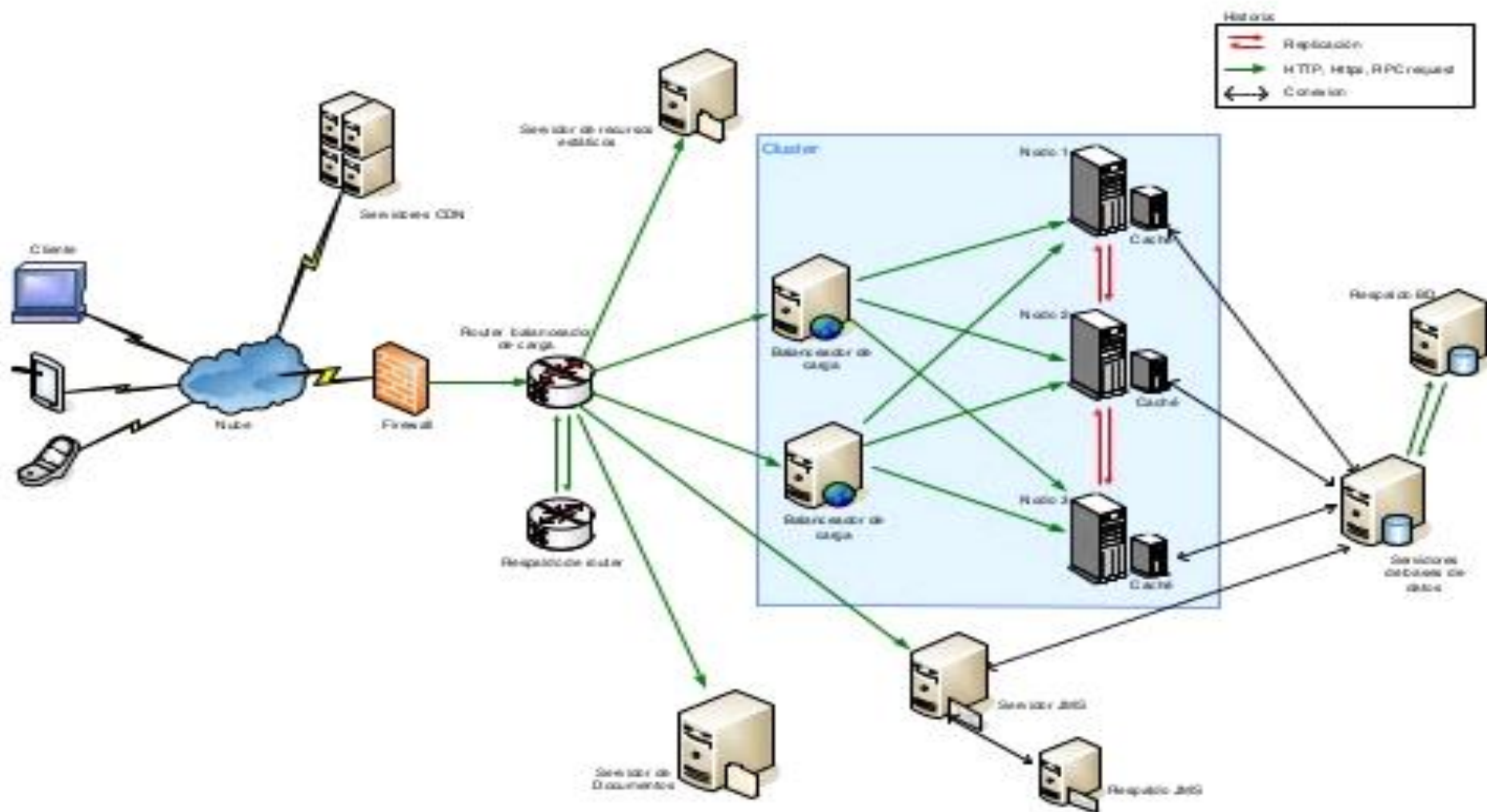


Arquitectura de aplicaciones



Arquitectura de aplicaciones

Los grandes productos a menudo se construyen en

- multi-tier architecture
- n-tier architecture

como a menudo le llaman.



Arquitectura física

La arquitectura física expresa cuáles son los componentes físicos (cliente, servidor, servidor web, BD, etc.) que participan en nuestra solución, así como la relación entre ellos.

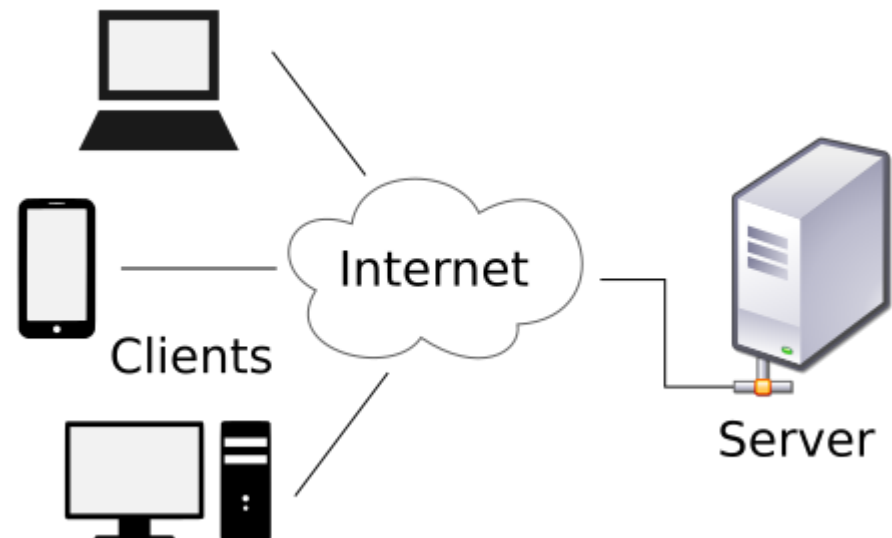
Arquitectura lógica

La arquitectura lógica expresa cuáles son los componentes lógicos (subsistemas, o macro-funciones) que participan en nuestra solución, y la relación entre ellos.

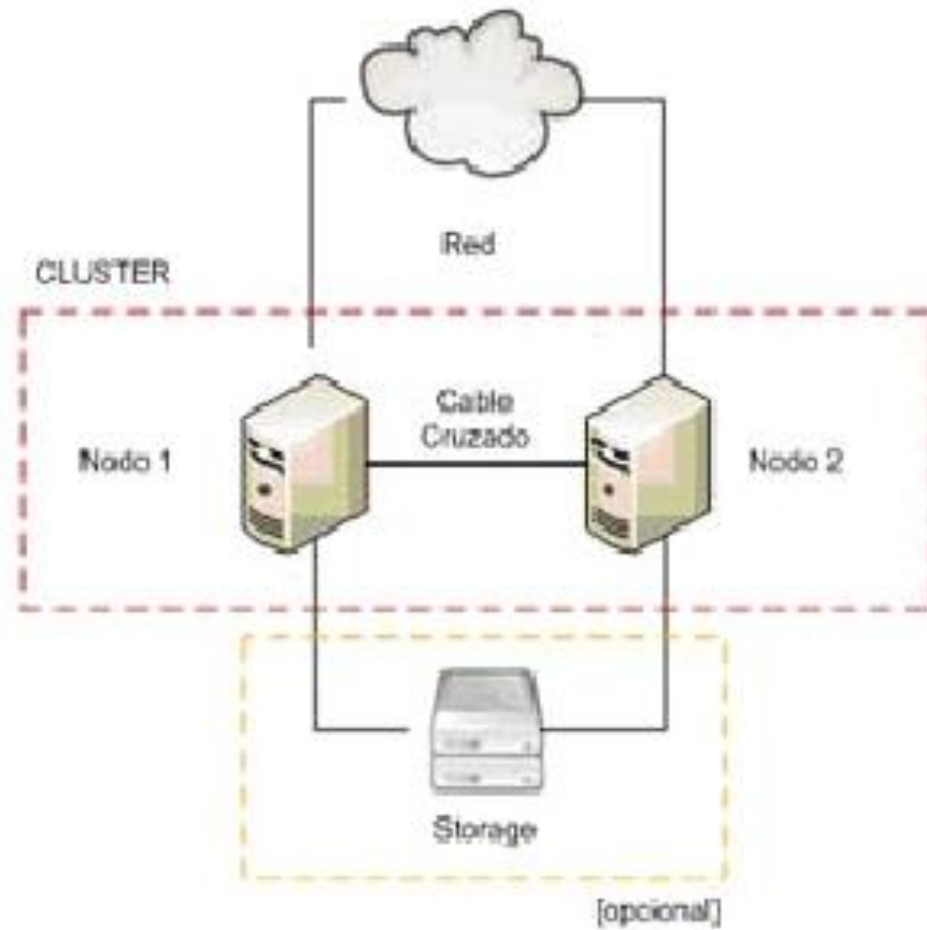
Ciente-servidor

Es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.



Cluster



N-Tier Architecture

1. El software está diseñado para tener las funciones de procesamiento, administración de datos y presentación separadas física y lógicamente.
2. Esto significa que estas diferentes funciones están alojadas en varias máquinas o clusters, lo que garantiza que los servicios se brinden sin compartir los recursos y, como tales, estos servicios se entregan a su máxima capacidad

N-Tier Architecture

1. No solo gana el software al poder obtener servicios a la mejor velocidad posible, sino que también es más fácil de administrar.
2. Esto se debe a que cuando trabajas en una sección, los cambios que realizas no afectarán a las otras funciones. Y si hay un problema, puede identificar fácilmente dónde se origina.
3. El uso más extendido de la arquitectura multitier es la arquitectura de tres niveles.

N-Tier Architecture

Se divide en tres niveles diferentes.

Estos serían:

1. Nivel lógico
2. Nivel de presentación
3. Nivel de datos

Presentation tier

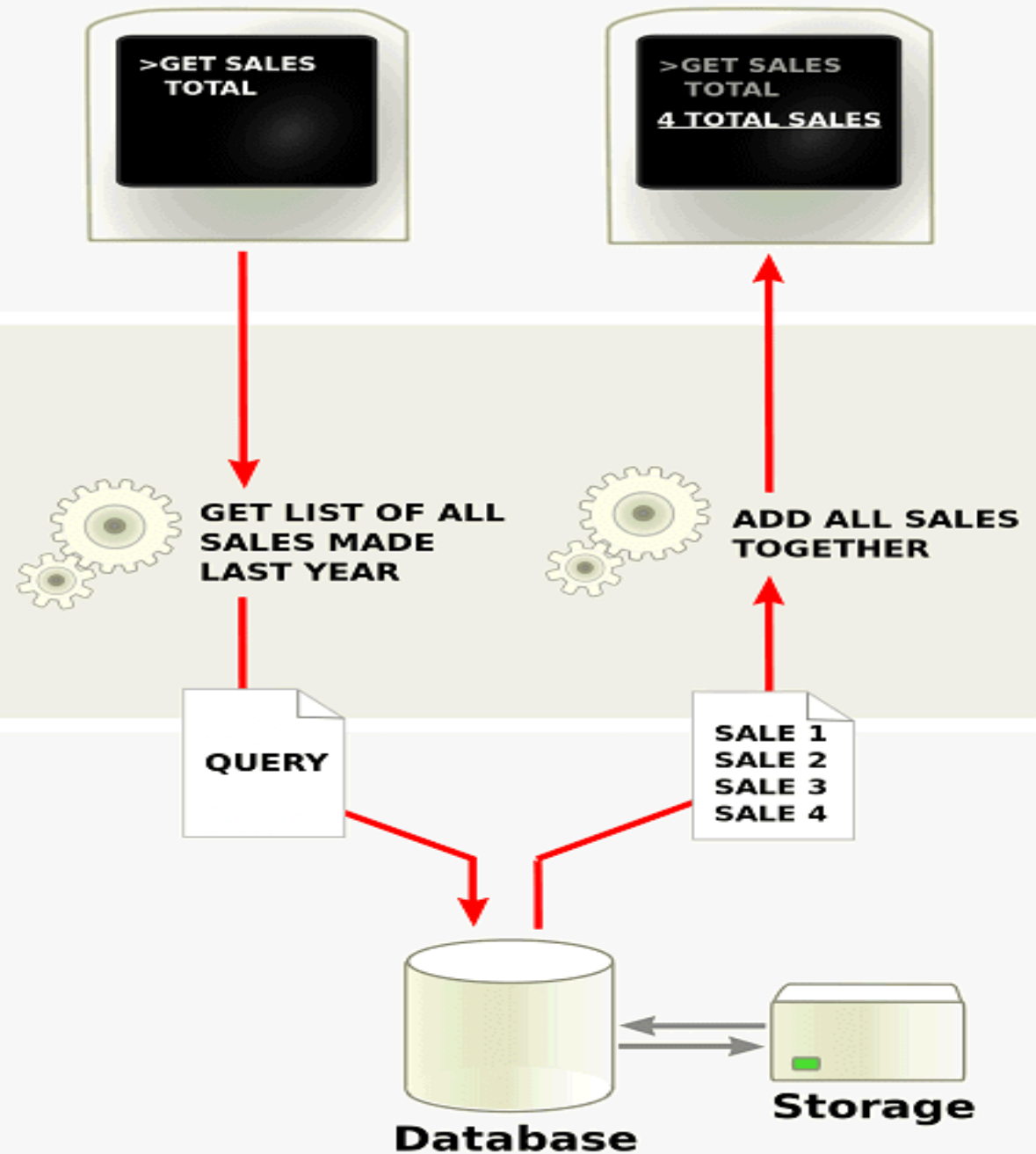
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



¿Cuáles son los beneficios de la arquitectura N-Tier?

1. Seguro: puede asegurar cada uno de los tres niveles por separado utilizando diferentes métodos.
2. Fácil de administrar: puede administrar cada nivel por separado, agregando o modificando cada nivel sin afectar los otros niveles.
3. Escalable: si necesita agregar más recursos, puede hacerlo por nivel, sin afectar los otros niveles.
4. Flexible: además de la escalabilidad aislada, también puede ampliar cada nivel de la manera que lo requieran sus requisitos.

N-Tier Architecture

Con la arquitectura n-tier, puede adoptar nuevas tecnologías y agregar más componentes sin tener que reescribir toda la aplicación o rediseñar todo su software, lo que hace que sea más fácil de escalar o mantener.

Mientras tanto, en términos de seguridad, puede almacenar información delicada o confidencial en el nivel lógico, manteniéndola fuera del nivel de presentación, lo que la hace más segura.

Otros beneficios incluyen:

1. Un desarrollo más eficiente.
2. La arquitectura N-tier es muy amigable para el desarrollo, ya que diferentes equipos pueden trabajar en cada nivel.
3. De esta forma, puede estar seguro de que los profesionales de diseño y presentación trabajan en el nivel de presentación y los expertos en bases de datos trabajan en el nivel de datos.

Otros beneficios incluyen:

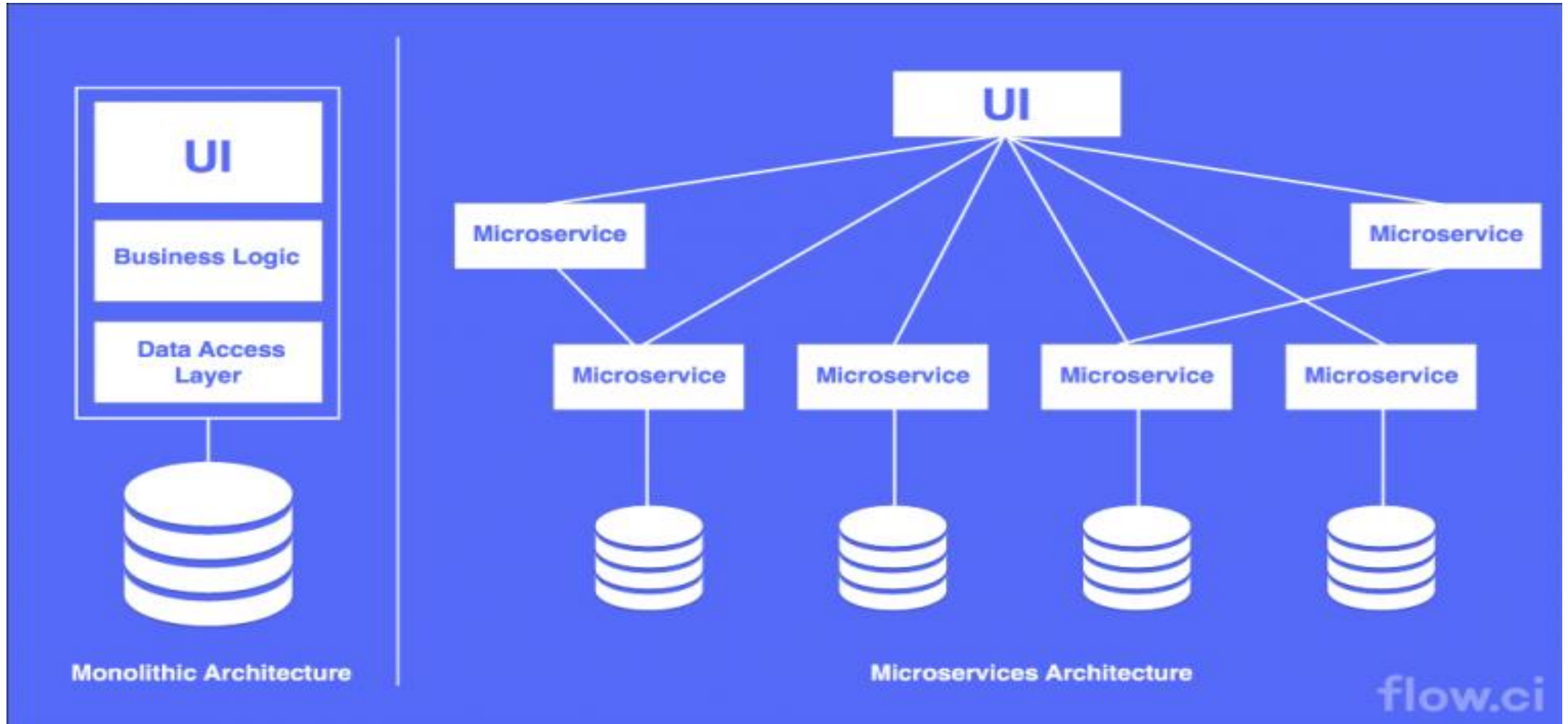
Fácil de agregar nuevas características. Si desea presentar una nueva característica, puede agregarla al nivel apropiado sin afectar los otros niveles.

Otros beneficios incluyen:

Fácil de reutilizar Debido a que la aplicación se divide en niveles independientes, puede reutilizar fácilmente cada nivel para otros proyectos de software.

Por ejemplo, si desea utilizar el mismo programa, pero para un conjunto de datos diferente, puede simplemente duplicar los niveles de lógica y presentación y luego crear un nuevo nivel de datos.

Micro Servicios vrs Aplicaciones monolíticas



Futuro Cercano

