

On the Standardization of a Testing Framework for Application Deployment on Grid and Cloud Infrastructures

Thomas Rings, Jens Grabowski
Institute of Computer Science, University of Göttingen
 Göttingen, Germany
 {rings, grabowski}@cs.uni-goettingen.de

Stephan Schulz
Conformiq
 Espoo, Finland
 stephan.schulz@conformiq.com

Abstract—An important requirement in the successful deployment of grid and cloud computing technology in industry or governmental institutions is the ability to compose their infrastructures using equipment from different vendors. These equipments have to be engineered for and assessed to assure a problem-free interoperation. Focusing on software interoperability, we present a testing framework for the assessment of interoperability of grid and cloud computing infrastructures. This testing framework is part of an initiative for standardizing the use of grid and cloud technology in the context of telecommunication at the European Telecommunications Standards Institute. Following the test development process developed and used at the European Telecommunications Standards Institute, its application is exemplified by the assessment of for resource reservation and application deployment onto grid and cloud infrastructures based on standardized Grid Component Model descriptors. The presented testing framework has been applied successfully in an interoperability event.

Keywords—standardization, interoperability, GCM, grid, cloud, testing

I. INTRODUCTION

Accessing data and computing power from anywhere in the world becomes more and more an obligatory requirement from customers in order to store data and utilize computing power. Systems for efficient usage of idling resources which are physically located all over the world are needed. Approaches for realizing such systems can be found in grid computing systems but also recently in cloud computing systems. Both offer services for obtaining, providing and selling computing power on demand. This is especially interesting in application domains where computing power is needed spontaneously and in unpredictable time intervals. Many grid and cloud providers have recently appeared on the market offering their own custom-made solutions to address this need. However, from a customer point of view, it is required to access several systems offered by different providers to use more resources, for example, for replication on different systems, but also to save money choosing the best solution. A way to achieve this goal is to make these systems interoperable.

Interoperability provides a mean to success, opens new markets, fosters innovation, and enables mass markets. Inter-

operability requires however also agreed specifications, e.g., standards, which capture requirements and functionality, define architectures and interfaces, and specify protocols to be used to communicate via these interfaces. Ideal specifications are independent from implementations and leave space for innovation. Even if specifications are assumed as unambiguous, which is rarely the case, testing is needed to show that implementations follow such specifications. A step further is to test if implementations are able to interoperate.

Proper testing - similar as specification - requires a well defined process and guidelines to be effective in its application. The purpose of testing frameworks is to define a structured approach to test specifications in a given domain, i.e., what to test as well as how to test certain aspects of a *System Under Test* (SUT). They help to increase the quality of test specifications.

At the *European Telecommunications Standards Institute* (ETSI), an initiative for standardizing the use of grid and cloud computing technology in the context of telecommunication, the *Technical Committee GRID* (TC GRID) [1], has been formed. Under its umbrella, standards and interoperability in grid, cloud and telecommunication systems are analyzed, developed and forwarded. These include analysis of grid and cloud interoperability gaps [2] and surveys [3], comparisons between grid, cloud and telecommunication systems [4], the grid standard *Grid Component Model* (GCM) [5]–[7], and analysis of architectural options for combining grid and the *Next Generation Network* (NGN) [8]. As part of this initiative towards standardization, we present an interoperability testing framework for grid and cloud computing systems following the systematic test development process [4] developed and used at ETSI. The framework was developed for applications in interoperability events. The testing framework unifies testing in very diverse domains such as grids and *Infrastructure as a Service* (IaaS) clouds systems, which are dominated by proprietary interfaces for similar functionalities.

This article is structured as follows: In Section II, we provide an overview and a comparison with related work in the domains of grid and cloud computing. Afterwards in Section III, we introduce different types of testing in

the context of standardization including conformance and interoperability testing. In Section IV, we first introduce the ETSI GCM standard, which provides the main context for our testing framework that is illustrated afterwards. This testing framework, our main contribution, is applicable for resource reservation and application deployment in grid and cloud infrastructures. In Section V, the application of the framework is exemplified by the grid middleware Globus Toolkit and the cloud computing system of Amazon. An event around the presented testing framework organized by ETSI is described in Section VI. Finally, we conclude with a summary and outlook in Section VII.

II. RELATED WORK

ETSI developed and published a methodology for interoperability testing [9] and automated interoperability testing [10]. In the latter, guidelines and best practice in testing interoperability in an automated manner are presented. This methodology has been applied successfully for the development of interoperability test specifications for technologies such as IPv6 [11] and *Internet Protocol (IP) Multimedia Subsystem (IMS)* [12], and put into practice in ETSI interoperability events [13]. Previously, this approach had however not been applied to grid or cloud computing technology.

Several interoperability initiatives for grid and cloud computing systems exist. For cloud systems, these include the *Open Grid Forum (OGF) Open Cloud Computing Interface (OCCI) Working Group* [14], and the *Distributed Management Task Force, Inc. (DMTF) Open Cloud Standards Incubator* [15]. However, since the concepts of cloud computing are not fully agreed, interoperability test specifications have not been published, yet.

Interoperability initiatives such as *OGF Grid Interoperability Now (GIN)* and standards bodies in grid computing are described in [4]. OGF interoperability test specifications for grid are rarely available and only for selected standards such as GridFTP. They also follow more ETSI's notion of conformance testing than interoperability testing.

Due to our knowledge, an interoperability testing framework for such diverse domain of grid and cloud computing infrastructures has not been published. In addition, the testing framework development followed the ETSI test development process for a step towards a systematic approach and application in interoperability testing.

III. CONFORMANCE VS. INTEROPERABILITY TESTING

Interoperability testing demonstrates that implementations provide end-to-end functionality as described or implied by a specification. In this section, the differences and commonalities among conformance and interoperability testing are described.

Conformance testing is generally used to check that an implementation follows the requirements stated in a specification whereas interoperability testing checks that

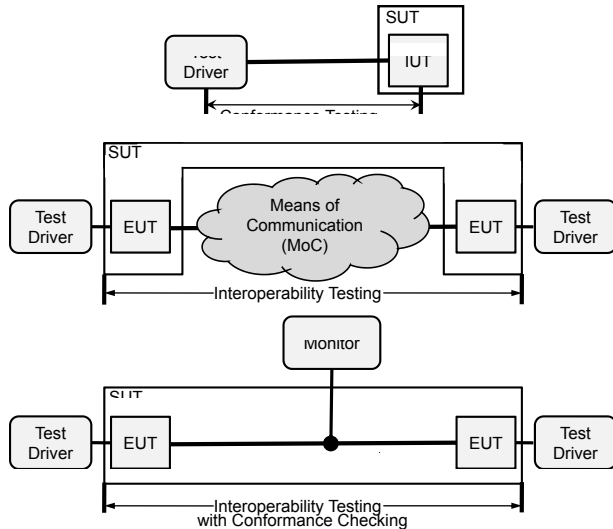


Figure 1. Three approaches to testing in standardization

a system composed of different equipment can provide a given service or functionality. In the following, we consider such a specification to be standardized and published by an organization like OGF, *World Wide Web Consortium (W3C)* or ETSI.

At ETSI, implementations of standards can be formally tested using conformance testing, interoperability testing, or interoperability testing with conformance checking [9]. The three approaches are illustrated in Figure 1. Each approach has benefits and limitations.

Conformance testing is functional black-box testing of one *Implementation Under Test (IUT)*. Hereby, it is checked if the IUT shows its conformity to a standard. The SUT embedding the IUT is a testing environment that includes parts which are required by the IUT to provide its service to the user. Conformance testing requires the development and implementation of sometimes sophisticated testing tools based, for example, on the *Testing and Test Control Notation (TTCN-3)* [16], which simulate the environment that the IUT is embedded in. However, even passing a conformance test does not automatically prove that the IUT is interoperable with other systems implementing the same standard. Standards need to be engineered for interoperability, because they may contain implementation options and leave space for interpreting requirement specifications which may lead to interoperability problems.

Interoperability testing checks if two or more *Equipment Under Test (EUT)* provide end-to-end functionality specified or implied by a standard. Each EUT is considered to be a complete system that may consist of several soft- and hardware components. EUTs interoperate via an abstract *Means of Communication (MoC)*. It is generally assumed that the communication services used between EUTs are compliant to underlying standards. Due to the proprietary

nature of end user interfaces, interoperability testing is usually driven manually and does per se not require any testing tool support. However, most interoperability problems in practice are often caused by incorrect or non compliant use of communication interfaces.

To address this problem, ETSI endorses a form of interoperability testing that includes conformance checking, i.e., a hybrid of the two testing approaches. This approach extends end-to-end interoperability testing with the monitoring of the communication among the EUTs. Monitors are used to check the conformance of the EUTs with the relevant protocol specifications within the SUT during the interoperability test. The ETSI experience with applying this hybrid approach during interoperability events [13] is that this approach provides valuable feedback to standardization. Even if end-to-end interoperability has been observed, EUTs did not communicate in a number of cases according to underlying standards. Although this approach is not a replacement for conformance testing, it offers an economic alternative to gain insights about the conformance of equipment participating in an interoperability test to a standard.

IV. TESTING FRAMEWORK

In this section, we present a testing framework for resource reservation and application deployment onto grid and cloud infrastructures which is based on the generic ETSI interoperability testing methodology. The framework is based on the ETSI GCM standards [5], [6] which provide a starting point for an extraction of testable requirements for an application deployment in grid and cloud infrastructures. However, the consolidating nature of the GCM standards allows the application of the presented framework outside of the specific context of GCM on other grid and cloud infrastructures. The presented testing framework includes interoperability test descriptions which provide a basis for interoperability test specifications that can be applied in the context of an interoperability event.

A. ETSI Grid Component Model

For users of grid or cloud communities, a provision of common interfaces for the allocation of resources for application deployment in different infrastructures becomes a crucial requirement, since the users wish to access multiple resources of several infrastructures simultaneously and in the most cost saving way. An approach towards such an interface is described in the GCM standards which have been standardized at the ETSI. The main objective of GCM is the creation of a uniform interface for allocating resources for applications whereas resources may be provided across different grid and cloud infrastructures. Resource information is specified in the form of *eXtensible Markup Language* (XML) descriptors, i.e., in GCM the *Deployment Descriptor* (DD) and the *Application Descriptor* (AD).

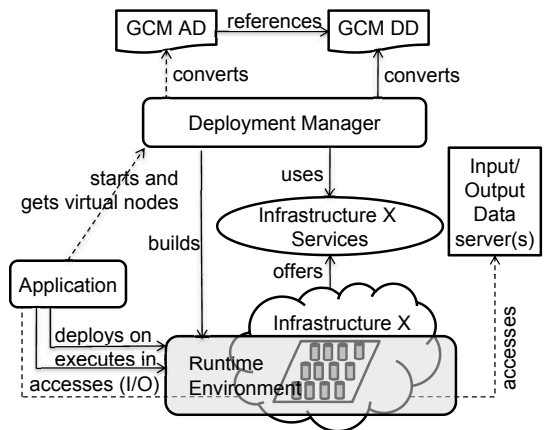


Figure 2. GCM Architecture

The content and concepts used in the GCM DD have been derived by abstracting different proprietary interfaces offered by commercial products in the grid, cloud, cluster and high performance computing domains. The key aspect of the GCM specification is the standardized mapping of this abstract interface to different proprietary interfaces of these systems as well as interfaces standardized for this purpose outside of ETSI, e.g., *Open Grid Service Architecture-Basic Execution Service* (OGSA-BES) [17]. Figure 2 shows a generic GCM architecture which focuses on the GCM AD and DD. It also introduces deployment manager and infrastructure entities to illustrate the likely separation of GCM descriptor processing and provision of the actual resource. Here, the user is assumed to provide a (test) application, a DD XML file, as well as optionally an AD XML file.

The GCM DD describes the resources that can be requested for the deployment of an application on one or more infrastructures. It is converted by the deployment manager into the invocation of specific infrastructure services or commands to reserve resources from the specified infrastructure(s). The GCM differentiates between infrastructures with direct access to their computing resource - as in the case of a cloud computing system or a set of desktop computers - and indirect access by using a job scheduler - as in the case of a cluster, or a grid middleware.

For the application deployment, a GCM AD specifies the mapping of virtual nodes to real resources as well as the location of input and output data server(s). If a GCM AD is provided, it is used to establish the runtime environment, which is required for application execution.

B. Test architecture

A test architecture for GCM-based application deployment which follows the concepts defined in Ref. [9], [10] is shown in Figure 3. The SUT consists of the deployment manager and at least one infrastructure. The different types of entities that compose the means of testing handle the

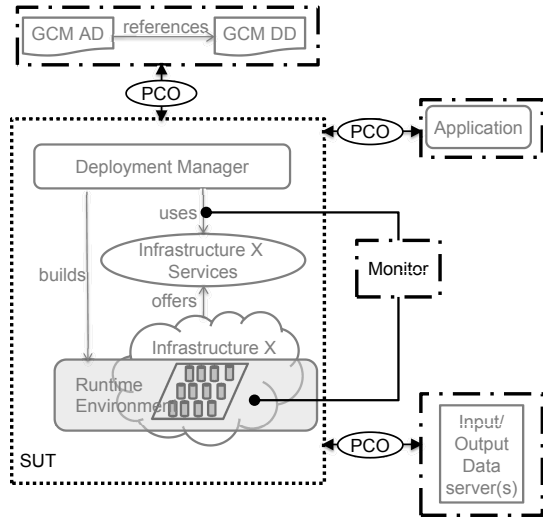


Figure 3. A test architecture for GCM-based deployment

provision of the GCM DD and AD files to the deployment manager. These entities associated with the infrastructure to be tested evaluate responses from the deployment manager, and analyze the output produced by the application via their *Point of Control and Observation* (PCO). In addition, the processes that run on each infrastructure as well as their interface(s) to the deployment manager and the input/output server(s) are monitored during tests execution.

The presented testing architecture can also be used to access other standards related to the deployment and execution of applications on grid or cloud infrastructures, e.g., an OGSA-BES web service based interface between the deployment manager and the infrastructure.

C. Test configurations

Test configurations are a refinement of the test architecture. They specify structural aspects of a test and define in detail all equipments participating in a test as well as their communication. Test configurations are then referenced during the specification of tests which mainly specify behavioral aspects. In this section, we introduce two GCM test configurations as an example. For a comprehensive list of test configurations the reader is referred to Ref. [18].

1) *Single infrastructure with a bridge*: The test configuration “Single infrastructure with a bridge” depicted in Figure 4 has two EUTs, whereas EUT A contains a deployment manager which is connected via a bridge to EUT B which contains a single infrastructure. Access to the deployment manager, the infrastructure, the test application to be executed, the GCM DD, and the GCM AD are distributed across two different physical machines. The user uses the deployment manager to load the GCM DD and, in case the test application is a GCM application, also the GCM AD as input, to connect remotely to the infrastructure

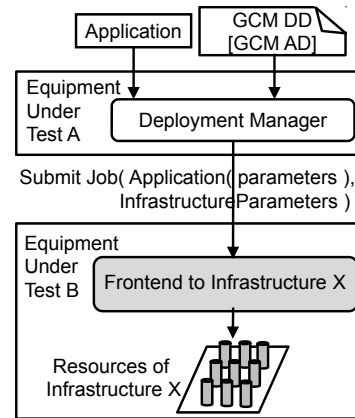


Figure 4. Single infrastructure with a bridge

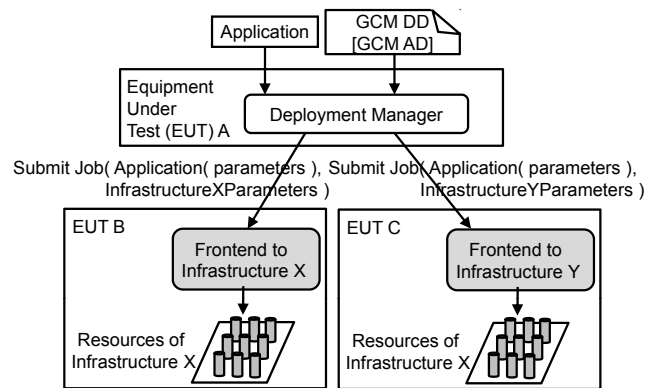


Figure 5. Two infrastructures and bridges

in order to establish the GCM runtime environment and to submit jobs related to the application from the remote machine. Depending on the type of the resource access provided by the infrastructure, a frontend may or may not be used to access resources.

2) *Two infrastructures and bridges*: The test configuration depicted in Figure 4 extends the test configuration described in the previous clause with a second infrastructure. This test configuration has three EUTs, whereas EUT A contains the deployment manager, EUT B contains the infrastructure X, and EUT C contains the infrastructure Y. Since the deployment manager controls both infrastructures at the same time, it has to be connected to each infrastructure via a bridge.

D. Test purpose specification

The first step in the development of ETSI test specifications is to analyze the base standard and extract testable requirements that are used to specify test purpose. A test purpose specifies if and how a catalogued requirement can be assessed in the context of a given test architecture, i.e., in the form of pre-conditions that are relevant to the requirement, and (a) stimulus and response pair(s). Each test purpose

TP ID:	TP_GCM_DD_DA_PA_001
Clause Ref:	ETSI TS 102 827 V1.1.1 clause 7.1
Configuration:	Single infrastructure or single infrastructure with a bridge
Summary:	Ensure that an infrastructure with direct resource access provides a single processor as specified in the GCM DD

Figure 6. Test purpose “Single processor with direct resource access”

includes at least one reference to the clause in a specification where the requirement to be assessed is described. It should have a unique identifier reflecting its place in the test suite structure.

The GCM standard defines the specification of deployment information and not an interface for the deployment. Therefore, the specification of test purposes for GCM descriptors is not a trivial task. In the case of GCM DD, the primary source of testable requirements is general information associated with resources, such as the number of offered processors or the number of threads per processor available for execution. The secondary source of testable requirements includes parameters that are common to a number of standardized GCM mappings to different infrastructures, e.g., wall time or maximum memory. However, these might not be supported by each infrastructure. Therefore, a test purpose should not be specific to a single mapping. A third source for additional test purposes includes variations of the requirements mentioned above based on different resource access methods, i.e., direct versus indirect as well as local versus remote access. In the presented testing framework, each test purpose is dedicated to one aspect of a specific requirement or concept defined in the GCM standard.

In Figure 6, an example test purpose for GCM DD is depicted. In this case, the support of the direct resource access is a precondition and a GCM DD with a single processor reservation is the stimulus. The success of the application execution determines the success of the resource reservation.

In the development of GCM AD test purposes, (re)assessing of GCM DD information should be avoided. For example, the test purposes for GCM AD should be applicable independently from the method the resources of an infrastructure are accessed (direct or indirect). This means that these test purposes focus on information and concepts specified in the GCM AD. Example source for test purposes is the handling of virtual nodes and input/output data location.

E. Specification of interoperability test descriptions

A test description is a detailed but informal specification of the pre-conditions and test steps needed to cover one or potentially more given test purposes. A test description shall contain the following information:

- **Identifier:** A unique identifier that relates a test to its group and sub-group.

- **Summary:** A unique description of the test purposes covered by this test.
- **Configuration:** A reference to all the equipments required for the execution of this test as well as their connection.
- **Specification References:** One or more references to clauses in the standard for which the test purposes have been specified
- **Test application:** A reference to the test application which is required to execute this test.
- **Pre-test conditions:** A list of all conditions that have to be fulfilled prior to the execution to a test. These conditions should identify the features that are required to be supported by participating equipment to be able to execute this test, requirements on GCM descriptors, as well as requirements on the parameterization of the test application.
- **Test sequence:** A test sequence is written in terms of external actors and their ability to interact and observe the services provided by the infrastructure, i.e., end-to-end behavior. Based on its success, a test verdict reflecting the interoperability of all EUTs in a test is derived.

The test description can also include a list of checks that should be performed when monitoring the EUT communication on standardized interfaces during the end-to-end test. In the case of GCM testing, this option is not directly relevant since the GCM standard does not intentionally define the interfaces between a deployment manager and infrastructures. However, checks can be formulated if an infrastructure implements interfaces standardized for resource reservation and application execution by other standardization organization, e.g., OGF or DMTF.

An example test description for the GCM DD test purpose shown in Figure 6 is depicted in Figure 7. This test description describes a test to check if an infrastructure with direct resource access provides a single processor as specified in the GCM DD. A complete list of the test descriptions can be found in Ref. [18].

F. Test application

For the assessment of the success and validity of each application deployment, a test application is executed on all involved infrastructures. The purpose behind these applications is not to perform complex, real world, computational tasks but to produce output that allows determining the real usage of resources and the behavior relevant to a test purpose covered by a test. The test application is parameterizable to allow its reuse across multiple tests.

An example for a test application is the single process batch job. This test application starts a single process on a single processor and consumes CPU and memory for a given amount of time. The application’s behavior including its execution time, the amount of memory to allocate, and

Interoperability Test Description		
Identifier:	TD GCM DD DA PA 001	
Summary:	Ensure that an infrastructure with direct resource access provides a single processor as specified in the GCM DD	
Configuration:	Single Infrastructure or single Infrastructure with a bridge	
Specification References:	GCM DD clause 7.1	
Test Application:	Single process batch job	
Pre-test conditions:	<ul style="list-style-type: none"> Infrastructure provides direct resource access GCM DD contains a direct group description with <code>hostList</code> containing one host and host description with <code>hostCapacity=1</code> for the infrastructure Infrastructure has a processor available for use 	
Test Sequence:	Step	
	Description	
	1	User loads the GCM DD and starts the test application on the infrastructure using the deployment manager
	2	Verify that the infrastructure has created and executed the process
3	Verify that returned application output is correct	

Figure 7. Test description “Single processor with direct resource access”

the number of threads can be controlled by parameters. The application prints all information required to determine if a test execution has succeeded or failed either to the standard output or a file. This includes the application start time, the value of each parameter, and the identifier of the application.

G. Test selection and execution

To determine the applicability of a test, all pre-conditions need to be evaluated. To speed up this process, an *Implementation Conformance Statement* (ICS) should be established to allow infrastructure providers to specify supported features prior to a test execution and support automatic test selection. A test should be selected for execution if all of its pre-conditions have been ensured. Common types of pre-conditions in the GCM tests include constraints on:

- the GCM DD and/or AD specifications,
- the infrastructure relating to the type of resource access, features that need to be supported by the EUTs, and available amount of resources,
- and the test application parameterization.

A test should not be selected and recorded as being not applicable if one of its pre-conditions is not met by one (or more) equipment part of the SUT.

A collection and specification of *Protocol Implementation Extra Information for Testing* (PIXIT) can be used to capture infrastructure specific aspects of a GCM DD such as the access details to an infrastructure and resource identifiers, and used to significantly speed up the execution of tests. The implementer of an IUT/EUT states the PIXIT that includes information according the IUT/EUT and its testing environment to enable runs of an appropriate test suite against the IUT/EUT [19].

Each grid and cloud infrastructure will be assessed under the same conditions based on a standardized ETSI test specification. Applicable tests are executed by uploading a test specific application, providing infrastructure and deployment information, e.g. via GCM descriptors, and observing

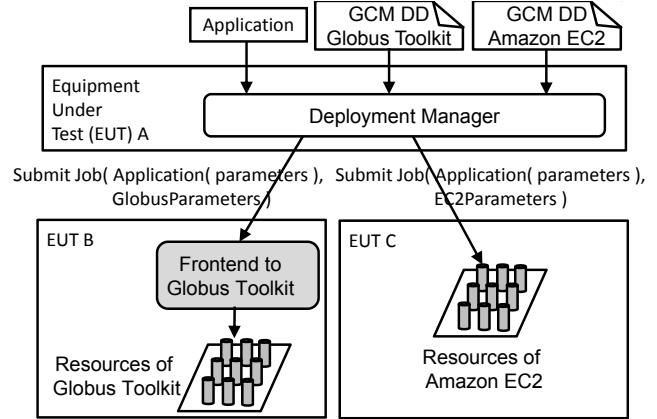


Figure 8. Test configuration “Two infrastructures and bridges” exemplified by Globus Toolkit and Amazon *Elastic Compute Cloud* (EC2)

the execution of the application as specified in the test specification.

V. EXAMPLE TEST SESSION

This section describes how the tests of the framework can be applied. For this, we apply the test configuration “Two infrastructures and bridges” more specifically as depicted in Figure 8. EUT B contains the grid middleware Globus Toolkit [20] whereas EUT C includes the cloud computing system Amazon *Elastic Compute Cloud* (EC2) [21] which is basically an IaaS. For each infrastructure, a GCM DD is needed.

The attributes for describing a DD for Globus Toolkit have already been specified in the GCM standard [5]. The DD used in this test session is depicted in Listing 1. Globus Toolkit is contacted via a *Secure Shell* (SSH)-bridge in order to access the Globus Toolkit frontend. An UNIX-based operating system runs on each computing node whereas each contains four processors which are represented by the element `hostCapacity` of the attribute `host`. Since Globus Toolkit is an infrastructure with indirect resource access, the total number of available processors is not specified.

A DD for the Amazon EC2 is depicted in Listing 2. A scheme for this infrastructure has not been specified yet, but to experience the application of Amazon EC2 and GCM, this example DD has been developed. In case of its successful deployment, it gives a base for an extension of the GCM standard by the specification of a GCM DD scheme for Amazon EC2. Since Amazon EC2 is an infrastructure with direct resource access, the number of included computing node needs to be specified. In our example, the Amazon EC2 contains ten computing nodes which are based on a Windows operating system. The infrastructure is accessed via an SSH-bridge. Both presented DDs can be merged into one DD file.

In this test session, we exemplify the test specified in the test description depicted in Figure 9. It will be checked

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <GCMDeployment xmlns="urn:gcm:deployment:1.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="urn:gcm:deployment:1.0
5     http://etsi.org/schemas/GCMDDSchemas/extensionSchemas.xsd">
6   <environment>
7     <javaPropertyVariable name="user.home" />
8   </environment>
9   <resources>
10    <bridge refid="globusGateway" />
11    <group refid="globusGrid">
12      <host refid="ComputeNodeUnix" />
13    </group>
14  </resources>
15  <infrastructure>
16    <hosts>
17      <host id="ComputeNodeUnix" os="unix" hostCapacity="4">
18        <homeDirectory base="root" relpath="{user.home}" />
19      </host>
20    </hosts>
21    <groups>
22      <globusGroup id="globusGrid"
23        hostname="globus.grid.local"
24        bookedNodesAccess="ssh"
25        queue="free">
26        <maxTime>5</maxTime>
27        <stdout>./output</stdout>
28        <stderr>./error</stderr>
29      </globusGroup>
30    </groups>
31    <bridges>
32      <sshBridge id="globusGateway"
33        hostname="grid.informatik.uni-goettingen.de"
34        username="globususer" />
35    </bridges>
36  </infrastructure>
37 </GCMDeployment>

```

Listing 1. GCM DD for Globus Toolkit

```

1 <GCMDeployment>
2   <environment>
3     <javaPropertyVariable name="user.home" />
4   </environment>
5   <resources>
6     <bridge refid="amazonCloudGateway" />
7     <group refid="amazonCloud">
8       <host refid="ComputeNodeWindows" />
9     </group>
10  </resources>
11  <infrastructure>
12    <hosts>
13      <host id="ComputeNodeWindows"
14        os="windows" hostCapacity="1">
15        <homeDirectory base="administrator" relpath="{user.home}" />
16      </host>
17    </hosts>
18    <groups>
19      <amazonCloudGroup id="amazonCloud"
20        hostList="node-[01-10]">
21      </amazonCloudGroup>
22    </groups>
23    <bridges>
24      <sshBridge id="amazonGateway"
25        hostname="aws.amazon.com"
26        username="amazonuser" />
27    </bridges>
28  </infrastructure>
29 </GCMDeployment>

```

Listing 2. GCM DD for Amazon EC2

if both infrastructures provide multiple processors for a parallel application. Therefore, the parallel application al-

Interoperability Test Description									
Identifier:	TD GCM DD DA IA PA 001								
Summary:	Ensure that an infrastructure with indirect resource access and an infrastructure with direct resource access provide multiple processors for a parallel application as specified in the GCM DD								
Configuration:	Two infrastructures and bridges								
Specification:	GCM DD clause 7.1, 7.2								
References:									
Test Application:	Parallel job								
Pre-test conditions:	<ul style="list-style-type: none"> One infrastructures provides indirect resource access One infrastructures provides direct resource access GCM DD contains one direct group description and one indirect group descriptions Communication between the infrastructures is supported Infrastructures have multiple processors available for use 								
Test Sequence:	<table border="1"> <thead> <tr> <th>Step</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>User loads the GCM DD and starts the test application on both infrastructures using the deployment manager</td> </tr> <tr> <td colspan="2" style="text-align: center;">executed in both infrastructures</td> </tr> <tr> <td>3</td> <td>Verify that returned application output is correct</td> </tr> </tbody> </table>	Step	Description	1	User loads the GCM DD and starts the test application on both infrastructures using the deployment manager	executed in both infrastructures		3	Verify that returned application output is correct
Step	Description								
1	User loads the GCM DD and starts the test application on both infrastructures using the deployment manager								
executed in both infrastructures									
3	Verify that returned application output is correct								

Figure 9. Test description “Multiple processors in infrastructures with indirect and direct resource access”

locates more than one processor in each infrastructure. The execution of the application will be logged in order to evaluate the result of the test. The Amazon EC2 infrastructure includes ten nodes as described in the DD and the number of nodes in the Globus Toolkit cannot be determined from its DD. Therefore, the parallel test application needs to start ten processes on the Amazon EC2 system and secondly four processes in the Globus Toolkit environment. If all the processes have been started successfully and if the test application writes its output as expected, the test can be evaluated as successful.

VI. INTEROPERABILITY EVENT

At ETSI, the validation of interoperability test specifications usually takes place in testing events. Such testing events also provide opportunities for vendors of systems for the technology under test to assess and demonstrate their interoperability with systems from other vendors. For the GCM testing framework, this validation took place in November 2009 as part of the ETSI Grids, Clouds, and Service Infrastructures event [22]. It provided an unique opportunity for standardization experts, operators, IT service providers, telecom equipment vendors to see available systems running.

A. Application of the Framework in an Interoperability Event

A requirement of the application of the presented testing framework is that a possible SUT needs to include an implementation of the GCM standard. However, the interoperability event included a variety of state-of-the-art systems that implement grid, cloud, cluster, or related technologies which fit into the idea of the GCM standard. Therefore, as

a first step, we compared and executed different state-of-the-art system to determine and evaluate their similarities and differences. The goal was to feed the result of the demonstrations of the systems of the participated vendors back into the standard to make GCM applicable for these systems.

B. Event Summary

The interoperability event attracted various actors of grid and cloud computing systems from academics, industry, and other standard organizations. In total, six exhibitor demonstrated their grid or cloud environments. The demonstration included resource reservation and application deployment onto different infrastructures. The basis for their evaluation was a questionnaire as well as use case scenarios defined in the ETSI GCM test specification. The questionnaire assessed interfaces for resource reservation and preparation of infrastructure as well as standard support. The test specification included use case scenarios for infrastructures offering direct and indirect access. Infrastructures were not required to support ETSI GCM standards so that custom-made interfaces were used for application deployment. In addition, capabilities beyond the requirements of the ETSI use cases could have been shown.

The systems of the vendors who participated in the event mainly implemented a deployment manager for IaaS. However, there were also a deployment manager for *Platform as a Service* (PaaS) and a cloud management system. All the solutions provided a portal or *Graphical User Interface* (GUI) for resource reservation. For an automated use, these have been realized either as *RESTful Web Service* (WS), *Command Line Interface* (CLI), or a Java/XML based API. In general, all the demonstrated systems shield users from complex details of resource reservation. Resource provision and resource requests were handled separately. Handling of data transfer to/from the computing node was mostly done by the application, e.g., using the *Secure Copy Protocol* (SCP) or FTP.

1) *Resource Request and Access*: Resource request and access were based on different requirements on resources such as computing, storage, or network resources and assessed by the test application referenced in the test description. The resources were selected based on requirements such as performance, service level agreements, application types, or objectives. Fixed or common concepts between the systems could not be identified. The reason may be the application domain specific implementations of the systems. For example, while one systems needs a detailed specification of resource requirements, another system only requires a specification of a class defined for resource requirements. In addition, the systems differed in the level of transparency of resource management. Therefore, there is a need for an appliance independent hypervisor that manages resources independent of the application.

2) *Standard Support*: For resource request and access, mainly ETSI GCM, OGF *Distributed Resource Management Application API* (DRMAA), and DMTF *Open Virtualization Format* (OVF) have been implemented. However, most systems use non compliant default configuration but also allow adaptation to DMTF OVF.

A minor standard support exist in commercial cloud systems since cloud computing is an emerging technology and standards are only slowly evolving. Most of the cloud systems provide proprietary RESTful WS and XML based interface to resources. This provides a simple basis and can be extended easily.

Weak points of existing standards are that they allow too many options such as in the OGF *Job Submission Description Language* (JSDL) or that they require to fix the location of resources. Most desired is a standardized API for virtual machine and resource management.

3) *Test automation*: The presented testing framework has been mainly developed for an application in an interoperability event. In such events, systems of different vendors are connected to each other. These can be then seen as black boxes connected and accessible only by their interfaces. Automating test executions in such a configuration is challenging since agreed standards are not available for the usage of grid or could infrastructures. Furthermore, in the setting of such events participants are usually know as late as two weeks before the event. However, test specification efforts usually are concluded months before an event. Therefore, and also due to time and budget limitations, test execution cannot be automated by the organizer. For participants, this would be an extra cost they are not willing to spend. Therefore, the test have been conducted manually.

4) *Reflections*: Key areas for standardization, i.e., clear boundaries of the state-of-the-art systems is an API for the provision of resources and for requests of resources. Open issues are the achievement of portability of appliances of the hypervisor, i.e., the management of different virtual machines images and resources. Minor concerns include lack in agreed terminology and the need for a strong common denominator.

VII. SUMMARY AND OUTLOOK

We presented a testing framework around the GCM standard developed by ETSI that is applicable for grid, cloud, and cluster management systems. This framework has been developed by following the ETSI test process. We described the GCM architecture, an applicable test architecture, and exemplified test configurations. Test purpose specification and interoperability test descriptions have been explained. Furthermore, we considered test selection and test execution according to the presented testing framework. The application of the framework has been exemplified by Globus Toolkit and Amazon EC2 as EUTs.

We believe that the GCM standard is a first step towards the use of resources from different infrastructures simultaneously in a standardized way. Infrastructure adapters can easily be incorporated into the standard. With the presented testing framework, it is possible to enhance and expand the standard to allow a wide adaption of several systems provided by different vendors.

The described testing framework is part of an initiative for standardizing the use of grid and cloud technology in the context of telecommunication at ETSI. We believe that this testing framework is a step towards systematic interoperability testing of grid and cloud computing environments in general [3].

The specification of executable test cases from GCM test descriptions is considered as future work. This step includes the further concretization of GCM test configurations including equipment user and monitor test components as well as the specification of their behavior.

REFERENCES

- [1] ETSI Technical Committee Grid (TC GRID), since 2010 TC CLOUD, [Online; <http://portal.etsi.org/cloud> fetched on 11-05-10].
- [2] "ETSI TR 102 659: GRID;Study of ICT Grid interoperability gaps," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.
- [3] "ETSI TR 102 766: GRID; ICT Grid Interoperability Testing Framework and survey of existing ICT Grid interoperability solutions," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.
- [4] T. Rings, G. Caryer, J. Gallop, J. Grabowski, T. Kovacikova, S. Schulz, and I. Stokes-Rees, "Grid and Cloud Computing: Opportunities for Integration with the Next Generation Network," *Journal of Grid Computing: Special Issue on Grid Interoperability, JOGC*, in press, 2009.
- [5] "ETSI TS 102 827: GRID;Grid Component Model (GCM);GCM Interoperability Deployment," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2008.
- [6] "ETSI TS 102 828: GRID;Grid Component Model (GCM);Part 2: GCM Application Description," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2008.
- [7] "ETSI TS 102 829: GRID;Grid Component Model (GCM);GCM Fractal Architecture Description Language (ADL)," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.
- [8] "ETSI TR 102 767: GRID;Grid Services and Telecom Networks;Architectural Options," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.
- [9] "ETSI ES 202 237: Methods for Testing and Specification (MTS);Internet Protocol Testing (IPT);Generic approach to interoperability testing," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2007.
- [10] "ETSI EG 202 810: Methods for Testing and Specification (MTS);Automated Interoperability Testing;Methodology and Framework," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, to be approved.
- [11] "ETSI TS 102 517: Methods for Testing and Specification (MTS);Internet Protocol Testing (IPT): IPv6 Core Protocol;Interoperability Test Suite (ITS)," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2008.
- [12] "ETSI TS 186 011-2: Technical Committee for IMS Network Testing (INT);IMS NNI Interworking Test Specifications;Part 2: Test descriptions for IMS NNI Interworking," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.
- [13] ETSI, "PlugtestsTM Interop Events," [Online; <http://www.etsi.com/WebSite/OurServices/plugtests/home.aspx> fetched on 11-05-10].
- [14] OGF, "Open Cloud Computing Interface Working Group," [Online; <http://www.occi-wg.org> fetched on 11-05-10].
- [15] DMTF, "Open Cloud Standards Incubator," [Online; <http://www.dmtf.org/about/cloud-incubator> fetched on 11-05-10].
- [16] ETSI, "ETSI Standard (ES) 201 873 V3.2.1: The Testing and Test Control Notation version 3; Parts 1-8," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, also published as ITU-T Recommendation series Z.140, 2007.
- [17] I. Foster, A. Grimshaw, P. Lane, W. Lee, M. Morgan, S. Newhouse, S. Pickles, D. Pulsipher, C. Smith, and M. Theimer, "OGSA Basic Execution Service Version 1.0, GFD-R.108," Open Grid Forum, 2008.
- [18] "ETSI TS 102 811: GRID;Grid Component Model (GCM);Interoperability test specification," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, to be approved.
- [19] ISO/IEC, "Information Technology – Open Systems Interconnection – Conformance testing methodology and framework," International ISO/IEC multipart standard No. 9646, 1994-1997.
- [20] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," in *Proceedings of the IFIP International Conference on Network and Parallel Computing (NPC05)*, ser. LNCS, vol. 3779. Springer, 2005.
- [21] "Amazon Elastic Compute Cloud (Amazon EC2)," [Online; <http://aws.amazon.com/de/ec2/> fetched on 11-05-10].
- [22] ETSI, "Grids, Clouds & Service Infrastructures: PlugtestsTM and Workshop," [Online; <http://www.etsi.com/plugtests/GRID09/GRID.htm> fetched on 11-05-10].