Service Description (SD) Identity Provider

**Abstract**

This document describes an Arrowhead compliant Identity Provider service. An identity provider service is a service that is able to in a secure manner determine the identity of a caller. Another system may in turn establish a new session that allows the system to present its identity to another system. It does this by providing a proof of identity to the system it wants to identify itself to.

Table of Contents

[1. Service Description Overview 3](#_Toc116902998)

[1.1. Significant Prior Art 3](#_Toc116902999)

[1.2. How This Service Is Meant To Be Used 3](#_Toc116903000)

[1.3. Important delimitations and dependencies 3](#_Toc116903001)

[2. Service Interface 3](#_Toc116903002)

[2.1. Operation GetProofOfIdentity(): Identity 3](#_Toc116903003)

[2.2. Operation ValidateIdentity(Identity) 4](#_Toc116903004)

[2.3. Identity Provider Sequence 4](#_Toc116903005)

[3. Information Model 4](#_Toc116903006)

[3.1. Struct Identity 4](#_Toc116903007)

[4. Non-functional Requirements 4](#_Toc116903008)

[5. References 5](#_Toc116903009)

[6. Revision history 5](#_Toc116903010)

[6.1. Amendments 5](#_Toc116903011)

[6.2. Quality Assurance 5](#_Toc116903012)

1. Service Description Overview

This document describes an Arrowhead compliant generic identity provider service. An identity provider service is a service that is able to in a secure manner determine the identity of a caller. Another system may in turn establish a new session that allows the system to present its identity to another system. It does this by providing a proof of identity to the system it wants to identify itself to.

# Significant Prior Art

None defined.

# How This Service Is Meant To Be Used

The Identity Provider service is used to create and validate identification tokens for users and systems in an Arrowhead cloud.

# Important delimitations and dependencies

None defined.

1. Service Interface

The abstract authentication interface has two functions that are, for one authentication chain, typically called by two different systems wanting to exchange information between each other. The “get proof-of-identity” call is called by a system (System A) with the intent to identify itself to another system (System B). The system providing the authentication service (System C) will sign a proof-of-identity to the calling system (System A) that the system can later present.

When system A later calls system B it includes this proof-of-identify in the call and system B can then later call the system providing the authentication service (System C) with the “validate identity” function to validate that the proof is indeed valid and signed by system C.

# Operation GetProofOfIdentity(): Identity

The get proof-of-identity call is used to get the proof that can later be provided to another system to validate the caller’s identity. The providing system might require the consuming system to prove its identity before returning a signed proof-of-identity. This can be done for example, but not limited to, username and password or certificates.

# Operation ValidateIdentity(Identity)

The validate identity call is used to validate a proof-of-identity presented in a call by a calling system. An implementation of this service description may do the validation function in several steps to facilitate high availability of the authentication feature.

# Identity Provider Sequence

The following sequence diagram shows a complete authentication sequence including getting the proof-of-delivery and then later using the proof to make a call to another system.

En bild som visar bord

Automatiskt genererad beskrivning

1. Information Model

# Struct Identity

The proof-of-identity is an object containing a signed claim about an identity, together with information about who signed the claim, in order to help validate it. The proof-of-identity could for example, but not limited to, be a JSON Web Token (JWT) or a SAML Response.

1. Non-functional Requirements

No non-functional requirements have been defined.

1. References

No references have been identified.

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2022-10-17 | 0.1 | Initial draft of specification | Per Olofsson |

# Quality Assurance

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Date | Version | Approved by |
| 1 |  |  |  |
| 2 |  |  |  |