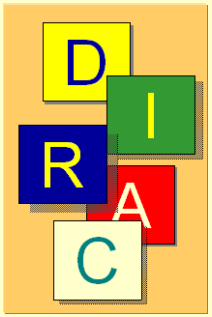


DIRAC, the LHCb Data Production and Distributed Analysis system

A.Tsaregorodtsev,
CPPM, Marseille

CHEP 2006 , 13-18 February 2006, Mumbai, India

Authors

- DIRAC development team

  TSAREGORODTSEV Andrei, GARONNE Vincent, STOKES-REES Ian, GRACIANI-DIAZ Ricardo, PATERSON Stuart, CASAJUS Adria, CLOSIER Joel, SABORIDO SILVA Juan, KUZNETSOV Gennady, CHARPENTIER Philippe, BROOK Nicholas, SMITH Andrew, SANCHEZ Manuel, SANTINELLI Roberto


University of Bristol

- Production site managers

  BLOUW Johan, PICKFORD Andrew , CARBONE Angelo, BERNET Roland, VAGNONI Vincenzo



ITEP



Universidade Federal do Rio de Janeiro

Outline

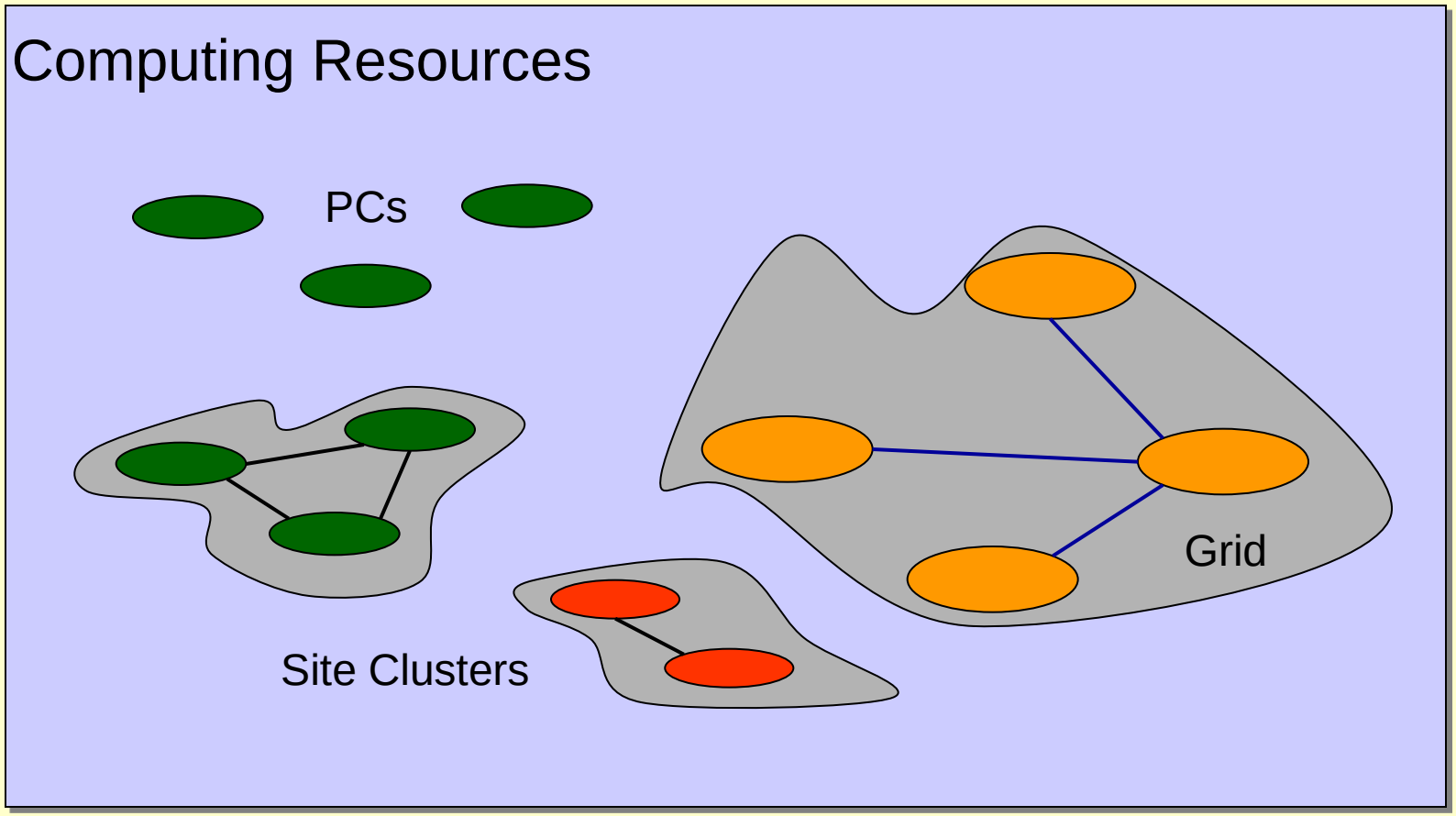
- DIRAC basic paradigm
- Architecture and components
- Operational experience
- Conclusion

Introduction

- DIRAC is a distributed data production and analysis system for the LHCb experiment
  - Includes workload and data management components
  - Was developed originally for the MC data production tasks
  - The goal was:
    - integrate all the heterogeneous computing resources available to LHCb
    - Minimize human intervention at LHCb sites
  - The resulting design led to an architecture based on a set of services and a network of light distributed agents

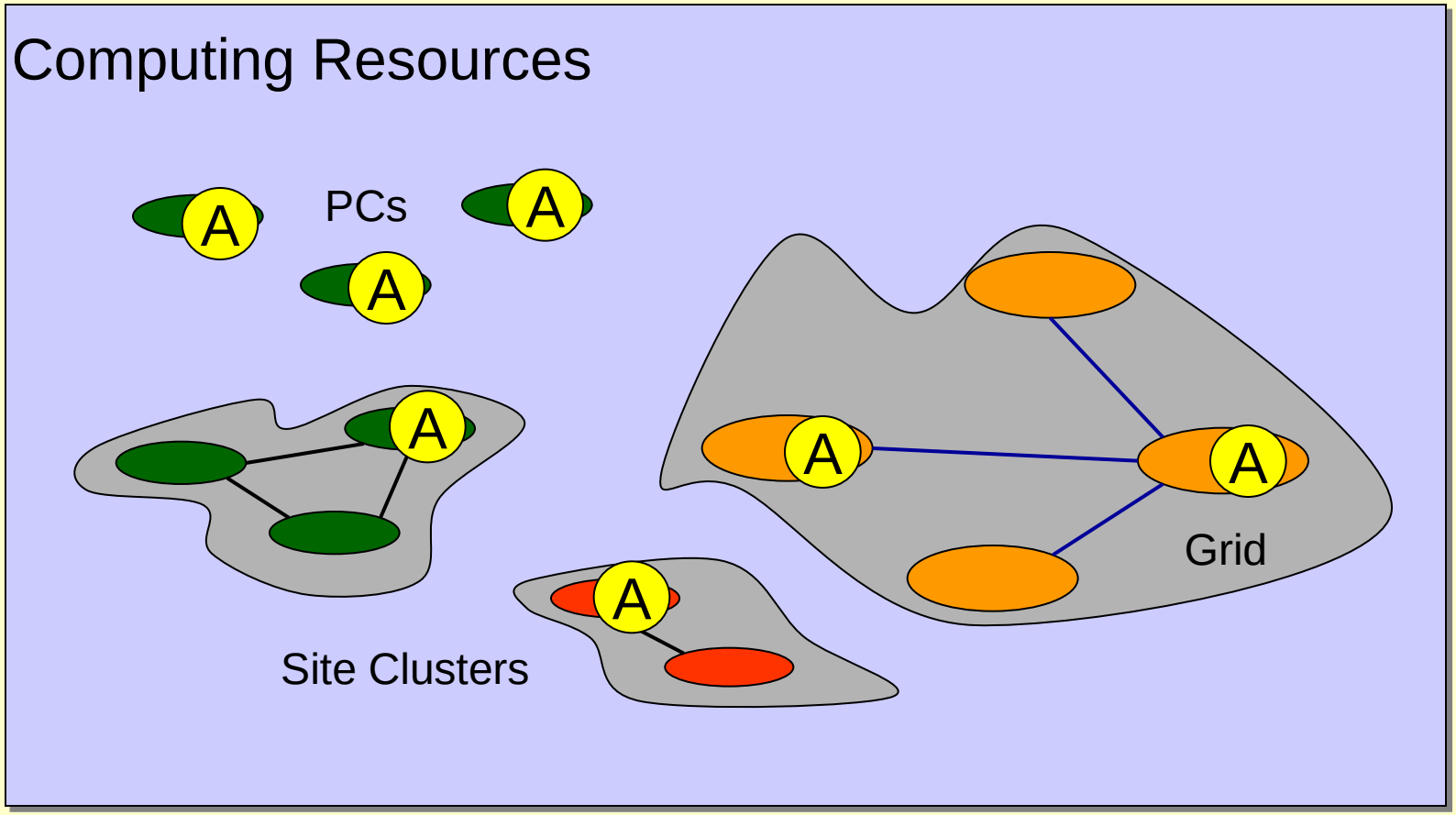
Overlay network paradigm

Various computing resources to deal with



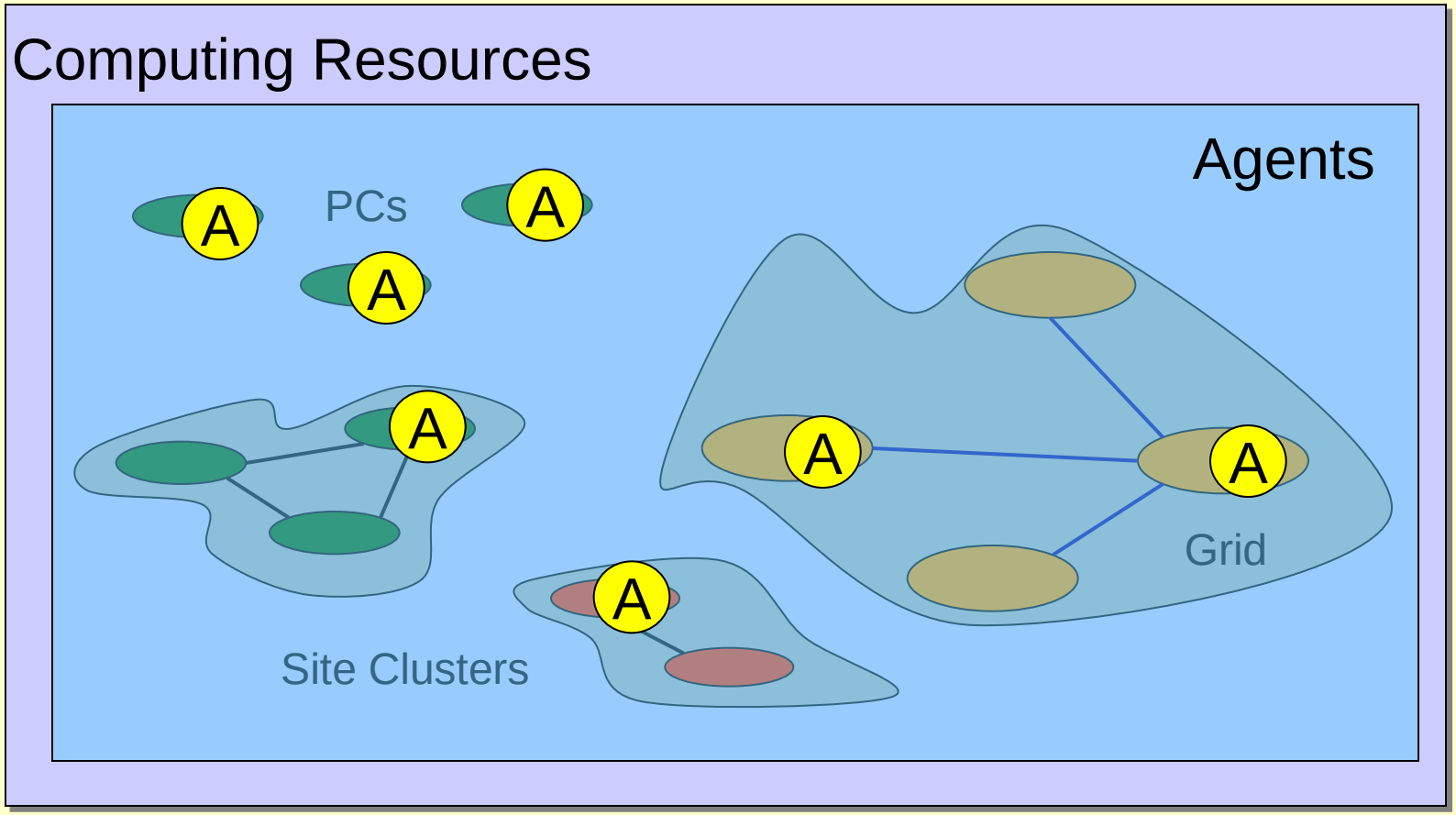
Overlay network paradigm

Placing agents close to resources



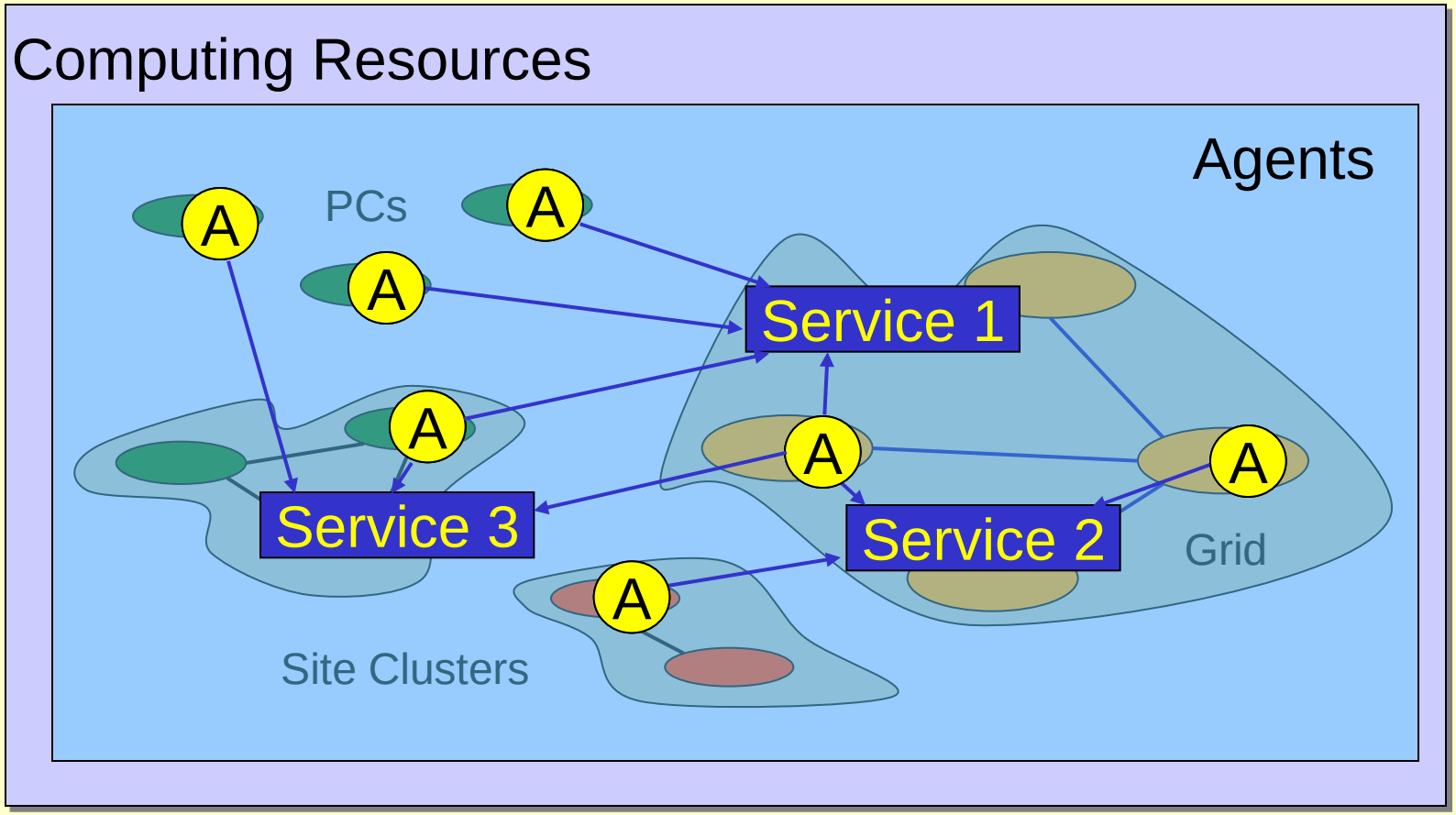
Overlay network paradigm

Agents form and overlay layer hiding the underlying diversity



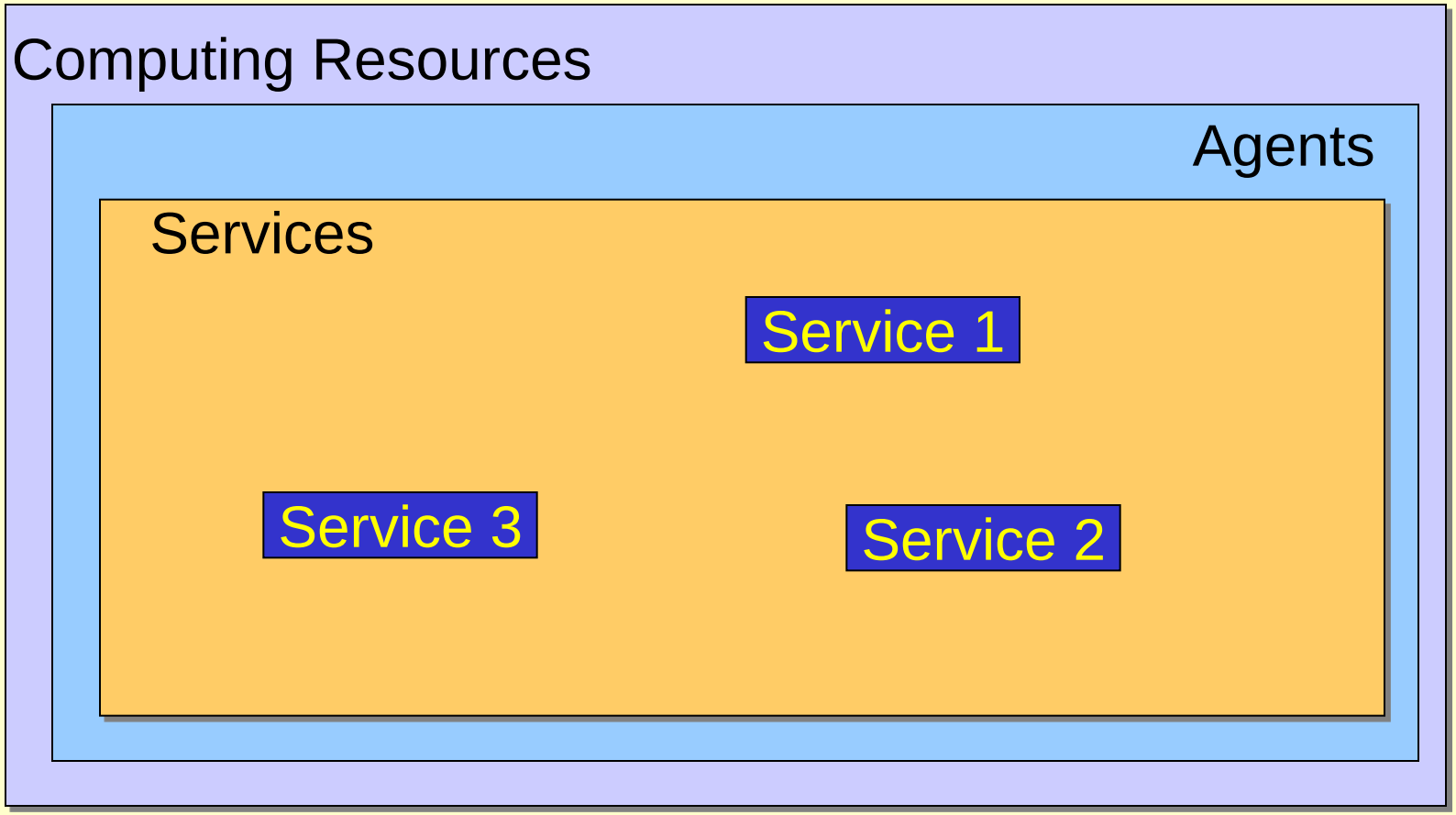
Overlay network paradigm

Agents form and overlay layer hiding the underlying diversity



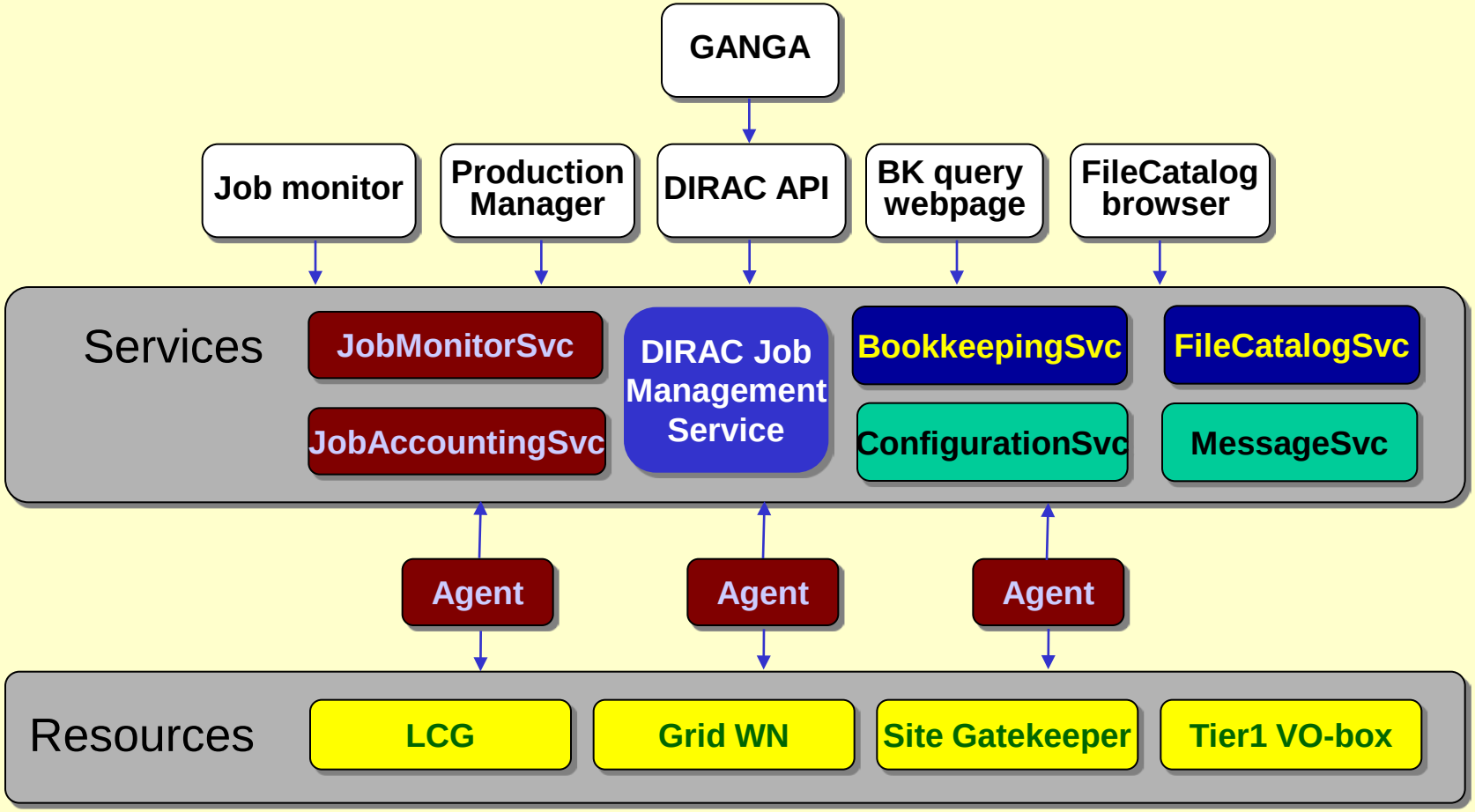
Services seen by users

Users communicate with services to execute their tasks



Architecture and components

DIRAC Services, Agents and Resources

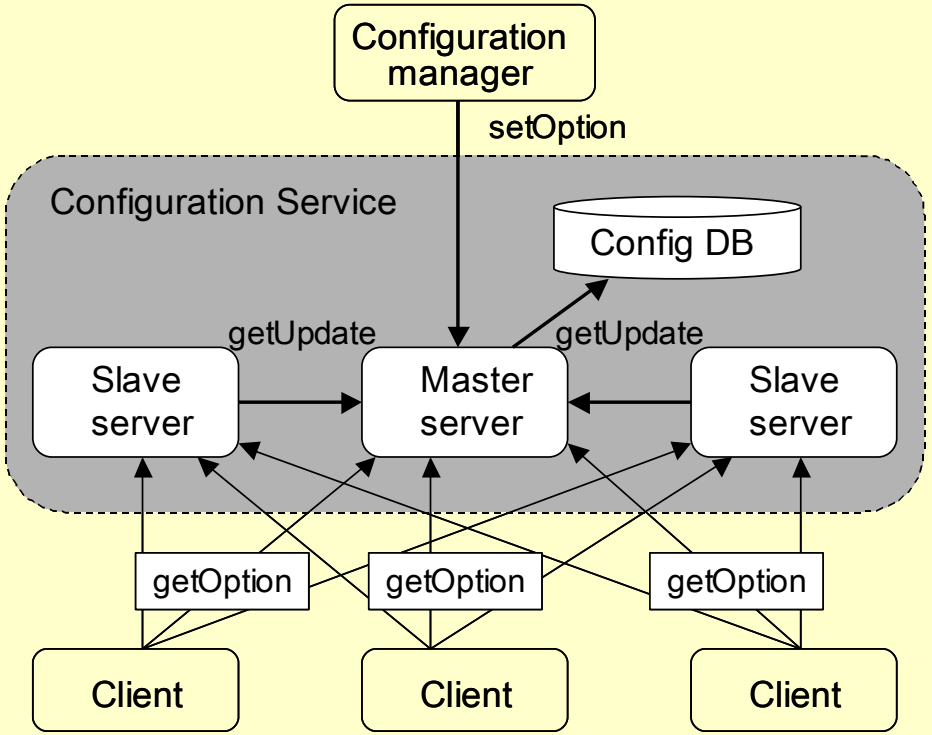


DIRAC Services

- DIRAC Services are permanent processes deployed centrally or running at the VO-boxes and accepting incoming connections from clients (UI, jobs, agents)
- Reliable and redundant deployment
  - Running with watchdog process for automatic restart on failure or reboot
  - Critical services have mirrors for extra redundancy and load balancing
- Secure service framework:
  - XML-RPC protocol for client/service communication with GSI authentication and fine grained authorization based on user identity, groups and roles
  - *See [112] DIRAC Security Infrastructure by Adria Casajus Ramo*

Configuration service

- Master server at CERN is the only one allowing write access
- Redundant system with multiple read-only slave servers running at sites on VO-boxes for load balancing and high availability
- Automatic slave updates from the master information
- Watchdog to restart the server in case of failures



WMS Service

- DIRAC Workload Management System is itself composed of a set of central services, pilot agents and job wrappers
- Realizes the ***PULL*** scheduling paradigm
  - Pilot agents deployed at LCG Worker Nodes pull the jobs from the central Task Queue
- The central Task Queue allows to apply easily the VO policies by prioritization of the user jobs
  - Using the accounting information and user identities, groups and roles
- The job scheduling is late
  - Job goes to a resource for immediate execution

File Catalog Service

- LFC is the main File Catalog
  - Chosen after trying out several options
  - Good performance after optimization done
  - One global catalog with several read-only mirrors for redundancy and load balancing
- Similar client API as for other DIRAC “File Catalog” services
  - Seamless file registration in several catalogs
  - E.g. Processing DB receiving data to be processed automatically

DIRAC Agents

- Light easy to deploy software components running close to a computing resource to accomplish specific tasks
  - Written in Python, need only the interpreter for deployment
  - Modular easily configurable for specific needs
  - Running in user space
  - Using only outbound connections
- Agents based on the same software framework are used in different contexts
  - Agents for centralized operations at CERN
    - E.g. Transfer Agents used in the SC3 Data Transfer phase
    - Production system agents
  - Agents at the LHCb VO-boxes
  - Pilot Agents deployed as LCG jobs

Pilot agents

- Pilot agents are deployed on the Worker Nodes as regular jobs using the standard LCG scheduling mechanism
  - Form a distributed Workload Management system
- Once started on the WN, the pilot agent performs some checks of the environment
  - Measures the CPU benchmark, disk and memory space
  - Installs the application software
- If the WN is OK the user job is retrieved from the central DIRAC Task Queue and executed
- In the end of execution some operations can be requested to be done asynchronously on the VO-box to accomplish the job

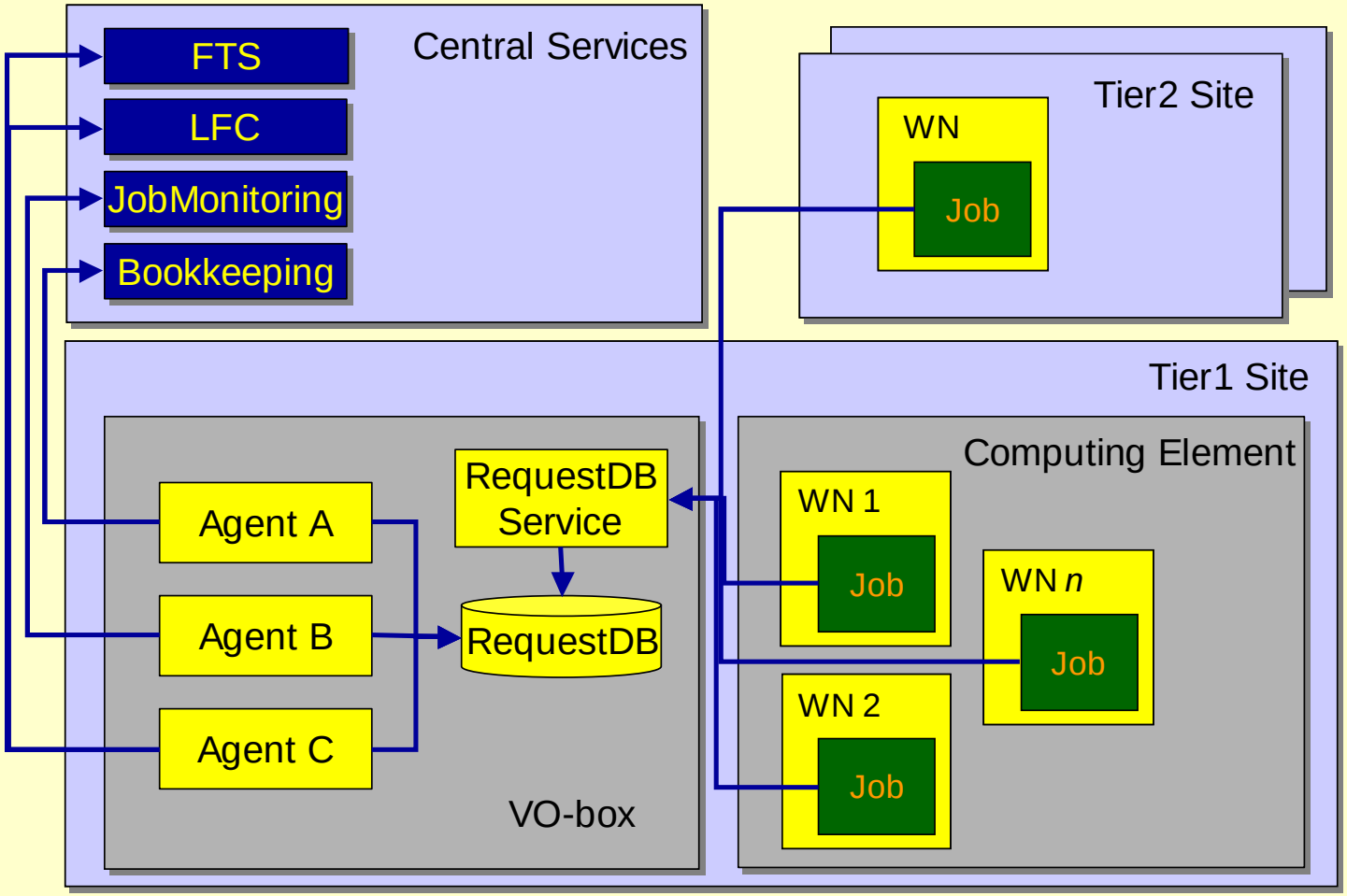
Pilot agents (2)

- The combination of pilot agents running right on the WNs with the central Task Queue allows fine optimization of the workload on the VO level
  - The WN reserved by the pilot agent is a first class resource - there is no more uncertainly due to delays in in the local batch queue
  - Pilot agent can perform different scenarios of user job execution:
    - Filling the time slot with more jobs
    - Running complementary jobs in parallel
    - Preemption of the low priority job
    - Etc
  - Especially interesting for the Distributed Analysis activity
    - *See [260] DIRAC Infrastructure for Distributed Analysis by S. Paterson*

VO-box

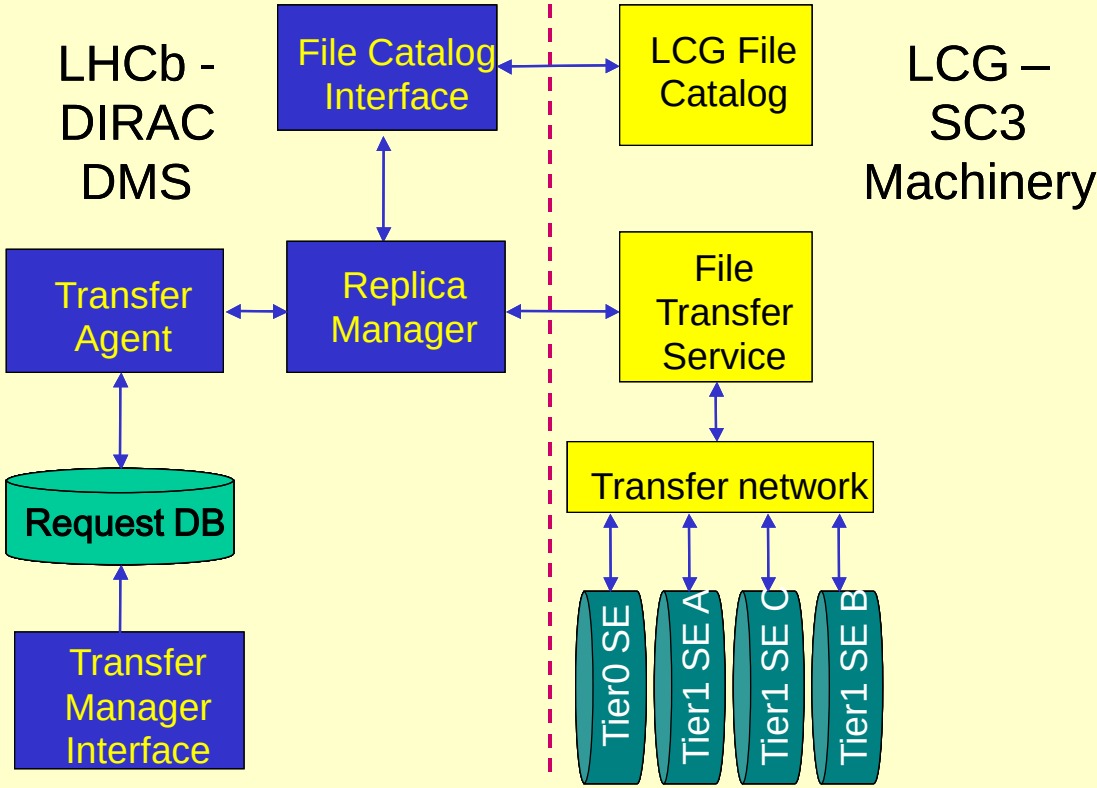
- VO-boxes are dedicated hosts at the Tier1 centers running specific LHCb services for
  - Reliability due to retrying failed operations
  - Efficiency due to early release of WNs and delegating data moving operations from jobs to the VO-box agents
- Agents on VO-boxes execute requests for various operations from local jobs:
  - Data Transfer requests
  - Bookkeeping, Status message requests

LHCb VO-box architecture



Transfer Agent example

- Request DB is populated with data transfer/replication requests from Data Manager or jobs
- Transfer Agent
  - checks the validity of request and passes to the FTS service
  - uses third party transfer in case of FTS channel unavailability
  - retries transfers in case of failures
  - registers the new replicas in the catalog

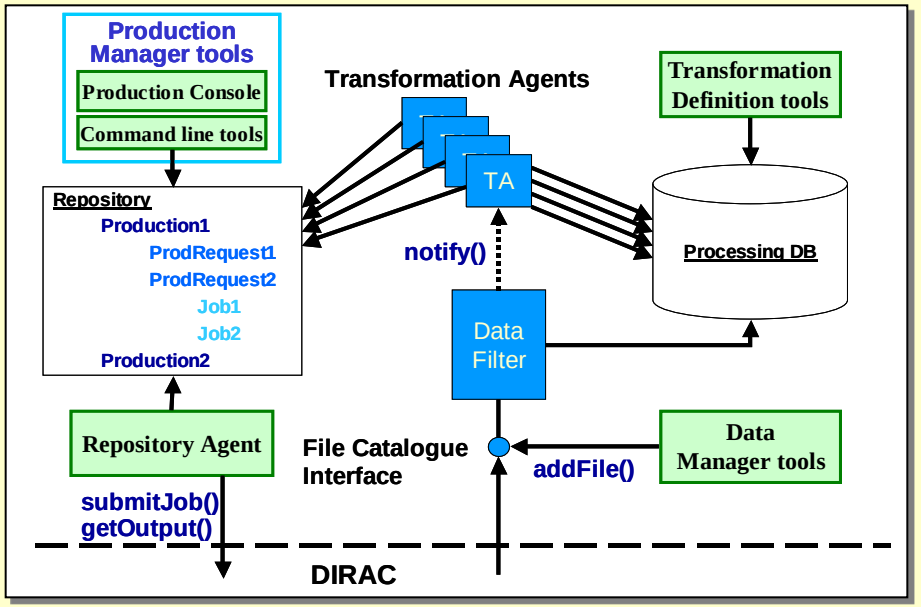


See [191] LHCb Data Replication in SC3 by A.Smith

Operation

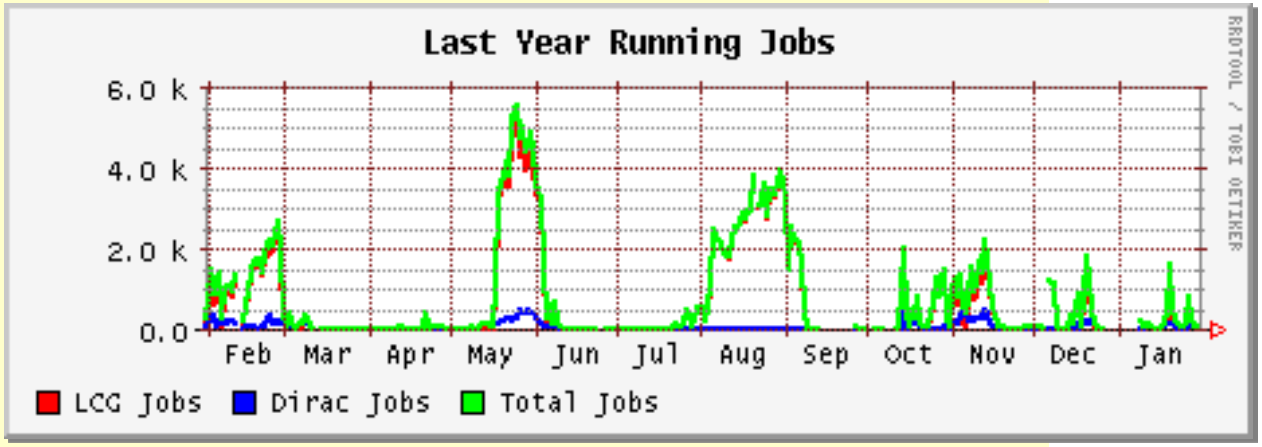
Processing Database

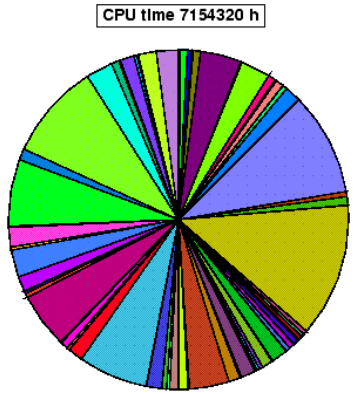
- The suite of Production Manager tools to facilitate the routine production tasks:
  - define complex production workflows
  - manage large numbers of production jobs

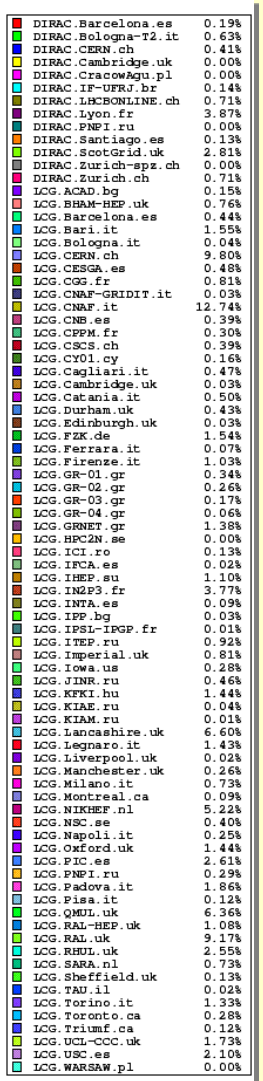


- Transformation Agents prepare data reprocessing jobs automatically as soon as the input files are registered in the Processing Database via a standard File Catalog interface
  - Minimize the human intervention, speed up standard production
- *See [95] DIRAC Production Console by G. Kuznetsov*

DIRAC production performance



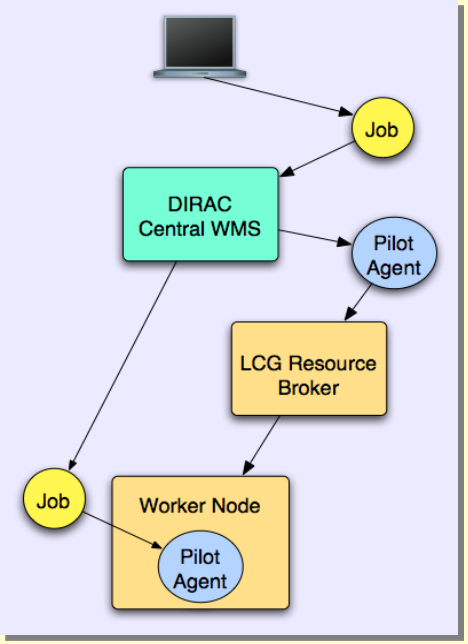




- Up to over 5000 simultaneous production jobs
  - The throughput is only limited by the capacity available on LCG
- ~80 distinct sites accessed through LCG or through DIRAC directly

Distributed Analysis

- The Pilot Agent paradigm was extended recently to the Distributed Analysis activity
- The advantages of this approach for users are:
  - Inefficiencies of the LCG grid are completely hidden from the users
  - Fine optimizations of the job turnaround
    - It also reduces the load on the LCG WMS



- The system was demonstrated to serve dozens of simultaneous users with about 2Hz submission rate
  - The limitation is mainly in the capacity of LCG RB to schedule this number of jobs
- *See [260] DIRAC Infrastructure for Distributed Analysis by S. Paterson*

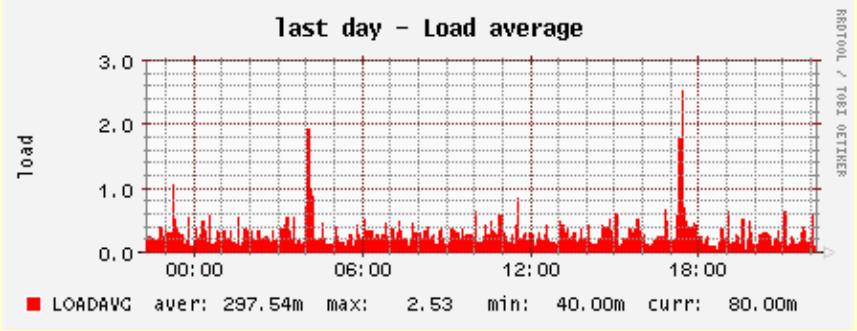
Conclusions

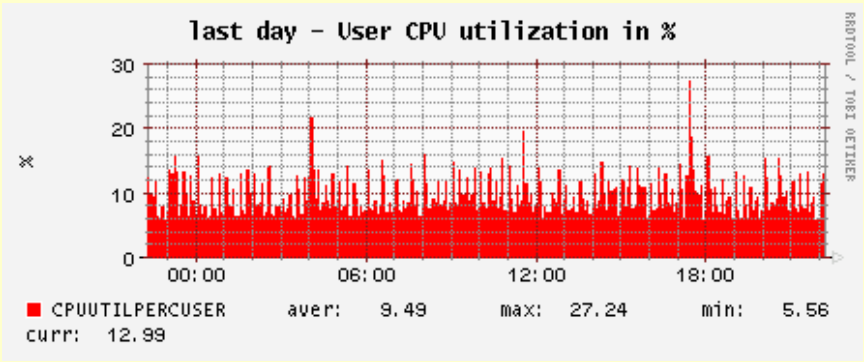
- The Overlay Network paradigm employed by the DIRAC system proved to be efficient in integrating heterogeneous resources in a single reliable system for simulation data production
- The system is now extended to deal with the Distributed Analysis tasks
- The LHCb Data Challenge 2006 will serve as un ultimate test of the DIRAC system on the eve of the LHC start

Back-up slides

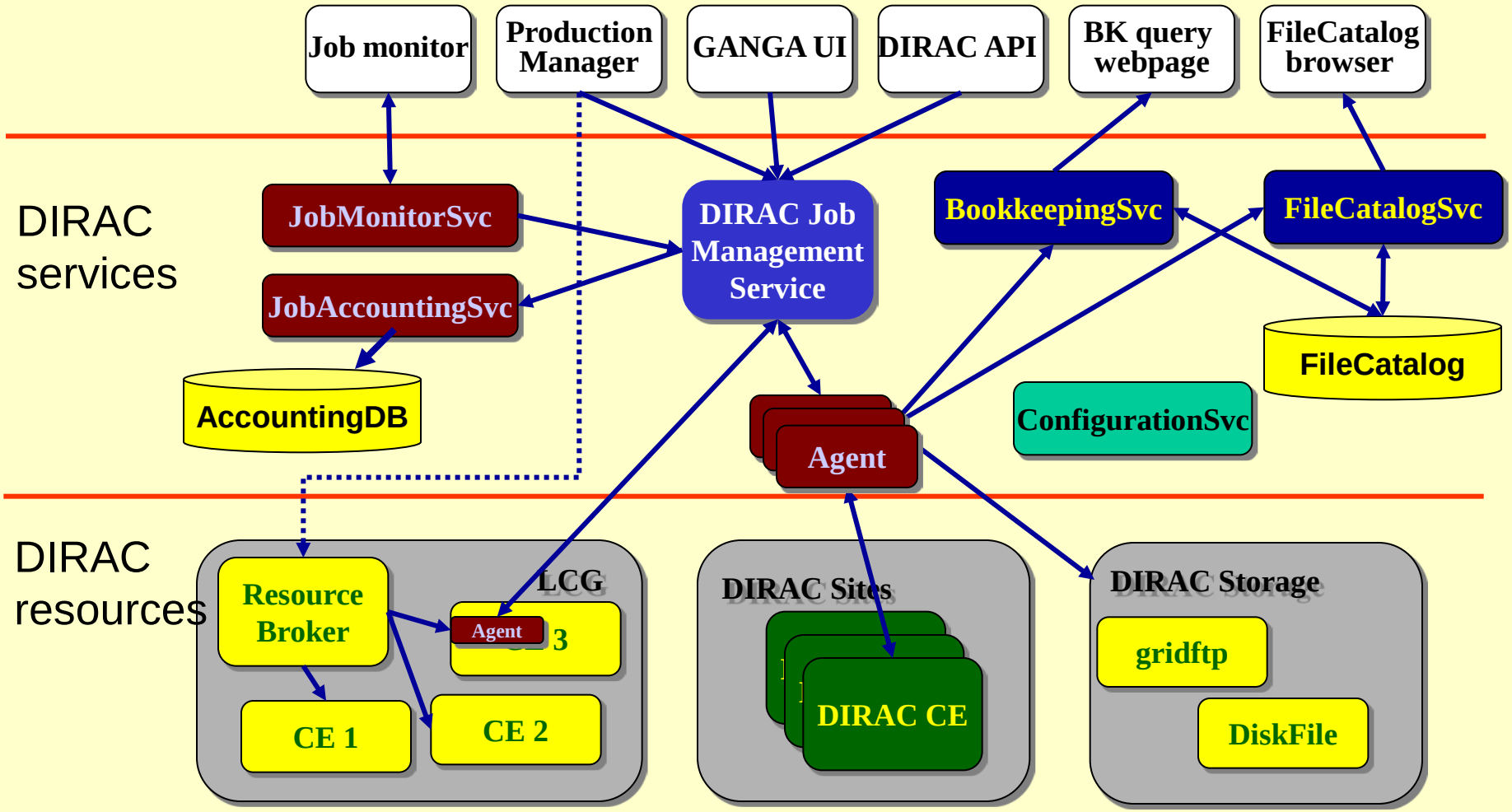
DIRAC performance

- Performance in the 2005 RTTC production
  - Over 5000 simultanuous jobs
    - Limited by the available resources
  - Far from the critical load on the DIRAC servers



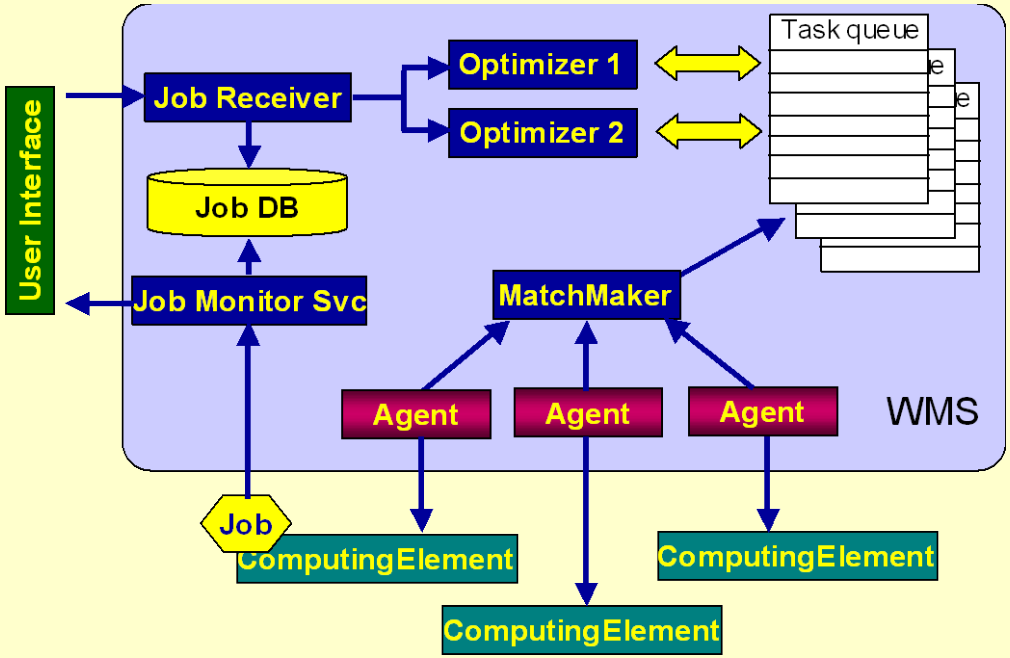


DIRAC Services and Resources



DIRAC workload management

- Realizes ***PULL*** scheduling paradigm
- Agents are requesting jobs whenever the corresponding resource is free
- Using Condor ClassAd and Matchmaker for finding jobs suitable to the resource profile
- Agents are steering job execution on site
- Jobs are reporting their state and environment to central Job Monitoring service



Other Services

- Job monitoring service
  - Getting job heartbeats and status reports
  - Service the job status to clients (users)
    - Web and scripting interfaces
- Bookkeeping service
  - Receiving, storing and serving job provenance information
- Accounting service
  - Receives accounting information for each job
  - Generates reports per time period, specific productions or user groups
  - Provides information for taking policy decisions