



# GrEASe

Grid Environment based on Agent Services

A. Boccalatte, S. Fazzari, S. Gatto, A. Grosso, C. Vecchiola  
I.i.d.o. – DIST University of Genova

speaker: Christian Vecchiola

# Agenda

- Introduction
  - Grids
  - Agents for the Grids
- GrEASe
  - Architecture
  - Lower Layer
  - Upper Layer
  - Use Case
- Conclusions



# Introduction

- Grids issues
  - Resource sharing
  - Resource localization
  - Resource storage
  - User profiling
  - Load balancing
  - Administration



# Introduction

- Grid environment
  - Complex
  - Heterogeneous
    - different hardware support on each node
    - different nodes can host different type of resources
  - Highly dynamic
    - nodes can be added or removed
    - links can be added or removed



# Introduction

## ■ Grids types

### – Computational Grids

- object: computational power of computers connected to the grid
- usage pattern: parallel and distributed algorithms
- example: SETI@home, gaming grids



# Introduction

## ■ Grids types

### – Data Grids

- object: huge collections of data distributed all over the network
- usage pattern: look for documents (image, text) that match some user defined criteria
- example: bio-informatic grids



# Introduction

## ■ Grids types

### – Service Grids

– the object is a service

– usage pattern:

– look for a particular kind of service

– build a composed service from the existing ones

– example:

– multimedia



# Introduction

## ■ Agents for the Grids

### – Agents features

- autonomous
- reactive
- proactive
- social ability

} flexible behaviour (intelligence)





# Introduction

## ■ Agents for the Grids

### – Multi-agent systems

– social ability is one of the most important features of agency

– agents interact by means of

– competition

– negotiation

– cooperation

– in order to better exploit this skill agents are arranged in communities called multi-agent systems (MASs)



# Introduction

## ■ Agents for the Grids

- Are these features valuable into a grid environment?
  - agent technology has been conceived for complex and highly dynamic systems
  - proactivity and reactivity become crucial in this kind of contexts where the single nodes need to adapt to:
    - the mutating conditions of the grid topology
    - the different load during their life cycle
  - social ability is important too:
    - cooperation is fundamental in resource discovery
    - cooperation and negotiation are winning practice for resource acquisition



# Introduction

## ■ Agents for the Grids

### – Some examples

- A4 methodology [J.Cao, CCGrid 01 – 02]
  - agents dynamically adapt to the conditions of the grid
  - agents are homogeneous and organized hierarchically
- MyGRiD [Moreau et al, CCGrid 03]
  - provides a collaborative environment for biologist researchers distributed in all the world
  - uses SoFAR as implementation technology
- Bond Agent Systems [L. Boloni, CIPC – 03]
  - extends the JADE framework with specific behaviours that abstract grid services



# GrEASe

## ■ GrEASe

Grid Environment based on Agent Services

GrEASe defines a grid environment

GrEASe uses the agent technology in order to define a simple, clear and easy to manage grid architecture.

GrEASe strongly relies on the feature of agency in the design of the different components of a Grid



# GrEASe

## ■ Features

- Based on the AgentService programming framework
- Uses a different approach to apply agents to Grids (functional decomposition)
  - Identifies all the core functionalities that characterize the system
  - Defines a role for each set of functionalities that make up a service
  - Defines an agent for each role previously identified
- Models the entire Grid as dynamic and distributed multi-agent system



# GrEASe

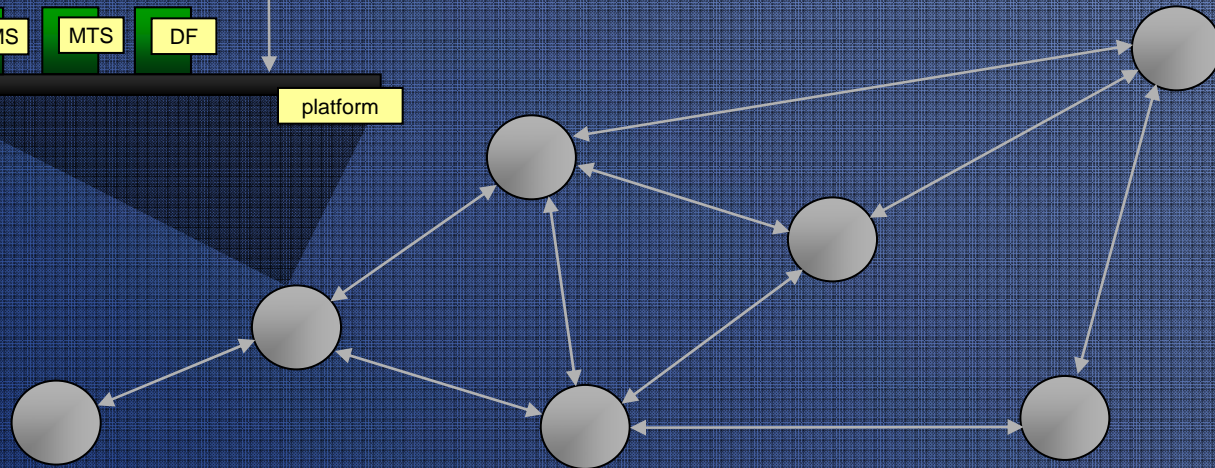
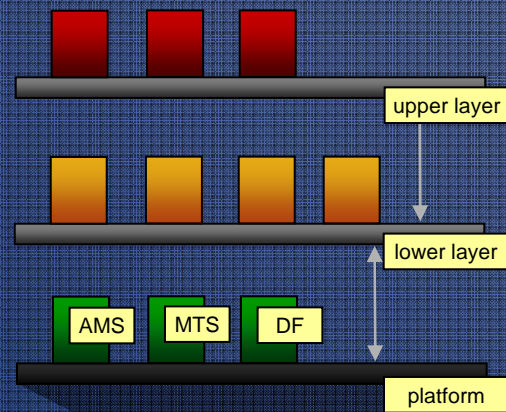
## ■ Architecture

- Each node is defined by an instance of the AgentService programming platform
- The instance of the platform host the portion of the multi-agent system related to the node
- Each node is organized into ...
  - lower layer
  - upper layer



# GrEASe

## Architecture



12/1/2004

Christian Vecchiola - GrEASe  
WOA 2004 - Torino



# GrEASe

## ■ Node Architecture

### – Lower Layer

- Core functionalities of each type of grid
  - Node management
  - Topology management
  - Resource discovery
  - Authentication
  - Data transfer
  - User interface
- One to one mapping between this services and the agents belonging to this layer





# GrEASe

## ■ Lower Layer

### – NodeManager

- Takes care of the list of the shared resources
- It is the access point to the node
- Delivers requests that it cannot handle to the other agents of the node



# GrEASe

## ■ Lower Layer

### – Dispatcher

- Handles the information about the topology of the node and neighbour nodes
- Forwards outgoing queries to the dispatchers of the neighbour nodes able to handle them
- Handler incoming queries by forwarding them to the neighbour dispatchers or to the NodeManager of the node



## ■ Lower Layer

### – ResourceProvider

- Can be more than one for each node
- Handles a subset (if not alone) of the resources of the node
  - Allocation status
  - Temporary owner
- Interacts with the upper layer agents that are bound to the specific resources handler by the provider



# GrEASe

## ■ Lower Layer

### – Carrier

- Handles the transfer of data among the nodes
- Uses different protocols in according to
  - The peer node
  - The type of the resource
- Interacts with the NodeManager to coordinate the delivery of a resource to the client



# GrEASe

## ■ Lower Layer

### – Authenticator

- Verifies the credential of end-user that want to access the grid (first level of authentication)
- Gives the user access to the specified resource in according to its security profile (second level of authentication)



# GrEASe

## ■ Lower Layer

### – UserAgent

- Represents the user into the multi-agent system hosted into the node where the user has logged in
- Not present in all the nodes
- Acts like a proxy of the user and handles all his requests
- Created at user login, destroyed the at logout



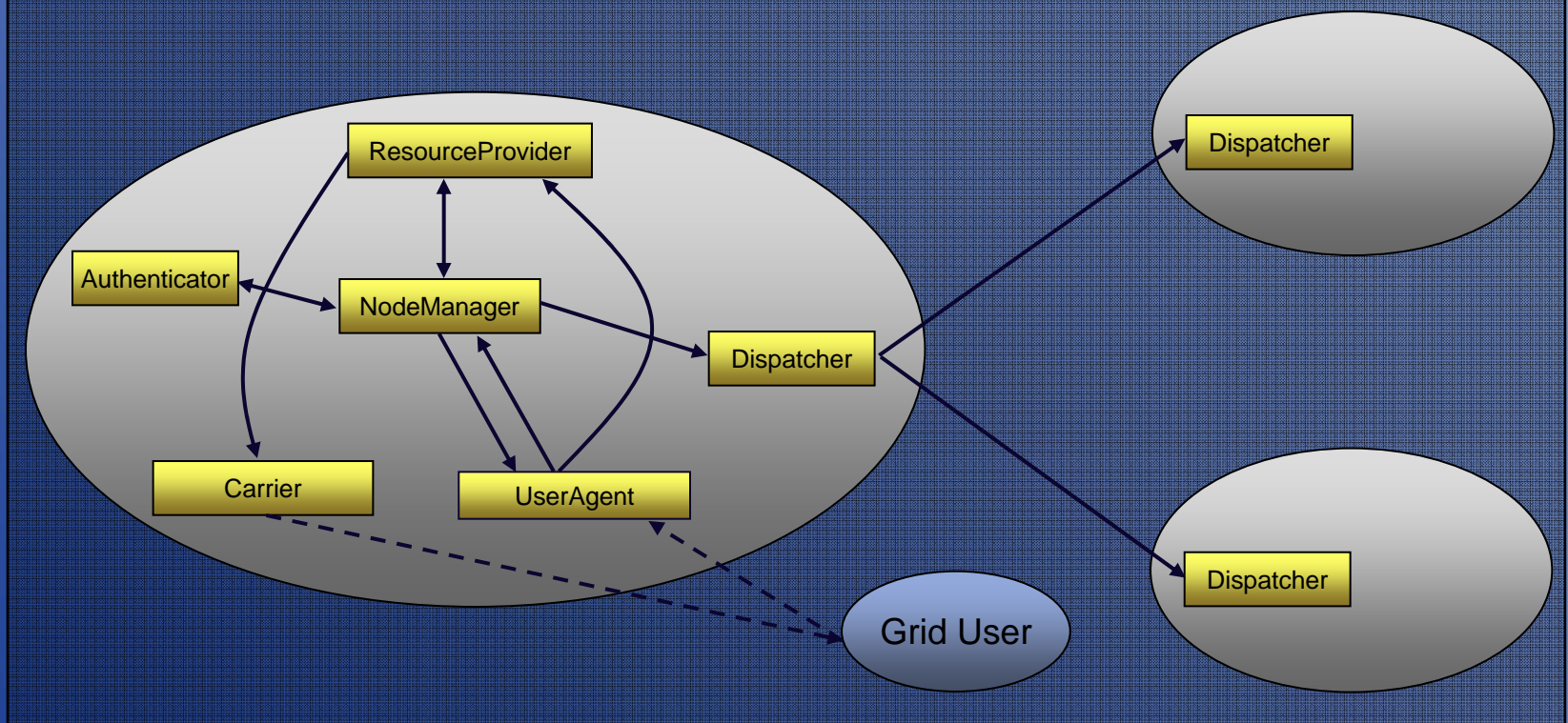
## ■ Upper Layer

- Grid type specific agents
  - Different agents have to be designed in order to deal with the nature of resources
    - computation
    - data
    - services
  - They share a common set of behaviours that are used to interact with the resource provider



# GrEASe

## ■ Use Case: search for a resource



12/1/2004

Christian Vecchiola - GrEASe  
WOA 2004 - Torino



# Conclusions

## ■ Some considerations

- Agent Technology can be an interesting approach to grid computing
- The use of functional decomposition lead to..
  - .. an intuitive, modular organization of the system
  - .. a system easy to mantain
  - .. a new a approach



# Conclusions

## ■ Future enhancements

- Integration and interoperation with existing grid systems (OGSI in particular)
- Improvement of
  - the routing techniques used by Dispatcher agents
  - the authentication method of Authenticator agents
  - the number of protocols used by Carrier agents





Thanks for the attention...