

GWD-I
Category: Informational

Ann Chervenak, USC Information Sciences Institute
Ben Clifford, USC Information Sciences Institute
Karl Czajkowski, USC Information Sciences Institute
Robert Schuler, USC Information Sciences Institute

OGSA Data Replication Services Working Group

May 14, 2004

GGF DOCUMENT SUBMISSION CHECKLIST (include as front page of submission)	
	COMPLETED (X) - Date
1. Author name(s), institution(s), and contact information	(X) – May 14, 2004
2. Date (original and, where applicable, latest revision date)	(X) - May 14, 2004
3. Title, table of contents, clearly numbered sections	(X) - May 14, 2004
4. Security Considerations	(X) - May 14, 2004
5. GGF Copyright statement inserted	(X) - May 14, 2004
6. GGF Intellectual Property statement inserted.	(X) – May 14, 2004
7. Document format : WORD	(X) - May 14, 2004

Web Services Replica Set Specification (WS-ReplicaSet)

Status of This Memo

This memo provides information to the Grid community regarding the design of ReplicaSets for the Web Service Resource Framework (WS-RF). It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

Copyright © Global Grid Forum (2004) All Rights Reserved.

Abstract

We describe issues relating to the design of a Web Services Resource Framework (WS-RF) compliant specification for ReplicaSets. This design extends the WS-ServiceGroup specification and reflects recent work in the DAIS Working Group on the design of OGSA Data Services.

Contents

Abstract.....	2
Introduction	4
1. Background.....	4
1.1 WS-Resources and Web Services.....	4
1.2 WS-ServiceGroups	4
1.3 OGSA Data Services	5
2. Overview of ReplicaSet ServiceGroup	6
3. ReplicaSets.....	8
3.1 ResourceProperties	8
3.2 ReplicaSet: Operations	8
4. ReplicaSetEntry	9
4.1 ReplicaSetEntry: Resource Property Declarations	9
4.2 ReplicaSetEntry: Message Exchanges.....	9
5. ReplicaSetRegistration	9
5.1 ReplicaSetRegistration: Resource Property Declarations	9
5.2 ReplicaSetRegistration: Message Exchanges	10
5.3 Additional Message Exchanges	10
6. ReplicaSet Factories.....	11
7. Standard Error Modes	11
8. Examples	11
8.1 Creating a New ReplicaSet for Replicated Data Resources.....	11
8.2 Adding a New Entry to a Set of Replicated Data Resources.....	12
8.3 Destroying a ReplicaSetEntry	13
8.4 Destroying a ReplicaSet and All Associated Entries	13
9. Implementation Issues.....	14

9.1	Scalability of WS-RF Compliant ReplicaSets.....	14
9.2	Policy Enforcement	15
9.3	Persistence	15
9.4	Reliability	16
10.	Discovery of ReplicaSet Services via Metadata Services or Indexes	16
11.	Hierarchies of ReplicaSet WS-Resources.....	16
12.	Summary.....	16
13.	Security Considerations.....	16
	Author Information	17
	Intellectual Property Statement	17
	Full Copyright Notice	17
	References	18

Introduction

The OGSA Data Replication Services Working Group (OREP) is standardizing this specification for replicaSets, which associate replicated data items. This version of the specification replaces earlier specifications that were based on the Open Grid Services Infrastructure version 1.0 specification [1]. The current specification is based on the proposed standards for the Web Services Resource Framework [2,3] or WS-RF, including WS-Resource Properties [4], WS-Resource Lifetime [5], WS-ServiceGroup [6], and OGSA Data Services [7].

1. Background

1.1 WS-Resources and Web Services

In the WS-RF specifications [2,3], Web services are stateless. State is associated with WS-Resources. A WS-Resource Properties document [4] associates state with a WS-Resource. A WS-Resource is accessed via a Web service. Both Web services and WS-Resources are addressed using Endpoint references, which are XML elements that conform to the WS-Addressing specification [8]. An endpoint reference that refers to a WS-Resource includes the URL of the Web service through which the resource is accessed plus some arbitrary key information to distinguish the particular resource.

The WS-RenewableReferences specification is being developed to allow renewal of references to Web services and WS-Addresses that move within the hosting environment or among hosting environments.

1.2 WS-ServiceGroups

Our specification will extend the Web Services ServiceGroup Specification (WS-ServiceGroup) [6]. ServiceGroups are WS-Resources that aggregate information about one or more other WS-Resources or Web services. A WS-ServiceGroup WS-Resource contains entries that are also WS-Resources. These entries associate the Web Service Addresses (WS-Addresses) of member WS-Resources or Web services with a WS-ServiceGroup. These associations are illustrated using dashed lines in Figure 1. Because each ServiceGroupEntry is itself a WS-Resource, we can manage the lifetimes of these entries individually.

The WS-RF ServiceGroup is similar to the ServiceGroup previously defined in the Open Grid Services (OGSI) specification [1]. OGSI ServiceGroups were Grid services that aggregated information about one or more other Grid services [1]; ServiceGroup entries could be Grid services to allow lifetime management of individual entries.

Like all WS-Resources, a WS-ServiceGroup is accessible via a Web service, as shown in Figure 1. A single Web service may provide access to multiple WS-ServiceGroup WS-Resources. Similarly, a single Web service may provide access to multiple ServiceGroupEntry WS-Resources. To access a WS-ServiceGroup, a client must know the WS-Address endpoint reference of the Web service and additional information required to identify a particular WS-Resource accessed through the service.

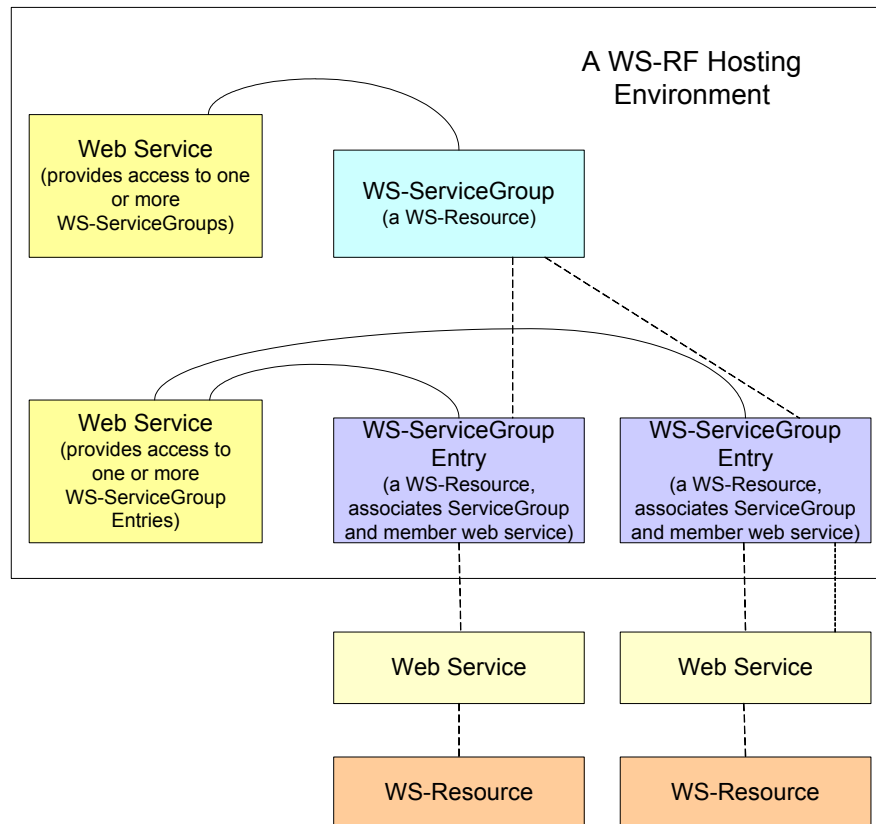


Figure 1: A WS-RF ServiceGroup

1.3 OGSA Data Services

The replicated data items that we register in this specification using replicaSets are OGSA Data Services. An OGSA data service [7] is a Web service that is used to access one or more underlying WS-Resource-compliant [4] resources. The following discussion summarizes relevant information from the OGSA Data Services specification [7] being developed through the DAIS Working Group.

The WS-Resource being accessed through an OGSA Data Service has a WS-Resource Properties document that describes key parameters of a *data virtualization*, which is an abstract view of some data available through the resource. For example, a WS-Resource Properties document might describe the schema of a relational database, a file system including its directories and file names, or the properties of a particular file. We can introspect on a WS-Resource properties document to discover the properties of a data virtualization. A data virtualization is defined by operations and attributes on the data [7].

An OGSA data service implements one or more of three interfaces. The Data Access interface provides operations to access or modify the contents of a data virtualization represented by a data service. The Data Factory interface provides the ability to create a new data service whose data virtualization is derived from the virtualization of a parent data service. The Data Management interface provides the ability to manage and monitor a data service's data virtualization.

An OGSA data service has a name, which is a WS-Addressing endpoint reference [8]. In general, this name does not provide a globally unique identifier for data, but the name is typically unique

within a particular namespace. All the interfaces for the OGSA Data Service extend the WS-Agreement [9].

Any data item (file, file system, database, etc.) for which we want to manage replicas through a replicaSet must be exposed as a WS-Resource available through an OGSA Data Service.

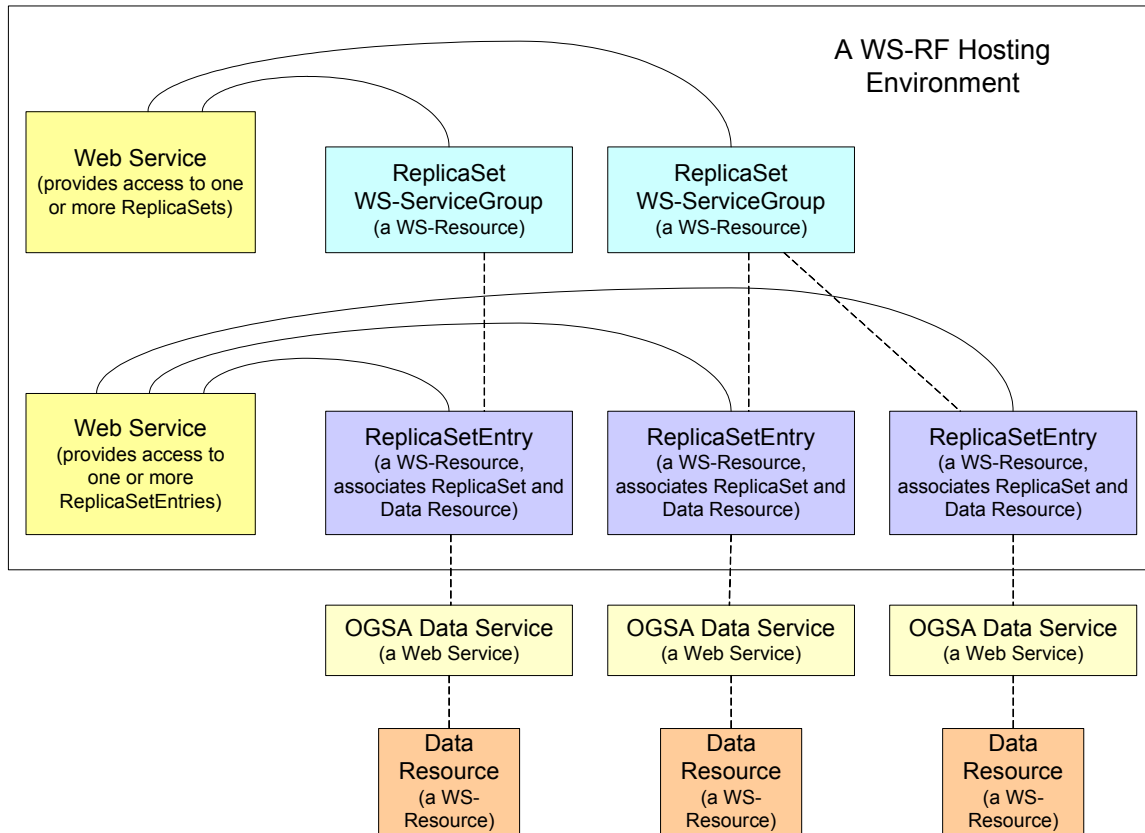


Figure 2: Shows ReplicaSets in the WS-RF environment. Both ServiceGroups and ServiceGroup Entries are WS-Resources. Web Service in the hosting environment provide access to one or more ServiceGroups or ServiceGroup entry resources. Data services that provide virtualizations of one or more Data resources may be in the same or in different hosting environments.

2. Overview of ReplicaSet ServiceGroup

Figure 2 shows replicaSets in the WS-RF environment. Aspects of a WS-RF replicaSet include the following:

- Replicated data items form a set that meets some semantic definition of replication. The *semantics* of replication and *policies* for maintaining consistency among replicas can be associated with a replicaSet.
- A replicaSet will be based on and extend the WS-ServiceGroup specification.
- A WS-ServiceGroup is a WS-Resource.
- The replicaSet ServiceGroup contains WS-ServiceGroupEntries called replicaSetEntries. These entries are WS-Resources. Each replicaSetEntry describes an association between the replicaSet and one or more members of the replicaSet. These associations are illustrated by dashed lines in Figure 2.

- The members of the replicaSet are Data Resources, which are WS-Resources. These resources are accessed via OGSA Data services, which are Web services that provide a virtualization of a data resource.
- Data services and associated resources may be in the same hosting environment as the replicaSets or in a remote hosting environment, as shown in Figure 2.
- Like all WS-Resources, replicaSets and replicaSetEntries are accessed via a Web service. A single Web service can provide access to multiple replicaSets or multiple replicaSetEntries. To address a ReplicaSet (or ReplicaSetEntry), the user must provide the WS-Address endpoint reference for the Web service as well as enough information to refer to the particular WS-Resource.
- The replicaSet ServiceGroup may have associated policies for authorization (who is allowed to add members to the replicaSet) and semantics (what constitutes a member of the replicaSet). These policies are associated with the replicaSet WS-Resources through the WS-Resource Properties document.
- There may be indexes for aggregating information about multiple replicaSet ServiceGroups. These indexes should also be designed as WS-ServiceGroups.

Figure 3 shows a UML diagram depicting the relationships among the OREP specification being defined here and other WS-RF and DAIS specifications.

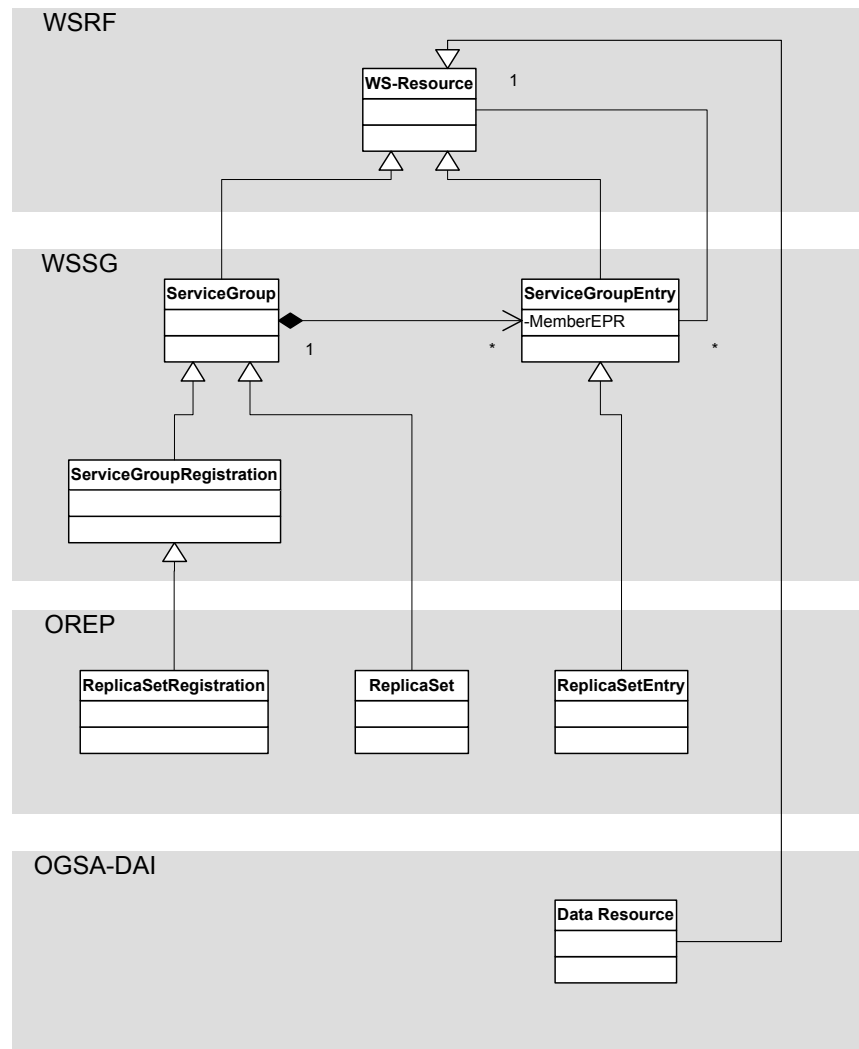


Figure 3: UML depiction of relationships among OREP and other specifications.

3. ReplicaSets

ReplicaSets extend WS-ServiceGroups [6].

3.1 ResourceProperties

According to the WS-ServiceGroup specification, the replicaSet MUST support the required message exchanges from the WS-ResourceProperties specification and MAY support optional message exchanges. The replicaSet resource property document MUST include the MembershipContentRule Resource Property and the Entry Resource Property.

The resource property document contains a potentially empty set of MembershipContentRule elements. These rules specify constraints on the membership of the ServiceGroup, for example, that members must implement a particular interface. If no MembershipContentRule elements are specified, then members of the ServiceGroup are not constrained.

The EntryResourceProperty aggregates information about the resource property documents of the replicaSetEntry resources. From the WS-ServiceGroup specification [6], an Entry Resource Property has the following components: the WS-Address endpoint references for the ServiceGroupEntry, for the member service, and for arbitrary content associated with the entry.

```
<wssg:Entry>
  <wssg:ServiceGroupEntryEPR>
    was:EndpointReferenceType
  </wssg:ServiceGroupEntryEPR>
  <wssg:MemberServiceEPR>
    was:EndpointReferenceType
  </wssg: MemberServiceEPR>
  <wssg:Content>
    {any}
  </wssg:Content> ?
</wssg:Entry>
```

To find the members of a replicaSet, a client can introspect on the values of the replicaSet's Entry Resource Properties. To perform this introspection, the client would use the GetResourceProperty, GetMultipleResourceProperties and QueryResourceProperties message exchanges defined in the WS-Resource Properties specification.

For the replicaSet extension of the WS-ServiceGroup, we will define additional resource properties that describe the policies that the replicaSet ServiceGroup supports. For example, these policies would specify what authorization is enforced by a particular replicaSet implementation, what replica semantics are associated with a replicaSet and how consistency among replicas is maintained, and the persistence and reliability characteristics supported by a replicaSet implementation.

3.2 ReplicaSet: Message Exchanges

The replicaSet interface defines no message exchanges. The replicaSet SHOULD implement one of the message exchanges from WS-ResourceLifetime if it needs to support immediate or scheduled resource destruction.

4. ReplicaSetEntry

ReplicaSetEntry extends WS-ServiceGroupEntry [6]. ReplicaSetEntry is a WS-Resource that represents a member within the replicaSet. The replicaSetEntry interface describes the requirements on the Web service through which entry management occurs.

Unlike in the WS-ServiceGroup specification, a member MUST NOT appear multiple times in a replicaSet.

A replicaSetEntry MUST belong to exactly one replicaSet.

A replicaSetEntry interface MAY provide additional management functions for a replicaSetEntry WS-Resource. It MAY provide independent lifetime management functions for replicaSetEntry WS_resources. If the replicaSetEntry Web service implements the message exchange sets defined in WS-Resource Lifetime, a replicaSetEntry MAY be removed from a replicaSet by managing the lifetime of the replicaSetEntry.

4.1 ReplicaSetEntry: Resource Property Declarations

The WS-ServiceGroup specification includes three resource property elements for ServiceGroupEntries [6]. We extend these for the replicaSetEntry.

The replicaSetEPR contains the WS-Addressing endpoint reference for the replicaSet for which this entry represents membership.

The memberEPR contains the WS-Addressing endpoint reference for the member service to which this entry pertains.

The content resource property element contains information associated with the replicaSetEntry. This content must conform to the membership content rule of the replicaSet containing this replicaSetEntry.

4.2 ReplicaSetEntry: Message Exchanges

The replicaSetEntry defines no operations. The replicaSetEntry SHOULD implement one of the message exchanges from WS-ResourceLifetime if it needs to support immediate or scheduled resource destruction. The replicaSetEntry interface SHOULD implement message exchanges and resource properties for the NotificationProducer interface.

5. ReplicaSetRegistration

ReplicaSetRegistration extends the ServiceGroupRegistration interface defined in the WS-ServiceGroup specification. ReplicaSetRegistration defines the message exchanges that allow a requestor to add entries to a replicaSet explicitly.

5.1 ReplicaSetRegistration: Resource Property Declarations

ReplicaSetRegistration defines no resource properties. The resource properties defined by the interfaces in WS-ResourceLifetime SHOULD be included in the resource property document of a replicaSetRegistration. The resource properties defined in the replicaSet interface MUST be included in the resource property document of a replicaSetRegistration.

5.2 ReplicaSetRegistration: Message Exchanges

The WS-ServiceGroup specification [6] describes the ServiceGroupRegistration interface's Add message, which includes arguments for the endpoint reference (WS-Address) of the member service to be added, a content field that contains information that should be associated with the member endpoint reference in the ServiceGroup, and an initial termination time for the ServiceGroupEntry WS-Resource.

We will extend the Add message of the ServiceGroupRegistration interface to provide additional arguments needed for ReplicaSet policy enforcement. For example, a particular ReplicaSet implementation might allow a client to assert a checksum value for a data item being added to a ServiceGroup. This checksum would be matched against the required checksum for a ReplicaSet before the Add operation was allowed to succeed. In the extended ServiceGroupRegistration Add message below, we include a replicaSetContent component that allows the replicaSet client to provide information specifically required for semantic or other policy enforcement.

```
<wssg:Add>
  <wssg:MemberEPR>
    wsa:EndpointReferenceType
  </wssg:MemberEPR>
  <wssg:Content>
    {any}
  </wssg:Content>
  <wsrs:replicaSetContent>
    {any}
  </wsrs:replicaSetContent>
  <wssg:InitialTerminationTime>
    xsd:dateTime
  </wssg:InitialTerminationTime>?
</wssg:Add>
```

If the ServiceGroupRegistration accepts the request to add a member to the ServiceGroup after performing policy enforcement, then it returns the following AddResponse message, which contains the EndpointReference of the ServiceGroupEntry WS-Resource that is created by the association of the member with the ServiceGroup..

```
<wssg:AddResponse>
  <wsa:endpointReferenceType>
</wssg:AddResponse>
```

WS-ServiceGroupEntries are deleted in two ways: either by an explicit delete message defined for the ServiceGroupEntry or because entries are removed when their lifetimes expire.

Policy enforcement, including authorization (who is allowed to add an entry to a ServiceGroup) and semantics (what is a valid replica Data Service), will be enforced by a particular ReplicaSet ServiceGroup implementation.

5.3 Additional Message Exchanges

We will provide an explicit Delete message exchange for entries based on the WS-Resource Lifetime specification [5].

In addition, the group needs to determine whether bulk add/delete operations should be supported in this specification. These are probably desirable for good performance of the

ReplicaSet for applications that manipulate large numbers of data objects, such as workflow managers.

6. ReplicaSet Factories

A replicaSet factory is required to create new replicaSets. A factory will create replicaSets that support particular policy assertions for authorization and replica semantics. A replicaSet factory is a WS-Resource whose ResourceProperties specify what kinds of replicaSets it can create.

A replicaSet factory imposes authorization restrictions on ReplicaSet creation. It determines whether the client who requested a replicaSet creation is allowed to perform that operation. The factory will use standard WS-RF security techniques for authentication and authorization.

The replicaSet factory will define a message exchange for the Create operation. The create message may include content related to policy enforcement, for example, a checksum that will be required for new members to be added to a replicaSet. The response to the create message will contain the endpoint reference of the newly-created replicaSet ServiceGroup WS-Resource.

```
<wsrsf:Create>
  <wsrsf:Content>
    {any}
  </wsrsf:Content>
</wsrsf:Create>

<wsrsf:CreateResponse>
  <wsa: endpointReferenceType
</wsrsf: CreateResponse>
```

7. Standard Error Modes

Still to be determined within the OREP Working Group is whether we should standardize on particular error modes for ReplicaSet operations. This is probably necessary. We will define a set of typical error modes that are based on the WS-BaseFaults specification [10].

An additional issue that should be addressed is how errors are reported that are not related to particular operations. For example, if a replicaSet is maintaining consistency among replicas, how does it inform a client or a subscriber that a replica has become inconsistent?

8. Examples

The following examples are provided to assist the reader in understanding the behavior expected in compliant replication systems for certain key activities. The activities described here include creation and destruction of ReplicaSet and ReplicaSetEntry resources.

8.1 Creating a New ReplicaSet for Replicated Data Resources

In Figure 4, a new ReplicaSet resource is created by an unspecified caller. Adhering to the implied usage pattern [2,3], the caller utilizes a factory to create a new ReplicaSet resource. The exact form of the factory is beyond the scope of this specification and may be considered implementation-specific. The factory is responsible for creating the ReplicaSet resource and

properly initializing its corresponding Resource Properties [4]. The result returned to the caller is an Endpoint Reference (EPR) [8], which effectively points to the ReplicaSet resource.

It may be of interest to note that when creating a new ReplicaSet, values may be passed by the caller to initialize the ReplicaSet resource's Resource Properties.

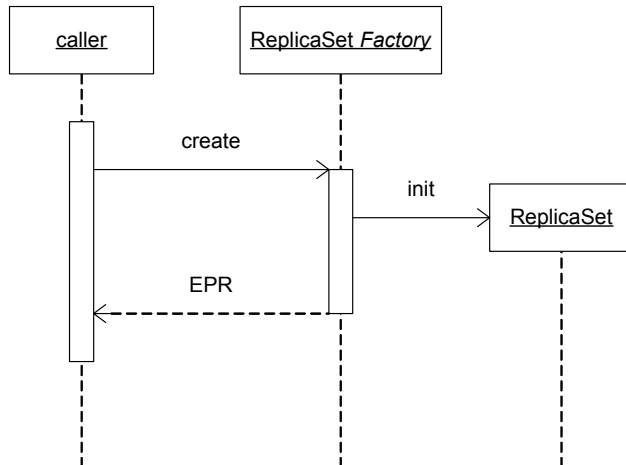


Figure 4: Illustrates creation of a ReplicaSet by employing a factory pattern.

8.2 Adding a New Entry to a Set of Replicated Data Resources

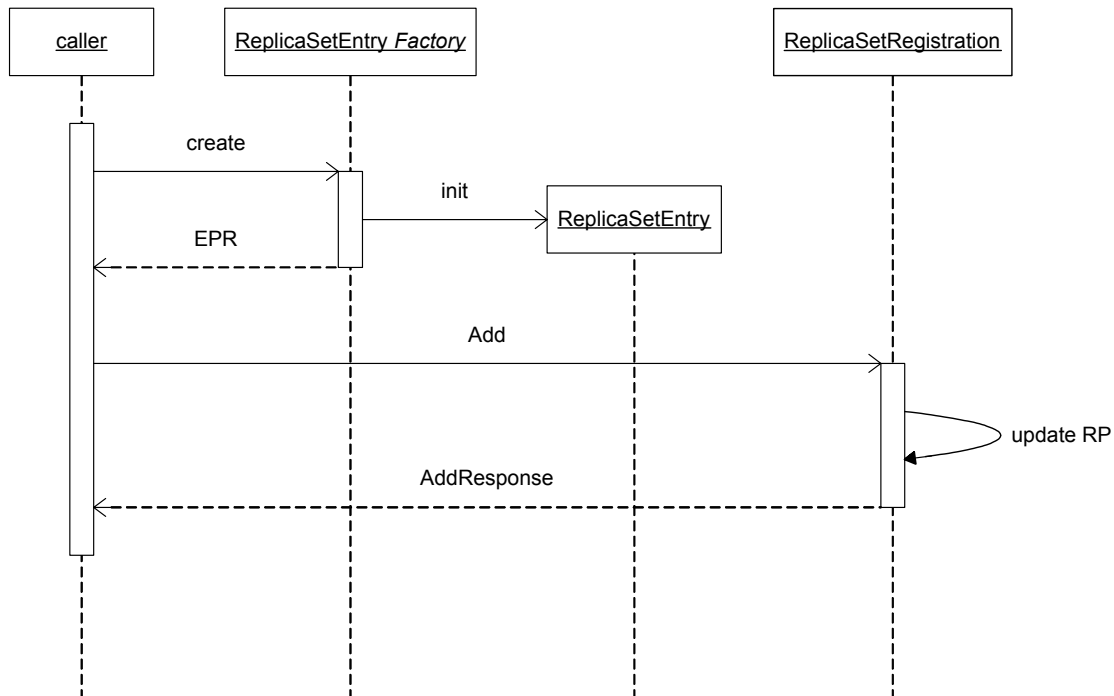


Figure 5: Illustrates creation of ReplicaSetEntry resources and adding them to a ReplicaSet by using the specified ReplicaSetRegistration resource.

In Figure 5, a new ReplicaSetEntry is created and added to an existing ReplicaSet. As with creating a ReplicaSet, a factory pattern may be provided as a means for a caller to create a new instance of a ReplicaSetEntry resource and to yield an EPR to the new resource. The caller then

invokes the Add message exchange [6] passing the Entry's EPR to the ReplicaSetRegistration resource. The ReplicaSetRegistration then updates the corresponding ReplicaSet resource's Resource Properties [6] to identify the new Entry in the set. The ReplicaSetRegistration concludes with the AddResponse message [6] to the caller.

8.3 Destroying a ReplicaSetEntry

In Figure 6, a caller destroys the K^{th} entry of a given ReplicaSet by using the ReplicaSetEntry resource's Destroy message exchange. Again the entry's EPR is used to identify it. Upon receipt of the Destroy message the ReplicaSetEntry informs the ReplicaSet of its impending destruction so that the ReplicaSet may update its Resource Properties to reflect the elimination of the entry, thus avoiding a dangling reference to the soon-to-be defunct entry. The entry then performs any necessary cleanup and issues a DestroyResponse message to the caller. The precise mechanism used to notify the ReplicaSet of the removal the entry before its destruction needs to be further defined. It is most likely within the scope of this specification.

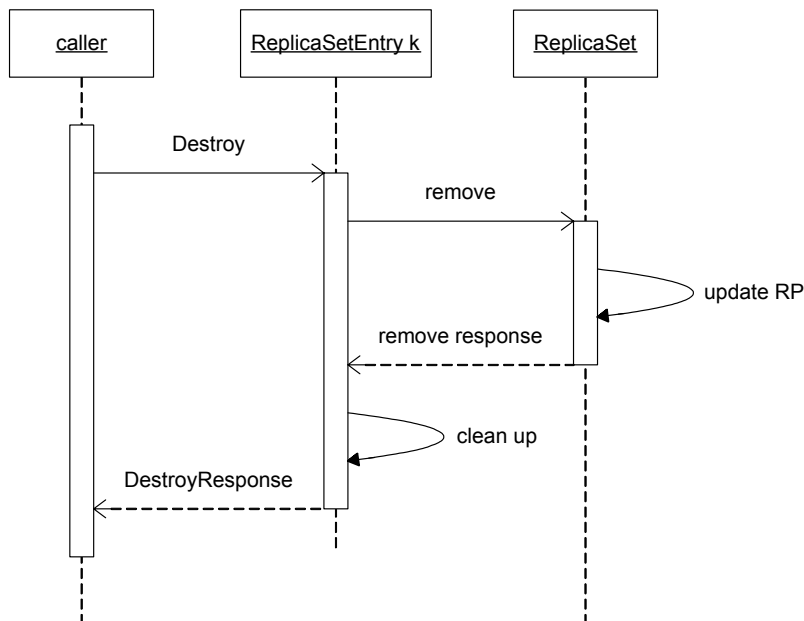


Figure 6: Illustrates destruction of a ReplicaSetEntry and removal from its associated set.

8.4 Destroying a ReplicaSet and All Associated Entries

In Figure 7, the caller initiates the immediate destruction of a ReplicaSet resource via the Destroy message exchange [5]. The ReplicaSet is identified using an EPR [9]. Upon receiving the Destroy message, the ReplicaSet enacts destruction of each of its entries, as recommended in [6]. The ReplicaSetEntry resources accept the Destroy message, perform any necessary cleanup and respond with the DestroyResponse [5]. Once all Entries are destroyed the ReplicaSet resource performs any necessary cleanup and issues a DestroyResponse to the caller. It can be expected that shortly after a terminated resource sends the DestroyResponse message, it will cease to exist. EPR's to destroyed resources should be promptly discarded.

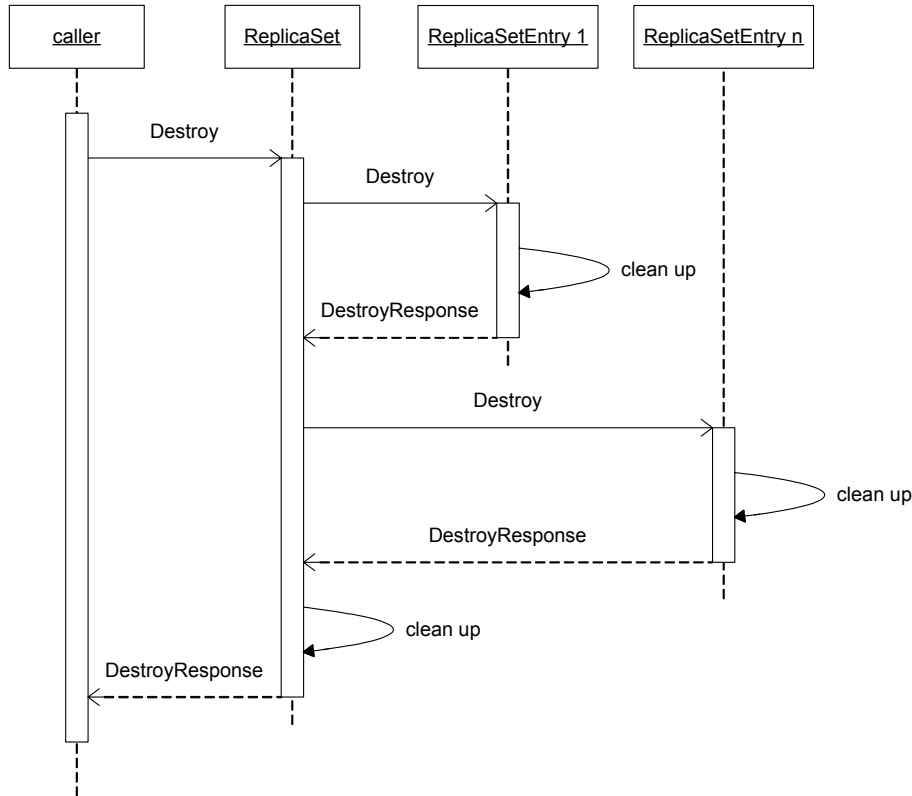


Figure 7: Illustrates destruction of a ReplicaSet and the subsequent destruction of all of the set's associated entries.

9. Implementation Issues

We address some general implementation issues, although these are outside the scope of the interface specification being developed by the OREP working group.

9.1 Scalability of WS-RF Compliant ReplicaSets

The separation of Web Services and WS-Resources is intended to provide greater scalability for WS-Resources and ServiceGroups than was possible in the OGSi environment. The WS-RF specification does not mandate any particular implementation for WS-Resources or that WS-Resources consume any resources in the hosting environment. For example, WS-resources might be implemented as rows in a relational database.

By contrast, every OGSi Grid service consumed approximately 100 Kbytes of memory in the hosting environment. This severely limited the number of OGSi-compliant ReplicaSet ServiceGroups that could be deployed in an OGSi hosting environment. Thus, it is unlikely that the previous ReplicaSet service specification could scale to support replicaSets for millions of replicated data objects.

In the WS-RF-based design, the scalability will be determined by the scalability of the WS-Resource implementation, for example, by the scalability of a back end database that implements the WS-Resources.

9.2 Policy Enforcement

A replicaSet ServiceGroup implementation provides a convenient point for enforcing policies such as access control and replica semantics. The policy enforcement will be specific to a particular implementation of the replicaSet.

First, the replicaSet ServiceGroup can enforce authorization policies about who is allowed to add new data services (or remove data services) as members of the equivalence set. The replicaSet ServiceGroup will only allow clients with whom it has an appropriate trust relationship to perform add operations.

Second, the replicaSet ServiceGroup can enforce semantic policies regarding the meaning of replication and which data services are allowed to be added as members of a replicaSet. For example, the replicaSet could support policies that replicas can only be added to the ReplicaSet if the data service is an exact copy of the replicas in the ReplicaSet or is a version within a certain range of allowed versions of the data. Some possible examples of standard semantic policies for replicas include the following:

- Byte-for-byte copy of data items, such as files
- Data objects that contain the same information in different formats
- Data objects that are equivalent to a specified degree
- Data objects that are derived from a common parent
- Versions of data objects
- Replicas that have been synchronized within a specific time period
- Partial replicas of data objects

The extent to which the assertions about replica semantics are verified or enforced depend on the replicaSet ServiceGroup implementation. Depending on the implementation, these assertions may or may not be enforced at the time when members are added to the replicaSet, and these assertions may or may not be maintained in response to changes in the content of OGSA data services that are members of the replicaSet. One option is to implement replicaSet services that maintain relationships among members of the replicaSet, such as byte-for-byte copy replication. We could subscribe to be notified of changes in member data services and then propagate these changes among replicas according to a particular coherency scheme. An alternative is to periodically introspect on the members of the replicaSet service to check coherence and remove non-complying members from the replicaSet equivalence class.

Finally, the replicaSet ServiceGroup can enforce policies about what attributes may be associated as WS Resource Properties of the replicaSet ServiceGroup or the ServiceGroupEntry resources.

9.3 Persistence

An important implementation issue for replicaSet ServiceGroups and ServiceGroupEntries will be persistence. Information about replicas should not be lost because of a hosting environment or other software or hardware failure. We assume that these WS-Resources will be stored in a persistent. The persistence scheme will depend on how ServiceGroup and ServiceGroupsEntries are implemented. For example, if these are implemented as rows in a relational database or as files or entries within a file, then we could use standard persistence and backup schemes for databases or file systems to guarantee the persistence of replica information.

9.4 Reliability

A desirable characteristic of a replicaSet is that it provide reliable operations. For example, replicaSetRegistration Add operation fails before successfully updating the replicaSet (for example, because of a container failure), we would like to maintain enough state about the outstanding Add operation that it could be restarted when the replicaSet service resumes operation. The amount of reliability provided by a replicaSet is dependent on the implementation. Resource properties that describe the reliability characteristics of a particular replicaSet implementation should be included in the replicaSet resource properties document.

10. Discovery of ReplicaSet Services via Metadata Services or Indexes

Since a replicaSet can only be addressed by its WS-Address and not by a human-readable name, we assume that there are other mechanisms in the overall service architecture by which logical names can be associated with replicaSets. A typical configuration would be to deploy a metadata service that associates descriptive attributes including a logical name with the identifier for a ReplicaSet. The specified descriptive attributes would apply to all replicas associated by a replicaSet WS-Resource.

A Web service environment may also contain a variety of index services that aggregate information about replicaSets and provide discovery and query capabilities.

Metadata and index services used for replicaSet discovery are outside the scope of this specification.

11. Hierarchies of ReplicaSet WS-Resources

We have indicated in Figure 2 that replicaSets and their associated entries are available via web services in a particular hosting environment. It may also be desirable to define a hierarchy of replicaSets, for example, a hierarchy that spans multiple Web service hosting environments. In this case, the members of a replicaSet could either be OGSA Data Services or they could be existing replicaSet WS-Resources. The current version of the specification does not define replicaSet hierarchies, but this would be good to include in future versions.

12. Summary

We have presented a design for ReplicaSet ServiceGroups based on the WS-ServiceGroup, WS-Resource and OGSA Data Service specifications. The ReplicaSet ServiceGroup is a WS-Resource that aggregates information about replica Data Services using ServiceGroup entries that are also WS-Resources. The design of the replicaSet extends WS-ServiceGroups. We have discussed extensions that are needed to ServiceGroups to specify replicaSet policies such as authorization and replica semantics. We have noted that enforcement of these policies will be provided by the implementation and will not have an impact on the interface specification.

Further refinement of this design will occur through subsequent meetings of the GGF OREP Working Group.

13. Security Considerations

This discussion relates to WS-RF-compliant replicaSet ServiceGroups for maintaining replica location information. Therefore, our service will have all the same security capabilities and issues as other WS-ServiceGroups. Additional security considerations such as access control over creation of replica mappings are discussed above.

Author Information

Ann L. Chervenak, USC Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292, USA, annc@isi.edu

Ben Clifford, USC Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292, USA, benc@isi.edu

Karl Czajkowski, USC Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292, USA, karlc@isi.edu

Robert Schuler, Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292, USA, schuler@isi.edu

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

Copyright (C) Global Grid Forum (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

References

- [1] Steve Tuecke, et. al, "Open Grid Services Infrastructure (OGSI) Version 1.0", Global Grid Forum Open Grid Services Infrastructure (OGSI) Working Group, June 27, 2003, <https://forge.gridforum.org/projects/ogsi-wg>.
- [2] Ian Foster, et. al, "Modeling Stateful Resources with Web Services version 1.0", <http://www-106.ibm.com/developerworks/library/ws-resource/-ws-modelingresources.pdf>
- [3] Karl Czajkowski, et. al, "The WS-Resource Framework Version 1.0", March 5, 2004, <http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf>.
- [4] Steve Graham, et. al, "Web Services Resource Properties (WS-ResourceProperties) Version 1.1", March 5, 2004, <http://www-106.ibm.com/developerworks/library/ws-resource/ws-resourceproperties.pdf>.
- [5] Jeffrey Frey, et. al, "Web Services Resource Lifetime (WS-ResourceLifetime) Version 1.1", March 5, 2004, <http://www-106.ibm.com/developerworks/library/ws-resource/ws-resourcelifetime.pdf>.
- [6] Steve Graham, et. al, "Web Services Service Group (WS-ServiceGroup) Version 1.0", March 31, 2004, <http://www-106.ibm.com/developerworks/library/ws-resource/ws-servicegroup.pdf>.
- [7] Ian Foster, et. al, "OGSA Data Services", February 24, 2004, Global Grid Forum DAIS Working Group, <http://forge.ggf.org/projects/dais-wg>.
- [8] Adam Bosworth, et. al, "Web Service Addressing (WS-Addressing)", March 2004, <ftp://www6.software.ibm.com/software/developer/library/ws-add200403.pdf>.
- [9] Alain Andrieux, et. al, "Web Service Agreement Specification (WS-Agreement) Version 1.0", February 21, 2004, Global Grid Forum Grid Resource Allocation Agreement Protocol (GRAAP) Working Group, <https://forge.gridforum.org/projects/graap-wg>.
- [10] Steve Tuecke, et. al, "Web Services Base Faults (WS-BaseFaults) Version 1.0", March 31, 2004, <http://www-106.ibm.com/developerworks/library/ws-resource/ws-basefaults.pdf>.