

Middleware del Internet de las Cosas

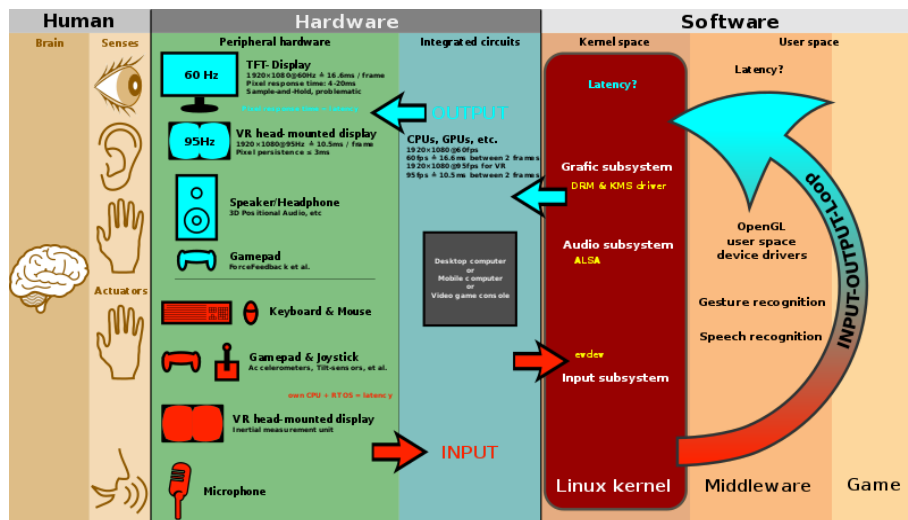
Juan Luis Sánchez Romero y Arturo Cortés Sánchez

Abstract. The abstract should summarize the contents of the paper in short terms, i.e. 150-250 words.

1 El middleware

1.1 Definición

El middleware es un software informático que proporciona servicios a aplicaciones de software más allá de las disponibles en el sistema operativo. Éste simplifica el trabajo de los programadores en la tarea de generar las conexiones y sincronizaciones que son necesarias en los sistemas distribuidos. Puede describirse como "pegamento software".



Todo lo que hay entre el kernel y las aplicaciones de usuario es considerado middleware.

El middleware facilita a los desarrolladores de software la implementación de la comunicación y la entrada/salida, funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores. El

middleware abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipos de servicios de middleware. Ganaron popularidad en la década de 1980 como solución al problema de cómo vincular las aplicaciones más nuevas a los sistemas heredados más antiguos, aunque el término había estado en uso desde 1968.

1.2 Middleware en sistemas distribuidos

El middleware en el contexto de las aplicaciones distribuidas es un software que proporciona servicios que van más allá de los que proporciona el sistema operativo para permitir que los diversos componentes de un sistema distribuido se comuniquen y gestionen datos. El middleware soporta y simplifica aplicaciones distribuidas complejas. Incluye servidores web, servidores de aplicaciones, mensajería y herramientas similares que soportan el desarrollo y la entrega de aplicaciones. El middleware es especialmente integral a la moderna tecnología de la información basada en XML, SOAP, servicios Web y arquitectura orientada a servicios.

El middleware a menudo permite la interoperabilidad entre aplicaciones que se ejecutan en diferentes sistemas operativos, suministrando servicios para que la aplicación pueda intercambiar datos de una manera basada en estándares. El middleware se sitúa "en el medio" entre el software de aplicación que puede estar trabajando en diferentes sistemas operativos. Es similar a la capa media de una arquitectura de sistema único de tres niveles, excepto que se extiende a través de múltiples sistemas o aplicaciones. Los ejemplos incluyen software de EAI, software de telecomunicaciones, monitores de transacciones y software de mensajería y cola.

La distinción entre la funcionalidad del sistema operativo y la del middleware es, hasta cierto punto, arbitraria. Mientras que la funcionalidad del núcleo sólo puede ser proporcionada por el propio sistema operativo, algunas de las funcionalidades que antes proporcionaba el middleware vendido por separado ahora están integradas en los sistemas operativos. Un ejemplo típico es la pila TCP/IP para telecomunicaciones, hoy en día incluida en prácticamente todos los sistemas operativos.

Sample Heading (Third Level). Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Forth Level). The contribution should contain no more than four levels of headings. The following Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	Lecture Notes	14 point, bold
1 st -level heading	1 Introduction	12 point, bold
2 nd -level heading	2.1 Printing Area	10 point, bold
3 rd -level heading	Run-in Heading in Bold. Text follows	10 point, bold
4 th -level heading	<i>Lowest Level Heading.</i> Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z \quad (1)$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

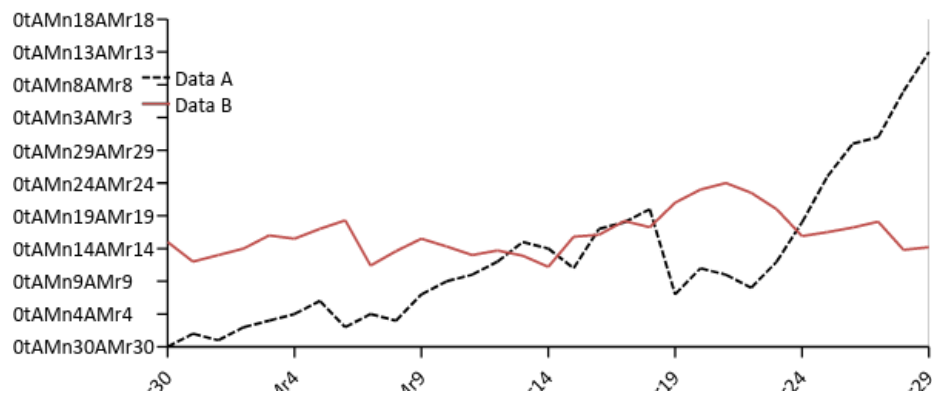


Fig. 1. A figure caption is always placed below the illustration. Short captions are centered, while long ones are justified. The macro button chooses the correct format automatically.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], as well as a URL [5].

References

1. Author, F.: Article title. Journal 2(5), 99–110 (2016).
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).

3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
4. Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
5. LNCS Homepage, <http://www.springer.com/lncs>, last accessed 2016/11/21.