

# 1 Strawman requirements for ACS discussion

2 This document describes strawman requirements for ACS discussion.

## 3 1 Functional Requirements

### 4 1.1 Everything required for a task in a single archive

5 Background:

6 In order to install and operate complex systems, a number of various application related  
7 information must be specified, especially for three-tier systems or Grid tasks. Keeping everything  
8 required for a task into a single archive will contribute to ease of use, reduction of the administrative  
9 cost and higher consistency. Examples of such information are the following:

- 10 • Program binaries and initial data: These are the main contents of the systems and/or the Grid  
11 tasks.
- 12 • Middleware and/or OS: These may be required for building execution environments in a data  
13 center required for Grid task execution.
- 14 • Policy rules: OGSA services for task execution management use this information to manage  
15 Grid tasks dynamically in case of, for example, erroneous conditions and/or increasing load.
- 16 • Fail-over and/or load-sharing configuration: It is an important Grid requirement to be able to  
17 cope with disaster recovery and load-sharing among geographically separated sites.
- 18 • Scheduling information or operational policy rules, etc.

19 Merits:

- 20 • (Ease of use) Business activity manager of a complex system can handle application contents  
21 easily, resulting in less human errors.
- 22 • (Administrative cost reduction) Administrative costs will be reduced by simplifying the  
23 handling procedures of application contents.
- 24 • (Consistency) Consistency of application contents can be validated more easily if they are in a  
25 single archive.

26 Notes:

27 The first and second items are among the main purposes of OGSA, and described as OGSA  
28 requirements in section 2.11 (Ease of Use and Extensibility) and 2.8 (Administrative Cost  
29 Reduction) of OGSA v1.0 [OGSA v1.0].

30 Requirements:

- 31 • [1.1-a] AAF/ARI: Application contents must be logically bundled in a single and consistent  
32 archive.
- 33 • [1.1-b] AAF: ACS should allow diversity of information as its contents, including application,  
34 middleware, OS, or firmware.
- 35 • [1.1-c] AAF: ACS should allow bundling a set of applications which are to be deployed and  
36 executed in geographically and/or administratively separated sites.

1 **1.2 Everything required for Grid in a single archive**

2 Background:

3 Grid application ready to be executed comprises various files such as the descriptive information to  
4 be used by the various services for scheduling and resource management. To operate the system  
5 more efficiently and automatically, those files need to be presented in a standard format. Examples  
6 of such information are the following:

- 7 • Configuration Description, Deployment and Lifecycle Management information, presented as  
8 CDL.
- 9 • Resource requirement description for hardware and/or software, presented as JSDL or WS-  
10 Agreement.
- 11 • Installable Unit Package Format (IUPF) being developed by the OASIS Solution Deployment  
12 Descriptor (SDD) TC.

13 Merits:

- 14 • (Completeness of the contents) Itemizing and collecting the requisite information in an archive  
15 contribute to clarify the completeness of the required information both for user and system.

16 Requirements:

- 17 • [1.2-a] AAF: All information required for Grid task execution must be logically bundled in a  
18 archive.
- 19 • [1.2-b] AAF: SDD shall be extended by using an extension point <xsd:ANY> in order to  
20 include ACS namespace elements such as owner, EPR, access rights, etc.

21 Notes:

- 22 • Requirement [1.2-a] above is relevant to requirement [4-d] in chapter 4 .

23 **1.3 A service comprising a part of systems**

24 Background:

25 The ACS will be a service comprising a part of systems, such as a Grid system. An archive  
26 repository can facilitates a stable source of the archive to collaborate with other services in the  
27 system. Complex system composed from multiple applications is at risk for troubles due to missing  
28 and/or inconsistency of application contents. It will execute a consistency check, especially at the  
29 time of modification of the application.

30 Changes, upgrades and bug-fixes of the application are envisioned during the lifecycle of  
31 application archive. The archive repository facilitates lifecycle management of application contents  
32 including version control, distributed deployment such as copying and relaying a software stack onto  
33 multiple machines when sharing application archives across administrative domain.

34 Merits:

- 35 • (Service Integration) Storing the contents of the Application Contents in a repository and  
36 providing to other services contributes to the system efficiency in executing a task.

- 1 • (Lifecycle management support) Version management, if combined with repository  
2 management, can reduce the cost of lifecycle management of application.

3 Requirements:

- 4 • [1.3-a] ARI: Archives must be stored in a repository, with a standard set of the CRUD  
5 functions, i.e. Create, Read, Update, and Delete. **Once registered, operations must be allowed**  
6 **in a smaller unit in the archive.**
- 7 • [1.3-b] ARI: Repository Interface must provide reference handles, with appropriate access  
8 control management for actors in both inside and outside of the system.
- 9 • **[1.3-c] AAF/ARI: Incremental upgrading of the application archive must be supported.**
- 10 • [1.3-d] AAF: Consistency check of the archive contents should be assisted through the ARI.

11 Notes:

- 12 • Further investigation may be desired as to in which grain size archive contents are retrieved.
- 13 • ACS assumes that there is an overall grid management which ensure that ACS and other  
14 resources are available to support the entire system. For example, an archive should not be  
15 deleted when it is in use. ACS is not responsible for this.

#### 16 **1.4 Reuse of an existing archive**

17 Background:

18 Considering situations such as repeating the same calculation with different parameters or  
19 performing similar services, in parallel or in different points of time, a business activity manager  
20 may feel convenient if the system allows creating a new task reusing the content for a running task  
21 or its subset. ACS should enable to create a new task using an existing archive in the repository that  
22 is previously registered for a similar task. A business activity manager may want to apply some  
23 updates on the archive to modify a task or to create a different sort of task. A business activity  
24 manager may also want to use the archive created by another business activity manager besides  
25 improving his/her own archive.

26 Merits:

- 27 • (Cost reduction) Reuse of an archive can reduce the development/administrative cost when  
28 creating similar or repetitive tasks.
- 29 • (Incremental improvement) Reuse of an archive enables incremental improvement of  
30 functionality and/or reliability.
- 31 • (Ease of use) Reuse of an archive contributes to the over all ease of use for the systems

32 Requirements:

- 33 • [1.4-b] ARI: Retrieving an archive by multiple different actors, under appropriate access  
34 control to the archive, must be allowed.

1 **2 Non-functional Requirements (or Design Goal)**

2 **2.1 Exchangeability, Interoperability**

3 Background:

4 An application archive is expected to be used for various administrative domains and/or  
5 heterogeneous environments. In order to address to diversity of the environments, standard  
6 specification needs to define a reliable universal format and/or interface which ensure reusability and  
7 exchangeability.

8 Requirements:

- 9 • [2.1-a] AAF/ARI: Standard format of archive and interface to the repository needs to be  
10 defined.

11 Notes:

- 12 • Further investigation should be performed as to at what level and to what extent the  
13 interoperability.  
14 • The following requirement (2.2 Extensibility) assumes this requirement.

15 **2.2 Extensibility**

16 Background:

17 Specification should allow incremental evolution of itself, to be adaptable for future technologies  
18 that are under development or not predicted today, including those in GGF. In addition, specification  
19 is desired to allow broader range of the industry to share its single framework. Application archives  
20 should accommodate various use cases, such as application installation on a single machine, copying  
21 a software stack onto multiple machines, and task execution on a Grid system. It should allow for  
22 diversity of system implementations to handle the application archives.

23 Requirements:

- 24 • [2.2-a] AAF/ARI: In addition to minimum set of requirements, ACS must allow optional or  
25 extension sets.

26 **2.3 Making use of the external ingenuity**

27 Background:

28 There are and will be ingenuity in the world to implement the feature, while satisfying the common  
29 set of the requirements and interface specifications. The specification should make full advantage of  
30 the external ingenuity in the field.

31 Requirements:

- 32 • [2.3-a] AAF/ARI: ACS specification will not define how to implement; it should provide  
33 common interfaces and mechanisms for implementing.

34 **2.4 Efficiency**

35 Background:

1 Complex application comes to be a larger set of files and data. In order to address to this, the  
2 references to the external storage that is persistent and stable should be allowed in the both archive  
3 and repository. Efficiency in data storage and efficiency in transport is also expected. Transport on  
4 network is expected in registration and retrieval of an archive file. ACS should seek to method to  
5 take advantage of the state of the art of various technologies in both storage and transport of archives.

6 Requirements:

- 7 • [2.4-a] AAF: To reduce the size of archives, reference to external storage/files should be  
8 allowed in them.
- 9 • [2.4-b] AAF/ARI: ACS must provide mechanisms for minimizing the network traffic by  
10 allowing differential application archives.
- 11 • [2.4-c] ARI: When managing contents of archives in the repository, mechanisms for  
12 eliminating redundancy and minimizing the disk usage should be allowed.
- 13 • [2.4-d] ARI: To make use of the efficient transport technologies available, the third party  
14 transports must be allowed in the repository interface.
- 15 • [2.4-e] ARI: ACS should allow ARI implementers to select a communication protocol for  
16 transport.

17 Notes:

- 18 • When allowing reference to external entities, consistency of application contents must be  
19 considered.

### 20 3 Security

21 Background:

22 Business/commercial applications may include confidential and/or critical information, and  
23 integrity of contents is vital.

24 Requirements:

- 25 • [3-a] ARI: Security features must be deployable according to the business requirements.
- 26 • [3-b] ARI: Appropriate access control to archives is required.
- 27 • [3-c] AAF/ARI: Secure transport of archives must be realized. For example, mechanisms for  
28 keeping confidentiality of archive contents and detection of alteration and spoofing are  
29 required.

30 The security design of ACS follows that of OGSA-WG Security Design Team.

### 31 4 OGSA-compliant

32 Background:

33 It is important that the ACS is an OGSA-service since the OGSA will be a harness framework for  
34 the future grid systems and ensure the interoperability of the diversity of the Grid system  
35 implementations.

36 Requirements:

- 1 • [4-a] ARI: The service must be implemented as an OGSA-service based on OGSA  
2 infrastructure services (such as XML, WSDL, WS-Addressing, WS-ResourceFramework, WS-  
3 Notification).
- 4 • [4-b] ARI: ACS must cooperate with other OGSA-services and must not overlap in capabilities  
5 with them.
- 6 • [4-c] ARI: Archive components must be retrievable by other OGSA-service entity such as  
7 CDDLM, Job Manager, and Candidate Set Generator.
- 8 • [4-d] AAF: Archive contents can be any opaque entities that are required for deployment and  
9 execution of application.

## 10 **Appendix**

11 References:

12 [OGSA v1.0] <https://forge.gridforum.org/projects/ogsa-wg/document/draft-ggf-ogsa-spec/en/19>

13