

Patrón de varios niveles (Multi-tier)

El patrón de niveles múltiples es un patrón de C&C o un patrón de asignación, dependiendo de los criterios utilizados para definir los niveles. Se pueden crear niveles para agrupar componentes de similares funcionalidades, en cuyo caso es un patrón C&C. Sin embargo, en muchos, si no en la mayoría, los niveles de casos se definen teniendo en cuenta el entorno informático en el que el software se ejecutará: un nivel de cliente en un sistema empresarial no se ejecutará en la computadora que aloja la base de datos. Eso lo convierte en un patrón de asignación, mapeo elementos de software, tal vez producidos mediante la aplicación de patrones de C&C, a la computación elementos. Por esa razón, hemos optado por incluirlo como patrón de asignación.

Contexto: en una implementación distribuida, a menudo es necesario distribuir la infraestructura de un sistema en distintos subconjuntos. Esto puede ser para operaciones o negocios. razones (por ejemplo, diferentes partes de la infraestructura pueden pertenecer a diferentes organizaciones).

Problema: ¿Cómo podemos dividir el sistema en una serie de estructuras de ejecución computacionalmente independientes (grupos de software y hardware) conectadas por algunos medios de comunicación? Esto se hace para proporcionar entornos de servidor específicos optimizados para los requisitos operativos y el uso de recursos.

Solución: Las estructuras de ejecución de muchos sistemas están organizadas como un conjunto de agrupaciones lógicas de componentes. Cada agrupación se denomina nivel. La agrupación de componentes en niveles puede basarse en una variedad de criterios, como el tipo de componente, compartiendo el mismo entorno de ejecución o teniendo el mismo tiempo de ejecución objetivo.

El uso de niveles se puede aplicar a cualquier colección (o patrón) de componente de tiempo de ejecución, aunque en la práctica se utiliza con mayor frecuencia en el contexto de patrones cliente-servidor. Los niveles inducen restricciones topológicas que restringen qué componentes pueden comunicarse con otros componentes. Específicamente, los conectores existen solo entre componentes del mismo nivel o que residen en niveles adyacentes. El patrón de varios niveles que se encuentra en muchas aplicaciones Java EE y Microsoft .NET es un ejemplo de organización en niveles derivados del patrón cliente-servidor.

Además, los niveles pueden restringir los tipos de comunicación que pueden tomar en niveles adyacentes. Por ejemplo, algunos patrones escalonados requieren devolución de llamada comunicación en una dirección, pero notificación basada en eventos en la otra.

La principal debilidad de la arquitectura de varios niveles es su costo y complejidad. Para sistemas simples, los beneficios de la arquitectura de varios niveles pueden no justificar sus costos iniciales y continuos, en términos de hardware, software y diseño y complejidad de la implementación.

Los niveles no son componentes, sino agrupaciones lógicas de componentes. También, ¡no confunda niveles con capas! La creación de capas es un patrón de módulos (una unidad de implementación), mientras que los niveles se aplican solo a las entidades en tiempo de ejecución.

La tabla 13.11 resume la parte de solución del patrón de varios niveles.

Los niveles facilitan garantizar la seguridad y optimizar el rendimiento y disponibilidad de formas especializadas. También mejoran la modificabilidad del sistema, ya que los subgrupos computacionalmente independientes necesitan acordar protocolos para la interacción, reduciendo así su acoplamiento.

La figura 13.15 utiliza una notación informal para describir la arquitectura de varios niveles de la aplicación Java EE del sitio web del consumidor. Esta aplicación es parte del sistema Adventure Builder. Muchos tipos de componentes y conectores son específicos de la plataforma de soporte, que es Java EE en este caso.

Tabla 13.11 Solución de patrón de varios niveles

Descripción general	Las estructuras de ejecución de muchos sistemas están organizadas como conjunto de agrupaciones lógicas de componentes. Cada agrupación se denomina un nivel. La agrupación de componentes en niveles puede basarse en una variedad de criterios, como el tipo de componente, que comparten el mismo entorno de ejecución, o tener el mismo propósito de tiempo de ejecución.
Elementos	Nivel, que es una agrupación lógica de componentes de software. Los niveles pueden formarse sobre la base de plataformas informáticas comunes, en cuyo caso esas plataformas también son elementos del patrón.
Relaciones	Es parte de agrupar componentes en niveles. Se comunica con, para mostrar cómo los niveles y los componentes que contienen interactúan entre sí. Asignado a, en el caso de que los niveles se asignen a plataformas informáticas.
Restricciones	Un componente de software pertenece exactamente a un nivel.
Debilidades	Costo inicial y complejidad sustanciales

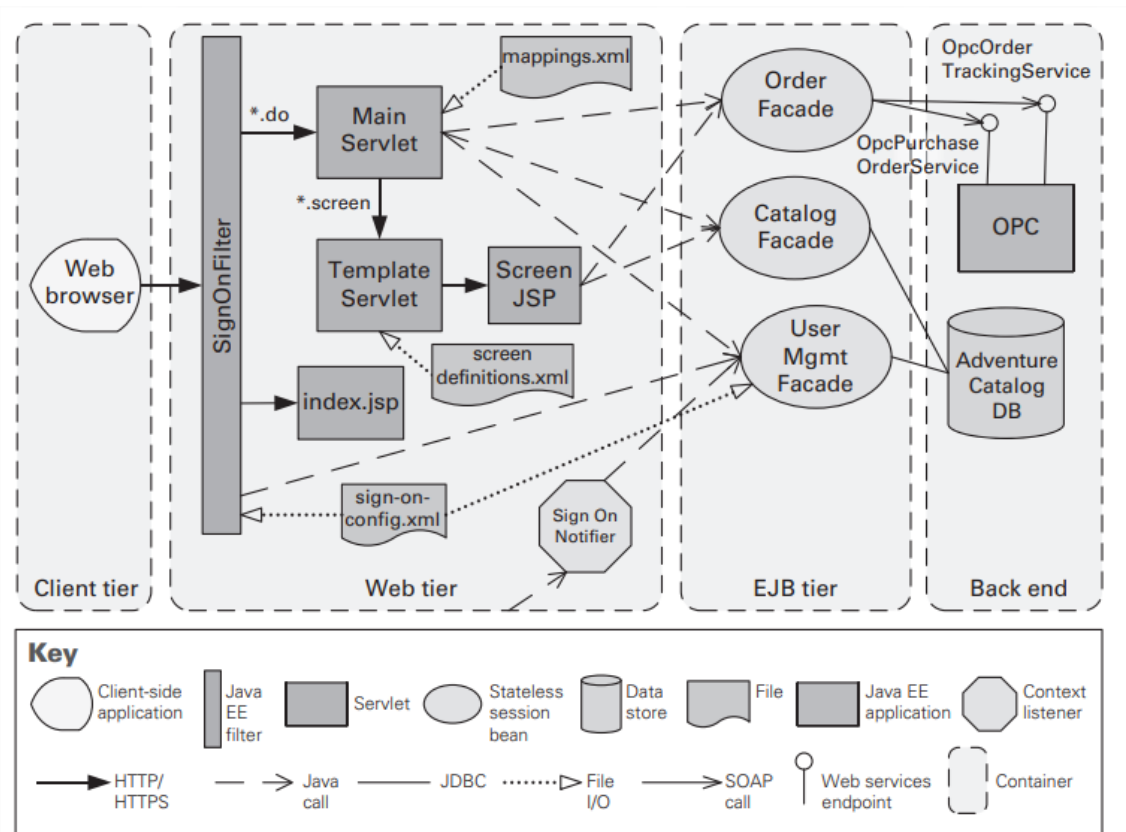


FIGURE 13.15 A multi-tier view of the Consumer Website Java EE application, which is part of the Adventure Builder system