

Comment	Originator	Comments	Outcome
1	Zach Hill	The inclusion of authorization information in the DEPR seems problematic as it requires additional security measures to protect the DEPR itself. EPRs generally name and describe how to access a resource, but do not provide the actual request parameters themselves. A better approach might be to separate the DEPR from the credentials required to access it by providing a delegation interface to the DTF itself.	
2	Duane Merrill	It's unclear how HTTP/SCP/etc would work. I could envision the destination DEPR having a file://blah protocol so that, if the DMI instance was co-located with the sink, you could use http as the source.	
3	Chris Sosa	I have a comment about the different states you mention in 5.4.1. I notice that the the transferrer may be in the process of "undoing" a transfer upon a failure. What state is returned during this process? I imagine an "undo" state would be different from both a transferring and failed states.	
4.1	Rob Schuler and Anne Chevernak	From section 3: "the ability to transfer data from one location to another" I had a bit of a misconception about the goal of the OGSA-DMI specification before I reviewed it. I previously thought it would be similar to the Globus Reliable File Transfer service or the Storage Resource Management srmCopy interface. These tools manage several transfers per request, whereas as I now understand it the OGSA-DMI interface is intended to support essentially a single "one location to another" data transfer. So it seems more akin to a Web service layer atop a data transfer utility. For instance, it appears to be a candidate to sit in front of the GridFTP servers as a WS interface for the control channel. Also, it could be a candidate to replace srmCopy though without support for multiple transfers. But as I understand it, it would not be a candidate for a service that schedules multiple data transfers per request.	We envisage that a DMI service can effect a multiple file transfer from a source to sink transfer - a DEPR may be used to encode an aggregate of data such as a collection of files, e.g. see last paragraph in section 13, so in theory a DMI service should be able to support this behaviour.
4.2		From section 3.2.1: "this version of the specification does not make use at all of the WSAddressing Endpoint Reference data structure" I don't necessarily have an issue with that decision. I just would like to have a better understanding of why it isn't used. For instance, what made WSA EPR unsuitable for this specification? Also, given that the name used is "Data Endpoint Reference (DEPR)" and in fact the data structure looks superficially similar to the WS-Addressing EPR (figure under 4.2.1.2.1), this seems sure to cause confusion.	
4.3		From section 4.2.1.2.1: DataLocations/Credentials element I think this is a good choice to include a Credentials element nested in the DataLocations element of the DataEPR. Ideally though, I would have a default Credentials element elsewhere that could apply to both the source and sink DEPRs, perhaps in the [transfer requirements] parameter of the create DTI call. I think it's likely that the user's credential often will be the same for source and sink. By only having this element nested within the source and sink DEPRs results in significant overhead.	
4.4		From section 4.2.1.3.1: EndNoLaterThan element The EndNoLaterThan setting could be problematic. Does a user really want to abort a transfer that is 99% complete? Is this feature mapped to a real user requirement? But if it is needed, why not just use the WS-ResourceLifetime TerminationTime interface?	The element is nillable so that an end time does not have to be specified explicitly. The EndNoLaterThan element may be use to indicate an end time if a user deems to be necessary even if his transfer is 99.9% complete.
4.5		From section 5.2: Support for Stop or Suspend Not all transfer protocols will be able to support the stop or suspend operations. In such cases, I suppose the DTI state becomes failed:unclean or perhaps the service returns an exception . I am not clear on that.	
4.6		General comment: Extensions It seems like there will be a lot of usage of the any elements to extend the interface. For instance, I can see [transfer requirements] being used for GridFTP specific settings like streams and buffer size in order to tune GridFTP transfers. While the nterface specification appears well defined and it is good to limit its scope, in practice the extended, implementation-dependent elements could be significant and undermine the objective of interoperability. That's more of a concern than an objection.	

4.7		<p>General comment: Support for notifications</p> <p>I don't see support for notifications or is WS-Notification an interface that implementations of the specification can optionally support. As much as possible, it would be good for the specification to reduce the need for client roundtrips, e.g. getting a state notification instead of polling the DTI resource to check whether the transfer is done. In fact, adding a "callback" EPR in the create DTI call could be a good way to eliminate the need for the client to make a second remote call (following the first remote call to create the resource) to set up notifications, if notifications are to be supported.</p>	
5.1	Dave Berry	<p>Overall: This is an important specification that plays a key part in the OGSA Data Architecture as well as being used standalone. It has the support of several major players. The specification seems sensibly scoped for a version 1.0 and is well written.</p>	No action required.
5.2		<p>* Abstract</p> <p>"will be greatly reduced" -&gt; "is greatly reduced"?</p> <p>"leverage off" -&gt; "leverage"</p>	These comments seem ok, have made the changes accordingly.
5.3		<p>* Introduction (opening section, before 1.1)</p> <p>I think you need to clarify that the DMI mechanism transfers a copy of the original data, i.e. that whether the source retains or deletes its copy is outside the scope of this specification. (Cf. POSIX "mv" vs "cp").</p> <p>I think the introduction should mention that the user can optionally specify a preferred or suggested transport protocol, i.e. the automatic negotiation is the ideal but can be bypassed if the client wishes.</p>	It is always a copy - right?
5.4		<p>* Architecture</p> <p>First bullet points: the trailing "and" looks as if something has been omitted from that line. I assume the intention is to link to the second bullet point, but it does not read well. Perhaps a semicolon is needed before it?</p>	Have removed the "and".
5.5		<p>Section 3.3.3</p> <p>The DEPR, as described here, seems to contain similar functionality to a WS-Name. Can a DEPR be built using a WS-Name?</p>	
5.6		<p>Section 4.1.2</p> <p>I don't understand what is meant by "undo strategy identifiers look like URLs but they are not necessarily so". What is the characteristic of a URL beyond syntax that you are referring to? Why don't you just say that they are URIs?</p> <p>I also don't understand how the undo strategies relate to the "Failed:*" states. The "full" strategy says that cleanup is "guaranteed"; can this ever fail and leave the system in the "Failed:Unclean" or "Failed:Unknown" states? Conversely, can the "none" strategy ever leave the system in the "Failed:Clean" state?</p>	
5.7		<p>Section 5.2.7.1.1</p> <p>"dmi:InstanceAtrributes" -&gt; "dmi:InstanceAttributes"</p>	2
5.8		<p>Section 5.4.1.8</p> <p>"Failed:Unkown" -&gt; "Failed:Unknown"</p>	typo fixed
5.9		<p>Section 5.4.2</p> <p>It would be useful to repeat here that the mechanism for emitting LifeCycle Events is not defined in this version of the specification. When I read the document, I missed the initial explanation of events and was caught by surprise when I reached this section. Anyone jumping straight to this section wouldn't understand the context.</p>	
6	Shahbaz Memon	<p>I don't have much to comment. But I can say the specification is well explained and understandable.</p> <p>For implementation, Grid middlewares providing services based on WSRF, the DMI's (WSRF) rendering will significantly be worth noting. Lets see how it comes.</p>	

<del>7.1</del>	Erwin Laure	In general, this is a well written spec suitable as version 1.0.	No action required.
7.2		In Section 3, please be consistent with the names in the text and figure 1; for instance "Factory Port Type" of the figure should be "Data Transfer Factory Port Type" etc.	
7.3		Section 4.1.1. requires an undo strategy for each supported protocol which is only described later in Section 4.1.2 - I would recommend to include Section 4.1.2 in 4.1.1	
7.4		The specification of the different undy strategies in 4.1.2 should move to the XML representation (4.1.2.1) to be consistent with the discussion in 4.1.1	
7.5		Section 4.2.1.2 (source and sink DEPR) would gain by providing one or two examples, for instance, how to render a gridftp DEPR or an SRM DEPR	
<del>8</del>	Clive Davenhall	<p>The architecture developed by NextGRID (<a href="http://www.nextgrid.org/">http://www.nextgrid.org/</a>) relies on DMI for data transfer. It fulfils an important role in our architecture by providing a structure and interface for data transfer services. An important feature is that it separates the transfer resources from individual data transfer protocols, such as GridFTP, etc.</p> <p>It also provides the flexibility to support 'third-party' transfers, which are important for efficient implementations in Grid systems.</p> <p>Overall it is good stuff.</p>	No action required.
<del>9.1</del>	James Casey	Generally, I think the spec is well rounded and has the right scope for a 1.0 version of this subject.	No action required.
9.2		pg6, para 2 "record the protocols that can be used". I didn't understand this well first time reading it (the usage of the word 'record'). Do you mean "query for the protocols", "negotiate the protocols"	
<del>9.3</del>		pp12, 4.1.2.1 Typo dmi:UndoStrategy -> dmi:UndoStrategy	Fixed