## Introduction

This document proposes a naming scheme for two Eclipse Arrowhead Systems, answering to Question #4 of the Eclipse Roadmap agenda for June 27th “How do we name the Orchestration System (Core) and the Choreography System (Support) such that we avoid unnecessary confusion from the Microservices communities?”

To solve this matter several questions have been placed in ChatGPT. The objective is to obtain the name that best represents what the system does without biasing the response introducing the names used in this moment.

First, ChatGPT was queried about the Eclipse Arrowhead Orchestrator Core System (Section 2). Next, ChatGPT was queried about the Eclipse Arrowhead Choreography Support System (Section 3). To clarify the difference between Orchestration and Choreography ChatGPT was queried again (Section 4). Finally, the proposed naming alternatives are presented in Section 5.

## Eclipse Arrowhead Orchestrator System (Core)

Questión: How would you call a software system component that puts together a service provider and a consumer after checking with the authorization rules, stores access rules for specific cosumer-provider pairs and enables on the fly service matching?

ChatGPT

You could call such a software system component a "Service Mediator" or a "Service Broker." This component acts as an intermediary between service providers and consumers, handling the authorization and access control process while facilitating dynamic service matching.

The term "Service Mediator" implies its role in mediating interactions between the service providers and consumers, ensuring that the authorization rules are enforced and access is granted according to the specified rules. It also suggests the component's ability to handle different service providers and consumers seamlessly.

Alternatively, the term "Service Broker" also captures the essence of this component. It brokers the interaction between service providers and consumers, managing access rules for specific consumer-provider pairs and enabling on-the-fly service matching. The term "broker" often implies a middleman that facilitates transactions or interactions, which aligns well with the described functionality.

Ultimately, the choice between "Service Mediator" and "Service Broker" depends on the specific context and the terminology used within your system or industry.

## Eclipse Arrowhead Choreography System (Support) or Workflow Manager

Question: How would you call a software system component which coordinates sequence of tasks/services that compose a process or workflow?

ChatGPT

A common term for a software system component that coordinates a sequence of tasks or services to compose a process or workflow is a "Workflow Engine" or "Orchestration Engine."

A Workflow Engine is responsible for managing and executing the various tasks or services in a defined order or according to a specified workflow logic. It typically handles the coordination, scheduling, and execution of these tasks, ensuring that the workflow progresses smoothly and efficiently.

An Orchestration Engine, on the other hand, implies a higher level of coordination and management. It not only handles the sequencing of tasks but also manages the overall flow and interaction between multiple services or systems involved in the process. The Orchestration Engine acts as a conductor, ensuring that each component performs its designated task in harmony to achieve the desired outcome.

Both terms, Workflow Engine and Orchestration Engine, are often used interchangeably depending on the specific context and the level of coordination required in the system. The choice between the two may depend on the granularity of control needed or the complexity of the workflows involved in your software system.

## Difference between Orchestration and Choreography

Question: What is the difference between orchestration and choreography?

ChatGPT

Orchestration and choreography are two different approaches to coordinating and managing interactions between components in a distributed software system. Here's a brief explanation of the differences:

1. Orchestration: In an orchestration-based system, there is a central controller known as the "orchestrator" that actively manages and controls the flow of the system. The orchestrator dictates the sequence and coordination of activities among the various components or services involved. It is responsible for defining the overall workflow, making decisions, and directing the execution of tasks. The orchestrator typically has a global view of the system and its state, and it actively drives the interactions between the components.
2. Choreography: In a choreography-based system, coordination is distributed among the participating components or services. Instead of a central controller, each component or service knows its responsibilities and interacts with other components based on predefined rules and protocols. Each component is autonomous and communicates directly with other components to fulfill its role in the system. The coordination emerges from the interactions between the components, and there is no centralized controller governing the entire process.

To summarize, the key differences between orchestration and choreography are:

* Control: Orchestration has a centralized controller (the orchestrator) that actively manages and controls the flow, while choreography distributes the coordination among the components themselves.
* Communication: In orchestration, the components communicate with the orchestrator, which then directs the communication with other components. In choreography, the components communicate directly with each other.
* View and knowledge: The orchestrator in orchestration has a global view of the system and its state, while components in choreography have local knowledge and only know about their own responsibilities.

The choice between orchestration and choreography depends on the specific requirements and characteristics of the system. Orchestration is often preferred when there is a need for central control and coordination, whereas choreography is useful in decentralized systems where components are more autonomous and the interaction patterns are more dynamic.

## Naming Proposal

Considering the replies obtained from ChatGPT and the authors previous knowledge/understanding, this document presents the following naming scheme. The document proposes a naming scheme that migrates from the actual naming to the proposed one “slowly”, considering that many developers know the systems with the actual naming, and they need to adapt to the changes.

|  |  |  |
| --- | --- | --- |
| Actual Name | Adaptation Time Name | Final Name |
| Orchestrator System (Core) | Orchestrator- Service Mediator System (Core) | Service Mediator System (Core) |
| Choreography System (Support) | Choreography -Orchestrator System (Support) | Orchestrator System (Support) |