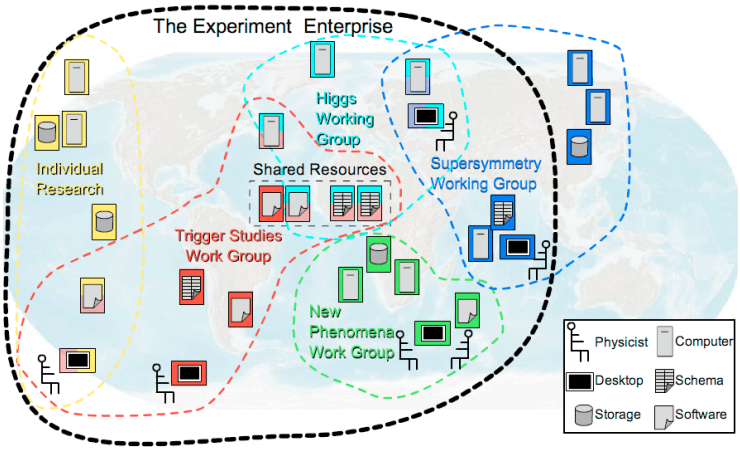
Grid-Centered Position Paper for the W3C Workshop on Constraints and Capabilities for Web Services.

Frank Siebenlist, Argonne National Laboratory (franks@mcs.anl.gov)
Takuya Mori, NEC Corporation (moritaku@bx.jp.nec.com)

This paper discusses constraints and capabilities as they are expressed in policies from the Grid community's perspective. The Virtual Organization is introduced as a concept to facilitate collaborations. The negotiated agreement is seen as the foundation for the Virtual Organization from which the policies are derived that are to be enforced and which defines the context in which interactions will occur. Use cases and scenarios are presented that highlight the desirability of close integration of policy- and high-level application languages and that discuss the importance of the ability for subjects to dynamically and ad-hoc empower other subject with capabilities and express constraints for resource access. Finally, two partial solutions are discussed for the use case in the call-for-participation that address the sharing and discovery of policies.

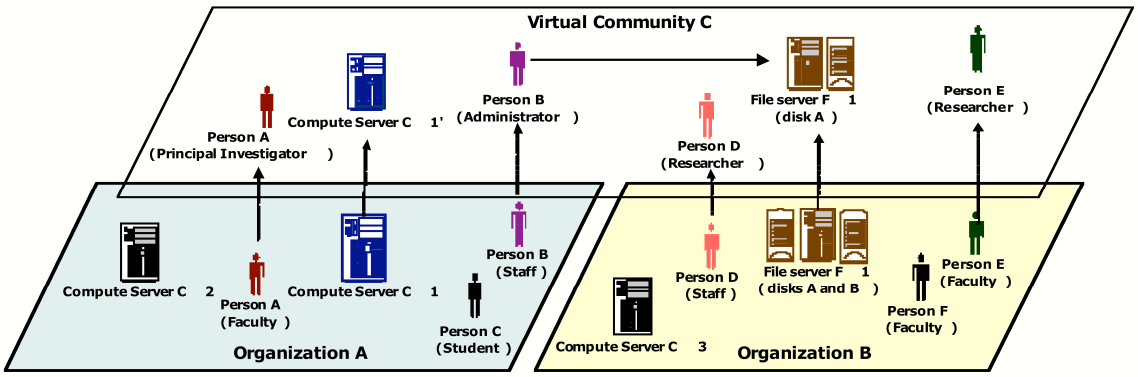
Grid Applications and Virtual Organizations



Cross-Organizational Collaborations

Grid-specific applications may span multiple administrative domains. This property implies that each of these domains will have its own business objectives to meet, which translates to the individual domains separately establishing and enforcing their own policies, which can differ greatly in complexity and strictness. Note that all interactions associated with a thread of work in a Grid application must therefore adhere to the domain-locally-enforced policies as well as to the policies established for the Virtual Organization (VO) — i.e. the cross-organizational (business) agreement. [GRID, GGF, OGSA, GLOBUS]

The Virtual Organization is an artifact defined by the agreement between the collaborating organizations.



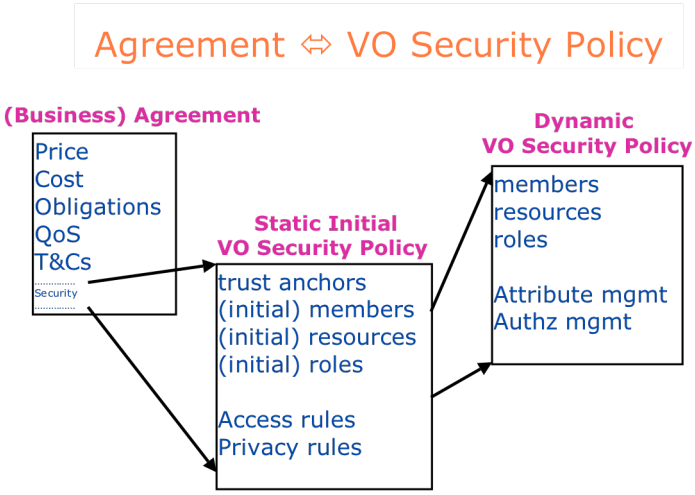
The members of a VO come from the participating organizations, but may have different roles and responsibilities. Resources owned by the different organizations may be made available for sharing with the VO-members, but may have different SLAs associated with them and different access control policies. In other words, the VO will have its own set of policies that are enforced with all interactions that are within the context of that VO.

Use Cases

In this section, we discuss two use cases. First, we argue that the foundation for collaborations are agreements from which policy are derived. Second, we make the observation that with the virtualization of resources one needs the ability to empower others to work on one's behalf and that the mismatch of application and policy language has the undesirable effect that constraints have to be relaxed or that intermediates are empowered with too many capbilities.

Agreement, Virtual Organization and Policy

From a high-level conceptual view, Virtual Organizations are created and used to meet the collaboration's objectives. These objectives are negotiated and stated in agreements. From these agreements, policies are derived that will govern the interactions between parties, and will include terms and conditions about the responsible entities and boundaries. In other words, the stated capabilities are granted and the constraints are enforced to help to meet the collaboration's objectives. Currently, many of these steps and dependencies are informal, disconnected and follow ill-defined procedures.



We need the ability to automate this process and the ability to derive the policy associated with negotiated agreements. This would allow VOs to be established more quickly with policies that are closely coupled to the agreement and are thus in a better position to support the VO's objectives. One needs policy related terms and conditions in the agreement languages such that the correct VO policy can be derived from a negotiated contract. [GRAAP]

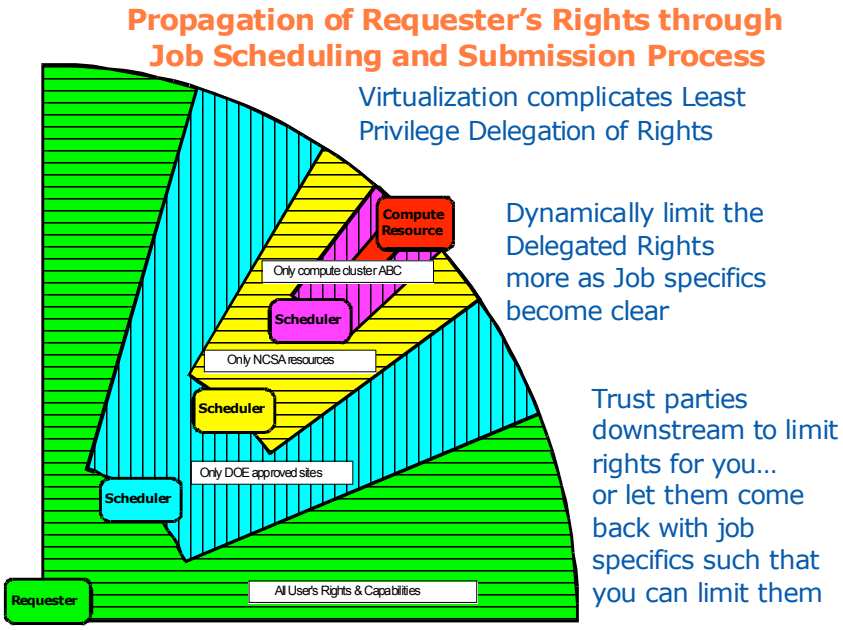
Integration of Application and Policy Languages

Many of the Grid applications use high-level languages for the negotiation of agreements, for scheduling and for job submission. There are currently many efforts at the Global Grid Forum [GGF] to improve and standardize these languages. The security policy is more naturally expressed in a language that uses the same primitives and is on the same level as these application languages. So far, however, the security policy languages that are used to enforce the policy do not match the application languages well as they are designed outside of the application domains. As a consequence, there are mismatches between the expressiveness and semantics of the application and policy languages, which results in a security policy enforcement that has to allow for more rights than necessary with the associated risks for compromise.
We need to minimize mismatch of agreement, job specification, and scheduling languages with the policy language that is used for enforcement. This requires a tight integration of the languages on the semantic level that can be achieved by deploying common ontologies as the basis. [JSDL, GRAAP]

This issue is illustrated in the example, where the resources that a requester wants to use are virtualized. In that case, the requester relies on scheduling and discovery services to work on his behalf and will accept any physical resource that will meet the required service levels and that adheres to its local policy. In other words, the requester will have to empower those intermediates with a subset of its rights such that those services can work on the requester's behalf. The intermediates may rely on yet another level of intermediates that again have to be empowered with enough capabilities to initiate the job invocation at the ultimate physical resource.

This delegation of rights is a fundamental capability needed to let services work on behalf of other entities. With this rights-delegation comes the associated risk that any of these services may be compromised and use those rights in inappropriate ways. To limit the exposure, one would like to limit the delegated rights to only those rights truly needed by the service. This *least-privilege*

delegation model requires that one is able to match the invoked service operations with the exact "amount" of rights, which is a non-trivial requirement. Many Grid applications use the concept of jobs, in which job directives are specified in their own language. The job requirements are then matched with the capabilities and availability of resources by discovery, brokers, and scheduler



services. The language used for the expression of these job directives and resource capabilities should be able to match up with the directives used to express the equivalent rights needed. Any mismatch is likely to result in a deployment where essentially too many rights will have to be given to services to ensure that the job directives can be executed.

W3C Use Case

For the workshop registration, the submitters were asked to address the following use case:

> *A Web service wishes to stipulate that clients are required support a reliable messaging protocol, and encrypt a specific header with WS-Security using a X.509 or user name security token in order to send an acceptable request message. Furthermore, the service has a P3P policy associated with its operations. Such constraints and capabilities might be associated with the Web service via a SOAP header or a WSDL file.*

Proposed Solution

A complete solution will have to be able to express the stated policies, be able to associate those policies clearly with the service provider instance, and be able to share that policy with the requesters such that they can decide whether the provider's policies overlaps with the requester's capabilities and requester's own policies.

It can be expected that a number of different mechanisms will be proposed for the format, association and sharing, and we would like to focus on a Web Services Resource Framework [WSRF] specific mechanism that could be used as a solution for the sharing of the provider's policies and a Grid-specific solution of pre-screening applicable service providers based on a policy profile.

WSRF resource properties for policy information sharing.

The WSRF specifications define so-called Resource Properties [WSRF-RP]:

The declaration of the WS-Resource's properties represents a projection of or a view on the WS-Resource's state. This projection is defined in terms of a resource properties document. This resource properties document serves to define a basis for access to the resource properties through Web service interfaces. This specification also defines a standard set of message exchanges that allow a requestor to query or update the property values of the WS-Resource. The set of properties defined in the resource properties document associated with the service interface defines the constraints on the valid contents of these message exchanges.

These resource properties can be used to make all kinds of resource related information available to requesters, including information about the policy that will be enforced.
In other words, the WSRF resource properties could give a standardized way for requesters to query for the policy information associated with the different operations that can be invoked on the WS-Resource.
Furthermore, if we standardize the xml-element definitions associated with the policy statements, then the same definitions could be used to share the policy through Resource Properties, through embedding in EPRs, or in dedicated query interfaces and directory services.

Meta-Data Services for service discovery through policy profiles

A number of Grid applications are working with so-called Meta-Data Services (MDS), which are services that will gather and collect knowledge-domain specific data. These MDS services could collect their information by querying for certain resource properties of the services that they are made aware off. Requesters who are trying to discover the available services that meet their own policy criteria will use their domain specific MDS servers to help them to locate service providers that will fit the right profile. Note that the discovery of applicable services is one of the key problem areas of Grid computing.
One could for example imagine a MDS service that would collect information about certain type of services (as in WSDL portType) that only support a reliable message interface and would adhere to certain P3P policies. A requester could use this service to obtain a short-list of EPRs to only those services that fits its profile.

References

[GRID] Foster, I. and Kesselman, C. eds. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann (2nd Edition), 2004.
[GGF] Global Grid Forum, http://www.ggf.org
[GLOBUS] Globus Project, http://www.globus.org
[GRAAP] Grid Resource Allocation Agreement Protocol working group, GGF, https://forge.gridforum.org/projects/graap-wg
[JSDL] Job Submission Description Language Working Group, http://www.epcc.ed.ac.uk/~ali/WORK/GGF/JSDL-WG/
[OGSA] Foster, I., et al The Open Grid Services Architecture, Version 1.0, http://forge.gridforum.org/projects/ogsa-wg
[WSRF] Web Services Resource Framework, OASIS TC, http://www.oasis-open.org/apps/org/workgroup/wsrf
[WSRF-RP] Web Services Resource Framework Resource Properties, OASIS, http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf