A Framework for Authentication and Authorization in Plug-in-Based Control System Software

Matthias Clausen, Jan Hatje, Helge Rickens (DESY, Hamburg), Jörg Rathlev (University of Hamburg)

Abstract

Preventing unauthorized use is a concern for many software systems, including control system software. The authorization mechanism used by a system should be pluggable, so that the software is not tied to a specific infrastructure. For the Control System Studio (CSS), we have developed a generic authorization framework which can be used by applications built on top of CSS to authorize user actions. For example, the framework provides support for the creation of menu items or graphical display elements that are automatically enabled and disabled based on the user's permissions. The framework is implemented in plug-ins which can be exchanged to interact with different infrastructures. Currently available implementations use standard Java authentication and authorization techniques to integrate with Kerberos and LDAP systems.

Authentication and Authorization in Control System Studio

Applications in the Control System Studio (CSS) are implemented as plug-ins. The CSS Platform provides a common basis for these application plug-ins. Several of the CSS applications have the requirement to enable or disable certain actions based on the permissions of the current user. To support this requirement, the CSS platform provides a login mechanism which verifies the user's identity (authentication), and a framework which can be used by application plug-ins to check the permissions of the user (authorization).

Using the Framework

There are two main entry points for using the framework, the Security Façade and the Activation Service.

The **Security Façade** is a low-level API which can be used to directly check whether a specific permission is granted to the current user. The Security Façade can also be used to register listeners which will be notified when the user's permissions change.

The **Activation Service** automatically enables and disables objects based on the current user's permissions. It uses adapters to enable and disable the objects, so it can manage all types of objects and different adapters can perform different actions (for example, enable/disable or show/hide). The main purpose of this service is to automatically enable and disable user interface widgets. The CSS Platform provides adapters for Eclipse actions and SWT widgets, and the SDS provides an adapter for SDS widgets.

Pluggable Implementation

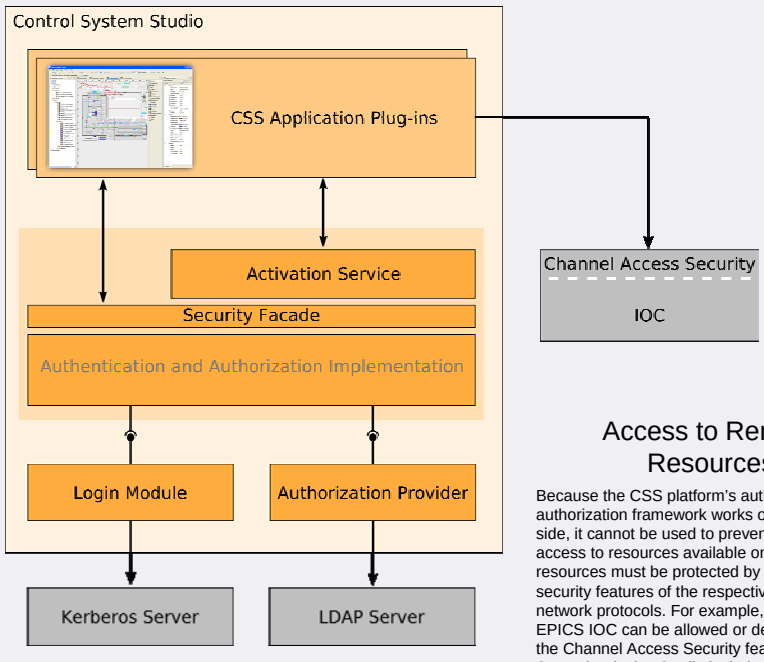
In its implementation, the authentication and authorization framework of CSS relies on two extension points through which login modules and authorization providers can be contributed. The login module is responsible for authenticating the user. The authorization provider is responsible for retrieving the list of permissions that are granted to the user and the permissions that are required for an action.

Implementations Provided by DESY

DESY provides a login module based on JAAS and an authorization provider based on LDAP.

JAAS Login Module: This module uses the Java Authentication and Authorization Service (JAAS) to authenticate the user. At DESY, we use this module to authenticate users against a Kerberos server. The same module could also be used with other authentication services that support the JAAS standard.

LDAP Authorization Provider: This plug-in reads permission information from an LDAP directory server. We encode permissions as role-group tuples. Each user can belong to one or more groups with one or more roles, and each action can be permitted for a number of role-group combinations. The permission information is stored in the LDAP server based on a custom LDAP schema.



Access to Remote Resources

Because the CSS platform's authentication and authorization framework works on the client side, it cannot be used to prevent unauthorized access to resources available on a server. Such resources must be protected by using the security features of the respective servers and network protocols. For example, access to an EPICS IOC can be allowed or denied by using the Channel Access Security feature. The CSS Synoptic Display Studio includes a feature to make the Channel Access Security settings visible in the user interface.