



Enabling Grids for E-science

gLite middleware IPv6 compliance

(sub-task TSA2.3.3)

Mario Reale - GARR

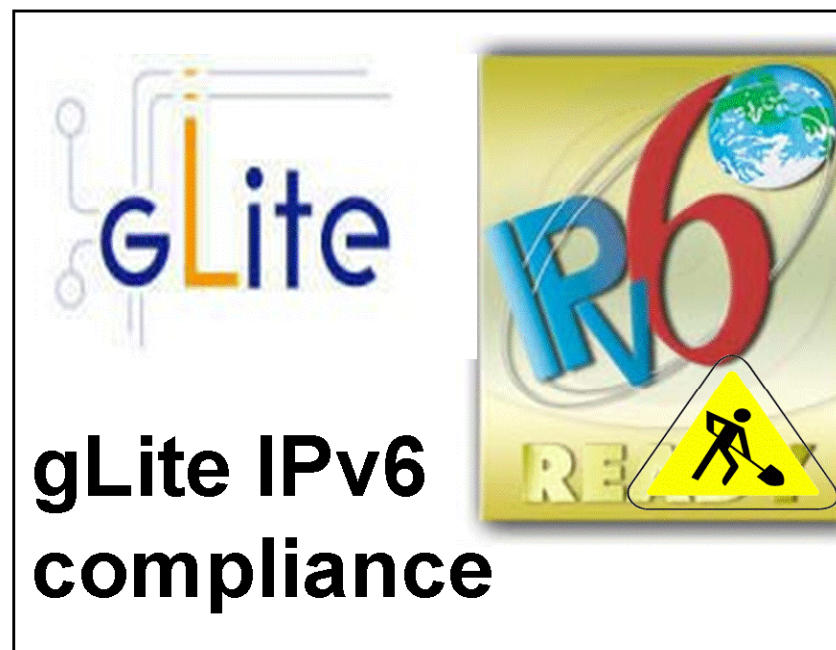
mario.reale@garr.it

EGEE SA2 – IPv6



EGEE08 - Istanbul

September 23rd, 2008



www.eu-egee.org



- **SA2 IPv6 testbed**
 - Layout and features

- **IPv6 testing of gLite components**
 - DPM, LFC, LCG-utils, WMproxy manual tests
 - ETICS based testing
 - Issues

- **gLite general IPv6 compliance assessment**
 - Overall situation
 - Bugs summary
 - One example correction for non-compliance:
 - AF-independent name resolving (*getaddrinfo*) in C/C++

- **Next steps and conclusions**

- SA2 manages an IPv6 testbed at GARR
deploying **gLite 3.1** on Scientific Linux CERN 4.6 nodes

- UI
- VOMS
- CE (lcg-CE) & Torque
- WN
- WMS
- LB
- BDII
- DPM-SE
- LFC



- Nodes are virtualized based on VM
- Hardware : HP ProLiant DL 380, 6 GB RAM, 2 TB HD)
- Supports the following gLite related VOs:
egee, dteam, infngrid, garr, eumed, euchina

- esxserver-cons.mng.dir.garr.it
 - DEV
 - DEV2
 - ETICS
 - GN-JRA1-TEST
 - GRID WN1 SLC 3.0.8
 - GRID WN2 SLC 3.0.8
 - SLC 4.6 0-UI2
 - SLC 4.6 9-RGMA-BDII
 - SLC 4.6 1-LFC
 - SLC 4.6 2-DPM1
 - SLC 4.6 3-WN1
 - SLC 4.6 4 -VOMS
 - SLC 4.6 5-DPM2**
 - SLC 4.6 7-WMS
 - SLC 4.6 8-CE
 - SLC 4.6 9-LB

SLC 4.6 5-DPM2

Summary Performance Events Console

General

Guest OS: **Other Linux (32-bit)**
 CPU: **1 vCPU**
 Memory: **256 MB**
 Memory Overhead: **68,36 MB**
 VMware Tools: **not installed**
 IP Address: ...
 DNS Name: ...
 State: **Powered Off**
 Host: **esxserver-cons.mng.dir.garr.it**
 Active Tasks:

Resources

CPU usage:
 Host memory
 Guest memory

Datstore	Capacity	Free
storage1	551.00 GB	160.86 GB

Network

if vlan 300-106

Commands

- Power on
- Edit Settings

Annotations

Notes: [Edit](#)

Recent Tasks

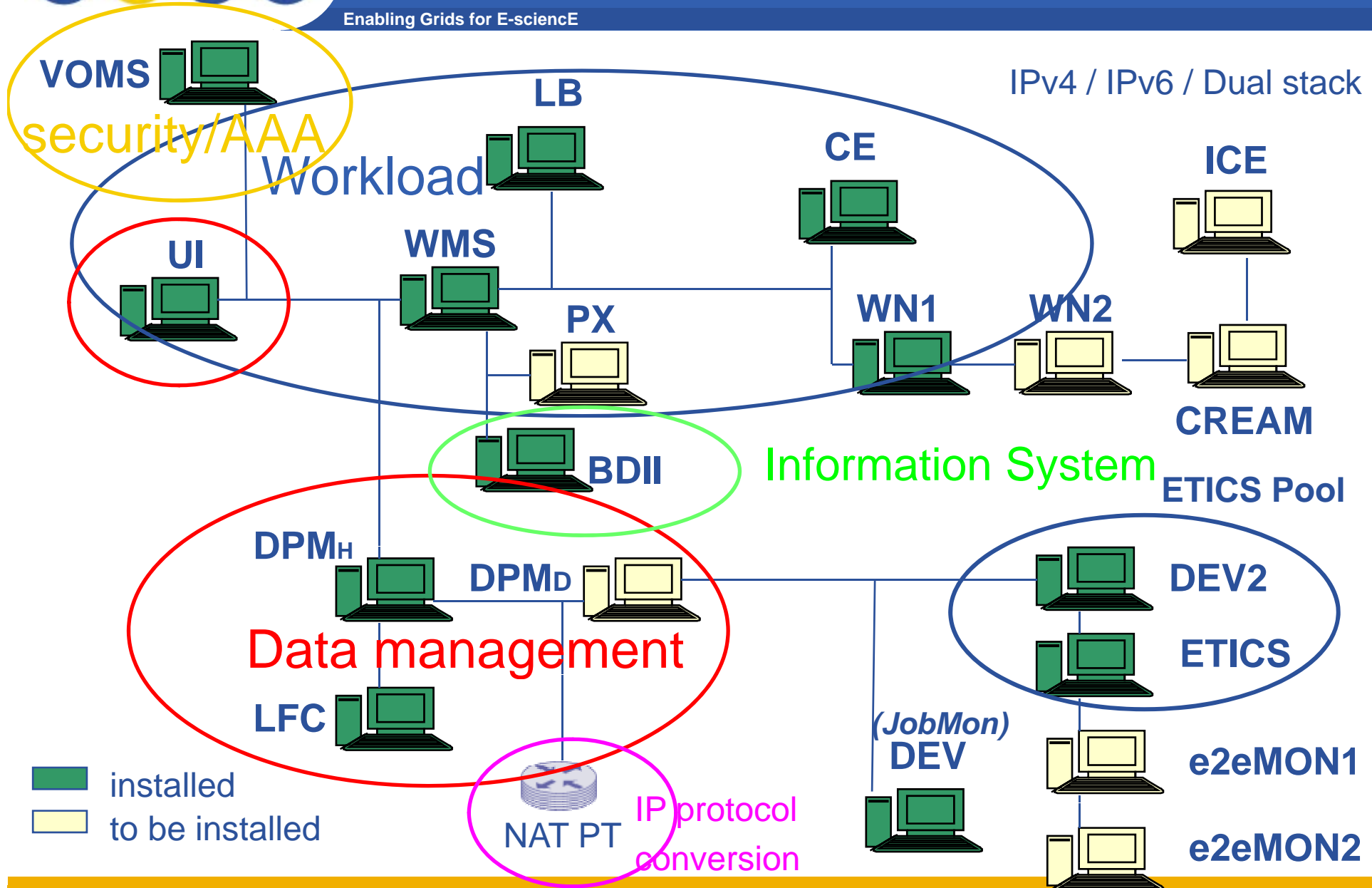
Name	Target	Status	Initiated by

SLC 4.6 0-UI2 on esxserver-cons.mng.dir.garr.it

```

Number of active connections has changed. There is now 1 active connection to this console
-rw-r--r-- 1 root root 62870 Oct 10 2007 install.log
-rw-r--r-- 1 root root 30353 Oct 10 2007 install.log.syslog
-rw-r--r-- 1 root root 48117502 May 2 2007 jdk-1_5_0_12-linux-i586
-rwxr-xr-x 1 root root 47793615 Mar 14 2008 jdk-1_5_0_12-linux-i586
-rw-r--r-- 1 root root 48130978 Oct 5 2007 jdk-1_5_0_14-linux-i586
-rwxr-xr-x 1 root root 47819569 Mar 14 2008 jdk-1_5_0_14-linux-i586
-rw-r--r-- 1 root root 48124055 Feb 9 2008 jdk-1_5_0_15-linux-i586
-rwxr-xr-x 1 root root 47811712 Oct 10 2007 jdk-1_5_0_15-linux-i586
-rw-r--r-- 1 root root 66590171 Dec 6 2007 jdk-6u6-linux-i586-rpm
-rw-r--r-- 1 root root 832 Aug 29 14:48 lista-di-rpm-externi.tx
-rw-r--r-- 1 root root 13117 Aug 29 14:49 lista-rpm-installati.li
-rw-r--r-- 1 root root 331404 Dec 1 2004 log4j-1.2.6-1.jpp.noarch
-rw-r--r-- 1 root root 656 Aug 29 14:49 risultato-dip-NON-trova
-rw-r--r-- 1 root root 176 Aug 29 14:49 risultato-dip-trovate.t
-rw-r--r-- 1 root root 1096 Aug 29 14:49 risultato.txt
-rw-r--r-- 1 root root 12865 Jan 2 2008 rpm-list-on-UI.txt
-rw-r--r-- 1 root root 42560 Mar 14 2008 rpm.php?jppversion=
-rw-r--r-- 1 root root 42560 Mar 14 2008 rpm.php?jppversion=.1
-rwxr-xr-x 1 root root 492 Aug 29 14:48 scanner-new.sh
drwxr-xr-x 2 root root 4096 Apr 2 13:22 siteinfo
drwx----- 2 root root 4096 Jul 8 10:23 .ssh
-rw-r--r-- 1 root root 25915 Dec 19 2007 storia-UOMS.txt
    
```

IPv4 / IPv6 / Dual stack



TESTBED NODES

Last Updated : Tuesday, Sept 16th, 2008

Node	IPv4	IPv6	Role	OS	Status	Notes	Scad.Certif.	Configuration Files
Gateway natpt	193.206.159.242 / 106.11	2001:760::159:242 / 2001:760:0:106:242 2001:760:0:159:242	ROUTER / NAT-PT		OK			
voms-4	193.206.106.20	2001:760:0:106::20	VOMS server	SLC 4.6	OK	installed 20080512 - Some issues with proxy		
ui-4	193.206.159.212	2001:760:0:159::212	PC mario NFS repository	SLC 4.6	OK	Install & Config 20080314 ETICS CLIENT		
ui2-4	193.206.106.21	2001:760:0:106::21	User Interface	SLC 4.6	OK	Install & Config 20080314		
wms-4	193.206.106.22	2001:760:0:106::22	WMS node	SLC 4.6	OK	Install & COnfig 20080801	20/12/2008 OK	
lb-4	193.206.106.50	2001:760:0:106::50	LB node	SLC 4.6	OK	Install & COnfig 20080805	4/8/2009 OK	
ce-4	193.206.106.23	2001:760:0:106::23	Computing Element	SLC 4.6	OK			
rgma-bdii-4	193.206.106.24	2001:760:0:106::24	site BD II	SLC 4.6	OK			
wn1-4	193.206.106.25	2001:760:0:106::25	Worker Node	SLC 4.6	OK	ETICS CLIENT		
etics-4	193.206.106.37			SLC 4.6	OK	ETICS NMI Pool node		
dev2-4	193.206.106.35			SLC 4.6	OK	ETICS NMI Pool node		
dev-4	193.206.106.34			SLC 4.6		CronJobs GRID Monitoring		
se-4	193.206.106.26		Storage Element	SLC 4.6		Not installed		
lfc-4	193.206.106.29	2001:760:0:106::29	LFC File Catalog IPv6	SLC4	OK	David Smith's ported vs		
dpm1-4	193.206.106.30	2001:760:0:106::30	DPM HEAD NODE	SLC4	OK	David Smith's ported vs		User dpmmgr:dpmmgr aggiunto a mano

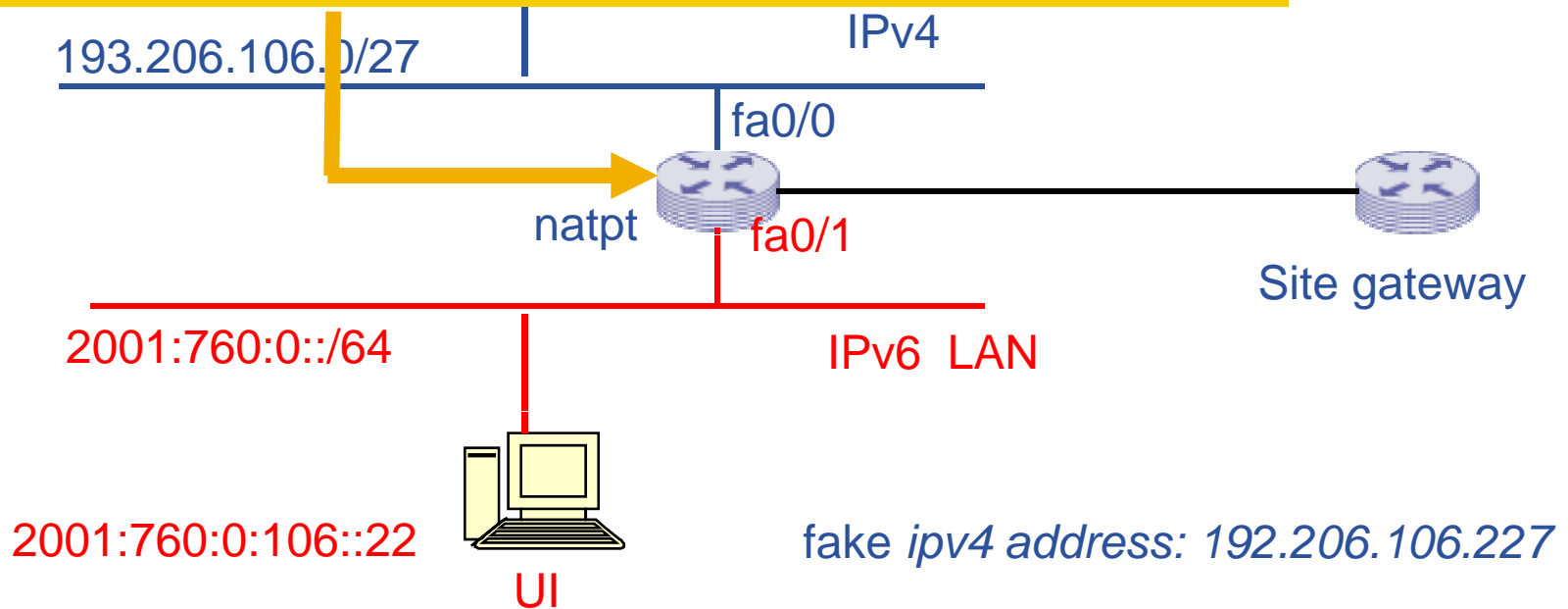
- Each node can be configured in IPv4, IPv6, Dual Stack
- Installed via YUM
- Configured via YAIM (*apart from VOMS (Python&XML)*)
- Static NAT-PT protocol conversion is available for specific Client/Server studies
 - IPv4 to IPv6
 - IPv6 to IPv4
- UI account available for testers/developers SA2/JRA1/SA3
 - other kinds of access to testbed nodes can be agreed

```

interface fa0/0
    ip address 193.206.106.242/24 255.255.255.0
    ipv6 nat prefix 2001:760:0:106:EC:/96
    ipv6 nat
interface fa0/1
    ipv6 address 2001:760:0::242/64
    ipv6 nat
ipv6 nat v6v4 source 2001:760:0:106::24 192.206.106.227
ipv6 nat v4v6 source 193.206.106.24 2001:760:0:106:EC::180
    
```

in the Grid

0:0:106:EC::180



- **A few gLite components have been tested for IPv6 compliance already**
 - DPM (ported to IPv6 on Dec 07 by David Smith/CERN)
 - LFC (ported to IPv6 on Dec 07 by David Smith/CERN)
 - LCGutils
 - Workload Management / WMproxy

- **DPM and LFC proven to be fully IPv6 compliant ☺**
 - Regression tests on IPv4 also performed and successfully passed
(<https://edms.cern.ch/file/946408/1/EGEE-III-SA2-TEC-946408-lfc-tests-using-ipv6-v1.0.doc>)

- **LCGutils and WMproxy showed non compliance ☹**
 - LCGutils: **gSOAP and LDAP connection problems**
 - CGSI-gSOAP: Could not open connection !
 - Can't contact LDAP server
 - WMproxy: **name resolving problems**
 - Error - Wrong Value Unable to resolve the hostname: wms.dir.garr.it

- Nodes have been **registered in DNS** with both an IP4 FQDN (*lfc-4.dir.garr.it ..*) and an IPv6 one (*lfc.dir.garr.it, t=AAAA*)
- First **both IPv4 and IPv6 have been turned on** on the nodes, and all IPv4 tests have carried out to verify expected functionality was there. (*regression tests*)
- Second, **IPv4 has been turned off on the UI** by de-configuring the IP address and the gateway, leaving only IPv6 up, and all functionality tests have been repeated.
- The **site BDII** has been configured in **dual stack mode** and both resource names have been entered in the LDAP (*lfc-4, lfc*)
- **IPv4 was not switched off on the LFC / DPM / WMS servers themselves** (*due to MySQL connection problems in IPv6*)

• Tested commands:

[IPv6 compliance (OK/not OK)]

- **lfc-ping** [OK]
- **lfc-ls** /grid [OK]
- **lfc-mkdir** /grid/infngrid/testipv4 [OK]
- **lfc-listgrpmap** --group infngrid [OK]

LFC

- **lcg-cr** --vo infngrid -l testipv4fileNew -d dpm1-4.dir.garr.it ipv4testfileNew [not OK]
- **lcg-cp** --vo infngrid lfn:testipv4fileNew file:testipv4fileNew-retrieved [not OK]
- **lcg-rep** --vo infngrid -d dpm1-4.dir.garr.it guid:c341fce4-7a31-4ef4-8697-57cda60108b4 [not OK]
- **lcg-lr** --vo infngrid lfn:/grid/infgrid/testipv4fileNew [OK]
- **lcg-lg** --vo infngrid srm://dpm1-4.dir.garr.it/dpm/dir.garr.it/home/infgrid/generated/2008-07-09/file808f7b86-33fc-4c66-afba-de5c804f51fc [OK]
- **lcg-infosites** --vo infngrid all [not OK]

LCGutils

- **dpm-ping** -h dpm1-4.dir.garr.it [OK]
- **dpm-reservespace** --gspace 2G --lifetime Inf --group infngrid --token_desc infngrid_ESD [OK]
- **dpm-qryconf** [OK]
- **dpns-ls** / [OK]

DPM

- [reale@ui2-4 JDL]\$ `glite-wms-job-delegate-proxy -d second`
Error - Wrong Value
Unable to resolve the hostname: wms.dir.garr.it
[not OK]
- [reale@ui2-4 JDL]\$ `glite-wms-job-submit -d first minimal.jdl`
Error - Wrong Value
Unable to resolve the hostname: wms.dir.garr.it
[not OK]
- [reale@ui2-4 JDL]\$ `glite-wms-job-list-match -d first minimal.jdl`
Error - Wrong Value
Unable to resolve the hostname: wms.dir.garr.it
[not OK]

WMproxy

- **Motivation**
 - Integrate the IPv6 tests into the ETICS build & test system to ease the verification of IPv6 compliance of gLite for developers testers, and release certifiers.
- **Created a reference project for IPv6 in ETICS called *gLite_ipv6***
 - org.glite.testsuites.ipv6 reference component
 - Various configurations for different testing scripts
 - corresponding CVS repository:
 - <http://glite.cvs.cern.ch:8180/cgi-bin/glite.cgi/org.glite.testsuites.ipv6/>
- **Included IPv6 resources into the ETICS metronome pool**
 - Currently 2 nodes at GARR
 - reachable selecting the IPv6 checkbox from WA
- **Performed many tests related to IPv6 up to now**
 - Automatic deployment of UI and BDII
 - BDII functional tests
 - Protocol switching tests on the nodes (IPv4 → IPv6-→IPv4)
 - DPM-LFC tests currently being ported
- **Move from a single prototypal *catch-all* component to the test method of each gLite component in the build&test system as soon as IPv6 tests are made available**
 - Make IPv6 tests within ETICS ordinary *business as usual* procedures

- **set up of IPv6 infrastructure and IPv6 match making**
- **ported IPv6 BDII server query from ETICS job**
 - Ported BD-II server in Paris was queried via NATPT by an ETICS job running at CERN – demonstrated our idea
- **UI and BDII installation in user space**
- **Protocol switching on the nodes and the installed UI**
- **Currently porting the DPM-LFC tests to fully automate them**

- **inclusion of IPv6 address format and URLs within the YAIM configuration scripts**
 - All IPv6 related changes have currently to be done by hand
- **availability of a gLite IPv6 repository**
 - installing a UI for example:.....
 - `cd /etc/yum.repos.d/`
 - `wget http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/glite-UI.repo`
 - `yum update`
 - `yum install glite-UI`**doesn't** currently work under IPv6
- **global IPv6 connectivity at CERN**
 - Although global WAN IPv6 connectivity is not essential to test basic functionality, is highly desirable to perform extended IPv6 tests
 - Including CERN would be important for its strategic role and would allow easier inclusion of IPv6 resources into ETICS

- Essentially all the gLite components (but a couple) show evident IPv6 non compliance in the source code
- Around **110 IPv6 bugs** have been submitted to the gLite bug tracking system
- Start point for bug submission have been the results of the IPv6 code checker (ETICS IPv6 metric):
 - seeking specific non IPv6 compliant calls/data structures in the source code
- Each bug (normally) reports filename, module name and line number of non compliant calls and suggested possible solutions (only a reference to the compliant equivalent functions)
 - Although some bugs are more collective and less precise
- 2 posted IPv6 bugs are **functional bugs** (LCGutils and WMproxy failures in IPv6) (bug# 39890, 41844)

- **A global analysis on all glite code has been carried out by means of the IPv6 metric of ETICS (IPv6 code checker)**
 - org.glite project on ETICS
 - org.glite CVS check out
- **The code checker is optimized for C/C++ but spots also evident JAVA and Python problems**
- **In C/C++ spread IPv4 only (=non IPv6 compliant) usage of**
 - AF_INET
 - sockaddr_in
 - inet_ntoa()
 - INADDR_ANY
 - gethostbyname()
 - gethostbyaddr()
- **Bugs have been posted according to the problems found.**

- AMGA	6	- LB	14
- APEL	1	- RGMA	6
- CE/CEmon	4	- SECURITY / VOMS	12
- Data Management	6	- Service Discovery	-
- DPM	1	- SLCS	-
- FTS	-	- WMS	8
- DGAS	10	- YAIM	-
- JP	2	- Gridsite	2
- E2EMONIT	2	- Hydra	-
- GPBOX	5	- SRM	-
- ICE	1	- JDL	-
- CREAM	3	- UI	3
- LFC	-		

- For AF independent / Network Transparent Programming it is important to pay attention to:
 - Use of name instead of address in applications is advisable; in fact, usually the hostname remains the same, while the address may change more easily. From application point of view the name resolution is a system-independent process.
 - Avoid the use of hard-coded numerical addresses and binary representation of addresses.
 - Use *getaddrinfo* and *getnameinfo* functions.

The **gethostbyname()** for IPv4 and **gethostbyname2()** function created for IPv6 **was deprecated** in RFC 2553 and was replaced by **getaddrinfo()** function.
(see RFC 3493 and RFC 3542)

```
#include <netdb.h>
struct hostent *gethostbyname(const char *name)
```



DEPRECATED

```
#include <netdb.h>
#include <sys/socket.h>
struct hostent *gethostbyname2(const char *name, int af)
```



DEPRECATED

```
#include <netdb.h>
#include <sys/socket.h>
int getaddrinfo(const char *nodename, const char *servname,
               const struct addrinfo *hints, struct addrinfo **res);
```

getaddrinfo() takes in input

- a service name like “http” or a numeric port number like “80”
- a FQDN

and returns

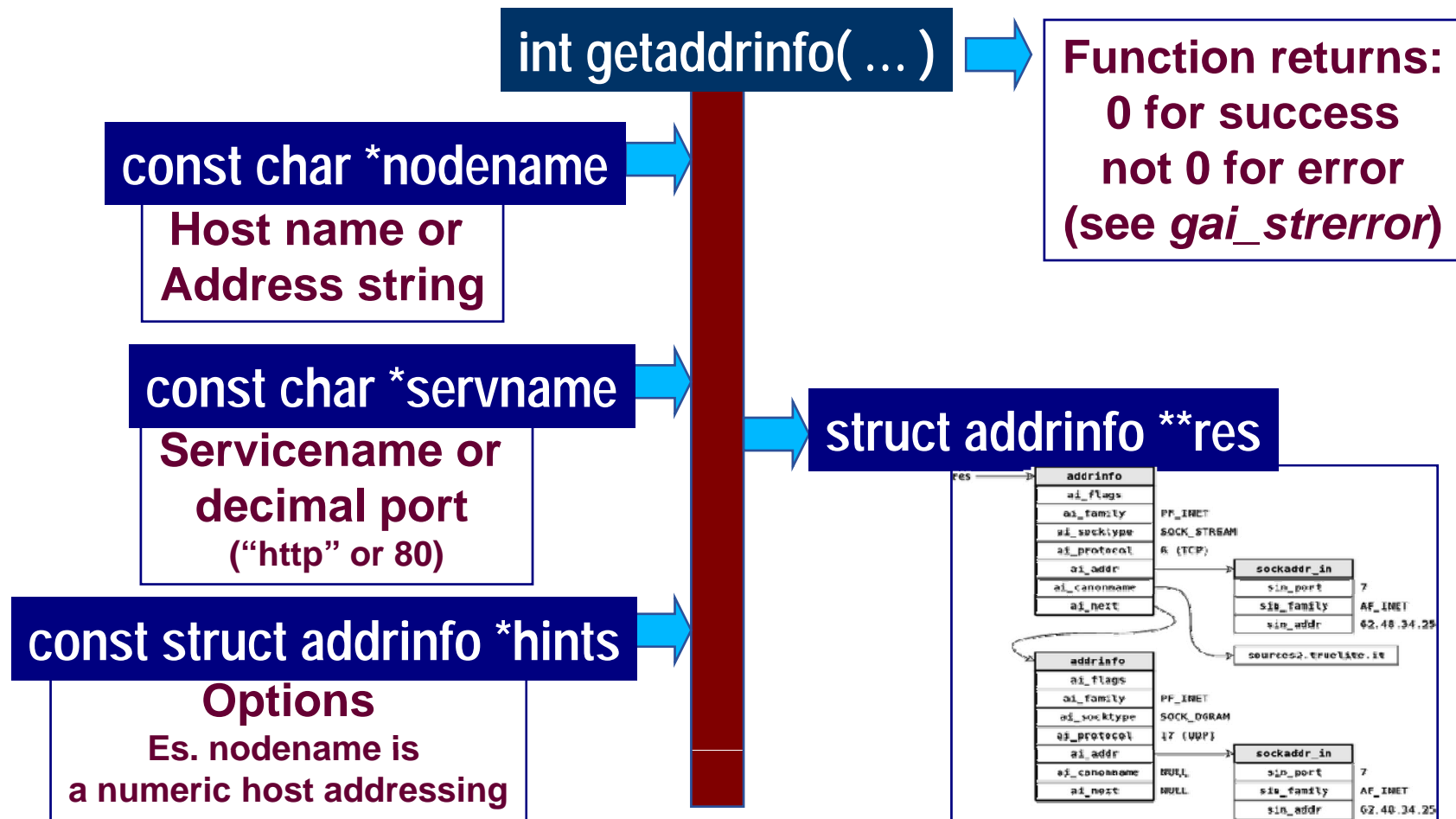
- a list of addresses along with the corresponding port number.

The *getaddrinfo* function is very flexible and has several modes of operation. It **returns a dynamically allocated linked list** of *addrinfo* structures containing useful information (for example, *sockaddr* structure ready for use).

```
#include <netdb.h>
#include <sys/socket.h>

int getaddrinfo(const char *nodename,
               const char *servname, const struct addrinfo *hints,
               struct addrinfo **res);
```

Nodename-to-address translation is done in a protocol-independent way using the *getaddrinfo()* function.



```
int getaddrinfo(const char *hostname, const char *servname,  
               const struct addrinfo *hints, struct addrinfo **res);
```

hostname is either a hostname or an address string.

servname is either a service name or decimal port number string.

hints is either a null pointer or a pointer to an addrinfo structure that the caller fills in with hints about the types of information he wants to be returned.

(see next slide)

The caller can set these values in the *hints* structure:

```

struct addrinfo {
    int      ai_flags;        // AI_PASSIVE, AI_CANONNAME, ..
    int      ai_family;      // AF_XXX
    int      ai_socktype;    // SOCK_XXX
    int      ai_protocol;    // 0 or IPPROTO_XXX for IPv4 and IPv6
    socklen_t ai_addrlen;    // length of ai_addr
    char     *ai_canonname;  // canonical name for nodename
    struct sockaddr *ai_addr; // binary address
    struct addrinfo *ai_next; // next structure in linked list
};
    
```

ai_family: the protocol family to return

AF_INET, AF_INET6, AF_UNSPEC [any OS supported protocol family])

ai_socktype: type of required socket:

SOCK_STREAM, SOCK_DGRAM, or SOCK_RAW.

(when *ai_socktype* is zero the caller will accept any socket type)

ai_protocol: Indicates which transport protocol is desired, IPPROTO_UDP or IPPROTO_TCP. If *ai_protocol* is zero the caller will accept any protocol.


```
struct addrinfo {  
    int      ai_flags;      // AI_PASSIVE, AI_CANONNAME, ..  
    [...]     
};
```

ai_flags shall be set to zero or be the bitwise-inclusive OR of one or more of the values:

AI_PASSIVE

if it is specified the caller requires addresses that are suitable for accepting incoming connections. When this flag is specified, *nodename* is usually *NULL*, and address field of the *ai_addr* member is filled with the "any" address (e.g. *INADDR_ANY* for an IPv4 or *IN6ADDR_ANY_INIT* for an IPv6).

AI_CANONNAME

the function shall attempt to determine the canonical name corresponding to *nodename* (The first element of the returned list has the *ai_canonname* filled in with the official name of the machine).

```
struct addrinfo {  
    int      ai_flags;      // AI_PASSIVE, AI_CANONNAME, ..  
    [...]     
};
```

AI_NUMERICHOST

specifies that nodename is a numeric host address string. Otherwise, an [EAI_NONAME] error is returned. This flag shall prevent any type of name resolution service (for example, the DNS) from being invoked.

AI_NUMERICSERV

specifies that servname is a numeric port string. Otherwise, an [EAI_NONAME] error shall be returned. This flag shall prevent any type of name resolution service (for example, NIS+) from being invoked.

AI_V4MAPPED

if no IPv6 addresses are matched, IPv4-mapped IPv6 addresses for IPv4 addresses that match *nodename* shall be returned. This flag is applicable only when *ai_family* is AF_INET6 in the hints structure.

```
struct addrinfo {  
    int      ai_flags;      // AI_PASSIVE, AI_CANONNAME, ..  
    [...]     
};
```

AI_ALL

If this flag is set along with AI_V4MAPPED when looking up IPv6 addresses the function will return all IPv6 addresses as well as all IPv4 addresses. The latter mapped to IPv6 format.

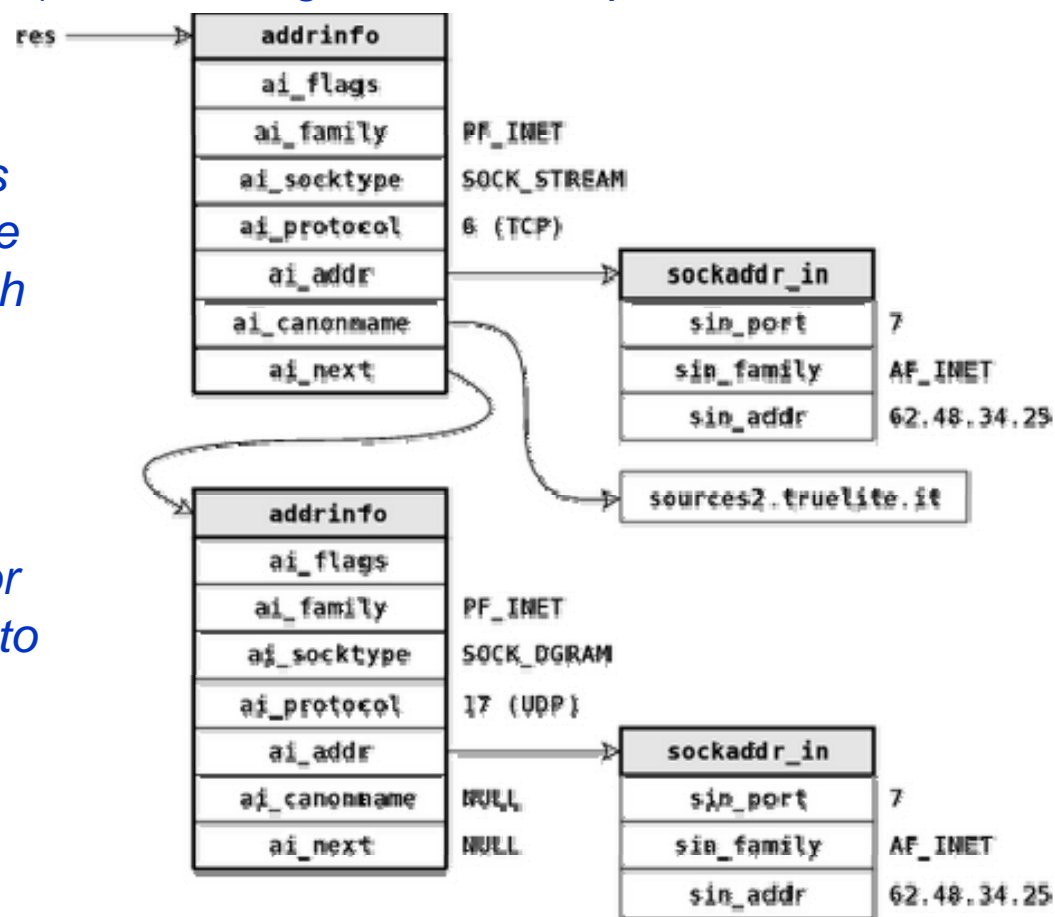
AI_ADDRCONFIG

Only addresses whose family is supported by the system will be returned: IPv4 addresses shall be returned only if an IPv4 address is configured on the local system, and IPv6 addresses shall be returned only if an IPv6 address is configured on the local system. The loopback address is not considered for this case as valid as a configured address.

If getaddrinfo returns 0 (success) *res* argument is filled in with a pointer to a linked list of addrinfo structures (linked through the *ai_next* pointer).

In case of multiple addresses associated with the hostname one struct is returned for each address (usable with hint.ai_family, if specified).

One struct is returned also for each socket type (according to hint.ai_socktype).



```

struct addrinfo {
    int      ai_flags;      /* AI_PASSIVE, AI_CANONNAME, .. */
    int      ai_family;    /* AF_XXX */
    int      ai_socktype;  /* SOCK_XXX */
    int      ai_protocol;  /* 0 or IPPROTO_XXX for IPv4 and IPv6 */
    socklen_t ai_addrlen;  /* length of ai_addr */
    char     *ai_canonname; /* canonical name for nodename */
    struct sockaddr *ai_addr; /* binary address */
    struct addrinfo *ai_next; /* next structure in linked list */
};
    
```

The information returned in the *addrinfo* structures is ready for socket calls and ready to use in the *connect*, *sendto* (for client) or *bind* (for server) function.

ai_addr is a pointer to a socket address structure.

ai_addrlen is the length of this socket address structure.

ai_canonname member of the first returned structure points to the canonical name of the host (if AI_CANONNAME flag is set in hints structure).

- **Inclusion of ICE / CREAM in the SA2 testbed**
 - Perform general purpose functionality tests in IPv6 for it
- **Analyze the behaviour of services when MySQL is configured to connect locally on the node (using a local unix socket instead of tcp/ip connection (configuration issues))**
 - For example starting from LFC/DPM components
- **Organize a gLite 3.1 IPv6 repository to provide RPMs and source code directly in IPv6**
- **Perform another systematic scan of the source code to complete the IPv6 bug posting**
 - Especially on all the configuration related components
 - Also with improved versions of the code checker once available

- **gLite is overall of course still non-IPv6 compliant ☹, but:**
 - IPv6 ported components started popping up
 - a much clearer view on the amount and exact location of IPv6 non-compliance is now available (w.r.t. only few months ago)
 - awareness about IPv6 non compliance has grown - globally
- **We are getting into a an interesting phase:**
 - we do not only see the top of the iceberg
 - a testing methodology and a set of tools is in place and operational
 - SA2 testbed
 - IPv6 expertise and knowledge
 - Some reference documents on IPv6 for the gLite community are now available
 - ETICS testing tools and their corresponding automation in testing procedures
- **There is a lot of work to do, but it's a very challenging activity whose outcome could finally allow us to make gLite IPv6 compliant**
 - a useful investment for its long term sustainability and for IPv6 users

- I'd like to thank Rino Nucara and Cristiano Valli/GARR for their support

References

EGEE SA2 IPv6 wiki at

<https://twiki.cern.ch/twiki/bin/view/EGEE/IPv6FollowUp>

AF-independent network programming:

<http://www.kame.net/newsletter/19980604/>

http://www.6journal.org/archive/00000047/01/porting_ipv4tov6.pdf

<http://people.redhat.com/drepper/userapi-ipv6.html>



GLite R6 READY

