GGF DOCUMENT SUBMISSION CHECKLIST (include as front page of submission)	
	COMPLETED (X) - Date
1. Author name(s), institution(s), and contact information	2005.06.01
2. Date (original and, where applicable, latest revision date)	2006.02.10
3. Title, table of contents, clearly numbered sections	2006.01.05
4. Security Considerations section	2006.01.05
5. GGF Copyright statement inserted (See below)	2005.06.01
6. GGF Intellectual Property statement inserted. (See below) **NOTE that authors should read the statement.**	2005.06.01
7. Document format - The GGF document format to be used for both GWD's and GFD's is available in MSWord, RTF, and PDF formats. (note that font type is not part of the requirement, however authors should avoid font sizes smaller than 10pt).	2005.06.01

Draft

Work in progress document

This page intentionally left blank to align document for 2-sided printing

Ralph Niederberger, Forschungszentrum Jülich GmbH (Editor)
William Allcock, Argonne National Laboratory
Leon Gommans,University of Amsterdam
Egon Grünter, Forschungszentrum Jülich GmbH
Thijs Metsch, German Aerospace Centre – DLR e.V.
Inder Monga, Nortel Networks
Gian-Luca Volpato, RRZN University of Hannover
January,2006

Firewall Issues overview.

Status of This Memo

This document is a working draft, ultimately to be submitted to the Global Grid Forum for consideration as informational document.

The latest version of this document can be found at:

https://forge.gridforum.org/projects/fi-rg

Abstract

To provide an overview, the document will describe several kinds of devices used to provide some level of protection against malicious attacks from somewhere in the public Internet. Then the document describes a number of grid related cases that experience issues with firewall type of devices. The document will use these experiences to classify and describe a number of issues.

Table of Contents

ABSTRACT 3

1 INTRODUCTION 7

2 CONVENTIONS USED IN THIS SPECIFICATION 8

3 DEFINITIONS 9

3.1 Firewall 9
3.1.1 Classification of firewalls 9

3.2 Firewall (global definition) 10

3.3 Network Address translators 11

3.4 Application level gateways 11

3.5 VPN gateways 12

4 GRID APPLICATIONS AND THEIR ISSUES WITH FIREWALLS 13

4.1 Grid and Application Technology Deployments 13
4.1.1 The Issue with "Net of Trust" or the "bastion hosts" solution 13
Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany 13
4.1.2 Impact of DCache deployment 15
Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany 15
4.1.3 Issues in enabling General Parallel File System, GPFS 17
Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany 17
4.1.4 The workflow management system TENT 18
Organization: German Aerospace Centre, Cologne, Germany 18

4.2 Grid Network Architectures and Protocols 20
4.2.1 GridFTP versus the Firewall 20
Organization: Argonne National Laboratory, ANL, US 20
4.2.2 UNICORE - The Seamless GRID Solution 21
Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany 21
4.2.3 Webservices Firewall Issues 25
Organization: Argonne National Laboratory, ANL, US 25
4.2.4 Firewalls and high bandwidth, long distance networks 27
Organization: University of Amsterdam, Amsterdam, The Netherlands 27

5 CLASSIFICATION OF FIREWALL ISSUES 28

6 SUMMARY 30

7 SECURITY CONSIDERATIONS 31

8 ACKNOWLEDGEMENTS 31

9 AUTHOR INFORMATION 31

10 GLOSSARY 32

11 APPENDIX 1: CLASSIFICATION OF FIREWALL ISSUES SEEN FROM THE USE CASES SIDE 33

12 INTELLECTUAL PROPERTY STATEMENT 42

13 FULL COPYRIGHT NOTICE 42

14 NORMATIVE REFERENCES 42

15 INFORMATIONAL REFERENCES 42

1 Introduction

Grid-Projects with external partners often lead to communication relationships between external and internal computer systems requiring special configurations at firewall systems. These configurations include:

- allowing access for communication sessions (ports)
- allowing access to single systems or sub networks in general

Additionally physical access may be provided implementing physical or logical links as fiber, wavelength, sub wavelength, VPN, VLAN, etc. assuming that this links can not be used by external sources. Often these links will not be secured by firewalls.

The configurations shown above result in:

- administrative overhead
- wildcard access rights (port not known, so give access to whole system)
- weaker policies or no security policies anymore
- general decreasing security level to that of the partner installation
- security vulnerability because of open ports for long time periods.

Because of the limitations of today's firewalls (limited to 1 Gb/s throughput often, some already allow 10 Gb/s) load balancing of multiple firewalls is done based on IP or MAC-address balancing often, i.e one stream will be executed by one firewall giving real balancing only with multiple communication streams. Grid applications with huge bandwidth demands (one data stream) do not have any advantage of these firewalls.

Some firewall clusters allow round-robin mechanisms, but are limited to lower speeds because of the extreme overhead needed for status information updates between the different firewall components

Only a small amount of firewall systems is able to handle applications with dynamically assigned ports. Some implementations are known for applications like ftp, h.323, and sip. But no general solution is available.

Often within a grid environment every installation has its own firewall system. All of them have to be traversed by grid applications. Because of the problems discussed in the introduction project networks are placed in a demilitarized zone in most of the cases. This implies that every computer system used in the project has to be secured. Bad or wrong configured systems lead to security vulnerabilities. Supercomputers or special systems may be connected via dedicated networks assuming a "Net of Trust", i.e. users at these systems will be trusted leading to insider security problem. Compromise of these systems leads to increased security problems. Many national and international activities/projects try to cope with these problems. Some of them are:

- D-Grid, a German project funded by BMBF, Germany [D-Grid]
- EGEE, a European Project funded by EU, [EGEE]
- MIDCOM [MIDCOM]
- OPSEC (CheckPoint), [OPSEC]
- University of Buffalo, Grid Computing Research projects, ACDC-Grid Firewall - „Advanced Computational Data Center Dynamic Firewall (ACDC Dyna-Fire) Development", http://www.ccr.buffalo.edu/grid/content/research [ACDC]

The examples above show that there are new demands to firewalls today.

This document tries to identify scenarios used today in grid environments. It structures these scenarios into use cases and classifies these cases into general communication concepts used

by grid applications. These classifications will provide a fundament for further investigation into possible solutions which will be discussed within a later FI-RG document.

The solutions senns so far can be divided into three categories:

a) A solution to the use case can be provided without any modification or additional software or hardware development (e.g. give access to a special port)

b) A soloution to the use case can be provided developing new software / hardware components, which allow handling of those special use cases or classes of use cases (e.g. as been done by the ftp protocol by checking the control communication stream and opening ports negotiated between the communication partners via the control connection.

c) A solution seems not feasible with the current kind of firewalls. New software / hardware models have to be developed.

The current document tries to pave the way to these classifications and wants to identify which of the current grid applications fall into which category and how to overcome use cases which are categorized into classes b) and c) above.

2 Conventions used in this Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

3 Definitions

The goal of this chapter is to provide an overview of the types of devices and software components that are used to protect grid applications and infrastructures from malicious attacks from the Internet.

3.1 Firewall

A **firewall** is a logical object (hardware and/or software) within a network environment which prevents communications forbidden by the security policy of an organization taking place, analogous to the function of firewalls in building construction. Often a firewall is also called a **packet filter**.
The basic task of a firewall is the control of traffic between different zones of trust and/or administrative authorities. Typical zones of trust include the Internet (a zone with no trust) and an internal network (a zone with high trust). The ultimate goal is to provide controlled connectivity between zones of differing trust levels through the enforcement of a security policy and connectivity model based on the least privilege principle.
Proper configuration of firewalls demands skill from the administrator. It requires considerable understanding of network protocols and of computer security. Small mistakes can lead to a firewall configuration worthless as a security tool and in extreme situations fake security where no security has been left.

3.1.1 Classification of firewalls

There are three basic criteria to categorize firewalls:

1. whether the communication occurs between a single host and a network, or between two or more networks;
2. whether the communication is intercepted at the network layer or at the application layer;
3. whether the communication status is tracked at the firewall or not.

With regard to the scope of filtered communication there are

- personal firewalls, i.e. software applications which normally filter traffic entering or leaving a single computer through the Internet;
- network firewalls, normally running on a dedicated network device or computer positioned on the boundary of two or more networks or DMZs (demilitarized zones). Such a firewall filters all traffic entering or leaving the connected networks.

The latter definition corresponds to the conventional, traditional meaning of "firewall" in networking. Additional firewalls may be located between administrative domains of an organization (e.g. between production, research, administration and finance departments).

In reference to the software layers where the traffic is intercepted, three main types of firewalls exist:

- network layer firewalls
- application layer firewalls
- application firewalls

The network-layer and application-layer types of firewalls may overlap, even though the personal firewall does not serve a network. Indeed there are examples of single systems that have implemented them both together.
Application firewalls are sometimes used in wide area network (WAN) networking on the world-wide-web and govern the system software. An extended description would place them at a lower level than application-layer firewalls, actually at the operating system layer, and they could alternatively be called operating system firewalls.

Lastly, depending on whether the firewalls track communication status, two categories of firewalls exist:

- stateful firewalls
- stateless firewalls

3.1.1.1 Network layer firewalls

Network layer firewalls operate at a (relatively low) level of the TCP/IP protocol stack as IP-packet filters, not allowing packets to pass through the firewall unless they match the filtering rules.
The firewall administrator defines the rules or default built-in rules may apply (as in some inflexible firewall systems). A more permissive setup could allow any packet to pass the filter as long as it does not match one or more "negative-rules", or "deny rules".
Today network firewalls are built into most computer operating system and network appliances.

3.1.1.2 Application-layer firewalls

Application-layer firewalls work at the application level of the TCP/IP stack and intercept all packets traveling to or from an application (HTTP traffic, telnet traffic, ftp traffic, etc.). They block unauthorized packets, usually dropping them without acknowledgement to the sender. In principle, application-layer firewalls can stop all unwanted incoming traffic from reaching protected machines.
By inspecting all packets for improper content, these firewalls can even prevent the spread of viruses. In practice, however, this becomes so complex and so difficult to attempt (given the variety of applications and the diversity of content each may allow in its packet traffic) that comprehensive firewall design does not generally attempt this approach.
The XML Firewall exemplifies a more recent kind of application-layer firewall.

3.1.1.3 Application firewalls

The term application firewalls is often used to describe security tools that control access to services that run on an operating system. They are composed of software components running on a system and securing this local system by checking which external (remote) hosts may access the special services running on this node. Often these firewalls are called operating system firewalls.
A well-known implementation of application firewalls is TCP wrapper.

3.1.1.4 Stateful/stateless firewalls

Modern network-layer firewalls can filter traffic based on many packet attributes like source IP, source port, destination IP or port, destination service like WWW or FTP. They can filter based on protocols, TTL values, netblock of originator, domain name of the source, and many other attributes.
Having the ability to look into the packets into more detail, allows monitoring the status of the transmission (based on TCP options or simulated status for stateless protocols) and implementing more complex filtering rules. A stateful firewall usually allows incoming TCP packets only when they belong to a connection started by a host in the protected network. Connection requests coming from untrusted networks are rejected.
In contrary to this behavoir a stateless firewall does not monitor the status of connections. Every packet has to be checked and mapped to a rule that either allows or denies it.

3.2 Firewall (global definition)

In a broader sense a firewall is the implementation of a security policy of an institution concerning traffic exchange between different security domains. It is not only a black box or single hardware. It can be much more. It is all the rules you specify, to become safe. It is the way you check the

compliance with these rule and it is the whole bunch of software and hardware you use to implement this.

3.3 Network Address translators

Network Address Translation (NAT) [RFC 1631] provides a way to map IP addresses from one IP network to another IP network allowing transparent routing between client and server hosts in distinct networks. Most times NAT will be used to connect private networks using private address space to the external INTERNET with officially registered addresses. Using this technique it is possible to solve the current problem of rare official IP addresses by reusing these for different hosts until new IP addresses (IPv6) are available respectively commonly used. The address reuse is normally done at the borders of private domains. This allows using this technique for security reason additionally. The main advantage of NAT is that it can be installed without changes to routers or hosts. Unfortunately the NAT function cannot support all applications since sometimes IP addresses are used in the packet payload itself. Therefore NAT must co-exist with application level gateways (ALGs) often.

Allowing transparent routing NAT devices have to modify host addresses in the packets on the fly and must maintain state information of communication flows. Packets belonging to the same communication stream have to be translated in the same manner i.e. to the same IP address.

Port Address Translation (PAT) or Network Address Port Translation (NAPT) [RFC 2663] enhances this technique one step further. Here different hosts may be translated to the same IP address using port information (source and destination port) to differentiate between different communication streams (e.g., TCP and UDP port numbers, ICMP query identifiers). Many internal private IP addresses can be translated to one official external IP address.

Often NAT and PAT are used as a security mechanism. Internal hosts are allowed to setup communication paths to external hosts, but connections from external hosts to internal hosts can be setup only if a translation is active currently (That means an internal host has already setup a connection). Here NAT is done dynamically making it harder for an attacker to point to any specific host in the NAT domain. NAT routers may be used in conjunction with firewalls to filter unwanted traffic. Often the firewall itself does the NAT.

Problems arise with end-to-end IPsec, because there cannot be a NAT device in the path. IPsec uses the source and destination address of the end-to end-communication. If NAT changes one of these addresses the IPsec communication will fail. A solution will be to use the NAT devices as tunnel end point of the IPsec connection.

“NAT devices, when combined with ALGs, can ensure that the datagrams injected into Internet have no private addresses in headers or payload. Applications that do not meet these requirements may be dropped using firewall filters. For this reason, it is not uncommon to find NAT, ALG and firewall functions co-exist to provide security at the borders of a private network. NAT gateways can be used as tunnel end points to provide secure VPN transport of packet data across an external network domain. (RFC 2663)”

3.4 Application level gateways

Not all applications lend themselves easily to translation by NAT devices; especially those that include IP addresses and TCP/UDP ports in the payload. Application Level Gateways (ALGs) are application specific translation agents that allow an application on a host in one address realm to connect to its counterpart running on a host in different realm transparently. An ALG may interact with NAT to set up state, use NAT state information, modify application specific payload and perform whatever else is necessary to get the application running across disparate address realms.

ALGs may not always utilize NAT state information. They may glean application payload and simply notify NAT to add additional state information in some cases. ALGs are similar to Proxies, in that, both ALGs and proxies facilitate application specific communication between clients and servers. Proxies use a special protocol to communicate with proxy clients and relay client data to servers and vice versa. Unlike Proxies, ALGs do not use a special protocol to communicate with application clients and do not require changes to application clients.

3.5 VPN gateways

A Virtual Private Network (VPN) gateway into a Corporate Network can be considered as the "employee entrance", whereas a firewall could be considered as a "public entrance". A corporate network is typically classified as a private network, created to support the business of an Enterprise, SMB, or any other organization with a need to protect its networked resources from public access. A VPN gateway uses credentials that are issued by the Corporate Network Administrator to create a security association between the Corporate VPN gateway and a remote VPN site. Remote sites can either be individual PC clients or other VPN gateways. A remote VPN gateway allows the Corporate Network to be securely extended into a branch office via an un-secure network. This setup is called a site-to-site VPN. A PC VPN Client allows individual employes to access the Corporate Network from the Internet when at home or traveling. Protocols, such as IPSec, L2TP, PP2P and SOCKS ensure authenticated and encrypted communication between VPN sites by creating a tunnel. On such connections, packets are constructed in a specific VPN protocol format and are encapsulated within some other base or carrier protocol, then transmitted between VPN client and server, and finally de-encapsulated on the receiving side. The base protocol for Internet is IP. Other cases, that typically use point-to-point connections, may use a layer 2 protocol.

Most VPN gateways offer similar functionalities as firewalls. Packet filtering and packet inspection are examples. It is therefore important to consider these types of devices in within the realm of this document.

4 Grid applications and their issues with firewalls

This chapter contains input from various organizations describing their issues with firewalls. This input may describe problems and suggested solutions to problems. When contributions were asked, no structure was suggested as to keep the input as broad as possible. The information within this section will be analyzed and classified in subsequent sections.

4.1 Grid and Application Technology Deployments

4.1.1 The Issue with "Net of Trust" or the "bastion hosts" solution

Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany

The Research Center Jülich has been involved in many networking projects over the last 10 years. Always these projects included research on new network technologies as well as its impact on applications. As a consequence firewall considerations have been of main interest always. As a conclusion we realized that constantly growing bandwidth demands on networks require a reconsidering of techniques. Using new generation networking techniques in Wide Area Networks implies the communication of hosts of different administrative security domains. Because of the high speed networks it will not be possible to inspect every packet. Firewalls cannot be faster as normal network interfaces as they use these interfaces, so there will be a time delay in implementing faster firewalls always. Because firewalls have to forward many communication streams in parallel providing access for many different host to host communications, this scenario increases the needed throughput bandwidth enormously. So traditional firewalls can not be used in futuristic scenarios. How to handle security issues in the future?

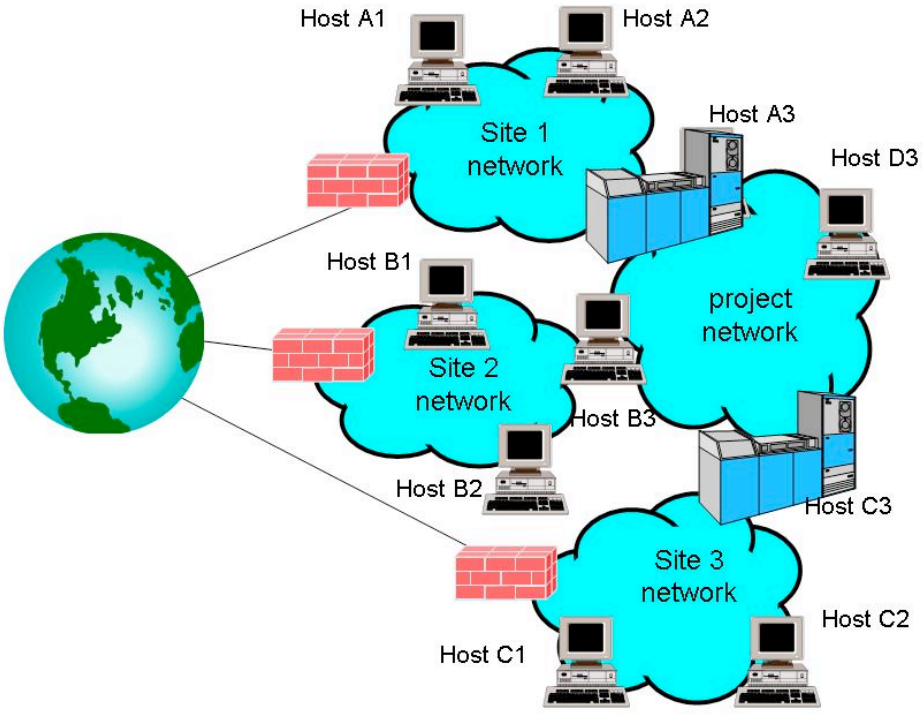

Figure 1: Securing project networks – "Net of Trust"

A generally used approach is to have a “Net of Trust”, which implies every node within the project network is assumed secure (or conforms to the security policies of all organizations).

This can be achieved by having every node secured by an organization firewall (site local firewall) prohibiting unauthorized access from remote sites and assuming that only authorized persons can access the project network directly. Hosts A3, B3, C3 and D3 are connected to their institution network and additionally to the project network. The project network cannot be access directly from outside (see figure1 above).

Alternatively every node within the project network may have been installed with highest security considerations (personal firewall, iptables, virus scanners, only really needed services installed and activated with minimum privileges,…).

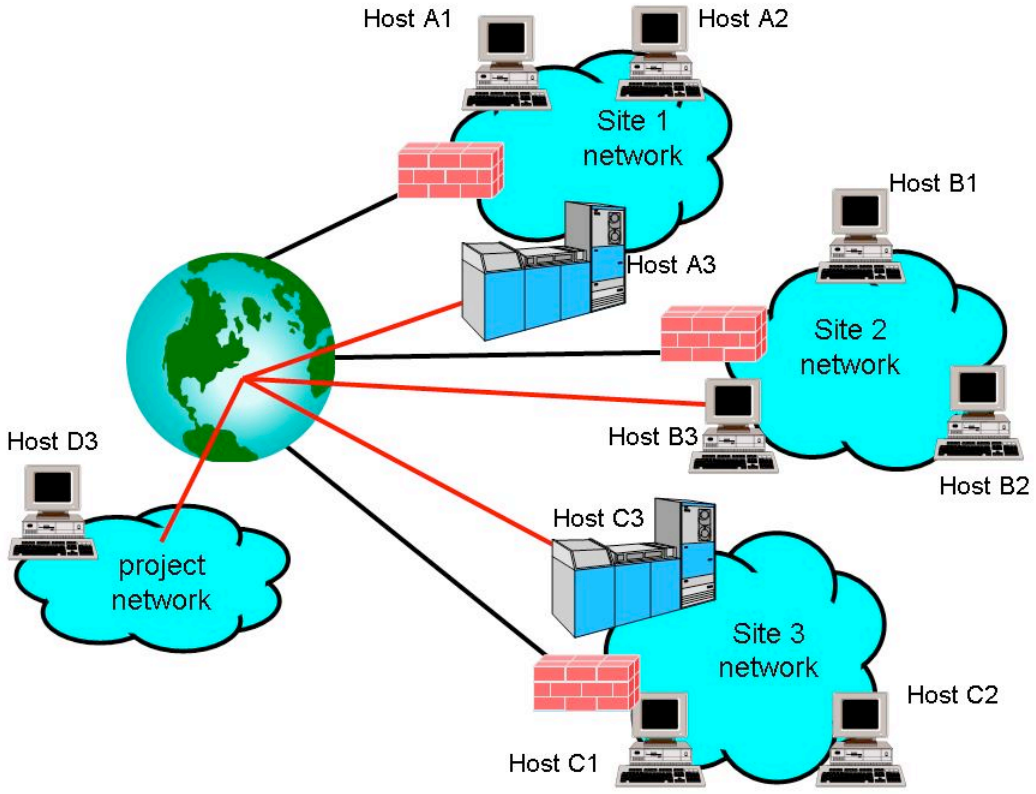

Figure 2: Securing project networks – “Bastion hosts”

These hosts are normally called bastion hosts, because they are located in an insecure environment and have been secured as a bastion against its enemies. Figure 2 shows the bastion host scenario, where hosts A3, B3, C3 and D3 are connected to their institution network as well as to the publicly accessible project network. All hosts within the project network have to be secured accordingly. Though these scenarios are a nightmare for firewall administrators and security officers they are often used because of missing alternatives. New ideas have to be developed in the future.

4.1.2 Impact of DCache deployment

Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany

dCache is a joint venture between the Deutsches Elektronen-Synchrotron, DESY and the Fermi National Accelerator Laboratory, FERMI.

dCache has been selected to be used in the German D-Grid project started in 2005. dCache allows to store and retrieve huge amounts of data, distributed among a number of heterogeneous server systems. These systems simulate a single virtual file system. Depending on the Persistency Model, dCache provides methods for exchanging data with backend (tertiary) Storage Systems as well as space management, pool attraction, dataset replication, hot spot determination and recovery from disk or node failures. Connected to a tertiary storage system, the cache simulates unlimited direct access storage space. Data exchanges to and from the underlying hierarchical storage manager, HSM, are performed automatically and invisibly to the user. File system namespace operations may be performed through a standard nfs interface allowing all regular file system operations except accessing the data directly. In addition to standard data access methods like Ftp, GFtp and Http, a native access protocol dCap may be used allowing POSIX file system operations. dCache has full control of the location and multiplicity of datasets. Non precious files are removed if space is running short. File replicas are generated if a certain pool becomes overloaded. Replicas are slowly removed if the situation improves. Pools are chosen for file transfers, either from clients or from the backend HSM, based on dynamic space and load parameters of the individual pools. In addition to the dynamic behavior, pools can be assigned to data according to the IP address of the client, the ordering mechanism of the backend HSM or special tags which can be given to subdirectory trees of the file space. [dCache-1]

Because dCache can be used in a local environment as well as over a Wide Area Network, firewall issues have to be considered as well.

If components of the dCache systems are installed distributed across multiples sites, some of these components have to be accessed from outside, which implies that firewalls have to be traversed. Protocols used within dCache are dCap, GSIdCap, GridFTP and SRM, storage resource manager. dCap should be used for local, trusted access only and therefore is not of any relevance for firewall considerations. GSIdCap extends the dCap protocol by using a GSI authentication wrapper (tunnel). Communicating with the GSIdCap servers (doors) requires opening ports into a firewall.

The GridFTP protocol has been described in a previous chapter above already.

The SRM protocol uses https as transport protocol and negotiates data transfers between the client and server as well as between different servers. One of the other protocols is used to transfer actual data.

A common solution to overcome the problem of dynamic client and server connections over not known ports in advance is to open a range of ports within the firewall. On the user's perspective this allows undisturbed usage of dCache services. From the firewall manager's perspective this implies a security hole within the security policies of the site. [dCache-2]

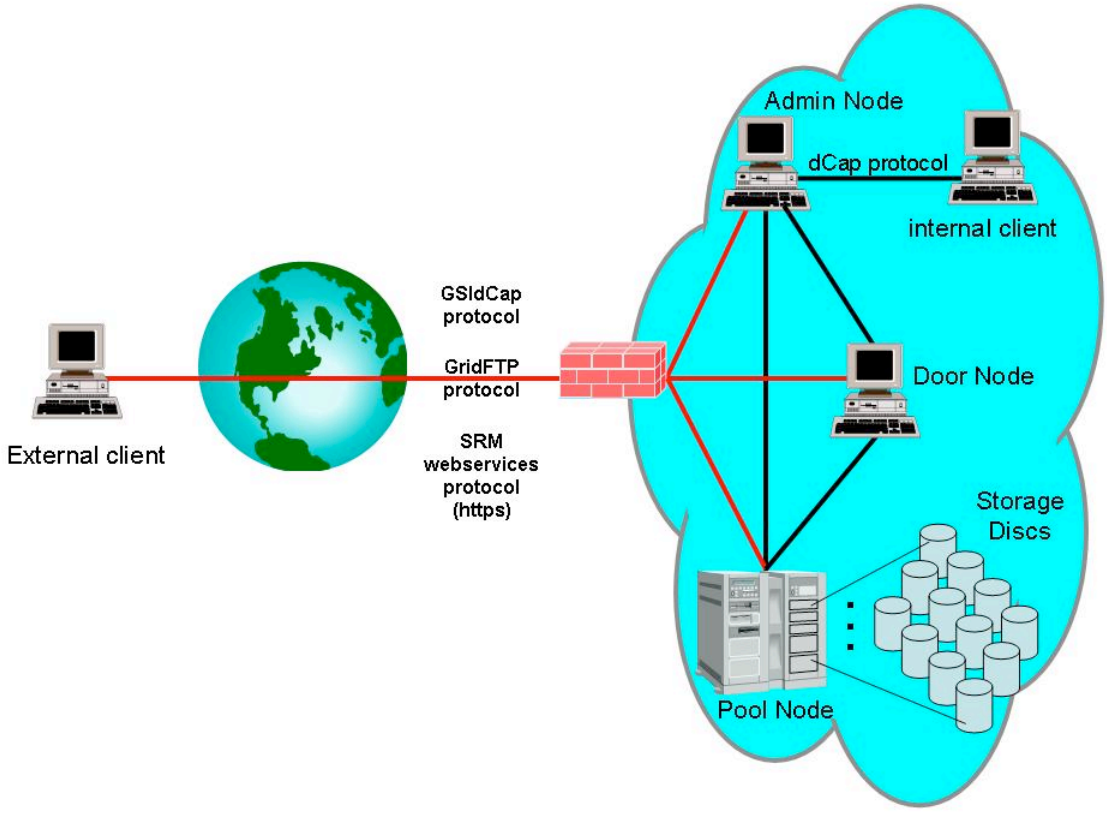

Figure 3: dCache, an overview, Nicolo Fioretti, Bari , Nov. 2005, http://www.dcache.org/manuals/dcache.nicolo.overview.small.jpg

4.1.3 Issues in enabling General Parallel File System, GPFS

Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany

The General Parallel File System, GPFS, developed by IBM, is a high-performance shared-disk file system. It provides fast, reliable data access from all nodes in a homogenous or heterogeneous cluster running an AIX or LINUX operating system.

GPFS allows parallel applications to simultaneously access one file or a set of files from any node that has the GPFS file system mounted while providing a high level of control over all file system operations.

GPFS has been designed to deliver much higher performance, scalability and failure recovery by accessing multiple file system nodes in parallel. Nevertheless it complies with normal UNIX file system standards.

GPFS provides high-performance I/O by "striping" blocks of data from individual files across multiple disks (on multiple storage devices) and reading/writing these blocks in parallel. In addition, GPFS can read or write large blocks of data in a single I/O operation, thereby minimizing overhead.

For optimal performance and reliability the data can flow between the storage and application node via multiple paths. GPFS availability is further improved by automatic logging and replication. Additionally GPFS can be configured to failover automatically in the event of a disk or server malfunction.

GPFS scalability and performance are designed to meet the needs of data-intensive applications such as engineering design, digital media, data mining, financial analysis, seismic data processing and scientific research. [GPFS-1]

The general communication scheme used by GPFS is a client server model. The GPFS deamon (mmfsd process) communicates between nodes in different clusters. The communication paths are established via TCP socket call. GPFS uses IANA assigned port 1191 by default, and is changeable via the *mmchconfig* command if required. So from a firewall perspective GPFS uses only one port when using GPFS-MC. This can be configured without any problems in standard firewalls. Because systems using GPFS are known in advance, a static access list can be configured. In future a problem could arise when GPFS would become publicly available and commonly used. In case of this the protocol itself would have to be analyzed and secured, so that no backdoors or vulnerabilities will open up wholes within normally strong protected network areas.

What makes GPFS interesting as a special firewall use case is the communication throughput of the protocol. Because of parallel streams transferring a file between systems a high bandwidth will be needed. Communication throughput is only dependend on the number of clients and I/O servers used within the GPFS installation. Data rates of 3 GB/s have already been examined. This implies high speed firewalls not available today or very good load balancing of a firewall cluster.

4.1.4 The workflow management system TENT

Organization: German Aerospace Centre, Cologne, Germany

This contribution describes the use case of the workflow management system TENT.

The workflow management system TENT (see figure 4 for a screenshot) has been developed at the German Aerospace Center over the last years. It allows engineers to easily setup and maintain workflows. Workflows are applications coupled together to form a process chain. Applications can be computational fluid dynamic (CFD) codes or graphical editors for visualization. Components can be numerical or functional units within a work flow, e. g. computational fluid dynamic (CFD) codes, graphical editors for visualization, or pre-/post-processors.

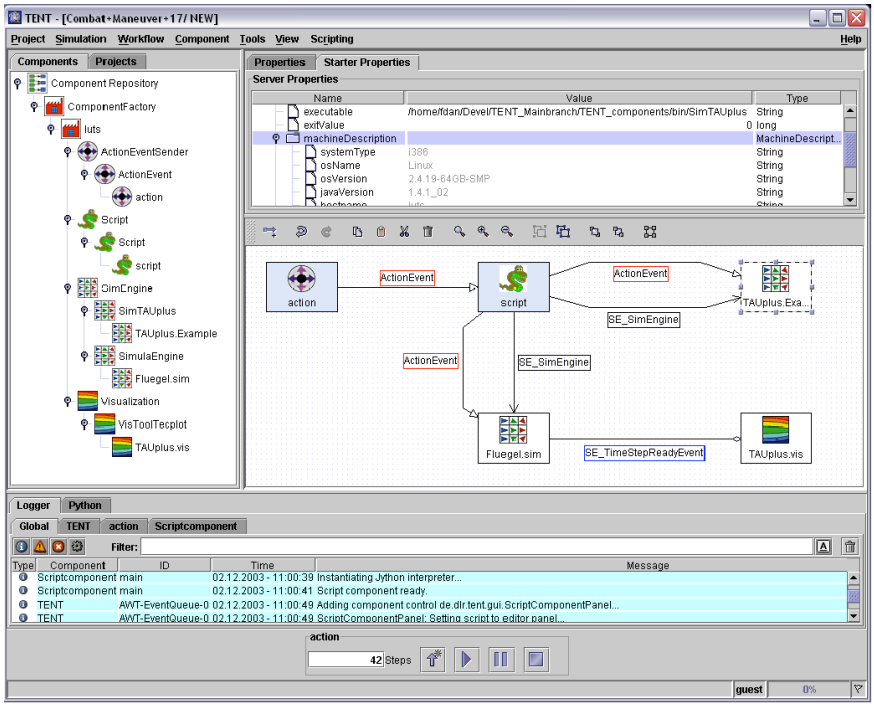

Figure 4: TENT GUI

This process chain can be used to solve fluid dynamics, structural mechanics, and thermodynamic computations. Components are the smallest elements in a workflow and can be localized on distributed resources. Coupling of resources can be achieved by means of Grid computing. Computational resources can be placed on different remote hosts. By creating Virtual Organizations (VOs) it is possible to use all these resources as one. The following functionalities are necessary to use TENT within Grids:

- Access to all resources of a Grid. Authentication mechanisms need to be provided.
- Data transfers between the resources of a Grid must be possible. Data transfers can either be Reliable File Transfers (RFT) or status messages (MPI based messaging) passed along.
- Execution of CFD codes on Grid resources. Job Managers and their queues should be accessible to the TENT system.

For all these communications TENT uses service based communication.

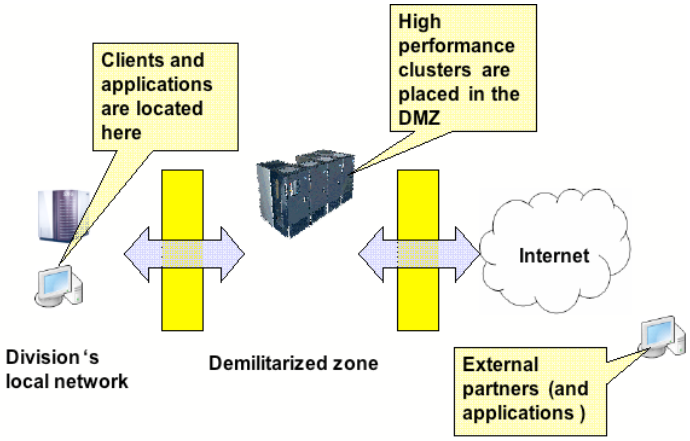

Figure 5: Closer look at possible firewall borders

The creation of VOs becomes obligatory when applications (and its matching licenses) and resources are located on either sides of a firewall. The creation of VOs can extend beyond the borders of companies. Therefore the location of the Grid resources is no longer bound to geographical positions. Figure 5: Closer look at possible firewall borders" gives a closer picture of the borders of a company. Firewalls form the borders of the local site. But applications of resources like high performance cluster are not located within the local (and easily accessible) network.

Due to the fact that most companies and organisations use firewalls, following problems arise:

- Several firewalls have to be passed (internally and externally). The administrators of these firewalls are not always directly available.
- Firewalls have to be opened for several TCP and UDP ports. Some port ranges are unknown during set-up. They will be initialized by the Grid middleware itself. So port ranges have to de defined.
- Data transfers have to be allowed beyond the borders of a local site. This includes the transmission of data packets and status information.
- VPNs have to be initialized at the borders of a site. To increase security the traffic connections should be secured against wire tapping.

Security policies disallow the opening of firewalls in almost every case. Strict control of the incoming and outgoing traffic becomes a major issue. A lot of politics have to be dealt with when establishing connections beyond company's borders.

4.2 Grid Network Architectures and Protocols

4.2.1 GridFTP versus the Firewall

Organization: Argonne National Laboratory, ANL, US

GridFTP is a fairly troublesome application from the point of view of firewalls. It can use a significant number of ports that are in the ephemeral range and with today's protocols it is not possible to either know in advance the full 5 tuple that describes a connection, nor to limit the usage to two ports.

GridFTP, like FTP, has two channels, a control channel and a data channel. The control channel is relatively painless. It is always a single socket connection to a well known port. The connection is strongly authenticated, it is encrypted, integrity protected, and very low bandwidth, so this is something that firewall administrators are generally willing to deal with, and because it is low bandwidth, the firewall generally does not introduce any performance limitations.

Then there is the data channel which is very difficult especially for GridFTP. Why is it so difficult? There are several reasons. First, the data channel is a logical construct and can consist of an arbitrary number of sockets, which can vary in time. The protocol allows sockets to be added or removed arbitrarily anytime during a transfer. Second, the protocol requires that the sending sides perform the TCP connect so you do not have the option of having the client be passive to work around firewall restrictions. Third, the full 5-tuple for a given connection is known very late in the process and nothing has global knowledge of the logical connection between individual sockets that make up the logical data channel connection.

Some background on how GridFTP works will help explain this:

We will describe a third party transfer (a transfer between two servers mediated by the client). It is the most complex of the transfers and client/server transfers simply do only one half of the PASV/PORT command, since the client knows the other half internally when it is involved in the actual movement of data. We will describe a striped transfer, which involves m hosts on one end sending to n hosts on the other end, m and n are not required to be the same, and can be one, a non-striped transfer is for the purposes of this discussion, a striped transfer with m and n equal to one.

The client attempts to open a control channel connection on a well known port. Assuming this port is open on the firewall and it can establish a connection, it begins sending a series of commands that do authentication, and then begin to describe the transfer, like is it binary or ASCII, etc. If this server is the receiving server, it will sent the SPAS (striped passive) command. Each host will then listen on an arbitrary ephemeral port, and that list of listening ports is sent back to the client in the response to the SPAS command. At this point that server knows it will be contacted, but it does not know by whom.

The client now attempts to open a control channel to the sending server again on a well known port. It authenticates and begins its command sequence to the server, but this time it will send the SPOR (striped PORT) command. This command includes the list of listening ports that was returned in the response to the SPAS command. This tells the server that it *MAY* connect to this list of servers. It may connect to one, all, or some subset depending on the layout of the data. It does not yet know how many connections to make. That is determined when the OPTS RETR (retrieve options) command is sent. This indicates the minimum number of streams, the start number of streams, and the maximum number of streams. Note that it is the server who can decide to change the number of streams, within the limits specified, the client can not tell the server to add or remove streams, this means that there is no command sequence that can be trapped on the control channel to know when a new connection is being initiated. Once the RETR <filename> command is received, each host on the server side will determine which hosts in the SPOR list it needs to connect to and will initiate the connection, which will again be an arbitrary ephemeral port. It is only in the socket call when the connecting ephemeral port is chosen that the full 5-tuple is known.

We have been asked why we can't have a single data port. The problem is that you can only have one process using a port. The way the control channel works is that some daemon (typically inetd) is listening on the well known port. It gets a *single* connection, does a fork/exec, duplicates the socket, hands it off to the new process and then closes it's file descriptor. It is now ready to accept another connection on that port from anywhere other than a host and port that already has a connection to its port 2811.

However, let's assume that we wanted to have 2812 be the data channel port. The process listening on that port would need to be able to accept a connection, know which transfer that connection is associated with, and how many total connections were expected (all connections would have to be formed up front, this would not allow for additional connections later, a limitation of what the protocol allows, though probably not a big one). Once it had all the connections for a given transfer, it could then fork/exec a data node (GridFTP backend) dup all the necessary sockets to it, then it closes its socket, and that backend could go merrily upon its way. The problem is that there is no way, today, to know what transfer a connection is associated with and no way for that listener to know how many connections it should get.

4.2.2 UNICORE - The Seamless GRID Solution

Organisation: Forschungszentrum Jülich GmbH, Jülich, Germany

The UNICORE software (UNiform Interface to COmputing REsources) is a user-friendly software interface which allows easy and uniform access to distributed computing resources, and which provides support for running important scientific and engineering applications in a Grid environment. Scientists can use different supercomputers as well as other computing and storage resources without having to become experts in the special kind of access software and security policies of the various (super-)computer centers.

UNICORE provides a science and engineering Grid combining resources of supercomputer centers. It makes these resources available through the Internet. UNICORE uses a strong authentication and authorisation scheme in a consistent and transparent manner. Differences between platforms are hidden from the user. A seamless HPC portal for accessing supercomputers, compiling and running applications, and transferring input/output data has been developed.

Through using UNICORE end-users can concentrate onto their real application issues and therefore increase their productivity. Internal supercomputer specifics are hidden to these end-users which don't need to learn any kind of job control languages. So a more efficient job can be done.

The UNICORE user prepares or modifies structured jobs through a graphical user client interface on his local workstation or PC. Besides the UNICORE internal job description UNICORE also is able to handle XML-Jobs. After preparation the created job has to be submitted to one of the platforms of a UNICORE Grid. Here the user may monitor and control the submitted jobs through a second area in the UNICORE client.

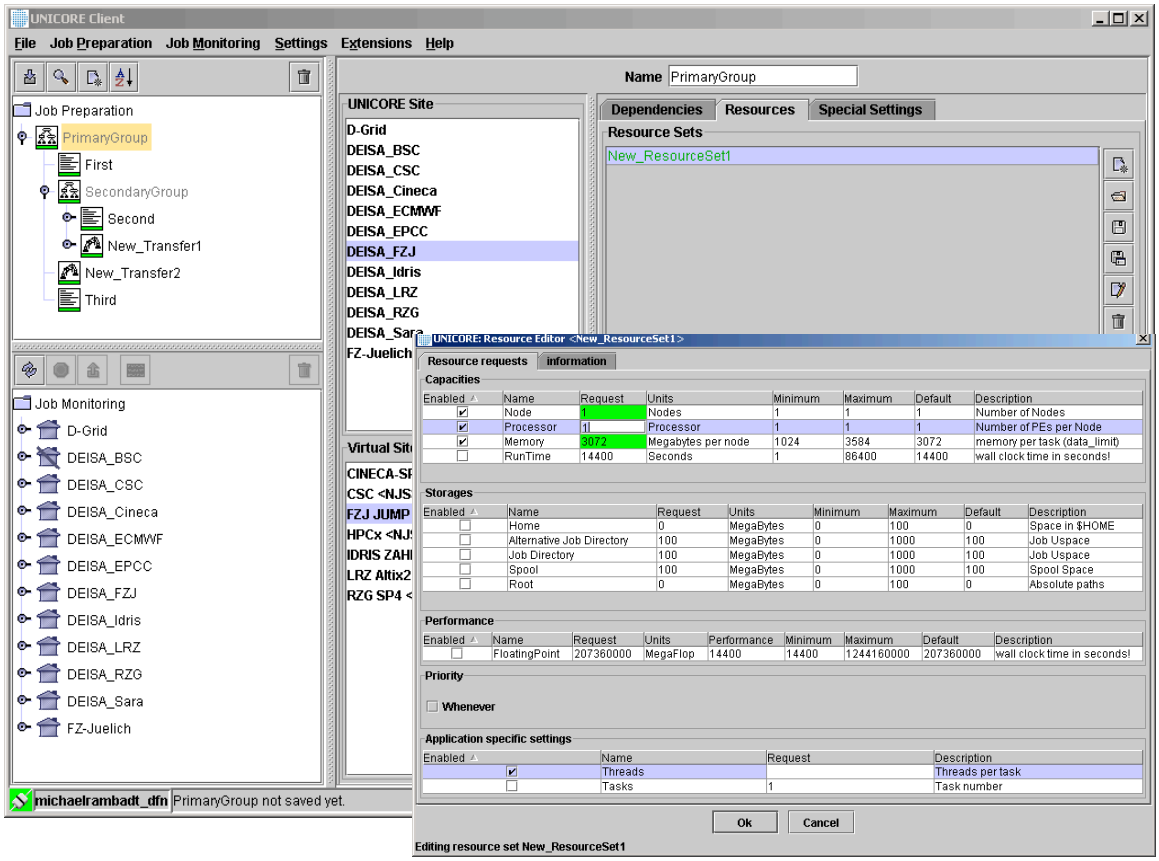

Figure 6: Job preparation: Definition of a job, adding dependencies and resource requests.

UNICORE allows to structure jobs, dividing them into independent tasks. Dependencies between these tasks can be assigned. The structured model allows executing a job, divided into subtasks to be run on different locations of the UNICORE Grid leading to hierarchical job structures and data locations. So UNICORE is able to manage complex multi-site and multi-step workflows efficiently.

UNICORE has a three tier architecture which consists of the Client, the Gateway, and the NJS /TSI. The NJS (Network Job Supervisor is responsible to map the abstract job description to concrete target system issues. This is done with the IDB (Incarnation Database). Also it authorises the user to access the target system. The NJS is the front-end to the target system. The TSI (Target System Interface) is a library of Perl modules being installed on the target system (e.g. the supercomputer) itself and providing an interface between the batch system and the UNICORE servers. While all other UNICORE components but the TSI are implemented in Java the UNICORE Client and servers are very platform independent.

UNICORE tasks and resources are represented in abstract terms and units, so that a server can translate them into the platform-specific commands and options. Input and output files are automatically imported / exported from / to the user's file space or transferred from earlier tasks of the same job. Explicit transfer tasks handle the high-speed data transfer between different sites. The UNICORE servers select the most efficient mechanism for each transfer.

For each job, the user specifies the intended target system and the task's resource requirements. The client software checks whether the resources requests by the end-user can be satisfied by the target system, and submits the job into the target system. To resubmit a job at a different system, the user simply changes the target system.

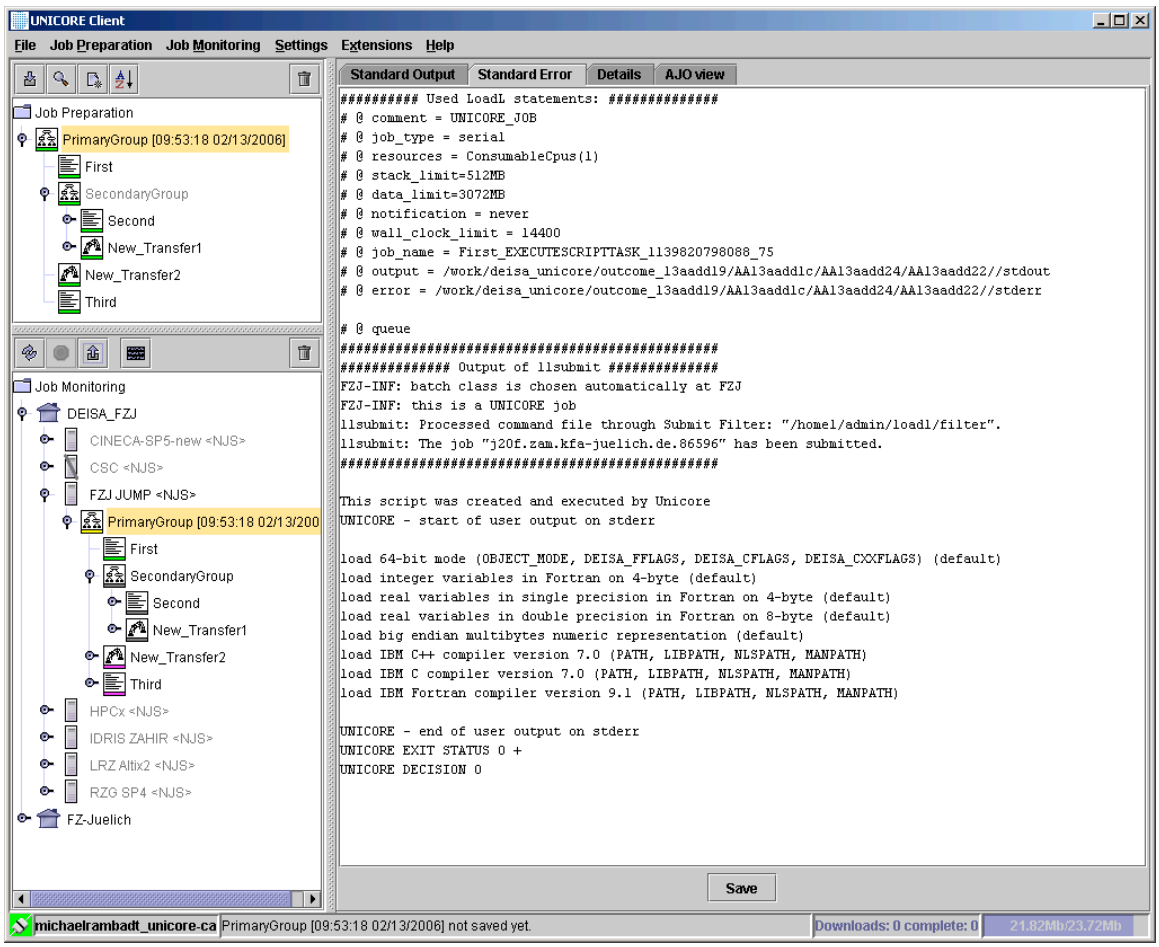

Figure 7: Job monitoring: inspect the status of running jobs and retrieve the output

In any case the users can monitor and control their jobs through the job monitor interface, which depicts the job status in a graphically manner. After job execution the output data of the job can be retrieved to the local workstation

User authentication is performed using X.509 certificates. A public-key infrastructure has been established for the German HPC centers that enforces rigorous control of certificates. Each UNICORE user has a personal user certificate signed by a trusted CA. The administrator himself is responsible to define the “Trusted CAs” in the UNICORE servers. Each job the user sends into the UNICORE Grid is signed by the private key of the certificate. User authorization is handled by the participating sites using their proven mechanisms. In this case UNICORE also completely retains the sites autonomy with authorizing users and to allocating resources to them. The UNICORE interface for the user authorisation is called UUDB (UNICORE User Database). This component maps the user’s public key of his personal certificate to the real Xlogin on the target system. So every time a job arrives in the UNICORE Grid the certificate is checked and compared with the entry in the UUDB. Only if both are identical the job will be transferred to the target system. To transfer jobs, control information and application data, SSL is used to guarantee data integrity and confidentiality. The signing of job representations with the originating user's private key prevents also third parties from tampering with the job contents.

The UNICORE gateway component authenticates connection requests by checking if the incoming certificate has been signed by a trusted CA. Also the Gateway checks if the presented user’s certificate has not been revoked and is still valid. The gateway can cooperate with firewalls

to permit only legitimate UNICORE traffic. It may reside outside the protected zone, in a demilitarized zone, or within the protected zone depending on the site's security setup. Using UNICORE only one port for the gateway has to be opened in the firewall.

While the Client-Gateway connection is necessarily SSL-secured the connection between Gateway and NJS is SSL-secured optional. While generally said the UNICORE NJS is in the safe intranet nevertheless it might be necessary or wished by the site's administrators to secure the Gateway-NJS connection via SSL, too. This is also one of the aspects why UNICORE does not influence the sites autonomy. While both the Gateway and the NJS component are signed with a server certificate the SSL handshake can be established between those components, too.

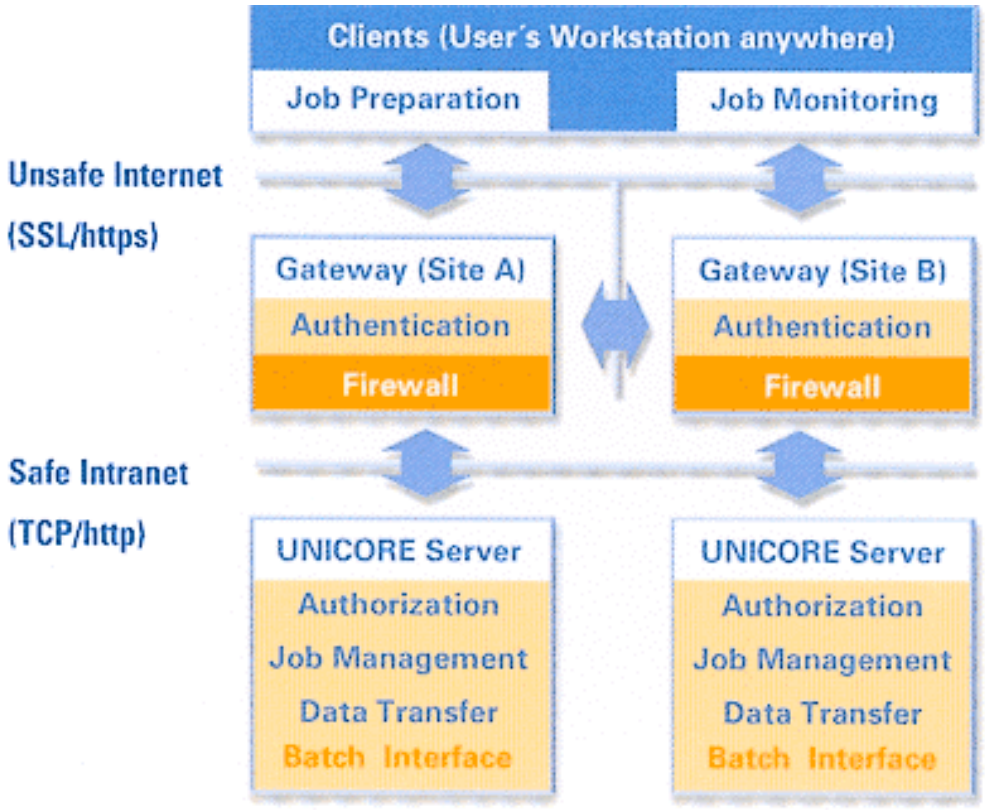

Figure 8: UNICORE architecture: system components and their interaction

The UNICORE client enables the user to create, submit and control jobs from any workstation or PC on the Internet. Required is only an installed UNICORE client. All user certificates are stored in the UNICORE keystore. So the user might just export this keystore to e.g. a memory stick and import it on the other machine again and he is able to access all his jobs and resources again.

The client connects to a UNICORE gateway which authenticates both users and other UNICORE servers, before contacting the UNICORE NJS, which in turn manages the submitted UNICORE jobs. They incarnate abstract tasks destined for local hosts into batch jobs and run them on the native batch subsystem. Tasks to be run at a remote site are transferred to a peer UNICORE gateway. All necessary data transfers and synchronizations are performed by the servers. They also retain status information and job output, passing it to the client upon user request.

The protocol between the components is defined in terms of Java objects. A low-level layer called the UNICORE Protocol Layer (UPL) handles authentication, SSL communication and transfer of

data as inlined byte-streams and a high-level layer (the Abstract Job Object or AJO class library) contains the classes to define UNICORE jobs, tasks and resource requests.

Third-party components can be integrated into the system: on top of UPL to create alternatives to the AJO layer, or within the AJO layer defining new classes. Thus, the functionality of clients and servers can be extended within the UNICORE framework by implementing so called Plugins. Plugins are also Java objects which allow integrating different applications into the UNICORE Grid software easily.

4.2.3 Webservices Firewall Issues

Organization: Argonne National Laboratory, ANL, US

This section enumerates some of the known issues concerning the webservices protocols and firewalls.

As the webservice protocol will most probably be used for the control channels and control-planes that manage GridFTP endpoints and/or dynamic firewall configurations, it is important to understand the issues and the associated requirements.

4.2.3.1 Internal vs External EPRs

The application service's EPR (End Point Reference) has an address that is used as a network endpoint for that service by the clients. As a result, when a service is located behind a firewall, then external clients outside the corporate firewall cannot use the same EPR that is used by the internal clients. If the access by external clients is allowed through an application-level firewall-proxy, then the external clients will have to be supplied with an external-EPR for the application service that will direct the client to send the messages to the SOAP-Proxy service, who, after policy enforcement, will forward the request to the application service behind the firewall.

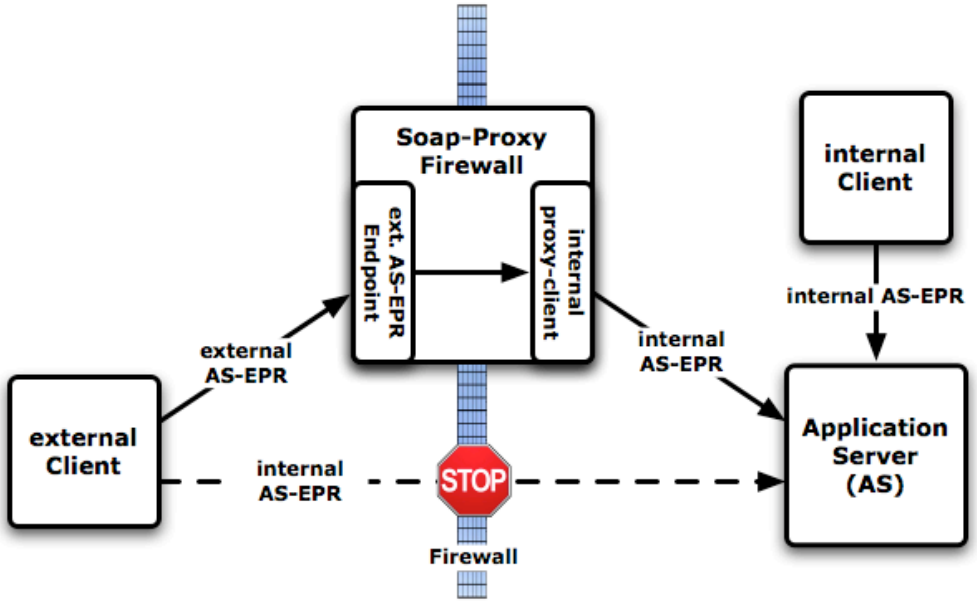

Figure 9: External Clients and Internal EPRs

The issue is depicted in Figure 9, where the external client's use of the application service's internal-EPR is blocked by the firewall, while the external-EPR is shown to route the external client's messages through the proxy service to the application service.

We have no standardized ways yet to:

- Augment the EPR with routing information
- Obtain an external EPR from an internal one
- Publish and discover the need for external EPRs
- Express policies to tell clients to extend the security context end-to-end

4.2.3.2 Ephemeral Internal EPRs

Even if we have a way to tell the external client that a soap-proxy service should be used to connect to the internal application service, we have an additional issue with factory-like patterns.

In a factory-pattern, a service is used to obtain a new EPR for a newly created or located resource. In other words, an EPR for that new service is returned in the message exchange with the factory service.

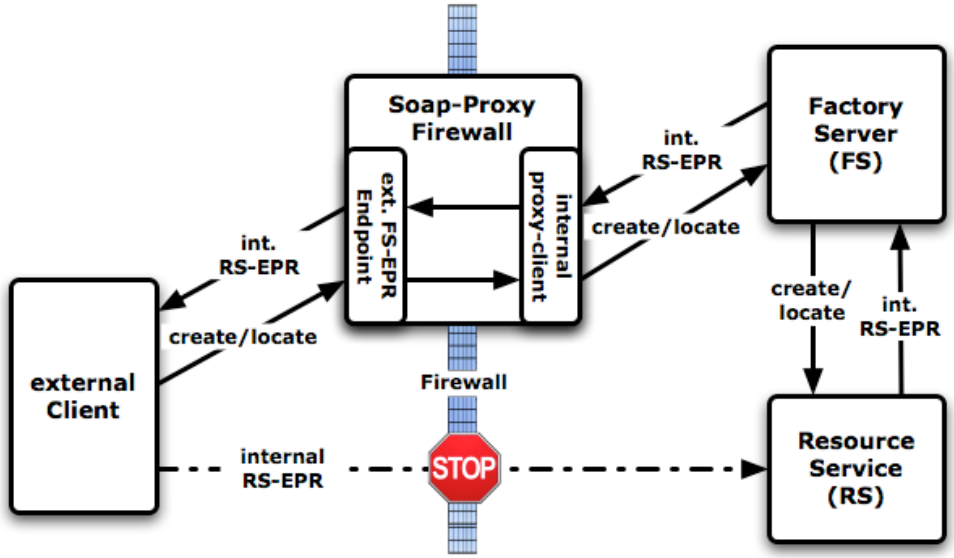

Figure 10: External Clients and Ephemeral EPRs

As shown in Figure 10, the issue with the returned EPRs is that by default they will be internal EPRs, and they should somehow be translated to external EPRs before the external client is able to use them.

The issues with ephemeral EPRs are:

- We have no standardized way for the firewall to discover which internal EPRs should be translated on the fly into external ones (feels like HTML rewriting for reverse-webproxies…)
- We have no way to express and enforce a policy that allows firewalls to deal with ephemeral EPRs, which may refer to a resource that is not in the same hosting

environment as the factory, or not even on the same host, and may not even use the same identity.

4.2.4 Firewalls and high bandwidth, long distance networks

Organization: University of Amsterdam, Amsterdam, The Netherlands

Grid applications often use high-bandwidth connections between grid locations over long distances. These applications will benefit from congestion-free connections. A modified TCP protocol behavior, which increases the rate of transmissions more rapidly after a congestion event, is needed to efficiently use such a connection. Such behavior makes these TCP streams unsuitable to share bandwidth with regular TCP streams, as their behavior is considered to be unfair. These TCP streams therefore typically by-pass the regular Internet using dedicated, mostly optical-, connections between grid locations. Research within the GHPN-RG is performed to create on-demand version of these connections, using switched optical network technologies. The GHPN group does not consider the involved network security architectures.

This chapter does consider the requirements towards a possible security architecture that could be used to connect a grid node both to the Internet and a long distance by-pass network.

The figure below shows a possible network layout involving firewalls. All Grid resources are located behind a typical two firewall-setup with a DMZ. Firewalls A and D have an additional connection that connects to the high bandwidth connection. Involving Grid middleware, a grid application may schedule a connection via the Multi-domain control and management plane. A Grid VO may be involved in the decision to provision the connection.

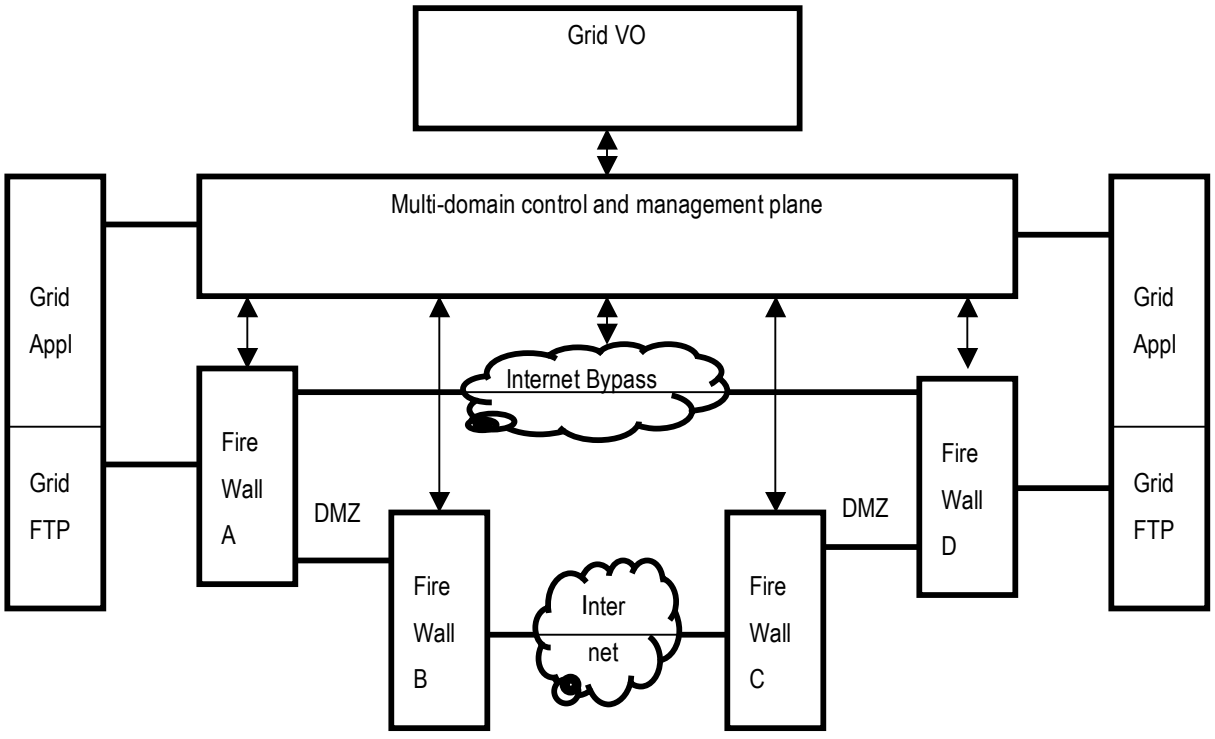

Figure 11: Possible network layout with a typical two firewall-setup with a DMZ

Considering an above network, the following requirements and issues can be defined towards such a network architecture for a Grid location:

1. The grid location must be protected against malicious attacks from the Internet. This is a general requirement for any node connected to the Internet.

2. The Internet must be protected from abusive use from a Grid cluster. As Grid locations are capable of generating vast amounts of IP traffic, it has the potential to attract malicious users. If access control fails, then a firewall should stop any attempt to misdirect IP traffic.

3. Memory shortage in any forwarding device will cause packet-loss. At least the performance of firewalls A and D must allow wire-speed operation without any congestion. Both firewalls must have enough memory to support the bandwidth delay product for each by-pass connection. Note that long distance connections inherently have large bandwidth-delay products.

4. Firewall architectures must be capable to splitting the high-bandwidth inter-grid location traffic from the regular traffic at a very early stage, e.g. at firewalls A and D. This will avoid high network loads at shared network resources downstream.

5. Adding nodes to a grid cluster should scale with the firewall infrastructure, such that congestion for Grid TCP streams, and unfairness to regular Internet resources, is avoided.

6. Usage of on-demand network resources between two Grid locations may need authorization. Firewalls A and D may act as an ingress/egress point of such a connection. Firewalls A and D could therefore act as an access enforcement point. Firewall A or D could act as a first point of contact to which applications could send requests to open up a chain of firewalls.

7. Firewalls A and D should enforce that only private (non-routable) addresses can be used via the by-pass connection. Routable addresses should be forwarded via the DMZ to firewalls B and C. Some address management for the private address spaces must be performed.

8. Both the Grid VO (or users authorized by the Grid VO) and network security administrator should have a stake in the control of a firewall.

5 Classification of Firewall Issues

After having selected different kinds of grid applications in chapter 4 and having shown their issues with firewalls located somewhere within the communication stream, we summarized in appendix A which particular problem every of these applications gets into because of traversing these firewalls.

Within this chapter we try to classify the kind of applications firewalls have to deal with.

The different kinds of grid applications we have examined can be sorted in the following manner:

- First there are applications which use special single well known ports which could be opened within a firewall. Depending in the specific application this should be not problem for firewall administrators. Nevertheless each of these applications has to be examined in detail, if the communication behaviour complies to the security policy of the organizational entity. E.g. ssh should be no problem, because users will be authenticated and authorized to login by the local ssh server. The same would be true for e.g. telnet. But telnet uses an unencrypted authentication scheme, sending userid and passwords in cleartext over the communication path. If a telnet communication has been recorded by an unauthorized person (hacker), the “man in the middle” has gained access to secure information and may afterwards get access to resources which shouldn’t be publicly available. So every grid application, though using

only single and fixed ports, has to be checked if it complies to the organizational security policies.

Another approach using these kind of single port streams is the tunneling of applications via e.g. port 80. These implementations have been developed to circumvent institutional security policies, because most organizations allow the use of the http protocol.

Though this would allow any kind of firewall traversal, it has been shown that using this approach as a general concept would allow also hackers and especially viruses to use thies method to overcome firewall barriers. Firewalll developers have taken application tunneling into account and try to fight against these methods. They implement software which is familiar with special kinds of protocols e.g http. A trivial port 80 tunneling can be recognized and stopped. Though this is no problem at all, because the tunneling application can be programmed to behave like an http protocol stream, it helps recognizing most of the trivial attacks tunneled from the outside. Taking into account these considerations application programmer should not use this kind of tunneling techniques.

- Secondly there exist applications which use control streams to signalize the communication behavior, e.g. exchange of dynamic port information for file transfers (data stream of an ftp session). These control streams can be analyzed via a firewall control program which allows to dynamically open special random ports allocated to the data stream. Thus grid applications are made firewall aware. This concept has been developed already with the ftp and h.323 protocol, where additionally to the control connection data streams have been issues. Up to date firewalls have implemented features to scan e.g. ftp streams.
- Third we have seen applications which use a control stream for exchanging of control information. But nevertheless not all information may be synchronized via this control channel. We examined GridFTP where port information is generated dynamically in a later stage of the communication protocol and in fact after the control channel has been closed. So this kind of application has to use a port range allowing to assign dynamic ports. This introduces a real nightmare for firewall administrators, problems we have to deal with.
- Fourth there are applications outside which use any kind of ports set up on when communication starts or within the already started communication session. These may only use one dynamic port or may be a combination of multiple parallel streams. These kind of application cannot be supported by a firewall administration team, because of number and times of firewall reconfiguration needs. An automatic configuration environment will be needed here.
- Fifth we have heard of applications which need high throughput data pipes. Often these cannot be interfered with normal traffic on common links, because the high speed communications have special service level agreements, SLAs. These SLAs could e.g. priorities packets leading to reduced communication throughput for normal organizational traffic. In some cases normal traffic could drop to zero, because of excessive use of those grid applications. As a result it should be feasible to bypass the *normal* institutional firewall. Therefore high throughput traffic has to be secured in a different manner.

As a summary we can classify the scenarios above, which arise problems for current firewalls, into different classes which may be structure into software, hardware, network and security policy issues:

Software:

- Port numbers and amount of ports are unknown until the application starts
- Consequence: big holes (many ports) are required if amount and/or port numbers are unknown, single hole case (e.g. HTTP port 80) causes referral problems.
- Only specific, predetermined applications that use a low number

- only very well defined ports (well known ports) can be supported adequately.

Hardware:

- unknown number and kind of firewalls are located within the routing path
- High performance data streams across long connections need enough buffer space and switching capacity
- Firewalls which are able to deal with multiple wavelengths on a single fiber not developed until now.
- If these wavelengths have been divided into individual fibers by DWDM equipment, firewalls are not able to deal with 16, 32 or 64 links of 10 Gb/s each currently

Network:

- Grid hardware resources running certain applications can not be place inside the DMZ.
- Sometimes applications must past more than 2 DMZs.
- Putting Grid applications inside the DMZ may not be avoidable sometimes.
- Firewalls, when involved in bypass connections must perform elaborate routing functions,

Security Policy:

- Firewalls may not be aware how many different applications may use the same port.
- Firewalls may not be aware of the amount of ports that are actually required v.s. configured.
- Firewalls may need to open up to 10.000 ports for certain applications
- Firewalls may not have enough information to authorize complex grid applications.
- Firewalls must not only protect from evil from the public network, but also prevent the public network from being abused.
- Firewalls may not be able to extend the security context between two applications.
- Firewalls may not be aware if a hosts connecting is actually trusted.

6 Summary

Within this document we have tried to give an overview about currently used grid applications. Of course we could not include any application used, but we tried to identify those which are examples for all the applications commonly used. We tried to classify these applications concerning their communication behavior, so that we got a good feeling which problems arise because of the existence of firewalls within the communication paths. Firewalls try to secure and control the traffic which is going in and out of an organization and there is no doubt that they are needed at all. On the other hand free research and information exchange between organizational entities is required also. Application programmers did not deal with firewalls in the past. Often applications have been developed in a local scenario without interfering with firewalls. After successful implementation they have been thought to be used in a more global environment,

often between different organizational entities. Then applications and firewalls ha d to interact with each other.

This document intends to give application programmers a feeling how to develop firewall aware applications and tries to pave the way for firewall developers to construct new kinds of firewalls, which can deal with new types of applications and network infrastructures.

Constantly growing bandwidth demands on networks require a reconsidering of techniques. It will not be possible to inspect every packet. Firewalls cannot be faster as normal network interfaces as they use these interfaces, so there will always be a time delay in implementing faster firewalls. New ideas have to be developed. Instead of inspecting single packets streams could be checked. This is already done within current firewalls with the port concept. Many connections will be allowed without checking the content of the connection. The connection will be allowed because of the fact that an instance, the destination system, checked the authorization.

Strategic objectives will be to define a standardized authorization mechanism accepted and implemented by firewall vendors into their systems so that grid enabled firewalls will become reality.

The GGF Firewall Issues- Research Group (FI-RG) intends to create another document which is a follow-up of this one and which will show a way out of the problems identified so far.

7 Security Considerations

This entire document is about security considerations.

It describes applications used across firewalls, tries to identify security risks and structures these risks into use cases. The document is intended to provide an overview of scenarios which have not yet been included into current firewall systems and tries to identify solutions for future developments.

8 Acknowledgements

To be done

9 Author Information

Ralph Niederberger (Editor)
Forschungszentrum Juelich GmbH
r.niederberger@fz-juelich.de

William, E. Allcock
Argonne National Laboratory
allcock@mcs.anl.gov

Leon Gommans
University of Amsterdam
lgommans@science.uva.nl

Egon Gruenter
Forschungszentrum Juelich GmbH
e.gruenter@fz-juelich.de

Thijs Metsch
Deutsches Zentrum für Luft- und Raumfahrt - DLR e.V.
thijs.metsch@dlr.de

Inder Monga
Nortel Networks
imonga@nortel.com

Gian Luca Volpata
RRZN University of Hannover
volpato@rrzn.uni-hannover.de

10 Glossary

To be done.

11 Appendix 1: Classification of firewall issues seen from the use cases side

<table>
<tr><td>Name</td><td colspan="5">The “Net of Trust Model”</td></tr>
<tr><td>Description</td><td colspan="5">Hosts within a project network spanning different organizational entities are secured via institutional firewalls. Between the project hosts no firewall is used. Every host and the users of these hosts are though as trustworthy.

Advantage and problem solved: Because of private networks, firewalls do not introduce a throughput bottleneck (10 Gb/s and more connections may be used)</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>low</td><td>low</td><td>high</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>management</td><td>management</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">No elements within communication path.</td><td>Own Software</td><td>Any kind of software can be used. Commercial, free software as well as experimental software.</td><td rowspan="4">No hardware restrictions. Because of free communication paths every kind of hardware using any kind of protocols (also non IP) may be used.</td><td rowspan="4">The network connecting the hosts is a private one. Could be IP or lower protocols.</td><td rowspan="4">The security policy on both sides has to agree with this net of trust concept. Hacking of one project host leads to security impacts on all connected institutional local networks.</td></tr>
<tr><td>Ports used</td><td>Because of no restriction, every port/port range may be used.</td></tr>
<tr><td>Protocol used</td><td>All kinds of protocols beneath TCP and UDP possible</td></tr>
<tr><td colspan="2"></td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">The “Bastion Host Model”</td></tr>
<tr><td>Description</td><td colspan="5">Hosts within a project network spanning different organizational entities are secured only by own security mechanisms (personal firewalls). The project hosts are freely accessible from the outside world. The project network security concept is based on the security of each individual host (bastion host).</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>low</td><td>low</td><td>high</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>management</td><td>management</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">No elements within communication path.</td><td>Own Software</td><td>Any software can be used assumed this software packet is secure and does not introduce any vulnerability.</td><td rowspan="4">No hardware restrictions. Because of free communication paths every kind of hardware using any kind of protocols (also non IP) may be used.

Prerequisite: Host can be configured secure (whatever this means).</td><td rowspan="4">The network connecting the hosts is an official one. Could be IP or lower protocols.</td><td rowspan="4">The bastion hosts are placed outside the institution networks. This implies that these networks are not affected. Nevertheless connections from the bastion hosts into the institution network are required normally. These communications have to be inspected and secured. Hacking of a project host does not directly lead to security impacts on the other project hosts. Every host is a standalone bastion.</td></tr>
<tr><td>Ports used</td><td>Because of no restriction, every port/port range may be used.</td></tr>
<tr><td>Protocol used</td><td>All kinds of protocols beneath TCP and UDP possible</td></tr>
<tr><td colspan="2"></td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">dCache</td></tr>
<tr><td>Description</td><td colspan="5">dCache allows to store and retrieve huge amounts of data, distributed among a number of heterogeneous server systems. These systems simulate a single virtual file system.</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>low</td><td>middle</td><td>high</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>management</td><td>management</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">Any kind of firewalls between the communicating entities.</td><td>Own Software</td><td>No. Software developed at DESA and FERMI.</td><td rowspan="4">No hardware restrictions.</td><td rowspan="4">Different kinds of configuration allowed. Some components must/may be placed within a DMZ, some of them must/may be placed internally into the site network.</td><td rowspan="4">Since most of the protocols use dynamic ports within a specified range, there have severe security impacts. If the protocols used haven’t been configured securely, backdoors may be introduced.</td></tr>
<tr><td>Ports used</td><td>Incoming: dCap TCP 22125
GSIdCap TCP 22128

GridFTP TCP 2811

And 20000-25000

SRM TCP 8443

Location Manager TCP 11111

Outgoing: any

All ports are configurable</td></tr>
<tr><td>Protocol used</td><td>TCP</td></tr>
<tr><td colspan="2"></td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">GPFS</td></tr>
<tr><td>Description</td><td colspan="5">The General Parallel File System is a high-performance shared-disk file system. It provides fast, reliable data from all nodes in a homogenous or heterogeneous cluster running an AIX or LINUX operating system.

GPFS allows parallel applications simultaneous access to one file or a set of files from any node that has the GPFS file system mounted using parallel streams for a single file transfer.</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>low</td><td>low</td><td>middle</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>NA</td><td>management</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">Any kind of firewalls between the communicating entities.</td><td>Own Software</td><td>No. Software developed by IBM.</td><td rowspan="4">No hardware restrictions.</td><td rowspan="4">Communication is done via normal communication paths.

(Site network –

provider network –

site network).</td><td rowspan="4">Protocol uses fixed configurable TCP port. Disadvantage: Communication including data is unencrypted.</td></tr>
<tr><td>Ports used</td><td>GPFS TCP 1191

Port is configurable</td></tr>
<tr><td>Protocol used</td><td>TCP</td></tr>
<tr><td colspan="2"></td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">The workflow management system TENT</td></tr>
<tr><td>Description</td><td colspan="5">This use case describes the firewall issues arise while integrating grid middleware software into the workflow management system TENT. The creation of a VO forms the major problem.</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>low</td><td>middle</td><td>high</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>management</td><td>management</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">Several packet filters located at the network borders of the participating organizations.</td><td>Own Software</td><td>Yes (TENT)
No (Globus Toolkit)</td><td rowspan="4">The hardware on which the software runs are not located in DMZs. Solutions with VPNs would end in the DMZ. Resources of the Grid cannot be relocated.</td><td rowspan="4">3 DMZs are located in the communication path.</td><td rowspan="4">The security policy on both sides does not allow the opening of ports without inspection.</td></tr>
<tr><td>Ports used</td><td>Unknown/port range used</td></tr>
<tr><td>Protocol used</td><td>tcp</td></tr>
<tr><td colspan="2">Globus requires several ports to be opened for e.g. MyProxy Server, Web Service Container. GridFTP uses an unknown port range.</td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">GridFTP vs the Firewall</td></tr>
<tr><td>Description</td><td colspan="5">GridFTP protocol specifics and the reason why firewalls are not able to deal with it well.</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">low</td><td>low</td><td>low</td><td>high</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>NA</td><td>management</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">Unknown number of Packet filters/stateful firewalls monitoring based on 5-tuple of an IP packet</td><td>Own Software</td><td>Yes - GridFTP</td><td rowspan="4">Runs on Grid resources. Grid resources cannot be placed in the DMZ</td><td rowspan="4">Runs on Grid resources. Depending on the number of streams and the throughput desired using GridFTP, the firewall hardware might be a performance bottleneck.</td><td rowspan="4">Requires static opening of a large number of ports (1000+ at least) in the dynamic port range all the time in Firewall. This leads to a big security hole that security and network administrators are challenged to endorse.</td></tr>
<tr><td>Ports used</td><td>Unknown numbers / dynamically decided</td></tr>
<tr><td>Protocol used</td><td>Tcp</td></tr>
<tr><td colspan="2">Software requires multiple ports to run. Sockets/connections are added and deleted dynamically. Sockets determined dynamically per connection</td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">UNICORE</td></tr>
<tr><td>Description</td><td colspan="5">The UNICORE software (UNiform Interface to COmputing REsources) is a user-friendly software interface which allows easy and uniform access to distributed computing resources, and which provides support for running important scientific and engineering applications in a Grid environment. Scientists can use different supercomputers as well as other computing and storage resources without having to become experts in the special kind of access software and security policies of the various (super-)computer centers.</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>low</td><td>low</td><td>low</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>NA</td><td>NA</td></tr>
<tr><td rowspan="4"></td><td rowspan="4">Any kind of firewalls between the communicating entities.</td><td>Own Software</td><td>Available via sourceforge.org</td><td rowspan="4">No hardware restrictions.</td><td rowspan="4">Communication is done via normal communication paths.

Unicore client program connects to Unicore gateway. This connects internally to the Network Job Supervisor service</td><td rowspan="4">Protocol uses fixed configurable TCP port. Communication and access is allowed with certificates only. So there is only low security impact.</td></tr>
<tr><td>Ports used</td><td>One TCP port

Port is configurable

Depending on location of the NJS an additional port may be needed to be opened</td></tr>
<tr><td>Protocol used</td><td>TCP</td></tr>
<tr><td colspan="2"></td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">Firewalls and high bandwidth, long distance networks</td></tr>
<tr><td>Description</td><td colspan="5">This use-case describes a setup that allows the creation of (optical) by-pass connections that span long distances which need to be connected via a firewall</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">Low</td><td>High</td><td>middle</td><td>high</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>performance</td><td>manage-ment</td><td>management</td></tr>
<tr><td></td><td>Enterprise and public firewalls at both ends of a connection. Enterprise firewall both connects to the DMZ and to an optical by-pass connection.</td><td>Own Software</td><td>Yes and No GridFTP or any other datamover may be used – requirements are independant</td><td>Switching performance and buffer space is critical for the enterprise side of the firewall.

Buffers should be able to contain the bandwitdh/delay product of a long haul connection.

Performance should be in the multi-Gb range.</td><td>Enterprise firewall may be involved in driving the request of a by-pass connections when detecting private address space ARP requests or handling application specific signals using some protocol</td><td>1. Requests from an application to access the optical by-pass should be authorized. The firewall should call out to obtain such authorizations or be provisioned with information that recognises an access request.

2. Security policies should prevent hi-bandwidth / non TCP transmission protocol conformant traffic to be leaked into the regular Internet.</td></tr>
<tr><td></td><td></td><td>Ports used</td><td>Globus port range or others</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>Protocol used</td><td>TCP and UDP in various flavours</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2"></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>Name</td><td colspan="5">Web Services Firewall Issues</td></tr>
<tr><td>Description</td><td colspan="5">Clients outside a network protected by a firewall must be able to refer to the Web Service End Point Reference (EPR)</td></tr>
<tr><td></td><td>Elements in communication path</td><td colspan="2">Software</td><td>Hardware</td><td>Network</td><td>Security Policy</td></tr>
<tr><td>Severity</td><td></td><td colspan="2">High</td><td>NA</td><td>Low</td><td>High</td></tr>
<tr><td>Occurrence</td><td></td><td colspan="2">NA</td><td>NA</td><td>NA</td><td>Management</td></tr>
<tr><td rowspan="3"></td><td rowspan="3">The server’s network is protected by a firewall and a SOAP-proxy firewall in parallel, which acts as a gateway between external clients and WS Application Server.

Any other kind of firewall may be located between the client and the server.</td><td>Own Soft-ware</td><td>External clients must know to refer to the SOAP-proxy in order to reach Web Service EPRs.

Internal EPRs must be translated to external EPRs, in order to be reached through the SOAP-proxy.</td><td rowspan="3">Web Services are running on hosts located in the internal network.</td><td rowspan="3">Firewalls in the communicati on path may not allow direct connections.</td><td rowspan="3">It is not possible to know how many Web Services are running on a single port.

No way to express a policy that informs client to extend the security context end-to-end when communicating through the SOAP-proxy.

The SOAP-proxy must have the same or higher level of trust when EPRs are communicated to external clients.</td></tr>
<tr><td>Ports used</td><td>SOAP over HTTP (port 80).

More than one Web Service may run on the same port.</td></tr>
<tr><td>Proto-col used</td><td>TCP</td></tr>
<tr><td></td><td></td><td colspan="3">There is no standard mechanism to
• Augment an EPR with routing information
• Obtain an external EPR from an internal EPR
• Publish and discover external EPRs</td><td></td></tr>
</table>

12 Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

13 Full Copyright Notice

14 Normative References

[RFC 1631] The IP Network Address Translator (NAT)

[RFC 2663] IP Network Address Translator (NAT) Terminology and Considerations

[RFC 3234] Middleboxes: Taxonomy and Issues, http://www.ietf.org/rfc/rfc3234.txt

[RFC 3303] Middlebox communication architecture and framework, http://www.ietf.org/rfc/rfc3303.txt

15 Informational References

[ACDC] ACDC-Grid Firewall - „Advanced Computational Data Center Dynamic

Firewall (ACDC Dyna-Fire) Development“, http://www.ccr.buffalo.edu/grid/content/research

[D-Grid] AP7 - Design and deployment of firewall concepts within grid environments, Performance and dynamic configuration, http://www.d-grid.de

[dCache-1] dCache - Scope of the project, http://www.dcache.org

[dCache-2] dCache, the Book, 2003-2005, http://www.dcache.org/manuals/Book/

[UNICORE] UNICORE – The seamless Grid solution, http://unicore.org

[EGEE] EGEE, a European Project funded by EU, Service Activity 1 http://egee-sa1.web.cern.ch/egee%2Dsa1/Security.htm

[GPFS-1] General Parallel File System,

http://www-03.ibm.com/servers/eserver/clusters/software/gpfs.html

[GridFTP-1] GWD-R (Recommendation) GridFTP: Protocol Extensions to FTP for the Grid, April 2003, http://www.ggf.org/documents/GFD.20.pdf

[GridFTP-2] GWD-E GridFTP Protocol Improvements, July 2003, http://www.ggf.org/documents/GFD.21.pdf

[GridFTP-3] GridFTP v2 Protocol Description, May 2005, http://www.ggf.org/documents/GFD.47.pdf

[MIDCOM] Firewall Communication Protocol, http://www.iptel.org/fcp/

[OPSEC] OPSEC, Open Platform for Security (CheckPoint), http://www.opsec.com