Summary of the session "Workload management systems and Workflow systems"

Harald Kornmayer (FZK)

3rd of march 2006

Contributors

- This summary is based on the contributions by the speakers and the discussions during the session!
- **Thanks to all contributors**:
  - **Logging and Bookkeeping and Job Provenance services**
    - Ludek Matyska
  - **The gLite Workload Management System**
    - Francesco Giacomini
  - **BOSS: the CMS interface for job summission, monitoring and bookkeeping**
    - Guiseppe Godispoti
  - **MOTEUR: a data intensive service-based workflow engine enactor**
    - Tristan Glatard
  - **K-Wf Grid: Knowledge-based Workflows in Grid**
    - Ladislav Hluchy
  - **G-PBox: A framework for grid policy management**
    - Andrea Caltroni
  - **IBM strategic directions in workload virtualization"**
    - Jean-Pierre Prost
- Thanks to personnel discussions over the last two days

My view point

- **One user of the EGEE infrastructure as a member of the MAGIC VO**
- **Some experience in industry**
  - Building a risk analysis system in a bank using SOA techniques
- **Some Grid experience**
  - CrossGrid WP leader
  - D-Grid
  - g-Eclipse (under negotiations)

General remarks

- **Workload management**
  - is important to make the Grid resources invisible
    - "I don't care where my job runs!"
  - must support users with the basic functionality to implement their IPO (Input, Processing, Output)
    - Input and Output is of same importance as Processing
    - This is the "atomic" transaction on a Grid
  - EGEE is the only big scale infrastructure which provides a scheduling service
    - Not in UNICORE
    - not in GT2, GT4

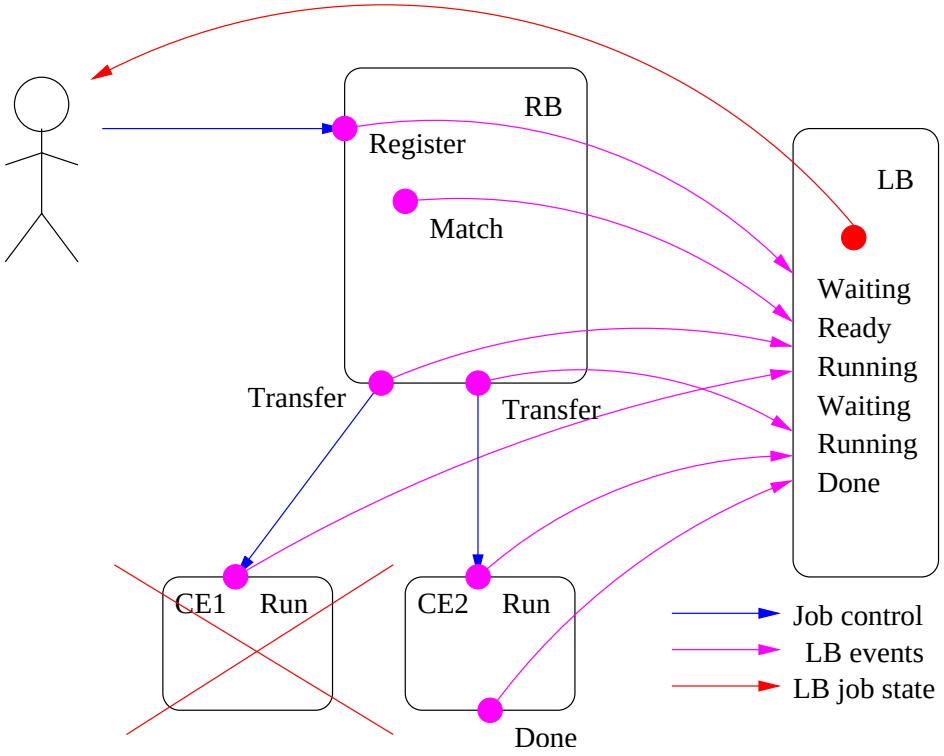
General remark II

- **Workflow systems**
  - Composition of jobs
  - Abstraction of (e-Science) processes
  - Fundamental for complex and automated analysis and business applications
  - Only documented workflows will enable “reproducible” results in the future
  - Workflows will be the tool in the future to ask “simple” question to the Grid infrastructure
    - Do the “Atlas/LHCb/Alice/CMS” measurements agree with the assumption of super symmetry with a given set of free parameters?

Logging and bookkeeping (L&B)

- **Motivation: Keep track of Grid jobs**
  - Part of Workload management system
- **L&B events as important points in the flow control of job**
  - Submission
  - Transfer between components
  - Match making and brokerage results
  - Starting/finishing job execution
  - Events generated directly by user
    - Only during the actual job execution
- **Information is available through the L&B querying mechanism**
  - Through the UI or public API
- **Users can store events in the bookkeeping DB**
  - Non-blocking reliable mechanism for passing job related information

Interaction overview (L&B)

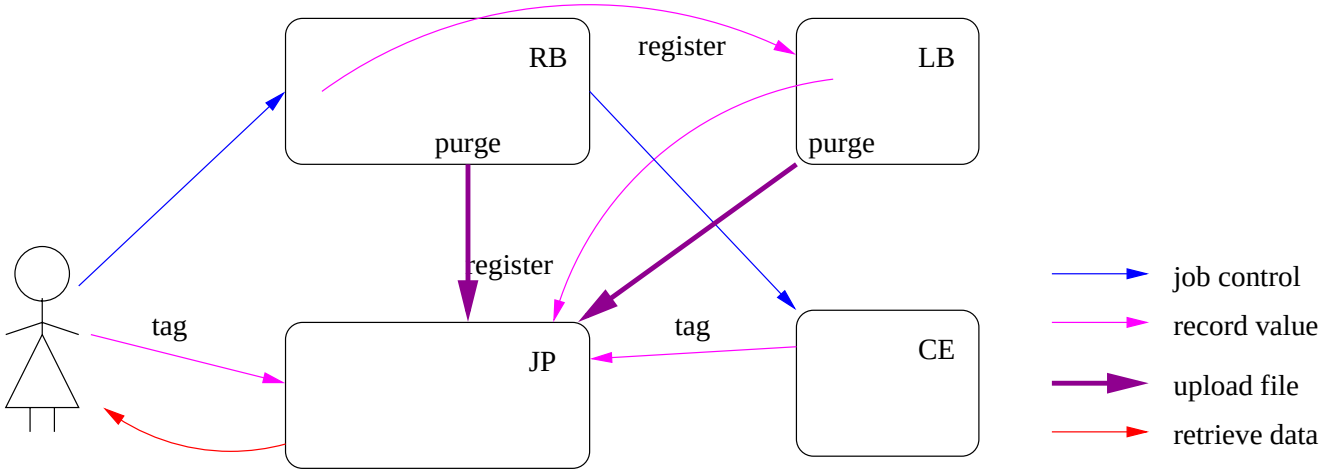


Open question from Users

- How can user defined events be instrumented in the application?
  → API
- Cyptical output of “get-logging-info” was mentioned by users
  - most information useless for non experts
  - documentation!
  - but tools exists
    → feed back from users needed

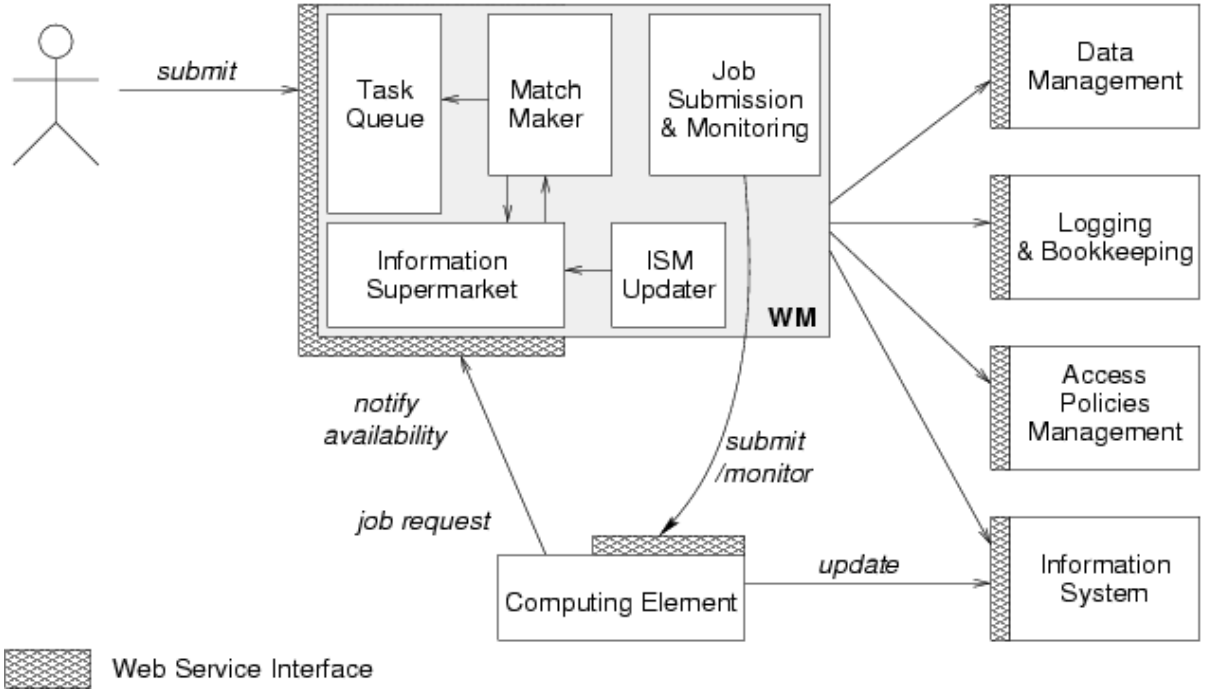
Job provenance (JB)

- **Motivation: provide long term information about jobs**
  - E.g. repeat a submission of a job executed year ago
  - The information about job control flow and job execution environment complements job results
- **Job Provenance**
  - Preserve information about Grid jobs (e.g. the input sandbox, …)
  - Allow data-mining in this information
  - Assist job re-submission
  - Additional system to WMS



gLite Workload management System

- **What it is:**
  - The Workload Management System (WMS) is a collection of components providing a service **responsible** for the **distribution** and **management** of tasks across resources available on a Grid, in such a way that applications are conveniently, efficiently and effectively executed
- **Keywords**
  - Matchmaking
  - Persistency
  - Job monitoring
  - Security
  - Genericity



gLite Workload Management System

- **Job Types of the new WMS system:**

Batch-like	**Simple workflow (as DAG)**
Collection	**Parametric**
Interactive	**MPI**
(Checkpointable)	**(Partitionable)**

- **Job Description Language**
  - Uniform language to express the characteristics, requirements and preferences of a job
    - What is about the status of a job, where the JDL defined registration of output files fail?
      - *DONE (SUCCESSFUL) without a successful registration of data*

- **WMS current topics:**
  - Adoption of standards: JSDL
    - WS-I, GGF and OASIS
  - Integration with other services
    - VO support (VOMS), policies (G-PBox), Accounting (DGAS)
  - Scalability
  - New features
    - Workflows
    - error recovery
    - integration with data management
- **Questions:**
  - Is a clustering of WMS possible to increase reliability and performance?
  - Can the system deal with fault tolerance?
    - (E.g rescheduling of jobs waiting in a queue)

BOSS

- **BOSS:**
  the CMS interface for job submission, monitoring and bookkeeping
  - A tool for batch job submission, real time monitoring and bookkeeping
    - interface to local and grid schedulers
    - retrieval of user defined info from process STDOUT/ERR
    - store job-specific logging and bookkeeping info in a relational DB
    - provide real time monitor
  - Optimized for use in distributed environment
    - localhost, LSF, PBS, Condor, LCG, gLite, …
  - A BOSS job is a single elaboration unit
    - Can be made of multiple processing steps (user executables)
    - allows complex workflows: executables chaining
    - Chaining tool may be external
  - Multiple identical jobs can be grouped in Tasks:
    - Logical grouping of jobs
    - compact description of multiple homogeneous jobs using iterators
    - Multiple iterators allowed
    - XML description

BOSS

- **Support for workflows**
  - A BOSS job is a single elaboration unit
    - Can be made of multiple processing steps (user executables)
    - allows complex workflows: executables chaining
    - Chaining tool may be external
  - Multiple identical jobs can be grouped in Tasks:
    - Logical grouping of jobs
    - compact description of multiple homogeneous jobs using iterators
    - Multiple iterators allowed
    - XML description
- **Logging provides long term storage of information**
- **Monitoring provides real time access to logging info**
- **In production since 4 years for CMS**

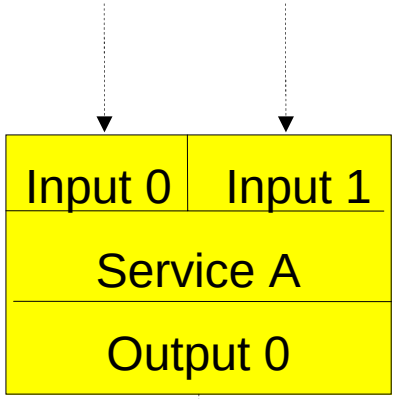
MOTEUR

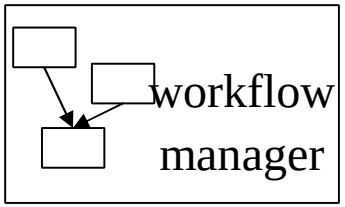
- MOTEUR: a data intensive service-based workflow engine
- Workflow approaches
  - Workflow description
    - Business workflows languages (*e.g.* BPEL)
      - *Control-centric*
    - Scientific workflows languages (*e.g.* Scuffl)
      - *Data-centric*
  - Workflow execution
    - **Task-based** workflows (*e.g.* DAGMan)
      - *Explicit mention of data dependencies*
      - *Complex workflow, simple optimisation*
    - **Service-based** workflows (*e.g.* Taverna, Triana, Kepler)
      - *Independent expression of processors and input data-sets*
      - *Simple workflows, complex optimisation*

← **CS friendly**

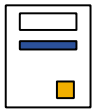
← **user friendly**

- **The service <<isolation>>**

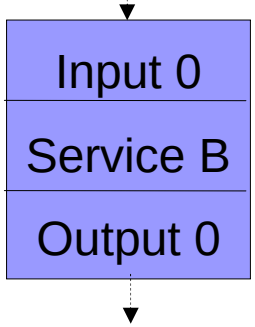




EGEE
User Interface



EGEE
Resources



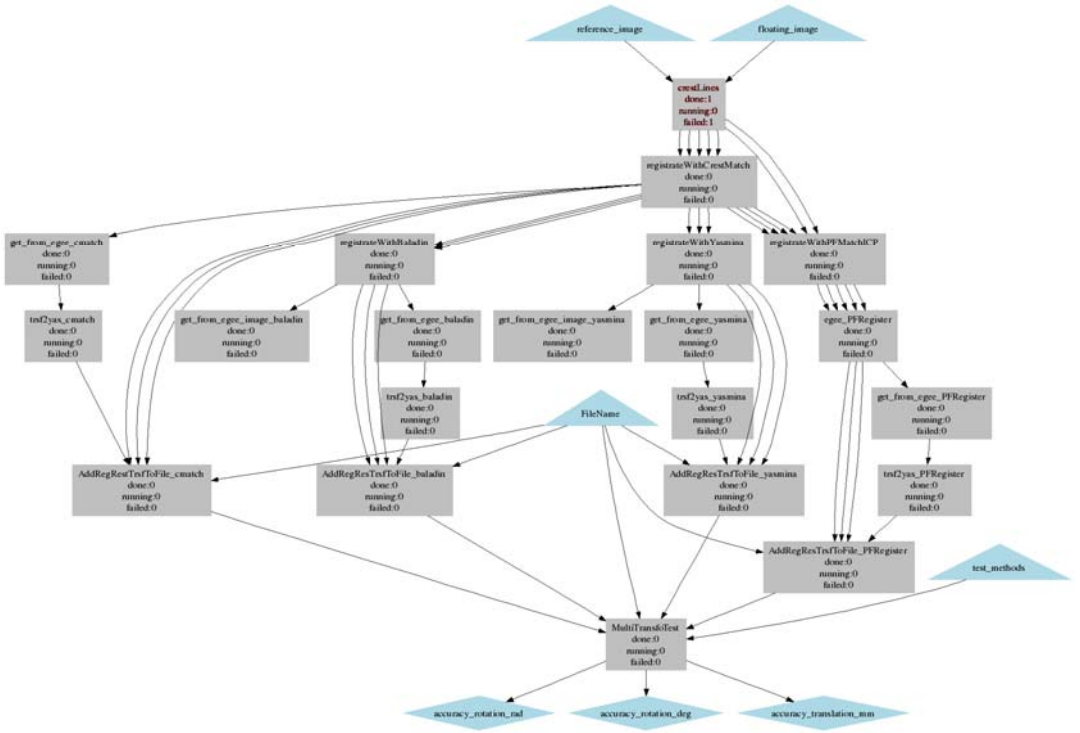
Workflow engine
- Standalone

Web service wrappers for the applications (service A,B)
Running on tomcat
“edg-job-submit”

Description of the workflow

MOTEUR

- **Standards used by Moteur**
  - Workflow language →Scufl
  - Services: → Web Services
- **Infrastructure used by Moteur**
  - Grid 5000 (national Grid infrastructure in France)
  - EGEE#
- **Feature of Moteur:**
  - Optimisation of workflows
    - Intrinsic parallelism
    - Data parallelism
    - …



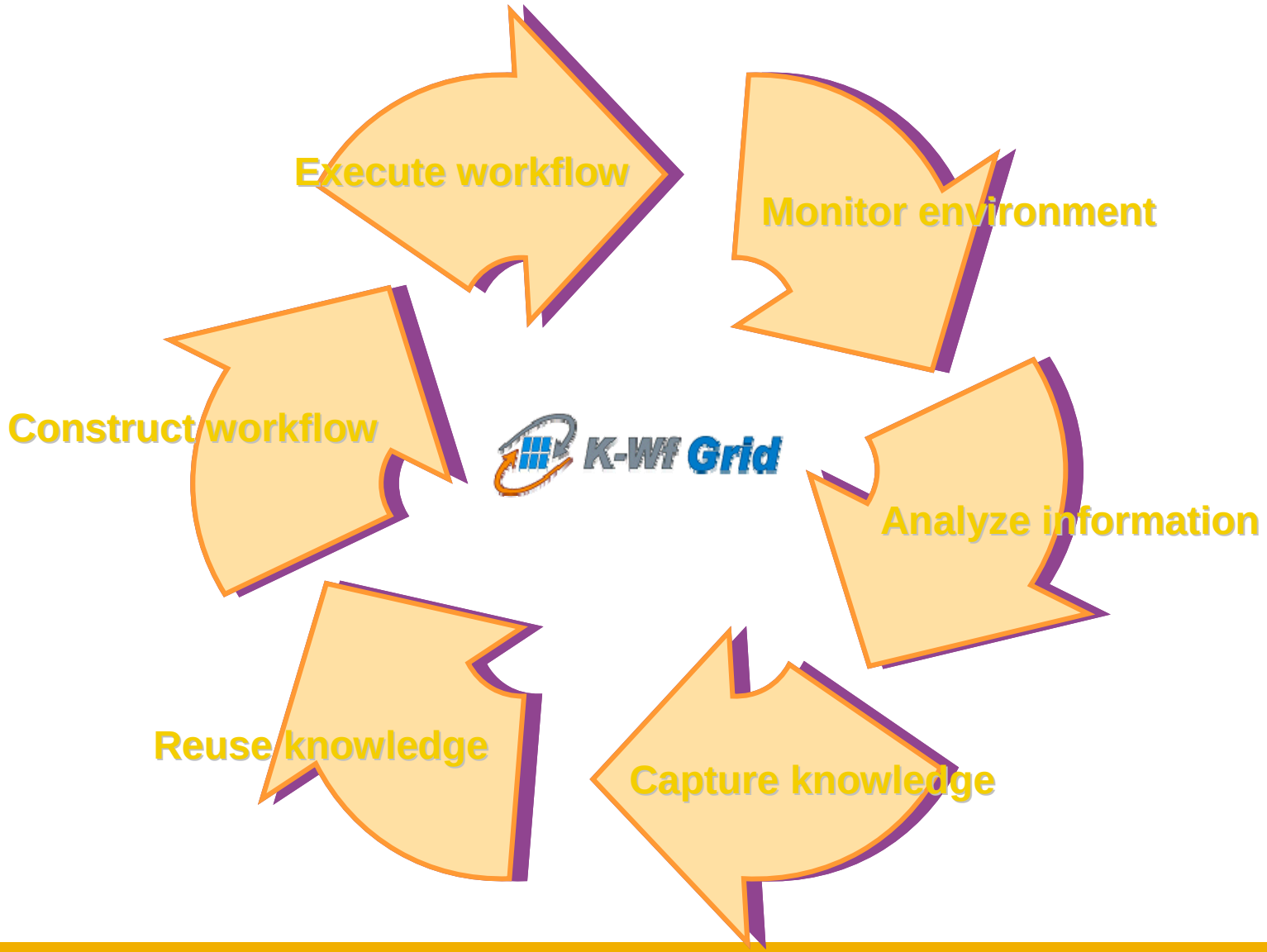
K-Wf Grid

Objectives of the K-WF Grid

- To enable users to **create complex workflows** and use grid resources without detailed knowledge of grid
- To construct workflows optimized for underlying infrastructure, **using its advantages and avoiding its bottlenecks**
- To (semi-)automatically **construct workflows based on user's requirements**, using semantic annotation of services, data, applications and resources
- To constantly renew information about the grid by using complex monitoring network – **to learn from experience**
- To provide **simple, easy-to-use interface** to K-Wf Grid services

- **Composition of workflow from a set of services**
  - System composes the workflow for you – just tell him **what you want to get at the end**
  - Less grid language, more application domain language

K-Wf Grid


Execute workflow
Monitor environment
Analyze information
Capture knowledge
Reuse knowledge
Construct workflow
K-Wf Grid

K-Wf Grid

- **WSRF interfaces to application services**
  - Globus Toolkit 4 implementation
  - Middleware is mostly WS services, deployable as stand-alone components with defined interfaces
- **Web Services – applications and middleware**
- **JSR-168 – user interface portlets**
  - We develop reusable portlets
- **XML**
  - Workflow definition language (GWorkflowDL) is a XML language, with open schema
  - Monitoring information and events are transported as XML-structured data, with open schema
- **Their engine can not run on the EGEE infrastructure currently**

- **Questions:**
  - What is needed from EGEE to benefit from this development?
    - Web services needs to be orchestrated,
      →but there are currently no plans to deploy dynamically webservices on CE/WN

G-PBox: A Framework for Grid Policy Management

- **Motivation:**
  - VO and Site administrators need:
    - A common interface to manage VO and site policies.
    - The possibility to express fine-grained policies
      - *Based on group/role combinations from VOMS.*
    - To distribute policies to other domains (e.g.: ban lists).
    - To be able to explicitly accept/reject policies from other domains
  - Resource owners must have absolute control on their own resources
  - Support the policy distribution process

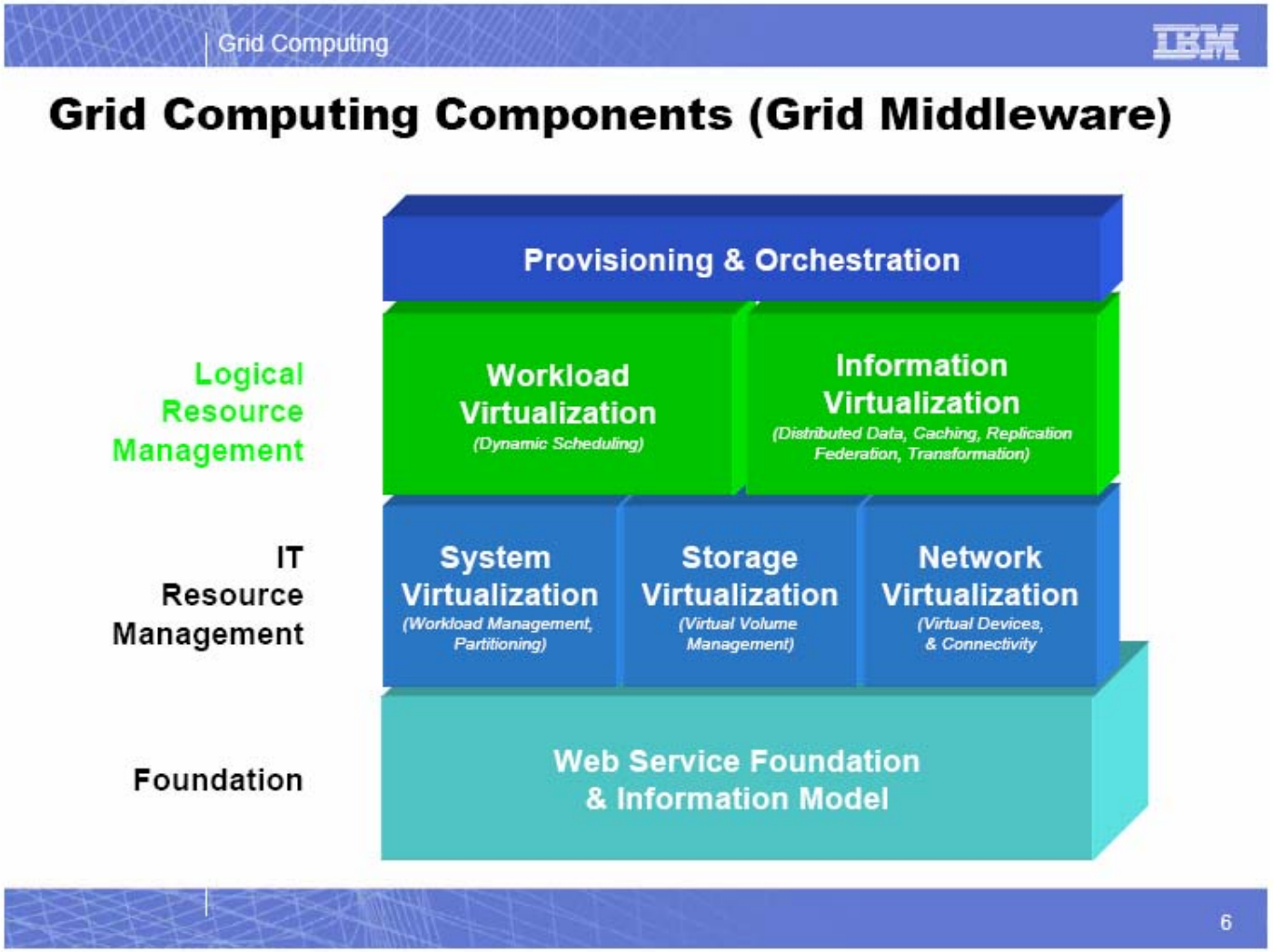
  - Part of gLite, but still under development
    - Feedback from VOs is important!!

G-PBox

Features:

- independent set of modules that can be “plugged in” into the current LCG/EGEE architecture.
- Taking advantage of current standards. (e.g. XACML)
  - XACML allows user-mapping policies (replacing the gridmap file)
- distributed architecture.
- capable of expressing many types of policies:
  - *ACLs*
  - *Management Policies*
  - *Policies depending on parameters of the Grid environment*

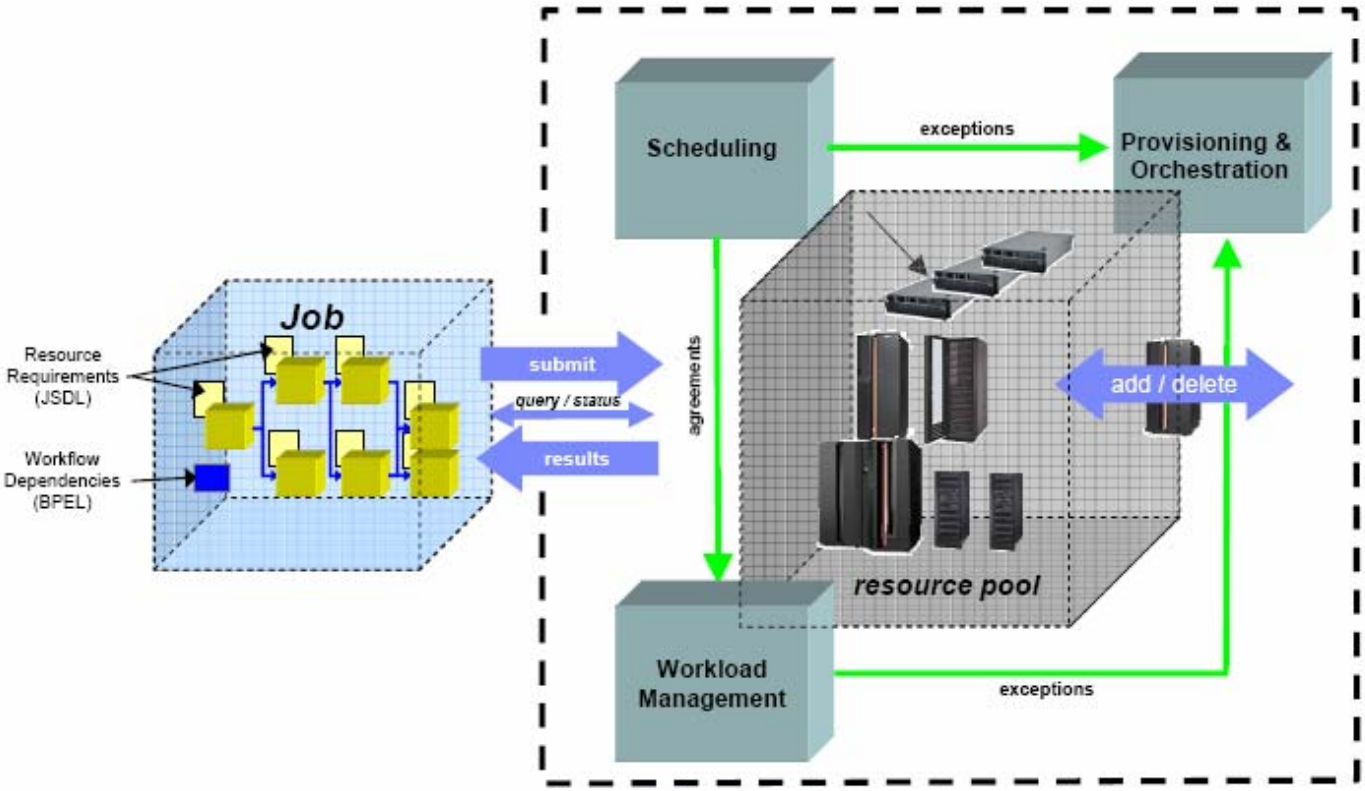
IBM Strategic Directions in Workload Virtualization



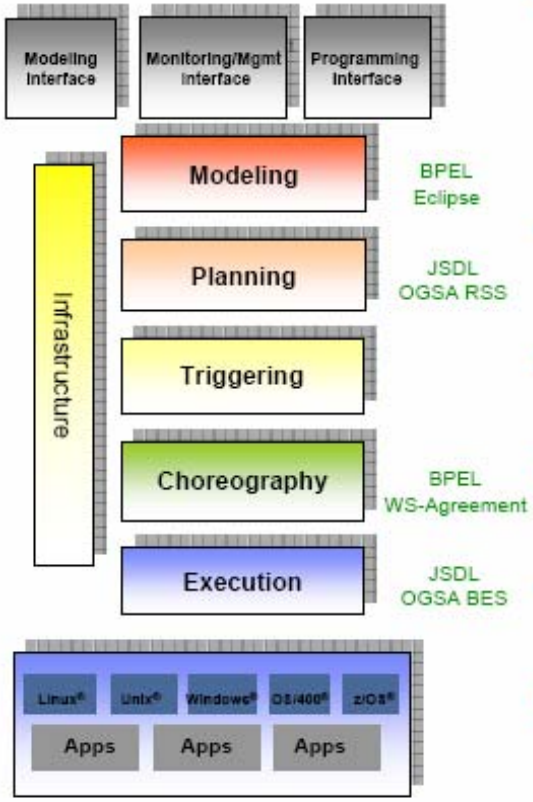
Grid Computing

IBM

Workload Virtualization

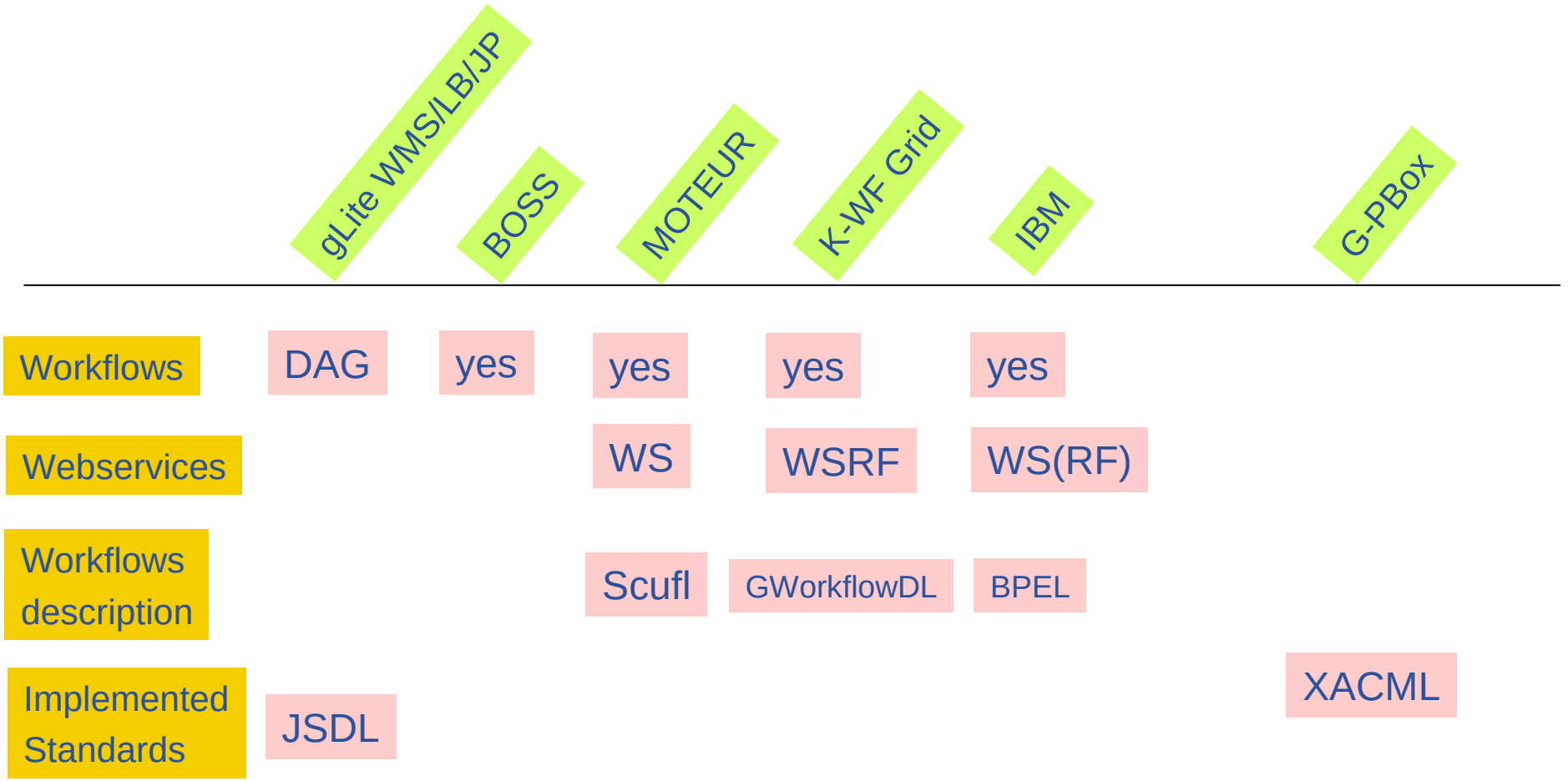


Towards SOA based scheduling



- **Modeling**
  - Models Jobs or Flows of Jobs, such as: System or application commands invocations, Business processes invocations JCLs, etc..
  - Assigns scheduling rules (calendars/times rules), such as
    - Every month on the 1st Monday and on last working day of the month
    - On monday and Friday every three weeks, at 100:00 EST
- **Planning (optional)**
  - Plans the execution of Jobs. Adjusts start times and start sequence, with the scope to optimize the resources usage and maximize the throughput.
- **Triggering**
  - When the specified scheduling rule for a Job or a Job-flow is met, triggers its execution.
- **Choreography**
  - Orchestrates execution of Jobs-flows, based on Jobs dependencies, resource requirements, policies, etc..
- **Execution**
  - Executes a Job on the target system and returns its completion information (return code, results, stdout, stderr, ..)
- **Monitoring and Management**
  - Provides interfaces to monitor and manage (i.e. re-run, kill, etc..) the execution of Jobs.

Summary


gLite WMS/LB/JP
BOSS
MOTEUR
K-WF Grid
IBM
G-PBox
Workflows
DAG
yes
yes
yes
yes
Webservices
WS
WSRF
WS(RF)
Workflows description
Scufl
GWorkflowDL
BPEL
Implemented Standards
JSDL
XACML

Summary II

- **Workload management (Job level) and complex workflows must clearly separated**
  - Don’t overload the WMS
  - Improvements in state machine of WMS possible
- **Job provenance/Workflow provenance is a needed tool for the future**
- **An EGEE workflow engine is needed**
  - Based on a reliable implementation
  - Based on standard workflow description (BPEL)
- **The deployment of application specific web services will help in the orchestration of complex workflows for VOs**
- **Policies managed by the G-PBox will have influence on the scheduling as well as emerging Quality-of-Service mechanisms**

Summary

- **Questions?**

→ Discussions!