AMI & ARDA

Solveig ALBRAND

LPSC Grenoble

http://atlasbkk1.in2p3.fr:8180/AMI

AMI History

- The AMI project started in the spring of 2000. The first application was an electronic notebook for Atlas LAr testbeam runs.
- Soon, we had other requests for other database projects:

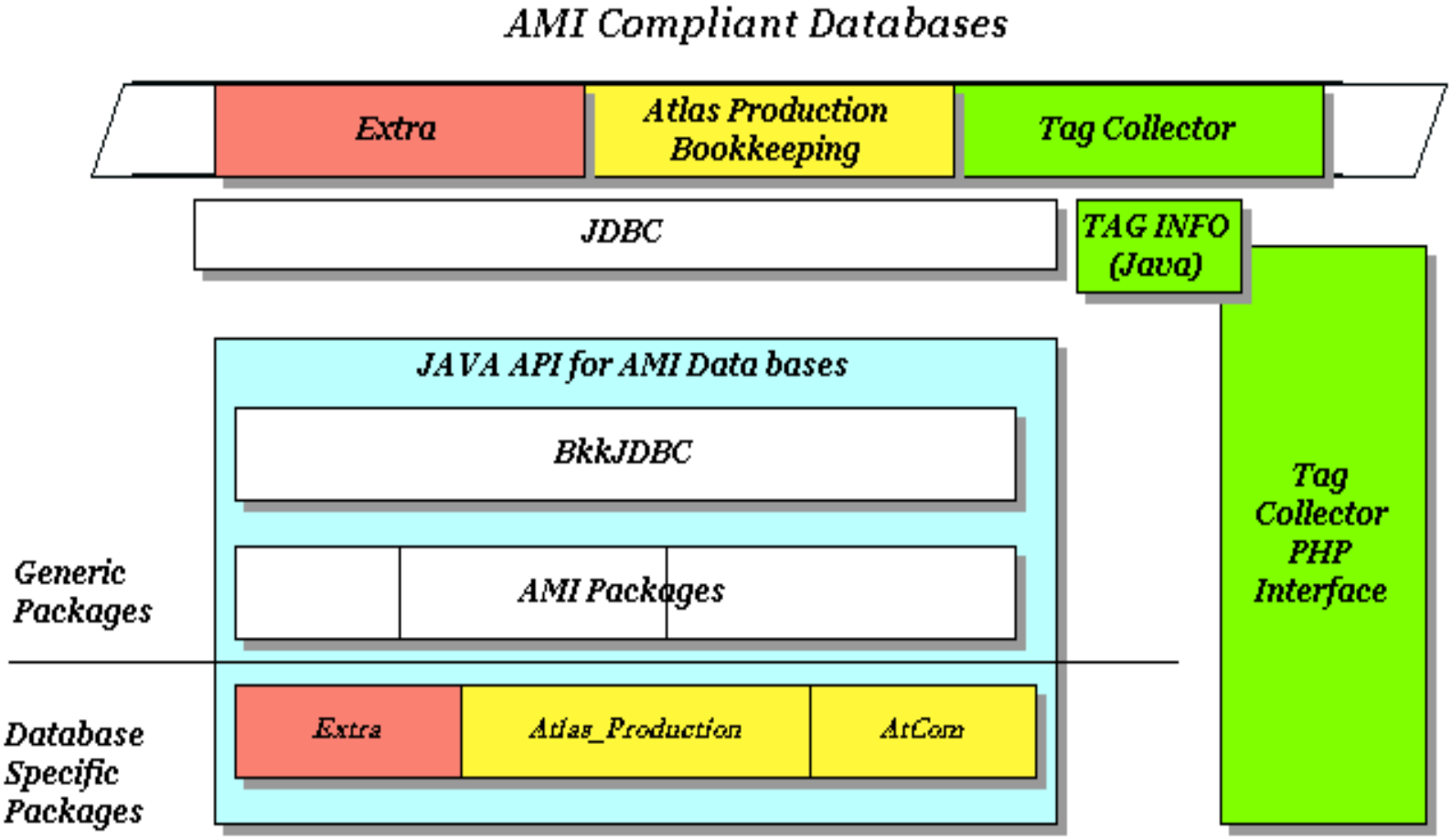
  ➔ Needed to reuse software, and to make databases "self describing" so they can be used in a generic way.

Some Design Principles

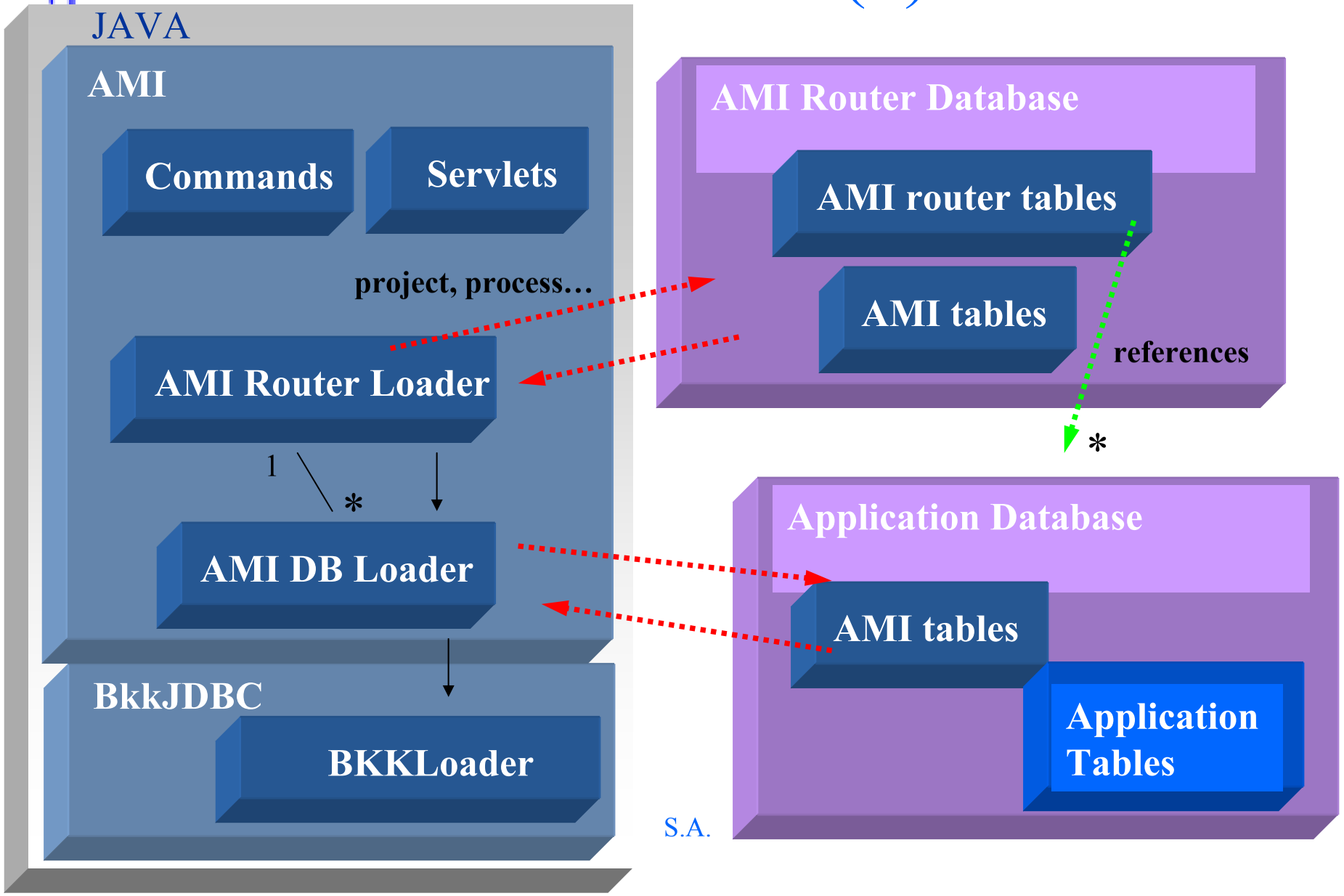
- Use an RDBMS.
- Use many RDBMS.
- Be generic. (Reuse as much SW as possible)
- Distribute geographically
- Manage long term projects (schema evolution)
- Place emphasis on user interfaces.

➔ *SQL and Java were chosen*

AMI Architecture



AMI Architecture (2)



Notes on the Schema

- Distinguish between the database schema and the application schema.
- An application has at least one project, corresponding to a version of the application schema.
- Data can be arranged by “project” but a project can contain several “processes”. Threaded queries can be launched over of all processes of a project.

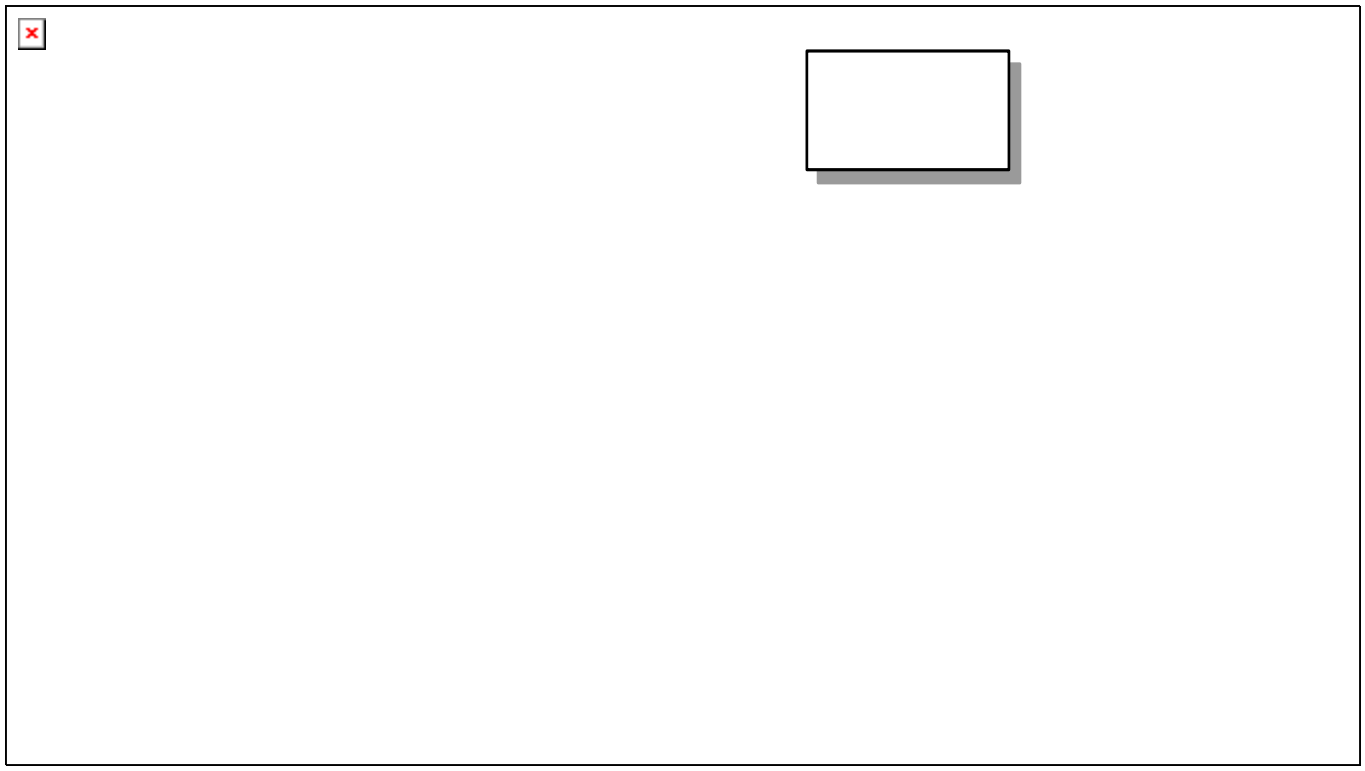
Client queries in terms of the application semantics

- Real database implementation “hidden”
- A user of the AMI production database should be able to work with a schema of datasets, files and events, whereas a tag collector user should be able to think in terms of releases and packages. The same client software could be used by both sets of users.

Schema Evolution

- A database supports schema versioning if different versions of its schema can be identified and selected by a suitable labeling system, such as symbolic naming, or time stamping. Schema versioning can be partial – in which case updating data is only allowed on one specific schema – usually the current one, and the old data recorded with a previous schema can be considered "read-only". Full schema versioning supports viewing and update of all data, through a user definable versioned interface.
- See “Managing Scheme Evolution in AMI” S.A. Dec 2003.

Implicit Schema Versioning

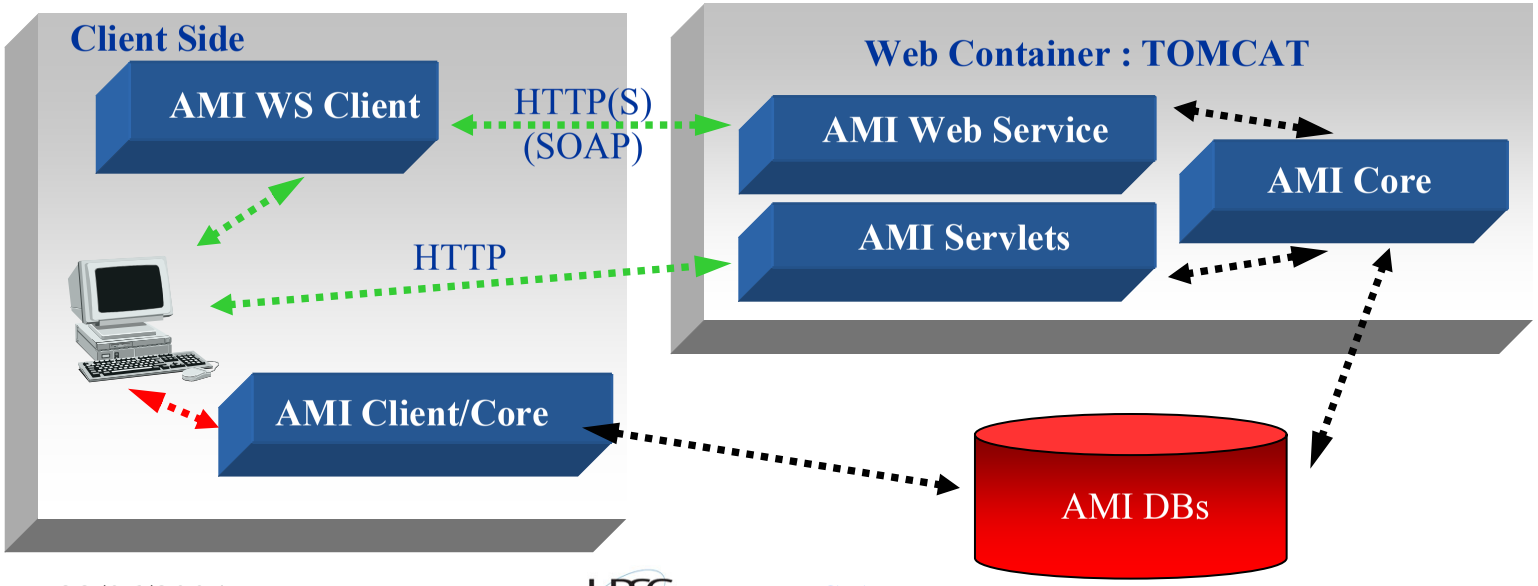


Web Services (1)

We decided to introduce a web service client in mid 2003.
The first version was released in December 2003.
Uses JWSDP technology with AXIS servlets.

The advantages are very obvious to us.

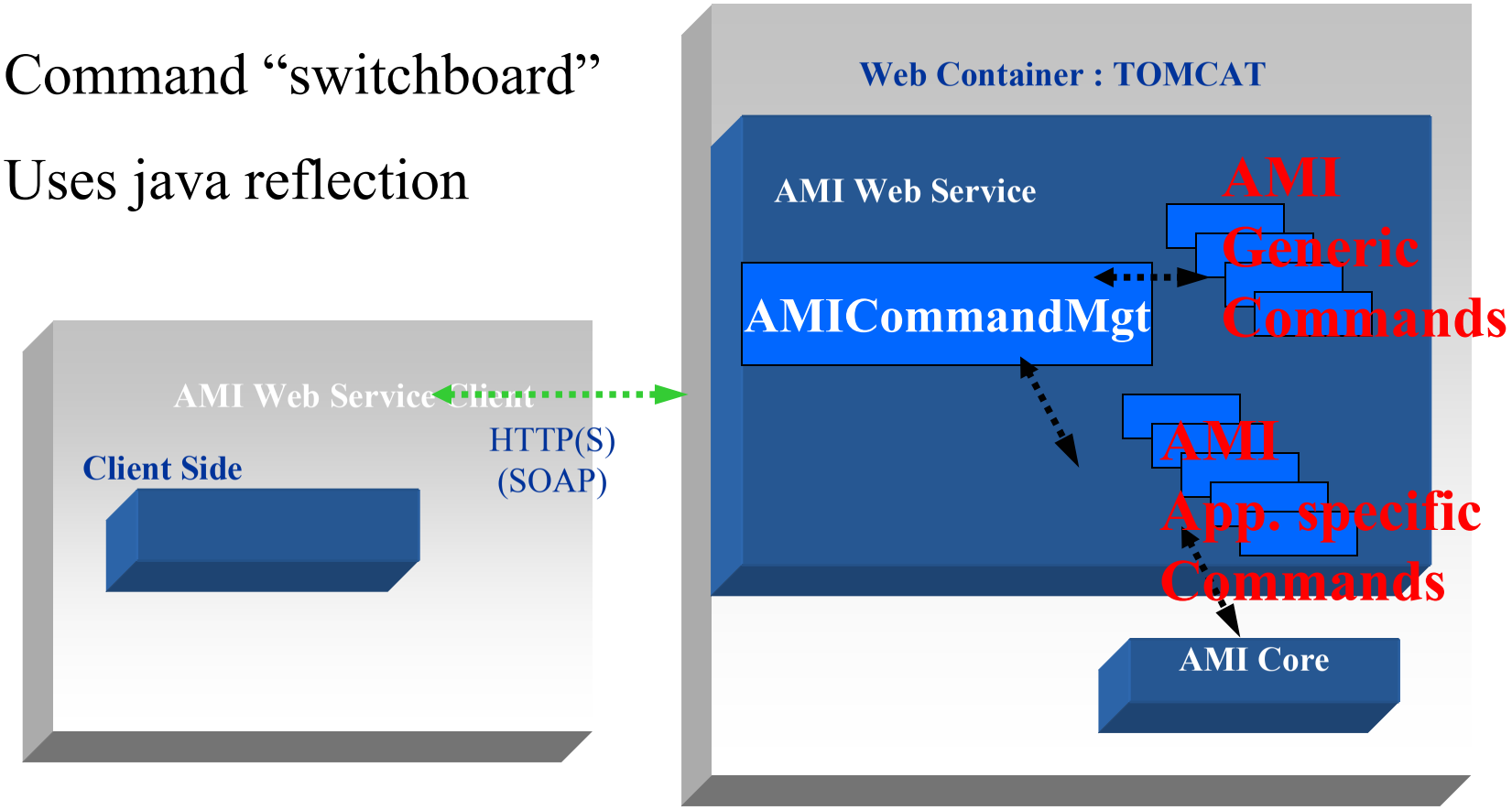
Maintainance, distribution, multi-platform, multi language support



A simple web service

Command “switchboard”

Uses java reflection



First experiences with Web Services (Client generation)

- The advantages are less obvious for clients than for developers.
- In theory users can implement their own clients using the WSDL file. In practice it is not so simple. There are teething problems, we have to provide client support. (web pages + example clients)
- C++ is a particular problem in our context.
  - Our C++ client with “Easy Soap” was easy, but doesn’t compile with gcc3.2.
  - Need LCG decision + support (gSoap ?).

First experiences with Web Services (Performance I)

- First users of the web service reported a significant performance drop.
- Although performance is not our primary concern, the problem required some immediate action. We have attained some improvement by code optimization, and configuration tuning.
- Perhaps SOAP/XML itself is the problem?
- And what are the requirements for performance?

Web Services (Performance II)

- SOAP/XML is a “heavy” protocol. The client needs to parse. This is time consuming. But everyone can read XML.
- Should we step back to “RPC”? Will this be as generic?
- Should we adopt something new? OGSA-DAI/WSRF/DSDL/ gLite“custom encoded”/…
- We do not currently have the manpower to try out new solutions – or even to follow them closely.
- We must stick to solutions which work, and concentrate on the functionality of our applications.
- Slow functions which work are less ridiculous than a fast application with no functions!

Web Services (Performance III)

- We need guidance:
  - on choice of technology,
  - on requirements.
- Our first step is to establish AMI metrics.
- PERFORMANCE is a multi dimensioned problem.
  - Technology (SOAP/ !SOAP) (and which SOAP package?)
  - Deployment (Load balancing…)
  - Architecture
  - DB design
  - Code efficiency
  - Network
  - Server and client configuration

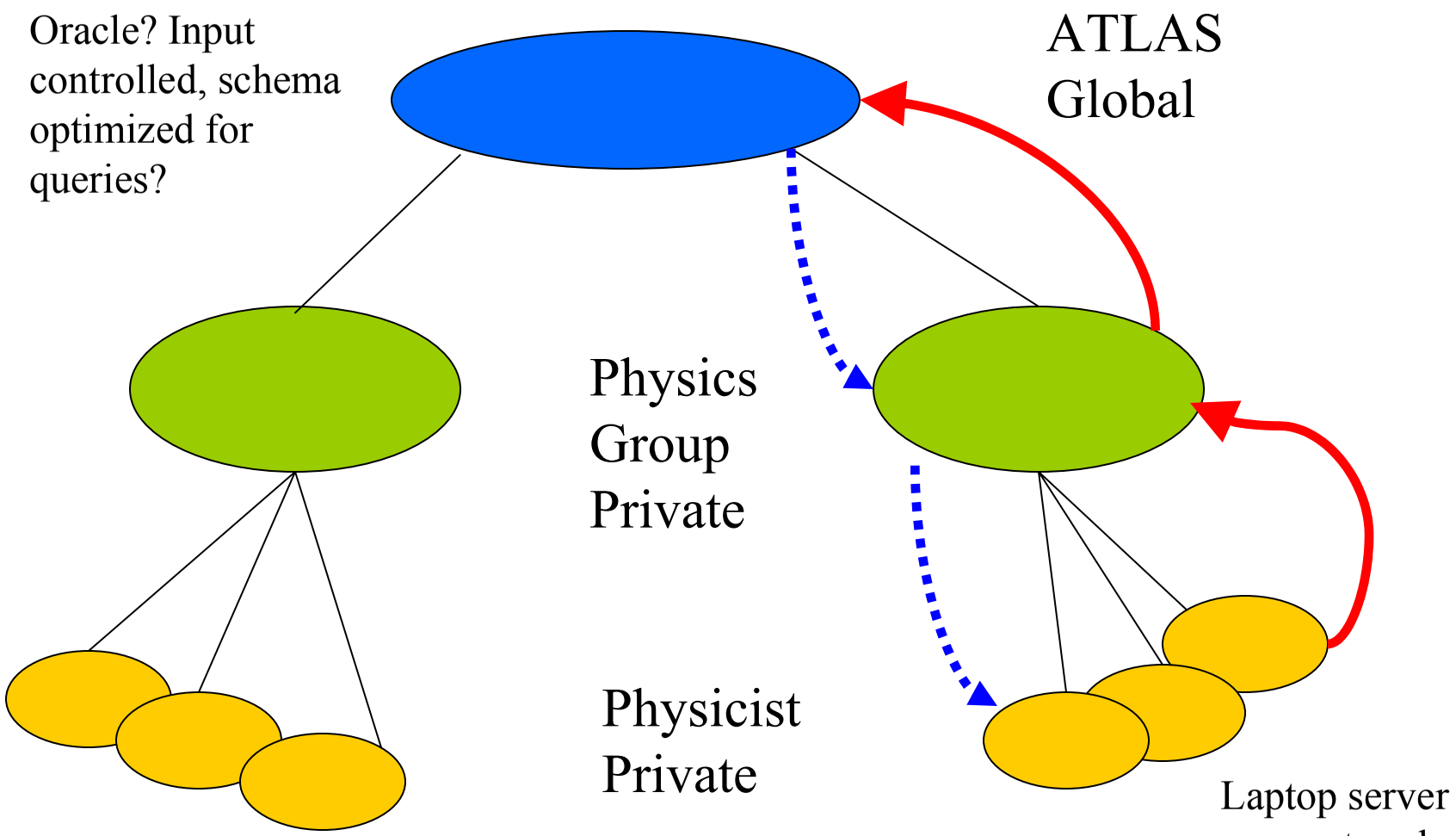
Other Web service problems

- How to we cope with very large query responses?
  - Paging ➔ active cursor
  - Sessions, Web service proxy
- Security/Authentification/Authorization
  - We installed EDG-WP2 Spitfire on a developement server, but the clients were not ready for this type of access.
  - Help from EGEE for “son of Spitfire” – work just beginning.

Other Concerns

- Prove multi RDBMS of AMI. (especially ORACLE)
- “Horizontal”Distribution :How to replicate? How to deploy servers? (Load balancing)
- “Vertical Distribution”: Who needs/does what? Same schema? (see next slide)
- User management for the application level.
- AMI Monitoring ➔ system for running tasks regularly or at a set time.
  - Who does what?
  - When did they do it?
  - Did anything go wrong?

Vertical Distribution

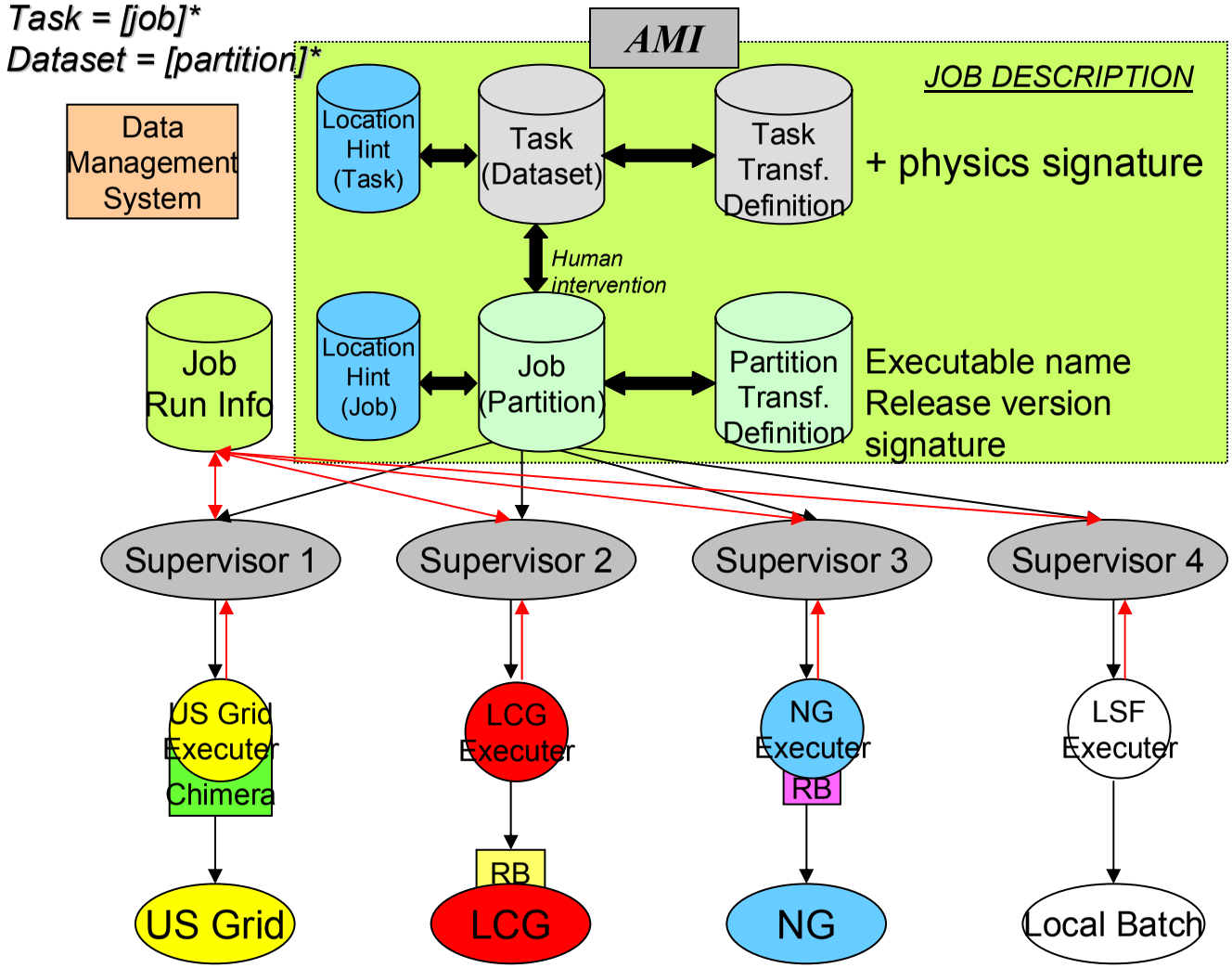


Some Short term plans

- Support for Atlas DC2 and CTB
- "Request a task interface"
- Links to other services CTB, DQ (see below)
- Improvements in the Generic Web search interface.
- Housekeeping – Documentation Web site cleanup.
- AMI installation, deployment , collaborative tools (CVS + WIKI)
- AMI database administration tools.
- (Rewrite Tag Collector)

ATLAS Production System schema

Taken from G. Poulard's presentation ATLAS SW week March 2003



Glasgow Metadata Group

- Initiative of Glasgow (Tony Doyle), finance GridPP,
- http://www.gridpp.ac.uk/datamanagement/metadata/
- "The purpose of the Metadata Group is to examine commonalities across all the PPE experiments' metadata handling, at the technology level, at the interface level, and at the schema level. “

Aim of the group

- “The final aim of the group will be to ensure that the metadata services are deployed as much as possible using common technologies (ideally industry standard technologies), that they are built using existing common interfaces so that the products are interchangeable, and that they use common schema for problems that are common across the experiments (e.g. data product navigation). The group will work within the ARDA process, together with the EGEE and LCG projects. “

Will it all work?

All the data (or perhaps instead “at least 10,000 events” or “retrievable within 2 days”) from first quarter 2007, taken in trigger configuration EFG, reconstructed with reco version 7.4 and calibration set 21.d3, for which p_t>150 GeV and a J/Psi was observed. (*Hepcal II page 11*)

- Will the right info be in the right DB. Have we forseen all the links needed?
- Need some kind of intelligence to pre-analyse and split queries.
- Already we have seen that the Application metadata “dataset catalogue” needs a link to the replica catalogue to satisfy user requirements of the web interface.