CENTER FOR COMPUTATION
& TECHNOLOGY

Bringing the Cactus Community to ExTENCI

Shantenu Jha, Ole Weidner, Hartmut Kaiser
Gabrielle Allen, Steve Brandt

http://saga.cct.lsu.edu

omii-uk
www.omii.ac.uk

- Intro to SAGA & Cactus
- Three other Interoperability Projects
  - EGEE-TG[-NAREGI]
    - "ONE PERSONS' TOOL IS ANOTHER PERSONS' APPLICATION"
  - NAREGI-TG
  - TG-DEISA
- Three Cactus Application Scenarios
  - Ensembles of Cactus Simulations
  - Real-time Spawning
  - Multi-Physics Simulations
- Plan for ExTENCI:
  - A Cactus GateWay is under development.
  - SAGA BigJob will be made to work with TG + OSG
  - Integrate SAGA BigJob Interface into GW

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

LSU
CENTER FOR COMPUTATION
& TECHNOLOGY

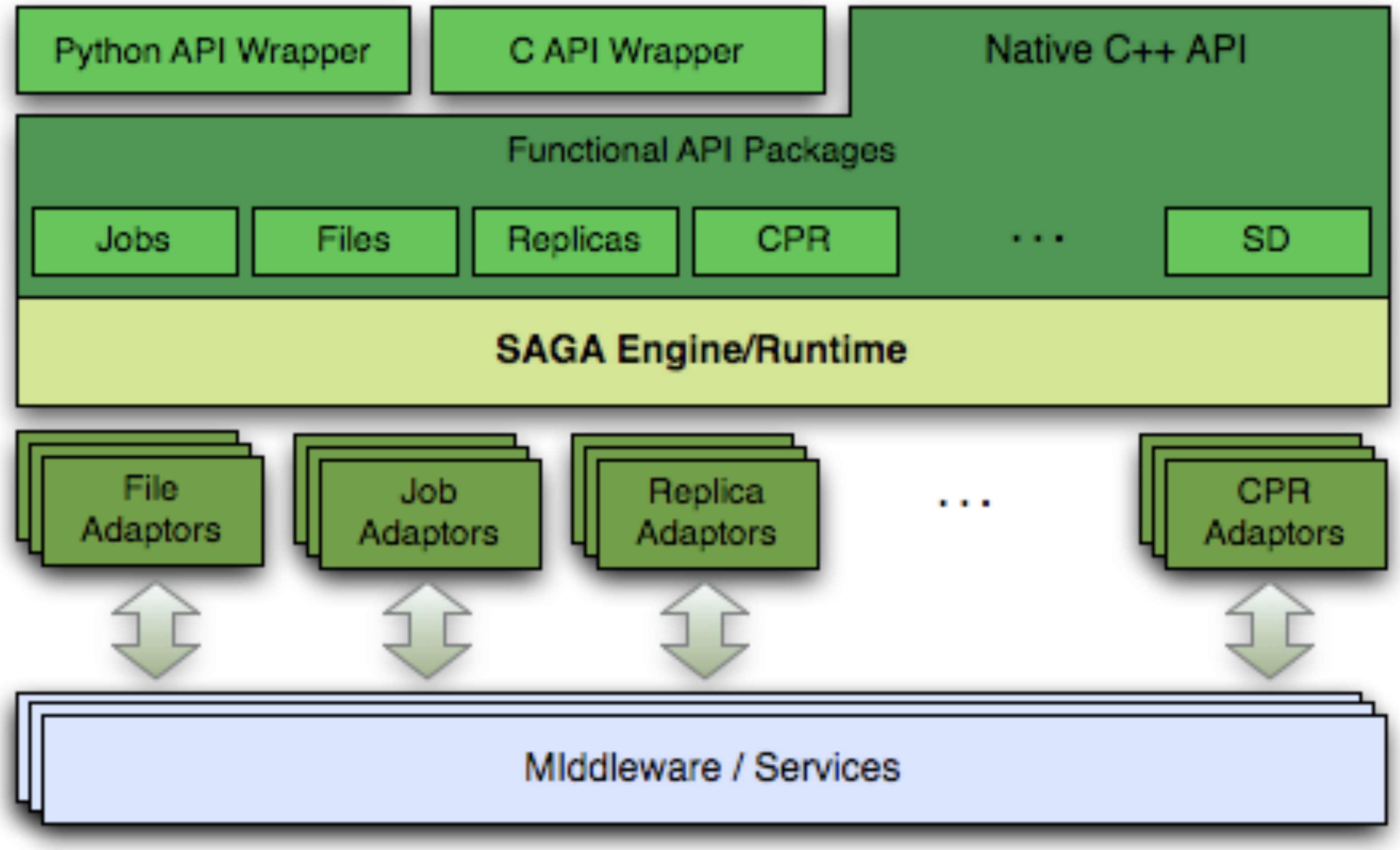
Understanding Distributed Applications

IDEAS: First Principles Development Objectives

- **Interoperability:** Ability to work across multiple distributed resources
- **Distributed Scale-Out:** The ability to utilize multiple distributed resources concurrently
- **Extensibility:** Support new patterns/abstractions, different programming systems, functionality & Infrastructure
- **Adaptivity:** Response to fluctuations in dynamic resource and availability of dynamic data
- **Simplicity:** Accommodate above distributed concerns at different levels easily…

Challenge: How to develop DA effectively and efficiently with the above as first-class objectives?

SAGA: Basics



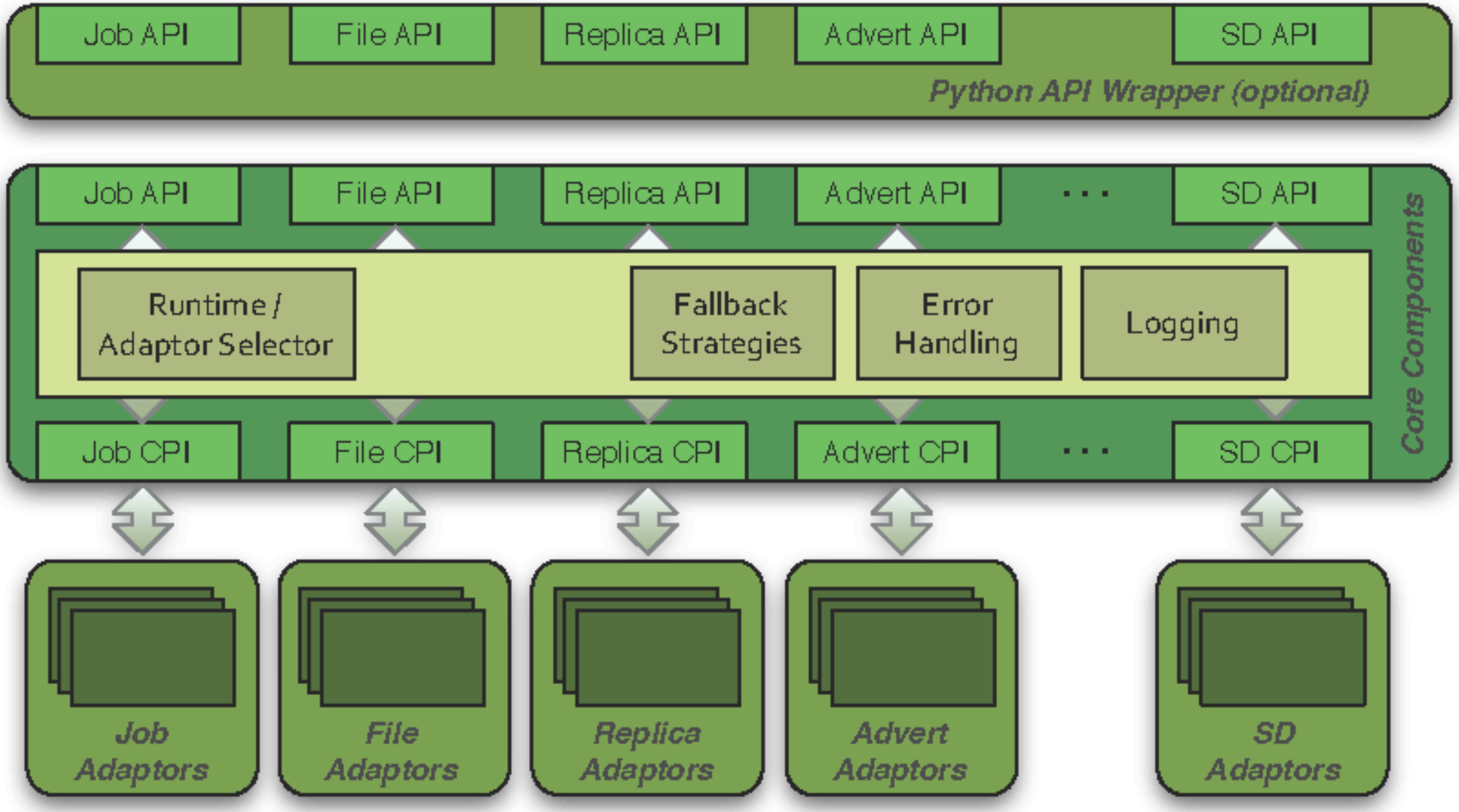
Python API Wrapper
C API Wrapper
Native C++ API
Functional API Packages
Jobs
Files
Replicas
CPR
...
SD
SAGA Engine/Runtime
File Adaptors
Job Adaptors
Replica Adaptors
...
CPR Adaptors
Middleware / Services

SAGA/CREAM Python Example

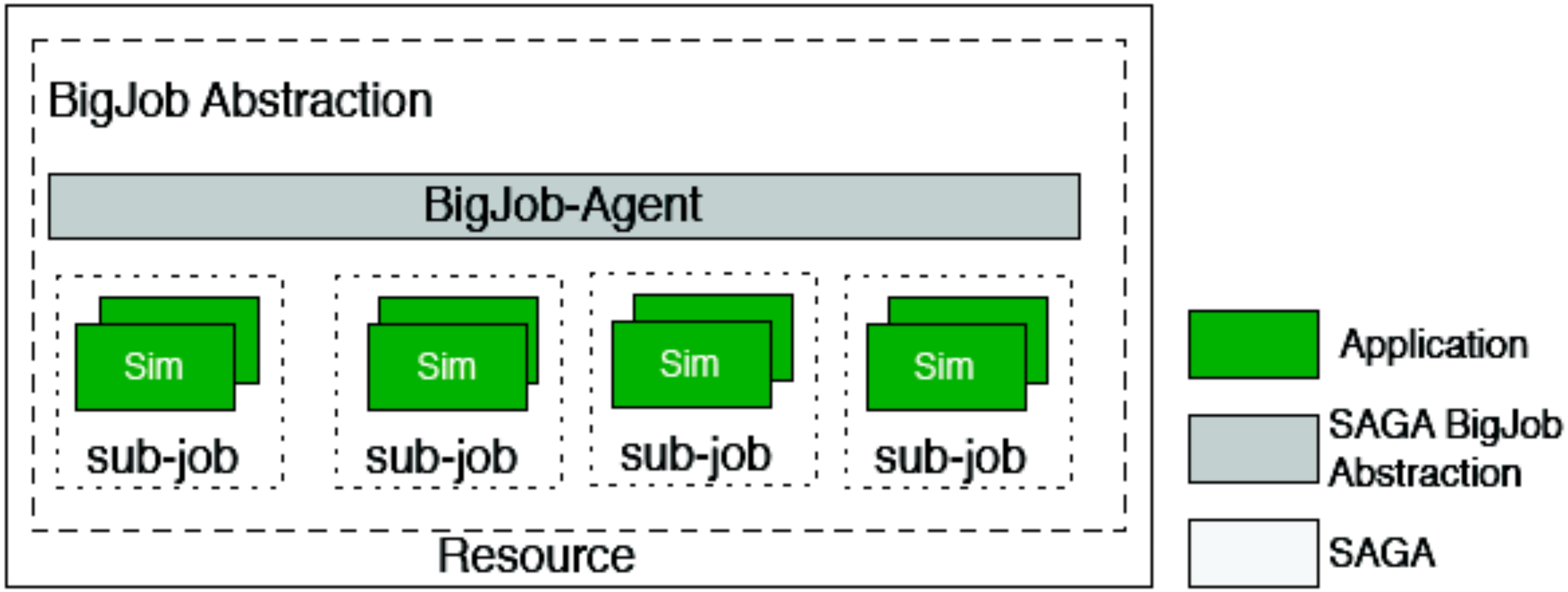
```
// Submitting a simple job to a CREAM CE
// with SAGA (C++ Example)

#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;
  }
}
```

LSU CENTER FOR COMPUTATION & TECHNOLOGY

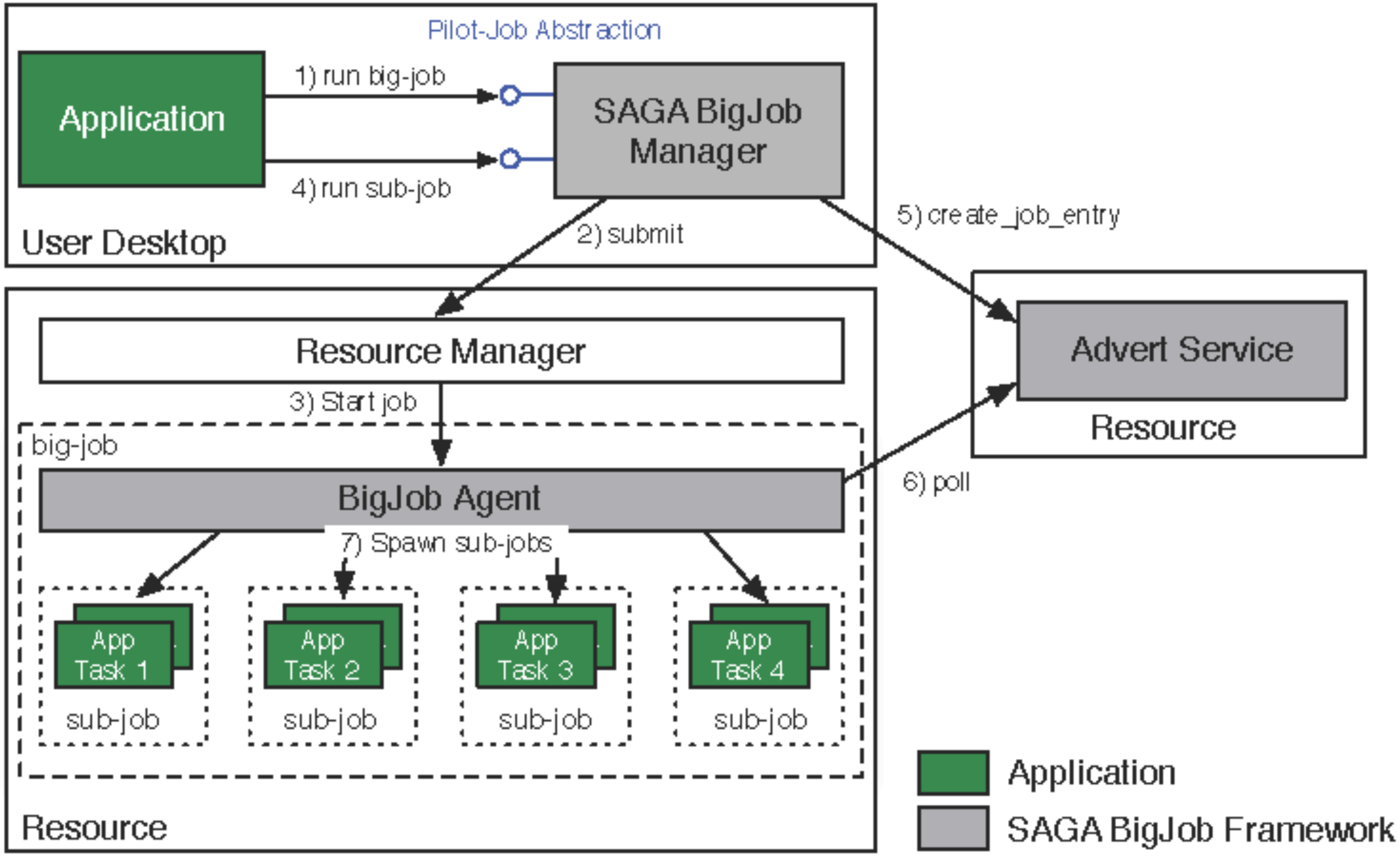
SAGA: Architecture



Abstractions for Dynamic Execution (1) Container Task



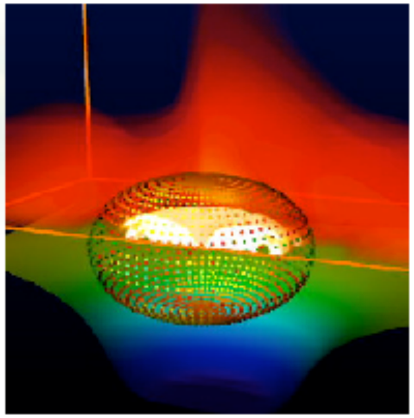
Abstractions for Dynamic Execution (2)
SAGA Pilot-Job (BigJob)

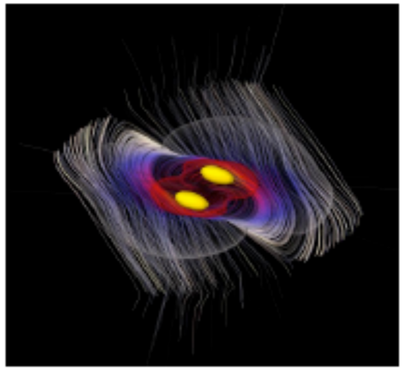


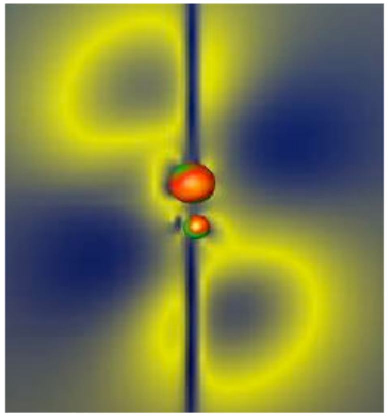
LSU
CENTER FOR COMPUTATION
& TECHNOLOGY

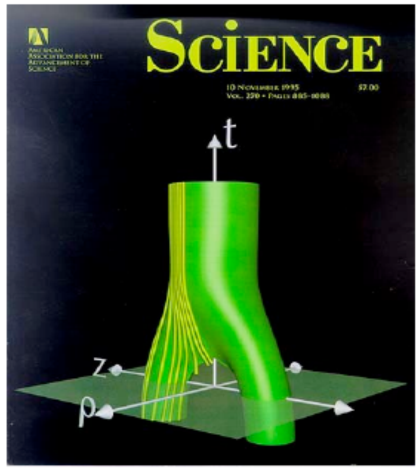
Cactus: Computational Framework

- Cactus is a modular, portable, manageable environment for parallel, high-performance simulation
    - Develop modules (components) based upon their expertise
    - No knowledge of the internals or operation of the other modules
    - Thorns = Components
- Thorns: chkpoint & restart, remote steering interface..

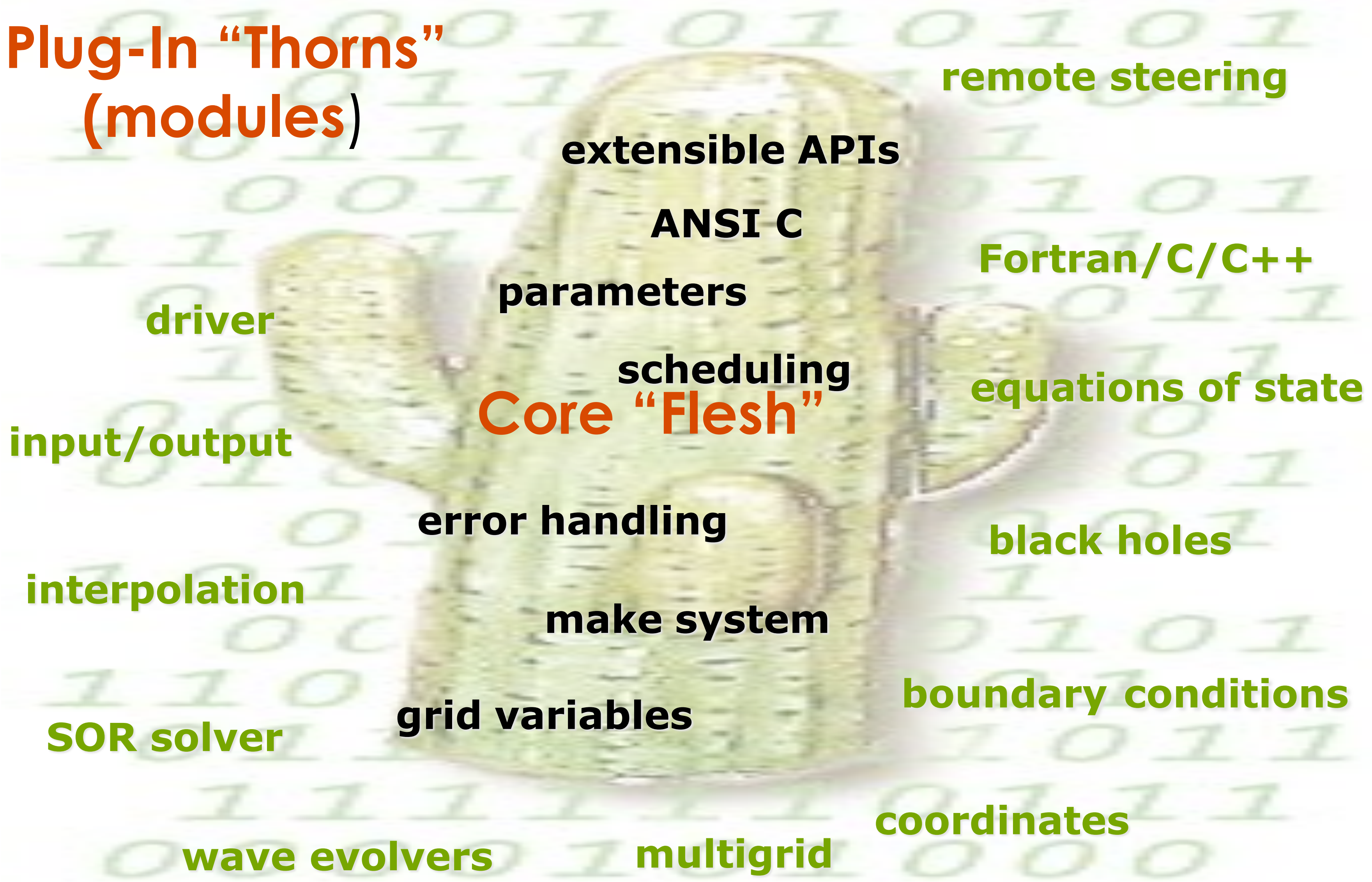








Cactus: Flesh and Thorns



CENTER FOR COMPUTATION & TECHNOLOGY

Cactus Arch: Flesh and Thorns

Cactus Flesh:

- Thorn manager; scheduling of routines of a thorn; data passing between thorns..
- Acts as utility/service library which thorns call for info or to request some action
- Contains abstracted APIs
    - Parallel operations, I/O, checkpointing...
- All actual functionality in provided by thorns

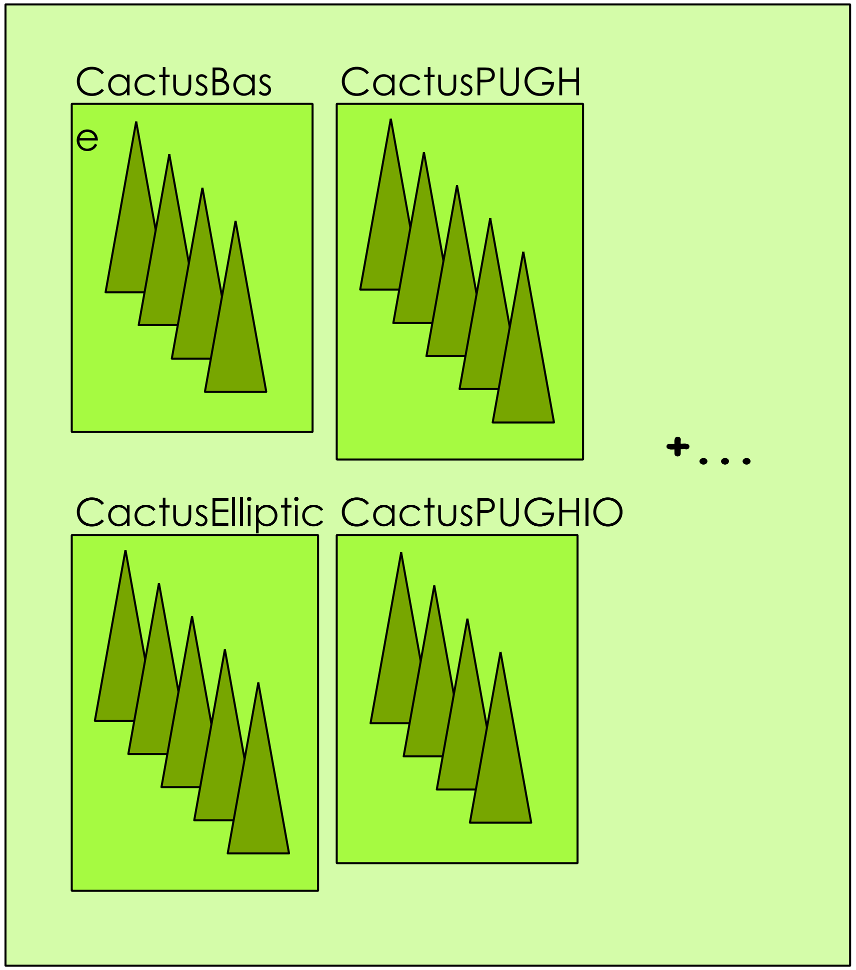
Cactus Thorns:

- Separate libraries encapsulating some specific functionality
- Different thorns can provide the same functionality; interchangable
- Can be written in any language (C, C++, F77..)

Cactus Abstractions: Thorns-Arrangements-Toolkit

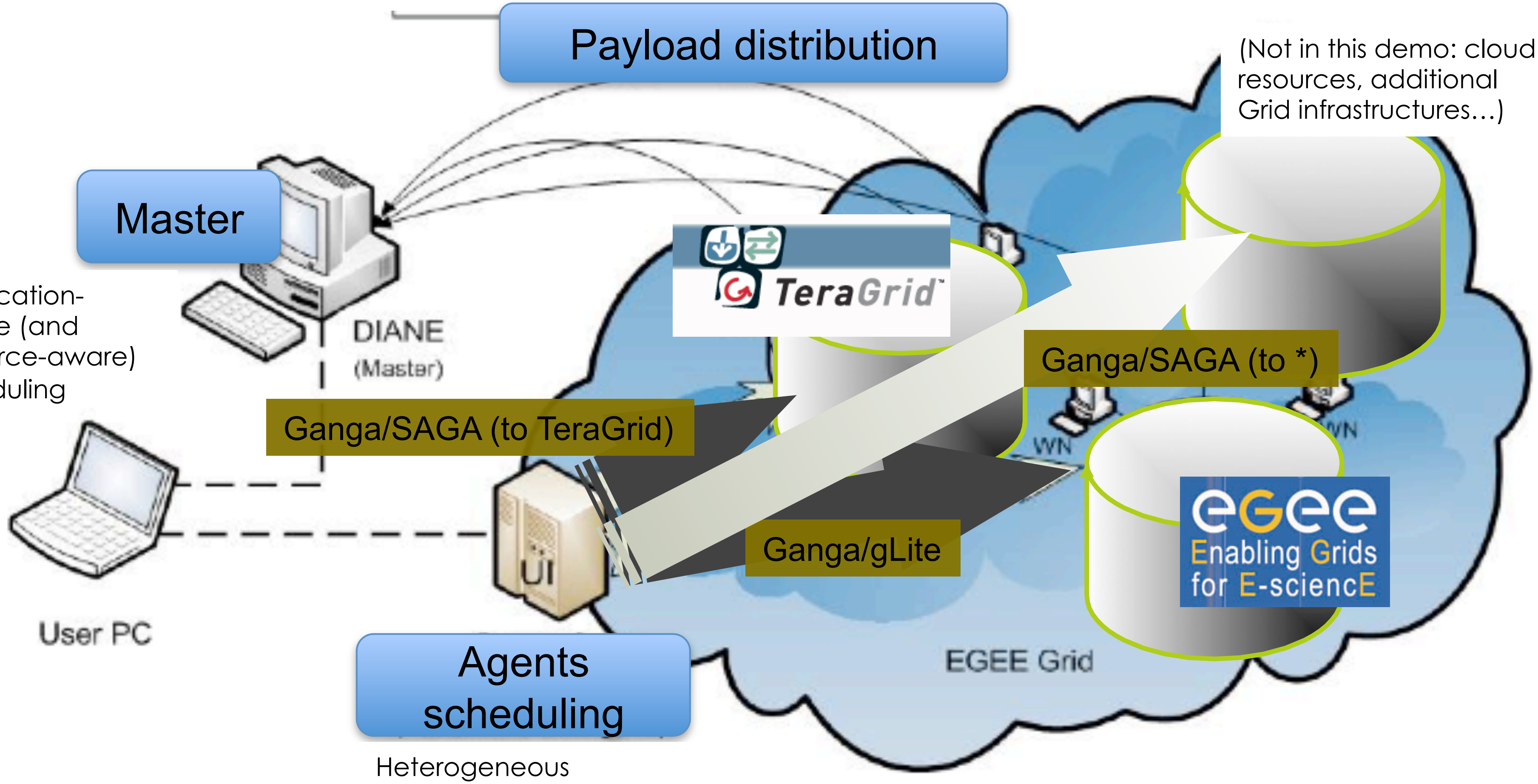
- Thorns are grouped into *arrangements* may:
  - have related functionality
  - contain everything needed for one problem
- A collection of arrangements, a toolkit e.g.
  - Cactus Computational TK
  - Cactus CFD & Relativity
  - Cactus Coastal Modelling...

Cactus Computational Toolkit



Federating resources! EGEE Conference (Apr'10)

(Not in this demo: cloud resources, additional Grid infrastructures…)



Application-aware (and resource-aware) scheduling

Heterogeneous resources allocation (Ganga + Ganga/SAGA)

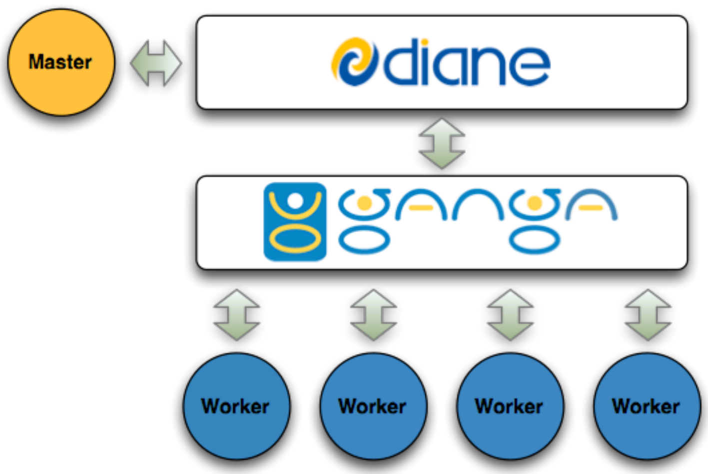
CENTER FOR COMPUTATION & TECHNOLOGY

gLite CREAM SAGA Adaptor

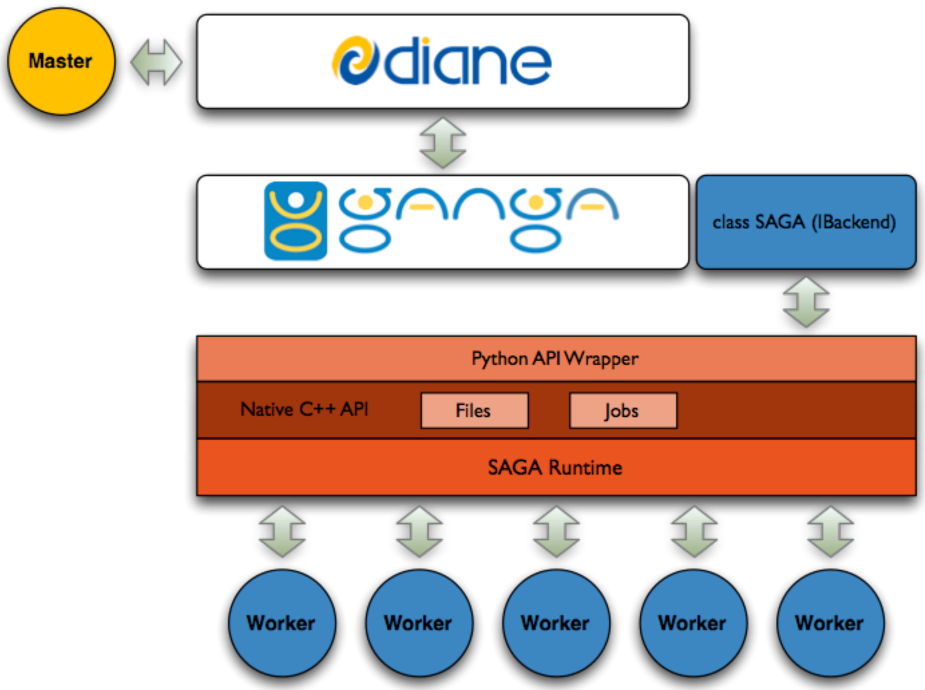
- SAGA provides a prototype *job adaptor* to access gLite CREAM CEs
- Implemented with the CREAM C++ API
- Currently provides the following features:
  - job description (saga-jd to JDL mapping)
  - job control (run, submit, suspend, cancel, etc...)
  - listing of jobs associated with a CE
  - re-connection to running jobs
  - Encapsulates (hides) proxy delegation
- The adaptor code is available via SVN: https://svn.cct.lsu.edu/repos/saga-adaptors/glite/

DIANE INTEGRATION

Diane without SAGA



Diane with SAGA



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

CE!

EUROPEAN MIDDLEWARE INITIATIVE

Service Discovery

SAGA Service Discovery

The SAGA Service Discovery API provides a uniform interface to access service details published by the information system.

C++ Users Guide
Java Users Guide
Java Doc

SAGA Information System Navigator

The SAGA Information System Navigator API provides a uniform interface to access additional details published by the information system.

C++ Users Guide
Java Users Guide
Java Doc

Web Interface

Information Navigator

Specification

SAGA
Service Discovery
Information System Navigator - work in progress

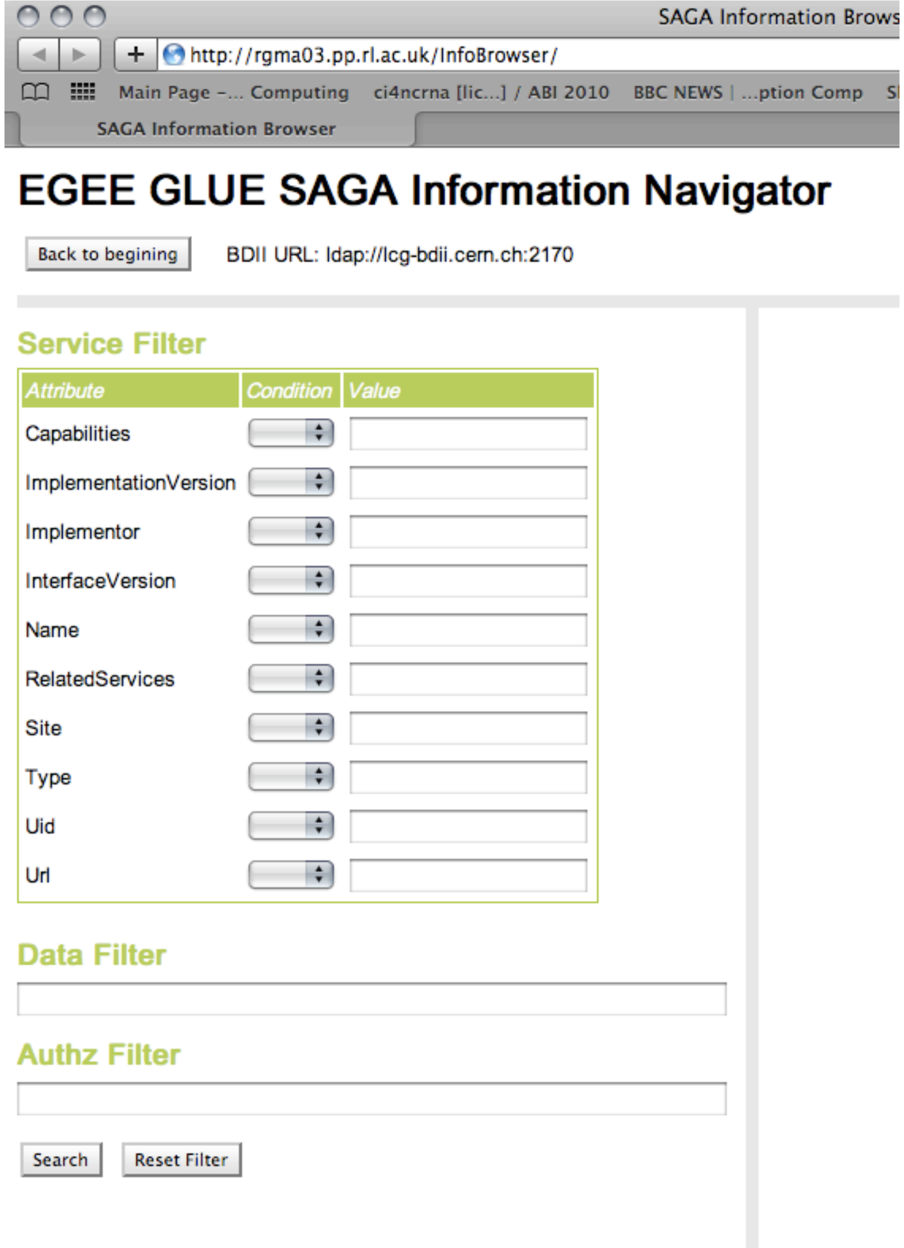
Design Notes

Service Discovery
Information System Navigator

Software Verification and Validation

Plan
Reports

CENTER FOR COMPUTATION
& TECHNOLOGY



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@ercim.org**

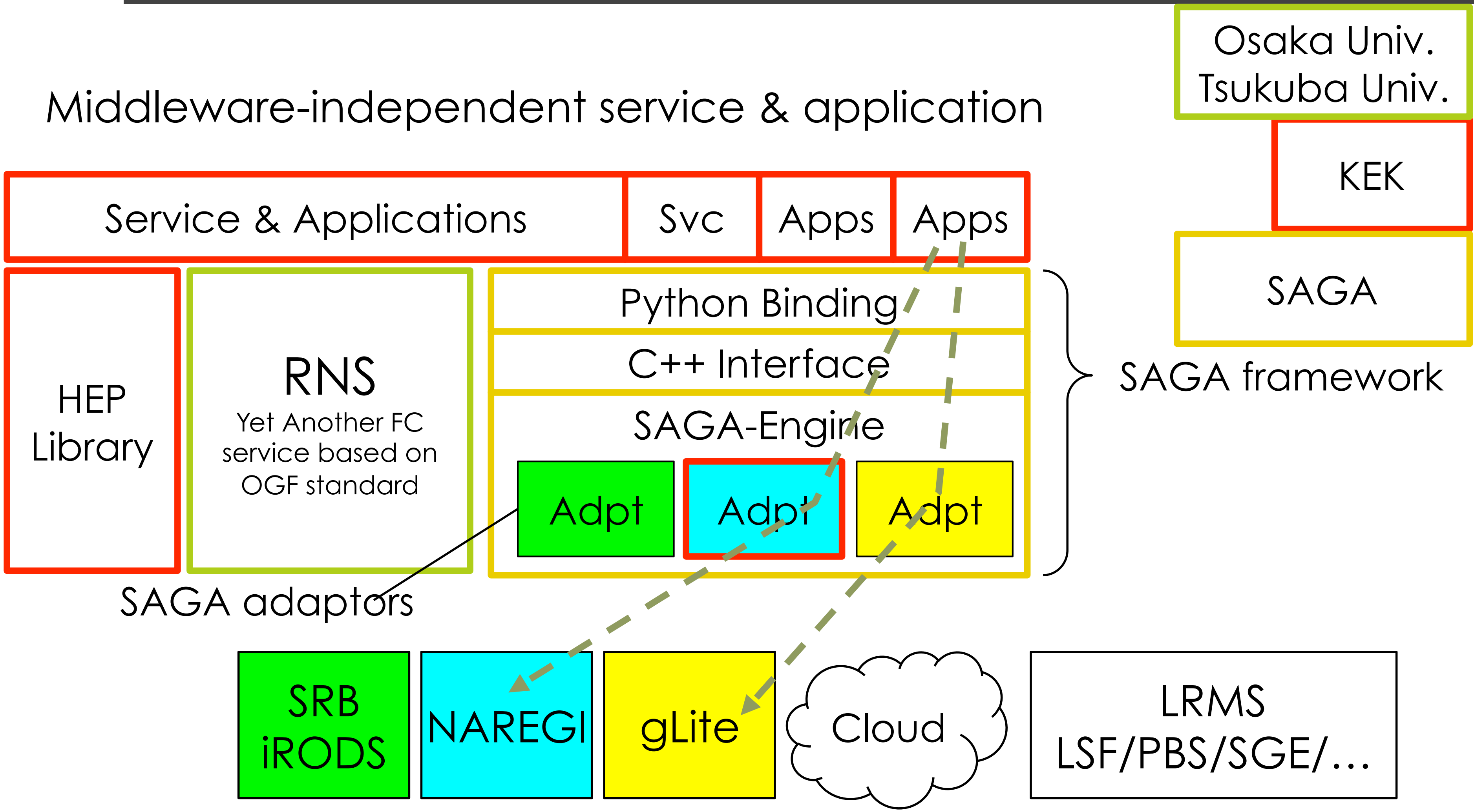
LONI-TeraGrid-DEISA Interoperability Project

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

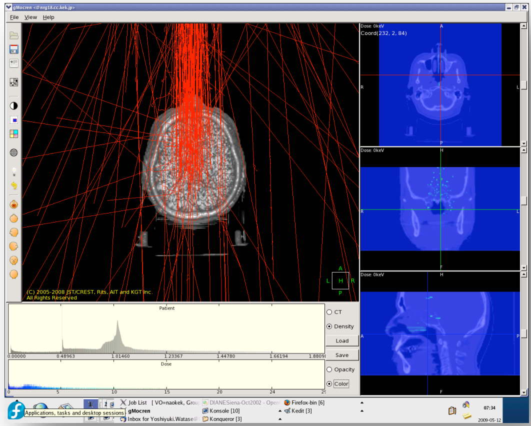
RENKEI Project Aims

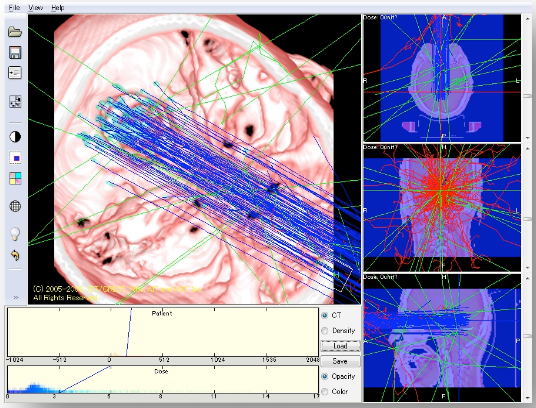


This activity is funded by MEXT as a part of RENKEI project which develops seamless linkage of resources in the Grids and the local one for e-Science.

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





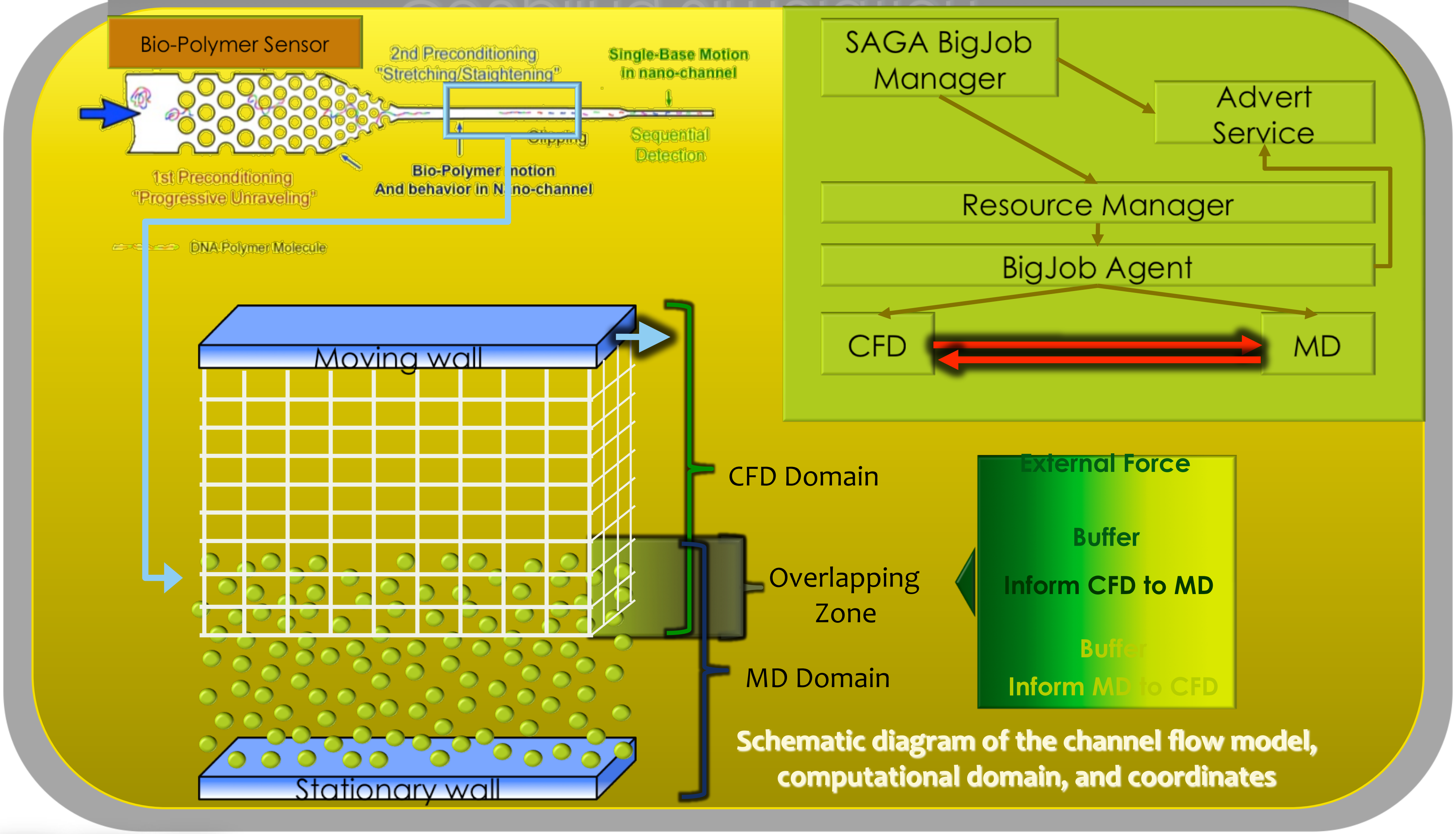
More application-wise development in 2010

CENTER FOR COMPUTATION & TECHNOLOGY

ExTENCI: 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

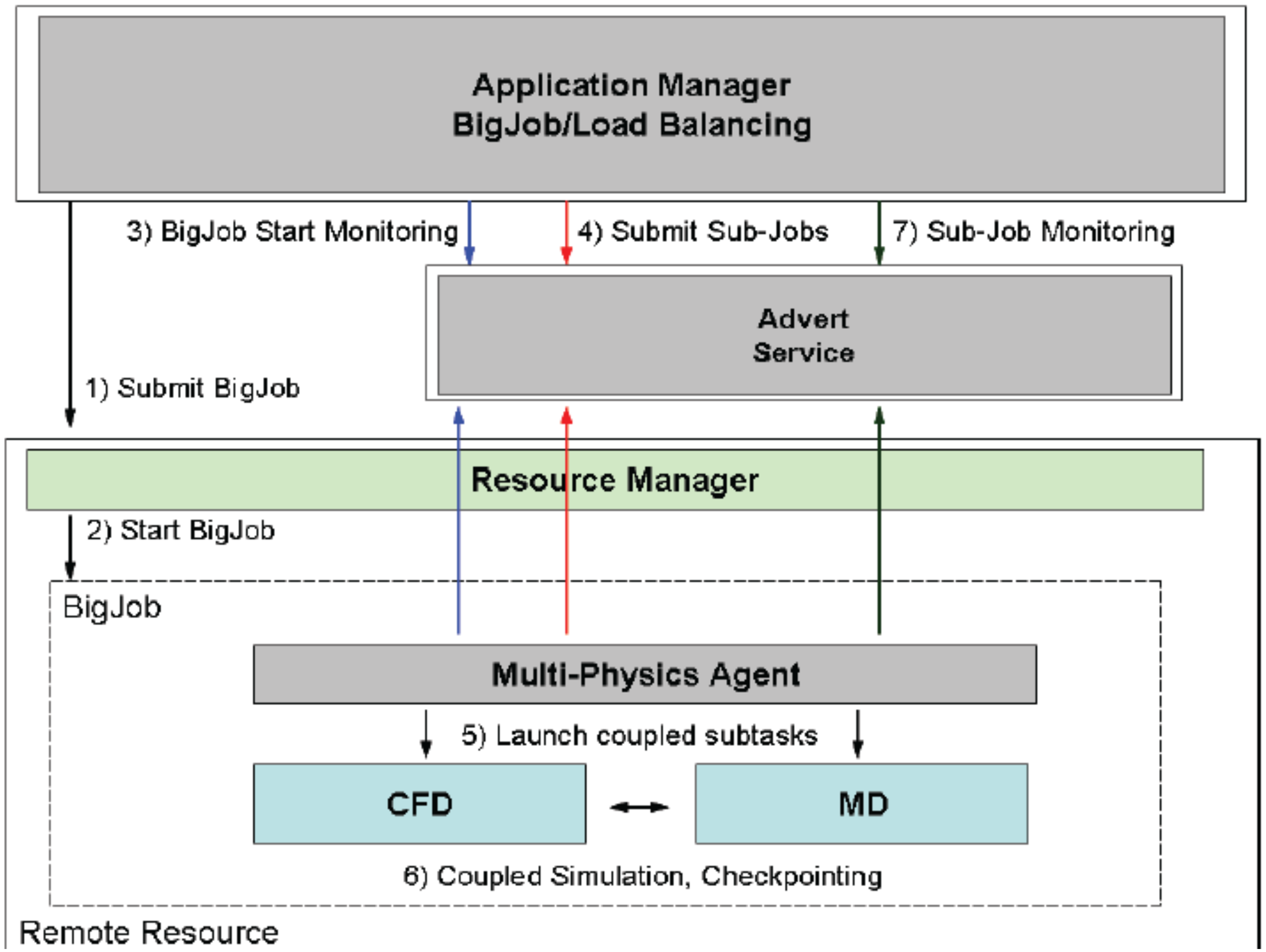
Atomistic-Continuum Hybrid Coupling Simulation



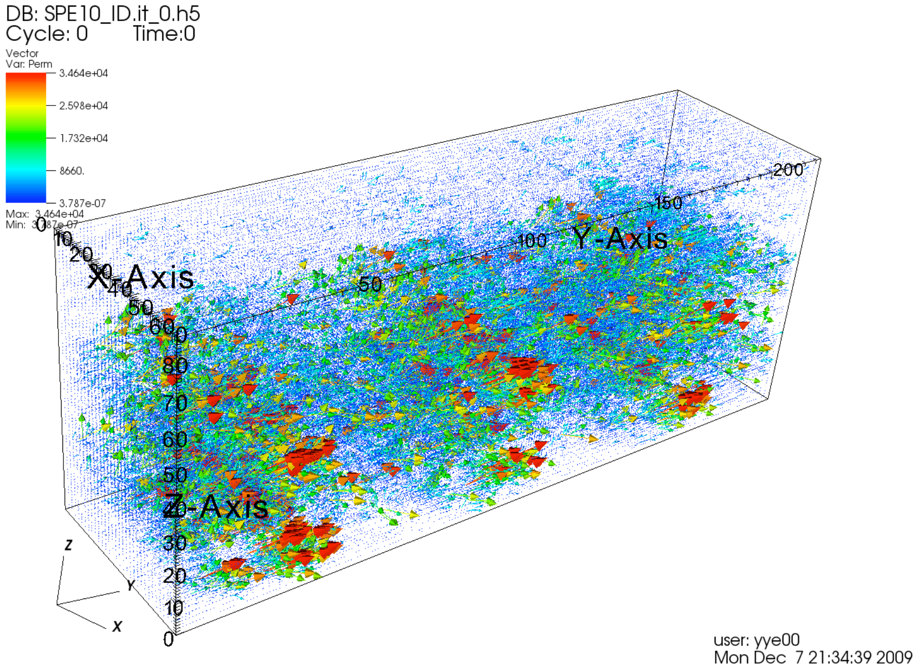
Schematic diagram of the channel flow model, computational domain, and coordinates

Multi-physics Execution Framework

❖ Structure of Multi-physics Execution Framework

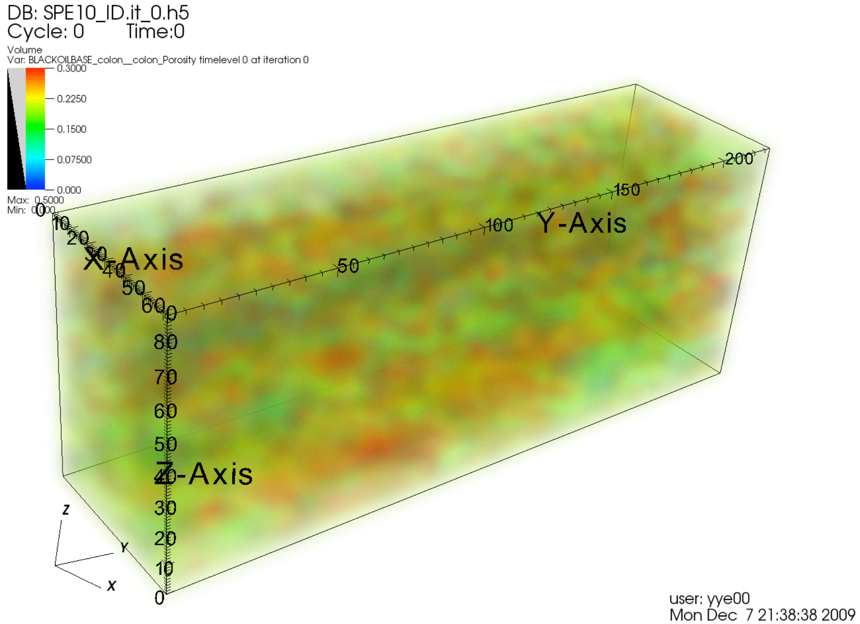


Characterizing Reservoirs: Permeability and Porosity



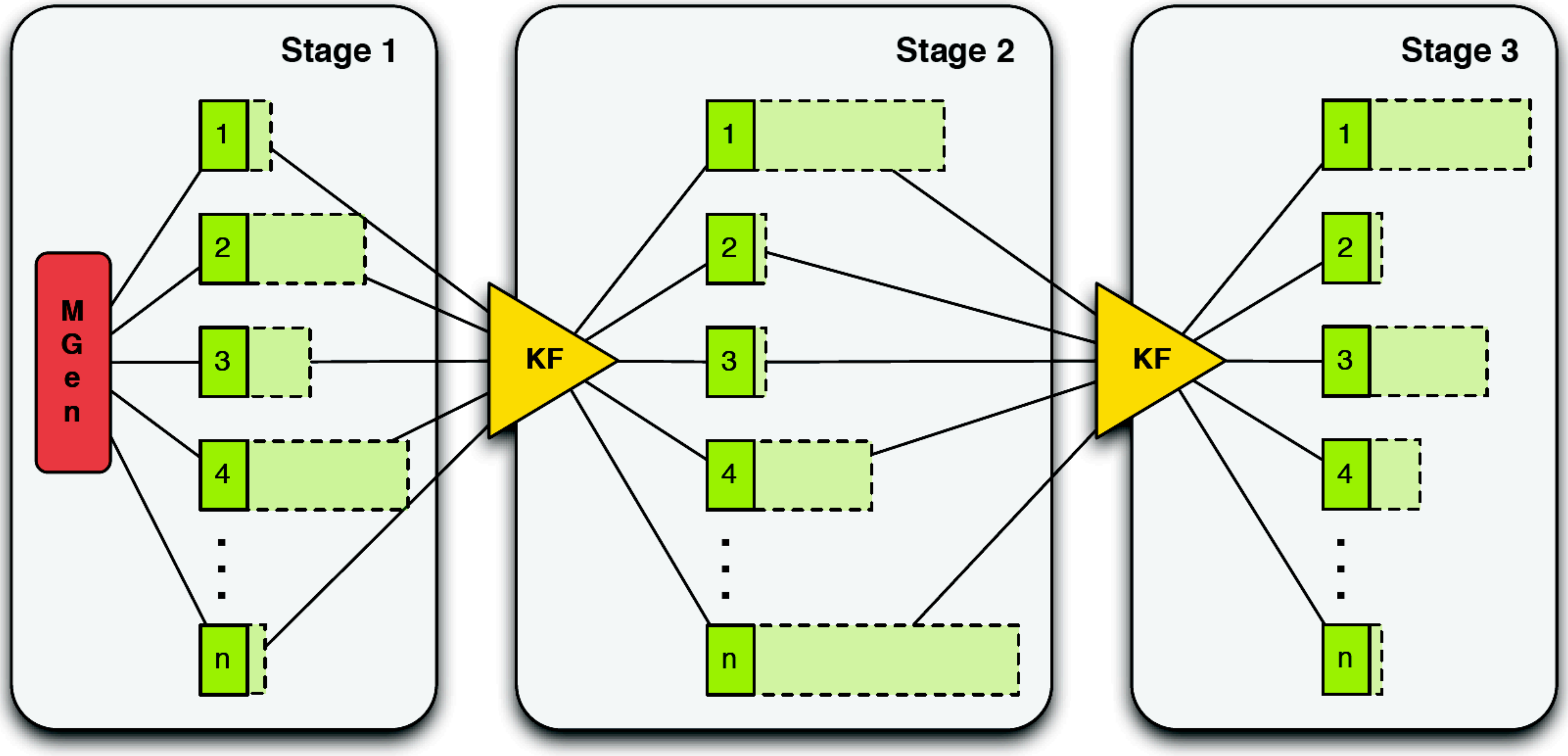
Porosity: Measure of capacity (buckets)

Permeability: Measure of flow (pipes)



LSU CENTER FOR COMPUTATION & TECHNOLOGY

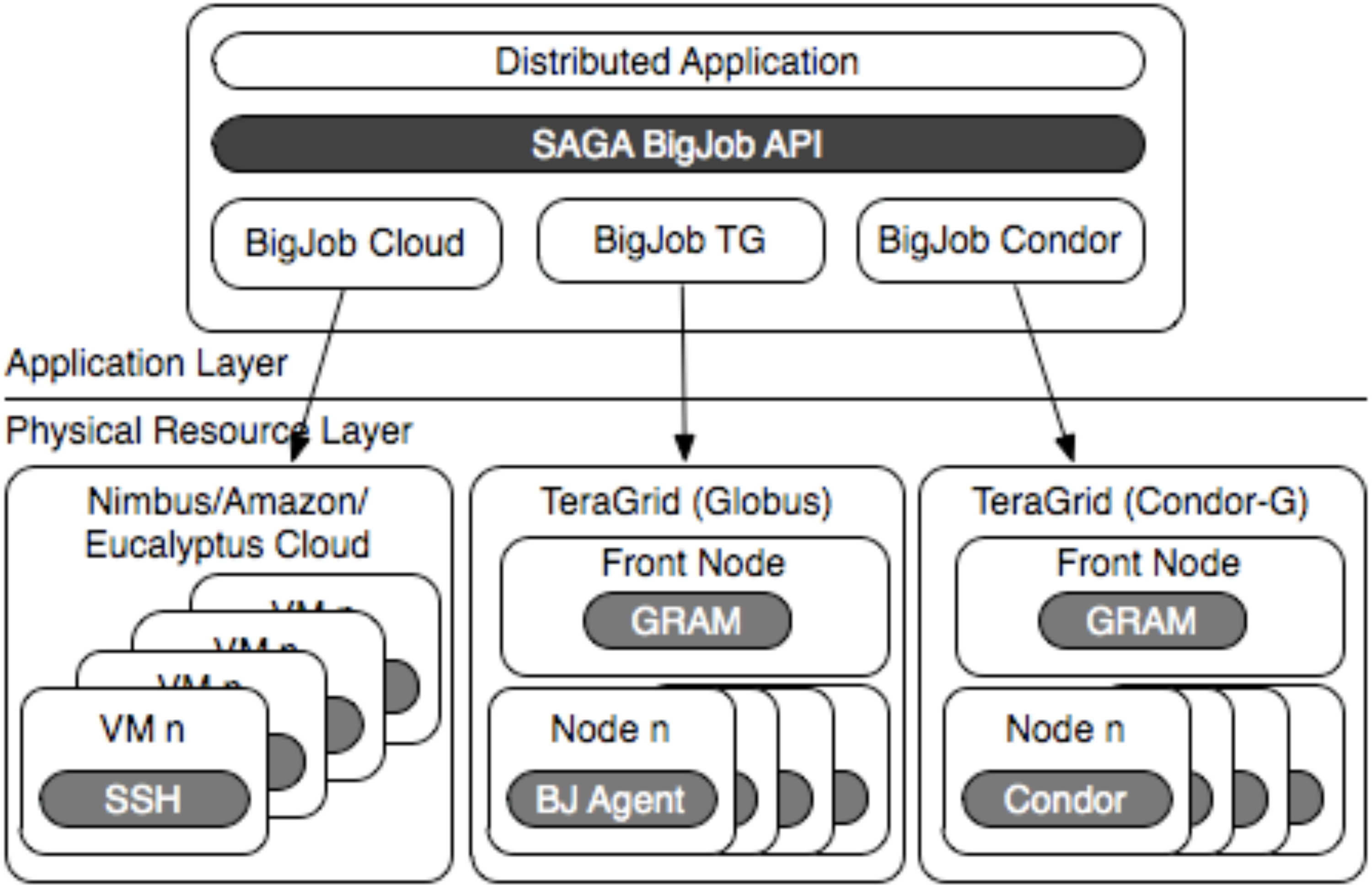
The Computation Problem



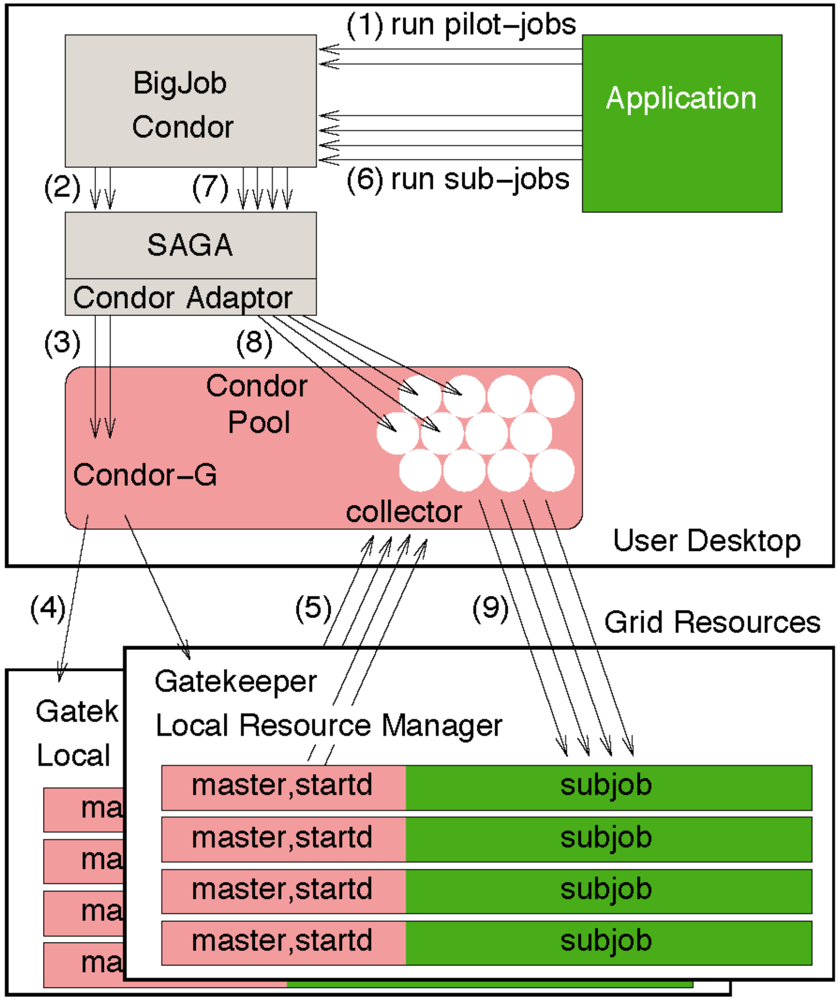
Computational Workload

- Ideally, an infinitely large ensemble would give infinitesimally small data assimilation errors.
  - In practice, an ensemble of 100~200 members is common
- For 100 ensemble members:
  - Each ensemble member (i.e. simulation) is a full 3D simulation, with anywhere from thousands to millions of grid cells
  - 1 million grid cell simulation, running on a single Ranger node for one forecast stage, requires roughly 2.5 hours and covers 300 time iterations
  - An ensemble of 100 such simulations, running for a total (15 matching+15 forecast+15 sequestration simulation years)*12 (month in a year) *2.5 (wall clock hours per simulation month)*16 (cores per node) * 100 (ensemble members) or 2,160,000 SUs on Ranger
- Tradeoffs and Improvements are possible and obviously necessary:
  - faster simulator, reduced wait time, reduced failure time, improved convergence and so on

BigJob: Infrastructure Independent Pilot-Job



BigJob: Preserving Glide-in Semantics and Interface



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

LSU CENTER FOR COMPUTATION & TECHNOLOGY

Job Overlay proposed Workplan

- Year 1: Development:
  - SAGA Interface – Condor Backend
  - SAGA BigJob Interface – Glide-in Integration
- Year 1: Validation
  - Command line SAGA-Bigjob Interface
  - Concurrent TG + OSG with Cactus application(s)
- Year 2: Development
  - Integration with Cactus Gateway
- Year 2: Validation and Extensibility
  - Scalability etc

CENTER FOR COMPUTATION
& TECHNOLOGY

A Fresh Perspective on Distributed Scientific Applications and Cyberinfrastructure

Shantenu Jha

CERN Computing Seminar, 08 Jan, 2010

http://saga.cct.lsu.edu

omii-uk
www.omii.ac.uk