AGRID - Distributed and Autonomous Agent-based Grid Resource Allocation Manager

Work in Progress

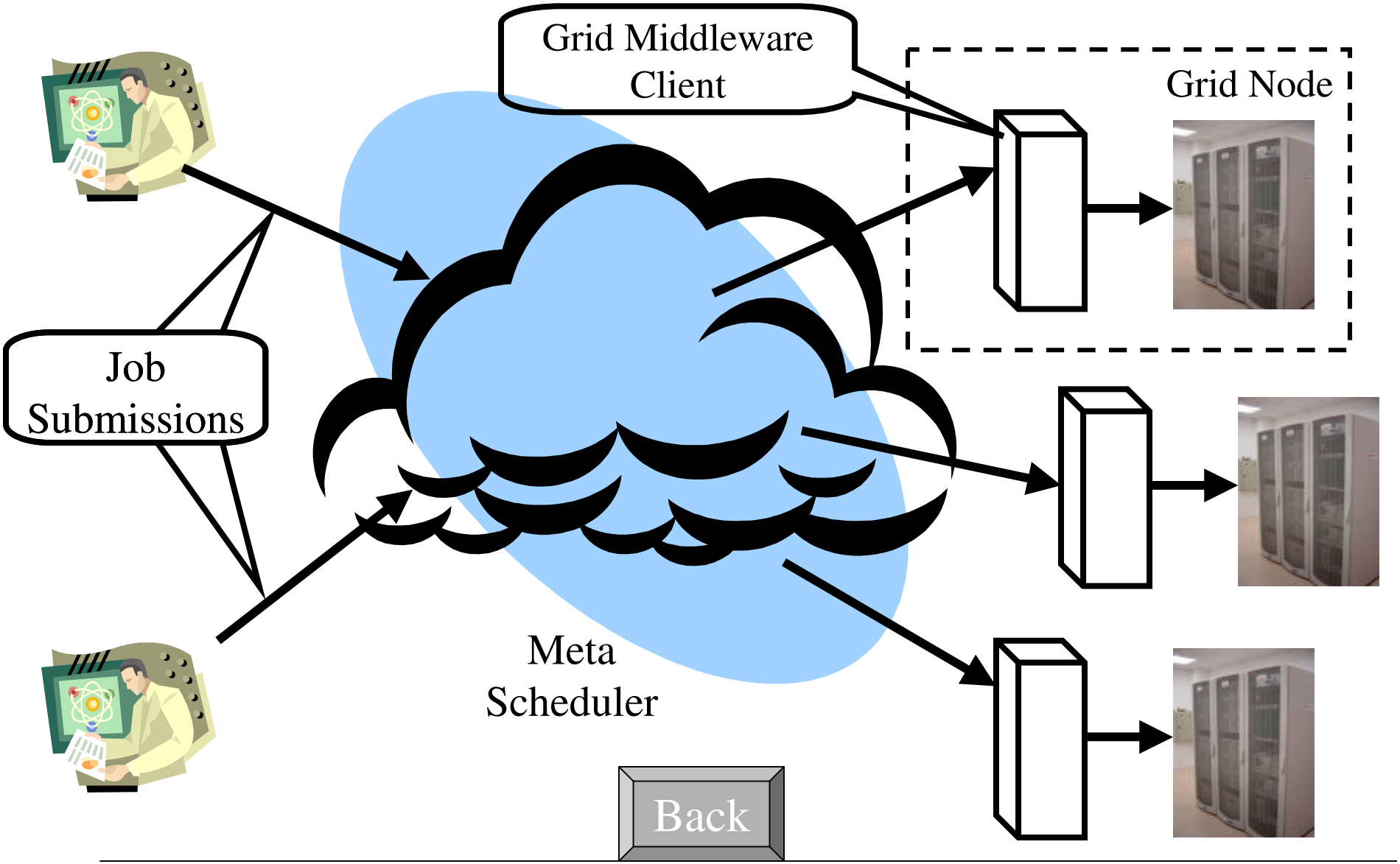
Tsahi Birk & Yoav Levy

Electrical Engineering Department, Technion

AGRID – Presentation Outline

- Defining the domain of application
  - Large collections of medium-sized compute clusters
- Defining the problem and the need
  - Fluctuations in resource availability ineffectively handled by centralized meta-schedulers
- Presenting the proposed solution
  - Distributed, two-tiered, market-based resource management

Typical Grid Environment



Grid Resource Allocation Management (GRAM)

- Resource Management is a central component of any grid system
- Typical issues in GRAM include:
  - Resource Discovery (global namespace)
  - Allocation & Scheduling Model
  - QoS and Prioritization
  - State Estimation
- Multi-cluster GRAM calls for different thinking

Diagram

Problem Statement

- Information on cluster resource availability is
  - Incomplete: hard to retrieve and manage extensive state at a central location
  - Volatile: part of the state changes at a high rate
- Ability of a central meta-scheduler to respond to changes in intra-cluster resource availability is limited
- Causes valuable usage slots to be wasted
  - Example: Aircraft takeoff / landing queue problem

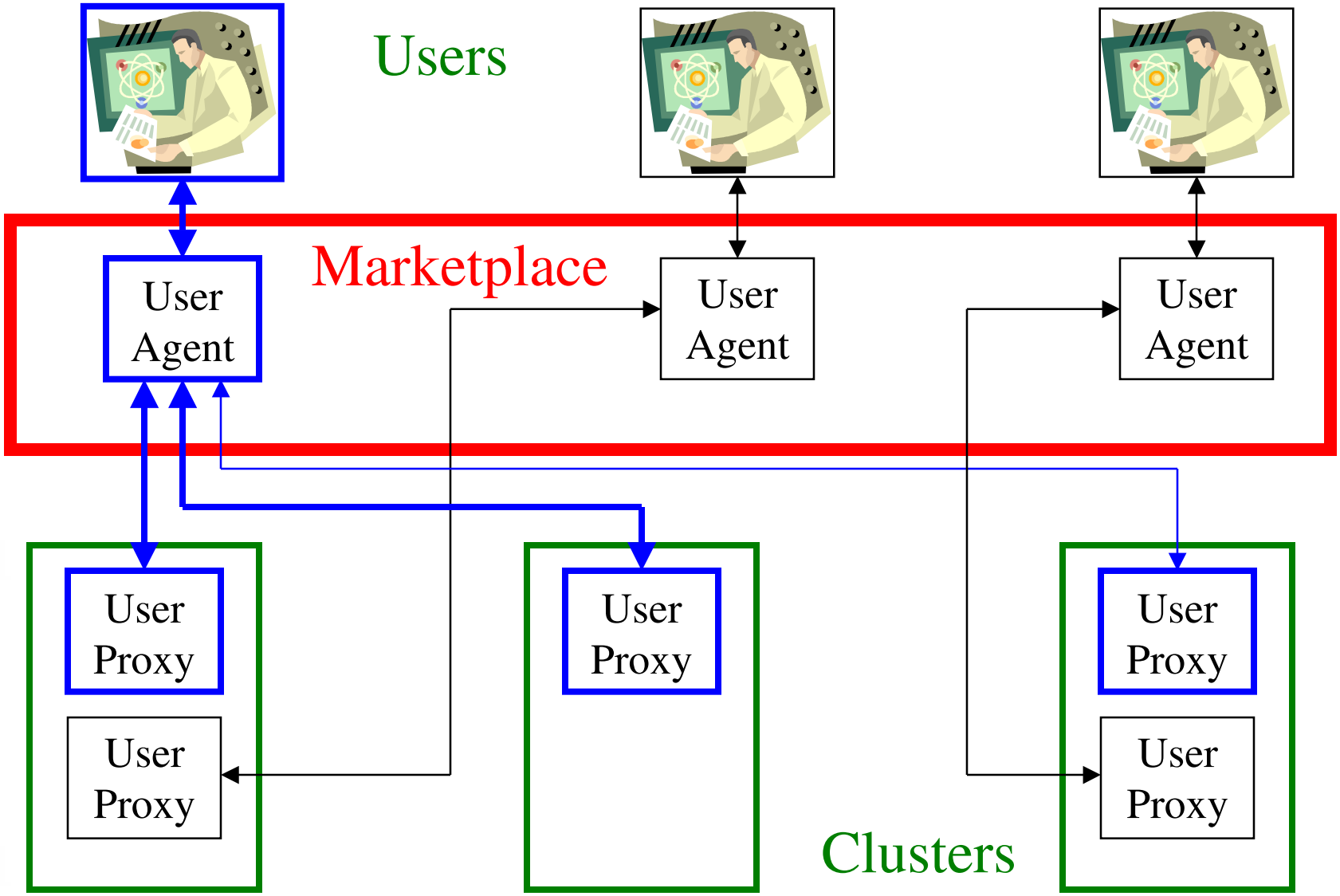
AGRID – The Agent Grid
Principles of Operation

- Two-tiered middleware composed of software agents
- Dynamic allocation of the resources of any given cluster to jobs running on it is handled at that cluster by competing software agents
- Slower, inter-cluster decisions are handled in a centralized "marketplace"
  - Placement of new jobs
  - Redirection of ongoing jobs in reaction to major, persistent state changes

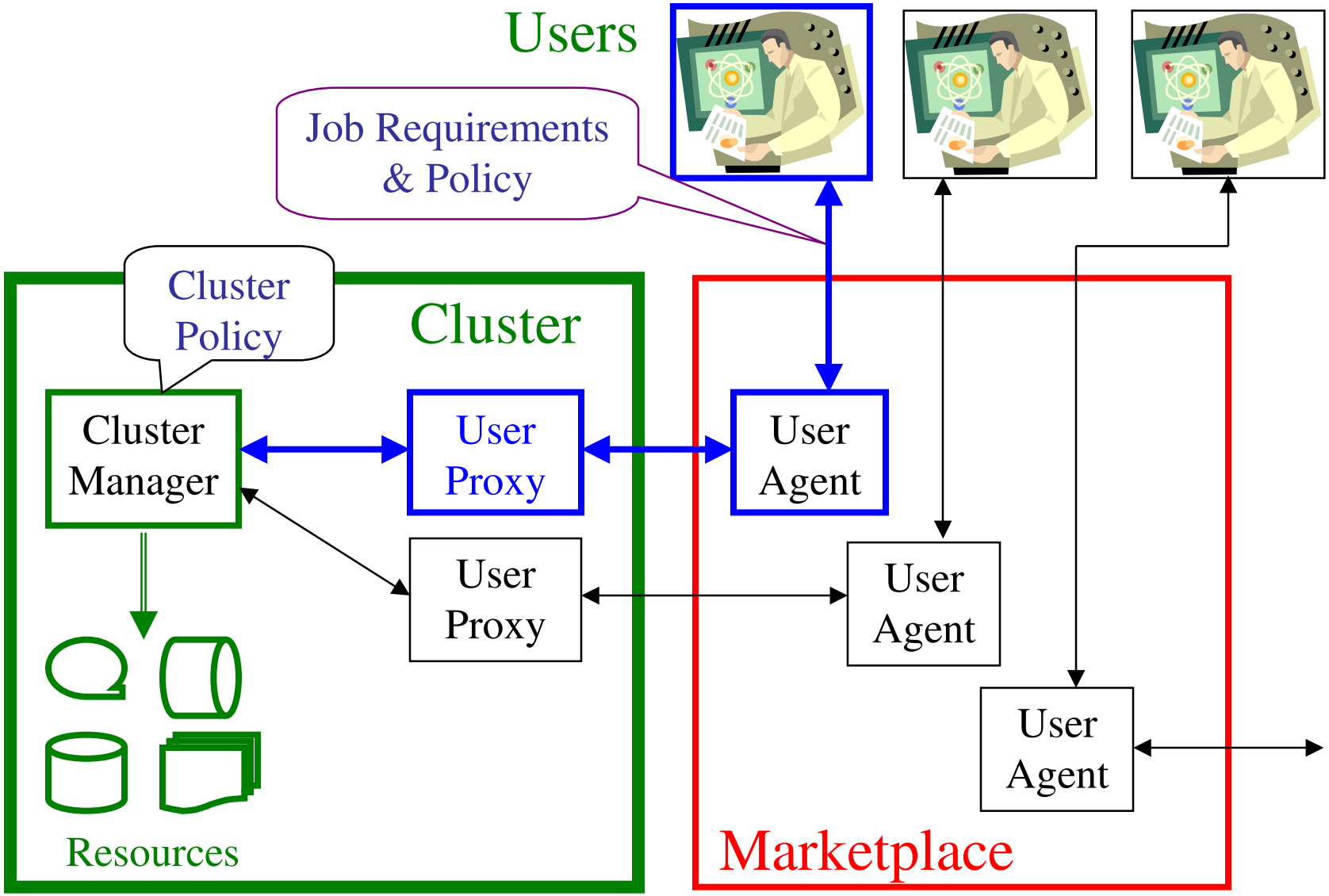
AGRID – The Agent Grid
Principles of Operation (contd.)

- **Market-based**: utility-based bidding for resources among competing parties
- **Agent Learning**: persistent agents gaining cumulative knowledge on site resource performance & pricing
- **Policy-based**: users and cluster owners interface middleware using policy functions

AGRID – Inter-Cluster Level



AGRID – Intra-Cluster Level



AGRID – Assumptions & Prerequisites

- Local cluster scheduler capable of
  - Fragmenting resource usage to temporal segments
  - Reporting resource performance
- Jobs can migrate (check-pointing supported)
- Several markets may exist in the “world”; users will select their preferred market resources are scheduled and run on one

Local Global

Algorithms – Design and Operation

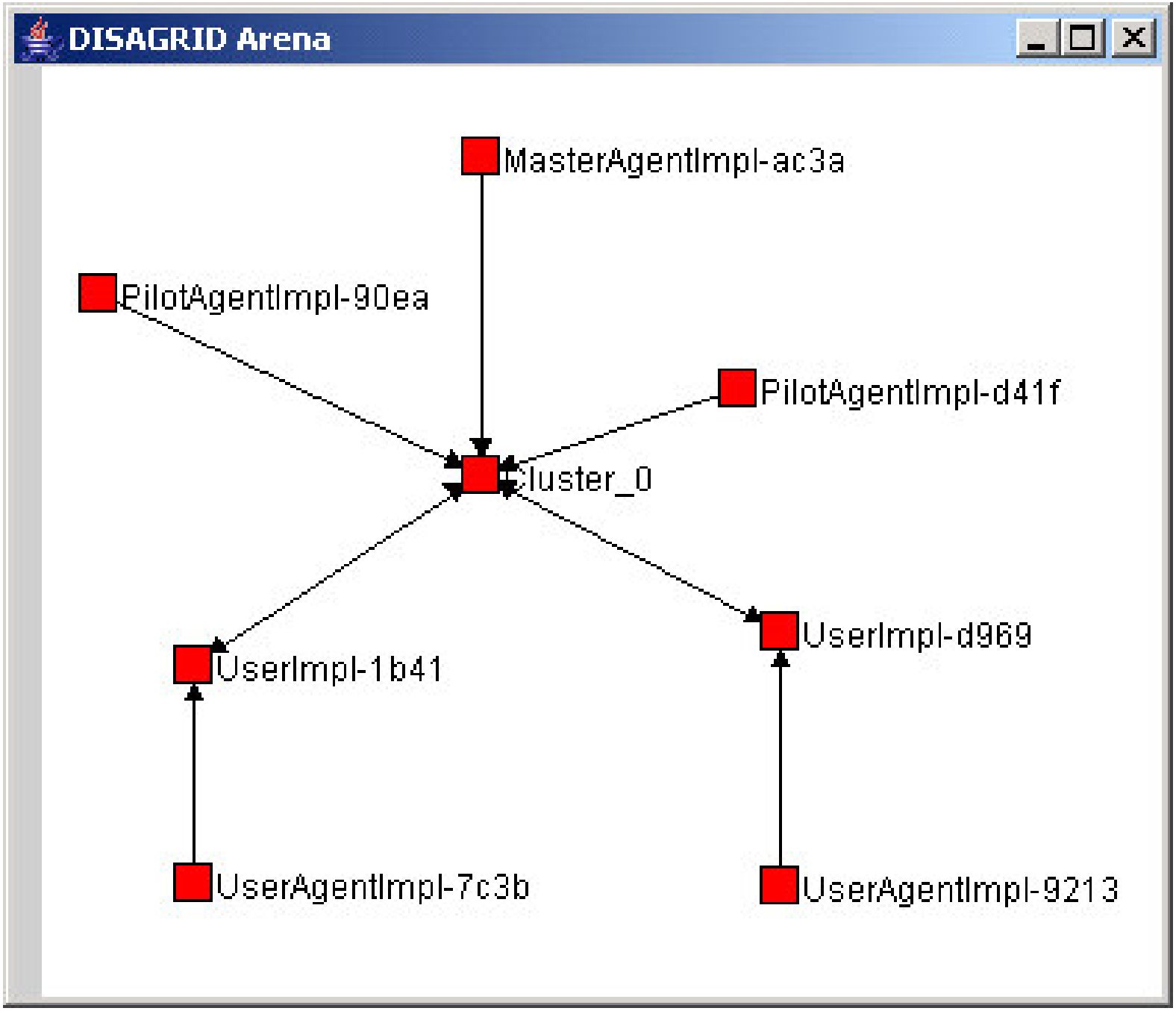
- Each user proxy periodically runs the same algorithm on its local database
- Goals:
  - Find locally optimal solutions to pending jobs
  - Prepare bids to be sent to cluster resource manager
- Utility of match will be affected by:
  - Target QoS
  - Transfer costs to a different resource
  - Risk level the job is at the time
  - Proximity of the Allocation Horizon
- Use back-tracking, multi-knapsack algorithm

Local

Global

DISAGRID – AGRID Simulator

- Designed to serve as a proof-of-concept as well as a testbed for algorithm and protocol evaluations
- Incorporates a thin abstraction layer
  - The rest is final code
- Java-based with extensive use of code generation
  - Use of XML, XSLT
  - Around 40% of semicolons are machine generated
- Uses existing visualization package - JUNG


DISAGRID Arena
MasterAgentImpl-ac3a
PilotAgentImpl-90ea
PilotAgentImpl-d41f
Cluster_0
UserImpl-d969
UserImpl-1b41
UserAgentImpl-7c3b
UserAgentImpl-9213

Current Status & Future Work

- Work done till now:
  - Simulation framework functioning
  - “Job Path” protocols implemented
  - Intra-cluster allocation algorithms implemented
- Future Work:
  - Complete inter-cluster tier
  - Investigate effect of various system factors:
    - Grid size
    - Cluster resource & performance variety & volatility levels
    - Interaction and division of responsibility between tiers
  - Develop a distributed prototype
  - Large-scale experimentation on a real-world testbed

Questions?
Comments?

For additional information visit:
http://tx.technion.ac.il/~yoavlevy/agrid/
(link will appear on seminar site)

Thank You!

(Enjoy the Food…)

Market-Based / Agent-Based GRAM – Related Work

- Popcorn
  - An infrastructure for programming distributed computations in Java
  - Uses auctions to match resource "buyers" and "sellers"
- Nimrod/G
  - Nimrod enables users to perform parametric experiments by declaring deadlines and prices
  - Nimrod/G extends Nimrod to work over the Globus toolkit
- Negotiating Agents
  - Grid and Agent technologies are intermingling
  - Developers of the Globus Toolkit, agree with agent experts on the advantages that the convergence of the two approaches may bring
- Market-based GRAM and agent-based negotiation gain interest but still built on top of existing centralized and hour-glass shaped resource management systems