|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 41 |
| Source:\* | Leila Le Brun, Orange ([leila.lebrun@orange.com](mailto:leila.lebrun@orange.com));  Chrystel Gaber, Orange ([chrystel.gaber@orange.com](mailto:chrystel.gaber@orange.com) ) |
| Date:\* | 2019-06-30 |
| Reason for Change/s:\* | To enhace dynamic authorization by adding nested token |
| CR against: Release\* | Rel-4 |
| CR against: WI\* | Active <Work Item number>  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0003 |
| Clauses \* | TS-0003 clause 7.3.2.2, clause 7.3.2.3 |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Other TS/TR(s) impacted | <TS/TR number>, <Version Number>, and <Description on which aspect should be reflected in this TS/TR> |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2019 (do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

If this is a correction, and the change applies to previous releases, a separate “mirror CR” should be posted at the same time as this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made e.g. a change impacting 5 tables should not only include a proposal to change only 3 tables. Include any changes to references, definitions, and abbreviations in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar.

Use change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the proposed new clause is located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to the content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

Ofter AEs are connected to several networks and AEs resources are requested and used by different originator AEs/CSEs. Unfrounitely it exposes AEs to the attacks when the hacker using one acces network/CSE to get a full control on AEs and to propage the attack to other networks to which AEs is connected.

Example of this attack if the attack to ‘Target’ retailer when 40 million individuals were exposed and when hackers have sucesseded to get credit card data including account numbers, CVV security codes, and expiration dates

Hackers began with stolen credentials from an HVAC company that acts as a contractor to several Target locations. (HVAC company IT is connected to Air Conditionner’s installed in the Target shops in order to monitor energy consumption and temperatures in the stores to save on costs and to alert store managers if rempertatures in the stored fluctuate otside of an acceptable range that could prevent customers from shopping in the store).

Once entered in the HVAC company IT, hackers have sucessed deployment of the malware on the Air Conditionners which where connected to the internal Target network. Via this internal network, the intruders pushed the card-stealing malicious software to the cash registers within Target stores.

Example Target attack using HVAC

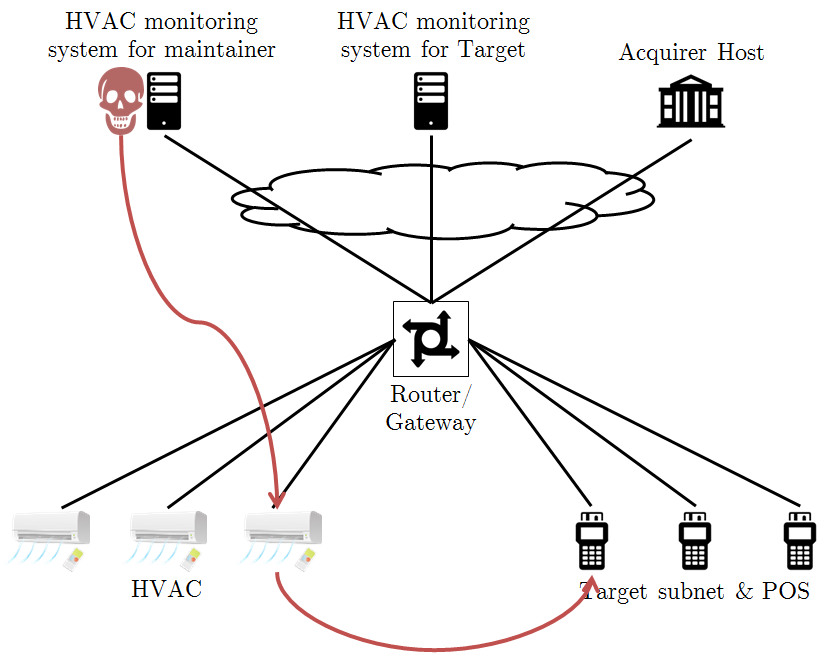


Figure 1: Attack on Target retailer via HVAC air conditionners

To prevent this kind of attacks, ODSI europenian project proposes to add an ASN node and use multi-tenant isolation within it. This will allow giving external company an access to the resources and in the same keep a strict control of ressources by the owner of AEs.

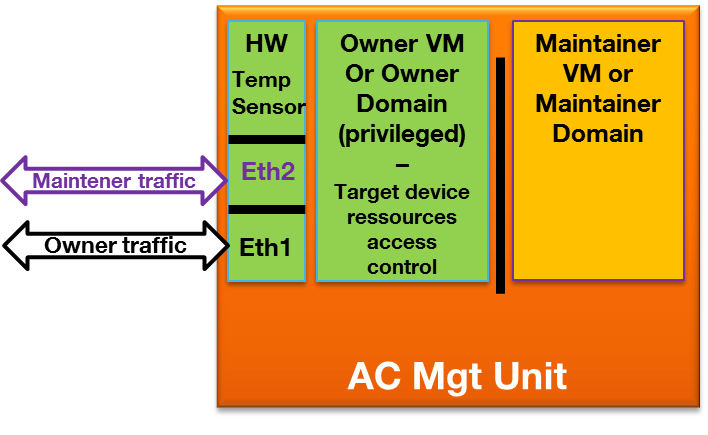


Figure 2: Multi-tenants isolation within the device

In case of multi-tenant isolation, it will be possible to control an access to the sensitive resources of the privileged VM/Domain. In order to allow this control, the proposal is to allow DAS from the Owner to control an access to the Owner VM/Owner Domain sensitive resources with Owner token. Then:

* Maintainer token will be used to control VM1 resources in case maintener needs to access VM1
* Owner will be used to control VM0 resources in case owner need to access VM0

**Multi-Tenant Use Cases**

Air Conditioners are installed as part of Smart Office.

Maintenance of Air Conditioners is subcontracted to a **maintainer (M)** who manages the device resources remotely (humidity level, temperature, on/off, power consumption, firmware version,…).

**Owner (O)** is also able to manage the device remotely (humidity level, temperature,…)

Each actor is associated to a Domain or VM in the device. The hypervisor & owner privileged VM manage the “real” resources and expose “virtual resources” to tenants (M).

Each Domain is associated to a ASN-CSE (ASN-CSE\_O, ASN-CSE\_M).

M has an algorithm for maintenance deployed on the HVAC. M therefore has access to device resources & domain resources (version, results of processing, alerts)

**Owner, Maintainer** each have their own DAS (DAS\_O & DAS\_M). ASN-CSE\_O validates Owner token(s) emitted by DAS\_O and ASN-CSE\_M validates Maintainer token(s) emitted by DAS\_M.

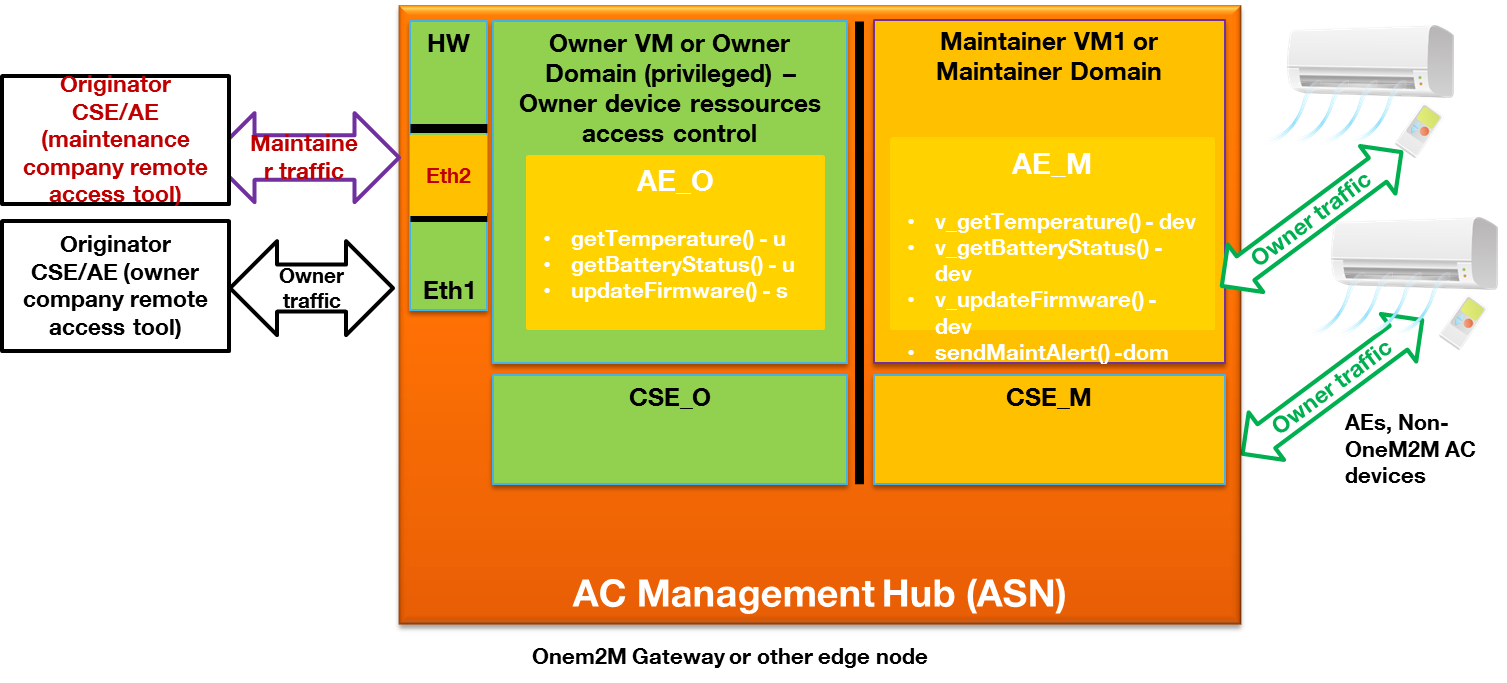


Figure 3: Air Conditionners multi-tenant Use Case

Let’s describe several Use Cases showing that different types of usage will require different tokens to isolate sensitive resources and to give different priviledges.

1. **Case 1:** If the administrator from O (Owner) wants to modify temperature (device resource), then O shall collect an Owner token from DAS\_O. In this case no need for the second Token:

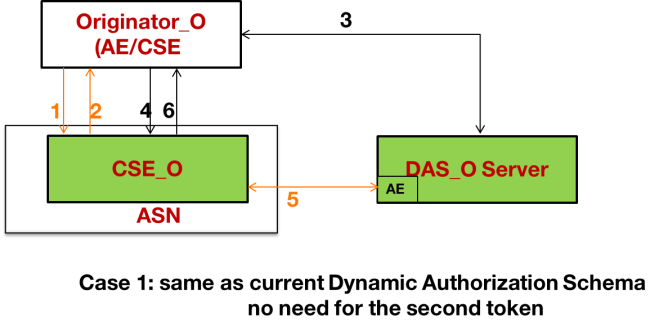


Figure 4: Case 1, no need for the the second Token

1. **Case 2**: If the administrator from M wants to update some AI parameters (M domain ressource), then M shall collect a Maintainer token from DAS\_M. In this case no need for the Owner Token:

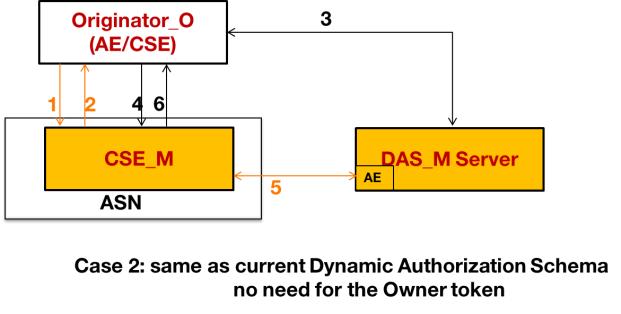


Figure 4: Case 2, no need for the Owner Token

1. **Case 3:** If the administrator from M wants to modify the temperature (unsensitive device resource) then M shall collect a Maintainer token from DAS\_M. In this case no need for the Owner Token:

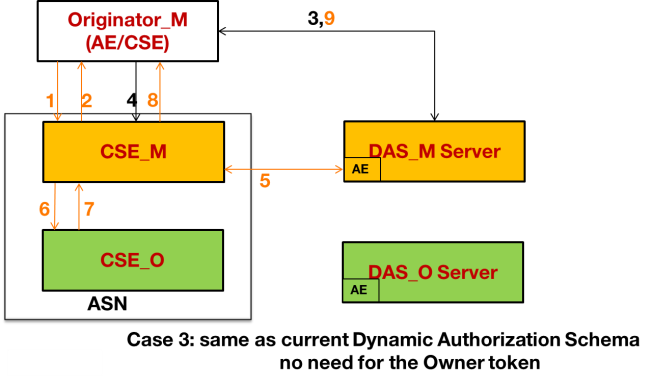


Figure 5: Case 3, no need for the Owner Token

1. **Case 4:** If the administrator from M wants to modify the firmware (sensitive device resource), then M shall collect a Maintainer token from DAS\_M which will collect an Owner token from DAS\_O and include it as a second token in the tokens list. In this case:

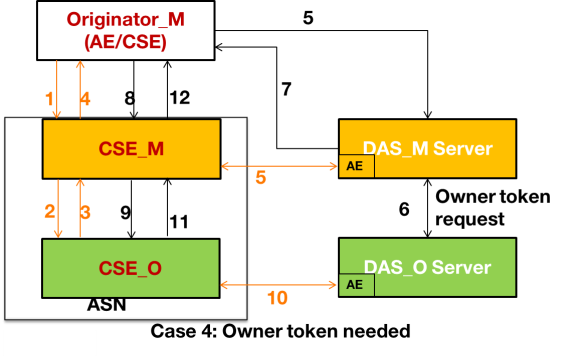


Figure 6: Case 4, Owner Token as a second token needed

For each request from M related to a device resource :

- DAS\_M needs to know whether a DAS\_O token is needed or not. DAS\_O can communicate this information through a security policy, documentation or a specific API.

- CSE\_M sends to the hypervisor & CSE\_O a request to read/modify

**Use Cases Summary:**

| **Case N** | **Actor performing an action on AE** | **Case Description** | **DAS providing token** | **Both DAS Owner and DAS Maintainer Tokens needed** |
| --- | --- | --- | --- | --- |
| Case 1 | O (Owner) | The administrator from O wants to modify temperature (device resource) | O collects a token from DAS\_O | no |
| Case 2 | M (Maintainer) | The administrator from M wants to update some AI parameters (M domain ressource), | M collects a token from DAS\_M | no |
| Case 3 | M (Maintainer) | The administrator from M wants to modify the temperature (unsensitive device resource) | M collects a token from DAS\_M | no |
| Case 4 | M (maintainer) | The administrator from M wants to modify the firmware (sensitive device resource), | M collects a token from DAS\_M which will collect a token from DAS\_O and include it as a second token in the list. *Issuer* field will indicate that the second token is provided by DAS\_O | yes |

Here are the call flow detailed on the Case 4, when the Owner Token as second token from different DAS is required:

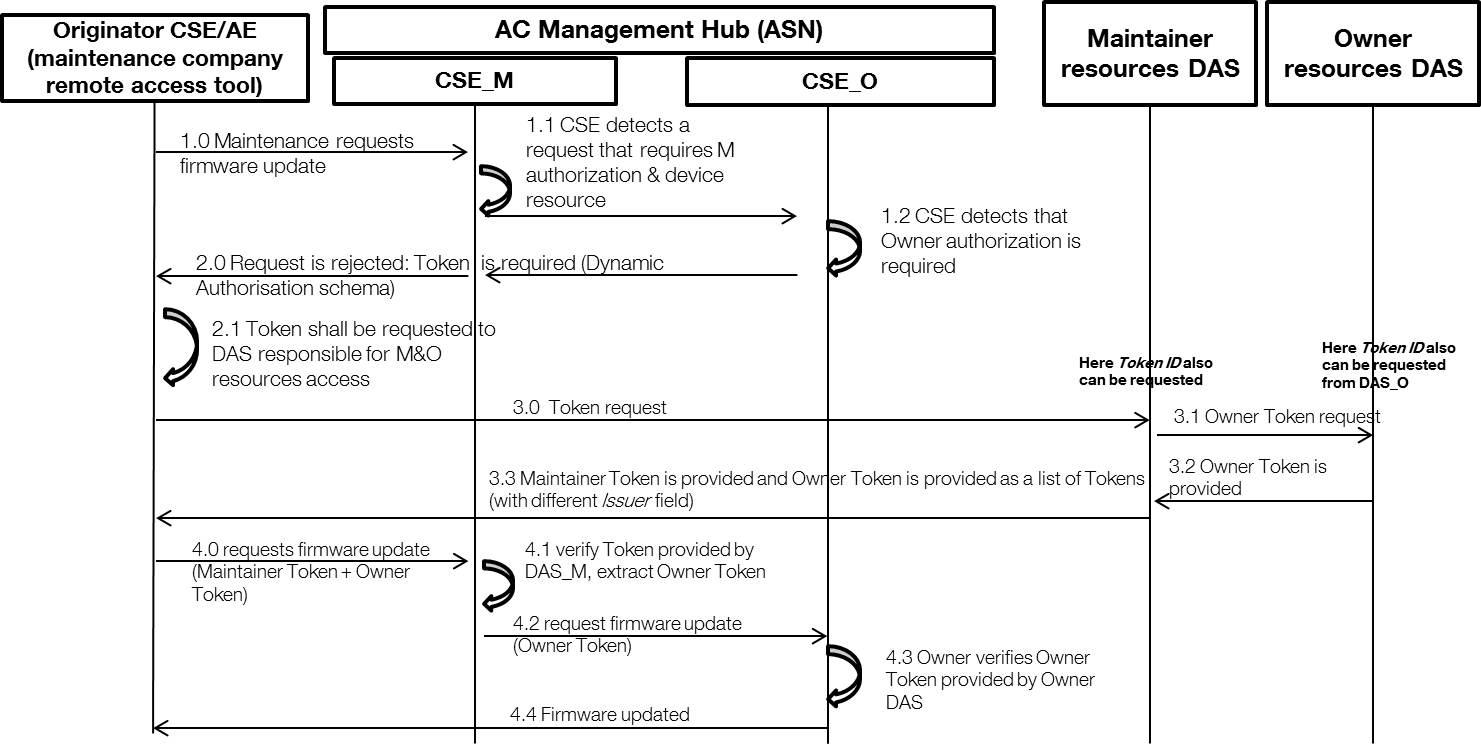


Figure 7: Case 4 Call Flow

-----------------------Start of TS-0003 change 1-------------------------------------------

#### 7.3.2.2 Direct Dynamic Authorization

The present document specifies the exchanged parameters and associated processing at the Hosting CSE. The transport of parameters is specified in clause 11.5.2, oneM2M TS-0001 [1].

The message flow for the Direct Dynamic Authorization is shown in figure 7.3.2.2-1, and described in the following text.



Figure 7.3.2.2-1: Message flow for Direct Dynamic Authorization

1. The Originator sends request (called the request from the Originator for this message flow) to the Hosting CSE. This request may include *Tokens* or *Token-IDs*; see the clause 7.3.2.3 "Indirect Dynamic Authorization".

2. Initial Hosting CSE processing:

2.1 If the request from the Originator includes ***Tokens*** or ***Token-IDs*** then these are processed as described in clause 7.3.2.3 "Indirect Dynamic Authorization". The Hosting CSE evaluates the access decision algorithm, but is unable to grant access for the request from the Originator based on configured access control policies.

2.2 The Hosting HCSE determines the set of DAS Server with which Direct Dynamic Authorization may be performed:

2.2.1 The HCSE examines all *accessControlRules* for which request satisfies the *accessControlOperations* and *accessControlContexts* in the <*accessControlPolicy*> resources linked to the requested resource. The HCSEcollects the set of all *Role-IDs* in the *accessControlOperators* ofthese *accessControlRules.* This *Role-IDs* are grouped according to the DAS Server AE-ID identified by the *Role-ID*.

NOTE 1: Regarding the Role-ID(s) parameter: The Originator would be granted access if a Token(s) is issued which associates the Originator with one or more of the Role-ID(s). Providing this list to the DAS Server allows the DAS Server to select a suitable set of one or more Role-ID(s) to associate with the Originator in Token(s), thereby authorizing the Originator to access the requested resources. The policies configured to the DAS Server would dictate which Role-ID(s) (if any) are included in Token(s) issued to the Originator.

2.2.2 The HCSE shall also collect the set of <*dynamicAuthorizationConsultation*> resources linked to the requested resource, and group these according to the DAS Server's dynamicAuthorizationPoA attribute of the <*dynamicAuthorizationConsultation*> resource.

2.3 The Hosting CSE selects a DAS Server (from the set determined in step 2.2) and sends a oneM2M request message containing the information described in table 7.3.2.2-1. The transport of parameters is specified in step 2.3, clause 11.5.2, oneM2M TS-0001 [1].

Table 7.3.2.2-1: Information sent from Hosting CSE to DAS Server  
during the Direct Dynamic Authorization

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Mandatory/Optional |
| Originator | Identifier of the Originator of the request received by the Receiver | M |
| Originator Resource Type | Type of resource targeted by originated request received by Receiver | M |
| Operation | Type of operation specified in originated request received by the Receiver | M |
| Originator IP Address | IP address of Originator of request received by Receiver | O |
| Originator Location | Location of Originator of request received by Receiver | O |
| Originator Role IDs | Role IDs of Originator of request received by Receiver | O |
| Request Timestamp | Timestamp when originated request was received by Receiver | O |
| Targeted Resource ID | Resource ID targeted by originated request received by Receiver | O |
| Proposed Privileges Lifetime | Proposed lifetime of authorization privileges requested by the Receiver | O |
| Role IDs From ACPs | The set of Dynamic Access Roles in the *accessControlDynAuthRole* parameters associated with the DAS Server AE-ID. | O |
| Token IDs | The set of token identifiers associated with the Originator | O |
| AuthorSignIndicator | An indicator included in the request received by Receiver to indicate the capability to sign for creating AuthorRelMapRecord when Originator is an AE. It is used in the case that the AE-ID-Stem is assigned by the Registrar CSE of the AE so that the AE-ID may change in a new registration (see clause 7.3.2.7.1). If the Hosting CSE does not support this parameter, then the Hosting CSE shall ignore it. | O |

3. DAS Server processing:

3.1 The DAS Server processes the received parameters. The DAS Server may decide to provide *Token(s)* and/or *dynamicACPInfo* which will be used by the Hosting CSE to create a dynamic <*accessControlPolicy*> resource. The DAS Server applies the policies with which it is configured to decide on the appropriate actions.

NOTE 2: The details of this decision are specific to the Dynamic Authorization System being employed; these details are not visible to the oneM2M system, and are not addressed in the present document.

NOTE 3 : DAS server selects suitable Token authorizing the Originator to access the requested resource. However, it is possible that Owner/Maintainer multi-tenants isolation is used in ASN (Access Service Node) with different access rights policies per tenant and different DAS servers per tenant. In such case DAS Owner server will manage requesting a second token from DAS Maintaner. DAS Owner will be in charge to return as list of Tokens to the Hosting CSE :

* + - DAS Owner token(s) authorizing Owner tenants resources access
    - and DAS Maintainer token(s) authorizing Maintainer tenants resources access .

*Issuer* field inside the Token structure (clause 7.3.3.1 "Token Structure") will indicate which DAS provided which token.

The Token(s) (if any) shall conform to clause 7.3.3.1 "Token Structure", with the following profile:

* The "holder" parameter shall contain the Originator's Absolute CSE-ID or AE-ID received from the HCSE, and may contain other CSE-IDs and AE-IDS.
* The "audience" parameter shall contain only the HCSEs CSE-ID.

The DAS Server shall apply a ESData protection option to the individual Tokens with the following requirements

* The DAS Server may encrypt the Token such that the Token can be decrypted by the Hosting CSE.
* The Hosting CSE shall be able to verify that the DAS Server issued the token.

The ESData processing results in an ESData envelope which is called the *ESData-protected Token* for the purposes of this message flow.

If the DAS Server decides to authorize the Hosting CSE to create a dynamic <*accessControlPolicy*> resource, then the DAS Server shall form a *dynamicACPInfo* parameter containing the following information are listed in table 7.3.2.2-2.

Table 7.3.2.2-2: Information included in the *dynamicACPInfo* parameter

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Mandatory/Optional |
| Granted Privileges | List of granted privileges | O |
| Privileges Lifetime | Lifetime of granted privileges | O |
| Tokens | List of issued tokens | O |

3.2 The DAS Server shall send the ESData-protected *Token(s)* (if any) and (optional) *dynamicACPInfo* parameter via the DAS Server AE to the Hosting CSE. The transport of parameters is specified in step 2.3, clause 11.5.2, oneM2M TS-0001 [1]. If the DAS Server receives the *AuthorSignIndicator* from the Hosting CSE and the DAS server itself also supports to trigger creating the authorization relationship mapping record, then the DAS Server shall send an *AuthorSignReqInfo* to the Hosting CSE to request the AE to create the authorization relationship mapping record.

4. HCSE Processing:

4.1 The HCSE processes the ESData-protected *Token(s)* (if present) and *dynamicACPInfo* parameter (if present):

4.1.1 The HCSE shall perform the following verifications for each ESData-protected *Token*:

4.1.1.1 The HCSE shall apply ESData processing to the ESData-protected *Token* to extract the authenticated Token.

4.1.1.2 The HCSE shall perform the following verifications:

4.1.1.2.1 The "issuer" parameter in the Token shall exactly match the identity of the DAS Server.

4.1.1.2.2 The HCSE's CSE-ID shall match the CSE-ID in the "audience" parameter in the Token.

4.1.1.2.3 The "holder" parameter in the Token shall exactly matches the Absolute CSE-ID or AE-ID of the Originator from whom the request was received.

4.1.1.2.4 The HCSE shall verify that the Token has not expired, by comparing the current time to the "notAfter" parameter in the Token.

4.1.1.3 The HCSE shall cache the verified Token, and may later delete the verified Token when the Token expires (as defined in step 4.1.2.4). If the Hosting CSE receives a *AuthorSignReqInfo* from DAS Server AE, then the Hosting CSE must make sure the Absolute AE-ID of the Originator shall be assigned to the *holder* attribute of the cached token.

4.1.2 If *dynamicACPInfo* is provided by the DAS Server, then the Hosting CSE shall create a dynamic <*accessControlPolicy*> resource matching the *dynamicACPInfo*.

4.2 The Hosting CSE repeats the access decision mechanism in clause 7.1.4 "Access Control Decision".

4.3 If access is granted, then the Hosting CSE performs the operation requested in the request from the Originator, resulting in the Hosting CSE sending a request to the Originator.

5. The Hosting CSE shall send a response message containing the ESData-protected Token(s) (if present) or TokenID(s), and *AuthorSignReqInfo* if the Hosting CSE receives and supports the *AuthorSignReqInfo* from DAS Server AE. If the *AuthorSignReqInfo* is not included in the response, then the steps 6-8 will not be applied.

6. Originator processing:

6.1 If the Originator receives AuthorSignReqInfo, then the Originator shall generate AuthorSign(s) on Token(s) or TokenID(s) for each Token.

NOTE 4: AuthorSign is a signature generated using the certificate of the AE or a MIC generated using a symmetric key shared between the AE and Hosting CSE. How a symmetric key is distributed to the AE and DAS server is not specified in this document.

NOTE 5: If the Originator includes the AuthorSignIndicator in step 1, but there is no AuthorSignReqInfo included in the response in step 5, then it indicates that the Hosting CSE or the DAS server doesn’t support creating the authorization relationship mapping record.

6.2 The Originator sends the AuthorSign(s) with the corresponding Token(s) or TokenID(s) to the Hosting CSE.

7. The Hosting CSE forwards the parameters from the Originator to the DAS server AE.

8. DAS server AE shall create AuthorRelMapRecord(s) containing the following information listed in table 7.3.2.2-3 for each Token:

Table 7.3.2.2-3: Information included in the AuthorRelMapRecord

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Mandatory/Optional |
| SubjectID | Absolute AE-ID of the AE | O |
| Token | The token issued for the AE | M |
| SignatureAuthorSign | Generated from Token or TokenID | M |
| ResourceID | The resource ID of the resource AE requests to access | O |

-----------------------End of TS-0003 change 1-------------------------------------------

-----------------------Start of TS-0003 change 2-------------------------------------------

#### 7.3.2.3 Indirect Dynamic Authorization

The present document specifies the exchanged parameters and associated processing at the Originator and Hosting CSE. The transport of parameters is specified in clause 11.5.3 of oneM2M TS-0001 [1].

The message flow for Indirect Dynamic Authorization is shown in figure 7.3.2.3-1, and described in the following text.



Figure 7.3.2.3-1: Message flow for Indirect Dynamic Authorization

1. (Optional) The Originator sends request to the Hosting CSE. The Originator includes an indication that the Originator is prepared to request Tokens from DAS Servers for this request. This request may include a combination of *Tokens*, *tokenIDs, Local-Token-IDs* but this message flow assumes that these do not provide sufficient permissions for accessing the requested resource.

2. (Optional) Initial Hosting CSE processing:

2.1 Hosting CSE performs the access decision for the request from the Originator. This call flow assumes that the request from the Originator is denied as a result of the access decision. The Hosting CSE observes the indication that the Originator prepared to request Tokens from DAS Servers for this request.

2.2 The Hosting CSE forms a list of DAS Server's and associated Role-ID(s) (if any) as described in step 2.2.1 of the Direct Dynamic Authorization in clause 7.3.2.3 "Direct Dynamic Authorization".

For each DAS Server, then Hosting CSE may apply ESData to the set of Role-IDs for decryption by the DAS Server. For example, the ESData may encrypt the set of Role-IDs so they are not visible to the Originator.

2.3 The Hosting CSE shall send an unsuccessful response to the Originator, including the list of DAS Servers and associated set of optionally-ESData-protected Role-IDs.

2.4 The Originator selects a DAS Server identified in the response.

3. The Originator shall interact with the DAS Server to request the issuance of a *Token*. The Originator can provide the optionally-ESData-protected set of Role-IDS to the DAS Server, and parameters from the original resource access request. If the Originator is an AE and the AE-ID-Stem is assigned by the Registrar CSE of the AE, and the Originator supports to create the authorization relationship mapping record, then the Originator shall provide the *AuthorSignIndicator* parameter in order to ask the DAS server to maintain the authorization relationship (see clause 7.3.2.7.2) in case the AE-ID of the Originator may change in a new registration. If the set of Role-IDS is protected using ESData, the DAS Server applies ESData to extract the set of Role-IDS. The DAS Server issues a Token(s) and provides the tokenID(s) and optionally the ESData-protected Token(s) to the Originator. The DAS Server can also provide the Originator with other parameters from the Token; for example, the time window in which the Token is valid. If the DAS Server receives the *AuthorSignIndicator* from the Originator, and the DAS server supports creating the authorization relationship mapping record, then the DAS server shall provide the Originator with a *AuthorSignReqInfo* to request the Originator to return *AuthorSign(s)* for each Token. This interaction is specific to the Dynamic Authorization System technology being used.

NOTE 1 : DAS server selects suitable Token authorizing the Originator to access the requested resource. However, it is possible that Owner/Maintainer multi-tenants isolation is used in ASN (Access Service Node) with different access rights policies per tenant and different DAS servers per tenant. In such case DAS Owner server will manage requesting a second token from DAS Maintaner. DAS Owner will be in charge to return as list of Tokens to the Hosting CSE :

* + - DAS Owner token(s) authorizing Owner tenants resources access
    - and DAS Maintainer token(s) authorizing Maintainer tenants resources access .

*Issuer* field inside the Token structure (clause 7.3.3.1 "Token Structure") will indicate which DAS provided which token.

4. If the Originator receives an *AuthorSignReqInfo* from DAS server, then the Originator shall return the AuthorSign(s) to DAS server:

4.1 The Originator generates *AuthorSign(s)* on Token(s) or TokenID(s) for each Token.

NOTE 2: AuthorSign are a signature generated using the certificate of the AE or a MIC generated using a symmetric key shared between the AE and DAS server. How a symmetric key is distributed to AE and DAS server is not specified in this document.

4.2 The Originator sends the AuthorSign(s) to DAS server with the corresponding Token(s) or TokenID(s).

4.3 The DAS server shall create AuthorRelMapRecord(s) containing the information listed in table 7.3.2.2-3 for each Token.

5. For request that the Originator wishes to have authorized using an issued Token, the Originator shall add ESData-protected Token provided by the DAS Server or *tokenID* (if no ESData-protected Token was provided) if the corresponding ESData-protected Token(s) was not provided by the DAS Server. In particular, if the request at step 1 was unsuccessful at step 2.3, then the Originator may repeat the request with new *Token(s)* and/or *tokenID(s)*. A token may be used in multiple request. If step 4 is performed, then the request shall contain the *AuthorRelIndicator* to indicate to the Hosting CSE that the relationship between the AE and the Token(s) are maintained in the DAS server.

The Originator shall send the request to the Hosting CSE.

6. (Optional) If the request includes *tokenID(s)*, then for each *tokenID* the Hosting CSE identifies the corresponding DAS Server AE from which to request the corresponding Token:

6.1 The Hosting CSE sends the tokenID(s) to the DAS Server via a DAS Server AE.

6.2 The DAS Server shall return the corresponding valid ESData-protected Token(s) to the Hosting CSE via the DAS Server AE.

7. Hosting CSE Processing:

7.1 Token Processing:

7.1.1 The Hosting CSE shall apply ESData to the ESData-protected Token(s), either provided in the request or retrieved from the DAS Server, to extract the authenticated Token(s).

7.1.2 If a Local-Token-ID was provided in the request, then the Hosting CSE attempts to retrieve the cached token.

7.1.3 The HCSE shall perform the following verifications for each authenticated and cached token associated with the request:

- The HCSE's CSE-ID shall match one of the Absolute CSE-IDs (optionally including wildcards) in the "audience" parameter in the Token.

- The "holder" parameter in the Token shall exactly match the Absolute CSE-ID or AE-ID of the Originator from whom the request was received.

- The HCSE shall verify that the Token is currently valid and not expired, by comparing the current time to the "notBefore" and "notAfter" parameter in the Token. If a cached Token has expired, then the Token may be removed from the cache.

7.1.4 If any identified Token could not be retrieved in steps 6 or 7.1.2, or if any ESData-protected Token-ID failed verification at step 7.1.1, or if any Token failed the verification at step 7.1.3, then the Hosting CSE shall respond with an error.

7.1.5 The Hosting CSE may cache any new Token(s). If the Hosting CSE receives the *AuthorRelIndicator* in step 5, then the Hosting CSE shall make sure the Absolute AE-ID of the Originatoris assigned to the *holder* attribute of the cached token.

7.2 The Hosting CSE may assign *Local-Token-ID(s)* to cached Token(s).

7.3 The Hosting CSE shall perform the access decision as described in clause 7.1.4, including the information in the Token(s) identified in the request. If access is granted, then the requested operation shall be performed.

8. Response:

8.1 The Hosting CSE sends a response to the Originator. For each new *Local-Token-ID*(s) that has been assigned, the Hosting CSE provides the *Local-Token-ID* and corresponding *tokenID* in the response parameters.

8.2 The Originator associates the *Local-Token-ID* with *tokenID*. In subsequent requests, the Originator may use the *Local-Token-ID* instead of the *Token* or *tokenID*.

-----------------------End of TS-0003 change 2-------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change? (Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.)
* Are multiple changes in this CR clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.?