gLite IPv6 compliance *task TSA2.3.3*

Overall end-of-2009 status
and roadmap until project end

v1.2

Mario Reale-GARR/EGEE SA2
mario.reale@garr.it

Etienne Dublé, Xavier Jeannin - CNRS/UREC EGEE SA2
etienne.duble@urec.cnrs.fr, xavier.jeannin@urec.cnrs.fr

Tuesday, December 15, 2009 – CERN – JRA1/SA3 All Hands

www.eu-egee.org

content

- **I. Current (12/2009) gLite IPv6 status**
  - Available IPv6 deployment modules/services
    - Next ones to be expected in the streamline
  - Overall status of IPv6 compliance in the source code
    - IPv6 bugs summary
- **II.Current set of SA2 provided tools and documents**
  - SA2 testbed
  - Static source code checker (ETICS IPv6 metric plugin)
  - IPV6 CARE (Dynamic IPv6 Checker / Patch Mode)
  - IPv6 programming guides
- **III.How to move on and what to achieve before the end of EGEE III**
  - IPv6 Certification
  - Milestone MSA2.5
  - EGEE Final Review

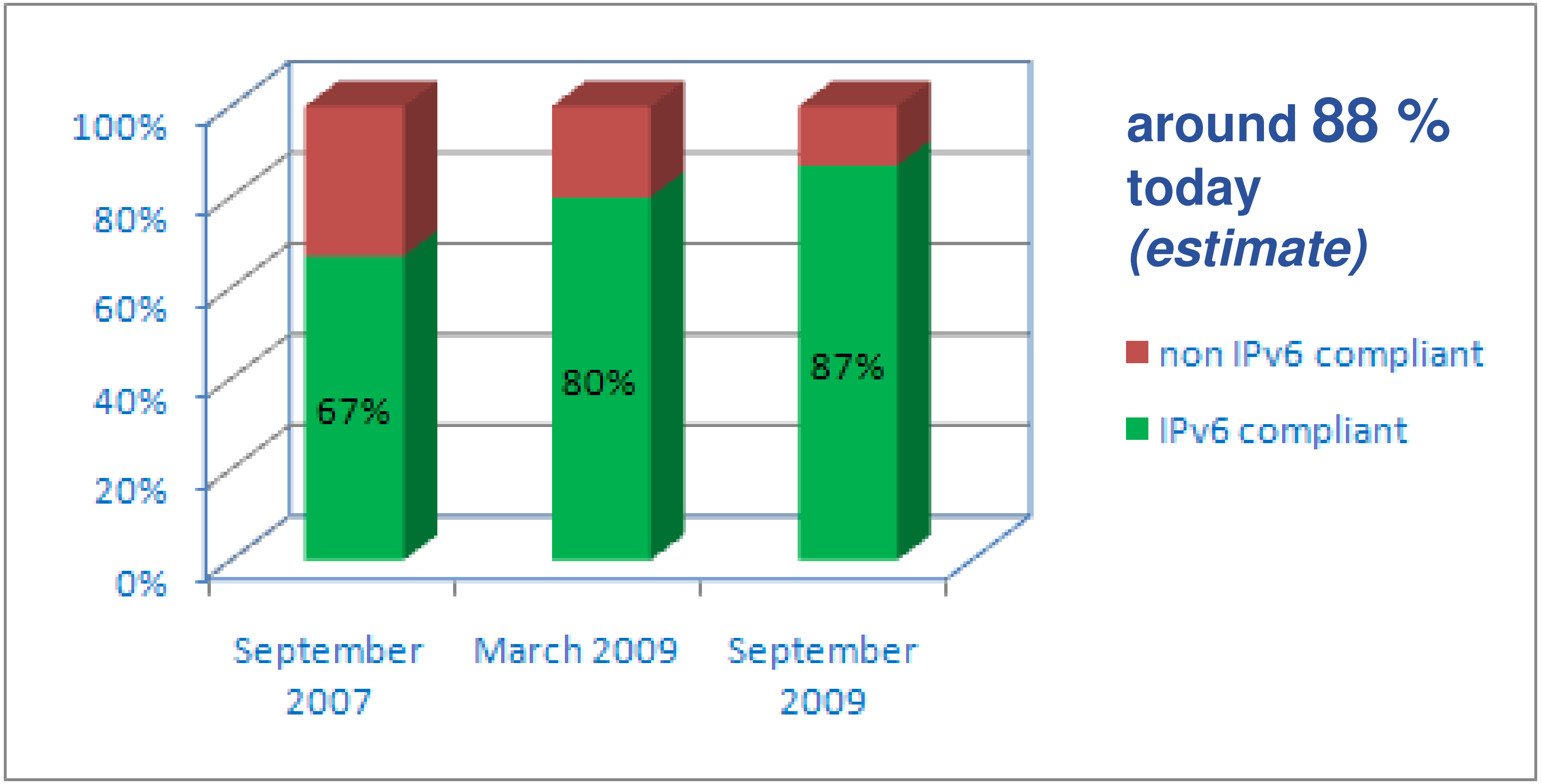
I. Current IPv6 status of gLite

After all…

- …we feel we can reach this goal:

A simple question

- **How far are we from having full IPv6 compliance in all gLite code ?**



around 88 % today *(estimate)*

(# compliant components / total # comp.)

IPv6 status of gLite as of 12/09

- **Already IPv6 compliant deployment modules/components:**
  - LFC, DPM-SE, BDII, WMS/WMSproxy, CREAM CE, FTS, BLAH
- **Work in progress (on-going porting):**
  - VOMS, LB , GridSite
- ***IPv6 Concerning (non compliant or unknown)*** **modules,components and tools:**
  - AMGA, e2emon, DGAS, LRMS batch managers like PBS/Torque, LSF, SGE (and their utils), Condor , lcg_utils , GFAL
  - YAIM
    - Able to deal with IPv6 address notation ?
    - untested
  - An IPv6-reachable gLite repository still missing
  - yum not completely straightforward over IPv6 ?
    - https://bugzilla.redhat.com/show_bug.cgi?id=539563

Current stand on gLite deploym.modules and IPv6

Full IPv6 compliance for the production version

- LFC
- DPM
- globus-url-copy/gridFTP
- BDII
- WMproxy/WMS
- CREAM
- CEmon

- FTS
- blah

TESTED BY
SA2

IPv6 porting currently on-going

- VOMS
- GridSite
- LB

IPv6 compliance unknown (but not aware of clear IPv6 non compliance)

- GFAL
- PX
- Hydra
- PBSutils
- dCache
- MON
- glexec
- VObox
- MPIutils
- Condorutils
- SGEutils
- Torqueutils
- SCAS
- WN

IPv6 compliance still an issue / porting not started

- DGAS
- lcg_utils
- AMGA

gLite deployment modules tested using IPv6

- **DPM Storage Element**
- **LFC File Catalog**
- **CREAM Computing Element / CEmon**
- **BDII**
- **globus-url-copy / GridFTP**
- **WMS/WMproxy**
  - LB pending
  - Full job submission chain still pending
  - Report in preparation

IPv6 bugs

- **115 bugs related to non IPv6 compliant function calls or data structures currently in savannah**
- **Many of them have been processed and corrected for at the source code level**
- **A summary: 62 Closed (more than ½ ☺) , 53 Open**
  - **Closed:**
    - **Verified and Closed: 34**
    - { Won’t Fix / Invalid / Duplicate } and Closed: { 21, 5, 2 }
  - **Open:**
    - **Remind: 24** *(e2emonit, jp, voms, dgas, amga, [globus])*
    - In progress: 22
    - Accepted: 1
    - Integration Candidate: 3
    - Fix not Certified: 2
    - Ready for Test: 1

Not a single bug has not been dealt with

→ No bugs in “None” ☺

Nightly Builds : IPv6 metric for gLite 3.2.0 dev: *glite_branch_3_2_0_dev*

- **The IPv6 source code checker (ETICS IPv6 metric) reports:**
  - **181** Successful Modules
  - **28** Failing Modules

gSOAP needs to be built with the WITH_IPV6 option

- **The 28 failing modules are:** ***(failure reason should be analyzed)***

  LCG-DM, SAGA.lsu-cpp.engine, lcg-mon-gridftp, amga.cli, apel.core, authz.pap-service, authz.pdp-pep-common, data.dpm-httpd-cgi, data.transfer-scripts, jobid.apic, lb.client, lb.logger, lb.server, lbjp-common.server-bones, security.gsoap-plugin, security.gss, security.lcmaps-plugins-scas-client, security.saml2-xacml2-c-lib, security.scas, security.voms, security.voms-admin-server, security.voms-api-java, voms.voms-testsuite-run, wms-ui.api-python, wms-utils.jobid, wms.wmproxy, wms.wmproxy-api-cpp, org.gridsite.core

II. Summary of SA2 provided tools and documents to deal with gLite and IPv6

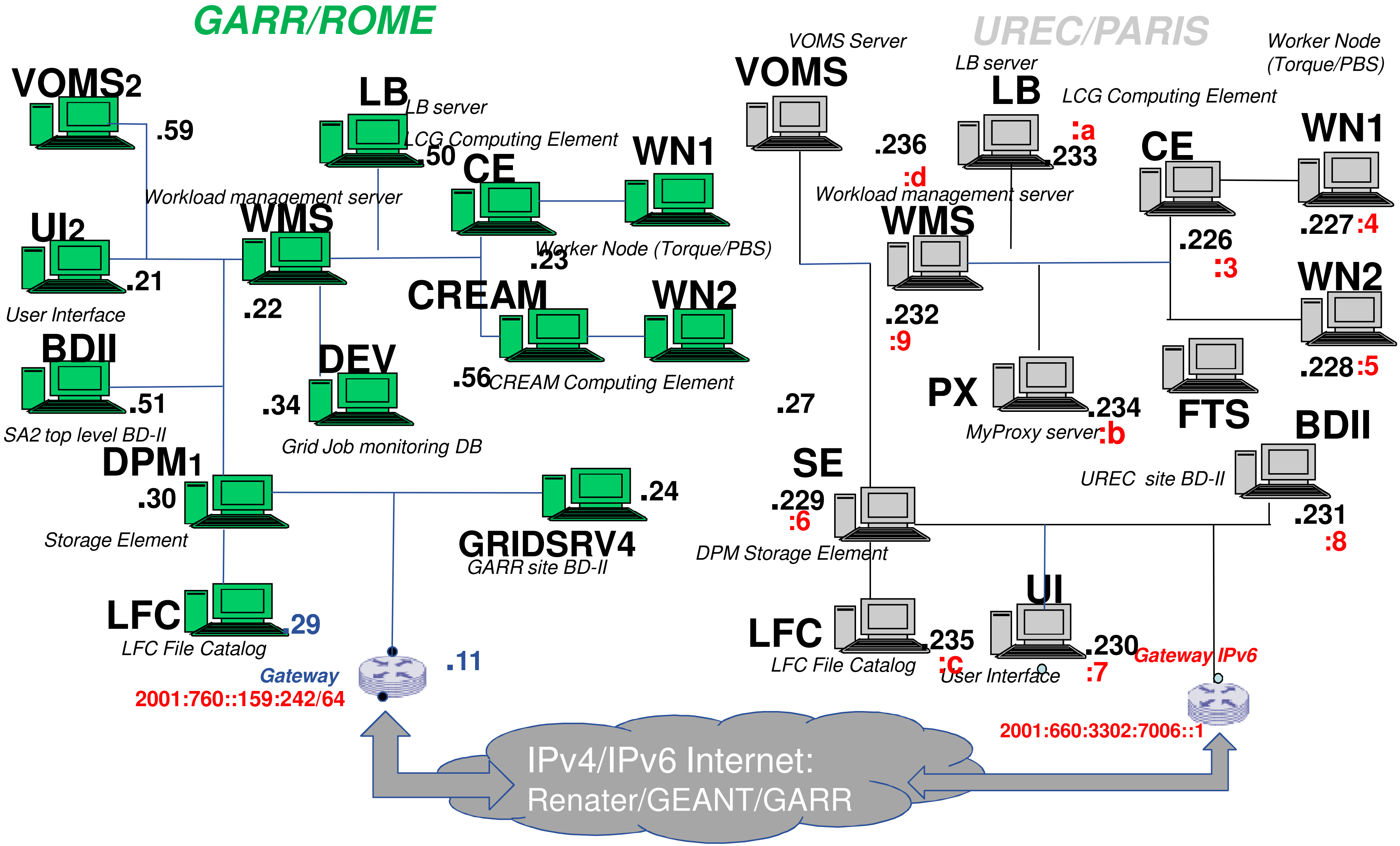
What SA provides around IPv6

- Guides for
  - IPv6 programming in C/C++, Java, Perl, Python
  - Test the IPv6 compliance of a socket server
- A general IPv6 introduction tutorial including exercises
- A distributed IPv6 capable testbed, including NATPT (protocol translator) at GARR(Rome) and UREC(Paris)
- IPv6 resources included in
  - The SA3 certification testbed
  - The ETICS metronome pool
- Both a static (source code) and a dynamic IPv6 checker
  - IPv6 metric of ETICS
  - IPv6 CARE Framework
- A set of specific IPv6 compliance test reports for
  - Selected external components
  - gLite deployment modules and their services
- An ETICS test project on IPv6 (ETICS provided):gLite_ipv6

SA2 provided documents

- **Reference documents on IPv6 for gLite developers: (all on SA2 EDMS or Wiki page)**
  - IPv6 Programming methods:
    - Guide to IPv6 compliant programming in C/C++, Java, Python and Perl:
      - *Provides a sample TCP client and server for each programming language*
      - *Explains advantages/drawbacks/limitations of each language regarding IPv6*
  - IPv6 Testing methods:
    - How to make sure the IPv6 behavior of your application is as expected
  - IPv6 Tests reports:
    - Assessment of the current status of the gLite external packages overall
    - 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,CREAM
- **Provisioning of specific IPv6 introductory tutorials for gLite developers**

SA2 gLite IPv6 testbeds



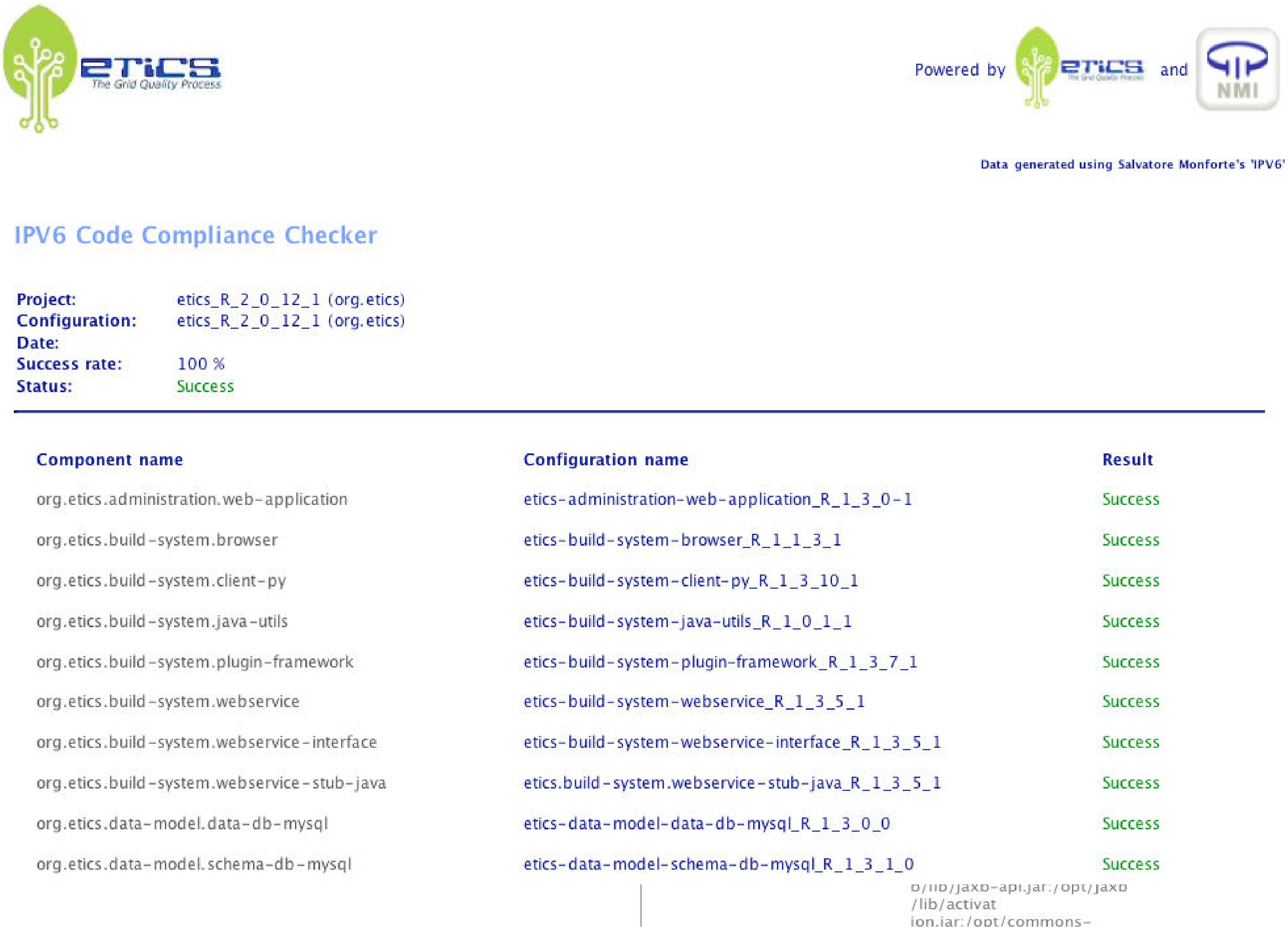
The IPv6 static code checker

- **What is it?**
  - A bash script seeking for evident non IPv6 compliant patterns in the source code
    - Available from http://ui2-4.dir.garr.it/GRID/ipv6_metric.tar.gz
- **How to use it?**
  - Using ETICS build system:
    - You can check the IPv6 metric on the ETICS UI (see next slides)
    - You can submit an IPv6 check job, for example on the org.glite.data.transfer-fts gLite component:

```
etics-submit build -p ipv6check="True“ \
          org.glite.data.transfer-fts
```

  - Optionally the code checker can also be used by hand

IPv6 code cheker usage example



Data generated using Salvatore Monforte's 'IPV6'

IPV6 Code Compliance Checker

Project:	etics_R_2_0_12_1 (org.etics)
Configuration:	etics_R_2_0_12_1 (org.etics)
Date:	
Success rate:	100 %
Status:	Success

Component name	Configuration name	Result
org.etics.administration.web-application	etics-administration-web-application_R_1_3_0-1	Success
org.etics.build-system.browser	etics-build-system-browser_R_1_1_3_1	Success
org.etics.build-system.client-py	etics-build-system-client-py_R_1_3_10_1	Success
org.etics.build-system.java-utils	etics-build-system-java-utils_R_1_0_1_1	Success
org.etics.build-system.plugin-framework	etics-build-system-plugin-framework_R_1_3_7_1	Success
org.etics.build-system.webservice	etics-build-system-webservice_R_1_3_5_1	Success
org.etics.build-system.webservice-interface	etics-build-system-webservice-interface_R_1_3_5_1	Success
org.etics.build-system.webservice-stub-java	etics.build-system.webservice-stub-java_R_1_3_5_1	Success
org.etics.data-model.data-db-mysql	etics-data-model-data-db-mysql_R_1_3_0_0	Success
org.etics.data-model.schema-db-mysql	etics-data-model-schema-db-mysql_R_1_3_1_0	Success

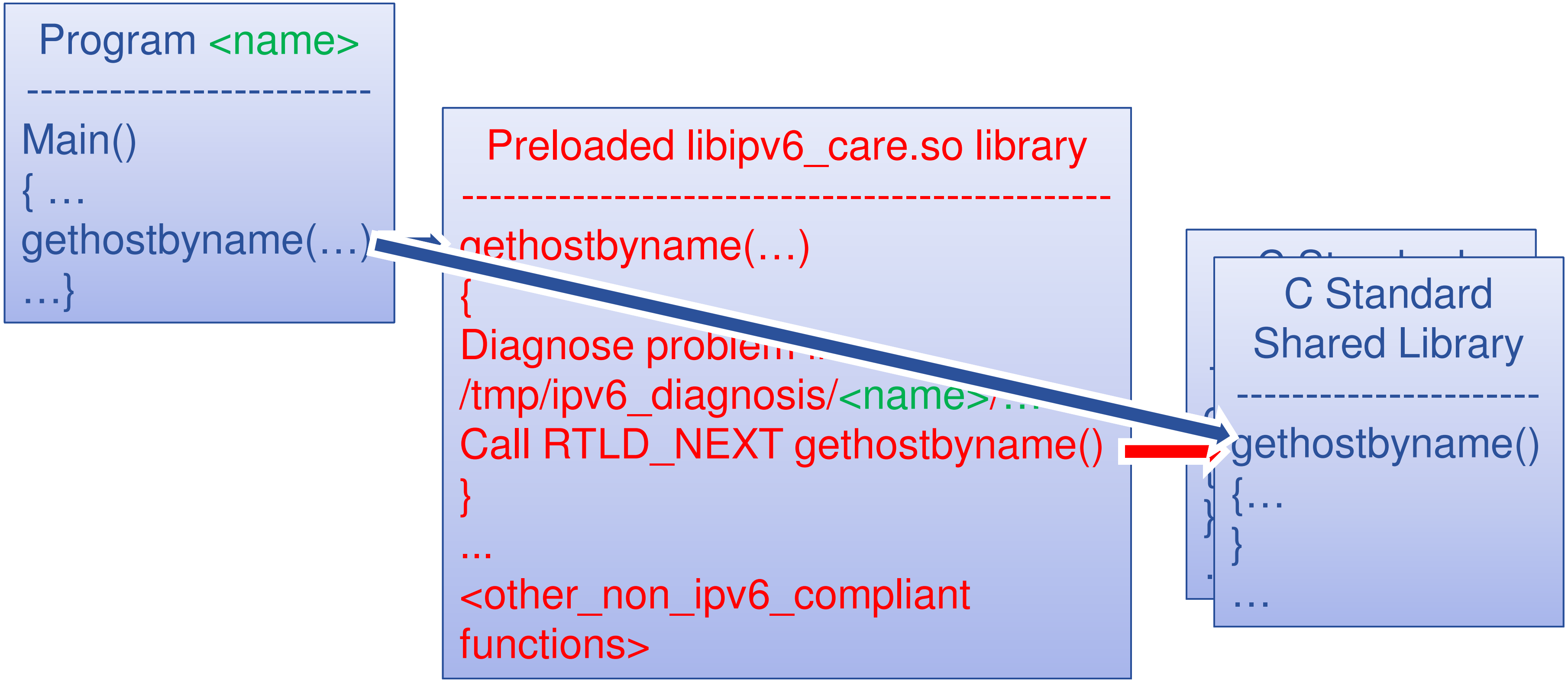
b/lib/jaxb-api.jar:/opt/jaxb
/lib/activat
ion.jar:/opt/commons-

IPv6 CARE

- **Linux toolbox about IPv6 compliance of applications**
  - ✓ « Checking » mode: diagnose IPv6 compliance of an application
    - ➢ Stable feature
  - ✓ « Patching » mode: correct non-IPv6 compliant behavior of an application on-the-fly, in order to make it compliant
    - ➢ Feature being tested
- **The tool works by detecting and analyzing / replacing the networking function calls performed by your program**
  => *no need to have the source code of the program being checked / patched*

IPv6 CARE mechanism

LD_PRELOAD=/path/to/libipv6_care.so



Program <name>

Main()
{ …
gethostbyname(…)
…}

Preloaded libipv6_care.so library
----
gethostbyname(…)
{
Diagnose problem in
/tmp/ipv6_diagnosis/<name>/…
Call RTLD_NEXT gethostbyname()
}
...
<other_non_ipv6_compliant
functions>

C Standard
Shared Library
----
gethostbyname()
{…
}
…

Advantages / Drawbacks

- **Advantages:**
  - **It works with all non-static programs**
  - **It does not affect the standard behavior of the program**
  - **It does not warn about parts of code which are actually not executed**

- **Drawbacks:**
  - IPv6 CARE only detects non-IPv6-compliant function calls. There may be other (less common) kinds of non-IPv6 compliance problems which will not be detected.

IPv6 CARE: Checking mode

- **Example: test of an old version of “telnet”**
  - One must prefix the command with “ipv6_care check [-v]”:

```
$ ipv6_care check -v telnet localhost 9876
```

- The output messages allow to diagnose IPv6 compliance
- If needed the whole diagnosis is available in the reported directory

IPv6 CARE: Checking mode

- **Example: test of an old version of “telnet”**
  - One must prefix the command with “ipv6_care check [-v]”:

```
$ 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 output messages allow to diagnose IPv6 compliance
- ➢ If needed the whole diagnosis is available in the reported directory

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
```

IPv6 CARE: Patching mode

- Exemple of mysqld:

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                                [  OK  ]
[root@quarks ~]#
```

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
```

IPv6 CARE: Patching mode

Enabling Grids for E-sciencE

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306     0.0.0.0:*    LISTEN      21591/mysqld
[root@quarks ~]#
```

IPv6 CARE: Patching mode

- Exemple of mysqld:

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                            [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN      21591/mysqld
[root@quarks ~]# /etc/init.d/mysqld stop
```

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                            [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN      21591/mysqld
[root@quarks ~]# /etc/init.d/mysqld stop
Stopping MySQL:                                            [  OK  ]
[root@quarks ~]#
```

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN      21591/mysqld
[root@quarks ~]# /etc/init.d/mysqld stop
Stopping MySQL:                                             [  OK  ]
[root@quarks ~]# ipv6_care patch /etc/init.d/mysqld start
```

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN      21591/mysqld
[root@quarks ~]# /etc/init.d/mysqld stop
Stopping MySQL:                                             [  OK  ]
[root@quarks ~]# ipv6_care patch /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]#
```

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN      21591/mysqld
[root@quarks ~]# /etc/init.d/mysqld stop
Stopping MySQL:                                             [  OK  ]
[root@quarks ~]# ipv6_care patch /etc/init.d/mysqld start
Starting MySQL:                                             [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
```

IPv6 CARE: Patching mode

- **Exemple of mysqld:**

```
[root@quarks ~]# /etc/init.d/mysqld start
Starting MySQL:                                            [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN
21591/mysqld
[root@quarks ~]# /etc/init.d/mysqld stop
Stopping MySQL:                                            [  OK  ]
[root@quarks ~]# ipv6_care patch /etc/init.d/mysqld start
Starting MySQL:                                            [  OK  ]
[root@quarks ~]# netstat -lnpt | grep mysqld
tcp        0      0 0.0.0.0:3306    0.0.0.0:*    LISTEN
21736/mysqld
tcp        0      0 :::3306         :::*         LISTEN
21736/mysqld
[root@quarks ~]#
```

IPv6 CARE: Patching mode

- An option allows to apply the patching-mode to *all processes started on the system*:
  `ipv6_care system patch`
  (This could for example make a whole gLite node IPv6 compliant)
- Patching-mode available from version 3.0 ongoing
- More tests required before releasing a stable 3.0 version
- Alpha version ipv6_care-3.0-alpha1-src.tar.gz available at: http://sourceforge.net/projects/ipv6-care/files/
- Any other info:
  http://sourceforge.net/projects/ipv6-care
  etienne.duble@urec.cnrs.fr

III. How to move on with IPv6 until the end of EGEE III

EGEE IPv6 Milestone: MSA2.5

- **"In this milestone, the IPv6 status in EGEE will be assessed in the middleware. An analysis of the different solutions for the IPv6 deployment over EGEE infrastructure will be given, and if possible demonstrated"**
  - Overall, EGEE has achieved already quite some results w.r.t. IPv6
    - Thanks to SA2, JRA1, SA3, ETICS
  - IPv6 status in gLite already assessed.
  - → We have the choice to demonstrate something with an IPv6 DEMO or not ☺
    - An IPv6 demo has been prepared for OGF25 in Catania showing: (https://twiki.cern.ch/twiki/bin/view/EGEE/OnlineDEMOS)
      - *Dual Stack doesn't affect gLite (specifically the Management)*
      - *BDII,LFC,DPM and globus-url-copy are working in IPv6*
    - this or something more/new might be of interest for the final review ?
      - *Especially if -for instance- the full Job Management will be exploitable*

IPv6 Certification: Discussion

- **We should address the reviewers' comments":**

  "complete the development , testing and certification of IPv6 support"

- **What kind of certification should be set up ?**
  - Limited basic functionality tests for the already available IPv6 compliant components
  - Certify gLite behavior on Dual Stack by setting up certification tests on SA2 IPv6-capable testbed
  - Automatic execution of IPv6-related analysis/tests
    - IPV6CARE
    - ETICS IPv6 metric (source code checker)
  - Should we only certify that the modifications related to IPv6 did not break IPv4? (if yes we have nothing more to do, current tests already do that)
  - Anything else ?

IPv6 Porting

- **JRA1:**
  - What would you need from SA2 ?
  - In what can we specifically help ?
  - What could a possible strategy to face the remaining non-compliance left (residual non compliance) look like in future (EGEE? EMI/UMD ? EGI ?)

The IPv6 legacy for the future

- **All installation and configuration tools**
  - IPv6 repository
  - YAIM
  - Quattor ?
  - PXE ?

- **All dashboard tools / SAM / Grid monitoring tools**
  - What is their level of IPv6 compliance ?

- **Training operation team**
  - Spread knowledge and form people in the operations onIPv6

Thank You.

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