

Minor comment on the Legend in Section 5 Constructs used in WS-Agreement – An Analysis

There seem to be two kinds of ways of denoting yes and no:
Y or y / N or n. Is there any difference in the meaning? If not then
there should be a consistent way of denoting.

Other wise, I think this document is very important in
showing how WS-Agreement is actually used in various projects.

Best Regards
Toshi

Replaced N by n and Y by y in the table.

Posted by: jens 2009-10-30 09:24:05 *Comments/review of WS-Agreement experiences document*

The goals of the WS-Agreement specification is to standardise the
terminology and the concepts, and the WSDL, for the WS-Agreement
protocol.

Example scenarios in GFD.107 cover job submission, advance
reservation of resources, and negotiating QoS (called "service
parameterization" in the document). All these scenarios appear to be
covered by the experiences document.

The experiences document does not discuss the expected behaviours of
the service entities (agreement providers and consumers) described in
GFD.107.

Added a clarification in Section 4.

The WS-Agreement implementations are used in grids with very diverse
middleware stacks (GT4, GRIA, Unicore-based, etc) -- this is good
because it means the software is applicable in a wide range of
contexts.

Made this explicit in Section 4.

It is not immediately clear how many independent implementations there
are. Section 5.1 suggests there are somewhere between six and eight
-- are they completely independent or do they share code or libraries,
eg. to implement the underlying WS-* protocols? More than one project
used WSAG4J. In any case, it would appear there are "enough"
implementations.

Added a clarification in Section 4 about the number of code independent implementations and those sharing a
common framework.

However, it would appear from section 7 that only two implementations
have been used for interoperation testing. It would be helpful to
have interoperation testing between other implementations.

The major purpose of section 7 is to show that interoperation is possible between two code-independent
implementations **and** across different middleware stacks. Due to the different goals, foci and use-cases showing
interoperability between implementations of the different projects would require a huge effort to converge to a

minimal common use-case. However, in many of the projects several components use WS-Agreement to negotiate agreements on resource usage, thus in each of these projects we have different implementations of providers and consumers of WS-Agreement (also across administrative domains) showing the interoperability on the level of the language and protocol specification of WS-Agreement.

Moreover, the interoperation testing appears to rely on XSLT to translate one WSRF format to another one, via an interoperability proxy. This may be outside the scope of WS-Agreement but does not bode well for interoperation. On the positive side, it does show that different implementations were used.

This translation with XSLT is required only because of the implementation of different WSRF versions in GT4 and UNICORE at the time of the interoperation tests.

Are there implementations in languages other than Java, specifically C or C++? None are mentioned in the experiences document. Successful interoperation between C/C++ implementations and Java implementations would be particularly convincing.

Added a clarification in Section 4.

There is no doubt the WS-Agreement specification has been useful to the projects referred to in the experiences document. Conversely, there is no doubt that having a standard for WS-Agreement is useful because the projects have a need for this type of protocol and it's clearly better to have standard than for each project to "roll its own."

I would expect that if the WSRF interoperation problems were addressed (obviously outside the scope of the WS-Agreement projects), the two WS-Agreement implementations would fully interoperate. In other words, they actually do interoperate, the proxy is only needed to make the WSRF implementations interoperate. If this were true, my concerns about interoperations would just be that interoperation has been tested on only two implementations, it would be nice to see wider interoperation testing.

While I am also concerned about the lack of C++ implementations, there is clearly a lot of experiences with WS-Agreement in different projects, and the protocol fills a need. A lot of high quality work has been done by the group, and it shows in the documents. If the remaining concerns could be addressed in a satisfactory way, I would recommend that GFD.107 be promoted to full OGF standard.

Regards
--jens