

4.2.1.3 Replaceability

[Return to Portability](#)

- ~~☒ [nick][✓ char, 2020-10-22]Char's review completed--~~

About

[Return to the Top](#)

Replaceability occurs when software components are developed using open, well written, standard specifications, usually captured as an [Application Programming Interface \(API\)](#). The standards can be either technical (i.e., developed by a [Standards Organization](#)) or *de facto* (i.e., developed by a for-profit or not-for-profit corporation). Replaceability is also a key factor in preventing [Vendor Lock-In](#).

Replaceability is not just about the ability to switch suppliers and avoid [Vendor Lock-In](#)¹⁾ of the components, it's also about managing risk to the target system. This is especially true because each component can have its own [Lifecycle](#) with its own [End-of-life \(EoL\)](#) timelines, independent of the target system. In addition to the components' lifecycle, many components are now [Open Source Software \(OSS\)](#), which can often have forks spawning newer and competing products with similar, but not identical APIs. A recent article describes the [Best Message Queue \(MQ\) Software of 2020](#). It describes 30 of the "top" [Message Queue\(MQ\) Message-Oriented Middleware \(MOM\)](#) software products.²⁾:

1. MuleSoft Anypoint Platform
2. IBM MQ
3. Azure Scheduler
4. Apache Kafka
5. Google Cloud Pub/Sub
6. Amazon MQ
7. RabbitMQ
8. Apache ActiveMQ
9. KubeMQ
10. IBM MQ on Cloud
11. Azure Queue Storage
12. Alibaba Message Queue
13. Alibaba Message Service
14. CloudAMQP
15. Intel MPI Library
16. ZeroMQ
17. Apache Qpid
18. Apache RocketMQ
19. IBM Compose for RabbitMQ
20. IronMQ
21. PubSub+
22. Red Hat AMQ

23. TIBCO Enterprise Message Service
24. Bottomline GTBridge
25. CloudMQTT
26. Compose Hosted RabbitMQ
27. deepestream.io
28. Enduro/Z
29. EnMasse
30. IBM Cloud Pak for Integration

Obviously, not all these products have the same API. They definitely do not have the same [wireproro](#), so it is very difficult to “mix and match” [Publishers](#) and [Subscribers](#). In a distributed system, this would require all the components within the system to upgrade at the same time, sometimes referred to as a “[Reboot the World Problem](#)”. And when the upgrades occur, each part of the system requires complete [Regression Testing](#) to make sure that all the parts of the system continue to operate and function according to the specifications.

Replaceability is closely related to [Adaptability](#) and the kinds of Installers [Architecture](#) and [Software Adaptability](#). Whenever there is an architecture or software adaptability issue, there is probably a Replaceability issue as well. Additionally, the concepts of [Installability](#) and Replaceability overlap. The harder a system or a product is to install, the greater the probability it will also be hard to replace.

Competitive products within the same domain are ideal candidates for Replaceability. However, replacing products should not just be driven by the cost of acquisition (i.e., purchase price) but also on the [Total Cost of Ownership \(TCO\)](#), which considers the cost throughout the lifecycle of the target system or component including maintenance (see [20_maintainability](#)). This cost can be tied to other tangential products such as debuggers, [performance](#) monitors, loggers, or any of the [Non-Functional Requirements](#).³⁾

DDS Specifics

[Return to the Top](#)

The [data_distribution_service_dds](#) ecosystem is built around the idea of technical standards. The technical standard specifications are open, transparent and readily available to all. The [Standards Developing Organization \(SDO\)](#) overseeing the [DDS Family of Standards](#) is the [omg](#). The OMG has strict policies and procedures to govern and maintain the specifications to ensure the standards remain open and vendor neutral. As a result, the standards cover a wide range of issues that allow for the products to be Replaceable. The standards cover:

- [01_core](#)
- [02_api](#)
- [03_ext](#)
- [04_gate](#)
- [05_wip](#)

These standards allow for components to be replaced at either the [Architecture](#) or [Software](#) level (see [Adaptability](#));

1)

Note: Vendors are not just proprietary corporations; Open Source projects produce and sell products also. The software might be “free”, but the producers are competitors that have the same drive to lock-in customers as the corporations

2)

Best Message Queue (MQ) Software, <https://www.g2.com/>,

3)

Portability Testing Guide with Practical Examples, Software Testing Help, 30 June 2020, Accessed 31 July 2020, <https://www.softwaretestinghelp.com/what-is-portability-testing/>

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc:10_portability:06_replace&rev=1605305471

Last update: **2020/11/13 17:11**

