

SAGA

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

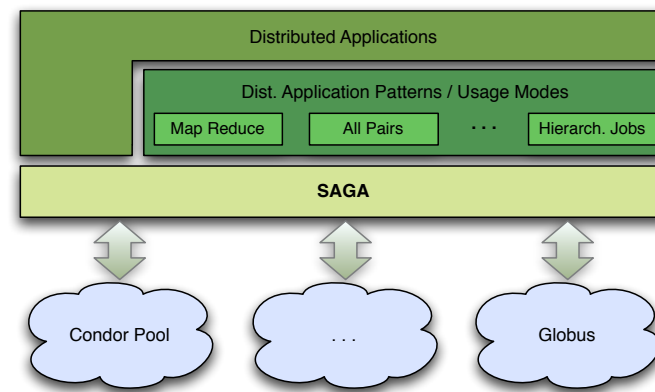
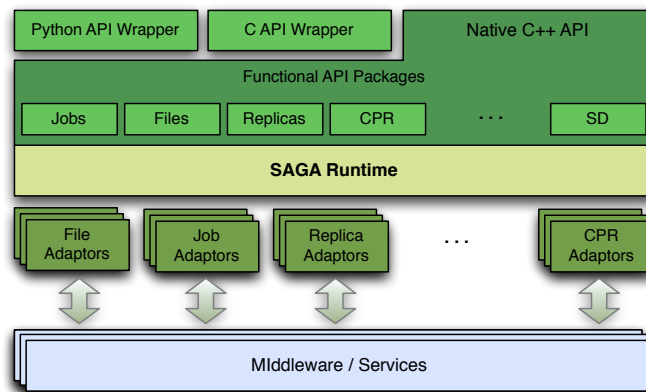
A Simple and Powerful Abstraction Layer

SAGA - the *Simple API for Grid Applications* is a proposed recommendation of the Open Grid Forum (OGF) and defines a high-level programmatic interface for developers of Distributed Applications. The fundamental idea of SAGA is to lower the barrier for applications and application scientists to utilize distributed infrastructure by providing a simple and uniform interface to the most often required functionality in order to construct general purpose, extensible, dynamic and scalable applications.

Implementation

The SAGA team at CCT strives to implement a fully OGF compliant SAGA C++ reference implementation including the runtime system, middleware adaptors and API language bindings for Python and C [1. fig.]. Currently, the implementation comprises the functional packages and middleware adaptors for:

- Job Submission & Management (Globus GRAM, OMII GridSAM, Condor, Fork)
- Filesystem Access (Local Filesystems, Globus GridFTP)
- Replica Management (SQL-based, Globus RLS)
- Advert Services (SQL-based)
- Checkpoint & Recovery Services (Migol)
- Stream Handling & Remote Procedure Calls



Enabling Distributed Application

SAGA has been used to support a wide range of applications types [fig. r.]:

- Applications with novel and agile distributed execution modes. For example, we have re-architected a Kalman-Filter based application to enable it to determine at run time, the optimal resource to launch sub-jobs.
- SAGA has also been used to implement an adaptive distributed replicaexchange application which can opportunistically exploit multiple resources concurrently without prior arrangement and dynamically reallocate resources.
- SAGA is also being used to develop novel applications written for explicit deployment on distributed infrastructure, i.e. first-principles SAGA can support a wide range of applications types. Besides developing SAGA itself, CCT is involved in several projects that use SAGA's powerful abstractions. d