SAGA

A Simple API for Grid Applications

Ole Weidner CCT/LSU

Genève July 22, 2009

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

WHAT IS SAGA

- SAGA is an Open Grid Forum (OGF) specification (GFD.90) for an extensible grid application API. It aims to fill the gap for:
  - Programmatic approaches that provide common grid functionality at an appropriate level of abstraction for application developers
  - Ability to hide underlying complexity of infrastructure, varying semantics, heterogeneity and changes from the application developer
- It provides a *uniform* API that allows application developers to work transparently across different distributed systems
- http://www.ogf.org/documents/GFD.90.pdf

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

WHAT IS SAGA

- SAG … or an exte…
  - Pr… at…
  - A… …antics, he…
- It pr… …work tran…
- http…

```
int copy_file (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);
                                   &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);
```

```
  (source_url.scheme_type == GLOBUS_URL_SCHEME_GSIFTP ||
   source_url.scheme_type == GLOBUS_URL_SCHEME_FTP    ) {
  globus_ftp_client_operationattr_init (&source_ftp_attr);
  globus_gass_copy_attr_set_ftp (&source_gass_copy_attr,
                                 &source_ftp_attr);
}
else {
  globus_gass_transfer_requestattr_init (&source_gass_attr,
                                         source_url.scheme);
  globus_gass_copy_attr_set_gass(&source_gass_copy_attr,
              &source_gass_attr);
}

output_file = globus_libc_open ((char*) target,
              O_WRONLY | O_TRUNC | O_CREAT,
              S_IRUSR  | S_IWUSR | S_IRGRP |
              S_IWGRP);
if ( output_file == -1 ) {
  printf ("could not open the file \"%s\"\n", target);
  return (-1);
}
/* convert stdout to be a globus_io_handle */
if ( globus_io_file_posix_convert (output_file, 0,
                                   &dest_io_handle)
     != GLOBUS_SUCCESS) {
  printf ("Error converting the file handle\n");
  return (-1);
}

result = globus_gass_copy_register_url_to_handle (
         &gass_copy_handle, (char*)source_URL,
         &source_gass_copy_attr, &dest_io_handle,
         my_callback, NULL);
if ( result != GLOBUS_SUCCESS ) {
  printf ("error: %s\n", globus_object_printable_to_string
          (globus_error_get (result)));
  return (-1);
}
globus_url_destroy  (&source_url);
return (0);
}
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

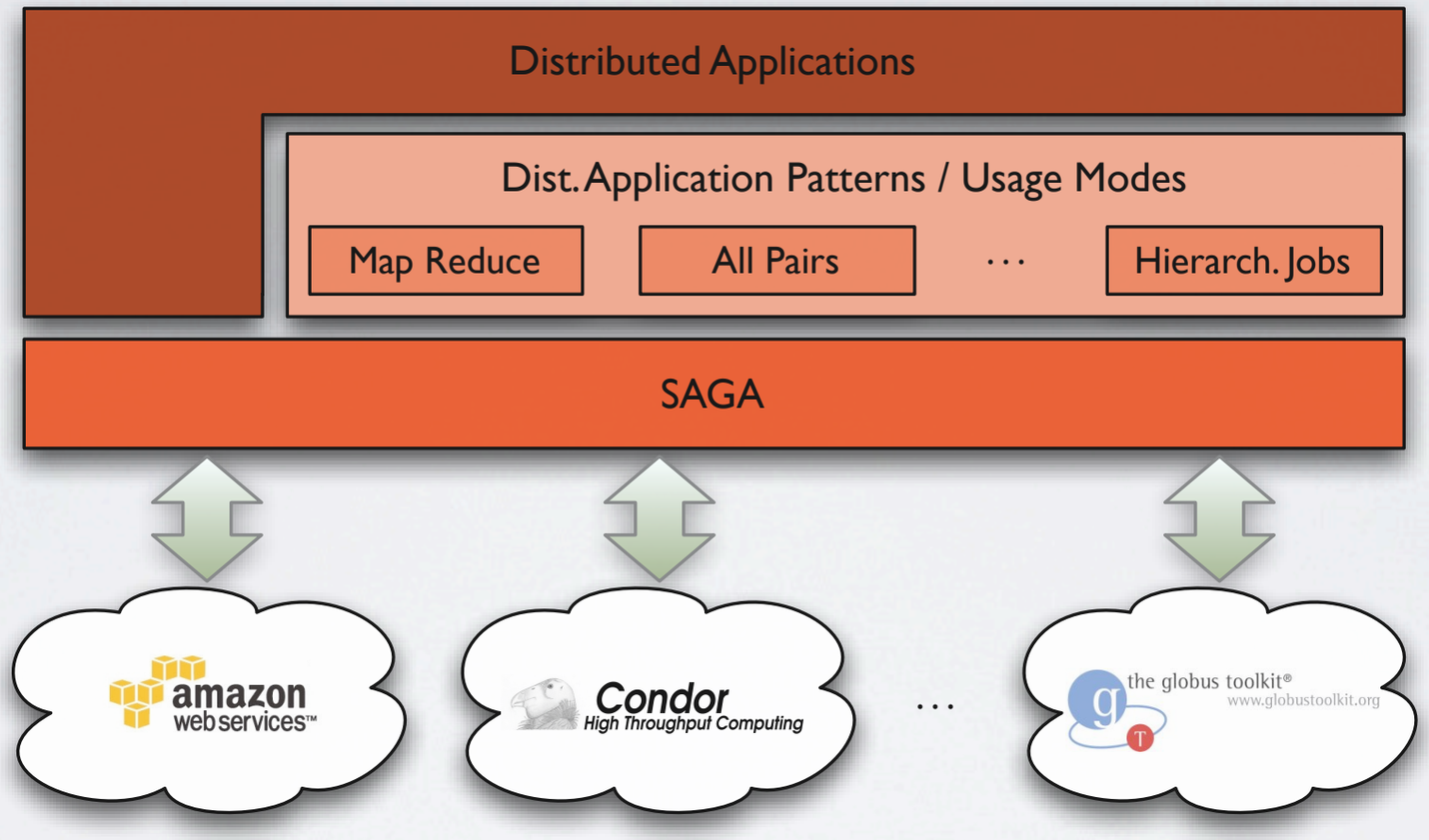
WHAT IS SAGA

- SAGA is an Open Grid Forum (OGF) specification (GFD.90) for an extensible grid application API. It aims to fill the gap for:
  - Programmatic approaches that provide common grid functionality at an appropriate level of abstraction for application developers
  - Ability to hide underlying complexity of infrastructure, varying semantics, heterogeneity and changes from the application developer
- It provides a *uniform* API that allows application developers to work transparently across different distributed systems
- http://www.ogf.org/documents/GFD.90.pdf

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

USAGE SCENARIOS



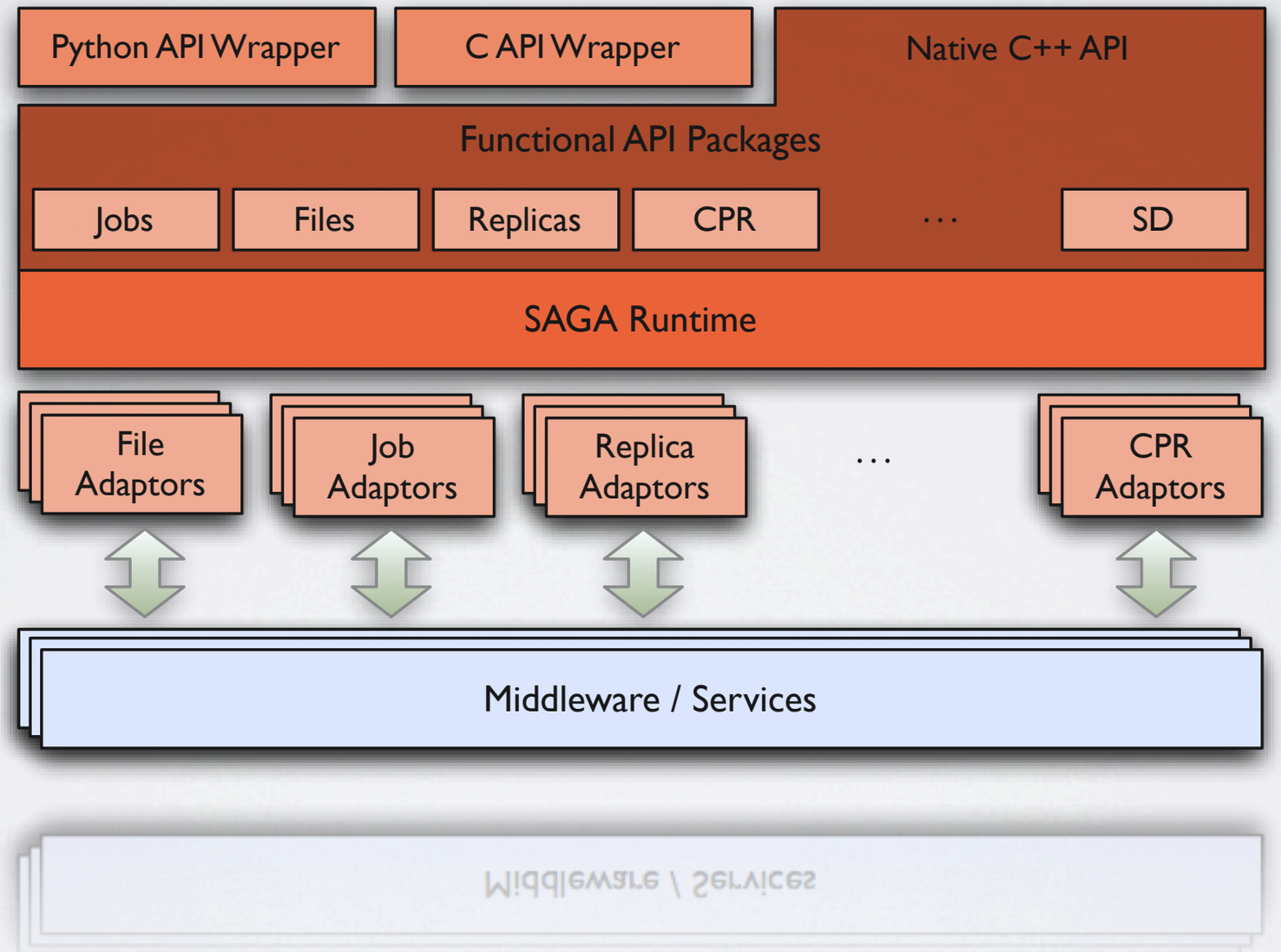
IMPLEMENTATIONS

- Currently there are several implementations under active development:
  - C++ Reference Implementation (LSU), funded by OMII-UK http://saga.cct.lsu.edu/cpp/
  - Java Implementation (VU Amsterdam), part of the OMII-UK project http://saga.cct.lsu.edu/java/
  - JSAGA (IN2P3/CNRS) http://grid.in2p3.fr/jsaga/
  - NARGI/KEK (?)

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA C++ ARCHITECTURE



SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA C++ QUICKTOUR

- Open Source - released under the Boost Software License 1.0
- Implemented as a set of libraries
  - SAGA Core - A light-weight engine / runtime that dispatches calls from the API to the appropriate middle-ware adaptors
  - SAGA functional packages - Groups of API calls for: jobs, files, service discovery, advert services, RPC, replicas, CPR, ... (extensible)
  - SAGA language wrappers - Thin Python and C layers on top of the native C++ API
  - SAGA middle-ware adaptors - Take care of the API call execution on the middle-ware
- Can be configured / packaged to suit your individual needs!

AVAILABLE ADAPTORS

- Job Adaptors
  - Fork (localhost), SSH, Condor, Globus GRAM2, OMII GridSAM, Amazon EC2, Platform LSF
- File Adaptors
  - Local FS, Globus GridFTP, Hadoop Distributed Filesystem (HDFS), CloudStore KFS, OpenCloud Sector-Sphere
- Replica Adaptors
  - PostgreSQL/SQLite3, Globus RLS
- Advert Adaptors
  - PostgreSQL/SQLite3, Hadoop H-Base, Hypertable

AVAILABLE ADAPTORS CONT.

- Other Adaptors
  - Default RPC / Stream / SD
- Planned Adaptors
  - CURL file adaptor, gLite job adaptor

- Open issues
  - We're in the process of consolidating the adaptor code base and adding rigorous tests in order to improve adaptor quality
  - Capability Provider Interface (CPI - the 'Adaptor API') is not documented or standardized (yet), but looking at existing adaptor code should get you started if you want to develop your own adaptor.

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA JOB INTERFACE C++

```
try {

    saga::url js_url (“gram://gatekeeper.lonestar.tacc.teragrid.org:2119/jobmanager-lsf”);

    saga::job::description jd;
    jd.set_attribute (attributes::description_executable, “/home/oweidner/tests/heat_transfer”);
    jd.set_attribute (attributes::description_number_of_processes, "2");
    jd.set_attribute (attributes::description_queue, "checkpoint");

    saga::job::service js (js_url);
    saga::job::job my_job = js.create_job (jd);

    my_job.run();

    std::cout << "Job ID    : " << my_job.get_job_id() << std::endl;
    std::cout << “Job STATE : " << my_job.get_state() << std::endl;

    my_job.suspend();
    my_job.resume();

    my_job.cancel();
}

catch (saga::exception const & e) {

    std::cerr << “Ooops: “ << e.what() << std::endl;
}
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA JOB INTERFACE PYTHON

```
import saga

try:

    js_url = saga.url("gram://gatekeeper.lonestar.tacc.teragrid.org:2119/jobmanager-lsf")

    jd = saga.job.description()
    jd.executable = “/home/oweidner/tests/heat_transfer"
    jd.number_of_processes = 2
    jd.queue = “checkpoint”

    js = saga.job.service(js_url)
    my_job = js.create_job(job_desc)

    my_job.run()

    print "Job ID    : " + my_job.get_job_id()
    print “Job STATE : " + my_job.get_state()

    my_job.suspend()
    my_job.resume()

    my_job.cancel()

except saga.exception, e:

     print “Ooops: “ + str(e)
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA FILE INTERFACE C++

```
try {

    saga::url file_url (“gsiftp://queenbee.loni-lsu.teragrid.org:2811//home/oweidner/.bashrc”);
    saga::filesystem::file f (file_url, saga::filesystem::Read);

    while ( true )
    {
        saga::size_t const n = 1024*64;
        saga::uint8_t data_buf[n+1];
        for ( unsigned int i = 0; i <= n; ++i ) { data_buf[i] = '\0'; }

        if ( f.read (saga::buffer (data_buf, n), n) )
        {
            // output buffer content
            std::cout << data_buf << std::flush;
        }
        else
        {
            break;
        }
    }
}
catch (saga::exception const & e) {

    std::cerr << “Ooops: “ << e.what() << std::endl;
}
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA FILE INTERFACE C++

```
int copy_file (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);
                                   &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);
```

```
 (source_url.scheme_type == GLOBUS_URL_SCHEME_GSIFTP ||
  source_url.scheme_type == GLOBUS_URL_SCHEME_FTP    ) {
  globus_ftp_client_operationattr_init (&source_ftp_attr);
  globus_gass_copy_attr_set_ftp (&source_gass_copy_attr,
                                 &source_ftp_attr);
}
else {
  globus_gass_transfer_requestattr_init (&source_gass_attr,
                                         source_url.scheme);
  globus_gass_copy_attr_set_gass(&source_gass_copy_attr,
              &source_gass_attr);
}

output_file = globus_libc_open ((char*) target,
              O_WRONLY | O_TRUNC | O_CREAT,
              S_IRUSR  | S_IWUSR | S_IRGRP |
              S_IWGRP);
if ( output_file == -1 ) {
  printf ("could not open the file \"%s\"\n", target);
  return (-1);
}
/* convert stdout to be a globus_io_handle */
if ( globus_io_file_posix_convert (output_file, 0,
                                   &dest_io_handle)
     != GLOBUS_SUCCESS) {
  printf ("Error converting the file handle\n");
  return (-1);
}

result = globus_gass_copy_register_url_to_handle (
         &gass_copy_handle, (char*)source_URL,
         &source_gass_copy_attr, &dest_io_handle,
         my_callback, NULL);
if ( result != GLOBUS_SUCCESS ) {
  printf ("error: %s\n", globus_object_printable_to_string
          (globus_error_get (result)));
  return (-1);
}
globus_url_destroy  (&source_url);
return (0);
}
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

SAGA FILE INTERFACE C++

```
try {

    saga::url file_url (“gsiftp://queenbee.loni-lsu.teragrid.org:2811//home/oweidner/.bashrc”);
    saga::filesystem::file f (file_url, saga::filesystem::Read);

    while ( true )
    {
        saga::size_t const n = 1024*64;
        saga::uint8_t data_buf[n+1];
        for ( unsigned int i = 0; i <= n; ++i ) { data_buf[i] = '\0'; }

        if ( f.read (saga::buffer (data_buf, n), n) )
        {
            // output buffer content
            std::cout << data_buf << std::flush;
        }
        else
        {
            break;
        }
    }
}
catch (saga::exception const & e) {

    std::cerr << “Ooops: “ << e.what() << std::endl;
}
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

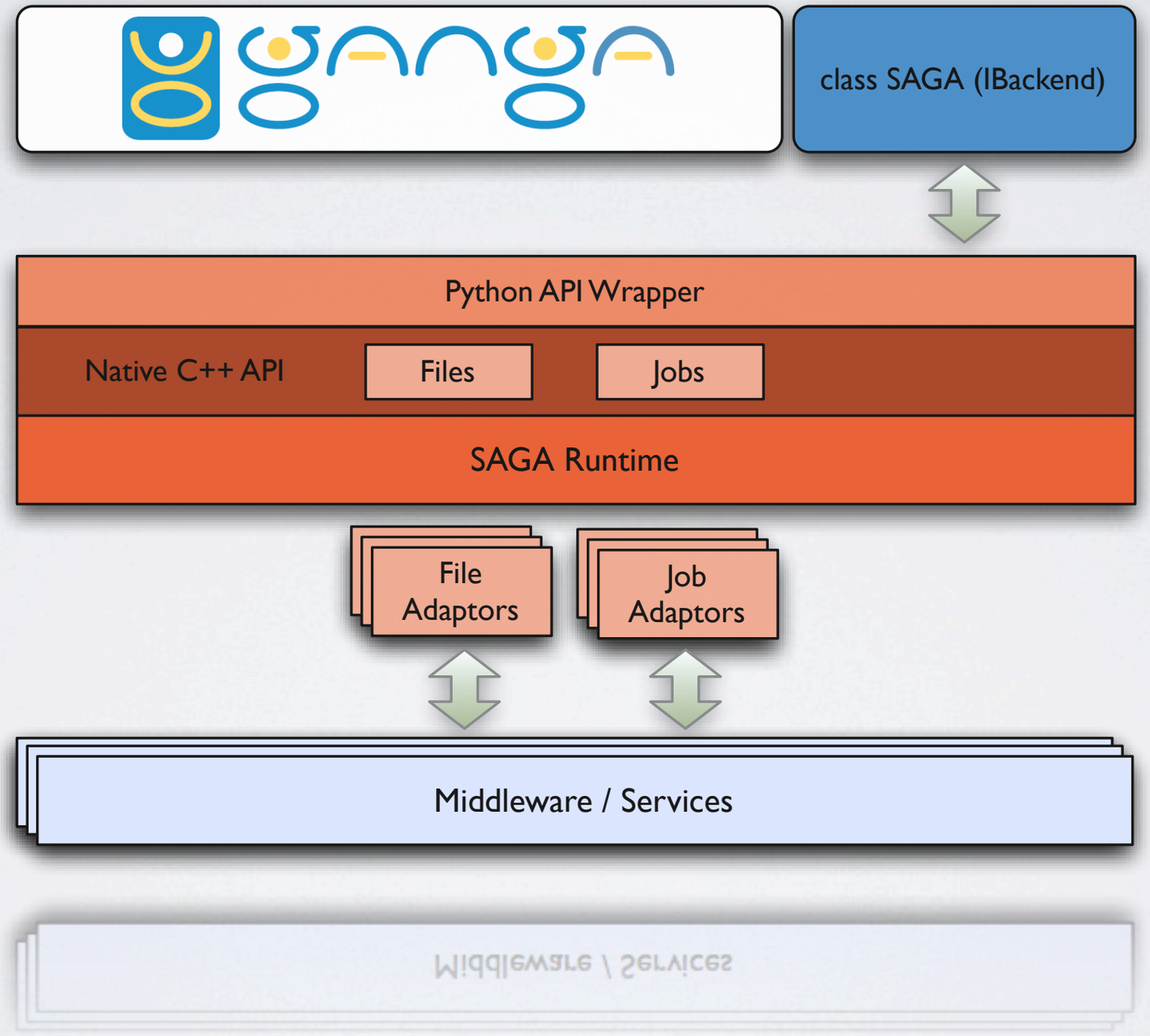
SAGA FILE INTERFACE PYTHON

```
import saga

try:

    file_url = saga.url("gsiftp://queenbee.loni-lsu.teragrid.org:2811//home/oweidner/.bashrc")

    f = saga.file.file(file_url)
    s = f.read()

    print s

except saga.exception, e:

    print “Ooops: “ + str(e)
```

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

GANGA INTEGRATION



SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

GANGA INTEGRATION CONT.

- SAGA job (for job control and monitoring) and file (input / output staging) API calls are wrapped into a ganga IBackend class
- Additional job attributes need to be defined for this backend and translated into a SAGA job description
  - rm_contact, queue, project_id, and other JSDL attributes
  - All of these attributes can be predefined (or stored in a configuration file or job template) for a specific ganga use-case / environment

SAGA

A Simple API for Grid Applications [http://saga.cct.lsu.edu]

GANGA INTEGRATION CONT.

- A first prototype is already available
- It works out-of-the-box as long as SAGA and the Python bindings are installed (and in your PYTONPATH). You also need to have a valid TeraGrid X.509 proxy certificate.

- EXAMPLE: submitting a bunch of jobs through ganga to three different TeraGrid resources:
  - DTF - San Diego Supercomputing Center (SDSC)
  - Ranger - Texas Advanced Computing Center (TACC)
  - QueenBee - Louisiana Optical Network Initiative (LONI)

NEXT STEPS: SAGA/GLITE

- Development of a SAGA gLite adaptor (which can then be used through ganga ;-)
- Many open issues:
  - WSDL (gSoap vs. AXIS) vs. CMD line tools
  - Is it possible to have a SAGA/gLite adaptor without the whole gLite stack installed on your system
  - How to handle gLite security (saga::context ?)
  - CREAM ?
  - etc.

THANKS

Questions / Comments ?