

Grid Architecture Component Class Specification

Component Class Designation: Middleware

Summary Overview

Middleware is software that sits between an operating system and applications that provides services to other software, in order to support activities such as data transmission and processing, bridging software functionality, and general cooperation among systems. Specifically, middleware plays a role in concurrent processing, load balancing and transaction management. It may also support connectivity to and security of back-end resources. This specification focuses on middleware that facilitates the Integration of software components and applications.

Definitions

Application (Software) – Any computer program designed to perform a set of functions that are beneficial to an end-user.

Service – A specific function, such as authentication or search, that is provided by an application.

Distributed System – A group of independent entities, such as computers or applications, connected by a network, which work together to accomplish a common goal by passing messages to each other.

Application Programming Interface (API) – A software library that provides functionality for building applications.

Component Class Description

Purpose and Function

The purpose of middleware is to connect software components or applications that were not originally specifically designed to interact with each other. Middleware functions as a mechanism for hiding the disparate details of applications, protocols, and other interfaces to support interoperability between applications and components. It is used to support collaboration among entities in a distributed system in a uniform and well-defined way. Middleware may use APIs provided by components to bridge their functionality into a useful system.

Key externally visible properties

N/A

Interface descriptions

Given that middleware is a broad term with many different forms, it is not useful to enumerate the myriad types of interfaces that middleware may expose or interact with. Middleware often uses message passing or invocation of API functions to transparently support collaboration between

applications, leaving components unaware of any change in their position in a data pipeline. Thus, message or object- oriented middleware may be thought of as software that conforms to the Interceptor [1] or Chain of Responsibility [2] design patterns.

Table of relevant standards

Standard Name	Description	Version/Date
CORBA	Common Object Request Broker Architecture	
MPI	Message Passing Interface	

List of Attached Documents

Document Name/File Name	Description	Version/Date

References

No.	Reference
1	Object Management Group (2012). About the Common Object Request Broker Architecture Specification Version 3.3, retrieved from https://www.omg.org/spec/CORBA/
2	Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading, Mass. Addison-Wesley.

Version History

Action	Date	Description	Author(s)
Created	2 November, 2018	Initial draft	B. Kelley