Network Monitoring Using Grid Jobs

v0.6

EGEE SA2

Xavier Jeannin, Etienne Dublé   - CNRS/UREC

Mario Reale, Alfredo Pagano – GARR

Friday, December 11, 2009 – Bologna – LHCOPN Meeting

www.eu-egee.org

Information Society
and Media

- **Network Monitoring for Grids**
- **The EGEE context   (vs. LHCOPN)**
- **The idea:** *using Grid Jobs to monitor the network for the Grid*
- **System design and operation**
- **Current status**
- **Future developments**
- **Short live DEMO**

www.eu-egee.org

Information Society
and Media

- **GRIDs are *big users* and they will exercise the network**
  - **The LHC Computing Grid moving across ~15 PetaBytes of raw data/year for sure will**

- **Grid middleware can benefit from monitoring:**
 - Example: Network aware job scheduling

Execution Time vs Number of Jobs
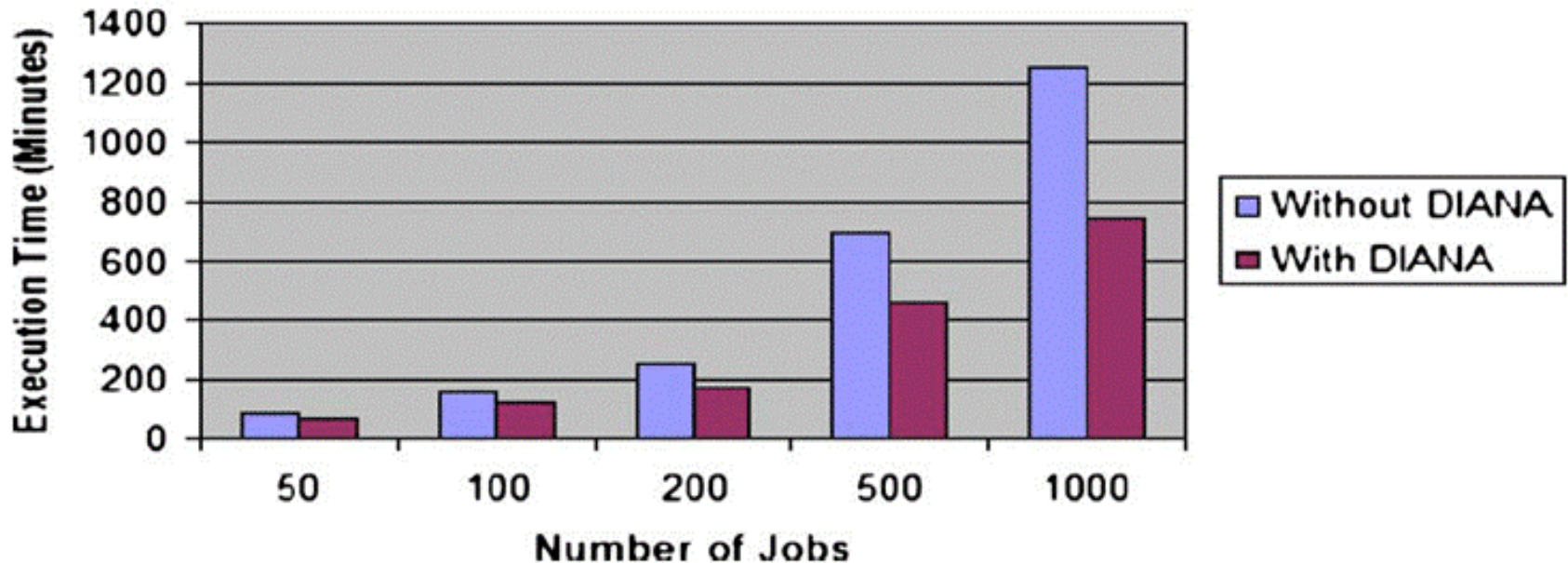
- **e2emonit (pingER, UDPmon, IPERF)**
- **NPM (Network Performance Monitor)**
  - PCP (Probe Control Protocol)
- **Diagnostic Tool**
- **PerfSONAR_Lite-TSS**
- **PerfSONAR-MDM**

- **Things you probably already know... Zzzzzzzzz ;-)**

- **The Current EGEE/W-LCG infrastructure is made up by 260+ sites**
  - 1 Tier 0 at CERN (DAQ)
  - 11 Tier 1 (online to the DAQ at CERN, data reconstruction)
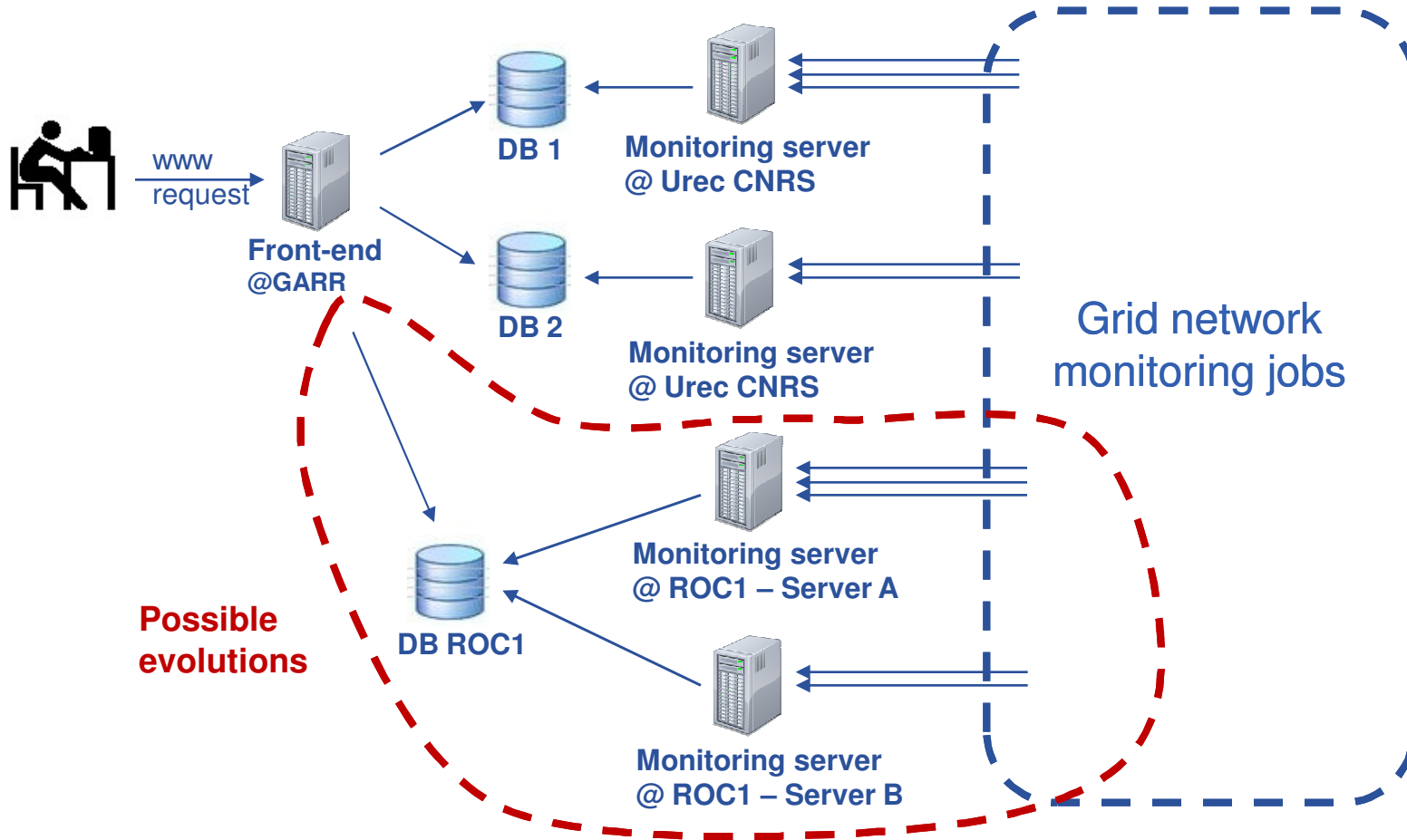 - **130+** Tier 2 (end users' analysis and simulation)
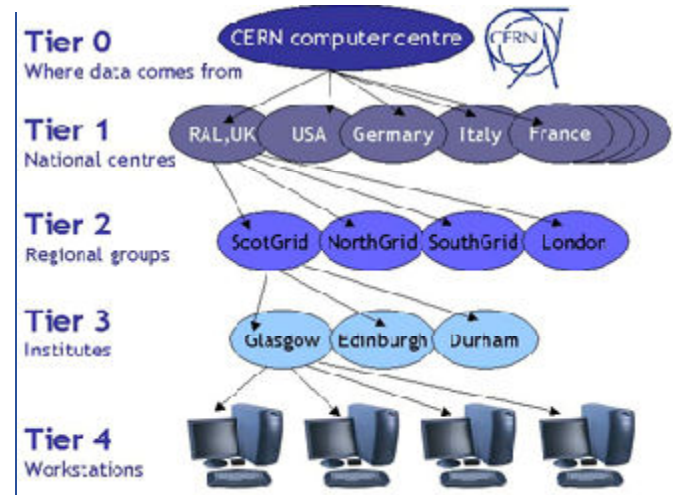
- **T0-T1s interconnected by the LCHOPN**

- **Monitoring appliance for T0/T1 already in place**
  - PerfSONAR MDM release for the LHCOPN
  - Dedicated deployment at sites (MoU, dedicated servers)
  - On-going – #(10) sites (11)

Enabling Grids for E-sciencE

- **No EGEE recommended general solution for network monitoring**

- **Seeking the general, light weighted solution for #(100) grid sites, SA2 started a task which**
  - Assessed the situation
  - Identified relevant metrics
  - Proposed an approach and prototyped it (that's what we are about to present here today ☺)

- **Addressing a complimentary approach with respect to**
  - The LHCOPN T0-T1 approach
  - The troubleshooting on demand tools proposed with PerfSONAR-Lite-TSS

Enabling Grids for E-sciencE

- **The approach we propose here today tries to take into account:**
  - High scalability
  - Light weighted deployment
  - Non-intrusiveness
  - Security
  - Reliability
  - Cost-effective

- **The tool currently implements active monitoring**

egee

Enabling Grids for E-sciencE

"Instead of installing a probe at each site, run a grid job"

eGee

www request

Front-end @GARR

DB 1

Monitoring server @ Urec CNRS

DB 2

Monitoring server @ Urec CNRS

Grid network monitoring jobs

Possible evolutions

DB ROC1

Monitoring server @ ROC1 – Server A

Monitoring server @ ROC1 – Server B

Frontend: Apache Tomcat, Ajax, Google Web Toolkit (GWT)

Backend: PostgreSQL

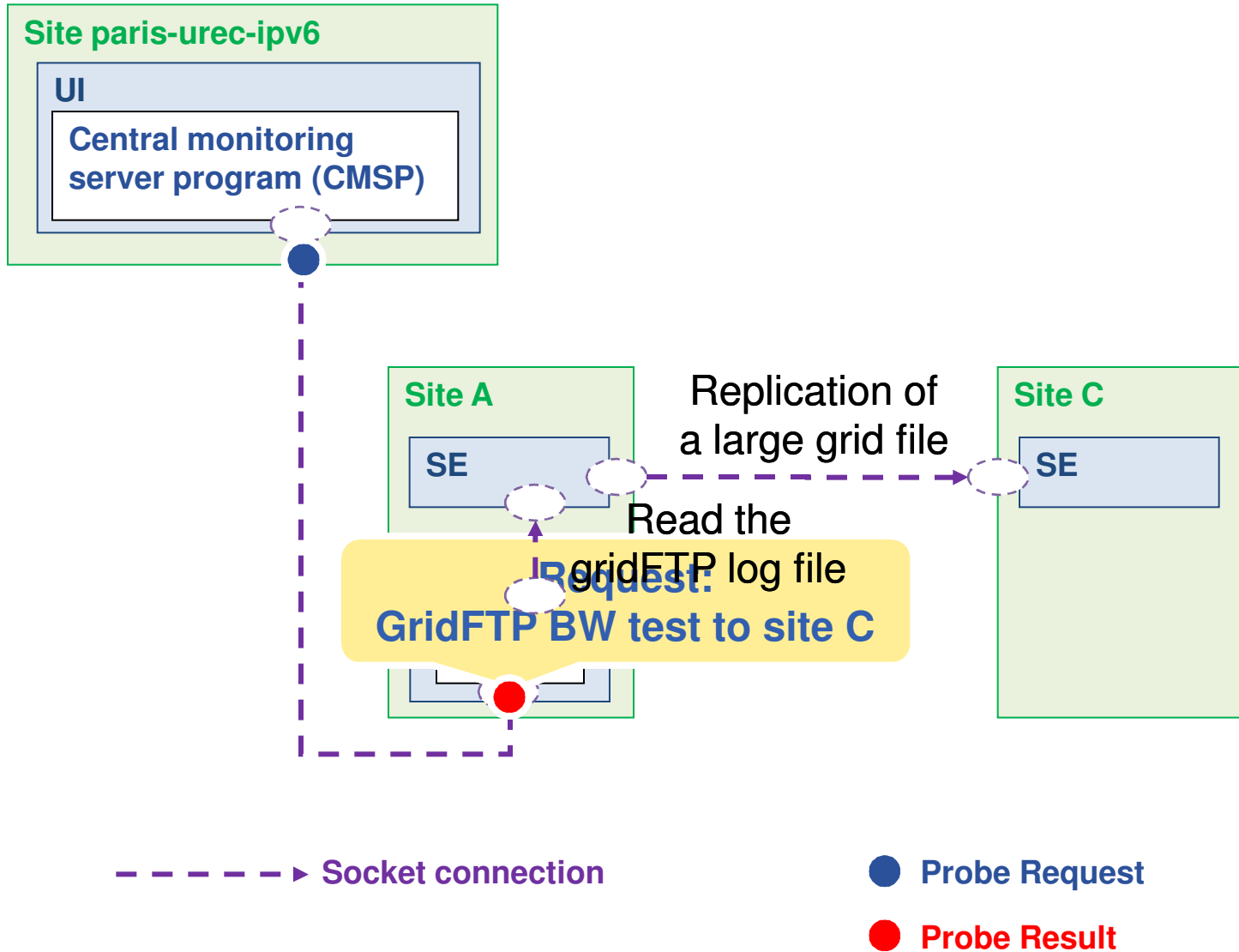Programming languages for server and jobs: Python, bash script

- **Technical constraints to be dealt with:**
  - When running a job, the grid user is mapped to a Linux user of the Worker Node (WN):
    - This means the job is **not running as root** on the WN
    - ➢ Some low level operations are not possible
      (for example **opening an ICMP listening socket** is **not allowed**)
  - Heterogeneity of the WN environments
    (various OS, 32/64 bits…)
    - Ex: making the job download and run an external tool may be tricky
      (except if it is written in an OS independent programming language)
  - The system has to deal with the grid mechanism overhead
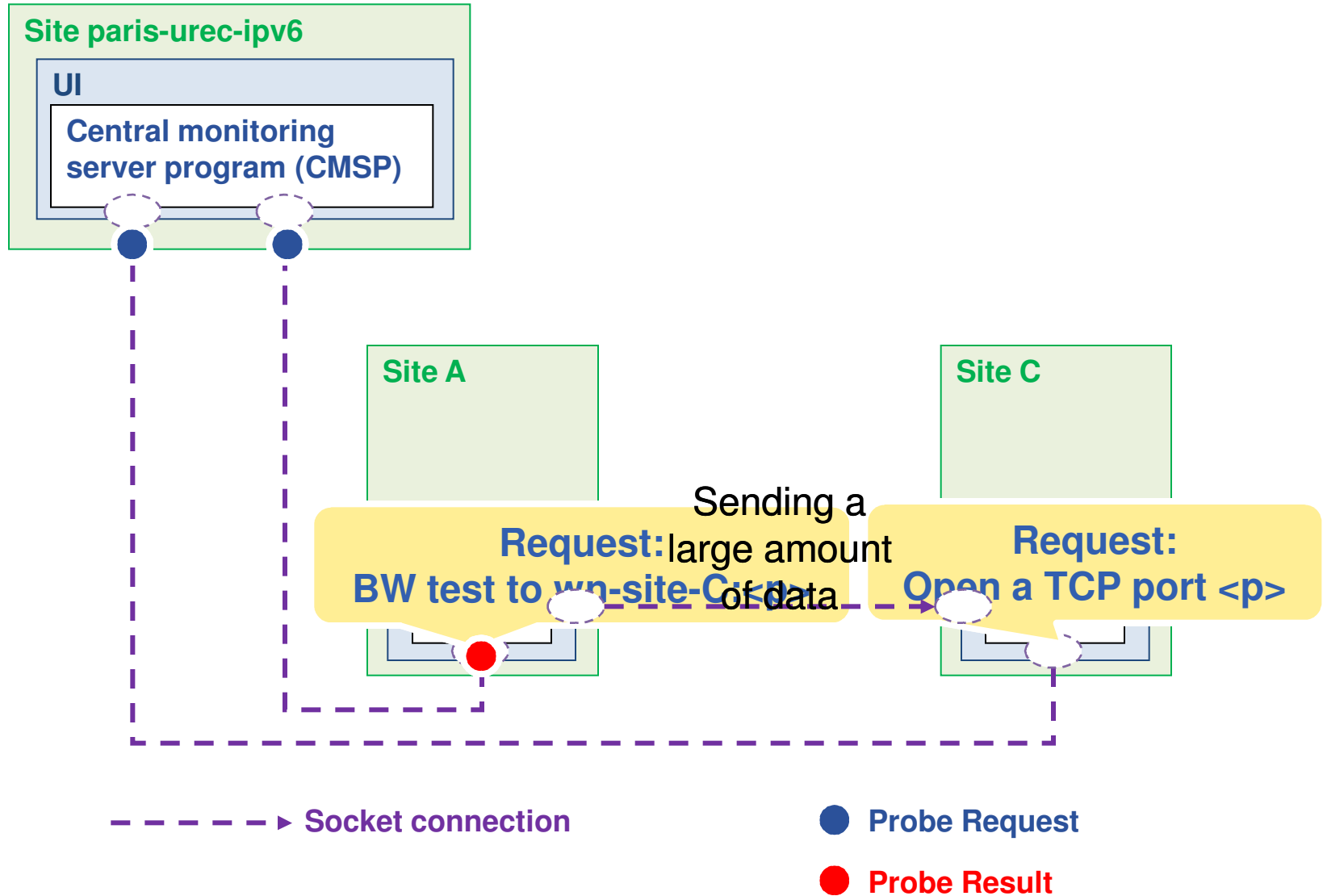   (delays…)
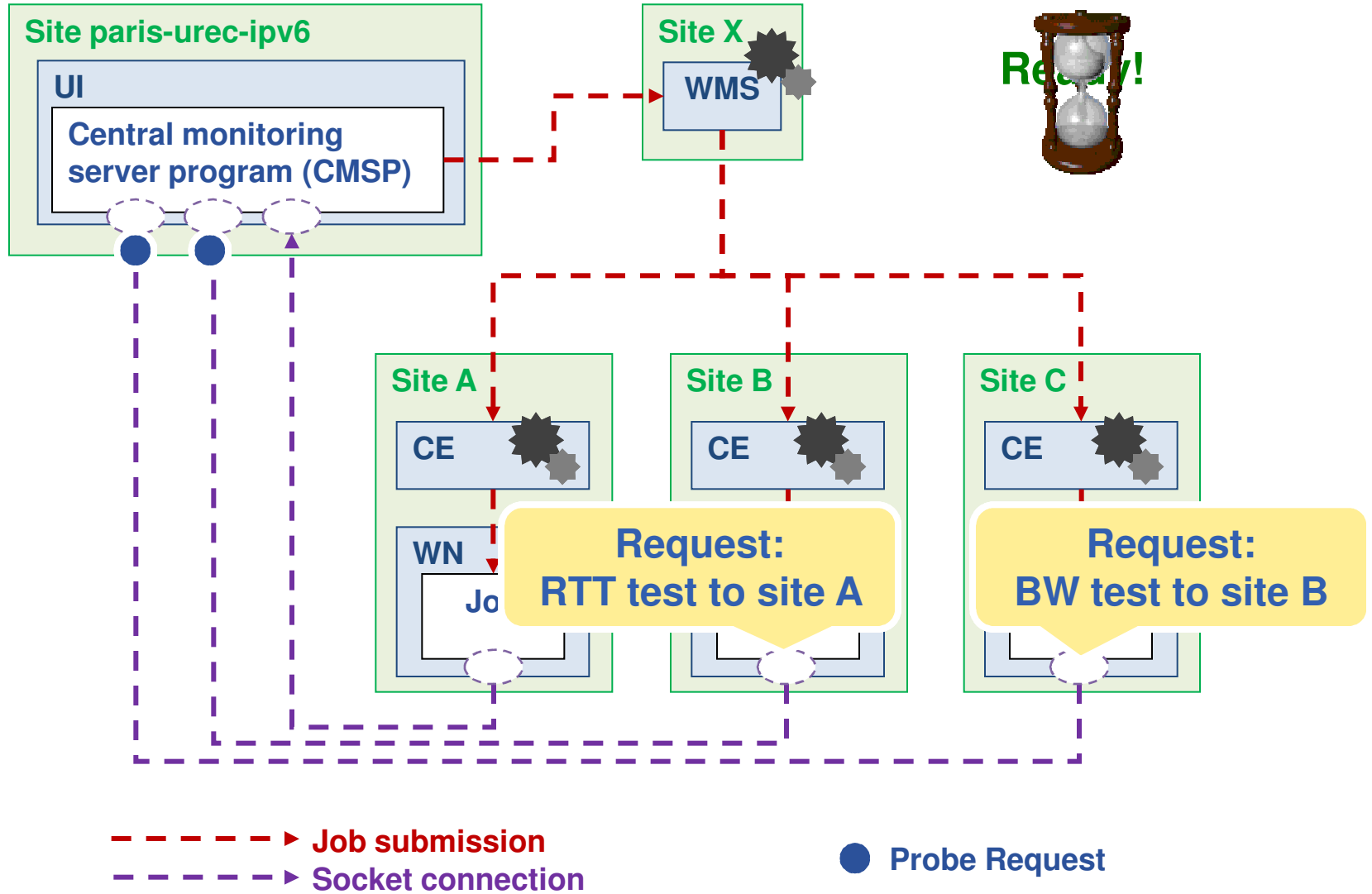
- **Chosen design is more efficient than starting a job for each probe → 1 jobs::more probes**
  - Considering delays
  - Considering the handling of middleware failures (the majority of failures occur at job submission, not once the job is running)
- **TCP connection is initiated by the job**
  - ➢ No open port needed on the WN → better for security of sites
- **An authentication mechanism is implemented between the job and the server**
- **High scalability**
- **A job cannot last forever (GlueCEPolicyMaxWallClockTime)**
  - ➢ So actually there are two jobs running at each site
    - A 'main' one, and
    - A 'redundant' one which is waiting and will become 'main' when the other one ends

Enabling Grids for E-sciencE
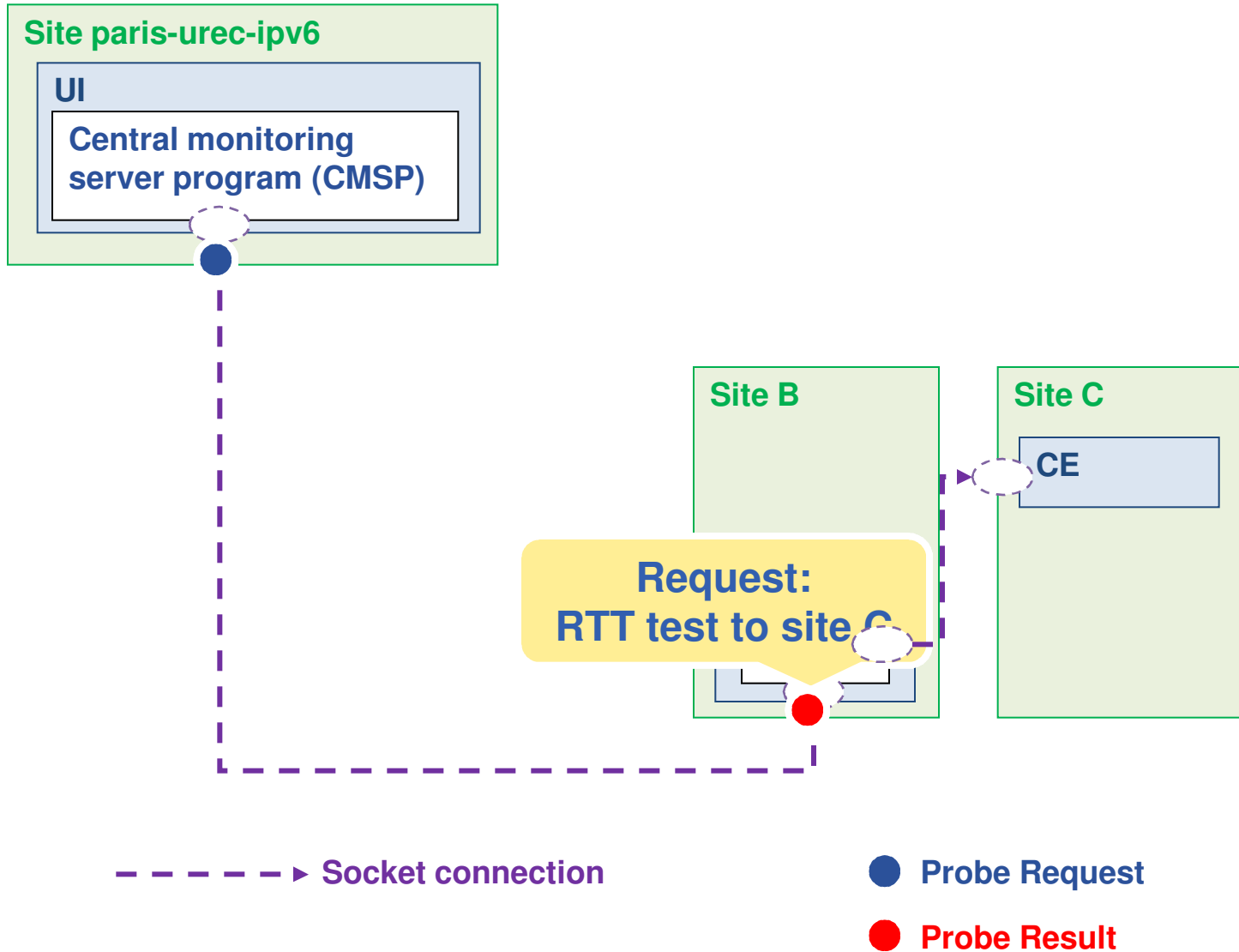
- The 'RTT' measure is the time a TCP 'connect()' function call takes:
    - Because a connect() call involves a round-trip of packets:
        - *SYN* ->
        - *SYN-ACQ* <- } Round trip
        - *ACQ* -> } Just sending => no network delay
    - Results very similar to the ones of 'ping'
- The MTU is given by the IP_MTU socket option
- The number of hops is calculated in an iterative way
- All these measures require:
    - To connect to an accessible port (1) on a machine of the remote site
    - To close the connection (no data is sent)
    - *Note: This (connect/disconnect) is detected in the application log*
- (1): We use the port of the gatekeeper of the CE since it is known to be accessible (it is used by gLite)

egee

Site paris-urec-ipv6

UI

Central monitoring server program (CMSP)

Site A

SE

Replication of a large grid file

Site C

SE

Read the gridFTP log file

Request: GridFTP BW test to site C

- - - - ► **Socket connection**

● **Probe Request**

● **Probe Result**

- **If the GridFTP log file is not accessible (cf. dCache?)**

  – We just do the transfer via globus-url-copy and measure the time it takes

  – This is slightly less precise

  – How many streams should we request in the command line?
    `globus-url-copy -p <num_streams> […]`

    ▪ `Default = 1`

    ▪ `Best streams number seems to be 3 (we are investigating on this)`

Site paris-urec-ipv6

UI

Central monitoring server program (CMSP)

Site A

Site C

Request:
BW test to wn-site-C <p>

Sending a large amount of data

Request:
Open a TCP port <p>

- - - - ► **Socket connection**

● **Probe Request**

● **Probe Result**

- **It requires the remote site to allow incoming TCP connections to the WN**
  - This is not a good thing regarding security
  - Sometimes it is not possible (WNs behind a NAT)
    - Some other derivative solutions could solve this BUT not in all cases
- **There is no real use case of these WN to WN transfers**
  - So what will this measure refer to?
  - The WN network connectivity may not be adapted
- ➤ **We might deprecate this method and use the GridFTP one instead (cf. previous slides)**
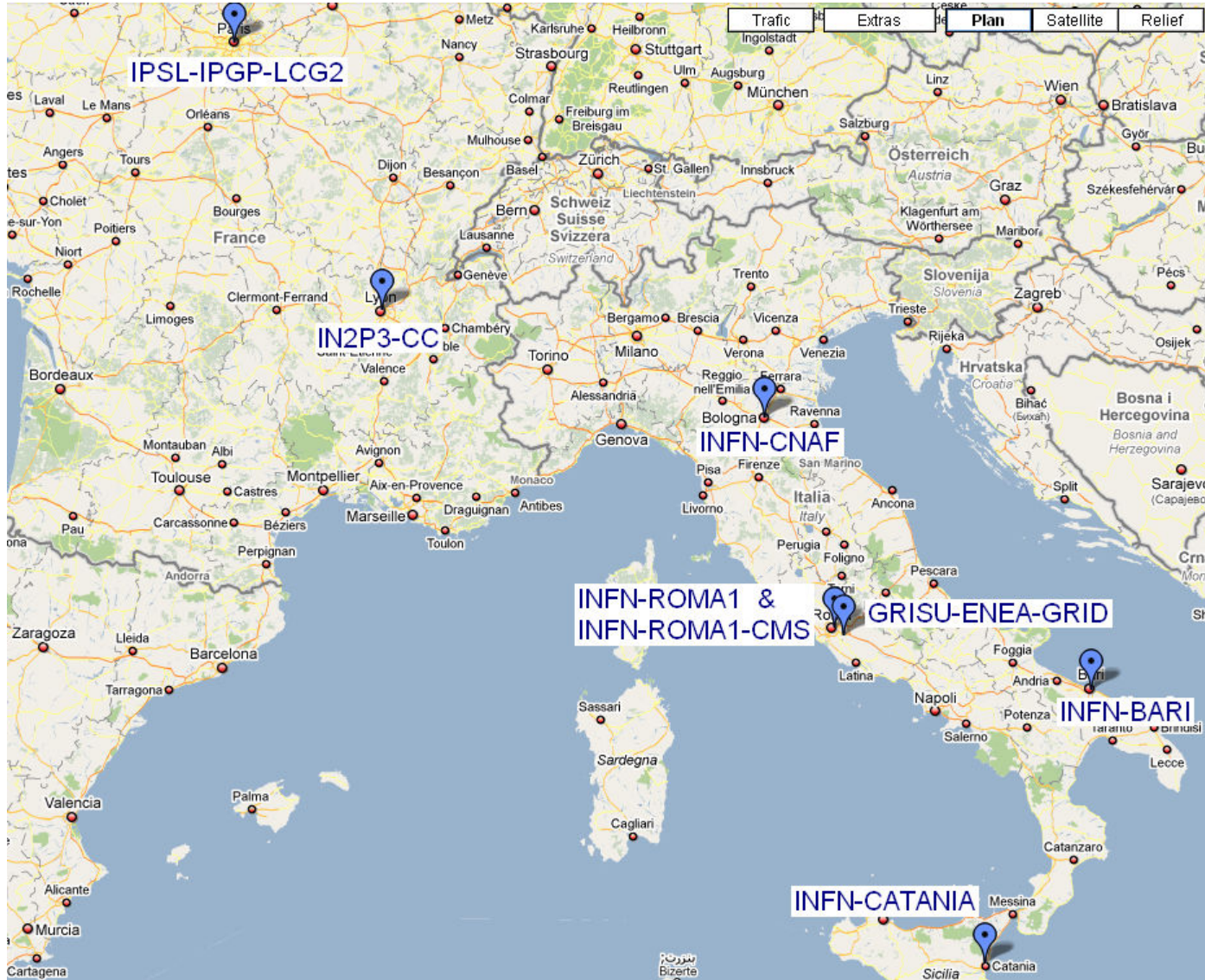
- **Monitor all possible site-to-site paths will be too much: N x (N-1) and N ~ 300 sites for a whole grid coverage**

- **We must restrict the number of these paths**
  - To a specific VO, to an experiment, to the most used paths, etc.
  - We have studied this at https://edms.cern.ch/document/1001777

- **… and/or start several server instances (in order to achieve desired performance)**

- **The system is completely configurable about these paths and the scheduling of measurements**
  - The admin specifies a list of scheduled tests, giving for each one:
    - The source site
    - The remote site
    - The type of test
    - The frequency of the test

- **Latency test**
  - TCP RTT
  - Every 10 minutes

- **Hop count**
  - Iterative connect() test
  - Every 10 minutes

- **MTU size**
  - Socket (IP_MTU socket option)
  - Every 10 minutes

- **Achievable Bandwidth**
  - TCP throughput transfer via GridFTP transfer between 2 Storage Elements
  - Every 8h

In order to avoid too many connections these three measurements are done in the same test

Ldap Authentication, based on Google Web Toolkit (GWT) framework

Enabling Grids for E-sciencE

1. **Triggering system to alert site and network admins**

2. **Frontend improvements**
   (plotting graphs)
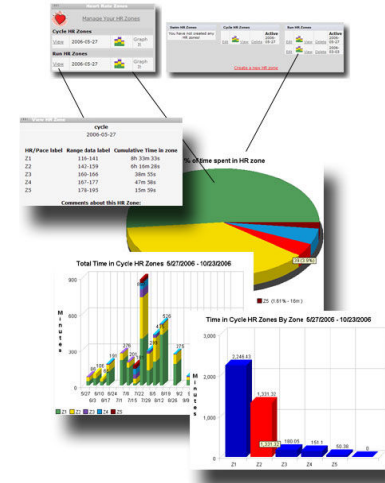
3. **Not only scheduled, but also 'on-demand measurements'**

Enabling Grids for E-sciencE

4. **Improve the server part (mainly regarding gLite failures)**

5. **Add more types of active measurements?**

6. **Add passive measurements?**

7. **Consider adding a dedicated box (VObox?)**

   ➢ If some of the metrics needed are not available with the job-based approach

      ▪ ex: low level measurements requiring root privileges

   ➢ The job would interact with this box and transport the results

   ➢ This might be done in a restricted set of major sites

8. **Consider interaction with other systems (some probes may be already installed at some sites, we could benefit from them)**

Enabling Grids for E-sciencE

- **Show the basic functionality provided by the tool**

- **Added value:**
  - No installation/deployment needed in the sites
    - Monitoring 10 or 300 sites is just a matter of configuration
  - A monitoring system running on a proven architecture (the grid)
  - Possibility to use grid services (ex: AuthN and AuthZ)
- **Limits:**
  - Some low-level metrics can't be implemented in the job itself
  - Comparing to a dedicated system installed in a site, no control of the Worker Node environment (hardware, software) is available

Thank You

Feedback or request?

http://egeemon.dir.garr.it:8080/NetMonDB/
https://twiki.cern.ch/twiki/bin/view/EGEE/GridNetworkMonitoring

Contacts: alfredo.pagano@garr.it
                 etienne.duble@urec.cnrs.fr