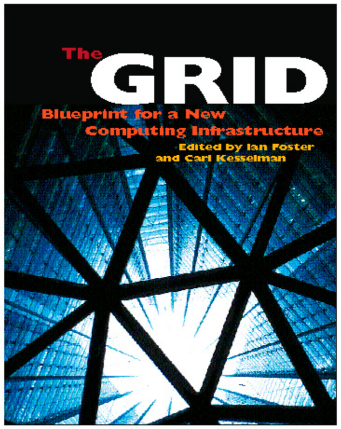
Globus

The INFNGrid Project Team

The Globus project

- Project led by I. Foster (ANL) and C. Kesselman (ISI), the “fathers” of the Grid
- Focus: research on Grid technologies
  - Security
  - Resource Management
  - Data Management
  - Information Services
  - …
- Development of the Globus toolkit
- Goal: “*Making Grid computing a reality*”
  - Very active in the Global Grid Forum


The GRID
Blueprint for a New Computing Infrastructure
Edited by Ian Foster and Carl Kesselman

http://www.globus.org

The Globus toolkit

- A software toolkit addressing key technical problems in the development of Grid enabled tools, services, and applications
  - Offer a modular “bag of technologies”
  - Enable *incremental* development of grid-enabled tools and applications
  - Implement standard Grid protocols and APIs
  - Available under open source license

- The following slides refer to Globus v. 2.x

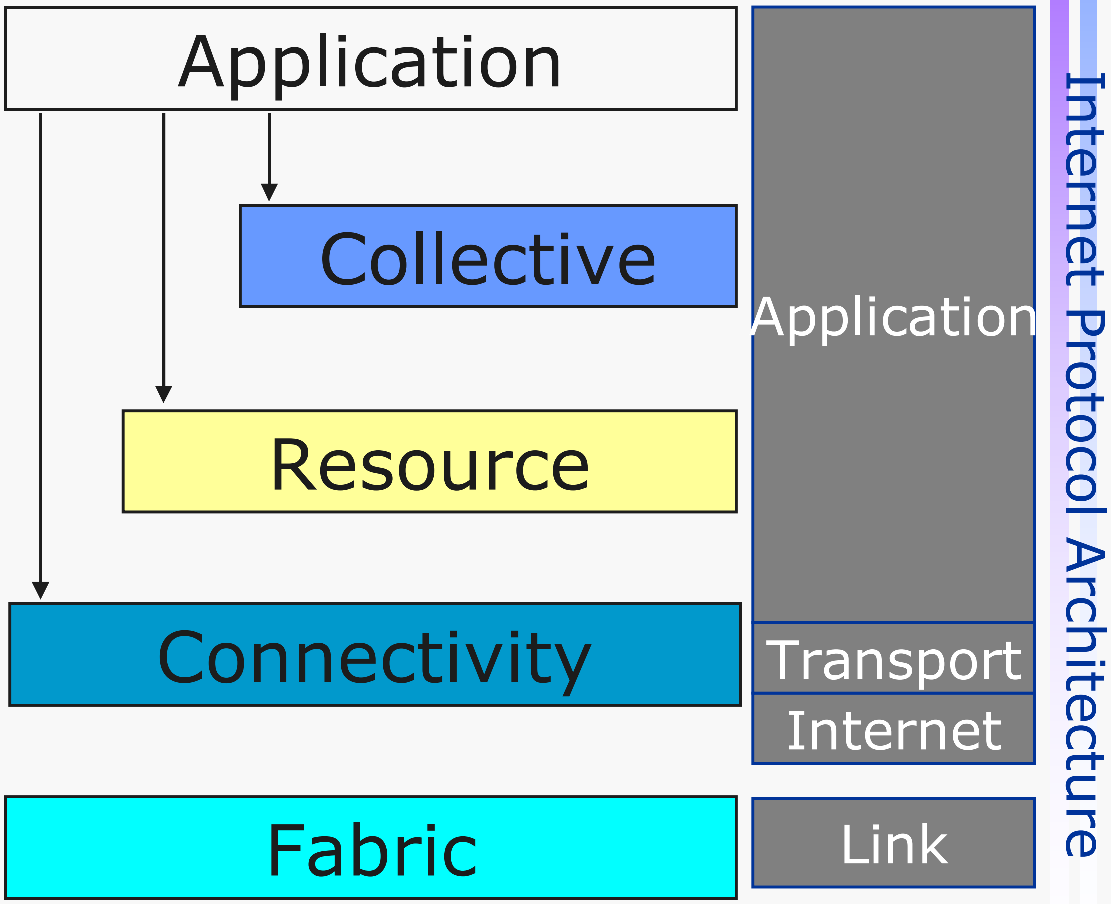
Layered Grid Architecture (By Analogy to Internet Architecture)

"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services

"Sharing single resources": negotiating access, controlling use

"Talking to things": communication (Internet protocols) & security

"Controlling things locally": Access to, & control of, resources



Application

Collective

Resource

Connectivity

Fabric

Application

Transport

Internet

Link

Internet Protocol Architecture

Where are we with this architecture ?

- No “official” standards exist
- But Globus Toolkit has emerged as the de facto standard for various important protocols:
  - Connectivity layer:
    - *Security*: Grid Security Infrastructure (GSI)
  - Resource layer:
    - *Resource Management*: Grid Resource Allocation Management (GRAM)
    - *Information Services*: Monitoring and Discovery Service (MDS)
    - *Data Transfer*: Grid File Transfer Protocol (GridFTP)
  - Also key collective layer protocols
    - Replica Management, etc.

Security in Globus

- Grid Security Infrastructure (GSI) for enabling secure authentication and communication over the Grid
- GSI advantages
  - Support of security across organizational boundaries, without the need of a centrally-managed security system
  - Secure communication (authenticated) between elements of a Grid
  - Single sign-on for users
    - Single authentication step allows access to all the Grid resources
    - No need to keep tracks of accounts and passwords at multiple sites
- Based on public key technology
  - Asymmetric keys (private and public keys)

Certificated based authentication

- Every user and service on the Grid is identified via a certificate, which includes:
  - A “Grid user/service” id: the Subject Name
    - E.g. */C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Massimo Sgaravatto/Email=massimo.sgaravatto@pd.infn.it*
  - The public key
  - The identity of the Certification Authority (CA) that has signed the certificate to certify that the public key and the identity belongs to the subject
  - GSI certificates encoded in X.509 standard format
- GSI authentication model
  - User sends certificate over the wire
  - Other end sends user a challenge string
  - User encodes the challenge string with private key
  - Public key is used to decode the challenge
  - If you can decode it, you know the user
- Treat your private key securely
  - In any case the file that contains the private key is encrypted via a pass-phrase
    - To use the GSI, the user must enter the pass phrase

GSI advantages: 1/2

- Mutual authentication
  - Authentication: Establishing identity
  - If two parties have certificates, and if both parties trust the CAs that signed each other’s certificates, then the two parties can prove to each other that they are who they say they are
- Communication integrity
  - An eavesdropper may be able to read communication between two parties, but is not able to modify the communication in any way
- Authorization
  - Authorization: Establishing rights
  - A “grid-map” file at each resource specifies grid-it to local-id mapping
  - E.g.: *"/C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Massimo Sgaravatto/Email=massimo.sgaravatto@pd.infn.it" sgaravat*

User proxies

- A proxy consists of a new certificate (user’s certificate slightly modified) + the user’s private key
- Once a proxy has been created, the proxy can be used for mutual authentication without the need of entering the pass phrase
- Proxies have limited lifetimes
- So a proxy is a temporary user’s credential which allows to act on behalf of the user

GSI advantages 2/2

- Single sign-on
  - Problem: a computation requires that several Grid resources (each one requiring mutual authentication) be used
  - Using proxies, it is not necessary to re-enter the pass-phrase for each resource
- Delegation
  - Problem: e.g. a user process running on a remote resource need to start an other process on an other resource
  - Delegation: remote creation of a user proxy
  - Allows remote processes to act on behalf of the user
  - Avoids sending passwords or private keys across the network

Globus authentication setup

- Before you can run Grid applications
  - Obtain a Grid certificate and have it signed by your CA
  - Contact sites to set up local accounts and grid-mapfile entries
- Logging on the Grid (creating a proxy certificate)
  - ***grid-proxy-init***
- Run your application
- Logging out (destroyng the proxy)
  - ***grid-proxy-destroy***

Resource management in Globus

- The Grid Resource Allocation Management (GRAM) protocol allows programs to be started on remote resources, despite local heterogeneity
- Resource Specification Language (RSL) is used to communicate requirements
- Uniform interface to various local resource management systems
  - LSF,PBS, fork, etc.
  - No need to learn and remember obscure command sequences at different sites

Resource Specification Language

- RSL provides two types of information:
  - Job configuration: Directory, executable, args, environment
  - Resource requirements: Machine type, number of nodes, memory, etc.
- Elementary form: parenthesis clauses
  - (attribute op value [value …])
- Operators Supported:
  - <, <=, =, >=, > , !=
- Some supported attributes:
  - executable, arguments, environment, stdin, stdout, stderr, resourceManagerContact, resourceManagerName
- Unknown attributes are passed through
  - May be handled by subsequent tools

Examples of RSL expressions

```
& (count>=5) (count<=10)(max_time=240) (memory>=64)
      (executable=myprog)
```

“Create 5-10 instances of myprog, each on a machine with at least 64 MB memory that is available to me for 4 hours”

```
& (executable=myprog)
    ( | (&(count=5)(memory>=64))
        (&(count=10)(memory>=32)))
```

“Create 5 instances of myprog on a machine that has at least 64MB of memory, or 10 instances on a machine with at least 32MB of memory”

```
+ (& (count=5)(memory>=64)(executable=p1))
  (&(network=atm) (executable=p2))
```

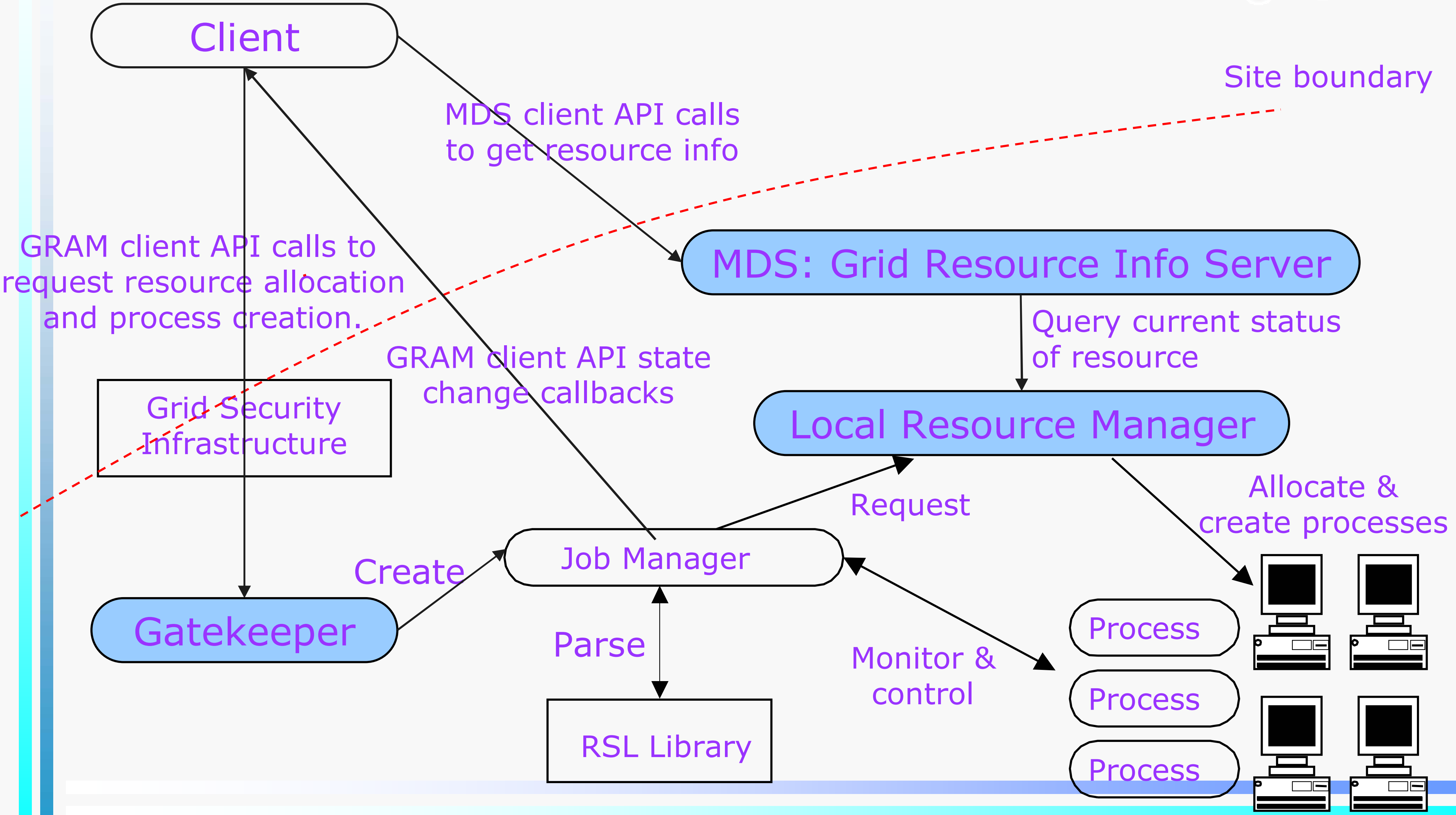
“Execute 5 instances of p1 on a machine with at least 64M of memory.

Execute p2 on a machine with an ATM connection”

GRAM implementation

- Gatekeeper
  - Single point of entry
  - Authenticates user, maps to local security environment, runs job managers
- Job manager
  - A gatekeeper service
  - Layers on top of local resource management system (e.g., PBS, LSF, etc.)
  - Handles remote interaction with the job

GRAM Components


Client
MDS client API calls to get resource info
Site boundary
GRAM client API calls to request resource allocation and process creation.
MDS: Grid Resource Info Server
Query current status of resource
GRAM client API state change callbacks
Grid Security Infrastructure
Local Resource Manager
Request
Allocate & create processes
Create
Job Manager
Gatekeeper
Parse
Monitor & control
Process
Process
Process
RSL Library

Job Submission Interfaces

- Globus Toolkit includes several command line programs for job submission
  - *globusrun*: Flexible scripting infrastructure
  - *globus-job-run*: To run interactive jobs
  - *globus-job-submit*: To submit batch jobs

  - *globus-job-run* and *globus-job-submit* simply rely on *globusrun*

globusrun

- Uses an RSL string to specify job request
- Supports both interactive and batch jobs
- E.g.:
  - ***globusrun -w -r lxde01.pd.infn.it/jobmanager-fork -f ts1.rsl***
  - ***globusrun -w -r lxde02.pd.infn.it/jobmanager-lsf -f ts1.rsl***
  - ***globusrun -w -r lxde03.pd.infn.it/jobmanager-pbs -f ts1.rsl***

  ts1.rls:

```
&
(executable=/bin/echo)
(arguments= "Hello World !")
(stdout=$(GLOBUSRUN_GASS_URL)/tmp/oo)
(count=1)
```

Local Resource Management Systems

- Complex to use
  - Must write RSL by hand
  - Generally users prefer using globus-job-* commands instead

globus-job-run

- For running of interactive jobs
- E.g.:

```
$ globus-job-run lxde01.pd.infn.it/jobmanager-pbs /bin/ls /tmp
edg-replica-manager-wp1-UegEHK.log
globus_job_run.sgaravat.rsl.26389
globus_job_run.sgaravat.tmp.26389
lost+found
mio
pim.info.lxde01.pd.infn.it
pim.log.lxde01.pd.infn.it
res.log.lxde01.pd.infn.it
sbatchd.log.lxde01.pd.infn.it
x509up_u51
```

globus-job-submit

- For running of batch jobs
  - ***globus-job-submit***: Submit job
    - Same interface as globus-job-run
    - Returns immediately
  - ***globus-job-status***: Check job status
  - ***globus-job-cancel***: Cancel job
  - ***globus-job-get-output***: Get job stdout/err
  - ***globus-job-clean***: Cleanup after job

Condor-G

- Enhanced version of Condor that uses Globus Toolkit to manage Grid jobs
- Advantages of using Condor-G to manage your Grid jobs
  - Full-featured queuing service
  - Credential Management
  - Fault-tolerance

Full-Featured Queue

- Persistent queue
- Many queue-manipulation tools
- Set up job dependencies (DAGman)
- Log files
- E-mail notification of events

Credential Management

- Authentication in Globus Toolkit is done with limited-lifetime X509 proxies
- Proxy may expire before jobs finish executing
- Condor-G can forward new proxy to execution sites

Fault Tolerance

- Local Crash
  - Queue state stored on disk
  - Reconnect to execute machines
- Network Failure
  - Wait until connectivity returns
  - Reconnect to execute machines
- Remote Crash – job still in queue
  - Job state stored on disk
  - Start new jobmanager to monitor job
- Remote Crash – job lost
  - Resubmit job

Example of CondorG submission

- ***condor_submit test.cnd***

  test.cnd:

  ```
  universe=globus
  executable=script1.sh
  Transfer_executable=true
  Transfer_output=true
  output=out.out.$(Process)
  error=err.err.$(Process)
  log=log.log
  x509userproxy=/home/edguser/sgara/sgaraproxy
  GlobusScheduler=lxde01.pd.infn.it/jobmanager-pbs
  GlobusRSL=(queue=short)
  queue 1
  ```

- *condor_q* to check the status of the jobs in the queue
- *condor_rm* to cancel a job

Data Management Services

- Two major data management services in the Globus toolkit
  - GridFTP
    - High-performance, secure data transfer protocol
  - Replica Location Service (RLS)
    - Maintains and provides mapping information from logical names for data to their physical instances
    - See Flavia's talk on data management services

GridFTP

- Extended version of popular FTP protocol for Grid data access and transfer
- High-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide area networks
- Some protocol features
  - GSI security on control and data channel
  - Multiple data channels for parallel transfers
  - Partial file transfers
  - Third party (direct server-to-server) transfers

Gridftp protocol implementation

- APIs and command line tool (globus-url-copy)
- Washington University’s FTP server (wu-ftpd) adapted to support a majority of the GridFTP protocol features

- E.g.

  globus-url-copy |
  gsiftp://host1.pd.infn.it/data/filename |
  file:///home/userx/myfile

Information Services

- Globus toolkit provides MDS
- See talk on Information Services

More info

- Globus
  - http://www.globus.org
- Condor and Condor-G
  - http://www.cs.wisc.edu/condor