LionShare Architectural Cases: Internal and External Connections

Marek Hatala, SFU, mhatala@sfu.ca
January 30, 2004
Version 0.2

Abstract:

The purpose of this document is to analyze possible connections between LionShare network and other tools and networks from the perspective of the architecture and methods for connecting those tools to the LionShare. The analysis is made from the perspective of preferred use of OKI OSID and OSID-based components to enhance interoperability. Alternative solutions are suggested for the situations not supported by the current OKI OSID.

1 Introduction

LionShare P2P client is being built on top of a Gnutella-based client called Limewire. LionShare will reuse the Gnutella P2P protocol and modify it to provide authenticated access[1] to the LionShare network. To support persistence of resources on the P2P network LionShare users have an ability to upload resources to the LionShare Peer Server that functions as another peer in the network. LionShare client will be able to discover and access resources in other connected systems and networks. Similarly, LionShare will provide access to other tools and networks to LionShare resources[2].

To facilitate the openness of the LionShare network a clear specification of the protocols and connecting middleware is needed. LionShare uses two complementary approaches to support interoperability: OKI Open System Interface Definitions (OSID) and eduSource Communication Layer (ECL). In sections below we analyze architectural cases for connecting other systems and networks with respect to required functionality and current capabilities of the above specifications. Another factor we consider is an effort required to connect these systems into the LionShare network. Based on the analysis we propose development of connecting middleware and guidelines for connecting other systems to the LionShare network.

2 Interoperability Background (OKI OSID and ECL capabilities)

OKI OSID is a set of APIs for separating client applications from the backend of the system. OSID specifies set of APIs for different functionality. Communication with other systems is via 'plugin' that is included into the application. The application communicates with plugin via OSID API and plugin communicates with other application using other system specific protocol. Different OSID plugins can call each other via their APIs. The application and plugins can exchange information by putting it into shared 'Context' object accessed via API.

ECL is an implementation of IMS DRI. ECL defines a SOAP based protocol with functionality for searching, submitting, and harvesting resources. ECL protocol is used for communication between repositories (such as federated search, harvesting) or between tools and repositories (search, submit). Subscribe/alert functionality is not implemented yet[3]. In addition to protocol definition ECL provides two middleware components. ECL connector provides fast and easy way for connecting existing systems to the ECL network. ECL gateway is designed as a framework enabling developers to define a mapping between ECL protocol and other network protocols. Currently, gateways to OAI, Edna, SMETE and SPLASH P2P network are implemented,

OKI/ECL plugin is an implementation of ECL connector that exposes the ECL Search functionality via OSID Digital repository (DR) interface. This includes definition of the OSID Types and Assets specific for the learning objects.

---

[1] This document does not deal with the specifics of the authentication neither for the Gnutella protocol nor other protocols mentioned din this document.
[2] Under certain conditions such as authentication requirements are met.
[3] As of time of writing January 22, 2004.

3 Basic LionShare Architecture

AC-3-1: Basic LionShare Architecture and LionShare Client

The basic LionShare architecture is shown in Figure 1.

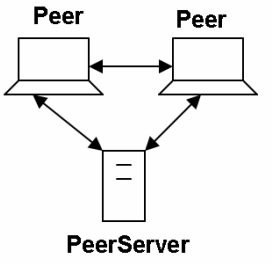

Figure 1 Basic LionShare Architecture

Peers communicate with each other and with a PeerServer using Gnutella protocol. The communication layers in the LionShare client are shown in Figure 2

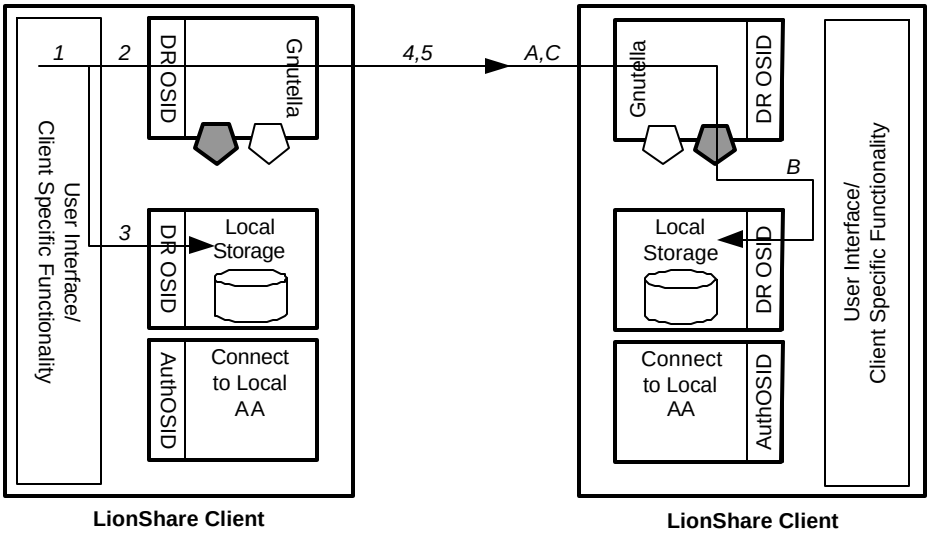

Figure 2 Communication Between Two LionShare Clients

The LionShare client utilizes OKI OSID for separating Client Specific Functionality accessible via User Interface from the underlying (reusable) infrastructure. The core of the LionShare Client consists of three OKI plugins.

3.2 LionShare Plugins

Authentication Plugin is accessible via Authentication OSID and connects to Authentication authority (AA) that can authenticate the user. Typically, that would be university or organization authority. Users in different organizations will need to include plugin for their specific AA, also

we expect several standard plugins to be developed and reused between institutions, such as Authentication plugin for Kerberos.

Local Storage Plugin is standard plugin that enables the LS Client to access local resources. (more needed here).

The OKI/Gnutella Plugin enables the client to query the LionShare (Gnutella) network via standard DR OSID interface as well as to receive Gnutella queries, convert them to DR OSID calls and direct them to the Local Storage plugin.

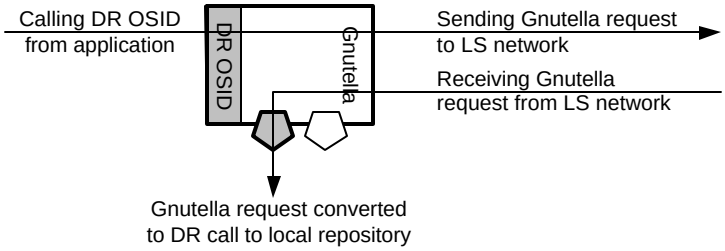

Figure 3 OKI/Gnutella Plugin

The OKI/Gnutella plugin is the most important plugin in the LionShare Client. It is a standardized plugin that can be widely distributed. It not only enables LS client to talk to other clients but it also enables to connect other OSID-enabled applications to be connected to the LionShare network.

Sending the Query

1- user formulates the query

2, 3 – the user is send to both Local Storage and OKI/Gnutella plugin (this job is done by OSID DR Manager).

4- the query is distributed via Gnutella standard protocol (NOTE: there are several possibilities how to do it in the DR OSID, but eventually Gnutella requests are sent to other LS Clients).

5- the response is received via Gnutella connection and the plugin transforms the results to assets that are available via AssetIterator in DR OSID.

Receiving and Processing the Query

A- OKI/Gnutella plugin receives gnutella query. It uses standard Gnutella mechanism to distribute query further in the network AND searhes local storage for matching resources.

B- OKI/Gnutella plugin converts search request for the local resources into standard DR OSID request fro the Local Storage Plugin.

C- Local results and results received from the Gnutella network are bundled and send back as a standard Gnutella response to the query.

4 LionShare Client Accessing Resources Outside of LionShare

AC-4-1: LionShare Clients Accessing Other Systems via OSID DR Plugin

As described above LionShare client's side implements OSID DR to send requests to other LionShare clients. Therefore, LionShare client can send request supported by OSID DR to other systems that provide their OSID DR plugins[4]. In this way the LionShare Client can send request for resources also to systems that do not use OSIDs internally. The only requirement is that DR OSID plugin exists for those systems. The situation is shown in Figure 4.

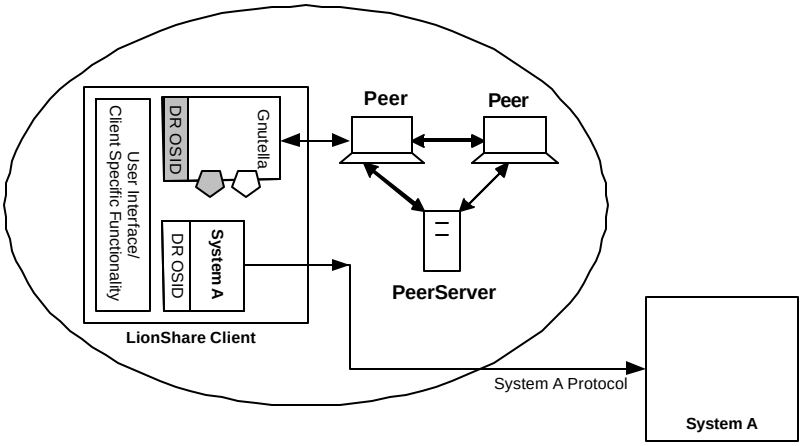

Figure 4 LionShare client sends requests to System A via specific OKI plugin

Pros:

1. If DR OSID Plugin exists leverages previous effort.

Cons:

1. DR OSID Plugin has to exist
2. For new systems DR OSID Plugin has to be installed by the user into existing Client. If code modification is needed (e.g. incompatible types/assets) then new verision of LionShare Client is needed.
3. Uni-directional access from client to the system. See AC-5-3: for set of issues with other direction.

Suitable for: Fedora, DSpace, SAKAI

[4] Note that this is also true for sending requests to other Lionshare clients via OSID DR/Gnutella plugin.

AC-4-2: LionShare Clients Directly Accessing ECL-enabled Repositories via OKI/ECL Plugin

In the previous case the LionShare client was using System A specific plugin that communicated with the System A using a specific protocol. For each new system a plugin had to be downloaded and integrated with LionShare client. To send requests to ECL-enabled system we use a generic OKI/ECL plugin. This enables LionShare Client to discover and send a request directly to any ECL-enabled system. There are 2 ways how to make other systems ECL compatible. If the system was built with OKI OSID layer the OKI/ECL plugin can be incorporated into the system In other cases ECL connector can be used to quickly connect systems that were not designed with OKI OSID specification layer. The benefit of this approach is that the LionShare client is distributed with the OKI/ECL plugin and the user can discover new ECL-enabled repositories by searching through the ECL registry.

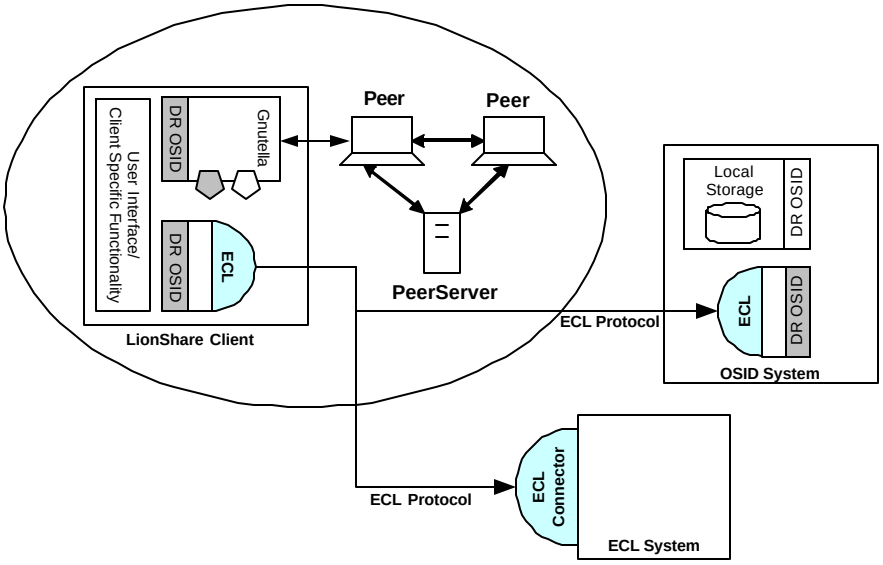

Figure 5 LionShare client sends requests to ECL enabled systems via generic OKI/ECL plugin

Pros:

1. Other systems accessible from both ECL tools AND OKI DR OSID tools (via OKI/ECL Plugin)
2. Systems can be discovered via ECL registry.
3. Possible to connect to the system not designed with OKI OSID. Use ECL Connector.
4. See AC-4-5:, AC-5-2: and AC-5-4: for complementary benefits.

Cons: none.

Other projects: Fedora, DSpace, SAKAI, eduSource network with connected systems

AC-4-3: LionShare Client Accessing DR OSID Compatible Systems via OKI/Gnutella Plugin

If another system itself implements DR OSID and wants to provide its resource to the LionShare network it can download and incorporate OKI/Gnutella Plugin into its implementation. From this point this repository can be searched as any other peer in the LionShare network. This case differs from case AC-2 in protocol used and a way how queries reach the System A. In AC-2 case queries would use System A's specific protocol and would be sent to System A directly from the querying LionShare client. In this case, the query hits System A only if it reaches System A while propagated in LionShare network. The situation is schematically captured in Figure 6.

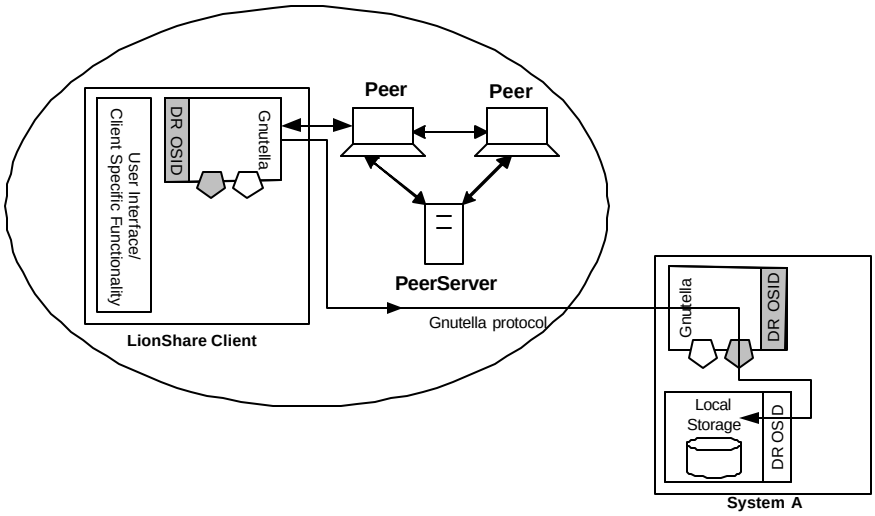

Figure 6 LionShare Client sending request to DR OSID based system incorporating OKI/Gnutella plugin

Pros:

1. Links other user systems into LionShare network as new LionShare clients.
2. New clients fully functional in the LionShare network.

Cons:

1. Cannot address request directly to the system – no guarantee the system will be reached.

Other projects: Chandler, Splash

AC-4-4: LionShare Client Accessing DR OSID Compatible Systems via OKI/ECL Plugin

4.2 EduSource Gateway

EduSource addresses the problem of outside interoperability by providing a second type of mediator simply called the eduSource Gateway. The eduSource Gateway is modeled after the design patter of an adapter functioning at the network level. The main function of the gateway is to mediate between ECL and communication protocols used by the outside systems.

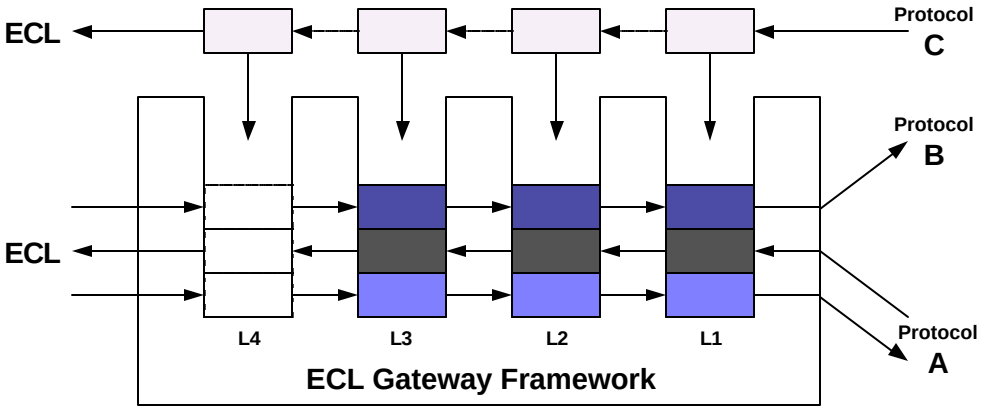

Figure 7 ECL Gateway Framework for mapping between ECL and other protocol

One side of the gateway is formed by the ECL connector. The other side of the gateway provides a framework (Figure 7) for defining a chain of handlers that perform a conversion between ECL protocol and the protocol of the external network. The gateway framework enables us to define the mapping between protocols at four levels:

L1. Communication protocol (HTTP, SOAP, XML-RPC, Peer-to-peer, etc.)
L2. Communication language (ECL, OAI, POOL, etc)
L3. Metadata (IMS, CanCore, Dublin core)
L4. Ontologies (vocabularies for metadata)

The eduSource Gateway is typically running on a dedicated computer and provides services for all participants in the eduSource network. The main benefit of placing the mapping functionality for an outside network onto a gateway instead of with each participant is that it can be easily updated if the change in the outside network protocol occurs. In such a case, a chain of mapping handlers is updated at the one place and all eduSource participants can continue to communicate with the gateway using ECL protocol without any change necessary. There can be several gateways for the same outside network if the traffic between the two networks is high. One gateway can provide services for several other networks.

The gateway also functions as a selector of internal eduSource services for the external requests. Currently, any request addressed directly to an eduSource node is forwarded to that node, while a request addressed to eduSource as a whole is distributed to all registered nodes providing the requested service. This is not the best way of distributing the requests and needs to be further addressed.

AC-4-5: LionShare Client Accesses Other Network via ECL Gateway

LionShare provides its users with an access to networks and systems that use other protocols than ECL via ECL gateway mechanism (Figure 8). This mechanism is suitable for the important well established networks and systems with well defined and stable protocol. The LionShare client communicates with the ECL gateway using ECL protocol via OKI/ECL Plugin. ECL gateway transforms the ECL request into Network A protocol request and distributes it to the Network A or sends to the specific system. The reverse process of querying LionShare network from other networks is described in case AC-5-1:.

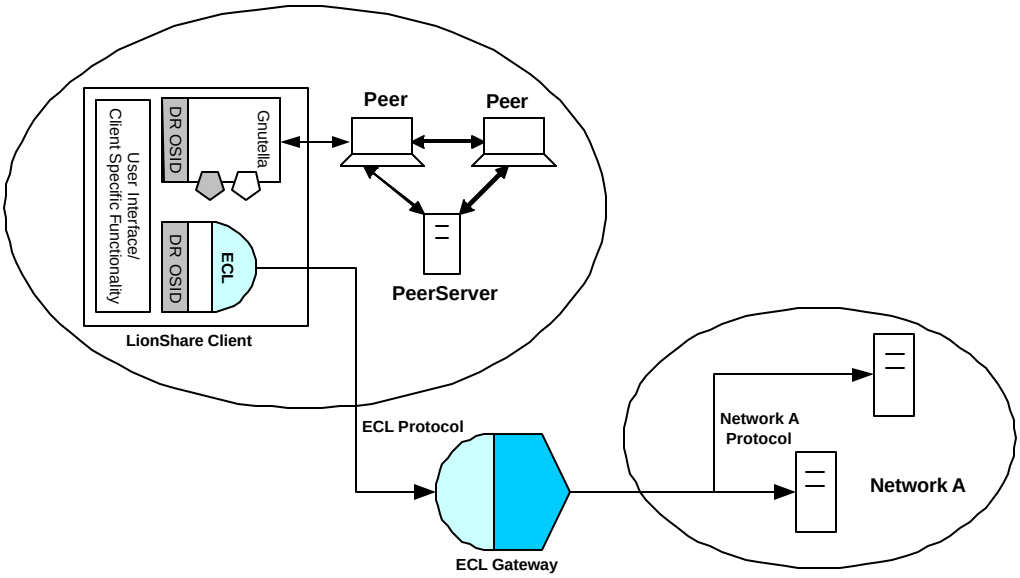

Figure 8 LionShare Client communicates with other network via OKI/ECL Plugin and ECL gateway

The mechanism allows accessing both server based and P2P networks. At the time of writing (January 29, 2004) gateways to four networks and systems were defined and operational:

- ECL ←→ OAI
- ECL ←→ EdNA
- ECL → SMETE
- ECL ←→ POOL peer-to-peer network

Pros:

1. Gateway can be discovered via ECL Registry
2. new networks can be added without affecting client
3. Changes in protocol A dealt with in one place

Cons:

1. Creates bottleneck – can be resolved by standard load balancing approach.

Other projects: DSpace, Fedora

5 Connecting to LionShare Client/Network

5.1 Connecting to LionShare Client

AC-5-1: Other Systems Accessing LionShare Client via OKI/Gnutella Plugin

This case is a complementary case to AC-4-3:. The outside system communicates with LionShare via OKI/Gnutella Plugin using Gnutella protocol[5]. The LionShare provides OSID DR/Gnutella plugin to facilitate connection of other clients implementing OKI DR OSID specification. The system queries LionShare network in exactly same way as a LionShare Client does. The system can be an end-user tool or other system, for example an aggregator or a repository providing federated search functionality that includes LionShare network. Figure 9 schematically captures this situation.

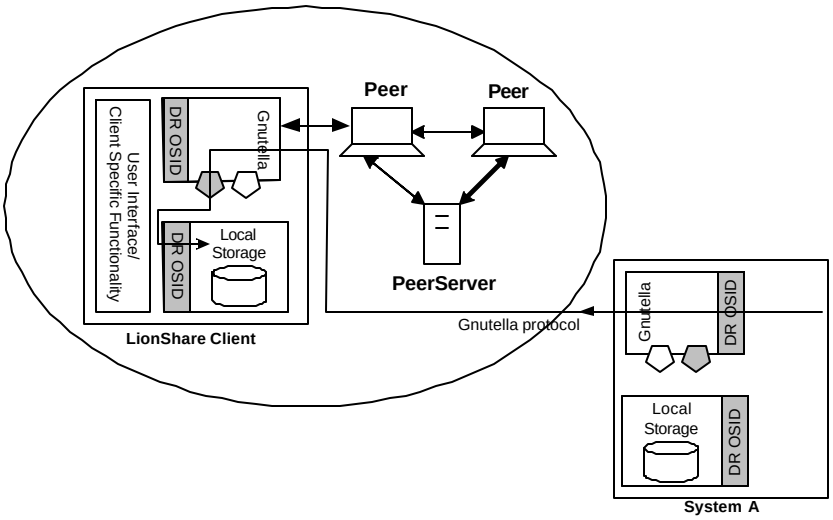

Figure 9 System A accesses to LionShare network via OKI/Gnutella plugin

Pros:

1. Reuse of effort invested in building DR OSID in other application and DR OSID Plugin
2. Communication/authentication mechanism coherent with LionShare network
3. Updates in the plugin can be synchronized at both LionShare Clients and other system
4. Natural role for end user tools.

Cons:

1. Outside system has to maintain a neighborhood of other LionShare peers. This is natural for end-user tools but not for repositories.
2. Reach of the queries from the repository-type system is limited to the neighborhood
3. Limits other system ability to search other networks connected to the LionShare network (unless OKI/ECL Plugin is used in which case AC-5-2: is preferable architecture).

Other projects: Chandler (OKI-systems with difficulties)

[5] Considering that it is properly authenticated.

AC-5-2: Other System Accessing LionShare Client via ECL Protocol

This case is similar to the case AC-5-1: and it is the reverse of AC-4-3:. Either ECL-connector OKI/ECL Plugin is used on the other system side and OKI/ECL Plugin is used in LionShare Client. ECL is used as a protocol. The pros and cons of this approach are same as in AC-5-1: except that the outside system benefits from having access to other ECL-enabled systems too.

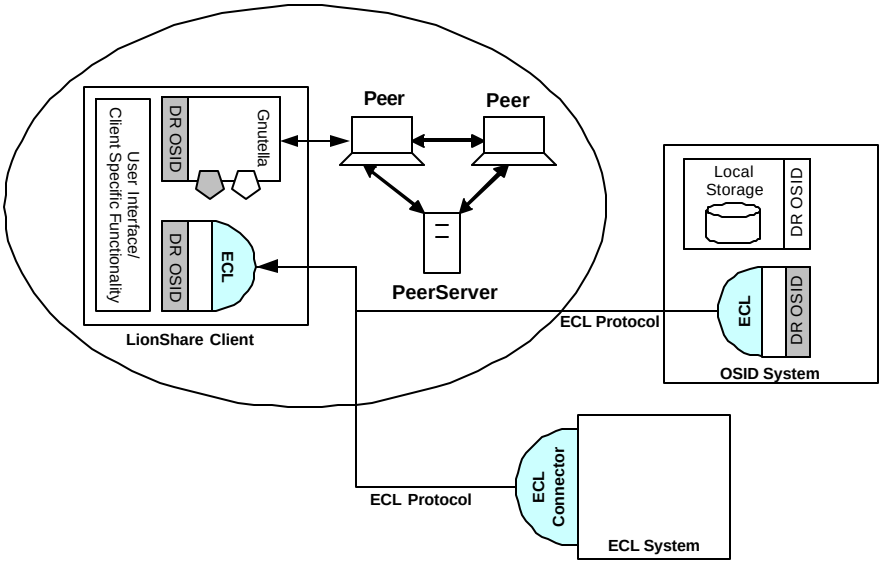

Figure 10 Other System Accessing LionShare Client via ECL Protocol

Pros:

1. Reuse of effort invested in building DR OSID in other application and DR OSID Plugin
2. Communication/authentication mechanism coherent with LionShare network
3. Access to ECL enabled system and networks.
4. Updates in the plugin can be synchronized at both LionShare Clients and other system
5. Natural for end user tools.

Cons:

1. Outside system has to maintain a neighborhood of other LionShare peers. This is natural for end-user tools but not for repositories.
2. Reach of the queries from the repository-type system is limited to the neighborhood

Other projects: Chandler, Splash (ECL-systems and OKI systems with difficulties)

AC-5-3: Other System Accessing LionShare Client via System Specific DR OSID Plugin

Specific Plugin in LionShare client.

Cons:

1. problem with plugin needing to know how exactly LionShare represents assets and types – unless direct match than plugin needs to be re-implemented.
2. problem with keeping clients neighborhood.

5.2 Connecting to LionShare Network

The nature of the peer-to-peer protocol such as Gnutella poses some challenges how the responses to the request are handled. First, peers in peer-to-peer network maintain knowledge of their network neighborhood of other peers they directly communicate with. A system outside of the peer to peer network need to first get knowledge of a peer or peers to send a query to. Once that is accomplished, the query is send to those peers and is distributed in the peer-to-peer network. Second problem is caused by query results coming back asynchronously in batches and they need to be combined into one result set before they are returned to an issuing client.

The most efficient way to solve both problems is to build a gateway between LionShare and other network. The gateway would be responsible for 2 tasks:

1. Keep connections with several peers in the LionShare network. A good candidates would be PeerServers that would distribute query into specific parts of the LionShare network, for example PeerServer at PennState would maintain a neighborhood of other LionShare peers and PeerServers at PennState.
2. Translate between Gnutella and another protocol.

AC-5-4: ECL System Accesses LionShare Network via ECL/LionShare Gateway

ECL/LionShare Gateway is a gateway that is designed to translate between ECL protocol and Gnutella protocol used in the LionShare network. The gateway also maintains connections to LionShare peers and reconciles differences in the nature of two protocols with respect to synchronization.

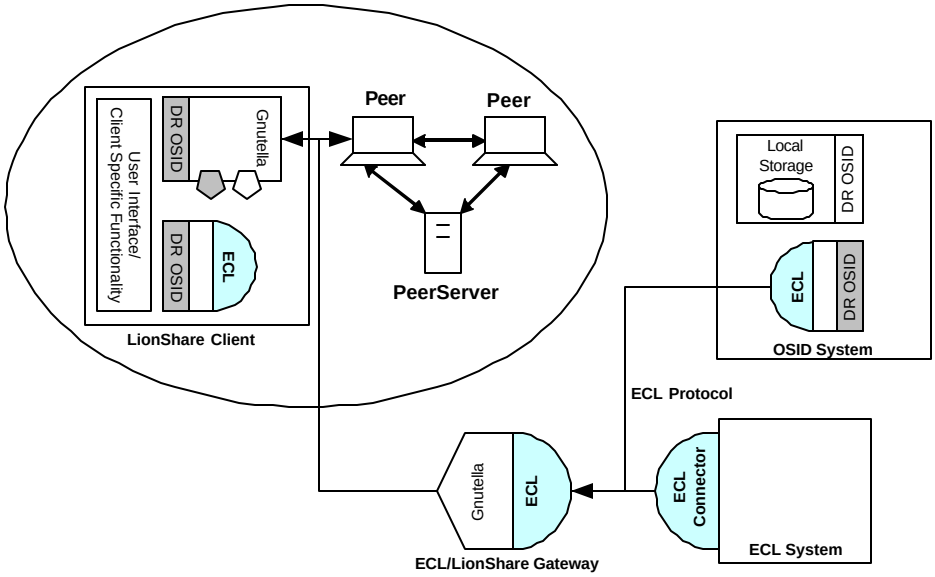

Figure 11 ECL/LionShare Gateway

Note 1: Although LionShare client can discover and access ECL-enabled system selectively (case AC-4-2:) this is not true in other direction. The ECL-enabled system views LionShare network as a whole and need to send request to LionShare network via ECL/LionShare gateway.

Note 2: As we assume LionShare network will eventually consists from several networks (for example campus-based) it can be possible to setup a gateway for each of these (sub-)networks individually.

Note 3: From the perspective of the ECL client the ECL/LionShare gateway looks and function like other nodes on the network. Specifically, it registers with the ECL registry and can be discovered via standard ECL search mechanism.

Note 4: Considering previous note and case AC-4-2: LionShare client can direct requests to other LionShare networks by using its ECL capability. It can discover other LionShare networks via ECL registry and utilize OKI/ECL plugin to send request to the ECL/LionShare gateways for discovered networks.

Pros:

1. Reuse of effort invested in building DR OSID in other application and DR OSID Plugin
2. Communication/authentication mechanism coherent with LionShare network

3. Access to ECL enabled system and networks.
4. Updates in the plugin can be synchronized at both LionShare Clients and other system
5. Natural for end user tools.

Cons:

1. Outside system has to maintain a neighborhood of other LionShare peers. This is natural for end-user tools but not for repositories.
2. Reach of the queries from the repository-type system is limited to the neighborhood

Other projects: Chandler, Splash (ECL-systems and OKI systems with difficulties)

AC-5-5: Non-ECL System Accesses LionShare Network

6 LionShare Network Architecture

Based on the cases analyzed in the previous sections the architecture connecting LionShare Clients and other systems is shown in Figure 12. The following criteria were used to define the architecture: maximizing utility/access to resources, supporting extensibility of the architecture, utilizing existing infrastructure, supporting discoverability of resources, minimizing implementation efforts via middleware components.

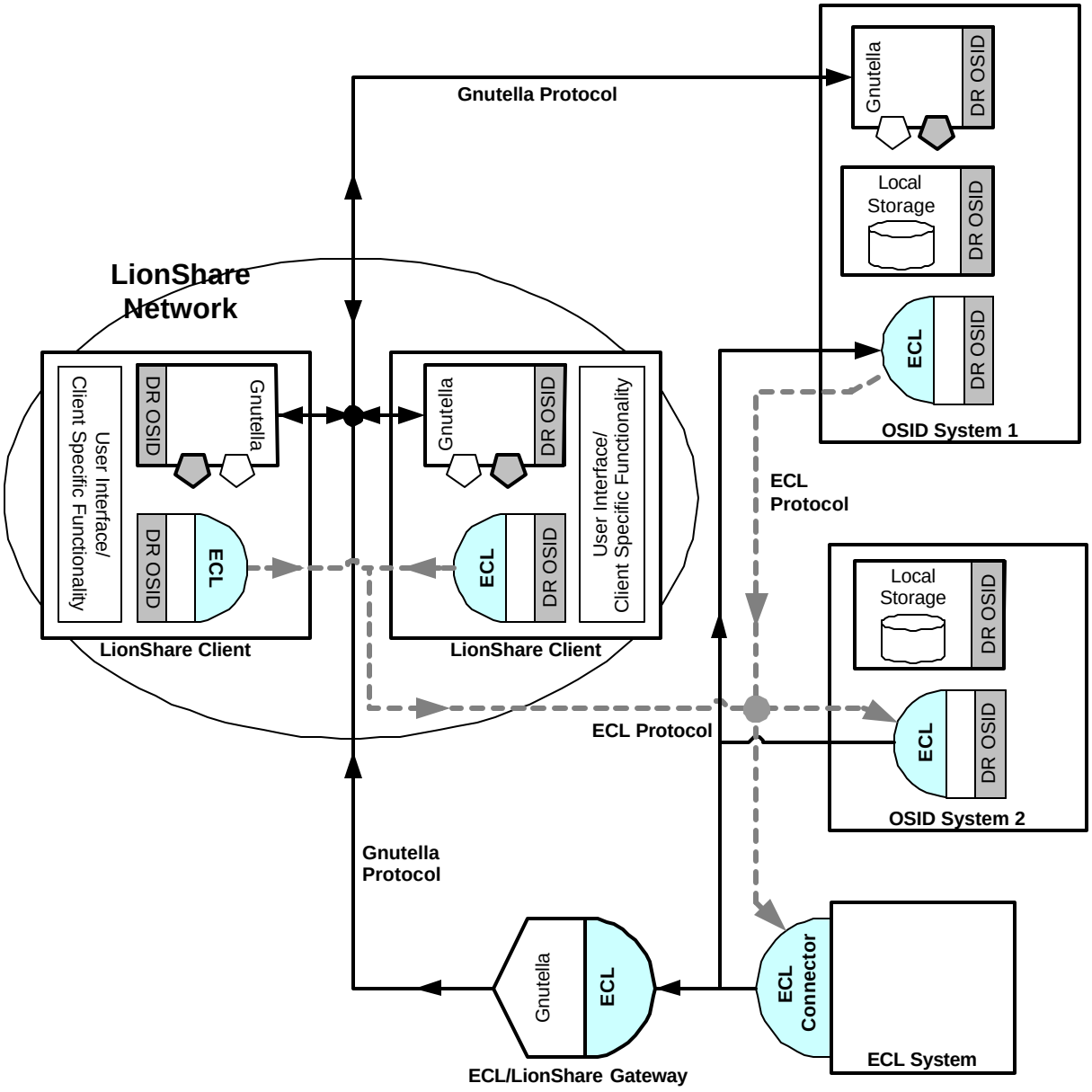

Figure 12 LionShare Network Architecture

6.1 Connected Systems

The systems connected into the architecture differ based on their function. The distinction is based to optimize the communication and support their functionality that is different for each system.

LionShare Client
LionShare Client is a client for individual user that participates in a peer-to-peer network. LionShare Client has its own repository of resources the user can share with others. The client's main functionality includes search and management of resources, authoring component for combining resources and creating e-portfolio views.

OSID System 1
This is first of two OSID type systems. This type represents a user specific client similar to LionShare Client. It is a type of tool that represents an individual user and his/her resources.
Examples: *Chandler, VUE, e-Portfolio Client, or SPLASH* (when re-implemented with OSID)

OSID System 2
Second type of OSID system is a system shared with many users, such as LMS or repository. Typically, users access the system via web interface but in general system does not 'represent' any of them as individuals. This type of a system typically holds large volume of resources (such as repository) or provides authenticated access to a specific collection of resources for selected group of users (such as courses on LMS).
Examples: *Fedora, SAKAI, e-Portfolio server, e-Portfolio Client, VUE, DSpace* (if rebuilt with OSID).

ECL System
This category includes both ECL client tools and ECL repositories. Existing legacy systems can be converted into the ECL-enabled system by implementing ECL connector. This category also includes other systems and networks that are connected to the ECL network via specific gateways.
Example repositories: *POND, CAREO, Explora, SMETE, EdNA, DSpace* (if connector implemented)
Example tools: *SPLASH, ALOHA, Explora tools, Splash Federated Search*

6.2 Communication Flows

LionShare architecture combines two protocols and several middleware components for maximum interoperability with other systems and networks. The communication flow chosen is based on the particular type of the system. Table below summarizes protocols used in communication:

From \ To	**LS Client**	**OSID 1**	**OSID 2**	**ECL**
LS Client	Gnutella	Gnutella	ECL	ECL
OSID 1	Gnutella	Gnutella	ECL	ECL
OSID 2	ECL/Gnutella	ECL/Gnutella	ECL	ECL
ECL	ECL/Gnutella	ECL/Gnutella	ECL	ECL