CENTER FOR COMPUTATION & TECHNOLOGY

Interoperabilty: The SAGA Approach and Experience

Shantenu Jha, Andre Merzky, Ole Weidner & * Collaborators

http://saga.cct.lsu.edu

omii-uk www.omii.ac.uk

CENTER FOR COMPUTATION & TECHNOLOGY

Outline

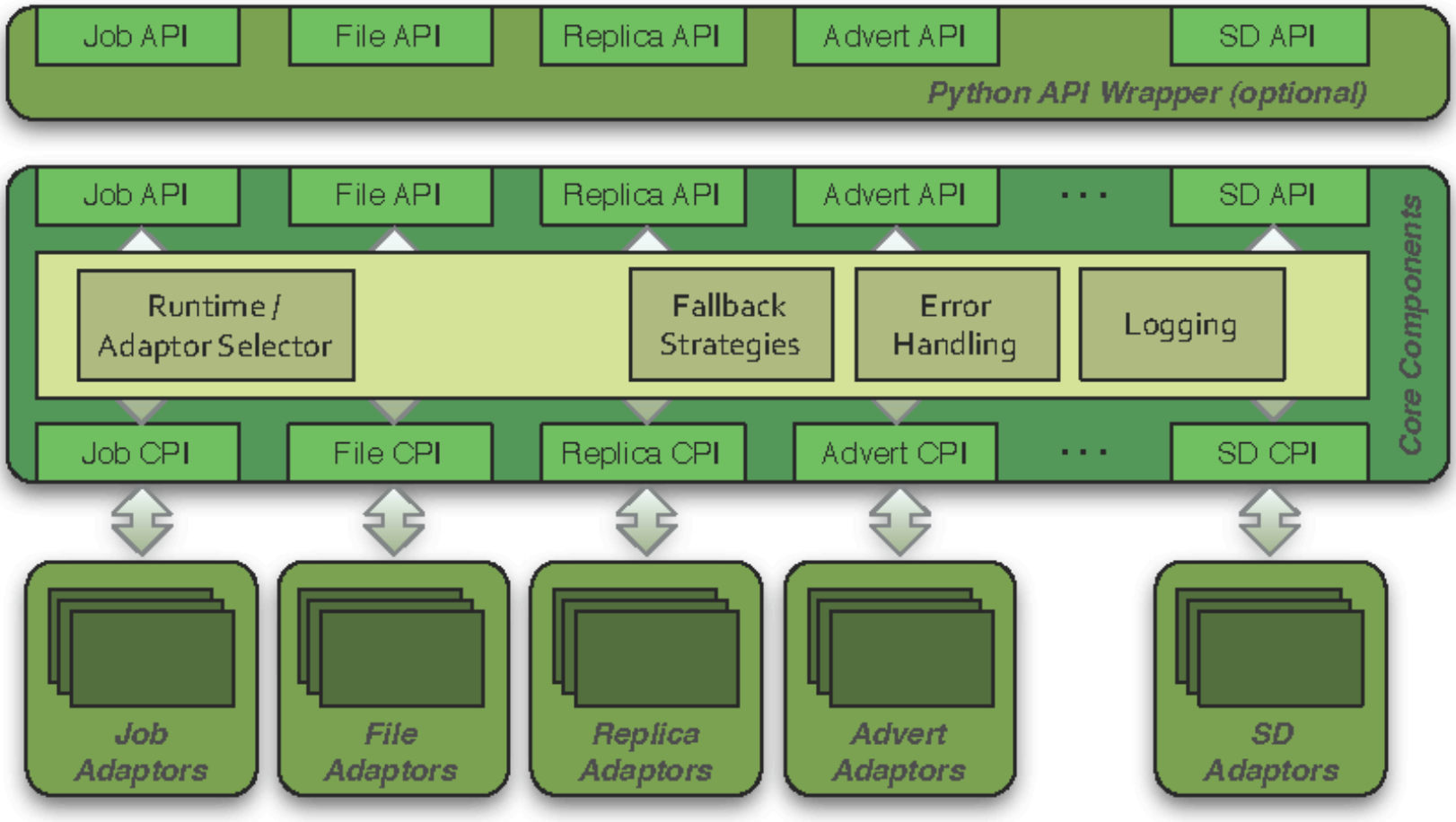
- Introduction to SAGA:
- Why SAGA for Interoperability?
  - Use of a standards-based approach for interoperability
- Four Interoperability Projects – access layers and tools
  - HPC-HTC 1: EGEE-TG[-NAREGI]
  - HPC-HTC 2: KEK/NAREGI-TG
  - HPC-HTC 3: ExTENCI [TG-OSG]
  - HPC-HPC 1: TG-DEISA
- Some thoughts on PGI Interoperability

SAGA: In a nutshell

- There exists a lack of programmatic approaches that:
  - Provide general-purpose, basic &common grid functionality for applications and thus hide underlying complexity, varying semantics..
  - The building blocks upon which to construct "consistent" higher-levels of functionality and abstractions
  - Meets the need for a Broad Spectrum of Application:
    - Simple scripts, Gateways, Smart Applications and Production Grade Tooling, Workflow…
- Simple, integrated, stable, uniform and high-level interface
  - Simple and Stable: 80:20 restricted scope and **Standard**
  - Integrated: Similar semantics & style across
  - Uniform: Same interface for different distributed systems

CENTER FOR COMPUTATION & TECHNOLOGY

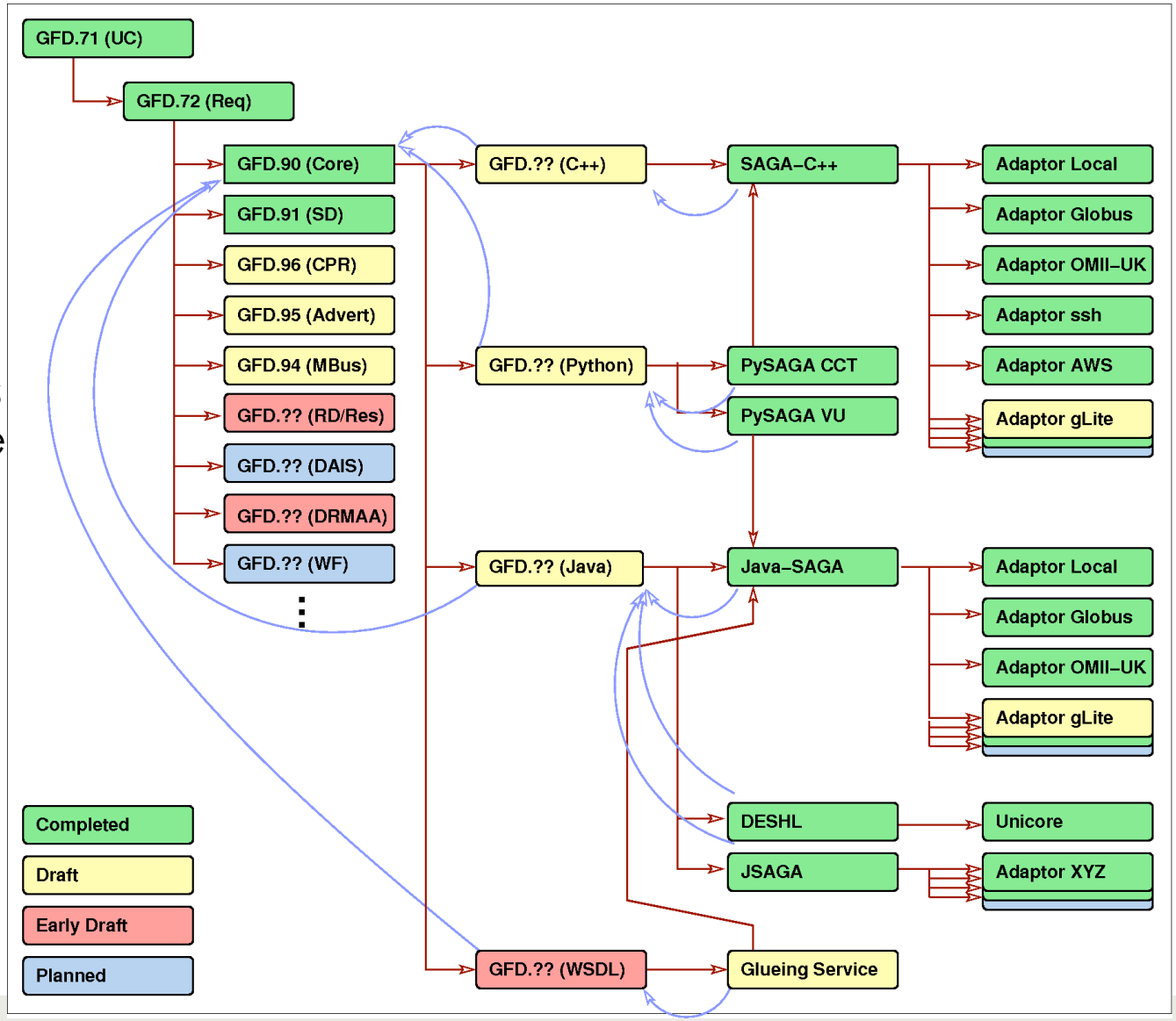
SAGA: Architecture



CENTER FOR COMPUTATION & TECHNOLOGY

SAGA: Specification Landscape

Blue lines show which packages have input in the Experience document



CENTER FOR COMPUTATION & TECHNOLOGY

SAGA/CREAM C++ Example

```
// Submitting a simple job to a CREAM CE
// with SAGA (C++ Example)

#include <saga/saga.hpp>

int main(int argc, char **argv)
{
  try {
    saga::job::description jd;

    jd.set_attribute (saga::job::attributes::description_executable, "/bin/date/");

    saga::job::service js("cream://cream-09.pd.infn.it:8443/cream-pbs-cream_A");
    saga::job::job cream_job = js.create_job(jd);

    cream_job.run();

    std::cout << "\nJob ID    : " << cream_job.get_job_id() << std::endl;
    std::cout << "Job State : "   << cream_job.get_state() << std::endl;

    cream_job.wait(-1.0);          // waits for state change

    std::cout << "Job State : "   << cream_job.get_state() << std::endl;
  }
  catch(saga::exception const & e) {
    std::cerr << "OOPS: " << e.what() << std::endl;
  }
}
```

CENTER FOR COMPUTAT
& TECHNOLOGY

SAGA API: Standards promote Interoperability

- The need for standard programming interface
  - Trade-off "Go it alone" versus "Community" model
  - Reinventing the wheel again, yet again, & then again
  - MPI a useful analogy of community standard
    - Vendors (Resource Provider), Software developers, users..
    - social/historic parallels also important
      - Time to adoption, after specification ....
- OGF the natural choice (SAGA-RG, SAGA-WG)
    - Spin-off of the Applications Research Group
    - Driven by UK, EU (German/Dutch), US
    - Design derived from 23 Use Cases
      - different projects, applications and functionality
      - biological, coastal modelling, visualization
- Will discuss the advantage of SAGA as a standard specification

SAGA-based Tools and Projects
Advantage of Standards

- JSAGA from IN2P3 (Lyon)
  - http://grid.in2p3.fr/jsaga/index.html
  - gLite adaptors exist
- JAVASAGA (Amsterdam)
  - Has a wide range of adaptors
  - JAVASAGA gets released by gLite (next few weeks)
- NAREGI/KEK (Active)
  - http://www.ogf.org/OGF27/materials/1767/OGF27_SAGA_KEK.pdf
- DEISA/DESHL
  - http://www.fz-juelich.de/nic-series/volume38/pringle.pdf)
  - http://deisa-jra7.forge.nesc.ac.uk/ and http://www.ogf.org/OGF19/materials/501/SAGA-DEISA.ppt
- XtreemOS
  - http://saga.cct.lsu.edu/index.php?option=com_content&task=view&id=95&Itemid=174

SAGA Implementation: Extensibility

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

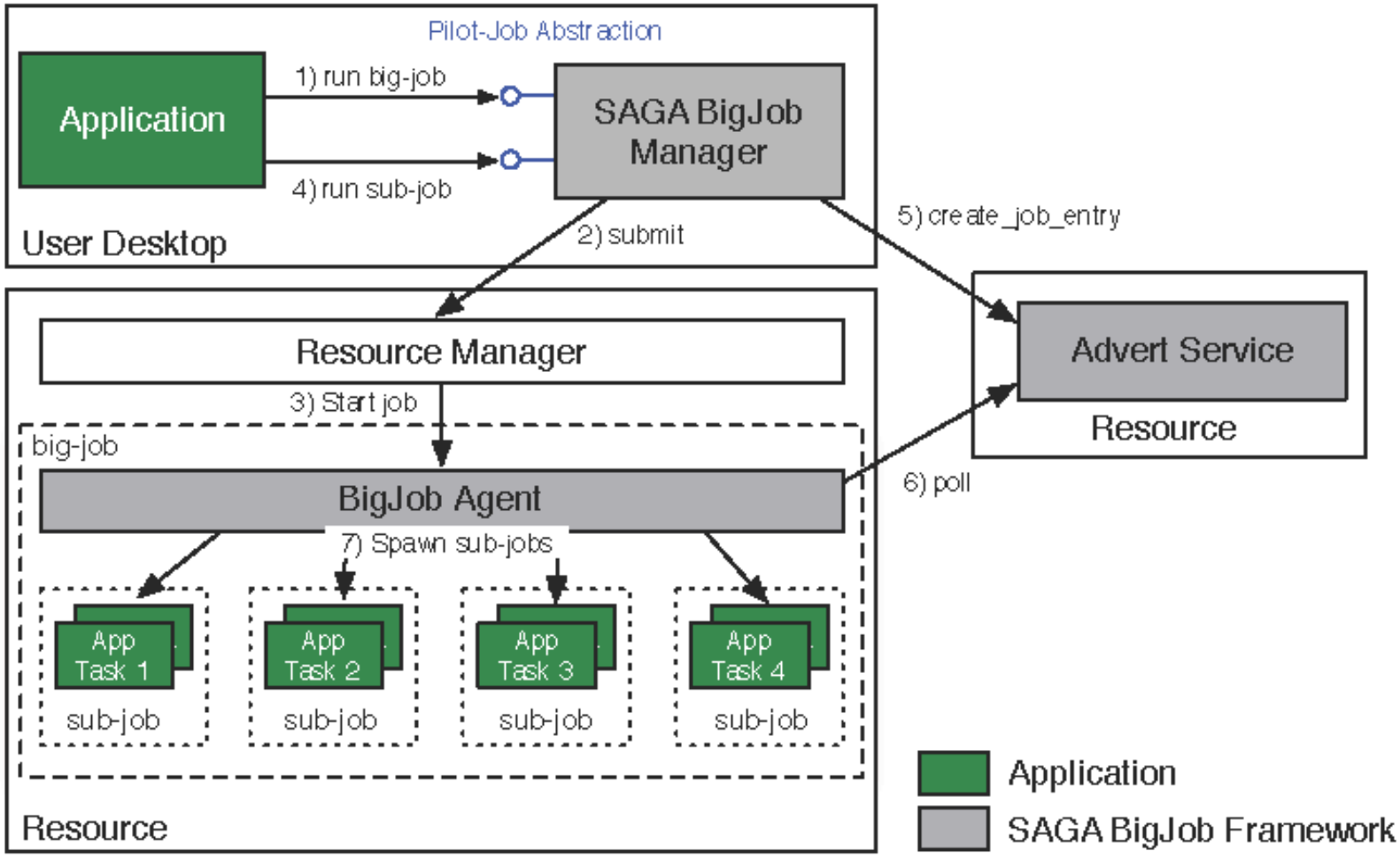
CENTER FOR COMPUTATION & TECHNOLOGY

SAGA: Access Layers
Challenge of many Adaptors

- Job Adaptors
  - **BES, UNICORE, Globus GRAM2, gLite**
  - Fork (localhost), SSH, Condor, 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
- Other Adaptors
  - Default RPC / Stream / SD

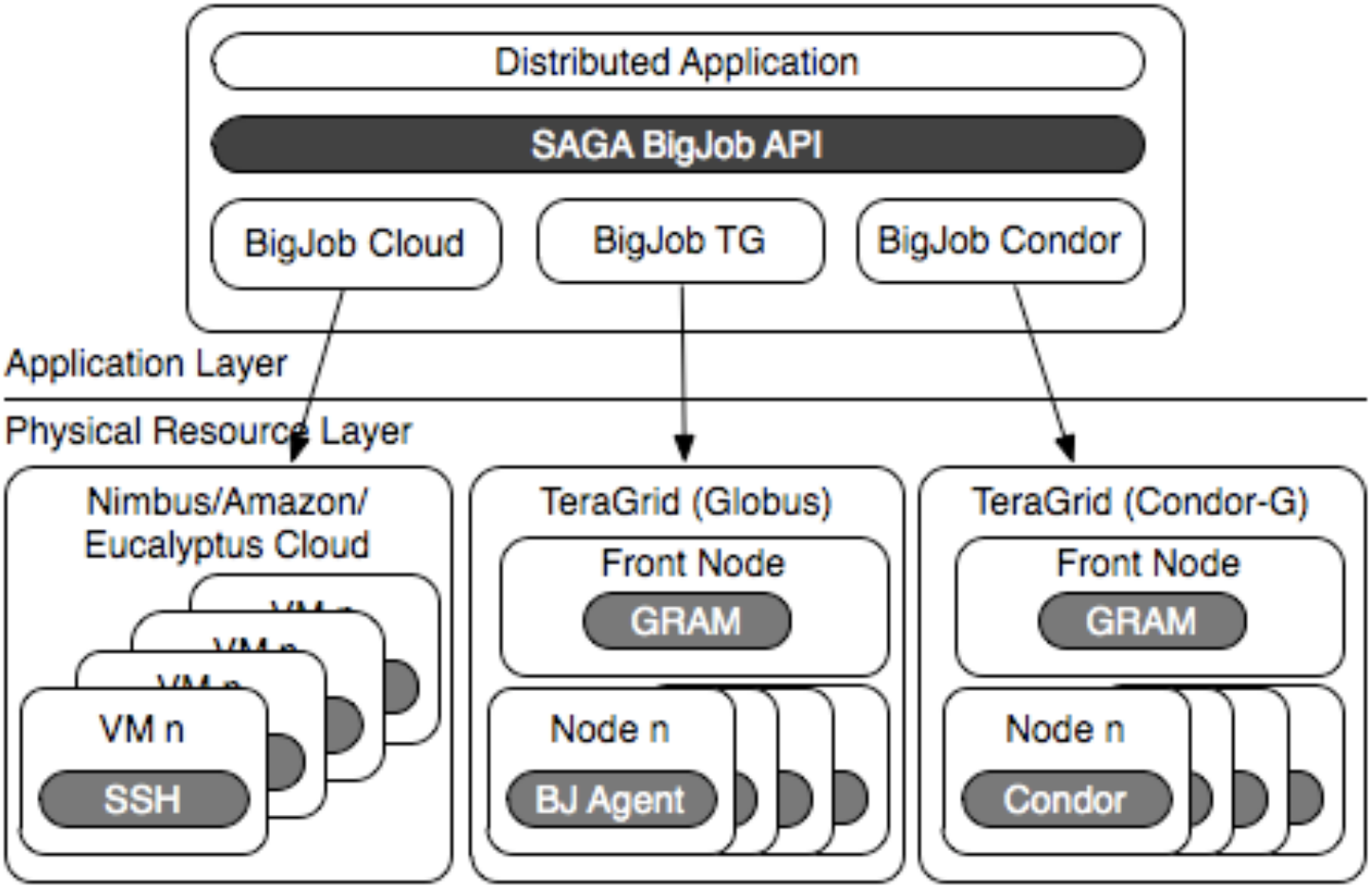
CENTER FOR COMPUTATION & TECHNOLOGY

Abstractions for Dynamic Execution SAGA Pilot-Job (BigJob)



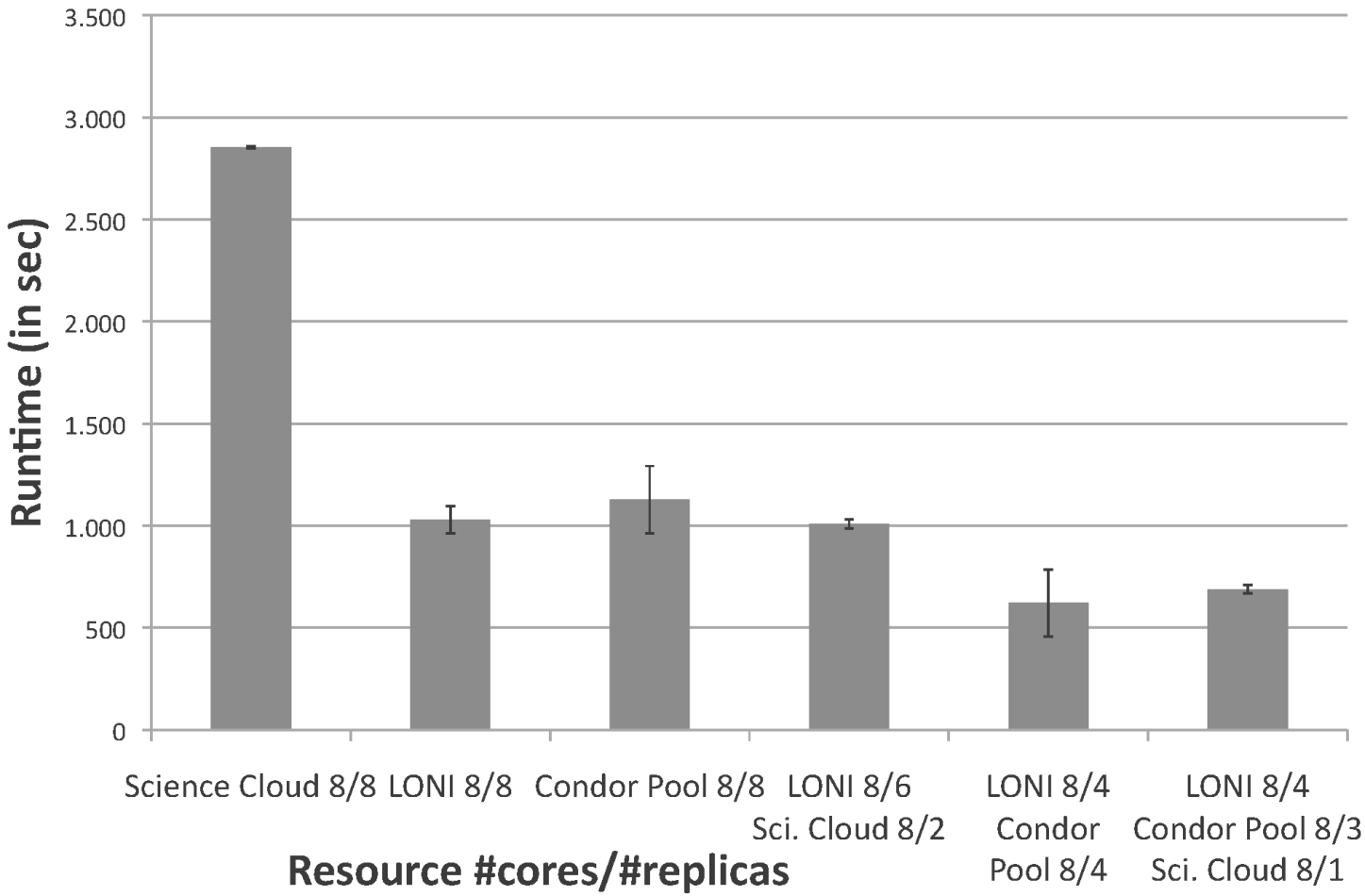
CENTER FOR COMPUTATION & TECHNOLOGY

BigJob: Infrastructure Independent Pilot-Job



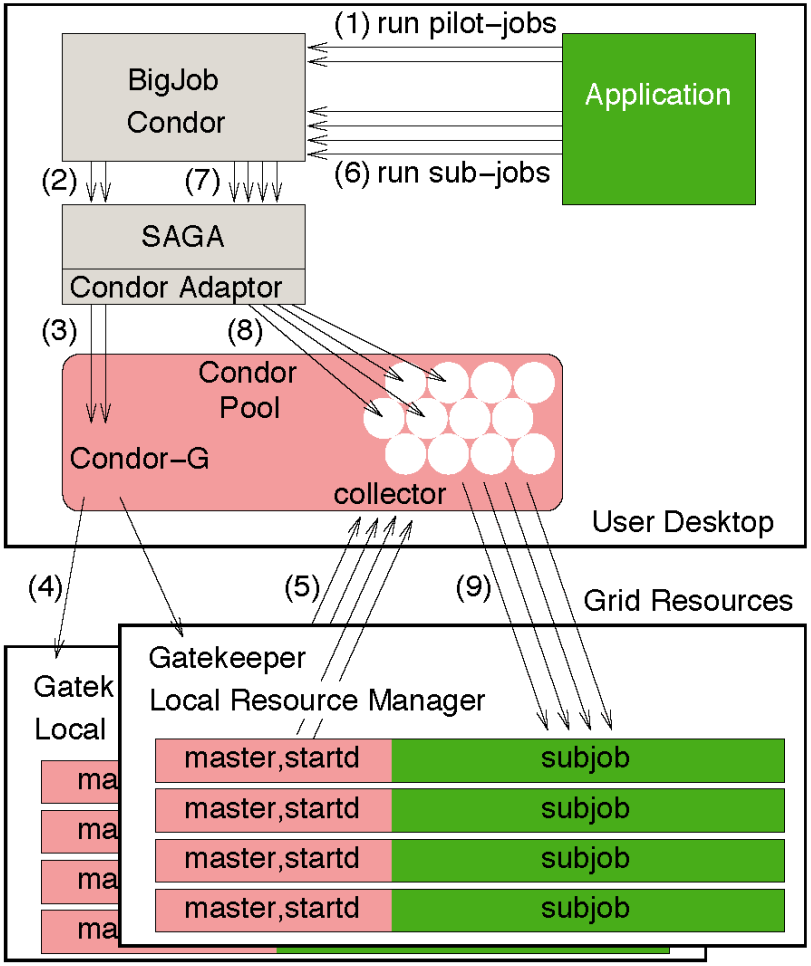
CENTER FOR COMPUTATION & TECHNOLOGY

BigJob: Infrastructure Independent Pilot-Job (Each sub-job is a MPI-based MD)



CENTER FOR COMPUTATION & TECHNOLOGY

BigJob: Preserving Glide-in Semantics and Interface



CENTER FOR COMPUTATION & TECHNOLOGY

SAGA Pilot-Jobs: What is different?

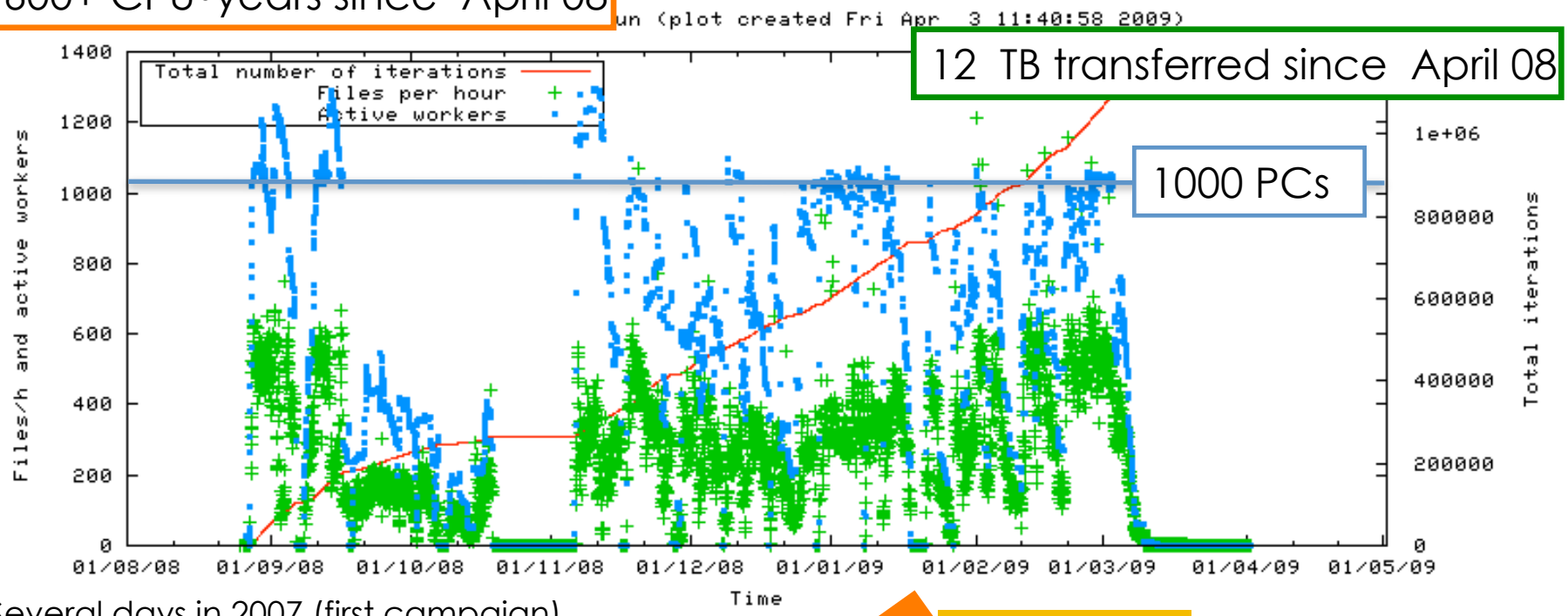
- Pilot-Jobs: Decouple Resource Allocation from Resource-Workload binding
- Pilot-Jobs are/have been typically used for:
  - Enhancing resource utilisation
  - Lowering wait time for multiple jobs (better predictibility)
  - Facilitate high-throughput simulations
  - Basis for Application-level Scheduling Resource binding
- Two unique aspects about the SAGA-based Pilot-Job:
  - Pilot-Jobs have not been used for Science Driven Objectives:
    - First demonstration of supporting multi-physics simulations
  - Infrastructure Independent
    - Falkon, Condor Glide-in, Ganga-Diane (EGEE/EGI), DIRAC/WMS, PANDA
      - Frameworks based upon PJs (pull model) for specific PGI/back-end
      - Do not support MPI
- SAGA-based Pilot-Job form the basis:
  - For autonomic scheduling and resource selection decisions
  - Advanced run-time frameworks for load-balancing and fault-tolerance

CENTER FOR COMPUTATION & TECHNOLOGY

Lattice QCD on the Grid

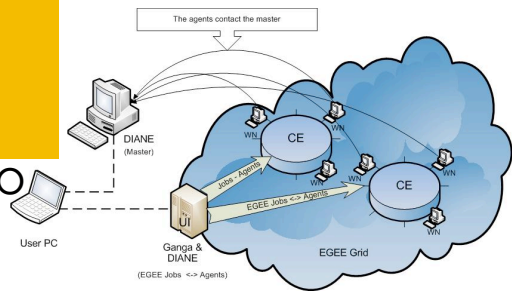
600+ CPU•years since April 08

12 TB transferred since April 08



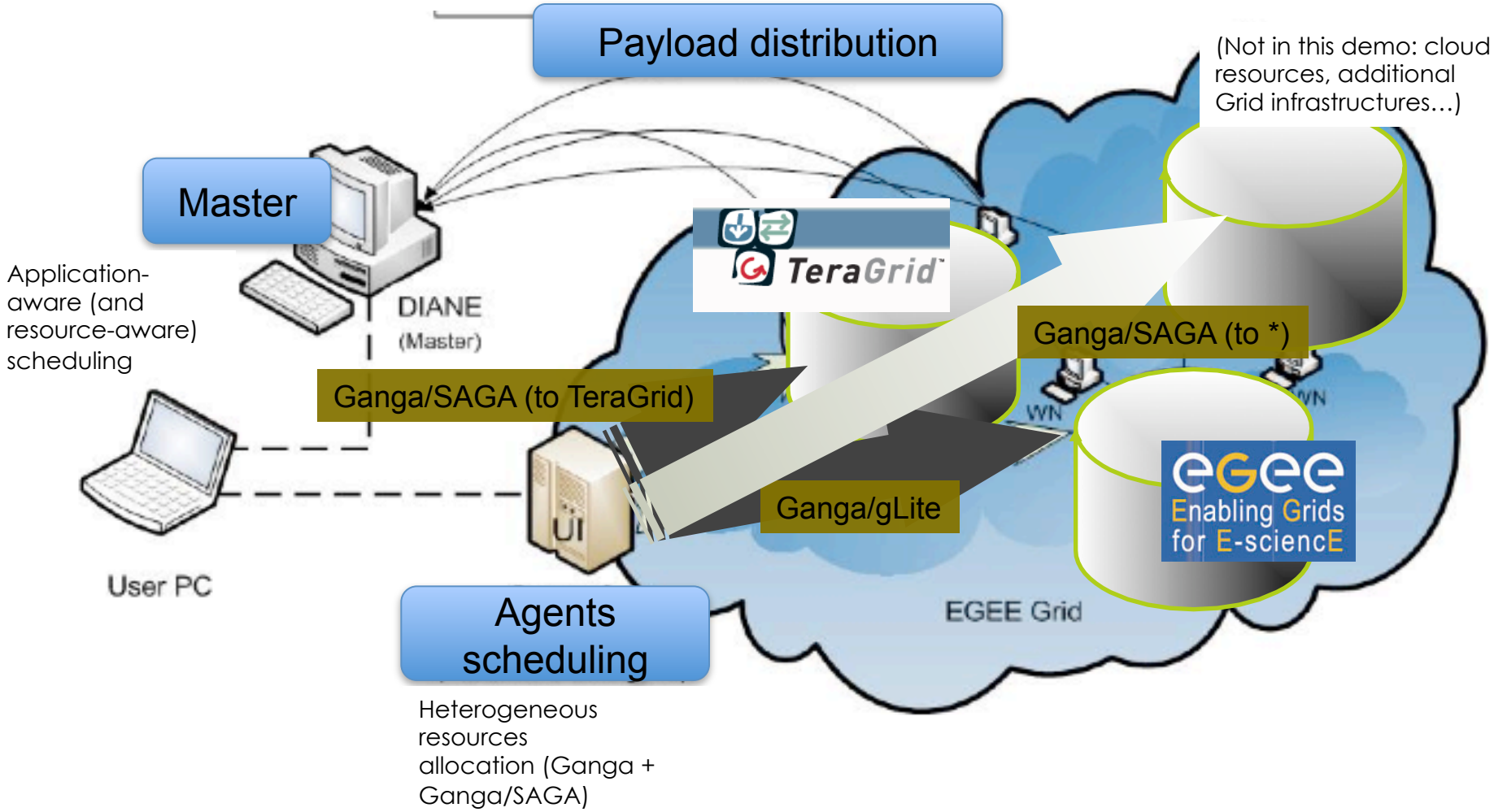
- Several days in 2007 (first campaign)
  - Enough for getting interesting results
- 12 months of running in 2008/9 (second campaign)
  - Long period needed (with many more CPUs), graph Sep08-Mar0
- **Now, not *simply* more CPUs but different resources (MPI jobs)**
  - **Tighter integration of the Grid and the supercomputer worlds**

"Natural" evolution of a scientific applicatio n!



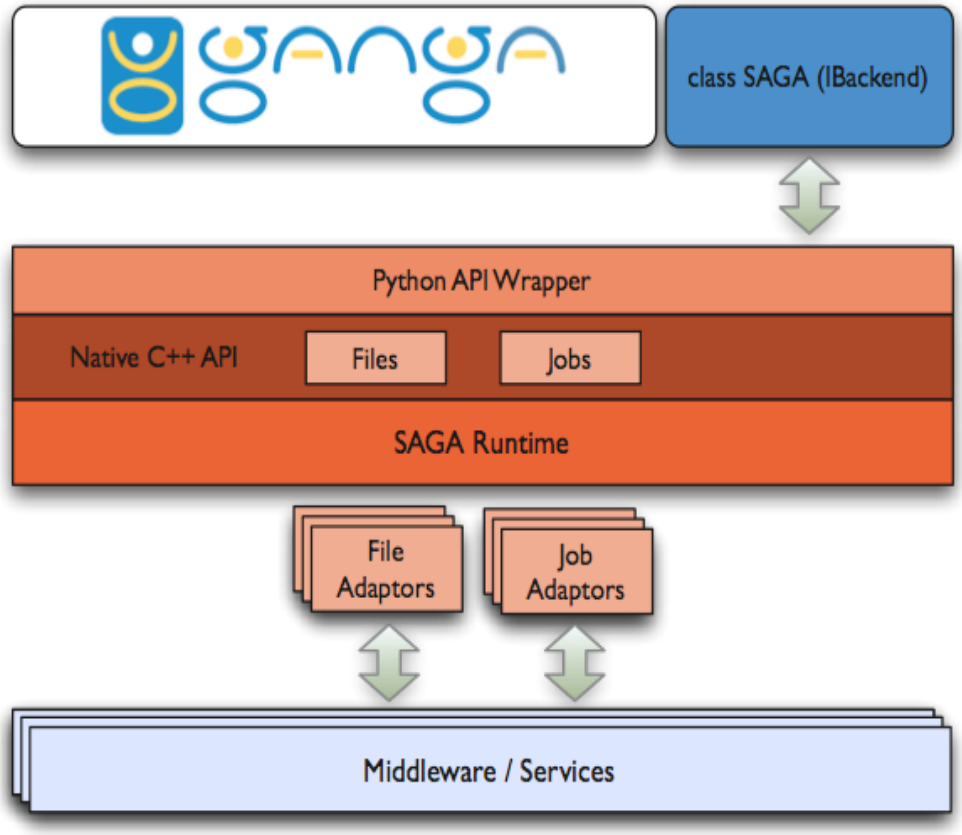
Lattice-QCD Applications on heterogeneous resources

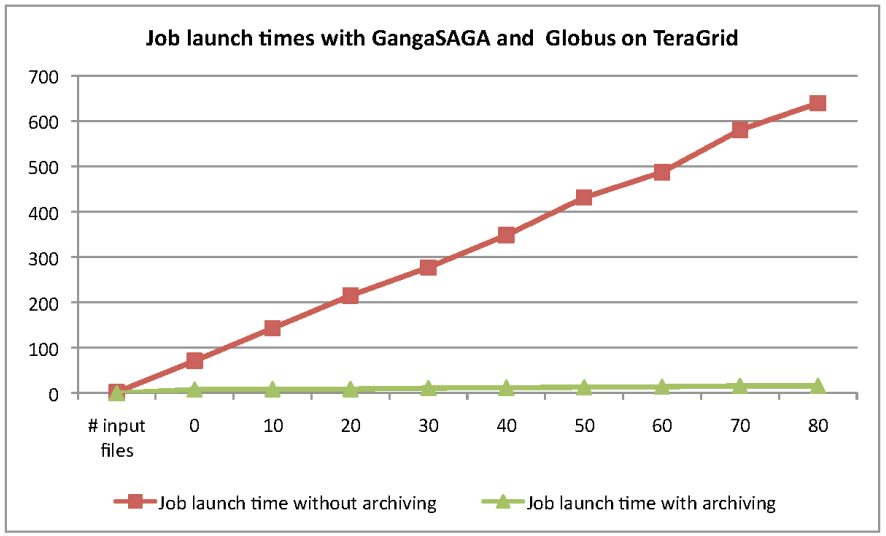
Federating resources! EGEE Conference (Apr'10)



CENTER FOR COMPUTATION & TECHNOLOGY

SAGA-GANGA Integration

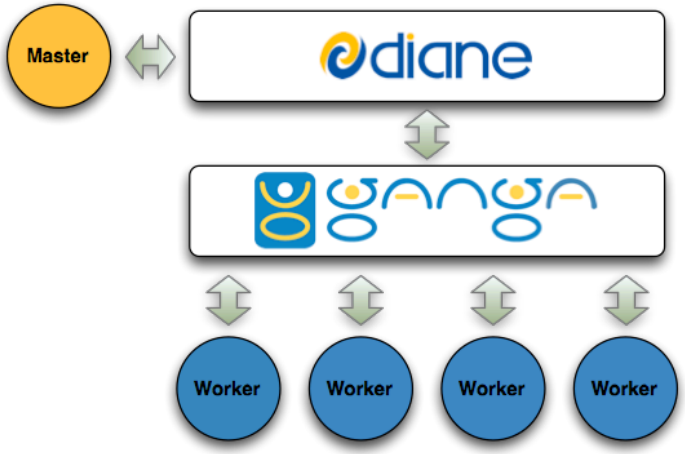




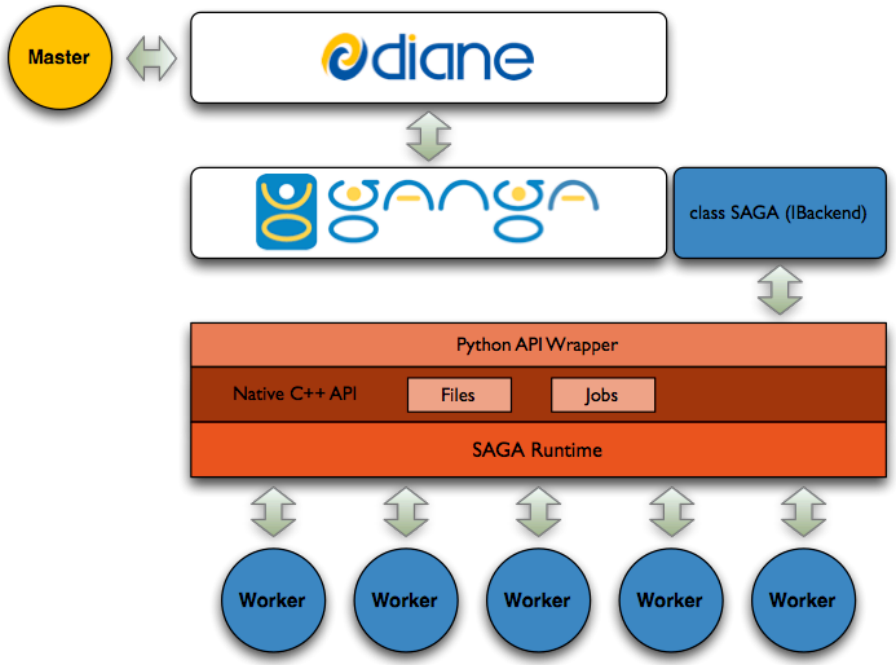
CENTER FOR COMPUTATION & TECHNOLOGY

DIANE INTEGRATION

Diane without SAGA



Diane with SAGA

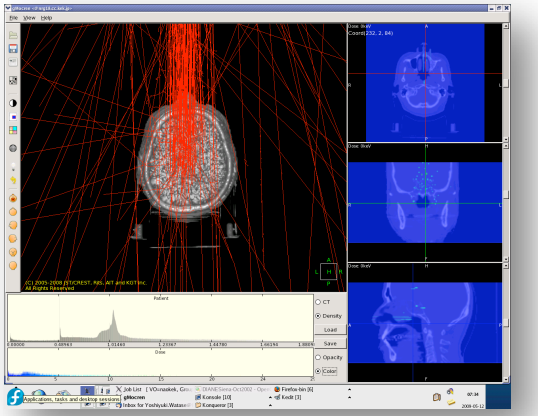


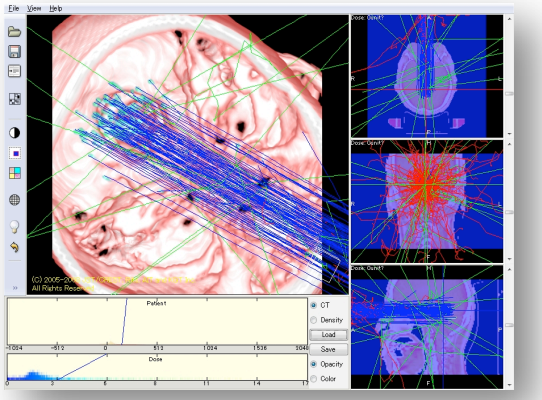
DIANE is an execution manager with support for pilot-jobs + worker agents (ID**E**AS Redux)

CENTER FOR COMPUTATION & TECHNOLOGY

NAREGI-TG: Practical Examples

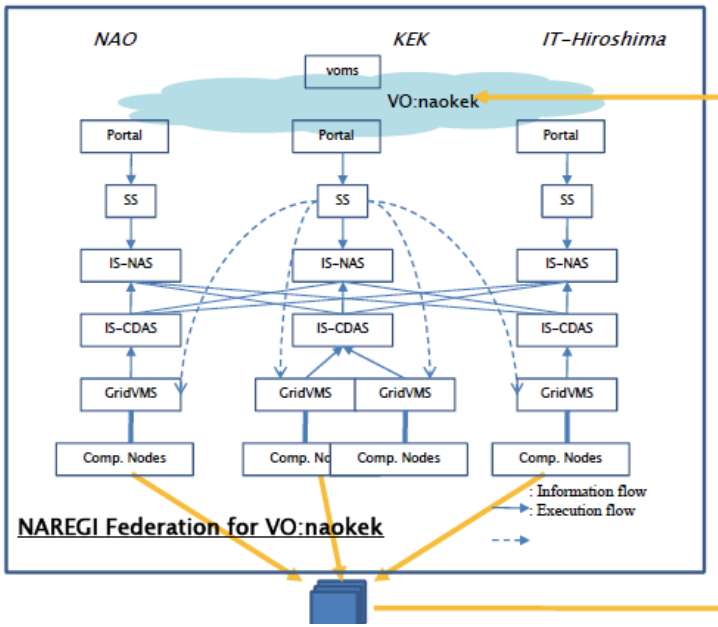
- Grid environment
  - MW: NAREGI v1.1 released in
  - VO scale: KEK, NAO, HIT, and NII
- SAGA adaptors:
  - NAREGI adaptor for job completed
  - Torque adaptor completed
- Demonstration in testbed
  - Particle therapy simulation based on Geant4 as the 1st practical example
  - Resource scale
    - 3 sites: KEK, NAO, HIT
    - CPU: 10 cores
    - OS: CentOS 5.2 x86_64
    - Memory: 2 GB each





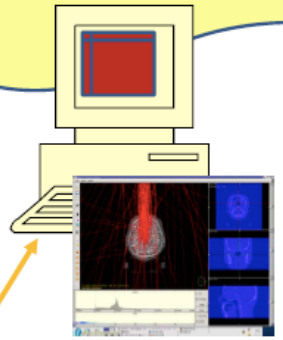
More application-wise development in 2010

Particle Therapy Simulation w/ SNA



Python Script on SAGA

- -Set # of parallel jobs
- -Build JSDLs
- -Submit job to GridVMS's
- -wait job finish
- -start display date



Client Host

- -SAGA Engine
- -NAREGI Adaptor
- -NAREGI CLI

User

Login

sg01.cc.kek.jp
SAGA Adaptor Host

SNA

STA

NAREGI Job

Torque Job

nrg04.cc.kek.jp
NAREGI/GVMS

dg02.cc.kek.jp
Torque

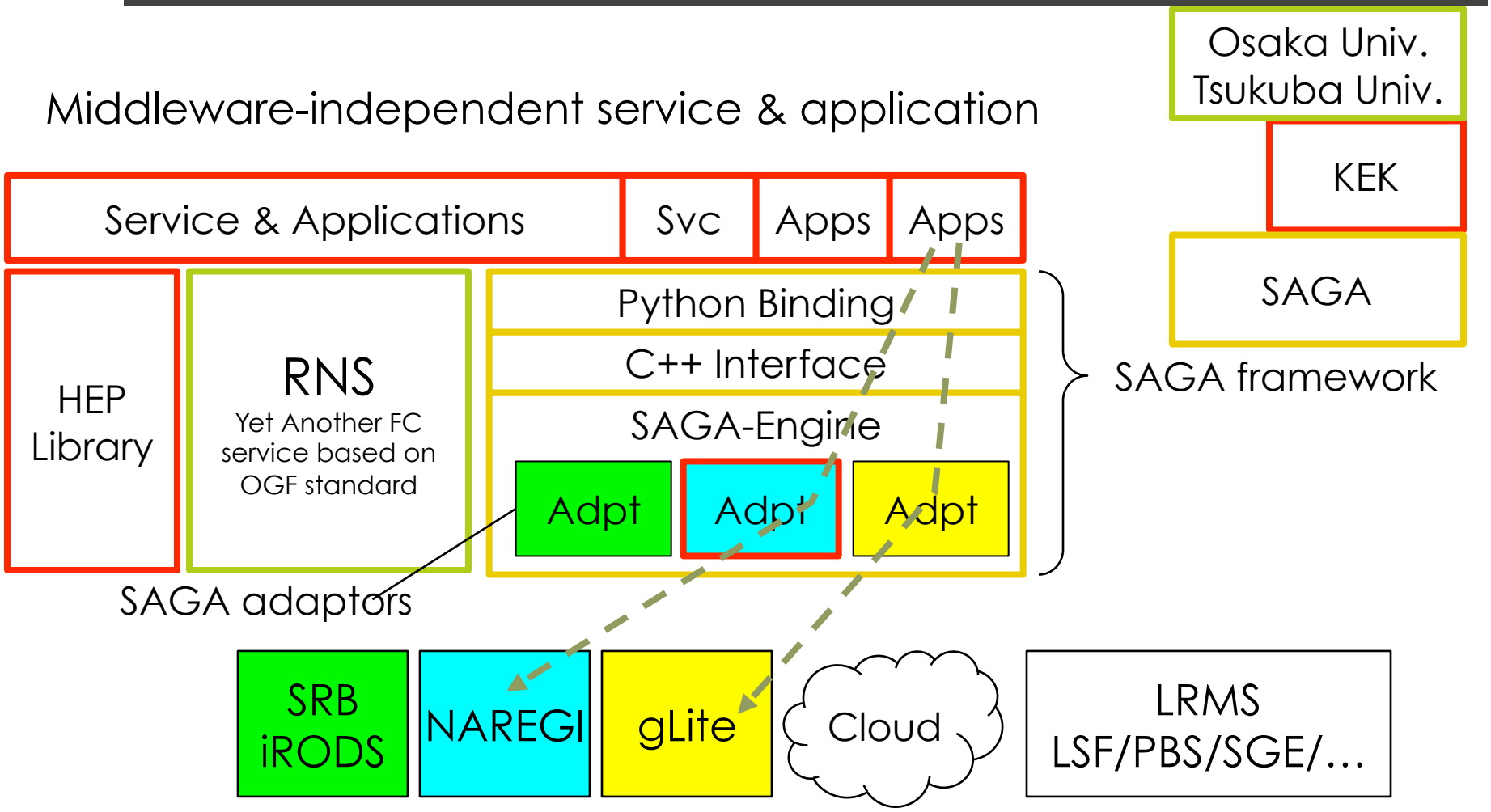
wn107 wn108 wn109

wn101 wn102 wn103

CENTER FOR COMPUTATION & TECHNOLOGY

RENKEI Project Aims

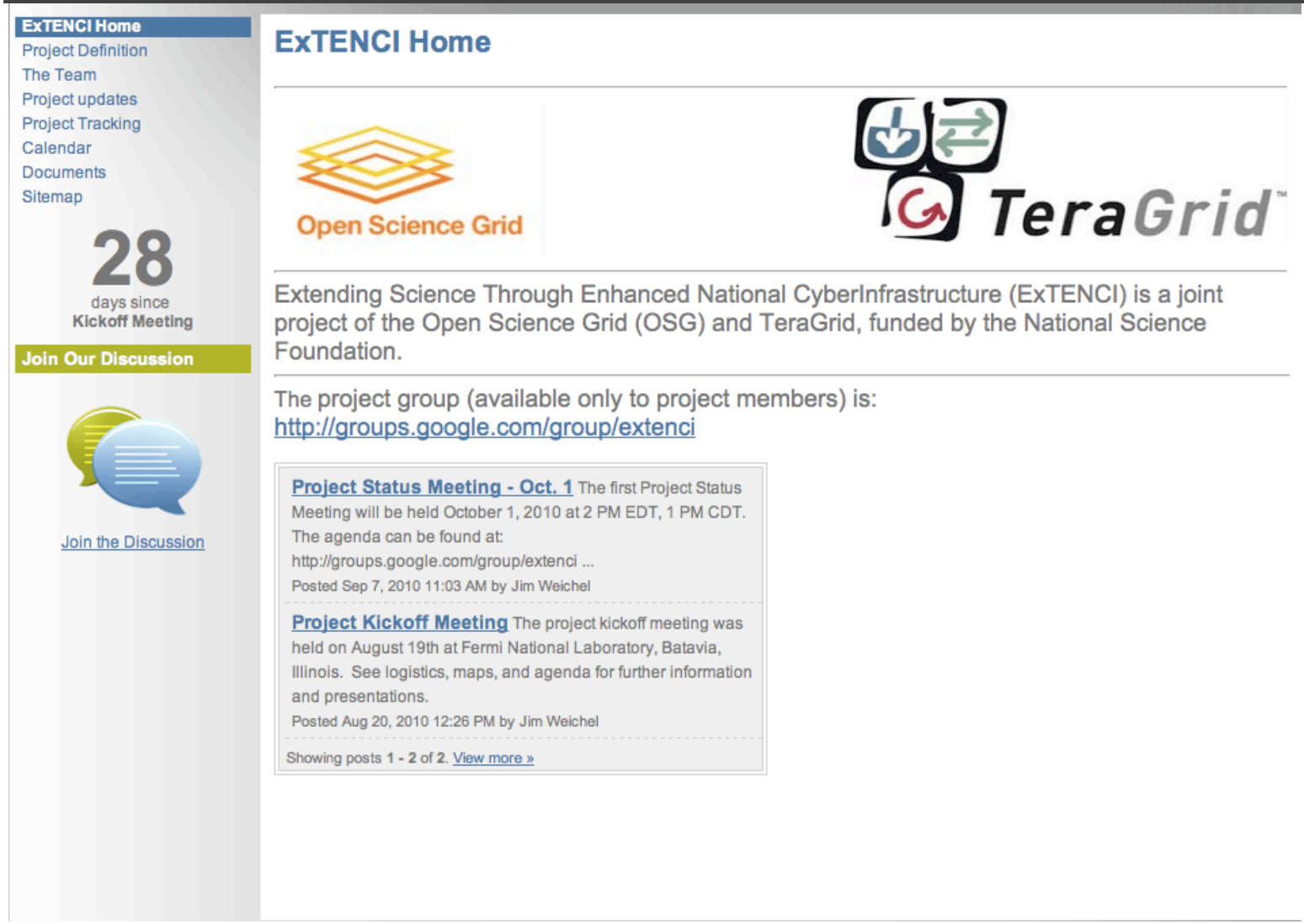
Middleware-independent service & application



This activity is funded by MEXT as a part of RENKEI project which develops seamless linkage of resources in the Grids and the local one for e-Science.

CENTER FOR COMPUTATION & TECHNOLOGY

ExTENCI – NSF funded TG-OSG



CENTER FOR COMPUTATION & TECHNOLOGY

ExTENCI: TeraGrid-OSG [2010-12] Cactus Application Scenarios

- Problem size varies – determinant of Infrastructure used
  - TG, OSG or either..
- MPI-based applications have a very complex SW environment that they need to worry about
- Application Scenarios/Usage Modes
  - 1. Ensemble of Cactus Simulations
    - NumRel, EnKF (Petroleum Eng)
  - 2. Multiphysics Code
    - GR-MHD, CFD-MD
  - 3. Spawning Simulations
    - Realtime 'outsourcing' from BlueWaters/Ranger to specialised architectures or less powerful resources

on a national basis, with no reference to EU funded initiatives.
The DEISA VPH Virtual Community is being administered by VPH NoE WP3 staff. The Application Hosting Environment, (see 'Simplifying grid computing for research and medical purposes' article), is a key component of the upcoming VPH NoE Toolkit release, and is the recommended way for VPH projects to access DEISA resources.
VPH NoE is continuing to work with DEISA to provide support for emergency medical computing requirements, by providing the ability to reserve in advance time on computational resources, so that it can be scheduled in to clinical workflows, as well as the ability to submit urgent ("emergency") jobs that preempt the current workload of the machine. VPH scenarios are also key to an NSF-funded project to enhance interoperability between DEISA and the US TeraGrid infrastructure (see box),
The Partnership for Advanced Computing in Europe (PRACE) is laying the groundwork for the creation of a persistent pan-European HPC service, which we expect will provide VPH researchers with access to capability computers that will form the top level of the European HPC ecosystem. ■

Peter Coveney was an invited speaker at this year's DEISA-PRACE Annual Symposium in Amsterdam 11-13 May, where he spoke under the heading "DEISA, PRACE & the Virtual Physiological Human". See link for further information:
http://www.deisa.eu/news_press/symposium/Amsterdam2009/deisa-symposium-amsterdam-may-11-13-2009
VPH-I projects wishing to make use of the allocation should contact our dedicated email allocations vph-allocations@cercim.org

LONI-TeraGrid-DEISA Interoperability Project

A year long Science-Driven Project Using Advanced Cyber Infrastructure funded by NSF via a HPCOPS award to LONI (one of the TeraGrid Resource Providers), aims to establish TeraGrid-DEISA Interoperabilty on a firm but extensible footing and began on 1 June 2009.
The high-level aim of this project is to enable scientific applications to utilise the federated capabilities of the TeraGrid, DEISA and LONI systems, to enhance the understanding of HIV-1 enzymes and epidermal growth factor receptors (EGFR) implicated in lung cancer. Specifically, the aim of this project is to use several Replica-based and Replica-Exchange simulations for HIV-1 & EGFR research on multiple TeraGrid, LONI and DEISA resources. The project will also work closely with researchers from the VPH-I project ContraCancrum. In addition to scientific advances, this project will provide working implementations and tools that can be utilised by a broad range of applications to utilize resources and effectively scale-out on the TeraGrid, DEISA and LONI.
This project is being co-led by Dr Shantenu Jha (Louisiana State University) and Prof. Peter Coveney (UCL). ■

Contact and further details can be found at:
http://www.teragridforum.org/mediawiki/index.php?title=LONI-TeraGrid-DEISA_Interoperabilty_Project#Kickoff_Meeting
Further details on the project can be found at:
http://www.teragridforum.org/mediawiki/index.php?title=LONI-TeraGrid-DEISA_Interoperabilty_Project


NGS
National Grid Service
Distributed
European
Infrastructure for
Supercomputing
Applications
TeraGrid™
EMPOWERING DISCOVERY
Grid Applications
Grid Middleware
Networking
NAREGI
eGee
Enabling Grids
for E-sciencE

CENTER FOR COMPUTATION & TECHNOLOGY

Some thoughts on PGI

- Interoperation is needed. Now! [And forever..!]
- The community has voted for Interoperation with their feet:
  - Application Scientists + Developers
  - Tool Developers
  - PGI - Resource Providers
- The question is not **whether to**, but **how to** provide interoperation?
  - Ideal world: Infrastructure would be interoperable "out-of-the-box"
  - Ditch SAGA: "Price of success should be irrelevance" ☺
  - Application level? versus Infrastructure level?
    - ALI: Simple, limited [User Access-layer]
    - RLI: Complex, complete [System Access Layer]
    - **SAGA CAN BE USED FOR BOTH !**
  - ALI vs RLI: Is there a difference in the time-scale of capability?
    - User Access-layer via SAGA Vs System Access-Layer