

gLite IPv6 compliance overall status and on going activities

v0.4

Etienne Dublé - CNRS/UREC EGEE SA2

etienne.duble@urec.cnrs.fr



Mario Reale – GARR

mario.reale@garr.it

EGEE SA2



Tuesday, September 22, 2009 – Barcelona – EGEE 09 Conference

- **gLite: a complex architecture**
- **EGEE achievements regarding IPv6**
- **Current status and foreseen evolution concerning IPv6 and gLite**
- **Next steps**

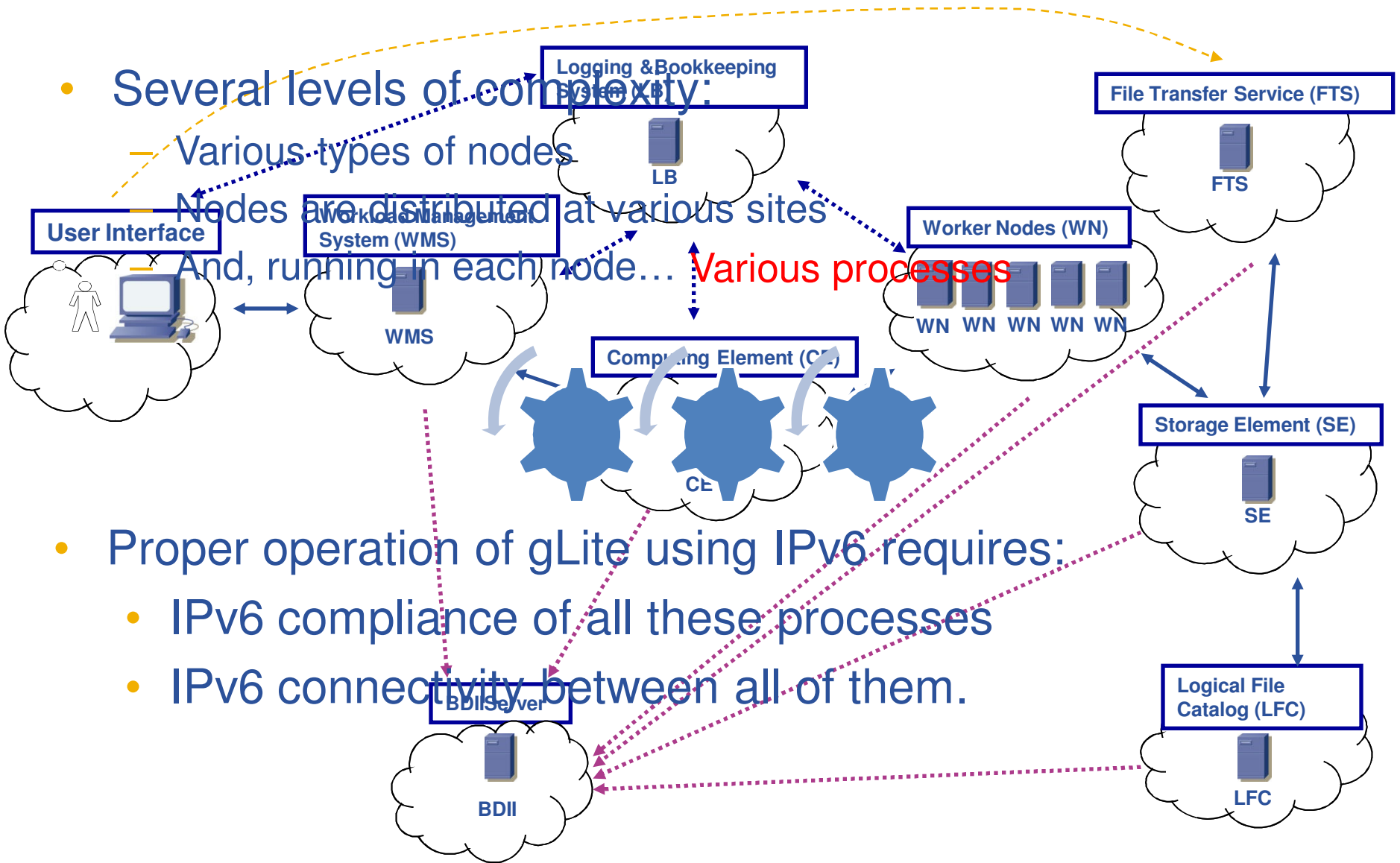
gLite: a complex architecture

- Several levels of complexity:

- Various types of nodes

- Nodes are distributed at various sites

- And, running in each node... **Various processes**



- Proper operation of gLite using IPv6 requires:

- IPv6 compliance of all these processes
 - IPv6 connectivity between all of them.

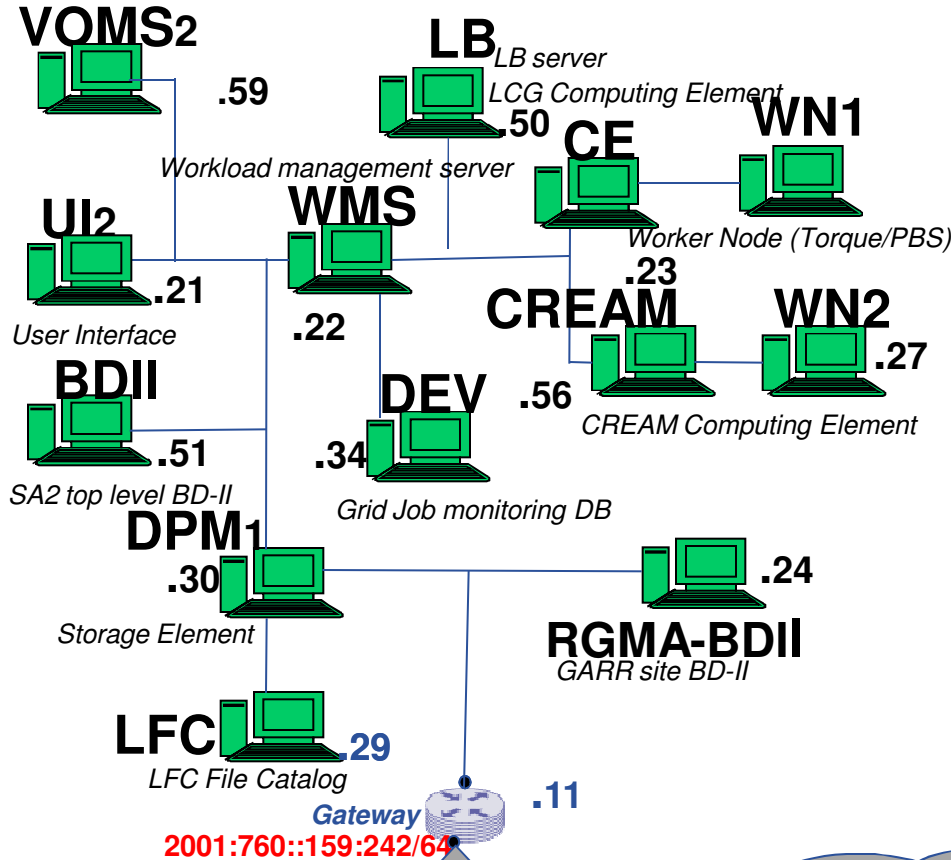
EGEE IPv6 achievements

- Definition of a methodology for IPv6 transition
- Collaboration with ETICS (*initially also with EuChinaGRID*):
 - Goal: automate IPv6 compliance testing of gLite packages
 - A static code checker was developed. Results are displayed as the “IPv6 metric” on ETICS website
- Development of a first IPv6-compliant gLite node: the BDII
- Building of a testbed in Paris and in Rome. Demos of:
 - The NAT-PT protocol usage (ex. with the ported BDII)
 - The dual-stack environment usage (ex. with the ported BDII)
- Report about IPv6 compliance of gLite external dependencies

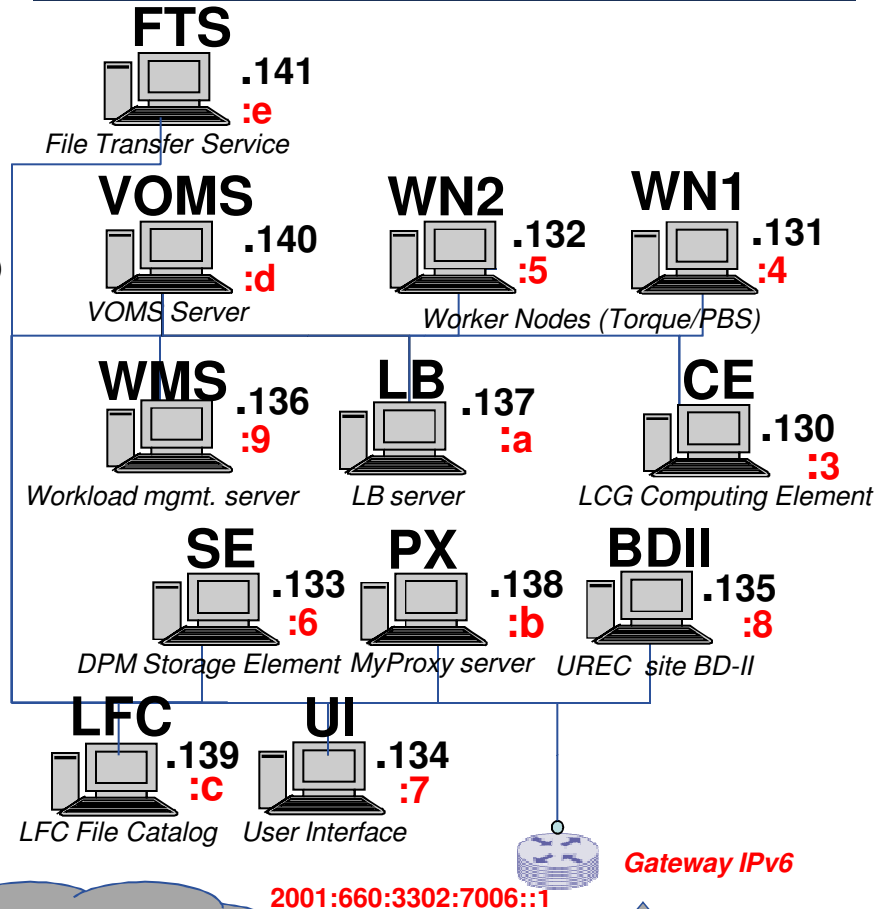
- Set up all elements needed to handle IPv6 in EGEE:
 - Middleware, collaboration with JRA1
 - gLite internal dependencies, IPv6 compliance
 - *Testbed*
 - *Support to developers*
 - gLite external dependencies
 - Certification process, collaboration with SA3:
 - Incorporate IPv6 in the certification process
 - IPv6 knowledge dissemination
 - Training course, presentation
- Assess and make available an operational EGEE IPv6 site (depending on the availability of IPv6 compliant gLite modules)

- It is a follow-up of the testbeds built during EGEE II
- Goals of this testbed:
 - Test IPv6 compliance of gLite components
 - Demonstrate current gLite IPv6 features
 - Integrate the certification pool (collaboration with SA3)
- It provides:
 - **The two first dual-stack sites in EGEE:**
 - One in Rome (GARR)
 - One in Paris (UREC)
 - Site-to-site IPv6 connectivity (through UREC / RAP / RENATER / GEANT / GARR networks)
 - Many relevant gLite nodes installed and configured: VOMS, UI, WMS, lcg-CE, CREAM, WNs, DPM-SE, LFC, PX, BDII, LB, FTS. All of them are dual-stack.

GARR/ROME



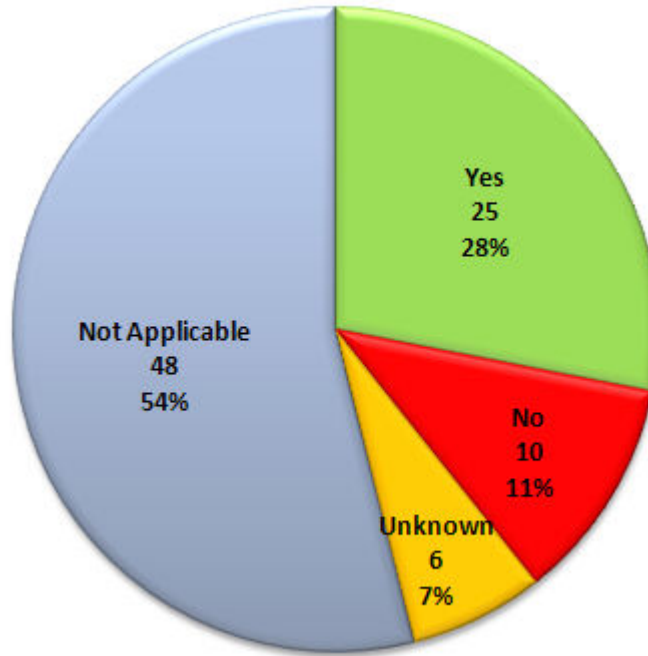
UREC/PARIS



IPv4/IPv6 Internet:
Renater/GEANT/GARR

- Analysis of the gLite source code and **bug reporting**
 - Using the IPv6 metric (IPv6 code checker) in ETICS
 - Around 110 bugs on non-compliant function calls and data structures in the code reported
- This analysis effectively incited developers to work on IPv6, and is a good metric regarding the work needed on this subject.

- **Analysis of all gLite external dependencies** to assess their IPv6 compliance



- Most of them are IPv6 compliant
- MySQL is the most relevant package which is not IPv6 compliant
 - 6 of the 10 non-compliant packages are MySQL-subpackages
 - However if the server and the client run on the same dual-stack node, there is no problem.

- **Other reports:**
 - IPv6 Programming methods:
 - [Guide](#) to IPv6 compliant programming in C/C++, Java, Python and Perl:
 - A sample TCP client and server for each programming language
 - Advantages/drawbacks/limitations of each language regarding IPv6
 - IPv6 Testing methods:
 - [Report](#) explaining how to test if a TCP server is IPv6 compliant
 - IPv6 Test reports:
 - Selected IPv6 compliance studies for specific packages: [gSOAP](#), [Axis](#), [Axis2](#), [Boost:asio](#), [gridFTP](#), [PythonZSI](#), [PerlSOAPLite](#)
 - [Assessment](#) of the IPv6 compliance of gLite components: DPM & LFC

- A tool called **IPv6 CARE**
 - Developed by SA2
 - Behavior:
 - It monitors the execution of programs and ***detects networking function calls***
 - The diagnosis generated helps to ***assess IPv6 compliance***
 - Very generic: you can use it for any program, even if you don't have the source code
 - Available on sourceforge: <http://sourceforge.net/projects/ipv6-care>

- **Usage example:** we want to test the “telnet” program
 - We run the program as usual except that we prefix the command by “`ipv6_care check`”:

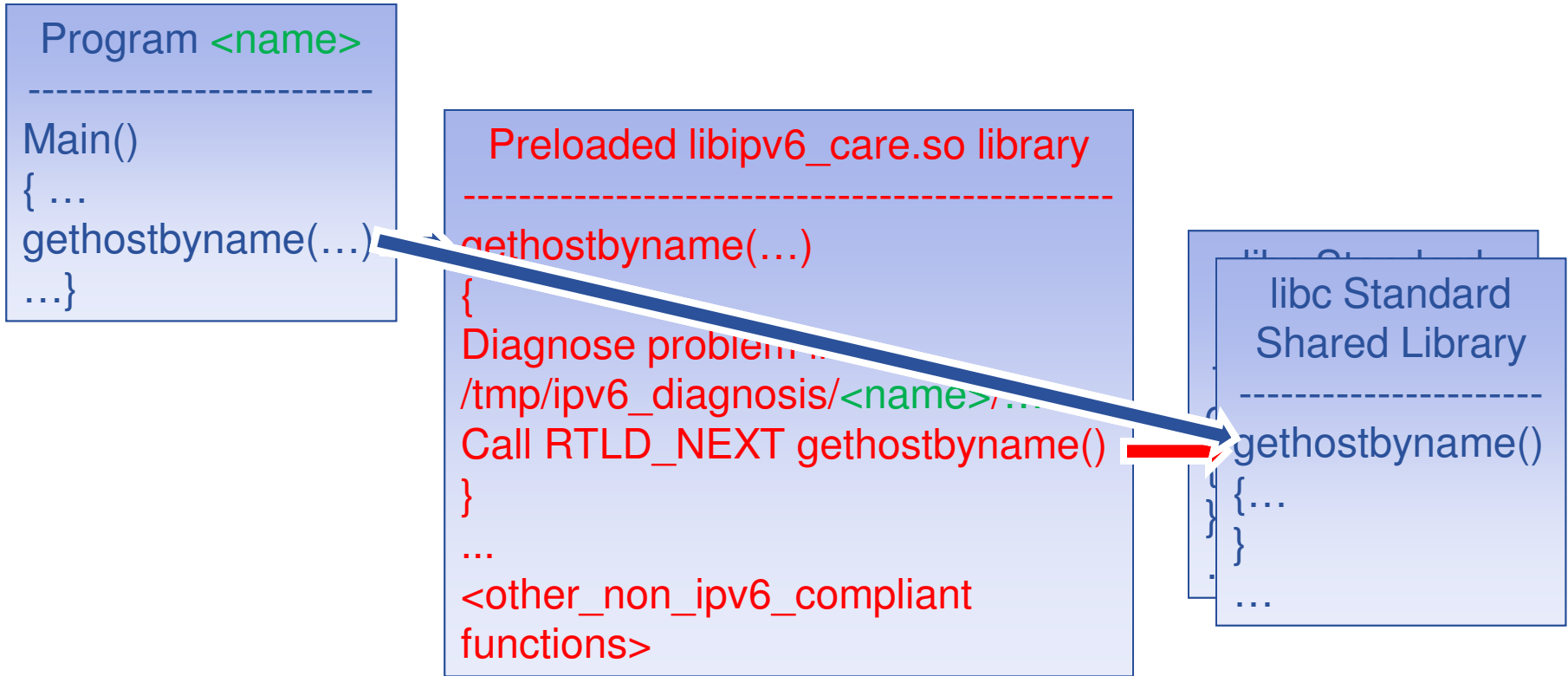
```

$ ipv6_care check -v telnet localhost 9876
IPV6 CARE detected: inet_addr() with [ cp=localhost ]
IPV6 CARE detected: gethostbyname() with [ name=localhost ]
IPV6 CARE detected: inet_ntoa() with [ in=127.0.0.1 ]
Trying 127.0.0.1...
IPV6 CARE detected: socket() with [ domain=AF_INET type=SOCK_STREAM protocol=ip ]
IPV6 CARE detected: connect() with [ socket=3 address.ip=127.0.0.1
address.port=9876 ]
telnet: Unable to connect to remote host: Connection refused
-----
IPV6 diagnosis for 'telnet localhost 9876' was generated in:
/tmp/ipv6_diagnosis/telnet/by_pid/pid_6541
-----
$

```

- The standard behavior of ‘telnet’ is not affected (it works as it should, see its output in black).
- The diagnosis (network function calls log file, non-IPv6 compliant calls detected, etc.) can be reviewed in the mentioned directory.

LD_PRELOAD=/path/to/libipv6_care.so



- Version 3.0 is being developed. This version will have two modes:
 - Check-mode: generate warnings (= current feature)
 - Patch-mode: ***dynamically correct the behavior of the running program*** in order to make it IPv6 compliant.
- There should also be a way to enable the patch-mode on all the programs started on the system (using `/etc/ld.so.preload`).
Ultimate goal: ***make glite nodes IPv6 compliant***
- Current prototyping tests of the patching mode show encouraging results (it works with simple programs like “nc”).

- **EGEE 08 conference, Istanbul**
 - IPv6 session targeted at gLite developers and testers
- **JRA1 / SA3 All-hands meeting, Prague**
 - “IPv6 Testing” session
 - IPv6 basics needed by developers & testers
 - IPv6 methods & tools for testing
 - “IPv6 Programming” session (Tutorial C/C++, Perl, Python, Java)
- **EGEE UF / OGF, Catania**
 - IPv6 session
 - Main work done / to be done about IPv6 in gLite and other middlewares
 - Point of view of the European Commission and of RIPE about IPv6
 - Posters “gLite IPv6 compliance”, “IPv6 compliance testing: dedicated tools”
 - An explanatory [video](#)
- **SA2 All Hands Meeting, Rome**
 - IPv6 session

- **EGEE UF / OGF, Catania demonstration**
 - Demonstration of the current IPv6 features of gLite (see slide “Main IPv6 features of gLite”)
 - Running on the two dual-stack sites (Rome and Paris)
- **This demonstration has been updated and is presented again at EGEE 09**
- **It is also available online at**
<https://twiki.cern.ch/twiki/bin/view/EGEE/OnlineDEMOS>

Current status and foreseen evolution for gLite and IPv6

- **gLite is working properly in a dual-stack environment** (i.e. when IPv6 is enabled on nodes)
 - **This is a major fact since it ensures a smooth transition to IPv6**

- **Some services are already IPv6 compliant**
 - The new (python) version of the BDII (Information System) is able to use IPv6
 - DPM & LFC (Storage Element) can be used over IPv6
 - gridFTP is IPv6 compliant
 - CREAM, WMS-server, WMproxy, blah have been ported (but they are not yet included in the production version of gLite)

IPv6 compliance

- Full IPv6 compliance – for the production version

LFC DPM globus-url-copy/gridFTP BDII_(python)

- Full IPv6 compliance – for a prototype version

CREAM*

- IPv6 compliance to be tested/verified by SA2 – gLite part of the deployment module claimed to be IPv6 compliant

WMS-server WMproxy/Job submission blah

- IPv6 porting currently on-going

gfal lcgutils VOMS LB FTS

- IPv6 porting plan exist

Condor_{utils}

- Currently no known porting plans – also includes non-gLite packages

PX VObox MON dCache Torque C/S MPIutils

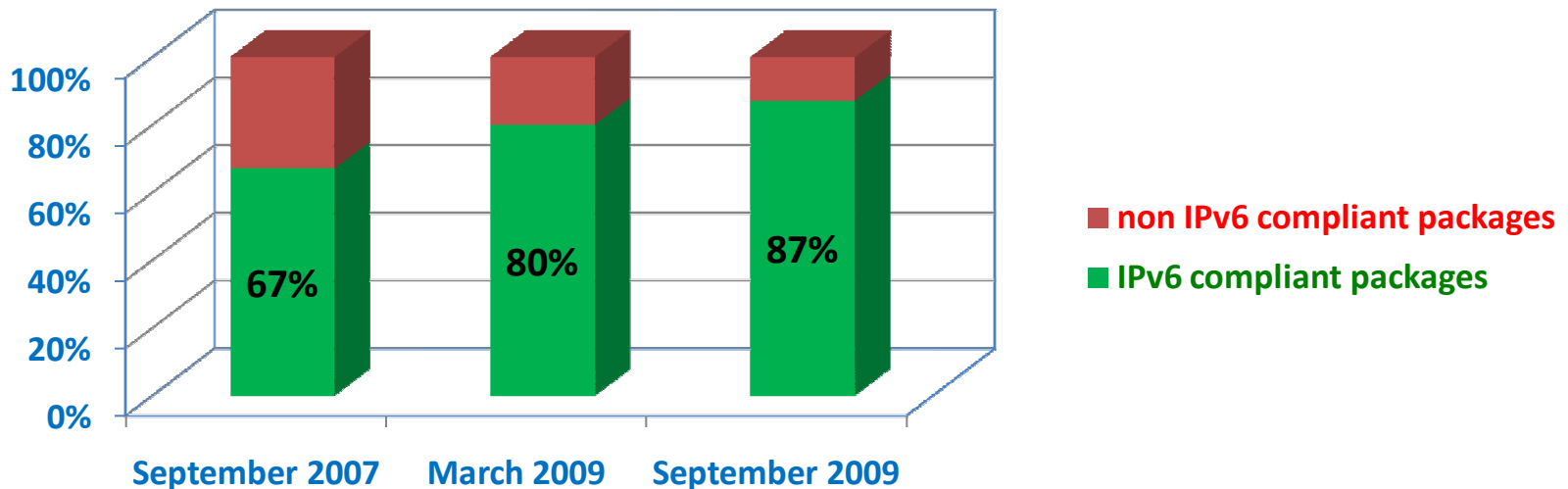
AMGA

* = gLite part of the component is IPv6 compliant / still pb with external dependency

- **Components which should be ported to IPv6 in the next months:**
 - VOMS
 - LB
 - LCG utils
 - GFAL
 - FTS
 - GridSite
- **An issue is what will happen to some relevant external component**
 - i.e. Condor, MySql

* = gLite part of the component is IPv6 compliant / still pb with external dependency

- Based on the results of the static code checker
 - Detection of non-IPv6 compliant patterns in the source code
 - This method is **not fully accurate, but gives a good estimate**
- Here is the evolution obtained on the gLite repository of **ETICS**:



- The value of september 2009 has been calculated by a **new version** of the **IPv6 static code checker**
 - Some false positives were removed (non compliance in code commented)

- 113** IPv6-related bugs in total: *78 open, 35 closed*

– None:	18	113	Entry point for bug lifecycle
– Accepted:	2	0	
– In Progress:	22	0	
– Integration Candidate:	10	0	
– Ready For Review:	10	0	Next ones in the SA2 streamline
– Ready For Test:	4	0	
– Fix not Certified:	11	0	
– Remind:	7	0	
– Verified:	2	0	
– Won't Fix:	21	0	kicked out of bug fixing process
– <i>Invalid:</i>	<i>5</i>	0	
		1 year ago	

Next steps

- **Main foreseen tasks**

- Test IPV6 compliance of ported nodes:
 - WMS/WMproxy currently on-going
- Collaboration with SA3 for the integration of IPv6 in the certification process
- Document our work in an EGEE milestone

- **Optional tasks:**

- Collaboration with the team responsible for the deployment tools, in order to have IPv6 enabled on nodes at installation time
- New complete overall analysis of the gLite source code and update of IPv6 non compliance bugs – in a few weeks from now.
- IPv6 CARE version 3.0
- Automation of the execution on IPv6 CARE within the ETICS test system
- Support to JRA1 and SA3 for anything needed

Thank You.

<https://twiki.cern.ch/twiki/bin/view/EGEE/IPv6FollowUp>
(or google “SA2 IPv6 follow up”)