Lambda Grid, Grid Computing and AAA

John Vollbrecht
April 2004

Abstract

This paper outlines the changing requirements of high performance computing applications, reviews work that has been done in the area of Lambda Grid[16] and Computing Grid[17] to address these requirements, describes some architectural issues that affect these applications and solutions, and presents proposals for extending existing work that include concepts defined by the AAAArch research group[65]. High performance computing capabilities available in clusters [63, 64] and high capacity communications using fiber optical resources have created a new capabilities and new architectural imperatives. The new high performance environment has been called the GRID to describe the massive interactions between devices within applications. The ability to control and integrate large parts of the "GRID" to control specific applications has been described as creating an OptIPuter[6].

Work to support and standardize control of Grid computing resources has been led by the Global Grid Forum.[17] Work to integrate the new high bandwidth fiber capabilities has been done in several "Lambda Grid" projects [34]. Current and future work supports applications that use resources from multiple organizations. The proposed work focuses on creating an independent server which uses concepts from a number of other organizations to create a software server that supports Virtual Organizations using concepts from Globus and from the Amsterdam Generic AAA toolkit. The intent is to make it easy to create and use Grid collaborations of people and organizations sharing high performance computing and communication capabilities.

1. A new computing paradigm - the elements of change

Several sets of activities have created the environment in which new paradigms for computing are evolving. Two parallel trends describe the changes: 1) the push caused by rapidly increasing capabilities of computing, storage and communication hardware, and 2) the pull from applications that do not fit the existing computing paradigm well.

1.1 Increasing Capabilities

The push from increasing capabilities has resulted in work to define new ways to interconnect and manage computing resources. The OptIPuter project [1] focuses on the changing relationships between computing elements, as well as on the increase in power of each type of element. Increases in computing power have a doubling time of 18 months, storage device capacity doubles every 12 months and communication speed doubles every 9 months. The difference in rate of increase is creating a situation in which although significantly more power is available in all areas, the traditional implementation tradeoffs between capabilities changes. This change in power, especially the increasing capability and decreasing costs of fiber, leads to the concept of the OptIPuter [5,6], which treats the communication elements as the backplane of the OptiPuter and computers and storage devices as the nodes. This paper investigates the use of AAA concepts to coordinate and control the resources of the OptiPuter as it carries out a variety of applications.

1.2 New Application Requirements

New applications are being developed in physics, biology, astronomy, visualization, meteorology, as well as in business and other areas that are not served well by existing computing concepts. Many of the anticipated applications have communication requirements between a small numbers of sites. This presents a very different requirement than presented by "typical" Internet use where the communication is among many sites. Cees DeLaat defines three classes of applications: 1) lightweight "classical" Internet applications (mail, browsing), 2) medium applications (business, streaming, VPN) and 3) heavyweight applications (e-science, computing, data Grids, virtual presence) [16]. The total bandwidth estimate for all users of each class of network application is 20 Gb/sec for the lightweight Internet, 40 Gb/sec for all users of the intermediate class of applications and 100 Gb/sec for the heavyweight applications. Note that the

heavyweight applications use significantly more bandwidth than the total bandwidth of all applications on the classical Internet. Differences application types value different capabilities. Lightweight Internet applications value interconnectivity, middleweight applications value throughput and QoS for long term connections but also connectivity to many places, while the heavyweight applications value throughput and performance between few places. This has been characterized as many-many, several-several, and few-few.

1.3 New classes of networked applications

The August 2003 issue of FGS [2] has a number of articles about experiments with future heavyweight (few-few) applications. A particularly important example, the Large Hadron Collider (LHC) in CERN [66], scheduled to come on line in 2007, has requirements well beyond what is available now. New computing mechanisms are being invented, implemented and tested at this time to enable ultra large scale computing [67]. The LHC will produce and need to distribute gigantic amounts of data. The data rates from the LHC in CERN are estimated to be in the order 100 to 1500 Megabytes per second [3]. Dedicated fiber will carry data from the Atlas Collector in CERN to the initial storage and computing devices around the world, bypassing the Internet. The creation of a "Lambda Grid" to support this class of applications has been initiated [22]. The LHC application is an example of new heavyweight network applications that require the capabilities of the OptIPuter and help shape the implementation of the AAA mechanisms to coordinate and control it.

In addition to transporting the huge amounts of data created at CERN and cached at various locations, the LHC computing plan is for groups of physicists from different organizations to collaborate to analyze specific problems. There will be many shifting collaborative groups, with multiple organizations contributing researchers and resources to one or more collaborations. Creating these short-term collaborative organizations is a technical challenge both in creating the necessary application/job management capabilities and in supporting the security requirements of users and resources from many different organizational domains [13].

Another example of heavyweight "few-few" application are the CAVE Research Network work being done at the Electronic Visualization Lab (EVL) at the University of Illinois at Chicago [8] which uses very large communication rates to implement total immersion projects. These projects also have the need for creating collaborative groups across organizational domains with domains contributing resources as well as collaborators. Other examples include the Electron Microscopy work being done at the University of California San Diego (UCSD) and Osaka [12].

2. The Global Grid Forum and Globus

A book called "The GRID" edited by Foster and Kellelman [4] and published in 1999 and revised in 2003 outlines many new high speed computing applications and requirements which taken together define a new computing paradigm called Grid Computing. The GRID computing paradigm, as defined in their book, is supports the kind of applications described in the previous section as well as a host of other distributed applications.

2.1 Global Grid Forum

To support the development of standards to support this new paradigm, the Global Grid Forum (GGF) was formed and had it first meeting in 1999. The group purpose was "developing standards and best practices for distributed computing ("Grids" and "Metacomputing") efforts including those specifically aimed at very large data sets, high performance computing but increasingly those efforts that industry is calling "Peer-to-Peer "[17]. GGF was modeled after the IETF, with working groups, periodic meetings, documentation control and standards approval process. At the last GGF meeting in Berlin in March 2004, about 600 people participated.

The founders of GGF were from GRID organizations in North America (GRID Forum), Europe (eGRID) and Asia Pacific. Kellerman and Foster (of Argonne) were among the founders of the organization, and Charles Catlett of Argonne [18] became the GGF Chair (and is planning to step down later this year).

GGF has worked create an architecture for the GRID called OGSA (Open GRID Services Architecture) [19].

GGF has spawned a number of working groups, which are working to develop standards for Workflow management, resource reservation, discovery, authorization, and others. The OGSA architecture is a powerful description of high-level requirements for implementing the Grids of Computing Clusters starting to be deployed for high-end computing. It is valuable in defining what is meant by GGF style GRID computing, and provides direction and guidance in specifying lower layer Infrastructure standards.

Two decisions announced at the Berlin GGF meeting in March have changed and solidified GGF working group goals. One decision is the acceptance of OGSA as the principle architecture to be supported by work done in GGF. The other is the move to use WSRF (Web Services Resource Framework) developed by a number of people at Globus as the underlying protocol to support Grid services. The second decision basically is to take work that had been done in GGF groups and move it to OASIS to fit with standards developed by the Web Services, and particularly the OASIS Consortium, which is developing e-business standards. This seems a good decision because it means that GRID services will be implemented on a foundation accepted by a large application community, and used by many commercial services. It does mean that much of what GGF does will move to OASIS. The people who worked on WSRF outside GGF are also key members of GGRF, and it is expected that some others from GGF will move to join OASIS in addition to GGF. As a personal comment, I am not sure how well GGF will survive with the infrastructure standards work moving to OASIS; the next year will be an interesting time to see how GGF evolves.

As a brief summary of the above, my understanding is that GGF is planning to provide high level standards for GRID Computing, but is adopting WSRF as its low level implementation. High-level standards might consist of things like Job Management, Resource Reservation and perhaps high level protocols for creating and maintaining trust relationships between Grid organizations and services. Low-level standards to be taken from OASIS WSRF will consist of message protocols and syntax and security and trust management between entities.

2.2 Globus Alliance

Globus Alliance has been involved in GRID Computing since about 1997. Its purpose is to promote Grid Computing by developing distributed architectures and by developing and providing open source software. "Just as the Web has revolutionized access to information, the Globus Alliance aims to achieve a similar result in computation"[23]. With the advent of GGF Globus is committed to following GGF standards. The Globus open source software is tracks standards as know at the time of implementation, and hence as standards change the Globus software implementations may take some time to implement new changes. In addition, the Globus software implements to specific architectures that may be a subset of the GGF architecture. The Grid Security Architecture described below is and example of such an architecture.

Many of the people involved in and leading GGF are also (as one might expect) in GLOBUS. GLOBUS has developed software tools, held workshops, and collaborated on work on applications. Its software toolkit provides many tools used in other applications. IBM [24] and probably others have repackaged GLOBUS software and also includes a number of Web Services Routines. The GLOBUS toolkit and repackaged version such as IBM's are used to create distributed applications.

Globus defines architectures as well as implementing software to support them. Its work defining WSRF and taking it to GGF is a recent example. It has also implemented a number of other capabilities that fit within the OGSA architecture including GRAM (Globus Resource Allocation Manager), GSI (Grid Security Infrastructure), GridFTP, MDS (Monitoring and Discovery Service), XIO (extensible Input and Output), as well as core message passing services (which will be converted to WSRF).

As noted above, GSI - Grid Security Infrastructure is an implementation of a security architecture for GRID services. It is not a GGF Standard, OGSA says that a standard implementation must be able to operate with multiple Security Infrastructures [19 p. 11] and that "security functions such as confidentiality, integrity and authentication fall within the scope of bindings and are thus outside the scope of the OGSA proper". [19, p.17] Nonetheless, GSI is a Security Infrastructure that is compatible with OGSA and has been used in

many implementations. The Architecture section of this paper will describe GSI and its use concepts as defined in AAAArch [25] (and other places) to analyze and suggest additions to a modified Grid Security Infrastructure.

Globus and GGF have thus far focused on supporting Computing and Storage Clusters. The advent of high-speed fiber and switches has spawned work to create and support what is being called Lambda Grids. Lambda Grid work does not fit the use cases that describe most GRID Computing solutions. The next section of this paper describes work that has been done to support Lambda GRIDS and how it is being developed. Following that is an architecture section that compares what is required to support both Computing and ultra high-speed Lambda Grid communicating.

3. Lambda Grid

The creation of high performance Communication capabilities at multiple organizations, and their interconnection into a high speed fiber communications mesh has been described as creation or the Lambda Grid [16] This work is to complement the development of computing GRIDS as defined in GGF, Globus, and other places – hence the use of "GRID". This paper looks at Lambda Grid implementations as generally falling into two high level models: 1) provider controlled networks and 2) user controlled networks.

3.1 Provider Controlled Networks

In a provider controlled network the user connects to the network edge and requests a path through the network to another edge. The network is responsible for finding and setting up an appropriate path and passing traffic through the path. The provider may also manage faults and provide internal backup paths in case of a failure in the network. The Optical Interconnect Forum (OIF) has defined interfaces to support this – the UNI for user interface to the network and the NNI for connecting between network segments. The GMPLS specifications [73,74] have created a modified MPLS protocol to support creation and grouping of paths, which include fibers. Some more about this protocol is covered in a few paragraphs. The figure below shows an example of a provider controlled network. It shows users connecting to the net edge router with UNI interfaces the routers finding creating paths between network end points and connecting to users at the other end. The pictures show two connections, many more are possible.

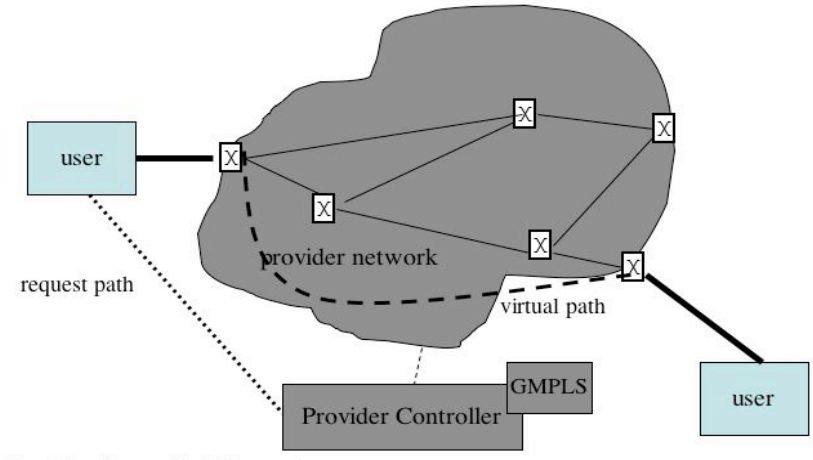

Figure 1

3.2 User Controlled Networks

User controlled networking is a concept being developed in a number of experiments to support applications which support class 2 (several to several) and especially class 3 (high performance few-few) networking. In the user controlled networking model the provider has a number of lightpaths from which it delegates subsets of resources to other organizations giving them the right to provision that subset of the provider's resources. Figure 2 illustrates a user controlled network.

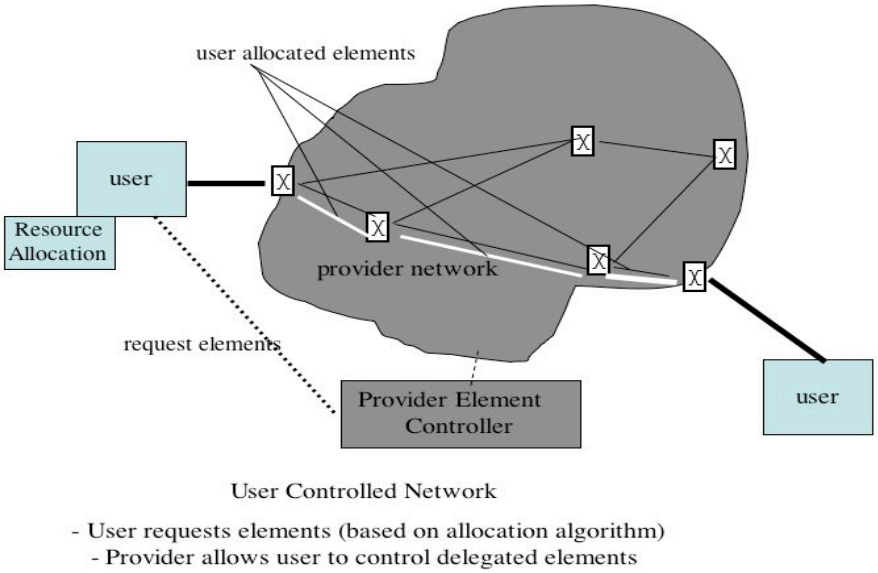

Figure 2

In Figure 2 the provider network is shown to own the network elements. The provider has multiple links between its switches (3 lines for each in the figure), and delegates to different users the right to allocate these links in any way it chooses. In fact there need not be a provider, and the switches could belong to different organizations each of which delegate the right to use specific links to particular users.

The implication of this is that there must be a control mechanism that allows a user to setup paths using the links delegated to it. In fact this is what is being done for several Lambda Grid Projects, as will be discussed in a few paragraphs. It also allows the use of fiber elements as resources to be allocated, just as GRID computing treats computing and storage elements as resources to be allocated to an application. This ability to treat Fiber links as resources may allow Grid software for controlling distributed computing to be used to interface with optical controllers. This is discussed in the Architecture section of this paper.

3.3 Provider Controlled Sercices

Provider Controlled Networking with photonic elements has been implemented in several places [29] using GMPLS [] to set up routes requested by users. GMPLS can be used with combinations of link types, copper as well as optical. It seems to be able to support aggregating IP traffic and sending it over the network, as well as providing level 2 and physical level paths to individual applications. ODIN (Optical Dynamic Intelligent Network) [22] is a service layer being designed and developed by ICAIR, which provides an interface to "photonic-empowered" [30] Applications. ODIN provides a single contact point for applications to get network resources from network provider without the application having to know about network complexity.

GMPLS allocates network resources, responding to demand, and works well within a single administrative domain. It can work across domains, but determining links between domains, if done is at a higher level than the individual path. Each domain determines the paths through its domain independently of paths

through other domains. From an optimization view this provides optimization within each domain but does not necessarily find the best path through all domains.

From a security point of view creating paths across domains requires trust relationships between domains and establishing and enforcing policy at each domain about what is allowed from other domains. This is not supported (at least at the individual links - as far as I can tell) in GMPLS. which means that GMPLS is good within domains but not between domains. It can be thought of as an internal routing protocol but not an external routing protocol.

In the next (architecture) section of this paper I discuss the possibility of using GMPLS in conjunction with Virtual Organizations. It seems a powerful possibility.

3.4 User Controlled Services

User Controlled Networking for photonic devices allows the user (or user agent) to select elements of a path and directly control switches to create it. In User controlled networking the network selection is done by the user or user agent. Several implementations support user controlled photonic networks. Some implementations of this concept are discussed below.

The most complete is provided by Canarie as its User Controlled Light Path (UCLP) [60]. Canarie has created a set of tools that allow users to allocate specific resources in the Canarie network, delegated to the user by Canarie.

The *Canarie Service* uses a Proxy between Web Services on the user side and TL1 on the Network Element side. The Canarie Service's Proxy server checks whether a requestor is allowed to use the requested resource and if so creates a TL1 request to the Network Element. The service and provides the ability for Grid/ Web Services applications to tell the proxy what to do on the resource provided to the user, while protecting other resources from being used. The Proxy issues commands that control Photonic elements, and presumably could be used to control other devices. I am not familiar with all the details of this implementation, but it seems a very good architectural approach, combining Grid/Web Services on the user side and device control on the network side. This seems a good basis for future development.

The University of Amsterdam and the University of Illinois at Chicago have each created servers, which control the creation of photonic paths across several links and across several administrative domains..

The Amsterdam software, called the Generic AAA Toolkit and the UIC server called PIN (Photonic Interdomain Network) both directly control photonic devices.

The Amsterdam and UIC controllers have worked together to support a common application, coordinating with each other but each controlling elements in their own domain. The resource allocation for these implementations have been relatively simple, and built into the application. [33]

The *Amsterdam Generic* AAA Toolkit software [74] supports a policy language [31], which combines the ability to evaluate expressions and to initiate actions. A prototype implementation of the Generic AAA Server has been documented as an IETF draft [32]. The Generic AAA server maps a "driving policy" with each request type, and uses the driving policy to authenticate and activate a set of devices as defined in the policy. The server is able to send requests and commands to other servers, possibly in other domains, to carry out policy defined for that request type in the remote domain. The current implementations use SOAP/ XML to communicate between servers. Next versions will move to support a full Web Services protocol between elements, just as is being done in Canarie. A prototype Generic AAA server has been used in demonstrations to show the ability to control devices across multiple domains [34]. In the next major (Architecture) section of this paper there is discussion of how this might be integrated with WSRF/Grid capabilities.

The *UIC PIN* server has some of the same characteristics of the Generic AAA server, but is oriented more directly to the control of multidomain devices and providing a policy for selecting and allocating devices. I have no published documentation of this, but my understanding is that PIN accepts requests for

connections between ports on equipment it has the right to control. It has a policy capability that supports allocating resources based on permissions for individual resources assigned to groups of users. PIN includes a table converts users to groups, and defines rights of groups to use particular devices. When a request is received PIN checks the user, then gets the group for the user. It then determines what, if any, path is available physically as well as permitted to the group. It then selects and implements the path. PIN policy seems a good mechanism for relatively small sets of photonic resources.

Note that the Canarie service is one which makes it possible to delegate control of individual elements of a network to other users. The Amsterdam and UIC services are aimed at selecting and specific elements and allocating them. Thus they could allocate resources delegated to them by Canarie.

All the implementations provide a way to control the photonic equipment. The Canarie method uses TL1 and seems more general that the direct control from other methods. I am not familiar enough with photonic equipment to be sure, but it seems that the Canarie proxy is quite general, and might well be incorporated by the other implementations as a way of actually controlling Network Elements.

The Canarie Service provides the ability to activate individual delegated network resources. The other methods provide tools for selecting and choosing the resources to be activated. The proposals section in this paper suggests building on the Canarie concepts and incorporating scripting capabilaity of the Amsterdam GAAA and allowing the policy capabilities of the UIC PIN.

4. Architecture

The first part of this section describes and discusses specific concepts that will be used later in defining potential projects. The concepts include 1) implications of the distinction between provider-controlled networking and user controlled networking, 2) Virtual organizations to support collaborative projects, 3) trust relations between domains, 4) driver policy for define and controlling applications, 5) allocation of resources across applications with and without delegation, 6) GMPLS as bandwidth policy language, and 7) attribute definitions and meanings across domains.

4.1 Provider/User Controlled networking

Provider-controlled networking and User-controlled networking differ principally in who ultimately controls allocation of network resources to specific projects. In provider-controlled networks the user asks for network resources, either by presenting traffic to the edge or by requesting specific bandwidth. The provider determines how and whether to fulfill the request.

In user-controlled networking the user either owns or is delegated the use of certain resources. The user then determines how to allocate resources to resolve a particular application or set of applications. The resources to be controlled may be owned by the controlling organization, may owned by a collaborating organization and delegated, or be owned by a provider network and delegated (perhaps via lease) to the controlling organization. The controlling organization makes the decision on how and whether to allocate the resources.

Provider-controlled networking examples include best-effort internet, frame relay, and implementations of implementations that provide VPNs or Fiber Ethernets [34]. In all these examples, the provider determines the internal path of the connection and determines who gets priority if there is a conflict.

User-controlled networking examples include academic experiments where multiple Universities collaborate in experiments, each contributing resources, including fiber links to the collaboration. The collaboration then control the contributed resources. In creating a network path the user allocates the links to create a desired path. If it is allocating delegated links then the delegating organization will check that the allocation is within what was delegated in order to control the delegation of resources.

In practice, it seems likely that a particular application will not allocate resources on its own but will go to a resource allocator, owned by the user's organization or by a group collaborating to perform this and

perhaps other applications, to request and activate resources. The availability of a server with the ability to allocate resources and to provide assurances of delegation be discussed in the final section of this paper.

The following fits these cases to the class A, B and C applications described in the first section. Provider controlled networking seems to be oriented to supporting communications for a large number of people as efficiently as possible. It seems to be good for class A (many-many) applications and to applications that have relatively small communications requirements.

User controlled networking seems well suited for class C (few-few) applications where the volume of data is high. This allows the user organization to make decisions about which applications to run on which elements, optimizing the implementation resources to optimize applications rather than the utilization of the network.

Class B applications may work on either. In practice providers have been working to create solutions which allow users to create paths to carry data across a network in addition to routing packets hop to hop through the network. These hybrid solutions do not give users control of individual network elements, but do some provide QoS guarantees[]. Where paths have segments across multiple providers, it seems possible for the user to control allocation of each path segment but not allocate resources that create the segments. This would be a hybrid of the provider and user controlled networking.

The focus of projects defined in this paper are to support User Controlled Networking, including the ability to request and maintain path segments from network suppliers using GMPLS or equivalent methods internally.

4.2 Virtual Organizations

Virtual Organizations (VOs) have been proposed by Globus as a way of supporting multi-domain networks [23.] This is a powerful concept which allows organizations that may want to collaborate on a project to create a temporary Virtual Organization to support the collaboration and to create policies for the VO that allow certain [collaborating] people to use certain [delegated] resources. Rather than each participating organization tracking which user from which domain is allowed to use a shared resource, each collaborating organization delegates the use of certain resources to the collaboration's VO, and/or assigns people from its organizations to the collaboration. A Virtual Organization uses resources that are delegated to it and supports users assigned to it

A Virtual Organization is the same as a real organization except that it is typically temporary (from seconds to years), and its resources are delegated from other organizations, and the members are members of a real organization. Virtual Organizations are supported by Globus, and can be supported by other software. The following gives a high level view of what Globus does and adds some possibilities not supported by Globus.

Creating a Virtual Organization requires that it be delegated resources, and that users from other organizations be assigned to it as members. These are described below.

4.2.1 Delegating Resources to Virtual Organization

In an environmnent supporting delegation, the "owning" organization may delegate one or more resource to an "operating" organization. The intent is to give the operating organization responsibility for allocating certain resources, while at the same time allowing the owning organization to check that each allocation is done by an organization to which that capability was delegated and withing the limits of what was delegated. This gives the operating organization the ability to apply any resource allocation algorithm it desires to reaources delegated to it, while the owning organization retains control (or veto) of the overall use of the resource. For example, University A with 60 Lambdas between node A and B may delegate the use of 9 Lambdas to VO2. VO2 can allocate up to 9 Lambda in any way it likes. When VO2 requests that A activate 5 Lambdas, A does a "delegation" check and then does the implementation. If VO2 were to then request an additional 5 Lambdas, A would check, find that this request would create a total of 10 Lambdas and that it had delegated only 9. It would then refuse the request (whether it refused all or implemented 4 of the 5 requested would be a matter of policy).

To support delegation of resources, an ability for an "owning" organization to delegate the resource and then to check that an allocation is for something it delegated. One way to do this is for the owning organization to delegate by giving the allocating organization a sign a certificate containing the resource, the allocating organization id and public key. When the allocating organization attempts to allocate the resource it sends sends a request signed with its private key and the "resource certificate". The owner uses the key and resource information in the first certificate, uses the user public key to check the second key, and then (if necessary) checks to see if requested resource is "allowable". The "if necessary" here is because if the user requested and is delegated port 2, then the check may not be necessary. If it is allocated 3 Lambdas and the request is for 2 Lambdas, then a check is made to see if 2 or more Lambdas have already been allocated by this user.

4.2.2 Assigning Users as members of VOs.

Users assigned to the VO want to use their local identity to access the VO. Several ways of accomplishing this are possible. Globus has proposed two ways – a mapping process that creates an entry in a VO directory with its "global" identity and an internal identity [] . When a user requests a VO resource the VO checks the validity of the identity and then checks it entry to see if the requestor is in the VO directory, and if so maps the requestor to an internal identity to use with the requested resource.

A second Globus method is to use the Globus CAS (Community Allocation Server) to get a proxy ID. When a user wants to use a VO resource it goes to the CAS and gets a proxy Certificate, signed by CAS and with attributes signifying which resources the user is authorized in the VO. The user presents the proxy certificate to the VO, which can then check it is from a valid CAS and then allows the user to request services. To reiterate, a proxy certificate is one that is signed by the VO and which contains a "proxy" attribute, a proxy-id and the id/ public key of the user. The user identifies himself by signing a request with his own public key and sending the proxy certificate with the request. The VO checks the request by checking proxy certificate and using the public key from the proxy to check the signed request.

A third approach not supported by Globus but suggested by the use of Shibboleth is to create a request with a certificate from the user's organization which verifies that the requestor is from the user organization and has the right to request VO resources. This is similar to the second method above, but in this case the certificate is created by the user's home organization while in the case above the proxy certificate is created by the CAS probably affiliated with the VO. In the second case, where the certificate is from the user's home organization, the VO will need to verify that the home organization is allowed to use the resource and that requests are within the limits of what that organization has been authorized

4.2.3 Creating a VO.

The creation of a VO requires certain cooperative elements to support the delegation and control of resources and the allocation of users and user rights between the VO and the organizations contributing to the VO. In particular, the VO must have a way of accepting delegated resources and of recognizing valid users, as described above. In addition, the supporting organizations must have the corresponding abilities. This is described in more detail below.

4.2.3.1 Delegation of Resources

When delegating resources, one must be able to delegate a resource to the VO, (if necessary) give the VO a certificate proving it was delegated the resource, and the owning organization must have the ability to know the request came from the VO and (if necessary) the request is within the limits of what was delegated. This assumes a relationship where what was delegated to the operating organization is part of the policy of the owning organization.

In Globus this can be done by creating and internal mapping of what is allowed to each user, and if the requestor is allowed to use a resource, mapping it into an internal ID that has that privilege. This simplifies what changes have to be made to support this service. It allows some control of what is allowed to each user, but does not give control of delegation in any easy way.

Support for delegation requires delegation control policy to be implemented at the owner organization at the same time that delegation to a VO is done. This permits the owner to check requests from the operating server and allow valid requests and deny requests outside the limits of what has been allocated.

4.2.3.2 Assignment of users

Globus requires all users to have an identity certificate, and that certificate can be used by the VO in one of two ways: the VO can map a global name into an internal name, with the mapping being part of the VO; or a VO can include a Community Access Server which will create a "proxy certificate" [75]. The proxy certificate is one signed in the CAS [], recognized by the VO, which permits the user to have certain rights in the VO. The VO may not need to map the proxy ID to an internal ID since the proxy certificate includes the rights of the requestor. Both of these methods require that adding a user include mapping a global user id to a set of internal rights, The mapping might be internal to the VO or it might be done in a CAS [76], depending on the implementation.

In addition to the Globus approaches, it is possible to have the user organization provide a certificate defining what the user is permitted in the VO. In this case, the user organization would need to know what to insert in the user certificate, and the VO would need to know what members of the user organization are permitted to request.

4.2.4 Virtual Organizations as a mechanism to support resource allocation

The use of Virtual Organizations to support multidomain resource sharing is a very powerful tool. One can imagine creating a Virtual Organization to support a specific collaboration – e.g. to support a group of physicists working on a specific grant. Without a VO to control delegated responses, the collaborating group when, for example, attempting to set long distance bandwidth, might try bandwidth broker techniques. These have, for backbone use at least, have been very difficult to support. The use of GMPLS makes the QoS across backbones possible, but even that does not allow the user to examine resources and apply them in a way to optimize a set of applications with different requirements and different execution strategies.

In this paper, the projects defined support the creation and control of Virtual Organizations and the supporting infratructure required in organizations delegating resources to them or assigning users to them. In addition, the projects assume that owners of VOs may want to have control of allocating the resources delegated to them. For example, a VO might have some specific resources delegated to it – e.g. it may have Lambda paths between 4 universities and 20 computers at 3 universities. It may also have access to a set of resources available to it such as best effort access to other computers and resources. The proposed solutions include the possibility for the collaboration to allocate delegated resources in a way that is more effective for a specific set of problems than might be available from standard allocation methods.

4.2.5 Virtual Organizations and support for multidomain applications

Virtual Organizations are a way of supporting multidomain applications. In particular it gathers resources and people from different domains, and support delegation of resources to the VO and allocation of users to the VO. While it is one way of supporting multidomain services, it is not the only way. A more general way would resources to be allowed to for certain users or organizations and have requesting organizations be given access to what they are permitted. The AAAArch research group has described ways to support the general case [25].

However, a VO is a subset of the general case that is extremely useful. Its delegation of control of resources by multiple organizations to a VO creates the possibility for a collaborating group to create VO to manage resources for that collaboration. Creating a VO requires the ability to delegate resources from one organization to another (possibly virtual) organization.

4.3 Security between organizations

Two Security concepts are discussed here. One is user authentication, the other is security of authorization information between systems. The first defines how the user and system are known to be who they claim to be, the second is how do I trust that information came from whoever it claims to be from.

4.2.3 **Authentication**

When two organizations want to exchange information they need a way to prove that a message one of them receives is really from the other. In order to do this on a computer they each must have a credential and a way to prove to the other that they own the credential. In some cases the credential may be between the two organizations, as when a user and his computer have a shared secret (e.g. a password) that is known (only) to each of them. (Note that sending an id/pw in the clear is considered bad form, but there are other algorithms which support shared secret without that security problem [42,43])

In cases where the user wants to access multiple applications with a single identity, some trusted third party knows the user and the application. Both Kerberos [37] and PKI [41] create credentials vouching for the user identity: Kerberos creates "tickets" and PKI methods create "certificates".

Web Services does not specify a specific authentication type, but does use credential, and may require information about "Authentication Context".[61]. The Globus GSI requires that users get a Public Key certificate that is granted by a known Certificate Authority. Thus this is more specific than what is required by general Web Services applications.

To be able to use applications requiring Public Key, a process to create a public key for Kerberos users has been developed at the University of Michigan [59].

A service called Shibboleth has been developed by a umber of universities and Internet 2 [62]. Shibboleth allows a user to authenticate to its home network, then has the home network create a credential which it passes the user who then passes it to the application. As far as I know this is not supported in Grid computing, but it seems that it the credential provided by Shibboleth could be adapted to be used by Web Services, and that the credential could be turned into a PKI Certificate to be used by GSI. This latter turning into a PKI Certificate would be equivalent to having the Community Access Server in Globus create a proxy Certificate based on a user Certificate.

These concepts seem to make it possible to use most common forms of third party authentication for Grid applications. Future work could develop this concept further, integrating common authentication methods into standard Grid Implementations.

4.2.4 **Trusted messages in trusted sessions**

When two entities want to set up a trusted communication a typical sequence is that they will authenticate each other as noted in the section above, and as part of the process of authentication create a common "session" key at each entity. This common key is used to encrypt and or sign messages between the entities.

Web Services [37] defines methods for communicating using messages between entities. Message between Web Services Entities may be protected by protecting the message itself and or by creating protected sessions between entities 9as described above) and sending messages over the protected paths. In some cases both methods could be used.

Web Services protocols are being developed by OASIS, an industry consortium creating standards. WS-Security is a part of Web Services and was recently approved by the OASIS board []. WS-Security defines how to use different security frameworks (e.g. PKI and Kerberos) to provide message integrity and encryption. SAML (Security Association Markup Language) also produced by OASIS defines a "framework for exchanging security information between business partners" [40] Using tools provided by WS one can create mesh of applications that for example, allow one to request a book, have the book provider check the availabilty in an inventory server, have the book provider check the payers credit on a credit server, have the provider initiate the book transport, and initiate a credit card transaction with the

credit card server. The sequence of messages can be initiated by different servers requiring information from other servers.

One can think of Grid Services as an instance of Web Services. In Grid Services the service being requested is typically a session with one or more resources. (e.g. 3 light paths, 4 computers, and two data sets) possibly in more than one domain.

The Grid Services sequence might be that: 1)the user generates a requests for a session that includes 3 resources, one from each of 3 different domains. 2) The user sends the request to a application server; 3) the application server asks a user profile server if the user is allowed to use the application and required resources. 4) If it is allowed, then the application server in each domain is queried to determine if the request is acceptable. 6) If that succeeds, the application server initiates "sessions" on each of the three resource servers.

In proposals for future work a variety of security associations will be supported and Web Services messaging will be used to send requests and responses between entities.

4.2.5 Globus, Grid Security Infrastructure, and AAAArch authorization models

Globus has defined a security architecture that requires PKI structures. Within this it has created an extension to standard PKI for Proxy Certificates. Adding proxy PKI "attribute" certificates provides a mechanisms for adding user permission attributes. In this model, one presents a PKI certificate signed by a trusted party to a resource provider to get access to the resource. The proxy certificate includes a "proxy" attribute, and a "rights" attributes given to the owner by the creator of the proxy. The proxy may be signed by another user with rights to the resource(s) or by a (Virtual) Organization's CAS [76].

The mechanism for getting a resource is 1) user gets a proxy certificate with "rights", 2) user presents the proxy certificate to the resource, 3) resource checks the validity of the certificate, and 4) the resource checks its policy to see if the request if acceptable. In terms of AAAArch concepts this is a push operation. Note that this works with standard certificate if proxy is not needed so changes to existing resource support code is very small, and well understood.

In a pull authorization the user would request the service, the service would check with the Proxy creator and get a response directly from the authorization authority. This is typical of network access authorization requests using RADIUS (or Diameter) [77,78]. It can provide performance gains over the pull method because the service can authenticate to the Proxy creator and and create a secure [e.g. SSL] channel. It can use the secure channel to send data rather than use more expensive public key methods to secure messages. Note that one can use proxy certificates in this method, which (as far as I can tell) is not supported by Globus.

In an agent authorization the user would request the service through the authorization authority, the authorization authority validates the user and requests the service. This is typical of the COPS [79] applications that provision a device based on requests from a user. It can provide significant performance gains by establishing a relationship with the authorization authority and then using a secure channel to communicate rather than protecting each message with public key techniques.

4.3 Driving Policy

When an application requires multiple resources and services, a way of describing the application in different places is needed. The Generic AAA group in Amsterdam has defined a "Driving Policy Language" (DPL) to support this. DPL is incorporated into the Amsterdam Generic AAA toolkit, and has been demonstrated in authorizing and initiating Lambda paths that use segments from different organizations.

Driving Policy is not the same as the policy that evaluates a set of attributes using a specified rules. Driving Policy does permit evaluation of attributes against rules, but also defines a way to get attributes, and to take actions based on the results of decisions. DPL provides a way to describe a set of activities to

be performed as part of a particular application. Others are working on similar ways to do this in OASIS [53] and GGF [54]. DPL provides tools in this area that are relatively easy to use. Where appropriate and possible, DPL should track evolving standards.

The use of rule based policy such as described by Matt Blaze [50], and that being described by XACML [51], is not required in DPL but can be included when needed. DPL does provide a way to do scripting of a transaction and includes simple logical operations but does not enforce the rigor described by Blaze or include the organized description of elements and rules in XACML. In my mind, DPL is a language that defines a whole set of "conversations" and can include the above concepts when appropriate.

DPL implements the concepts of the Generic AAA Server RFC 2903 [70]. It does logical operations itself and can call a method of method of evaluating other rules. It can respond to a request by performing actions on the basis of its evaluation, getting results and doing additional evaluations possibly based on the results of the actions. The actions in the description are "generic" and are performed by a "Application Specific Module" (ASM). In the implemented demos [34] a number of ASM's have been implemented, and a widely used server would need to have several capabilities implemented as standard, rather than as ASMs. Note that the a number of functions could be provided by "standard" ASMs that are provided with a server.

DPL is a powerful language for scripting applications that require operations on multiple devices. It fits well into the Web services [81] model where it acts as the central "Service" and uses Web Service messaging to communicate with other "Services". In the Grid model communication paths, computers, data sources and other resources are implemented as services. Currently DPL is implemented in Generic AAA servers with ASMs that talk to other Generic AAA servers. An interesting research project would be to implement DPL as a Web Service calling other Services using standard Web Service protocols and evaluate it as a way of controlling Lambda and Computing Grid applications. This is discussed further in the next major section "proposals".

4.4 Allocation of Resources

In Lambda Grid environment users and user applications control resources. Scheduling is tricky, and a number of mechanisms have been implemented to support it. In computers one has task schedulers, in computing clusters one has meta-schedulers like Condor [64] or Rock [63], and in IP networks one has best effort routing and some path scheduling using MPLS. All these assume a set of resources controlled by a single organization (and all are good in the single domain environment).

Managing a set of resources across many organizations is a difficult problem, and a subject of continuing research [54]. For example, suppose one has a set of resources available from several organizations and creates a Virtual Organization to control these resources. Suppose also that one has another set of applications to be performed using these and other publicly available resources (e.g. best effort Internet). Finally assume that there are different algorithms available to execute the applications, depending on what resources available. Finding a solution that applies available resources to complete the applications in an optimal time is not trivial.

A specific difficulty when combining communication resources and computing resources is making some sort of tradeoff between them. For example, if one has access to several lambda segments, many computers at different sites, and data available from different sites, one must determine which sites are to be connected, possibly at different speeds.

For the short term at least this seems an area where further research will be done. The proposed projects include the creation of a "stateful" resource repository as part of a Virtual Organization and a way for (experimental) applications to read and reserve these resources.

4.4.3 **Allocating delegated resources using GMPLS to other protocol**

Generalized Multiprotocol Label Switching [55, 56] has been developed to "adapt to the peculiarities of photonic switches" [55, p. 146]. GMPLS is being used to control fiber networks [30]. It is included here

because it is an example of a way that Network Providers can control Optical Paths within their networks. In general this results in the situation where the user requests network resource and the network determines how and if to provide the resource.

Canarie has turned this concept around [57]. It has implemented a mechanism in which the user may be allocated certain optical resource and is able to allocate them directly. In this case the user may use GMPLS to determine how to allocate resources and then the network (through a MonFox Proxy [58]). The implementation of the MonFox Proxy allows the Network provider to give users control of "delegated" resources within in the network, and the user controls how the resources are combined to create the desired paths. This lets Canarie delegate the control of specific resources to other networks without compromising the use of other resources. It also allows Canarie customers create their own network using resources delegated to it by Canarie. This seems a very powerful concept.

The point here is that a (Virtual) Organization may be delegated resources from another organization. controls those resources as if they were their own. In the case where Canarie delegates optical resources to Enterprise A, GMPLS can be used to allocate the delegated resources from Canarie along with other resources it may control. In this case GMPLS becomes a resource allocation algorithm for the (Virtual) Organization using the resources. Other existing algorithms may also be used.

4.5 Interfacing between Applications and Resource Managers/ GRAM

The OGSA architecture assumes that resources are tied to the Grid infrastructure with WSRF interfaces. Most devices do not understand Grid/ Web services commands, so an interface to them is needed. Globus supports software called GRAM (Globus Resource Allocation Manager) [48] which is interface between Grid applications and device controllers, and requires the user to write (or borrow) a code that maps specific Globus Commands to what is required by the Controller. GRAM thus ties the device to the Globus/Grid infrastructure

The concept of a common module that will interface to device controllers is important. It creates a common way to support devices, and in the Driver Policy Language it allows all devices to be treated a "GRAM" devices, with the specifics of the device hidden from it.

Further discussion in the next section proposes using the GRAM interface (evolving it if necessary) to support user-controlled network device allocations.

4.6 Attribute meaning across domains

A major issue with the cross domain authorization is the meaning of particular attributes. For example, the meaning of "Student" at one institution might mean any who has been enrolled anytime in the past 3 years, while at another institution it means anyone who is currently enrolled in one or more courses. The difference in meaning can be a serious problem when trying to make decisions based on the value of attributes from different organizations.

Attempts to solve this problem include creating a set of common attributes across a set domains [Shibboleth, SAML], creating a "attribute translation" mechanism used to interpret discussion between organizations, creating unique attributes within a (Virtual) organization used by anyone requesting service from that VO, and structuring services such that attributes are only evaluated in their own domain. A brief discussion of each of these follows.

Creating a common set of attributes as is done by Shibboleth [] allows a set of users to make decisions based on attributes associated with the request. One reason for Shibboleth to do this is to allow anonymity of the user while providing information about the class or capabilities of the user. The creation of a set of attributes (like edu-student) is the creation of a "Virtual User" set, applicable within the set of institutions that agree to use the definitions. Note that someone from a different instution could use the same attribute but with different meaning. The attribute in Shibboleth's case is assigned by the user's institution, and users of the attribute must check that the institution is one which has agreed to the set of attributes.

The use of GRAM as a common interface to all devices creates a similar situation. In order for this to work there needs to be a common set of attributes (e.g. status= on). But there are likely to be other attributes that are specific to the device which will different between devices and device types. This commonality creates the ability to perform common functions across all devices, while the unique attributes provide the ability to support additional features. This is similar to what is done with RADIUS where certain attributes are defined to have certain meanings, and there are "vendor supplied" attributes which have meaning for specific device types or applications.

Creating an attribute translation mechanism between domains allows one to define a meaning for a particular attribute when used between the domains. For example, the attribute "blue service" when used by domain 1 to request service from ISP-A could mean "this is permitted to access the entire network", while "blue service" between domain 2 and ISP-A could mean "this user is permitted to access only company specified sites on the internet". The creation of such a translation is done as part of the SLA agreement between organizations.

Structuring services such that attributes are only used within their own domain implies that one does not pass attributes between organizations, but my pass queries. A query might be "is person in this request authorized to read this document?" In this case the organization doing the querying sends information from the original request to the domain that owns the user of the request. The attributes in the request are from the same domain as the queried domain so no translation is needed. The response is a simple yes or no.

Note that combinations of all these strategies may be used in a particular service. The section of proposals will discuss this further.

5. Proposals for future work

This section lists some proposals for future work, expanding on work described in the previous sections.

The general goal of all the proposals is to provide capabilities that build on the standards of Grid computing and communication and to use information gained from implementations to inform future standards. Wherever possible they use concepts from Web Services [80] and the Global Grid Forum, [17]

5.1 Design and document methods to use Kerberos, Shibboleth and Shared Key authentication in Grid Services applications. The intent is to make Grid Services available to organizations that do not support Public Key infrastructure, and to make capabilities of Shibboleth available to Grid Applications.

5.1.1 Implement one or more or these with a partner, perhaps UM for Kerberos and Internet 2 for Shibboleth.

5.2 Proposals for creating a software server
These projects include writing some unique software, and integrating software from the Generic AAA toolkit with software from Globus, Canarie, and others. Specific projects can be done independently and eventually combined to provide a general server that can be used by organizations and Virtual organizations to support Grid applications. Pictures of the integrated Server, which I have at least temporaily named Grid Organization Application Deployment Server (GOADS) are shown at the end of this section. Section 6 describes an integrated set of GOADS Servers to support multidomain Grid Services and includes a picture of a possible system of such Servers.

The following are descriptions of specific projects which can be implemented in an order appropriate to specific pilot projects. All will be built with a common software communication and control core.

5.2.1 Create a Multidomain resource controller using the Generic AAA toolkit as a base that controls optical infrastructure using the Web Services and MonFox Proxy capabilities of Canarie. Use this to

control Fiberpath allocation within and between domains, perhaps in Chicago, Michigan and Amsterdam. This is an extension of existing Generic AAA Server to include MonFox.

5.2.1.1 Use the controller to get usage and performance information. Modify the Server based on learning from the needs of users in operational system

5.2.2 Extend the Generic AAA software to be a Web Service. Use concepts from AAAArch Research group to create intra and inter domain coordination between servers. Incorporate PIN scheduling capabilities to support allocation management.

5.2.3 Build a Virtual Organization Server to support the creation and control of a Virtual Organization. Include the ability to keep and share state information with (possibly experimental) allocation processes, and to accept allocation decisions from external processes. Include the ability to add and delete users, accept delegated resources and ensure that the delegating organization agrees to the delegation. Also include the ability to create and manage stateful session information, and the ability to find other resources and request resources from them.

This will create the ability to manage delegated resources by the Virtual Organization and to make the capability to manage the resources available to researchers studying different allocation algorithms.

Create an Application Server that incorporates supports the creation and execution of Driving Policy to describe the operation of multi-resource, multi-domain applications. This will allow applications to be stored and run on the server, making it possible for users to execute or modify a job from a remote device with limited computing capability that is not part of the Web Service framework.

5.2.4 Implement a Service using above Servers to allow control of Computing and Lambda Grid components. Use the implemented software to support applications with resources owned by one or more partner. Controlling lightpaths between Amsterdam and Chicago is a simple application.

5.2.5 Create a Resource Server that communicates with the Application Server and interfaces with device drivers. This supports standard requests for resources and applications and translates them into commands appropriate for the device driver. The Resource Server also keeps track of session information for devices or applications, and applies its policy to requests for services. The Resource Server is controlled by the organization that owns the device, so policy may check for specific rules like time of day for requests. The MonFox Proxy seems to provide some of this capability, and may be used in situations where it is appropriate. [Note: the intent is that the Resource Server could support a “virtual resource” such as an overlay network, and call other resource servers to get actual resources.]

5.2.5 Add session and resource tracking and other usage information to the server. Work with experimenters to define and collect types of information useful to actual experiments. This could be very useful in providing tools for Lambda Grid types of information.

5.2.6Define a set of attributes that can be used across domains and across providers to support different classes of resource. Defining Lambda attributes would be a good example.

5.2.7 Design and implement an architecture of distributed Web Services that can operate together to support Grid AAA capabilities. This would include how to name requests and find servers to support them, how to authenticate users, how to create and distribute usage and fault information, how to track and coordinate useage of resources, and how to define and control applications.

5.2.8 Make a server publicly available – perhaps like Apache – that provides a implementation of the Grid capabilities. This would be build using Globus tools where possible, but would act as a server, not as a resource provider

5.3 Block diagrams of Grid Organization Application Deployment Server.

The following diagrams show the Grid Organiaztion Application Deployment Server and Associated Local Resource Controller. Figure 3 shows a block diagram of a server with all capabilities included in a single server, with the ability to communicate to other servers using Web Services to get additional resources. Figure 4 shows a configuration of Servers that allows a high level server to control applications that use resources shared across organizations. Discussions of each follows.

5.3.1 Elements of Grid Organization Application Deployment Server

Figure 3 shows a block diagram of the proposed server. The following are elements used by the server.

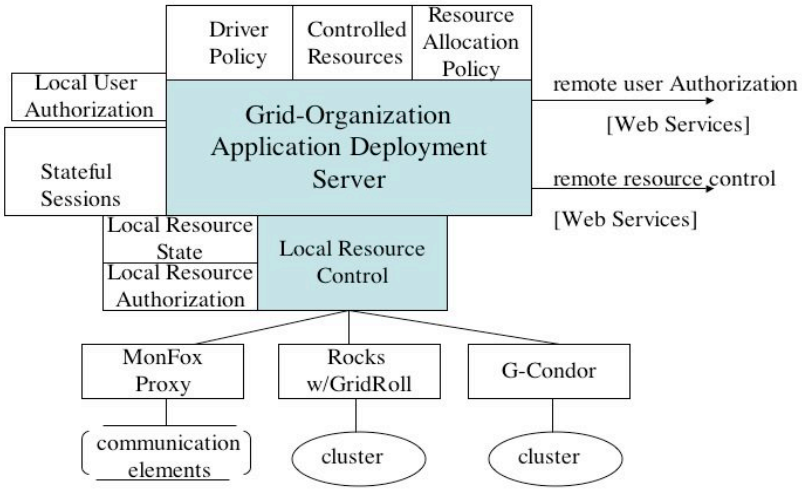

Figure 3

a) Driver Policy describes each application and the steps needed to implement it. This is based on the Generic AAA Server. Routines to create and modify the contents of the Policy repository will be created.

b) Controlled Resources are all resources owned or delegated to the organization. Routines to create and modify contents of the Controlled resources repository will be created.

c) Resource Allocation Policy describes rules for allocating resources. This policy will use information from the Controlled Resources repository as well as the Stateful Session information. This will be built based on existing resource management routines such as PIN, or by other mechanisms. For some networking applications this could be GMPLS. [Note: In virtual networking applications where the a user is delegated certain resources in the provider net, the user may be said to create an "overlay" network]

d) Local Resource Control is a part of the Server that interfaces to other Resource Controllers. It provides translation between the resource control commands from the GOADS server and the device controller.

e) The local Resource authorization module checks requests for Resource use and determines if the request is allowable. Checks can be made based on attributes of the user and/or on delegation of resources to another organization.

f) Stateful Sessions table keeps track of sessions, both active and inactive. It can be used for tracking and recreating resource usage, and for planning future resource allocation. It is a new feature, but can be built on the capabilities of the Generic AAA toolkit.

g) Local User authorization provides in formation about each user that is authorized by this Server. Note that authorization could be done at other servers, using Web Services communication methods. The

method of naming and finding the remotes servers is to be defined, but might use the NAI structure defined by IETF []

5.3.2 System of GOADS Servers to support fiber allocation across domains

The servers shown in Figure 4 include a high level Virtual Organization Server which controls resources delegated to it, and Resource Control Servers owned by the two cooperating organizations – in this case Amsterdam and UIC. The servers communicate using Web Services methods.

The operations is as follows: A user sends a request for a light service to the VO GOADS server. The server selects a Driving Policy based on the request. Based on the Driving Policy, it sends an authorization request to the appropriate authorization server at Amsterdam or UIC. It calls the resource allocation policy routine to select appropriate resources for the request. It sends requests for resources to both the Resource Controllers at the organizations.

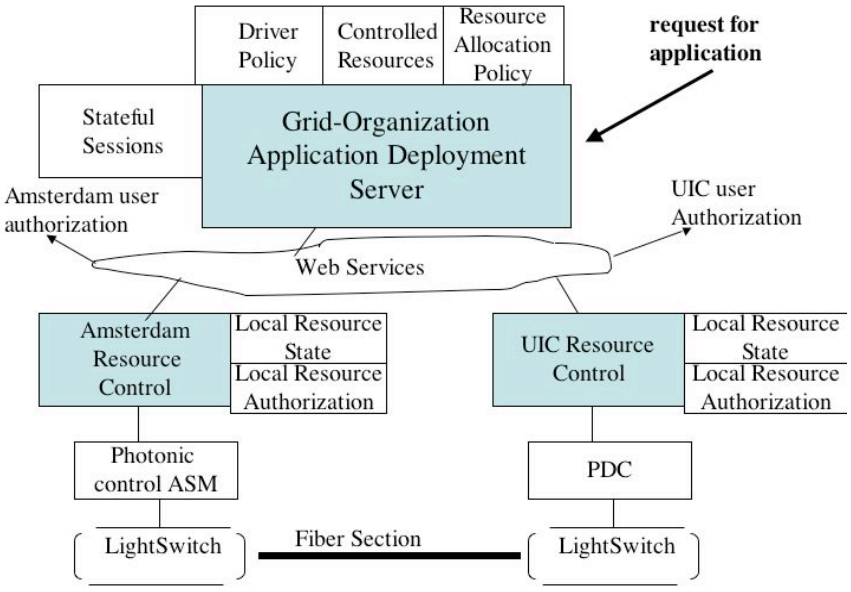

Figure 4

Each Resource Controller checks the requests it receives against its own policy, and if the check is positive it allocates the resource.

If the request is successful a session is created in both the stateful sessions table of the VO server and in the local resources state of the Resource Controller. If a failure occurs any resource allocation must be backed out and information stored in an inactive record.

Note that the resouce allocation will use methods being defined and developed in WSRF, the communication method that is being added to the OASIS Web Services, and which is going to replace and upgrade work done on OGSI by the GGF. [83].

6. Proposal for a Grid Organization Application Deployment Server

6.1 GOAD Server Overview

This section contains a proposal for developing a set of cooperating servers that will support Grid applications that may have users from many organizations, as well as the rights to use resources from many organizations. This server can be configured as a high level as well as as a low layer resource manager, as described in the section above and shown in figure 4 [The name (GOAD) is still preliminary]. The intent of this section is to present concepts for future projects as a way of initiating conversation with potential collaborators

The proposal is to use the server to control fiber paths between UIC, Amsterdam, University of Michigan, and others at Starlight. The same concepts can be used to interconnect with the Canarie Canadian Network or Lambda Rail.

The proposed servers will support the use of delegated resources by a VO/Application server and to check the resource utilization at the Resource Control server which oversees the use of the actual resource for the provider. It will work for permanent organizations and for Virtual Organizations.

Each server is part of a set of interoperating servers that support organizations that are collaborating to solve problems (either academic or commercial collaboration would be possible). A server communicates with other GOAD servers using Web Services messaging. It will use and track standards developed by the OASIS Web Services organization and by the Global Grid Forum. It will use free software from Globus and IBM to implement this where possible.

The server will support user authentication with a number of methods, including Globus user authentication, Kerberos, Shibboleth, and perhaps some EAP methods. It will also allow id/pw authentication to support existing clients.

The server will provide a repository for application scripts, called “driving policy” [32], and will have capability to build and edit these on the server itself as well as to move scripts to the server that are developed on other machines. The “driving policy” will be an extension of what is in the current Generic AAA toolkit.

The Server will initially be built and tested to run on a standard Linux platform. It may run by itself or may run with other applications on that platform. A client will be provided to make application requests to the server. The client uses a web interface to format the request. In addition to the client, a libraries will be provided to allow other applications or scripts to make a request.

A web-based client will be provided that allows a user to create a temporary organization, applications for the organization and list of permitted users or sets of users in the temporary organization. It will also provide the ability to delegate a resource from one organization to another: this will require changes to the “policy” for both organizations.

The Generic AAA toolkit will be extended to include and display tables of trust and security associations with other organizations and with other network elements to which the server communicates. The tables will contain information describing the network organizations with which it has a trust relationship based on contract or federation and those which it has a permanent or temporary security association. It will also contain information about specific elements with which it has or has had a security association. As an example of a difference between a trust relation and security association, UIC may have an agreement with Northwestern that allows sharing of fiber; Northwestern and UIC both have a common CA, but have no direct security association. Any time Northwestern and UIC want to share information they have to create a temporary security association using the common Certificate Authority (CA).

Supporting the creation, editing and display of this information in a single location seems a good operational management capability. It will be used by the server to determine the validity of information received from a particular device, to determine how to create temporary security associations if required, and to provide variables used when evaluating policy rules. It will also be use to display existing trust and

security associations. The expectation is that in the future this mapping will be enhanced in a number of ways to give good visual understanding of current and potential relationships.

The Server will support policy across multiple domains. The expectation is that each domain will have a GOAD server or will have its own section of a shared server [this may be especially useful when creating Virtual Organizations]. The intent for first implementation is that each domain will have a GOAD Resource Control Server which communicates to device controllers such as CONDOR-G, the Canarie MonFox Proxy, the UIC PIN/ PDC, or specific Control modules written for the server itself (as in the Generic AAA Server prototype implementation for SC2003).

Resource Allocation will be supported initially by a simple routine based on models developed for use by PIN at UIC. The server will be structured so that more complicated algorithms can be used to support allocation if developed. The expectation is that projects will be done that experiment with allocation algorithms appropriate to specific applications. The TerraPath proposal supports such experimentation.

The GOAD server will initially support "conversational" state, tracking the sequence of messages and responses used in a single authorization and allocation. It will be enhanced to provide stateful sessions which define an application across time. GOAD Severs will be able keep a record of what resources are being used and to match their state information with state from specific devices being used in the "session". For example a GOAD Server may request the allocation of devices from two organizations. Its session will say include the two devices and a pointer to the Resource Control Server that did the actual allocation. The Resource Control Server will also keep track of the use of the resouce (an perhaps many other resources). It will point to the state information for the device controller (or whatever is appropriate). This will create the ability to create and track sessions and session state for applications with resources in multiple domains. This will allow an application to add and delete resources during execution, will allow handling of resource change during an application, and will allow graceful termination of all resources in an application when it terminates. It will also allow coordinated usage reporting (and charging if desired).

6.2 Pilot Rollout for initial GOAD Server

The following describes a possible pilot implementation will between NetherLight, StarLight, UIC, Amsterdam, and Michigan. [Note: this is an imaginary project, but seems possible.]

In this project NetherLight and StarLight own optical switches, with fiber connecting between the switches and a number of user fiber connections to each switch. UIC, Amsterdam and Michigan create a Virtual Organization to collaborate on some project and are delegated the right to use a Lambda of the path between Amsterdam and Starlight as well as Lambdas from Michigan and UIC to StarLight and a Lambda from Amsterdam to NetherLight.

This will require a set of GOAD Servers: 2 Resource Servers, one at NetherLight and one at StarLight, and one Virtual Organization - Application Server that supports users from UIC, Amsterdam and Michigan. NetherLight and StarLight will delegate resources to the VO server. The VO will set up a path between any two of the organizations in response to application requests.

VO users from each of the organizations will authenticate to their local authentication server and get a credential to present to the GOAD Application Server. A client will be available for [Linux?] which will allow a user on that system to authenticate to a local server and then call the VO Server. The call to the VO Server will be for an application that is stored on the VO server, and asks for network resources to support the application. A demonstration application for each of the paths [Amsterdam – UIC, Amsterdam, UM, UIC- UM] will be made available as part of the pilot.

A delegation client is able to execute resource delegation by modifying policy on both Resource Server and Application Server simultaneously. The delegation client will need to be able to establish security relationship with both servers in order to be successful.having privileges to modify policy tables in both the Application Server and the Resource Server(s) in a single operation.

Figure 5 shows a conceptual layout of the pilot system. It shows the relationship between Se;rvers and between a user client and servers. It shows logical a relationship between optical switch and Resource controller, but does not show intermediate resource controllers. Note that each resource controller may interact with many Application controllers.

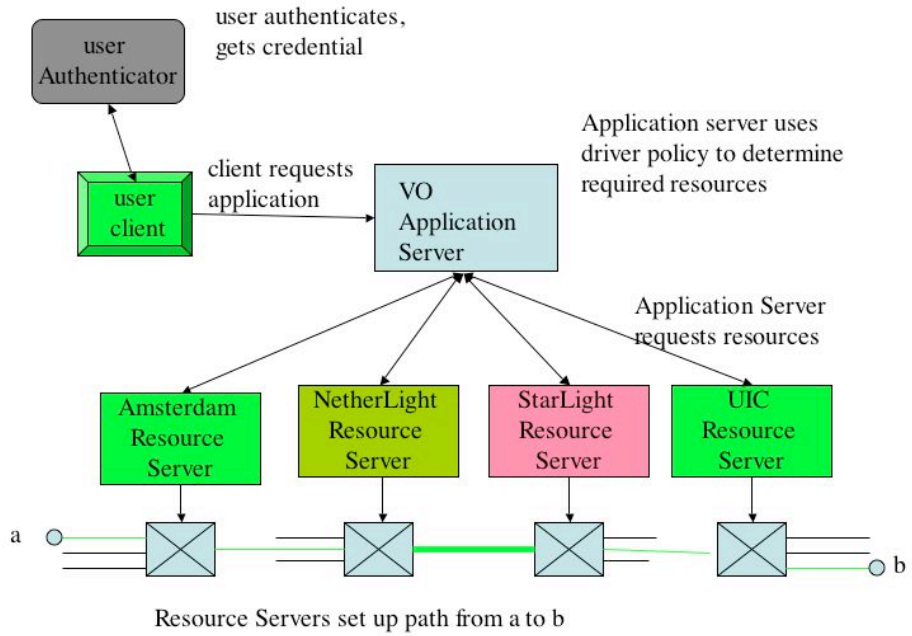

Figure 5

Figure 6 shows the relationship of a delegation Client and a pair of Application / resource Servers. The intent of the picture is to show how the delegation client can make it simple to track resources in the application server and resource policy in the Resource Server.

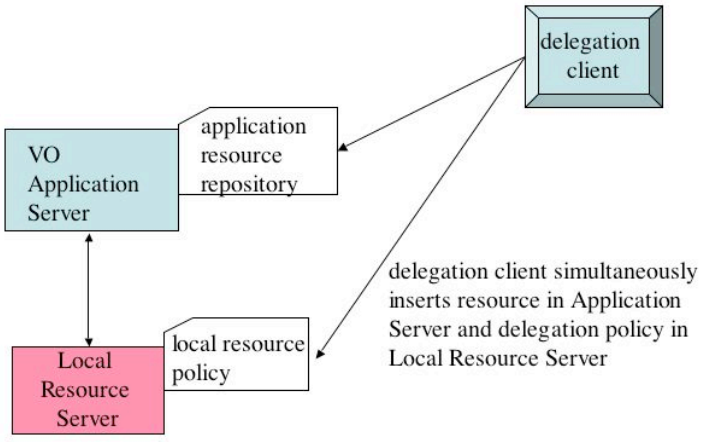

Figure 6 - delegation Client

6.3.1 Time Frames and Level of Effort for Pilot

The budget and level of effort included here for the pilot are necessarily very rough. I am not familiar with the exact level of current implementation of various projects, and have no current relationship with some of the groups that would be involved. Nonetheless, I offer the following as a strawman for further discussion, hoping that there is interest in taking up this project or variations on it.

The intent in this project is that it would include people resources at Merit/UM in Ann Arbor, in Amstedam, and at UIC. An assumption is that we can get access to Canarie Resource Code and MonFox Proxy. There is likely to be some cost for theMonFox proxy, so that may be an extra.

The pilot project will last 18 - 24 months. An initial rough estimate is that this will require 6 FTEs for that period of time. In addition there will be a need for travel between the geographic groups for development and project coordination. Some of this can be done via teleconference, but some travel will be needed. In addition some senior staff will need to travel to conferences. The estimate is that there will be a need for 12 long distance trips per year with an average cost of about $2000/ trip.

Resources required will be standard overhead for personnel (office space, personal computer, etc.) + 8 Linux computers for use in developing and deploying Software, access to fiber switches at each location, and software from groups like Canarie and MonFox.

A total cost for this is estimated as

Year 1	Year 2
6-FTE's – 600k	600k
12-trips - 24k	24k
software – 25k	25k
hardware – 25k	10k
674k	659K

6.3.1.1 Pilot Team
Ann Arbor

- 1.0 – Project Coordinator / architect
- 0.5 - Sys Admin/ programmer
- 0.5 – Web Services programmer
- 0.5 – Application developer

Amsterdam

- 1.0Server Coordinator / architect
- 0.5 – Senior Developer
- 1.0 Programmer/Documenter
- 0.5 -Application developer

UIC
0.25 - Application developer
0.25 – Device programmer/ coordinator

First application – grant +8 months

- Implement phase 1 control of fiber between UM/UIC/Amsterdam
  Demonstrate at appropriate conference/show
  - Implement initial tables of security and trust associations and use in control policy
  - Implement initial web interface for establishing temporary organizations with resources and people from UM, UIC, and Amsterdam
  - Document Implementation authorization flow.
  - Implement interface to Kerberos and PKI authentication
  - Implement initial delegation client

- Integrate Canarie/ MonFox resource controller
- Produce Application Driver Language inteface

Second milestone – grant + 14 months

- Implement stateful resource objects for GOAD servers, and use to manage appilication "lifecycle"
    - Use stateful resouce objects to provide usage and billing information
- Enhance Driver Policy to support sequencing stateful resources, including semaphores, parallel processing, and "rollback after error"
- Implement sample "network" organization to support overlay networks
    - This allows creation of an organization with delegated elements from many organiztions to provide "overlay" network resource to a number of Virtual Organizations.
- Integrate with Optiputer

Third milestone – grant +24 months

- Develop and demonstrate applications that include use of computing resources, especially computing clusters, like Condor or Rock.
- Include additional delegated resources in the Application Server, possibly from LamdaRail or Canarie.
- Incorporate Shibboleth credentials in server
- Publish descriptions of Pilot
- Feedback

7. Author's note

The work done on this paper was largely encouraged by Cees DeLaat at University of Amsterdam, Tom DeFanti at University of Illinois at Chicago and Hunt Williams at Merit Network, Inc. My thanks to them for their encouragement and interactions. I have enjoyed working on this; I have learned a lot and look forward to learning more. This is an exciting field, and a lot of people have done and are doing work that will create a new computing paradigm. I look forward to being a part of it.

Please send questions and comments to me [John Vollbrecht] at jrv@umich.edu

---

1. Newman, H., Ellisman, M., Orcutt, J. Data-Intensive E-Science Frontier Research, *Communications of the ACM 46, 11* (Nov. 2003) pp. 69-75
2. Smarr, L., Chien, A., DeFanti, T., Papadopoulos, P. The OptIPuter, *Communications of the ACM 46, 11* (Nov. 2003) pp. 59-66
3. DeFanti, T., Brown, M. DeLaat, C. (Editors) *FGCS Future Generation Computer Systems 19, 6* (August 2003)
4. Foster, I., Kellelman, C., The Grid, Morgan Kaufman Publishers, Inc, San Francisco 1999
5. The OptIPuter at Electronic Visualization Lab, University of Illinois at Chicago http://www.evl.uic.edu/cavern/optiputer/
6. The OptIPuter project site http://www.optiputer.net/
7. StarLight http://www.startap.net/starlight/
8. Leigh, J., Renambot, L., DeFanti, T., Brown, M., He, E. Krishnaprasad, N. Meerasa, J., Livingston, C., McLaughlin, M. An Experimental OptIPuter Architecture for Data-Intensive Collaborative Visualization, 3rd Workshop on Advanced Collaborative Environments (in conjunction with the High Performance Distributed Computing Conference), Seattle, WA 06/22/2003 - 06/22/2003 http://www.evl.uic.edu/papers/pap_project.php3?indi=197
9. N. Taesombut and A. Chien, "Distributed Virtual Computer (DVC): Simplifying the Development of High Performance Grid Applications," Proceedings of the Workshop on Grids and Advanced Networks (GAN '04), Chicago, Illinois, held in conjunction with the IEEE Cluster Computing and the Grid (CCGrid2004) Conference, April 2004.
10. G. M. Kent, J. Orcutt, L. Smarr, J. Leigh, A. Nayak, D. Kilb, L. Renambot, S. Venkataraman, T. DeFanti, Y. Fialko, P. Papadopoulos, G. Hidley, D. Hutches, M. Brown, "The OptIPuter: a new approach to volume visualization of large seismic datasets," Ocean Technology Conference, May 2004
11. The Biomedical Informatics Research Network (BIRN), a National Institutes of Health (NIH)-National Center for Research Resources (NCRR)
12. The Telescience Project A Collaborative Environment for Telemicroscopy and Remote Science https://telescience.ucsd.edu/
13. The Grid: The Future of High Energy Physics Computing? Lecture by Shawn McKee (University of Michigan) 1/7/2002 http://www.wlap.org/umich/phys/seminars/hep-astro/2002/mckee/
14. New Frontiers in High Energy Physics: The CERN Large Hadron Collider Lecture by Homer Neal (University of Michigan), 20 November 2003 http://wlap.physics.lsa.umich.edu/wl-browser/browser.php?ID=20031120-annarbor-01-neal
15. MGRID http://www.mgrid.umich.edu/front.shtml
16. DeLaat, C., Presentation at University of Michigan Nov. 3, 2003 http://carol.wins.uva.nl/~delaat/talks/cdl-2003-10-03.pdf
17. GRID Forum, History and Background http://www.ggf.org/L_About/hist&back.htm
18. Catlett Home Page http://www.ggf.org/people/catlett/default_b.htm
19. Foster, I, Gannon, D., Kishimoto, H. The Open Grid Services Architecture, GGF Draft, https://forge.gridforum.org/projects/ogsa-wg/document/draft-ggf-ogsa-spec/en
20. Czajkowski, K., Ferguson, D., Fopster, I., Frey J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenenepe, W., The WS-Resource Framework, http://www.globus.org/wsrf/#motivation
21. OASIS Consortium http://www.oasis-open.org/who/
22. Mambretti, J. Creating a Global Lambda GRID, Presentation at 10 GE Conference, San Diego, October 2002 http://www.sdsc.edu/10GigE/presentations/jmabretti_10GigII.pdf.
23. GLOBUS Background http://www.globus.org/about/default.asp
24. IBM Alpha Works, Grid Technology Techttp://www.alphaworks.ibm.com/grid
25. Vollbrecht, J., Calhoun, P., Farrell, S. Gommans, L. Gross, G. deBruin, B., DeLaat, C., Holdredge, M., Spence, D., AAA Authorization Framework, IETF RFC 2904 Auguset 2000 http://www.ietf.org/rfc/rfc2904.txt?number=2904
26. D.Papadimitriou, M.Fontana, G.Grammel, others, Optical Network-to-Network Interface Framework and Signaling Requirement IETF Draft May 2001 http://www.cse.ohio-state.edu/~jain/ietf/ftp/draft-papadimitriou-onni-frame-01.txt

27. Erning Ye, Intra-Carrier Solutions Enabled by the OIF NNI, OIF Presentation http://www.oiforum.com/public/downloads/Ye.ppt
28. Box,D., Curbera, F., and others, Web Services Policy Framework (WS-Policy), 28 May 2003 http://av.rds.yahoo.com/_ylt=A9ibyKDy04ZANVwAahlrCqMX;_ylu=X3oDMTBvdmM3bGlxBHBndANhdl93ZWJfcmVzdWx0BHNlYwNzcg--/SIG=11mq38om9/**http%3a//ftpna2.bea.com/pub/downloads/WS-Policy.pdf
29. Pendarakis, D. OIF NNI: The Roadmap to Non-Disruptive Control Plane Interoperability OIF Presentation http://www.oiforum.com/public/downloads/Pendarakis.ppt
30. Chen, Y., GMPLS Actively Managed WDM Testbed, Presentation, 2003 http://www.umiacs.umd.edu/partnerships/ltsdocs/Chen_presentation.pdf
31. The International Center for Advanced Internet Research (iCAIR) and iGRID2002, http://www.icair.org/igrid2002/
32. Taal, A., Sliepen, G., DeLaat, C., A grammar for Policies in a Generic AAA Environment, IETF draft March 2004 http://www.ietf.org/internet-drafts/draft-irtf-aaaarch-generic-policy-04.txt
33. van Oudenaarde, S.,Gommans, L., DeLaat, C., Dijkstra, F., Taal, A. Prototype of a Generic AAA Server, IETF Draft March 2004, http://www.ietf.org/internet-drafts/draft-irtf-aaaarch-prototype-00.txt
34. Gommans, L.,Generic AAA Based Provisioning Of Network Elements, Status update UIC-EVL 9/10/03
35. National LambdaRail Architecture http://www.nationallambdarail.org/architecture.html
36. Shirey, R., Internet Security Glossary, IETF RFC 2828, 2004, May 2000, http://www.ietf.org/rfc/rfc2828.txt?number=2828
37. Kohl, J., Neumann, C., The Kerberos Network Authentication Service (V5), IETF RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt?number=1510
38. IBM Developer Works, Web Services Security (WS-Security) Version 1.0, April 5, 2002, http://www-106.ibm.com/developerworks/webservices/library/ws-secure/
39. SOAP Version 1.2 Part 0: Primer, W3C Recommendation 24 June 2003 http://www.w3.org/TR/soap12-part0/
40. Hughes, J., Maler, E., Technical Overview of the OASIS Security Assertion Markup Languatge (SAML) V 1.1, March 30, 2004, http://www.oasis-open.org/committees/download.php/6193/sstc-saml-tech-overview-1.1-draft-04.pdf
41. Housley, R., Polk, T., Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure , John Wiley, 2001
42. Phoenix Technologies, History of Zero Knowledge Password Authentication http://www.integritysciences.com/history.html
43. Schneier, B., *Applied Cryptography Second Edition*,John Wiley & Sons, 1996.
44. Booth, D., Hugo, H., and others,Web Services Architecture, W3C Working Group Note 11 February 2004 http://www.w3.org/TR/ws-arch/
45. Cerami, E., Web Services Essentials, O'Reilly & Associates, (February 2002)
46. Charter for OASIS Web Services Resource Framework TC Oasis TC, http://www.oasis-open.org/committees/wsrf/charter.php
47. Foster, I., Frey, J., Graham, S., Tuecke, S., and others, Modeling Stateful Resources with Web Services, v 1.1 March 5, 2004, http://www.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf
48. Globus Resource Management Document http://www-unix.globus.org/developer/resource-management.html
49. MonFox Dynamic TL1 Web Page http://www.monfox.com/dtl1.html
50. Blaze, M., Feigenbaum, J., Ioannidis, J., keromytis, D., The Role of Trust Management in Distributed Systems Security. Chapter in Secure Internet Programming: Security Issues for Mobile and Distributed Objects, (Vitek and Jensen, eds.) Springer-Verlag, 1999. http://www.crypto.com/papers/trustmgt.pdf
51. Moses, T., ed. XACML profile for Web-services, OASIS working draft 03, 23 July 2003, http://www.altavista.com/web/results?q=XACML+Web+Services&kgs=1&kls=0
52. Web page of OASIS Web Services Composite Application Framework TC,
53. Web Page for Scheduling and Resource Management. GGF. https://forge.gridforum.org/projects/srm/

54. Yu, D., McKee, s., Cottrell, R., Robertazzi, T., Thomas, T., Principal Investigators, TerraPaths: A QoS Enabled Collaborative Dat Sharing Infrastructure for Peta-scale Computing Research, A DOE SciDAC and MICS Proosal for the period July1, 2004 to June 30, 2007
55. Banerjee,A., Drake, J., Lang, j., Turner, B., Kompella, K., Rekhter, Y., Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements IEEE Communications Magazine, January 2001, http://av.rds.yahoo.com/_ylt=A9ibyKdmDIhA_psAAUprCqMX;_ylu=X3oDMTBvdmM3bGlxBHBndANhdl93ZWJfcmVzdWx0BHNlYwNzcg--/SIG=11bssks1h/**http%3a//www.calient.net/files/GMPLS.pdf
56. Multiprotocol Label Switch Protocol Web Page, IETF working group http://www.ietf.org/html.charters/mpls-charter.html
57. Arnoud, W., MonFox TL1 Proxy, Presentation at ON*Vector Workshop, March 1, 2004
58. MonFox Home Web Page, http://www.monfox.com/
59. Kerberos Leveraged PKI, Web Page from CITI at University of Michigan, http://www.citi.umich.edu/projects/kerb_pki/
60. User Controlled LightPaths Definition Document, Canarie Presentation , Dec 19, 2002http://www.canarie.ca/canet4/uclp/index.html
61. Authentication Context, OASIS Security TC, Working Draft 03, 19 February 2004 http://www.oasis-open.org/committees/download.php/5556/sstc-saml-authn-context-2.0-draft-03.pdf
62. Shibboleth home page at Internet 2, http://shibboleth.internet2.edu/
63. Rocks Cluster Distribution home page, Open Source Performance Linux Cluster Solution, rockscluster.org, http://www.rocksclusters.org/Rocks/
64. Condor Home Page, University of Wisconsin, http://www.cs.wisc.edu/condor/
65. AAAArch Research Group, IRTF Group investigation AAA Architecture issues, http://www.aaaarch.org/
66. LHC Computing Grid Project, CERN Web Page, http://lcg.web.cern.ch/LCG/
67. Department of Energy, Particle Physics Data Grid Web Page, http://www.ppdg.net/related_grid_activities.htm
68. Hacker, T., Athey, B., A Methodology for Account Management in Grid Computing Environments, Proceedings of the 2nd International Workshop on Grid Computing, November 2001, http://www-personal.engin.umich.edu/~hacker/papers/A%20Protocol%20for%20Account%20Management%20in%20Grid%20Computing%20Enviro%85.pdf
69. Doster, W., Watts, M., Hyde, D., The KX509 Protocol, technical report from CITI at University of Michigan, http://www.citi.umich.edu/techreports/reports/citi-tr-01-2.pdf
70. Kornievskaia , O., Honeyman, P., Doster, W., Coffman, K., Kerberized Credential Translation: A Solution to Web Access Control. technical report from CITI at University of Michigan, http://www.citi.umich.edu/u/aglo/papers/
71. Adamson, W. Kornievskaia, O., A Practical Distributed Authorization System for GARA to appear at the Infrastructure Security Conference, Bristol, England, October 2002. http://www.citi.umich.edu/u/aglo/papers/infrasec2002.ps
72. Berger, L., Editor, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description, IETF RFC 3471, http://www.ietf.org/rfc/rfc3471.txt?number=347
73. Ashwood-Smith, P., Berger, L., Editors, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions, RFC 3472 January 2003, http://www.ietf.org/rfc/rfc3472.txt?number=3472
74. v. Oudenaarde, B., Gommans, L., GenericAuthorization Authentication Accounting, Status Report from University of Amsterdam to Data Tag, http://www.science.uva.nl/research/air/projects/aaa/index_en.html
75. Globus Web Page, Overview of the Grid Security Infrastructure , The Globus Alliance, http://www.globus.org/security/overview.html
76. Pearlman, L., Welch, V., Foster, I., Kellelman, C., Tuecke, S., A Community Authorization Service for Group Collaboration, http://www.globus.org/security/CAS/GT3/

77. Rigney C., Willats W. Calhoun P., RADIUS Extensions, IETF RFC 2869, June 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2869.txt
78. Calhoun P. Loughney J. Guttman E. Zorn G.. Arkko J., Diameter Base Protocol, IETF RFC 3588 September 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3588.txt
79. Durham, D., Ed. The COPS (Common Open Policy Service) Protocol, IETF RFC 2748 January 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2748.txt
80. de Laat ,C., Gross G., . Gommans L., Vollbrecht J. Spence D., Generic AAA Architecture, IETF RFC 2903, August 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2903.txt
81. Web Server Architecture. W3C Working Group Note 11, Feb 2004, http://www.w3.org/TR/ws-arch/
82. Aboba B. Beadles M., The Network Access Identifier, IETF RFC 2486 , January 1999 , ftp://ftp.rfc-editor.org/in-notes/rfc2486.txt
83. Czajlpwslo. K., Ferguson, D., Foster, I., Frey, J., Maguire, T., Snelling, D., Tuecke, S., From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring and Evolution, Globus Report, http://www.ibm.com/developerworks/library/ws-resource/ogsi_to_wsrf_1.0.pdf