NORDUGRID

Grid Solution for Wide Area Computing and Data Handling

NorduGrid Tutorial

How it works

Aleksandr Konstantinov
Vilnius University/Lithuania and
University of Oslo/Norway
Oxana Smirnova
Lund University/CERN

University of Iceland, Reykjavik, November 17, 2004

- Introduction
- Who am I on Grid
- Services which make NorduGrid/ARC
- Job flow

Introduction

NorduGrid delivers middleware (mediator software to serve as a layer between user and available resources) called **<u>A</u>dvanced <u>R</u>esource <u>C</u>onnector** with purpose to manage computational jobs in distributed environment.

Currently ARC runs on various Linux and few other UNIX-like distributions. Sorry, Microsoft Windows platform is not supported.

- This part of tutorial consists of
  - Description of parts which make ARC
  - Explanation of job flow
- Next part will show how to use ARC in examples.

Complete technical description and user manuals can be found at http://www.nordugrid.org/papers.html

ARC is mostly built on top of Globus Toolkit™. Desription of Globus Toolkit can be found at http://www.globus.org

Security

Security infrastructure is inherited from Globus Toolkit and is based on Public Key Infrastructure (Asymmetric Cryptography).

- Each participant posses 2 digital key: public and private.
  - Private key is kept secure, preferably encrypted by password
  - Public key is freely accessible and can be used to decrypt/check data encrypted/signed with private key.
  - Public keys are signed by higher level participants up to Certificate Authorities, hence validity of each public key can be test using chain of signers. Public keys of Certificate authorities are supposed to be delivered to all participants in some trusted way and are last in chain.
- Globus introduced mediating pair of keys called **proxy** in order not to enter password for every communication. **Proxy** has limited lifetime, is stored in file system and is protected by file system permissions.
- Procedure of creating **proxy** is often referred as "logging into Grid".

```
$ grid-proxy-init
Your identity: /O=Grid/O=NorduGrid/OU=uio.no/CN=Aleksandr Konstantinov
Creating proxy ........................................ Done
Your proxy is valid until: Mon Nov 15 00:33:53 2004
```

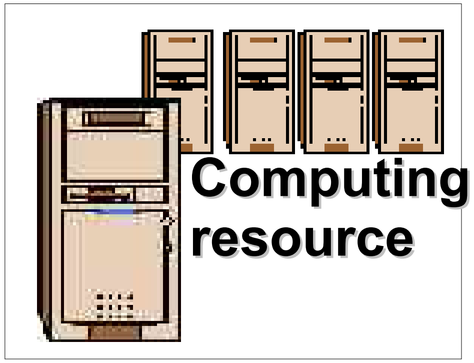
Authorisation

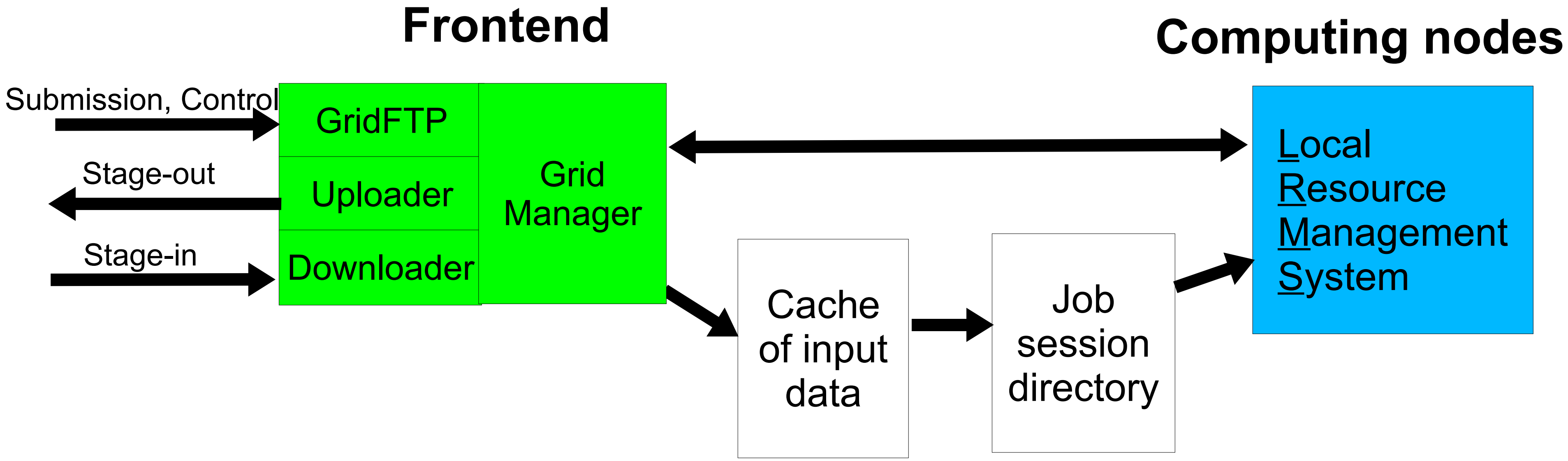
In distributed environment sites have different policies. Instead of asking every site for account **Virtual Organisations** were introduced.

- Each VO groups users which have something in common.
- VOs communicate with resource owners to have access for all members. There are solutions to run sophisticated VO structure with different roles, but they are not widely deployed yet. Hence, so far plain lists of members.
  - NorduGrid Guests VO is probably good start for trying Grid. Contact nordugrid-support@nordugrid.org for membership.
- Once You are member of VO, You will have an access to some resources.

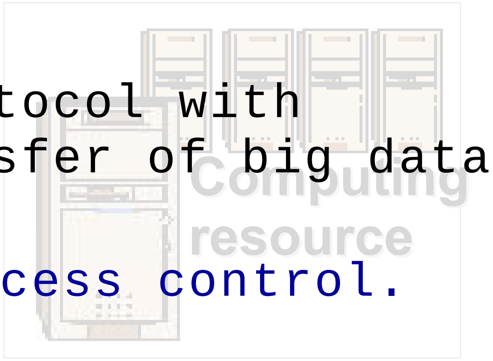
Computing Resource - Grid Manager

- Computing resource
  - Usually cluster of PCs.
  - Runs service responsible for job management called "Grid Manager".
    - Controlled through GridFTP interface.
    - Provide stage-in and stage-out of data.
    - Communicates with cluster software.

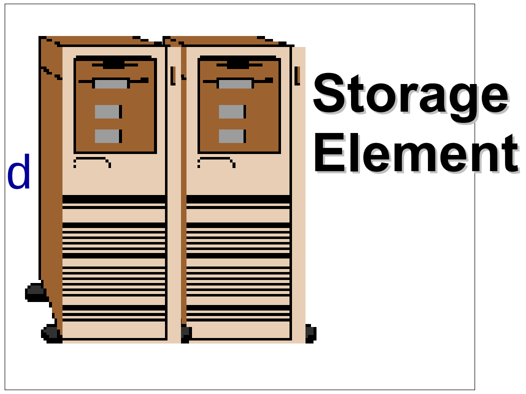




Storage Element - GridFTP and SSE



- Classic Storage Element
  - GridFTP server.
    - GridFTP is an extension for FTP protocol with emphasize on security and fast transfer of big data chunks.
  - Additional features to enhance data access control.
  - Flexible set of backends/plugins.
- "Smart" Storage Element
  - More standard protocols: HTTPS/G, SOAP.
  - Flexible access control integrated.
  - Data transfer without client's control with data integrity check.
  - Integrated support for data replication.
  - Direct interface to Data Indexing Services.
  - Interface to Global Grid Forum standard SRM v2 is currently being developed.

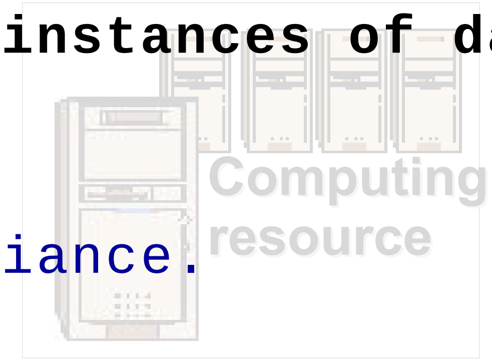


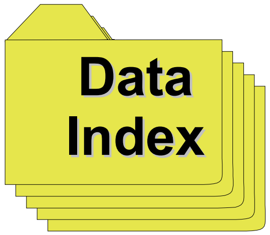
Data Infexing Services - RLS and RC

In distributed environment additional service is needed to track and locate multiple instances of data - Data Indexing Service.

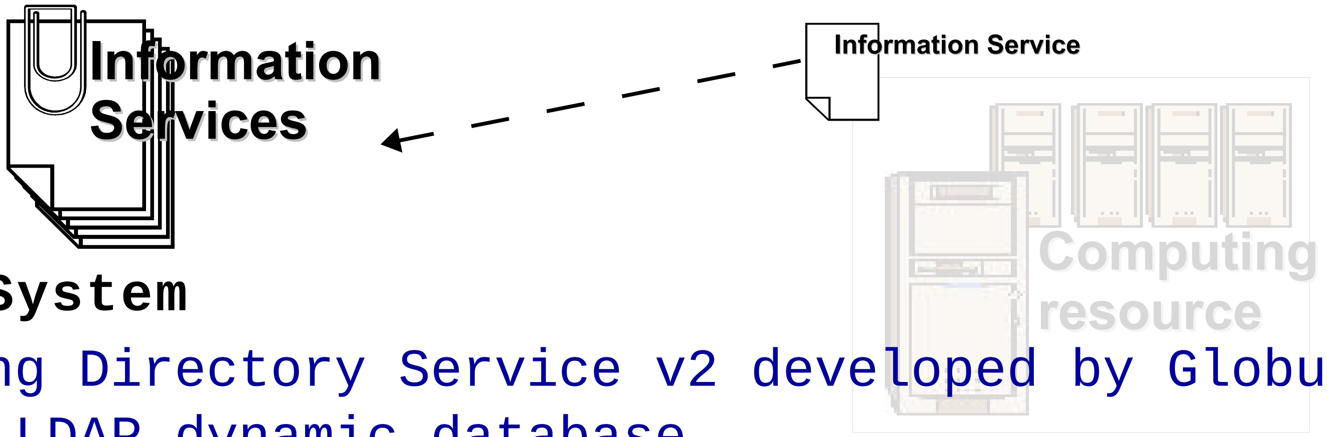
- Replica Catalog
  - Obsolete development of Globus Alliance.
  - Still used for compatibility.
- Replica Location Service
  - Integrated security
  - Distributed (hence scaling) architecture
  - Still not very flexible and requires supplemental services to provide full required functionality

New flexible Indexing Service is currently under consideration.



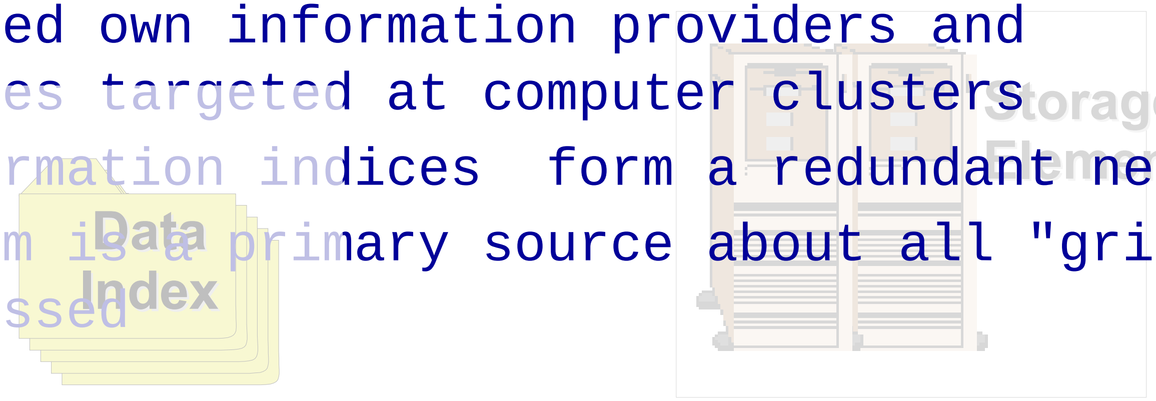


Information System - MDS2

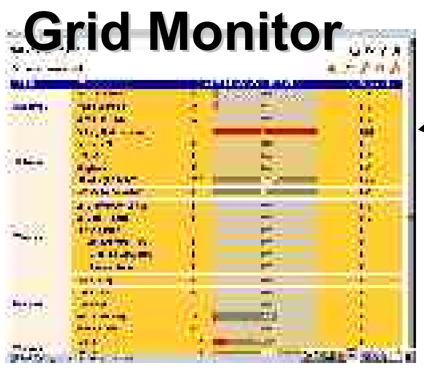


- **Information System**
  - Metacomputing Directory Service v2 developed by Globus Alliance is LDAP dynamic database
  - Collects information about available resources and passes it to clients.
  - Information providers run on each resource.
  - Information indices are used to register resources

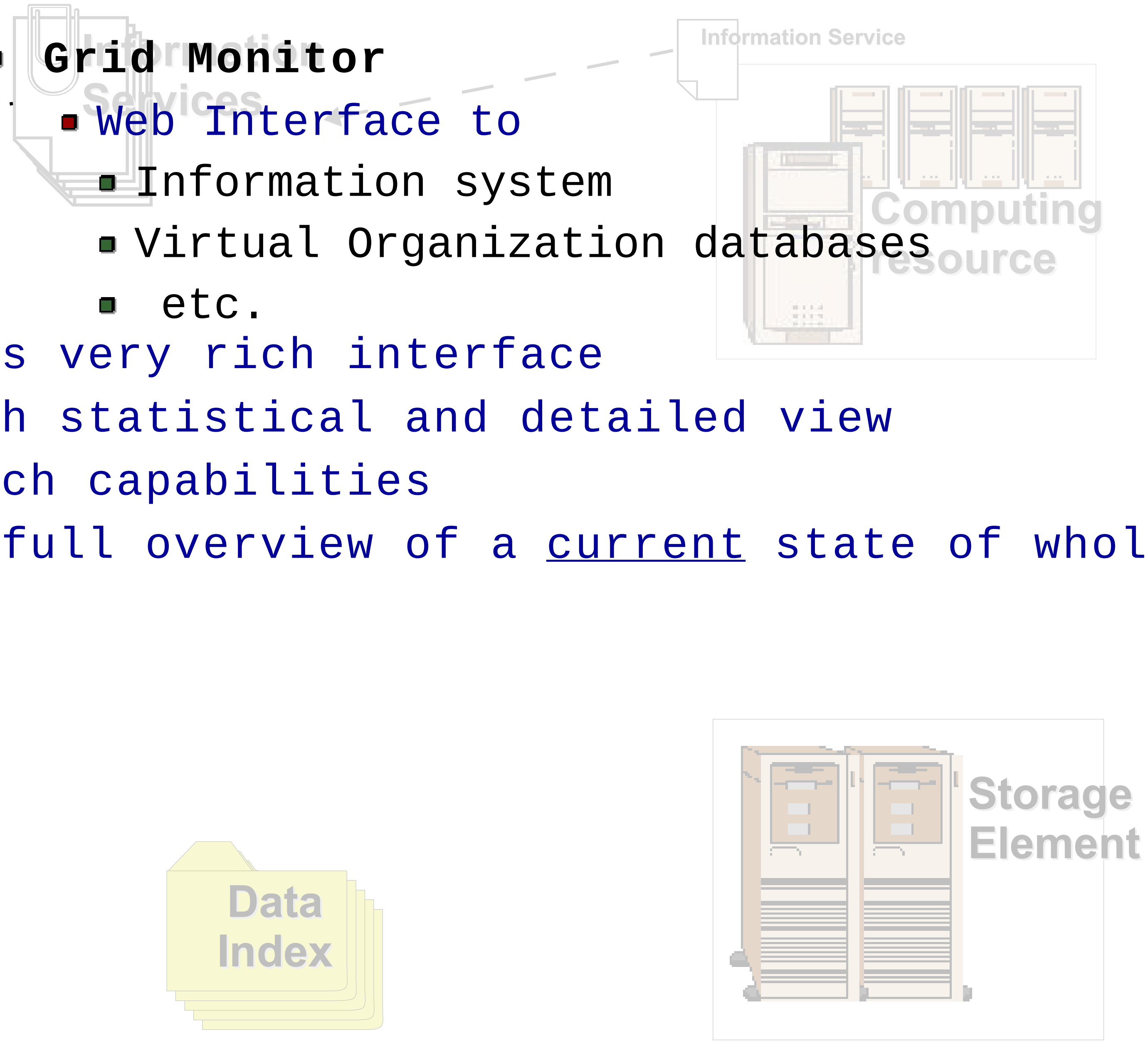
  - NorduGrid developed own information providers and information schemes targeted at computer clusters
  - Structure of information indices form a redundant net.
  - Information System is a primary source about all "grid jobs" being processed



Information System - Grid Monitor

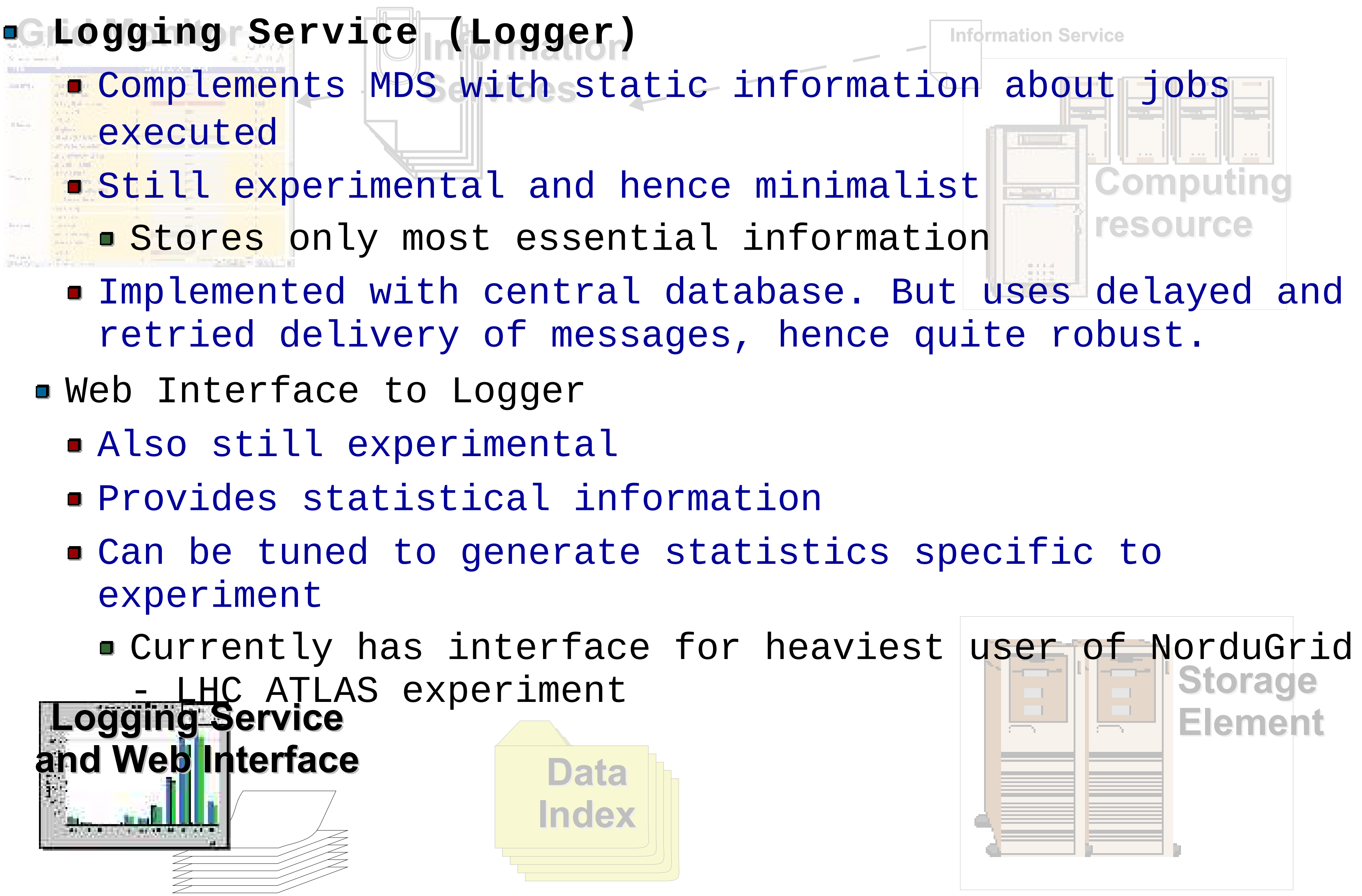


- **Grid Monitor**
  - Web Interface to
    - Information system
    - Virtual Organization databases
    - etc.
- Implements very rich interface
- Gives both statistical and detailed view
- Have search capabilities
- Provides full overview of a current state of whole system



Information System - Logger

- Logging Service (Logger)
  - Complements MDS with static information about jobs executed
  - Still experimental and hence minimalist
    - Stores only most essential information
  - Implemented with central database. But uses delayed and retried delivery of messages, hence quite robust.
- Web Interface to Logger
  - Also still experimental
  - Provides statistical information
  - Can be tuned to generate statistics specific to experiment
    - Currently has interface for heaviest user of NorduGrid - LHC ATLAS experiment



Runtime Environments

Due to distributed nature of computing resources user can't expect homogeneity through all computing resources. Few approaches can be taken to overcome this problem:

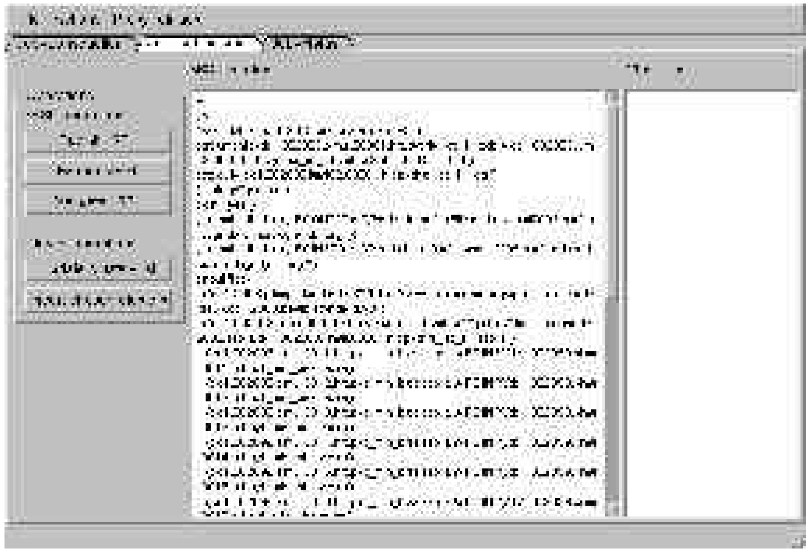
- Provide virtual layer over operating system with unified API. Disadvantage: Every user's application must be rewritten. No way to run legacy applications.
- Run virtual machine for every job. This also gives benefit of very high security. Disadvantage: very heavy solution, especially if architectures do not match.
- Publish most essential attributes of environment available on computing resource. Disadvantage: job can't use all resources, full set of attributes never exists.

ARC uses last approach. Additionally to let users avoid uploading big software packages Runtime Environments are introduced. Each RE defines a set of preinstalled software and a way for application to find it (mostly through environment variables).

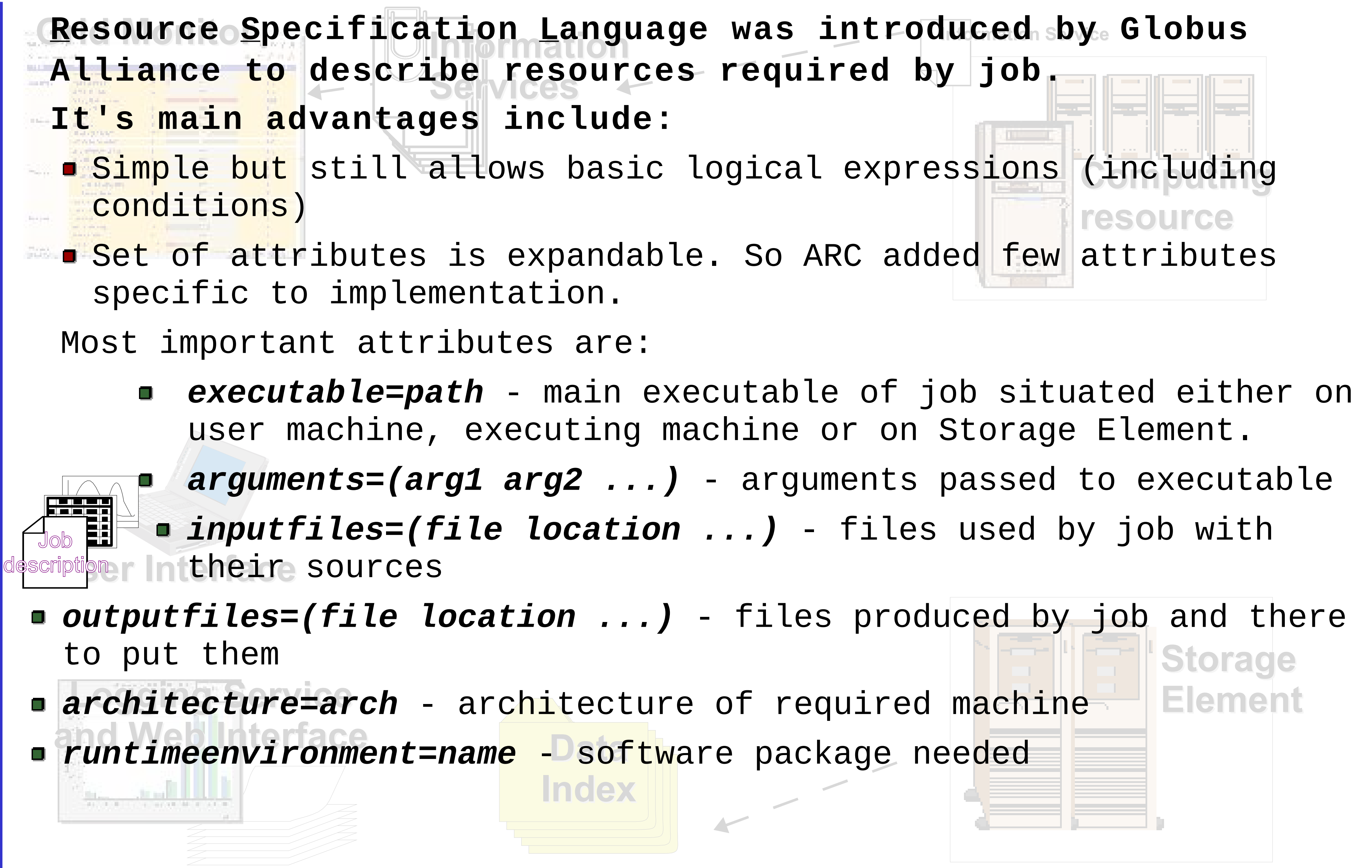
User Interface

- **Command Line User Interface**
  - Set of commands which allow to submit job, control it's execution, obtain results, transfer data.
  - Personal Resource Broker is included in User Interface. It's task is to query information available through defined interfaces and to choose most suitable resources for job execution.
  - So far limited to *NIX systems. Attempts to port it to Windows platform so far unsuccessful.

- **Graphical User interface**
  - Currently few implementations being developed. But none reached usability level yet.



Job Description - xRSL



Resource Specification Language was introduced by Globus Alliance to describe resources required by job.

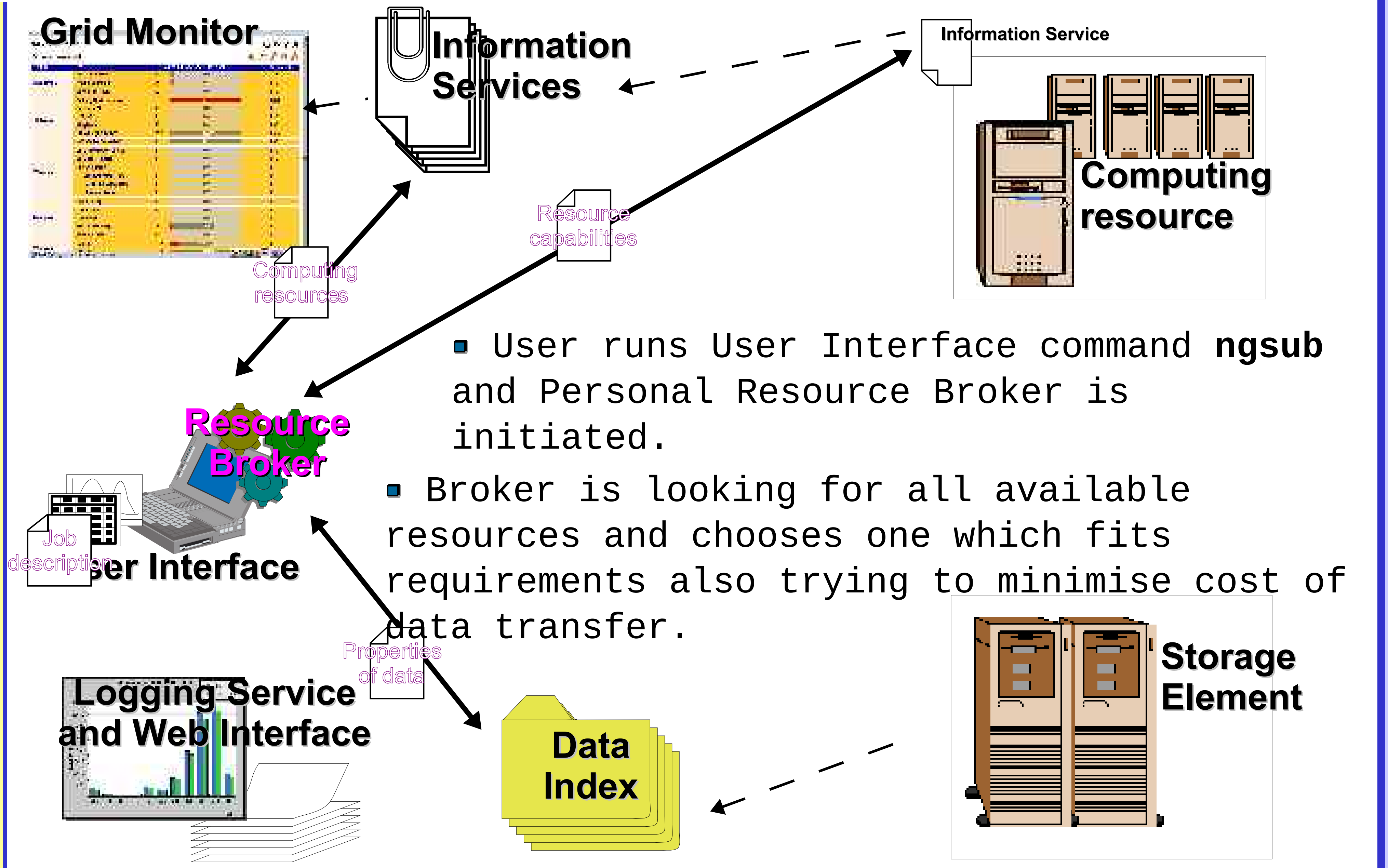
It's main advantages include:

- Simple but still allows basic logical expressions (including conditions)
- Set of attributes is expandable. So ARC added few attributes specific to implementation.

Most important attributes are:

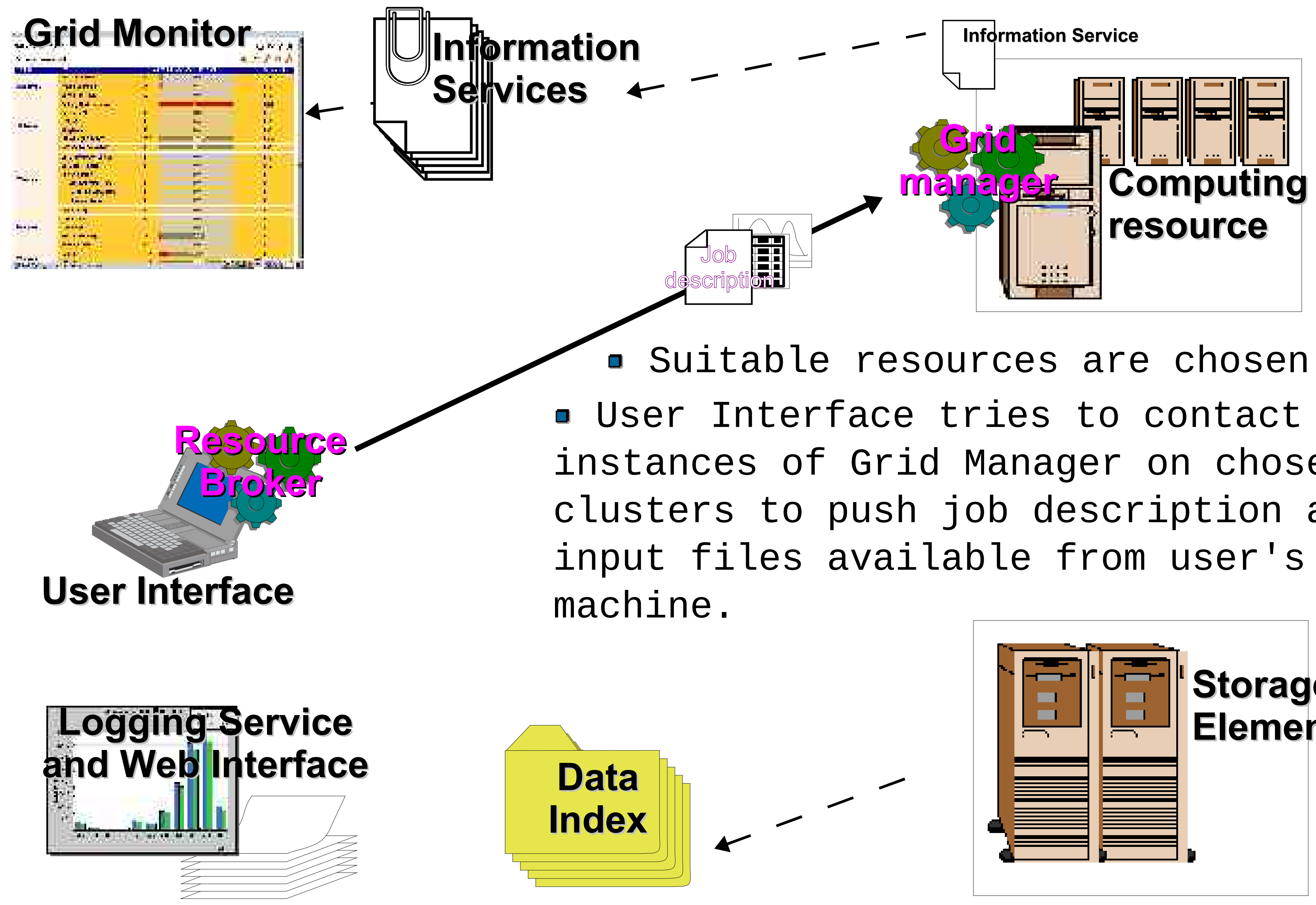
- ***executable=path*** - main executable of job situated either on user machine, executing machine or on Storage Element.
- ***arguments=(arg1 arg2 ...)*** - arguments passed to executable
- ***inputfiles=(file location ...)*** - files used by job with their sources
- ***outputfiles=(file location ...)*** - files produced by job and there to put them
- ***architecture=arch*** - architecture of required machine
- ***runtimeenvironment=name*** - software package needed

Job Flow - Looking for resources



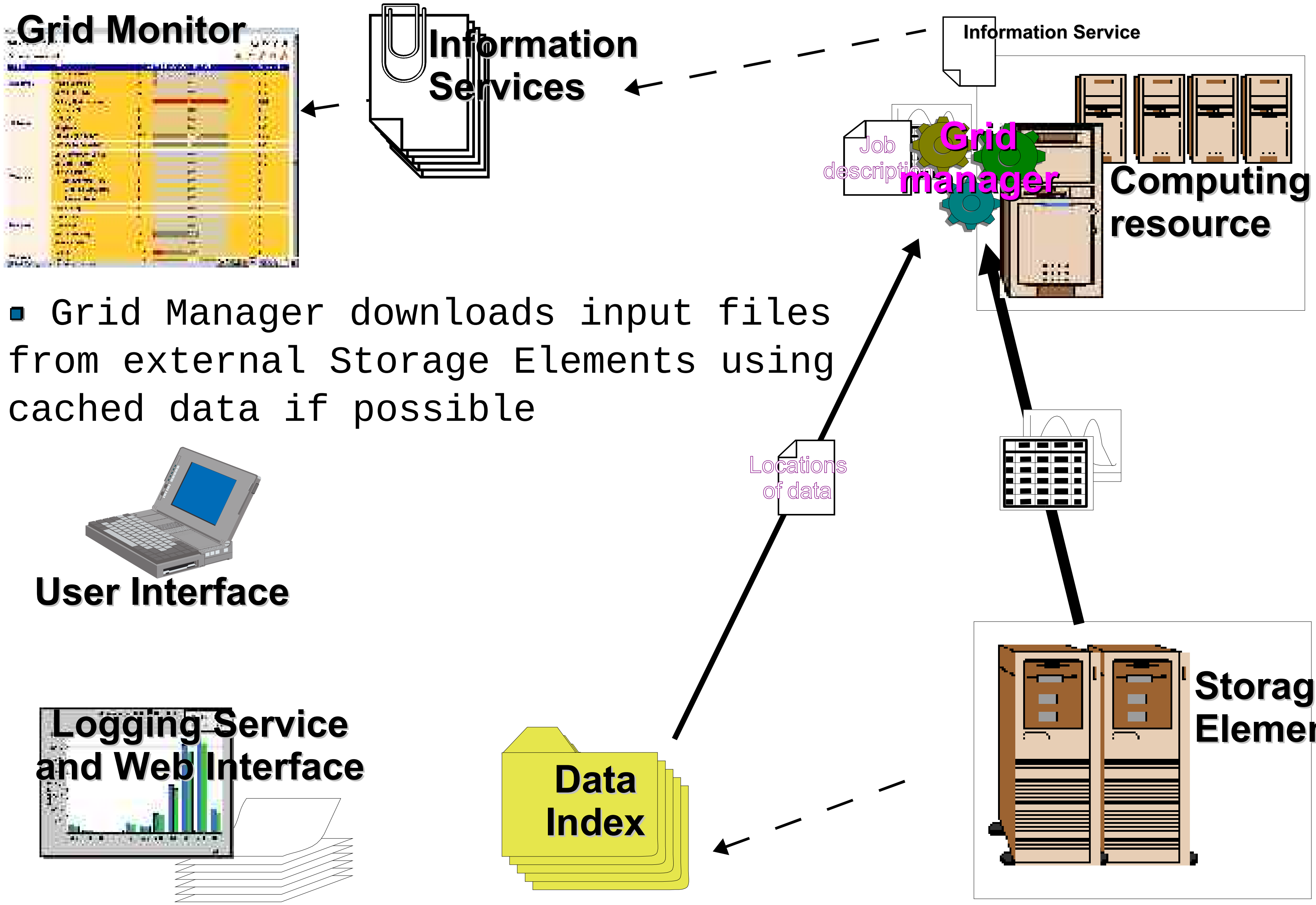
- User runs User Interface command **ngsub** and Personal Resource Broker is initiated.
- Broker is looking for all available resources and chooses one which fits requirements also trying to minimise cost of data transfer.

Job Flow - Initiating job



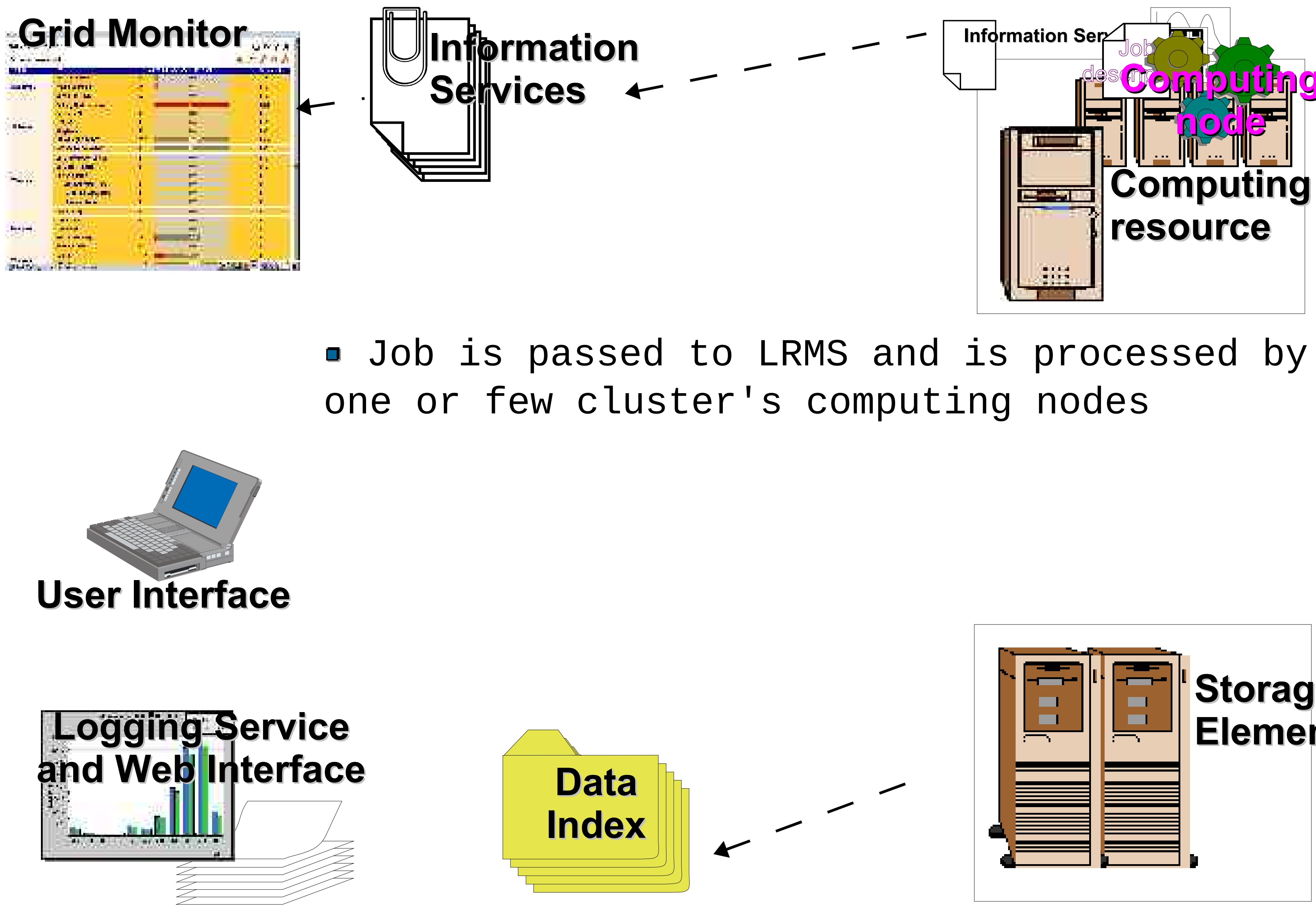
- Suitable resources are chosen
- User Interface tries to contact instances of Grid Manager on chosen clusters to push job description and input files available from user's machine.

Job Flow - Preprocessing job



- Grid Manager downloads input files from external Storage Elements using cached data if possible

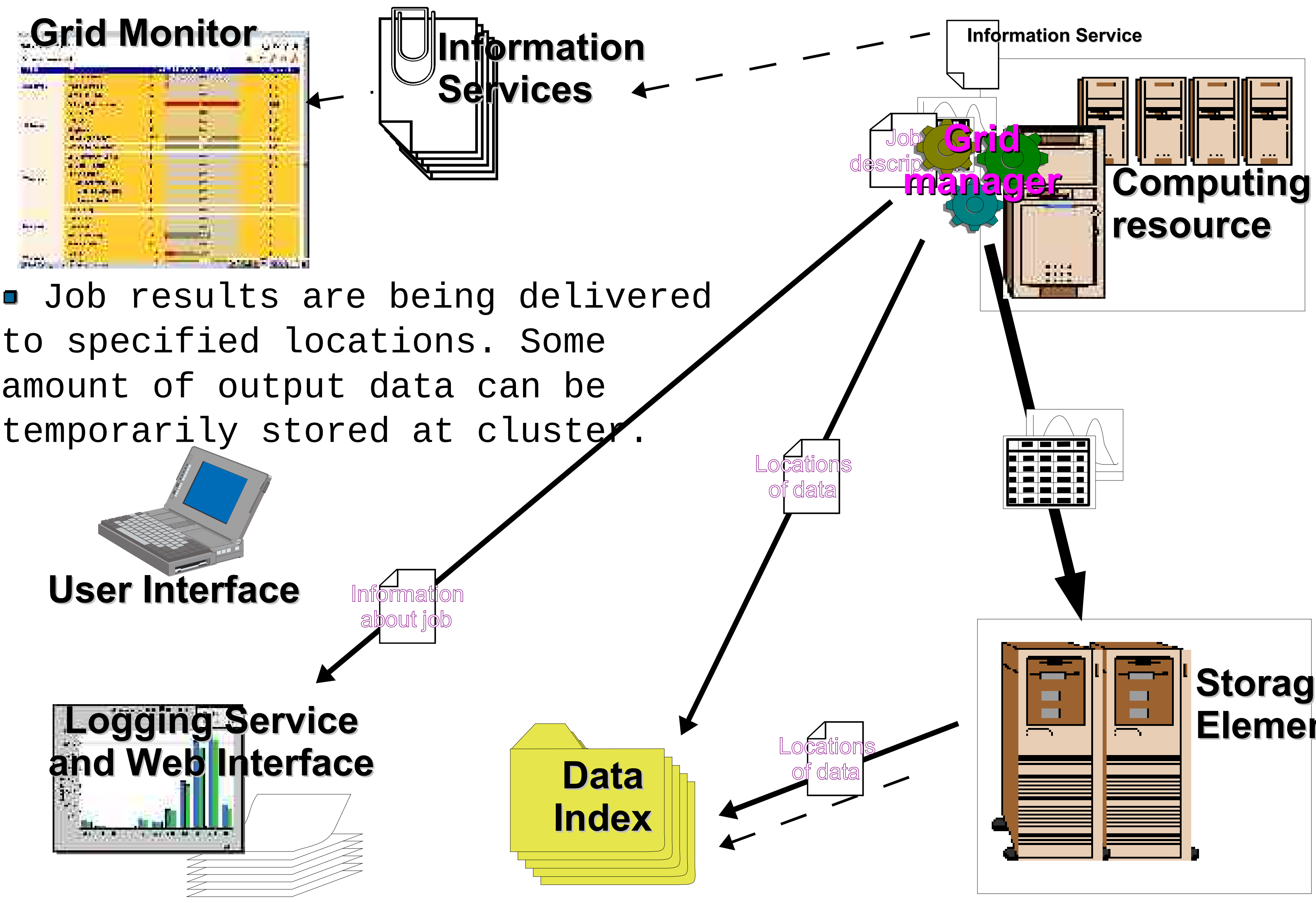
Job Flow - Executing job



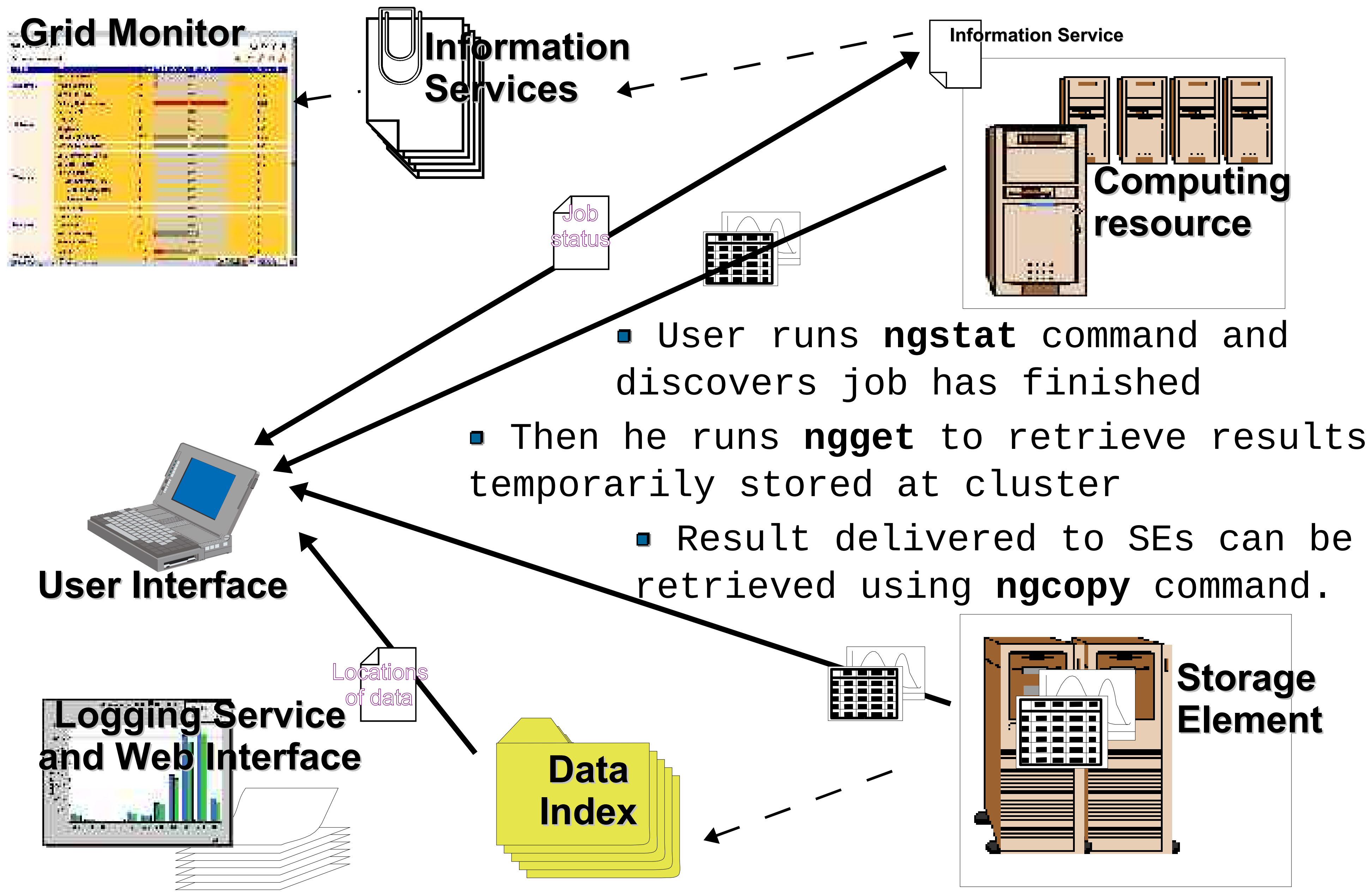
- Job is passed to LRMS and is processed by one or few cluster's computing nodes

Job Flow - Postprocessing job

- Job results are being delivered to specified locations. Some amount of output data can be temporarily stored at cluster.

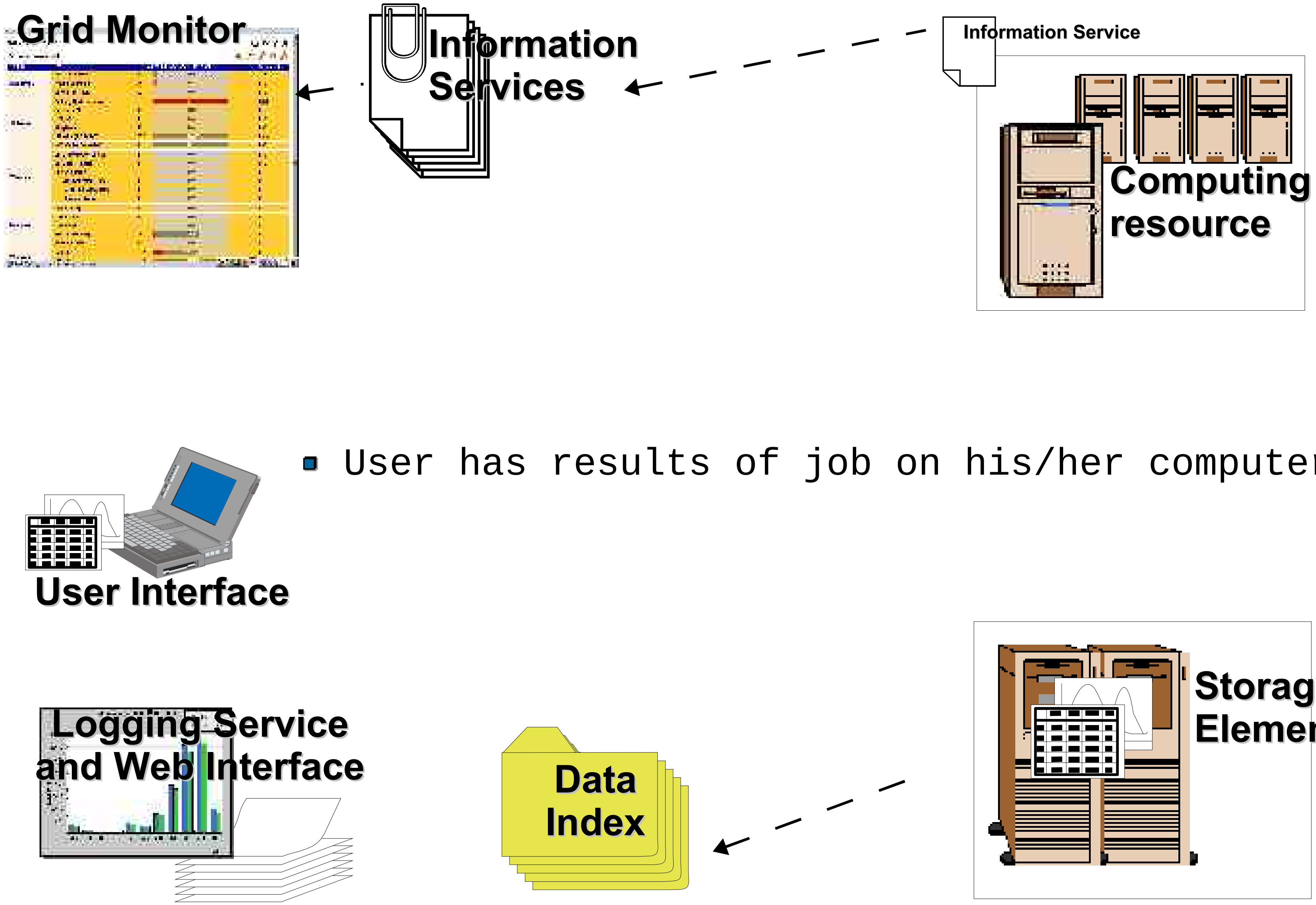


Job Flow - Gathering results



- User runs **ngstat** command and discovers job has finished
- Then he runs **ngget** to retrieve results temporarily stored at cluster
- Result delivered to SEs can be retrieved using **ngcopy** command.

Job Flow



- User has results of job on his/her computer.