WEBIST 2007 - Barcelona

Grid Infrastructure Architecture

A Modular Approach from CoreGRID

Augusto Ciuffoletti - INFN - Italy
Antonio Congiusta - DEIS – Italy
Gracjan Jankowski - PSNC – Poland
Michal Jankowski - PSNC – Poland
Ondrej Krajicek – Univ. Brno – Czeck Rep.
Norbert Meyer - PSNC – Poland

Outlook of a Grid

- A collection of resources
- Available through a transparent interface
- Their use is optimized according to user requests
- Requests are organized into complex tasks
- Resource usage is accounted and billed

Requirements

- Scalability
- Fault tolerance
- Security

The middleware

- The middleware is the complex network of softare functionalities that sits between the resources on one side, and the users on the other.
- It is in charge of implementing the functionalities mentioned in the previous slide.

Separation of concern

- To design the complex layout of a Grid middleware we need to identify basic functionalities.
- This
  - helps a collaborative approach to the design
  - splits the whole concept in a number of more manageable entities
  - improves reusability, robustness...

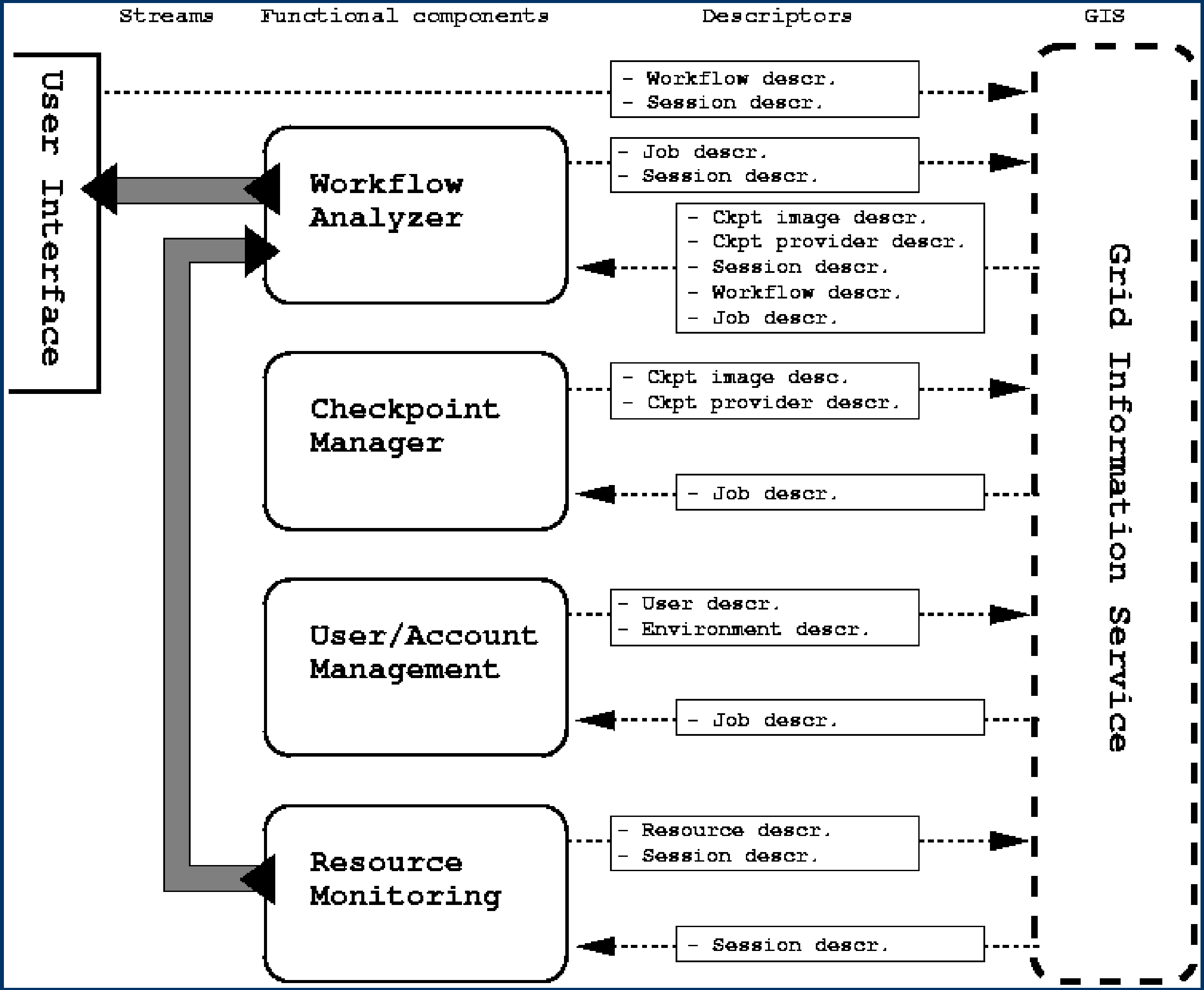
Functional components

- We identify a number of functional components which contribute to the implementation of the Grid Middleware.
- Such components are distributed entities, in their turn consisting in an aggregation of agents interacting according to a lower level pattern.
- Our paper explores the functionalities and the interactions among functional components, and simply outlines their internal structure.

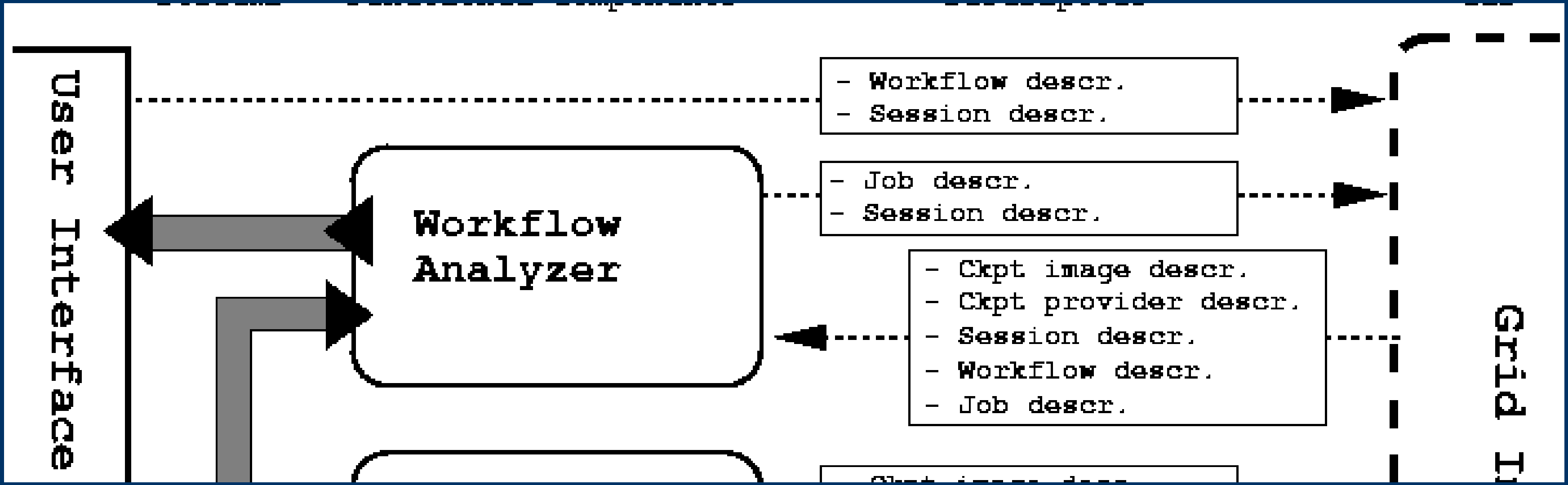
Middleware: functional outline

- We distinguish five functional components, one of them serving as backbone among the others:
  - Workflow Analyzer: user interface, task decomposition, task monitoring
  - Checkpoint Manager: synchronization issues, fault tolerance
  - User/Account Manager: authenticates users, protected environments, accounting
  - Resource Monitoring: observes resource performance, delivers observations
  - Grid Information Service: the backbone.

Middleware layout

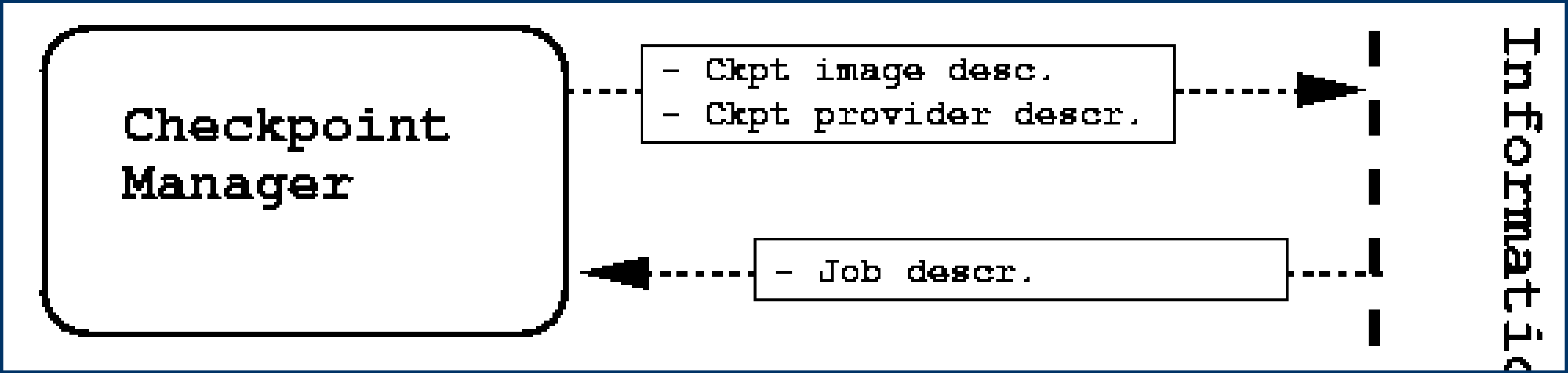


Workflow Analyzer



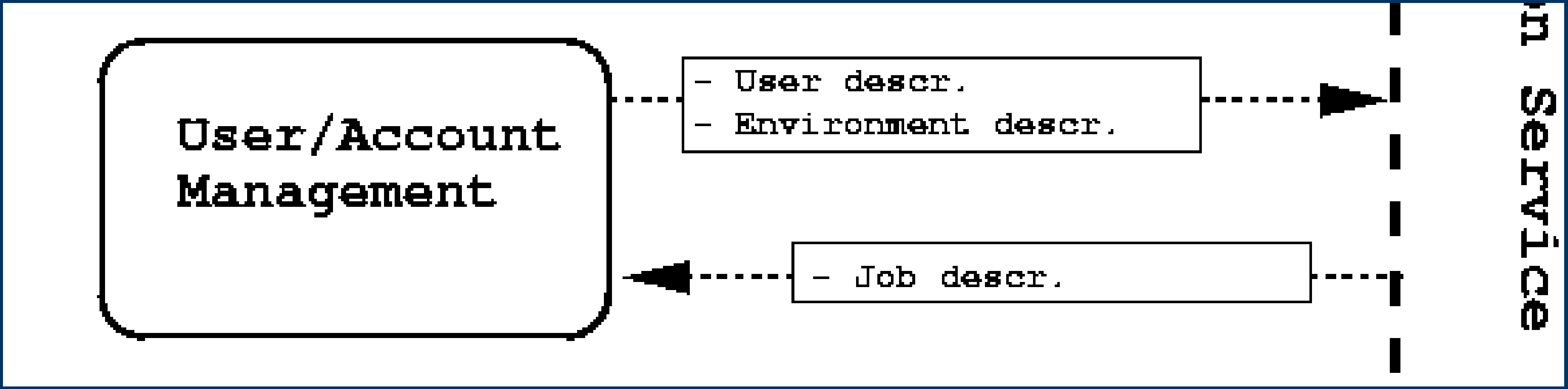
- Interacts with the user (via the GIS) to obtain workflow descriptions and to monitor their execution
- Obtains checkpoint availability from the GIS
- Exchanges job descriptions with the GIS

Checkpoint Manager



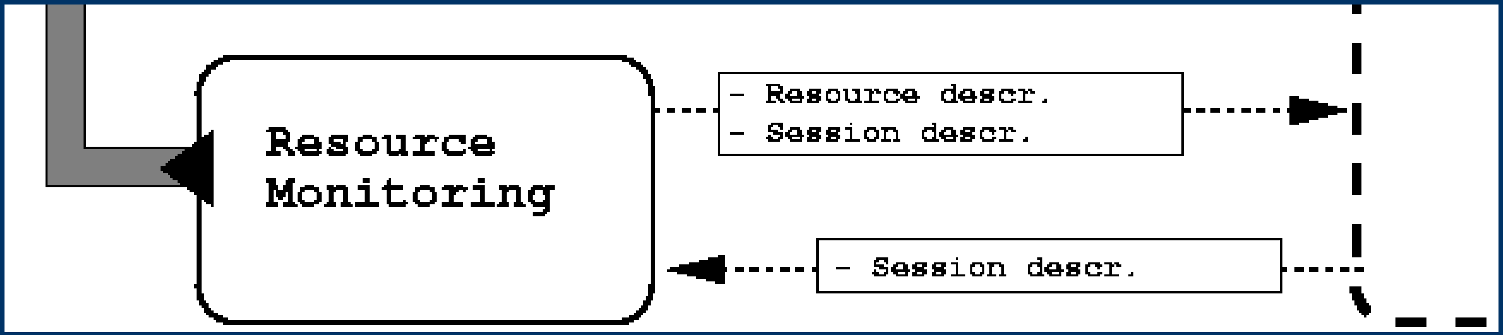
- Fetches job descriptions from the GIS.
- Performs checkpointing/recovery operations including the allocation of dedicated resources.
- Returns checkpointing/recovery features for jobs.

User and Account management



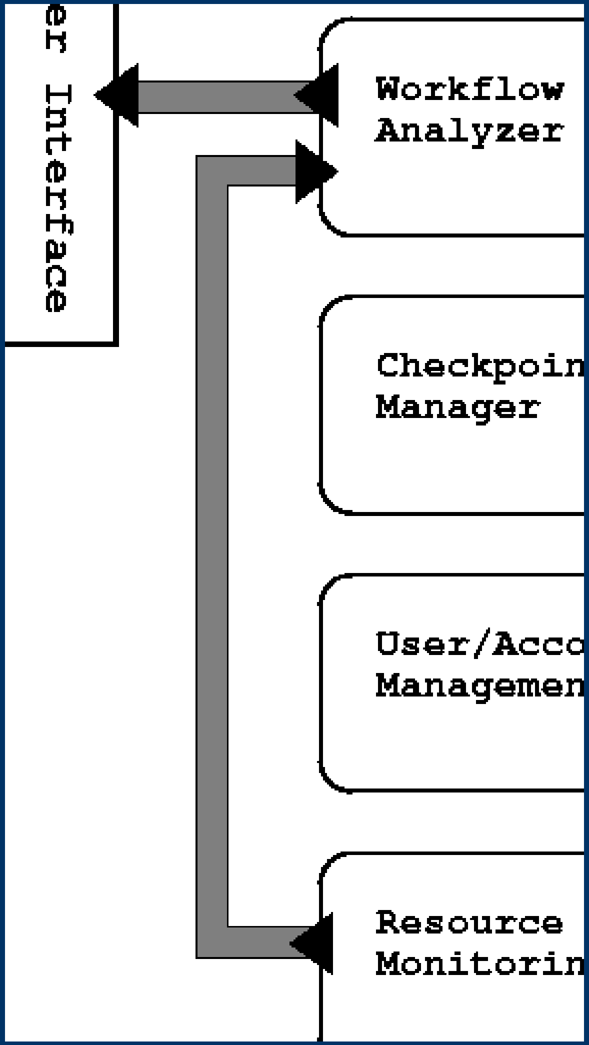
- Fetches job descriptions from the GIS.
- Configures protected execution environments for jobs.
- Publishes user credentials and environment access rules

Resource Monitoring



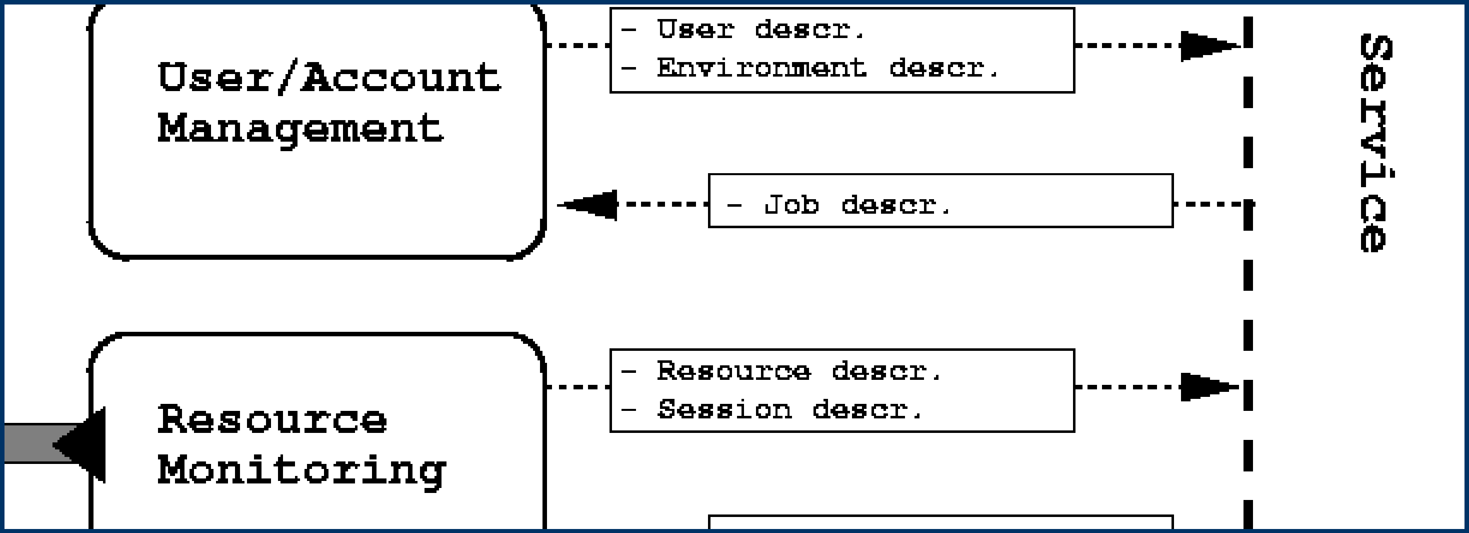
- Fetches monitoring session descriptions from the GIS
- Updates resource descriptions in the GIS
- Updates session descriptions in the GIS (e.g. splits “high level” requests into specific monitoring tasks)

Data Flows



- Traffic related to resource and workflow monitoring has the characteristics of a continuous flow of data
- Routing it through the GIS seems inappropriate (scalability)
- A flow oriented management seems more appropriate: multicast incapsulated into a session

Resources



- Not a component of the middleware, therefore not represented in the layout
- Resources are subject to resource monitoring, managed by user and account management:
  - session description
  - user and environment description

Conclusions

- The architecture tries to modularize the complex structure of the middleware
- We isolate complex functionalities, implemented by coordinated agents
- We describe the communication between those functionalities using data structures published through a Grid Information System
- We take into account security issues
- We consider recent perspective in the implementation of each functionality, using the experience of participating teams