Document title		Document type
SimpleStoreServiceOrchestration Core System		**SysD**
Date		Version
2025-12-01		**5.2.0**
Author		Status
Katinka Jakó		**DRAFT**
Contact		Page
jako.katinka@aitia.ai		**1 (9)**

SimpleStoreServiceOrchestration Core System

System Description

Abstract

This document provides system description for the **SimpleStoreServiceOrchestration Core System**.

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
2 (9)

ARROWHEAD

Contents

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
3 (9)

1 Overview

This document describes the SimpleStoreServiceOrchestration Core System, which exists to find matching providers for the consumer's specification within an Eclipse Arrowhead Local Cloud (LC). This can be achieved by various strategies, but the SimpleStoreServiceOrchestration Core System implements the simple-store orchestration strategy. This recommended system uses predefined rules to find appropriate providers for the customer.

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]

A service orchestration system is a central component in any Service-Oriented Architecture (SOA). In applications, the use of SOA for a massive distributed System of Systems requires orchestration. It is utilised to dynamically allow the re-use of existing services and systems in order to create new services and functionality.

There are some key differences, even on conceptual level, between the previous versions (Orchestrator 4.6.x) and this version:

- The previous versions contained three (or two in earlier versions) kind of orchestration strategies: a store containing simple, peer-to-peer rules, another store containing more flexible rules and a dynamic orchestration method which used the live data of the Service Registry to achieve its goal. The current version separates the three strategies into three different systems (but with the same orchestration service operations), and the Local Cloud's administrator can decide which strategy to support for their use case.

- The previous versions were named the system as Orchestrator. Because this expression has a different meaning in some related domains, it is decided that the current version uses the name <Strategy>ServiceOrchestration to avoid confusion.

- There was no data storage separation requirement: the Orchestrator's data storage was interconnected to other systems' storage. In the current version, data storage separation is mandatory.

- Only the orchestration pull was supported in which the consumer could start an orchestration process for itself. The current version also supports orchestration push: the consumers can subscribe to a service orchestration (or another Support/Application system can subscribe them) and after the subscription and whenever a system notifies the SimpleStoreServiceOrchestration system, it performs the orchestration for the related subscribers.

- X.509 certificate trust chains was used as authentication mechanism. The current version can support any type of authentication methods by using a dedicated Authentication Core system.

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
4 (9)

1.2   How This System Is Meant to Be Used

SimpleStoreServiceOrchestration is a recommended core system of Eclipse Arrowhead Local Cloud and is responsible for finding and pairing service consumers and providers.

There are two ways to use the offered functionality:

- An application that wants to consume a service should ask the SimpleStoreServiceOrchestration to find one or more accessible providers that meet the necessary requirements. The SimpleStoreServiceOrchestration returns the information that the application needs to consume the specified service.

- An application can subscribe for orchestration with the necessary requirements (or can be subscribed by another Application/Support system). Whenever a system notifies the SimpleStoreServiceOrchestration system, it does the orchestration and sends the orchestration information to the subscriber on the specified channel. Optionally, subscribers can ask that an orchestration process run and return immediately after the subscription.

The *information* that is provided to the consumer whenever an orchestration is done depends on the orchestration strategy. In case of simple-store orchestration strategy, the SimpleStoreServiceOrchestration relies on the predefined rules that are stored in its database. Since this system does not contact the ServiceRegistry or any other Core/Support system, the orchestration response does not contain all the information that is needed for the consumer to perform a service operation consumption (access details, authorization tokens, etc...). It is the consumer's responsibility to collect the additional information.

1.3   System functionalities and properties

1.3.1   Functional properties of the system

SimpleStoreServiceOrchestration solves the following needs to fulfill the requirements of orchestration.

- Enables the application and Core/Support systems to find the appropriate providers to consume their services.

- Enables the application and Core/Support systems to subscribe/unsubscribe to repeated orchestration (orchestration push).

- Enables the application and Core/Support systems with administrative rights to notify the ServiceOrchestration system to orchestrate for the related subscribers.

- Enables the application and Core/Support systems with administrative rights to subscribe/unsubscribe consumers to repeated orchestrations.

- Enables the application and Core/Support systems with administrative rights to manage the orchestration rules.

1.3.2   Data stored by the system

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

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
5 (9)

- **Orchestration store**: A storage to manage the predefined orchestration rules. It consists of the consumer system name, the service definition, the service instance identifier, the priority of the rule, the creator of the rule, the timestamp of the creation, the modifier of the rule and the timestamp of the last update.

- **Orchestration history**: A storage to save data about the performed orchestration processes. It consists of an orchestration type, a requester system name, a target system name, a service definition and a timestamp.

1.4   Important Delimitations

- If the Local Cloud does not contain an Authentication system or does not using X.509 certificates, there is no way for the SimpleStoreServiceOrchestration to verify the requester system. In that case, the SimpleStoreServiceOrchestration will consider the authentication data that comes from the requester as valid.

- An orchestration result does not contain enough information to consume a service instance. The consumer at least needs to contact the ServiceRegistry Core system using the service instance id to acquire the access information of the referenced service instance. If this instance requires authorization token as a security measure, the consumer also has to consult with the ConsumerAuthorization Core system to acquire one.

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
6 (9)

2 Services produced

2.1 service serviceOrchestration

The purpose of this service is to find information about providers that meet the requirements. It also provided subscription functionality to repeated orchestrations (orchestration push). The service is offered for both application and Core/Support systems.

2.2 service serviceOrchestrationPushManagement

The purpose of this service is to manage orchestration push subscriptions in bulk. It also allows to signal the SimpleStoreServiceOrchestration system to orchestrate for the related subscribers. The service is offered for Core/Support systems.

2.3 service serviceOrchestrationStoreManagement

The purpose of this service is to query, create, update and remove orchestration rules. The service is offered for Core/Support systems.

2.4 service serviceOrchestrationHistoryManagement

Recommended service. Its purpose is to give information about the orchestration processes performed by the system. The service is offered for Core/Support systems.

2.5 service monitor

Recommended service. Its purpose is to give information about the provider system. The service is offered for both application and Core/Support systems.

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
7 (9)

3 Security

For authentication, the SimpleStoreServiceOrchestration can utilize another Core system, the Authentication system's service to verify the identities of the requester systems. If no Authentication system is deployed into the Local Cloud, the SimpleStoreServiceOrchestration can accept X.509 certificates for identification. If none of these are available, the SimpleStoreServiceOrchestration trusts the requester system's self-provided identity.

For authorization, the system uses another Core System, the ConsumerAuthorization system to decide whether a consumer can use its services or not. If the ConsumerAuthorization Core System is not present in the Local Cloud, then the SimpleStoreServiceOrchestration may allow for anyone in the Local Cloud to use its services. The following service operations can always be used without any authorization rules:

- *serviceOrchestration* service's *pull* operation,

- *serviceOrchestration* service's *subscribe* operation,

- *serviceOrchestration* service's *unsubscribe* operation.

The implementation of the SimpleStoreServiceOrchestration can decide about the encryption of the connection between the SimpleStoreServiceOrchestration and other systems.

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
8 (9)

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

Document title
SimpleStoreServiceOrchestration Core System
Date
2025-12-01

Version
5.2.0
Status
DRAFT
Page
9 (9)

5   Revision History

5.1   Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.2.0		Xxx Yyy

5.2   Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.2.0	