

SAGA: Software Providers Track

Shantenu Jha, Thilo Kielmann and Andre Merzky

OGF IPR Policies Apply

- “I acknowledge that participation in this meeting is subject to the OGF Intellectual Property Policy.”
- **Intellectual Property Notices Note Well:** All statements related to the activities of the OGF and addressed to the OGF are subject to all provisions of Appendix B of GFD-C.1, which grants to the OGF and its participants certain licenses and rights in such statements. Such statements include verbal statements in OGF meetings, as well as written and electronic communications made at any time or place, which are addressed to:
 - the OGF plenary session,
 - any OGF working group or portion thereof,
 - the OGF Board of Directors, the GFSG, or any member thereof on behalf of the OGF,
 - the ADCOM, or any member thereof on behalf of the ADCOM,
 - any OGF mailing list, including any group list, or any other list functioning under OGF auspices,
 - the OGF Editor or the document authoring and review process
- Statements made outside of a OGF meeting, mailing list or other function, that are clearly not intended to be input to an OGF activity, group or function, are not subject to these provisions.
- Excerpt from Appendix B of GFD-C.1: “Where the OGF knows of rights, or claimed rights, the OGF secretariat shall attempt to obtain from the claimant of such rights, a written assurance that upon approval by the GFSG of the relevant OGF document(s), any party will be able to obtain the right to implement, use and distribute the technology or works when implementing, using or distributing technology based upon the specific specification(s) under openly specified, reasonable, non-discriminatory terms. The working group or research group proposing the use of the technology with respect to which the proprietary rights are claimed may assist the OGF secretariat in this effort. The results of this procedure shall not affect advancement of document, except that the GFSG may defer approval where a delay may facilitate the obtaining of such assurances. The results will, however, be recorded by the OGF Secretariat, and made available. The GFSG may also direct that a summary of the results be included in any GFD published containing the specification.”
- OGF Intellectual Property Policies are adapted from the IETF Intellectual Property Policies that support the Internet Standards Process.

Outline

- SAGA in a Nutshell
- Application-level Interoperability
- SAGA: Some Practical Details..
- SAGA Overview-II
 - Code examples, Functional-package
- Software development
 - C++ (Andre)
 - Java (Thilo)

SAGA in a Nutshell



- Why are there so few grid applications out there?
- A lack of **simple, stable, integrated** and **uniform high-level** programming interface that provides the most common grid programming abstractions?
- Need to hide underlying complexities, varying semantics, heterogenities and changes from application program(er)
- Measure(s) of success:
 - Does SAGA enable quick development of “new” grid applications?
 - Does it enable greater functionality using less code?

Copy a File: SAGA

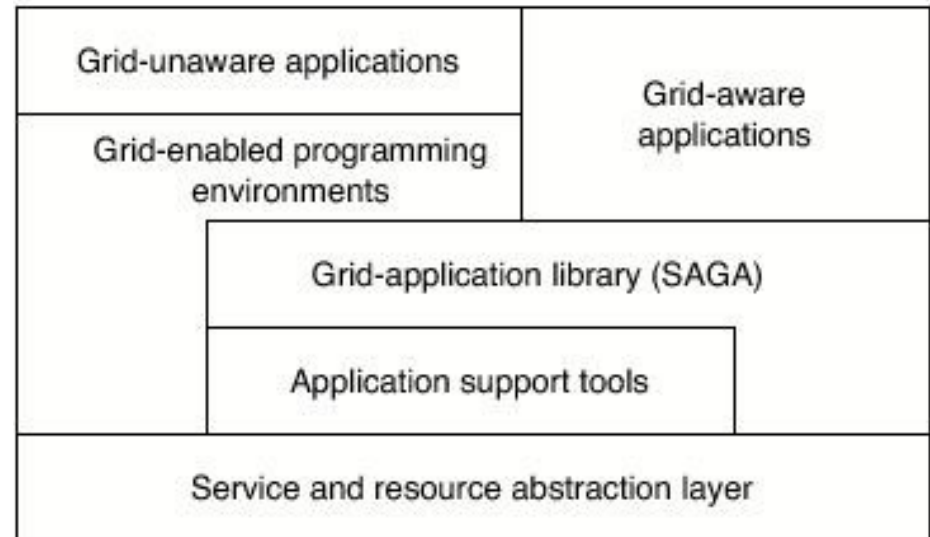


```
#include <string>
#include <saga/saga.hpp>

void copy_file(std::string source_url, std::string target_url)
{
    try {
        saga::file f(source_url);
        f.copy(target_url);
    }
    catch (saga::exception const &e) {
        std::cerr << e.what() << std::endl;
    }
}
```

- Provides the high level abstraction layer, that application programmers need
- Like MapReduce – leave details of distribution *etc.* out
- Shields details of lower-level m/w system

Rough Taxonomy of Applications



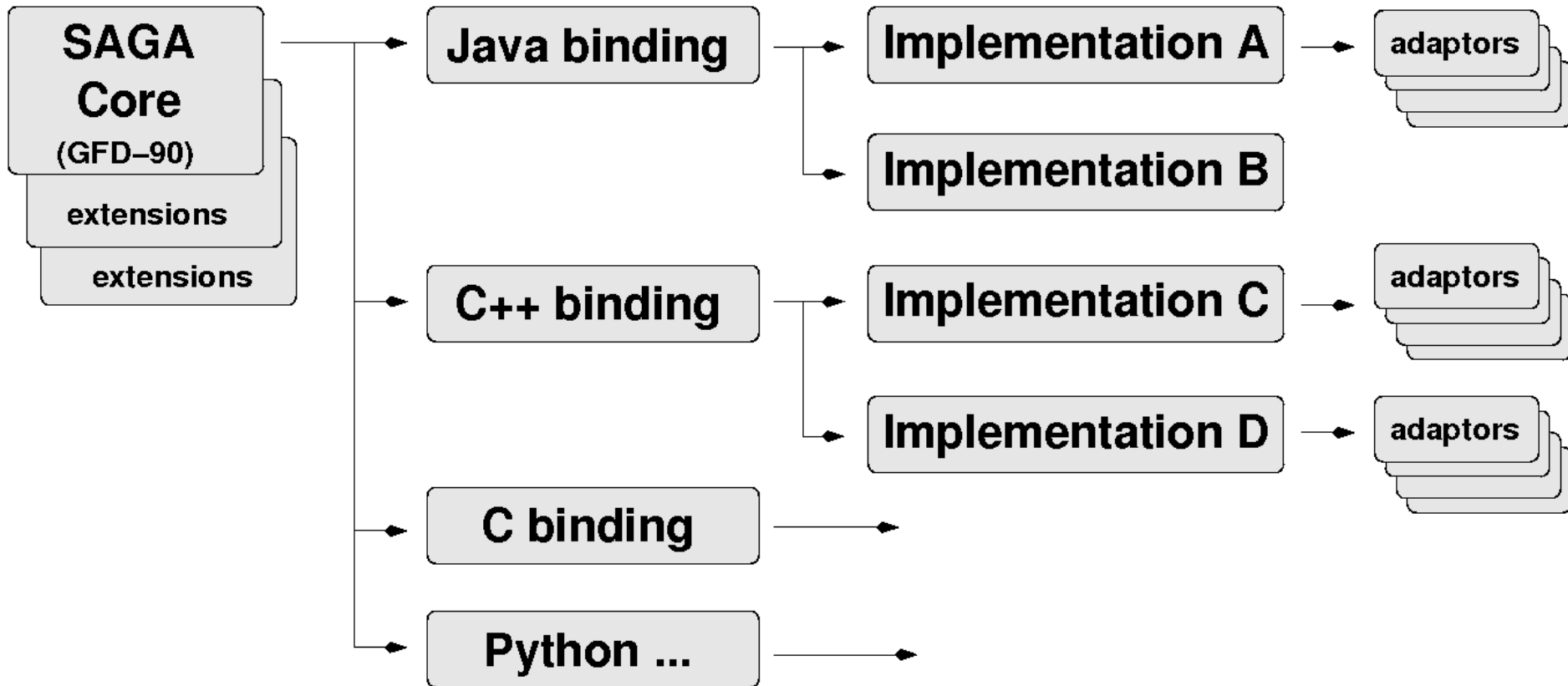
- Some applications are Grid-unaware and want to remain so
 - Use tools/environments (e.g, NanoHub, GridChem)
 - May run on Grid-aware/Grid-enabled environments (e.g. Condor) or programming environment (e.g, MPICH-G2)
- Some applications are explicitly Grid-aware
 - Control, Interact & Exploit distributed systems at the *application level*

Application Level Interoperability

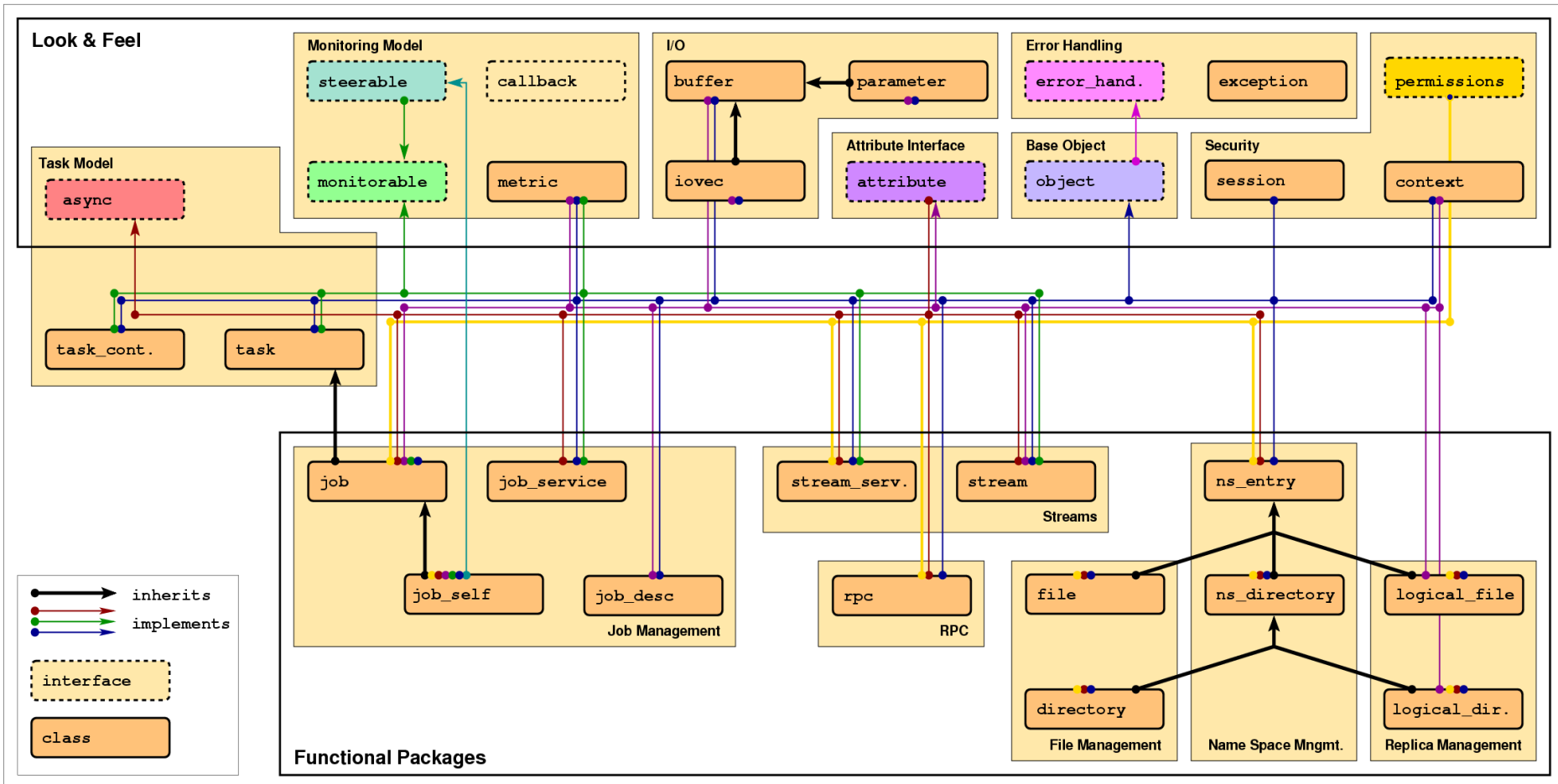


- “Service Level Interoperability”
 - A Bottom-up versus top-down approach
 - “create the infrastructure; then get the applications”
- Specific Applications
- Application Level Interoperability:
 - Other than compiling on a different or new platform, there are no further changes required of the application
 - Automated, scalable and extensible solution to use new resources, and not via bilateral or customized arrangements
 - Semantics of any services that an application depends upon are consistent and similar, eg error handling

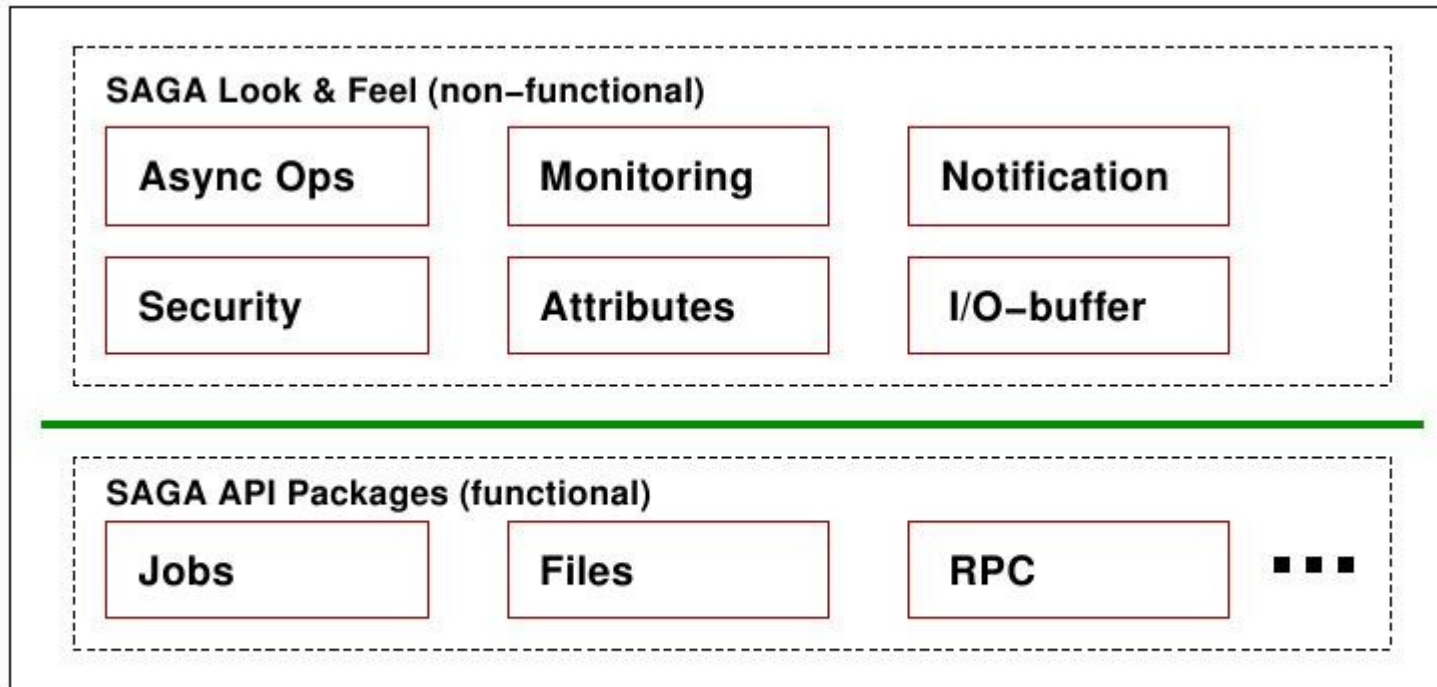
The SAGA Landscape



SAGA v1.0 (almost) – Full Glory



Packages: Functional vs Non-functional

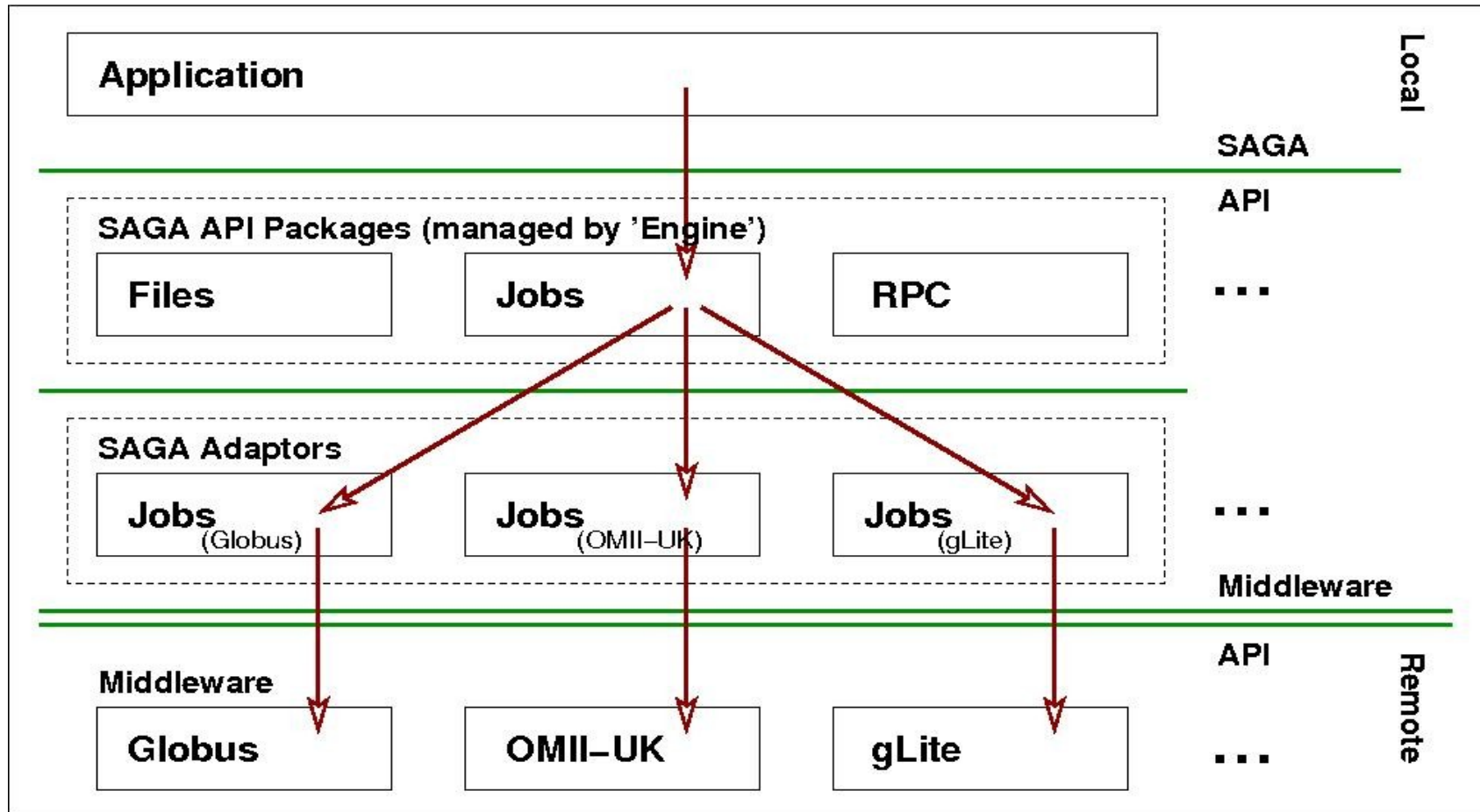


- Stable Non-functional; extensible functional packages
- Functional use non-functional packages
- Packages define one or more related classes; partial implementations implement sub-set of packages

Job Submission API

```
01: // Submitting a simple job and wait for completion
02: //
03: saga::job_description jobdef;
04: jobdef.set_attribute ("Executable", "job.sh");
05:
06: saga::job_service js;
07: saga::job job = js.create_job ("remote.host.net", jobdef);
08:
09: job.run();
10:
11: while( job.get_state() == saga::job::Running )
12: {
13:     std::cout << "Job running with ID: "
14:               << job.get_attribute("JobID") << std::endl;
15:     sleep(1);
16: }
```

SAGA: In action



- file management, job management, remote procedure calls, replica management, data streaming

SAGA Implementation: Some Req



- Non-trivial set of requirements:
 - Allow heterogenous middleware to co-exist
 - Cope with evolving grid environments; dyn resources
 - Future SAGA API extensions
 - Portable, syntactically and semantically platform independent; permit latency hiding mechanisms
 - Ease of deployment, configuration, multiple-language support, documentation etc.
 - Provide synchronous, asynchronous & task versions
- Portability, modularity, flexibility, adaptability, extensibility

SAGA API: Towards a Standard

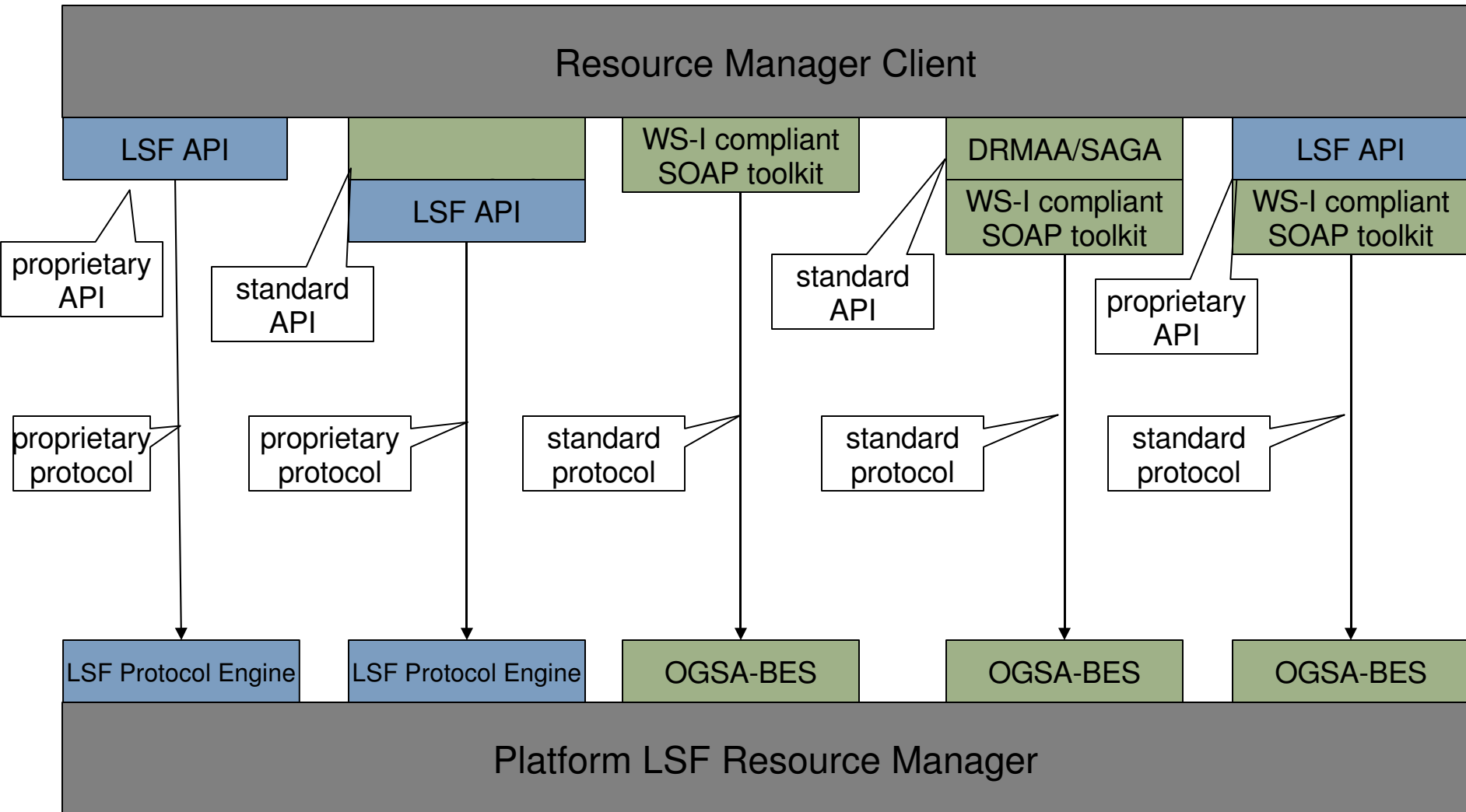


- The need for a standard programming interface
 - “Go it alone” versus “Community” model
 - Reinventing the wheel again, yet again, and again
 - MPI as a useful analogy of community standard
 - OGF the natural choice; establish SAGA-RG
- “Tedium” of the standardisation process?
 - Not all technology needs to be standardised upfront
 - Standardisation not a guarantee to success
 - Requirements Document
 - Design and requirements derived from 23 UC
 - Different projects, applications and functionality

Why Standardize?

- What is the value of implementing standards?
- For vendors
 - meet customer demand for interoperability
- For developers
 - leverage the expertise of other developers
 - offer a choice of tools and platforms in order to speed implementations
- For end-users
 - reduce the costs and risks of adopting grid technology
 - get insight into the best practices of the industry at large

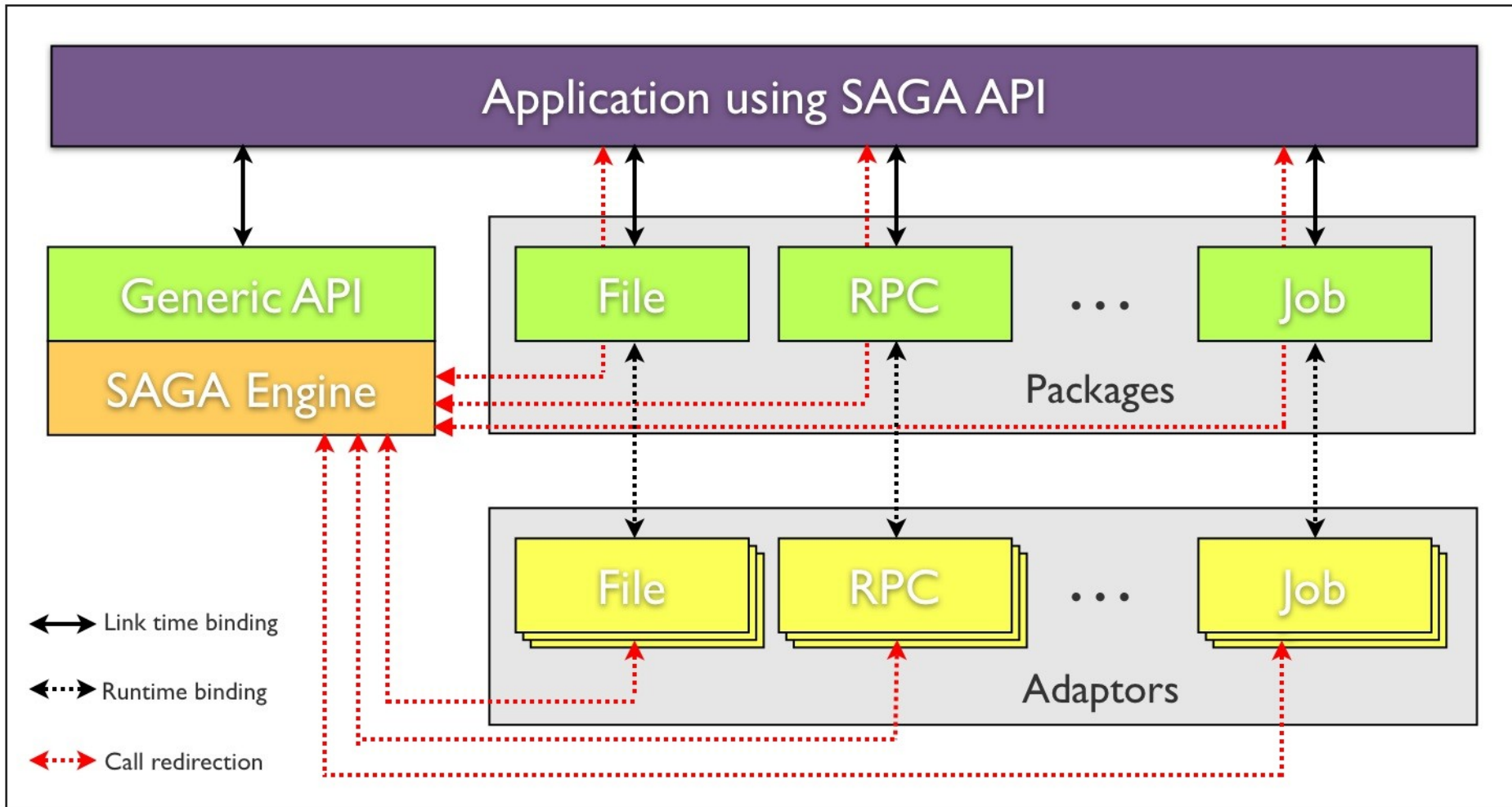
Standard APIs vs Protocols



Implementations

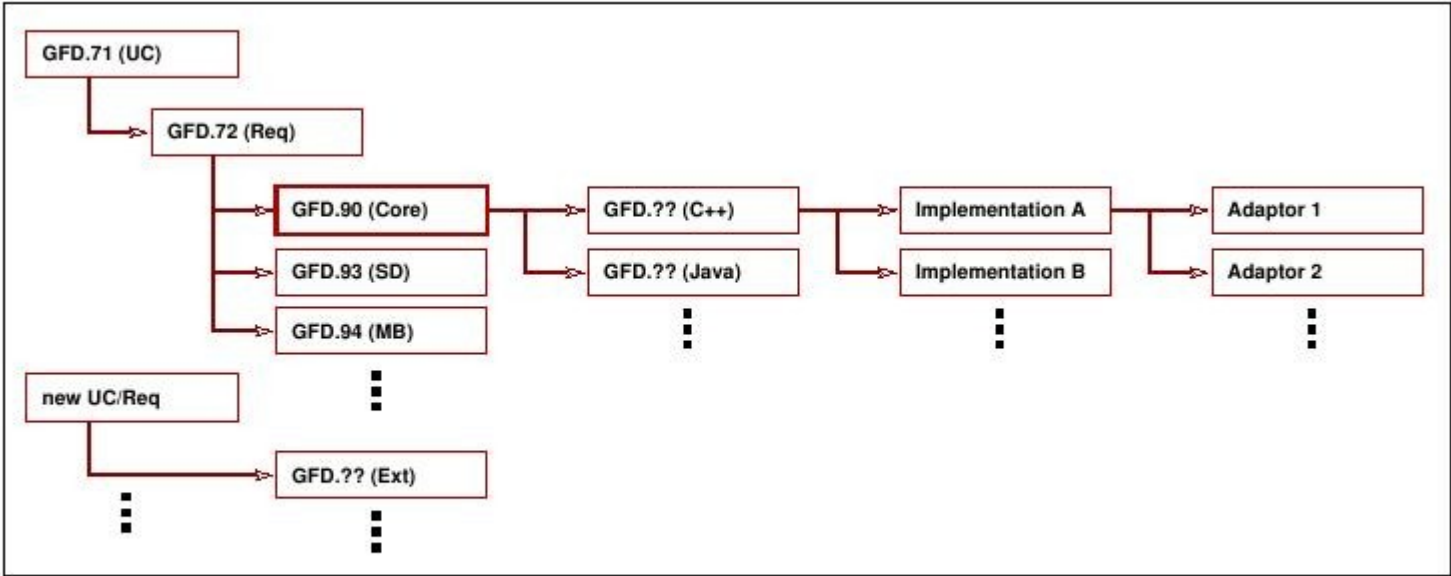
- OGF Standard: Two independent implementations of a specification are required
- LSU: C++
 - STATUS: Usable, mostly done, In Progress
- VU (Amsterdam): Java
 - Part of the OMII-UK Project
 - Builds on EU-GAT project
 - STATUS: In progress

SAGA C++ (LSU)



- Horizontal Extensibility – API Packages
 - Current packages:
 - file management, job management, remote procedure calls, replica management, data streaming
 - Steering, information services, checkpoint in pipeline
- Vertical Extensibility – Middleware Bindings
 - Different adaptors for different middleware
 - Set of ‘local’ adaptors
- Extensibility for Optimization and Features
 - Bulk optimization, modular design

SAGA: Landscape



http://forge.ogf.org/sf/projects/saga-rg



SourceForge : Project Home - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://forge.ogf.org/sf/projects/saga-rg


SAGA :: A Simple API for Grid Appl... SourceForge : Project Home SourceForge : Project Home Distributed Programming Abstractio... (Untitled)

GridForge Home Projects Search

Project Home Tracker Documents Tasks Source Code Discussions File Releases Wiki Project Admin SnipSnap OGFCalendar

Project: SAGA-RG Project Home

Project Home



SAGA-RG

Simple API for Grid Applications RG

Project Created: 05/19/2004

[Project News](#) (0 Items)

Project Members

Total Project Members:22

Project Administrators:

- [Andre Merzky](#)
- [Ole Weidner](#)
- [Pascal Kleijer](#)
- [Shantenu Jha](#)
- [Thilo Kielmann](#)
- [Tom Goodale](#)

Project Info

Simple API For Grid Apps RG (SAGA-RG)

Group Information

Group Type: Research Group

Group Chair(s): [Shantenu Jha](#), [Thilo Kielmann](#)

Group Email: saga-rg@ogf.org ([Subscribe](#)) ([Archive by date](#))

Group Description

The SAGA Research Group provides a forum in GGF to consolidate application driven API specifications.

Group Focus and Scope

The group will build on the results and feedback of the work of the former SAGA RG. As such, it will provide a forum in GGF to consolidate application driven API specifications. The primary goals of the group are:

1. To provide general guidelines for application driven API specifications in GGF (security guidelines, common look & feel, interoperability guidelines, conflict resolution, etc.)
2. The group will form, on an ad hoc basis, design teams which investigate new API areas for inclusion in the SAGA-API. The design teams will
 - solicit related application use cases
 - issue an informal use case document
 - issue an informal requirement document
3. Spawn off focused working groups which will work on API specifications (standard recommendation track) for specific subsystems, e.g. RPC, CPR, application steering etc, after a design team has identified and explored an area as described in (2) above, or identified respective prior work by other GGF groups.
4. Coordinate the activities of the SAGA working groups and make sure that the resulting APIs are in line with the SAGA goals.

The SAGA API does not strive to replace Globus or similar middleware systems, and does not target middleware developers, but solely application developers with no assumed background on Grid Computing who wish to grid enable their applications whilst spending as little time as possible learning new paradigms. Such developers typically wish to devote their time to their own goals and minimise the time spent coding infrastructure functionality. The API will insulate application developers from middleware.

The specification of services, and the protocols to interact with them, is out of the scope of the RG. Rather, the API

http://saga.cct.lsu.edu



SAGA :: A Simple API for Grid Applications - Home - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://saga.cct.lsu.edu/

CCT | Center for Com... google Guardian NYT BBC washingtonpost.com - ... Nature Publishing Grou... Prospect - events globeandmail.com - Ca... W Main Page - Wikipedia,... Camb Gmail - Inbox (2) Linux

SAGA :: A Simple API for Grid A... SourceForge : Project Home SourceForge : Project Home Distributed Programming Abstractio... (Untitled)

0.6 RELEASED

SAGA Framework

A Simple API for Grid Applications

Home About Download Documentation Publications Trac Community Wiki

Download

- Source Releases
- SVN Repository

Documentation

- Tutorials
- Build Instructions

Publications

- Conference Papers
- Presentation Slides

Get the SAGA Spec. v1.0-RC6

Metrics by Ohloh

Latest News

- 06/12/2007 [SAGA C++ Framework Version 0.6 released!](#)
- 04/04/2007 [Code repository migration from CVS to SVN complete](#)
- 03/29/2007 [SAGA C++ Framework Version 0.5 released!](#)
- 03/20/2007 [Slides from the SURA All-hands meeting SAGA presentation online](#)

[Read more...](#)

Welcome to SAGA

The advocates of grid computing promise a world where large, shared, scientific research instruments, experimental data, numerical simulations, analysis tools, research and development platforms, as well as people, are closely coordinated and integrated in 'virtual organizations'.

Still, relatively few grid-enabled applications exist that exploit the full potential of grid environments. This must be largely attributed to the difficulties faced by application developers in trying to master the complex interplay of the various components like resource reservation, security, accounting, and communication. Moreover, grid middleware like Globus, Condor-G, and Unicore are still undergoing many changes, with new software releases appearing frequently.

[Read more...](#)

