

Table of Contents

Transport Review	1
Are we doing things consistently for all of our applications?.....	1
If we are doing things in multiple ways, is there a good reason for it?.....	1
Are there any new technologies that would help us here?.....	1
How long would it take to change?.....	2
How would that impact the other layers?.....	2
Conclusion.....	2

Transport Review

This document is the output of the Transport working group: Alex and Robert.

As requested in the WLCG monitoring consolidation meeting 18 July 2013, we attempt to answer the following questions regarding the transport layer:

- Are we doing things consistently for all of our applications?
- If we are doing things in multiple ways, is there a good reason for it?
- Are there any new technologies that would help us here?
- How long would it take to change?
- How would that impact the other layers?

Are we doing things consistently for all of our applications?

Two modes of Transport are used by our applications:

- messaging
- http

There is also the MonALISA (CMS) collector from which we scrape messages from the log file and insert them into the database.

Current status of implementation

- dashboard
 - ◆ each has its own implementation on top of the recommended messaging libraries
 - ◆ different http libraries
- SAM
 - ◆ ATP OSG downtimes (python consumer custom)
 - ◆ perl-GridMon (perl publisher/consumer using msg-to-handler), but also additional custom library (nagios send-to-msg probe)
 - ◆ python-GridMon (python publisher, custom for WN tests)
 - ◆ OSG publisher (python wlcg-msg-publisher 2004, publisher)
 - ◆ msg-consume2db (python consumer, includes filters and backend abstractions)
 - ◆ msg-to-handler (perl consumer, used by MRS bootstrapper - discontinued)
 - ◆ fts/dax consumer (python custom by IT/ES)

If we are doing things in multiple ways, is there a good reason for it?

The split between http and messaging makes sense, as they serve different purposes as transport layers. However, there is no good reason for not consolidating the messaging layer.

Are there any new technologies that would help us here?

There are several new technologies for distributed messaging, however a more thorough future tech review is needed. Also, it may be better to have this decision taken by the experts in the messaging team on who we rely on to provide the libraries anyhow.

It should be noted that the messaging team has relatively new messaging libraries for a while now and not all components have migrated to them. We should probably focus on consolidation and migration to the recommended tools from the messaging team.

How long would it take to change?

Dashboards should be relatively easy as they only consume data, however SAM consolidation could take several months to get it implemented correctly and tested. Work has already been done in SAM consolidation with the MEG project.

How would that impact the other layers?

Should not impact other layers in any way. Migration would be transparent.

Conclusion

- Split between messaging and http is ok.
- We all want use the same libraries.
- Consolidation will take some time.

This topic: LCG > TransportReview

Topic revision: r1 - 2013-08-02 - RobertVeznaver



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback