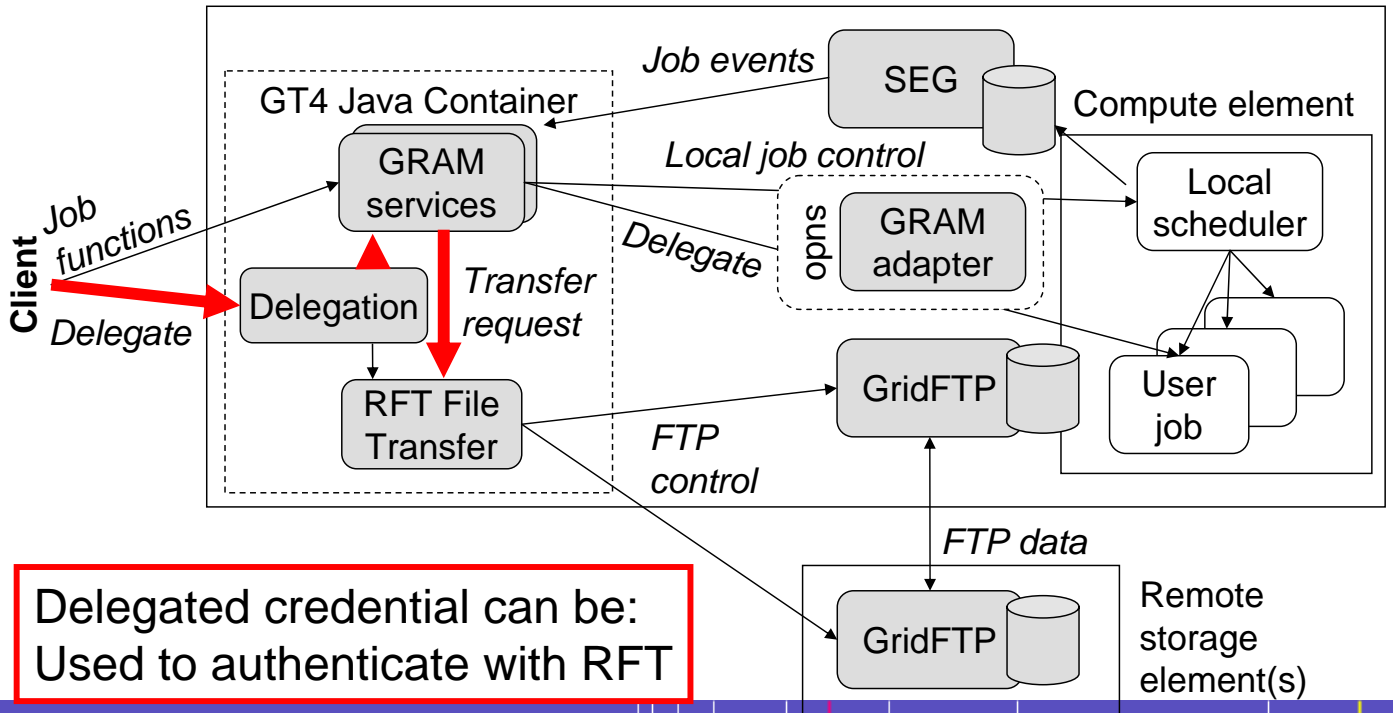
# GT4 WS GRAM Architecture

Service host(s) and compute element(s)



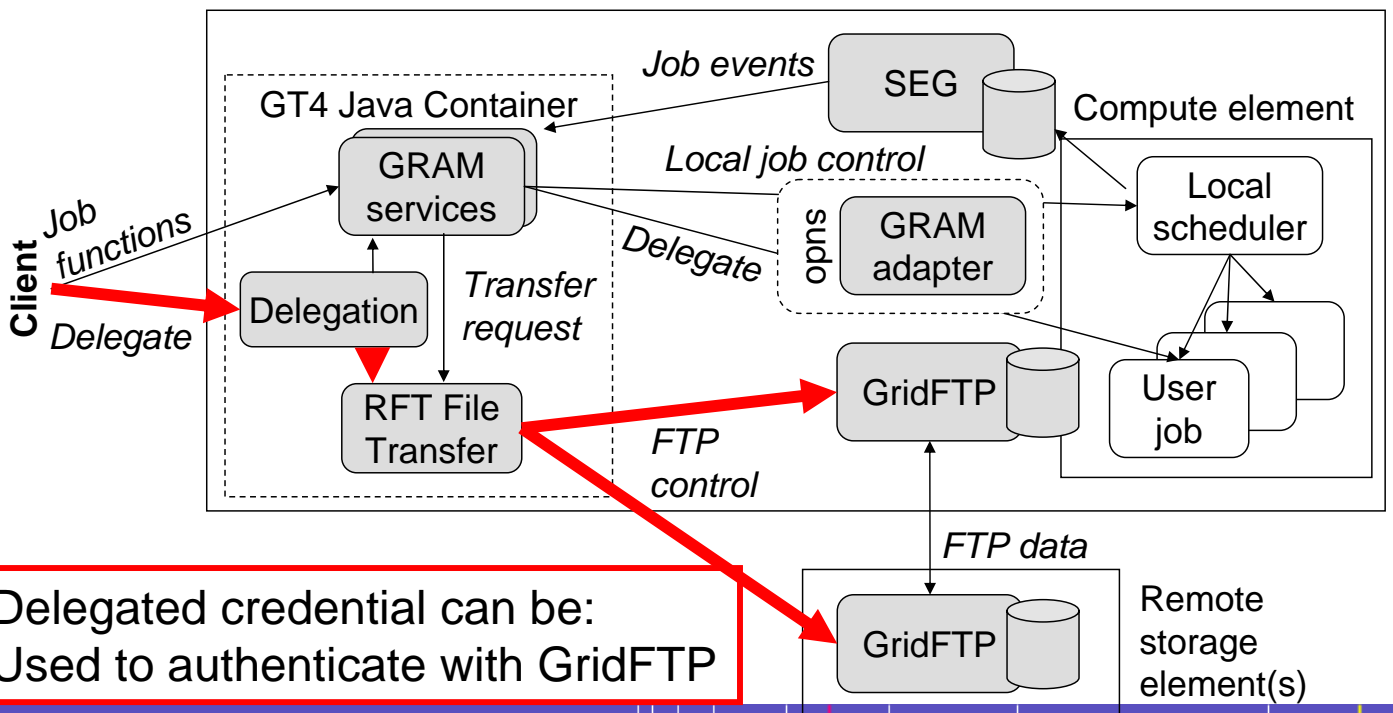
# GT4 WS GRAM Architecture

Service host(s) and compute element(s)



# GT4 WS GRAM Architecture

Service host(s) and compute element(s)



# Further look at Globus

- <http://www.globus.org>
- <http://www.globus.org/toolkit/presentations/>

## II. Deploying a Globus Grid into Grid 5000

# Needs

- A volatile Globus grid
- Must be deployed effortlessly by users
  - don't worry about tools configuration
  - ready to run tools for common use
  - don't worry about authentication / authorization configuration
- Deploy a Globus grid on a subset of the whole grid
- Deploy several Globus grids at the same time in Grid 5000
  - interconnected or not
  - deployed by the same or different user(s)
- Use of local schedulers
- MPI support
- Batch mode support

# Our approach

Slightly different than the way Globus toolkit was built for!

GOAL : build a deployable system image with « Globus inside »

Methodology :

- Install and configure Globus on 1 node
- Duplicate the first installation on a second node
- Adapt configuration for the second node
  - gives elements that must be automatically tuned
- Automatization

The adaptation step highlighted the need of 2 kinds of Globus node :

- One master : used for centralized services (Simple Certification Authority, Monitoring and Discovery Service)
- and slaves

# Self configurable image

## Problem :

Image provided to Grid 5000 users must be ready to use but majority of tools are user / host – dependant.

## Solution :

Install and configure Globus Tools on the fly with the help of scripts.

This method is used for following tools :

- Security
- Globus Gatekeeper (GT2)
- GridFTP / RFT (and its database)
- GRAM
- WebMDS
- MyProxyServer

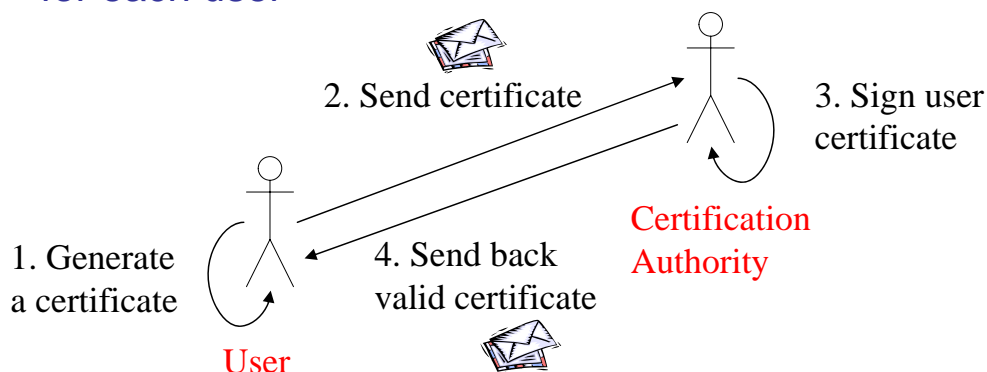
# GSI configuration

We choose the simple-ca package

## Problem :

We must generate automatically signed certificates :

- for each node (host) and Globus container (that host Globus WS)
- for each user



# GSI configuration

## Solution :

Use ssh communications instead of mail

To allow communications between globus nodes an ssh key has been generated in the Globus Image.

The corresponding public key has been added to the `~/.ssh/authorized_keys` file of root and globus user accounts.

This solution is adaptable to use an external certification authority.



## Provided Environment

An image that can be installed on Grid5000 nodes with kadeploy.

Shell-scripts are added to automate the install and configuration process. Some install scripts are also provided in order to allow administrators, but also simple users, to build a globus image based on their own environment.

The deployment architecture is as follows :

- Master node
  - contains the Certification Authority
  - contains the Monitoring and Discovery System (MDS)
  - has users credentials

- Slave nodes

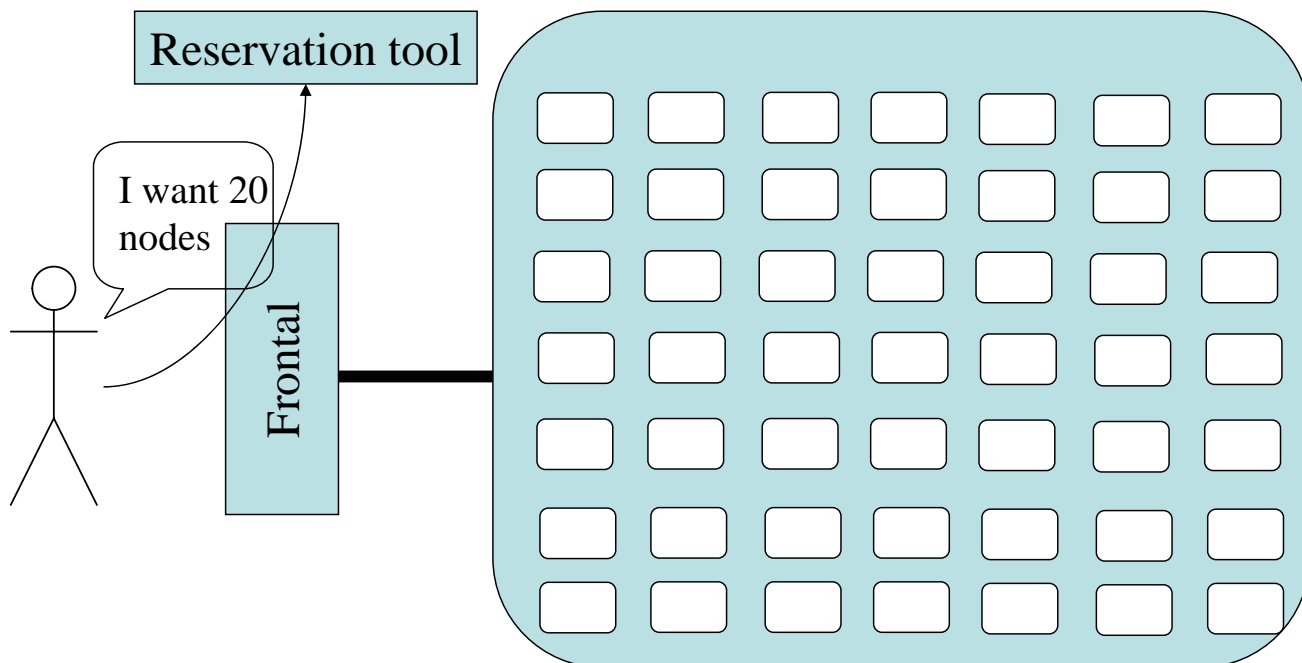
Each node as also :

- a Globus container ready to be activated
- MDS service activated
- GRAM for local jobs submission (fork)
- RFT and GridFTP activated
- Security service activated.

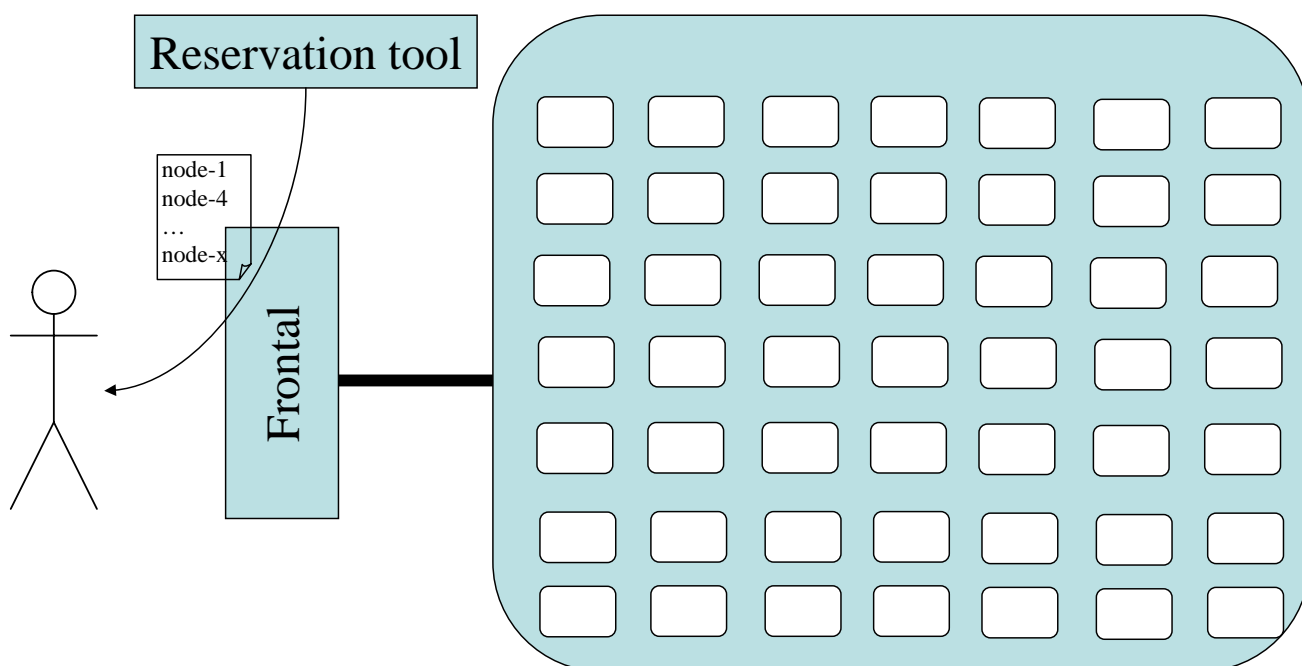
Other services may be present but not properly configured.



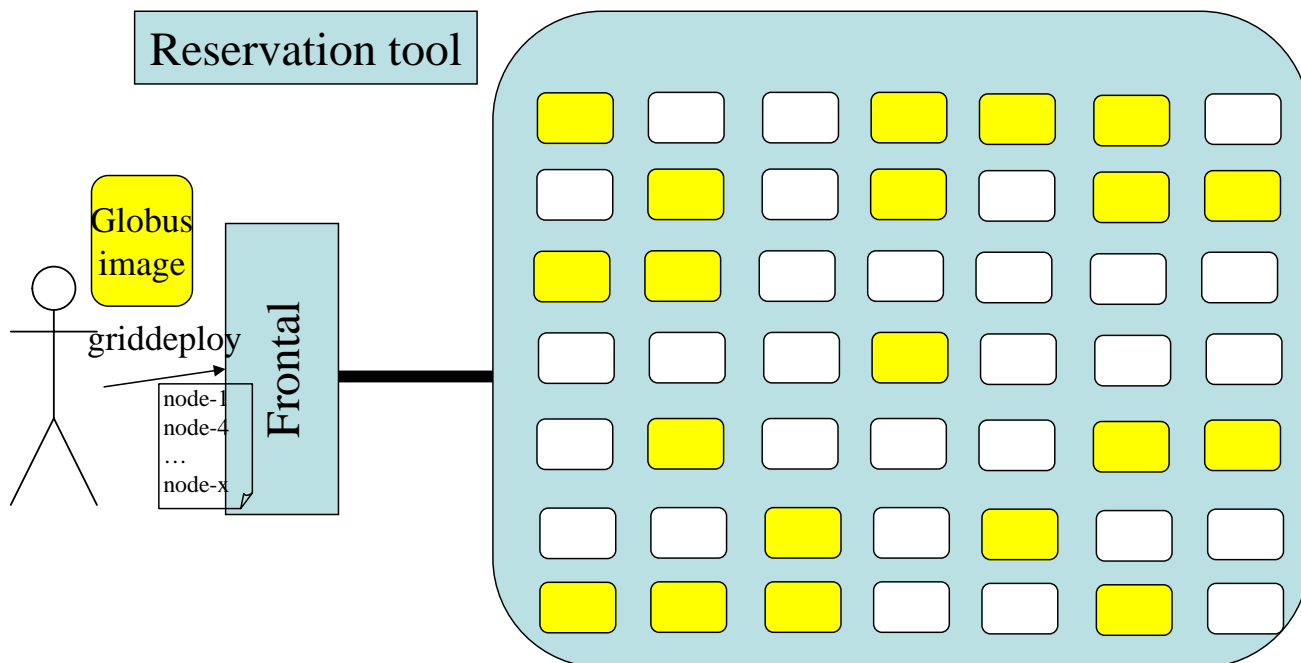
# Globus deployment overview



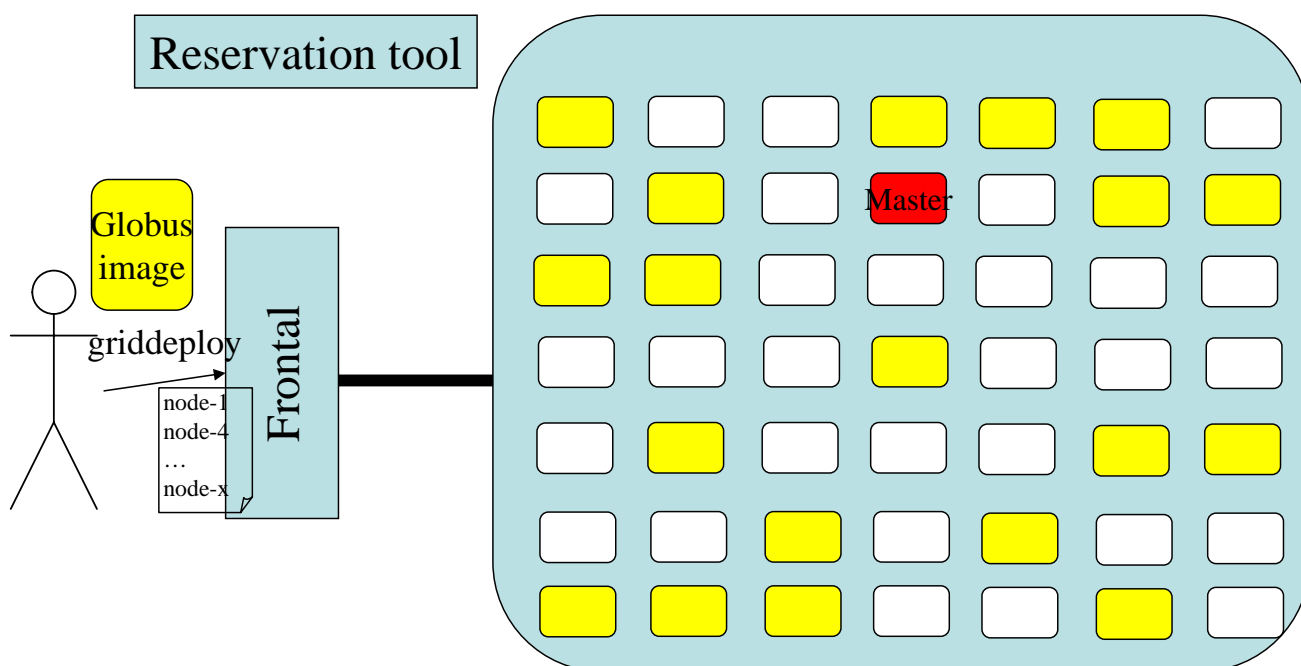
# Globus deployment overview



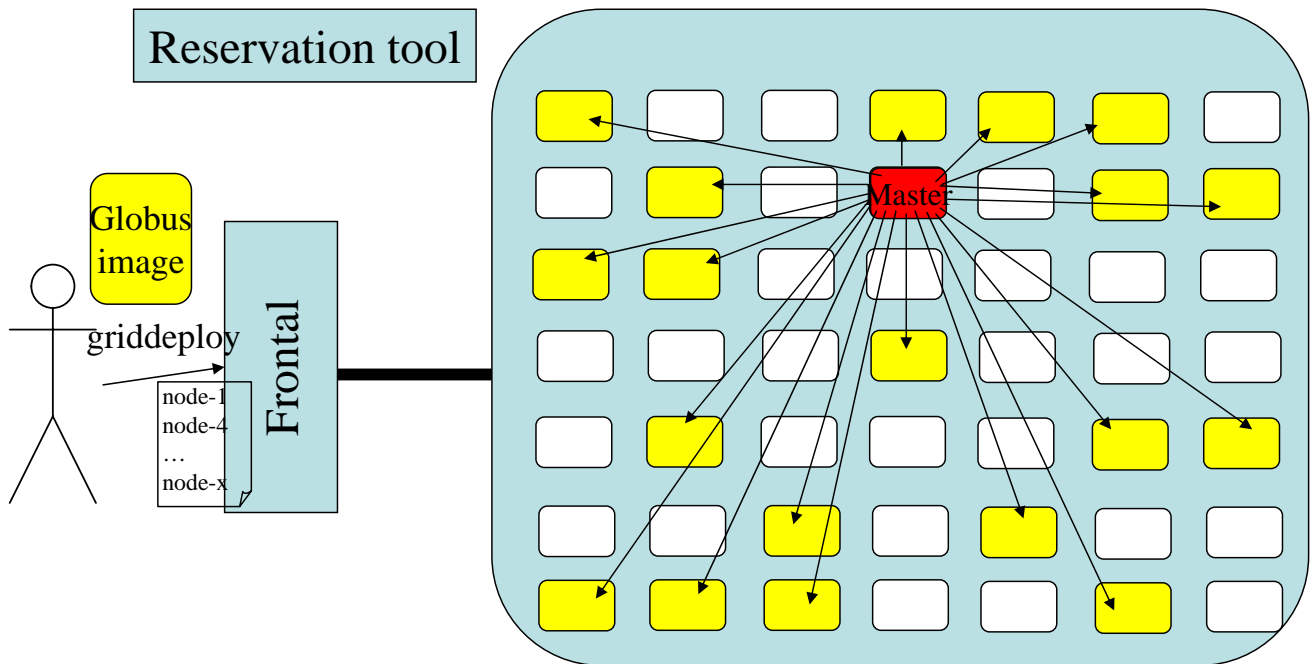
# Globus deployment overview



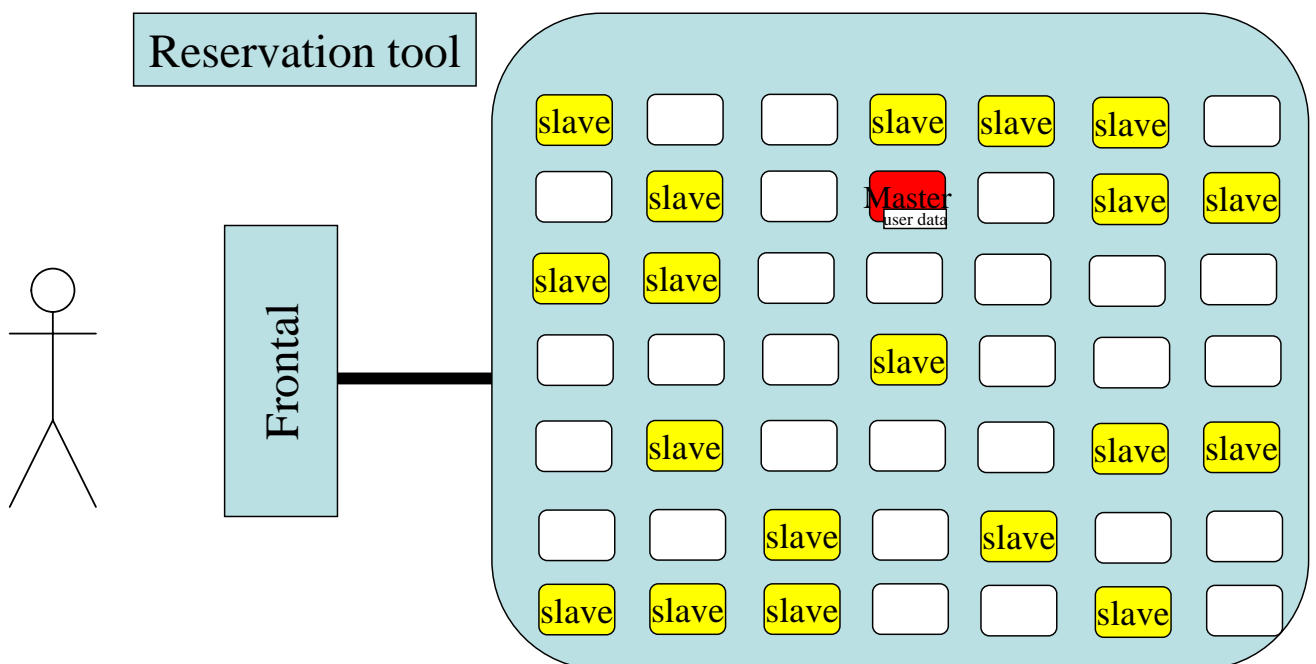
# Globus deployment overview



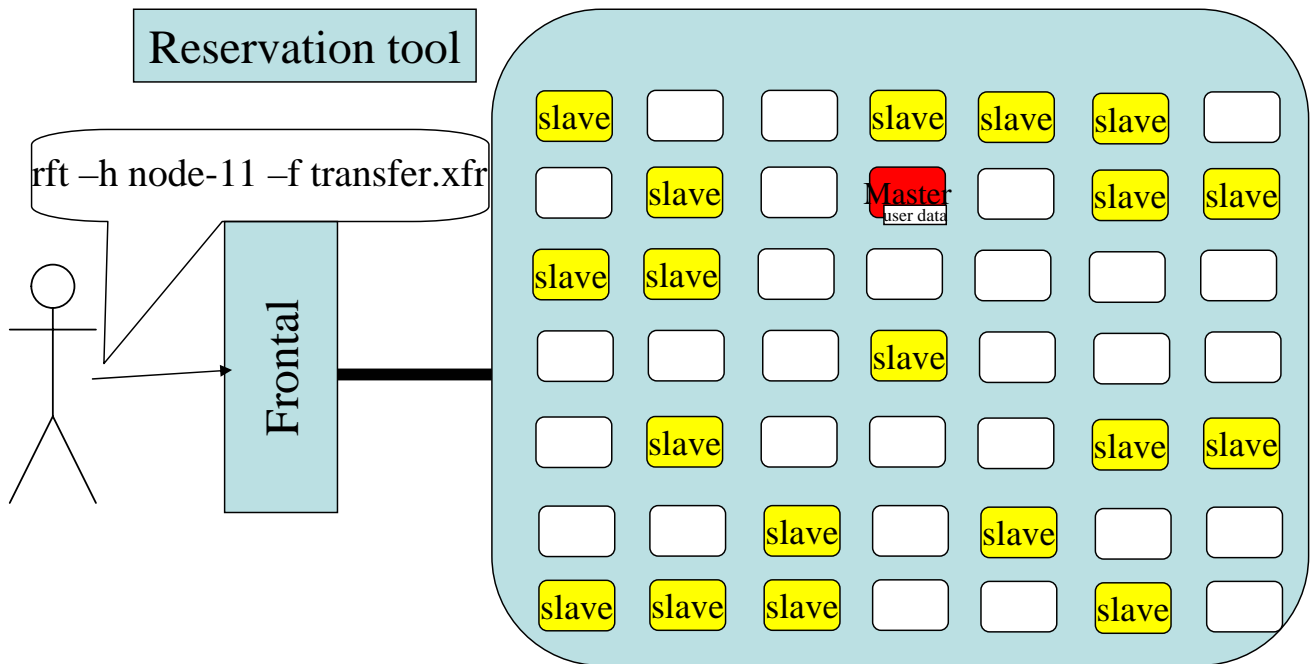
# Globus deployment overview



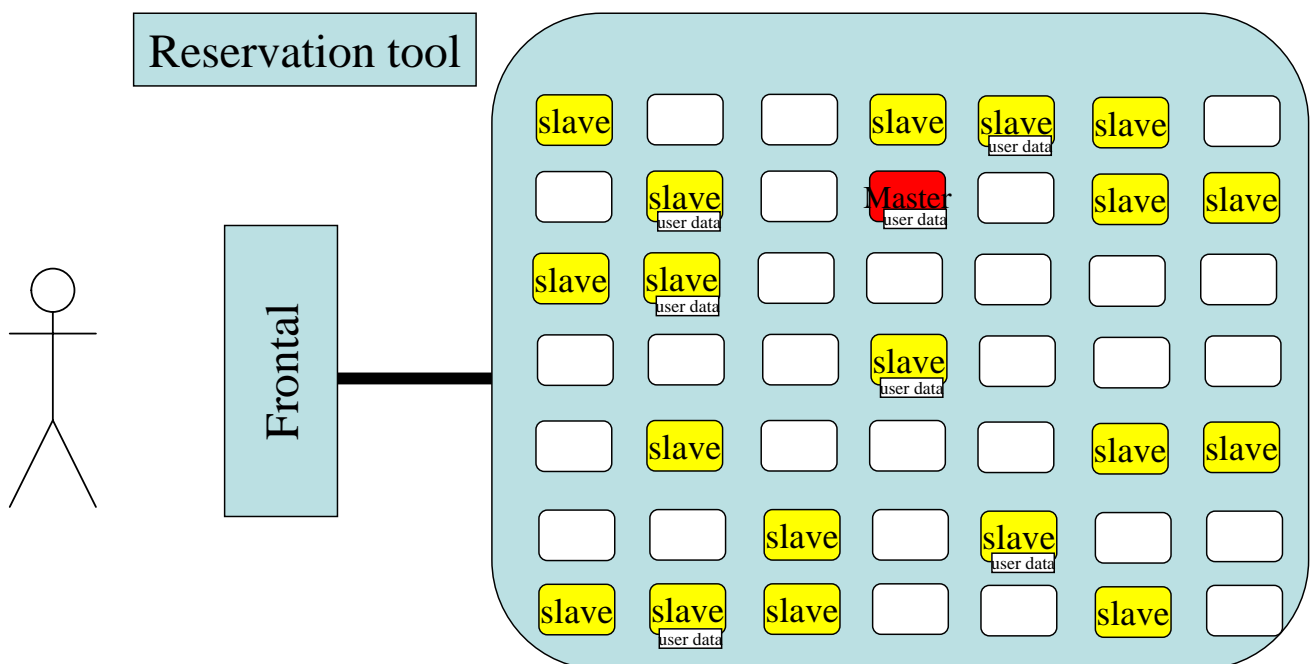
# Globus deployment overview



# Globus data transfer overview



# Globus data transfer overview





## III. Demonstration with ParadisEO-G



### The experiment

#### ParadisEO-G (Open Source Framework, Dolphin Team)

- PARAllel and DIStributed Evolving Objects for Grid
- enables the easy and fast development of the most common parallel and distributed models of hybrid metaheuristics.

#### The Traveling Salesman Problem (TSP)

- Data : a collection of cities and the cost of travel between each pair of them
- Goal : find the cheapest way of visiting all of the cities and returning to your starting point

#### The demonstration :

- Find the solution for a given TSP problem
- Computing done on Grid 5000 with ParadisEO and Globus



# RoadMap to run the experiment (interactive mode)

## I. Prepare the Grid

- Reserve nodes :

```
oargridsub -q deploy -w 08:00:00 lille:nodes=10,bordeaux:nodes=10,sophia:nodes=20
```

```
[OAR_GRIDSUB] Reservation success on lille : batchId = 1995, nbNodes = 10, weight = 0, properties = "", queue = deploy  
[OAR_GRIDSUB] Reservation success on bordeaux : batchId = 14615, nbNodes = 10, weight = 0, properties = "", queue = deploy  
[OAR_GRIDSUB] Reservation success on sophia : batchId = 146084, nbNodes = 20, weight = 0, properties = "", queue = deploy  
[OAR_GRIDSUB] Grid reservation id = 1879
```

- and store nodes in a file : `oargridstat -l 1879 > mynodes`

- Deploy the globus Image on all sites with this command :

```
griddeploy.sh mynodes globus4
```

- Choose a master node (anyone).

Then transfer the file containing the list of nodes on the master

```
scp mynodes root@<MASTER>:
```

- and finally log in as root on the master and run :

```
# ./globus/install.sh mynodes
```



# RoadMap to run the experiment (interactive mode)

## II. Run the experiment

- Transfert data to a globus node : `scp -r frontale.lille.grid5000.fr:<your_data_dir> .`

- Dump a job description file (RSL) :

```
mpirun -dumprsl -np <processor_number> <my_prog> <my_arguments> > <my_prog>.rsl
```

```
cat <my_prog>.rsl | sed -e 's|(executable="(.*\|"))|(executable=$(GLOBUSRUN_GASS_URL)\1)|g' > test.rsl
```

In our example :

```
mpirun -dumprsl -np 40 para_ga @tsp.param para_ga.rsl
```

```
cat para_ga.rsl | sed -e 's|(executable="(.*\|"))|(executable=$(GLOBUSRUN_GASS_URL)\1)|g' > test.rsl
```

The second line with sed substitutions is only necessary to stage in your data (including binary) if they are not present on each node (you can transfer it with RFT for instance).

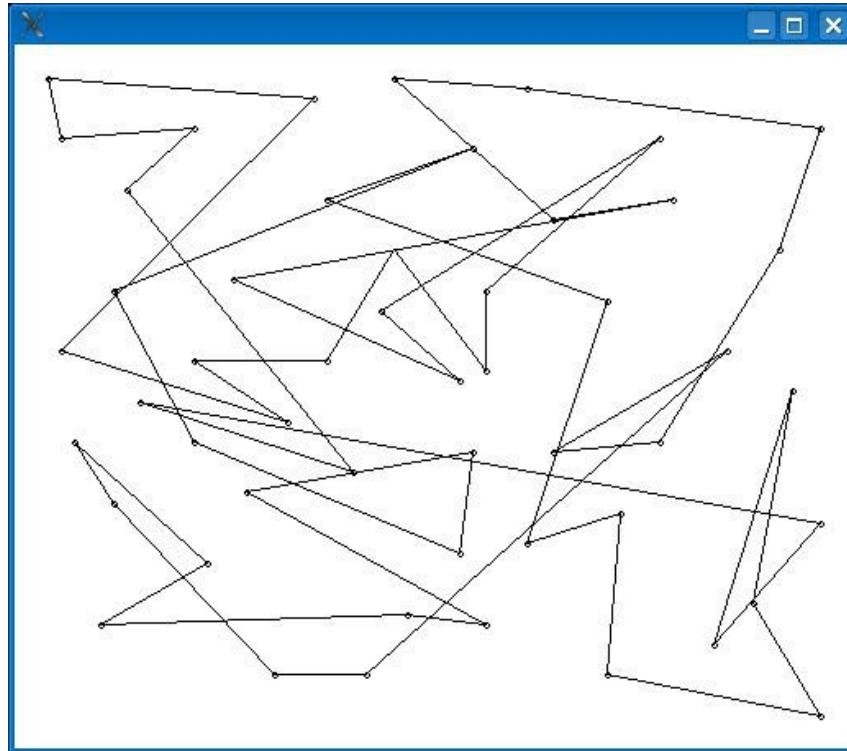
- Run the demo :

```
globusrun -s -f test.rsl
```

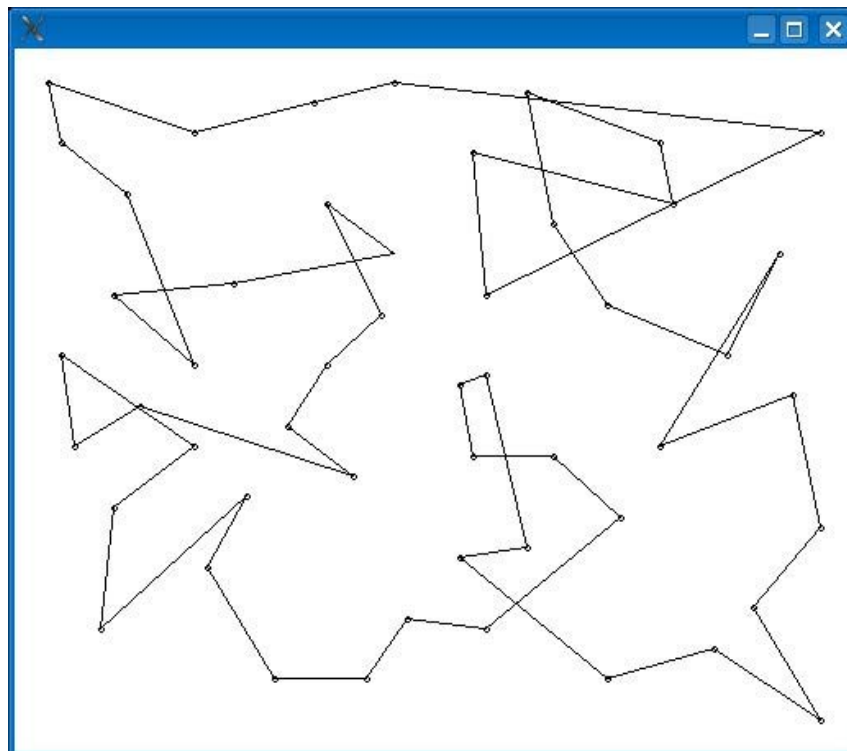
Note : In batch mode, all this stuff is automatically done for you.



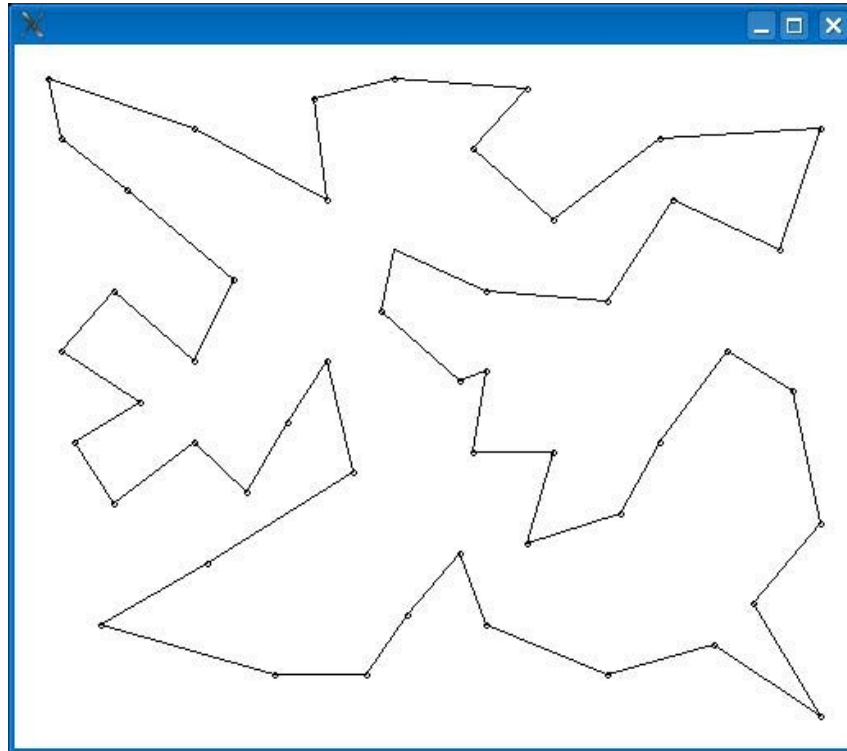
## ParadisEO-G solving a TSP Problem (1/3)



## ParadisEO-G solving a TSP Problem (2/3)



## ParadisEO-G solving a TSP Problem (3/3)



Thanks to :

- J. Garett, C. Frézier, A. Dumez, C. Inglebert, N. Niclausse, J. Bertot (INRIA Futurs & Sophia)
- S. Cahon, N. Melab, E-G. Talbi (INRIA Futurs, Dolphin Team)
- Ian Foster, Lee Liming (U. of Chicago and Argonne National Laboratory)

# QUESTIONS ?