# Grid & Virtualization Working Group

## OGF21 gridvirt-wg

Erol Bozak, Chair
SAP, Development Architect

Wolfgang Reichert, Co-Chair
IBM, Senior Technical Staff Member

October 2007
Seattle, WA

# OGF IPR Policies Apply

- "I acknowledge that participation in this meeting is subject to the OGF Intellectual Property Policy."
- Intellectual Property Notices Note Well: All statements related to the activities of the OGF and addressed to the OGF are subject to all provisions of Appendix B of GFD-C.1, which grants to the OGF and its participants certain licenses and rights in such statements. Such statements include verbal statements in OGF meetings, as well as written and electronic communications made at any time or place, which are addressed to:
  - the OGF plenary session,
  - any OGF working group or portion thereof,
  - the OGF Board of Directors, the GFSG, or any member thereof on behalf of the OGF,
  - the ADCOM, or any member thereof on behalf of the ADCOM,
  - any OGF mailing list, including any group list, or any other list functioning under OGF auspices,
  - the OGF Editor or the document authoring and review process
- Statements made outside of a OGF meeting, mailing list or other function, that are clearly not intended to be input to an OGF activity, group or function, are not subject to these provisions.
- Excerpt from Appendix B of GFD-C.1: "Where the OGF knows of rights, or claimed rights, the OGF secretariat shall attempt to obtain from the claimant of such rights, a written assurance that upon approval by the GFSG of the relevant OGF document(s), any party will be able to obtain the right to implement, use and distribute the technology or works when implementing, using or distributing technology based upon the specific specification(s) under openly specified, reasonable, non-discriminatory terms. The working group or research group proposing the use of the technology with respect to which the proprietary rights are claimed may assist the OGF secretariat in this effort. The results of this procedure shall not affect advancement of document, except that the GFSG may defer approval where a delay may facilitate the obtaining of such assurances. The results will, however, be recorded by the OGF Secretariat, and made available. The GFSG may also direct that a summary of the results be included in any GFD published containing the specification."

- OGF Intellectual Property Policies are adapted from the IETF Intellectual Property Policies that support the Internet Standards Process.

# Agenda

- Goals, Milestones & Status
- Recap of Previous Sessions
- Refine One Selected Use Case
- Working Streams & Next Steps

# Goals of the WG

1. Verification that within existing Grid standards the specifications are neutral to virtualized systems and resources
   - The request for “resources” may be satisfied either by “virtualized resources” or “physical resources”
   - Is virtualization “transparent to the current Grid standards”?

2. Explore how virtualization technologies can be exploited to better support Grid use cases
   - Define the use cases / scenarios wherein the Grid infrastructure is seen interacting with system virtualization platforms and making use of its capabilities
   - Exploit existing concepts and work, e.g.
     - System Virtualization, Partitioning and Clustering (SVPC) – (DMTF)
     - “Virtual Workspaces” (http://workspace.globus.org/papers/)

# Goals of the WG

3. Define the requirements to the Grid architecture for integration with system virtualization platforms

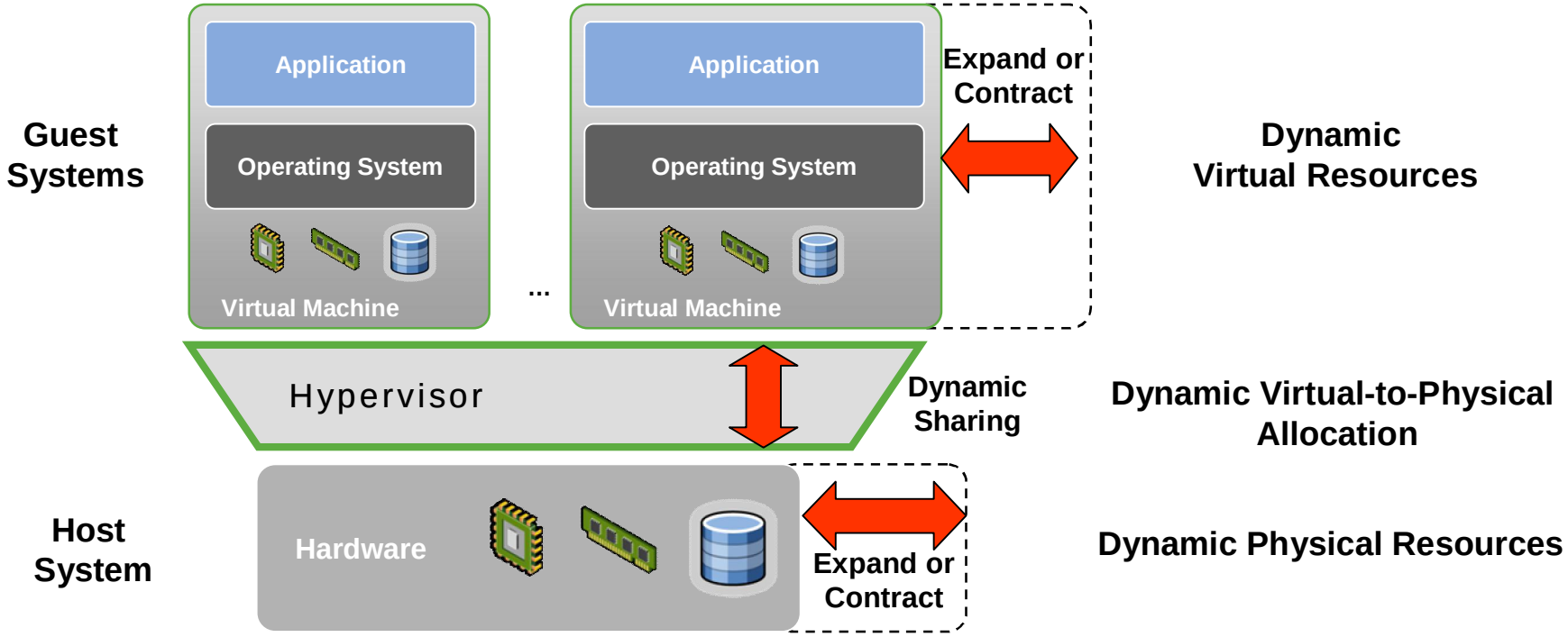
4. Define profile(s)
   - Align with existing standards:
     - DMTF - System Virtualization, Partitioning and Clustering (SVPC)
     - etc.

# Milestones and Status


- √ Milestone 1 (OGF 19)
  - Introduction of the workgroup
- √ Milestone 2 (OGF 20)
  - Terminology definition
  - Collection of use cases
  - Determine relations to other OGF WGs and SDOs
- Milestone 3 (OGF 21)
  - Requirements collection
  - Determine relation to other standards
  - First draft of a profile
- Milestone 4 (OGF 22)
  - Augmented profile for broader (external) review (e.g. DMTF)

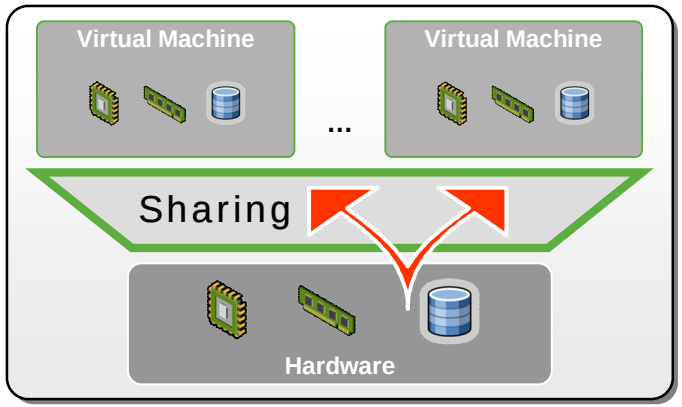
…

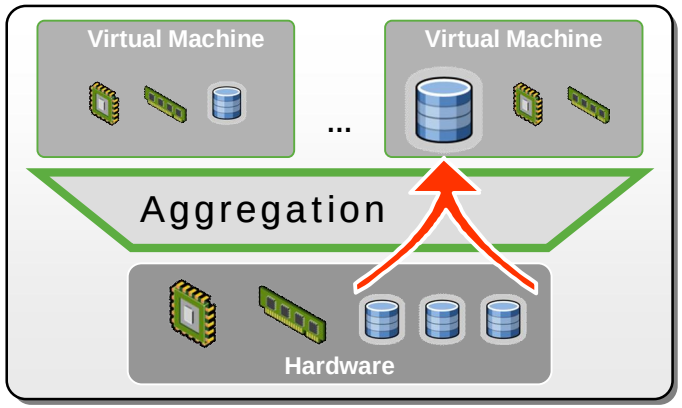
# Recap: Virtualization Concept

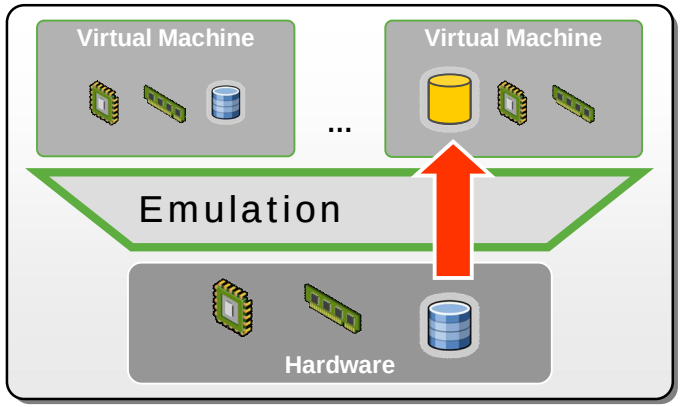


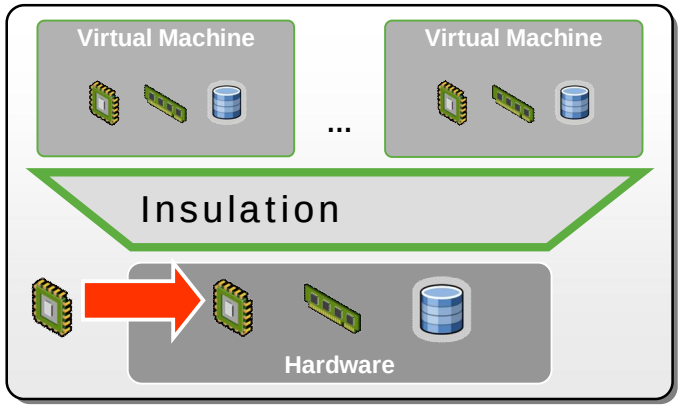

- Virtualization decouples presentation of resources to consumers (applications) from actual resources through a virtualization layer (Hypervisor)
- Several virtual machines (VMs) may run on a single physical host
- Each VM has its own installed operating system and applications

# Recap: Virtualization Concept



# Recap:
# Key Capabilities of Virtualization

- **Creation of virtual systems on-demand**
  - Specify the environment the application / jobs needs to run
  - The environment of the allocating can be pre-configured an persisted as images that can be activated on creation (multiple times if necessary)
- **Dynamic resizing**
  - Change the configuration of virtual system
- **Isolation**
  - Applications / jobs can run isolated from each other
- **Snapshotting**
  - Suspending the virtual system and persisting the state which can be reactivated again
- **Migration**
  - Movement of virtual system among host systems (physical systems)

# Recap: Scenario Examples

- „The Grid“ (Job / Execution Managers, Resource Managers etc.) does not only react on the „static“ configuration of the landscape → it may requests to create the proper environment for the applications / jobs
- The job manager may request that the application / job should run isolated because of security policies, to isolate the job malware etc.
- The application / job may require more resources to run → the Grid can dynamically adjust resource allocation (e.g. insulate more memory, CPU capacity, network bandwidth etc.)
- etc.

# Recap: Use Cases Overview

**Virtualization use cases**

- Power saving
- Planned maintenance
- Changing capacity requirements
- Changing capacity offering/availability
- Stateful cloning
- Protecting long running jobs from system failures
- Reproducing situations
- Metering of job resource consumption
- Resource consumption enforcement
- Protection against malware
- Ensuring Security
- Avoiding conflicts
- Emulating an environment for legacy jobs

**Virtualization capabilities**

Live migration

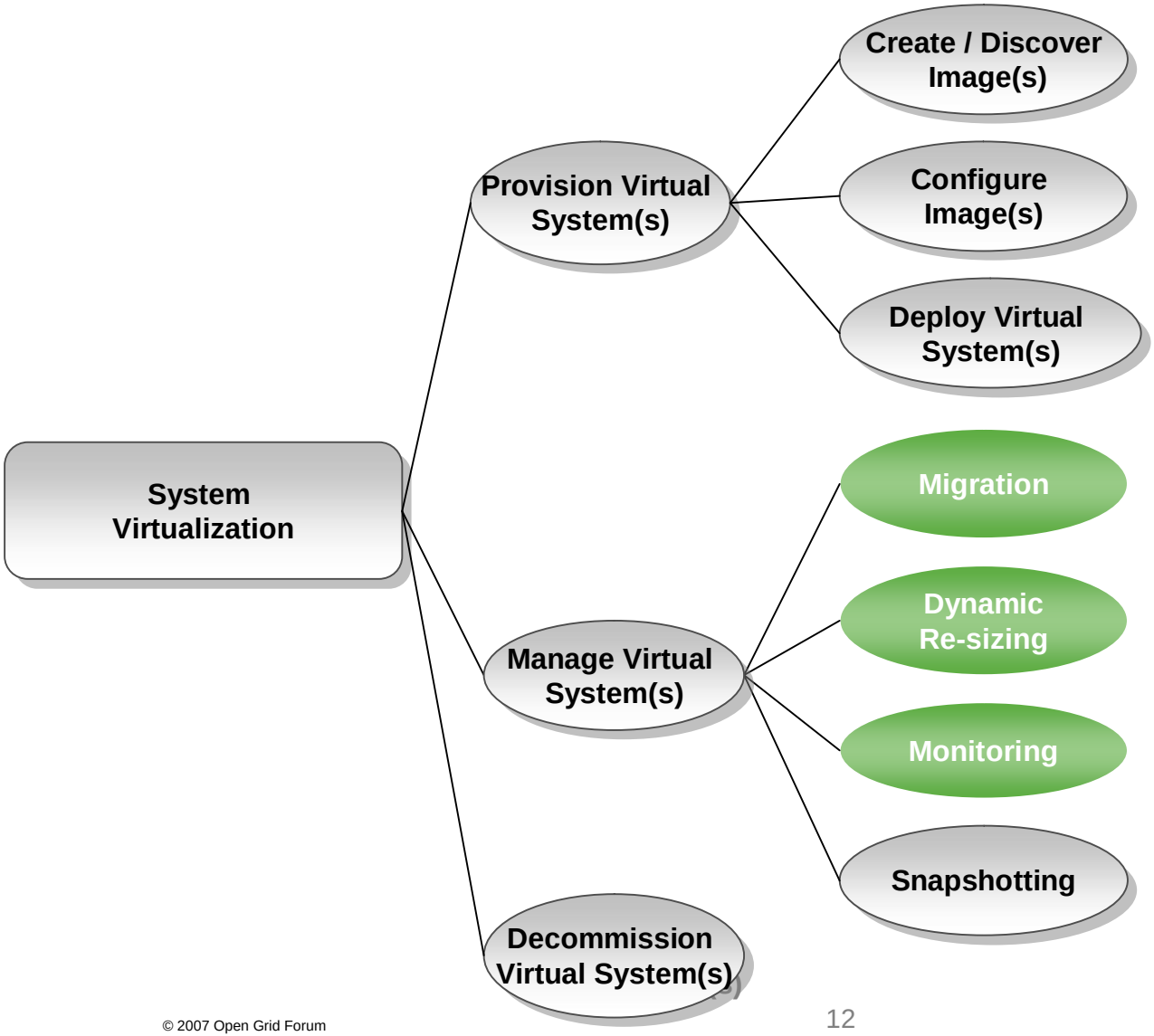
Dynamic resizing

Snapshotting

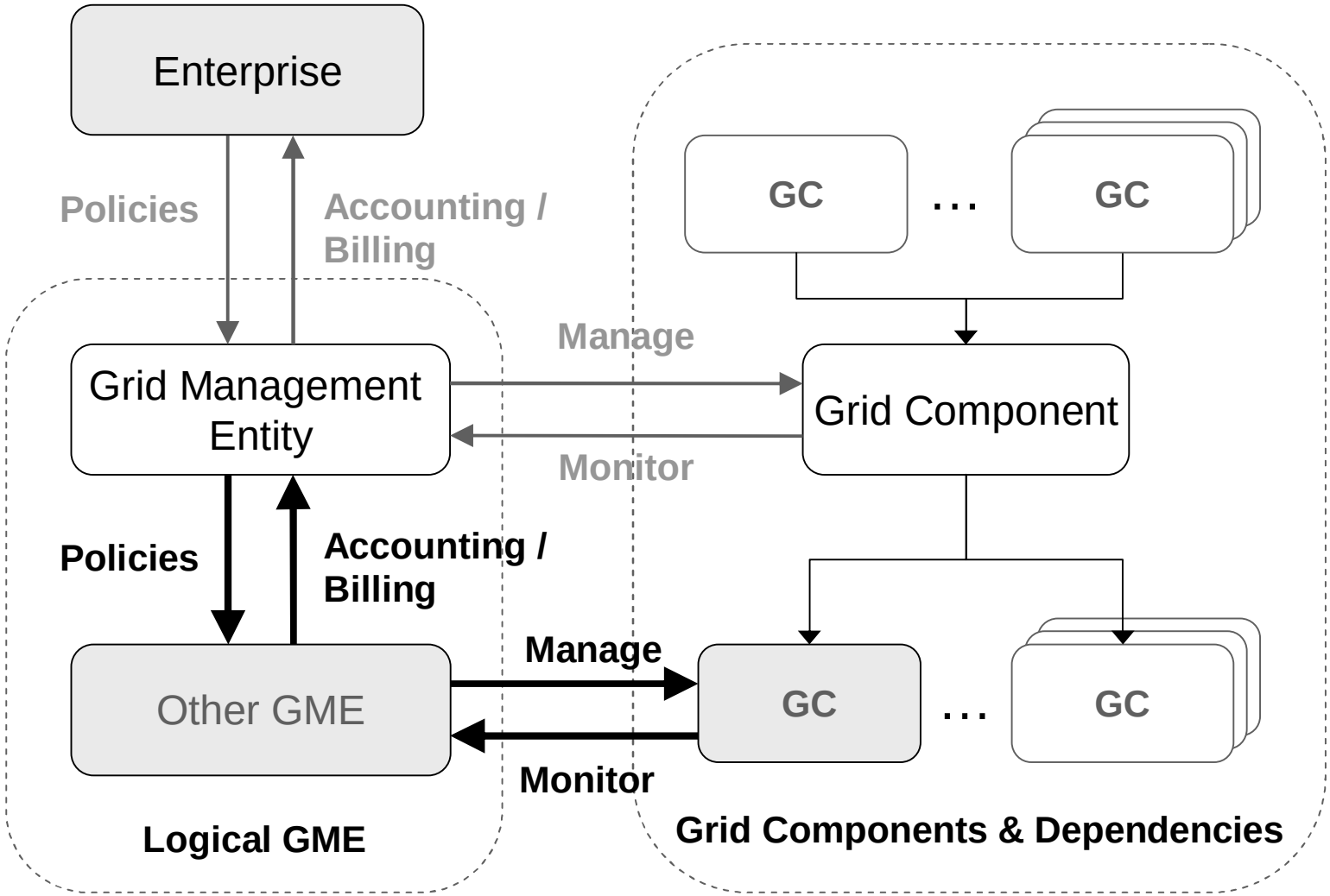
Isolation

Provisioning

# Recap: Use Cases & Capabilities

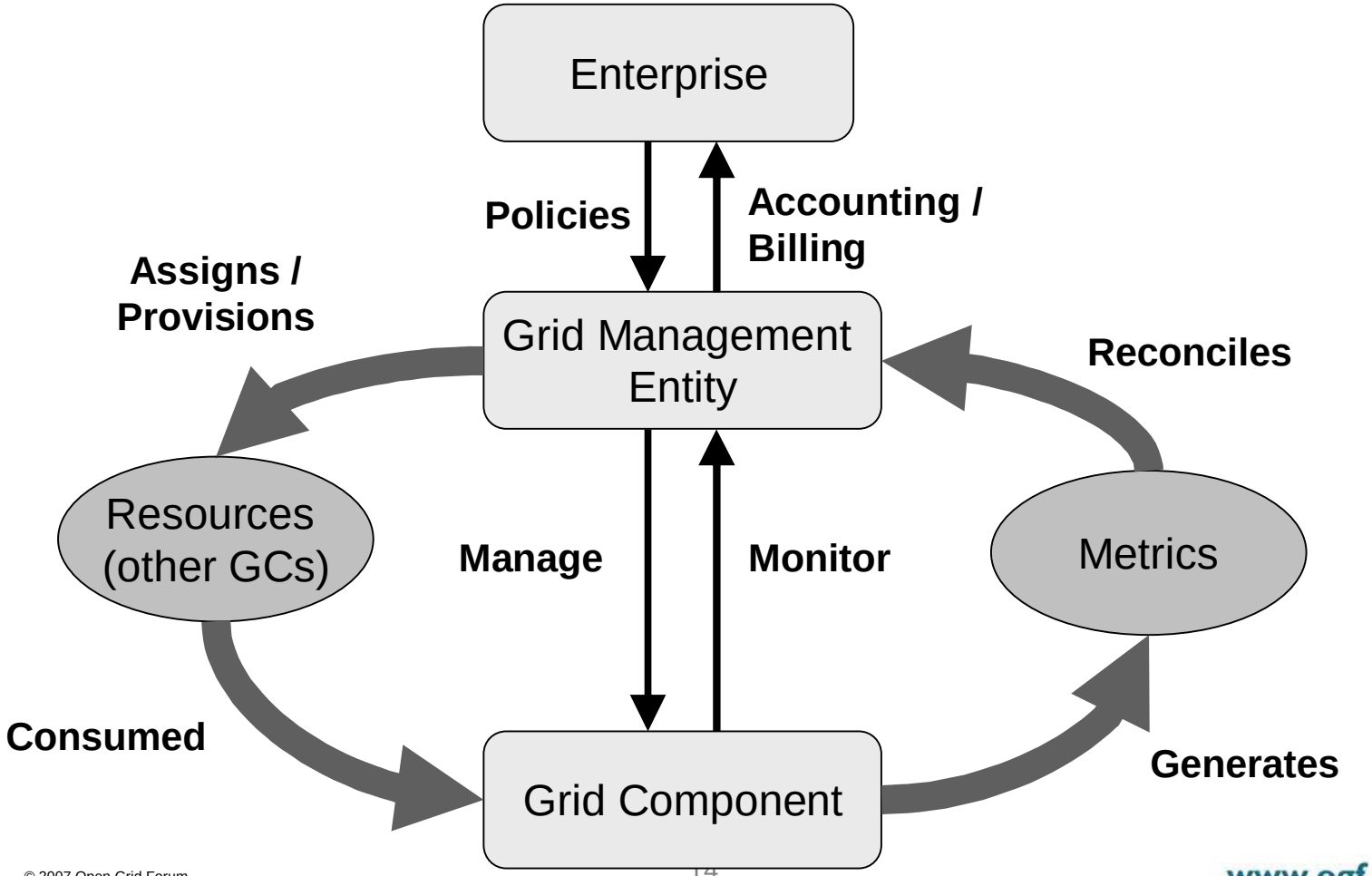


# OGF Reference Model



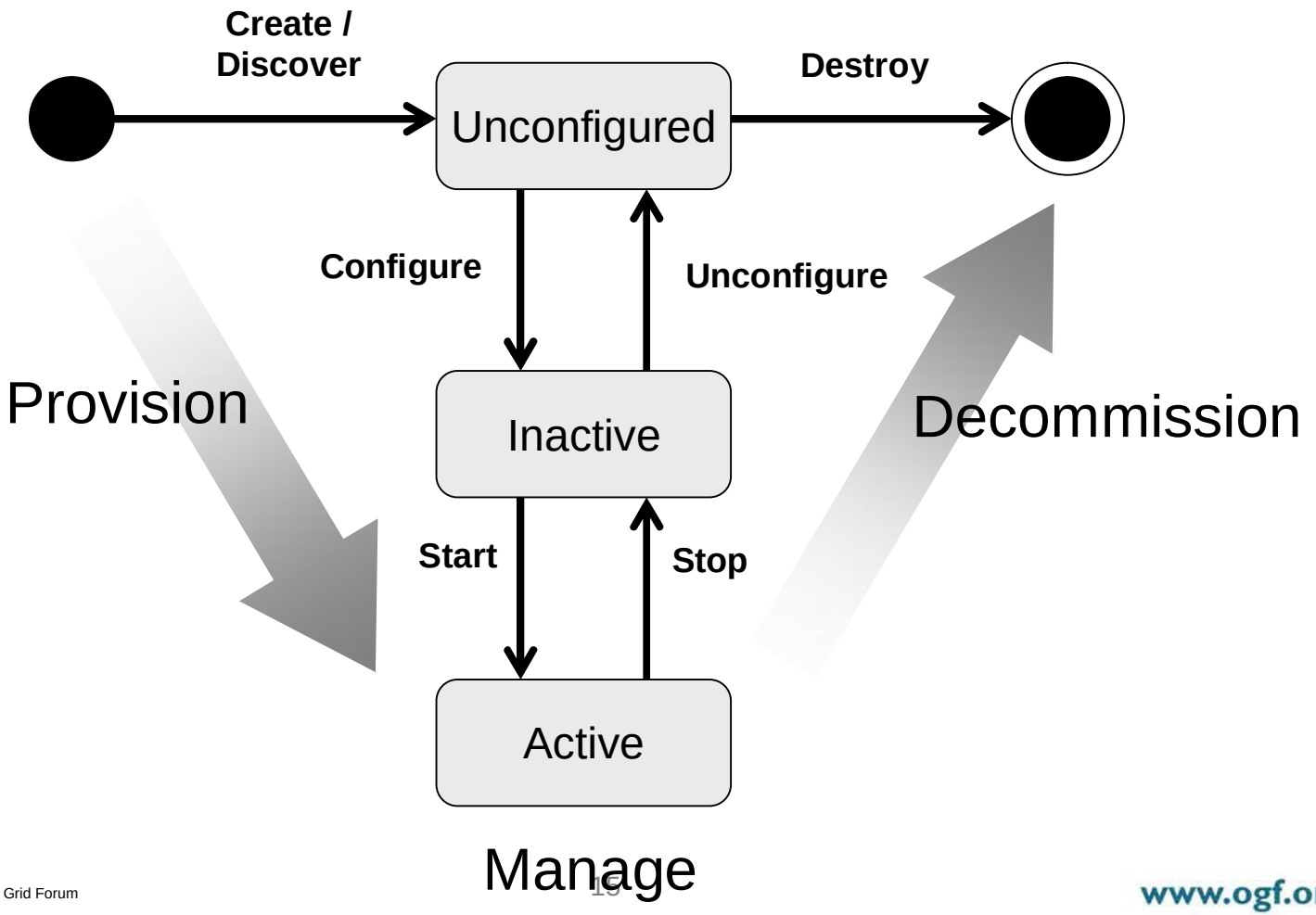
# OGF Reference Model

## Service Level Management



# OGF Reference Model

## Lifecycle of a Grid Component



# Refine Selected Use Cases


**Use cases**

- **Dynamically changing capacity requirements**

  During runtime the job may require additional capacity (e.g. CPU capacity, Memory capacity, I/O bandwidth etc.).

  If the underlying physical system is able to serve the requirements more capacity for the job / virtual system can be provided locally on the same physical system. If the requirements can be better fulfilled on another physical system the virtual system might be migrated.

- **Dynamically changing capacity offering / availability**

  Capacity availability may change in the physical system (e.g. CPU capacity, Memory capacity, I/O bandwidth etc.) because of recently freed resources by the completed jobs. In these situations available capacity can be utilized for the running jobs.

  Additional resources might become available on another physical systems which can be utilized.

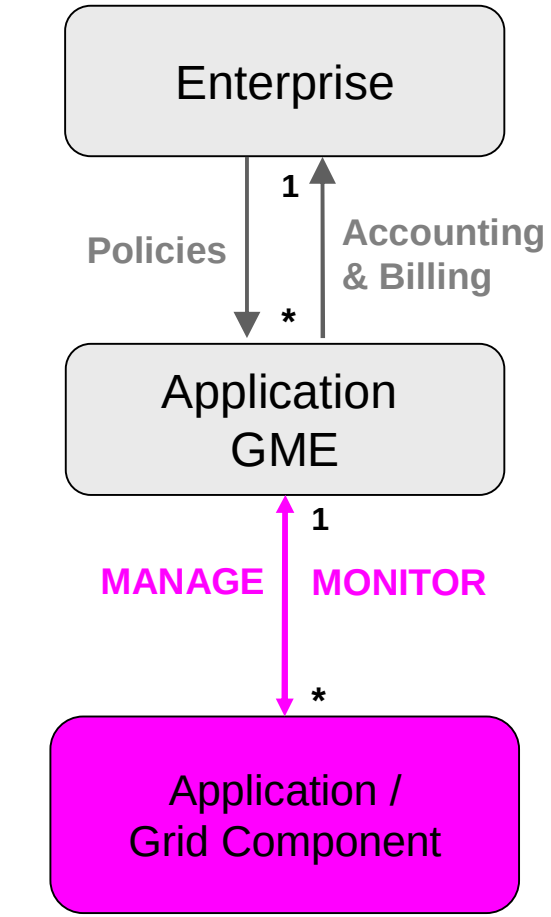
**Related virtualization capabilities**

- **Dynamic system resizing**
- **Live migration of virtual system during runtime**

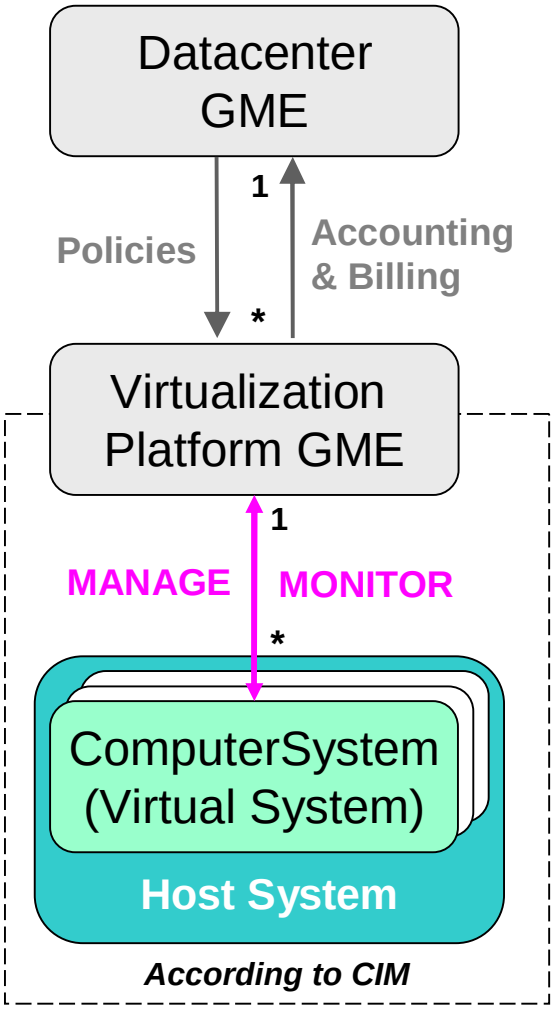
# Remarks

- Two related use cases have been selected: Dynamic Resizing and Live Migration. Both are about dynamic resource availability and changing environments during the lifetime of an application.
- Any management capabilities and communication with the ComputerSystem should be handled by the GME. The GME needs to be virtualization aware and enabled. The application code should not have any “intelligence” or knowledge to manage dynamic environments. This is in accordance with the OGF Reference Model.
- It might be desired that the application is “virtualization friendly”. The application shall allocate resources only when needed and deallocate them if no longer needed. It shall not assume that any allocated resources are physical and dedicated.
- The Virtualization Platform GME is considered to be an CIM Object Manager, i.e. an implementation of CIM profiles and classes (like CIM_ComputerSystem). It is assumed that the application GME understands CIM to the extent it exploits system capabilities. It directly communicates with the respective CIM Object Manager(s).

# Managing Hierarchy

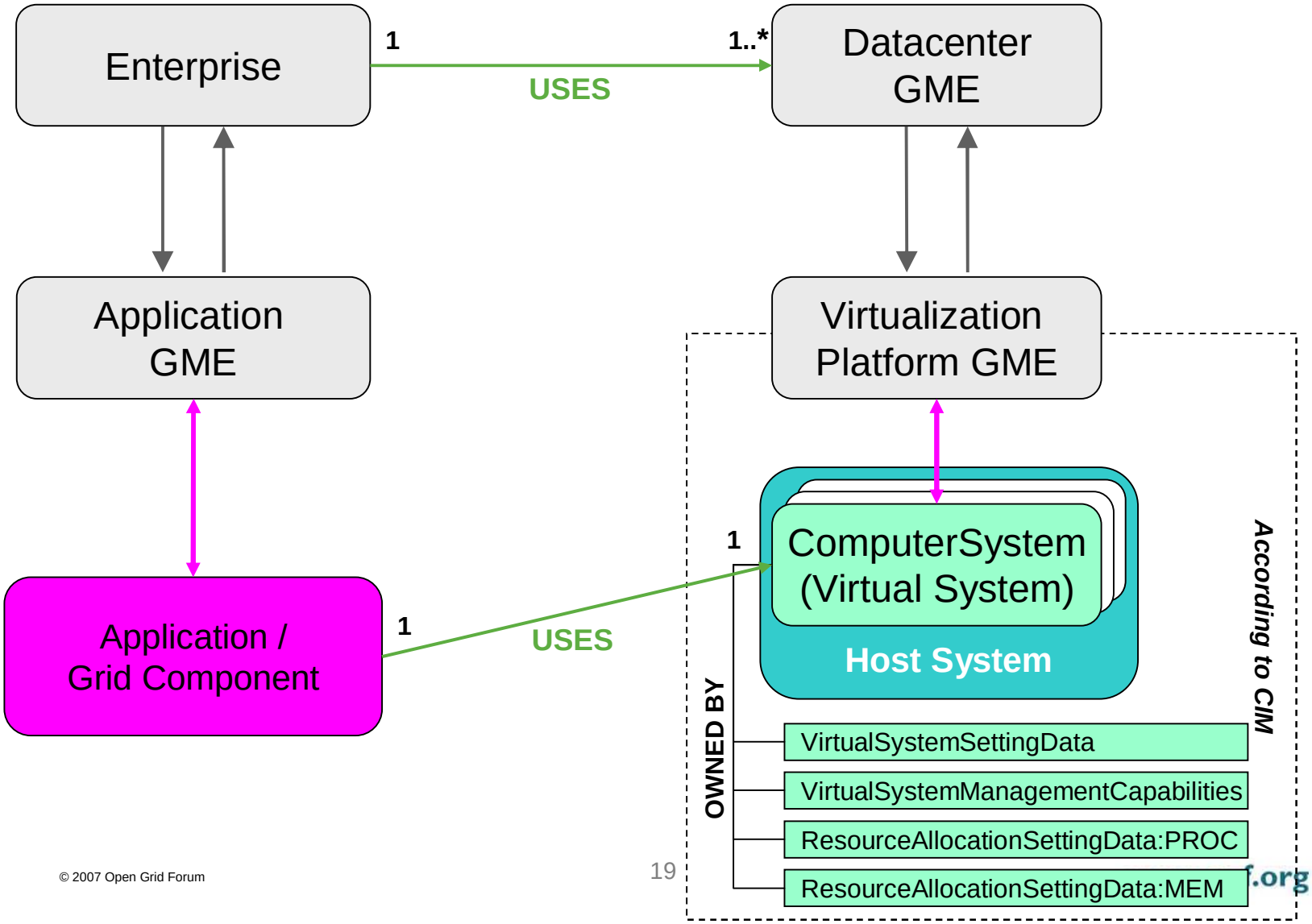



APPLICATION VIEW

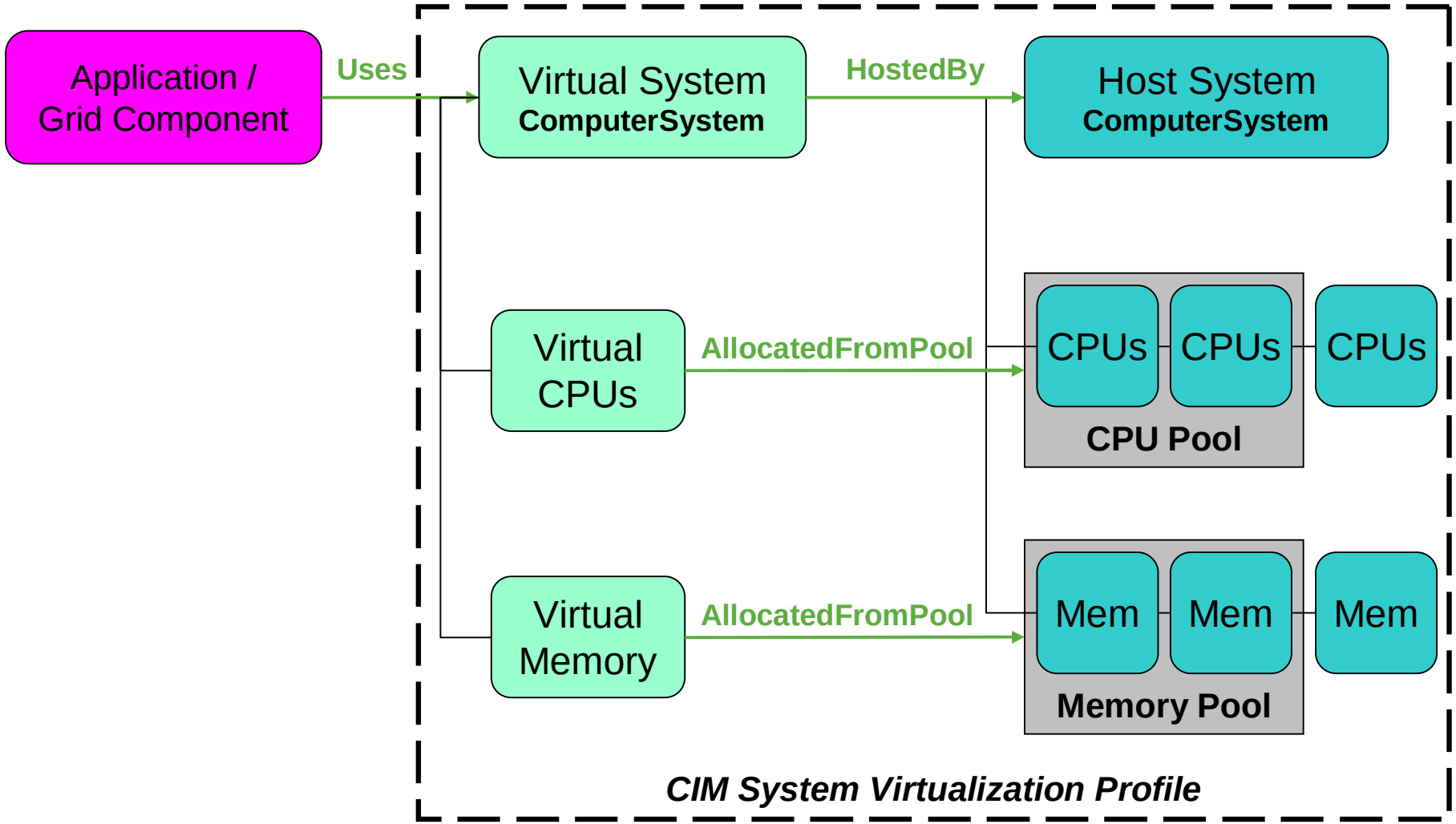



SERVER VIEW

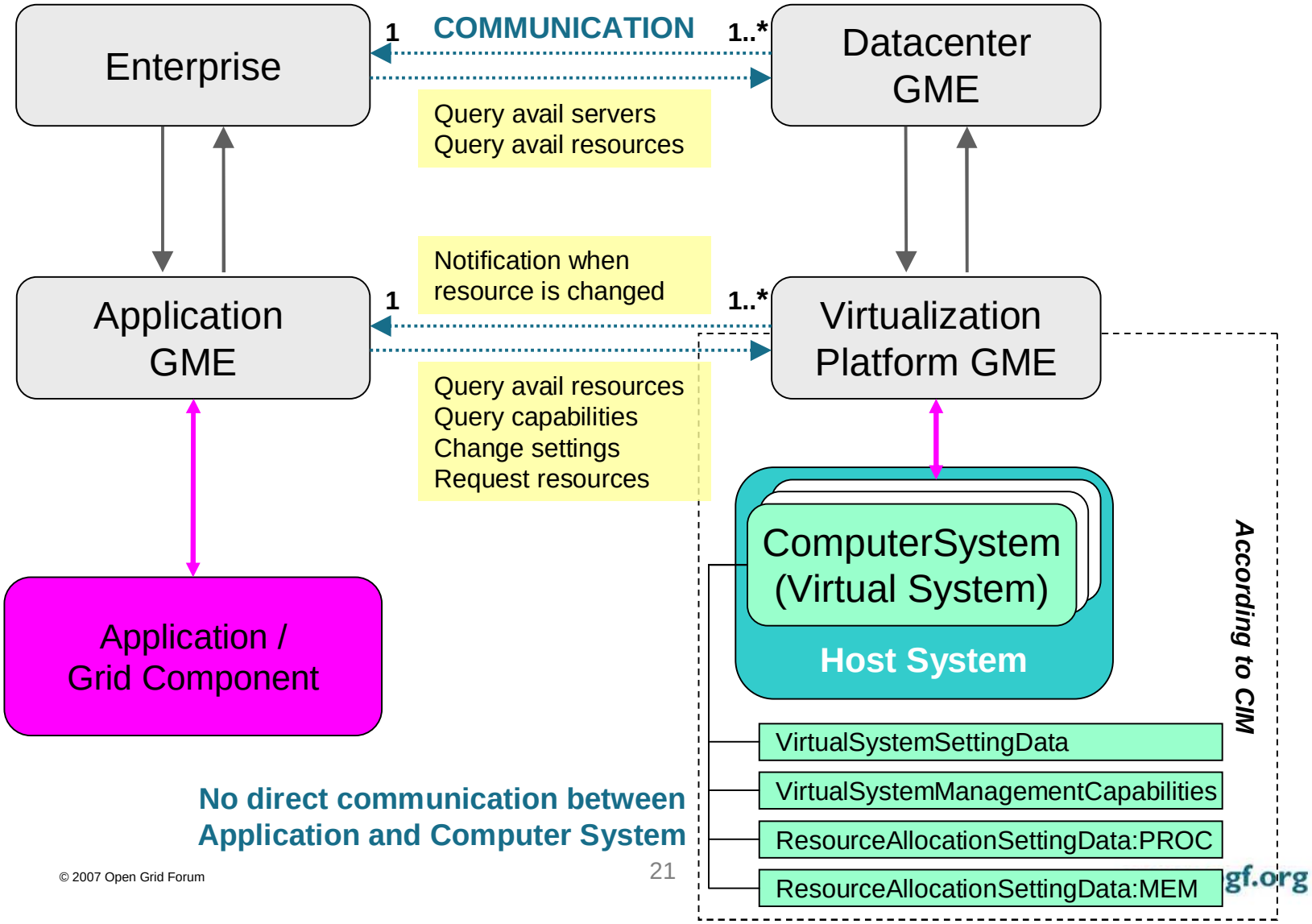
# Use Relationship



# Relation to CIM System Virtualization Profile

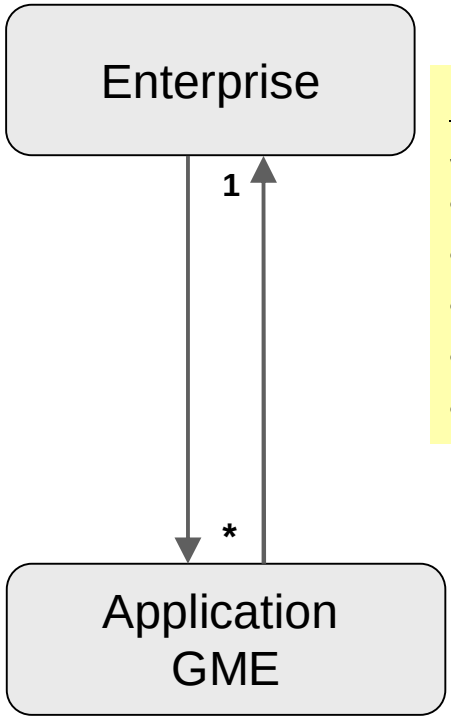


# Communication Paths




**No direct communication between Application and Computer System**

# Policy Hierarchy




**Business Policy**

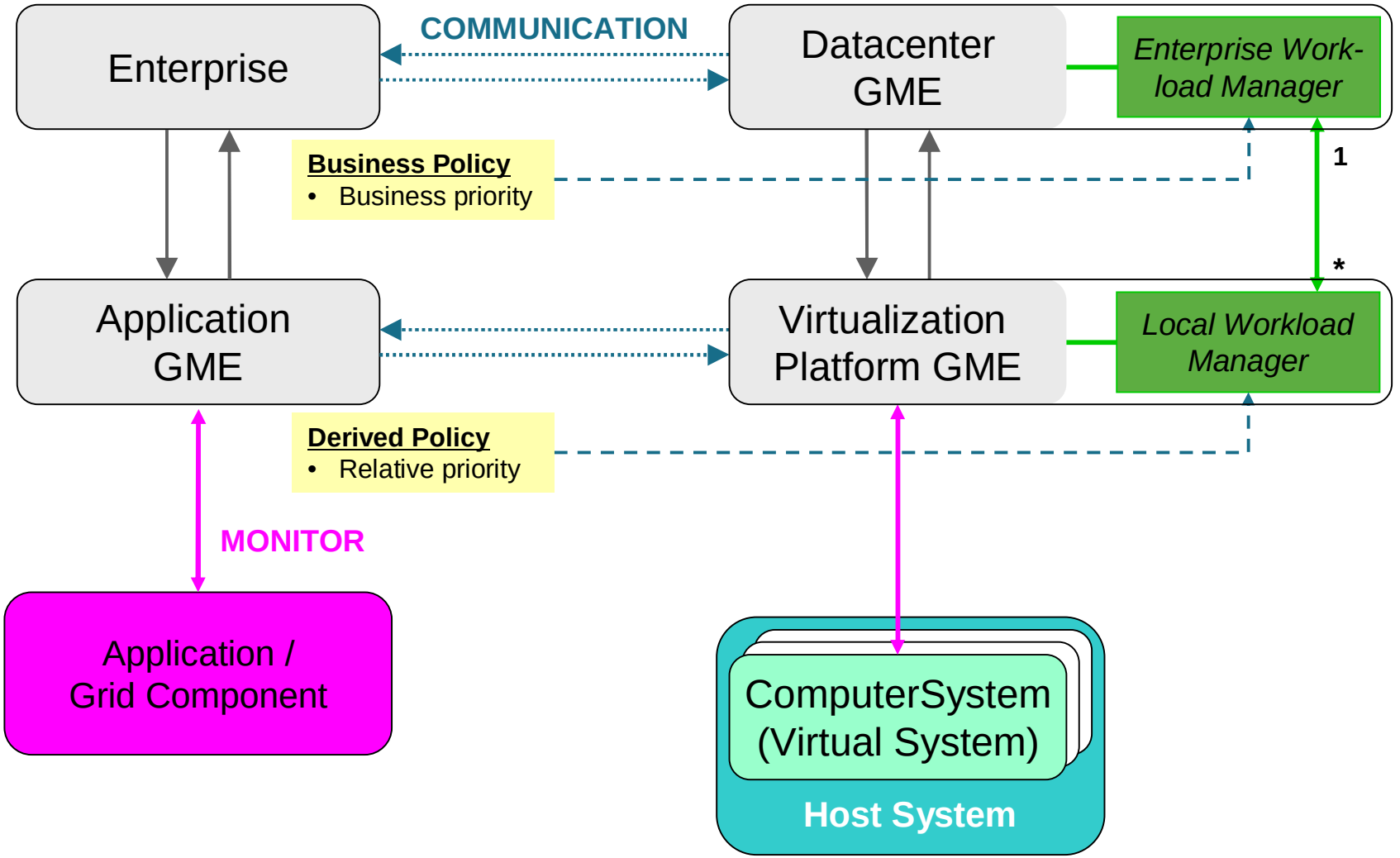
**Service Level**

- End-to-end response time / overall throughput
- Max/min resource allocation (processor, memory, …)
- Business priority
- Security level
- Availability level

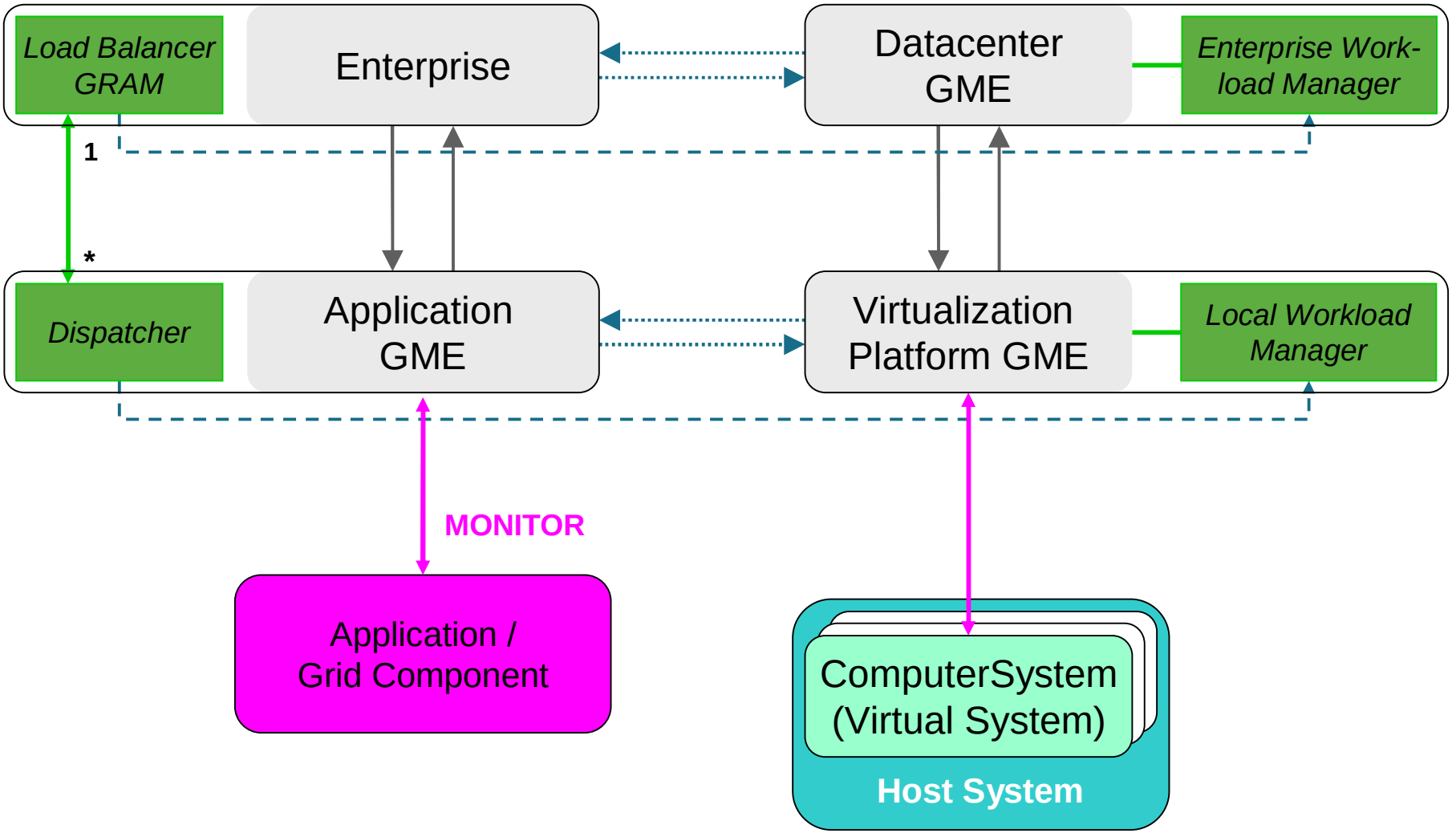
**Derived Policy**

- Response time / throughput
- Max/min resource allocation (processor, memory, …)
- Relative priority / weight
- Security settings (isolation, encryption)
- Availability settings (HighAvailability, multipathing)

# Workload Management



# Workload Management



# Workstreams

- Workstream 1: *High-level Use Cases Description* & Alignment of *Grid Reference Architecture* in the Context of System Virtualization
  - Define the requirements to the grid architecture for integration with system virtualization platforms
- Workstream 2: Refine the *Resizing and Migration* Use Case
  - Define Interaction among the components in the architecture to create / discover, configure and start a Virtual System
  - Describe relationship to SVPC Model
- Workstream 3: Refine the *Provisioning* Use Case
  - Define Interaction among the components in the architecture to create / discover, configure and start a Virtual System
  - Describe relationship to SVPC Model & OVF (Open Virtual Machine Format)