



Using Grids

Niels Drost

(slides stolen from) Rob van Nieuwpoort

niels@cs.vu.nl



Introduction

Up till now:

- “How to write grid applications”
 - Programming Models
 - Communication Libraries
 - Connectivity Trouble

This talk and the next (last) talk:

- “How to run applications on the grid”



In Other words

File View Options Help

Experiment Applications Clusters

Experiment Editor

Pool Name: demo

Select Application

Broker x 1

Select Cluster

Chiba x 1

Experiment Monitor

Zoom Rotate

	pool	name	job status	hub status	cluster	middleware	application	process count	resource count	output
▶	demo	AU-Hiroshi	ERROR	ERROR	Hiroshi	ssh	Server	4	4	output
□	demo	Broker	DEPLOYED	DEPLOYED	local	local	Broker	1	1	output
▶	demo	Client	DONE	DEPLOYED	local	local	Client	1	1	output
□	demo	JP-Chiba	WAITING	UPLOADING	Chiba	zorilla	Server	8	0	output
□	demo	JP-Tsukuba	WAITING	UPLOADING	Tsukuba	zorilla	Server	8	8	output
□	demo	NL-DesktopGrid	WAITING	UPLOADING	DesktopGrid	zorilla	Server	4	0	output
□	demo	NL-Leiden	WAITING	UPLOADING	Leiden	globus	Server	8	8	output
□	demo	NL-MultimediaN	WAITING	UPLOADING	MultimediaN	globus	Server	8	8	output
□	demo	NL-Sedna	WAITING	UPLOADING	Sedna	ssh	Server	1	1	output
▶	demo	NL-UvA	ERROR	DEPLOYED	UvA	globus	Server	8	8	output

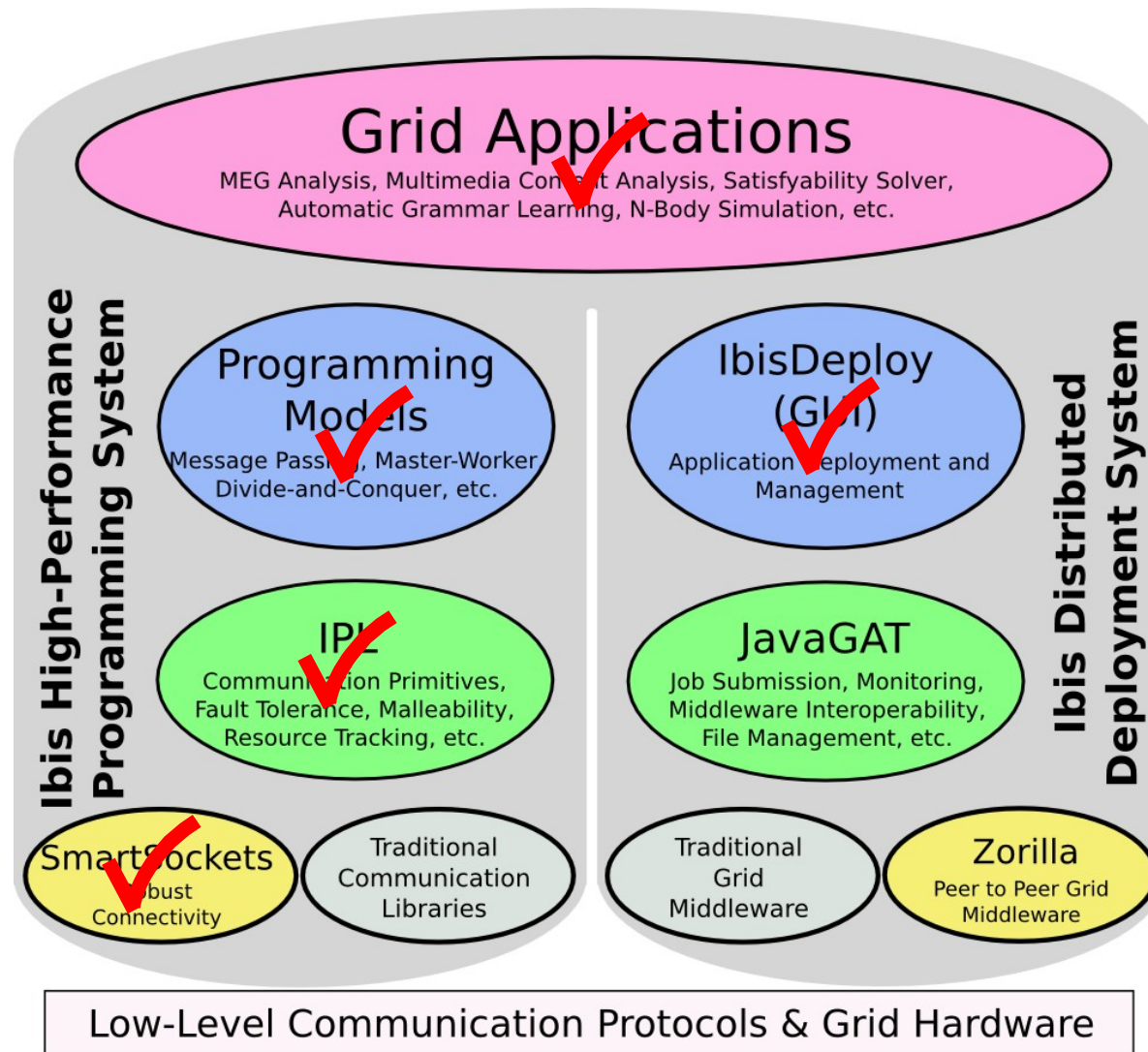
Start All Stop All Remove All

Start Selected Stop Selected Remove Selected

How did we (try to) create this?



Overview



Writing Grid Applications

Grid Application



Writing Grid Applications

File.copy(...)

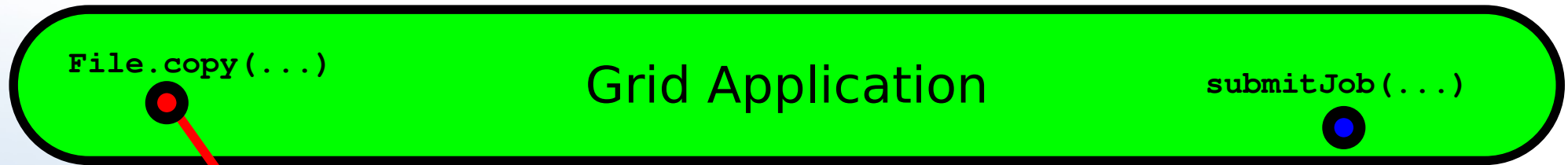


Grid Application

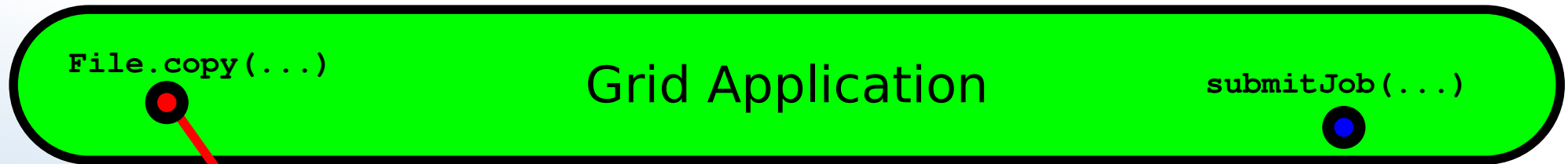
submitJob(...)



Writing Grid Applications



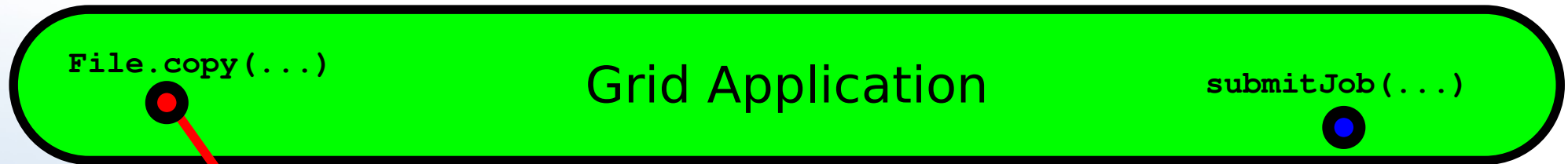
Writing Grid Applications



cp
ftp



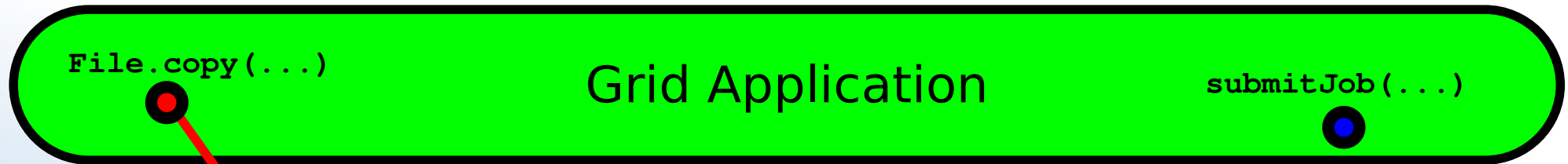
Writing Grid Applications



cp
ftp
gridftp



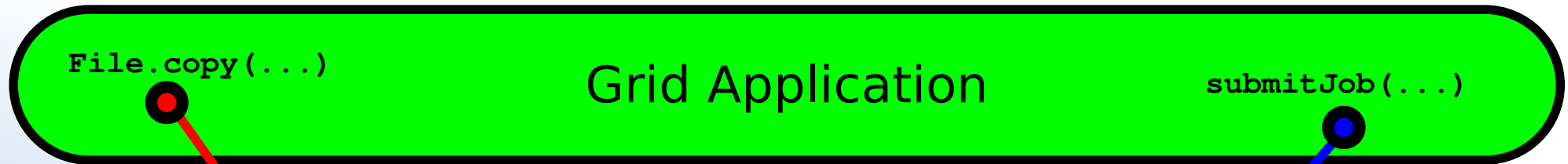
Writing Grid Applications



cp
ftp
gridftp
scp



Writing Grid Applications



cp
ftp
gridftp
scp

fork
pbs
condor
unicore
globus



Writing Grid Applications



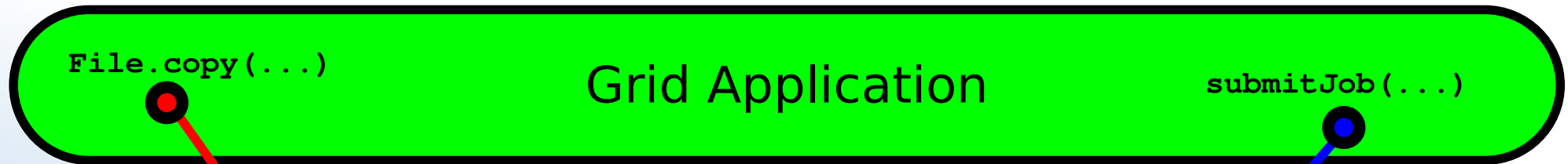
```
cp  
ftp  
gridftp  
scp  
http
```

```
fork  
pbs  
condor  
unicore  
globus
```

- Which should you use?



Writing Grid Applications



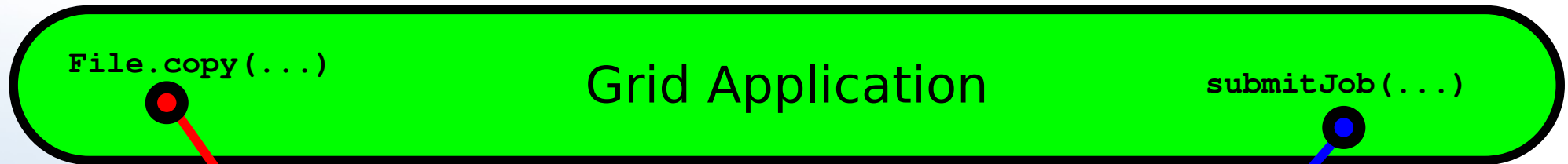
```
cp  
ftp  
gridftp  
scp  
http
```

```
fork  
lfs  
condor  
unicore  
globus
```

- Which should you use?
- Some might not be available on all sites



Writing Grid Applications



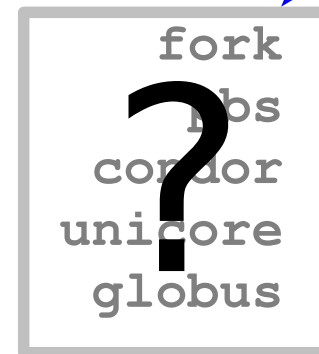
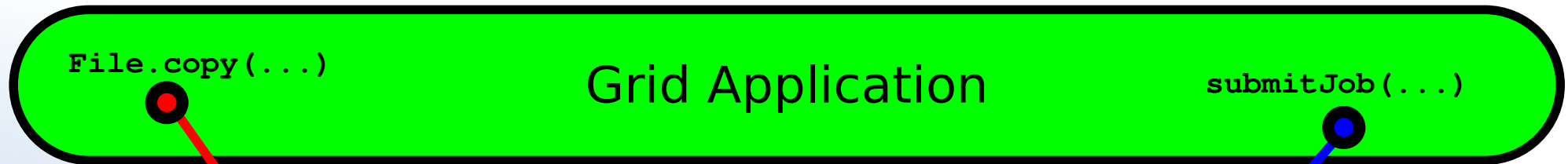
```
cp  
ftp  
gridftp  
scp  
http
```

```
fork  
lfs  
condor  
unicore  
globus
```

- Which should you use?
- Some might not be available on all sites
- Some may not work for all users (certificates)



Writing Grid Applications



- Which should you use?
- Some might not be available on all sites
- Some may not work for all users (certificates)
- Version differences (Globus changes interface every 6 months)



Globus File copy (C++)

```
int RemoteFile::GetFile (char const* source,  char const* target) {
globus_url_t          source_url;
globus_io_handle_t    dest_io_handle;
globus_ftp_client_operationattr_t source_ftp_attr;
globus_result_t       result;
globus_gass_transfer_requestattr_t source_gass_attr;
globus_gass_copy_attr_t source_gass_copy_attr;
globus_gass_copy_handle_t gass_copy_handle;
globus_gass_copy_handleattr_t gass_copy_handleattr;
globus_ftp_client_handleattr_t ftp_handleattr;
globus_io_attr_t       io_attr;
int                    output_file = -1;

if ( globus_url_parse (source_URL, &source_url) != GLOBUS_SUCCESS ) {
    printf ("can not parse source_URL \"%s\"\n", source_URL);
    return (-1);
}

if ( source_url.scheme_type != GLOBUS_URL_SCHEME_GSIFTP &&
    source_url.scheme_type != GLOBUS_URL_SCHEME_FTP &&
    source_url.scheme_type != GLOBUS_URL_SCHEME_HTTP &&
    source_url.scheme_type != GLOBUS_URL_SCHEME_HTTPS ) {
    printf ("can not copy from %s - wrong prot\n", source_URL);
    return (-1);
}

globus_gass_copy_handleattr_init (&gass_copy_handleattr);
globus_gass_copy_attr_init (&source_gass_copy_attr);

globus_ftp_client_handleattr_init (&ftp_handleattr);
globus_io_fileattr_init (&io_attr);

globus_gass_copy_attr_set_io (&source_gass_copy_attr, &io_attr);
globus_gass_copy_handleattr_set_ftp_attr (&gass_copy_handleattr,
&ftp_handleattr);
globus_gass_copy_handle_init (&gass_copy_handle,
&gass_copy_handleattr);
```

```
int RemoteFile::GetFile (char const* source,  char const* target) {
globus_url_t          source_url;
globus_io_handle_t    dest_io_handle;
globus_ftp_client_operationattr_t source_ftp_attr;
globus_result_t       result;
globus_gass_transfer_requestattr_t source_gass_attr;
globus_gass_copy_attr_t source_gass_copy_attr;
globus_gass_copy_handle_t gass_copy_handle;
globus_gass_copy_handleattr_t gass_copy_handleattr;
globus_ftp_client_handleattr_t ftp_handleattr;
globus_io_attr_t       io_attr;
int                    output_file = -1;

if ( globus_url_parse (source_URL, &source_url) != GLOBUS_SUCCESS ) {
    printf ("can not parse source_URL \"%s\"\n", source_URL);
    return (-1);
}

if ( source_url.scheme_type != GLOBUS_URL_SCHEME_GSIFTP &&
    source_url.scheme_type != GLOBUS_URL_SCHEME_FTP &&
    source_url.scheme_type != GLOBUS_URL_SCHEME_HTTP &&
    source_url.scheme_type != GLOBUS_URL_SCHEME_HTTPS ) {
    printf ("can not copy from %s - wrong prot\n", source_URL);
    return (-1);
}

globus_gass_copy_handleattr_init (&gass_copy_handleattr);
globus_gass_copy_attr_init (&source_gass_copy_attr);

globus_ftp_client_handleattr_init (&ftp_handleattr);
globus_io_fileattr_init (&io_attr);

globus_gass_copy_attr_set_io (&source_gass_copy_attr, &io_attr);
globus_gass_copy_handleattr_set_ftp_attr (&gass_copy_handleattr,
&ftp_handleattr);
globus_gass_copy_handle_init (&gass_copy_handle,
&gass_copy_handleattr);
```



CoG/RFT File Copy (C++)

```
package org.globus.ogsa.gui;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.net.URL;
import java.util.Date;
import java.util.Vector;
import javax.xml.rpc.Stub;
import org.apache.axis.message.MessageElement;
import org.apache.axis.utils.XMLUtils;
import org.globus.*
import org.gridforum.ogsi.*
import org.gridforum.ogsi.holders.TerminationTimeTypeHolder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class RFTClient {
    public static void copy (String source_url, String target_url) {
        try {
            File requestFile = new File (source_url);
            BufferedReader reader = null;
            try {
                reader = new BufferedReader (new FileReader (requestFile));
            } catch (java.io.FileNotFoundException fnfe) { }
            Vector requestData = new Vector ();
            requestData.add (target_url);
            TransferType[] transfers1 = new TransferType[transferCount];
            RFTOptionsType multirftOptions = new RFTOptionsType ();

            multirftOptions.setBinary (Boolean.valueOf (
                (String)requestData.elementAt (0)).booleanValue ());
            multirftOptions.setBlockSize (Integer.valueOf (
                (String)requestData.elementAt (1)).intValue ());
            multirftOptions.setTcpBufferSize (Integer.valueOf (
                (String)requestData.elementAt (2)).intValue ());
            multirftOptions.setNotpt (Boolean.valueOf (
                (String)requestData.elementAt (3)).booleanValue ());
            multirftOptions.setParallelStreams (Integer.valueOf (
                (String)requestData.elementAt (4)).intValue ());
            multirftOptions.setDcau (Boolean.valueOf (
                (String)requestData.elementAt (5)).booleanValue ());

            int i = 7;
            for (int j = 0; j < transfers1.length; j++)
            {
                transfers1[j] = new TransferType ();

                transfers1[j].setTransferId (j);
                transfers1[j].setSourceUrl ((String)requestData.elementAt (i++));
                transfers1[j].setDestinationUrl ((String)requestData.elementAt (i++));
                transfers1[j].setRftOptions (multirftOptions);
            }
        }
    }
}
```

```
TransferRequestType transferRequest = new TransferRequestType ();
transferRequest.setTransferArray (transfers1);

int concurrency = Integer.valueOf
    ((String)requestData.elementAt(6)).intValue();

if (concurrency > transfers1.length)
{
    System.out.println ("Concurrency should be less than the number"
        "of transfers in the request");
    System.exit (0);
}
transferRequest.setConcurrency (concurrency);

TransferRequestElement requestElement = new TransferRequestElement ();
requestElement.setTransferRequest (transferRequest);

ExtensibilityType extension = new ExtensibilityType ();
extension = AnyHelper.getExtensibility (requestElement);

OGSIServiceGridLocator factoryService = new OGSIServiceGridLocator ();
Factory factory = factoryService.getFactoryPort (new URL (source_url));
GridServiceFactory gridFactory = new GridServiceFactory (factory);

LocatorType locator = gridFactory.createService (extension);
System.out.println ("Created an instance of Multi-RFT");

MultiFileRFTDefinitionServiceGridLocator loc
    = new MultiFileRFTDefinitionServiceGridLocator();
RFTPortType rftPort = loc.getMultiFileRFTDefinitionPort (locator);
((Stub)rftPort)._setProperty (Constants.AUTHORIZATION,
    NoAuthorization.getInstance());
((Stub)rftPort)._setProperty (GSIConstants.GSI_MODE,
    GSIConstants.GSI_MODE_FULL_DELEG);
((Stub)rftPort)._setProperty (Constants.GSI_SEC_CONV,
    Constants.SIGNATURE);
((Stub)rftPort)._setProperty (Constants.GRIM_POLICY_HANDLER,
    new IgnoreProxyPolicyHandler ());

int requestid = rftPort.start ();
System.out.println ("Request id: " + requestid);

}
catch (Exception e)
{
    System.err.println (MessageUtils.toString (e));
}
}
```



Problems of Grid Applications

- Too many different middlewares / interfaces.
- Which are too hard to use
- And change too much over time
- And contain too many bugs, omissions and obfuscations



Overview

- ✓ Problems of Grid applications
 - JavaGAT Overview
 - JavaGAT API basics
 - Security API
 - Grid I/O



The (Java)GAT

- GAT: Grid Application Toolkit
 - API and Toolkit for developing and running portable grid applications independently of the underlying grid infrastructure and available services
- Simple API
- Uses adaptors (plugins) to access actual grid middleware
- Used with, not instead of existing middleware



GAT API features

- Security (deal with passwords, credentials, etc)
- Grid I/O
 - File operations, remote file access, file replication
 - Inter-process communication
- Resource Management
 - Resource brokering
 - Forking grid applications, job management

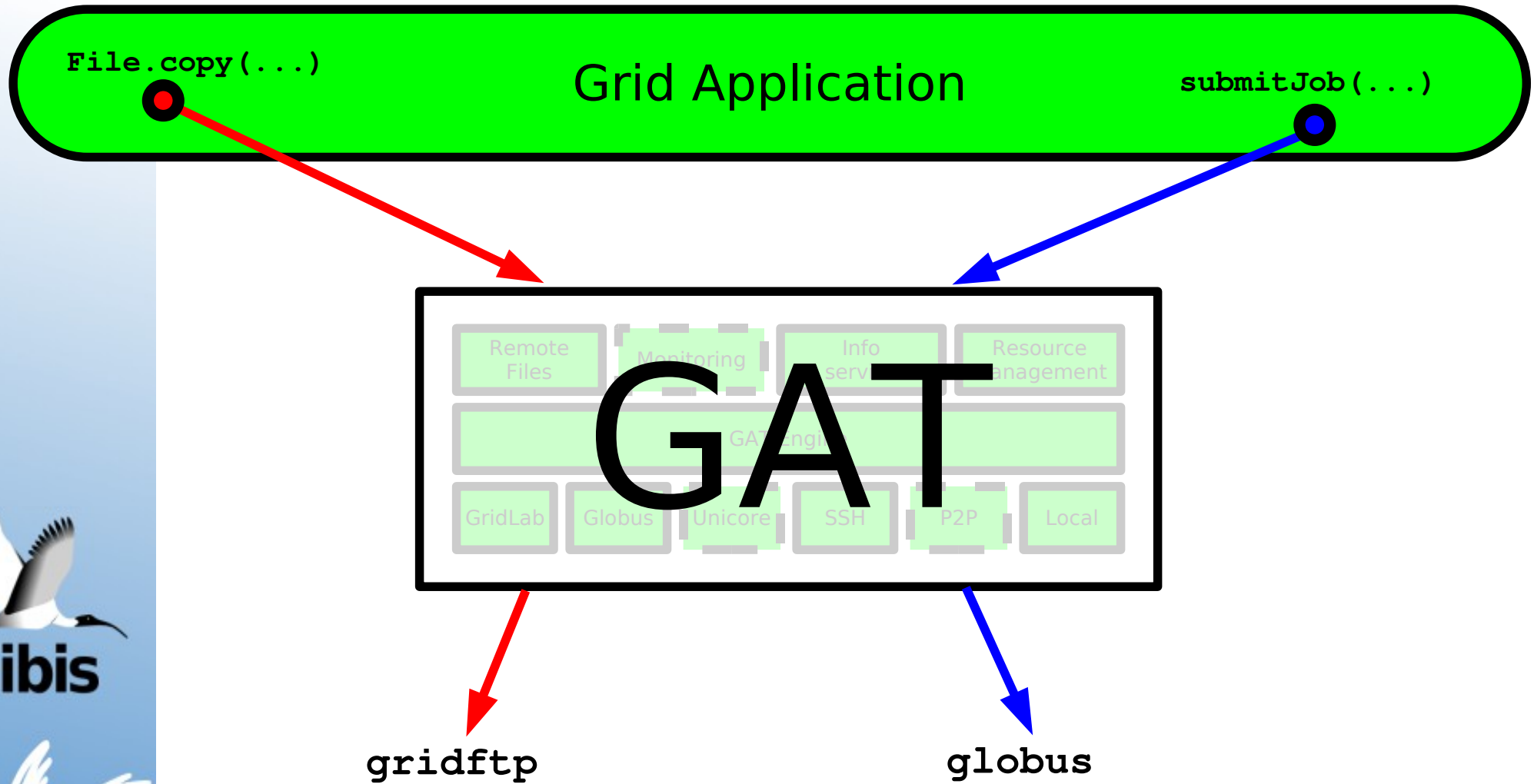


GAT API Features (Cond')

- Application Information Management
 - Global repository for application specific information
 - Query this information repository
- Monitoring
 - Grid monitoring
 - Application monitoring and steering



Grid Applications with GAT



File Copy with JavaGAT

```
import org.gridlab.gat.*;
import org.gridlab.gat.io.File;

public class RemoteCopy {
    public static void main(String[] args) throws Exception {
        GATContext context = new GATContext();
        File file = GAT.createFile(context, new URI(args[0]));
        file.copy(new URI(args[1]));
    }
}
```



File Copy with JavaGAT

```
import org.gridlab.gat.*;
import org.gridlab.gat.io.File;

public class RemoteCopy {
    public static void main(String[] args)
        throws Exception {
        File file = GAT.createFile(new URI(args[0]));
        file.copy(new URI(args[1]));
        GAT.end();
    }
}
```



Java GAT Structure

Grid Application

Files

Monitoring

Info
service

Resource
Management

GAT Engine

SGE

Globus

Unicore

SSH

Zorilla

Local

Java GAT Structure

Grid Application

Files

Monitoring

Info
service

Resource
Management

GAT Engine

SGE

Globus

Unicore

SSH

Zorilla

Local

API

Java GAT Structure

Grid Application

API

Files

Monitoring

Info
service

Resource
Management

GAT Engine

ADAPT.

SGE

Globus

Unicore

SSH

Zorilla

Local

Java GAT Structure

`File.copy(...)`



Grid Application

API

Files

Monitoring

Info
service

Resource
Management

GAT Engine

ADAPT.

SGE

Globus

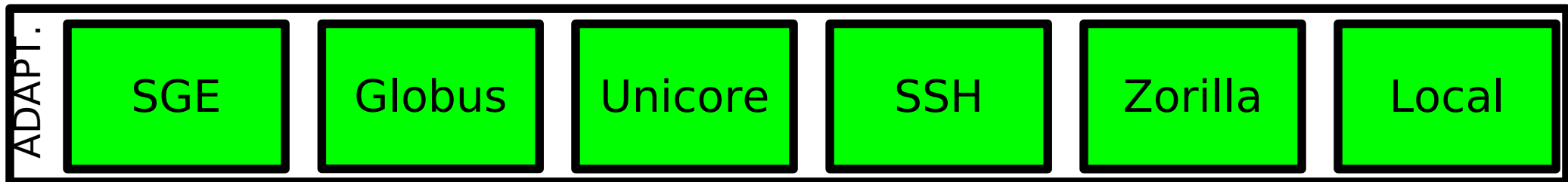
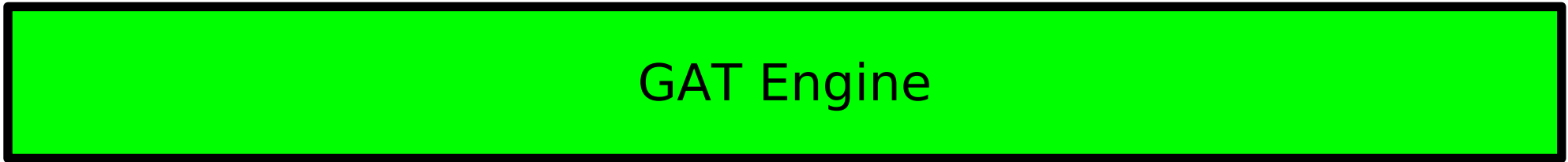
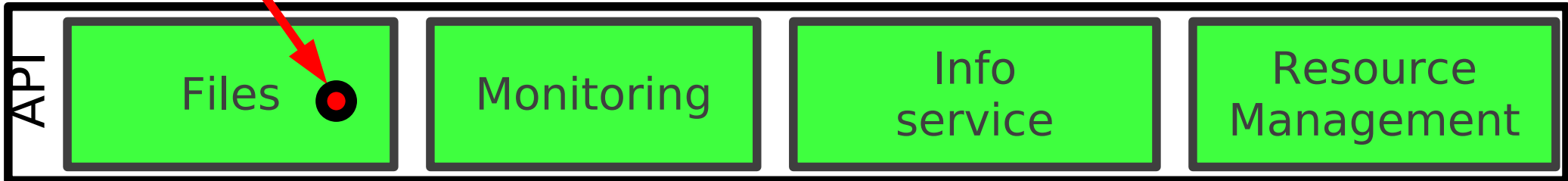
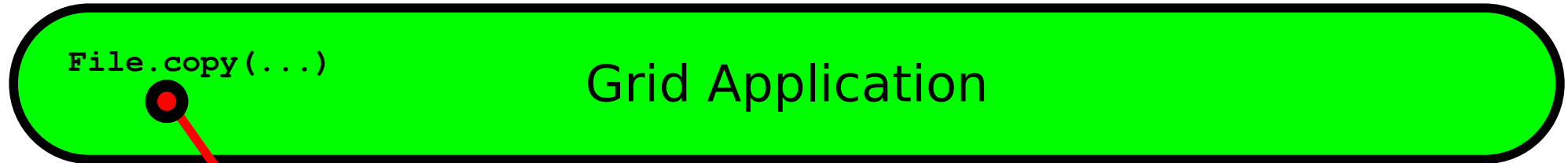
Unicore

SSH

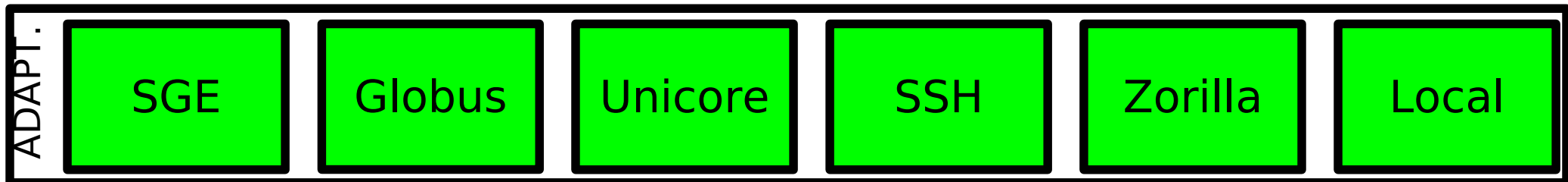
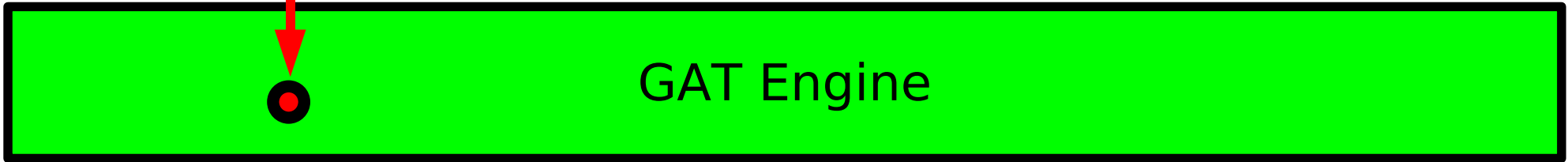
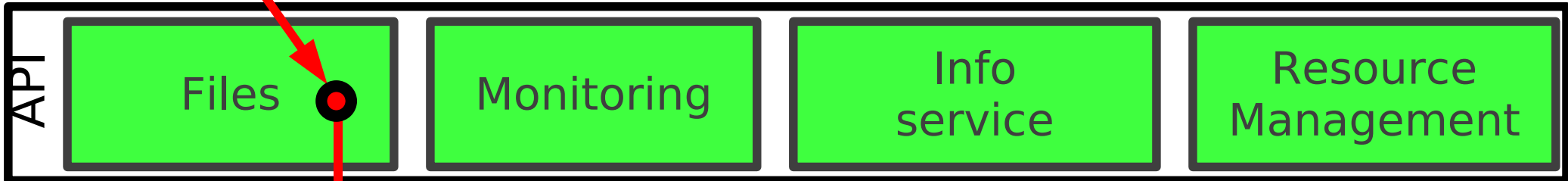
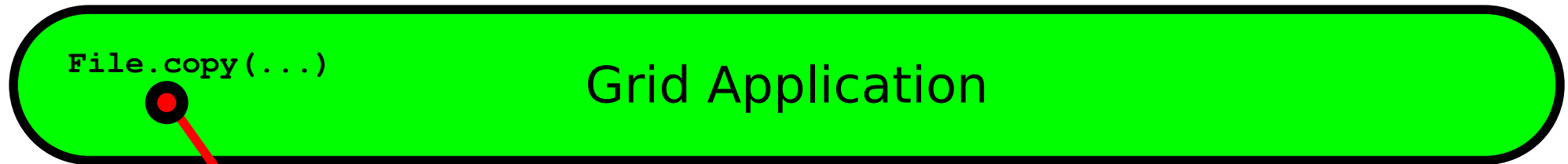
Zorilla

Local

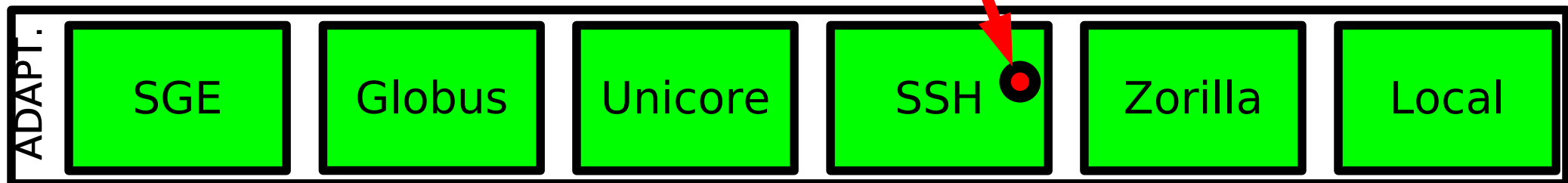
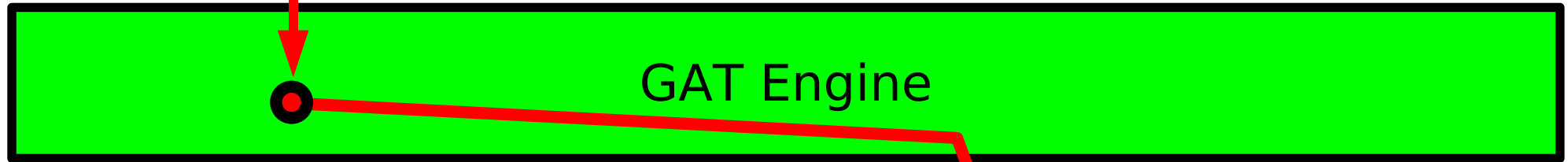
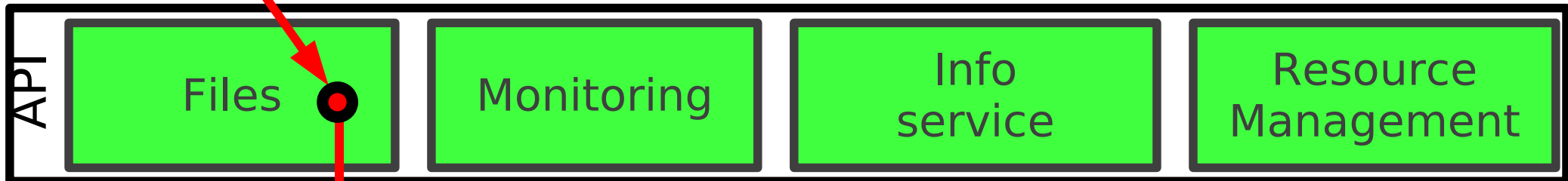
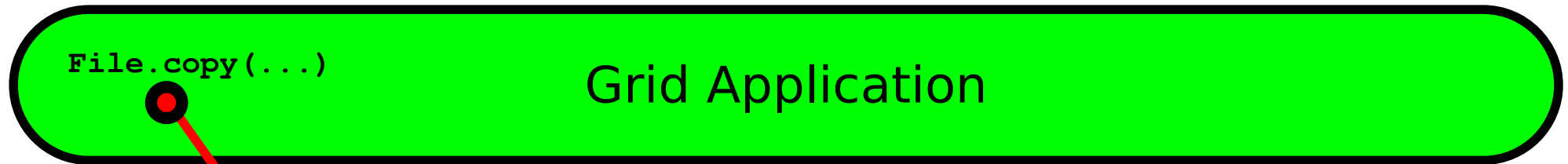
Java GAT Structure



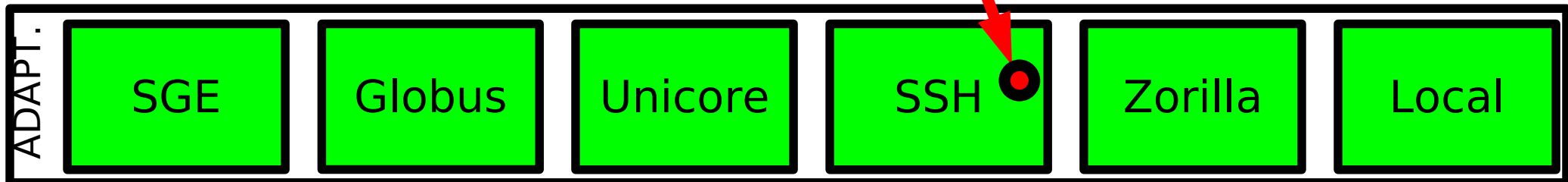
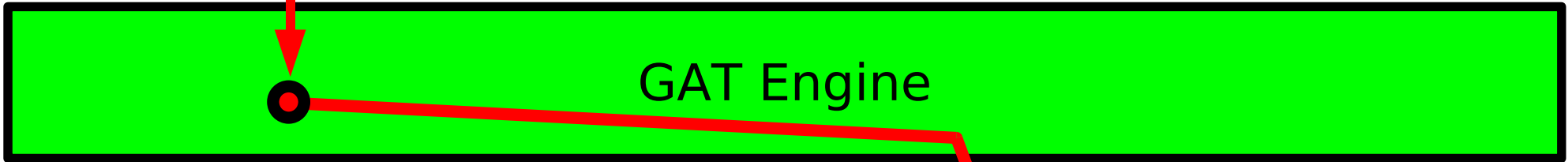
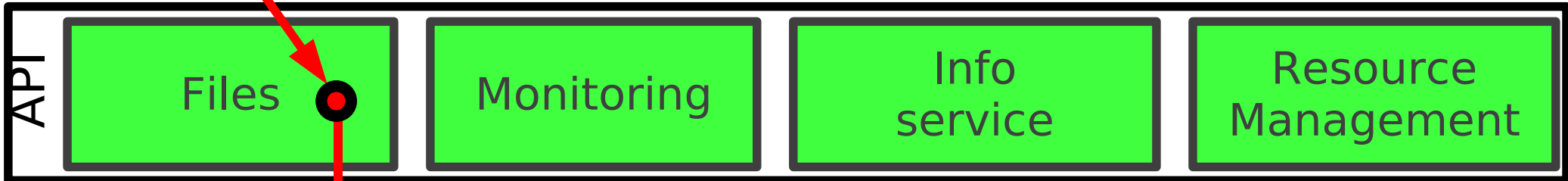
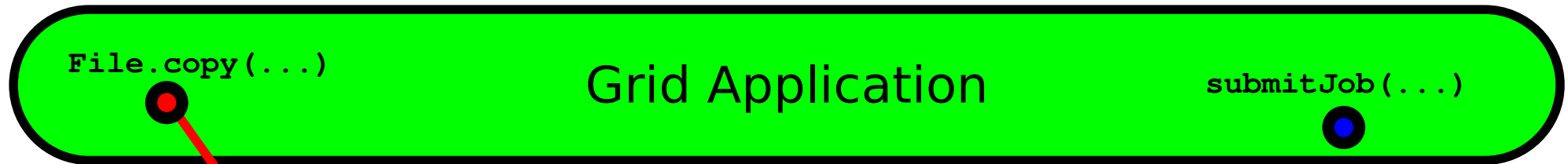
Java GAT Structure



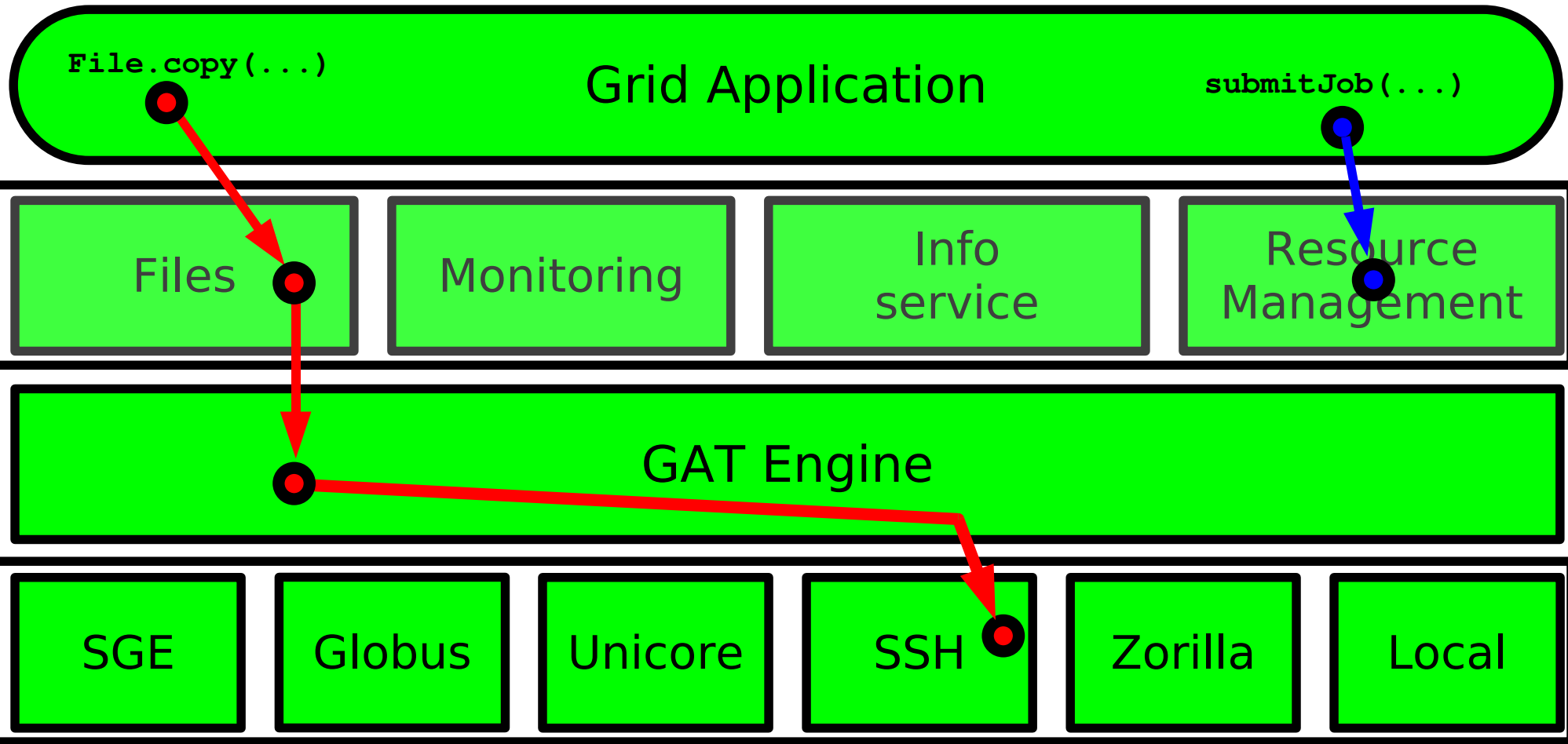
Java GAT Structure



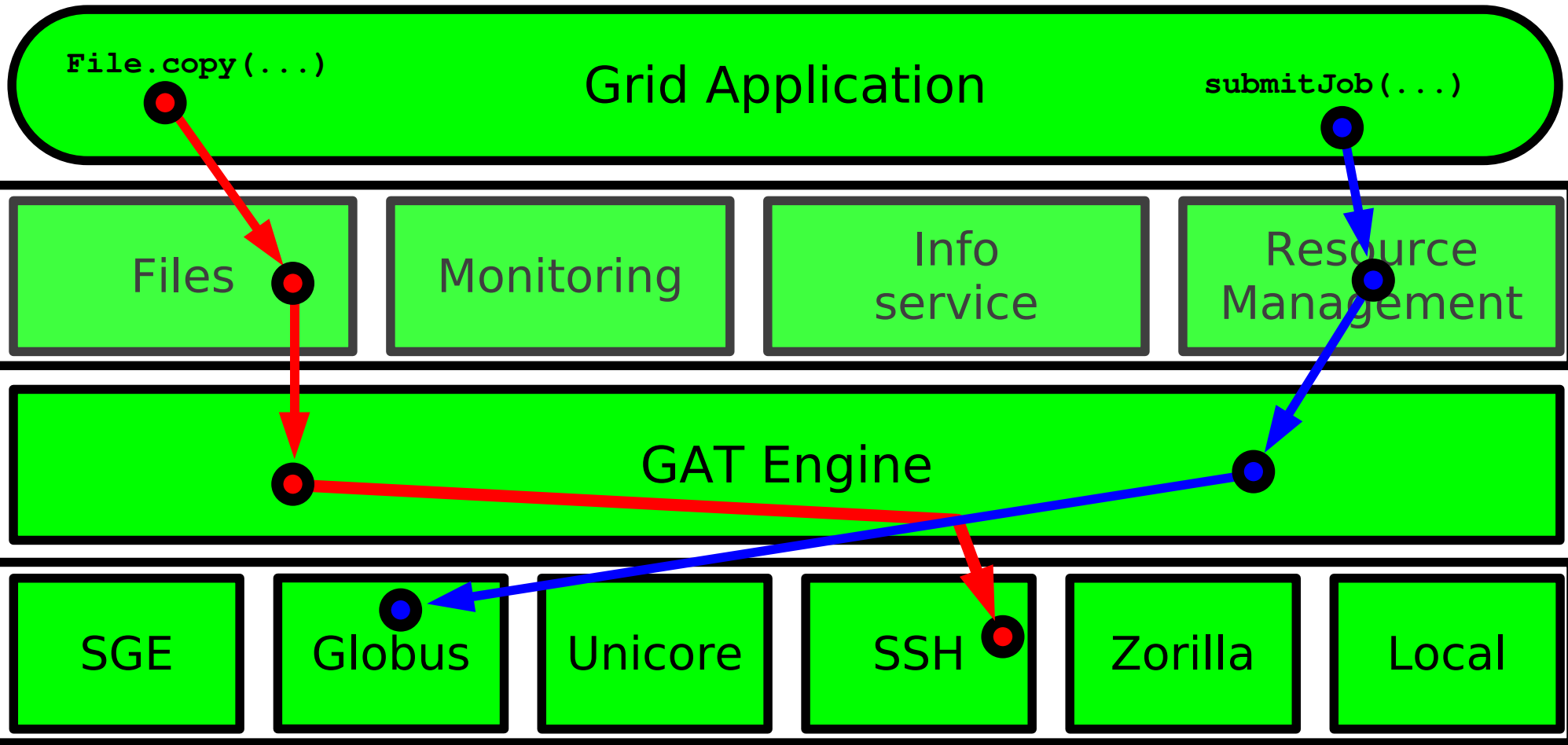
Java GAT Structure



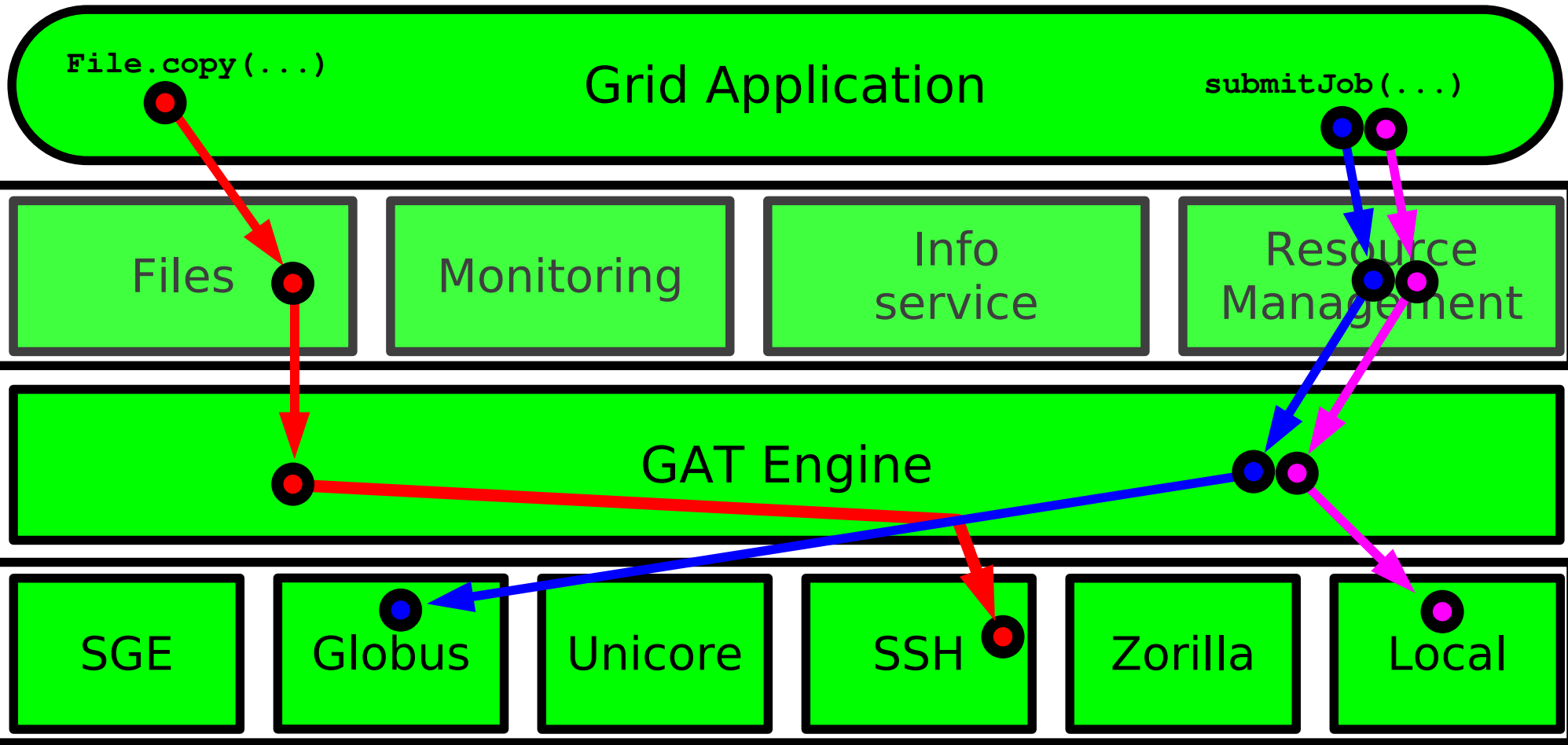
Java GAT Structure



Java GAT Structure



Java GAT Structure



Implementation

- Adaptors are Java JAR files, dynamically loaded into the application.
- Late binding: The GAT engine selects the best adaptor for each **method**.
- Example: Create file object.
- File.copy from site A to site B and C.
 - A -> B copy with GridFTP.
 - A -> C copy with SSH.
- Provides flexibility
- Provides fault tolerance.



Some Basic GAT Objects

- Preferences
 - Key value pairs
 - Adaptor-specific instructions
 - Global or local, local overrides global
 - Example: ("File.adaptor.name", "globus")
- GATContext
 - Contains security information
 - Contains global preferences
 - There can be more than one context



Some Basic GAT Objects

- GAT
 - Factory for all GAT objects
- GAT Exceptions
 - Nested, helps debugging (Needed because of late binding)



URI's in the JavaGAT

- Slightly different semantics compared to java.net.URI
- Use the right number of /'s in the URI's
- Full URI is easy:
`protocol://machine/<path>file`
- But some fields may be blank
- Special “any” scheme to let JavaGAT decide the protocol/adaptor used



URI Examples

- Local file in current directory:
 - file:///output
- Local file in root (/) directory
 - file:///output
- Local file in /tmp directory
 - file:///tmp/output
- Remote file in default ftp dir
 - ftp://10.0.0.1/output
- A remote file
 - any://kits.cs.vu.nl/file.txt



GAT Security

- SecurityContext
 - A container for security Information.
- Abstract, use subclasses
 - PasswordSecurityContext
 - CertificateSecurityContext
 - MyProxyServerCredentialSecurityContext
- Typically not needed if default credentials / private keys are used



GAT Security example

```
GATContext context = new GATContext();
```

```
SecurityContext pwd =  
    new PasswordSecurityContext(username, password);
```

```
SecurityContext cert =  
    new CertificateSecurityContext(keyfile, username,  
    passphrase);
```

```
// add them to the GAT context  
context.addSecurityContext(pwd);  
context.addSecurityContext(cert);
```



GAT API features

- Security
- Grid I/O
- Resource Management
- Application Information Management
- Monitoring



GAT API Features (Cond')

- Application Information Management
- Monitoring



Grid I/O: Use Cases

- Copy, move, read, write files on the grid
- Random access to remote files
- Replicate files between different grid sites
- Inter-process communication



Grid I/O JavaGAT Classes (1)

- File
- FileInputStream / FileOutputStream
- RandomAccessFile

- All extend java.io classes
- Makes it easy to “grid enable” applications by simply using GAT.create() instead of normal constructor



Grid I/O JavaGAT Classes (2)

- LogicalFile
 - Replicated file support
- Basic Inter-process communication
 - Endpoint
 - Pipe
 - PipeListener



GAT File

- Represents both files and directories (like java.io)
 - canRead, canWrite
 - delete
 - mkdir
 - list
 - copy
 - move
 - ...



GAT File Example (again)

```
import org.gridlab.gat.*;
import org.gridlab.gat.io.File;

public class RemoteCopy {
    public static void main(String[] args)
        throws Exception {
        File file = GAT.createFile(new URI(args[0]));
        file.copy(new URI(args[1]));
        GAT.end();
    }
}
```



GAT File Streaming Example

```
package tutorial;

class RemoteCat {
    public static void main(String[] args) throws Exception {
        GATContext context = new GATContext();
        URI loc = new URI(args[0]);

        FileInputStream in = GAT.createFileInputStream(context, loc);
        InputStreamReader reader = new InputStreamReader(in);
        BufferedReader buf = new BufferedReader(reader);

        while(true) {
            String result = buf.readLine();
            if(result == null) break;
            System.out.println(result);
        }
        in.close();
        GAT.end();
    }
}
```



The future of GAT

- GAT will be supported for the foreseeable future
- GAT is being standardized within GGF: SAGA (see next talk)
- Java reference implementation of SAGA is being developed by us
 - Implemented on top of the JavaGAT
- It is currently unclear if SAGA will replace JavaGAT altogether
 - But we will definitely support the JavaGAT API (e.g. on top of SAGA)



JavaGAT users

- Download is anonymous, so we don't know
- Max Planck Institute for Astrophysics in Garching
- D-Grid
- Astrogrid
- Louisiana State University
- University of Texas
- AMOLF, Institute for Atomic and Molecular Physics
- The Dutch Virtual Labs for E-science project (VI-e)
- The workflow system Triana (University of Cardiff)
- Georgia State University
- Vrije Universiteit Amsterdam (Ibis, teaching)
- The Multimedien project
- Zuse Institute Berlin, Germany
- VU Medical Center Amsterdam



Conclusion

- The GAT provides a simple and stable API to various Grid environments
- Powerful
- Independent of grid middleware
- Portable

Downloads: www.cs.vu.nl/ibis

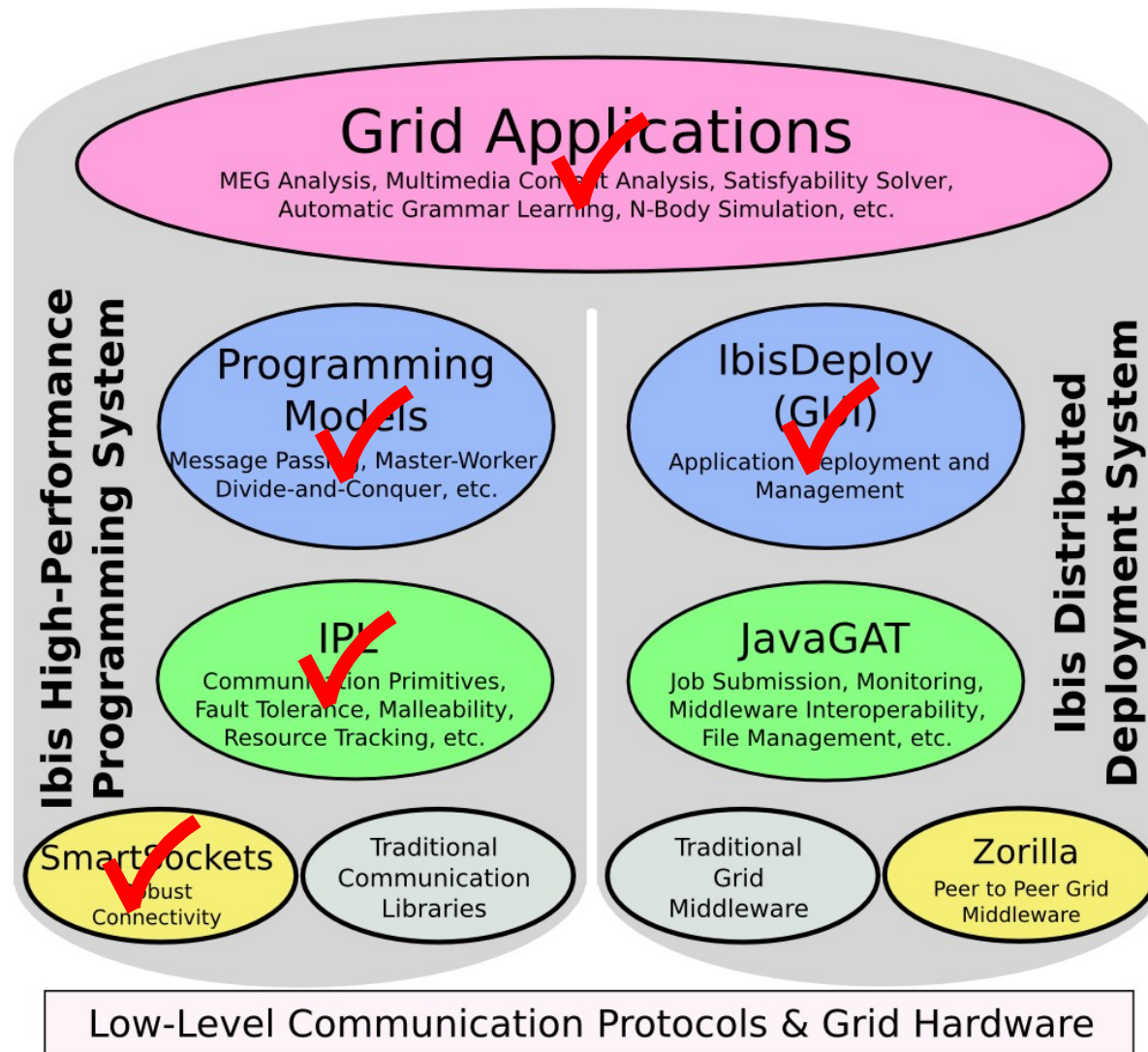


But Wait!

- I don't have any middleware!



Overview



Current Middleware

- Hard to install
- Hard to maintain
- Centralized (not very fault tolerant)
- Usually no Co-Allocation

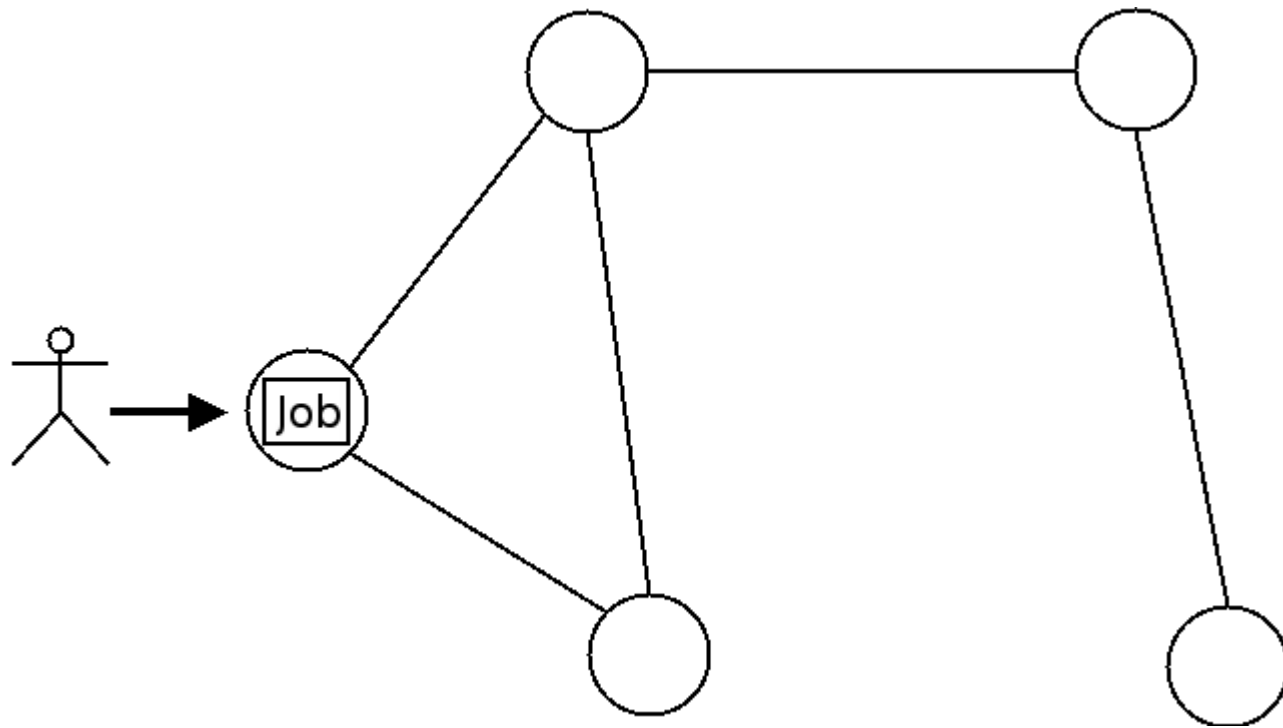


Alternative: Zorilla

- Prototype Java Peer-to-Peer middleware
- Fully distributed
- Easy to setup (just add Java)
- Fault tolerant (no single point of failure)
- P2P: Limited trust/security
- Interface: JavaGAT



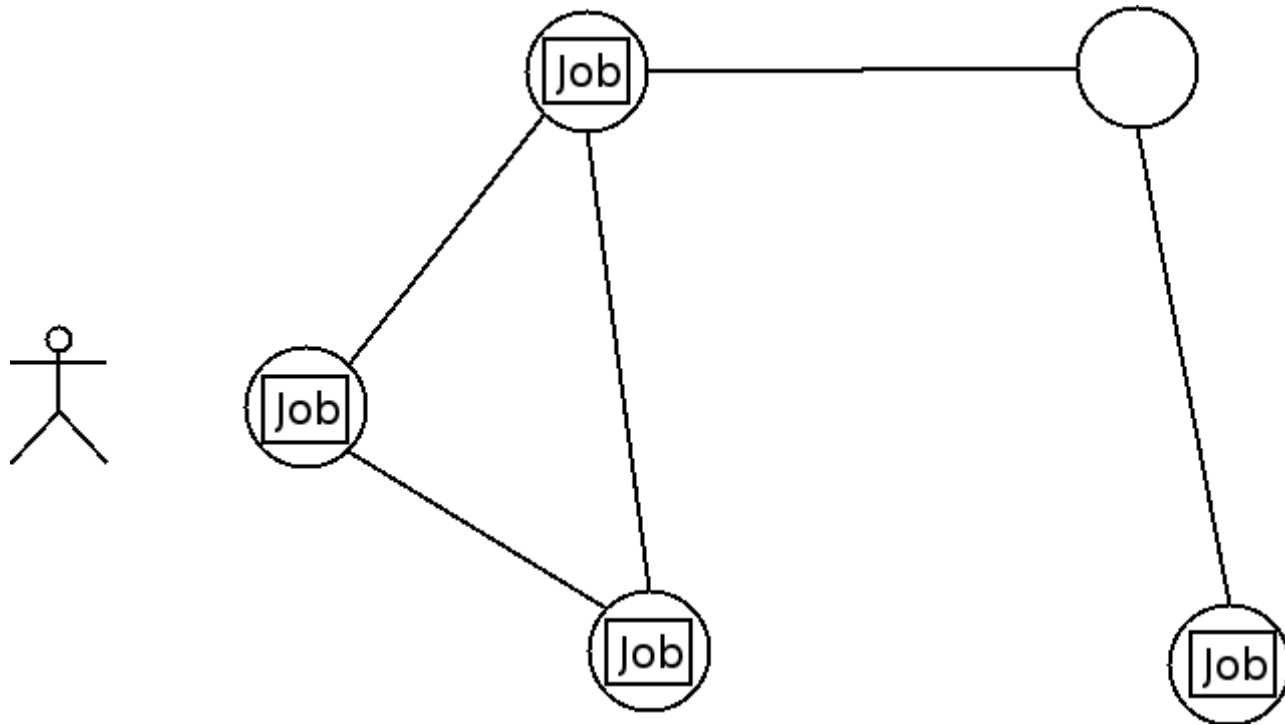
Life of a job (1/4)



- Submit



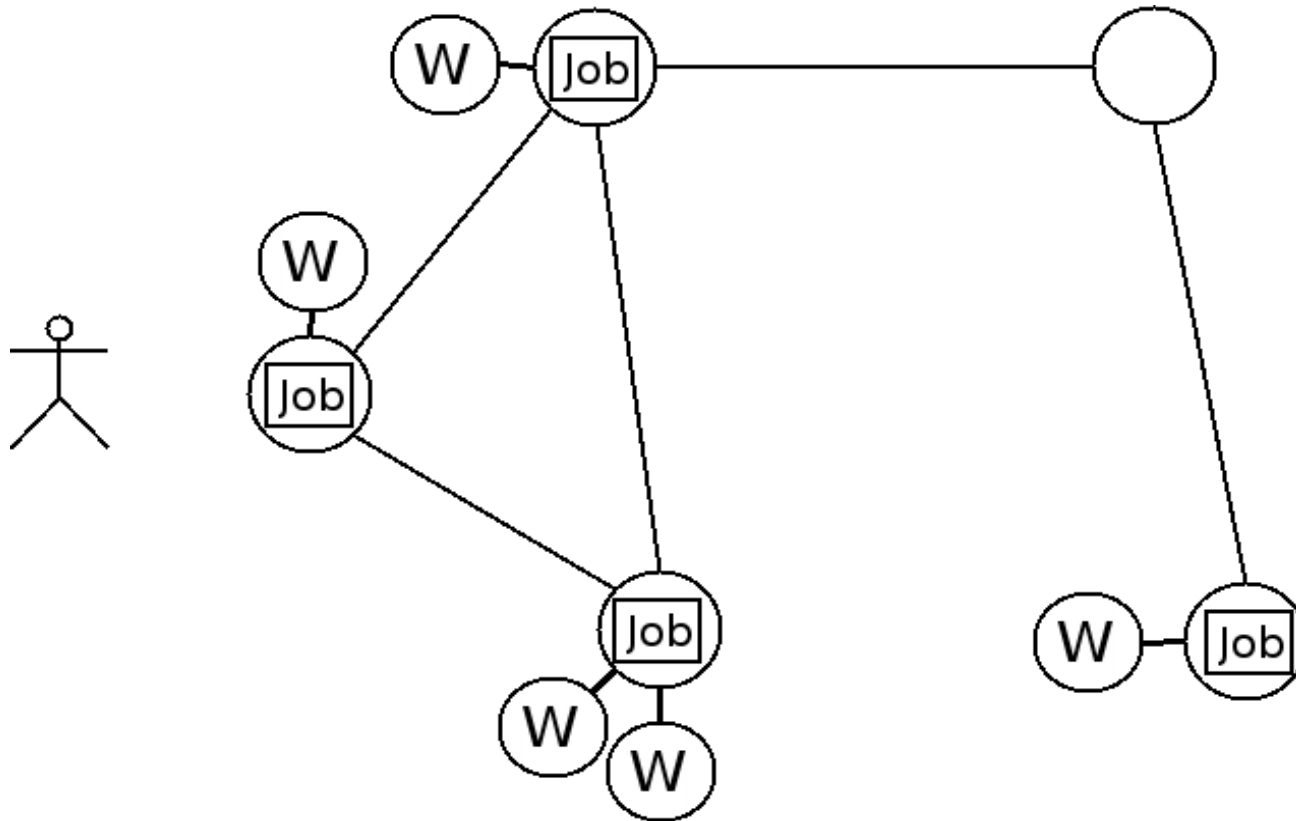
Life of a job (2/4)



- Resource Location



Life of a job (3/4)

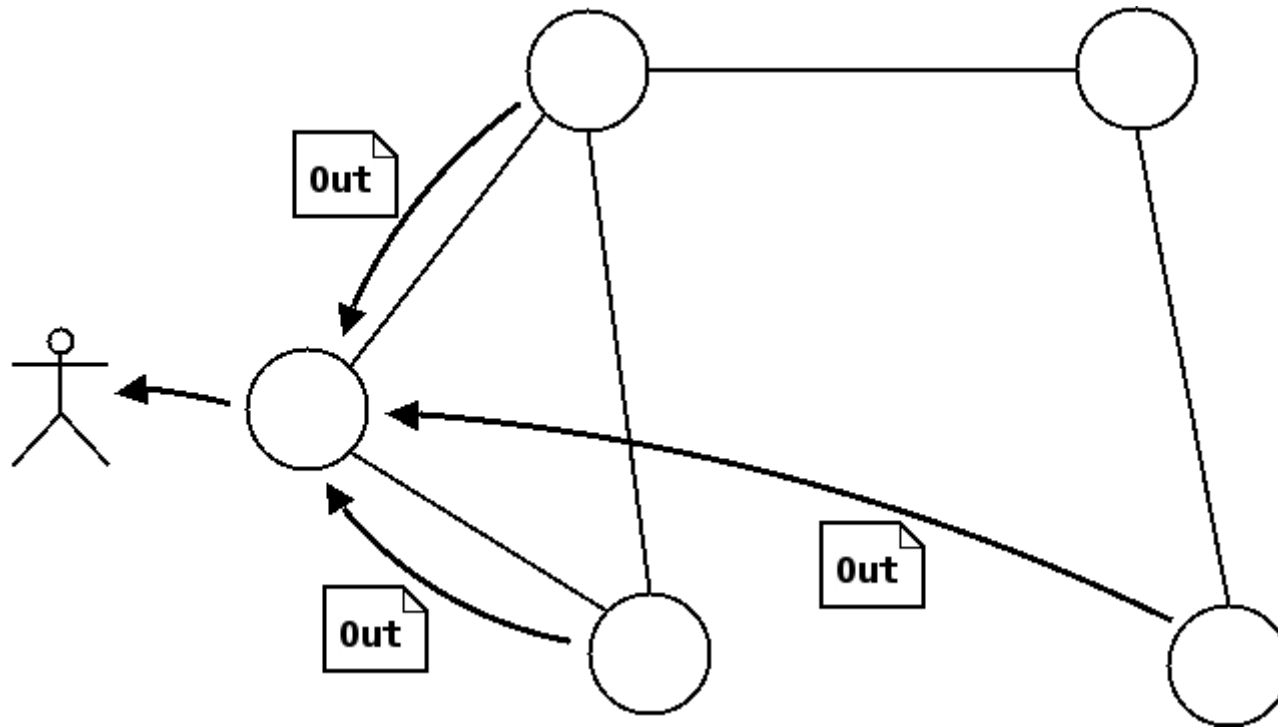


- Workers Started

W = Worker running Application

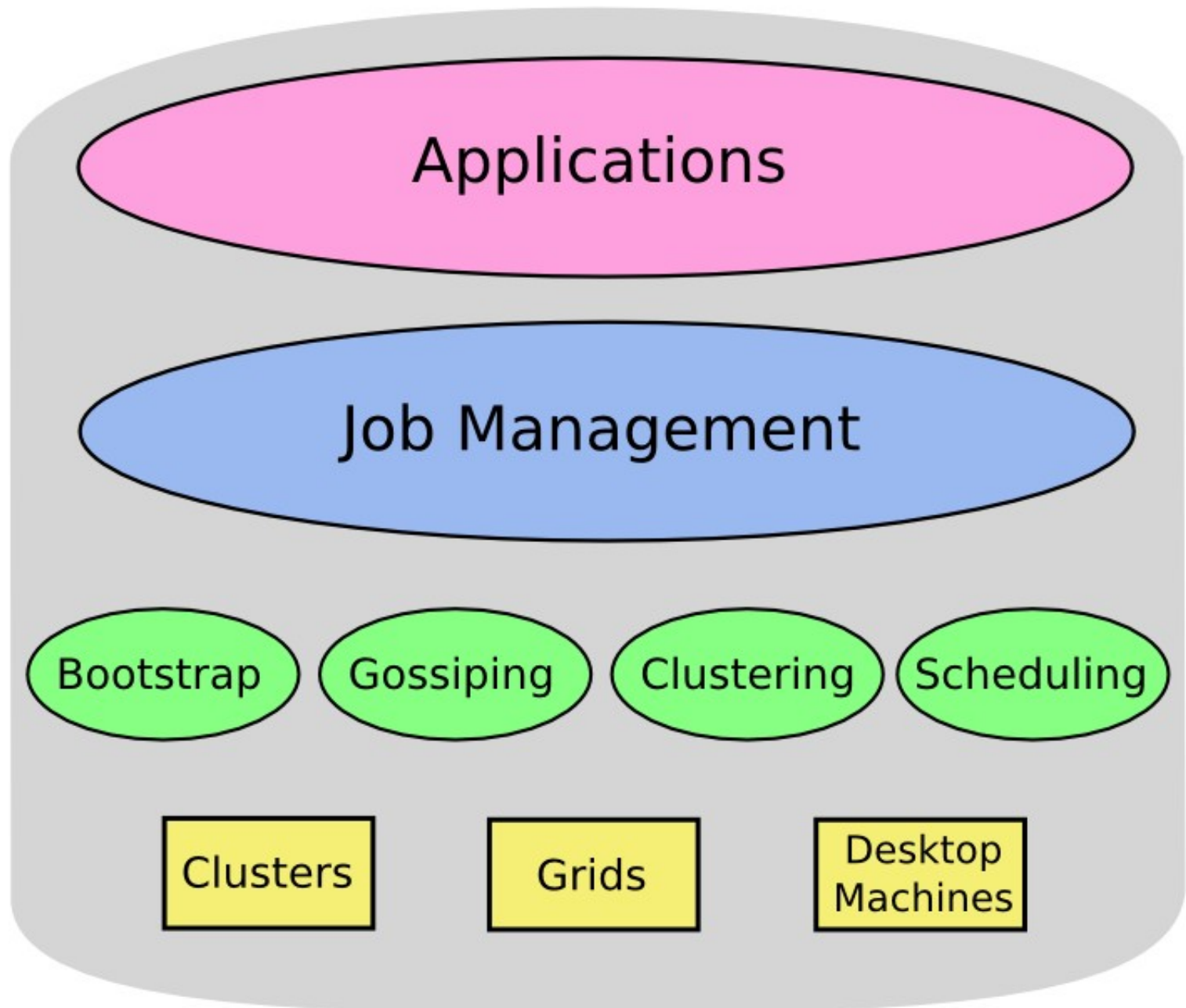


Life of a job (4/4)



- Results returned to user

Zorilla Overview



Zorilla Components

- Bootstrap
 - Initial set of contact points
 - UDP broadcast or provided by user
- Gossip overlay network
 - Actualized Robust Random Gossip (ARRG)
 - Withstands Firewalls et al.
- Clustering
 - Nearest neighbor list



Zorilla Components (2)

- Flood scheduling
 - Incrementally search for resources at more and more distant nodes
- Job Management
 - Status (scheduling, running, done, etc)
 - File transfers
 - Malleability / crashes



Zorilla Usage

- 1) Install recent JVM
- 2) Download Zorilla at
<http://www.cs.vu.nl/ibis>
- 3) Run “zorilla”
(see --help for options)
- 4) Repeat 1-3 for all machines
- 5a) Start application using JavaGAT
- 5b) Start application using “zubmit”

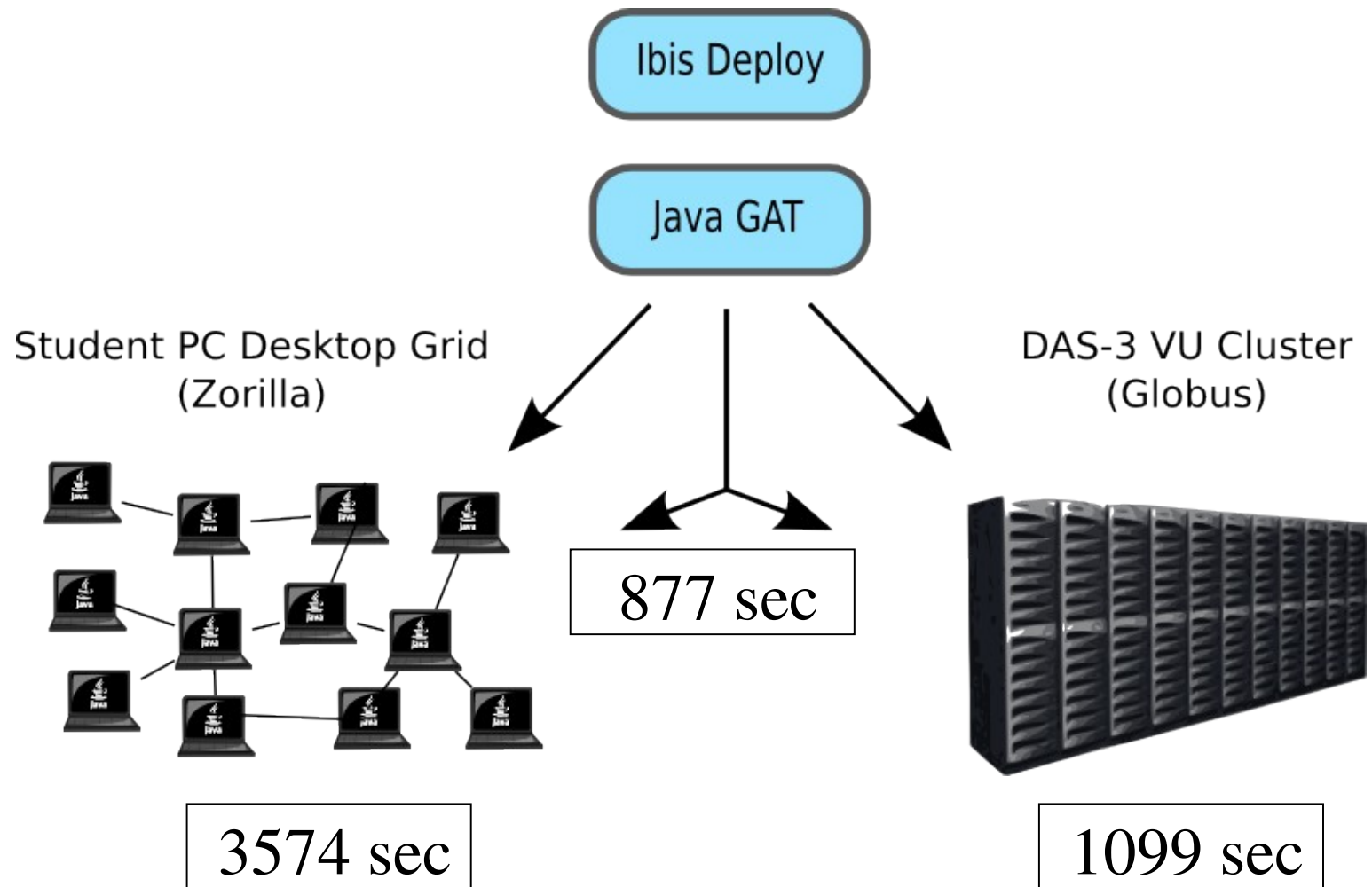


Desktop Grid Experiment

- Small experimental desktop grid setup
 - Student PCs running Zorilla overnight
 - PCs with 1 CPU, 1GB memory, 1Gb/s Ethernet
- Experiment: gene sequence application
 - 16 cores of DAS-3 with Globus
 - 16 core desktop grid with Zorilla
 - Combination, using Ibis-Deploy



Desktop Grid Experiment



Conclusion (Revisited)

- The GAT provides a simple and stable API to various Grid environments
- Powerful
- Independent of grid middleware
- Portable
- If you don't have any middleware, we can supply one :)

