

# Controlling Access to Data in a Web-based Campus GIS

Nana Y. Dei, Hartwig H. Hochmair, Jim Q. Chen

St. Cloud State University  
720 Fourth Avenue South  
St. Cloud, MN 56301, USA  
[dena0501@stcloudstate.edu](mailto:dena0501@stcloudstate.edu)  
[hhhochmair@stcloudstate.edu](mailto:hhochmair@stcloudstate.edu)  
[jchen@stcloudstate.edu](mailto:jchen@stcloudstate.edu)

**Abstract.** In GIS Web applications on an intranet, sharing data among multiple users is a critical component for an effective working environment. However, little attention is given to securing the data from unauthorized access or restricting sensitive information on a need-to-see basis.

This research investigates methods in an ArcSDE environment to achieve data security on a Web-based Campus GIS to provide reading and editing capabilities on shared data for various user groups. A subset of data from a campus facilities data model, incorporating mainly building floor plans and master plan data was used for testing.

The database back-end and the application front-end were examined with ArcGIS Server 9.2, ArcSDE 9.2, Microsoft SQL Server 2000 and Visual Basic .NET, using various database security methods and fine-grained ArcObjects. Implementation problems and possible approaches for solutions are discussed.

## 1. Introduction

Over the past couple of decades, the technology of Information Systems has grown in every facet of life. The need for information in a timely manner has necessitated the growth in this industry.

Geographic Information Systems, which is a branch of the larger picture of IS, continues to play an important role in the delivery and analysis of spatial information for reconnaissance or decision-making purposes. With this growth in technology, the sharing of geospatial data among personnel and the publishing of maps containing these datasets on the intranet/internet is now a commonplace. In today's multi-user GIS, with the growing use of such spatial data in a variety of GIS applications and the confidentiality of some of these data, there is the need to incorporate security into the system in a way that users get access to only the requisite data to accomplish their tasks.

## **2. Background**

Particular geographic layers and/or attributes of such layers may be categorized as being sensitive and hence need to be protected from an entire user group. In a recent article in the Federal Computer Watch (Moore, 2007), Clark Clara County in California decided to temporarily stop the sale of government geospatial data online due to the existence of sensitive information in some of these datasets and hence need not be in the hands of every buyer.

Such sensitive data must be hidden from the public domain but allowed for persons based on their job description. This research rather focuses on "how to" control access to sensitive data from authorized users on an intranet web application than to establish guidelines on "what is" deemed as being sensitive.

Using ArcSDE 9.2, SQL Server 2000 and ArcGIS Server, various security methods are examined to assess how best this objective may be achieved. A prototype campus geodatabase is designed with different user groups created for different access levels to data and resources. A secured GIS web application is then created authorizing users on the campus network to these resources. The users are hence presented with an application consisting of data based on their group membership.

### 3. Implemented System Architecture

The implemented system design presents a 3-tiered architecture; the Client, Application and Database tier (see Figure 1). In the client tier, users use the web browser to send requests for geographic layers and perform functions such as, visualization, querying and editing. The Application tier involves Internet Information Services (IIS), which hosts GIS Web application and processes requests from clients using the .NET ADF via the server objects running in ArcGIS Server. Within the database tier, the requested data layers are then retrieved from SQL Server, which stores the spatial data via ArcSDE.

Due to an easier management of a centralized GIS architecture and to ensure that different pockets of information are not contained in the various departments, all data and software components were installed on a Windows 2003 Server system. In the web server configuration, IIS was configured for Integrated Windows Authentication, denying anonymous access. This configuration was chosen for this setup as users would only need to go through a single log-in onto the domain and subsequently access the mapping application.

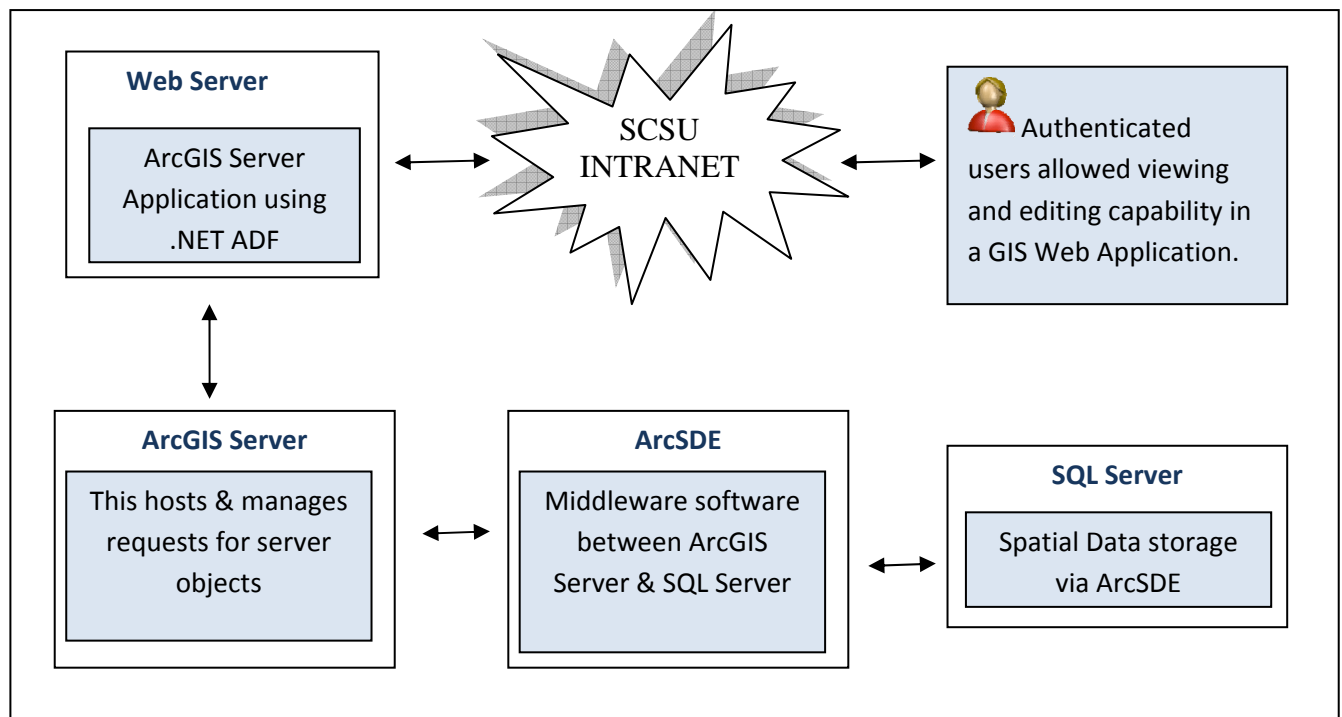


Figure 1: Implemented GIS System Architecture

## **4. User Groups**

To differentiate the users who access the mapping site, two local user groups were created on the Windows Server and SQL Server. These are the “GISSCSU\_Admin” and “GISSCSU\_Editors” user groups. The “GISSCSU\_Admin” user group is a collection of users who make administrative decisions using various layers and attribute information and would hence need access to the entire set of data to which they may have the need for. As opposed to this, the “GISSCSU\_Editors” user group is a set of users who perform edits on a version of the underlying database.

To minimize the overhead of managing user accounts, all other users who do not belong to any of these user groups are classified in an “average user group”. This group is served with a restricted data version of the “GISSCSU\_Admin” group; these users are hence described as having no need for such data.

The “ArcSDE\_Admin” user role is the last group created to perform database administrative functions. It involves a small number of users who are responsible for the performing tasks such as designing (geo) databases, backing up data, assigning user permissions, etc. To provide an additional layer of data security, the “ArcSDE\_Admin” role performs checks on the data edited by the “GISSCSU\_Editors” before being updated into the parent database. In this case, the data is not automatically updated into the underlying database after user-edits, but are analyzed by the “ArcSDE\_Admin” to correct conflicting records from the editors pending synchronization with the underlying database.

## **5. Database Design**

A simple prototype database was designed as a subset of a facility management data model. The key thematic data layers comprised of campus regions, buildings, building floor plans and master plan data with emergency lights and parking layers acting as secondary layers. An Entity-Relationship diagram for this prototype is available in Appendix A.

Since the geodatabase would be served to the user groups defined above, spatial views<sup>1</sup> were built and integrated. For illustrative purposes, we assume that the fields “Funding\_Type”, “LYear\_Funded” and “LYear\_Funding” from the Building table and the Master Plan layer are used for administrative purposes, and hence would be restricted from the average user.

## 6. Web Application

To curtail the number of map documents being served to various user groups; a map document was created for both Admin and Editor User groups and published on ArcGIS Server. The Editor group is served with a version of the database to perform edits, whilst the average users are served with the same but restricted map document of the Admin group. A web application utilizing some of the out-of-the-box functionality provided by ESRI for ArcGIS Server was designed in ASP.NET. To identify a caller for a mapping resource, some security elements available in the .NET Framework were incorporated into the web application.

Using mainly the IPrincipal and IIdentity interfaces in the .NET Framework, checks are performed within ASP.NET to identify the caller and send the appropriate map resource and attribute information. For the available query definitions on the mapping website, ASP.NET’s LoginView control was used to show or hide a query based on a user’s group membership. An additional query tool was defined based on the master plan data and hidden from the average user. A list of restrictions for the various user groups is given below.

Table 1: List of restrictions for the different user groups

USER GROUP	ATTRIBUTE	Layer	Query	Editing
AVERAGE_USER	BUILDING ( Funding_Type, LYear_Funded, LYear_Funding)	Master Plan	Master Plan Query Definition	Restricted
GISSCSU_ADMIN	No restriction	No restriction	No restriction	Restricted
GISSCSU_EDITOR	No restriction	No restriction	No restriction	No restriction

<sup>1</sup> This is a table comprising a subset of attributes from one or more tables in a spatial database.

## **7. Authentication / Authorization**

Security in the mapping application is built on the idea that every user must have a Windows domain account to use the resources it provides. When a user logs onto the domain, the Windows domain controller validates the user's username and password, thereby establishing the user's network authentication credentials. Once authenticated on the system, the user can be authorized or, assigned roles and permissions to use resources available on the network, including the mapping site.

A request by the user for resources on the mapping site would then have to go through various channels for data retrieval. IIS will authenticate the caller, create a logon session and an access token for the caller, and flow the collected details to ASP.NET to verify if the caller has permissions to view the resource. Once ArcGIS Server is set to run under a particular user identity within the web application, access to the application is then granted to the user with the appropriate layers, attribute information and data queries.

## **8. Examined Methods**

### **(i) Controlled Access with Spatial Views**

A spatial view was created from the “building” table, without including the “Funding\_Type”, “LYear\_Funded” and “LYear\_Funding” fields. Unlike a traditional database application where relations are served from the database backend to the application frontend without any intervening medium, the onus of the task was to investigate ways to incorporate the “building\_view” into the map document and/or server object since the server object depends on the map document.

All feature classes were cartographically designed in one map document. From the web application, the “building” and “building\_view” layers were removed from the server object based on group membership. Though the spatial view approach was successful, it had a drawback. The views needed to be re-created each time the schema in the parent table changes. This can be difficult to manage in an application which involves several layers and spatial views.

## (ii) Role-based Security

Security logins were created in SQL Server based on the local user groups defined in the Windows Management Console. The geodatabase was then mapped to the “GISSCSU\_Admin” and the “GISSCSU\_Editors” groups and were subsequently granted the “db\_datareader” and “db\_datawriter” roles respectively. For testing purposes, the Admin group was denied access to specific columns for some layers in the database. Data request from a client in the Admin group revealed that, the restricted columns could still be accessed from the web application.

In the propagation of a caller’s credentials within the entire architecture, IIS and the web application (housed in ASP.NET) are able to identify the caller for the mapping resource, but this is the farthest point that the credentials can traverse. The credentials of the caller cannot be flowed to the SDE as the ArcGIS Container account is solely responsible for the entire data in the map document. This account fetches the data in the map document regardless of who the caller is. Restricting the container account access to a column of a layer or the layer itself would result in the map not being rendered in the application. Hence to ArcGIS Server Object Manager, the caller for the resource is the user impersonated in the “agsusers” local group.

With this observation, it was impossible to control access to attribute information from the database using defined user groups.

## (iii) Controlling Access using ArcObjects

- Map Document – COM Interoperability

This approach of controlling access to the attribute information was considered after the database backend provided little desired functionality. The quest was to find a medium between the database backend and application frontend to apply restrictions to the columns before a map resource is presented to the user on request.

ArcGIS Server hosts and manages the map server objects which serve the map documents designed in ArcMap (ESRI, 2004). These map documents are composed of layers obtained from the database; and it also stores among other properties, the references to the layers and their behavior. In the creation of the server object, various parameters including the map resource that the server object must connect to are defined. When the object is started, it

automatically links with the defined map resource. The behaviors of the layers within the map document are then maintained and served up to the web browser by ArcGIS Server. The assumption then is, if restrictions are applied to the map document, the server object links with the map resource with defined restrictions, and serves that to the various users.

Using ArcObjects, a VBA script was written to customize the map document and automate the process of restricting columns based on user group. Since the written script is COM-based, and role checks need to be performed to assign appropriate data, Windows user checks existing in the .NET framework were exposed to the COM client to take advantage of this functionality.

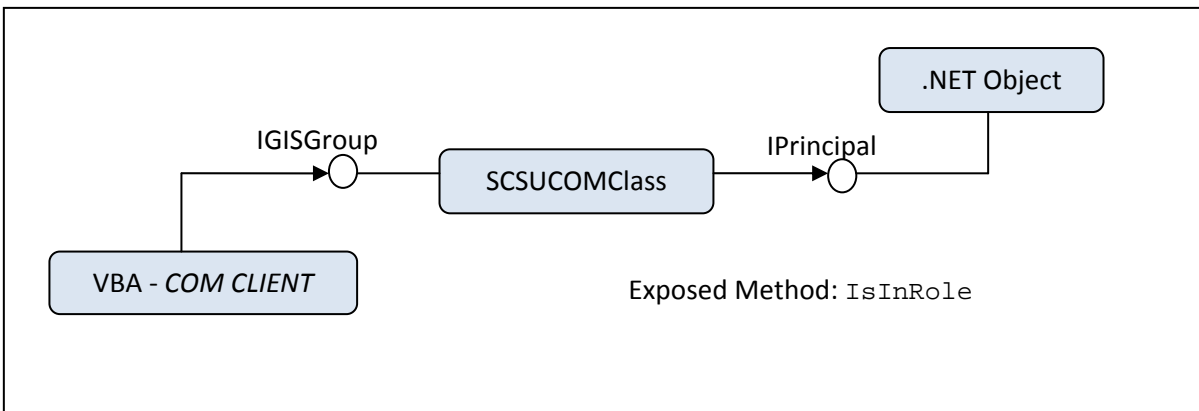


Figure 2: Calling IPrincipal from VBA

Two COM callable wrappers (CCW) were written to expose some properties and methods from the IPrincipal and IIdentity interfaces in the .NET framework to the COM environment as these are not available in the COM architecture. This wraps the .NET object and makes it look like a COM object within VBA. In Figure 2, the IsInRole method in the IPrincipal interface can be accessed in VBA through the created wrapper (SCSUCOMClass). The wrappers were then compiled for COM Interoperability and the respective classes added in VBA as references (see Figure 3).



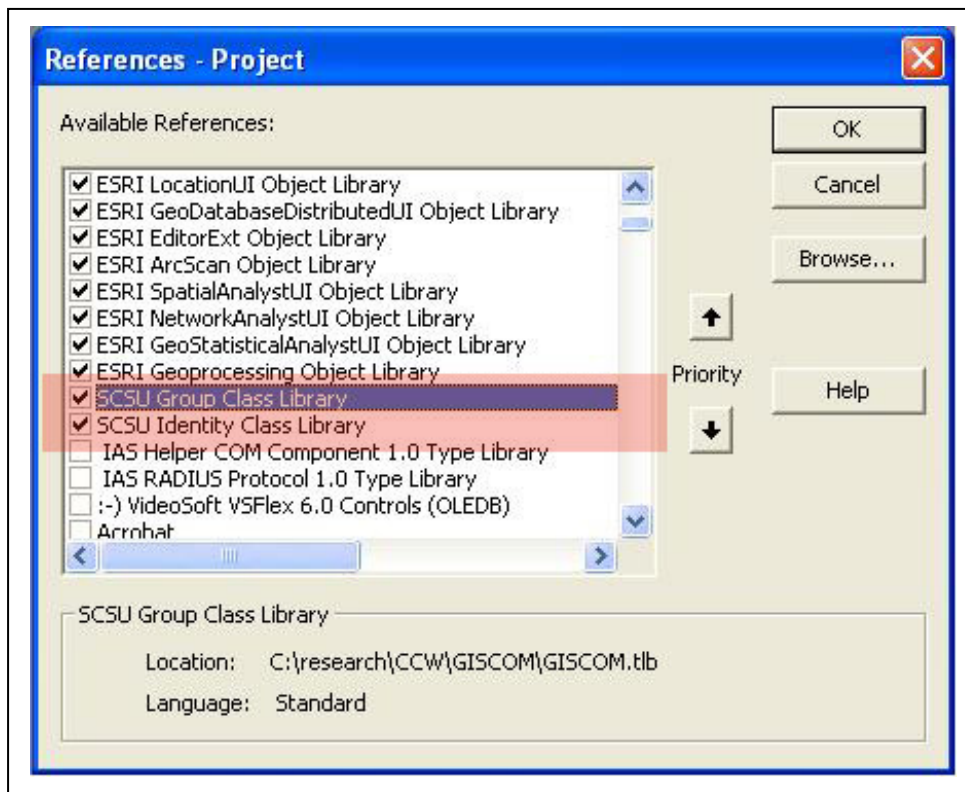


Figure 3: COM Class added as a reference

To check for group membership, an instance was made to the created interface (IGISGroup). The call is then sent to SCSUCOMClass which converts the native COM types to .NET strings and subsequently forwards it to IPrincipal (Platt D. S., 2002). The group membership check is then performed and results are sent back and converted to COM types. From these results, the caller is either allowed or denied access to the attribute information. Though this concept worked with ArcMap as the client, this research encountered some difficulties in implementing it from the web server end.

As already stated, when a server object is initialized it connects to the map resource and subsequently serves the map on user request. Since the written code and the CCW are attached to the map document, executing the embedded code before the resource is served was a difficult hurdle which required knowledge of the semantics between the server object and map resource.

This research was unsuccessful in making the concept produce the desired functionality due to little knowledge of this communication between the map document and the server object.

However, having a medium between the server object and the database to filter layers and attribute information would help define the desired data for users before a security-oriented web application is designed.

- ArcObjects – ASP.NET Application

The last examined method dealt with controlling access to the attribute data within the ASP.NET application using a non-pooled server object, as ArcGIS Server allows applications to utilize ArcObjects to achieve advanced and fine-grained GIS functionality.

A .NET code was written using ArcObjects in conjunction with the .NET Web ADF and applied to the “average-user group” on map request. Hence, after the map resource is initialized within the ASP.NET life cycle, the code is applied before the map is presented to the user. The code first loops through the various layers within the map resource and identify the specific layers from which attribute information should be denied access. On obtaining these layers, the respective attribute field names are then located and subsequently rendered invisible. In this manner, when authorizations are being controlled for the attribute data, the "state"<sup>2</sup> of the server object is temporarily changed for the session to each client. The various user groups then view the same data differently as depicted in Figure 4.

---

<sup>2</sup> This refers to the initial configuration of the map resource.

Administrative User	Average User																																																						
<input checked="" type="checkbox"/> Centennial Hall <table border="1"> <tr><td>OBJECTID</td><td>11</td></tr> <tr><td>Use</td><td>Instruction</td></tr> <tr><td>Closed Lab</td><td>0</td></tr> <tr><td>Open Lab</td><td>0</td></tr> <tr><td>Smart Classroom</td><td>11</td></tr> <tr><td>Built</td><td>1971</td></tr> <tr><td>GSF</td><td>161939</td></tr> <tr><td>Name</td><td>Centennial Hall</td></tr> <tr><td>Region_ID</td><td>C4</td></tr> <tr><td>Abbreviation</td><td>CH</td></tr> <tr><td>Funding Type</td><td>M&amp;E</td></tr> <tr><td>LYear Funded</td><td>2005</td></tr> <tr><td>LYear Funding</td><td>20000000</td></tr> <tr><td>Shape_Length</td><td>246.62</td></tr> <tr><td>Shape_Area</td><td>2090.59</td></tr> </table>	OBJECTID	11	Use	Instruction	Closed Lab	0	Open Lab	0	Smart Classroom	11	Built	1971	GSF	161939	Name	Centennial Hall	Region_ID	C4	Abbreviation	CH	Funding Type	M&E	LYear Funded	2005	LYear Funding	20000000	Shape_Length	246.62	Shape_Area	2090.59	<input checked="" type="checkbox"/> Centennial Hall <table border="1"> <tr><td>OBJECTID</td><td>11</td></tr> <tr><td>Use</td><td>Instruction</td></tr> <tr><td>Closed Lab</td><td>0</td></tr> <tr><td>Open Lab</td><td>0</td></tr> <tr><td>Smart Classroom</td><td>11</td></tr> <tr><td>Built</td><td>1971</td></tr> <tr><td>GSF</td><td>161939</td></tr> <tr><td>Name</td><td>Centennial Hall</td></tr> <tr><td>Region_ID</td><td>C4</td></tr> <tr><td>Abbreviation</td><td>CH</td></tr> <tr><td>Shape_Length</td><td>246.62</td></tr> <tr><td>Shape_Area</td><td>2090.59</td></tr> </table>	OBJECTID	11	Use	Instruction	Closed Lab	0	Open Lab	0	Smart Classroom	11	Built	1971	GSF	161939	Name	Centennial Hall	Region_ID	C4	Abbreviation	CH	Shape_Length	246.62	Shape_Area	2090.59
OBJECTID	11																																																						
Use	Instruction																																																						
Closed Lab	0																																																						
Open Lab	0																																																						
Smart Classroom	11																																																						
Built	1971																																																						
GSF	161939																																																						
Name	Centennial Hall																																																						
Region_ID	C4																																																						
Abbreviation	CH																																																						
Funding Type	M&E																																																						
LYear Funded	2005																																																						
LYear Funding	20000000																																																						
Shape_Length	246.62																																																						
Shape_Area	2090.59																																																						
OBJECTID	11																																																						
Use	Instruction																																																						
Closed Lab	0																																																						
Open Lab	0																																																						
Smart Classroom	11																																																						
Built	1971																																																						
GSF	161939																																																						
Name	Centennial Hall																																																						
Region_ID	C4																																																						
Abbreviation	CH																																																						
Shape_Length	246.62																																																						
Shape_Area	2090.59																																																						

Figure 4: Attribute Information viewed differently

The application of this concept of controlling access to attribute data within the web application has two main advantages. The hidden fields remain invisible for the entire duration of the user session and all form of queries involving the layers. Also, once the code only takes into consideration of the field(s) to be controlled, schema changes to the database does not affect the rendered fields in the application. The programming code is only slightly changed if a hidden attribute is deleted from the database or additional attributes are to be hidden.

## 9. Conclusion

ArcSDE and ArcGIS Server do not provide a clear-cut functionality to achieve attribute-level permissions for various user groups via the web browser. Various tweaks and workarounds may exist to achieving this objective, but the best options would be to control access from either the

database backend or between the database and the application frontend, before a map is requested by a user.

As these areas provided little desired functionality and in the COM case, inconclusive results, the only option left was to examine controlling access within the web application. A workaround to achieving attribute-level permissions was obtained by using ArcObjects within ASP.NET utilizing a non-pooled server object, controlling access to attribute data for an entire application session.

## **10. Future Work**

A pooled server object as against a non-pooled server object supports more users with a lesser amount of resources (ESRI, 2004). Hence, developing a “stateful” application making use of a pooled server object (stateless use of object) would be advantageous in large environments. For this reason, there is an interest in investigating more carefully the extension of this technique in utilizing a pooled server object.

## **REFERENCES**

- Platt, D.S. (2002). Introducing Microsoft .NET 2<sup>nd</sup> Edition. *Microsoft Press*. pp 46-48.
- ESRI. (2004). ArcGIS Server Administrator and Developer Guide. Redlands, California. ESRI Press.
- Moore, J. (2007). Putting security on the map: California map flap rekindles debate about public access to government geospatial data. Retrieved August 22, 2007, from the Federal Computer Watch Website: <http://www.fcw.com/article102637-05-07-07-Print>.

# Appendix A

## Conceptual Model (prototype)

