

Scheduling with markets

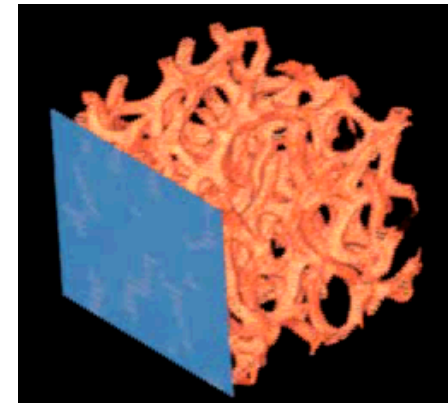
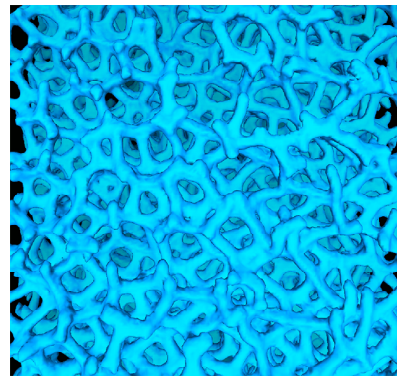
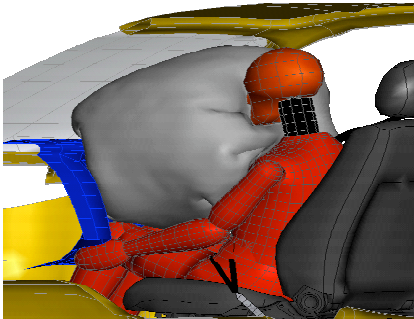
Mathias Dalheimer
dalheimer@itwm.fhg.de

Mathias Dalheimer
PhD-student
PLEASE INTERRUPT, ASK!



- Working at the Fraunhofer Institute for Applied mathematics
- Kaiserslautern (close to Frankfurt)
 - 150 Employees & PhD students
 - several departments with strong focus on math
 - > “Mathematics as a technology”
 - We have an energy-efficient building (we don’t need cooling during the winter season!)

Departments



Transport Processes

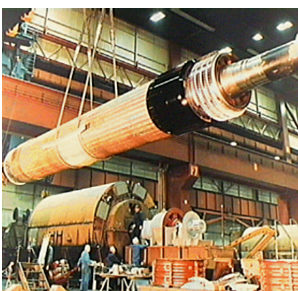
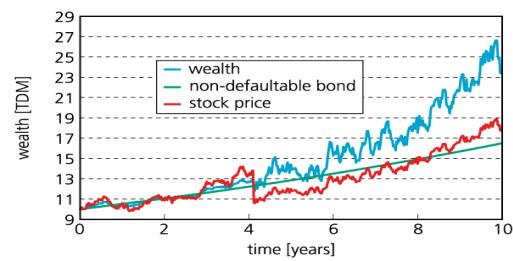
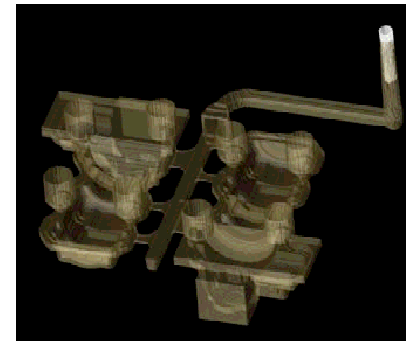


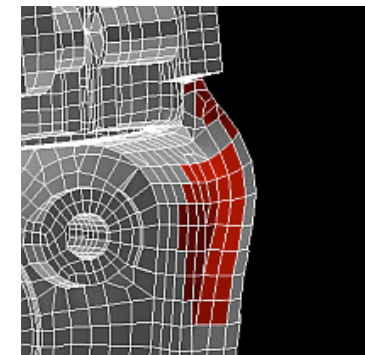
Image Analysis



Optimization



Flow in Complex Structures



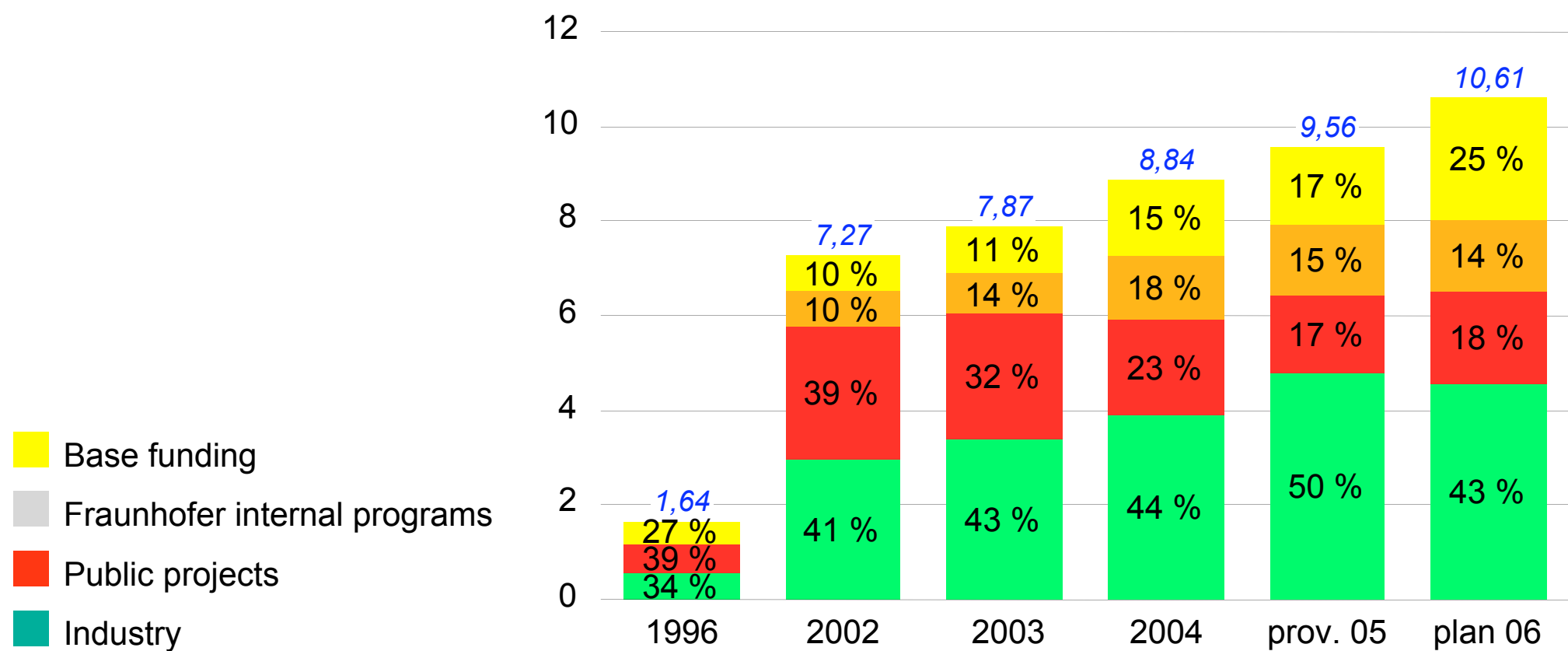
Adaptive Systems

Financial mathematics

HPC + Visualisation

Dynamics + Durability

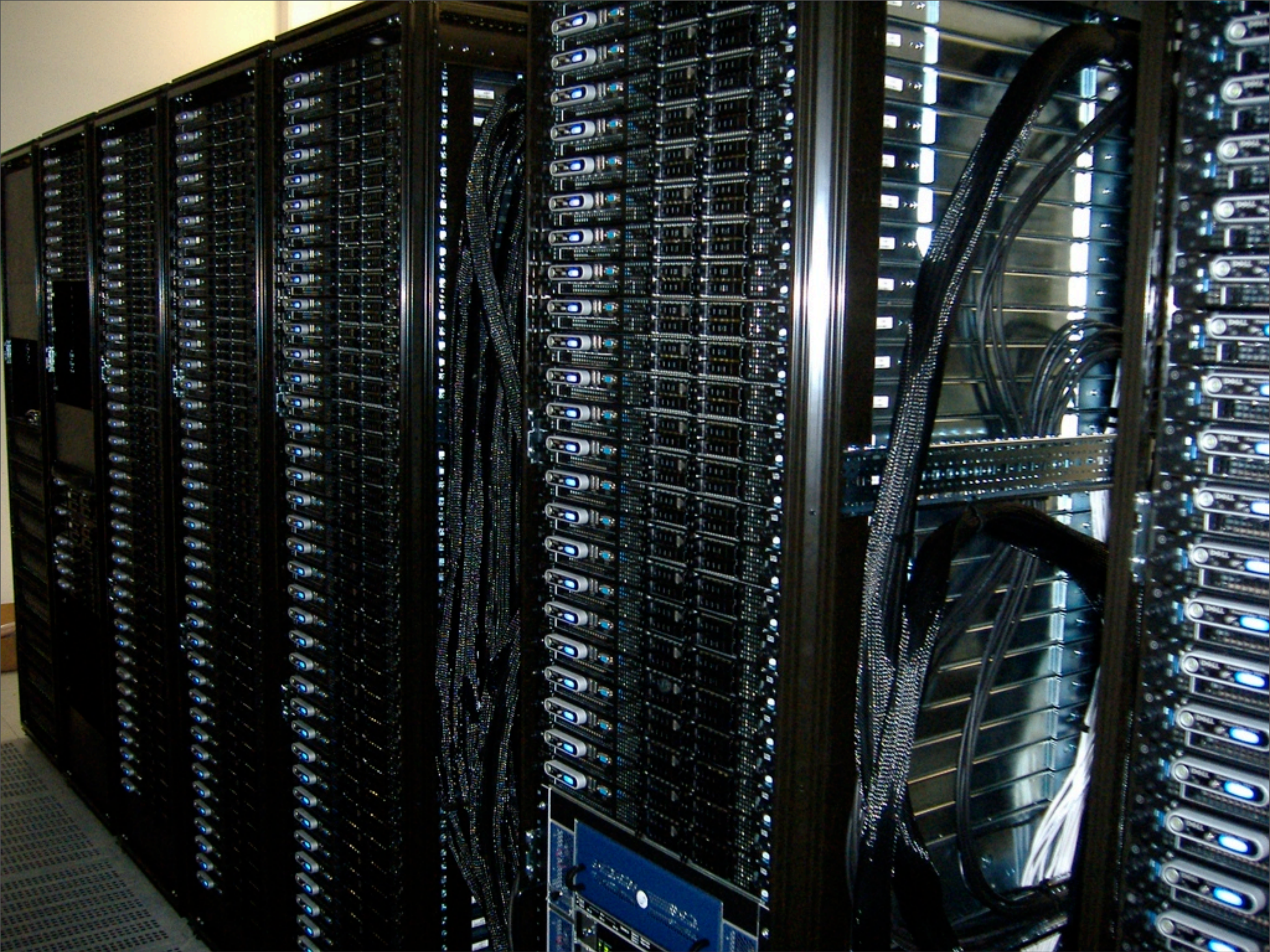
Budget Development





The CC-HPC is engaged in various industry and research projects

- Oil industry
- Financial business
- ...



Our new cluster system “hercules”

- 1000 core cluster, Intel Woodcrest, >250 nodes
- 8GB RAM per node
- Infiniband DDR interconnect

-> Up until end of the month

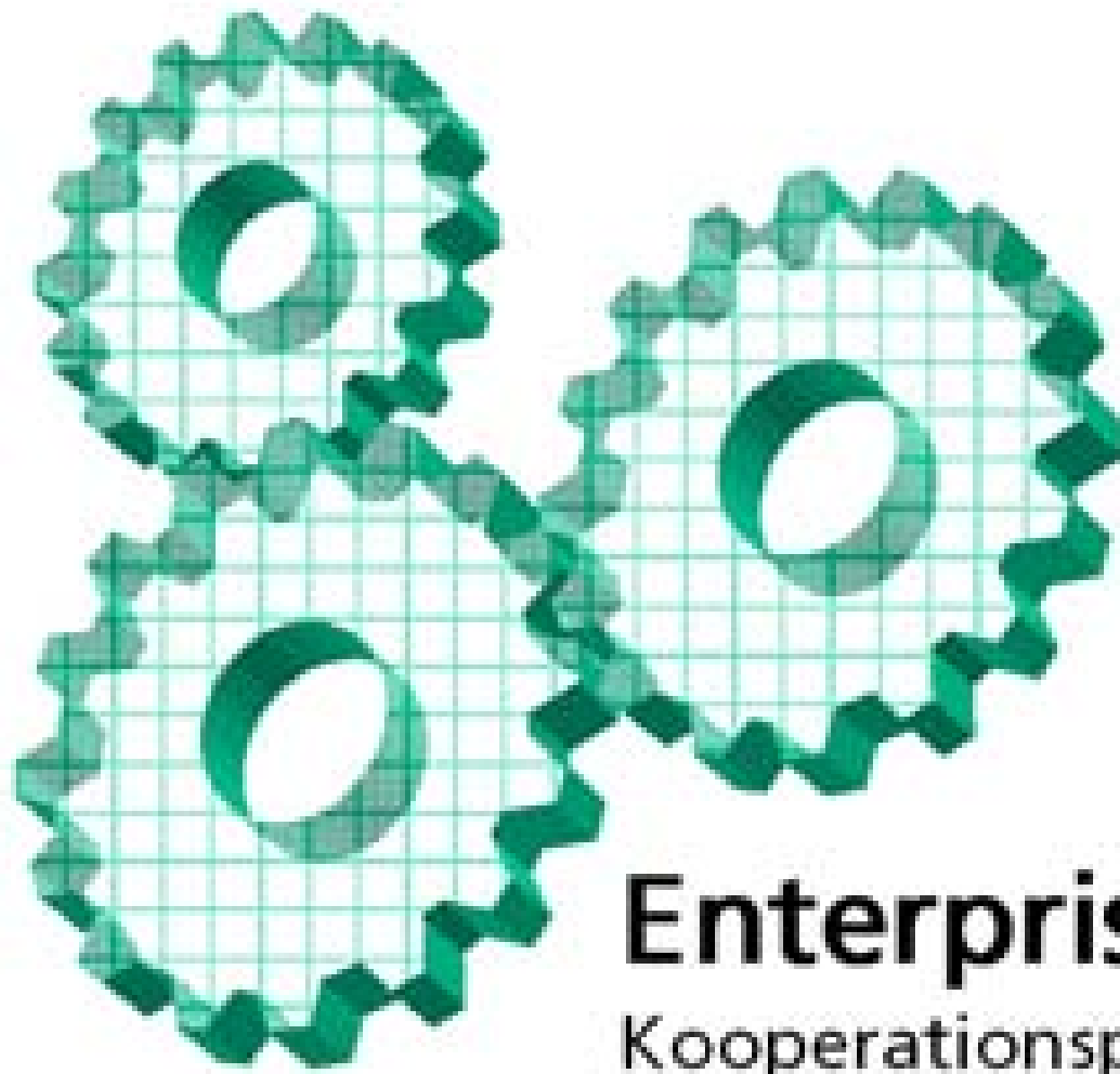
Grid Projects:

-EGEE

-Jawari

Projects I am not involved:

- EGEE
- Jawari

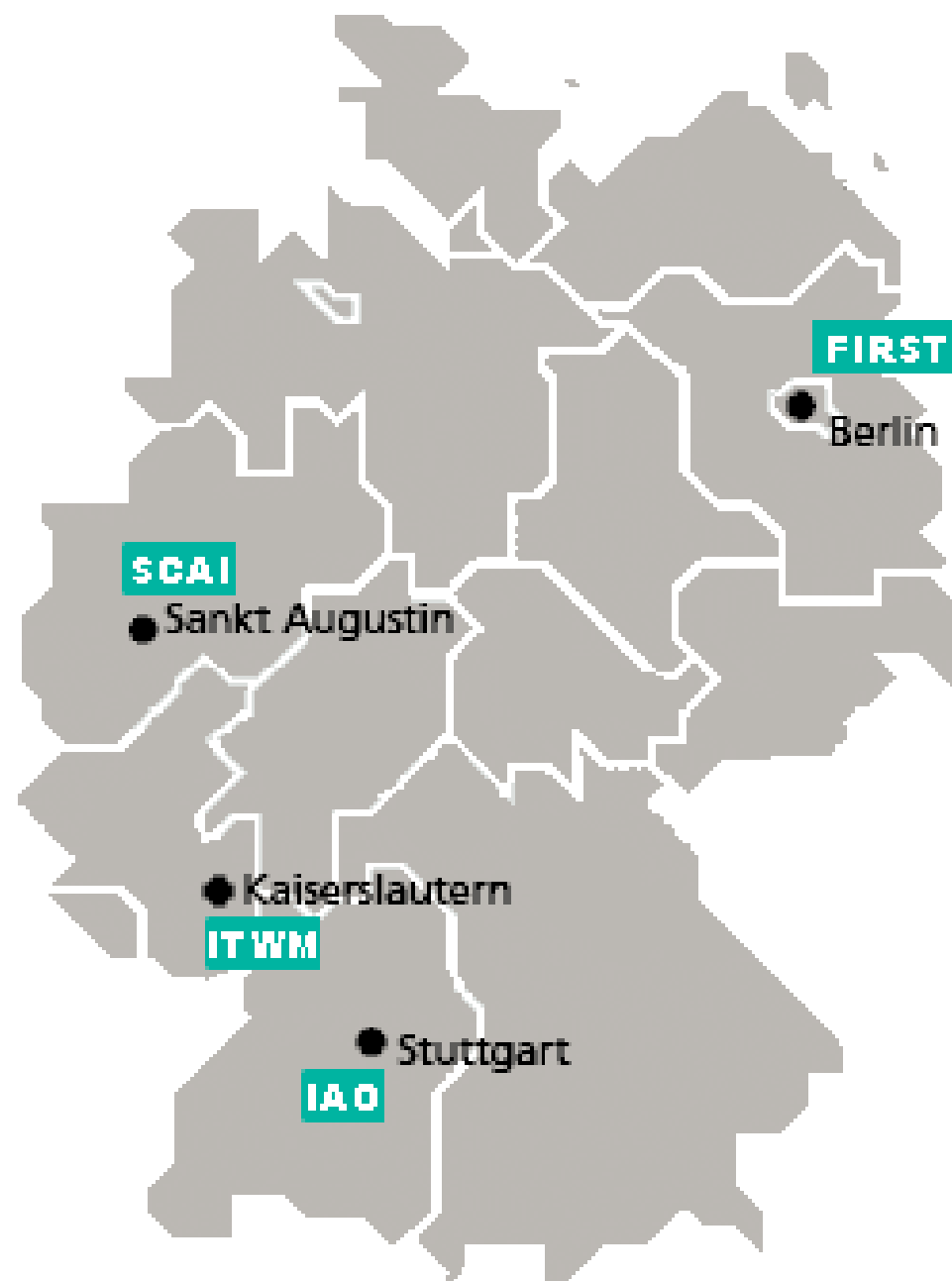


Enterprise Grids

Kooperationsprojekt WISA

A Fraunhofer-internal project to bring grid technology to industry users
– clear commercial focus

Ziel: Konkrete Wirtschaftserträge!



- Member Institutes:
- IAO: Portal, Einführungsprozesse
 - SCAI: MPCCI
 - ITWM: Operating, PHASTGrid
 - FIRST: Grid Workflow Management

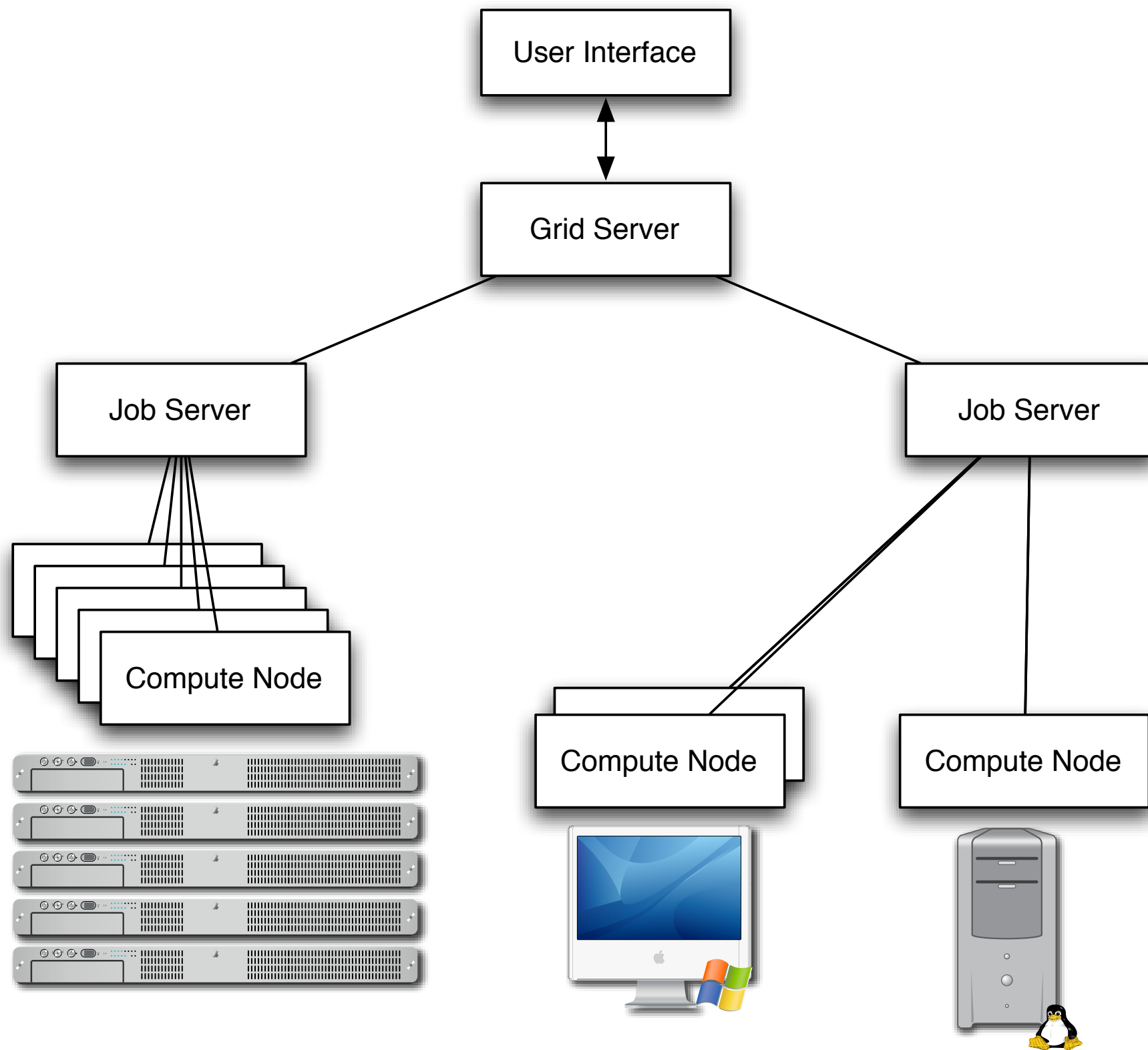


People at ITWM (beside me)
Tiberiu: PHASTGrid-Development
Kai: Deployment, Test infrastructure, support
Me: Coordination, Standardization our efforts (PHASTGrid, Calana, Xenbee)

Fraunhofer PHASTGrid



PHASTGrid as Execution-plane-product
– maybe depict architecture



Financial application



Trading room



Portfolio value
Portfolio risk



Trading System

Generates a collection
of jobs

Job Submission

Ergebnis

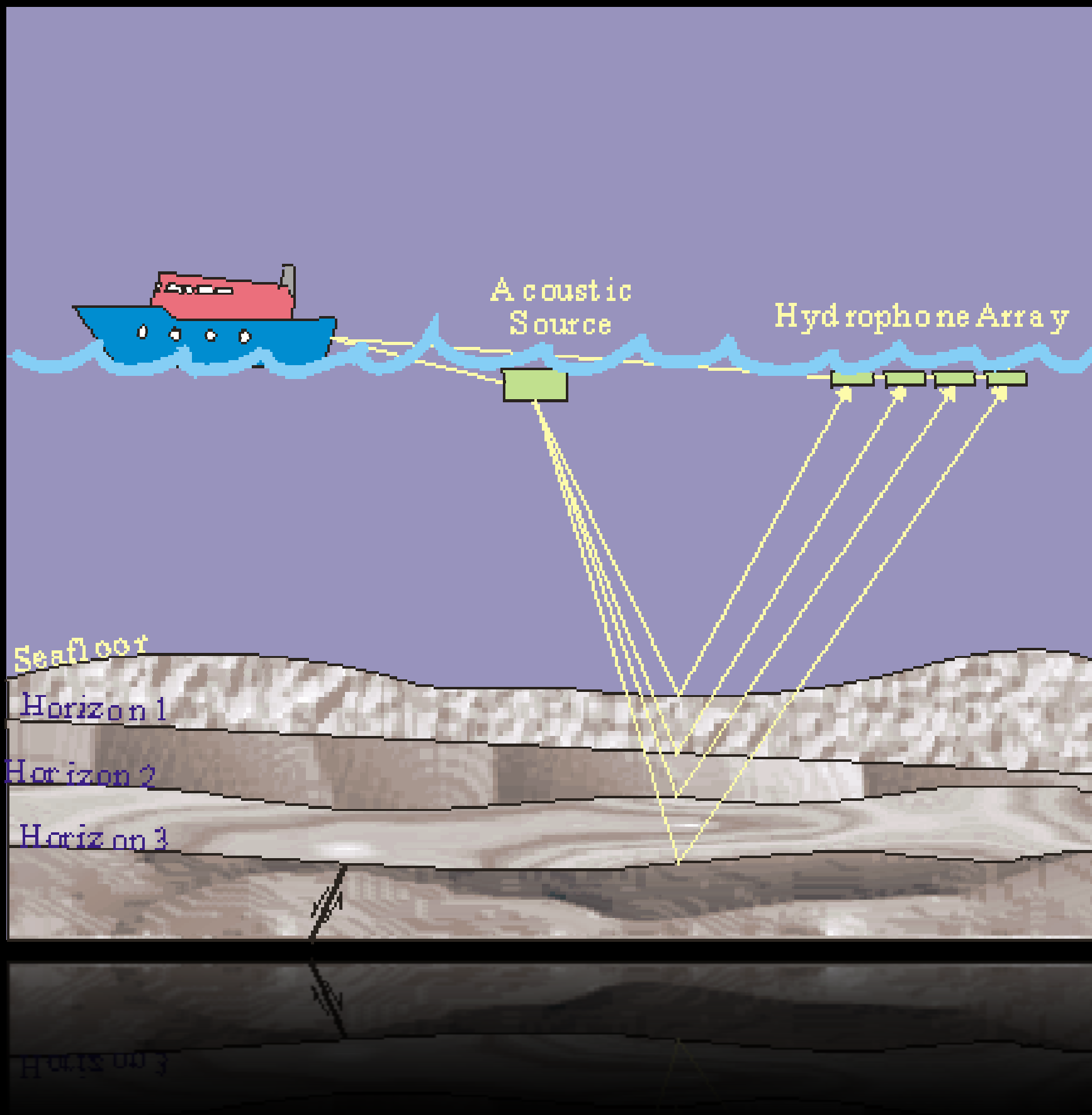
PHAST Grid System

Performance Prediction
Auto parallelization
Failure tolerant execution



More 3000 processes

- high throughput, 50 jobs/sec
- 24/7 on two datacenters for 3 years



Seismic migration processing

- Computations that take 6 Months on 6000 CPUs
- MPI-based parallel application
- Machine crashes unavoidable – checkpointing, ...
- > Integration of the algorithm into PHASTGrid, no MPI necessary.

PartnerGrid

15

German Project (approval pending) within D-Grid

3 Fraunhofer Institutes, DLR (NASA pendant)

2 Industry partners: Magma, GNS (FEM-Simulation INDEED)

- Fraunhofer SCAI, ITWM, IAO

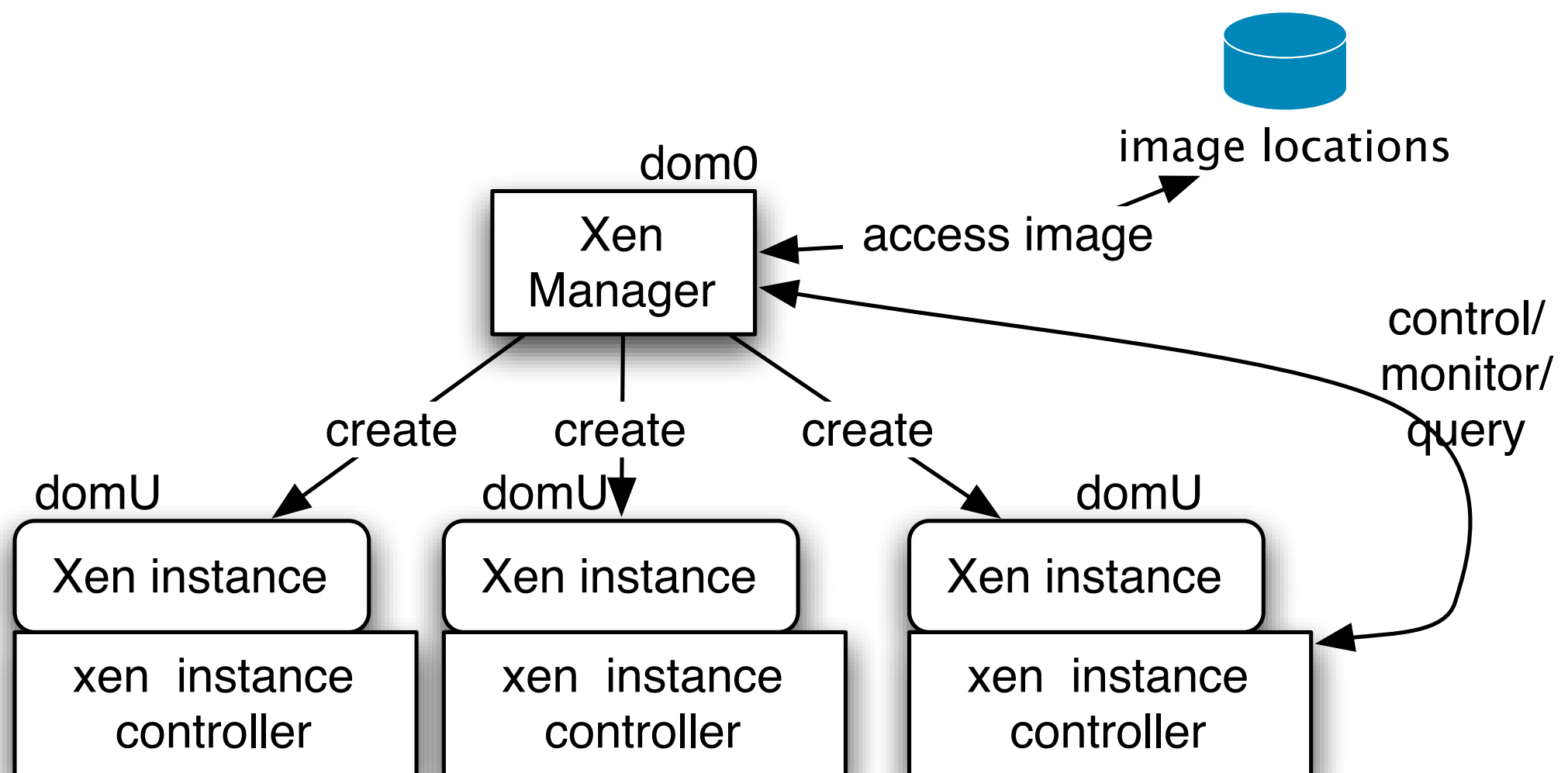
- DLR Simulations- und Softwaretechnik (DLR-Sistec)

- Industriepartner: Magma, Obermeyer (Bau), GNS (FEM-Software INDEED, Umformungsprozesse)

Using grid technology in industry settings

16

- e.g. Magma:
Simulation of metal in a mold
- Many SME customers of Magma let process their jobs at Magma
 - Magma wants to move those jobs to service providers



Part of our contribution:

Management of virtual machines based on Xen

- Application-images can be shipped completely with input data and license information to the service provider



- a virtual filesystem acts as a container
- as long as the kernel is suitable, there are no real dependencies on the target system
- easy to maintain: only a master image must be updated/certified



Secured Containers:

It is also possible to protect the containers using encryption.

- customer data is only accessible when the computation runs
- different provider levels: gold, silver, platinum, depending on the security a provider enforces

A full-page background image of red theater curtains. The curtains are closed, showing deep vertical folds and a rich, slightly dark red color. The lighting is even across the fabric.

Calana

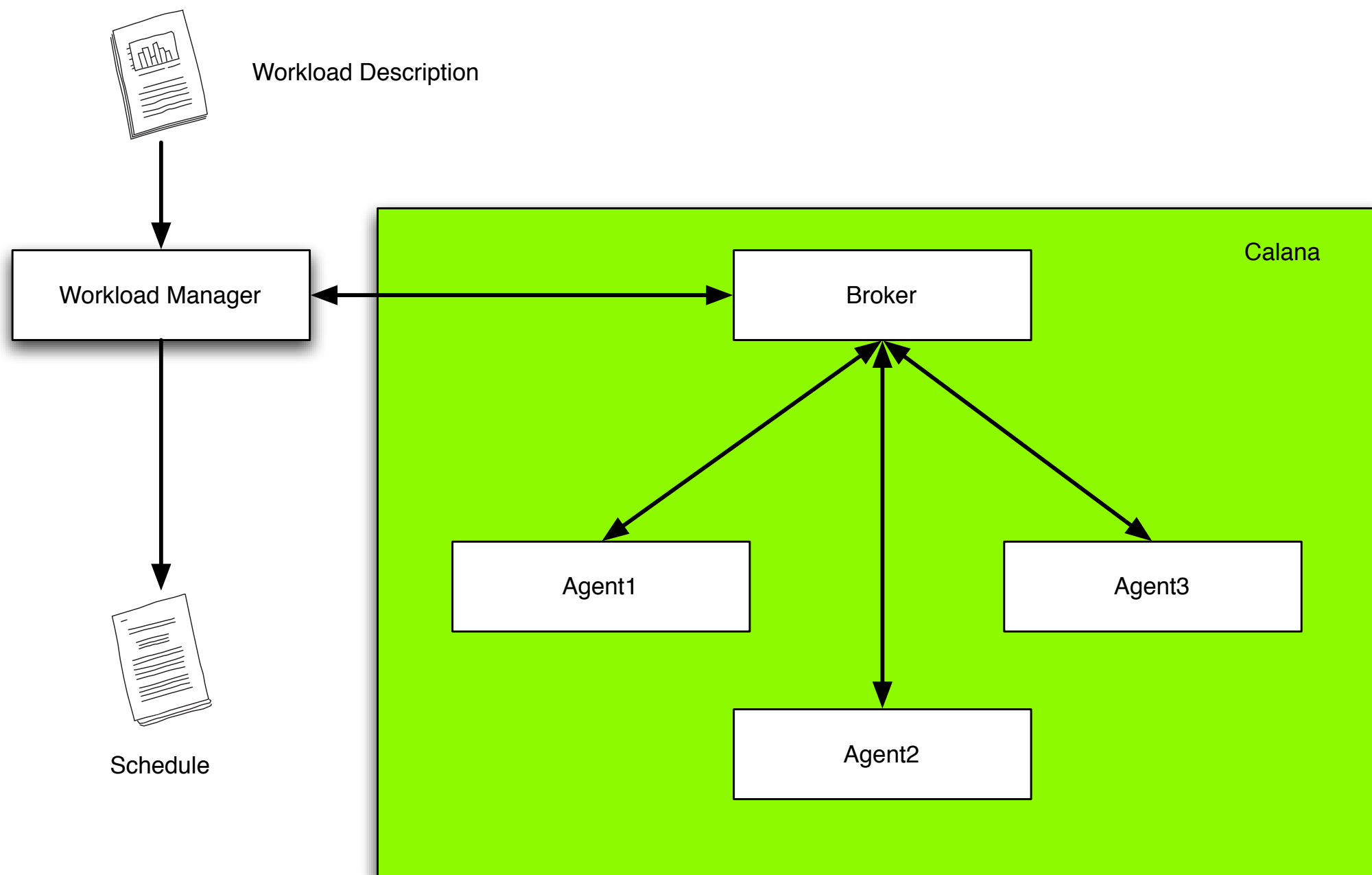
Today: Hierarchical Scheduler

21

Architecture of today's scheduling systems:
Mostly hierarchical. Resources propagate their usage in information systems and schedulers work on this information.
Typically no reservations etc in this setup, which makes it difficult to do coallocation.

Architektur:
Zentralisierte Scheduler, heute eher hierarchisch.
Ressourcen propagieren Auslastung in Informationssysteme, Scheduler arbeiten darauf.

Aber: Üblicherweise keine festen Zusagen (Reservierungen) in dieser Umgebung.

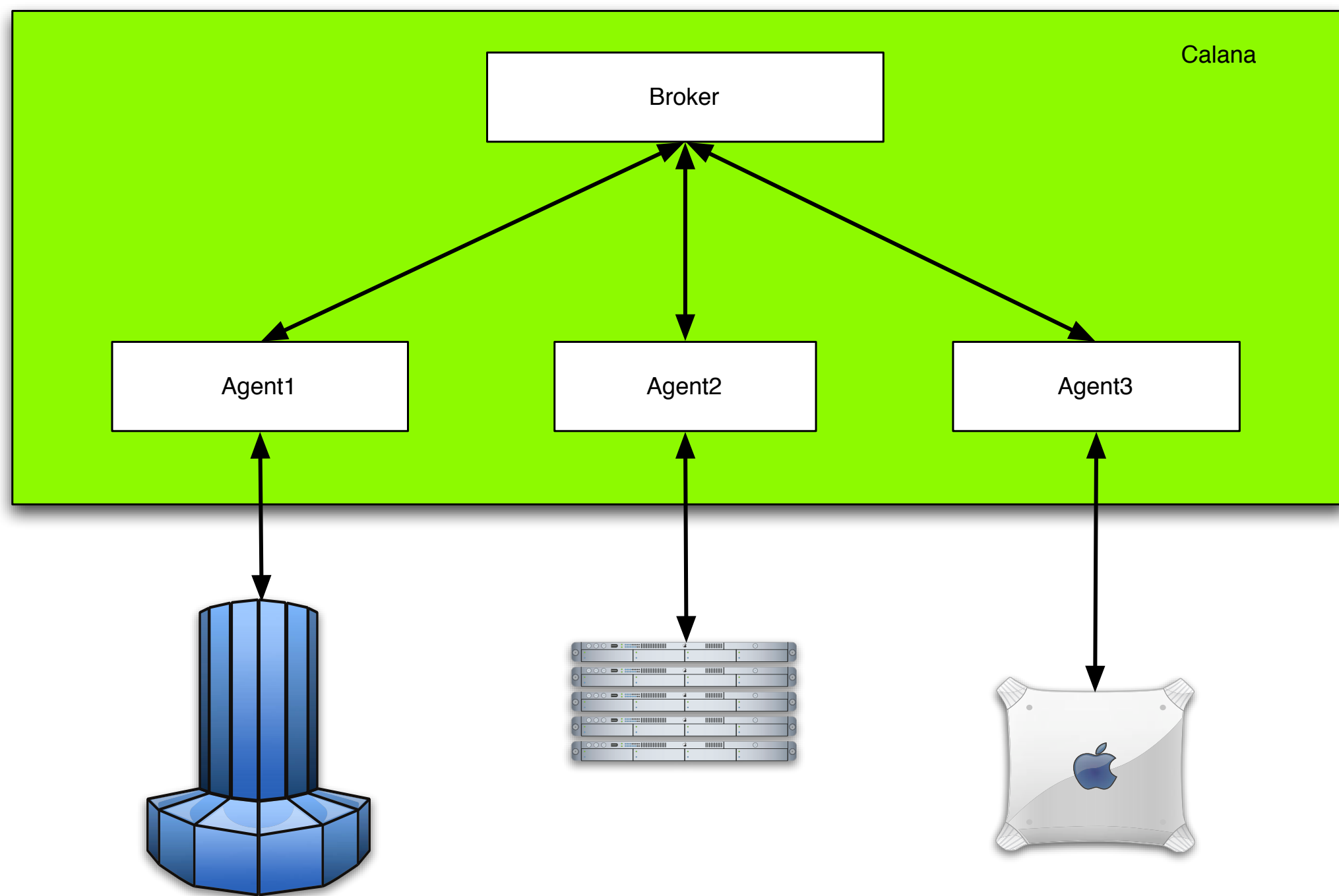


Setup Calana:

- (1) A workload manager reads the job description and tries to execute it.
- (2) Communication with Calana (rather complex)
- (3) A schedule is built and can be used.

Calana:

- (1) Ein Workloadmanager übernimmt eine Jobbeschreibung und versucht diese auszuführen.
- (2) Kommunikation mit Calana
- (3) Schedule entsteht.



Calana in more detail:
The broker communicates with the agents. The agents are placeholders for individual resources and provide an active abstraction layer.

Request comes in, the broker starts an auction. The agents consider the request and create bids on the execution of the job. The broker compares the bids and selects the best one.

*Sutherland '68:
"A futures market
in computer time"*

- > Simple daily auction for allocating slots on a PDP-1
- Other systems like Waldspurger et al, "Spawn", or "Popcorn"

Two Questions

- How does an agent decide to bid on the execution of a job?
- How to the user's preferences look like?



One user might want to have his jobs running *now*, as fast as possible. This might impose high resource usage costs (no optimization by the scheduler possible, “gaps” in the schedule)

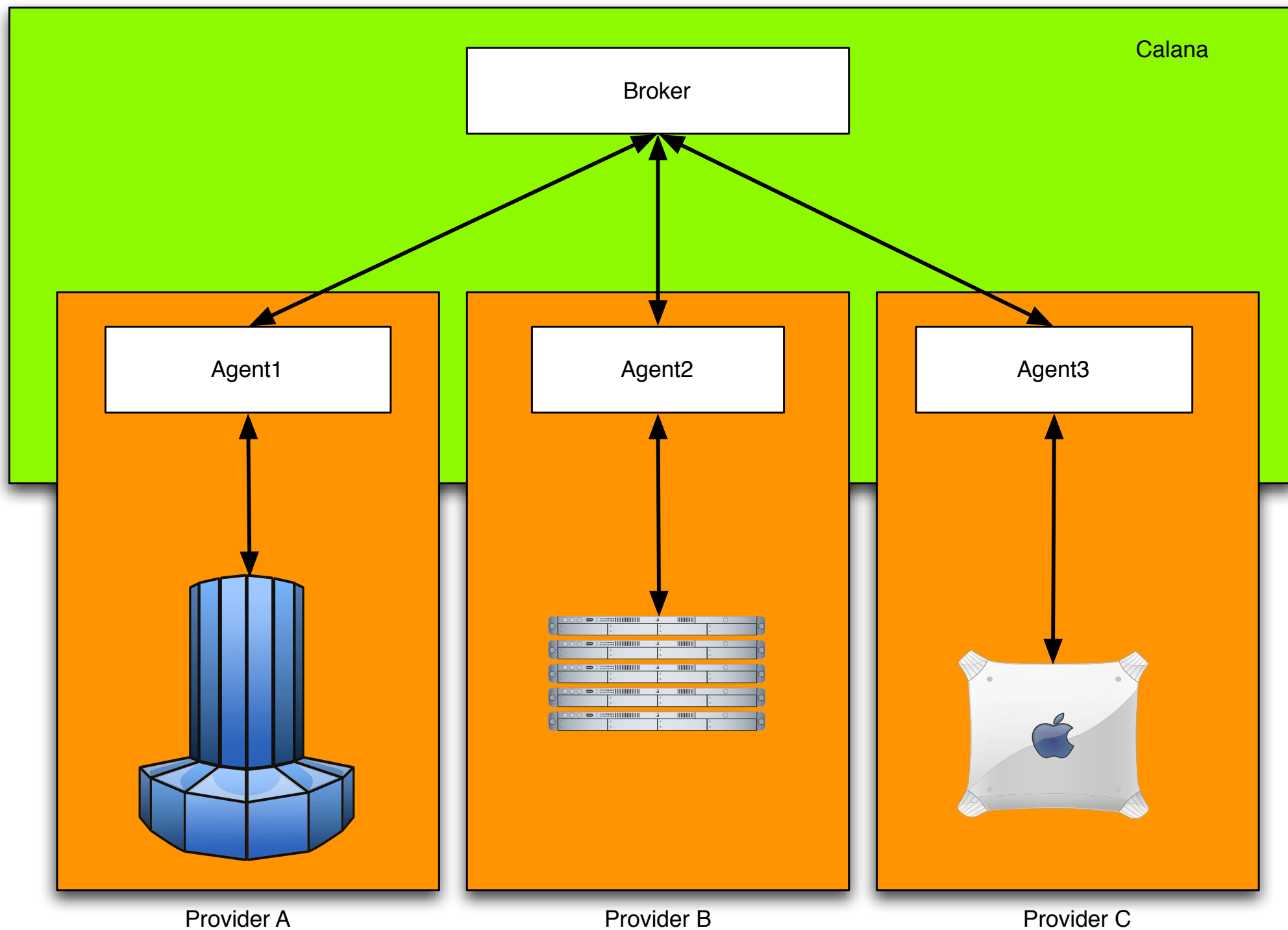
The other user might want to run her jobs as cheap as possible – speed is not the main criteria.

Compare to queue priorities in cluster schedulers.

$$v_i = f(p_i, t_i^f) = g \cdot \frac{p_i}{p_{max}} + (1 - g) \cdot \frac{t_i^f}{t_{max}^f}$$

So, a bid consists of a tuple (p, tf). How can the broker judge the bids? Compare the relative values of the bids.

g: The user's price preference.



The agents may implement any strategies, depending on their providers. The behaviour of the whole system depends on the behaviour of the agents.

To prevent fraud, the system must make the agents accountable for their actions. They need an incentive to behave correctly.

Those systems exist: The stock market is a system that works this way (somewhat successful).

Market Stability



29

collusion effects:

- information trade, syndicates
- > Countermeasure: Auction protocol to use (Vickrey-Auction)

fraud:

- > Allocation needs to be surveilled by the broker.
- > Rather complex communication protocol.

Phase 1: Find an allocation
(do an auction)

Phase 2: Survey the allocation

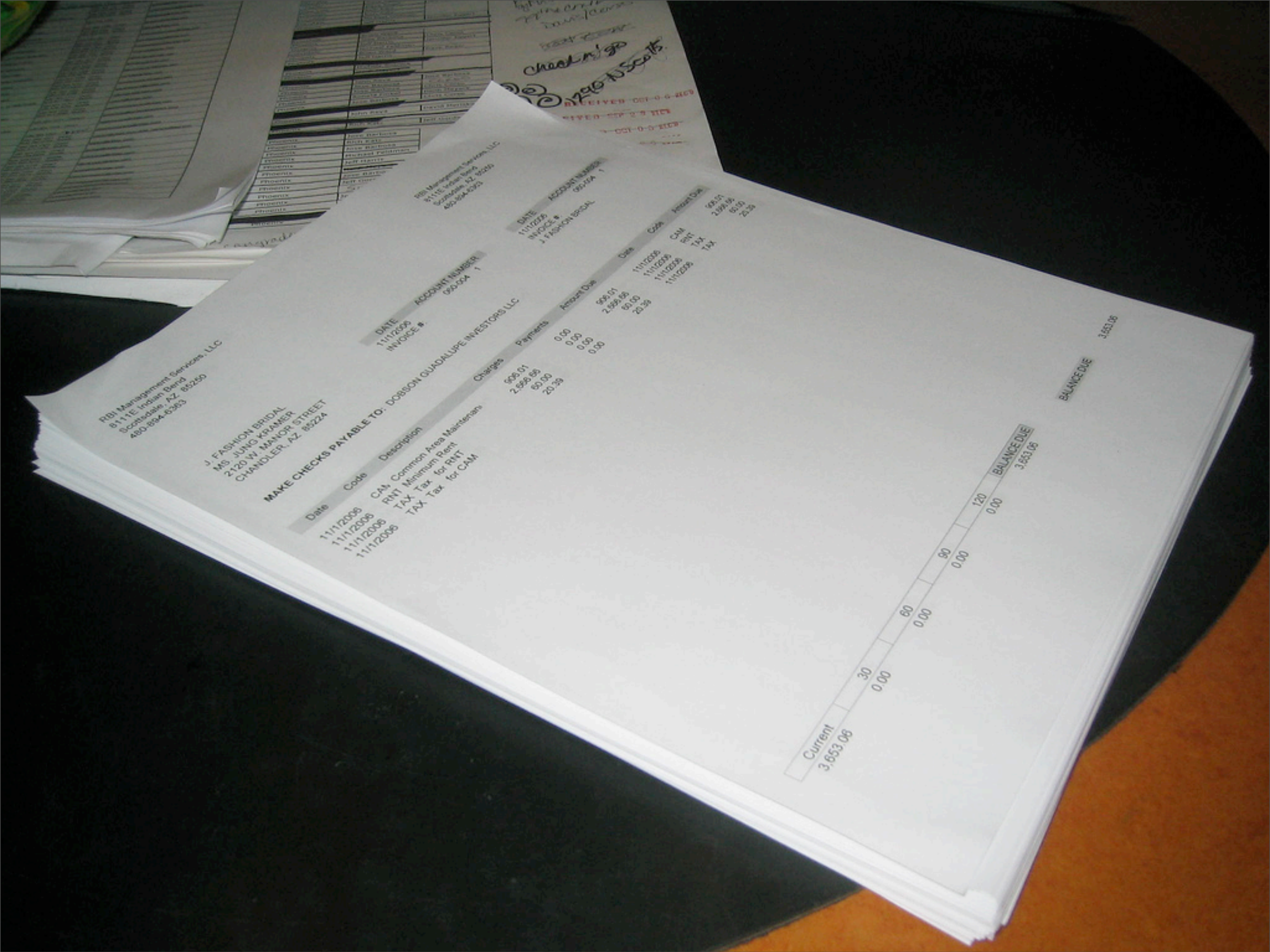
Phase 3: Account for the allocation

What I add to these mechanisms is accountability for actions:

scheduling needs to be more: its more transaction-based!

→ three phases

→ If one fails to fulfill a contract, a penalty payment is due. this is necessary for market stability.



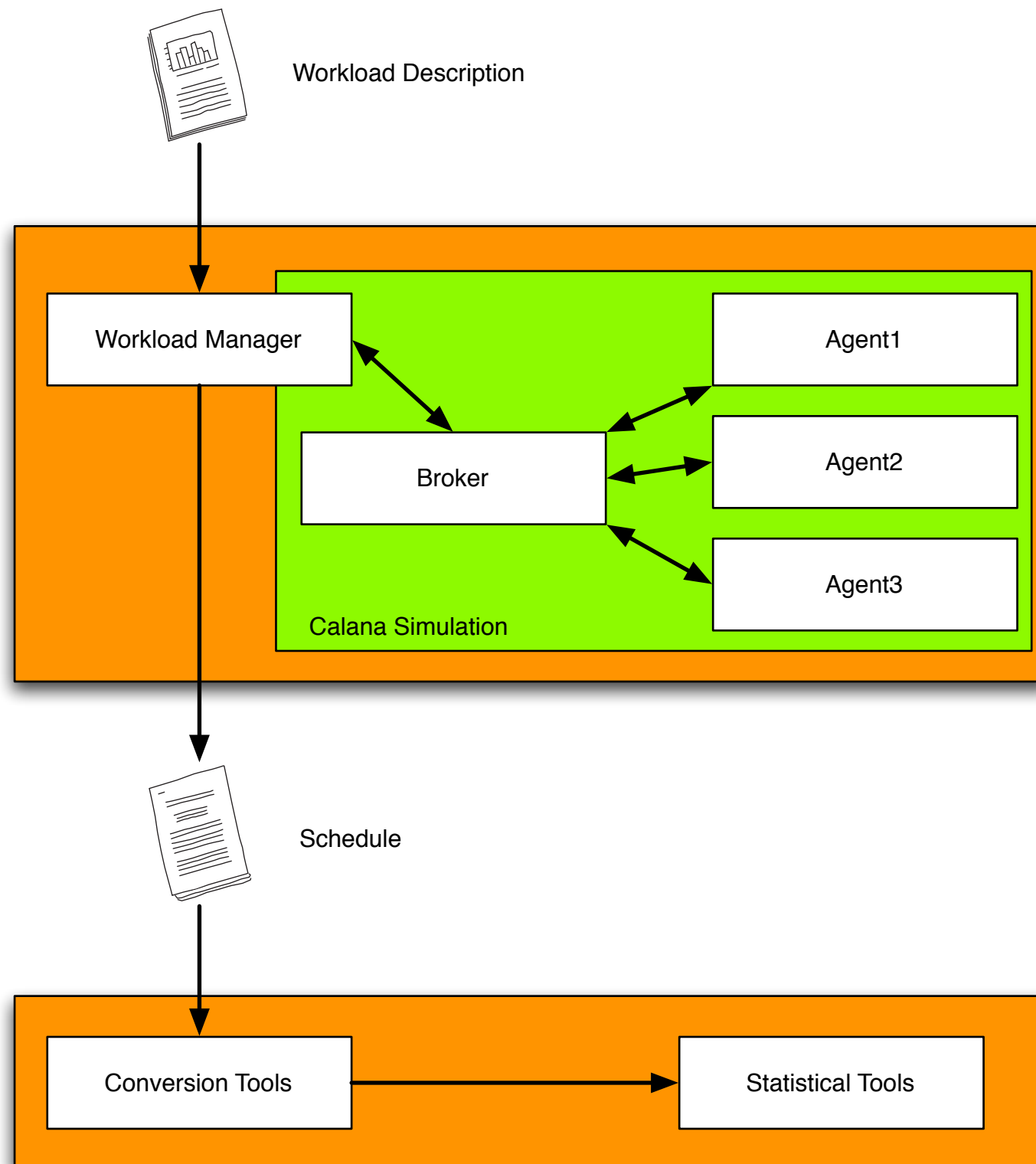
The broker knows all transactions and their results

- easy to do accounting
- billing easy since pricing is known



For commercial applications, it is important to the user to know who is processing the job

- for each job, there is a contract made
- this way, a user could prevent that sensitive data is processed at a not trustworthy provider



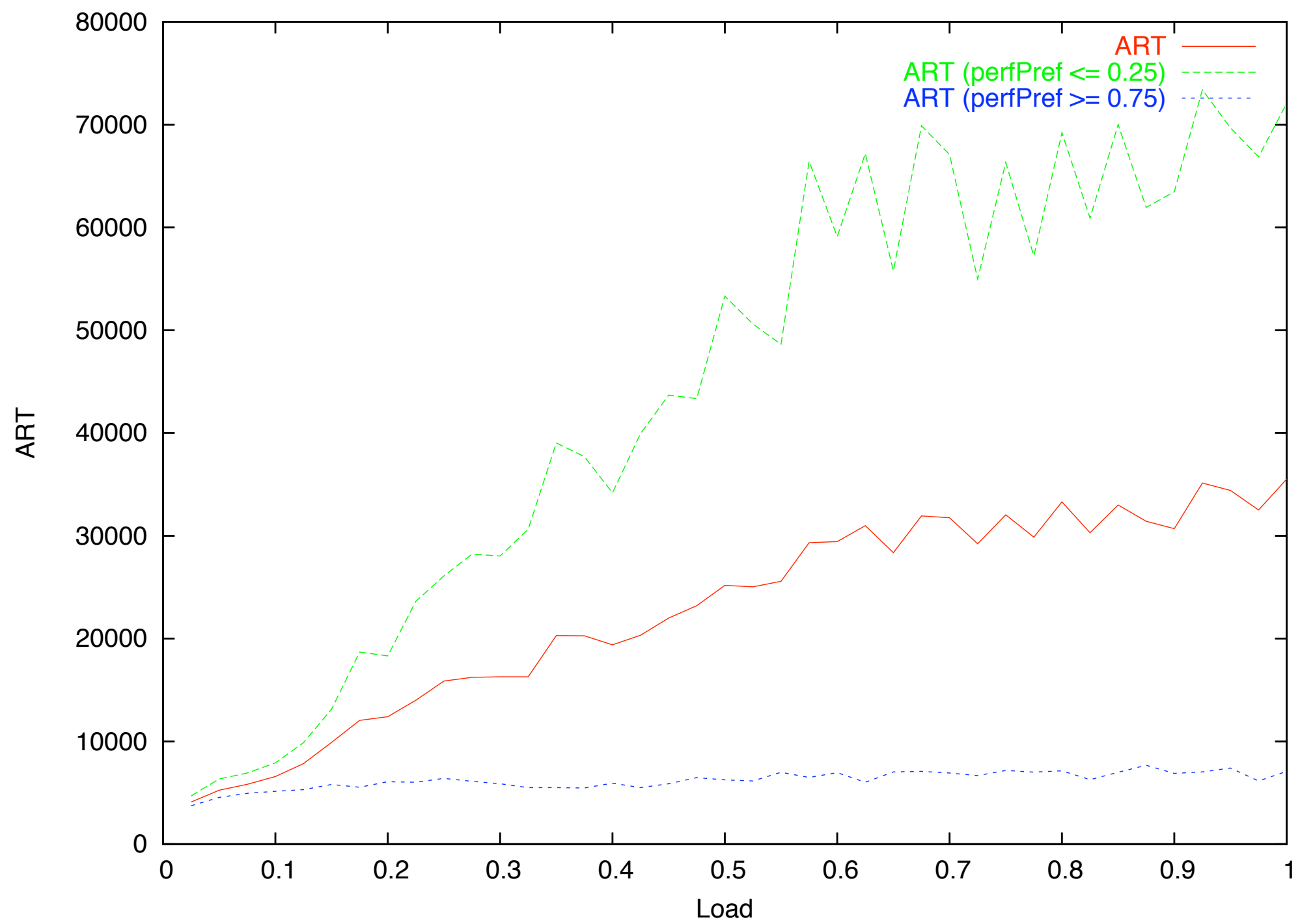
Simulation Toolchain

Problem: You need to simulate online (timing has a huge influence on scheduling results)
→ Alex implemented an eventbased simulation

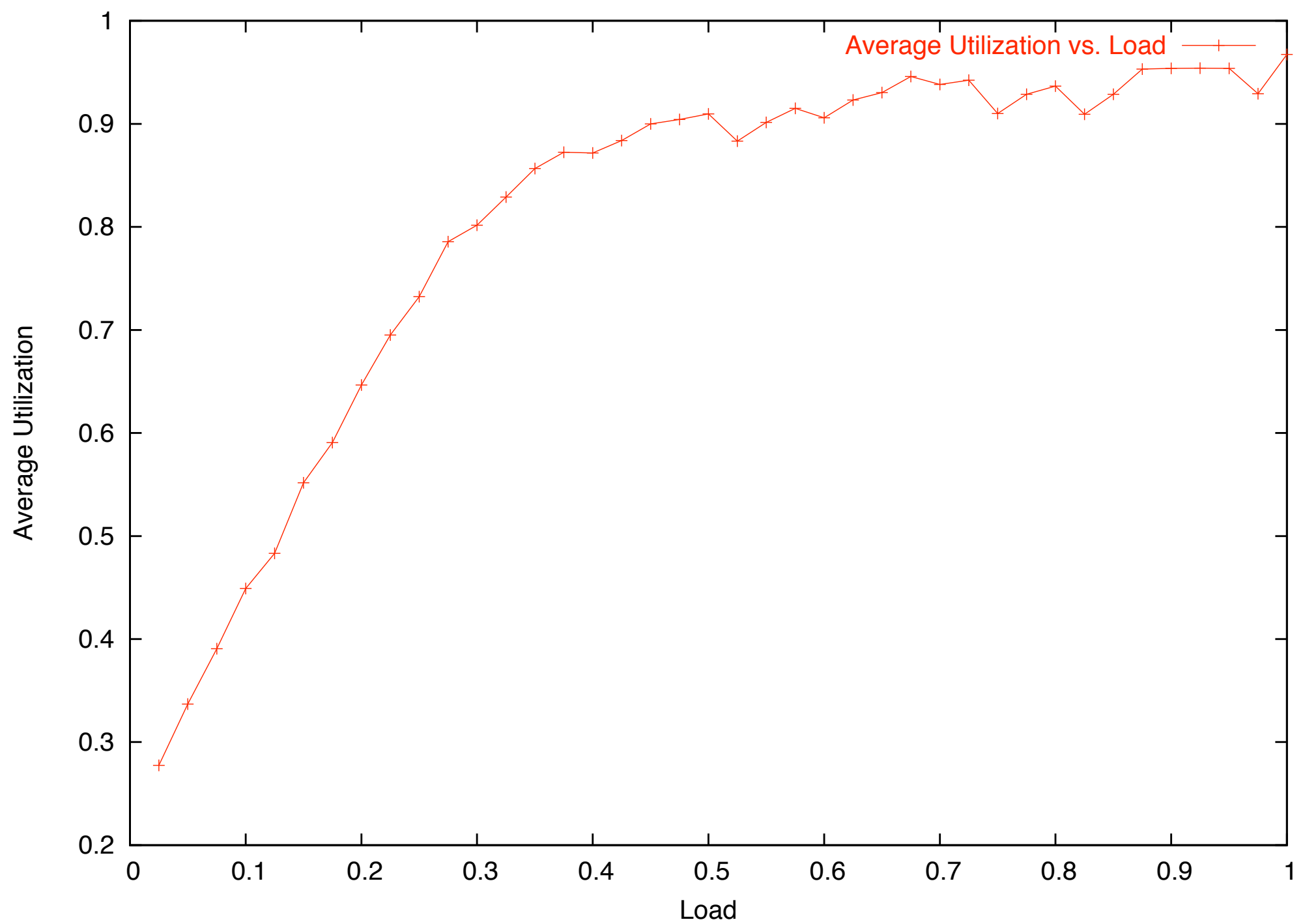
Problem: The workloads need to be scaled
→ developed a workload generator as well.

*Experiment 1:
50 Agents, random
preferences for
1000 users.*

50 agents in charge of one CPU each.
1000 users with random preferences,
the input workload consists of 10000 jobs.

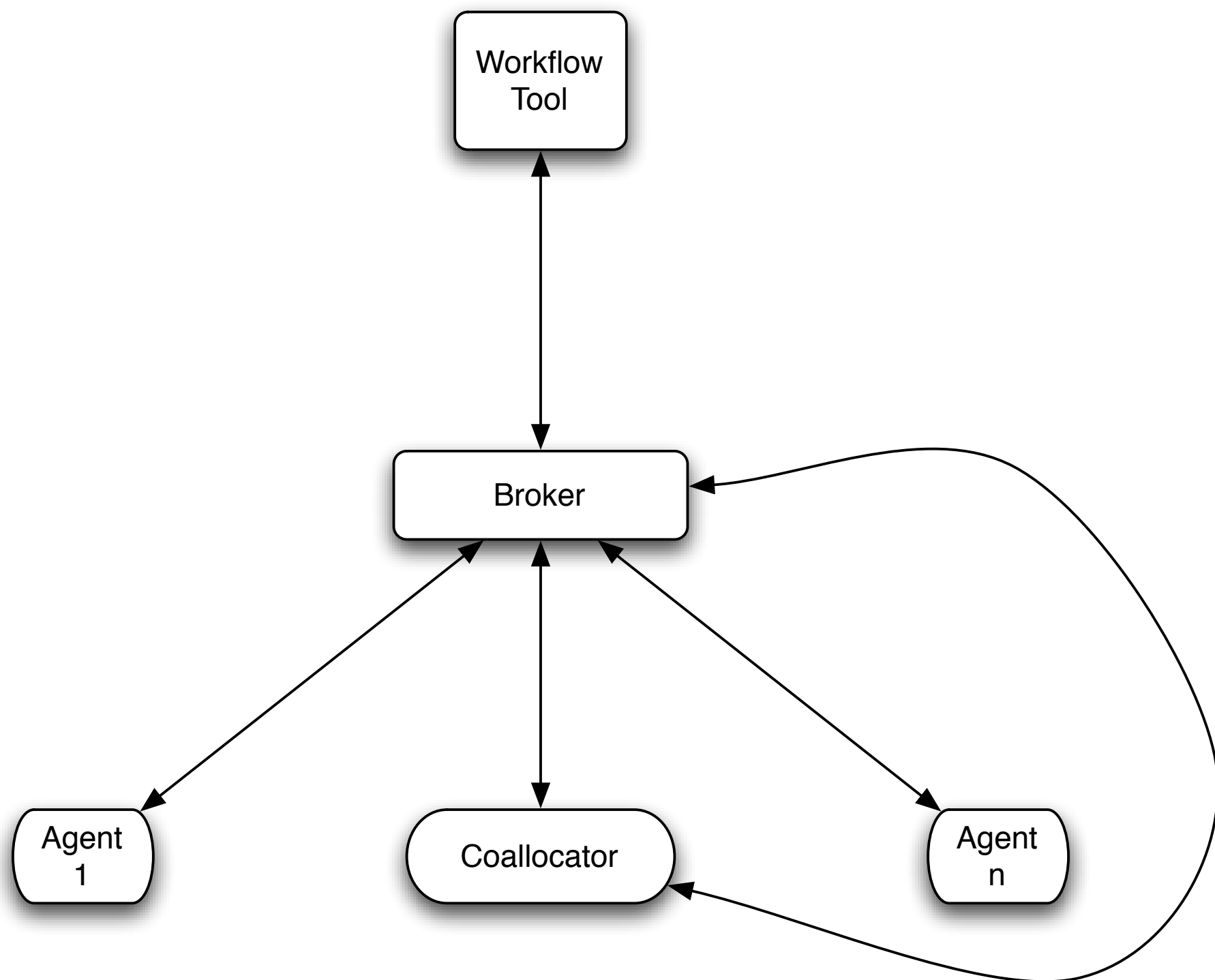


Blue: high performance preference \rightarrow the ART is almost constant
 Green: price preference dominates.

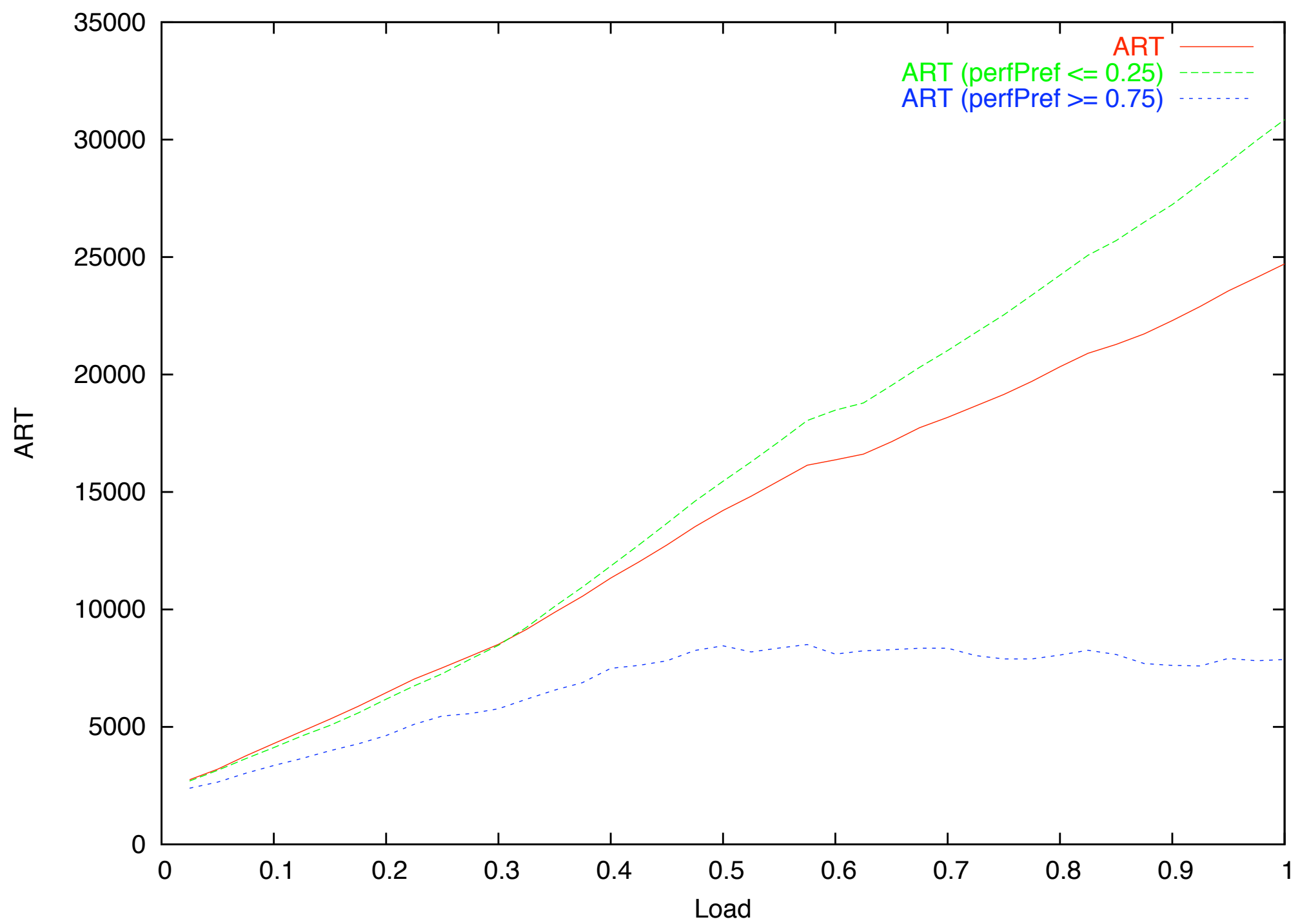


The system is almost saturated – workload contains idle times, so no full utilization possible.

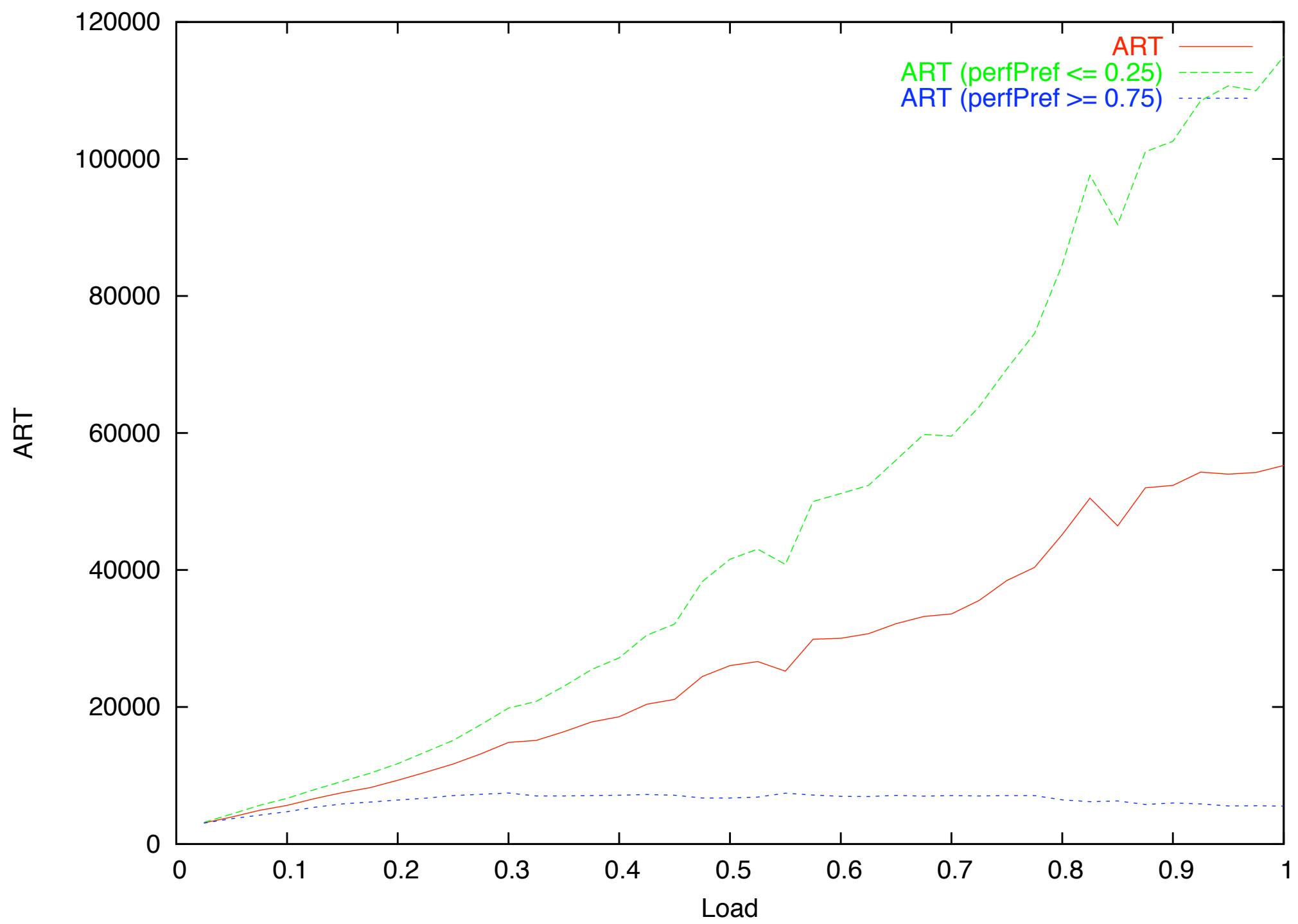
Experiment 2: Small Grid with Clusters, Coallocation



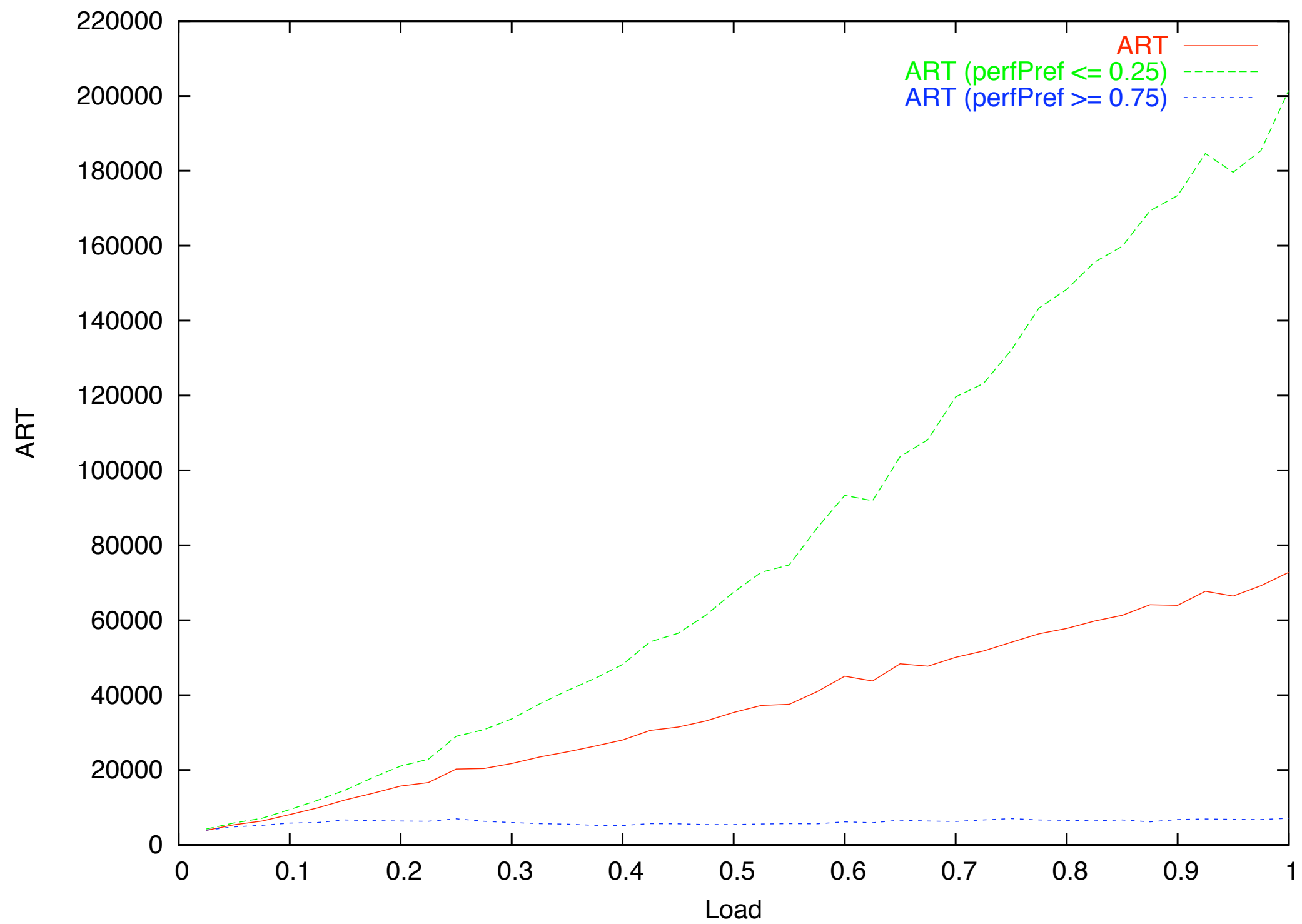
Coallocation can be done by a separate agent which aligns the bids of others. Auctions can be nested.
-> Very flexible.



reference: no coallocation here.



10% of the jobs ask for coallocation over several clusters



20 % coallocation jobs

Outlook (1)

- *Integrate Coallocation in the production code*
- *Add java messaging service layer*
- *Create agents for Xen installations*

Outlook (II)

- *Add a data management layer*
- *Investigate bidding strategies*
- *Change auction type*



Fraunhofer Institut
Techno- und
Wirtschaftsmathematik

Wirtschaftsmathematik
Techno- und

dalheimer@itwm.fhg.de