AN IDENTITY SERVER PORTAL FOR GLOBAL ACCELERATOR (GAN) AND DETECTOR (GDN) NETWORKS

G. Grygiel, R. Kammering, S. Karstensen, K. Rehlich
Deutsches Elektronen Synchrotron DESY, Hamburg
in der Helmholtz-Gemeinschaft

Abstract

The next generations of large colliders and their experiments will have the advantage that groups from all over the world will participate with their competence to meet the challenges of the future. Therefore it's necessary to become even more global than in the past, giving members the option of remote access to most controlling parts of this facilities. The experience in the past has shown that a number of problems result from the existing variety of computer systems and their graphical user interfaces which are incompatible to other systems and the possible options to reach them from outside the experimental area. A group at Trieste and DESY is working inside the GANMVL (Global Accelerator Network Multipurpose Virtual Laboratory)[1] project to solve this problem, finding a simple way for the consumer to have remote access with single sign-on personalisation and admission to several systems. We determine problems arising in the implementation of user friendly interfaces, in achieving a look and feel close to the real hardware and in handling software. Also in the future it should be possible to have access simply via any internet browser, without any knowledge about the computer operating systems inside the large facilities. Only one login procedure should be necessary to have access to every integrated system. The current project status shall be outlined.

1 OVERVIEW STRUCTURE

Goal of the GANMVL is to design and build a system with several tools for remote control actions and observation of accelerator facilities including their experiments.

Figure 1 shows the structure of GANMVL components. Virtual instruments, electronic logbooks, high quality video and audio systems for eye-contact video conferences with the latest MPEG-4 based technology (which is not available on the market up to now) will also included in this tool as file and information sharing for accelerator and detector controls.

The system includes also a compact and transportable hardware setup with 3D-video screens, computer terminals, and instrumentation for measuring signals, virtual versions of oscilloscopes, network analyzers, spectrum analyzers and sockets for flexible network connections.

Integration of controls is mandatory, where problems have to be solved because of the numerous different systems. Case to case solution depends on the technology and safety rules of the control system to be connected to.

Solutions have to be discovering for TTF (thin clients (x-terminal)), HERA (thick clients, tine applications) and windows platforms. Some common tasks have to be found to make MVL aware to control rooms and users.

Desirable to do this in a generic way, this needs careful thinking.

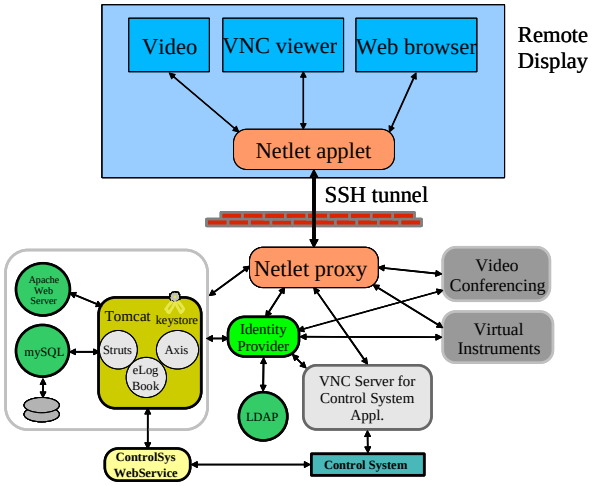

Figure 1

IDENTITY SERVER COMPONENTS

To understand the following chapters, it's evident explaining a few parts of the server mechanism.

1.1 Netlet - Applet and Proxy

Netlet [2] is an applet that runs on the browser and enables a secure connection between an arbitrary client on a system that is running a Java-enabled browser and a network resource behind a corporate firewall. The client can be behind a remote firewall and SSL proxy or directly connected to the Internet. Netlet can provide secure access to fixed port applications and some dynamic port applications that are available on the intranet from outside the intranet. All the secure connections made from outside the intranet to the intranet applications through the Netlet are controlled by Netlet rules.

Netlet listens to and accepts connections on preconfigured ports and routes both incoming and outgoing traffic between the client and the destination server. Both incoming and outgoing traffic is encrypted

using an encryption algorithm selected by the user, or configured by the administrator. The Netlet rule contains the details of all servers, ports, and encryption algorithms used in a connection. Administrators create Netlet rules by using the Identity Server administration console.

To provide the required service, Secure Remote Access includes the following components:

- Netlet applet
- Eproxy (Encryption proxy)
- Netlet proxy

1.2 NetFile

NetFile enables remote access and operation of file systems that reside within the corporate intranet in a secure manner, when accessed through the gateway. NetFile includes NetFile Java 1, an AWT-based user interface, and NetFile Java 2, a Swing-based user interface.

1.3 Load Balancer

Load balancers are network appliances with secure, real-time, embedded operating systems that intelligently load balance IP traffic across multiple servers. Load balancers optimize the performance of the site by distributing client requests across multiple servers, dramatically reducing the cost of providing large-scale Internet services and accelerating user access to those applications.

1.4 Rewriter Proxy

If a Rewriter proxy is installed, HTTP requests are redirected to the Rewriter proxy instead of directly to the destination host. The Rewriter proxy in turn sends the request to the destination server. Using the Rewriter proxy enables secure HTTP traffic between the gateway and intranet computers and offers two advantages:

- If there is a firewall between the gateway and server, the firewall needs to open only two ports—one between the gateway and the Rewriter proxy, and another between the gateway and the Portal Server.
- HTTP traffic is now secure between the gateway and the intranet even if the destination server only supports HTTP protocol (no HTTPS).

1.5 Eproxy and Rproxy

The gateway consists of Eproxy (Encrypted proxy) and Rproxy (Reverse proxy) subcomponents. Eproxy is responsible for handling Netlet components. Rproxy is responsible for all non-Netlet components. Depending on the type of request, the gateway performs the following:

- *Netlet request* - Eproxy checks session validity and routes the request (traffic) to the requested server. The destination of the Netlet traffic is determined by the Netlet rule that the user clicked in the Desktop.
- *HTTP(S) traffic* - Eproxy forwards the request to the Reverse proxy. When the Reverse proxy receives the request, it checks the session validity and forwards the request to the server as specified by the HTTP header. Upon receiving a response from the server, the Reverse proxy translates the response so that all intranet links within the response will work on the extranet.

2 AUTHENTICATION

Though Netlet does not authenticate users, it needs a valid SSO token for carrying out its operations. Netlet passes the SSO token with every request that is sent from the client to the Eproxy. Upon successful validation of the SSO token, the traffic is routed to the destination server. Additionally, the Netlet applet also displays a warning message before accepting any new connections (on the machine where the applet is running). The user must acknowledge this message and type the Netlet password before accepting new connections. You can configure this message to be a check box, or you can force the user to type a password before the connection is allowed.

3 TECHNICAL STRUCTURE

The topic here is to explain the connection to control system via VPN (virtual private network), particularly the relationship of Identity Server and Netlet [1] components, which are shown in Figure 2 and Figure 3.

3.1 Netlet structure

Figure 2 shows how Netlet fits into the Secure Remote Access platform. Here, each Netlet connection that is made to the destination host is described by a rule. The rule also describes the encryption algorithm to be used for that particular connection.

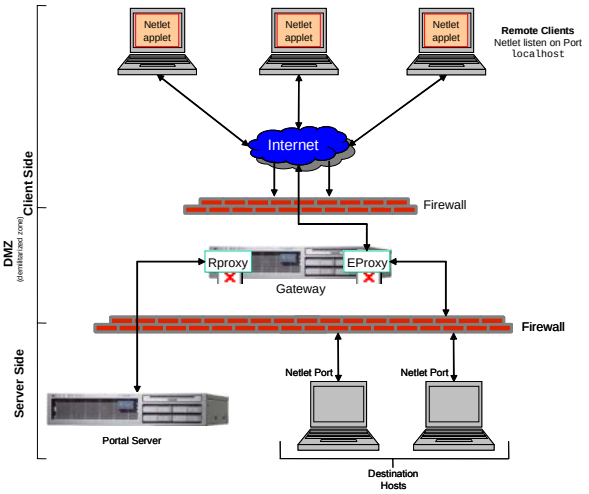

Figure 2

The Listen Port on the `localhost` is the client machine port on which the Netlet applet listens. The Netlet applet sets up an encrypted TCP/IP tunnel between the remote client machine and intranet applications on the remote hosts. The applet encrypts the packets and sends them to the gateway, and decrypts the response packets from the gateway and sends them to the local application.

All client requests are routed through the EProxy. EProxy handles only Netlet requests and passes any other request to the Rproxy. EProxy parses the Netlet requests and passes them to the Netlet proxy (if it is enabled) or directly to the destination host.

3.2 Netlet proxy

is an optional component. You can choose not to install it, or install it later. Netlet proxy extends the secure tunnel from the client, through the gateway to the Netlet proxy that resides in the intranet. This restricts the number of open ports in a firewall between the demilitarized zone (DMZ) and the intranet.

3.3 Netlet and Netfile structure

In Figure 3, two client browsers are redirected by a load balancer, which sits in the DMZ, to one of two gateways, also located in the DMZ. `Client_1` is performing a NetFile transaction. The NetFile traffic is routed by `Gateway_1` to `Portal Server_1`, whose Rewriter proxy directs the traffic to `Host_1`. `Client_2` is performing both Netlet and NetFile transactions. `Client_2`'s Netlet and NetFile requests are handled by `Gateway 2`, which routes the traffic to `Portal_Server_2`. The Rewriter proxy on this host directs the NetFile traffic to `Host_2`. The Netlet proxy on this host directs the Netlet request to `Application_Host_3`.

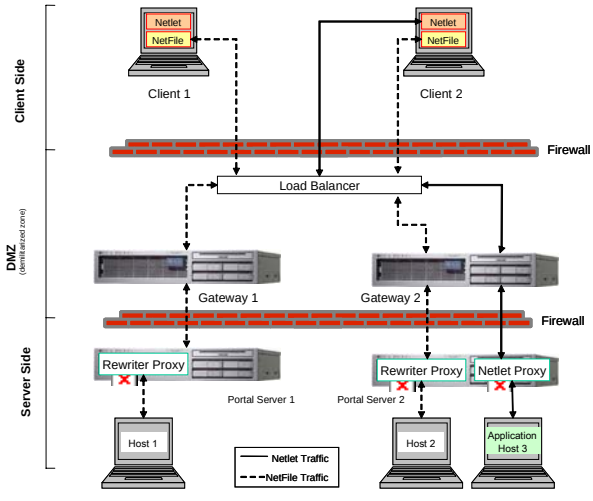

Figure 3

Secure mode uses the Secure Remote Access gateway, which typically resides in the DMZ. The gateway provides a single secure access point to all intranet URLs and applications, thus reducing the number of ports to be opened in the firewall. All other Portal Server services such as Session, Authentication, and the Desktop reside behind the DMZ in the secured intranet.

Communication from the client browser to the gateway is encrypted using HTTP over Secure Sockets Layer (HTTPS). Communication from the gateway to the server and intranet resources can be either HTTP or HTTPS.

4 ADVANTAGES

Netlet applet and server are tunnelling all information's, also for VNC, user permission, etc. This reduces many times over the costs and effort of administration as well as development. Since the applet has been written in JAVA, it is apparent, that all browsers, which are on the market now, will be able to use this product, without any adaptations.

5 PORTAL SERVER

The *Browser Desktop* [Figure 4] is the primary interface for the user to portal content. The Desktop service is implemented through a servlet, provider APIs, various channels, and various other support APIs and utilities. The Desktop uses programmatic entities called *providers* to generate content. A single unit of content is called a *channel.*

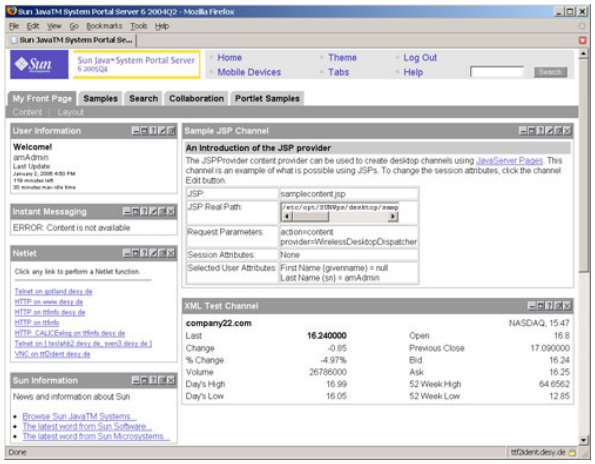

Figure 4

Multiple channels of content can be aggregated together into *container channels* and arranged in a variety of formats such as tables or tabs on the Desktop. When a user accesses the portal, the Desktop references a *display profile* which stores content provider and channel data used to generate the user's content. As confusing as it may sound, the display profile does not actually define the overall layout, display, or organization of what users see on their Desktops. Fundamentally, the display profile exists only to provide property values for channels. Actually, the Desktop uses multiple display profiles

stored as LDAP attributes at various levels or nodes in the Directory Server (top-most, organization, role, and user levels) to determine the content for a user. XML documents are used to define the display profile properties for each level and upload the property values into the LDAP node. At runtime, a user's display profile is created by merging the display profile properties defined at each level. Although a display profile document can be defined at each level, you do not need to have a display profile document at each level.

6 HARDWARE

Chip Multi-Threaded (CMT) processors provide support for many simultaneous hardware threads of execution in various ways, including Simultaneous Multithreading (SMT) and Chip Multiprocessing (CMP). CMT processors are especially suited to server workloads, which generally have high levels of Thread-Level Parallelism (TLP) [3].

For that reason we will use the new SUN Fire T2000 Server [Figure 5] with the following specifications:

- 8 core 1.0GHz UltraSPARC T1 processor
- 16GB DDR2 memory (16 * 1GB DIMMs)
- 2 * 73GB 2.5" 10K rpm SAS hard disk drives
- 1 DVD-RO/CD-RW slimline drive
- 2 (N+1) power supplies
- 4 10/100/1000 ethernet ports
- Solaris 10 and Java Enterprise System software
- Power consumption: 250 – 310 Watts

Figure 5

There will become also tests with standard PC's under Linux and shareware programs, to investigate the differences, pros and cons and, of cause, the price-performance ration.

7 PRACTICAL APPLICATION

As a real test object we are using existing parts of DESY network infrastructure. On one hand we have the well known TTF electronic Logbook [4] which is running at a UNIX web server, on the other hand we have a LINUX system with the PVSS [5] control system and a VNC server [6]. Figure 6 shows the functional structure of the portal server test device, including all parts described before.

Examining, if everything will work also under normal conditions during real action, we mixed all components without taking considerations for performance or special programming code.

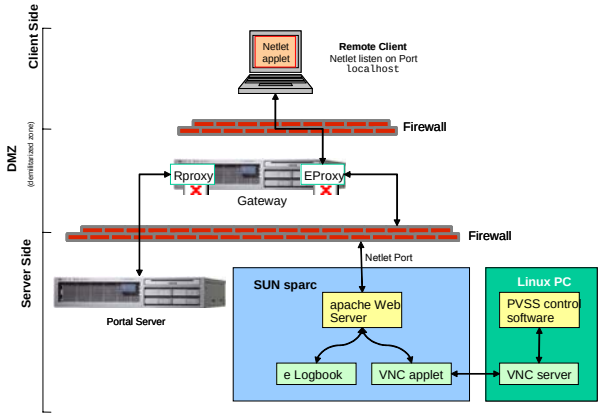

Figure 6

The result up to now is convincing [Figure 7]. Fast access is possible, depending of the Ethernet speed. Further swiftness tests with lower speed, like WLAN, are in preparation. Interesting to see is the address in the browsers navigation toolbar, which pointed to the localhost address 127.0.0.1 with port 40001, indicates the existence of Netlet.

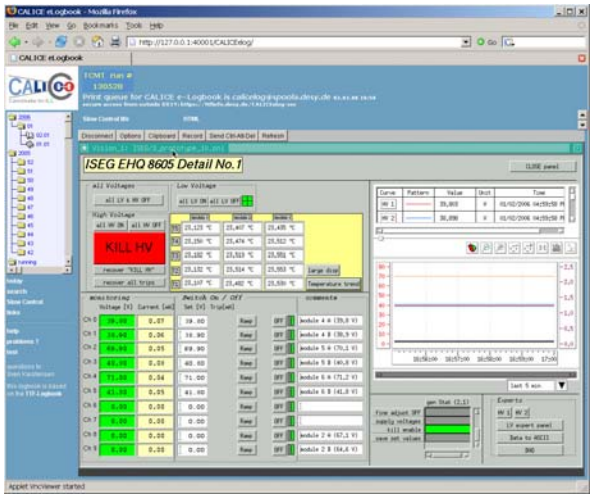

Figure 7

8 REFERENCES

This work is supported by the Commission of the European Communities under the 6th Framework Programme "Structuring the European Research Area", contract number RIDS-011899.

Please get the Poster for more detailed graphics at: http://www.desy.de/~sven/Poster

[1] http://www.eurotev.org/e558/e943
[2] Deployment Guide Sun ONE Portal Server
[3] Chip Multithreading: Opportunities and Challenges Lawrence Spracklen & Santosh G. Abraham, SUN
[4] http://tesla.desy.de/doocs/elogbook
[5] http://www.pvss.com
[6] http://www.realvnc.com