identity

Service Description

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

Contents

1 Overview

This document describes the **identity** service, which enables both application and core/support systems to get and release a proof of identity token which also can be verified. Furthermore, it also allows to a system to change its own credentials. A provider system for this service is necessary if other core/support system are using *outsourced* authentication policy. An example of this interaction when an application system get a proof of identity before registering itself into the Service Registry. An other example when the Service Registry verifies the proof of identity before accepting that registration request. To enable other systems to use, to consume it, this service needs to be offered through the Service Registry.

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

- *login* acquires a proof of identity token;
- *logout* invalidates a proof of identity token;
- *change* changes the credentials;
- *verify* checks the validity of a provided token and acquire information about the verified system.

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

Systems should call the *login* operation to acquire a proof of identity token. To get the necessary information about the *identity* service, systems can lookup for it in the Service Registry without any authentication. When a system wants to use a service of the core/support systems it has to include its token to the request. Core/support system should use the *verify* operation to check the validation of the identity token and to acquire information about the requester before serving the response. Systems should call the *logout* operation when finishing their work.

1.2 Important Delimitations

To avoid possible identity theft, it is important that systems only share their identity tokens with trusted systems: these are the core/support systems of the Arrowhead Local Cloud.

1.3 Access policy

Available for anyone within the local cloud.

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 login (IdentityRequest) : IdentityLoginResponse / ErrorResponse

The login data must meet the following criteria:

- System name is mandatory. System names are case insensitive and have to be unique within the Local Cloud.
- Credential map is mandatory. Since the operation can support various authentication methods (or more than one), only the implementation can specify what kind of credentials are needed.

2.2 operation logout (IdentityRequest) : OperationStatus / ErrorResponse

The logout data must meet the following criteria:

- It requires the same input data as *login* operation to avoid the possibility of session invalidation by an outside actor (without proper permissions).
- System name is mandatory. System names are case insensitive and have to be unique within the Local Cloud.
- Credential map is mandatory. Since the operation can support various authentication methods (or more than one), only the implementation can specify what kind of credentials are needed.

2.3 operation change (IdentityChangeRequest) : OperationStatus / ErrorResponse

The change data must meet the following criteria:

- With this operation a requester system can only change its own credentials and only after a successful authentication.
- This operation is not allows to change the assigned authentication method.
- System name is mandatory. System names are case insensitive and have to be unique within the Local Cloud.
- Credential map is mandatory. Since the operation can support various authentication methods (or more than one), only the implementation can specify what kind of credentials are needed.
- New credential map is mandatory. Since the operation can support various authentication methods (or more than one), only the implementation can specify what kind of credentials are needed (same requirements are applied than in case of the previous credential map).

2.4 operation verify (IdentityVerifyRequest) : IdentityVerifyResponse / ErrorResponse

The input data must meet the following criteria:

- The requester must have a valid identity to use this operation

3 Information Model

Here, all data objects that can be part of the **identity** 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 3.9, which are used to represent things like hashes and identifiers.

3.1 struct IdentityRequest

Field	Type	Mandatory	Description
systemName	Name	yes	The requester of the operation.
credentials	Credentials	yes	Credential information related to the system.

3.2 struct Credentials

An Object which maps String keys String values.

3.3 struct IdentityLoginResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
token	Identity	Proof of identity token that assigned to the requester system for a session.
expirationTime	DateTime	Token is valid until this time.

3.4 struct Identity

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

3.5 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.

3.6 struct IdentityChangeRequest

Field	Type	Mandatory	Description
authentication	Name	yes	The requester of the operation.
credentials	Credentials	yes	Credential information related to the system.
newCredentials	Credentials	yes	The new credential information that replace the current one.

3.7 struct IdentityVerifyRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
token	Identity	yes	The target identity token that the requester wants to verify.

3.8 struct IdentityVerifyResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
verified	Boolean	The result of the verification.
systemName	Name	The name of the verified system. Empty if verification was unsuccessful.
sysop	Name	A flag that determines whether the verified system has higher level administrative rights. Empty if verification was unsuccessful.
loginTime	DateTime	System was started their active session at this time. Empty if verification was unsuccessful.
expirationTime	DateTime	The verified token is valid until this time. Empty if verification was unsuccessful.

3.9 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
Boolean	One out of true or false.
DateTime	Pinpoints a specific moment in time.
ErrorType	Any suitable type chosen by the implementor of service.
Name	A string identifier that is intended to be both human and machine-readable.
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.
String	A chain of characters.

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	