SeLeNe – Self E-Learning Networks

IST-2001-39045

WP3 Deliverable 3:

A Grid Service Framework for Self e-Learning Networks

George Samaras **Kyriakos Karenos**

Eleni Christodoulou

University Of Cyprus
{cssmamra, cs98kk2, cseleni}@ucy.ac.cy

Abstract

In this report we propose a set of services that are applicable in the case of SeLeNe in alignment to the *Open Grid Services Architecture (OGSA)*. In general, a service-based architecture is proposed that allows a user to submit requests to the SeLeNe Network, viewed as a whole, for the completion of some specific task. At this point, the SeLeNe project concentrates on providing services for the utilization of Learning Objects' (LO) *metadata* based on the RDF/S standards. Service placement produces a number of possible architectural alternatives. Finally, we present issues not extensively covered by the SeLeNe project but constitute basic challenges for future work.

Nicosia, Cyprus

August 16, 2003

SeLeNe Project

Life-long learning and the knowledge economy have brought about the need to support a broad and diverse community of learners throughout their lifetimes. These learners are geographically distributed and highly heterogeneous in their educational backgrounds and learning needs. The number of learning resources available on the Web is continuously increasing, thus indicating Web's enormous potential as a significant resource of educational material both for learners and instructors.

The SeLeNe Project is aiming to elaborate new educational metaphors and tools in order to facilitate the formation of learning communities who require world-wide discovery and assimilation of knowledge. To realize this vision, SeLeNe is relying on semantic metadata describing educational material. SeLeNe offers advanced services for the discovery, sharing, and collaborative creation of learning resources, facilitating a syndicated and personalised access to such resources. These resources may be seen as the modern equivalent of textbooks, comprising rich composition structures, "how to read" prerequisite paths, subject indexes, and detailed learning objectives.

The SeLeNe Project (IST-2001-39045) is a one-year Accompanying Measure funded by EU FP5, running from 1st November 2002 to 31st October 2003. The project falls into action line V.1.9 CPA9 of the IST 2002 Work Programme, and is contributing to the objectives of Information and Knowledge Grids by allowing access to widespread information and knowledge, with e-Learning as the test-bed application. The project is conducting a feasibility study of using Semantic Web technology for syndicating knowledge-intensive resources (such as learning objects) and for creating personalized views over such a Knowledge Grid.

Executive summary

This deliverable is part of the SeLeNe Workpackage 3, on the architecture of Self e-Learning Networks. The objectives of Workpackage 3 are:

- To identify technologies for managing RDF descriptions of educational resources in an open, evolving environment such as a Self e-Learning Network, including issues such as distribution, replication, scalability, load and locality, and define a set of services supporting SeLeNe's functionality.
- To design the high-level system architecture of SeLeNe.
- To produce an Exploitation Plan for the outcomes of SeLeNe.

The traditional approach of storing data in database management systems (DBMS) is not suitable in a Web environment. With current DBMS technology, data are relatively homogeneous, exhibit a small degree of distribution (on relatively few network sites), remain unchanged unless explicitly updated, and are largely static (their location remains fixed). These assumptions do not hold in the semi-structured computing world of the Web, thus creating the need for new foundations for all data management aspects: modelling, storage, and querying. Metadata management in such an environment is crucial. To this end, we envisage that metadata (RDF/S descriptions) and services related to learning objects will be maintained in dedicated metadata stores. These metadata stores will create a backbone of knowledge and services and provide the means for browsing and searching for learning objects The proposed technologies must exhibit a high degree of scalability, allowing new metadata stores to easily join the backbone.

The scope of Deliverable 3, is to discuss techniques for handling the distribution aspects of SeLeNe via the proposition of a set of services to satisfy the User Requirements. These services are based on the OGSA model layering and aim to provide the building blocks for the formation of a set of architectural alternatives for SeLeNe, emerging through different service placement scenarios. Proposed services will relate to the actual areas which the SeLeNe project specifically addresses. However, open issues that are not addressed due to time and resource limitations (namely Security and Replication), are discussed herein. We aim to provide a set of services specifically designed for metadata (RDF/S) manipulation and although investigated in the context of SeLeNe, much of this framework is generally appropriate for other Grid applications that choose to make use of RDF/S.

Revision Information

Revision date	Version	Changes	Authors/Collaboratos
January 15, 2003	0.1	Outline	G. Samaras, K. Karenos, E.Christodoulou
May 2, 2003	0.8	First Draft	Same as above and C. Spyrou, E. Charalambous
June 2, 2003	0.9	Second draft (SeLeNe addresses LO descriptions)	Same
July 22, 2003	1.0	Third draft (After revised User Requirements)	Same
August 15, 2003	2.0	Final version	Same

Table of Contents

1 Introduction

Grid Technology has found uses in a wide area of applications that usually address large scale, process and data intensive problems. Our effort is to bring data-centric services adjusted to the Grid environment and to expand its functionality in the area of resource sharing using e-Learning as the testbed application. As we elaborate in §3, we consider *metadata management* (viewed as semantically meaningful resource descriptions of learning material), crucial especially as the Grid expands to be supplemented with capabilities towards supporting (and incorporating) technologies from the Semantic Web, termed the "Semantic Grid" [39] under the guidelines of the Global Grid Forum (GGF) [13]. We propose a set of data-oriented services, and more particularly, metadata-oriented, that are generic enough to be reusable and flexible enough to fit in different architectural models in terms of service placement. As the spectrum of possible services is quite large and extending in various directions (most of which today constitute individual research areas) we will provide a functional set of services that SeLeNe will specifically concentrate on.

Grid-based educational systems open new ways in the usability of the Grid as their primary requirements include the provision of adequate services for sharing, syndicating heterogeneous resources and relevant content discovery. Efforts are already under way: In [8] an attempt is made to provide an infrastructure for future eScience. Of our interest in this work, is the adoption of a service-based perspective to meet the needs of a global and flexible collaborative system for educational purposes. Three conceptual layers are proposed: The Data layer that deals with the low level, computational and storage resources, the Information layer that is related to the data representation, access and maintenance and the Knowledge layer that addresses some specific high-level problem or objectives while supporting and monitoring the learning procedures. Note that these levels meet the SeLeNe "three levels of abstraction" defined in the User Requirements Document [27]. In [8] the Grid is described as a collection of service *providers* and service *consumers* brought together in a "marketplace", initiated and managed by a "marketplace owner". We parallelize the "marketplace" to a SeLeNe and the "marketplace owners" to the system point of entry, which provides reliability in accessing the system. These we later refer to as *Authority sites*.

It is envisioned that the SeLeNe service layering will enable generic services to be provided which will support the high level Application-specific services. OGSA *GridService* [19] can be adapted to the requirements of an e-learning environment as it provides a process-oriented model on which our data-oriented model services will be based. An example of such an operation could be the submission of a query or a request for collaboration with another user. However, the various services - as described in the OGSA Layers - need not be deployed within every physical site. Each node may create and offer different services to the system, included in a predefined set. It is apparent, also, that services may require the collaboration of many sites (e.g. a query service) functioning in a distributed manner.

An educational environment such as the one envisioned by SeLeNe, however, exceeds the requirement of a standard client-server Grid model. Firstly, information sharing must be extended to the *semantic* level. The semantic extension of the SeLeNe-offered services will aim to address the diversity among consumers and producers of LO descriptions (in addition to services) in terms of ontological contexts. These requirements (among others, discussed in more detail in [27]) would require high coupling among services and the ability for the combination of these services towards the completion of specific e-learning tasks. In addition, the need for personalization - which requires each participating learner or site to be viewed as individual - and for collaboration - which requires for a global view of the SeLeNe components' interaction - impose a model that should functionally allow for the handling of both cases. Although we do view the problem through Grid lenses, we identify the need to incorporate techniques from both Grid *and* P2P technologies. Efforts have already been initiated towards the incorporation of P2P capabilities to the OGSA framework by the GGF community. Although during the composition of this report this effort was still at an early draft state, one can clearly see the practical need for P2P-usable OGSA[1]. To this end, the most relevant work to SeLeNe is done within the SWAP project [40], which combines P2P and Semantic Web technologies to support knowledge sharing. Web Services technologies [44] provide an excellent infrastructure on which SeLeNe services can be build. However we also consider other alternatives, especially in the light of P2P/Grid requirements mentioned earlier. The JXTA project framework [34] offers a purely Java-based services core and concentrates on a P2P-oriented model. On the other hand the Globus project [14] provides a range of basic services for the construction of Grid-based systems. These technologies have been studied extensively as part of previous projects [40, 10, 32]. For the SeLeNe project, we are working on the high-level definition of the required services and the architectural model that would suit the user requirements, assigning much less weight on the possible future implementation alternatives.

We next provide an example of cases of service acquisition to initiate the reader to such a possible model. The simplest scenario is the following: A node 'A' contacts a centralized, known server 'S' and requests a service 's1'. 'S' replies with the location and API description of service 's1' at a server 'B'. Node 'A' communicates in a P2P manner with 'B'. Later the reverse procedure may take place if, for example, 'A' is the owner of a service 's2' requested by 'B'. The difference in SeLeNe is that the Grid itself takes the role of 'S' (by the use of distributed interaction of services.) It is, thus, possible that, after services are discovered, a task may be completed by combining (reserving) a number of available services by the requesting node/site (Fig. 1a) or by the collaboration of several sites through a service composition procedure (Fig. 1b).

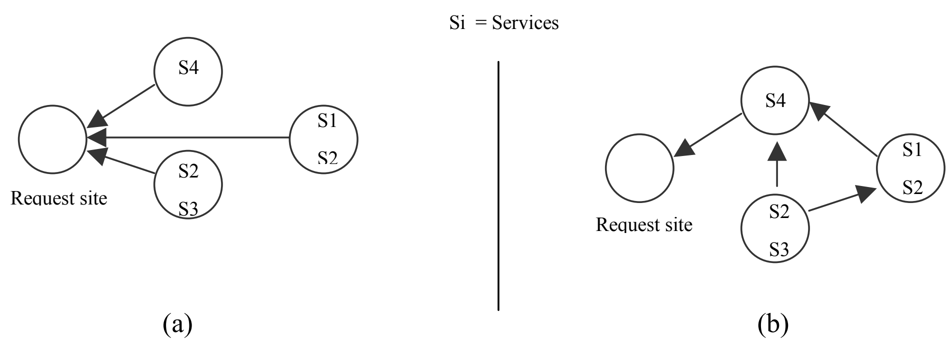

Figure 1: Acquisition of Services

It is important to notice that some vital services need to be available at all times (e.g. registration, mediated querying etc) as well as the fact that we need to provide some method for information integration. Therefore we propose that "authority" sites should be present (such as in a super-peer model [6, 7]) that will be more reliable and may acquire the role of mediator (e.g. to interconnect related sites by clustering) or coordinator (e.g. to support ontology mappings when and if necessary). In more open and dynamic architectures such as P2P, assignment of authorities is a more difficult and therefore we assume that these sites will act, at least, as entry points.

[1] The relevant GGF8 submission is titled "Peer-to-Peer Requirements on The Open Grid Services Architecture Framework", found in the P2P section of the GGF webpage [13].

2 Grid Computing

One definition of the Grid (or Grid Computing) could be:

Applying the resources of many computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data.

It can also be viewed from a software/hardware combination perspective comprising an infrastructure for resource access and sharing within and among virtual organizations [17]. The term "virtual organization" (VO) refers to the individuals and/or institutions that participate in a conditional sharing system. (Rules that may change over time define what is shared and who may access it). VOs are not actually the participants of a Grid; rather a Grid comes into play to address the requirements of VOs: both at the higher level (i.e. solving the problem itself posed by the VOs) and at the lower level (i.e. the problems appearing due to the VOs need for interaction.) Therefore we can now conceive the "Grid problem" definition as being:

The coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.

The key requirement of the Grids is *Interoperability.* However good the mechanisms may be, the global system's benefit may be reduced due to the environmental, platform and language diversity among Grid sites.

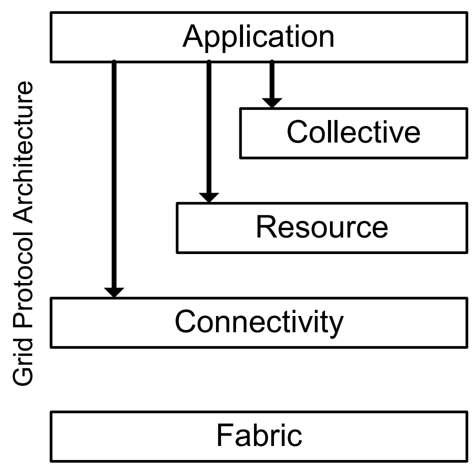

Figure 2: The OGSA Layers

Three basic constructs must be considered to support interoperability: Protocols, Services and APIs / SDKs. Protocols play a primary role to the Grid Architecture which, in fact, is a protocol architecture. Protocols aim at the externals (interactions) of VOs rather than internals (software, resource characteristics) allowing for different implementations for each VO with respect to its internal functional structuring and management. Standard services are also of fundamental importance towards interoperability and are defined within the context of their protocol and behavior. Finally, APIs and SDKs are required to provide abstractions for sophisticated application development.

One proposed Grid architecture, which is widely acceptable in terms of being largely used, follows the “hourglass” principles. The narrow neck defines a limited number of core abstractions and protocols. This is the Resource and Connectivity layer on top of which the broad, Collective layer is built that defines the high level behaviors. The bottom layer, called Fabric, includes the underlying technologies and diverse resources. Figure 2 represents this design.

- **Fabric** The Grid Fabric layer provides the resources to which shared access is mediated by Grid protocols: for example, computational resources, storage systems, catalogs, network resources, and sensors. A “resource”' may be a logical entity, such as a distributed file system, computer cluster, or distributed pool of computers. In such cases, a resource implementation may involve internal protocols (e.g., the NFS storage access protocol or a cluster resource management system's process management protocol), but these are not the concern of Grid architecture.
- **Connectivity** The Connectivity layer defines core communication and authentication protocols required for Grid-specific network transactions. Communication protocols enable the exchange of data between Fabric layer resources. Authentication protocols build on communication services to provide cryptographically secure mechanisms for verifying the identity of users and resources.
- **Resource** The Resource layer builds on Connectivity layer communication and authentication protocols to define protocols (and APIs and SDKs) for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources. Resource layer implementations of these protocols call Fabric layer functions to access and control local resources. Resource layer protocols are concerned entirely with individual resources and hence ignore issues of global state and atomic actions across distributed collections; such issues are the concern of the Collective layer discussed next.
- **Collective** While the Resource layer is focused on interactions with a single resource, the next layer in the architecture contains protocols and services (and APIs and SDKs) that are not associated with any specific resource but rather are global in nature and capture interactions across collections of resources. Because Collective components build on the narrow Resource and Connectivity layer 'neck' in the protocol hourglass, they can implement a wide variety of sharing behaviors without placing new requirements on the resources being shared. Collective functions can be implemented as persistent services, with associated protocols, or as SDKs (with associated APIs) designed to be linked with applications. In both cases, their implementation can build on Resource layer (or other Collective layer) protocols and APIs.
- **Applications** The final layer in the Grid architecture comprises the user applications that operate within a VO environment. Figure 3 illustrates an application programmer's view of Grid architecture. Applications are constructed in terms of, and by calling upon, services defined at any layer. At each layer, well-defined protocols provide access to some useful service: resource management, data access, resource discovery, and so forth. At each layer, APIs may also be defined whose implementation (ideally provided by third-party SDKs) exchange protocol messages with the appropriate service(s) to perform desired actions.

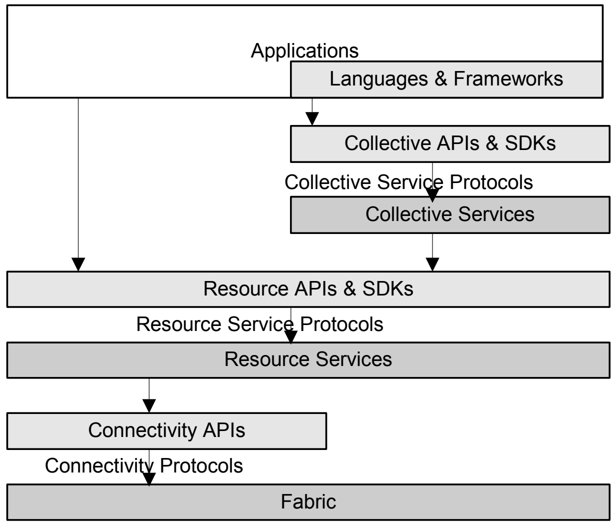

Figure 3: Protocols & APIs in OGSA[2]

[2] Figures 2 and 3 are extracted from [19]

3 An OGSA-guided, Metadata-centric Architecture

It has already been mentioned that SeLeNe is concentrating on the management of LO metadata having in mind, however, the possibility of dealing with LO management itself in a future project. Based on the OGSA service layering presented in §2, we construct in this section a corresponding layered set of services required for a SeLeNe. It is noted that these services derive from the User Requirements document [27] and cover the functionality and specific issues that are investigated more extensively by the SeLeNe project.

Management and manipulation of LO metadata is at least as important and critical as LO management itself. In an educational system, content descriptions are crucial in order to provide a uniform method for the discovery of LOs relevant to the user's queries and for combining multiple such descriptions to realize specific tasks that lead, eventually, to supporting the learning objectives. Additional requirements such as personalization support, change notification and automatic/semi-automatic metadata generation are only indications of the demanding nature of metadata management. When addressing large data set services in OGSA layering, metadata handling is usually present at the Resource and Collective layers. In our case, as metadata is the actual shared resource and due to the mentioned requirements we believe that it is required to provide metadata services covering all layers, thus making service break-down concrete. For example there exists the need for descriptions of LOs to be accessed, manipulated and stored in an RDF repository (i.e. the Repository's API). This is suitable to be included to the Fabric layer services since a number of different storage alternatives may be present. On the other hand there exist high-level services that will support Trails and Personalization of LO descriptions that need to be placed at the Application layer.

3.1 Service Classification

We have mentioned that it is understandable that not all services can be deployed at each and every SeLeNe-participating site. However, we feel that it is a requirement that there should be an as-small-as-possible set of specific services that each SeLeNe site will assume present in all other SeLeNe sites (i.e. provided by all participants or their immediate proxies). The basic reason for this is to make sure that at least communication and discovery of available services will be possible as soon as a single SeLeNe site is identified as an entry point. These services, we can call *Core Services.* Additional A*ppended Services* will be present for the materialization of the larger percentage of SeLeNe functionality.

Regarding “core” Grid services in general, there has been extensive discussion and it is still rather unclear as to which services should be characterized as such. There has been an initial proposal for generic service support interfaces for OGSA, introduced in the “Physiology of the Grid”[19]. This work attempts a functionality-oriented approach to OGSA services. More specifically, the OGSA Grid interfaces include the following[3]:

- GridService: Provides information about Grid Services, multiple query support and manages the termination time for a service instance. We note that this is actually the *only* required Grid interface, whereas the following are optional.
- Notification: A subscription service to carry out synchronous delivery of notification messages regarding service-related events.
- Registry: Provides the service handles registration/deregistration facility.
- Factory: Allows for the creation of service instances.

[3] As mentioned in the same work, the reference interfaces are likely to change in the future.

- HandleMap: Provides references to services' handles.

One can clearly see that the proposed interfaces are, to a major degree, process-centric. In SeLeNe, however, RDF metadata is the actual resource and for this reason we should provide additional or adapted interfaces to meet a, to a large extent, data-oriented system. This characteristic does not, in any way, cancel or even restrict the use of a service-supported scheme. For example, consider SeLeNe's data-oriented requirements compared with the EGSO project [11], which indeed concentrates on combining heterogeneous solar observation data to create a “virtual dataset” (in a way, related to the SeLeNe LO metadata learning space). What we are envisioning is that the set of proposed services will be possible to be deployed in alignment to the OGSA guidelines (and possibly over widely used grid technologies such as Web Services and Globus) but also extended to provide additional functionality that is missing from today’s Grids but required by an e-learning network (i.e. P2P support and expanded semantic (RDF) metadata usage). In this sense, existing infrastructure can be utilized to *mediate* the underlying service functionality described below but targeting to support RDF descriptions as the requested resource (i.e. instead of computation cycles, storage, large data objects etc.). As argued next, generic RDF services can then be adopted by other grid systems.

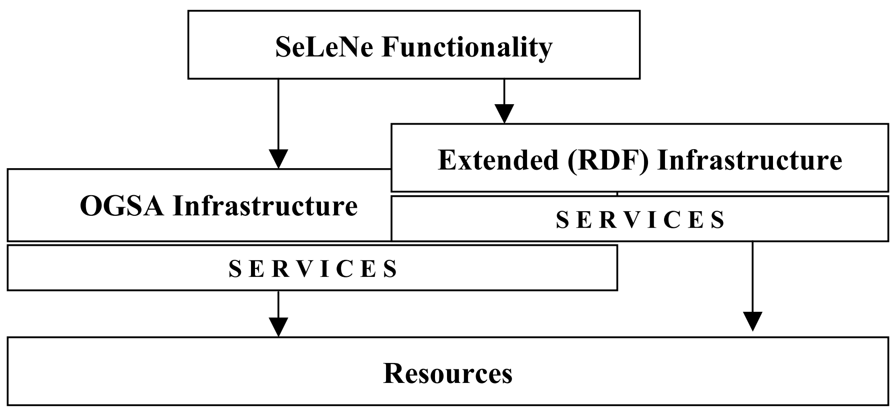

Figure 4: SeLeNe and the Grid

Besides characterizing services as being either core or appended, one other important distinctive factor for offered services is whether a service is *generic* or *application specific* (i.e. SeLeNe specific). Generic services will most possibly reside at the “hourglass neck” of the OGSA layers. These services will be usable for other applications or systems that require or make use of RDF. Examples of generic services include RDF view creation and change notification. On the other hand, application specific services concentrate on the specifics within the SeLeNe project with respect to the e-learning requirements such as trail management and personalization services.

In the following subsections we will describe the high level, expected functionality for each of the proposed services and indicate the specific research issues that concern the SeLeNe project and, when necessary, refer to relevant in-depth work in other Work Packages.

3.2 Core Services

- **Access Service:** This service is located at the lower layer of the Grid Architecture (Fabric). This service provides the direct access *API to the local RDF Repository*. It includes access methods for local requests as well as appropriate manipulation of locally stored descriptions (i.e. insert, delete and update of the repository content) irrespective of its low-level implementation. The actual *storage repository* can be realized over a number of implementation alternatives such as Sesame RDF Repository [35], Jena toolkit [24] and the ICS-FORTH RDFSuite [22].

- **Communication Service:** This service provides the basic communication mechanisms for exchanging data. Current protocols may be used on which communication links can be established (such as TCP/IP) but we should also consider creating a simple "SeLeNe specific" communication service (i.e. for the exchange of specific types of messages e.g. task request submission.) Possible example technologies that can support this "SeLeNe specific" communication service are SOAP [37] and RPC techniques (e.g. Java RMI), however the message content and structure is expected to be defined in detail in a subsequent project as it is not part of our current investigations. RPC is generally more appropriate for more formalized and concrete (e.g. local) communications and can be used in a local SeLeNe (e.g. installed at a single institution). On the other hand SOAP addresses incompatibility problems among multiple communicating and possibly remote groups.
- **Information Service:** The Information service provides the capabilities of acquiring descriptive information on some SeLeNe site. Informally, it will be able to answer questions of the form: "what does this node understand in terms of metadata?" It provides the profile of the site (not the user). Put in another way, it provides metadata on metadata and more specifically the *Namespaces* used and the *RDF Schema(s)* for that specific site. The Information service is built on top of the Access service. It does not raise any new research issues for us.
- **Query:** The Query Service is of great important: we need to define a powerful query language that will allow for the extraction of results from multiple, local RDF repositories. This is to be addressed in depth in *deliverables 4.2 and 4.3*. The Query Service should be distributed and should allow for search message routing in order to forward sub-queries to sites that can provide answers. It may also need to call the Syndication service to translate queries expressed against one RDF taxonomy to sub-queries expressed against different local taxonomies. It then passes a subquery to the Access service supported by a particular peer, expressed in terms of that peer's local RDF Schema. Another issue is the exploitation of the semantic meaning of our data to relate users of similar interests. A good, super-peer based technique is provided in [28] where a clustering technique is used to mediate heterogeneous schemas. Authority sites can become responsible for keeping semantically meaningful indexes about other neighboring sites.

3.3 Appended Services

- **Sign-on:** A site is able to register to the SeLeNe in order to advertise its content and services. Also, in this way, it should be able to make its presence known to other sites. Sign-on allows for the update of the indexes of neighbors as well as the directly connected authority site(s).
- **Locate:** This service relates to the OGSI GridService and makes requested service lookup possible. As soon as a site is connected, it should be able to discover where there are services that will be used, along with any required parameters that these services will need. We assume for now standard registry techniques depending on the architectural deployment of SeLeNe. A distributed cataloging scheme could suffice in this case (e.g. UDDI [41].) Semantic service descriptions is an issue not addressed within SeLeNe for now although it does pose an interesting future research issue for the evolution and expansion of the proposed set of services.

- **Syndication:** The Syndication service is responsible for the translation between different RDF schemas. This is accomplished by using the user-supplied mappings between heterogeneous schemas. This implies both data-to-data and query-to-query translations. Syndication issues are also of high importance and will be addressed in *deliverable 4.1*
- **Update:** The Update Service is used to appropriately transfer updates to descriptions expressed in diverse schemas. By analogy to the Query service, this service will take an update request for Peer 1 expressed in some RDF_Schema_2 and translate it into the equivalent update expressed in terms of RDF_Schema_1 by using the Syndication service. The Update Service would then request for the invocation of the appropriate operation of the Access service at Peer 1 to enact the actual update on its local RDF repository.
- **ECA:** LO descriptions are gradually updated and enhanced due to the ongoing learning process. Users should be able to register their interest to receive changes when they occur that are relevant to metadata that are of their interest. This feature should be provided by the ECA Service, which will propagate updates and notifications to registered sites. *Deliverable 4.4* describes in detail the functionality of such a service.
- **View:** The View Service provides the functionality of creating personalized views by structuring (and re-structuring) virtual resource descriptions among the SeLeNe LO descriptions' space. By this way we allow for the user to actually built-up her own virtual learning environment which she can navigate and expand. The View Service will can be realized over RVL that is able to, additionally, allow the definition of *virtual schemas* and thus amplifies the personalization capabilities of the SeLeNe system. RDF view creation is addressed in *deliverable 4.3* in this project.
- **LO Registration:** This service provides the API for submitting a new LO by providing its description to the SeLeNe. Storing LO descriptions is handled by the use of the Access service. The registration process makes use of the Syndication service and allows the registration of both atomic and composite LOs. We refer the deliverable 4.1 for a more formal explanation of the registration of composite LOs.
- **User Registration:** The user will be registering to a SeLeNe in order to create and later use her profile and thus acquire a personalized view of the system. User descriptions are also stored using the Access service. Issues of costing are not considered at this moment as we focus mainly on the personalization/profile creation aspect of the User Registration facility discussed in more detail in d*eliverable 4.2*. Security issues also remain an open issue discussed in §5.
- **Trails & Personalization:** The Trails & Personalization Service is related to a specific user or group of users. It is concentrated on the educational characteristics of the user and provides the API to extract user-profiling information. It is proposed that this service should run as a user-side agent when possible while trails could be formed and managed by message exchanging of the participating person or group agent or agents. *Deliverable 4.2* discusses in further detail personalization issues.
- **Collaboration:** A Collaboration Service should allow the communication between users and groups of users and it is proposed that this is mediated by a central authority site. At least two sites should request the creation of a collaboration session and others may be added later. Collaboration services may include already available systems such as Blackboards, Message Boards, CVS (for collaborative code writing) or e-mail and instant messaging services. The SeLeNe Collaboration Service lies above these services in order to provide connections to other SeLeNe services. Collaboration issues are also present in *deliverables 4.1 and 4.2*
- **Presentation:** Based mainly on the user profile, the Presentation service should be able to produce graphical visualization of metadata. This could, for example, be a RDF graph. It could also be produced locally or via a web-based engine. Since visualization and presentation are highly related to the learning experience itself, there is no simplified methodology for it and will most possibly require much work. Some

presentation aspects are included in *deliverable 4.2* but are generally beyond the scope of the present SeLeNe project. Detailed GUI issues may be addressed in a subsequent project.

	Service Name	Core	OGSA Layer
Application Specific	Presentation		Application
	Collaboration		
	Trails & Personalization		
	User Registration		
	LO Registration		
Generic	View		Collective
	ECA		
	Query	X	
	Update		
	Syndication		
	Locate		Resource
	Information	X	
	Sign-on		Connectivity
	Communication	X	
	Access Service	X	Fabric

Table 1: SeLeNe Services

3.4 Availability and the Need for Authority Sites

Defining the set of services needs to be complemented with a discussion of the need of these services being available to the users and that indeed the user will be able to access it to be able to exploit basic and advanced functionality. We thus propose the existence of "authority" sites primarily acting as the system's entry point. Until now, a general understanding of the functionality of "authorities" has been discussed in this document. In this subsection we consolidate our previous dispersed references to "authorities".

Authorities are similar to super-peers in a hybrid P2P model or brokers in a Grid model (although they provide more than just brokering services in the SeLeNe context). They are considered more reliable, offer persistent services and are characterized by their static nature regarding their participation in the system. As such they can be used to provide persistent services including Sign-on, Syndication, Collaboration and User Registration.

Authorities may take care of search message routing as well as integration/mediation of metadata [4]. They enable the creation of clusters of relevant sites via evolving indexes based on the semantic description of the sites' content.

Authorities can also serve as the third party in an authentication process between two sites as well as allow for the collaboration of multiple sites for the creation of collaborative trails.

The second important aspect of the system's functionality that must be ensured is service availability, which unfortunately does not seem to be extensively addressed in the various Grid-based systems. Informally, the question is: "Can we be sure that a service that will be requested will in fact be available?" The primary goal is to have multiple instance of the same service running at different sites, therefore having more than one "starting point" (such as in DNS).

A functional framework, with respect to SeLeNe, for supporting service availability is provided in [3]. The proposed framework is based on the assumption that a set of services exists (such as our proposed SeLeNe services) on servers able to communicate with each other (also possible with SeLeNe). Each service is stateful and allows access to user sessions. During a session, other servers are notified of the current state of the user. Therefore, in case of failure, another server can continue offering the service. Not surprisingly, the authors in this paper selected a "search service" (query) and an "educational service" (where a user/student accesses LOs) as two of the basic example scenarios.

Some additional ideas on how we can maximize the possibility of some service being available, are proposed below:

- Basic services are made core.
- Reliable sites (authorities) provide popular/demanding services.
- Allow for appended services to be added (installed) onto sites (Service replication).
- Break services to smaller components (or services).
- A site may include some (appended) service but may not have it "running" (e.g. to preserve resources). Remotely starting a service may be done via any one RPC implementation.
- Handling of predictable service shutdown (i.e. when a site voluntarily disconnects from the Grid).

4 Approaches to Service Placement

In the following subsections we will give some examples of the architectural designs that result due to different service placement decisions. We also mention some relevant projects and systems that deal with similar problems and comment on the selected architectural choices.

4.1 Centralized

One first approach is to take a look at the centralized scheme. In such an architecture, a number of "fixed" service providers exist which are highly available and powerful enough to accommodate a large number of services. Provider servers are inter-connected and together they provide a service provision cluster. Clients (or consumers) connect to the cluster via a specific entry point or an Authority. Authority servers are also used for coordination purposes.

Metadata located at consumer sites needs also to be registered at some cluster server. In this sense, servers act as metadata repositories for LOs. Query and Integration/Mediation services are provided for metadata among the servers and replies are sent back to the requester. Since all tasks are handled within the group of servers, consumer sites are not actually aware of each other.

The previously described strategy for placement of services is similar to a brokering system. An example of such a system is the EducaNext/Universal Project [9]. Universal uses a centralized web-based front-end for the submission, retrieval and costing of LOs.

Metadata is provided directly by the submitting side, again via a standardized input form.

4.2 Autonomic

An autonomic system is characterized by the fact that each site is autonomous in terms of service provision (i.e. each site may provide any number of services). In such cases, a core services requirement is the existence of a Service Discovery protocol (such as the previously described Discovery Service), which should be completely distributed.

Metadata is maintained at each site and there is no centralization. Therefore, a distributed and possibly partially replicated metadata catalog should exist to address intermittent connectivity issues. Each site must also act as a message forwarder. Intelligent techniques in this case need to be devised to enable efficient and semantically meaningful query message forwarding [34]. Also, since there is no centralization, other issues are raised related to service availability.

One such autonomic (P2P) approach is found in the SWAP project. The core difference however is that SWAP is component-based, not service-based.

We now elaborate more on autonomic sites systems identifying specific characteristics of SeLeNe that require P2P functionality support. The first logical observation is that a single SeLeNe site may represent both an LO producer and consumer simultaneously. Additionally, as experience and real systems have shown, the essence of Grid and P2P computing is *resource sharing.* In this sense, novelties and initiatives from both “worlds” can be blended together to address fine-grained user requirements. Below, we provide a list –in order of importance with respect to the SeLeNe goals– of some specific requirements that directly relate to P2P characteristics:

- *Cases for Autonomy:* One of the top contributions of P2P computing is support for autonomic behavior. Nodes may enter and leave the network without, in most cases, seriously affecting the rest of the participating peers. In addition, a peer may perform system activities solely based on its locally installed components. In an educational system, in this sense, autonomy is important. Independently holding information at individual sites, either as an LO producer or consumer, will satisfy two particular needs: distribution (i.e. avoiding centralized, single-point-of-failure storage of both local information as well as global catalogues) and personalization (i.e. local profile which do not completely depend on a remote broker and can be edited at any time). Finally, a SeLeNe should support *monitoring* capabilities: seamless metadata change propagation or data pulling and user-transparent trail/personalization monitoring (i.e. user agents).
- *Scalability:* The vision is to be able to have ranges of participation, from single institution to multi-institutional SeLeNes. In this way it could accommodate "small", "medium" and "large" users in terms of contribution to the system and this is exactly how the selected architectural design is affected. This also relates to the types of supported learning (directed, undirected, self learning). For example, a university providing a curriculum is a "large" LO producer but if it participates in a community of universities providing that curriculum, its contribution may range.
- *Intermittent Participation:* It is unavoidable that a participant will enter and leave the network in a non-deterministic manner. Therefore it is highly possible that information (metadata) will not be available at any time. Thus SeLeNe should be able to cope with connection intermissions as well as unpredicted failures during system functions especially in the cases of querying, system login and registration.
- *Ad-hoc collaboration:* Collaboration was one of the driving forces behind the idea of P2P systems that bring together the network edges. As soon as the collaboration nodes identify each other we would prefer that the joint production of composite LOs and their metadata will also be possible without brokering/authority intercession. This would also require that the collaborating sides collectively support the required services to complete such a task.

One final characteristic that is desirable in SeLeNe, and that also constitutes a P2P requirement, is that each site may act as both provider and consumer. In this case the shared contribution of the "peers" is services. However, a single site may not be able to accommodate all peer components (or all peer services) in which case the whole view of the system somewhat diverges from a true P2P paradigm. In this case the service provision scenario fits better. A "thin" client may be able to complete a complex task via the utilization of SeLeNe-provided services. (For example, consider a web-based portal service that provides querying, visualization and integration capabilities to a PDA user.)

We have already mentioned the efforts taking place at GGF towards OGSA P2P support. In the document produced, it is also stated – and we agree – that extensive support for P2P environments will require a new global infrastructure. Therefore, in addition to these efforts it is expected (again as this report is being produced) that the new version of the Globus Toolkit (GT3) will adopt open protocol standards also applied in Web Services technologies. An improved OGSA specification in combination with GT3 support for standard technologies will bring this goal closer to realization.

4.3 Mediator-Based

In a mediation-based scheme, consumers and producers (of both LOs and Services) are logically clustered around mediators/brokers that in our case will be taking the role of Authorities. This is also known as the Consumer-Broker-Producer model (in terms of services) and also resembles the super peer scheme (in terms of content). Sites may be both LO producers and providers but need to register their content to a broker which will provide the means for communication with other sites by creating logical communities. This last characteristic is highly desirable in SeLeNe.

Edutella [10] is a mediation-based educational system, probably the most relevant to SeLeNe, built on the infrastructure provided by the JXTA Framework.

5 Open Issues

5.1 Security

In this subsection we will describe the general issues and concerns relevant to security aspects of SeLeNe. The vastness of security concerns makes it impossible to concentrate on all of them at once. We can distinguish three very wide categories regarding security: physical, technical and legal. Herein, we will focus more on the technical aspects. Still, this is again a very large area, by itself, which includes, among many others, the definition of system security requirements, policies for the grid, protocols, management, performance and accounting. Therefore, we will attempt to identify the intersections between high-level grid security requirements (and proposed solutions) and a self e-learning network.

5.1.1 Overview of Underlying Platform Security Infrastructure

The Grid: OGSA and Globus

Security concerns in a Grid environment go far beyond the traditional establishment of trust relationships between a client and a server. Multiple distributed and heterogeneous individuals providing resources and services to the Grid form virtual organizations (VOs) each of which must confront diversities of access technologies among its participants but also among other VOs. Security Policies for VOs produce an *overlay* that bands together the various individuals under a common trust domain [43]. The emergence of such overlay is due to the requirement for secure communication and coordination among multiple processes spanning the grid network and the its dynamic nature which cannot guarantee stability in the establishment of trust relationships.

We provide below the description of three of what we believe are the major characteristics of the grid that directly relate to our high-level security view of this project. We do acknowledge that all characteristics must be considered, however, as discussed below, our considerations for now will not concentrate on deployment specifics but on the most basic requirement for ensuring the functionality provided by the proposed services.

- Single sign-on: This characteristic is particularly important as it allows for a user to connect to the system once, provide its credentials and then allow for the execution of the tasks requested by different service calls, residing at different sites.
- Interoperability: It cannot be assumed that each site will change its local security mechanisms to be admitted and participate in the system. Thus, a layer of interconnection among domain specific and local access mechanisms should be supported.
- Security for collaborating groups: Support for group communication is a feature currently missing from the grid. As collaboration is a key requirement for an educational system, solutions should be provided to coordinate processes originating from requests for collaborative group tasks.

Work within the GGF concerning security has produced the Generic Security Services Application Programming Interface (GSS-API [29]), which can be implemented to allow for the authentication procedure to be completed among diverse local mechanisms. The Globus Toolkit's implementation of the GSS-API is named Grid Security Infrastructure (GSI) [18] and defines and implements a set of protocols that provide a security solution for the coordination of diverse access control policies.

Web Services

Web Services provide an opportunity for the exploration of Grid Services deployment. The main focus is currently the production of security specifications that mainly encode established security solutions in SOAP messages. These include: WS-Security [20], a specification of the format of messages for security tokens exchange, WS-Policy [5] to support the advertisement of the communicating sides' policies and WS-Trust [21] (which builds on WS-security) to manage trust relationships.

JXTA

In JXTA it is also expected that existing, trusted technologies, will be easily incorporated to work with the provided infrastructure. However, in JXTA another interesting issue is emerging which relates to the fact that P2P environments will have different requirements from the well-established centralized security solutions. For example, in such environments with no centralization assumed, a peer may choose to avoid contacting a third party certificate authority. To this end JXTA has adopted Transport Layer Security (TLS) to support reliable private connections between peers and uses X509.V3 digital certificates to allow for both the usage of centralized Certification Authorities, if required, as well as each peer acting as its own certification authority.

5.1.2 SeLeNe Considerations

Although the SeLeNe project does not specifically address issues of security it still emphasizes the possibilities of making the most of the current work and underlying security infrastructure being developed for the Grid environment including previously mentioned solutions provided in GSI, Web Services and JTXA for challenges such as single sign-on and delegation of trust for service interactions. In this sense participating description producers and consumers will be, themselves, forming a trust domain i.e. the SeLeNe "virtual organization" with authority sites taking the role of Certification Authorities.

We expect to be able to utilize existing infrastructure, having SeLeNe services operating over or in parallel to the existing technology and raise issues for extending them to address other functionality:

- A basic issue is the definition of the content of the local security policies that must: (1) define the restrictions regarding the creation and execution of service threads as well as which *parts* of the API can be executed and by whom and (2) define access rights for the underlying description stores (over the RDF repository facilities.)
- SeLeNe aims to be a collaborative system in which sharing is part of its core functionality, secure group communication and access is a principal issue. As mentioned earlier there's still work to be done at the GGF regarding the proposal of guidelines and solutions on secure group communication[4].
- Define the point of interaction among SeLeNe Services and the existing security technologies (possibly via the GSS-API implementation.)
- Deciding which sites will be acting as Certification Authorities but also allow for more dynamic environments to be functional when a centralized solution is not completely applicable (e.g. exploration the case of JXTA Security alternatives)

[4] More information on current work can be found at GGF [13] – Security Working Group.

- Additional attention must be given to issues of digital rights management to enforce copyrights associated with Web resources. This can be investigated with respect to the referenced resources within the descriptions.

5.2 Replication / Caching

5.2.1 The Role of Metadata Replication

Replication is the service that allows for multiple copies of the same object to be available at multiple locations of the Grid. The importance of replication has been stressed a number of times therefore we will very briefly repeat the need for practicing it. Our attention turns to specifically addressing replication issues for RDF metadata. Caching can be considered a loose form of Replication in that the latter requires that all copies be kept in a consistent state whereas, for the former, an invalidation massage may suffice. Some aspects of caching which are related to semantically meaningful data are described in a Section 5.2.3.

Two main benefits of replication are the achievement of lower data access response time and higher data availability. One would argue that this is the case usually for large data sets (such as in the case of EU DataGrid Project [38]) that require large time intervals to be downloaded. However, as metadata provide the means for looking up and discovering the actual objects, the total access time can be radically reduced with metadata replication. The main reason for this is not the actual size of metadata - besides its size *should* be small - but its high distribution to a large number of sites. Keeping the search space limited will increase efficiency. On the other hand availability is a key issue for metadata. In dynamic environments where providers enter and leave unexpectedly, we need to provide methods for some kind of guaranteed lookups. Especially, in the case of LO metadata – of *semantic* extension over the use of RDF – the additional benefit is clearer as RDF metadata encapsulate more information than plain data object:s a user may discover information on some requested LO even if the LO itself is unavailable at the time through its contextual metadata.

Another important benefit with the ability to replicate is that interesting LO metadata to some learner can be "cached" locally and later receive updates on changes by the originator. This process is particularly helpful in the educational environment where knowledge is frequently extended (e.g. new learning resources attached or readjusted to some educational ensemble such as a class.)

Finally, metadata can be forwarded and/or copied and then processed at sites that provide some specific service not available at the source site [46]. In this way we can take advantage of the full, provided functionality.

With replication we come face to face with the classic trade-off of availability and low delay versus consistency. This is related to the two main issues arising with replication, discussed in the following subsection. These are *replica placement* and *consistency* i.e. update propagation. Of course there is not one single perfect solution as a number of factors affect the replication strategies, especially since we are addressing RDF metadata replication. These include:

- *The Architecture:* Obviously an architectural design with a small number of centralized Grid servers tends to be less complex than an open and unstructured P2P environment.
- *Replication Policies:* Policies answer questions such as who owns a replica, whether a primary replica exists and whether updates will be *pushed* or *pulled* to or from one or among many (all) replicas.
- *Degree of Heterogeneity* and supported protocols: The replication service must be built on some common ground so that participants will be able to inter-communicate for message

exchange. Of course, this is still an ongoing activity within the standards organizations and working groups (such as GGF).

- *Network:* At a lower layer, the actual network infrastructure plays an important role. Replication and data transfer over high-speed networks will pose different challenges compared to a low bandwidth (e.g. wireless or mobile) environment.

5.2.2 Functionality

Generally there are two issues to be considered, which are highly dependent on the individual application. The first is *Data Placement*, which deals with the algorithmic details of when and where the replication process should be initiated as well as the building up of the replica catalog. The second is *Replica Consistency*, which deals with replica update propagation. The basic assumption within SeLeNe is that a specific portion of metadata is kept by a single owner and may be changed only by that owner. We mention “portion” (also referred to as “fragment”) of metadata, not metadata on some specific LO since LO metadata may be under group ownership.

5.2.2.1 Replica Placement

Replica Placement is probably the most complex problem with respect to replication as it presents a number of challenges. Herein, we will not consider cases of the centralized client-server model but instead we look at cases of multiple sites where storage is possible, namely from hybrid (brokering) models to open P2P systems. These challenges include:

- Coordination among the participating sites. As sites become more and more autonomous the coordination becomes more complicated as nodes enter and leave dynamically.
- Sites may have different contributions to the system. In addition – and relevant to this – each site may have different storage and processing capabilities.
- The issue of adjustability is of primary importance, as popularity of available metadata will be changing over time.

In turn Replica Placement comprises of two lower level services that are responsible for: (a) running the algorithm(s) for deciding when to initiate the replication process and where to place the replica (or replica fragment), (b) building up the replica catalog which provides the means of mapping logical requests for a part of the distributed catalog to the actual physical location of RDF fragments at some site metadata repository.

Based on the challenges mentioned earlier, replica placement can be examined from multiple angles. First one can investigate algorithms and policies for the initiation of the replication process based on different or joint costs (such as response time, transfer delay, processing load, reads-vs-writes requests etc.) for multiple autonomous or semi-autonomous sites [16, 31, 23]. The particular case of Grids has also been studied in [16] concentrating, however, more on large data object replication and issues of load balancing rather than metadata. Another way to look at the replication process is through different structures (as part of the architectural design) [25, 45, 16] as well as each specific architectural approach chosen as presented in §4. This last perspective is closer to the scope of our project as, due to its support for semantic data interpretation, it allows for the building up of communities. As examples we note the so-called *Clustering techniques* proposed for XML-based data placement and lookups [4, 28]. It is highly recommended that clustering techniques should be used mainly because of the fact that access to popular metadata may imply access to metadata referenced by this metadata.

Replica Cataloging deals with creating the mapping structure from logical to physical entities (a one-to-many relationship). Of course it is not completely independent of the placement algorithm but, as a service, it requires a separate access interface. As our RDF metadata space will be used for resource (LO) discovery, we investigate similarities to the UDDI cataloging specification and in particular, its replication scheme [42]. There are actually three important analogies: (a) The whole system appears as a single service regardless of the number of replicas. (b) In UDDI, each node is assigned "custody" to portions of the catalog. (c) Changes only take place at the "owner" of each portion (the *primary copy* approach). Change notifications, update request and update propagation exchange messages in UDDI is achieved using XML-SOAP.

5.2.2.2 Replica Consistency

Again there exist two issues related to consistency: first, the architectural model to use i.e. the change originators, receivers and any components in between; second, the change propagation strategy i.e. the actual change/update communication protocol.

A widely used solution that fits the service-based model is the *Publisher, Subscriber and Distributor metaphor*. As implied by the name, three components are implicated: the Publisher (or Publication service) which is responsible for providing the data to be replicated, the Subscriber (or Subscription service) which allows for the replication site to request updates and the Distributor (or Distribution service) which is responsible for a storing and forwarding updates from publisher to subscribers. The distribution process can be a remote replication sub-service, most probably available at Authorities thus it will not be required that a distribution service is running at each publishing site. In [31] a three tier architecture is proposed for managing distributed metadata based on the publish/subscribe paradigm making, however a clear distinction among metadata providers and clients who gain access via Local Metadata Repositories. Although the use of intermediate Repositories that handle query processing is an interesting idea, still in SeLeNe we need to go further and also consider the fact that sites can be both publisher and subscribers.

With respect to the change propagation protocol, Nejdl et al. propose a modification exchange language for RDF in [46]. In the same work some additional issues on choosing the proper replication strategy are presented. However issues of change detection are not addressed. In the publish/subscribe model presented in [31], rules are used to register the interest of subscribers, which is closer to our proposed service approach. More specifically, in SeLeNe we are able to provide update capabilities supported by ECA rules framework, which allows for the definition of the update firing conditions as well as update propagation itself (see *deliverable 4.4*).

Transactions

The assumption made in SeLeNe concerning RDF descriptions for a specific LO or user, is that these descriptions are stored in their totality on one node. Local transactions' issues are well understood and are assumed to be handled by the underlying local repository as part of its provided functionality. (For example Jena's SQL backend supports transactions through the transaction isolation capability of the underlying database. Also, at the application layer, the transactional aspects of the Universal Project mainly rely on InterBase's facilities).

Deliverable 4.4 provides an in-depth look on a proposed notification mechanism based on ECA rules. We clarify that when initiating ECA rule execution, atomicity is crucial. The whole nested and possibly distributed transaction should either all be executed or none of it. As the architecture moves from a centralized to a P2P model, synchronicity becomes more complex due to possible node unavailability. This issue is discussed further in *deliverable 4.4.*

We currently adopt the model of single owner for a specific description for which this owner and this owner only can provide updates to. Therefore the following cases are possible raising transactional issues:

- Updates produced from single owner to her descriptions. Correct execution of transactions is handled by the local repository.
- Descriptions created under group ownership. We need to consider the possibility of having a primary copy placed at an Authority site. In Edutella's MEL, transactions comprise of a group of modification commands, executed as one chunk. In this case insertion, deletion and update of RDF triples is, in essence, analogous to the multi-client, single server model.
- Creating taxonomical descriptions of composite learning objects from the taxonomical descriptions of their constituent objects, which may change. The composite object description, if materialized, needs to be considered as the primary copy and placed at a specific Authority site or, alternatively, at its single registration site. This is a more complex problem and analogous to the data-warehousing problem.

5.2.3 Caching

As argued earlier, caching can be viewed as a relaxed form of replication. Still, (and possibly for this reason) it has received much attention especially as web caching has been (and still is) a highly addressed research area. Caching in our context has more to do with availability observed by an individual user or group of users linked by common interests rather than response time with respect to the previous discussion on RDF replication. In this way, the caching service will be able to serve a local or multiple remote participants (possibly not able to deploy caching services themselves locally).

We identify two basic aspects in this area: caching of RDF query results in a centralized repository and caching distributed RDF query results. As in RDF description semantics are inherent, we note the relevance of the work on "semantic caching". The main goal is the grouping of semantic regions that satisfy some constraint (in a way similar to communities). This is then reflected in SeLeNe through the process of querying i.e. "semantic regions" or communities correspond to queries and vice versa. Additionally, we can then be able to create views forming virtual communities. Semantic caching is addressed both for centralized environments [2] as well as distributed ones [36]. Distributed caching can also be distinguished into *local cache per site*, i.e. applying caching mechanism for each single site in a autonomous way and *distributed cache per "community"* which would require an underlying network structure e.g. the use of node adjacency techniques from structured P2P networks. We next list a number of basic considerations related to caching:

- As results that need to be cached are represented by queries, it is important to identify when the materialized query results are to be stored at the site running the caching service. One possibility is extending containment query mechanisms to materialize the result (locally) when appropriate. Proposals for clustering structuring are present in [28] and the JXTASearch architecture as part of Project JXTA.
- A second consideration is where caching takes place. This is also dependent on the architectural model selected.
- Another basic issue is the selection among replacement policies. Common techniques such as LRU and MRU may suffice for specific local caching policies but may not be adequate for distributed community caching. Additionally, these techniques do not consider dependencies and relationships among descriptions.
- Possibilities to be investigated also include caching of *namespaces* and caching of *trails.* In the Sesame/SAIL API [35] selective caching of namespaces is supported as an optimization.
- We also propose the investigation of extending or reengineering caching techniques for unstructured networks in which case site autonomy and intercommunication complexity with respect to caching policies is increased.

6 Conclusion

In this report we have investigated the architectural aspects of SeLeNe and its relationship to Grid technology. We have proposed a set of services that form the high-level architecture of SeLeNe, reflecting its functionality. We have also discussed the possibility of combining the proposed services' functionality with existing Grid Infrastructure and more specifically the OGSA service layering guidelines. This interaction can provide the ability to other Grid projects, beyond Educational Grids, to exploit the expressiveness of semantic resource descriptions (RDF) via our generic services. We have also discussed open issues related to the SeLeNe project that are not specifically addressed by the proposed services.

The proposed services set comprises the basis for the emergence of different architectural alternatives via various service placement scenarios as a single static solution may not be sufficient for all possible cases as discussed in *deliverable 2.2. Deliverable 5* is based on the outputs of this report and also *Workpackage 4* to define the architectural alternatives and service interactions that substantiate SeLeNe's functionality.

Acknowledgments

We would like to thank Mr. C. Spyrou and Ms E. Charalambous for their assistance throughout the composition of this report.

7 References

[1] Aimilia Magkanaraki, Val Tannen, Vassilis Christophides, Dimitris Plexousakis: Viewing the Semantic Web through RVL Lenses. In Proceedings of the Second International Semantic Web Conference 2003 (ISWC 2003), 20-23 October 2003, Sundial Resort, Sanibel Island, Florida, USA

[2] Arthur M. Keller, Julie Basu: A Predicate-based Caching Scheme for Client-Server Database Architectures. VLDB Journal 5(1): pg. 35-47 (1996)

[3] Alan Fekete, Idit Keidar: A Framework for Highly Available Services Based on Group Communication. In the IEEE International Workshop on Applied Reliable Group Communication (WARGC), in conjunction with ICDCS 2001, pages 57-62, Phoenix, Arizona, April 2001.

[4] Alexander Löser, Walfgang Nejdl, Martin Wolpers, Wolf Siberski: Information Integration in Schema-Based Peer-to-Peer Networks. Conference on Advanced Information Systems Engineering (CAiSE) 2003, pages 258-272

[5] BEA, IBM, Microsoft and SAP: Web Services Policy Language (WS-Policy), 2002.

[6] Beverly Yang, Hector Garcia-Molina: Comparing Hybrid Peer-to-Peer Systems. In Proceedings of the 21st International Conference on Very Large Databases (VLDB'01), 2001

[7] Beverly Yang, Hector Garcia-Molina: Designing a Super-Peer Network. In Proc. 19th Int'l Conf. Data Engineering, Mar. 2003. IEEE Computer Society Press, Los Alamitos, CA.

[8] David De Roure, Nicholas R. Jennings, Nigel R. Shadbolt: The Semantic Grid: A Future eScience Infrastructure. National e-Science Centre.University of Edinburgh, UK. UKeS-2002-02, 12 2001.

[9] EducaNext (Universal) Project: Available at www.educanext.org

[10] Edutella Project: Available at edutella.jxta.org

[11] European Grid of Solar Observations (EGSO): Available at www.mssl.ucl.ac.uk/grid/egso

[12] Francisco Matias Cuenca-Acuna, Ritchard P. Martin, Thu D. Nguyen: Autonomous Replication for High Availability in Unstructured P2P Systems, In Proceedings of the SC Conference 2003, Phoenix, AZ, 2003.

[13] Global Grid Forum (GGF): Available at www.gridforum.org

[14] Globus Toolkit: Available at www.globus.org

[15] Globus CoG Kit: Available at www-unix.globus.org/cog/java

[16] Houda Lamehamedi, Boleslaw Szymanski, and Zujun Shentu, Ewa Deelman: Data Replication Strategies in Grid Environments. In IPDPS 2003, pg. 100

[17] Ian Foster, Carl Kesselman, Steven Tuecke: The Anatomy of the Grid Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.

[18] Ian Foster, Carl Kesselman, Gene Tsudik, Steven Tuecke: A Security Architecture for Computational Grids: Proc. 5th ACM Conference on Computer and Communications Security Conference, pg. 83-92, 1998.

[19] I. Foster, C. Kesselman, J. Nick, and S. Tuecke: The Physiology of the Grid: An open grid services architecture for distributed systems integration. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.

[20] IBM, Microsoft and VeriSign: Web Services Security Language (WS-Security), 2002

[21] IBM, Microsoft, RSA Security and VeriSign: Web Services Trust Language (WS-Trust), 2002

[22] ICS-FORTH RDFSuite: High-level Scalable Tools for the Semantic Web: Available at http://139.91.183.30:9090/RDF/index.html

[23] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In Proceedings of ACM SIGCOMM'01, San Diego, September 2001.

[24] Jena Semantic Web Toolkit: Available at http://www.hpl.hp.com/semweb/jena.htm

[25] Jim Gray, Pat Helland,Patrick O'Neil, Dennis Shasha: The Dangers of Replication and a Solution. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996, pages 173-182

[26] Karl Aberer: P-Grid: A self-organizing access structure for P2P information systems. Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS 2001), 2001.

[27] Kevin Keenoy, George Papamarkos, Alexandra Poulovassilis, Mark Levene, Don peterson, Peter Wood, George Loizou: Self e-Learning Networks - Functionality, User Requirements and Exploitation Scenarios, SeLeNe Project Deliverable 2.2, August 2003.

[28] Leonidas Galanis, Yuan Wang Shawn, R. Jeffery David J. DeWitt: Processing XML Containment Queries in a Large Peer-to-Peer System. CAiSE 2003, pages 273-288

[29] Linn, J: Generic Security Service Application Program Interface, Version 2. INTERNET RFC 2078, 1997

[30] Luciano Serafini, Fausto Giunchiglia, John Mylopoulos, Philip A. Bernstein, The Local Relational Model: Model and Proof Theory. Modeling and Using Context 4th International and Interdisciplinary Conference, CONTEXT 2003, Stanford, CA, USA, June 23-25, 2003 LNAI 2680, page 286-299, Springer Verlag, 2003.

[31] Markus Keidl, Alexander Kreutz, Alfons Kemper, DonaldKossmann: A Publish & Subscribe Architecture for Distributed Metadata Management. Proceedings of the 18th International Conference on Data Engineering IEEE Computer Society, pages 309-320, San Jose, USA, February 2002.

[32] OntoWeb: Available at www.ontoweb.org

[33] Peter Triantafillou, Chryssafi Xiruhaki, Manolis Koubarakis, Nikolaos Ntarmos: Towards High Performance Peer-to-Peer Content and Resource Sharing Systems. Proc. of CIDR, 2003.

[34] Project JXTA: Available at www.jxta.org

[35] Sesame Open Source RDF Schema-based Repository and Querying facility: Available at http://sesame.aidministrator.nl/

[36] Shaul Dar, Michael J. Franklin, Bjorn T. Jonsson, Divesh Srivatava and Michael Tan, Semantic Data Caching and Replacement, Proceedings of the VLDB Conference, 1996

[37] Simple Object Access Protocol (SOAP): Available at www.w3.org/TR/SOAP

[38] The DataGrid Project: Available at www.eu-datagrid.org

[39] The Semantic Grid Community Portal: Available at http://www.semanticgrid.org/

[40] The SWAP System available at swap.semanticweb.org/public/index.htm

[41] The Universal Description, Discovery and Integration Protocol: Available at www.uddi.org

[42] UDDI Verison 2.03 Replication Specification

[43] Von Welch, Frank Siebenlist, Ian Foster et al: Security for Grid Services, Proc. Of the 12th IEEE International Symposium on High Performance Distributed Computing (HPDC'03), 2003, page 48

[44] Web Services at W3C: Available at www.w3.org/2002/ws

[45] Wee Siong Ng, Beng Chin Ooi, Kian-Lee Tan, Aoying Zhou, PeerDB: A P2P-based System for Distributed Data Sharing. SIGMOD Conference 2003, page 659

[46] Wolfgang Nejdl, Wolf Siberski, Bernd Simon, Julien Tane: Towards a Modification Exchange Language for Distributed RDF Repositories. International Semantic Web Conference 2002, pages 236-249