Glossary:

VPA = Virtual Private ARCHIBUS restriction
SSO = Single Sign-On
LDAP = Lightweight Directory Access Protocol
WebCentral = ARCHIBUS Core Engine
IIS = Internet Information Services (IIS, formerly Internet Information Server) is an extensible web server created by Microsoft
SS = Security Service

1.1 ARCHIBUS System Security

There are more security levels in ARCHIBUS, as described bellow.

1.1.1 ARCHIBUS Web Central Security

Covers the Spring Security framework used by Web Central. Those methods are applied to WebCentral applications. The Authentication methods are:

1. ARCHIBUS Security
2. Single Sign-On (SSO)
3. LDAP
4. Mixed Case: SSO + ARCHIBUS Security or LDAP

1. ARCHIBUS Security

The default scenario, which uses ARCHIBUS security and Web Central as the authentication server. The security service:

a) Presents a login dialog.
b) Receives the request with the login credentials (Username and password).
c) Loads the UserAccount object from a record in the afm_users table for a given Username.
d) Compares the password against the property of the UserAccount (authentication).
e) Uses the UserAccount properties (security groups, security groups from the user role, VPAs) for the authorization.

The password can be stored in authentication repository in clear text or encrypted. It can be encrypted using the older ARCHIBUS format, the newer ARCHIBUS format (PasswordEncoderVersion2Impl), or some other encoding (such as MD5 or SHA).

2. Single Sign-On (SSO)

In this scenario, the site uses an external authentication server to manage passwords. All Web Central requests are routed to this external single-sign on server for authentication.

The Essential SSO Sequence :

1. The Web Server/Application Server receives a request for the Web Central resource.
2. The SSO server authenticates the user.
3. The Web Server/Application Server inserts the SSO Username into the request header, and forwards the request to Web Central. For example, the IIS filter gets Username for the remote user, and inserts this value as the remote user value so that in Tomcat HttpServletRequest.getRemoteUser() will return the Username.
4. The security service loads the UserAccount object from a record in afm_users table for a given Username.
5. The security service uses the UserAccount properties (security groups and VPAs) for the authorization.

Project ID Options :

Option: projectID (such as the project name in afm-projects.xml) can be specified in the request header or in the property file. The specified project will be used as context.

Retrieving the Username from the Request

The Security service gets the Username from the request. It can do so:

- from the request header
- from a cookie
- from HttpServletRequest.getRemoteUser()
- from the request parameter

Mapping SSO Users to ARCHIBUS Users

- The use cases for mapping SSO users to ARCHIBUS user accounts within the security service (SS) are these:
- SSO Username is used as SS Username (one-to-one).
- All SSO Usernames are mapped to single SS Username (many-to-one).
- SSO Usernames are mapped to SS Usernames (one-to-one). The site would need to implement synchronization (one-way) of LDAP usernames with afm_users usernames
- SSO Usernames are mapped to SS Usernames (one-to-one). If there is no matching SS Username, use Guest Username.
  - The mapping can happen in the Web Server/Application Server, or in the SS.
  - Example of the mapping in Web Server/Application Server: IIS filter gets Username for remote user, calls LDAP server with SSO Username and password, LDAP server

authenticates the SSO user credentials, and returns the SS Username for the given SSO Username.

- IIS filter inserts the SS Username as remote user value into the request header.
- Example of the mapping in SS: SS gets SSO Username from the request, calls LDAP server with SSO Username and password, LDAP server authenticates the SSO user credentials, and returns the SS Username for the given SSO Username.

3. LDAP

The Essential LDAP Scenario : In this scenario, user credentials are kept in an LDAP server external to Web Central.

The security service:

1. Presents the login dialog.
2. Receives the request with the login credentials (Username and password).
3. Calls the LDAP server with the Username and password, LDAP server authenticates the user credentials.
4. Loads the UserAccount object from a record in afm_users table for a given Username.
5. Uses the UserAccount properties (security groups, VPAs) for the authorization.

Note: The LDAP configurations are compatible with any LDAP server; however, ARCHIBUS has not tested LDAP configurations with non-AD servers.

There are three Active Directory authentication scenarios provided for Web Central.

One-to-One Configuration
In this configuration, Active Directory (AD) users are mapped to their own unique ARCHIBUS identity. For instance, BIGUNIV\smith is mapped to the smith ARCHIBUS user, and BIGUNIV\davies is mapped to the davies ARCHIBUS user.
Many-to-One Configuration
In this configuration, all Active Directory (AD) authenticated users become one Web Central common/shared user. As an example, AD users BIGUNIV\smith and BIGUNIV\davies will both become a common/shared user on Web Central. By default both users will become the AFM user.
Authority-by-Prefix Configuration
In this configuration, Active Directory (AD) users are mapped to a common/shared user in Web Central according to their LDAP Group assignments.

4. Mixed Case: SSO + ARCHIBUS Security or LDAP
Multiple Authentication Types Required
Some users use computers that belong to the domain, so they are already authenticated by the SSO server.
Other users use computers outside of the domain, so they are not authenticated by the SSO server.
These two categories of users are mapped to different instances of Web Central. The two instances of Web Central use the “clustered set of application servers” feature of Web Central. This avoids license file copying.
The domain users use Web Central instance configured for SSO.
The users outside of the domain use Web Central instance configured for ARCHIBUS security or LDAP. That instance has to use secure HTTP channel, if the domain username and password (LDAP) will be transmitted over this channel.

1.1.2 ARCHIBUS hierarchical security

Hierarchical security is applied on Windows applications and on web applications too. This security level is a refinement to the regular security group codes and further controls access to columns of data (fields). It adds a flexible system for organizing security access into hierarchies so that the application security organization can reflect the structure of your organization. It also gives you the ability to aggregate security groups into

roles so that all of the permissions that each type of staff member needs to execute their mission can be assigned in one step.
Hierarchical security enables you to assign roles access to domain, activity, or functional role tasks in one step. The roles can then be assigned to users. This makes it possible for large numbers of schema elements (e.g. fields, tasks) to be aggregated according to function. It also means that these aggregates can be assigned in powerful and flexible ways.
Hierarchical security regroups the ARCHIBUS tasks and fields into roles and groups. Implementing security becomes a matter of deciding how roles at your site differ from the standard.
For instance, do your CIOs also want to see all of the review-level detail? If so, add the %rev% group to their role along with the %cio group.
Two key goals of hierarchical security are:

- Reduce the number of groups that a site needs to define and maintain.
- Ship a default security schema that needs minimal modification for use at end-user sites.

Roles and Groups
Hierarchical security for group codes covers most of the distance to achieving these goals, with roles and VPAs giving sites the flexibility to further combine and map these security groups according to their own needs.
Roles
Roles correspond to the types of users (and therefore the types of access each user needs).
Each role is like a key ring giving access to a select set of areas of the application. Roles can include parameters controlling menu access, row access (VPA), and column access (security groups and hierarchical security groups).
Groups
If a role is like a key ring, the individual groups are the keys.
These keys can grant access to Navigator items, Hotlist items, processes on the Process Navigator, and edit and review rights on individual fields.
You assign groups to roles to assemble the selection of keys or rights that that role should have. You can also assign groups to individual users, but you should favor assigning them to roles, since this method reduces the amount of security administration you need.
You can use regular groups (which must match exactly) or hierarchical groups. Hierarchical groups act like a hierarchy of master keys, with each master being able to open an entire set of related doors. Just as with physical keys, using these "master keys" or hierarchical groups can greatly reduce the

number of groups you need to define, maintain, and assign.

Users

At sites using security, users are those allowed to log in to the system. When they log in, users are granted the rights associated with their role. They may have other per-user settings, such as a VPA restriction.

Processes

If your site is using the Process Navigator interface, each user may be assigned one or more processes (e.g. the Craftsperson or the Supervisor process). These determine what role-specific tasks appear on your Process Navigator menu when you log in.

1.1.3 Virtual private ARCHIBUS

This security level is applied on Windows applications and on web applications. This is an extension of application security that controls access to rows of data (records). With this, you can partition your entire database by region, division, or organization, yet keep all of your data in one central database with a common set of rollup or validating codes.
The Virtual Private ARCHIBUS restriction is like a view-to-view restriction you set on a table in one view (e.g. "bl.bl_id LIKE HQ%"), that is to be applied to all subsequent views that get loaded in that session. However:

- It is defined on a per-role basis and is initialized and added to each user's profile on login.
- It is established when the user logs into the database and remains for the duration of the session.
- It applies to the Select Values dialog as well as to the view.
- It applies to the Drawing List in both ARCHIBUS and the Overlay (hence you don't see drawings managed by other sites).
- It can be set globally on all similar tables or fields with a single statement.
- It cannot be cleared with the Clear Restriction command.

Note: VPA applies to the data retrieved by the program, but not to the calculations or actions. For instance, if a staff member runs the recalculate chargeback task, it recalculates for all data.
Note: If more than one VPA restriction is specified, the restrictions will be joined with an "AND".
ARCHIBUS uses two types of VPA restrictions:

- Default Site and Building Code VPA Restrictions in the A/FM Users Table.
  Most sites establish VPA restrictions based on geographic responsibilities. For this usage, ARCHIBUS has a short-hand for specifying the VPA restriction. In the A/FM Users table, you enter the comma-delimited list of Building Codes or Site Codes to which each user should have access.

This feature is useful for implementations that have dozens, or even hundreds, of sites and buildings to manage data access for.

Default VPA Rules

When you enter values in the Building Code List or Site Code List of the A/FM Users table, the program uses the following rules in establishing the VPA restriction.

- List Items. List Items add a WHERE IN clause:
  - ‘HQ’ will add the clause (bl_id IN (‘HQ’))
  - ‘JFK-A, JFK-B’ will add the clause (bl_id IN (‘JFK-A’, ‘JFK-B’))
- Nulls. NULL will add an IS NULL clause:
  - NULL will add the clause (bl_id IS NULL)
- Wildcards. Items with wildcards will add a LIKE clause:
  - HQ% will add the clause (bl_id LIKE ‘HQ%")
- Compound Conditions. Multiple conditions will be OR’ed together:
  - ‘NULL,HQ%, JFK-A, JFK-B’
  - will add the clause
  - ((bl_id IS NULL) OR (bl_id LIKE ‘HQ%") OR (bl_id IN (‘JFK-A’, ‘JFK-B’)))
- Table and Field Names. Table and field names will be replaced as appropriate for the table. For instance, the validated "Building Code" fields in the in the Move Order table would be mo.bl_id_from and mo.bl_id_to.
- Building and Site Restrictions. Restrictions on the Building Code List and restrictions on the Site Code List are AND’ed together. This is because they are actually separate VPA restrictions, and all separate VPA restrictions are AND’ed.
- Default VPA Details -- The default VPA restriction establishes a validating table VPA on the Buildings table and/or on the Sites table.
- VPA Restrictions Entered in the A/FM Roles Table The role is then assigned to a user.
  You specify VPAs per-role, that is, in the A/FM Roles table. There are a number of options to VPAs that are summarized below. The use of these options will become clearer in the examples in the How to procedures listed below:
  A/FM Role VPA Restriction Format
  When specifying a restriction in the VPA Restriction field of the A/FM Roles table you use an XML format.
  The VPA has three forms:
- A restriction with type sql specific to a particular table. This is used when the restriction is on a single table, the restriction must compound restrictions using OR, or the restriction must relate tables (and so must state the tablename qualifier and field names explicitly).
- A restriction with type ForValidatedTables template for a restriction that can be expanded for all tables with a given name or that hold fields that validate on the table with the given name. This is the most common form of VPA restriction.
- A restriction with type ForFields template for a restriction that can be expanded for all fields with a given name. This is typically used for "generic" restrictions on non-validated fields, such as the tc_service enumeration field.