**TSG-SA5 Meeting #162 *S5-253953d1***

**, , – Revision of S5-253465**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** | **0011** | **rev** | **1** | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: compr*  *ehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **X** | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Rel-18 CR TS 28.319 Correct errors on the MSAC information model | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** | SA5 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | MSAC | | | | |  | ***Date:*** | | | 2025-08-14 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | | Rel-18 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | On the one hand, there are two figures providing description of the information model for MSAC: Figure 6-1 and Figure 7.2-1. However, it is not clear why both figures are needed, since they aim to pursue the same objective. In addition, their content is misaligned (e.g., attributes for Identity and Role classes are missing in Figure 6-1, attributes for AccessRule class do not match, etc).  Notification types is not componentC, but componentA instead.  On the other hand, there are errors in the properties of some attributes. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | * Update Figure 6-1, keeping only class names and removing constituent attributes. This allows defining a clear boundary with respect the intention of Figure 7-2.1. * Update Figure 7.2-1, by fixing the type of some attributes (e.g., identityName), removing unneeded attributes (i.e., roleList) and rename wrong attributes (i.e., replace resources by dataNodeSelector). * Update componentCData description. * Fix errors in the properties of some attributes specified in Clause 7.4. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The content of Figures 6-1 and 7.2-1 do not match the information model defined for MSAC.  Wrong specification of some class attributes. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 6, 7.2, 7.4.1 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **x** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | | No stage 3 | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

|  |
| --- |
| **1st Change** |

# 6 High level solution description

This clause elaborates the information architecture of the classes necessary to enable the role based access control explained in the clause above. This clause elaborates the concept into further logical classes that need to interact to enable the design.

The classes, attributes specified here shall be seen as concepts that might not have a direct mapping in some solution sets.

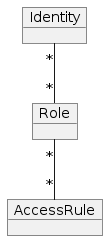


Figure 6-1: Information model for role based access control relationship

The information model above depicts the following to realise a role based access control relationship:

- One Identity could be associated to one or more Roles.

- One Role might be associated to one or more Access Rules. One Role might be associated to one or more Identities.

- The Access rule class allows protecting resources, by specifying which permissions are eligible for each resource. For more information, see clause 7.3.3.

The classes represent the information architecture that is necessary to implement an access control system. They are and not meant to be provisioned and managed like the NRM classes.

This clause provides an overview of the relationships between relevant classes in UML format.

As mentioned in the use cases the classes are provisioned during the integration time by a network operator. The data is sent to an authentication and authorization service producer.

Post this during integration time when a MnS consumer invokes an operation on the MnS producer, the authentication and authorization service producer validates the action on the resource to enable the decision for the MnS producer. The decision could be that the MnS producer allows or disallows the action on the resource.

The information architecture translates to the design which considers the below:

- The authentication function mainly contains the 'who' of the MnS consumers. The authorization function contains the information of the resource and the action of the MnS producer associated to the 'who' of the MnS consumers. These functions could be collocated or distributed.

- The MnS consumer is associated to 'who' is carrying out the operation and has to be known to the resource owner who will provide this information. The MnS consumer interacts with an authentication function to identify itself.

- The MnS consumer interacts with the authorization function whether it can carry out the action on the resource. The resource owner also provisions the information associating various resources and corresponding actions to valid MnS consumers as a pre-step.

- The MnS consumer interacts with the MnS producer after getting responses from the authentication and authorization functions. The MnS producer further does a validation if the action on its resource can be allowed or not.

The above interactions can be realized in any implementation.

|  |
| --- |
| **Next Change** |

## 7.2 Class diagrams

This clause specifies the class diagram.

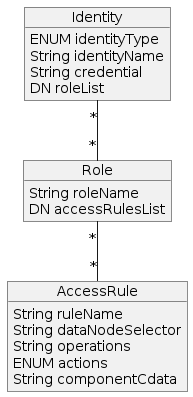


Figure 7.2-1: Classes for role based access control

## 7.3 Class definitions

### 7.3.1 Identity

#### 7.3.1.1 Definition

This class represents an identity of a MnS consumer. It is used for authentication and authorization.

The MnS consumer can be a human or a machine user. This class enables the creation and storage of an identity of a MnS consumer. The information in this class is the starting point for a MnS consumer to identity who it is. This is validated against an authentication service producer.

For the authentication operation to take place the identity related information has to be provisioned into the system by a network operator who could be an administrator. The administrator adds the identityType attribute and identityName attribute which characterizes a machine user or human user respectively. For example, an operator might have an identity like a tenant mapped to the relevant list of roles.

Attribute credential is used to provide information for the credential used together with identity when requesting authentication. The examples of credential are password, certificate, biometric, etc.

The roleList attribute defines the role names associated to a particular Identity.

The class stores the details of the expected tasks to be performed by an identity. The tasks are what is to be done on the network management system. To ease the administration on the system, the tasks are organised as roles. The user can be associated to one or more roles.

#### 7.3.1.2 Attribute

The Identity class includes the following attributes:

|  |  |
| --- | --- |
| **Attribute Name** | **S** |
| identityType | M |
| identityName | M |
| credential | O |
| **Attributes related to role** |  |
| roleList | M |

#### 7.3.1.3 Attribute constraints

None.

### 7.3.2 Role

#### 7.3.2.1 Definition

The Role class represents a task or collection of tasks in a network management system.

The Role class enables the storage of information as to what resources and actions an identity can work upon. This class maintains the resources that are known to the management system. This contains all the granular level resources and the corresponding actions.

The roleName attribute defines the name of a role.

The accessRulesList attribute contains a list of access rules that contain the list of granular permission sets. This could be the possible order in which the access rules are considered by the MnS producer.

#### 7.3.2.2 Attribute

|  |  |
| --- | --- |
| **Attribute Name** | **S** |
| roleName | M |
| **Attribute related to role** |  |
| accessRulesList | M |

#### 7.3.2.3 Attribute constraints

None.

### 7.3.3 AccessRule

#### 7.3.3.1 Definition

The AccessRuleclass represents the granular resource and actions in a network management system on which an action has to be performed.

This class enables the storage of the resource types in the system and the possible actions that are allowed on it. The permutations and combinations of these permissions are assigned to a role.

The roleName attribute binds the instances in the network as well as the permissions and the operations allowed upon it.

The dataNodeSelector attribute defines the resources. The resources are classes(IOC) or instances of classes(MOI) in the network that need to be access controlled. The resources define the root instances or the leaf instances. For example, the Managed Element could be the root object and the attributes could be referred to as the leaf objects. The resources here could be whole classes or specific instances of classes with a known DN value or could be an expression(e.g.: XPATH or JEX) that could be resolved by the producer to get the nodes at runtime.

Examples of the resources attribute value could be as below. Please note this is not an exhaustive set of examples and shown for depiction purpose.

a) Values related to IOC:

- Description: this means that:

- all attributes of an IOC are eligible for the access rule.

- at operation time, all instances of this IOC are eligible for the access rule.

Examples 1:

1) IOC name : "ManagedElement"

2) Expression resolving to IOCs under a subnetwork SN1:

"/SubNetwork[id="SN1"]/ManagedElement"

b) Values related to one or more instances of an IOC:

- Description: this means that:

- all attributes of the IOC are eligible for the access rule.

- at operation time, only the specified instances of this IOC are eligible for the access rule.

Examples 2:

Specific instance of IOC name : "SN1/ME1"

c) Values related to one or more IOC attributes:

- Description: this means that:

- only the specified attributes of the IOC are eligible for the access rule.

- at operation time, all attributes of the instances of this IOC are eligible for the access rule.

Examples 3:

1) Attribute name: "SubNetwork/ManagedElement/vendorName"

2) Expression resolving to specific instance of attribute name "/SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/attributes[vendorName="Company XY"]"

3) Specific attribute instance: "SN1/ME1/vendorName='Company XY'"

d) Any combination between a-c.

The operations attribute defines the list of operations that are permitted on the resources value encompassed under this ruleName.

The actions is an optional attribute which specifies whether the operation allows to permit all or deny all and maybe used depending on the solution set.

The componentCData is an optional attribute which specifies notification types and performance metric names. The dataNodeSelector shall specify objects when access rights for notifications and performance metrics are specified.

#### 7.3.3.2 Attribute

|  |  |
| --- | --- |
| **Attribute Name** | **S** |
| ruleName | M |
| dataNodeSelector | M |
| operations | M |
| actions | O |
| componentCData | O |

#### 7.3.3.3 Attribute constraints

None.

|  |
| --- |
| **Next Change** |

## 7.4 Attribute definitions

### 7.4.1 Attribute properties

The following table defines the properties of attributes specified in the present document.

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| identityType | This indicates a type of identifier  AllowedValues: username, email address, phone number, IP address, machine user | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| identityName | This defines a readable string to uniquely represent an identity  AllowedValues: N/A | type: String  multiplicity: 1  isOrdered: NA  isUnique: NA  defaultValue: None  isNullable: False |
| credential | The credential of an MnS consumer or producer used for authentication with authentication service producer. It could be password, certificate, key, pass phrase, etc., based on authentication protocol and factor.  AllowedValues: N/A | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| roleList | This defines the list of roles (represented by Role resources) associated with an identity  AllowedValues: N/A | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| roleName | This string defines a unique representation of the name of a role  AllowedValues: N/A | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| accessRuleList | This defines the list of access rules (represented by AccessRule resources) associated with a role  AllowedValues: N/A | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| ruleName | This string defines a unique representation of the name of an access rule.  The name of the access rule could also contain the name of the management service  AllowedValues: N/A | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| dataNodeSelector | This attribute contains an expression allowing to select data nodes (Component type B). The expression semantic and syntax is SS specific.  AllowedValues: N/A | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| operations | This defines the Component A related operations.  The operations related to attributes are also contained in this set.  The operations are of the MnS as defined in TS 28.532 [6].  AllowedValues: N/A | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| actions | This defines whether the operation is allowed or denied on the operation  AllowedValues: allow, deny | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| componentCData | This attribute defines alarm types and performance metrics.    AllowedValues: N/A | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |

|  |
| --- |
| **End of Changes** |