License4Grid: Adopting DRM for Licensed Content in Grid Environments

Joachim Götzte, Tino Fleuren, Paul Müller
Dept. of Computer Science, ICSY Lab
University of Kaiserslautern
Kaiserslautern, Germany
Email: {j_goezte, fleuren, pmueller}@cs.uni-kl.de

Simon Schwantzer
imc information multimedia communication AG
Saarbrücken, Germany
Email: simon.schwantzer@im-c.de

Abstract—Processing of licensed content with automatic compliance checking of the license terms is currently not supported in Grid environments. However, applications processing large amounts of data is a topic recently gaining more and more attention. The use of high performance computing capabilities, e.g., provided by Grid environments, is an obvious choice to speed up the processing time.

Currently, most of the input data required in such Grid applications is freely accessible by standard Grid technology. However, many upcoming applications for Grid Computing require data provided under a specific license, also leading to license violations on a regular basis, because license compliance can currently not be validated. Data under a specific license is often retrieved from outside the Grid over individual portals directly from a content provider or distributor. Beside the additional efforts for user and security management, such external distribution approaches prevent an association between the license information and the content.

In this work, the internal distribution approach for licensed content in Grid environments (License4Grid) is designed, maintaining the association between the license information and the corresponding data. The design respects the requirements emerging from handling either unprotected and protected content and makes use of the user and security mechanisms provided by common Grid technologies in order to fit into the environment homogeneously.

Keywords—Grid Computing; Digital Rights Management; Licensed Content

I. MOTIVATION

The term Grid Computing is based on an idea first mentioned in the 1960s. But only in recent years the network technology has reached a level to allow for the interconnection of high-performance computing centers in order to cooperate for solving complex problems with algorithms that require huge computing power and process large amounts of data. While in the beginning of the current era of Grid Computing the focus has been mostly on the basic realization of the interconnection and cooperation of the participating computing centers, in the recent years, more and more the applications are coming into focus. Although Grid Computing can provide access to high-performance computing resources located somewhere around the world, reduce the calculation time of the algorithms mentioned earlier, and provide a huge performance impact, these improvements only include calculations based on data that is already available in a Grid environment.

Some applications require that licensed content is available in a Grid environment. Examples for such applications are flood simulations or simulations of noise propagation, where the simulation is based on data describing the environmental conditions and spatial character of the simulated environment. This kind of content is often licensed content, i.e., a license defining how the data can be used and distributed.

Currently, licensed content is either provided manually or with a kind of distribution technology that cannot be integrated directly into a Grid application. In order to improve the performance of such applications and to allow for on-demand simulations where needed, the content should be directly integrated and accessible from within a Grid environment.

With current Grid technology, compliance with the license cannot be guaranteed, because of the missing link between license information and content. In order to make licensed content available in a Grid environment, several problems have to be resolved. Content providers have to recognize the need to provide their content using technology that can be easily integrated into a Grid environment. However, even more important is the enhancement of the current Grid environments with the ability to maintain and work with licensed content. In the following, different approaches for the work with licensed content are analyzed and a solution allowing for the integration of licensed content into a Grid environment is described (keeping the association between license and data intact).\(^1\)

This paper is organized as follows. In section II, the basics for licensed content is described, while section III analyzes different distribution approaches of licensed content in Grid environments and extracts requirements that need to be fulfilled by a distribution framework. In section IV the design of the proposed distribution framework License4Grid is described. Implementation aspects and considerations are discussed in section V. The paper concludes with related work and a summary.

II. LICENSED CONTENT

The focus of this work is content in a digital environment. Several different terms concentrate on different aspects of content, e.g., “intangible assets” focussing on

\(^1\)This work is funded by the German Federal Ministry of Education and Research (BMBF).
copyright, “information goods” on the trade of content elements, and “paid content” on the payment aspect. Here we use the term “licensed content”, because the major contribution of this work is on content provisioning while the association between license and data is maintained.

A. Aspects of Data as Subject of License Conditions

Similar to licensed software, the acquisition of digital content does usually not include the ownership, but more or less restricted usage rights. Common restrictions affect the purpose of usage, the period of time the content can be used, and its binding to humans and technical resources.

The primary characteristic of such acquired content is its binding to a specific license, which defines the rights of content owner and content user. If we talk of licensed content, we include all data as subject of license conditions independent from its digital form, e.g., files or database entries. Aside from paid content, this definition also includes content which is available for the licensee without a (financial) equivalent value, e.g., a license model based on a free non-profit use like many academic software licenses.

A possible categorization of licensed content is:

- Media, e.g., documents, audio, image, and video data.
- Information goods, e.g., statistical and trend analysis.
- Data used for information processing, e.g., satellite images for weather forecast.

In Grid environments, the last content category is the most significant one, because data processing often requires large computational resources. Grids are designed to offer these resources to business and research facilities. An example is given with the research project Spatial Data Infrastructure Grid GDI-Grid [8] (part of the German Grid Initiative D-Grid [3]), which aims to achieve the efficient mining and processing of spatial data for the simulation of noise dispersion and disaster management.

In 2007, the European Commission passed the “Flood Directive” (Directive 2007/60/EC) [4] dealing with the identification of inundated areas and the creation of flood risk maps. The basis for flood modeling is provided by computationally and storage-intensive flow simulations. A typical desktop computer is not capable of handling the data of more than a few square kilometers at a time, and it takes hours to complete a discretization process. In particular the use of high-resolution topographic data across large areas and the evaluation of the detailed simulation results creates the need for sophisticated processing techniques and storage management.

The focus of the project GDI-Grid is the introduction of Grid technology into the geospatial applications. Flood modeling is an example for such a scenario investigated in this project [5]. Services for flood modeling require and produce a large amount of data used in multiple resource-intensive processing steps.

Currently, the required input data, e.g., cartographical maps, are acquired and paid for in a manual process. It is not unlikely, that many persons are involved in this process and that the data is retrieved by an HTTP download. Not only is this approach slowing down the above described process considerably, but also the input data – which is licensed content – is provided without the possibility of recreating the association between the license and the data.

B. Handling of Licensed Content

Licensed content in a digital environment requires a digital rights management (DRM). A DRM includes the following steps of a DRM chain:

- Often a description is added to the content as meta information. The meta information can be used to search and browse content available from a provider.
- An essential part of a DRM chain is the identification of data. The identification is unique for each different data package and, if necessary, for equal packages delivered to different users.
- In a non-free distribution, a step of trade is added to the DRM chain, which includes billing and payment.
- An optional step is the protection of the content, e.g., by encrypting the content with a key only available for specific users or nodes.
- Depending on the complexity of rights management, specific control mechanisms for the data can be used.
- Beside the protection, mechanisms to trace the content can be implemented to monitor its distribution, e.g., watermarking techniques.

Each of these steps can be omitted, which significantly defines the complexity of a specific DRM chain. Note that the data protection, for example by encryption, is not necessarily a required feature of a DRM system.

A DRM system is responsible for the management of digital rights, as well as for the digital management of rights. The first part includes the identification of data, the creation of metadata, and the creation of mapping to a license. The second part includes steps of protection, distribution, and control.²

When we take the step to a computational environment, we talk of licensed data representing the licensed content. The criteria for handling data are defined by the activities to maintain a DRM chain. These activities are different for every element of the chain and depend on the specific environment the element is located. Therefore, a heterogeneous environment increases the complexity of handling licensed content.

To enable a DRM chain, some properties are required for the participating nodes and the communication between them:

1) To ensure that only authenticated nodes take part in the DRM chain, the communication partners have to identify each other in a secure way (mutual authentication).
2) The communication has to be protected against unauthorized access and modification. Therefore, the transmission has to be secured depending on the degree of protection chosen for the DRM chain.
3) As being the primary element for licensed content, the identification of the data has to be kept up

²Often DRM is reduced to step 2, i.e., the digital management of rights.
at the nodes and during the transmission between nodes. This also includes possibly related rights management objects.

In [20] the terms “Content Object” (CO) and “Rights Object” (RO) were introduced, whereas the CO is a container of media objects and the RO is the related management data, i.e., the identification, metadata, and management data related to the content. In this work, the term “Content Object” is not only the data representing the licensed content but also the surrounding container.

III. LICENSED CONTENT IN GRID ENVIRONMENTS

The previous section discussed the basics of licensed content. In order to identify the requirements of content distribution in a Grid environment, the different possible distribution approaches have to be evaluated in detail and scenarios have to be identified showing the typical course of events in such a distribution framework.

A. External and Internal Distribution Approach

We can differentiate between two basic approaches for content distribution. If the distribution node is part of the observed environment, we talk of an internal distribution approach, otherwise of an external distribution approach.

An external distribution gives more flexibility to the distributor, because he can freely define the interfaces available for accessing the data. The major drawback of an external distribution approach is the total loss of control over the data after its distribution.

An internal distribution binds the distributor to the technical conditions of the specific environment. On the other hand, mechanisms for security and user control provided by the environment can be used for implementing the DRM chain.

In the area of media distribution, the common way of distribution is an internal distribution that is implemented by creating an environment with the binding to specific software and hardware. Taking a look at so-called DRM-free distributions, not only the protection is removed, but also the binding of the CO and the RO by dropping the RO. Therefore, this distribution type can be categorized as an external distribution approach.

Current licensed content distribution solutions in Grid environments support only external distribution approaches, although this results in the previously described disadvantages and the different Grid environments offer comprehensive user and security features that could be made use of.

B. Scenarios of an Internal Distribution for Grid Environments

In this section, two scenarios for an internal distribution in Grid environments are described. These scenarios differ in complexity caused by the aspect of content protection.

Scenario 1 describes a content distribution without content protection. Participants are the licenser, the distributor, the licensee, and the computing resource. Taking a look at the involved Grid resources, four different resources are taken into consideration:

- Licensor service: Using this service, the client can acquire a license for the licensed content.
- Distribution service: The licensed content is distributed to authorized users by the distribution service.
- User access client: The node where the licensee is acting from.
- Computing resource: The computing resource is responsible for the data processing using the licensed content.

The course of events in the first scenario is as follows:
1) The licensor creates data packages at a distributor.
2) The licensee selects such a data package and acquires a license.
3) With the acquired license, the licensee can gain access to the data package at a suitable distributor.
4) A Grid service is used by the licensee, utilizing the required data, in order to execute the desired function on a computing resource.

The data package contains the CO and the RO. The packaging is maintained until the data processing is initialized.

Whereas the first scenario is the simplest scenario including only the maintenance of the mapping between the CO and RO, scenario 2 adds a content protection mechanism to the distribution model. Participants and resources are the same as in the first scenario.

As an additional step aside of packaging the CO and the RO, the data is encrypted by the distributor during the initial preparation phase. Only the computing resource is authorized to decrypt the data. Both scenarios divide into two different ways of data access. The first approach allows a direct user access to the encrypted data similar to the first scenario. The second approach describes an indirect data access: The user receives only the access information, but is not authorized to request the data. He forwards the access information to the computing resource, which requests the data itself from the distributor during a preparation phase. This approach enables a fast access to large data sets, if a high-speed connection is provided by the distributor node as well as the computing resource.

C. Requirements for an Internal Distribution of Licensed Content

In the previous section the different scenarios for an internal distribution have been described. Based on these scenarios, the requirements for a solution of internal distribution of licensed content have been extracted.

The minimal requirements for a non-protected content distribution (scenario 1) are:

3In the presented scenarios, it is not necessary to differentiate between a licensor and an independent institution handling the payment (clearing house) [15].

4The workflow does not include a license validity check.
1) Mutual authentication of participating users and services.
2) Restricted data access to the distribution service.
3) Creation of a container for combining data (CO) and license information (RO).
4) Support for maintaining this container at the user access client and the computing resource.
5) Possibility for data extraction at the computing resource.

For a distribution of protected content (scenario 2) additional requirements have to be fulfilled:
1) Data encryption at the distribution service.
2) A preparation phase including license validation request and decryption at the computing resource.

IV. DESIGN OF A DISTRIBUTION FRAMEWORK

Based on the scenarios of the previous section and the extracted requirements, the distribution framework License4Grid has been designed. The required services are observed in their functionality and their interaction. The services are grouped by licenser service, distributor service and computing resource (see subsection III-B).

We recommend scenario 2 with content protection and indirect data access, because it provides the optimal means to guarantee that the DRM chain is always undamaged. Further analysis of the advantages and disadvantages of the proposed scenario can be found in subsection IV-D.

In such an environment, all parties will be bound together through trust relationships. The party providing the licensed content relies on all other parties abiding by the terms of the license. A service provider can utilize the infrastructure to develop services that are able to handle licensed content. In order to do so, not only the service provider has to comply with the terms of the license, but the computing resources, too. Thus, all three parties need to collaborate in order to uphold the DRM chain. Additionally, all three parties are independent and neither content nor software or computing resources have to be provided by a single institution.

In order to process licensed content in the proposed distribution framework, three phases have to be supported: preparation, obtaining a license, and using a computing resource. First the data is uploaded and a license is created, then the client obtains a specific license allowing him to access a data package that can be processed by a computing resource. Especially in the last phase, the scenarios differ depending on direct or indirect access to the licensed content. When using the scenario of direct access, the client downloads the licensed content from a distribution service, and then he delivers the data to the service providing the desired computation functionality. In the indirect access scenario, the computing resource downloads the licensed data on behalf of the client, and then it starts the processing. The following subsections describe those phases in detail.

A. Preparation

Before licensed content can be made available in the Grid environment, the licenser and the distributor have to traverse a preparation phase.

The licenser has to manage multiple pieces of information regarding the content he intends to provide: The data representing the content, an unique identifier generated for its distribution, and metadata containing description and related license information. The step “Preparation” in figure 2 refers to figure 1 showing the process of creating data packages by the licenser at a distribution service. In the preparation phase, the licenser requests a data upload (see D1 in figure 3) for a globally unique identifier he generated for the specific content (content-id). The distributor creates an empty container for data and identifier and opens a door for file transfer to a directory dedicated to this container and only accessible by the licenser. If the data is to be protected, he also generates a key-pair for encryption. The distributor sends the URL of the upload directory and optionally the decryption key to the licenser. Now, the licenser can upload the data with a common file transfer client. When the upload is complete, he sends a signal back to the distributor, who packs the data into the container. As part of the packaging, the data may be encrypted with the secret encryption key of the key-pair. The current state of the packaging can be pending, packaging, complete and error.

The file transfer door can be closed after the data package was successfully created. To update a data package, an upload for an already existing identifier (content-id) can be requested in order to restart the process of file transfer and

5Identification and metadata in DRM systems is discussed in [17].
packaging. The content-id, encryption key, and decryption
key for the data package will not change during the update.

After completing this preparation phase, the licensed
content, i.e. the data package, is now available. In order to
improve the availability of a data package, the distribution
services provide the functions replicateContent() (D3) and
storeReplica() (D4). Using these operations, data packages
can be replicated to other distribution services.

B. Obtaining a License

In order to get access to licensed content, the client
needs to interact with the licenser service.

In both scenarios, the licensee calls the browseContent() operation (L1) of the licenser service in order to browse
the provided content based on its metadata information.
Then, the client chooses the desired data and requests
the content selected by its identifier content-id (L2). After
optional steps of billing and payment, the licenser creates
a license-id by linking the content-id with the user identi-
fication of the licensee (licensee-id) and the corresponding
license terms. With this license-id, the licensee can access
the data package in a direct or indirect way from a
distributor (or a list of distributors). A list of distributors
for a specific package can be requested from the licenser
(L3).

A license-id is a reference managed by the licenser
service in order to keep track of the licensee, the acquired
content, and the corresponding license terms. The use
of the licensed content is strongly regulated by the license
terms, because there the terms of usage are specified like
for example lifetime of the license, number of downloads
of the data package, availability for direct access by the
licensee, or number of uses by a computing resource.

C. Using a Computing Resource

When using a computing resource, all four participants
have to interact. However, from this point on, the scenarios
differ depending on using direct or indirect access to the
data. Therefore, scenario 2 with indirect access will be
described first and later on the differences for a direct
access scenario will be highlighted.

In the case of distribution of protected content with
indirect access, the computing resource has to retrieve
the data package on behalf of the user. Therefore, the client
calls the invokeService() operation (C2) of the computing
resource (see step 3 in figure 2), which then retrieves the
necessary data package from a suitable distribution service
calling the retrieveContent() operation (D2). The requester
will be authenticated, i.e., the computing resource will
send its resource-id, so that the licenser service can test if
the computing resource is providing a trusted service and
allowed to retrieve the licensed content on behalf of the
licensee (see step 3.1.2, L7). In addition, the distribution
service uses the validateLicense() operation (L4) to check
if the license allows the data retrieval (see step 3.1.1). Af-
fter that, the computing resource receives the data package,
<table>
<thead>
<tr>
<th>Services</th>
<th>ID</th>
<th>Functions</th>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licenser service</td>
<td>L1</td>
<td>browseContent()</td>
<td>-</td>
<td>A possibility to browse the content.</td>
</tr>
<tr>
<td></td>
<td>L2</td>
<td>getLicense()</td>
<td>content-id, licensee-id</td>
<td>Obtain a license for a specific data package.</td>
</tr>
<tr>
<td></td>
<td>L3</td>
<td>getListOfDistributors()</td>
<td>content-id</td>
<td>Retrieve a list of distribution services allowing the download of a specific data package.</td>
</tr>
<tr>
<td></td>
<td>L4</td>
<td>validateLicense()</td>
<td>content-id, licensee-id, license-id</td>
<td>Check if a specific user has a valid license for a specific data package.</td>
</tr>
<tr>
<td></td>
<td>L5</td>
<td>getDecryptionKey()</td>
<td>content-id, licensee-id, license-id, resource-id</td>
<td>Retrieve the corresponding decryption key of a data package.</td>
</tr>
<tr>
<td></td>
<td>L6</td>
<td>retrieveAllLicenses()</td>
<td>license-id</td>
<td>Obtain a list of all licenses of a certain licensee.</td>
</tr>
<tr>
<td></td>
<td>L7</td>
<td>isTrustedService()</td>
<td>resource-id</td>
<td>Check if a service is validated as a trusted service.</td>
</tr>
<tr>
<td>Distribution service</td>
<td>D1</td>
<td>requestUpload()</td>
<td>licenser-id, content-id</td>
<td>Requesting the preparation of a data package, receiving an upload URL and (optionally) a decryption key</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>retrieveContent()</td>
<td>content-id, licensee-id, license-id, [resource-id]</td>
<td>Retrieve a specific data package. If the licenser permits access to the data package only to trusted services, the resource-id of the trusted service is required.</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>replicateContent()</td>
<td>content-id, distributor-id</td>
<td>Initiate a data replication to another distribution service.</td>
</tr>
<tr>
<td></td>
<td>D4</td>
<td>storeReplica()</td>
<td>licenser-id, content-id</td>
<td>Receive a replica from another distribution service.</td>
</tr>
<tr>
<td>Computing resource</td>
<td>C1</td>
<td>browseService()</td>
<td>-</td>
<td>Retrieve a list of available service functionalities at this computing resource (trusted service).</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>invokeService()</td>
<td>function, content-id, licensee-id, license-id</td>
<td>Access a specific service functionality and providing the required data packages.</td>
</tr>
<tr>
<td></td>
<td>C3</td>
<td>getStatus()</td>
<td>licensee-id, job-id</td>
<td>Obtain the current status of a previously requested service functionality.</td>
</tr>
</tbody>
</table>

which can be either encrypted or not. When the data has been transferred to the computing resource, it will also call the licenser service to confirm the validity of the license (see step 3.2). After that the computing resource may need to retrieve the decryption key, if the data is encrypted, by calling the `getDecryptionKey()` operation (see step 3.3). Then, the computations are executed and the results will be delivered to the client.

The common approach to deliver a job description created by a user to a Grid node cannot be used in a DRM-aware system, mainly caused by the usage of executables and data packages which should not be used directly as input for job execution, because the license compliance cannot be tested. Even if a job description is extended with steps of unpackaging and decrypting, the data would be freely accessible by the user on the computing resource and hence can be extracted by the user running the job. The extracted data can no longer be identified and linked to the related license. While this extraction is always possible in a non-protected usage scenario, also protected content can be accessed this way after decryption.

The computing resource is a service provider managing compute jobs. It acts as a portal for compute resources allocated in the Grid and performs the staging phase allowing the usage of data packages of licensed content (including decryption). The client can use the `browseServices()` function (C1) in order to find a suitable functionality. The computation itself still runs on behalf of the licensee, so an additional step of locking the computing resource is required when the decrypted data should not be (directly) accessible to the licensee. The locking of a computing resource can be ranging from an exclusive access to the computing resource for the specific job to a job running in a closed sandbox environment. The main aspect of locking a computing resource is to prevent direct access to the data package by the user or by any other than the corresponding job. A computing resource, which is validated to provide these kinds of services is called trusted.

The alternative approach, which allows direct access to the licensed content, differs from the approach with indirect access only by the way the data packages are retrieved and provided to the computing resource. Here,
the getDistributors() function (L3) can be used to find a suitable distribution service providing access to the data. The client calls the retrieveContent() operation (D2) in order to retrieve the data. In these scenarios, the resource-id of the computing resource is optional. When calling the invokeService() operation (C2) of the computing resource, the user is responsible for providing the data packages himself.

D. Discussion

When the computing resource orders the licensed data directly from the distribution service (see figure 2), the user never gets access to licensed content. This is the recommended scenario, because the content provider can be sure that no one can manipulate or make a copy of the data. In this scenario, a trusted relationship between the content provider and the computing infrastructure is essential. With his valid license and the id of the data, the user is able to start a new computation as often as agreed on in the terms of the license. Although he can not use the data directly, he can retrieve the results of the computation, which is his actual interest. When invoking the service, the data will be transferred using the powerful transference capabilities of the Grid and not the client’s low bandwidth. Of course, the distribution service and the Grid environment must guarantee that the data will always be available and accessible with adequate bandwidth.

In the scenarios with direct access, the user keeps his copy of the licensed data. However, he cannot use it directly as the data may be encrypted. In addition, he cannot process the data by applications in his own infrastructure without integrating the validation process of the license service. Therefore, the data can only be used inside the Grid environment, and the DRM chain is still undamaged. Starting a computation with this data will make the computing resource validate the license by the licenser service. In this scenario, the user will have to store data, which he cannot use outside the Grid environment, thus using unnecessary storage capacity.

V. IMPLEMENTATION

In order to implement the previously described design of the distribution framework License4Grid, several existing Grid technologies are useful. The two main aspects of Grid Computing that are most important for the integration of a distribution model for licensed content into a Grid environment are user management and data access. The following gives an overview of the chosen technologies for implementing the described distribution framework.

A. User Management

The basis for authentication and user management in Grid environments is the public-key-infrastructure. Every user and service on the Grid is identified via a GSI certificate, which is encoded in the X.509 certificate format [22]. The primary tasks of GSI are secure communication (authenticated and confidential communication between elements of a Grid), security across organizational boundaries (without the need for a centrally-managed security system), and single sign-on support including delegation for the users. The delegation capability is an extension of the standard SSL protocol which reduces the number of times the user must enter his password. If a Grid computation requires that several Grid resources are to be used, the need to re-enter the user’s password can be avoided by delegation, i.e., creating a proxy.

The distribution framework uses GSI as a basis to ensure security and mutual authentication of the participating communication partners. Especially, the distinguished name of the user certificates and host certificates can be utilized as the parameters licensee-id and resource-id.

B. Accessing Data in a Grid Environment

Prerequisite for the distribution of content in a Grid environment is a facility to store content and provide access to this content for its users. There are several solutions available allowing access to data in a Grid environment. Some of these solutions will be described in the following with focus on the usability as a distribution platform for licensed data.

The Grid middleware Globus Toolkit [10] provides GridFTP [11] as the basic access technology. GridFTP is a high-performance data transfer protocol optimized for high-bandwidth wide-area networks. The GridFTP protocol is based on the Internet file transfer protocol FTP, but certain protocol features and extensions defined already in IETF RFCs have been added to meet requirements from various Grid projects [12]. GSI is used as the basis for security (see section V-A).

The Reliable File Transfer service (RFT) [18] uses standard Web service technology to manage GridFTP file transfers. Various transfer parameters of the GridFTP control channel are controlled by the RFT service. The properly authenticated and authorized user creates a RFT resource by submitting a transfer request (consisting of a set of GridFTP transfers) to the RFT factory service.

OGSA-DAI [16] is a middleware for wide-area or Internet-scale data integration and allows data resources like file systems or relational databases to be accessed, federated and integrated across the network. The data can be accessed via Web services, HTTP, FTP or GridFTP. The proposed distribution framework uses OGSA-DAI and the related transfer protocols in order to support a huge variety of data sources. The flexibility introduced by the use of OGSA-DAI ensures the usability of the distribution framework within multiple application domains.

VI. RELATED WORK

Most publications containing information about licensed content in digital environments can be found in the area of digital rights management. Basics of DRM and DRM
systems can be found in [19]. [15] describes concepts and participants of content distribution in DRM systems.

Here, the requirements for a content distribution system in Grid environments were examined. While licensed content is frequently used in data processing tasks, so far no specific mechanism was provided to deploy the content in the Grid and maintaining the relation between the content and its license information. Examples for an approach to bring the data into the Grid by the user, and the drawbacks of this approaches, have been discussed for example within the project GDI-Grid (see section II).

A distribution can only take place in a distributed environment. General requirements for the distribution of protected content in distributed environments are discussed in [20], but with a strong focus on media, and not suitable for data processing scenarios in Grid environments.

VII. Conclusion and Future Work

The early work on Grid Computing has focused mostly on the basic realization of the interconnection and cooperation of the participating computing centers, but in the recent years more and more the applications are coming into focus. Some applications require that licensed content is made available in a Grid environment. An example for such an applications has been discussed in the context of spatial data processing (see section II), where content is often licensed content.

Currently, licensed content is either provided manually or with a kind of distribution technology that cannot be integrated directly into a Grid application. In order to improve current Grid applications, the licensed content should be directly integrated and accessible from within a Grid environment. The distribution framework proposed here, provides a solution for an internal distribution of protected as well as unprotected licensed content.

In order to make licensed content in a Grid environment available, several problems have to be resolved. In the solution proposed here, the distribution of licensed content is directly integrated into the Grid environment. Therefore, the requirements for such a solution have been analyzed and used as the foundation for the design of License4Grid.

The solution defines a set of services for licensor, distributor, and Grid node. These services are introduced with example workflows discussing the point-of-view of the service provider. The distribution framework enables the integration and access to licensed content as well as the use of this content on trusted Grid nodes.

Currently, the prototype implementation of the proposed distribution framework is in its early stages. A complete implementation of the discussed services is to be expected some time this year. The prototype of the distribution framework will then be evaluated during its use within the German Grid initiative D-Grid.

REFERENCES


