Development of a file-sharing system for educational collaboration among higher-education institutions

Takuya Matsuhira, Yoshiya Kasahara, and Yoshihiro Takata

***Abstract*—**Opportunities for educational, research-oriented, and business-related collaborations among universities are increasing. A file-sharing system that handles reports, documents, and research papers is required to facilitate collaboration. However, file sharing across different universities under a distributed environment is not easy because of the difficulty of authentication and authorization. In this context, we have developed the ARchive system for cross-referencing across a distributed environment (ARCADE), using which users can share files securely with other users across organizational boundaries. Note that even those who lack skills in ICT can use the system, as ARCADE is implemented as a Java GUI application. In this paper, we introduce the configuration of our system and discuss its adaptation to an educational setting.

***Keywords*—**attribute sharing, shibboleth, single sign-on, federation

I. Introduction

Opportunities for collaboration among universities have recently increased in many areas such as education, research, and business. In particular, in Japan, organizations called consortium of universities have been established to facilitate collaboration among higher-education institutions such as universities, junior colleges, and technical colleges by educational exchange, information sharing, research, and investigation. Each prefecture has established a consortium. Kanazawa University has joined the University Consortium of Ishikawa, which consists of approximately 20 higher-education organizations in Ishikawa Prefecture. A typical collaborative activity is transferring credits for students. Students can take courses at other universities using e-learning and/or distance learning. In such situations, a file-sharing system to facilitate collaboration is required. File types include reports, documents, and research papers. However, file sharing across different universities under a distributed environment is not easy. One reason is the difficulty of authenticating users belonging to different universities. Another reason is authorization. It is not sufficient to define data access policies for files simply as "fully open access" or "subscribers only." Various data access policies are possible depending on the file.

In this context, we have developed the ARchive system for cross-referencing across a distributed environment (ARCADE), a data management system that allows data owners to share data easily with appropriate users across organizational boundaries. This system uses the open-source Shibboleth software package, which is based on the SAML 2.0 protocol for single sign-on across or within organizational boundaries. This software can be used to share, upload, and download data files among closed communities. Some studies have examined Shibboleth [1],[2],[3],[4], and systems similar to ARCADE have also been developed [5]. However, ARCADE is implemented in Java and has a simple GUI. Therefore, users can interact with the system simply by using drag-and-drop actions. Moreover, users can easily control access policies on the basis of attribute-based access control (ABAC) [6],[7],[8],[9],[10].

This system is expected to be highly useful; thus, we attempted to apply it to the University Consortium of Ishikawa. We adapted ARCADE to share educational materials and demonstrated its use to share student reports and teaching materials.

In this paper, we first discuss the problems that must be addressed. Second, we explain how ARCADE works. Third, we present an example of how ARCADE can be adapted to an educational environment. Finally, we present our conclusion and describe our future work.

II. Problems to be Addressed

In this section, we discuss important challenges in designing ARCADE properly. The problems include access control management and usability.

A. Access Control Management

In this section, we discuss the management of access control when collaborating. The management model of collaboration is presented schematically in Fig. 1. For example, consider a faculty member at A University who manages his/her course material. If this faculty member collaborates with a faculty member at C University, he/she can also use the educational materials for courses offered by the faculty member at C University. Moreover, a student at University A can access courses offered by the faculty at both Universities A and B. However, faculty members can exchange their educational materials, but students can only read them. Moreover, the faculty member at University A may collaborate with the

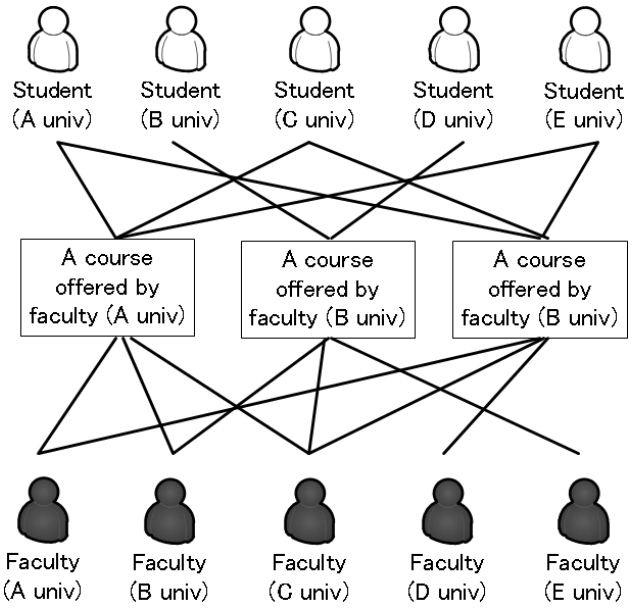

Fig. 1 Concept of management model of collaboration

faculty member at University B, but reports submitted by students who attend the courses at University A can be accessed only by the faculty members of that University. Thus, ARCADE must handle access control properly. We employ user attributes that determine user access to materials. Therefore, when user attributes must be changed, we need to change them rapidly. To realize this, the attributes of students and faculty are managed by the University with which they are affiliated.

B. *Usability of ARCADE*

We must solve the problems explained in section A, but the solution should not produce a complex system. In particular, the system must not rely on users' IT skills. Hence, we consider this an important goal of any implementation.

C. *Development Policy*

To manage users, we constructed a system using the lightweight directory access protocol (LDAP) technology. We used the OpenLDAP software [11], which is an open-source software. To manage access control among organizations, we used the open-source software Shibboleth. Shibboleth interacts with LDAP to manage users and control access by employing user attributes. Moreover, to achieve usability, we developed an interface as a Java application so that users can open, download, and upload files easily.

III. Development of ARCADE

A. *About Shibboleth*

We used the Shibboleth [12] architecture as the ARCADE framework. Shibboleth is a project of the Middleware Architecture Committee for Education of Internet2 [13]. It is based on SAML2.0 [14]; hence, we can implement single sign-on and attribute sharing among different information systems. Shibboleth consists of three systems. The identity provider (IdP) authenticates users and sends their attributes to the service provider (SP). The SP requests users' authentication from the IdP and sends user attributes to applications running on the SP server. The discovery service (DS) provides information to ensure that users choose the correct IdP when multiple IdPs are available.

The operation of Shibboleth is presented schematically in Fig. 2. First, a user at University A accesses University B's SP. Second, the SP redirects to the DS so that the user chooses the IdP of the user's organization. Third, the user selects the proper IdP and is authenticated by his or her ID/Password. Fourth, the IdP returns the authentication result; when the result is true, the SP requests the attributes that it needs to authorize the user. Fifth, the SP offers the service requested by the user.

B. *ARCADE Specifications*

This section presents an overview of the ARCADE specifications. Fig. 3 shows a conceptual image of ARCADE in operation. The following explanations assume a consortium between three organizations: Kanazawa University, Sample University, and Test University.

1) *User Management*

For user management, as explained in the preceding section, we constructed a directory using LDAP so that users can be managed at each organization and by common user information shared among organizations. Table 1 shows the designed LDAP schema. Users have the following attributes: Last Name, First Name, Login ID, Password, Organization Name, Affiliation, Position, Email Address, and Remarks. We use the attribute types of the LDAP standard for wide compatibility.

An LDAP directory contains the authentication and attribute information used by Shibboleth's IdP, as explained above. Because of the interaction with Shibboleth, we need not construct an authentication mechanism for each information

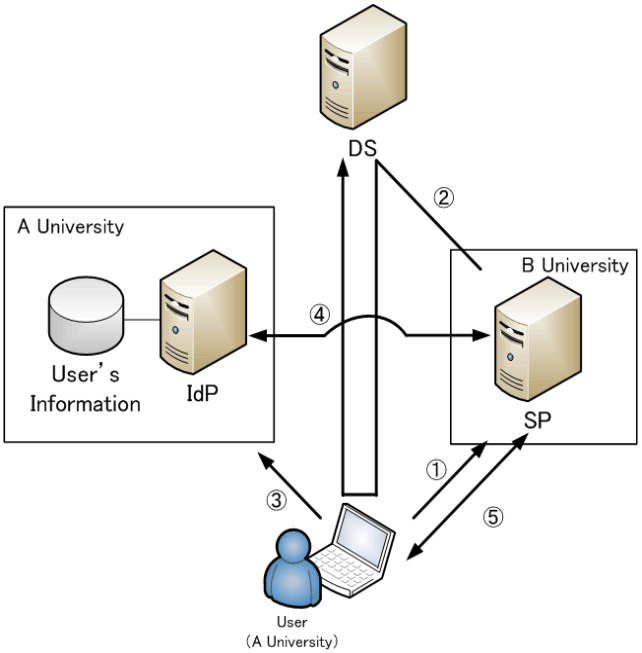

Fig. 2 Conceptual image of Shibboleth

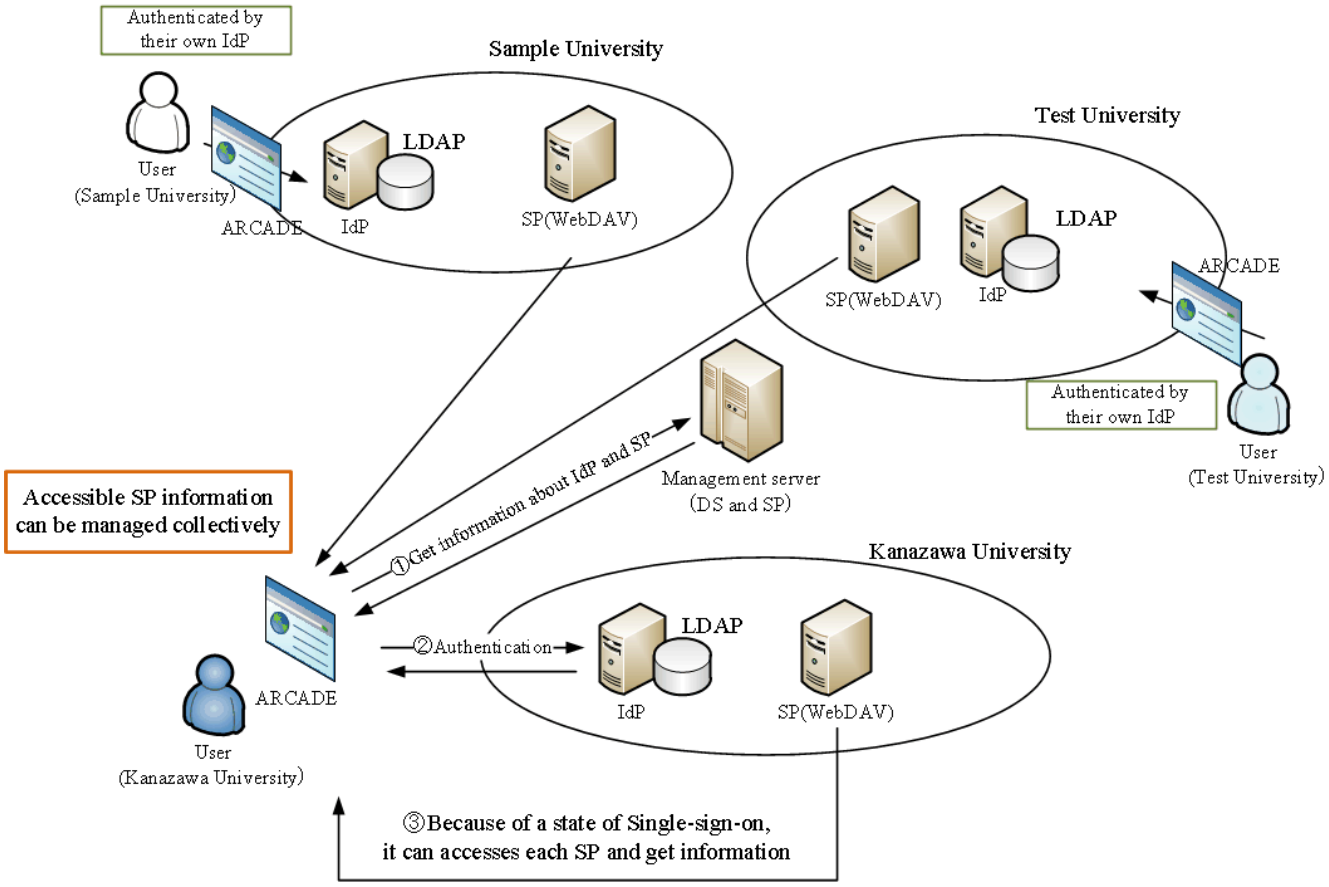

Fig. 3 Conceptual image of ARCADE

system, and we can manage only our own organization's users.

2) Access Control Management

As described in the preceding section, we applied the Shibboleth environment to control public access easily and correctly. As a Web service, Shibboleth can control access by using information in the LDAP directory. Because Shibboleth and Apache are highly compatible, we can easily restrict access in each directory by using Apache's .httaccess file. Fig. 4 shows an example of the settings. Here, "uid," which appears as "Login ID" in Table 1 as a key, gives the reference authority of the current directory to the user "ishikawa" and organization "Kanazawa University." Thus, it becomes possible to restrict access easily according to user attributes by using Apache's access control method.

Educational materials are stored in each organization's SP. We use Web-based distributed authoring and versioning (WebDAV) as the mechanism by which users upload and download their educational materials. WebDAV is a protocol that allows file sharing among machines running different OSes using Web access technology. In summary, files and directories on the Web server can be managed by the client. As WebDAV can use the protocol "https," we can use only port 443 even if Shibboleth and WebDAV are combined. Because this corresponds to the standard for WebDAV server functions in Apache, easy access control is built in, and the SP was constructed using Apache in the present study.

3) Usability of ARCADE

Strong usability is essential because ARCADE users may have only basic IT skills. In particular, we have attempted to

```
AuthType shibboleth
ShibRequireSession On
require uid ishikawa
require o KanazawaUniversity
```

Fig. 4 Example of authentication

Table 1 Attributes in LDAP

Attributes Type	LDAP	Example Value
Last Name	sn	Ishikawa
First Name	givenName	Taro
Login ID	uid	ishikawa
Password	userPassword	ishikawasan
Organization Name	o	KanazawaUniversity
Affiliation	ou	IshikawaLab
Position	employeeType	Faculty
Mail address	mail	ishikawa@kanazawa-u.ac.jp
Remark	description	administrator

hide from users the existence of servers such as the IdP, SP, DS, and LDAP directory. Thus, we developed ARCADE as a Java application.

We implemented ARCADE by using the standard widget toolkit (SWT) [15], which is constructed by Eclipse [16]. However, SWT does not depend on Eclipse, so we can use SWT as a single GUI toolkit library. One of its best features is that it uses the native OS's widgets such as buttons and textboxes. As a result, ARCADE runs rapidly and its appearance resembles that of the OS. Furthermore, ARCADE is launched by Java Web Start [17]. Thus, users can start ARCADE via a Web browser. The advantages of Java Web Start are as follows.

- Users simply click a link to start ARCADE.
- Users can use the latest version.
- Users do not need to install or update software.

These reasons make ARCADE easy to use, even for those with low levels of IT skills.

C. Using ARCADE

1) Authentication

The server that runs the ARCADE application is defined as the management server. The management server also plays the roles of SP and DS.

When a user accesses ARCADE via a Web browser, he/she is redirected to the DS to authenticate to his or her organization's IdP. The user selects the correct IdP (Fig. 5) and then enters an ID and password (Fig. 6).

After the user is authenticated, ARCADE can retrieve the SP's information.

2) Access Control Method

We next discuss access control in ARCADE after users successfully authenticate. An overview of data management in ARCADE is shown in Fig. 7. Data management consists of three parts: display of directories, display of files, and setting of access control.

a) Display of directories

All SPs accessible to a user are displayed in the same way that the OS displays disk drives. The user can view the directory tree if he/she has appropriate permission. The user can create or delete a directory simply by right clicking on a directory, and subtrees can also be created.

b) Display of files

When the user selects a directory, ARCADE displays information about the files in the directory. The user can drag and drop files if he/she has the appropriate permission.

c) Setting of access control

The user can set three types of restrictions on each directory. All settings are controlled by the user attributes. These attributes include the user's email address, organization name, and user ID shown in Table 1.

Directory Access controls whether to display the directory in the display of directories. This can be set for each directory.

File Put Access controls whether the user is permitted to upload files in the directory. The method of restriction uses the attribute value in the LDAP directory as well as the Directory Access setting. However, as it is difficult to control access restrictions on uploading and downloading files in Apache, we implemented an access control function similar to the method used for Directory Access. Depending on the user's attribute values, he/she first accesses the management server. Then, after the authentication with the IdP of each organization is complete, the user's attribute values are sent from the IdP to the management server, which acts as the SP. ARCADE receives the attribute values, and file uploading and downloading are restricted on the basis of this information.

File Get Access controls whether the user is permitted to download files from the directory. These control settings can also be set for each directory.

Only users to whom these three activities have been permitted can change the restriction settings. This prevents operational errors in the access control settings.

IV. Adapting ARCADE for an Educational Setting

This section presents a demonstration of the proposed system in an educational setting.

A. Example of using ARCADE

In this section, we outline a proof of concept.

1) Test Conditions

Fig. 8 shows a conceptual image of the test environment. The test was carried out as follows.

- A consortium named University Consortium of Japan exists.
- This consortium consists of three organizations: Kanazawa University, Sample University, and Test University.

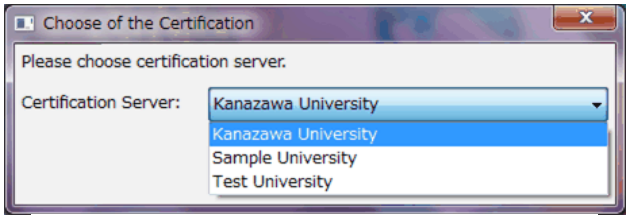

Fig. 5 Snapshot of DS page (IdP selection)

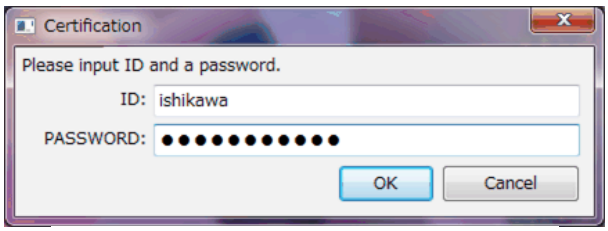

Fig. 6 Snapshot of authentication page

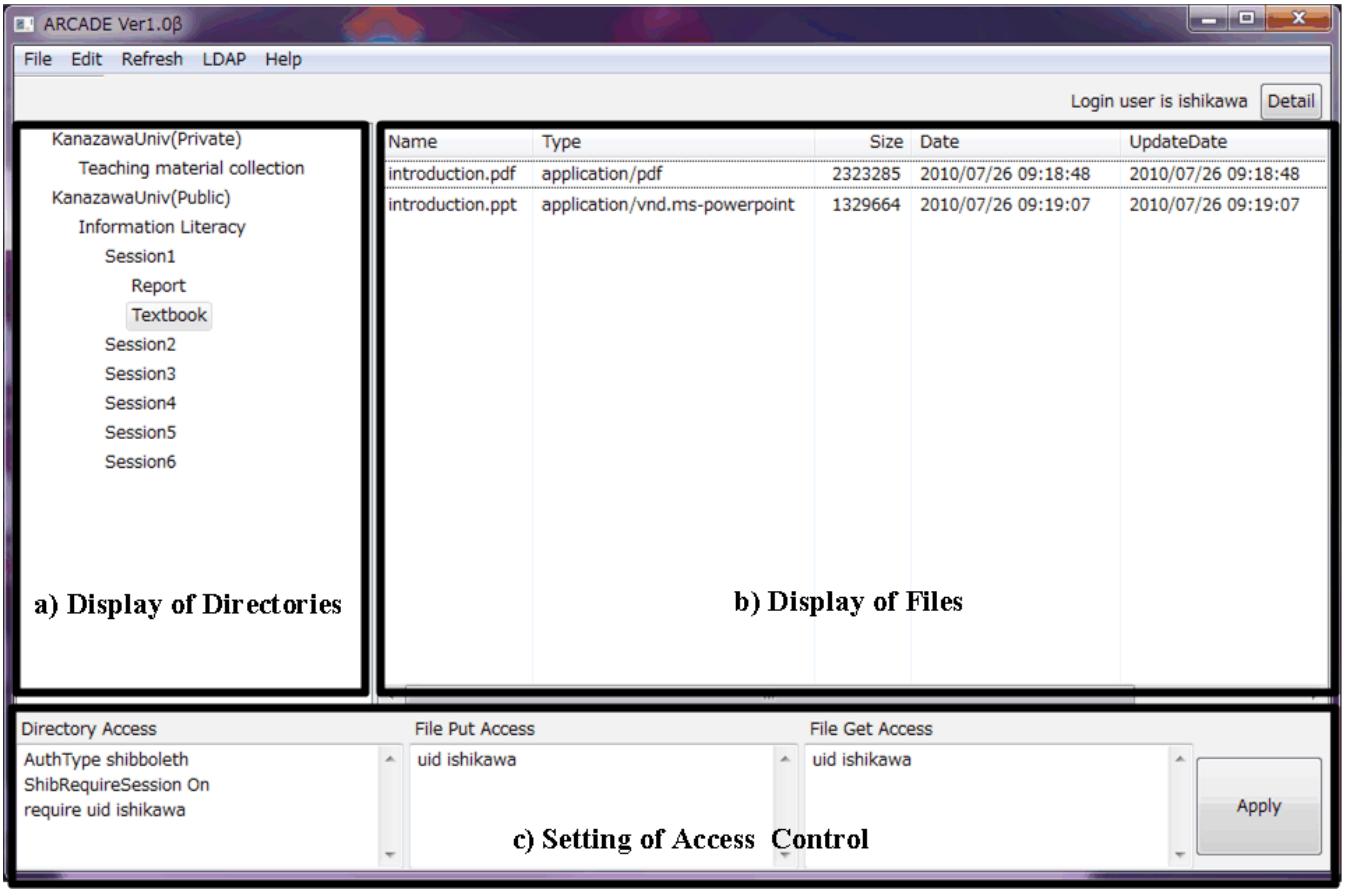

Fig. 7 Snapshot of ARCADE page

- Kanazawa University is responsible for this consortium; hence, the management server is located there.
- Kanazawa University's user, whose ID is ishikawa, offers a course called Information Literacy.
- Test University's user, whose ID is test01, collaborates with ishikawa for this lecture.
- Sample University's user, whose ID is sample01, takes this course.

2) Flow before the course begins

Each university has two types of SP: data management servers and lecture information servers. Data management servers handle educational materials via ARCADE, while lecture information servers handle lecture information via a Web browser.

First, user ishikawa accesses the lecture information server via a Web browser to register his course. When ishikawa accesses the SP, he is redirected to the DS and chooses Kanazawa University's IdP. After authentication, the attributes of ishikawa are sent to the lecture information server's SP. Kanazawa University's SP must meet the following conditions.

- The Organization Name attribute is "Kanazawa University."
- The Position attribute is "Faculty."

When these conditions are not satisfied, access is denied. Ishikawa registers his lecture. He then needs to register the course's title and outline. In this example, the course is titled "Information Literacy." When the registration is complete, this information is sent to the data management server's SP, on which a new directory is created automatically. The name of the directory is the new course name registered by ishikawa. Therefore, in this case, the directory name is "Information Literacy." Only this user can access this directory. The condition is that the email address attribute is his/her email address, ishikawa@kanazawa-u.ac.jp. The Directory Access, File Put Access, and File Get Access settings are set to his email address. When ishikawa accesses the data management server's SP via ARCADE, he can see the directory "Information Literacy" and create or delete subdirectories easily.

Second, user test01, who is at Test University, accesses the lecture information server's SP at Kanazawa University via a Web browser to collaborate for the lecture. She is also redirected to the DS and chooses the IdP of Test University. After the authentication, the attributes of test01 are sent to the lecture information server's SP. When the Organization Name attribute is not "Kanazawa University" but the Position attribute is "Faculty," the user can register for lectures provided by Kanazawa University as a collaborative member. In this case, test01 registers for "Information Literacy." The attributes

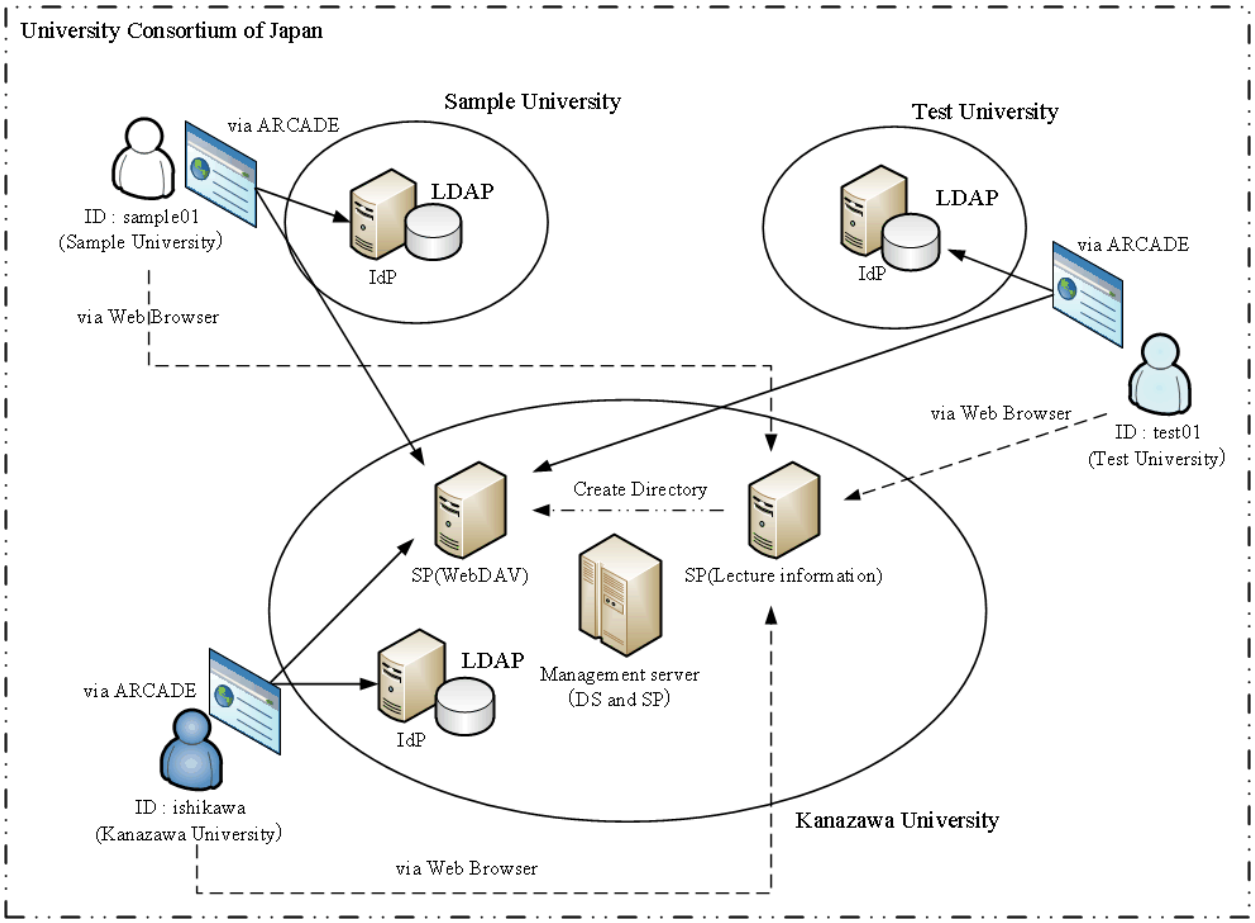

Fig. 8 Conceptual image of test environment

of test01 are recorded to determine whether he/she is the person in question, and access control is applied at the data management server's SP.

Third, user sample01, who is a student at Sample University, accesses the SP of the lecture information server at Kanazawa University via a Web browser to register for the course. He is also redirected to the DS and chooses the IdP of Sample University. After the authentication, the attributes of sample01 are sent to the lecture information server's SP. When the Position attribute is "Student," the user can register for lectures provided by Kanazawa University as a participant. Hence, in this case, sample01 registers for "Information Literacy." The attributes of sample01 are recorded to determine whether he is the person in question, and access control is applied at the data management server's SP.

Fourth, after the enrollment period ends, ishikawa accesses the lecture information server's SP to check the collaborative and participant members. After ishikawa checks the registered content, the Email Address attribute is sent to the data management server's SP. As test01 is a collaborative member, the Directory Access, File Put Access, and File Get Access settings are set for his/her mail address. As sample01 is a participant member, only Directory Access is set for his mail address.

3) Flow after the course begins

It is no longer necessary for all users to access the lecture information server's SP. Ishikawa creates directories for each session as subdirectories of Information Literacy and sets the access control for each directory. Each session has the subdirectories Textbook and Report. The former contains teaching materials and the latter acts as a collection box into which participants upload reports.

B. Results of using ARCADE

To the conditions given above, we add the following.

- Session 3 of this course is complete, so sample01 can download the Textbook material and upload reports through session 3.
- Test01 can download and upload Textbook material and access ishikawa's private teaching materials.
- Test01 cannot access each session's reports.

The directories displayed to each user are shown in Fig. 9. User ishikawa can access all SPs and directories and can set all permissions. User test01 can access "Teaching material collection" and the Textbook directories for all sessions. User test01 can only download "Teaching material collection" and both download and upload Textbook files. Student reports

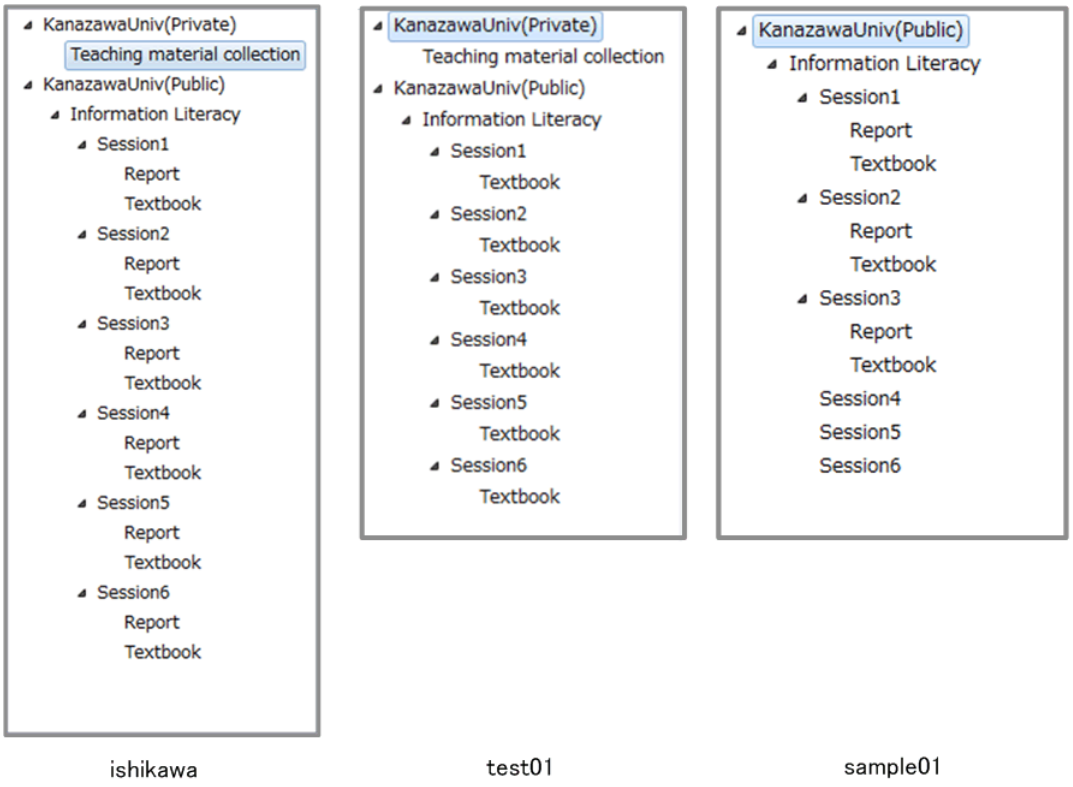

Fig. 9 Available directories according to user access control policies

cannot be accessed by test01. Kanazawa University's private SP is not displayed to sample01, who can download the Textbook files and upload reports through session 3. Neither test01 nor sample01 can change the access control for each directory.

V. Conclusion

In this study, we developed the ARCADE system, using which users can share files easily and safely across organizational boundaries. ARCADE is implemented as a Java GUI application; thus, users with only basic IT skills can also use the system to share and access files.

To confirm the utility of ARCADE in higher education, we have tried to adapt ARCADE for an educational setting. The proof of the concept was successful on the actual hardware, and therefore, ARCADE can be used in many situations associated with education. In this demonstration, during enrollment, users need to access the lecture information server's SP via a Web browser. Thus, we need to improve ARCADE so that it does not need a Web browser.

One of the problems that should be considered while developing ARCADE in the future is the need to create multiple servers. The management server in particular is responsible for the key roles of ARCADE distribution, DS, and SP. Therefore, two or more management servers must be prepared; this can be a time consuming process.

Looking toward the future, we plan to adapt ARCADE for an actual educational situation and extend its capabilities for further practical use. Work at the National Institute of Informatics [18] is crucial and includes the development of Gakunin [19], a tool for maintaining an authentication infrastructure constructed using Shibboleth. Because ARCADE can also be used with Gakunin, its practical use in the future can be expected.

Acknowledgment

This study was supported by a Grant-in-Aid for Young Scientists (B) from the Japan Society for the Promotion of Science (22700809).

References

[1] S. Encheva and S. Tumin, "Enterprise Logon Server for Domain Wide Web-Based Applications," *Proceedings of the 7th WSEAS Int. Conf. on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL and SIGNAL PROCESSING (CSECS'08)*, 2008, pp. 36–39.

[2] S. Encheva and S. Tumin, "Authentication and Authorization User Management within a Collaborative Community," *Proceedings of the 11th WSEAS Int. Conf. on COMPUTERS*, 2007, pp. 565–570.

[3] S. Encheva and S. Tumin, "Decentralized Administration in Collaborating Organizations," *Proceedings of the 6th WSEAS Int. Conf. on E-ACTIVITIES (E-Learning, E-Communities, E-Commerce, E-Management, E-Marketing, E-Governance, Tele-Working) (E-ACTIVITIES '07)*, 2007, pp. 353–355.

[4] J. Rivington, R. Kent, A. Aggarwal, and P. Preney, "A Service Oriented Architecture for Authorization of Unknown Entities in a Grid Environment," *Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION*, 2005, pp. 13–18.
[5] L. Ngo and A. Apon, "Using Shibboleth for Authorization and Authentication to the Subversion Version Control Repository System," *International Conference on Information Technology (ITNG'07),* 2007, pp. 760–765.
[6] E. Yuan and J. Tong, "Attributed Based Access Control (ABAC) for Web Services," *IEEE International Conference on Web Services (ICWS'05)*, 2005, pp. 561–569.
[7] C. Shang, Z. Yang, Q. Liu, and C. Zhao, "A Context Based Dynamic Access Control Model for Web Service," *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008, pp. 339–343.
[8] R. Wonohoesodo and Z. Tari, "A Role based Access Control for Web Services," *Services Computing, 2004 IEEE International Conference on (SCC'04)*, 2004, pp. 49–56.
[9] H.-b Shen and F. Hong, "An Attribute-Based Access Control Model for Web Services," *Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*, 2006, pp. 74–79.
[10] L. Su, "User Behavior Based Access Control Decision," *2010 International Conference on E-Business and E-Government*, 2010, pp. 1312–1376.
[11] OpenLDAP, http://www.openldap.org/ (accessed 2010.10)
[12] Shibboleth, http://shibboleth.internet2.edu/ (accessed 2010.10)
[13] MACE, http://middleware.internet2.edu/MACE/ (accessed 2010.10)
[14] SAML2.0, http://www.oasis-open.org/specs/index.php (accessed 2010.10)
[15] SWT, http://www.eclipse.org/swt/ (accessed 2010.10)
[16] Eclipse, http://www.eclipse.org/ (accessed 2010.10)
[17] Java Web Start, http://java.sun.com/javase/technologies/desktop/javawebstart/index.jsp (accessed 2010.10)
[18] National Institute of Informatics, http://www.nii.ac.jp/en/ (accessed 2010.10)
[19] GakuNin Project, https://upki-portal.nii.ac.jp/docs/fed (accessed 2010.10)