authorizationTokenManagement

Service Description

Abstract
This document provides service description for the **authorizationTokenManagement** service.

Contents

1 Overview

This document describes the **authorizationTokenManagement** service, which allows systems (with operator role or proper permission) to manage the service access tokens in bulk and on behalf of the consumer and provider systems. Access tokens enable the verification of service consumption permissions on the provider system side, and the application of session-based service consumption control between the consumer and provider systems. An example of this interaction when a Core/Support system generates tokens for a consumer system for multiple service instances.

The **authorizationTokenManagement** service contains the following operations:

- *generate-tokens* verifies the given consumer systems' permissions and produces the tokens of defined types for the targeted service instances;
- *query-tokens* lists the access tokens that match the filtering requirements;
- *revoke-tokens* remove access token records by token references;
- *add-encryption-keys* stores the defined encryption keys that can be used to encrypt the raw tokens generated for any service of the associated provider systems.
- *remove-encryption-keys* removes the encryption keys associated with the given provider system names.

The rest of this document is organized as follows. In Section 2, we describe the abstract message operations provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned operations.

1.1 How This Service Is Meant to Be Used

The purpose of this service is to handle the access tokens centrally and in bulk. This approach makes it possible for consumer systems to avoid generating tokens for multiple service instances individually. Also, provider systems don't have to register/unregister their token encryption keys on their own, it can be outsourced.

Several token variants could be available, that are gouped into the following types:

Simple tokens are not holding any kind of information. These can only be verified by consulting the system implementing the authorizationToken service.

Self-contained tokens are holding all the necessary information for the provider system to verify it independently. However, the way of accessing to the token payload could differ according to the exact self-contained token variant. Since these kinds of tokens are holding sensitive information, there is an option to encrypt the tokens using an encryption key associated with the provider.

1.2 Important Delimitations

The requester has to identify itself to use any of the operations.

1.3 Access policy

The service is only available for operators, dedicated Core/Support systems and those who have the proper authorization rights to consume it.

2 Service Operations

This section describes the abstract signatures of each operations of the service. In particular, each subsection names an operation, an input type and one or two output types (unsuccessful operations can return different structure), in that order. The input type is named inside parentheses, while the output type is preceded by a colon. If the operation has two output types, they are separated by a slash. Input and output types are only denoted when accepted or returned, respectively, by the operation in question. All abstract data types named in this section are defined in Section 3.

2.1 operation generate-tokens (AuthorizationTokenGenerationListMgmtRequest) : AuthorizationTokenMgmtListResponse / ErrorResponse

Operation *generate-tokens* verifies the given consumer systems' permissions to the targeted service/service-operation instances and produces expiring tokens of defined types associated with the consumer system, service provider system and service instance triplets. The operation returns the generated tokens and the belonged details.

2.2 operation query-tokens (AuthorizationTokenQueryRequest) : AuthorizationTokenMgmtListResponse / ErrorResponse

Operation *query-tokens* lists the access tokens that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- There is an AND relation between different kind of filters

2.3 operation revoke-tokens (AuthorizationTokenRemoveRequest) : OperationStatus / ErrorResponse

Operation *revoke-tokens* deletes the access token records associated with the given token references. In case of simple tokens this operation can close a session between a consumer and provider.

2.4 operation add-encryption-keys (AuthorizationEncryptionKeyMgmtListRequest) : AuthorizationEncryptionKeyMgmtListResponse / ErrorResponse

Operation *add-encryption-keys* saves and stores String key and encryption algorithm identifier pairs belonging to the given provider systems. If the given algorithm is using any addition (besides the encryption key) for the encryption process, such as salt or initialization vector for example, then this addition is returned to the requester system. Any time when a self-contained token is generated that is associated with a provider that has an encryption key saved, the token will be provided to the token requester encrypted, using the specified key and encryption algorithm.

2.5 operation remove-encryption-keys (AuthorizationEncryptionKeyRemoveMgmtRequest) : OperationStatus / ErrorResponse

Operation *remove-encryption-keys* deletes the stored encryption key and algorithm identifier pairs associated with the given provider systems. The result of this operation is that further tokens generated to any service of the affected provider systems, won't be encrypted.

3 Information Model

Here, all data objects that can be part of the **authorizationTokenManagement** service are listed and must be respected by the hosting system. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.15, which are used to represent things like hashes and identifiers.

3.1 struct AuthorizationTokenGenerationListMgmtRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
list	List<AuthorizationTokenGenerationMgtmRequest>	yes	List of token requests.

3.2 struct Identity

An Object which describes the identity of a system. It also contains whether the identified system has higher level administrative rights.

3.3 struct AuthorizationTokenGenerationMgtmRequest

Field	Type	Mandatory	Description
tokenVariant	AccessTokenVariant	yes	Exact type of token technology.
targetType	AccessTargetType	yes	Type of the targeted resource.
consumerCloud	CloudIdentifier	no	Cloud of the consumer.
consumer	SystemName	yes	Name of consumer.
provider	SystemName	yes	Name of the targeted provider system.
target	ServiceName / EventTypeName	yes	Target of the token.
scope	AccessTokenScope	no	Scope of the token. Only matters when the target is a service definition.
expiresAt	DateTime	no	Token will be valid until this timestamp. Only in case of time limited tokens. Default time limit is applied if not defined.
usageLimit	Number	no	How many times the token will be valid. Only in case of usage limited tokens. Default usage limit is applied if not defined.

3.4 struct AccessTokenScope

A String which specifies the scope of the token. It can be ServiceOperationName if the token target is a ServiceName and the token is limited to one particular service-operation or it can be empty if the token is not limited or the target is an EventTypeName.

3.5 struct AuthorizationTokenMgmtListResponse

Field	Type	Description
entries	List<AuthorizationTokenMgmtResponse>	List of token results.
count	Number	Number of returned results.

3.6 struct AuthorizationTokenMgmtResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
tokenType	AccessTokenType	Type of token technology group.
variant	AccesTokenVariant	Exact type of token technology.
token	AccessToken	The token itself.
tokenReference	AccessTokenReference	Reference of the token record.
requester	SystemName	Name of the system that requested the token.
consumerCloud	CloudIdentifier	Cloud of the consumer.
consumer	SystemName	Name of the consumer system.
provider	SystemName	Name of the provider system.
targetType	AccessTargetType	Type of the targeted resource.
target	ServiceName / EventTypeName	Target of the token.
scope	AccessTokenScope	Scope of the token.
createdAt	DateTime	Token was generated at this timestamp.
usageLimit	Number	Maximum number of token usage, if any.
usageLeft	Numer	The token can still be used this many times, if any.
expiresAt	DateTime	Token is valid until this timestamp, if any.

3.7 struct ErrorResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
errorMessage	String	Description of the error.
errorCode	Number	Numerical code of the error.
type	ErrorType	Type of the error.
origin	String	Origin of the error.

3.8 struct AuthorizationTokenQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.
pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
requester	SystemName	no	Requester is looking for tokens that were generated by the request of the specified system.
tokenType	AccessTokenType	no	Requester is looking for tokens that belong to the specified technology group.
consumerCloud	CloudIndetifier	no	Requester is looking for tokens that belong to the specified consumer cloud.
consumer	SystemName	no	Requester is looking for tokens that belong to the specified consumer system.
provider	SystemName	no	Requester is looking for tokens that belong to the specified provider system.
targetType	AccessTargetType	no	Requester is looking for tokens that belong to the specified target type.
target	ServiceName / EventTypeName	no	Requester is looking for tokens that belong to the specified service definition or event type.

3.9 struct AuthorizationTokenRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
list	List<AccessTokenReference>	yes	List of token references.

3.10 struct AuthorizationEncryptionKeyMgmtListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
list	List<AuthorizationEncryptionKeyMgmtRequest>	yes	List of encryption key requests.

3.11 struct AuthorizationEncryptionKeyMgmtRequest

Field	Type	Mandatory	Description
systemName	SystemName	yes	Name of the associated system.
key	String	yes	A secret key.
algorithm	EncryptionAlgorithmName	yes	A specific algorithm.

3.12 struct AuthorizationEncryptionKeyMgmtListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<AuthorizationEncryptionKeyMgmtResponse>	Result entries.
count	Number	Numer of the result entries.

3.13 struct AuthorizationEncryptionKeyMgmtResponse

Field	Type	Description
systemName	SystemName	Name of the associated system.
rawKey	String	The raw string key.
algorithm	EncryptionAlgorithmName	Name of the ecnryption algorithm.
keyAdditive	String	Any string addition that the defined algorithm is using, if any.
createdAt	DateTime	The encryprtion key was registered at this timestamp.

3.14 struct AuthorizationEncryptionKeyRemoveMgmtRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
list	List<SystemName>	yes	System name list of associated keys to be deleted.

3.15 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
AccessTargetType	A string reference that specifies the type of the targeted resource.
AccessToken	A possibly unique string of characters that is issued for a beneficiary system and is associated at least with a provider system, a target and is expiring.
AccessTokenReference	A string reference that is associated with the access token record.
AccessTokenType	A string reference that specifies a token technology group.
AccessTokenVariant	A string reference that specifies an exact token technology variant.
CloudIdentifier	A composite string identifier that is intended to be both human and machine-readable. It consists of the cloud name and the organization name that is managing the cloud. Each part must follow the PascalCase naming convention.
DateTime	Pinpoints a specific moment in time.
Direction	The direction of a sorting operation. Possible values are the representation of ascending or descending order.
ErrorType	Any suitable type chosen by the implementor of service.
EncryptionAlgorithmName	A string identifier that belongs to an encryption algorithm.
EventTypeName	A string identifier that is intended to be both human and machine-readable. Must follow camelCase naming convention.
List<A>	An *array* of a known number of items, each having type A.
Number	Decimal number.
Object	Set of primitives and possible further objects.
OperationStatus	Logical, textual or numerical value that indicates whether an operation is a success or a failure. Multiple values can be used for success and error cases to give additional information about the nature of the result.
ServiceName	A string identifier that is intended to be both human and machine-readable. Must follow camelCase naming convention.
ServiceOperationName	A string identifier that is intended to be both human and machine-readable. Must follow kebab-case naming convention.
String	A chain of characters.
SystemName	A string identifier that is intended to be both human and machine-readable. Must follow PascalCase naming convention.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.0.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.0.0	