The CMS Grid Analysis Environment GAE

(The CAIGEE Architecture)

CMS Analysis – an Interactive Grid Enabled Environment

See http://ultralight.caltech.edu/gaeweb

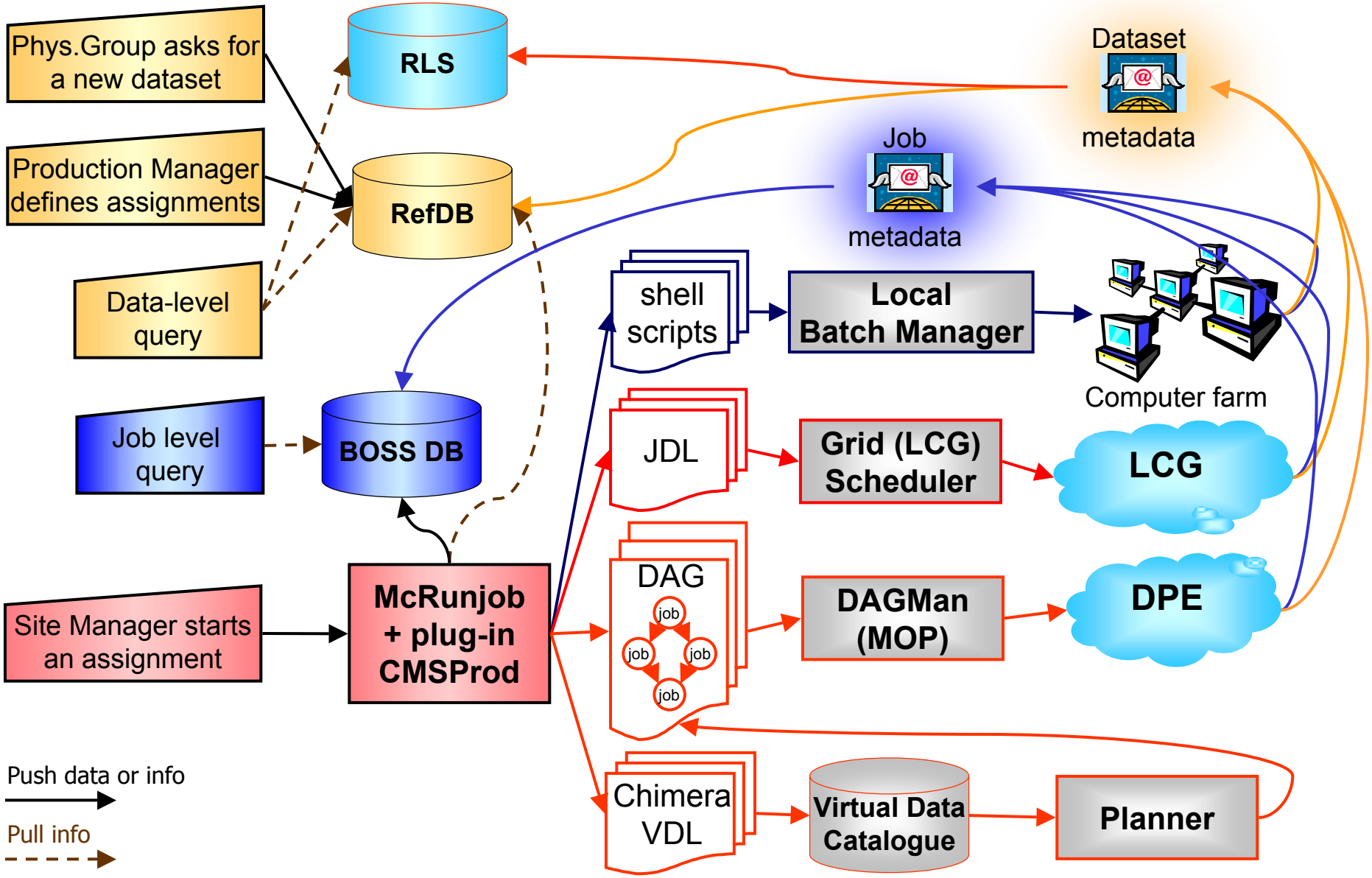
Outline

- The Current Grid System used by CMS
- The CMS Grid Analysis Environment being developed at Caltech, Florida and FNAL
- Summary

Current CMS distributed system

- Supports “traditional” and “Grid” productions
  - OCTOPUS is the glue
- Grid systems use VDT and LCG services
  - middleware provides
    - AAA
    - Information System and Information providers (including CE and SE)
    - Data Management System and File Catalogues
    - Workload Management System
    - Monitoring
  - CMS has developed (within OCTOPUS):
    - Job preparation (splitting) tools (McRunJob and CMSProd)
    - Job-level monitoring (BOSS)
    - Dataset Metadata Catalogue and Data Provenance System (RefDB)
    - Software packaging and distribution tools (DAR)

OCTOPUS Production System



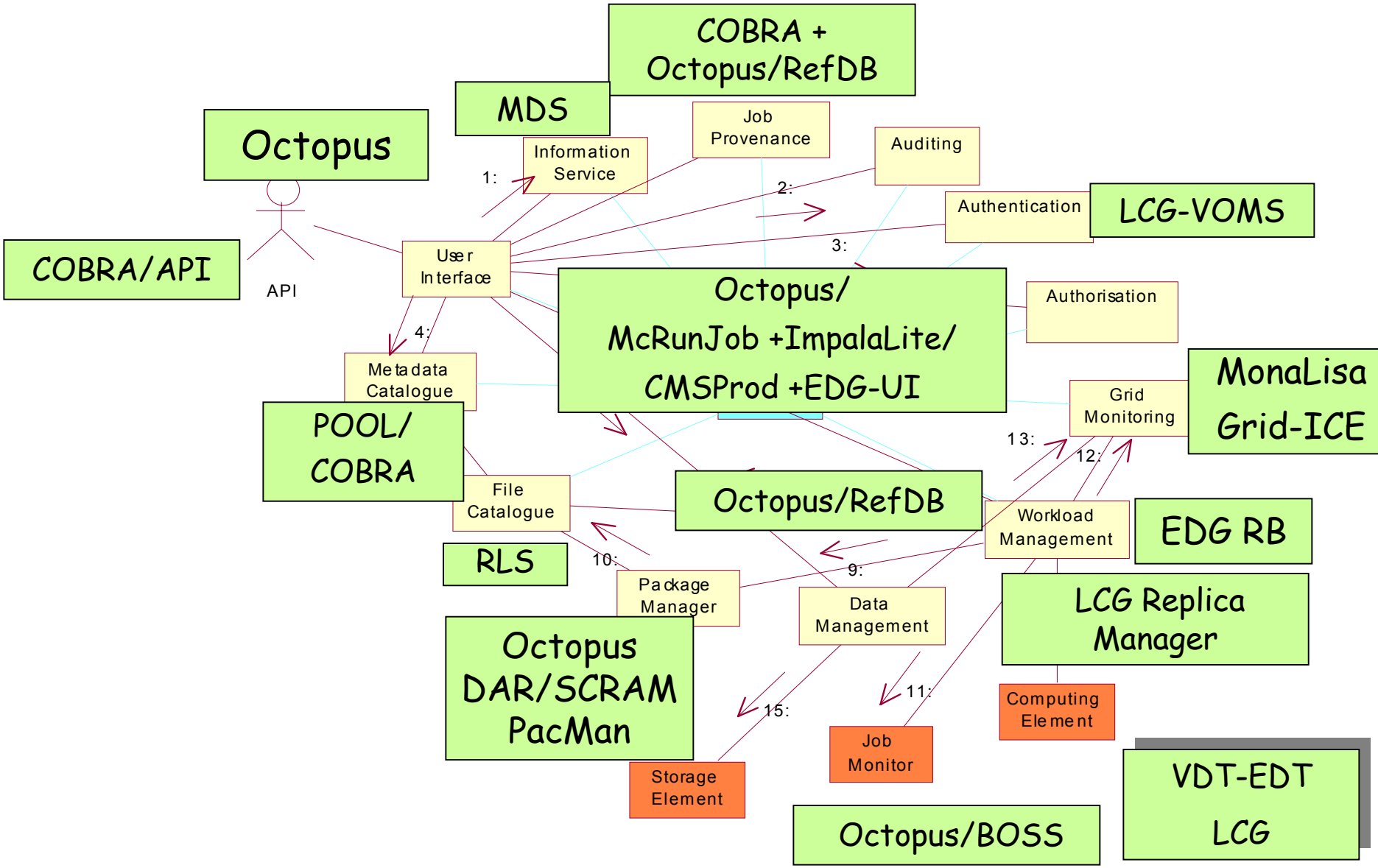
Support for distributed analysis (I)

- CMS software framework (COBRA) supports creation:
  - of Tags (datasets with direct pointers to selected events)
  - of reclustered datasets (deep copies of events)
  - of private datasets (redefinition of events)
- COBRA now uses POOL
  - uses POOL catalogue to store part of the dataset metadata
  - directly interfaced to a distributed file catalogue (RLS)
  - supports automatic download of remote files at run-time
- GFAL-enabled version of COBRA to directly access data on SRM-based (local) Storage Elements
- Developing a system for private software distribution
  - based on SCRAM
- Modifying OCTOPUS components to use grid AAA
  - RefDB, BOSS

Support for distributed analysis (II)

- Job preparation
  - RefDB has concepts of *requests* and *assignments*
    - are equivalent to a high-level query on metadata
    - optimized for production
    - allows automatic preparation of jobs for a variety of environments
  - CMS is extending the *request* object to support user queries on metadata
  - if a common Dataset Metadata Catalogue is provided by LCG, CMS can migrate part of the functionality of RefDB
    - otherwise continue to use the CMS specific implementation
  - CMS is already using DAG's on VDT. EDG (LCG-*N*) will support DAG submissions through the Resource Broker

CMS grid components and ARDA

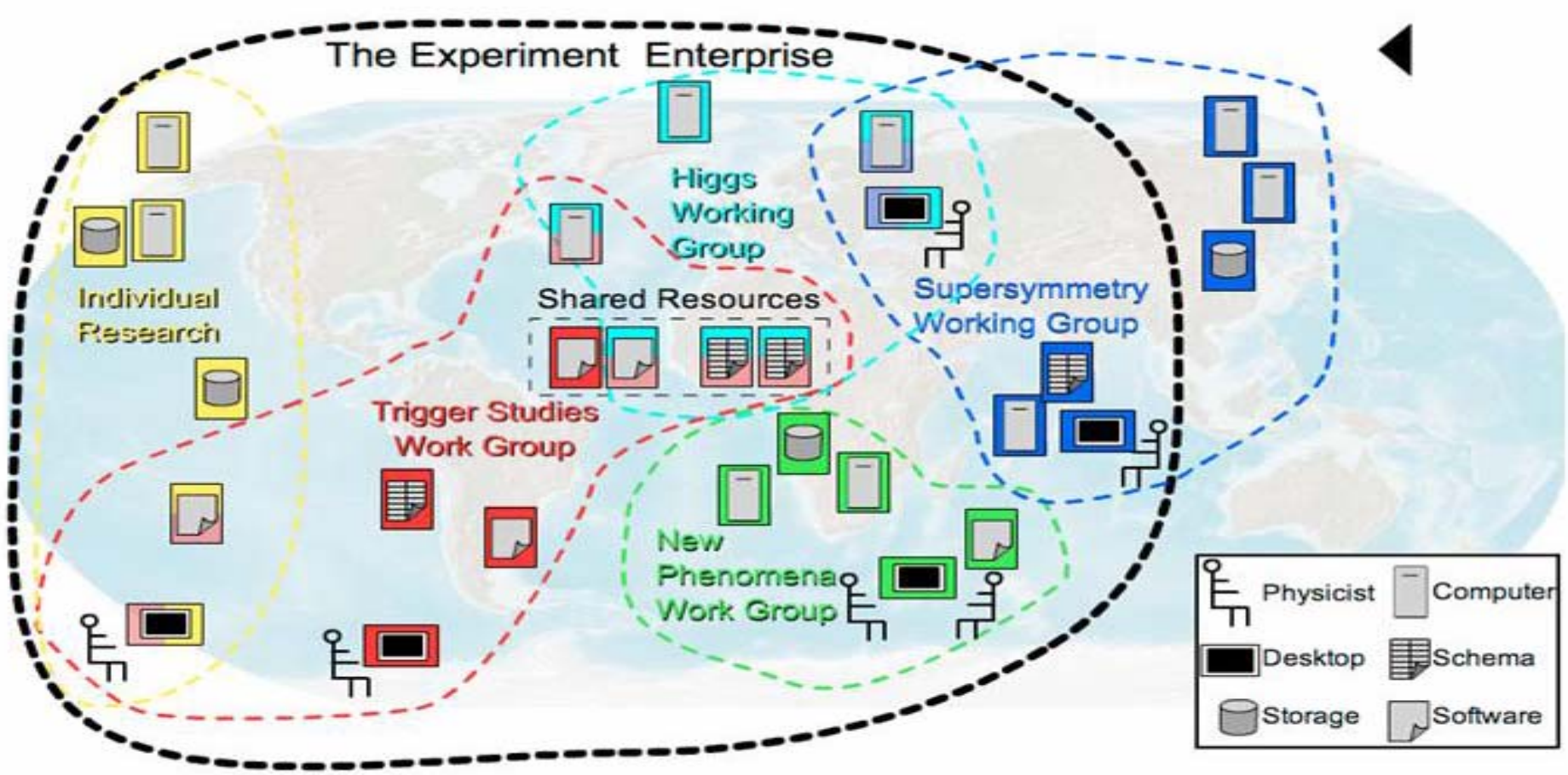


The Grid Analysis Environment

(GAE)

The Importance of the GAE

- The GAE is critical to the success or failure of physics, and Grid-based LHC applications:
  - Where the physics gets done
  - Where we learn how to collaborate globally on physics analyses



GAE Challenges

- The real “Grid Challenge” lies in the development of a physics analysis environment that integrates with the Grid systems
  - To be used by diverse worldwide communities of physicists; sharing resources within and among global “Virtual Organizations”
  - 100s - 1000s of tasks, with a wide range of demands for computing, data and network resources
  - Needs a coherent set of priorities, taking Collaboration strategy and policy into account
  - Needs Security
- To successfully construct and operate such a global system, we need
  - To build Grid End-to-End Service and Grid Application Software Layers
  - To Monitor, Track and Visualise the system end-to-end
  - To Answer the question:
    How much automation is possible. How much is desirable?

These challenges motivated the CAIGEE Project proposal, in Early 2002.

What does the GAE Address ?

- The full continuum of analysis tasks, from trivial (interactive) response to many days execution. It is especially focussed on:
  - ➔Low latency, Quick turnaround
  - ➔Unpredictable, sometimes chaotic system demands
  - ➔Smooth transition to longer, more “batch”-like tasks
- The seamless integration of existing analysis tools into the GAE
- The needs of CMS analysis, but with a strong intent to be useful in other LHC Grid environments.

GAE Development Strategy

- Build on the existing CS11 (PPDG Analysis group) requirements, HEPCAL (single user) use cases, & ARDA (Roadmap) documents
- Work with, and use results of existing efforts as much as possible
- Adapt and integrate tools in common use (like ROOT) in the GAE
- Support full spectrum of analysis tasks: from interactive to batch/production analysis
  - Low latency, fast interactive tasks, together with longer running batch tasks.
- Design for, and show Scalability
  - E.g. with Peer to Peer architectures
- Ensure fault tolerance
  - No single point of failure, e.g. no centralised critical services
- Support Security
  - Authentication and Authorization
- Build in Adaptability
  - Exploit usage patterns and policies to optimize the overall system performance
- Support the full range of heavyweight and lightweight clients
  - Clusters, Servers, Desktops, Browsers, Handhelds, WAP phones etc.
- Strongly favour implementations that are language and OS neutral

Work on the GAE

- General Scope:
  - GAE involves several US Tier2 sites and the FNAL Tier1
  - We are willing to actively participate in ARDA
  - We have been working with the PPDG analysis group (CS11) on definition of the Grid Analysis services
- GAE portal (Clarens):
  - Python and Java versions
  - Many Clarens clients developed and deployed (e.g. ROOT, JAS, handhelds etc.)
  - Clarens servers widely deployed
    - UFL (service host for the Sphinx Grid Scheduler)
    - CERN for CMS and ATLAS
    - Fermilab
    - UCSD
    - NUST and NCP (Pakistan)
    - Upcoming: FIU, UERJ (Rio), …
- GAE Monitoring
  - We use MonALISA to monitor the Grid2003 and the global VRVS reflector network

GAE System features

- Hierarchical Dynamic Distributed System Architecture*
- Dynamic service discovery
  - Distributed Lookup
  - Dynamic host discovery
  - Distributed and automatic service lookup
  - Better able to handle single node failure
  - E.g. JINI today (working now!)
- Support for Peer to Peer Communications
  - Hierarchical and Functional peer groupings
  - Access to Result Sets, data and applications
- Web Services APIs
  - Language and OS neutral
  - Standards Based
  - Emerging de-facto standard in academia and industry
- Full Architecture documentation
  - *See http://ultralight.caltech.edu/gaeweb/gae_services.pdf*

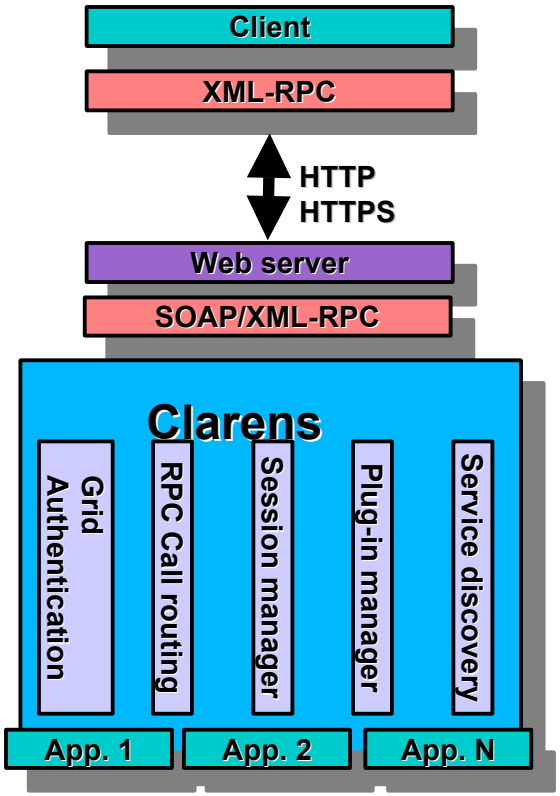
**Further info on the dynamic services architecture is available in: "Grid Computing 2002 – Making the Global Infrastructure a Reality", Chapter 39*

GAE Adopted Standards

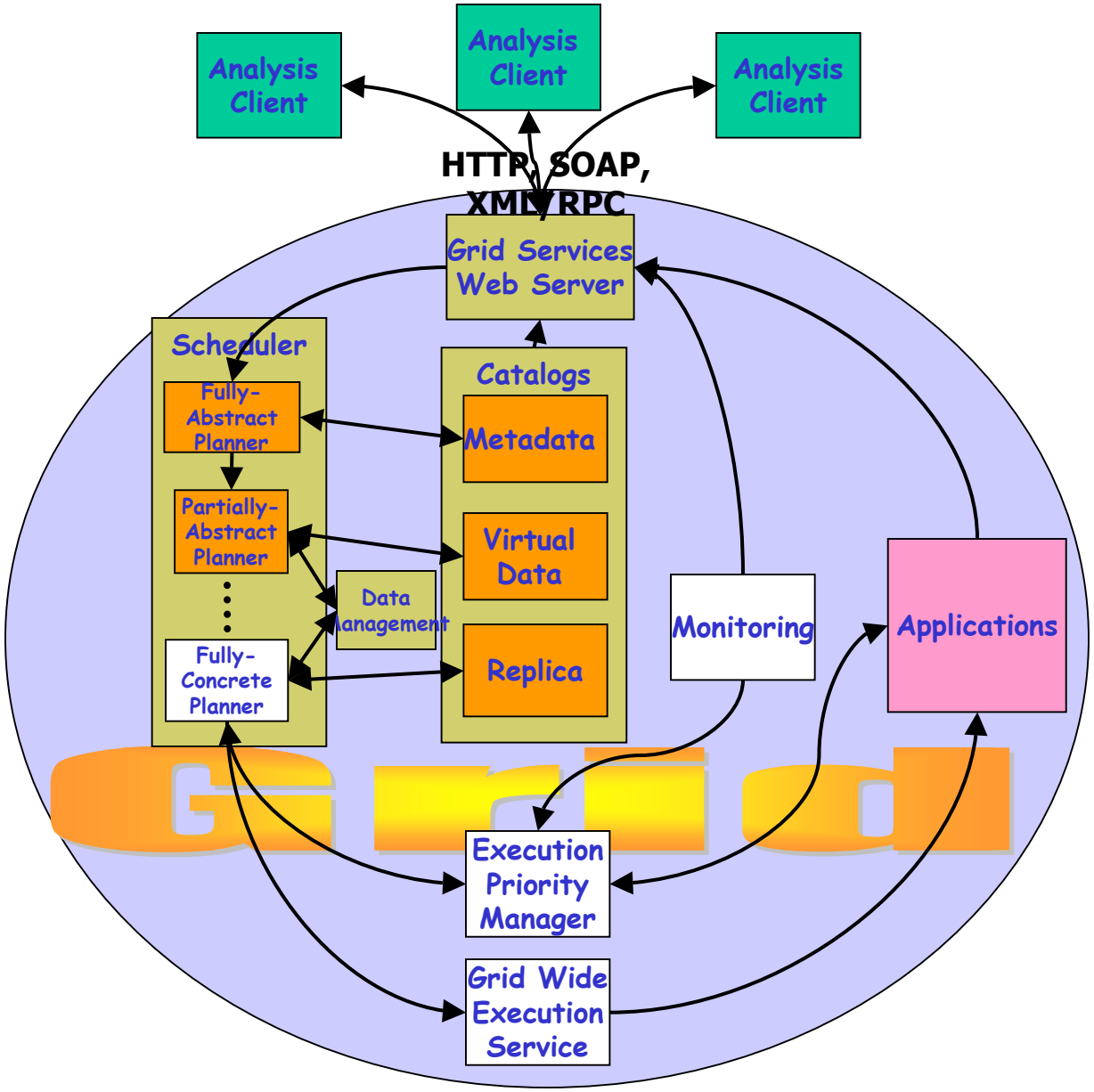
- Access Web Services using HTTP , SOAP and XML-RPC
- Describe Web Services using WSDL
- Use X509/PKI based security via HTTP, HTTPS and GSI
- GAE will track the emerging OGSA standard with the goal of becoming compliant with it

Clarens Web Services Framework

- Clarens is a Portal system that provides a common infrastructure for deploying Grid-enabled web services. Features:
  - ➔ Access control to services
  - ➔ Session management
  - ➔ Service discovery and invocation
  - ➔ Virtual Organization management
  - ➔ PKI-based security, including credential delegation
- Role in GAE:
  - ➔ Connects clients (top) to Grid or Analysis applications
  - ➔ Acts in concert with other Clarens servers to form a P2P network of service providers
- Two implementations: wide applicability
  - ➔ Python/C using the Apache web server
  - ➔ Java using Tomcat Servlets



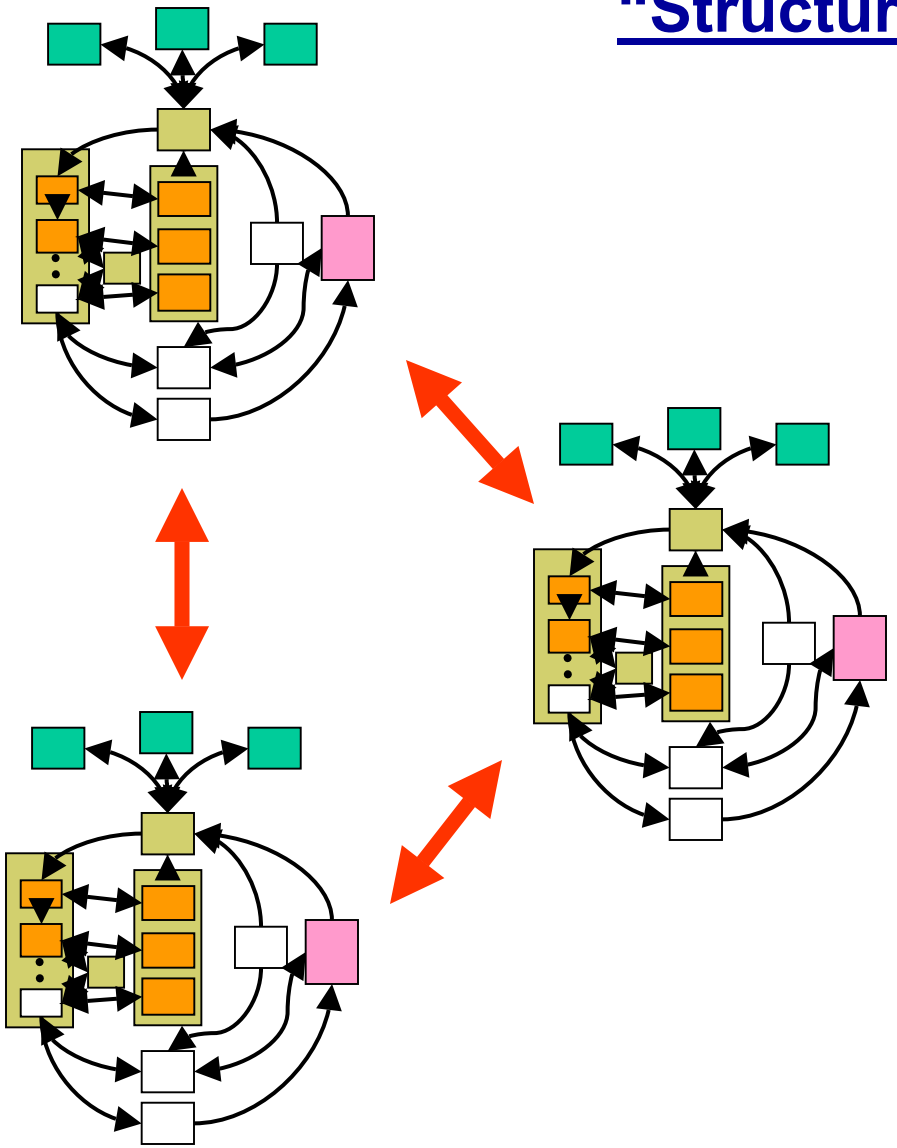
GAE Architecture



- Analysis Clients talk standard protocols to the "Grid Services Web Server", a.k.a. the Clarens data/services portal.
- Simple Web service API allows Analysis Clients (simple or complex) to operate in this architecture.
- Typical clients: ROOT, Web Browser, IGUANA, COJAC
- The Clarens portal hides the complexity of the Grid Services from the client, but can expose it in as much detail as req'd for e.g. monitoring.
- Key features: *Global* Scheduler, Catalogs, Monitoring, and Grid-wide Execution service.

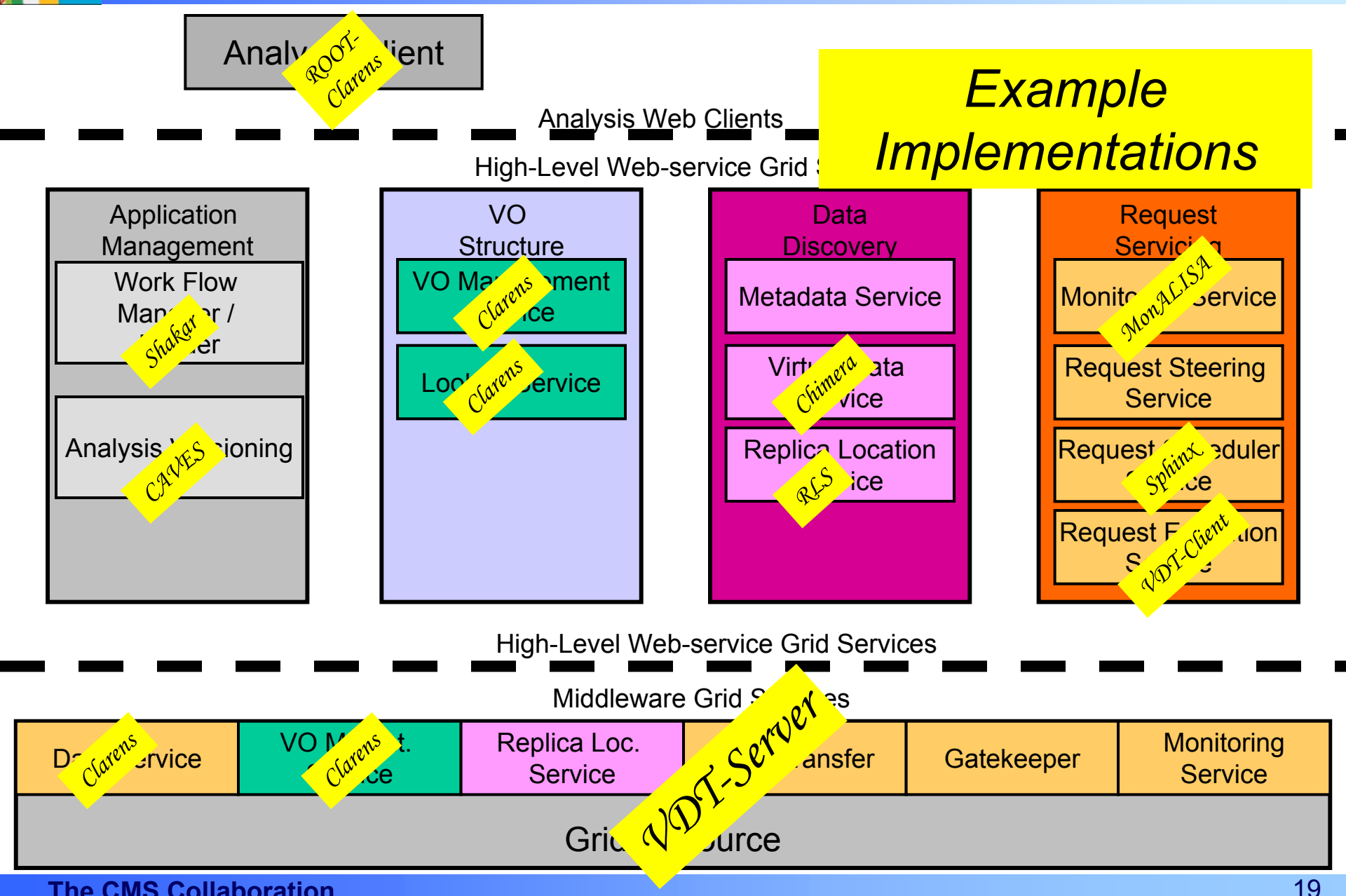
GAE Architecture

"Structured Peer-to-Peer"



- The GAE, based on Clarens and Web services, easily allows a "Peer-to-Peer" configuration to be built, with associated robustness and scalability features.
- *Flexible:* allows easy creation, use and management of highly complex VO structures.
- A typical Peer-to-Peer scheme would involve the Clarens servers acting as *"Global Peers,"* that broker GAE client requests among all the Clarens servers available worldwide.

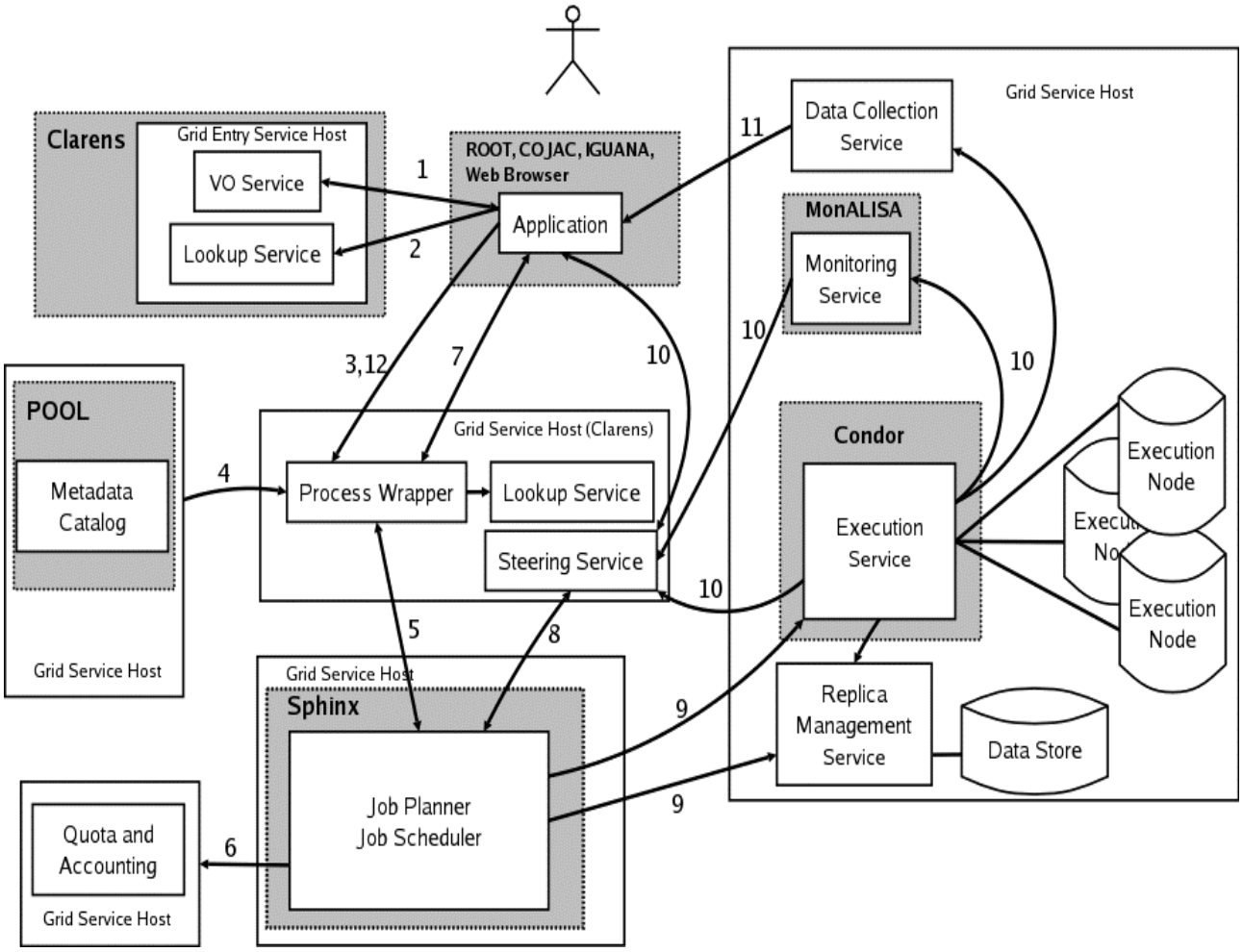
Services in the GAE Architecture



GAE Service Flow

The diagram depicts the sequence of operations and messages that occur during a Grid Enabled Analysis task execution in the GAE scheme:

- The sequence starts with the client communicating VO credentials to the Clarens server, and entrusting Clarens with the potentially complicated task of communicating with and marshalling the required Grid resources.
- Eventually the task execution results are returned, via Clarens, to the client from the Grid.
- During all this, the Monitoring service provides the user with constant feedback.



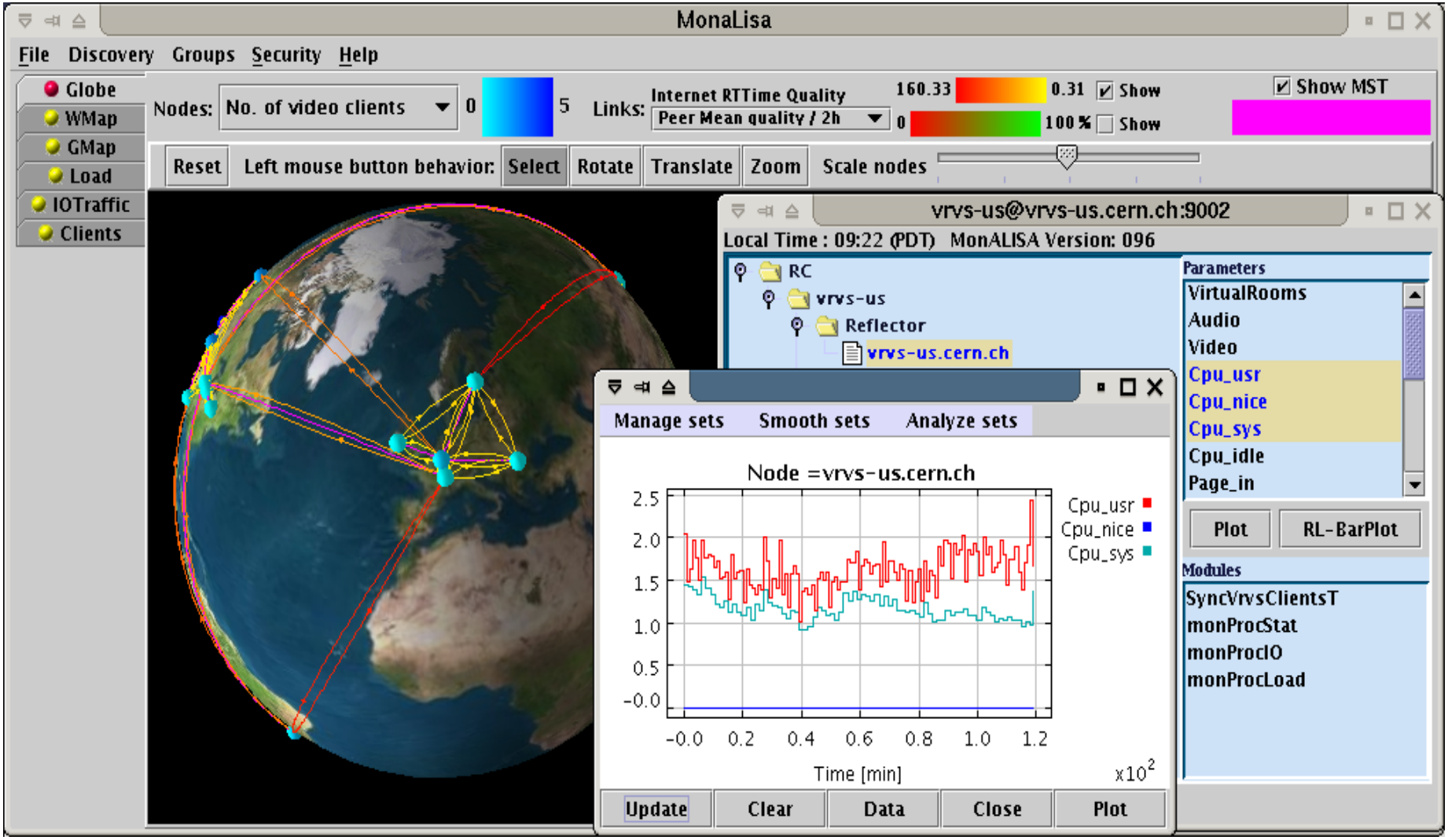
See http://ultralight.caltech.edu/gaeweb/gae_services.pdf

ARDA and GAE Ontology

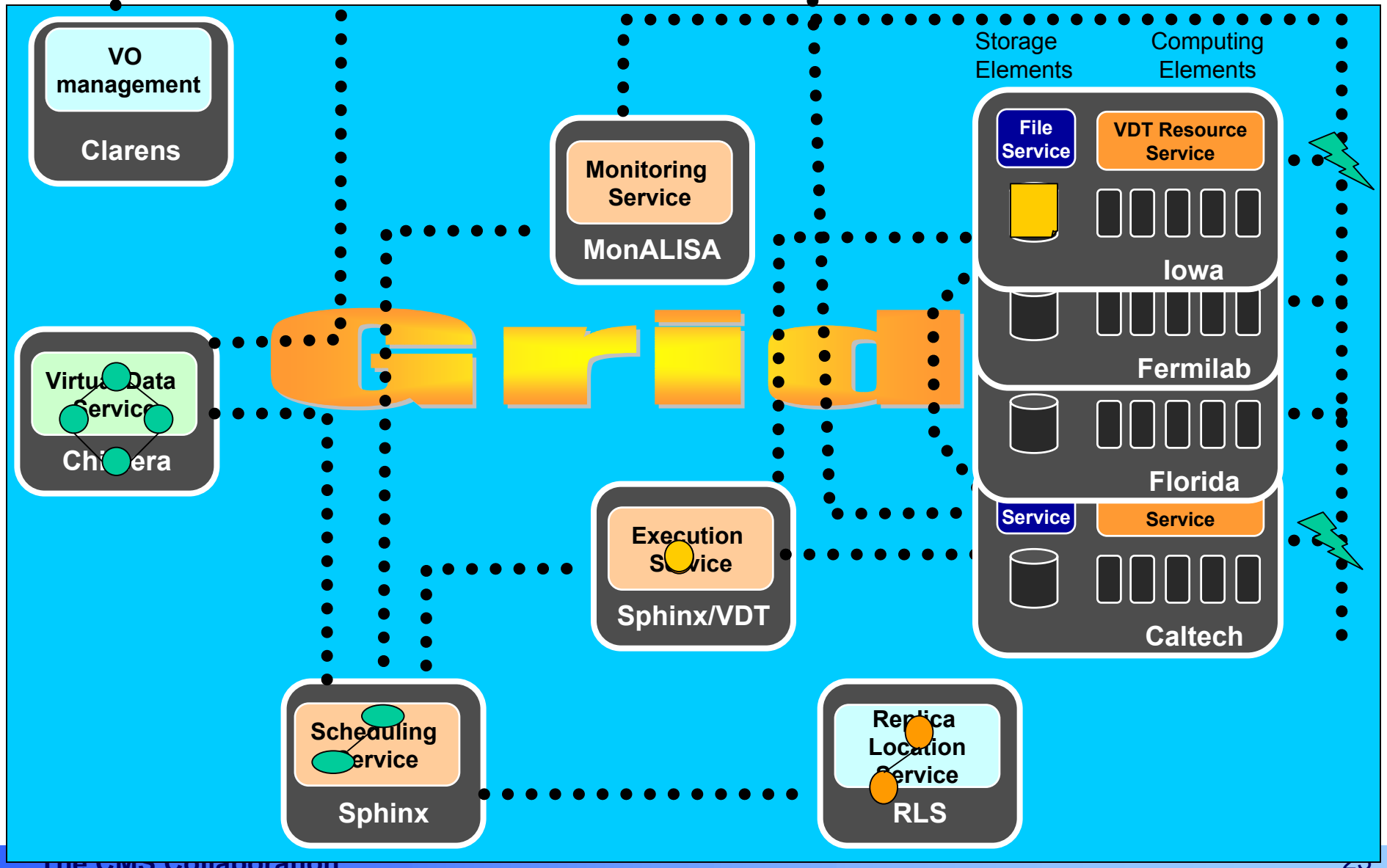
ARDA term	***GAE term***
Metadata Catalogue, Authentication, Authorization	
Job Monitor	Monitor, Steering
Grid Monitor	Monitor
Workload Management	Scheduler
Job Provenance	Virtual Data
Information	Lookup
Auditing	Monitor
File catalogue	Replica Location and Selection
???	Workflow manager/builder
???	Analysis versioning service
Package Manager	???

Monitoring the Global GAE System

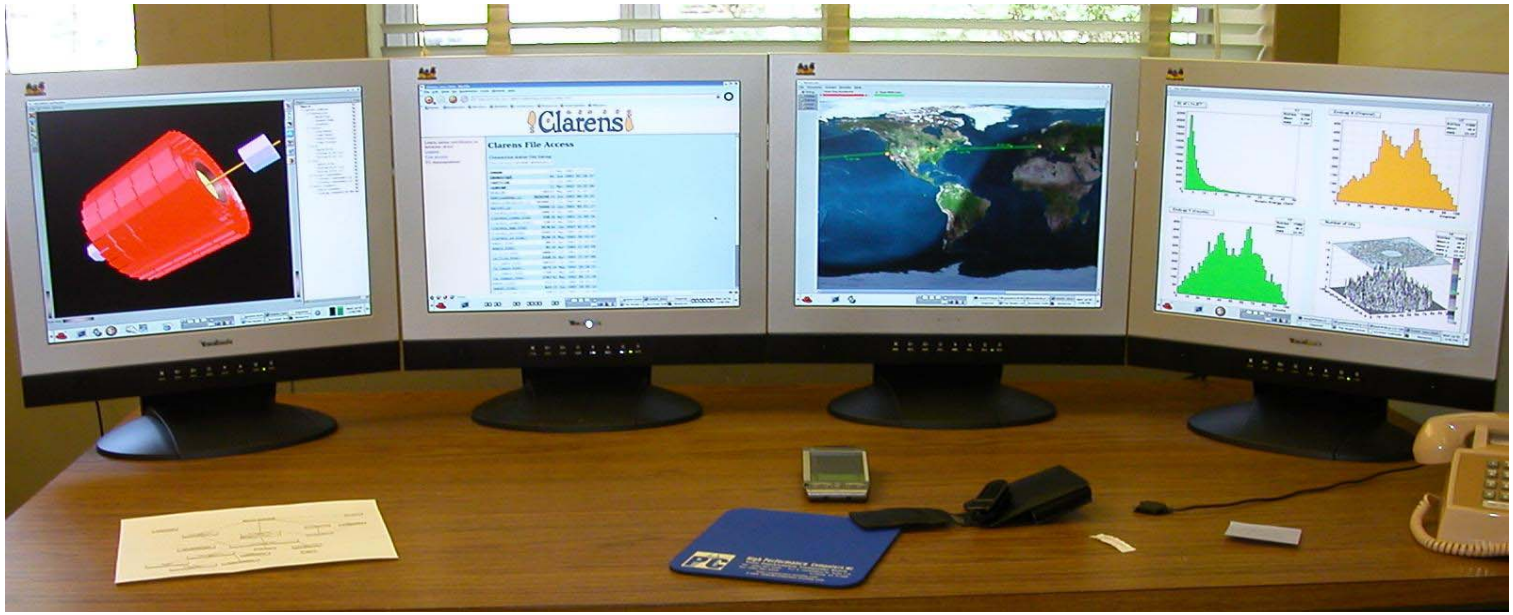
MonALISA: http://monalisa.cacr.caltech.edu



Distributed Services for Grid Analysis Environment SC03 Demo



A Candidate GAE Desktop



Four-screen Analysis Desktop
4 Flat Panels: 6400 X 1200

Driven by a single server and two graphics cards

Cost: a few thousand dollars

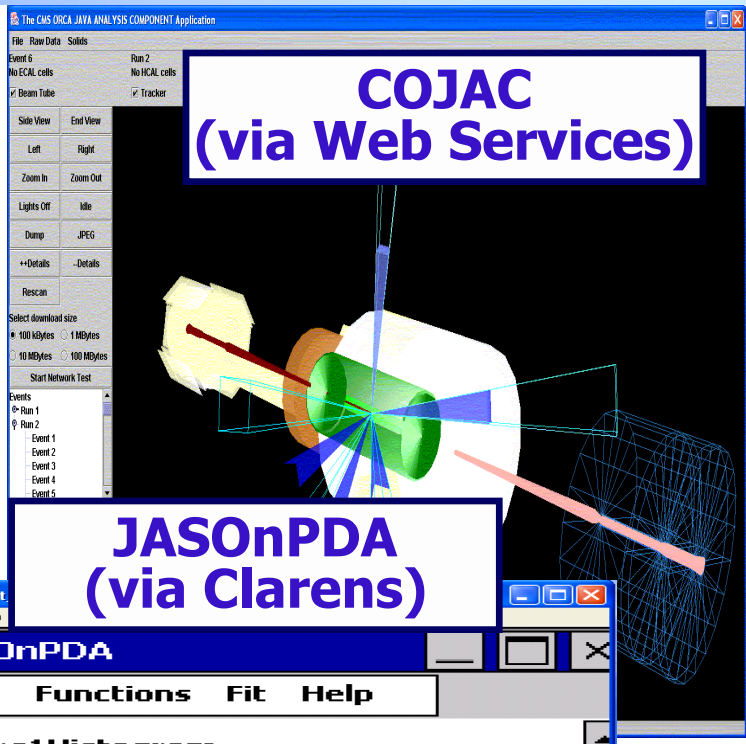
Allows simultaneous work on:

- **Traditional analysis tools (e.g. ROOT)**
- **Software development**
- **Event displays**
- **MonALISA monitoring displays;**
- **Other "Grid Views"**
- **Job-progress Views**
- **Persistent collaboration (VRVS; shared windows)**
- **Online event or detector monitoring**
- **Web browsing, email**
- ***Et cetera***

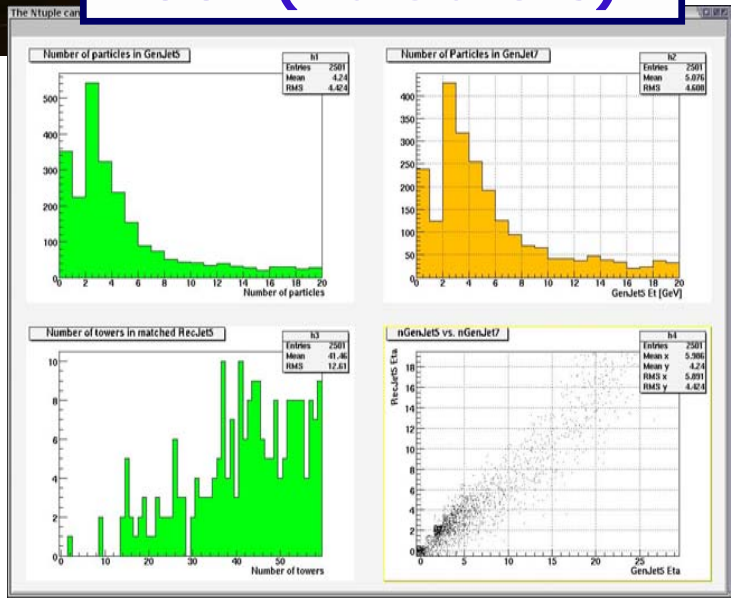
Grid-Enabled Analysis Prototypes

Collaboration Analysis Desktop

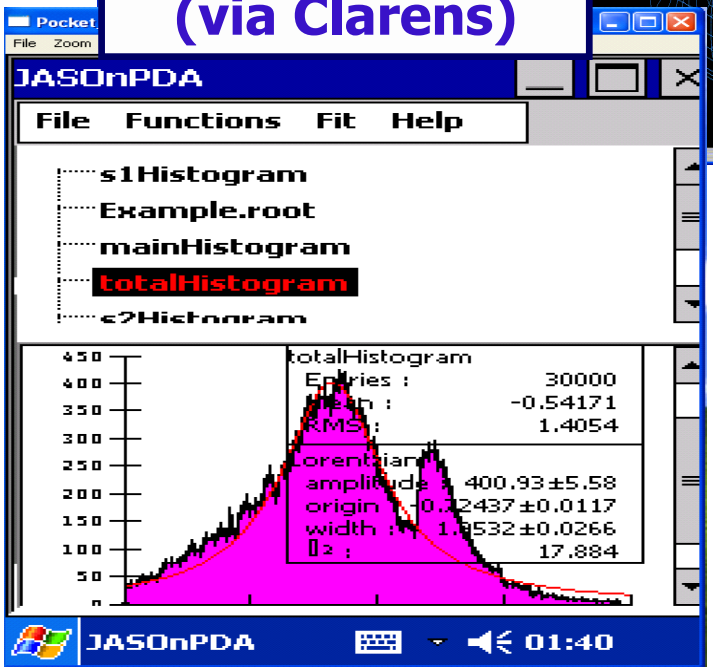
COJAC (via Web Services)



ROOT (via Clarens)



JASOnPDA (via Clarens)



Upcoming GAE Developments

- **Asynchronous service support** using alternative transport protocols
- **Higher level services** to exploit distributed nature of resources
- Define **standard APIs for similar services** provided by different existing components (such as data catalogs)
- Implement a **realistic end-to-end prototype** based on the SC2003 GAE demonstration
  - ➔Make more existing components web-service enabled

Summary

- Distributed Production
  - CMS is already using successfully LCG and VDT grid middleware to build its distributed production system
  - A distributed batch-analysis system is being developed on top of the same LCG and VDT software
  - CMS suggests that in the short term (≤6 months) ARDA extends the functionality of the LCG middleware to meet the architecture described in the ARDA document
- Grid Analysis Environment
  - CMS asks that the GAE be adopted as the basis for the ARDA work on a system that supports interactive and batch analysis. Support for Interactive Analysis is the crucial goal in this context.