Authentication Core System

System Description

Abstract
This document provides system description for the **Authentication Core System**.

Contents

1 Overview

This document describes the Authentication core system, which exists to provide, manage and validate system identities within an Eclipse Arrowhead Local Cloud (LC).

The rest of this document is organized as follows. In Section 1.1, we reference major prior art capabilities of the system. In Section 1.2, we describe the intended usage of the system. In Section 1.3, we describe fundamental properties provided by the system. In Section 1.4, we describe delimitations of capabilities of the system. In Section 2, we describe the abstract services produced by the system. In Section 3, we describe the security capabilities of the system.

1.1 Significant Prior Art

The strong development on cloud technology and various requirements for digitisation and automation has led to the concept of Local Clouds (LC).

"The concept takes the view that specific geographically local automation tasks should be encapsulated and protected." [1]

One of the main building blocks when realizing such Local Cloud is the capability of authenticating all the systems when wanted to join to a given LC and verifying the system identities before enabling service sessions between them.

The previous versions of Arrowhead had no separated or outsourced authentication mechanisms, this task relied entirely on X.509 certificates. While it is working well and reliably, it might be too resource intensive for edge devices or it could be exaggerated for certain use cases. In order to fulfill the needs for various authentication mechanisms, Arrowhead 5.0 is introducing the Authentication Core System which enables to implement and deploy the preferred methods and utilize them in a standard manner.

1.2 How This System Is Meant to Be Used

The Authentication is a recommended core system of Eclipse Arrowhead LC and is responsible for the fundamental system identity control functionality by storing system related data in order to being able to verify the provided credentials and assign identity tokens to the systems. Also, this core system has the responsibility to verify the identity tokens upon requests.

IMPORTANT! Application systems should not share their identity tokens with other systems than the arrowhead core or support systems. For authenticated communication between application systems the authorization token should be used. A typical way of working from a consumer system perspective is described n Figure 1

- The Figure describes a use case with a dynamic orchestration from the view of the consumer.
- Besides the consumer, every interaction with the core systems from the provider side and even the internal communication between the core systems requires identity tokens, but these are omitted here for the sake of easier understanding.
- Interactions between the consumer and provider does not contain identity token in order to prevent any kind of identity theft. Identities should be only shared with the trusted core and support systems.

- The difference between the identity and authorization tokens is that while identity tokens identify a system only, the authentication tokens are representing a consumer-provider service (optionally operation) exchange session.
- Having an identity token is a must for all the systems, but requiring an authorization token is a matter of choice of the provider systems.

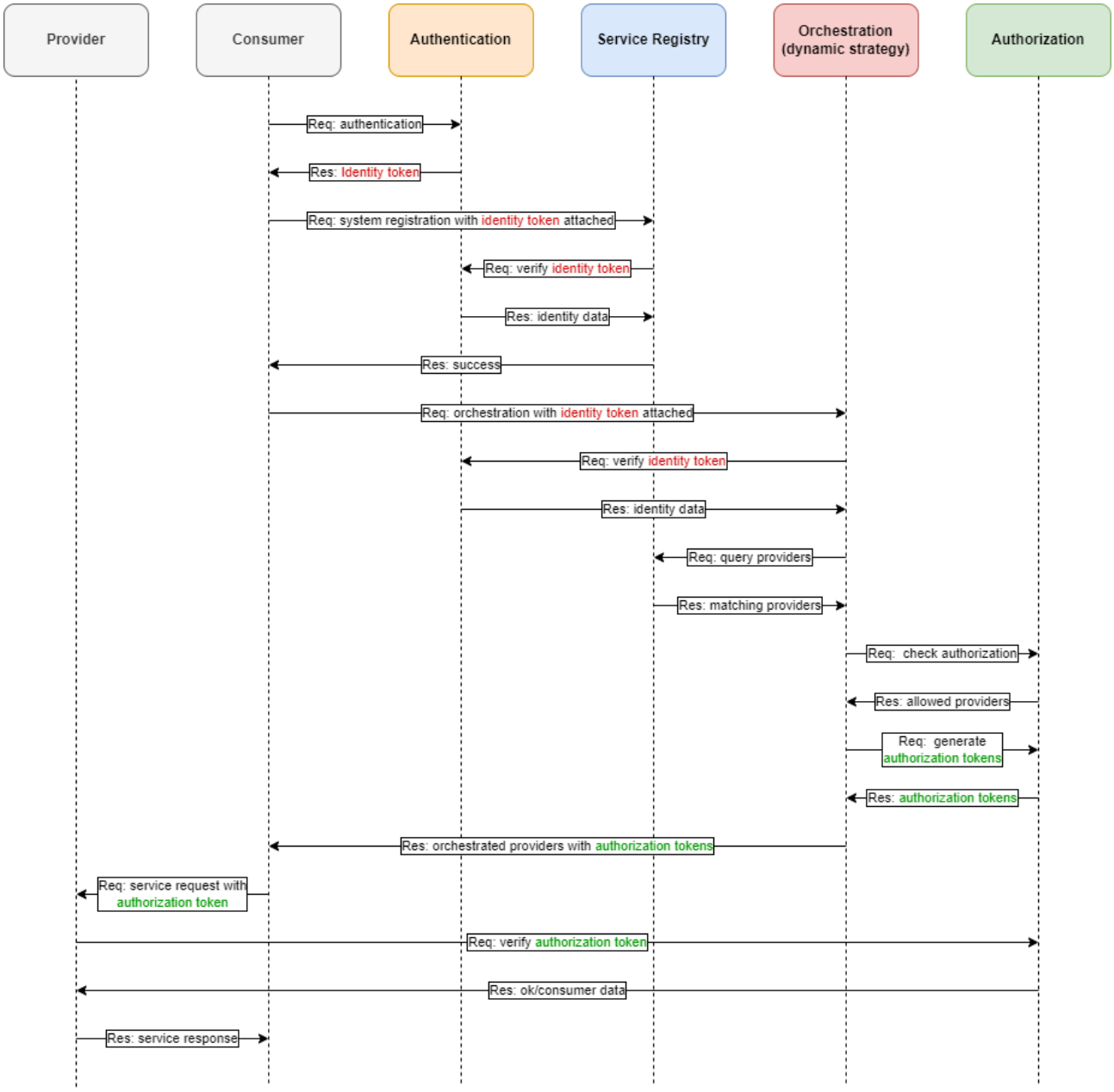

Figure 1: Interaction diagram of a typical use case.

1.3 System functionalities and properties

1.3.1 Functional properties of the system

Authentication solves the following needs to fulfill the requirements of authentication and system identity management.

- Enables the core, support and application systems to login/logout into/from the LC.
- Enables the core/support systems to validate that an other system is authenticated and already part of the LC.
- Enables the core/support systems to gather identity related data about a requesting system.
- Enabled the core/support systems with management rights to add, query and revoke system credentials.
- Enabled the core/support systems with management rights to query and close active login sessions.
- Enables the cloud operators to implement different kind of authentication methods.

1.3.2 Non functional properties of the system

-

1.3.3 Data stored by the system

In order to achieve the mentioned functionalities, Authentication is capable to store the following information set:

- **System authentication method**: What type of authentication method is assigned to a system by system name.
- **System authentication info**: What is required to verify upon a login request by system name and whether the system has access rights to the management services by default or not.
- **Identity tokens**: the assigned identity tokens by system-name and with the token properties like start date, expiration date, etc... .

1.4 Important Delimitations

-

2 Services produced

2.1 service identity

The purpose of this service is to enable the core/support and application systems to login/logout into/out from the Local Cloud and also to enable core/support systems to verify a requester application system's identity.

2.2 service identity-management

Its main purpose is to manage the system credentials and the active sessions in bulk. It also provides querying functionalities. The service is offered for core and administrative support systems.

2.3 service monitor

Recommended service. Its purpose is to give information about the Authentication system itself. The service is offered for both application and core/support systems.

3 Security

The actual implementation of the Authentication Core System can decide about the encryption of the connection between itself and other systems.

4 References

[1] J. Delsing and P. Varga, *Local automation clouds*. Boca Raton: Taylor & Francis Group, 2017, p. 28. [Online]. Available: https://doi.org/10.1201/9781315367897

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	