The ARDA Architectural Roadmap for Distributed Analysis

L.A.T.Bauerdick, Fermilab

LHC Substantial Experience w/ Grids

LHC experiments working with Grid developers and integrators

- EDG testbeds and application integration, DataTag, national Grid projects in Europe and elsewhere
- U.S. CMS and U.S. Atlas working with PPDG, GriPhyN, iVDGL

Virtual Data Toolkit VDT and EDG components

LCG-1 Service and U.S. Grid3

LHC Distributed Production of Simulated Events

- millions of events produced, thousands of CPUs used, Terabytes of data moved and stored, every day!

LHC Data Analysis: New Challenges and New Approaches

Mandate of ARDA RTAG

"Requirements and Technical Assessment Group" started by LCG SC2

Mandate for the ARDA RTAG

- To review the current DA activities and to capture their architectures in a consistent way
- To confront these existing projects to the HEPCAL II use cases and the user's potential work environments in order to explore potential shortcomings.
- To consider the interfaces between Grid, LCG and experiment-specific services
  - Review the functionality of experiment-specific packages, state of advancement and role in the experiment.
  - Identify similar functionalities in the different packages
  - Identify functionalities and components that could be integrated in the generic GRID middleware
- To confront the current projects with critical GRID areas
- To develop a roadmap specifying wherever possible the architecture, the components and potential sources of deliverables to guide the medium term (2 year) work of the LCG and the DA planning in the experiments.

ARDA Report Delivered Oct 2003

CERN-LCG-2003-033

LHC Computing Grid Project

ARCHITECTURAL ROADMAP TOWARDS DISTRIBUTED ANALYSIS

Document identifier:	**CERN-LCG-2003-033**
Date:	**31-Oct-2003**
Authors:	**ALICE:** **P. Buncic (CERN/IKF Frankfurt)** **F. Rademakers (CERN)** **ATLAS:** **R. Jones (U.Lancaster)** **R. Gardner (U.Chicago)** **CMS:** **L.A.T. Bauerdick (Fermilab), chair** **L. Silvestris (INFN Bari)** **LHCb:** **P. Charpentier (CERN)** **A. Tsaregorodtsev(IN2P3 Marseille)** **LCG:** **D. Foster (CERN)** + M.Lamanna (CERN) **T. Wenaus (BNL)** **GAG:** **F. Carminati (CERN)**
Document status:	**v1.01**

Abstract:

General ARDA Roadmap

Emerging picture of "waypoints" on the ARDA "roadmap"

- ARDA RTAG report
  - review of existing projects, common architectures, component decomposition & re-factoring
  - recommendations for prototypical architecture of ARDA services, and definition of prototypical functionality and a development strategy
- Development of a prototype for the ARDA services and initial release
  - Integration with and deployment on LHC Grid resources and services
- Re-engineering of prototypical ARDA services, as required

GGF OGSI gives a framework in which to implement / run ARDA services

- Addresses the "services architecture"
- Provides a framework for advanced interactions with the Grid

OGSI is not a proven technology

- Need to address issues of performance and scalability up-front
- Importance of modeling, plan for scaling up, engineering of underlying services infrastructure

Roadmap to a GS Architecture for the LHC

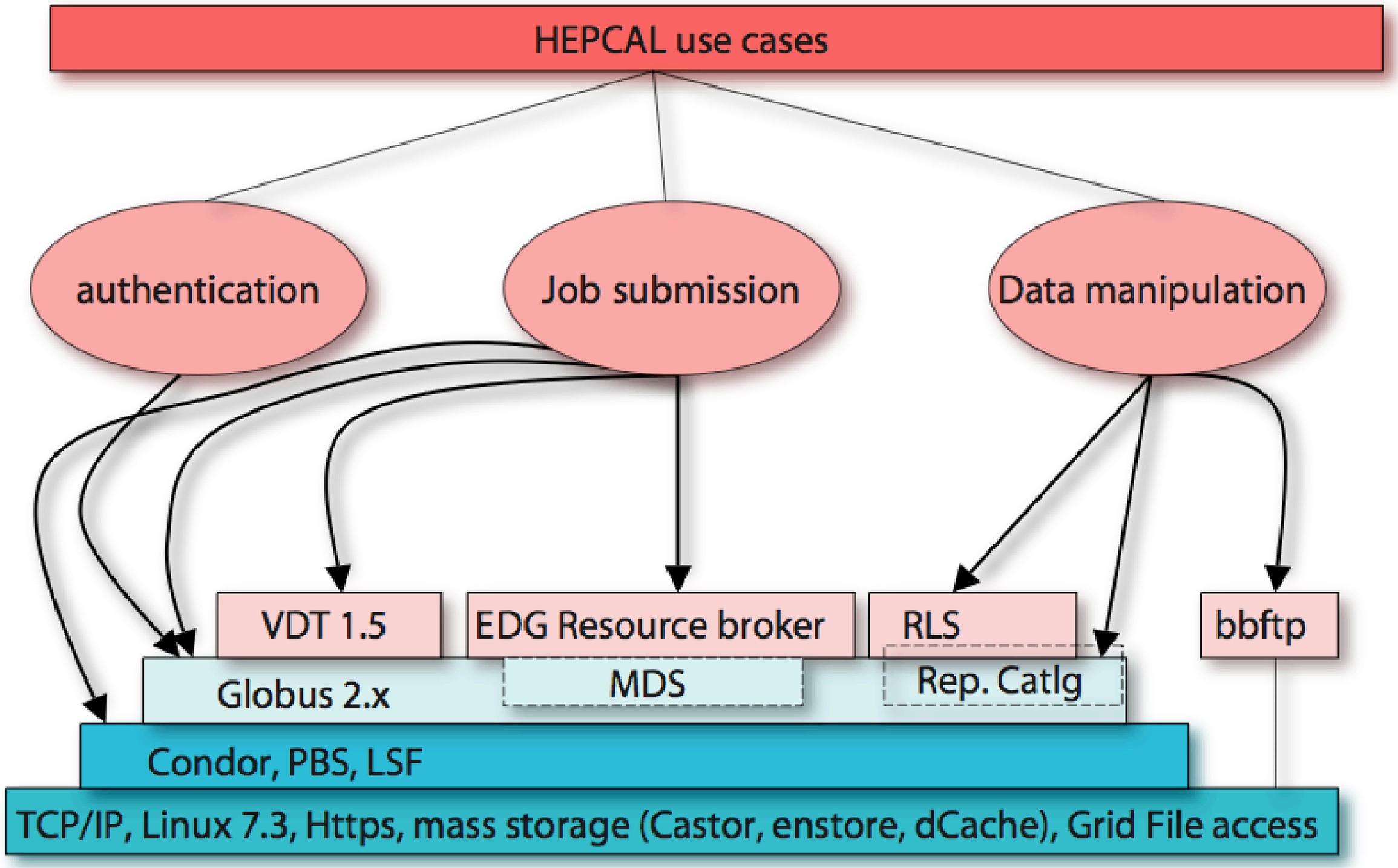
Transition to Grid Services explicitly addressed in several existing projects

- CMS Caltech's GAE and MonaLisa
  - Based on web services for communication, Jini-based agent architecture
- LHCb's Dirac
  - Based on "intelligent agents" working within batch environments
- Alice's AliEn
  - Based on web services and communication to distributed database backend
- Atlas' DIAL
  - OGSA interfaces
- LCG-GTA initial work on OGSA
  - GT3 prototyping

Leverage from experience gained in Grid & M/W R&D projects

LCG Software Stack

LHC distributed environment for Event Simulation Production



Layers and Middleware

Layers to organize Architecture and Responsibility/Charge

Experiment Layer
Frameworks, Grid shells, Portals, ...
Experiments, Physics Grid Projects, ...

Application Middleware
Pool, Object Server, Higher Level Services, ...
LCG AA, Experiments, GAE, Grid Projects, ...

Grid Middleware
Workload Systems, Replica Managers, ...
VDT, LCG/EGEE, Grid Projects, GGF, ...

Facilities and Fabrics
Worker Nodes, Storage Managers, ...
LCG, Regional Centers, Grid Projects, ...

Physics "Services"

Experiment's "Services" go end-to-end!

Experiment Layer
Frameworks, Grid shells, Portals, ...
Experiments, Physics Grid Projects, ...

Application Middleware
Pool, Object Servers, Higher Level Services,
LCG AA, Experiments, GAE, Grid Projects,

Grid Middleware
Workload Systems, Replica Managers, ...
VDT, LCG/EGEE, Grid Projects, GGF, ...

Facilities and Fabrics
Worker Nodes, Storage Managers, ...
LCG, Regional Centers, Grid Projects, ...

Physics "Data Services"

CMS Physics Finder

POOL Collection Manager, Object Server, ...

Replica System, Data Access Optimizer, ...

Storage Managers, Disk Caches, ...

Physics "Tasks Manager"

Atlas Physics Portal

MC Executer, Physics Prioritizer, ...

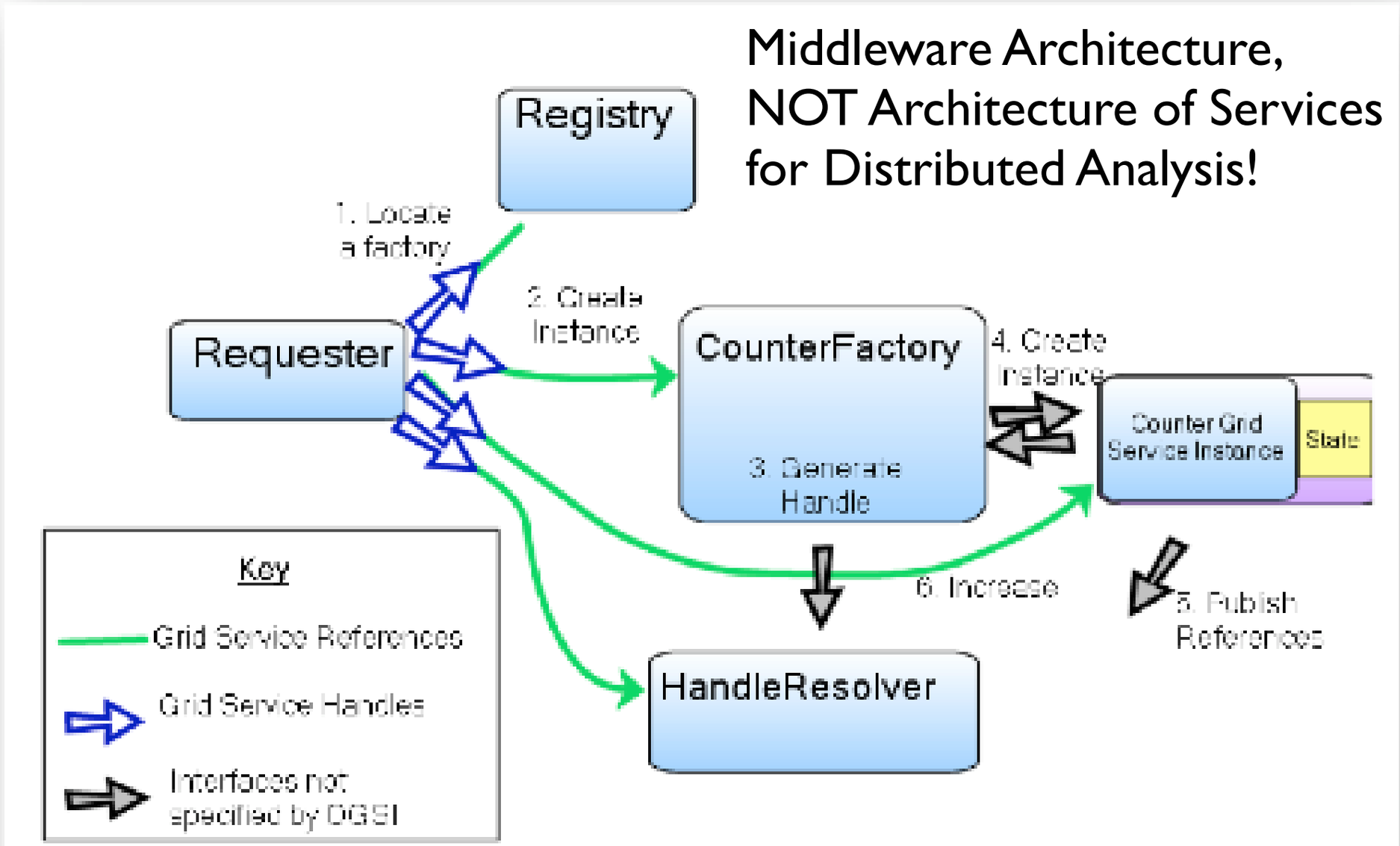
Grid Executer, Workload Optimizer, ...

Worker Node Managers, Queues, ...

Grid Services

Grid Services Architecture vs. Architecture of the Services for Analysis:

- ARDA is NOT about how the middleware services function!



Middleware Architecture, NOT Architecture of Services for Distributed Analysis!

Example of a Grid Services Architecture

Service Architecture vs Architecture of Services

ARDA did not go into "middleware architecture",
except assuming web services based implementation or "OGSI"

- OGSI "envelope of services", enabling loosely coupled services
  - "middleware developers" provide essential architectural components
    - service management and infrastructure functions
    - robust protocols, workload balancing, lifetime support, discovery mechanisms, service registration, service factories, notifications, diagnosis functions, agents and higher level services, etc
- an essential and wide field of work for LCG/EGEE, VDT and others: provide a solid foundation upon which to build end-to-end systems
- quality of which defines what (physics) functionalities are feasible or not...!

ARDA roadmap proposes approach to define services and interfaces:

- define the semantics of the physics services needed to implement use cases, that is, messages one uses to interact with, behavior expected in response, state of services and associated resources

Hepcal-II Analysis Requirements

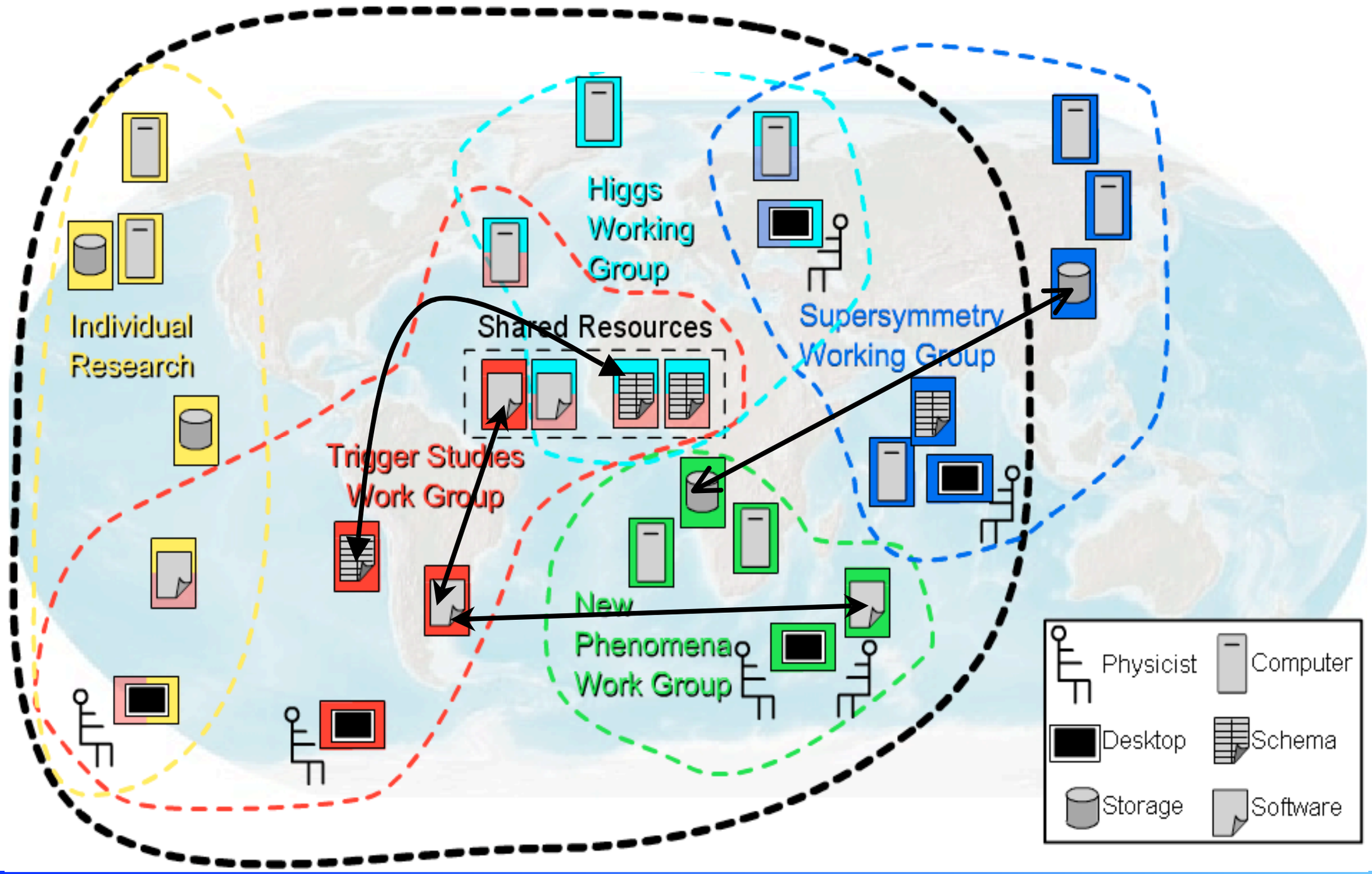
ARDA use cases based on (prelim. version of) GAG HEPCAL-II report

- Setup: authenticate to system, which creates the analysis context ensure the correct installation of software packages
- Determine data sets and eventually event components
  - Input data are selected via a query to a metadata catalogue
- Perform iterative analysis activity
  - pass selection and algorithm together with spec of the execution environment to a workload management system,
  - Algorithms are executed on one or many nodes
  - User monitors progress of job execution
  - Results are gathered together and passed back to the job owner
  - Resulting datasets can be published to be accessible to other users

Distributed Analysis Scenarios

Communities of Scientists Using the Grid for Distributed Analysis

Infrastructure for sharing, consistency of physics and calibration data, software



Distributed Analysis Scenarios

- Persistency of the analysis workspace that a user can later re-connect to, re-submit the analysis with modified parameters or code, check the status, merge results between analyses, share datasets with other users and analysis workspaces, while the system keeps provenance information about jobs and datasets.
- To carry out the analysis tasks users are accessing shared computing resources. To do so, they must be registered with their Virtual Organization (VO), authenticated and their actions must be authorized according to their roles within the VO
- The user specifies the necessary execution environment (software packages, databases, system requirements, etc) and the system insures it on the execution node. In particular, the necessary environment can be installed according to the needs of a particular job
- The execution of the user job may trigger transfers of various datasets between a user interface computer, execution nodes and storage elements. These transfers are transparent for the user

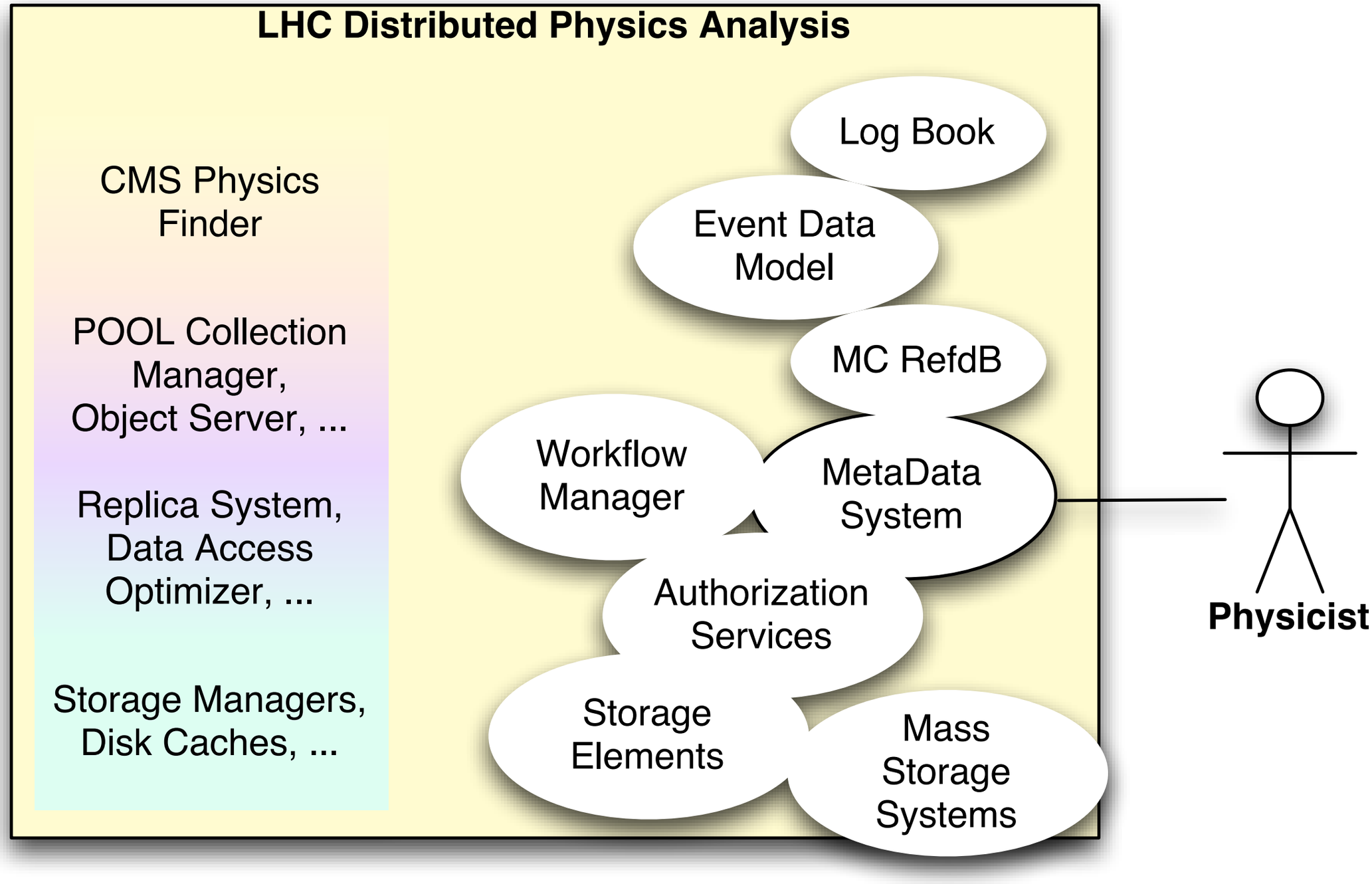
Distributed Analysis Scenarios

- System extracts a subset of the datasets from the virtual file catalogue using metadata conditions provided by the user.
- System splits the tasks according to the location of data sets.
  - balancing between local data access and data replication.
- Spawn sub-jobs and submit to Workload Management with precise job descriptions
  - User can control the results while and after data are processed
- Collect and Merge available results from all terminated sub-jobs on request
- Analysis objects associated with the analysis task remains persistent in the Grid environment so the user can go offline and reload an analysis task at a later date, check the status, merge current results or resubmit the same task with modified analysis code.

Also looked at Synchronous Analysis Scenario:
using PROOF in a Grid environment

Scenarios Guide ARDA Decomposition

Approach:
decomposition into services implementing required functionalities



ARDA Roadmap "Informed" By DA Implementations

The ARDA RTAG reviewed existing projects

- aiming to capture their architecture in a consistent way, confronting them with these use cases where possible.
- Several projects address DA by providing a set of web services or intelligent agents that communicate through a well-defined protocol. ARDA adopted that approach and studied a decomposition of the distributed analysis system into a set of services that would implement the major use cases and provide the basic required functionality for distributed physics analysis.

AliEn had most complete distributed analysis functionality, implemented through a set of web services. It addressed a large part of the DA use cases and had a substantial history of successful use.

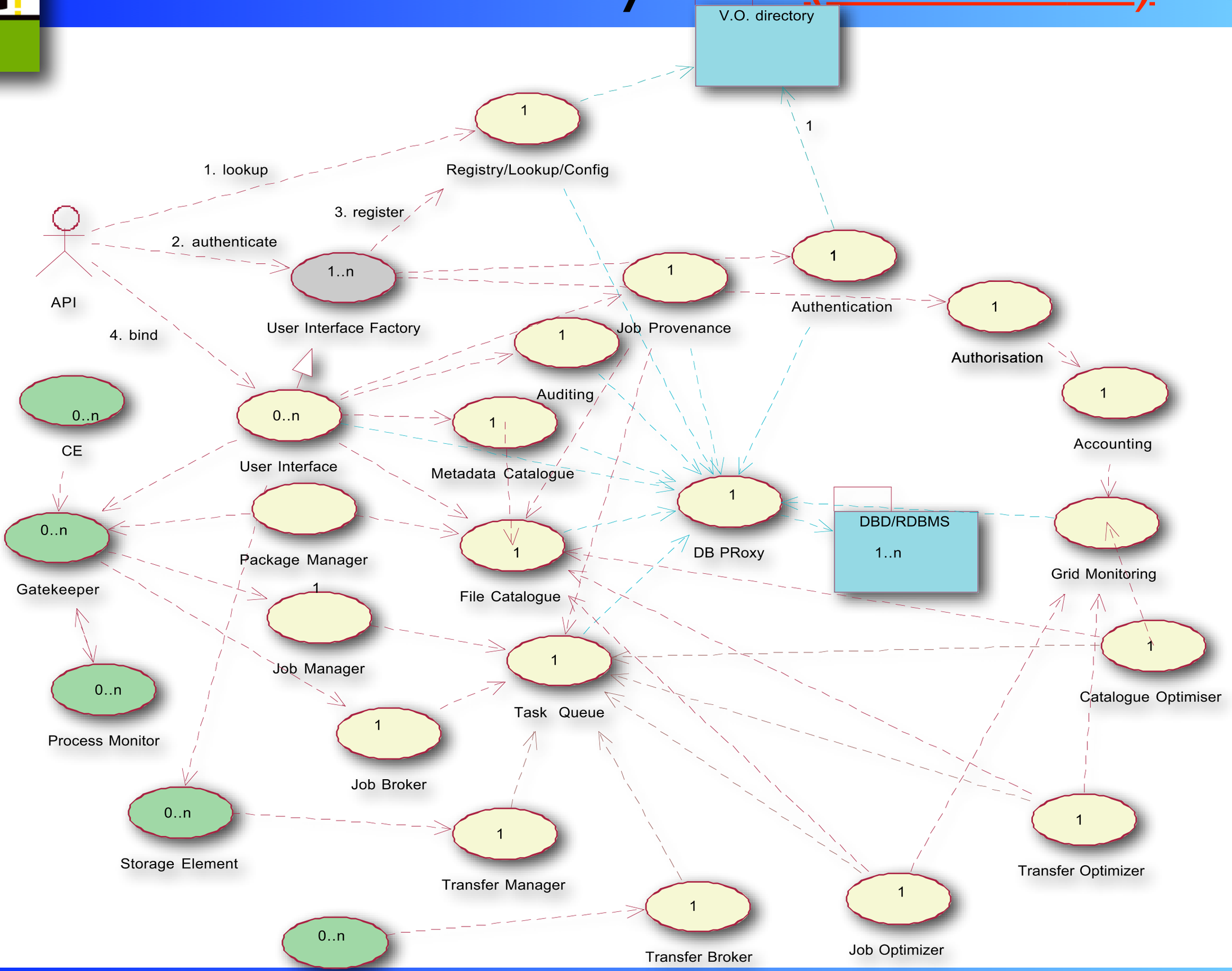
Example: Asynchronous Analysis

Running Grid-based analysis from inside ROOT (adapted from AliEn example)

- ROOT calling the ARDA API from the command prompt

```
// connect + authenticate to the GRID Service arda as “latb”
TGrid *arda = TGrid::Connect(”arda","”latb”,"","");

// create a new analysis Object ( <unique ID>, <title>, #subjobs)
TArdaAnalysis* analysis = new TArdaAnalysis(“pass001","“MyAnalysis",10);

// set the program, which executes the Analysis Macro/Script
analysis->Exec("ArdaRoot.sh”,"file:/home/latb/test.C");  // script to execute

// setup the event metadata query
analysis->Query("2003-09/V6.08.Rev.04/00110/%gjetmet.root?pt>0.2");

// specify job splitting and run
analysis->OutputFileAutoMerge(true); // merge all produced .root files
analysis->Split(); // split the task in subjobs
analysis->Run(); // submit all subjobs to the ARDA queue

// asynchronously, at any time get the (partial or complete) results
analysis->GetResults(); // download partial/final results and merge them
analysis->Info(); // display job information
```

AliEn End-to-end System (re-factored)



ARDA Decomposition into Services

ARDA RTAG was to derive a decomposition of the distributed analysis environment into a web services description, and to define their functions and interfaces. For this the RTAG started from the AliEn web services, re-factoring and synthesizing with the input gained from the other projects examined. The proposed domain decomposition into the ARDA set of services is described in the document.

The RTAG then went back to the other Grid projects related to distributed analysis, to study how these fit into the ARDA decomposition and what other and possibly complementary services are provided addressing distributed analysis functionality.

As a result, the RTAG defines the ARDA architecture as a set of services, with specific behavior and interfaces. Proceeding this way allowed the RTAG to base its recommendations on an analysis and synthesis of several projects, with at least one existing implementation. The resulting ARDA blueprint is presented in the document.

ARDA Distributed Analysis Services

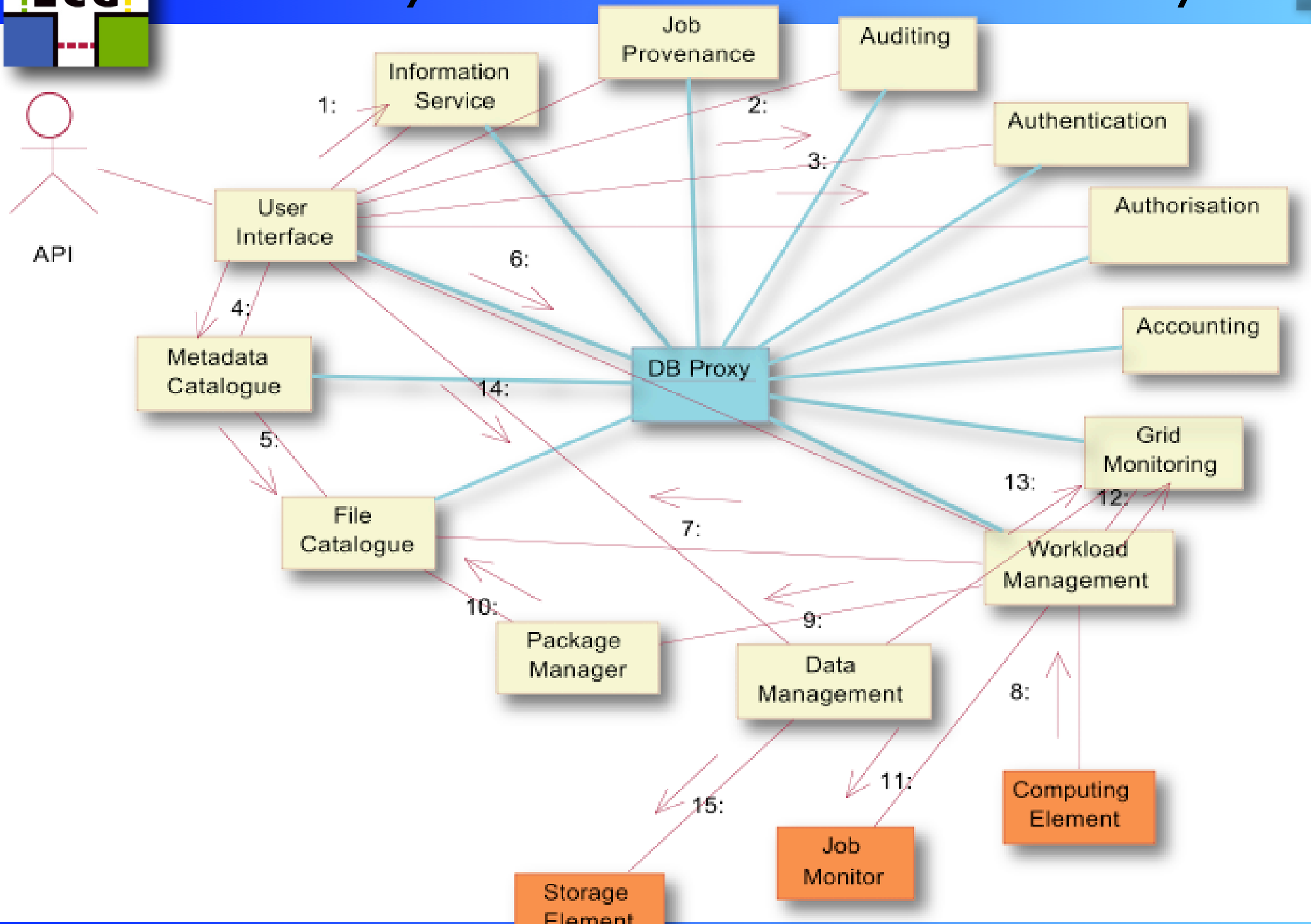
Grid Services for Distributed Analysis

- ARDA Services should be OGSI compliant -- built upon OGSI middleware
- ARDA API for experiment frameworks and applications
  with bindings to C++, Java, Python, PERL, …
  - interface through Grid Access Service Factory/API
    -- authentication, persistent "session"
- Fabric Interface to resources through CE, SE services
  - job description language, based on Condor ClassAds and matchmaking
  - statefulness and persistence through database(ses) / database proxy

We arrived at a decomposition into the following key services

- API and User Interface
- Authentication, Authorization, Accounting and Auditing services
- Workload Management and Data Management services
- File and (event) Metadata Catalog services
- Information service
- Grid and Job Monitoring services
- Storage Element and Computing Element services
- Package Manager and Job Provenance services

ARDA Key Services for Distributed Analysis



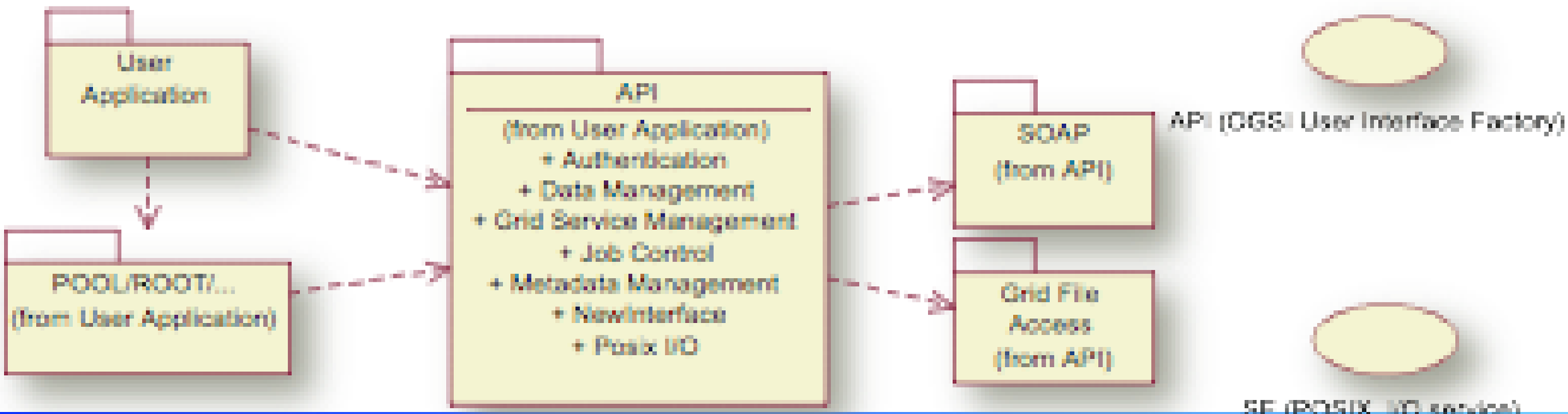
API to Grid services

ARDA "Grid Access Service Factory"/API

- Interface services to higher level software
  - Experiment frameworks
  - Analysis shells, e.g. ROOT, JAS et al.
  - Grid portals and other forms of user interactions with environment
  - Advanced services e.g. virtual data, analysis logbooks etc
- interface to application layer related services, e.g. exp. data and metadata management systems

Provides an API that others can project against

- Process to get a common API between experiments ⇒ goal of the prototype
- The GAS/API can use the Condor ClassAds and extensions as a Job Description Language
  - compatibility with existing job execution services, in particular LCG-n



Architectural "Talking Points"

Horizontally structured system of services with well-defined GAS/API and a database backend

- Can be extended with additional services, new implementations can be moved in, alternative approaches tested and commissioned — modular, flexible, open design

Interface to LCG-1 infrastructure

- VDT/EDG interface through CE, SE and the use of (A)JDL, compatible with existing i/s
- ARDA VO services can build on emerging VO management infrastructure

ARDA initially looked at file based datasets, not object collection

- talk with POOL how to extend the file concept to a more generic collection concept
- investigate experiment's metadata/file catalog interaction

VO system and site security

- Jobs executed on behalf of VO, but users fully traceable - address security model upfront!

How do policies get implemented, e.g. analysis priorities, MoUed contributions etc

- Auditing and accounting system, priorities through special "optimizers"
- accounting of site "contributions", that depend what resources sites "expose"

Database back-end

- Addresses latency, stability and scalability issues; "people know how to do large databases" well founded principle (see e.g. SAM for RunII), with many possible migration paths
- In a sense, the system **is** the database (possibly federated and distributed) that contains all there is to know about all jobs, files, metadata, algorithms of all users within a VO
- set of OGSI grid services provide "views" into the database, API provides the user access
- enables structuring into federated grids and "dynamic workspaces"

ARDA Roadmap for Prototype

Recommended development and deployment of initial prototype of ARDA architecture

- end-to-end: did not see an easy "evolutionary" path from existing GT2-based grids

Main Goal: provide a more complete blueprint and develop the specifications for functionality and interfaces of the ARDA services and API

- build functional prototype based on refactoring existing implementations
- provide main physics functionality to guide work on interfacing frameworks and analysis shells to the distributed computing environment
- provide interaction points with community: enable broad contributions
- gain experience with web-services based architectures and understand issues related to security infrastructure, VO management, information management, interfacing to fabrics, etc

Grid service instances and descriptions should be defined early in the process

- forum of developers and expert users to agree on service specifications, implementation priorities, integration plans, user feedback
- descriptions and specifications of ARDA services, interfaces, methods, definitions of external interfaces, i.e. the API and interfaces to infrastructures, facilities, fabrics
- "pro-active editor" (as proposed by Frédéric)

Investigate how end-to-end ARDA prototype would "run" in LHC Grid environment

The ARDA Project

RTAG report proposed a four-prong approach towards these goals:

- 1) Re-factoring of AliEn and other services into ARDA, with an initial release; consolidation of the API working with the experiments and the LCG-AA; release of a fully functional prototype. Subsequently implementation of agreed interfaces, testing and release of the prototype implementation.
- 2) Modeling of an OGSI-based services infrastructure, performance tests and quality assurance of the prototype implementation
- 3) Interfacing to LCG-AA software like POOL and ROOT
- 4) Interfacing to experiment's frameworks, with specific meta-data handlers and experiment specific services

The LCG Application Area, the LCG Grid Technology Area, and the EGEE middleware team should be involved in the ARDA project.

ARDA Prototyping

ARDA prototype would define the semantics of the initial set of services and their interfaces. Timescale: ~6 months

Important to involve experiments, LCG, other projects at the right level

- Continue initial modeling of GT3-based services
- Interface to major cross-exp packages: POOL, ROOT, PROOF, others
- Program experiment frameworks against ARDA API, integrate with experiment environments
- Expose services and UI/API to other LHC projects to allow synergies
- Spend required effort to document, package, release, deploy

Once ARDA prototype is delivered and evaluated, implement system

- Scale up, re-engineer as needed: OGSI, databases, information services
- Deploy and interface to site and grid operations, VO management etc
- Build higher-level services and experiment specific functionality
- Work on interactive analysis interfaces and new functionalities

Evolving ARDA to LHC scale

ARDA roadmap based on a well-factored prototype implementation that allows evolutionary development into a complete system that evolves to the full LHC scale

- ARDA initial prototype would be pretty lightweight
  - HEP-specific services, however based on generic OGSI-compliant services
- rapid delivery of prototype followed by short evolution cycles

Expect LCG/EGEE middleware effort to play major role to evolve the foundations of ARDA services, of the concepts and the implementation

- re-casting the (HEP-specific event-data analysis oriented) services into more general services, from which the ARDA services would be derived
- addressing major issues like a solid OGSI foundation, robustness, resilience, fault recovery, operation and debugging
- well-aligned further developments and inclusion of advanced functionalities

Expect US middleware and Grid projects to be involved in this!

ARDA "Architecture of Collaboration"

Major Challenge of ARDA project (and LCG in general)

- how does ARDA collaborate and get contributions from "outside"?
- managing the feedback loop
- other projects and groups, non-LHC experiments, etc

Foremost the LHC experiments themselves --- given the pressures

- "We recommend the experiments be involved from the beginning by working on the interface with the frameworks and integrating into the experiment's data and metadata management environments."

Keep others engaged without being overwhelmed

- "Other LHC-related projects should be engaged by exposing services and GAS/API definitions early to allow synergistic developments from these projects."

Foremost important that other know what is going on in ARDA

- "It is important that the appropriate emphasis and effort is being put on documentation, packaging, releases, deployment and support issues."