

# A Grid Job Monitoring System

Subir Sarkar, INFN Pisa

Contributors

Catalin Dumitrescu, FNAL

Andreas Nowack, RWTH Aachen

Sanjay Padhi, UCSD

## Introduction

The Job Monitoring system is a web based framework that enables users to track jobs in substantial detail in quasi-real time. This is the first such monitoring application for the Grid jobs where an X.509 authenticated web interface can be seamlessly accessed by both end-users and site administrators. A site administrator has access to all the information, whereas the visibility of jobs of a Virtual Organization (VO) to users depends on the privacy setting of the site.

The framework supports LSF, Condor and PBS like batch systems through plug-ins and can be easily extended to integrate other batch systems. The system is being extended to monitor glideinWMS instances also. This broadens the scope significantly, allowing the system to monitor jobs over multiple sites.

## Aim

user centric

- track jobs using global job id
  - summary information, process list
  - CPU and memory usage with time
  - job and working directory listing
  - log and error log
  - worker node status
- select jobs based on
  - RB/WMS
  - computing element
  - submission/start time

administrator oriented

- track jobs using local job id
  - for worker nodes
  - for local queues/users
  - mis-behaving jobs (0 load, expiring Grid proxy etc.)
- use a single, intuitive web interface
- a central page may link single jobs
- respect privacy
  - base access on Grid certificate

## Description

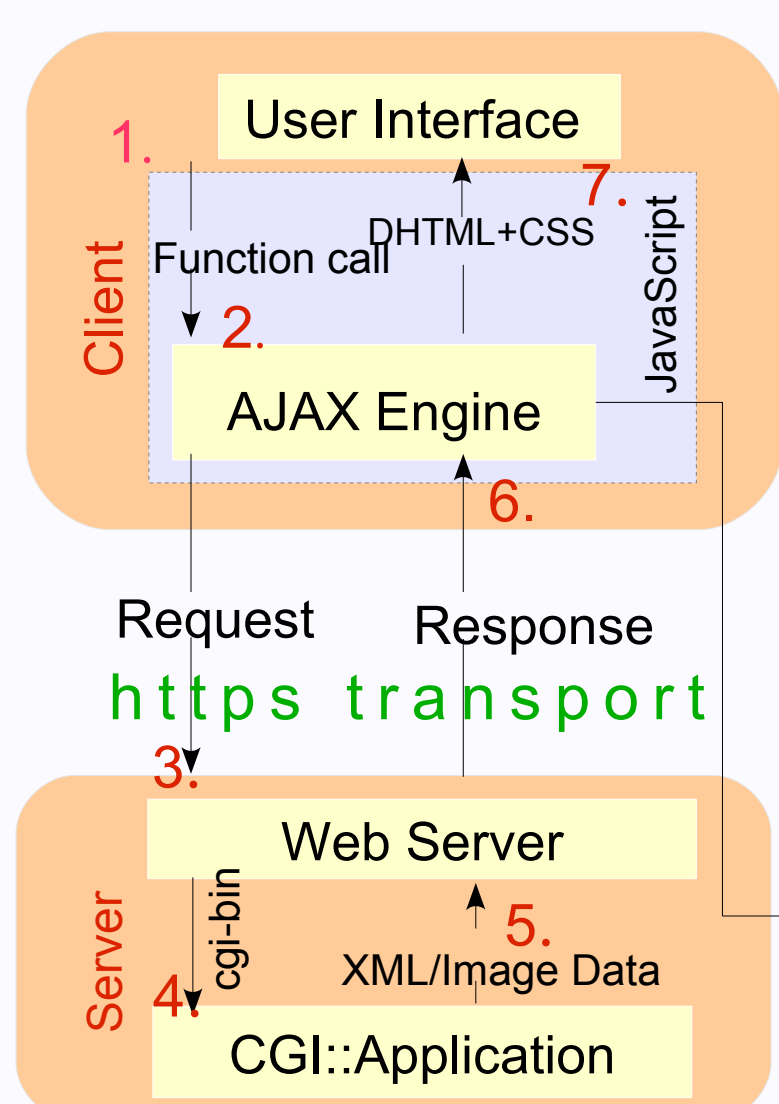
The framework consists of several independent components:

- a set of sensors that run on the site's
  - computing elements and worker nodes
  - and update a database
- a simple yet extensible web services framework
- an Ajax powered web interface with desktop like look-and-feel and control and supports several possible deployment scenarios:
- deployed fully at a site running a supported batch system
- existing site sensors adapted and reused with the web services components
- sites build the web server independently and use only the web interface

## Privacy of Information

- web access requires a valid Grid certificate of the client
  - true even for the site administrators
- several distinct, configurable levels
  - site administrators can access all the information
  - a list of VO administrators may access all the detail of each job of that VO
  - end users can access
    - detailed view of jobs from the Distinguished Name (DN) of the Grid certificate
    - summary view of jobs for the VOs the client is a member of
  - if strict privacy enforced
    - a client can access only jobs belonging to the certificate
- for VO level access, DN level privacy enforced for stdout, stderr etc.

## Client-Server Communication



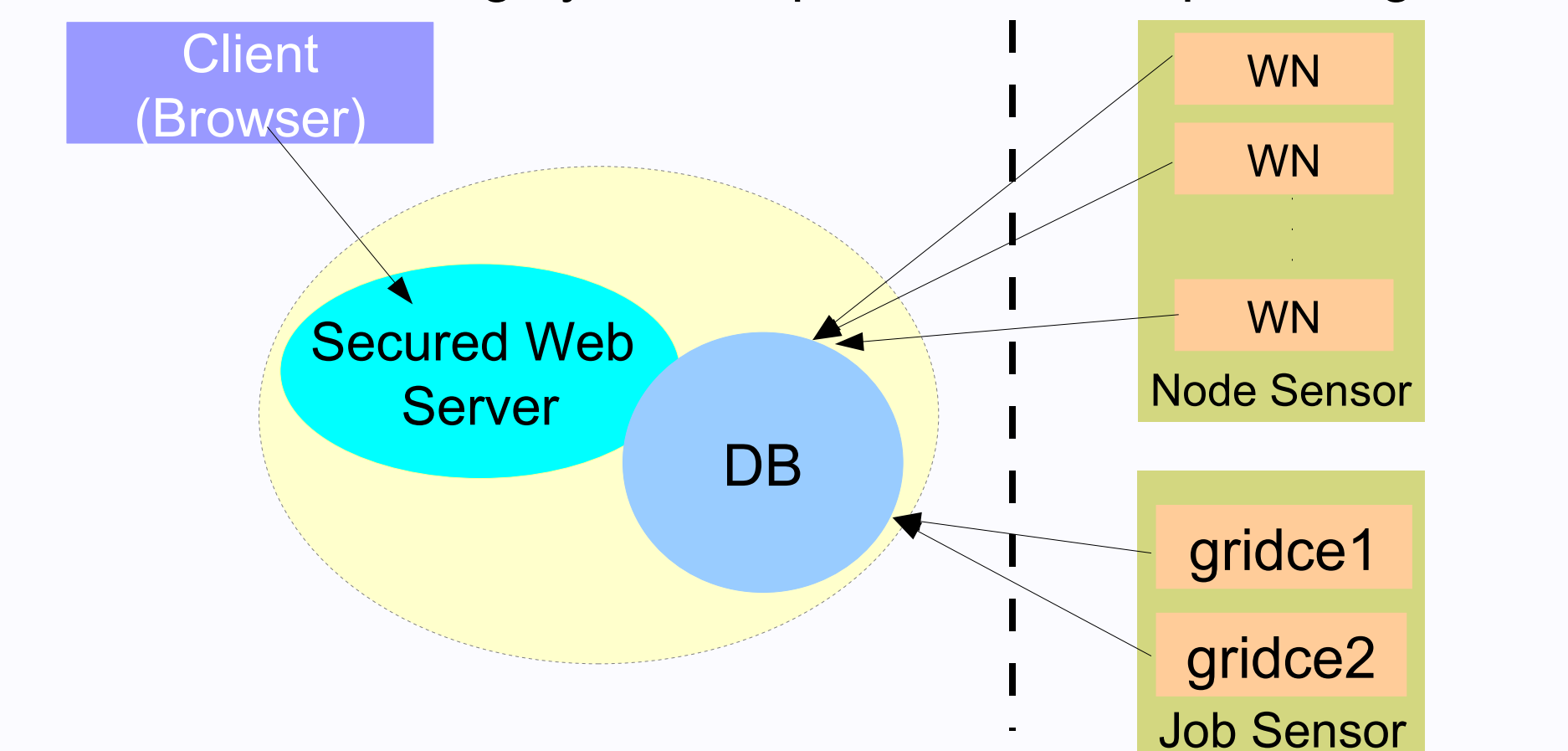
JavaScript at the heart of action

- HTML page loaded only once
- JavaScript manages Client-Server communication
- server sends XML, Text, JSON and image data
- JavaScript updates relevant parts
- asynchronous update key to responsiveness

XMLHttpRequest  
Document Object Model (DOM)  
eXtensible Markup Language (XML)  
XSLT

## Schematic View – Native Setup

The Job monitoring system implements a simple design



The sensors may be easily replaced with other source of data, e.g. Condor Quill DB, glideinWMS Collectors, Dashboard etc.

## Browser Support

The client side uses the jQuery library for

- basic effects and event handling
- User Interface
- Ajax calls

Fully supported browsers

- Firefox 2+
- SeaMonkey 1.1.13+
- IE 7+
- Safari 3+
- Opera 9.6+

Browsers with limited/no support

- IE 6, works partially; no further plans
- Google Chrome, not tested yet

## Web Interface

The web interface is divided into two parts:

- Control Panel
  - job list with global/local jobid
  - job state, selection and diagnosis panels
- View
  - a job summary table
  - graphical history of CPU load and memory usage by the job
  - a table with Grid related information for a job
  - several tabs that show detailed information of a running job
    - associated processes
    - job and work directories
    - last lines of output and error log files
    - worker node processes

A configuration panel allows to control the Client-Server communication and flow of information

## Monitor in Action

## Site Monitors & Documentation

Condor

T1\_US\_FNAL [https://cmsjobmon.fnal.gov:8443/jobmon/fnal\\_t1\\_newi](https://cmsjobmon.fnal.gov:8443/jobmon/fnal_t1_newi)

LPCCAF\_FNAL [https://cmsjobmon.fnal.gov:8443/jobmon/fnal\\_newi](https://cmsjobmon.fnal.gov:8443/jobmon/fnal_newi)

T2\_US\_UCSD <https://glidein-mon.t2.ucsd.edu/jobmon/ucsd>

LSF

T2\_IT\_Pisa <https://gridse.sns.it/jobmon/pisa>

PBS

T2\_DE\_RWTH <https://grid-mon.physik.rwth-aachen.de/jobmon/rwth-aachen/jobmon.html>

glideinWMS

UCSD [https://glidein-mon.t2.ucsd.edu/jobmon/ucsd\\_g](https://glidein-mon.t2.ucsd.edu/jobmon/ucsd_g)

<https://twiki.cern.ch/twiki/bin/view/CMS/ItalianT2ToolsJobMonitor>

<http://cmswiki.fnal.gov/twiki/bin/view/USCMS/Jobmon>

## Conclusion

- considered as a useful end-user tool
- pretty robust due to inherently distributed design
- in-built support for major batch systems, easily extensible through plug-ins
- scalability:
  - scales easily even for the largest T2s
  - a number of scalability issues addressed by the T1 implementation at FNAL
  - a large glideinWMS instance poses a challenge at a different scale
- security:
  - uses a basic secured web server; security issues not yet addressed

## Acknowledgements

We are thankful to Giuseppe Bagliesi, Jon Bakken, Stefano Belforte, Giacinto Donvito, Igor Sfiligoi and Frank Wuerthwein for helpful suggestions and support.