# Virtual Execution Environments and the Negotiation of SLAs in Grid Systems


**Matthias Hovestadt**
maho@cs.tu-berlin.de

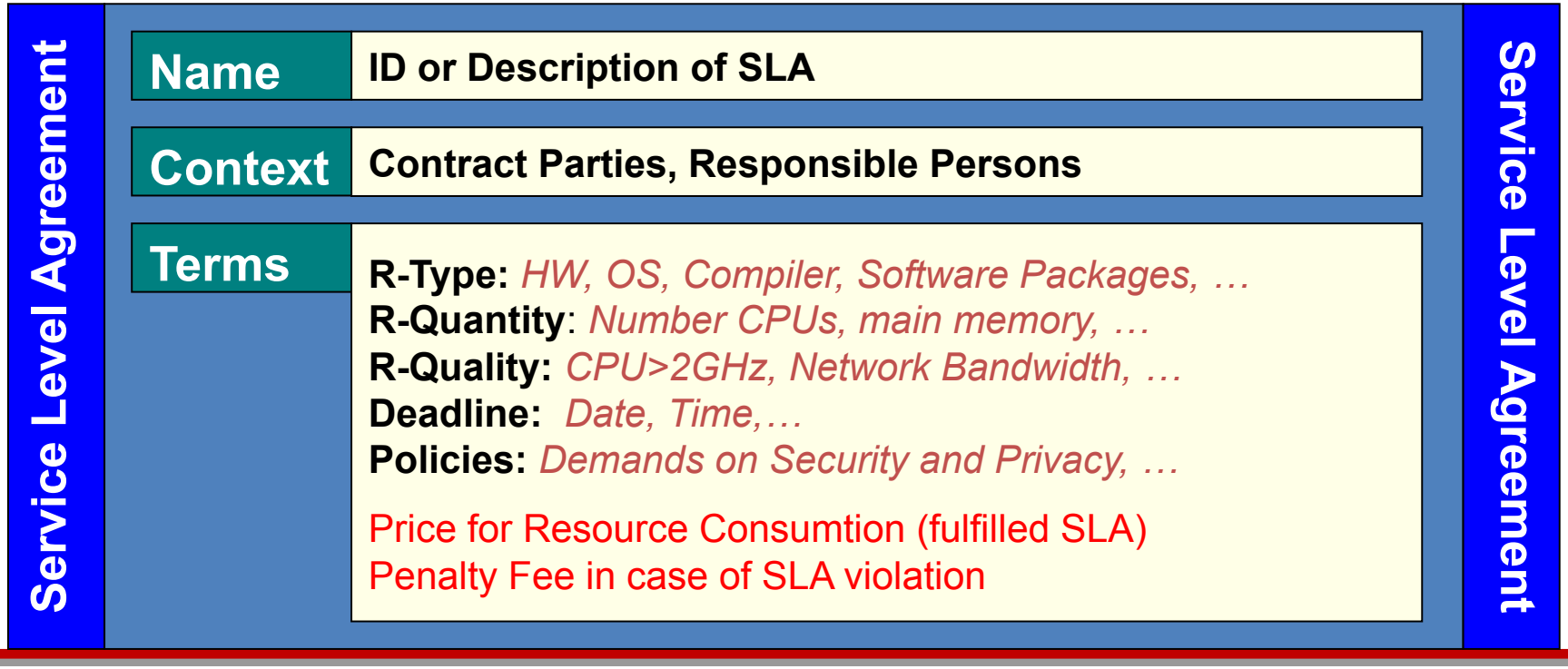
Technische Universität Berlin
Germany

# Grid Computing Today

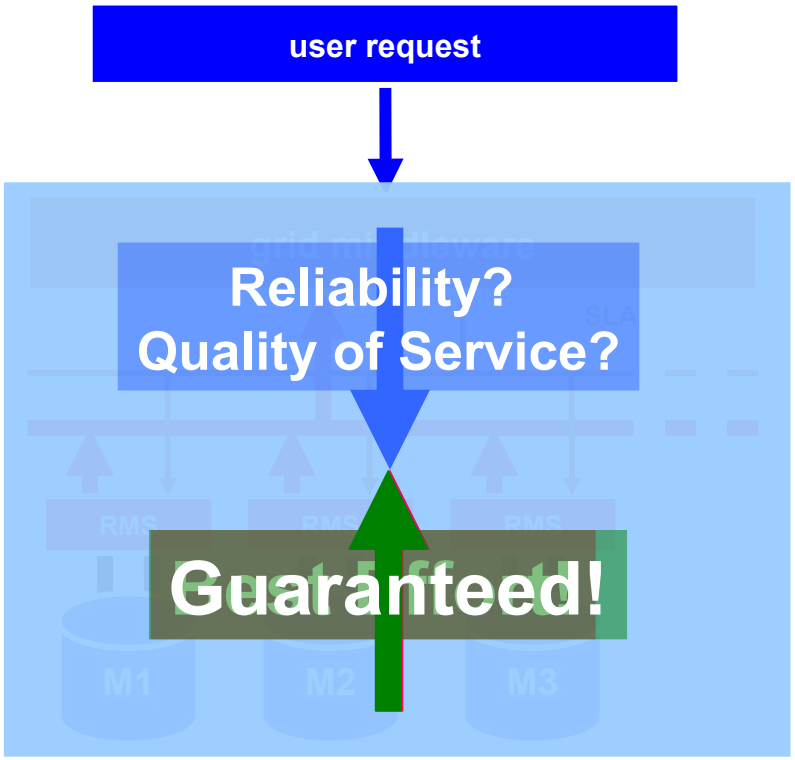
- How do Grids look like today?
  - Grids are in use, but…
    - … commercial usage is rare and limited
      - only isolated applications
    - … mostly used as a prototypic solution in research
      - testbeds within research projects
- Important problem
  - No contractually fixed QoS levels
    - Deadline bounded business critical jobs

# What is a Service Level Agreement

- Service Level Agreement (SLA)
  - Contract between provider and customer
    - Describes all obligations and expectations
  - Flexible formulation for each use case



# The Gap between RMS and the Grid

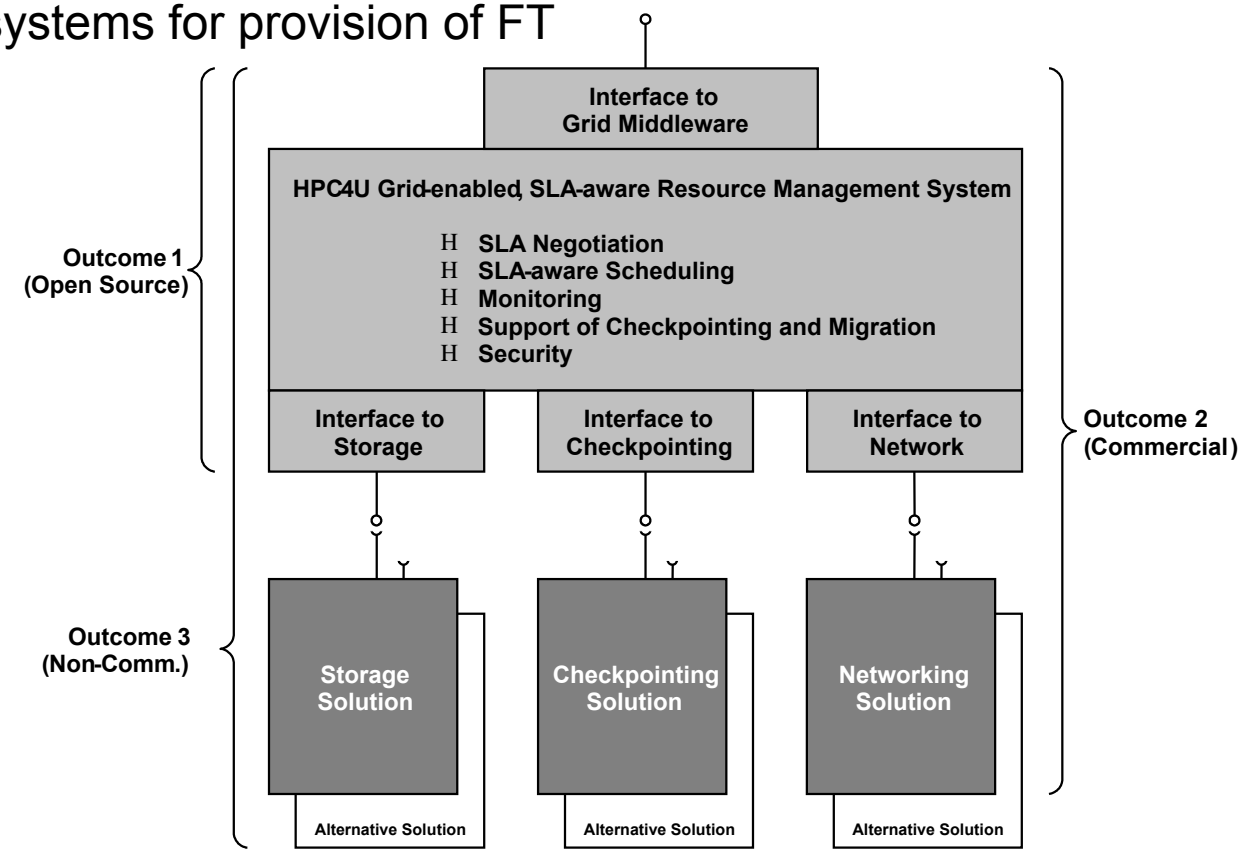



- User asks for Service Level Agreement
- Grid Middleware realizes job by means of local RMS systems
- BUT: These RMS only offer best effort!

- Goal: SLA-aware RMS
  - Runtime responsibility
  - Reliability
    - Fault tolerance

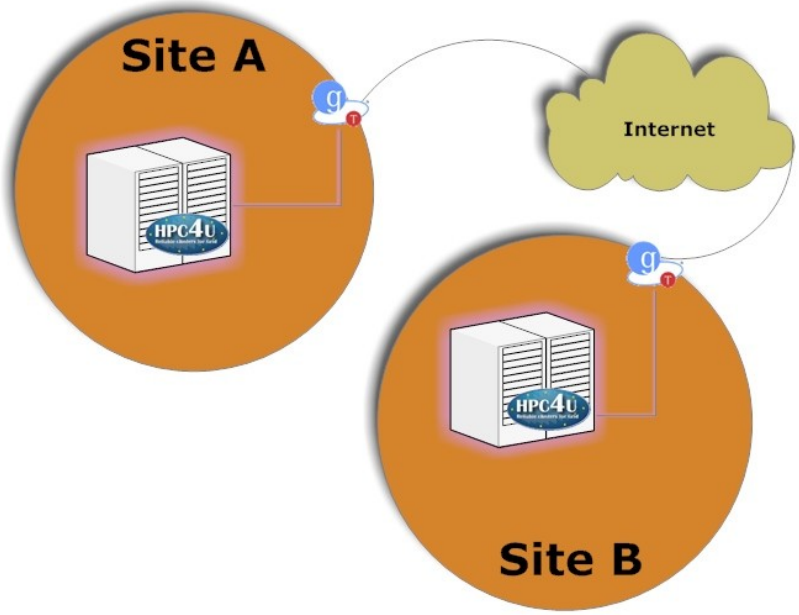
# SLA-aware RMS

- Central component
  - Interface to Grid middleware for SLA-Negotiation
  - Interfaces to Subsystems for provision of FT
- Tasks
  - SLA Negotiation
  - Policies
    - security, …
  - Monitoring
  - FT
    - checkpoints
    - migration
- Open interfaces



# Grid Migration

- Resource Outage at Site A
  - Cannot be compensated locally
- Compatibility
  - Kernel-level checkpointing
    - ♦ Demands on target resource
  - Grid are heterogeneous
- Compatibility profile
  - Fine-grained requirements of job
  - Part of SLA-negotiation
  - High chance of successful restart
  - But: small number of matching resources

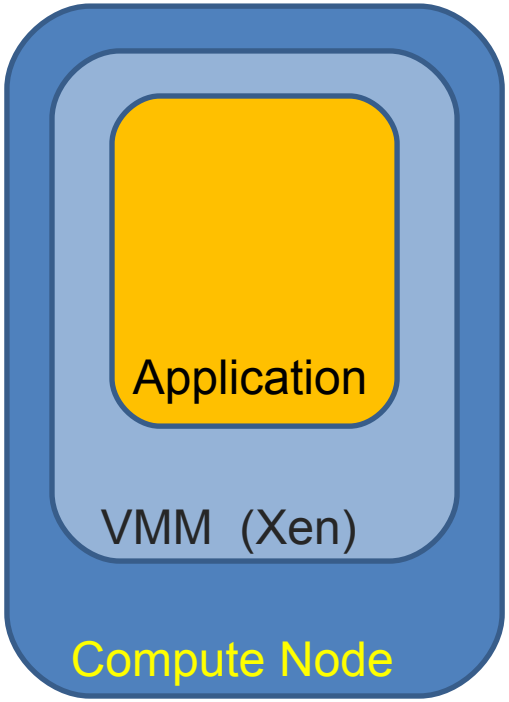


# Virtual Environments

- Status quo:
  Execution of application on compute node

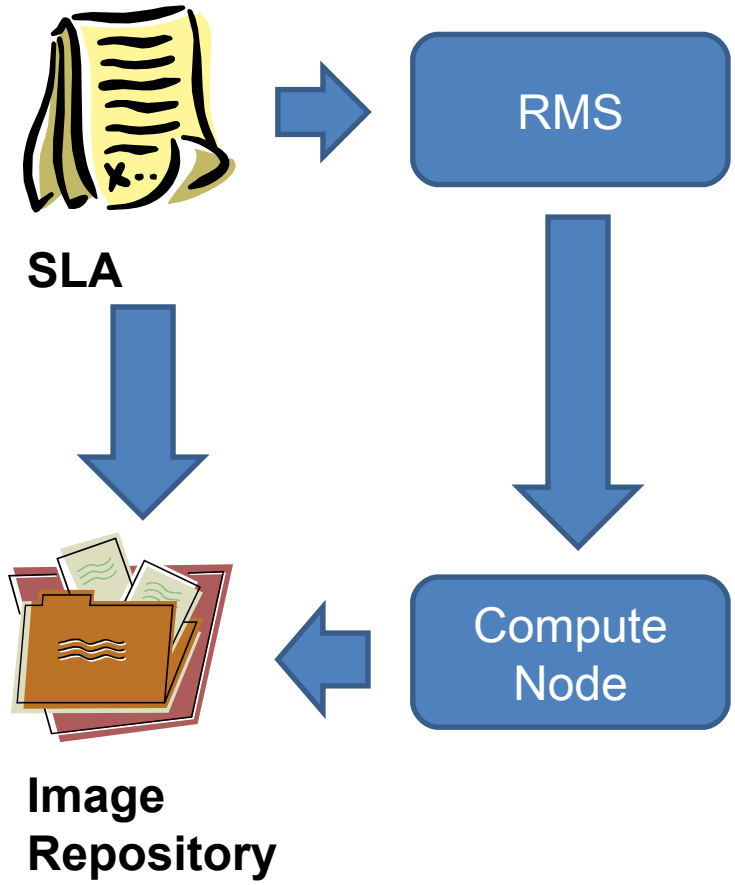
- Idea:
  Establishment of VMM on node
  - Startup of application within VMM

- VMM to match all requirements
  - Compatibility with job
  - Compatibility with migration dataset
  - Isolation of compute node from application



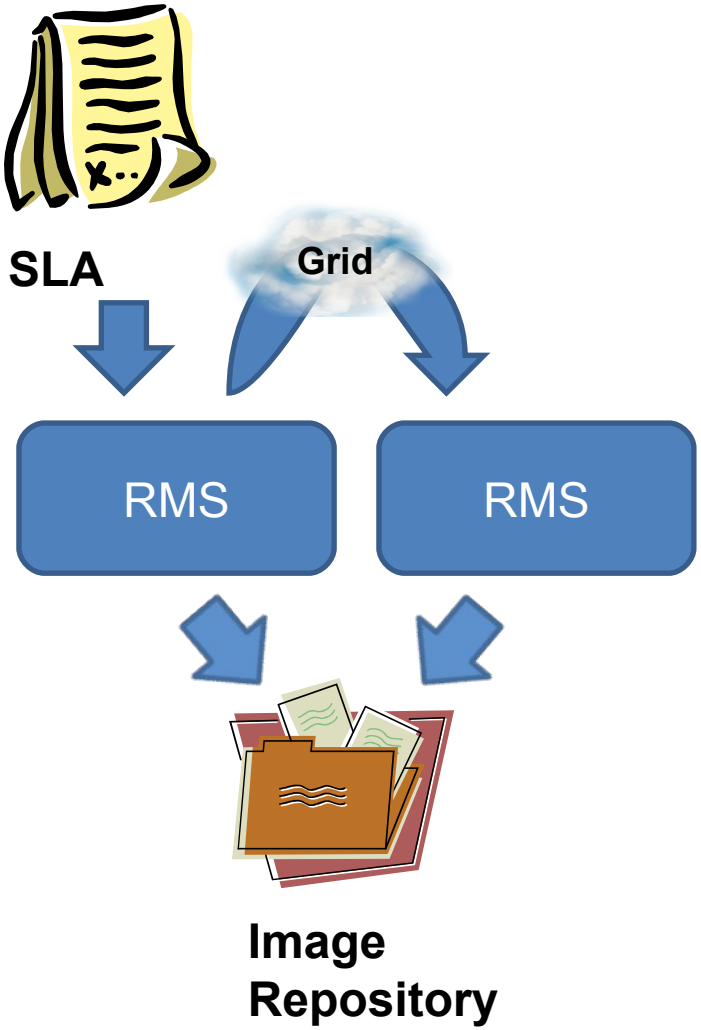
# Provision of Execution Environments

- Image repository with set of default system images
- SLA refers to system image as execution environment
- RMS initializes compute node
- Compute node establishes virtual environment



# Impact on Compatibility

- Source RMS provided execution environment according to SLA
- ID of system image is part of the SLA at migration time
- Remote RMS establishes same execution environment
- Compatibility at restart time



# Standard Negotiation Sequence

getTemplate

return Template

Fill in Template

Call is binding for the user

createAgreement

Check Availability

return SLA

# Enhanced Negotiation Sequence

getTemplate

return Template

AssessGrid enhancement

Fill in Template

getQuote

Quotes are not binding

Check Availability

Return quotes array

Select best offer

Call is binding for the user

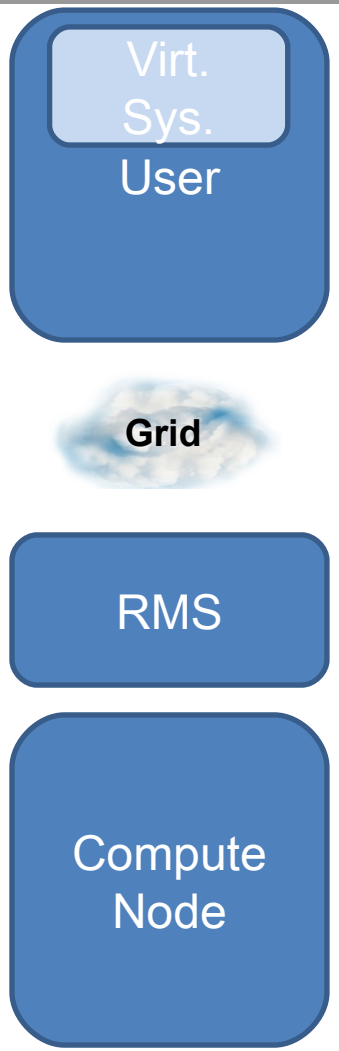
create Agreement

return SLA

Check Availability

# Next Step: Execution of Virtual Systems

- User provides virtual system
  - Ready configured to execute the user‘s application
- Virtual system is transferred to RMS
- RMS uses image for node initialization
  - Compute node establishes virtual system
  - Application starts in user-defined and user-provided environment



# Conclusion

- SLA-awareness in Resource Management Systems
  - Fault Tolerance for handling resource outages
  - Kernel-level checkpointing
  - Job Migration: Transfer of checkpoints to remote systems
- Virtual Execution Environments
  - Migration puts low-level demands on target system
  - Ensuring compatible environments on target system
  - Increasing number of potential migration targets
- Enhancement of SLA-Negotiation process
- Feel free to download the software stack:
  - http://www.openccs.eu, http://www.assessgrid.eu