View of "program execution" as presented by Ravi Subramaniam (Intel) – OGSA F2F (Sunnyvale, CA)

Service Category include but not limited to the followings:

5.1 Core Services
5.1.2 Service Interaction
5.1.3 Service Management
5.1.4 Service Communication
5.1.5 Policy and Agreement
5.1.6 Security
5.2 Data Services
5.3 Program Execution

Provide summary of the service category and include overall architectural figure if appropriate.

5   OGSA Platform Services

5.1   Workload realization (generalizes program execution)

Workload realization is the general set of use cases that take workload requests and map them to appropriate resources within the grid that can realize these workloads. Grid services manage and co-ordinate access and consumption of geographically distributed resources by workloads realized on those resources.

The overall conceptual architecture can be visualized as shown in Figure 1. Workload realization (or execution) can be visualized as a mapping between "demand" in the form of workloads and "supply" in the form of the *available* Grid resources. In a fundamental scenario, the system has to map the demand to the supply and provide the mechanism to realize these workloads on the resources. This primary mode can be augmented with other services, mechanism and capabilities that provide alternate modes of interaction including optimization of the mapping and scheduling a temporal and topological execution profile. In addition, other services manage and enforce of the service level agreements with the user and still other services tweak the resources and manage the available capacity to ensure a desired quality of service is delivered. The alternate modes are not 'core' to the execution framework but can be added to augment the efficacy and efficiency of the overall system.

Services that belong to the *Resource Optimization Framework* are focused on the optimization of the supply side of the mapping. This can be done by admission control, resource utilization monitoring and metering, capacity projections, resource provisioning and load balancing across equivalent resources, negotiation with workload optimization and/or management services to migrate workloads onto other resources so as to maximize resource utilization.

Services that belong to the *Workload Optimization Framework* are focused on the demand side of the mapping. These services may queue requests to prevent resource saturation and manage relative priorities in requests, perform post-balancing by migration workloads to appropriate resources depending on the potential to violate or be rewarded for missing or exceeding SLAs respectively.

Services in the *Optimizing Framework* are focused on resolving any contentions that the myopic views of the respective resource or workload optimization frameworks may create. These services arbitrate and modulate the primary interactions either in an 'in-band' or 'out-of-band' manner.
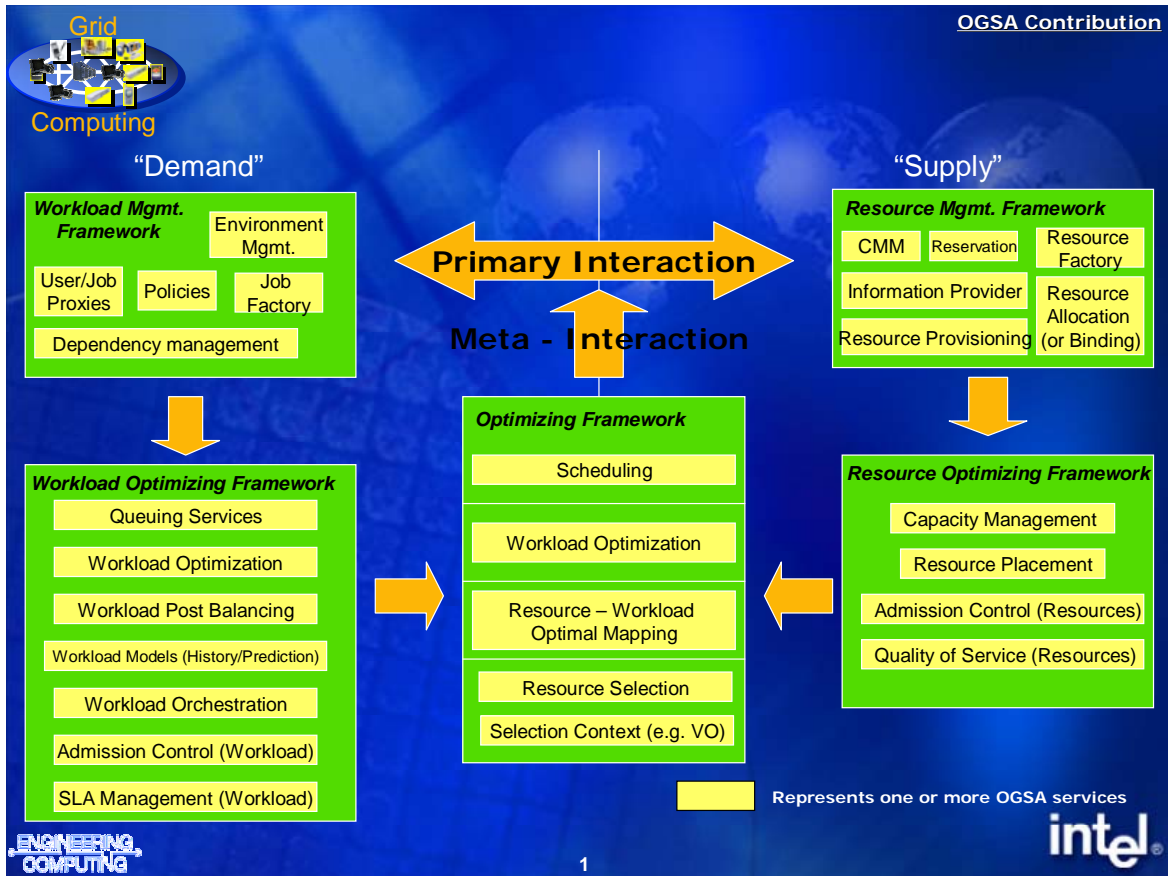
Figure 1: Grid Frameworks - Execution

Resources are used in its most general sense and can include virtualized physical resources like CPU, storage, memory and/or virtual resources like software licenses or data. Workloads are composite entities and have multiple levels of "execution entities"; workloads are made of jobs which in turn are made of tasks which in turn are made of tasklets. Each of these composite entities has a manager.
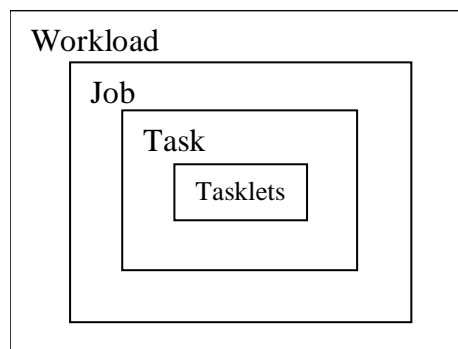
A possible set of interactions between Grid services to execute workloads are shown in Figure 2. The services can be at multiple levels in the Grid and can be composed of other services or service groups.
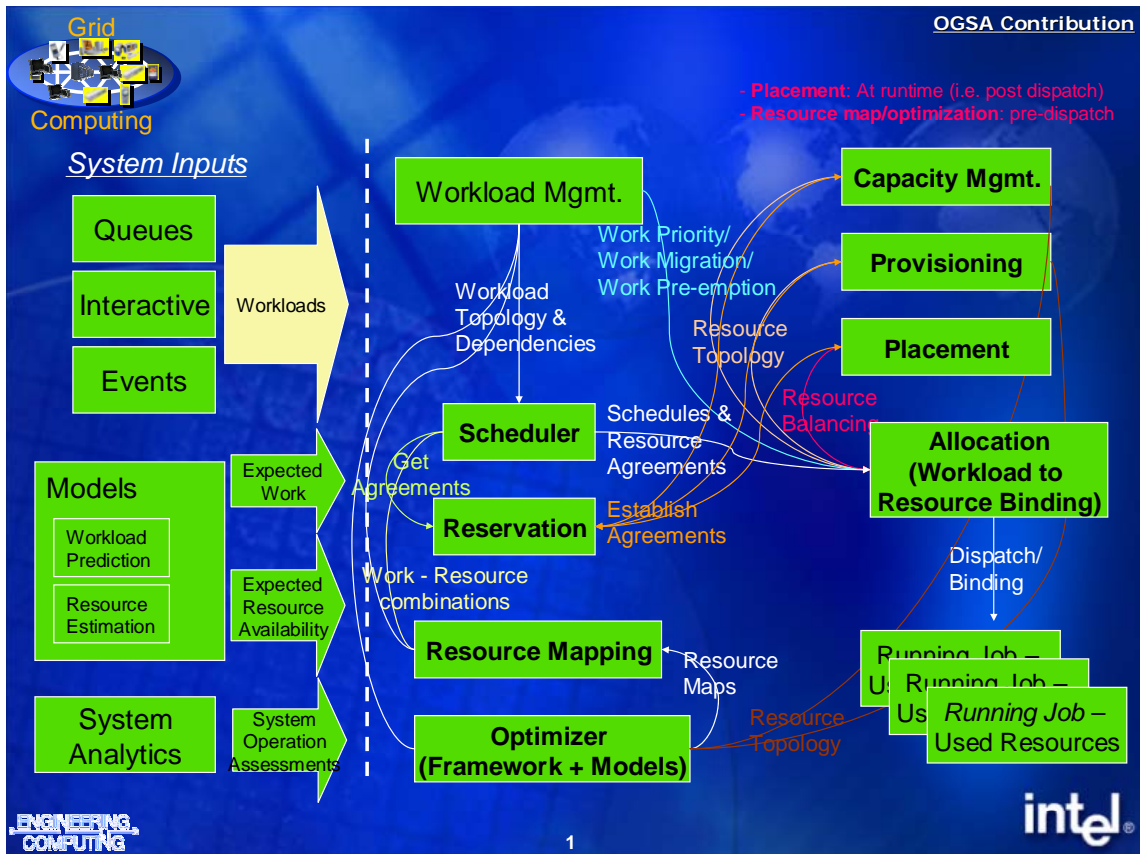
Figure 2: Service interactions for workload execution (partial enumeration)

A physical organization of the services and one of the possible collaborations scenarios is shown in Figure 3.
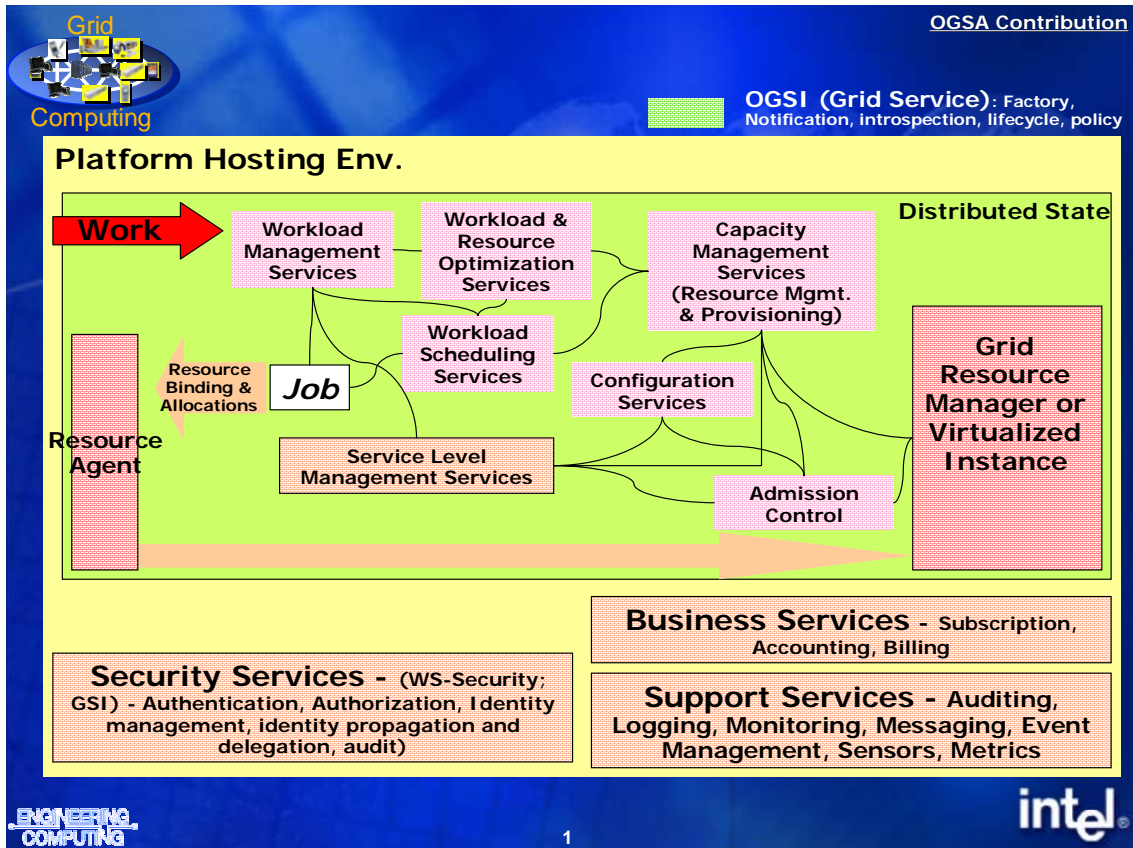
Figure 3: Example Grid execution services interaction

5.1.1 <Service Name>

Different from service category, service is a non decomposable entity from the view point of the archtecture.

> *Issue#1: It is unclear how to handle composite services. E.g. Meta-Scheduler consists of Job Service and Reservation Service. Is Meta-Scheduler first class citizen or not?*

Provide 1/2 page summary of the service. What it covers and does not. Explain its properties and attributes. Please include figures if possible.

5.1.1.1 *List of interfaces*

Please list up interfaces provided by this service. Explain each interface by a couple of paragraphs.

It may include portType level definition but such descriptions intend to give concrete idea what this service is but these definition is just an example (informative). Instead, owner-WG should define normative interface in detail.

5.1.1.2 *Standard schema/document*

Agreements: In this discussion an agreement is assumed to be a service that manages and represents an agreement between an initiator and grantor and/or a signed document that is signed by both parties in the agreement. Agreements established between two parties can be delegated to other services to be exercised on behalf of either party. Agreements can be based on specific instances of workload/jobs to specific instances of resources or can be from classes of workloads to classes of resources or a combination of classes or single instances. In the case of the classes of resources, the system may substitute any instance that meets the requirements embedded in the agreement to fulfill that agreement.

5.1.1.3   Underlying services
Explain which services (infrastructures) are extended by this service.

5.1.1.4   Related services
Explain which services are used by this service. Related services are peer not underlying.

5.1.1.5   Related standard
If one or more related standards are already exist, please specify them and which standard body define them.

5.1.1.6   Owner WG(s)
Please specify owner GGF-WG of this service if exist. Owner WG is expected to define actual service interface or portType. OGSA-WG will talk with owner-WG and agree the definition of the service and schedule for the specification creation.

If there is no appropriate WG, OGSA-WG will spawn new WG for this service.