Architectural Models for Resource Management in the Grid

Supervisor: PhD. Pham Tran Vu

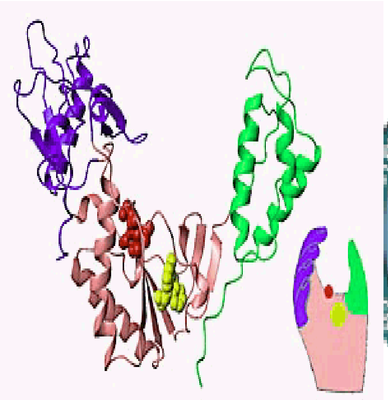
Presenters: Tran Minh Hung
Nguyen Manh Tuan
Nguyen Phan Thien Bach

Outline

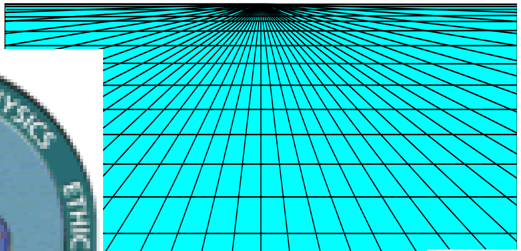
1. ***Introduction***
2. Hierarchical Resource Management
3. Abstract Owner (AO) Model
4. Economy/Market Model
5. Summary

High Performance Computing

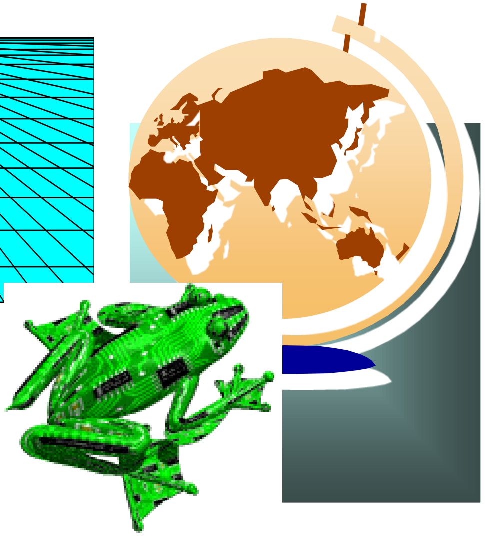
- **Solving grand challenge applications using computer *modeling*, *simulation* and *analysis***

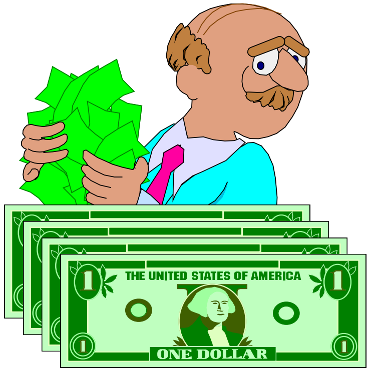


Life Sciences

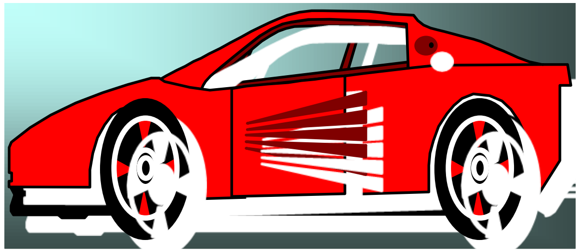


Aerospace

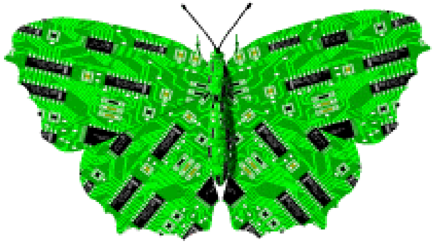




Internet & Ecommerce



CAD/CAM



Digital Biology

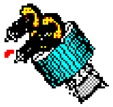
Military Applications

Towards Grid Computing

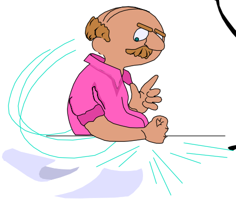
Unification of geographically distributed resources

What is Grid ?

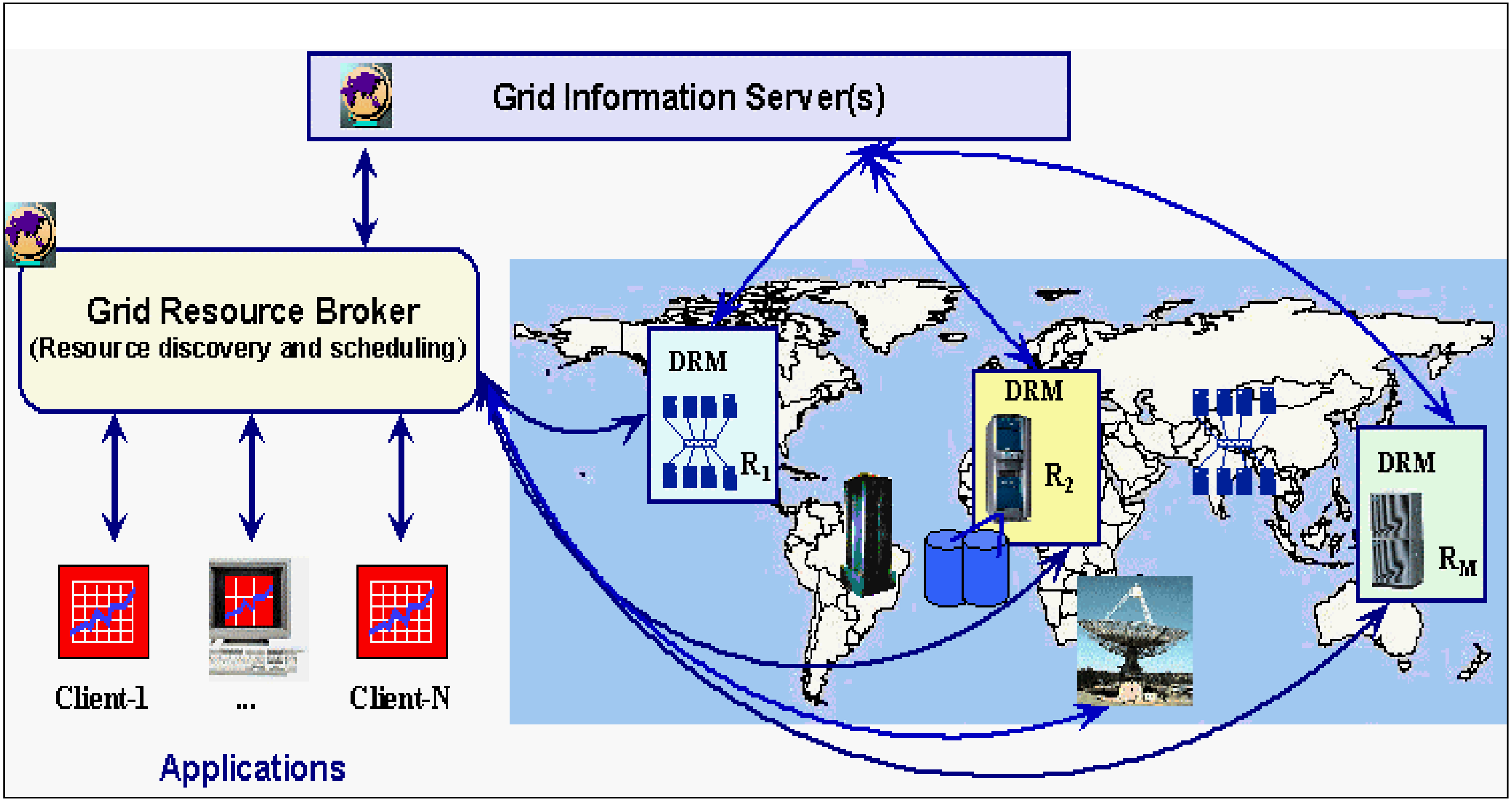
- **An infrastructure that couples**
  - **Computers – PCs, workstations, clusters, supercomputers, laptops, notebooks, mobile devices, PDA, etc**
  - **Software – e.g., ASPs renting expensive special purpose applications on demand**
  - **Catalogued data and databases – e.g. transparent access to human genome database**
  - **Special devices – e.g., radio telescope – SETI@Home searching for life in galaxy, Austrophysics@Swinburne for pulsars)**
  - **People/collaborators**



- **Offers dependable, consistent, pervasive access to resources**



A Example Grid Infrastructure

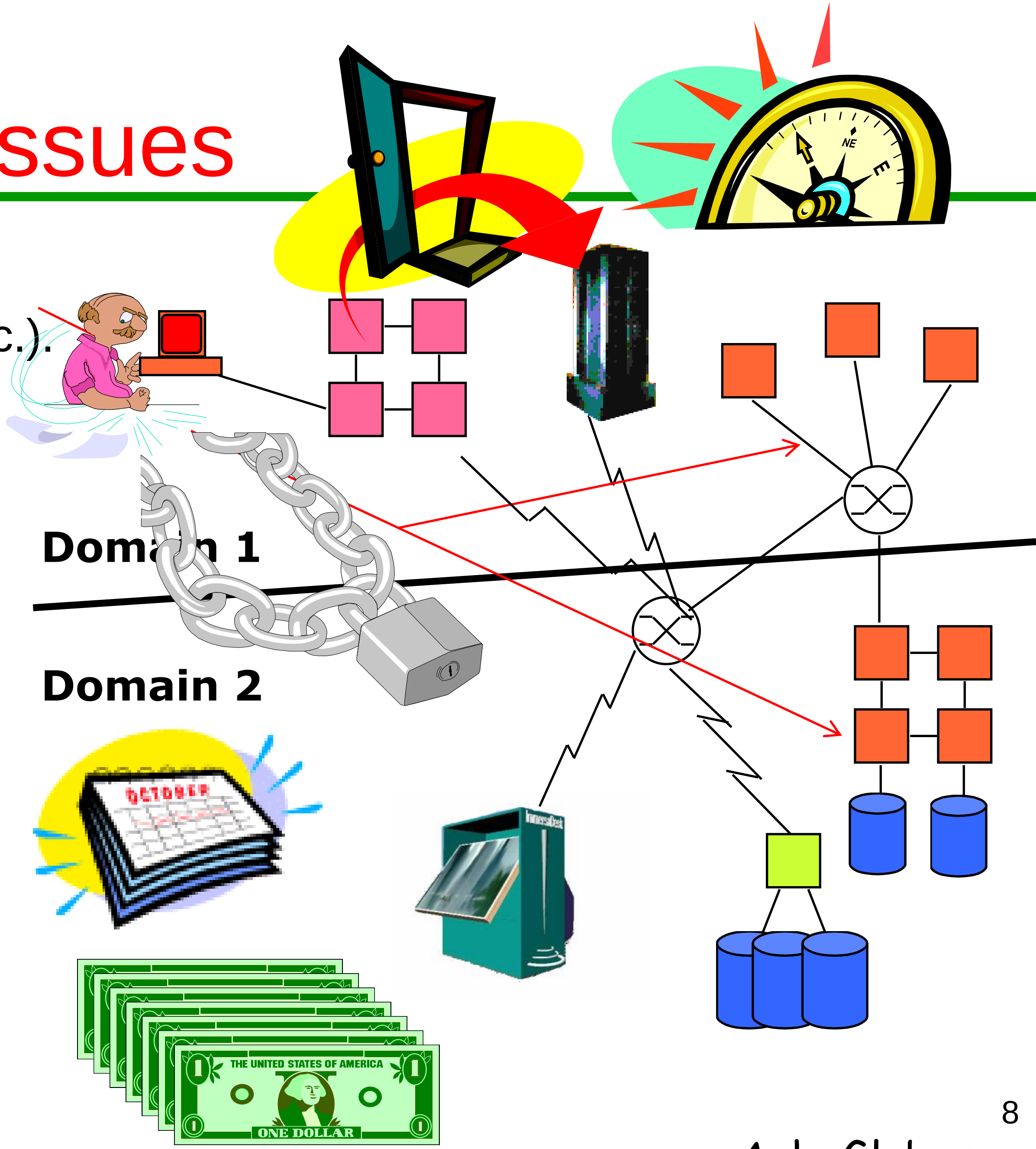


Sources of Complexity in Grid Resource Management

- **No single administrative control.**
- **No single ownership policy:**
  - **Each resource owner has their own policies or scheduling mechanisms;**
  - **Users must honour them (particularly external Grid users).**
- **Heterogeneity of resources.**
- **Dynamic availability – may appear and disappear…**
- **Unreliable resource – disappear from view!**
- **No uniform cost model - varies from one user’s resource to another and from time of day.**
- **No single access mechanism – Web, custom interfaces, command line…**

Grid Resource Management Issues

- Authentication (once).
- Specify (code, resources, etc.).
- Discover resources.
- Negotiate authorization, acceptable use, Cost, etc.
- Acquire resources.
- Schedule Jobs.
- Initiate computation.
- Steer computation.
- Access remote data-sets.
- Collaborate with results.
- Account for usage.



Ack: Globus..

Architectural Models

- ➢ Need to encourage resource owners to contribute their resources, offer a fair basis for sharing resources among users, and regulate resource demand and supply.
- ➢ Influence the way scheduling systems are built as they are responsible for mapping user requests to the right set of resources.
- ➢The grid scheduling systems need to follow multilevel scheduling architecture as
  - each resource has its own scheduling system
  - users schedule their applications on the grid using super-schedulers called resource brokers

Architectural Models

MODEL	REMARKS	Systems
Hierarchical	It captures model followed in most contemporary systems.	Globus, Legion, CCS, Apples, NetSolve, Ninf.
Abstract Owner (AO)	Order and delivery model and focuses on long term goals.	Expected to emerge and most peer-2-peer computing systems likely to be based on this.
Market Model	It follows economic model for resource discover, sharing, & scheduling.	GRACE, Nimrod/G, JavaMarket, Mariposa.

Outline

1. Introduction
2. ***Hierarchical Resource Management***
3. Abstract Owner (AO) Model
4. Economy/Market Model
5. Summary

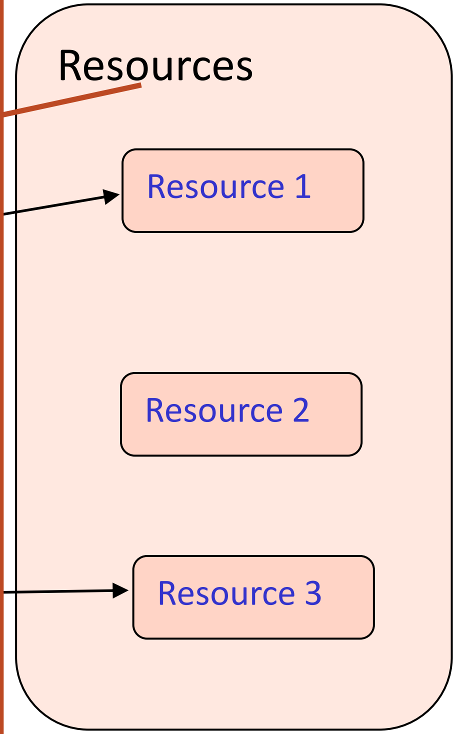
Hierarchical Resource Management

- **Passive components**
  - Resources
  - Tasks
  - Jobs
  - Schedules

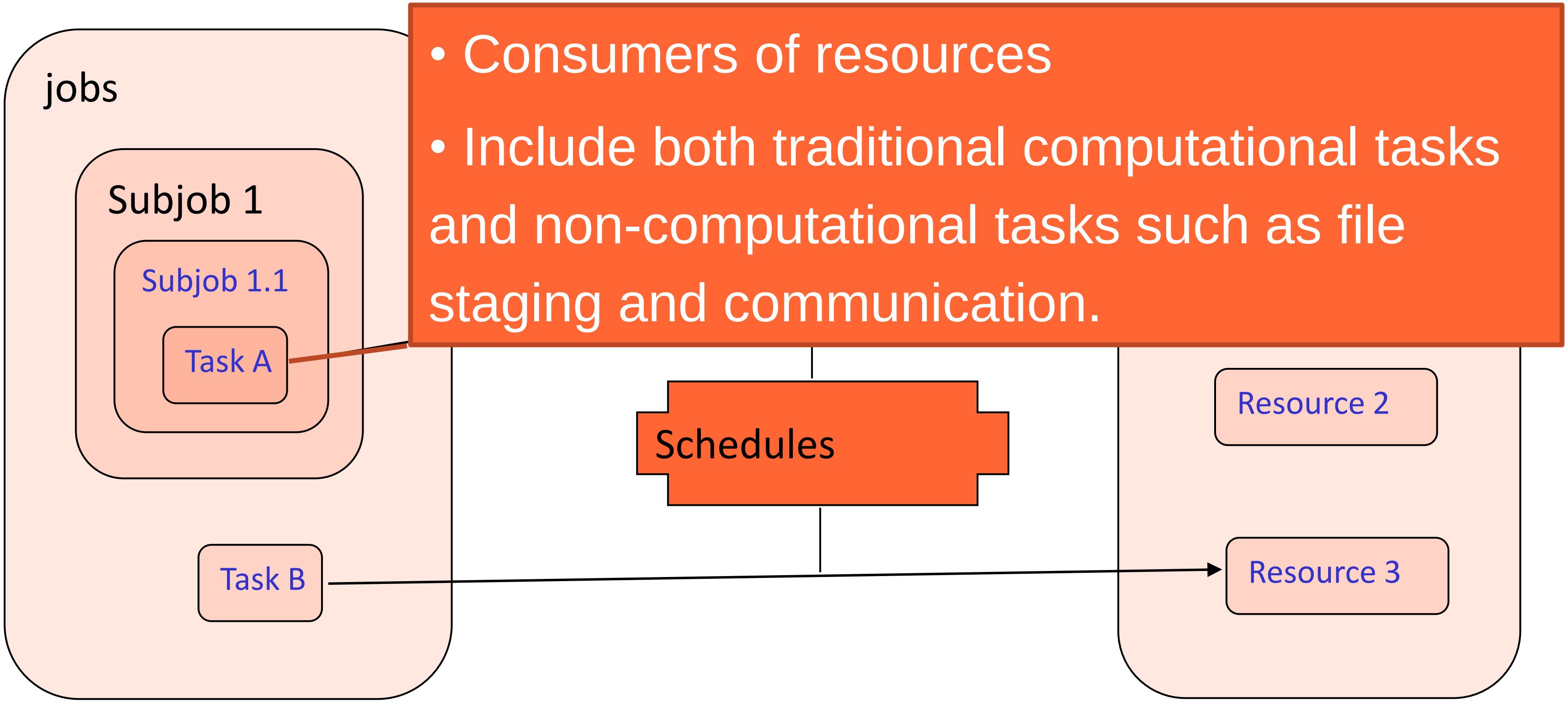
- **Active components**
  - Schedulers
  - Information services
  - Domain control agents
  - Deployment agents
  - Users
  - Admission control agents
  - Monitors
  - Job control agents

Passive Components - Resource

- Things that can be used for a period of time, may or may not be renewable.
- Have owners, who may charge others for using resources
- Can be shared, or exclusive.
- Might be explicitly named, or be described parametrically.
- Examples: disk space, network bandwidth, specialized device time, and CPU time

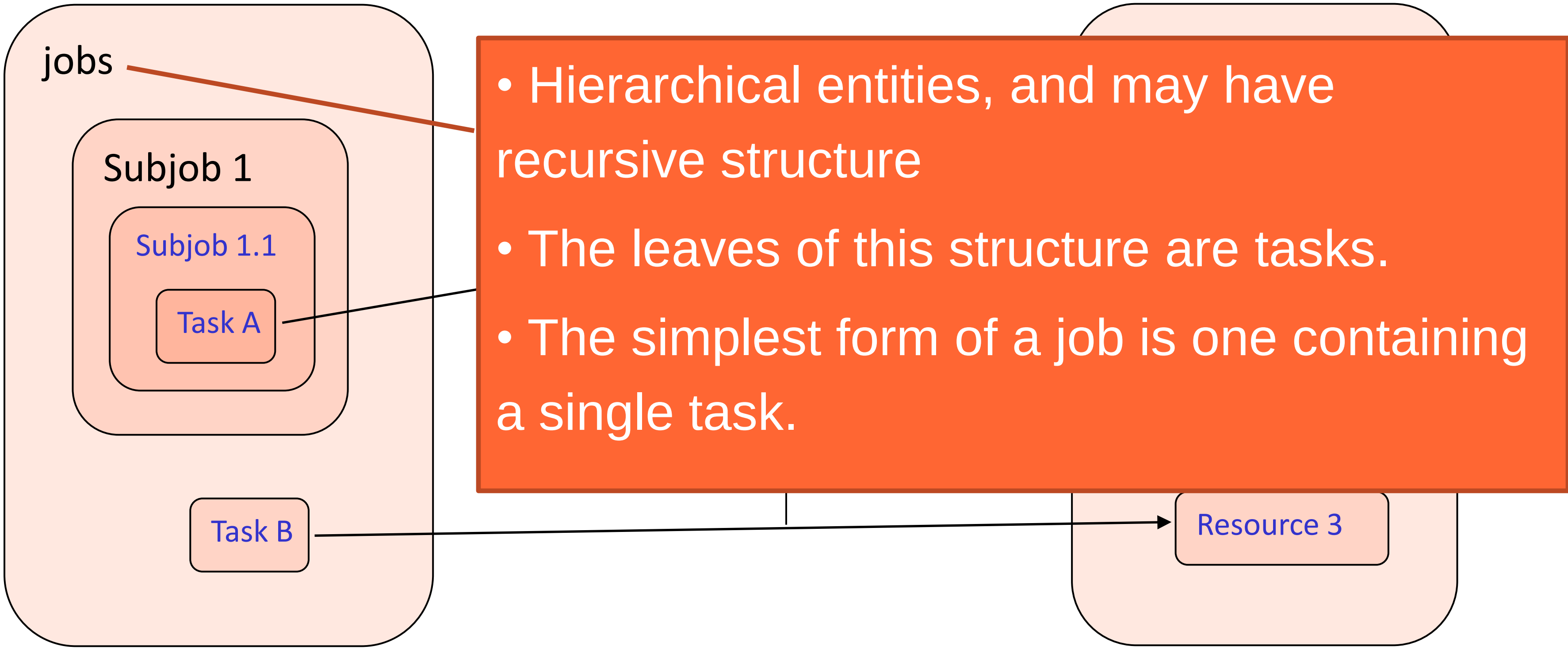


Passive Components - Task



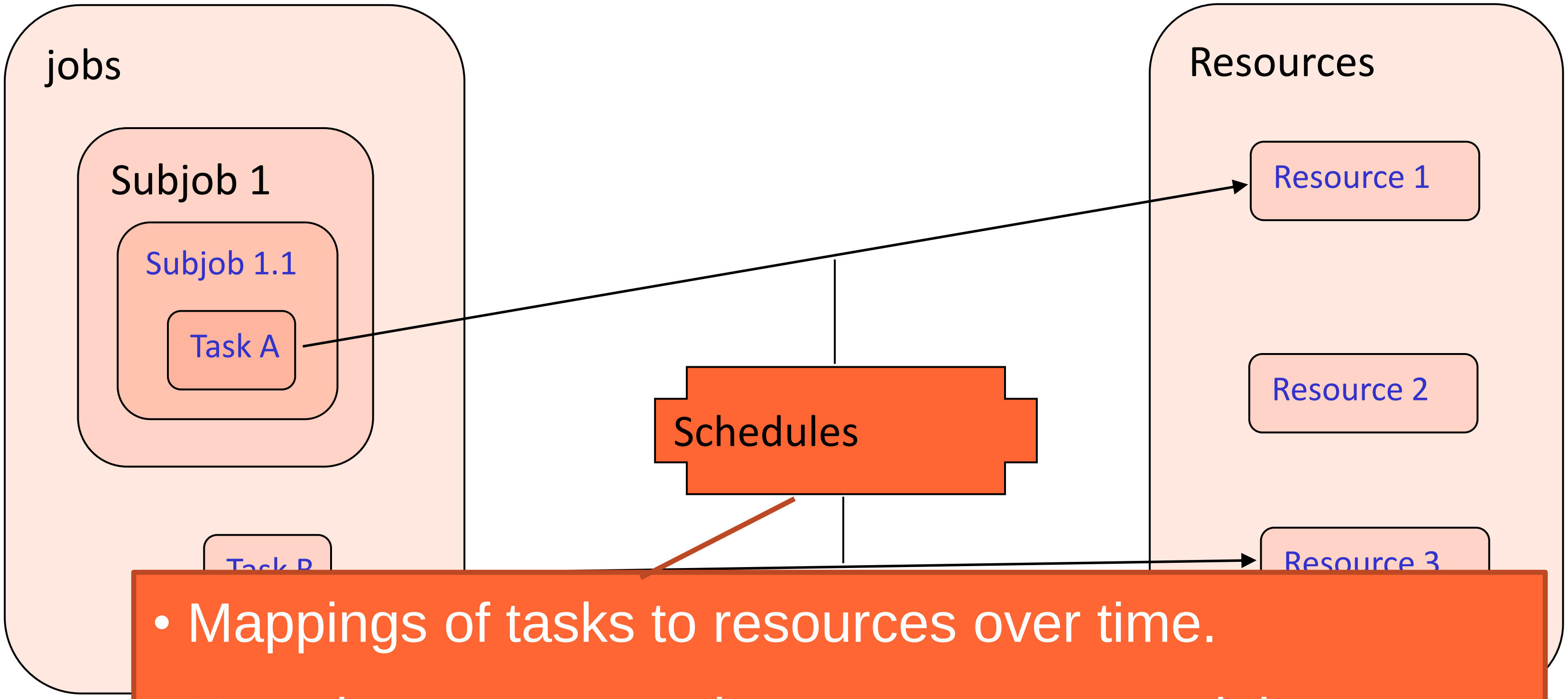
- Consumers of resources
- Include both traditional computational tasks and non-computational tasks such as file staging and communication.

Passive Components - Job



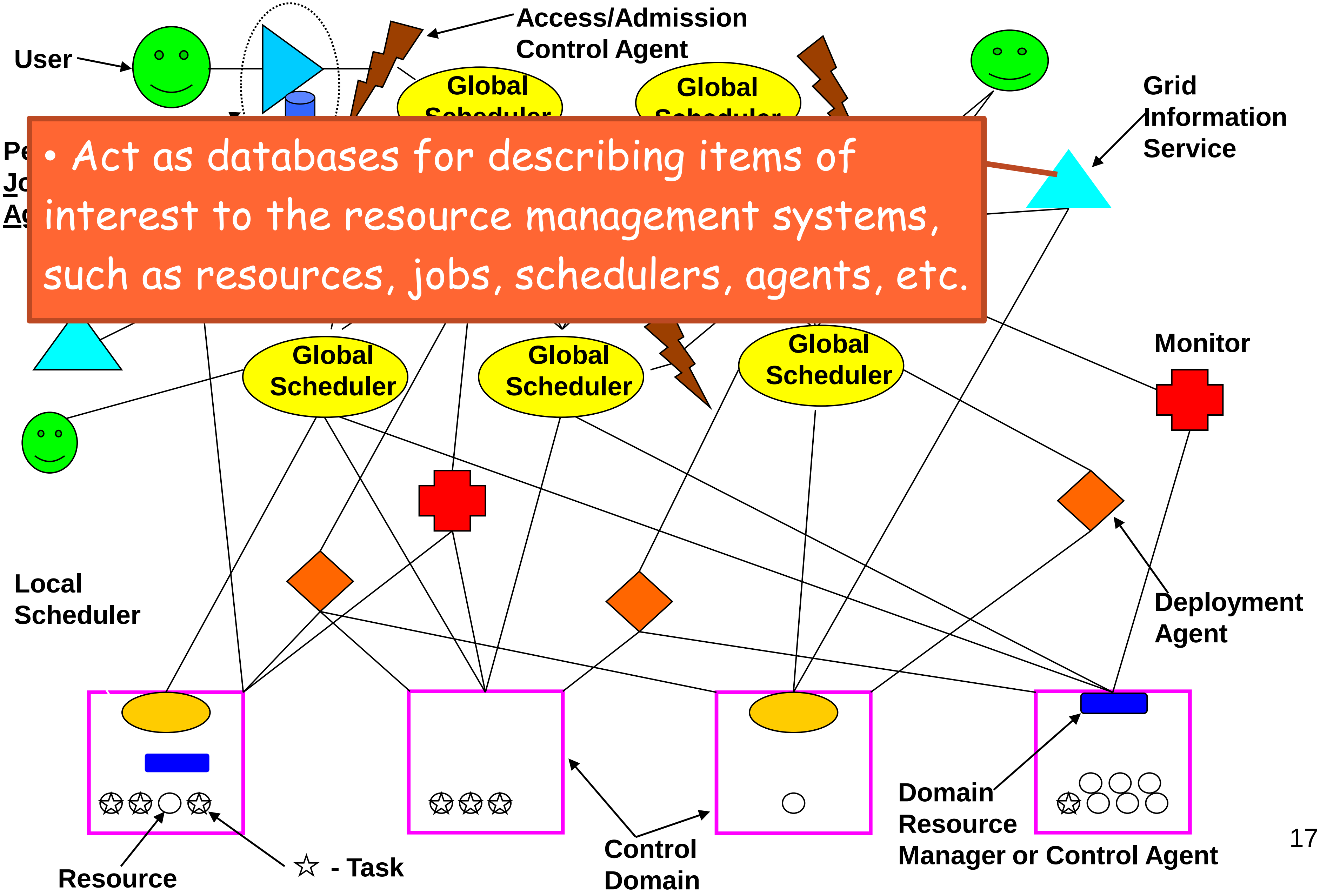
- Hierarchical entities, and may have recursive structure
- The leaves of this structure are tasks.
- The simplest form of a job is one containing a single task.

Passive Components - Schedule



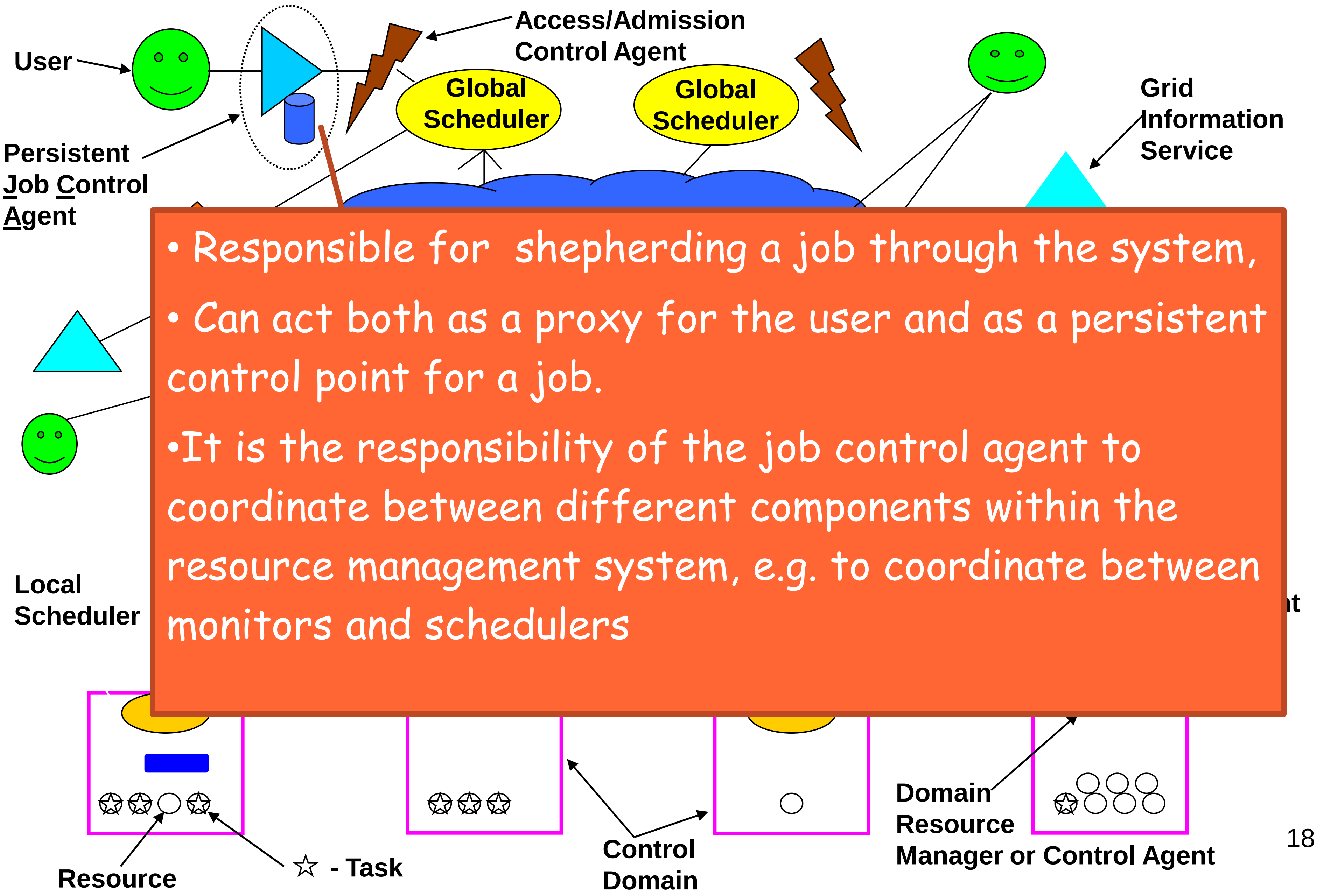
- Mappings of tasks to resources over time.
- Note that we map tasks to resources, not jobs, because jobs are containers for tasks, and tasks are the actual resource consumers.

Active Components – Information Service



- Act as databases for describing items of interest to the resource management systems, such as resources, jobs, schedulers, agents, etc.

Active Components – Job Control Agent



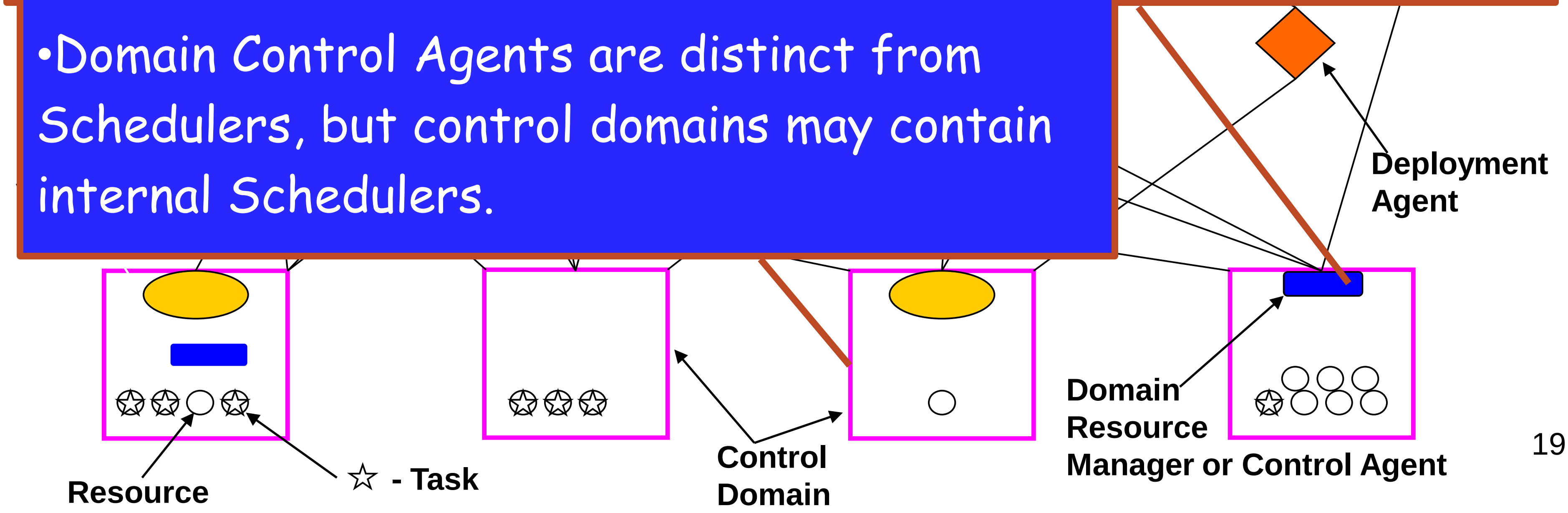
- Responsible for shepherding a job through the system,
- Can act both as a proxy for the user and as a persistent control point for a job.
- It is the responsibility of the job control agent to coordinate between different components within the resource management system, e.g. to coordinate between monitors and schedulers

Active Components – Domain Control Agent

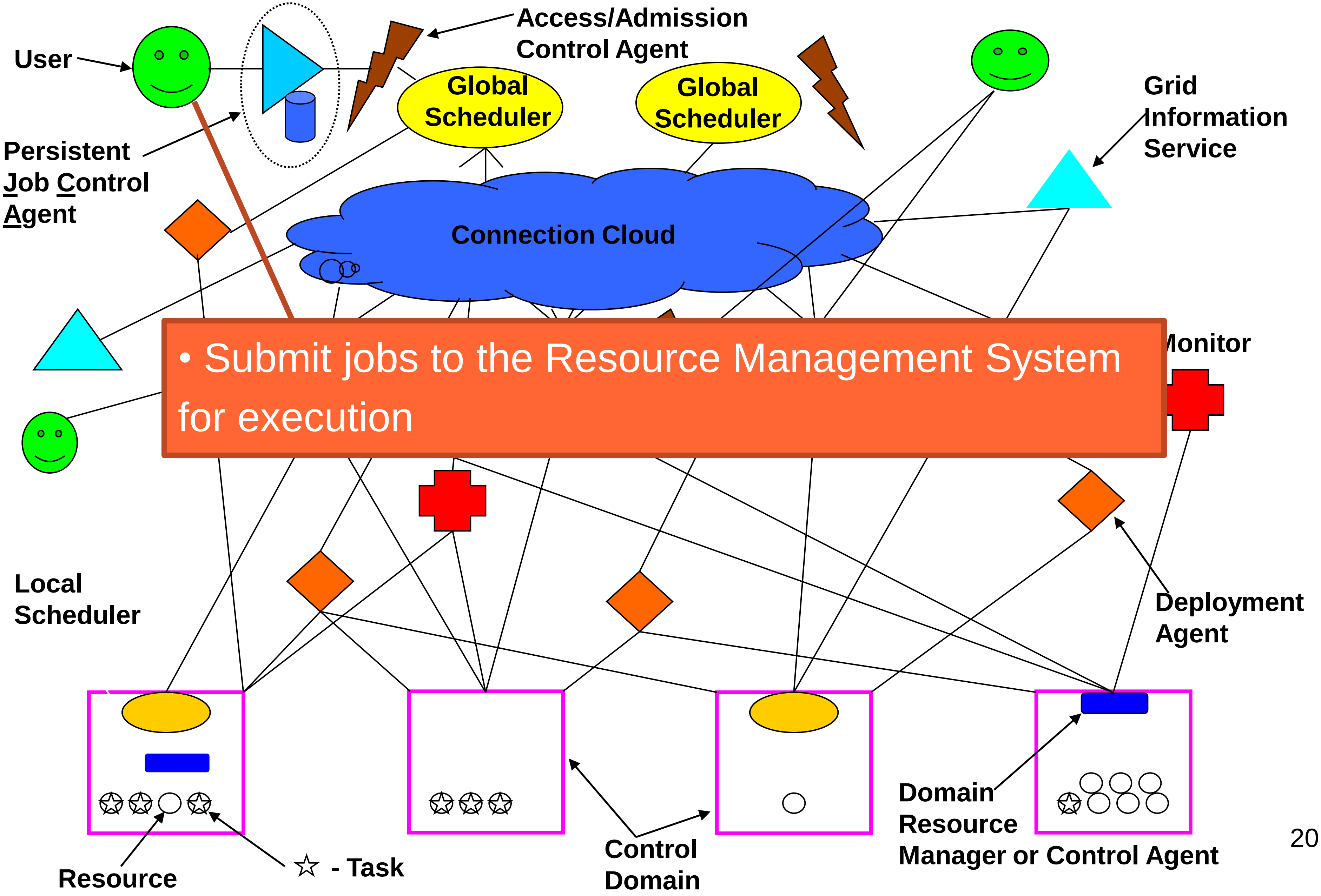
- Commit resources for use; expected to support reservations.
- Provide state information, either through publishing in an Information Service or via direct querying.
- This is what some people mean when they say local resource manager.

Maui Scheduler,

- The set of resources controlled by an agent is a control domain.
- Domain Control Agents are distinct from Schedulers, but control domains may contain internal Schedulers.

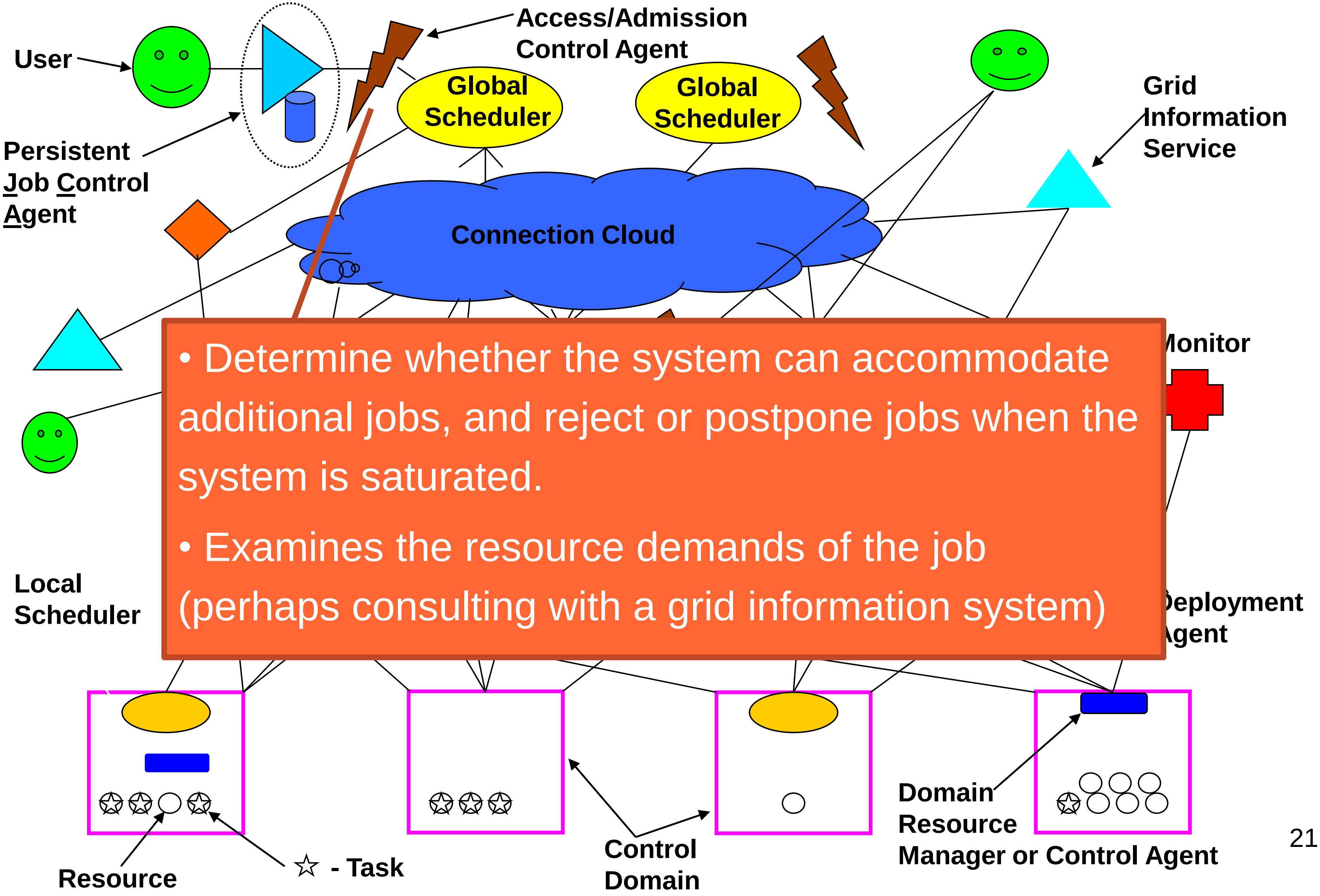


Active Components - User



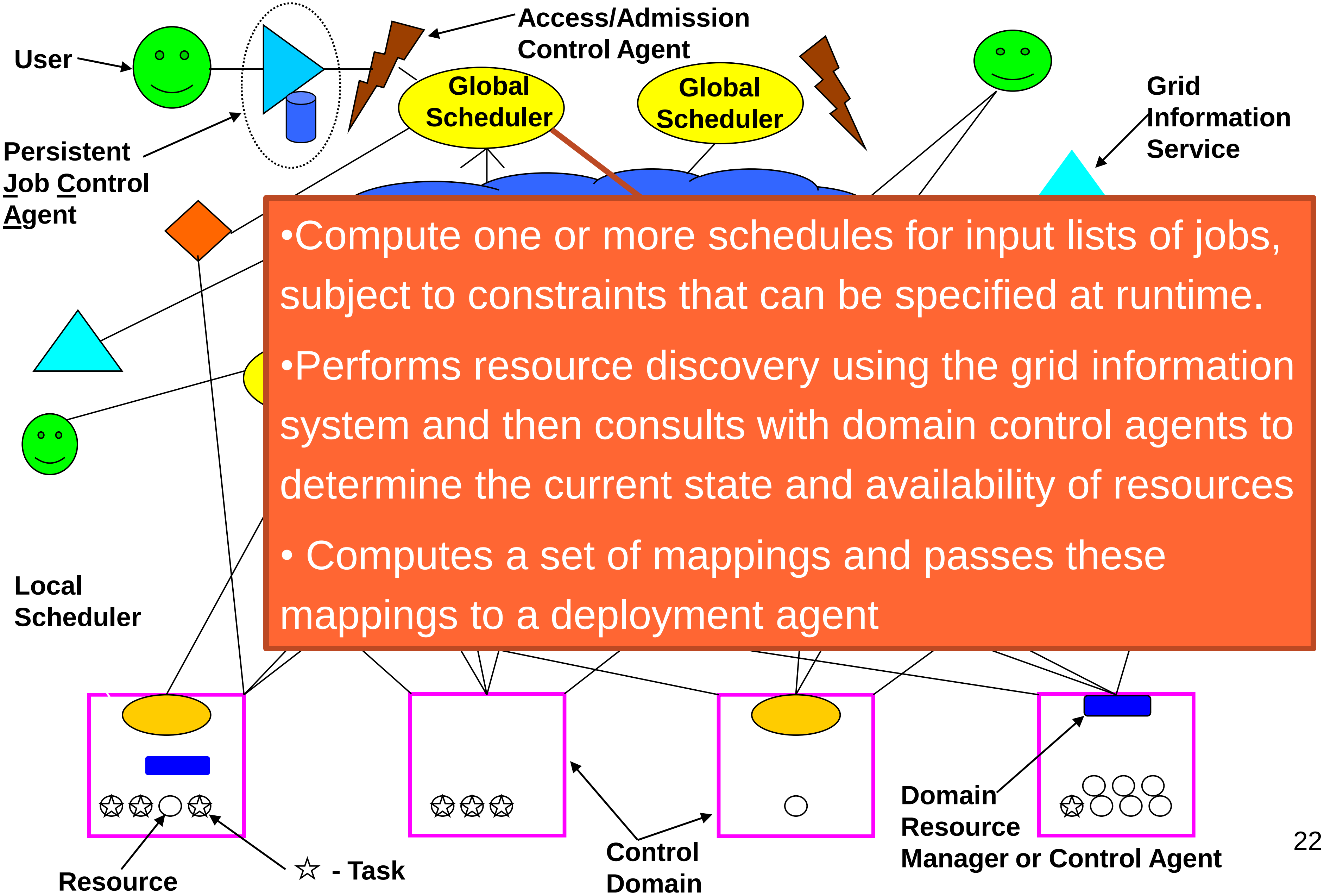
- Submit jobs to the Resource Management System for execution

Active Components – Admission Control Agent



- Determine whether the system can accommodate additional jobs, and reject or postpone jobs when the system is saturated.
- Examines the resource demands of the job (perhaps consulting with a grid information system)

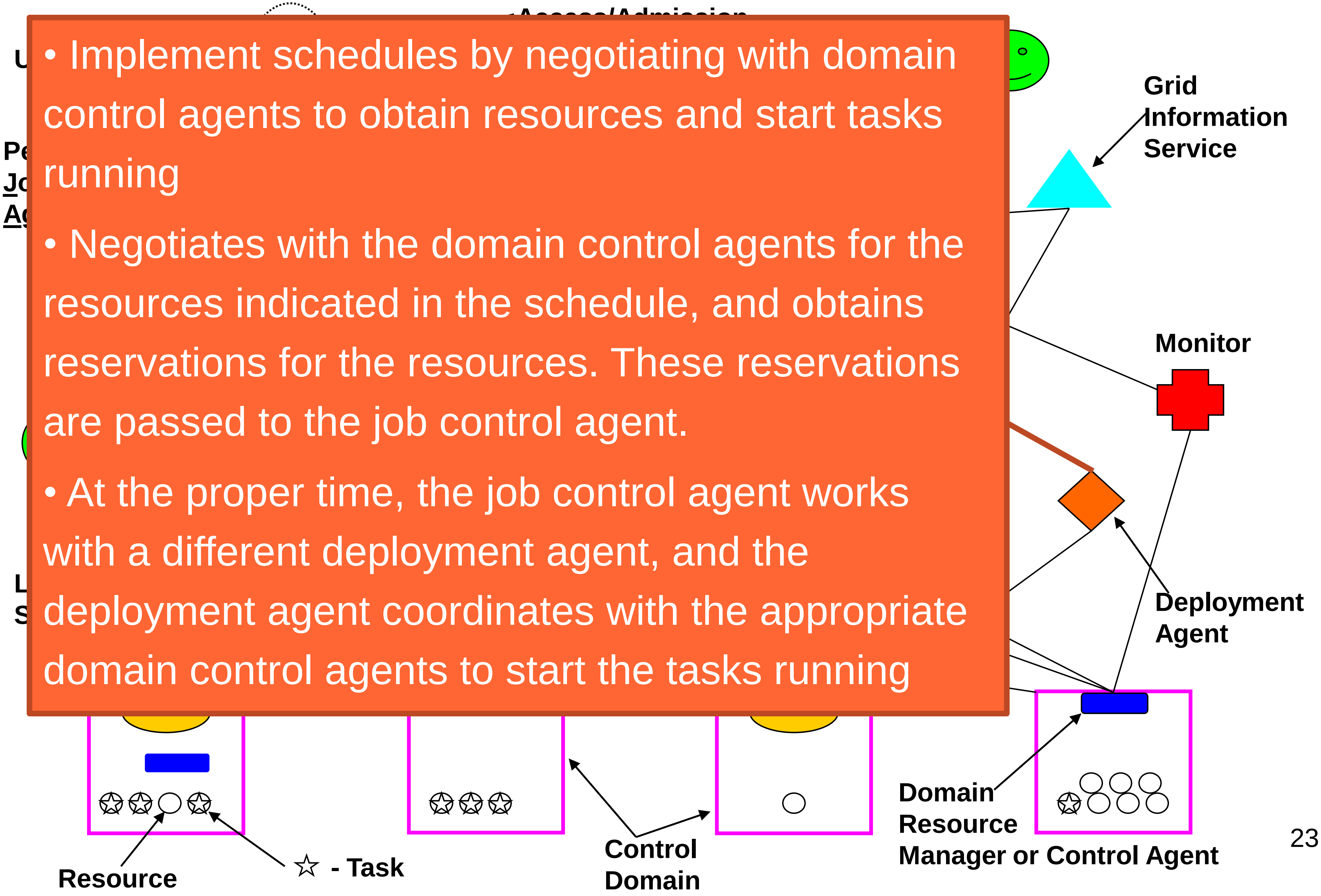
Active Components - Scheduler



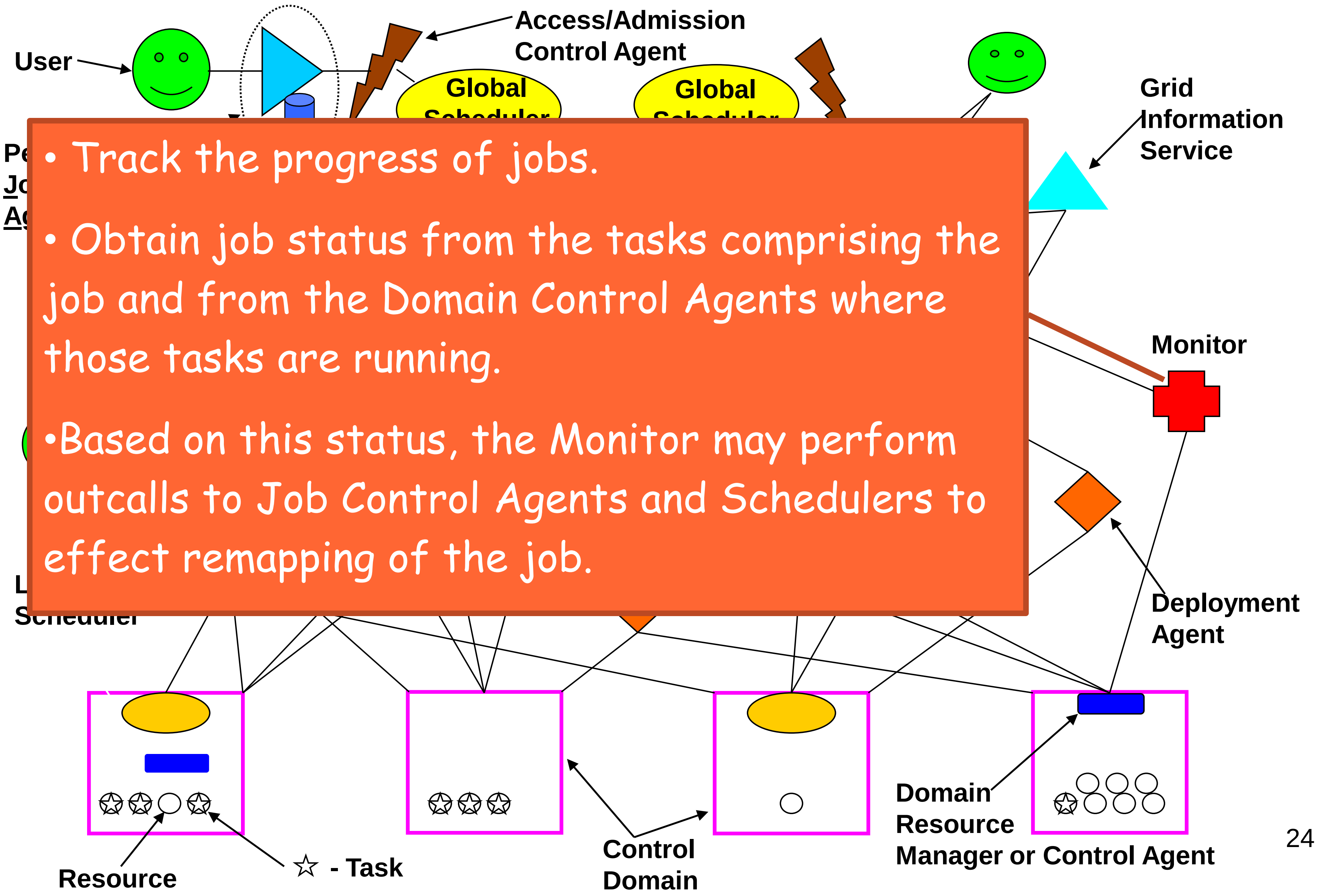
- Compute one or more schedules for input lists of jobs, subject to constraints that can be specified at runtime.
- Performs resource discovery using the grid information system and then consults with domain control agents to determine the current state and availability of resources
- Computes a set of mappings and passes these mappings to a deployment agent

Active Components – Deployment Agent

- Implement schedules by negotiating with domain control agents to obtain resources and start tasks running
- Negotiates with the domain control agents for the resources indicated in the schedule, and obtains reservations for the resources. These reservations are passed to the job control agent.
- At the proper time, the job control agent works with a different deployment agent, and the deployment agent coordinates with the appropriate domain control agents to start the tasks running



Active Components - Monitor



- Track the progress of jobs.

- Obtain job status from the tasks comprising the job and from the Domain Control Agents where those tasks are running.

- Based on this status, the Monitor may perform outcalls to Job Control Agents and Schedulers to effect remapping of the job.

Hierarchical Resource Management

- We have striven to be as general as is feasible in our definitions. Many of these distinctions are logical distinctions.
- For example, we have divided the responsibilities of schedulers, deployment agents, and monitors, although it is entirely reasonable and expected that some scheduling systems may combine two or all three of these in a single program.

Hierarchical Resource Management

- We intentionally referred to control domains as “the box” because it connotes an important separation of “inside the box” vs. “outside the box”. Actions outside the box are requests; actions inside the box may be commands.
- Schedulers outside control domains cannot commit resources; these are known as metaschedulers or super schedulers. Entire grid scheduling systems may exist inside the box.
- Therefore, we can treat the control domain as a black box from the outside

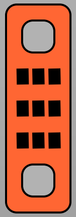
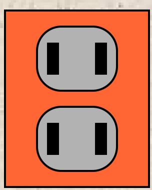
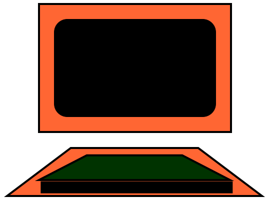
Outline

1. Introduction
2. Hierarchical Resource Management
3. ***Abstract Owner (AO) Model***
4. Economy/Market Model
5. Summary

Abstract Owner (AO) model

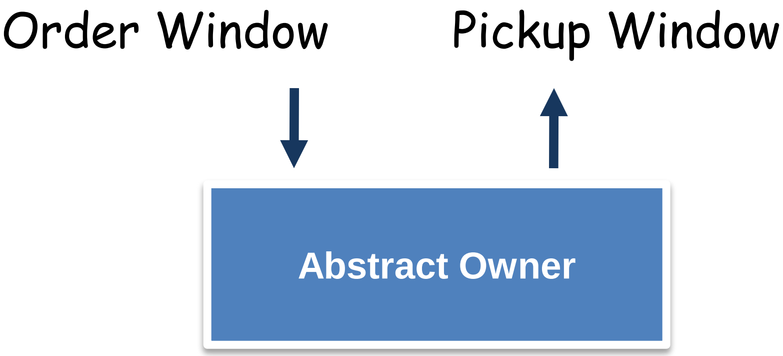
- Introduction
- External view of AO model
- Internal view of AO model
- AO resources
- Negotiating with AO
- Job shops
- AO Summary

Who owns the GRID?

I want to:	Talk to people	Power appliances	Use GRID resources
My interface is:			
I arrange service and payments with a: (may be many choices)	Phone co.	Electric co.	**Abstract Owner (AO)**
But resources I "get" may belong to others:	Antennae Cable/fiber Switches	Generators Power lines Transformers	HPC Networks Instruments People

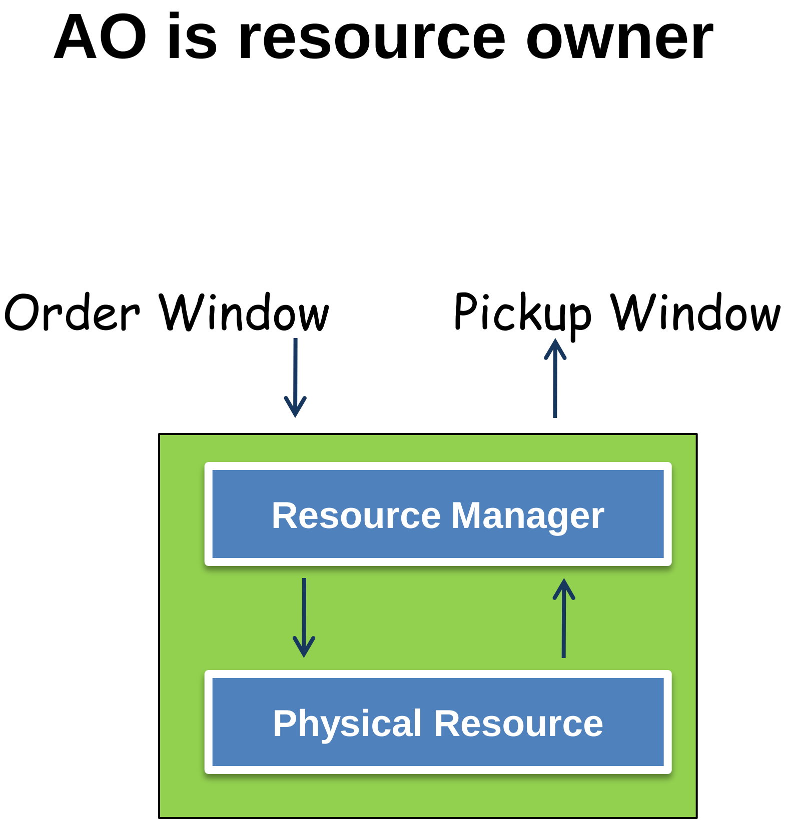
External view of AO model

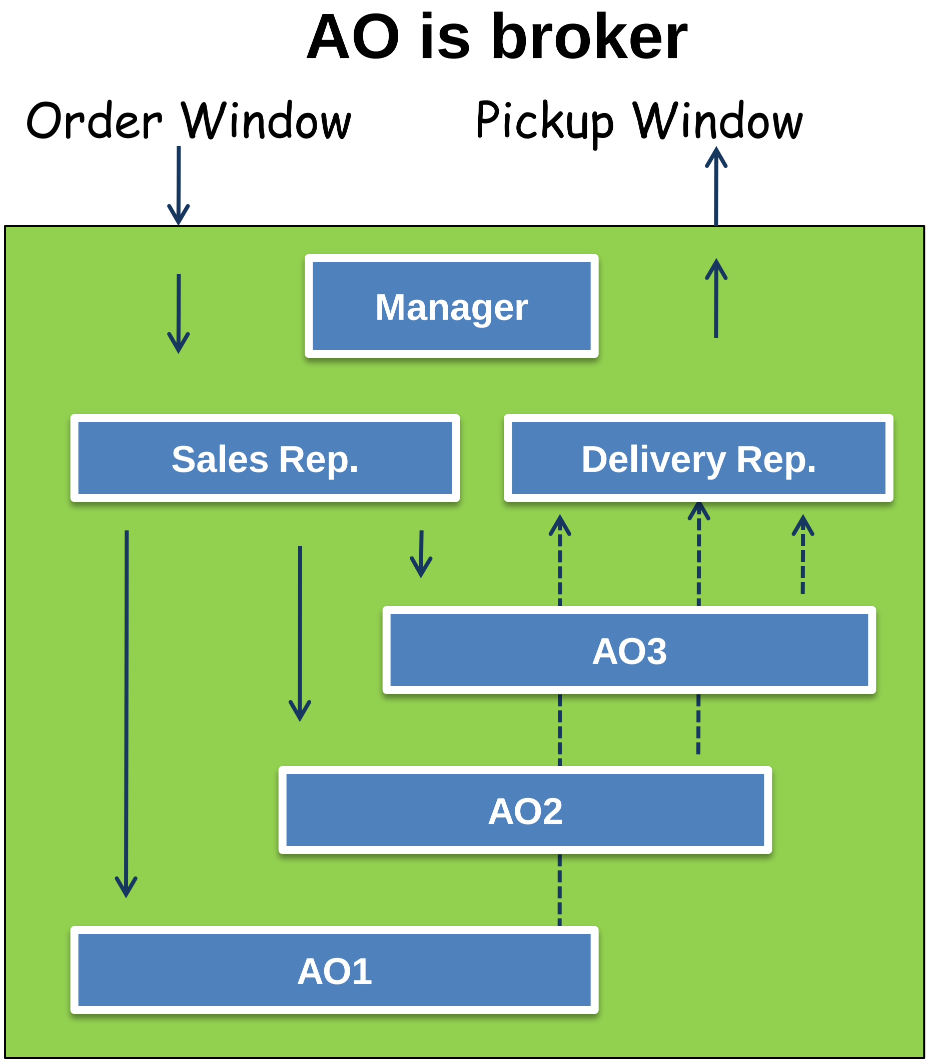
- **AO resembles a fast-food restaurant**
  - Negotiate
  - Order resources
  - Pick up resources



- **Window**
  - Remotely accessible
  - Support a standard procedure-like interface in which values are passed to and returned from the window

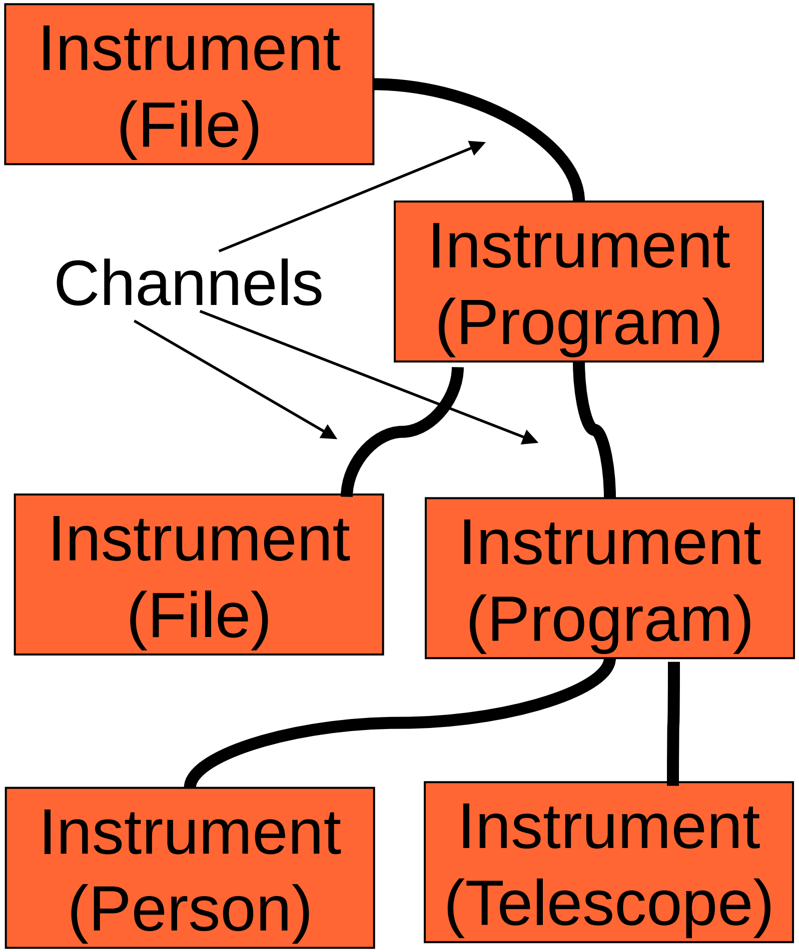
Internal view of AO model





AO Resources

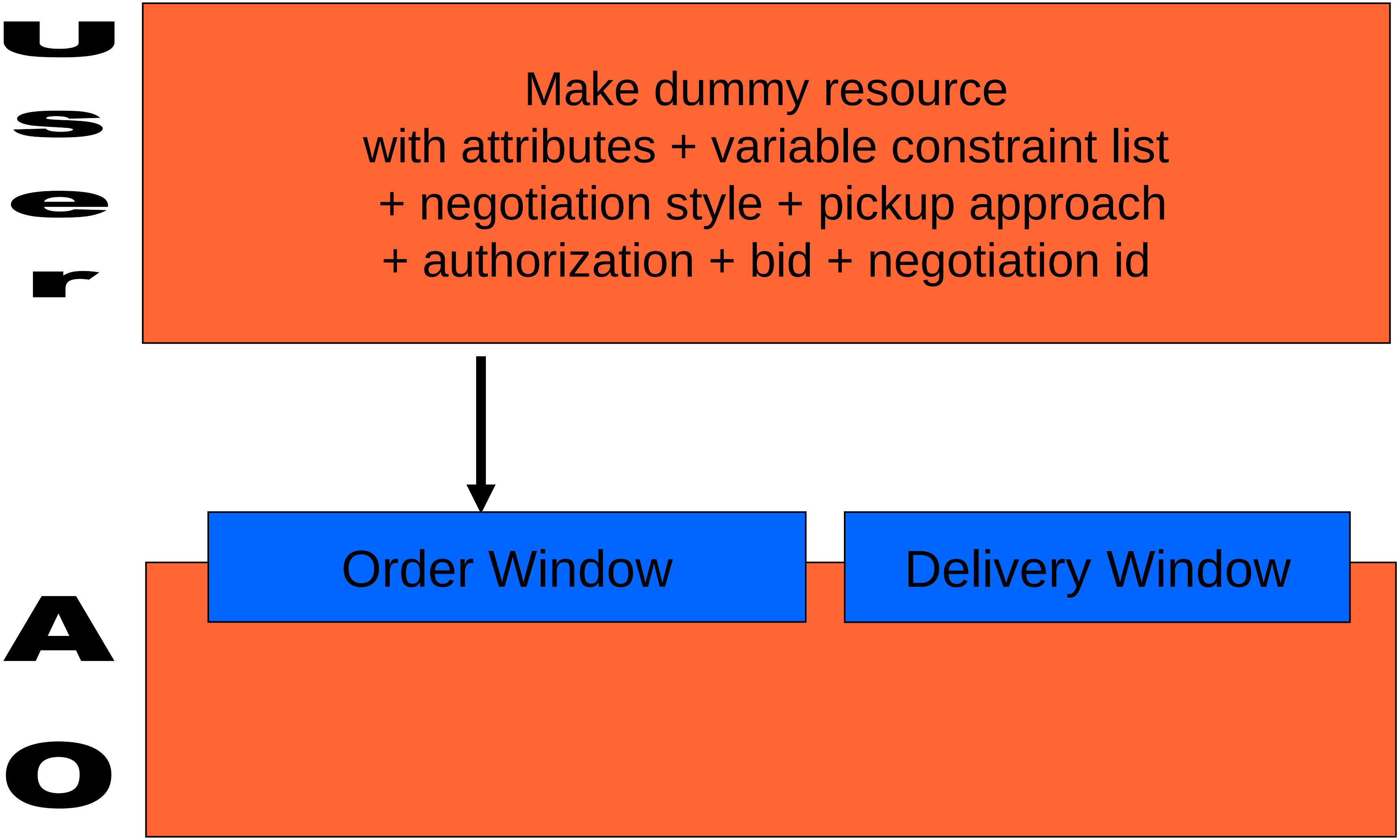
- **Resources are objects**
- **Classes are**
  - **Instrument**
    - » Data source, sink, transform
    - » e.g. programs, people, files, data collection devices
  - **Channel**
    - » Moves data among instruments
  - **Complexes of above**
- **Attributes define sizes, times, connections, etc.**



Instrument class

- **Compute instrument**
  - A processor or set of processors along with associated memory, temp file, software, …
- **Archival instrument**
  - Persistent storage of information
- **Personal instrument**
  - Assumed to interface directly to a human being
  - Ranging from a simple terminal to speech recognition/synthesis device,…
  - Its specification may include the identity of the person involved
- **Other machines and instruments**
  - Telescopes, electron microscopes or any other sink or source for grid data

Negotiating with an AO

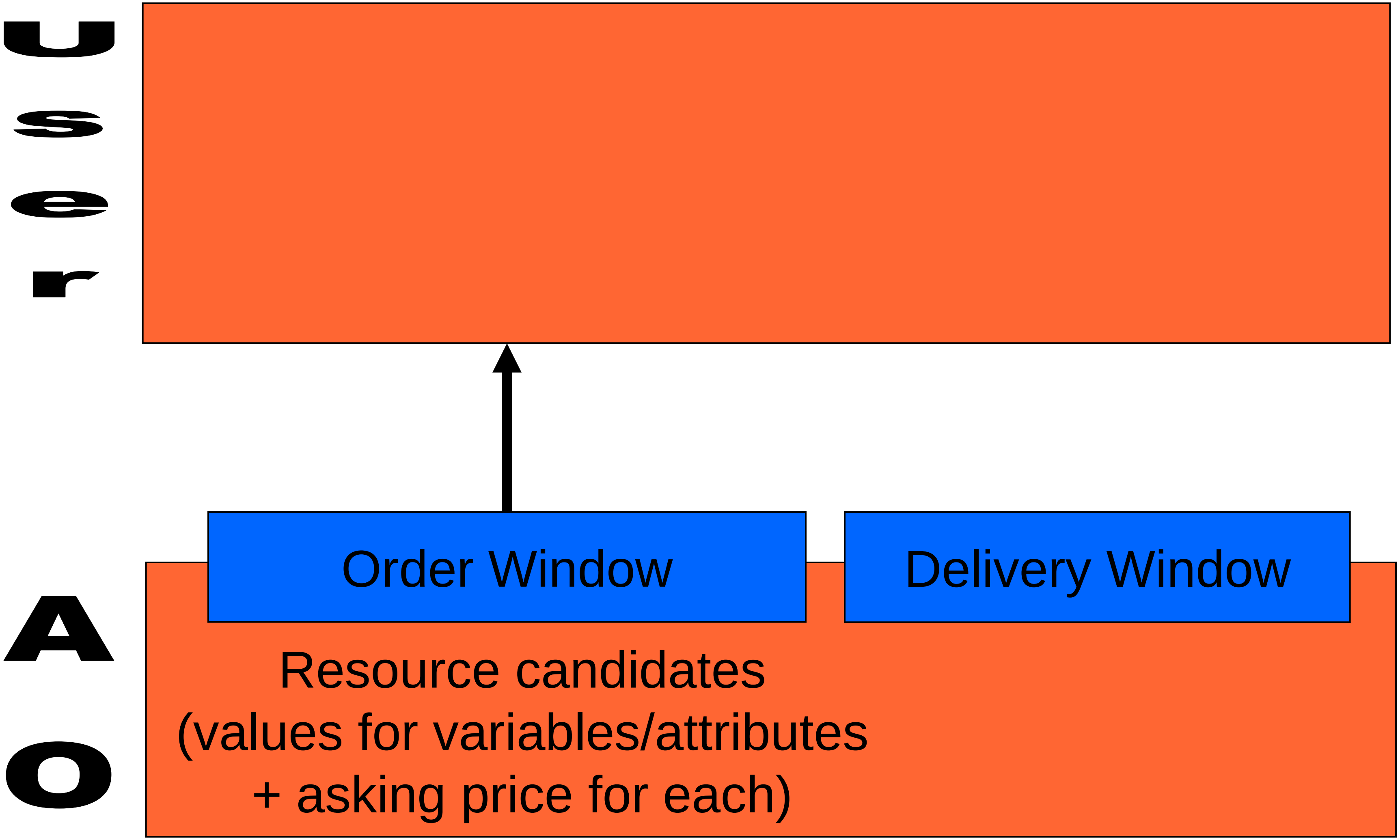


Make dummy resource
with attributes + variable constraint list
+ negotiation style + pickup approach
+ authorization + bid + negotiation id

Dummy resource

- **Attribute**
  - Constant value – “don’t care” value – Variable name
- **Negotiation style**
  - Immediate – Pending – Confirmation – Cancel
- **Pickup approach**
  - Alert client with a signal, interrupt, message
  - Client poll the pickup window or expect to find the resource ready at the window at a specified time
- **Authorization** - Key allows AO to determine the authority of the client
- **Bid** - Maximum price that the client is willing to pay for the resource
- **Negotiation ID** - Cookie

Negotiating with an AO



User

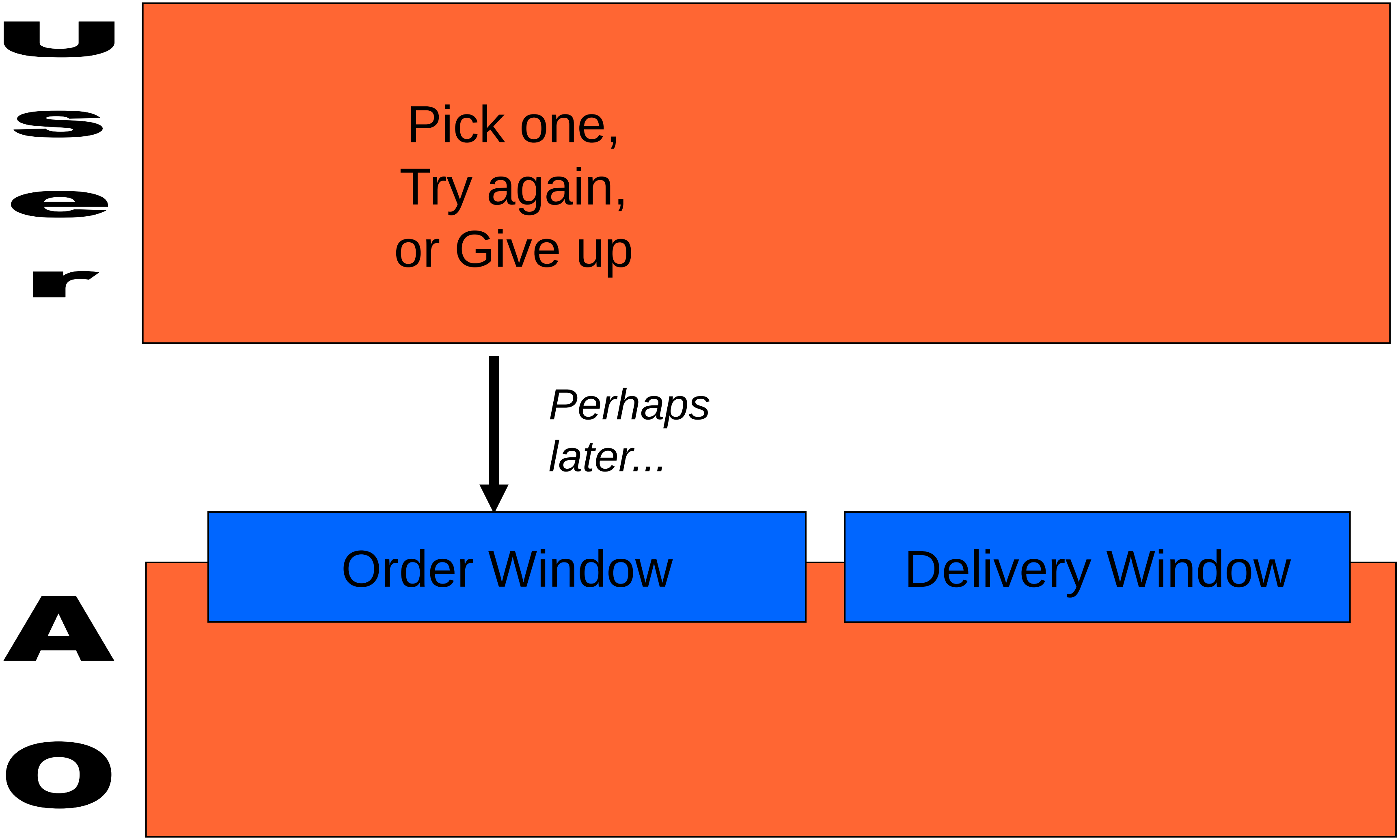
AO

Order Window

Delivery Window

Resource candidates
(values for variables/attributes
+ asking price for each)

Negotiating with an AO



User

Pick one,
Try again,
or Give up

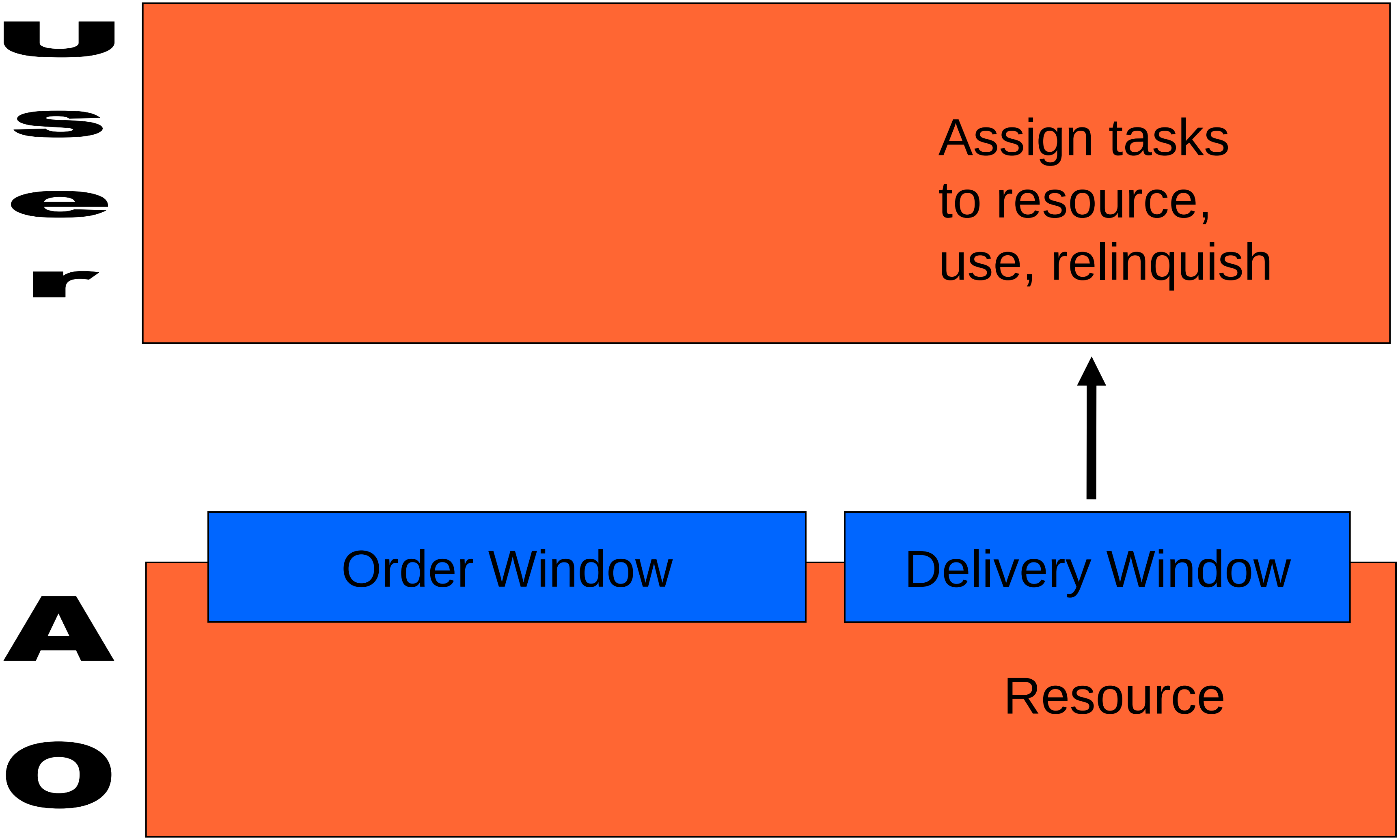
Perhaps later...

AO

Order Window

Delivery Window

Negotiating with an AO



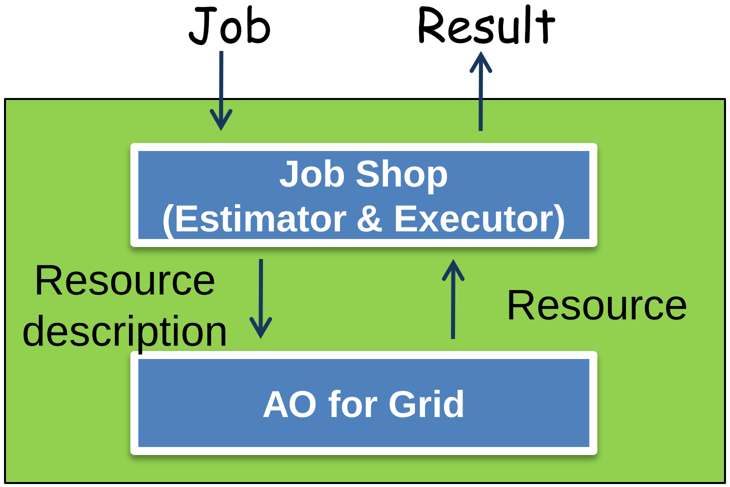
Job shops

Estimator

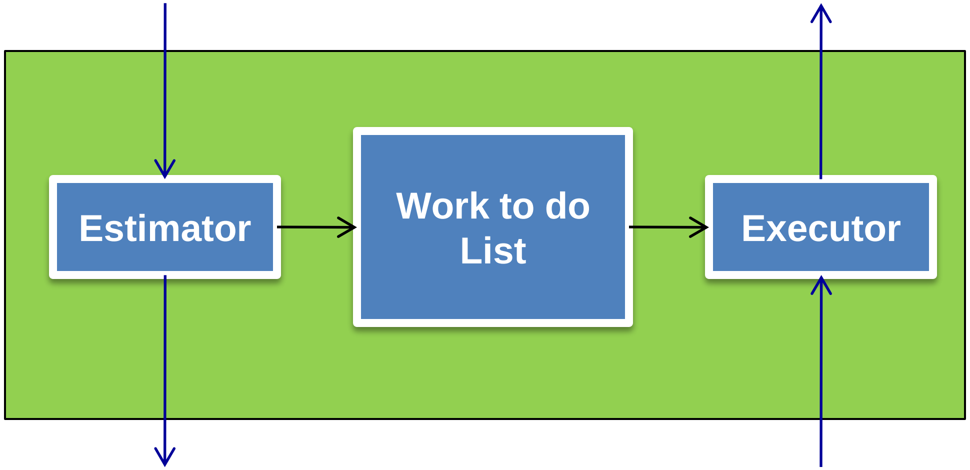
- Deal with the customer to determine when the job might be done, how much it might cost…
- Request resources from Grid AO
- Record what needs to be done in the job queue

Executor

- Take resources from AO delivery window
- Dequeue work from the job queue
- Build necessary environment for the tasks, initiate tasks
- Collect answers, notify, return the result to the client



Job scheduling atop AO



Job Shop

AO Summary

Many remaining gaps, both in details and in functionality

- How any client finds AOs that own the desired resources
- One approach is to imagine a tree of AOs, with the client always interacting with the root AO, but it is unrealistic to consider this tree as being hardwired when residing in an environment as dynamic as a computational grid
- A potentially useful and well-defined AO protocol will not be viable unless it can coexist with other contemporary approaches

Outline

1. Introduction
2. Hierarchical Resource Management
3. Abstract Owner (AO) Model
4. ***Economy/Market Model***
5. Summary

Economy / Market Model

- Introduction
- Grid resource broker - GRB
- Grid middleware
- Grid service provider
- Economic Models in a Grid Context

Introduction

- Who contributed to resources & why ?
  - Volunteers: for fun, challenge, fame, public good like SETI@Home & distributed.net projects.
  - Collaborators: sharing resources while developing new technologies of common interest – Globus, Legion, Ecogrid.

- How long ?
  - Short duration: GUSTO decommissioned.
  - What do we need ? Grid Marketplace!

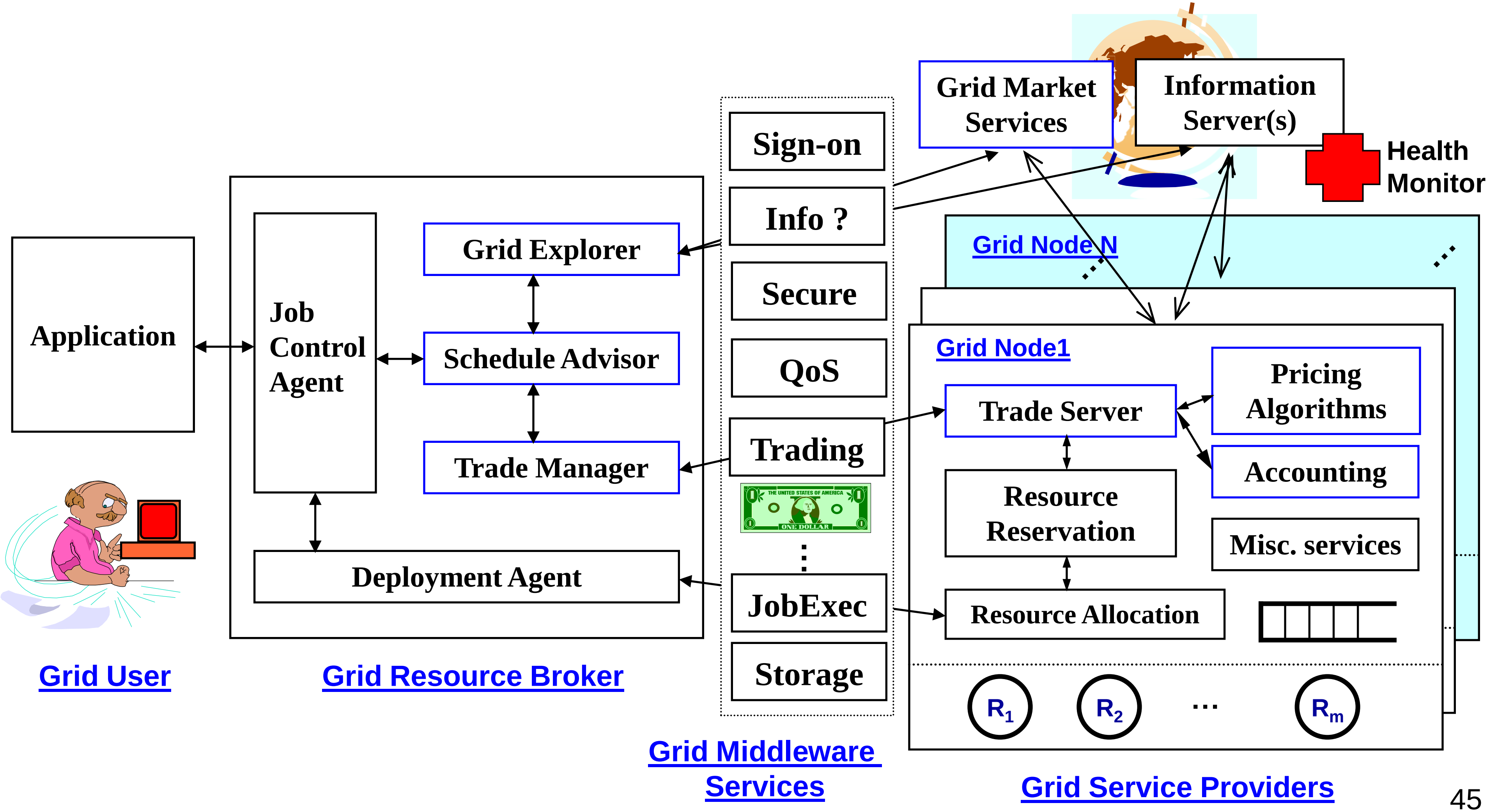
- What do we need ? Grid Marketplace!

Introduction

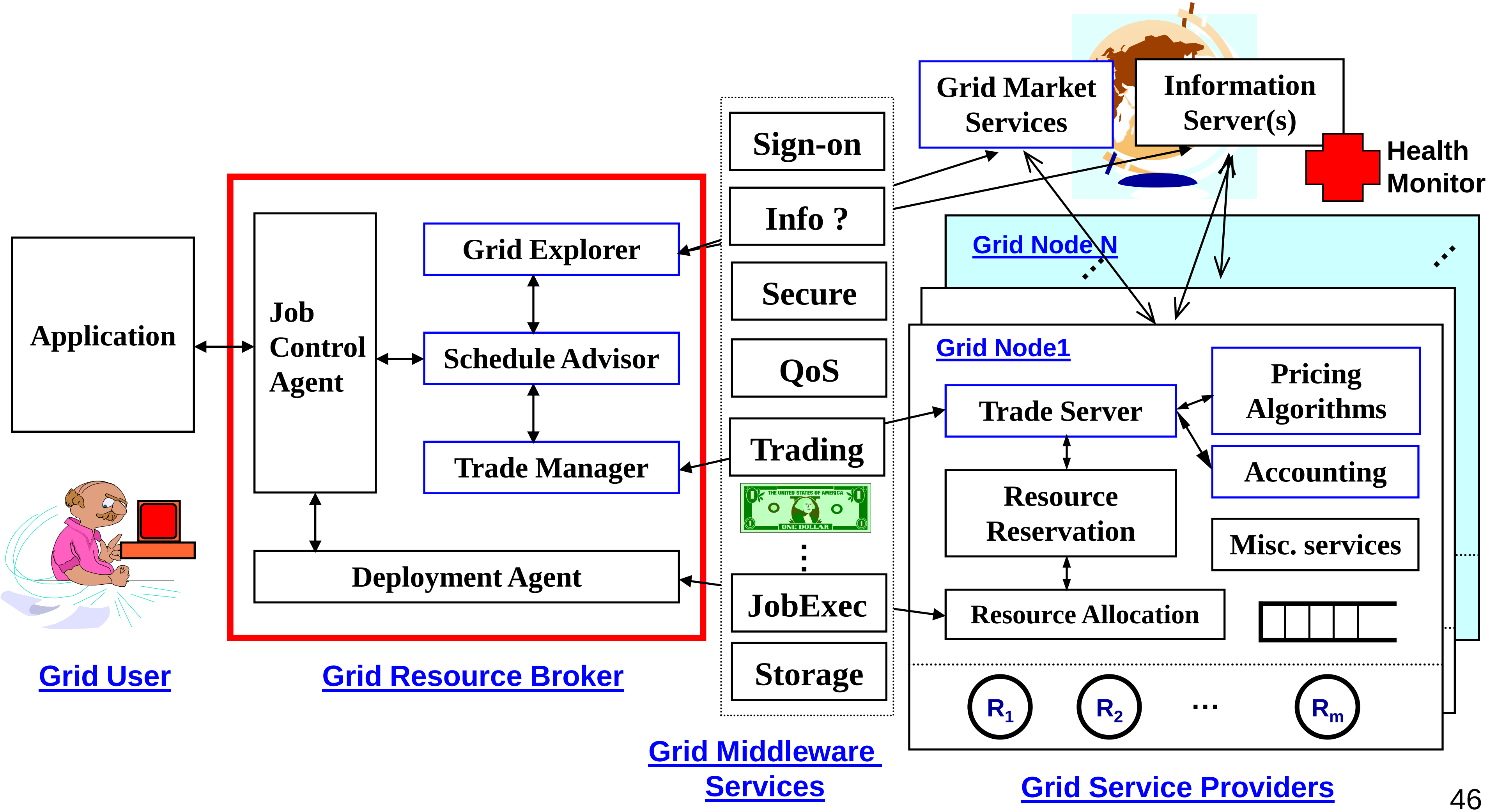
- In a computational market environment:
  - Resource users want to minimize their expenses
  - Owners want to maximize their return-on-investment.

- Key components of economy-driven resource management system:
  - User Applications
  - Grid Resource Broker
  - Grid Middleware
  - Grid service provider

Economy Model



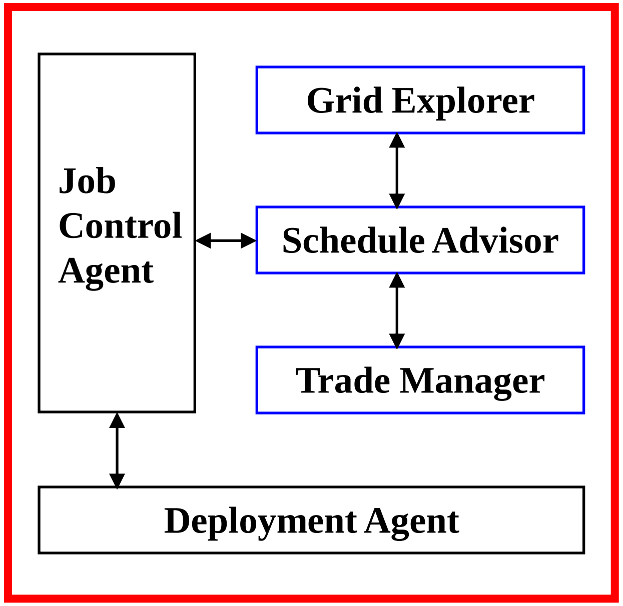
Grid resource broker – GRB



Grid resource broker – GRB

- Acts as a mediator between the user and grid resources using middleware services.

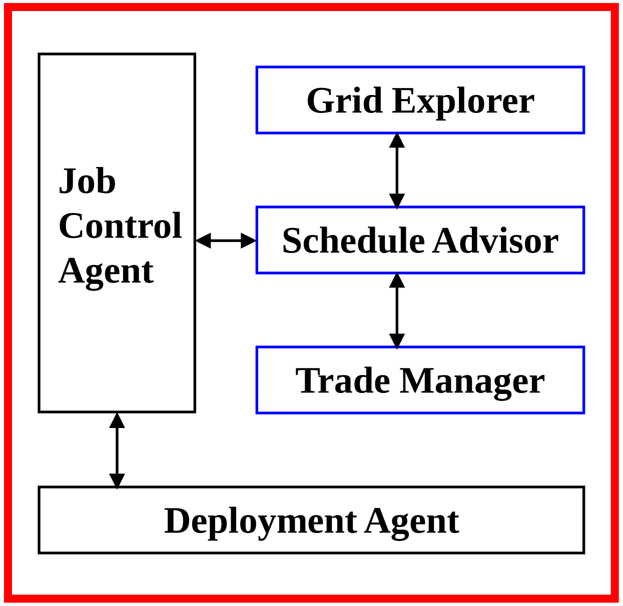
- Responsible for:
  - Resource discovery
  - Resource selection,
  - Binding of software (application), data, and hardware resources
  - Initiating computations
  - Adapting to the changes in grid resources
  - Presenting the grid to the user as a single, unified resource.



Grid Resource Broker

Grid resource broker components

- **Job Control Agent** (JCA): schedule generation, the actual creation of jobs, maintenance of job status, interacting with clients/users, schedule advisor, and dispatcher.
- **Schedule Advisor** (Scheduler): responsible for resource discovery (using grid explorer), resource selection, and job assignment (schedule generation).
- **Grid Explorer**: responsible for resource discovery by interacting with grid-information server and identifying the list of authorized machines, and keeping track of resource status information.

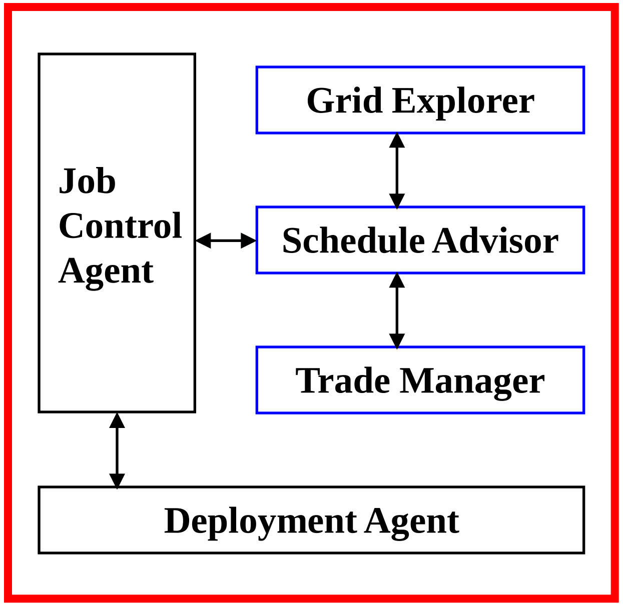


Grid Resource Broker

Grid resource broker components

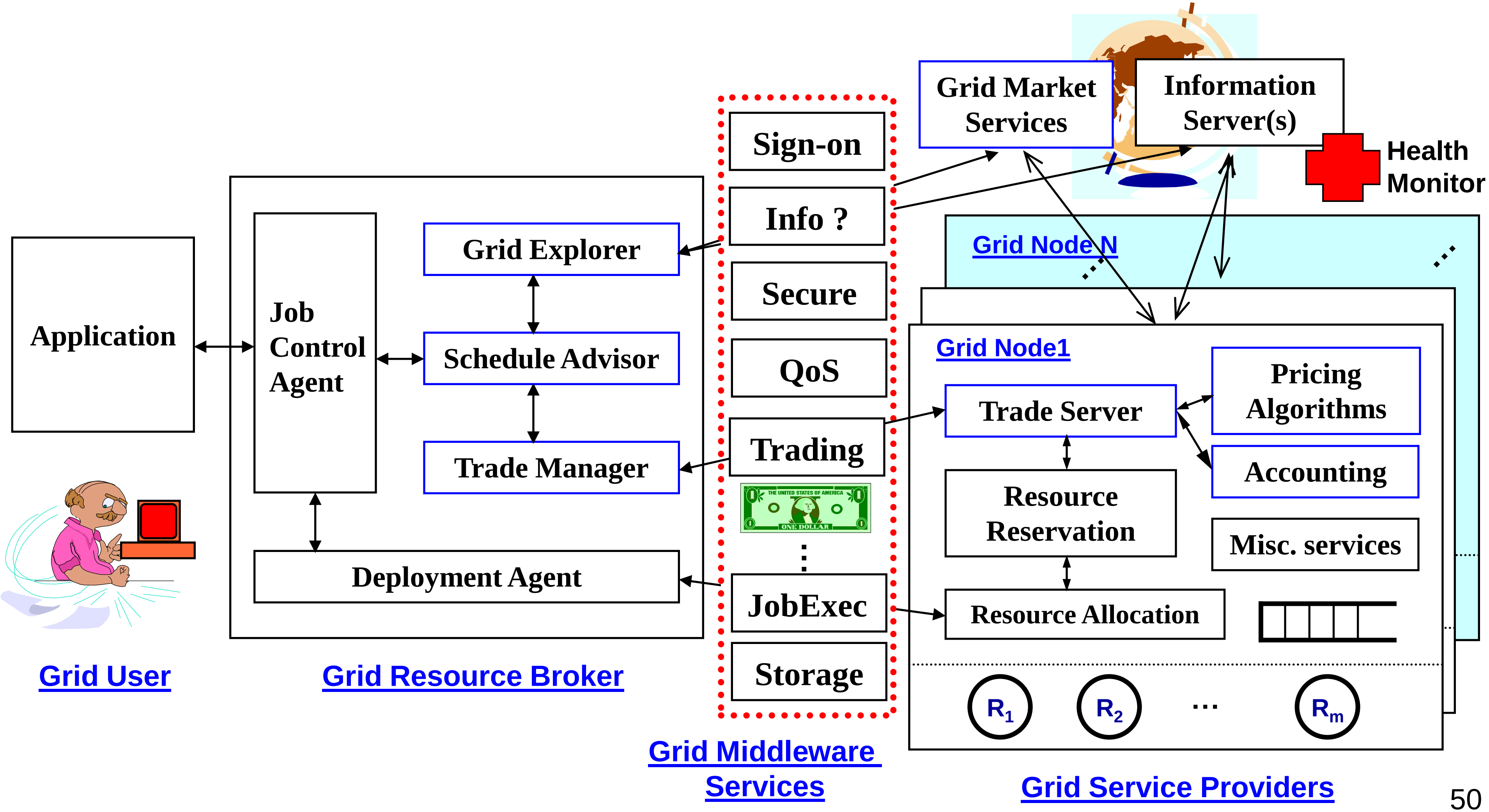
- **Trade Manager (TM):** works under the direction of resource selection algorithm (schedule advisor) to identify resource access costs. It can find out access cost through grid information server if owners post it.

- **Deployment Agent:** responsible for activating task execution on the selected resource as per the scheduler's instruction. It periodically updates the status of task execution to JCA.



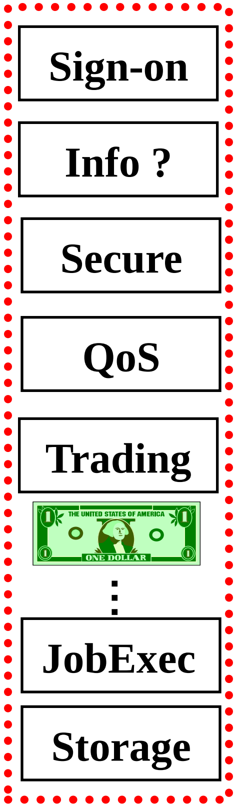
Grid Resource Broker

Grid middleware services



Grid middleware services

- Offers services that help in coupling a grid user and (remote) resources through a resource broker or grid enabled application.
  - Core services such as remote process management, co-allocation of resources, storage access, information (directory), security, authentication
  - Quality of Service (QoS) such as resource reservation for guaranteed availability and trading for minimising computational cost



Sign-on

Info ?

Secure

QoS

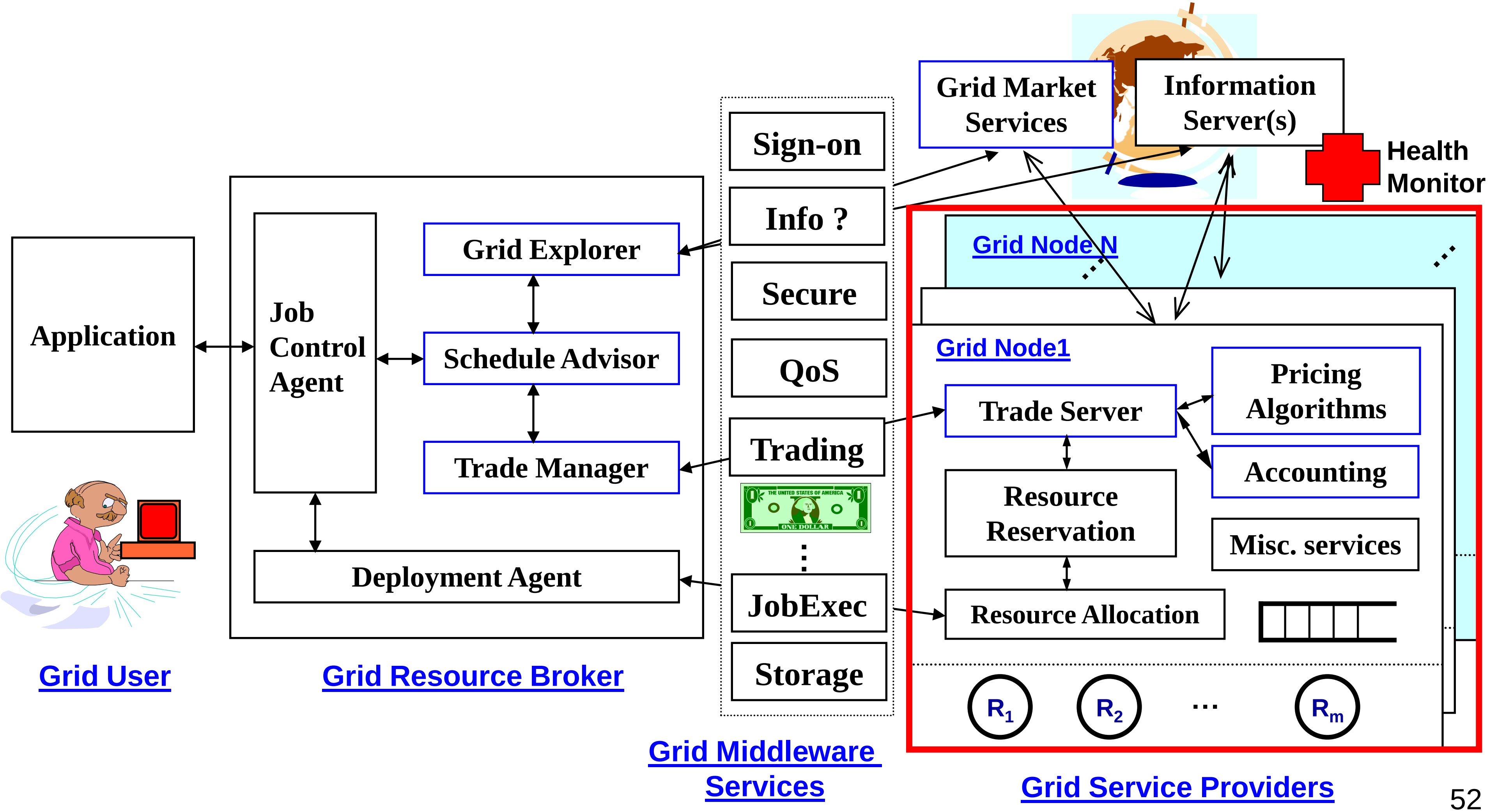
Trading

JobExec

Storage

Grid Middleware Services

Grid service provider



Application

Job Control Agent

Grid Explorer

Schedule Advisor

Trade Manager

Deployment Agent

Sign-on

Info ?

Secure

QoS

Trading

⋮

JobExec

Storage

Grid Market Services

Information Server(s)

Health Monitor

Grid Node N

Grid Node1

Trade Server

Pricing Algorithms

Accounting

Resource Reservation

Misc. services

Resource Allocation

R_1 R_2 … R_m

Grid User

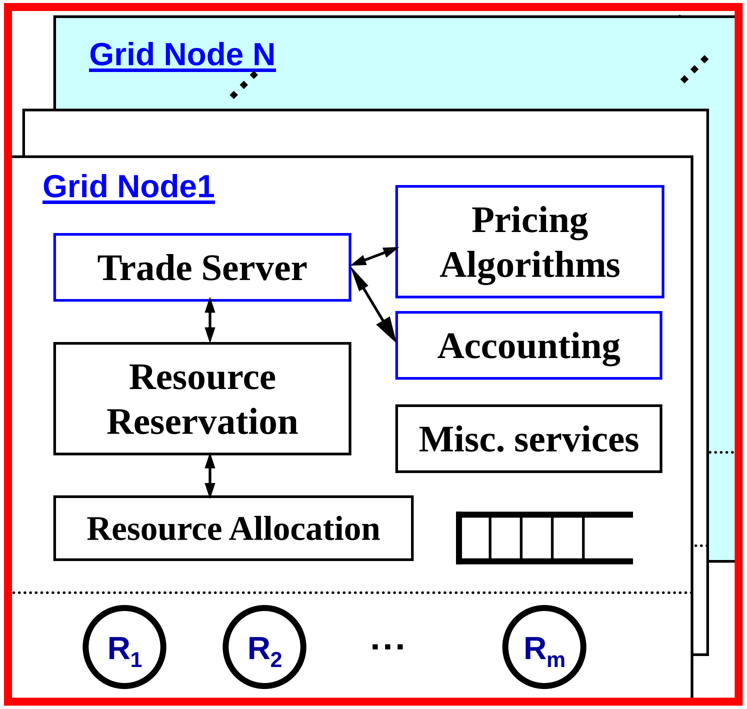
Grid Resource Broker

Grid Middleware Services

Grid Service Providers

Grid service provider

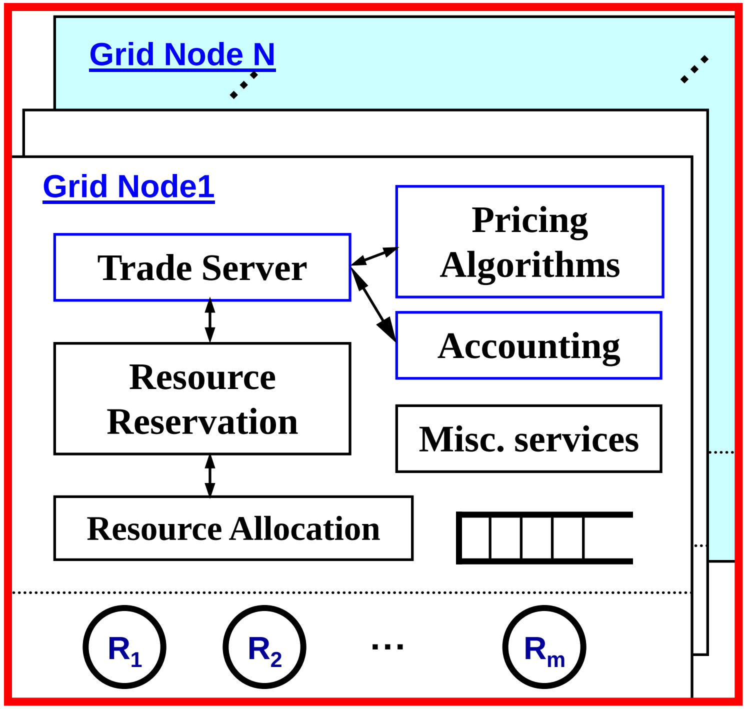
- Providing the traditional role of producer.
- Make their resources Grid enabled by running software systems along with Grid Trading Services (GTS) to enable resource trading and execution of consumer requests directed through GRBs.



Grid Service Providers

Grid service provider

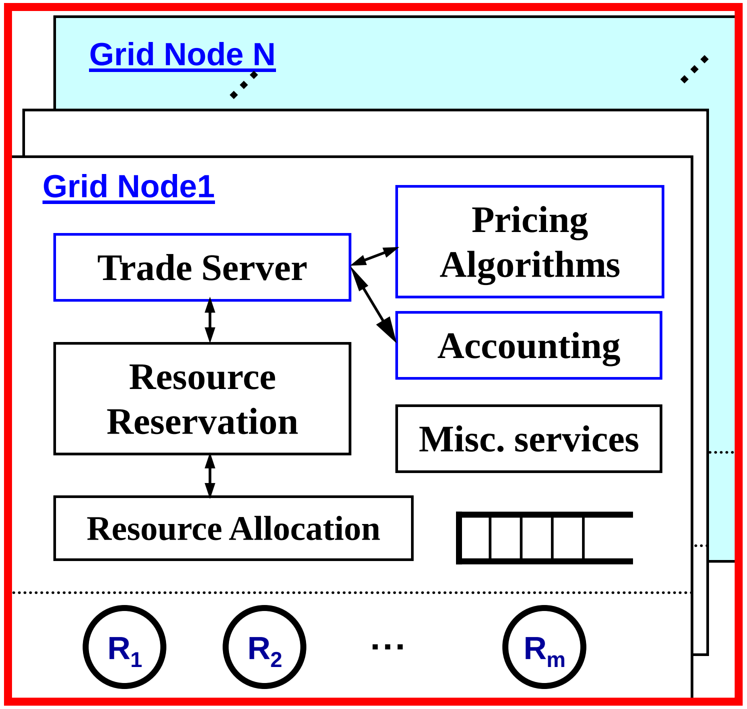
- The interaction between GRBs and GSPs during resource trading (service cost establishment) is mediated through a Grid Market Directory (GMD).
- The Grid trading server (GTS) can employ different economic models in providing services. The simplest would be a commodity model wherein the resource owners define pricing strategies including those driven by the demand and resource availability.



Grid Service Providers

Grid service provider

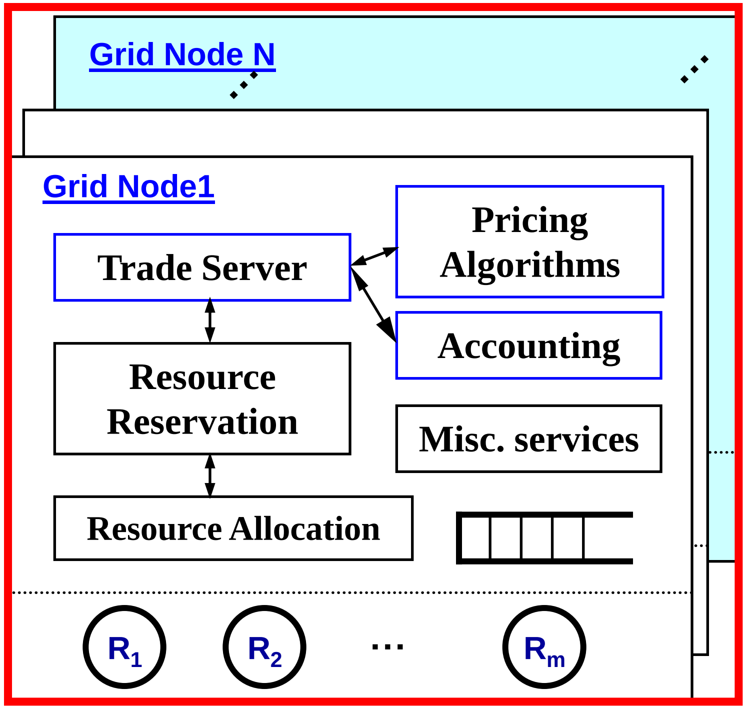
- **Trade Server (TS):**
  - A resource owner agent that negotiates with resource users and sells access to resources.
  - Aims to maximize the resource utility and profit for its owner
  - Consults pricing algorithms/models defined by the users during negotiation and directs the accounting system to record resource usage.
- **Pricing Algorithms/Methods**:
  - Define the prices that resource owners would like to charge users.



Grid Service Providers

Grid service provider

- **Accounting System**:
  - Responsible for recording resource usage and bills the user as per the usage agreement between resource broker (TM, user agent) and trade server (resource owner agent).
- **Local resource managers**
  - managing and scheduling computations across local resources such as workstations and clusters.
  - offering access to storage devices, databases, and special scientific instruments such as a radio telescope.



Grid Service Providers

Economic Models in a Grid Context

- Commodity Market Model
- Posted Price Model
- Bargaining Model
- Tendering/Contract-Net Model
- Auction Model
- Bid-based Proportional Resource Sharing Model
- Community/Coalition/Bartering Model
- Monopoly and Oligopoly

Outline

1. Introduction
2. Hierarchical Resource Management
3. Abstract Owner (AO) Model
4. Economy/Market Model
5. ***Summary***

Summary

MODEL	REMARKS	Systems
Hierarchical	It captures model followed in most contemporary systems.	Globus, Legion, CCS, Apples, NetSolve, Ninf.
Abstract Owner (AO)	Order and delivery model and focuses on long term goals.	Expected to emerge and most peer-2-peer computing systems likely to be based on this.
Market Model	It follows economic model for resource discover, sharing, & scheduling.	GRACE, Nimrod/G, JavaMarket, Mariposa.

Summary

- We have attempted to present these models in abstract high-level form as much as possible and have skipped low-level details for developers to decide.
- Many of the existing, upcoming and future grid systems can easily be mapped to one or more of the models discussed here.
- Real grid systems (as they evolve) are most likely to combine many of these ideas into a hybridized model (that captures essentials of all models) in their architecture

References

1. Rajkumar Buyya, Steve Chapin, and David DiNucci, Architectural Models for Resource Management in the Grid, USA, 2000

2. Rajkumar Buyya, David Abramson, and Jonathan Giddy, An Economy Driven Resource Management Architecture for Global Computational Power Grids

3. Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger, Economic Models for Resource Management and Scheduling in Grid Computing

4. David Abramson, Rajkumar Buyya, Jon Giddy. School of Computer Science and Software Engineering. Monash University, Melbourne, Australia.
Nimrod/G GRID Resource Broker and Computational Economy

5. Klaus Krauter1, Rajkumar Buyya and Muthucumaru Maheswaran. A taxonomy and survey of grid resource management systems for distributed computing.

Q&A