International Conference on Computing
in High Energy and Nuclear Physics 2010 (CHEP 2010)

USING WIDGETS TO MONITOR THE LHC EXPERIMENTS

I. González Caballero (U. Oviedo, Spain)
S. Sarkar (SNS, Pisa, Italy)

Taipei (Taiwan) – October 2010

Disclaimer…

- During this presentations several popular commercial companies and services will be mentioned. We are not affiliated by any means with any of them.

…and some remarks

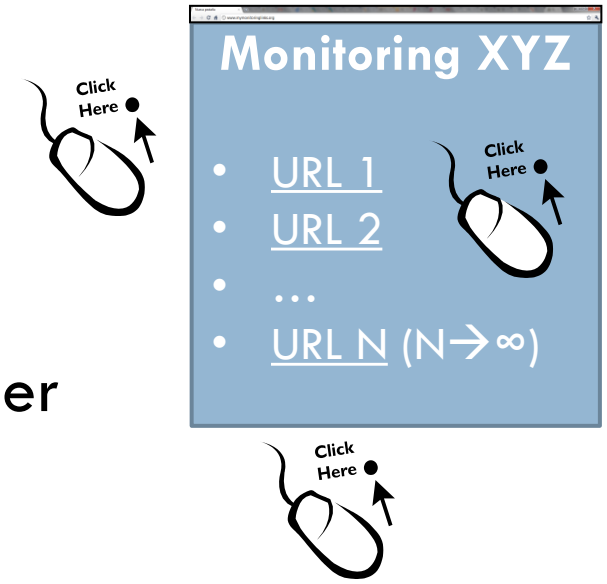
- We developed our experience with widgets in the context of the computing tasks at the CMS collaboration
  - …but we believe this work can be extrapolated to many other environments
- While we have made an effort to show examples for every widget platform supported, we have no easy access to some of them and therefore you might not see how our work looks like there
  - Just try and let us know!
- This presentation will be mostly devoted to why widgets are a good idea to monitor stuff, how they can be built and some examples
  - For a comprehensive list of widgets developed in the CMS environment, please refer to the documentation

Monitoring the world…

- The data useful for monitoring is usually scattered over several databases, web pages (central or local) and systems
  - A myriad of more or less complex web pages provide access to (usually) project related plots and numbers
- Users, experiment supervisors, service operators and system administrators are interested in different types of information
  - Different operators need to monitor different data from, sometimes, the same sources
  - Responsibilities are split in different ways at different places or may change over time
- People looking after something (site, service,…) know better the strong and weak points
  - They would like to focus their monitoring efforts on those aspects that are unstable or more prone to failure

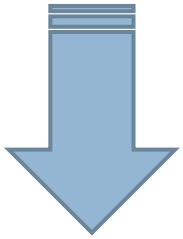
Monitoring solution 0 (or even -1)

- Scanning recursively through a set of web pages
  - Not very practical
  - Click! Click! Click!…
  - … even if you keep all the links together in another web page
  - … even if you manage to have all the links up to date
  - …

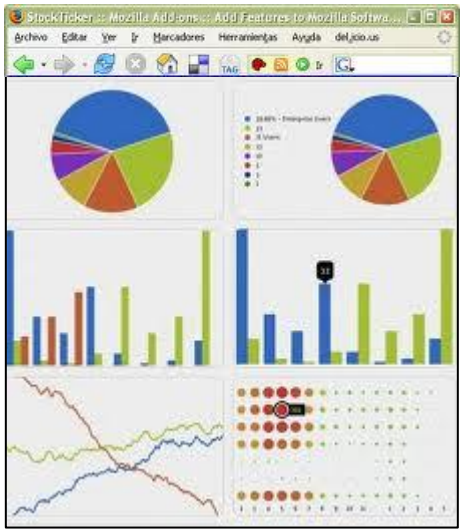


Monitoring solution 1: Customized web

- Fortunately most systems can be thought also as web services providing monitoring data through a more or less simple API



- Personal/Institutional monitoring web pages aggregating these data can be built easily based on that
  - Each individual needs to build its own HTML file
  - API changes (they happen!) means all individuals have to change their HTML files
  - Each «user» needs to master HTML (and possibly Javascript)

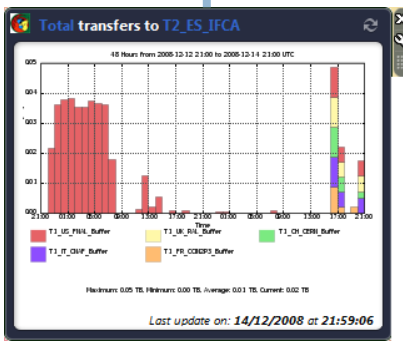


Monitoring solution 2: Widgets

- Fortunately most systems can be thought also as web services providing monitoring data through a more or less simple API

Web widget (*Wikipedia*): *It's a small application that can be installed and executed within a web page by an end user... Other terms used to describe web widgets include: gadget, badge, module, webjit, capsule, snippet, mini and flake.*

- Provide a set of more or less general customizable widgets that can be aggregated to form private or public monitoring web pages
  - Widgets are single, although not necessarily simple, units of information (monitoring data, for example) …
  - … that can also be built on top of those web services…
  - … that users can select and aggregate to (at least one of) the platforms in the market: Netvibes, iGoogle, Apple DB, etc…
  - … knowing that if some API or service is modified, the fixes propagates automatically to them



Universal Widget API (UWA)

- It is a widget Application Programming Interface developed by well known web portal Netvibes
- Widgets developed using UWA are compatible with the main widget platforms

  - It may also be easily added to any web page (i.e. blogs)
  - Users may use the platform they like most or feel more comfortable with
- Supports modern web technologies (AJAX)
  - A typical widget is only a bit of (X)HTML and JavaScript
- The API and part of the server technologies behind are released as Open Source through:

http://www.netvibes.org

Pros and Cons of UWA

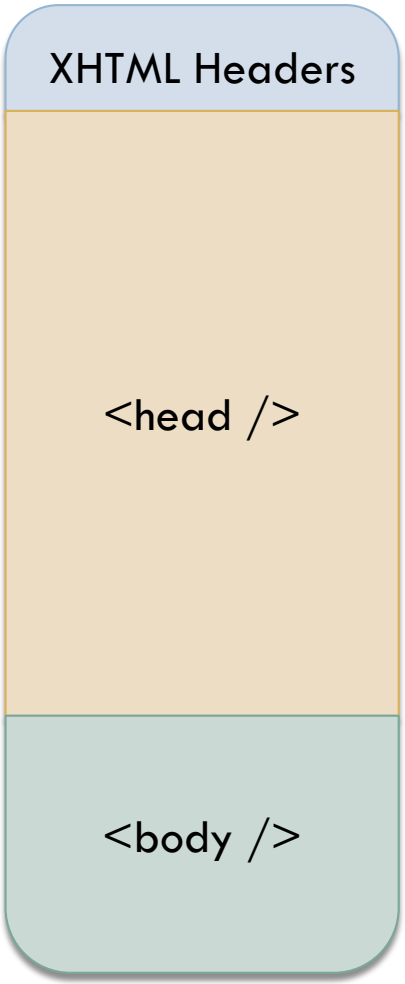
Pros

- Single development valid on many platforms
  - No specific widget platform is imposed on the users/admins/operators
- Open Source
  - Customization possible at many different levels
  - Including forking the project
- Great set of web tools:
  - Ecosystem: to expose widgets
  - Localization supported
  - Developers documentation and fora
- Many complementary utility widgets available:
  - (Multi) RSS feeds, automatic parser, the usual clocks, calendars, …

Cons

- Answers to technical questions and support a bit slow
- No current way to provide a simple URL where a widget with a given set of values for the options can be downloaded or installed
  - May be done by installing and hacking a server
- Still no support for modular programming
  - Each widget file (html) must have all the code it uses
  - No "includes" are allowed
  - Support for this very important use case in the long term requested
- Authentication through certificates might be not supported
  - Some web services require it!

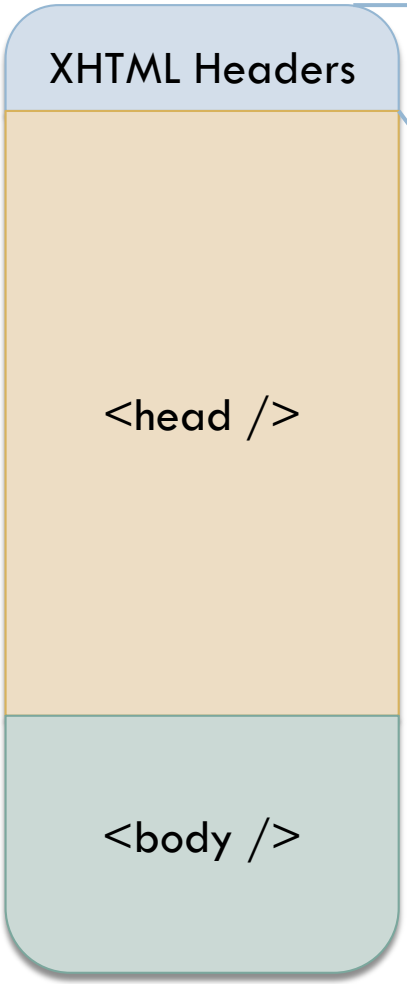
UWA widget basic structure



- A UWA widget is an XHTML file with
  - Some HTML
  - Some CSS
  - Some JavaScript
  - A few XML fields used for the preferences
- There are three main blocks in a UWA widget
  - A set of standard XHTML Headers
  - The core of the widget where the behavior is defined
  - The HTML skeleton of the widget

http://dev.netvibes.com/doc/uwa/documentation/anatomy_of_a_uwa_widget

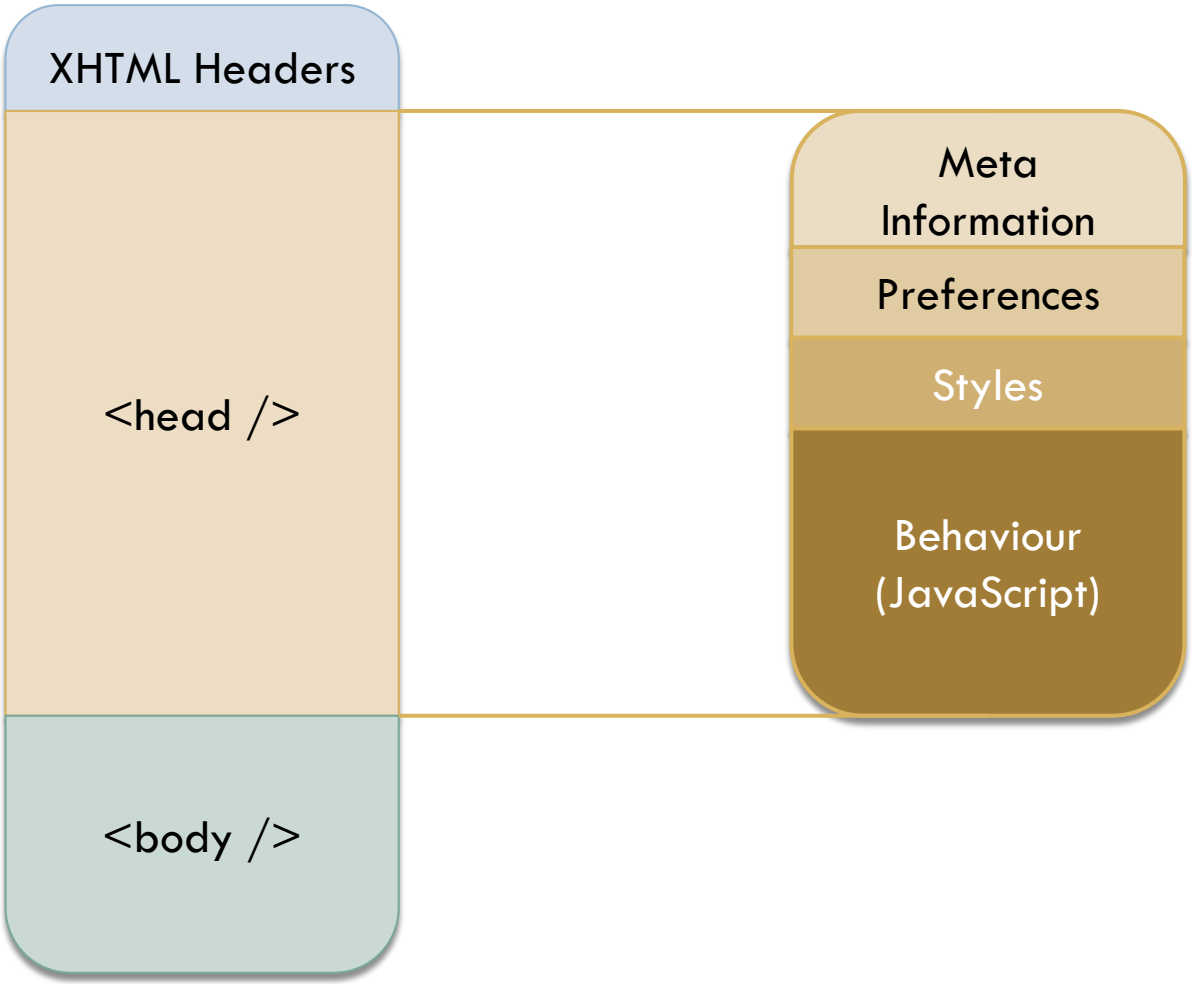
UWA widget base structure - Headers



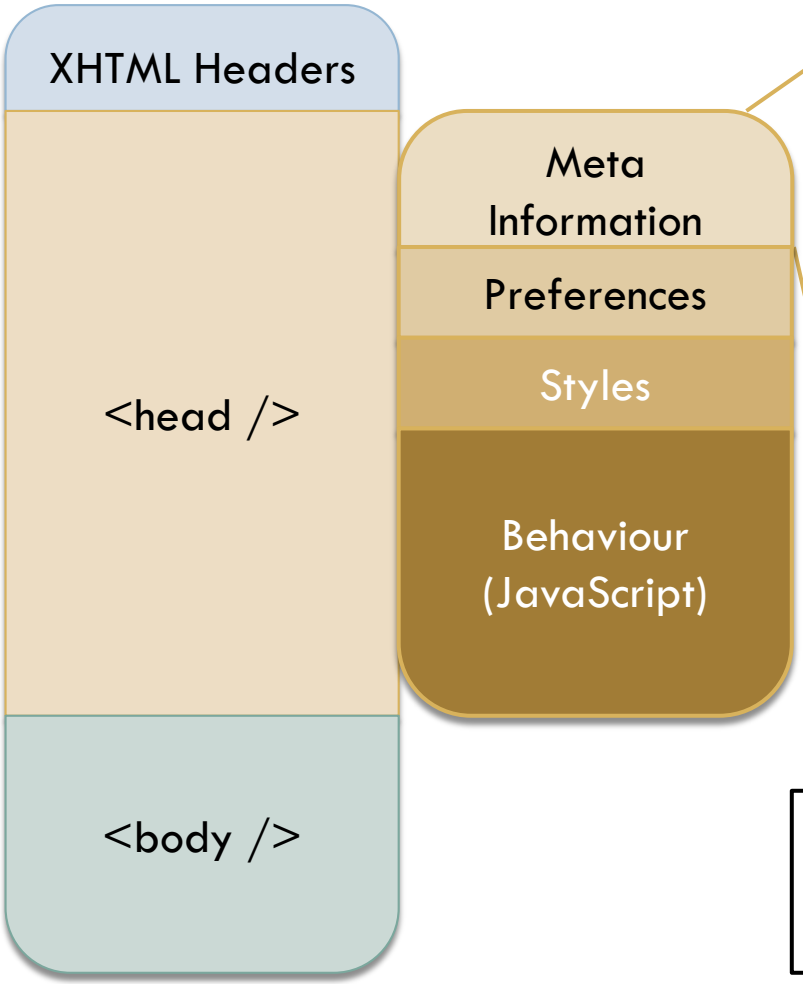

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:widget="http://www.netvibes.com/ns/">
```

- Standar headers
  - The widgets should use the utf-8 encoding

UWA widget base structure - <head/>



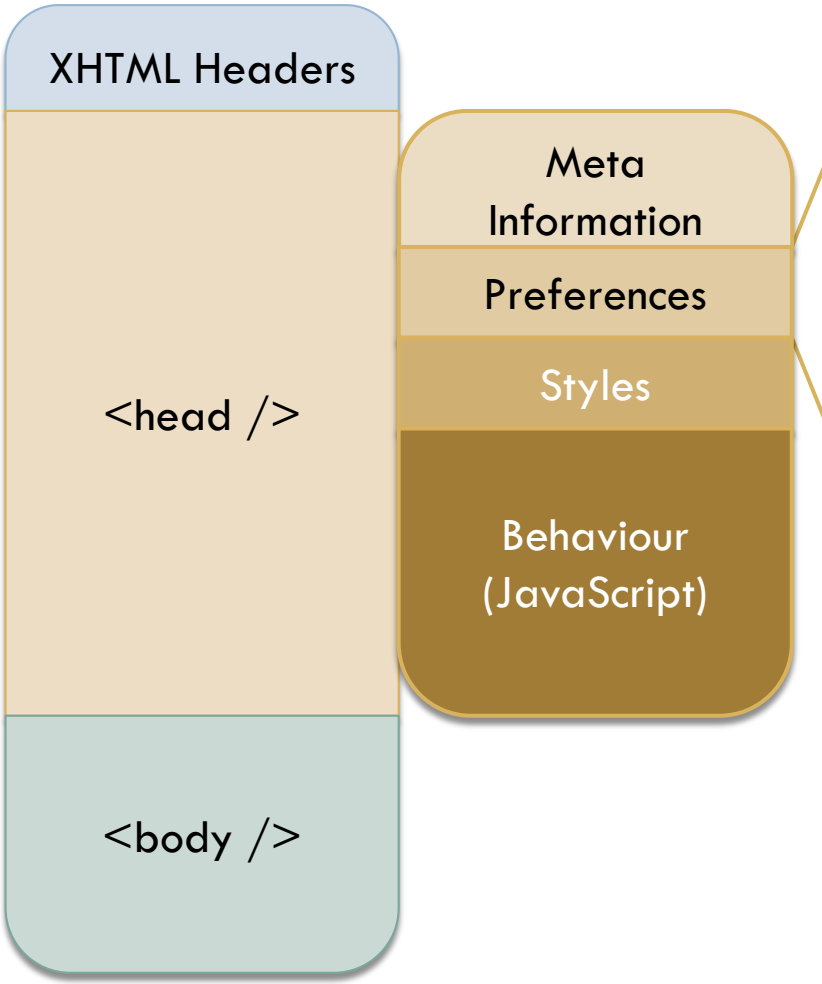
UWA widget base structure - <head/>




```
<meta name="author" content="IGC"/>
<meta name="description"
   content="Hello world"/>
<meta name="website"
   content="http://example.com/bla"/>
<meta name="version" content="1.1"/>
<meta name="keywords"
   content="CHEP2010, widgets, example"/>
<meta name="thumbnail"
   content="http://example.com/bla.gif"/>
<meta name="apiVersion" content="1.0"/>
<meta name="debugMode" content="false"/>
```

- Small description of the widget
  - Author, thumbnail, keywords…

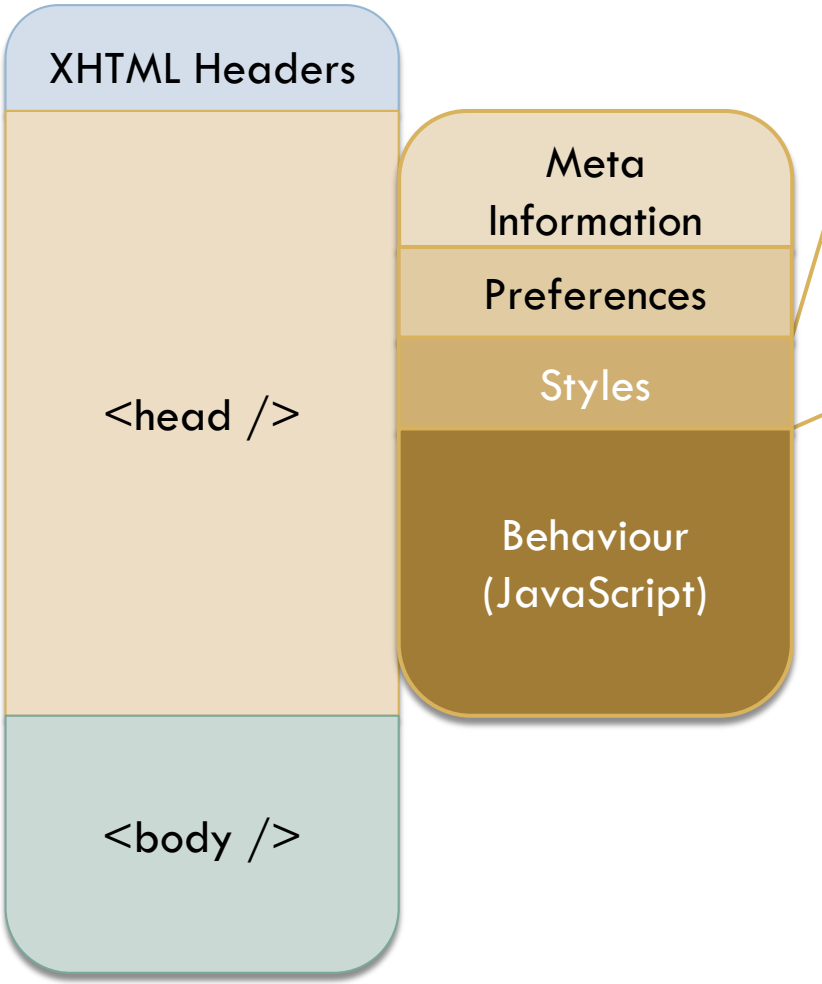
UWA widget base structure - <head/>




```
<widget:preferences>
  <preference name="thename" type="text"
label="Name" defaultValue="unknown"/>
  <preference name="service" type="list"
label="Service" defaultValue="CE">
     <option value="CE" label="CE"/>
     <option value="SRM" label="SRM"/>
  </preference>
  …
</widget:preferences>
```

- Defines how the users interact with the widget
- Possible options include checkboxes, list boxes, text fields, passwords…

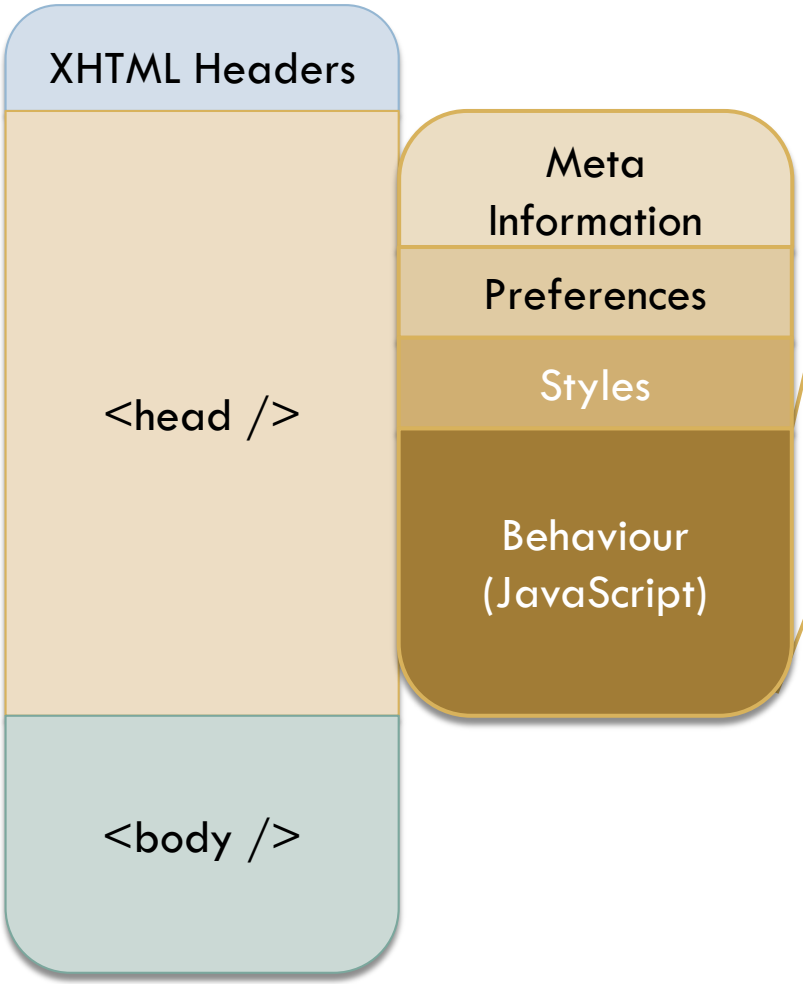
UWA widget base structure - <head/>




```
<style type="text/css">
  a:hover, a:active, a:visited {
    text-decoration: none;
    border: none;
  }
</style>
```

- CSS styling syntax
  - Not mandatory: One can use the usual HTML code to style the widget elements
  - …but it is a good coding practice

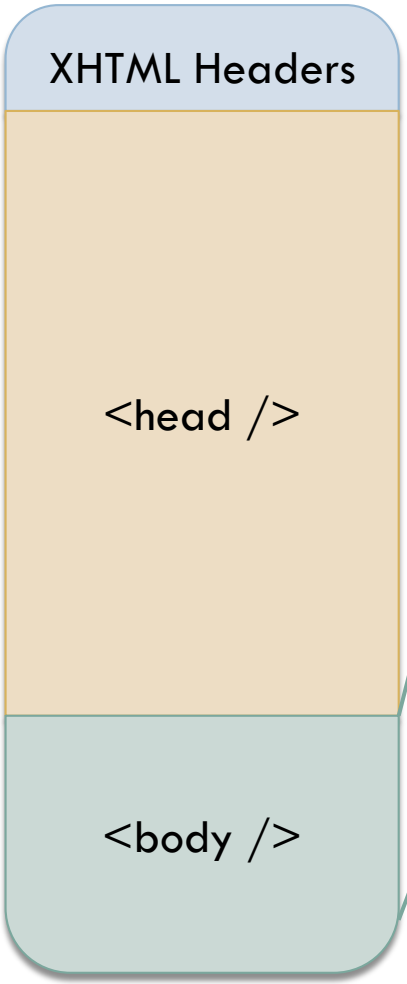
UWA widget base structure - <head/>




```
<script type="text/javascript">
 widget.onLoad = function(){
   var name = widget.getValue("thename");
   widget.setBody("Hello" + name);
 }
</script>
```

- Use JavaScript to:
  - Find out the preferences
  - Find the information from the web services in JSON, XML, HTML or plain text formats
  - Build the final information and present it to the users
  - May interact with the DOM in the <body/> part

UWA widget base structure - <body/>



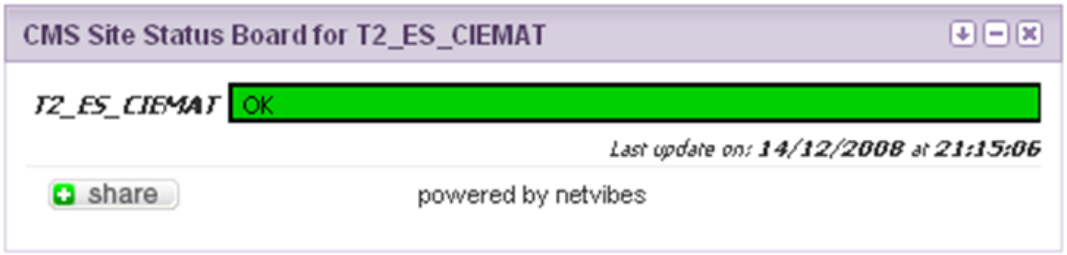

```
<body>
  <p>This widget is an example for
    <a href="http://event.twgrid.org/chep2010/">
       CHEP 2010
    </a>
  </p>
  <p>Loading...</p>
</body>
```

- Defines the HTML skeleton for the widget
- The JavaScript can interact with the elements in this section

Improvements in the widgets

- We have extended some of our widgets by adding extra functionality to them:
  - Versioning support
  - Debug information
  - Standard date format for the footer
- Still not used in all our widgets
  - We have to use cut & paste since there is no modular support in UWA yet
  - A kind pre-compiler may help adopting it

Example widgets: Site Monitoring



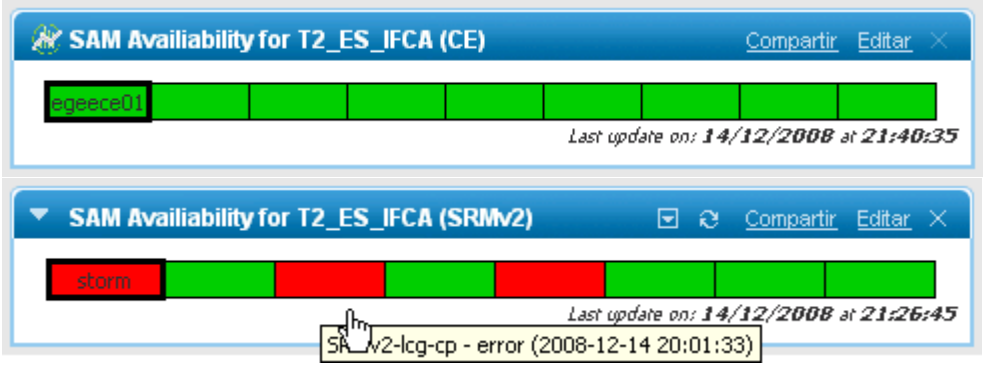
- Site Status Board -

- Shows the status of the Site according to the Site Status Board metrics defined in CMS

http://eco.netvibes.com/widgets/296601/cms-site-status-board

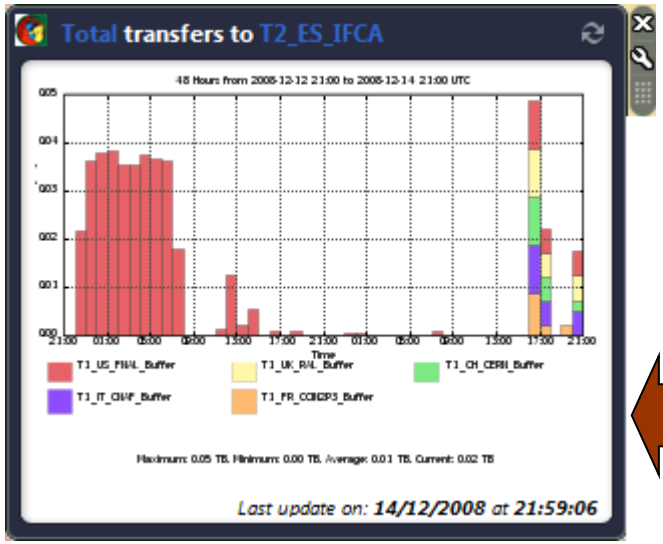
- SAM Tests -

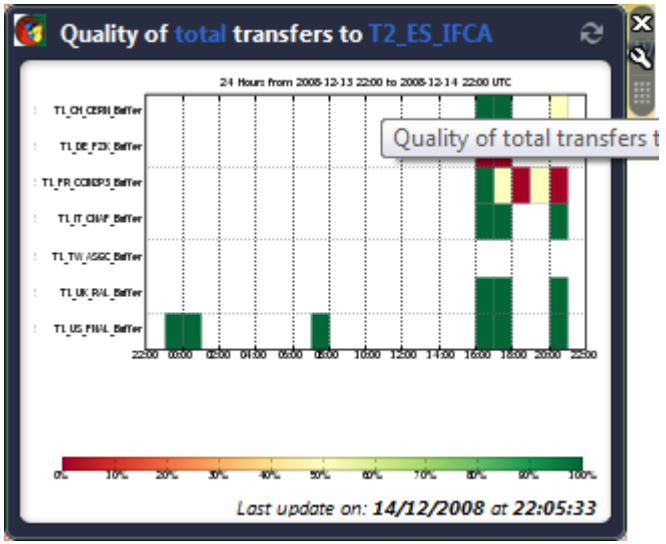
- Available LCG and CREAM CE, SRM & SRMv2
- Information on particular tests and overall status
  - From the Dashboard



http://eco.netvibes.com/widgets/233892/cms-site-availability-monitoring-sam

Example widgets: Data Transfers



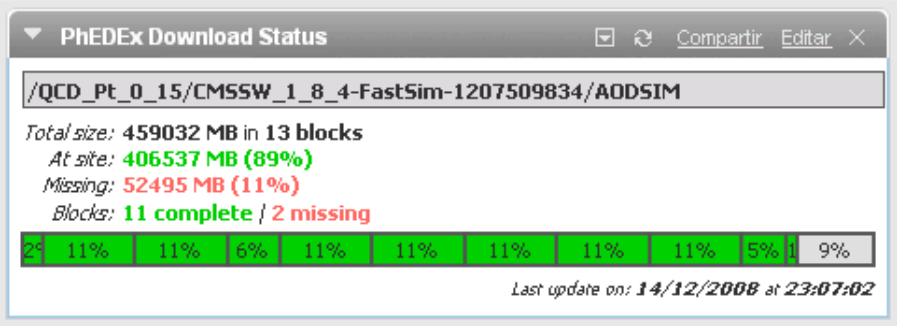


- Transfer Rate/Volume or **Transfer Quality -**

- Direction: Up/Down link
- Instance: Prod, Debug, Dev, All
- Period shown: Last day, week, month…

Transfers: http://eco.netvibes.com/widgets/206202/phedex-transfer-monitoring
Quality: http://eco.netvibes.com/widgets/206702/phedex-transfer-quality-monitoring

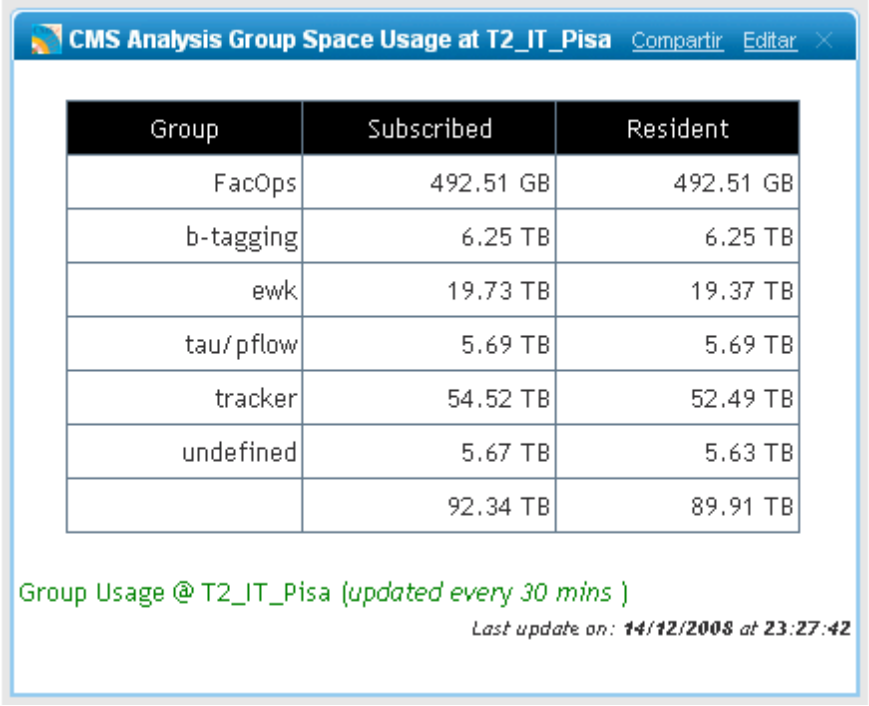
Example widgets: Data Management



- Dataset Download Status -

- Shows how much (block, files) of a given dataset are at a site
- Uses DBS/PhEDEx web services
- We are studying ways of providing a link to a customized widget each time a dataset is approved for transfer

http://tinyurl.com/datasetdownloadstatus



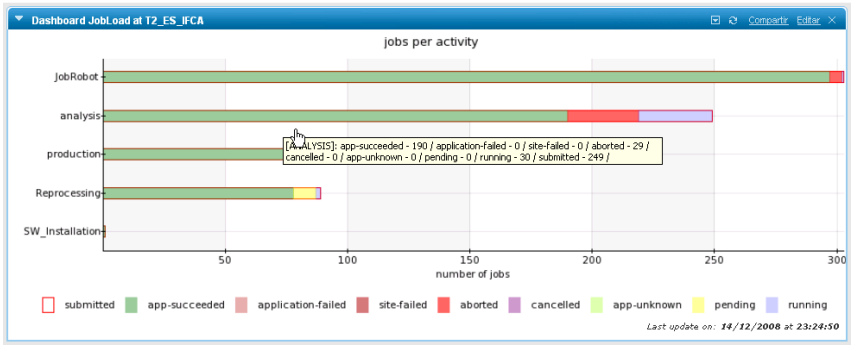
Group	Subscribed	Resident
FacOps	492.51 GB	492.51 GB
b-tagging	6.25 TB	6.25 TB
ewk	19.73 TB	19.37 TB
tau/pflow	5.69 TB	5.69 TB
tracker	54.52 TB	52.49 TB
undefined	5.67 TB	5.63 TB
	92.34 TB	89.91 TB

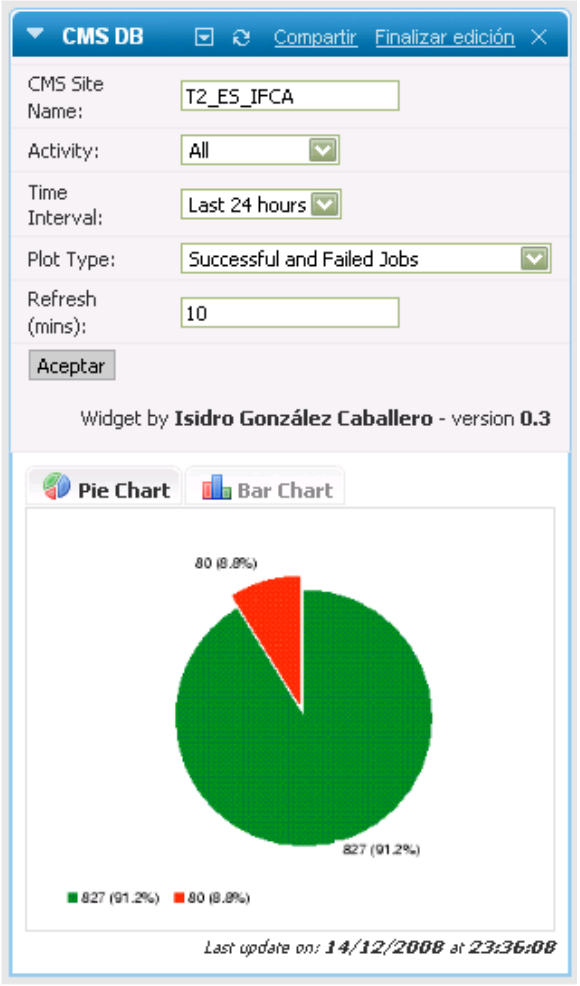
- Group Space Usage -

- Visualize the space each physics or detector group is using at a given site

http://farmsmon.pi.infn.it/dcachemon/GroupUsage.html

Example widgets: Job load



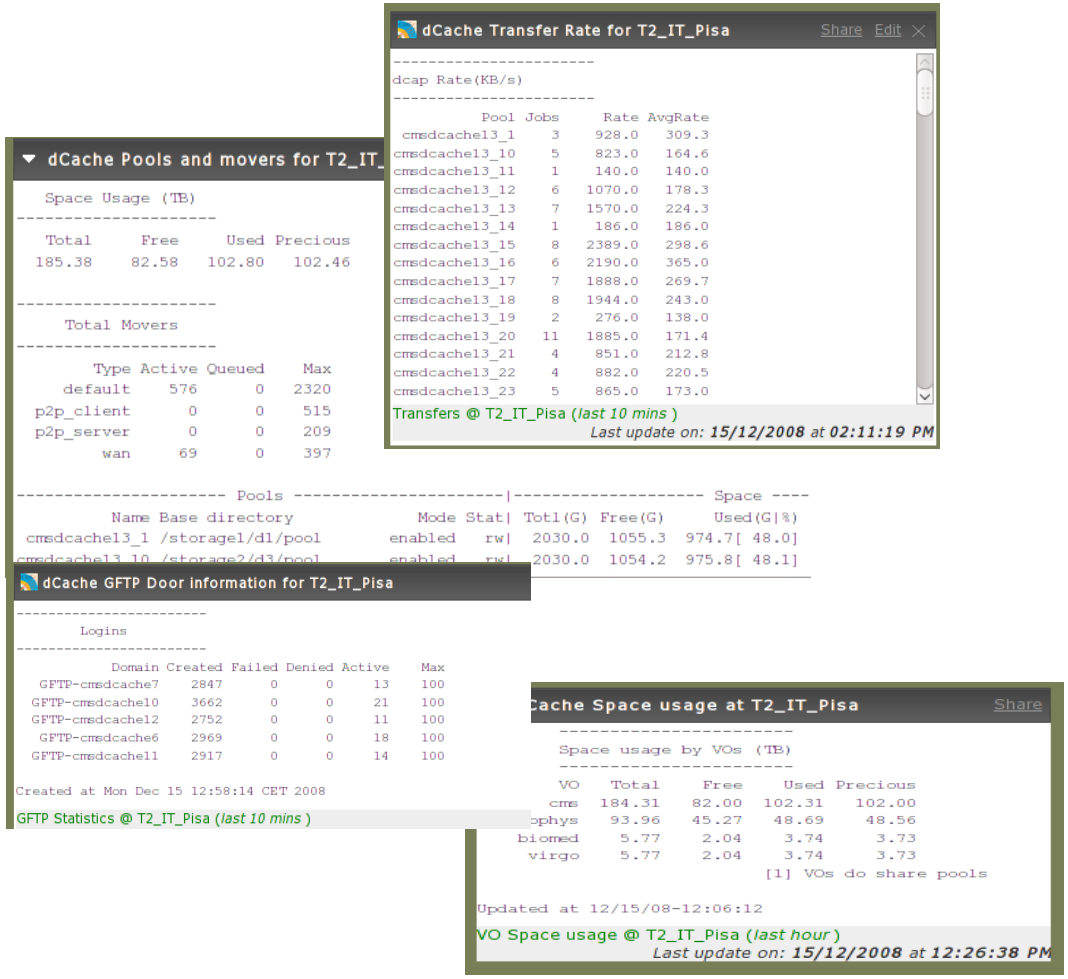


- Interactive View or **Historic View -**

- Show plot for any activity or for a specific one
- Sort by Dataset/Activity/User (interactive)
- 3 different plots based on error codes (historic)
- Pie/Bar Chart (historic)

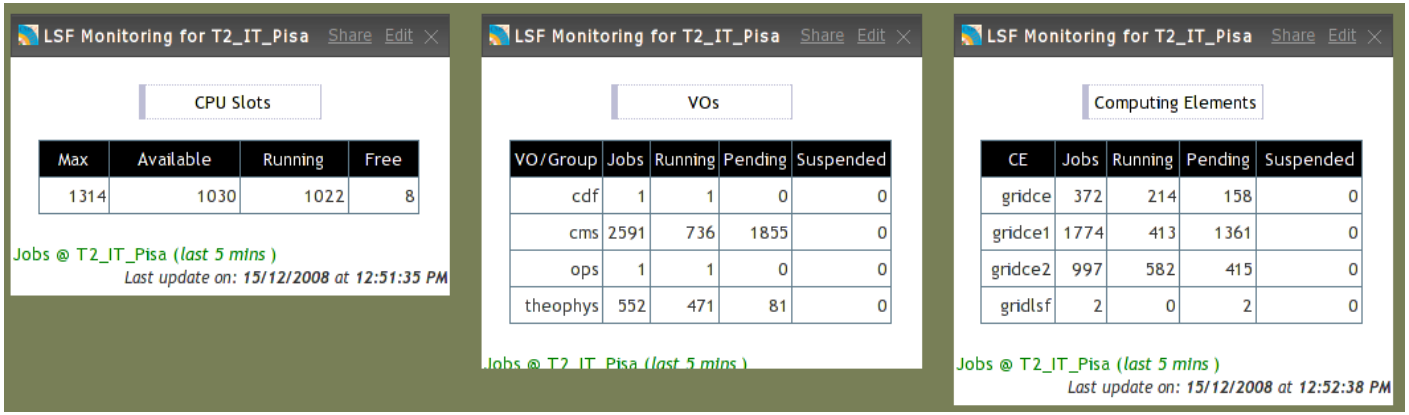
Interactive: http://tinyurl.com/interactivejobload
Historic: http://tinyurl.com/historicaljobload

Example widgets: dCache monitoring



- Built to monitor a local service (dCache) in a site
- May be used by:
  - Other dCache sites
  - To customize private monitoring pages
- Includes:
  - dCache pools
  - Internal transfer rates
  - Door information
  - Local space monitoring

Example widgets: Local batch system

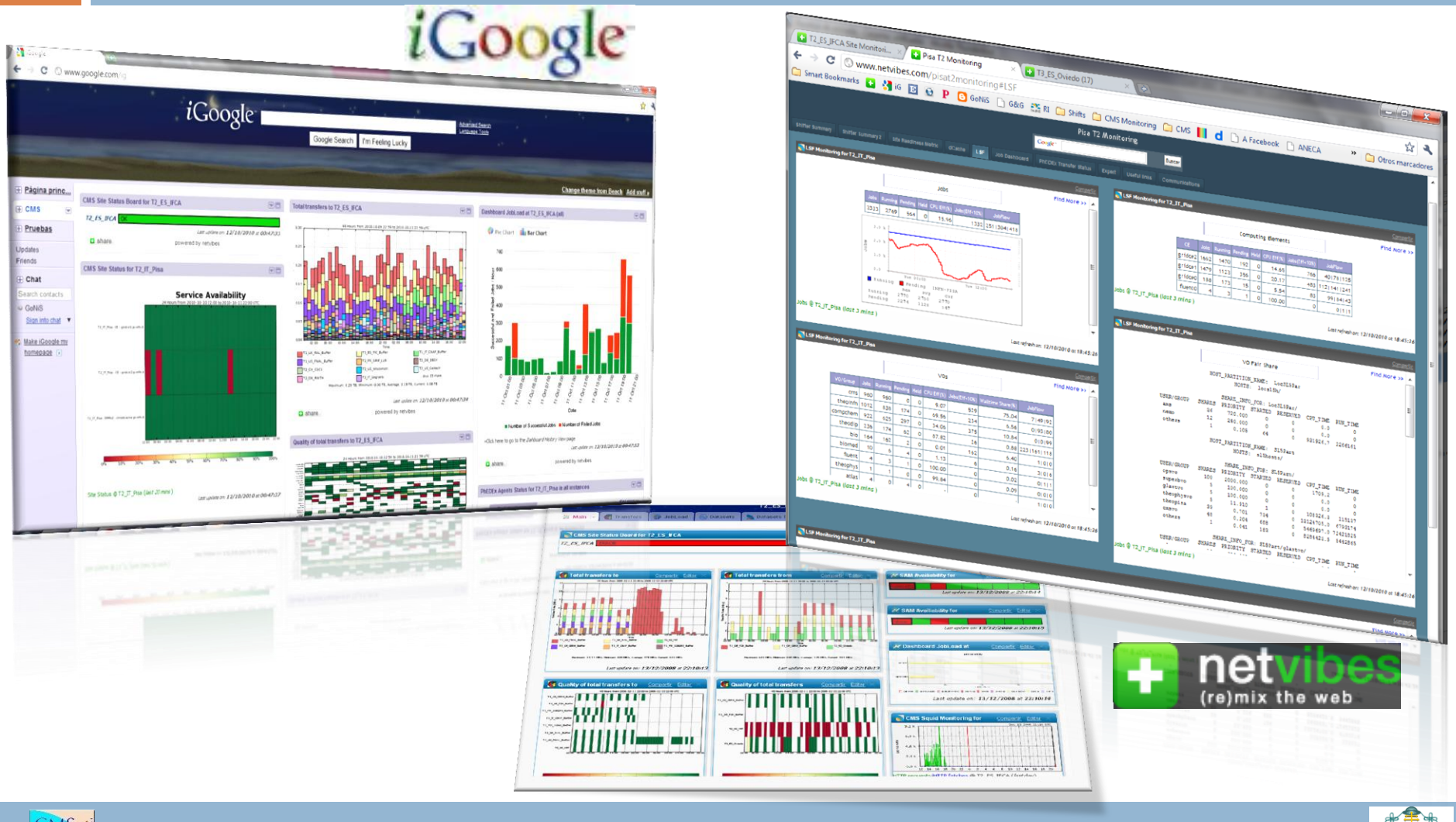


LSF Monitoring for T2_IT_Pisa — Share Edit

User DN

Group	Jobs	Running	Pending	Suspended	DN
theophys	400	400	0	0	/C=IT/O=INFN/OU=Personal Certificate/L=Fisica Pisa/CN=Guido Cossu
cms	1770	179	1591	0	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=fiori/CN=664352/CN=Francesco Fiori
cms	200	143	57	0	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=wardrope/CN=645210/CN=David Wardrope
cms	161	134	27	0	/C=IT/O=INFN/OU=Personal Certificate/L=Fisica Milano Bicocca/CN=Andrea Benaglia
cms	100	100	0	0	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=jdamgov/CN=537562/CN=Jordan Damgov
cms	65	65	0	0	/C=IT/O=INFN/OU=Personal Certificate/L=Milano Bicocca/CN=Giuseppe Cerati
cms	36	34	2	0	/DC=org/DC=doegrids/OU=People/CN=Sean Simon 537713
cms	27	27	0	0	/C=CN/O=HEP/O=PKU/OU=PHYS/CN=Zhen HU
cms	21	21	0	0	/C=IT/O=INFN/OU=Personal Certificate/L=Trieste/CN=stefano belforte
babar	16	15	1	0	/C=IT/O=INFN/OU=Personal Certificate/L=CNAF/CN=Armando Fella

Examples: Agregated in web portals



Conclusions

- The widgets technology provides a different way of visualizing the same monitoring data
  - Enabling a great level of personal customization
  - Solving the problems coming from having a unique monitoring tool that suits everybody
- Widgets based on UWA can be used in a great variety of platforms
  - Including the main ones (by number of users) in the market
  - All of them with their advantages, tricks and nice fireworks
- Developing new widgets is rather easy
  - Only a bit of XHTML and JavaScript is required
  - Provided there are good web services accessible
  - Changes in the API or improvements are quickly propagated to the users
- The tools presented here are not to be seen as replacements to the current various dedicated and quite complete monitoring systems out there:
  - It should be better though as a orthogonal complement to them
- Widgets are being efficiently used to monitor the CMS Computing services

References

- CMS
  - Twiki for widgets: https://twiki.cern.ch/twiki/bin/viewauth/CMS/WidgetsMonitoring
  - Widgets Code: http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/IGonzalez/widgets/
- Universal Widget API:
  - Documentation http://dev.netvibes.com
  - Netvibes Ecosystem: http://eco.netvibes.com/
- Portals:
  - Netvibes: http://www.netvibes.com
  - iGoogle: http://www.google.com/ig
- …

This is the end…

THANK YOU VERY MUCH FOR YOUR ATTENTION

Questions and comments are very welcome