Access Control

# Beschreibung

Das Access Control (AC) ist eine Service Fabric Application (SFA) die zum Speichern und Verwalten von Identitäts- und -berechtigungsinformationen in Verbindung zu Benutzern und Gruppen bestimmt ist.

AC ist dabei so modular aufgebaut, dass es die unterschiedlichen Datenprovider in eigene Komponenten kapselt die durch gemeinsame Interfaces austauschbar sind.

Derzeit existiert als Authentifizierungskomponente der Breanos Identity Provider, welcher über ein Entity Framework-Datenbankmodell Informationen zu registrierten Benutzern und deren Gruppierungen in einer SQL Datenbank ablegt und bereitstellt.

Als weitere Komponente existiert außerdem der Menu Provider.

Dieser nimmt einerseits KPU-seitig definierte Menüstrukturen entgegen und kann andererseits bei Anmeldung eines Benutzers die Einzelmenüs in ein auf den jeweiligen Benutzer zugeschnittenes Menü aggregieren und konsolidieren.

Kommunikation zum Access Control verläuft über das SFA-eigene Remoting über das InterfaceAccessControlService.Interfaces.IAccessControlService, das Getter- und Setter- Methoden für die unterschiedlichen Datensätze bereitstellt.

# Installation

AC wird auf einem Service Fabric Cluster als eigene SFA auf einer beliebigen, über die Parameter des SFA-Ausrollungsprojektes konfigurierbare Anzahl an Knoten installiert.

# Konfiguration

AC und seine Unterkomponenten können wie folgt konfiguriert werden:

Im Unterordner ./PackageRoot/Config von AC befindet sich die Date „settings.xml“ mit Parameterwerten für die einzelnen Komponenten.

Es lässt sich einstellen welche Untermodule verwendet werden sollen (z.B. Breanos Identity Provider als Identity Provider). Wird eine neue Providerkomponente hinzugefügt muss AC entsprechend angepasst werden:

* Das neue Modul muss zu den Abhängigkeiten hinzugefügt werden,
* Der Source Code von AC muss angepasst werden um eine entsprechende Einstellung in der settings.xml für das neue Modul zu erkennen.
* Die settings.xml kann angepasst werden um im AC das neue Modul zu verwenden (z.B. Wechsel vom BreanosIdentityProvider zu einem ActiveDirectoryIdentityProvider)

# Abhängigkeiten

AC benötigt die folgenden Bibliotheken, zusätzlich zu denjenigen die über die offiziellen Nugetserver verfügbar sind (alternativ bei Verfügbarkeit der Quellen auch als direkte Verweise):

* AccessControlService.Data
* AccessControlService.Interfaces
* BreanosConnectors.Kpu.Communication.Common
* BreanosConnector.SerializationHelper.Standard
* BreanosIdentitiyProvider
* IdentityProvider.Interfaces
* MenuProvider.Interfaces
* WpfMenuProvider

# Anforderungen

AC kann zum laufenden Betrieb aktive Datenbankverbindungen benötigen

Einige Komponenten von AC können über eine konfigurierbare Datenbankverbindung Logmeldungen in einer Datenbank speichern

Diese Umstände können eine verfügbare Datenbank notwendig machen.

# Description

Access Control (AC) is a Service Fabric Aplication (SFA) that can store and handle user credentials and permission information in relationship to users / user groups.

AC is built modularly, encapsulating the information providers provider (currently user authentication and client menu generation) into its own component so it can be exchanged more easily.

At present, only the Breanos Identity Provider is implemented as Identity Provider which uses an Entity Framework database model to persist and provide user identity information and grouping in an SQL database.

Furthermore, AC holds a Menu Provider, in a similar fashion to the Identity Provider, which is used for pre-processing KPU-defined menus into a menu structure usable by the client software to render the available menu options.

Interfacing with AC works via SFA Remoting over the interface AccessControlService.Interfaces.IAccessControlService which provides Getter- and Setter- methods for the various different sets of data.

# Installation

AC is to be installed on a Service Fabric Cluster and runs as a stateless service on a configurable number of nodes.

# Configuration

AC can be configured along with its sub modules (Identity Provider and Menu Provider)

AC has a file called “settings.xml” located in ./PackageRoot/Config/ which contains the parameter values for each of its components.

AC itself can be configured by defining which Identity Provider / Menu Provider to use.

If a new component is being added, AC itself must be adapted accordingly:

* The new module must be added to AC’s dependencies
* AC’s source code must be adapted to recognize the relevant settings and setting-values for use of the new module
* Settings.xml itself can be added to change use from the old module to the new one (e.g. switch from BreanosIdentityProvider to ActiveDirectoryIdentityProvider)

AC and its currently implemented components can further be configured by providing a connection string for the SQL database to use for storage and retrieval of the user data.

# Dependencies

AC requires the following libraries additionally to the ones available via public Nuget server (alternatively if sources are available, direct dependencies to the libraries can be used):

* AccessControlService.Data
* AccessControlService.Interfaces
* BreanosConnectors.Kpu.Communication.Common
* BreanosConnector.SerializationHelper.Standard
* BreanosIdentitiyProvider
* IdentityProvider.Interfaces
* MenuProvider.Interfaces
* WpfMenuProvider

# Requirements

AC may use database connections during its normal operation

Some components of AC may store log information in a database

These circumstances may effectuate the requirement of an available database.