5. Gestión de Tenant

Fundamentos de la gestión de tenant

Para comprender mejor el alcance y la naturaleza de la gestión de tenant, comencemos observando los componentes principales del universo de gestión de tenant.

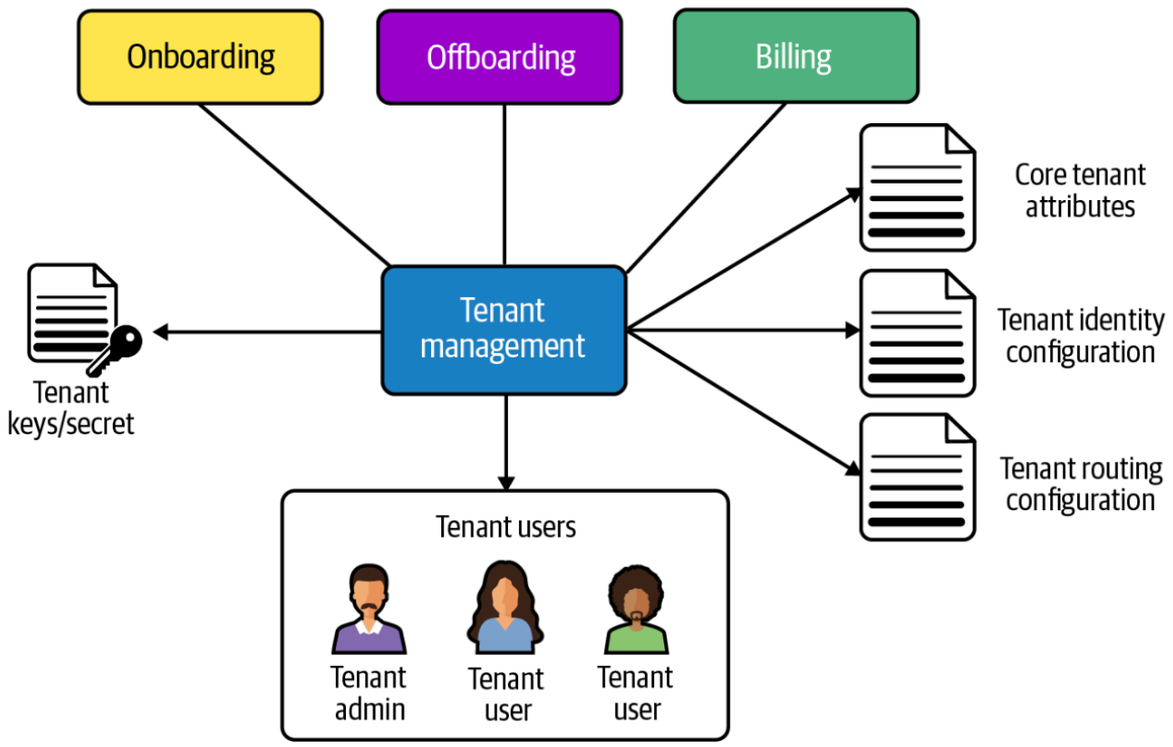

Figure 5-1. Tenant management's influence

En el lado derecho de la Figura 5-1, verás dos categorías diferentes de datos que se gestionan y configuran a través del servicio de gestión de tenant. **En este agrupamiento, encontrarás lo que he etiquetado como atributos principales del tenant.** Esto representa los datos básicos y fundamentales que se necesitan para la mayoría de tenants: identificador de tenant, estado (habilitado/deshabilitado), nivel de servicio, nombre de la empresa, estado del onboarding, fecha del último acceso, y así sucesivamente.

También he incluido un espacio separado para los parámetros de identidad. Estos parámetros contienen las diferentes propiedades de autenticación del tenant que se pueden configurar a través del servicio de gestión de tenant. Aquí es donde residen la MFA, políticas de contraseñas, asignaciones de proveedores de identidad y otros parámetros relacionados con la identidad.

En el lado izquierdo de la Figura 5-1, se muestra la configuración de claves y secretos. Estos parámetros se utilizan para configurar diferentes aspectos de seguridad de tu entorno. Podrías, por ejemplo, tener secretos o claves de cifrado por tenant que se gestionan a través de esta experiencia.

El administrador del tenant se crea cuando el tenant se introduce por primera vez en el sistema. Sin embargo, después del onboarding, el tenant también puede crear usuarios adicionales en su sistema, incluyendo otros administradores del tenant.

Construcción de un servicio de gestión de tenant

La interfaz de este servicio típicamente se divide en dos categorías lógicas.

- En primer lugar, tendrás un conjunto de operaciones enfocadas en la gestión básica de datos de configuración. Estas operaciones típicamente se exponen a través de una interfaz de crear, leer, actualizar y eliminar (CRUD).
- La otra categoría de operaciones se centra en operaciones más amplias de gestión de tenant (desactivación de tenant, desmantelamiento de tenant, y así sucesivamente). Estas operaciones tienden a contribuir significativamente a la complejidad general del servicio de gestión de tenant.

El primer conjunto de funciones gestiona la configuración, y los puntos de entrada inferiores soportan las diferentes operaciones relacionadas con la gestión.

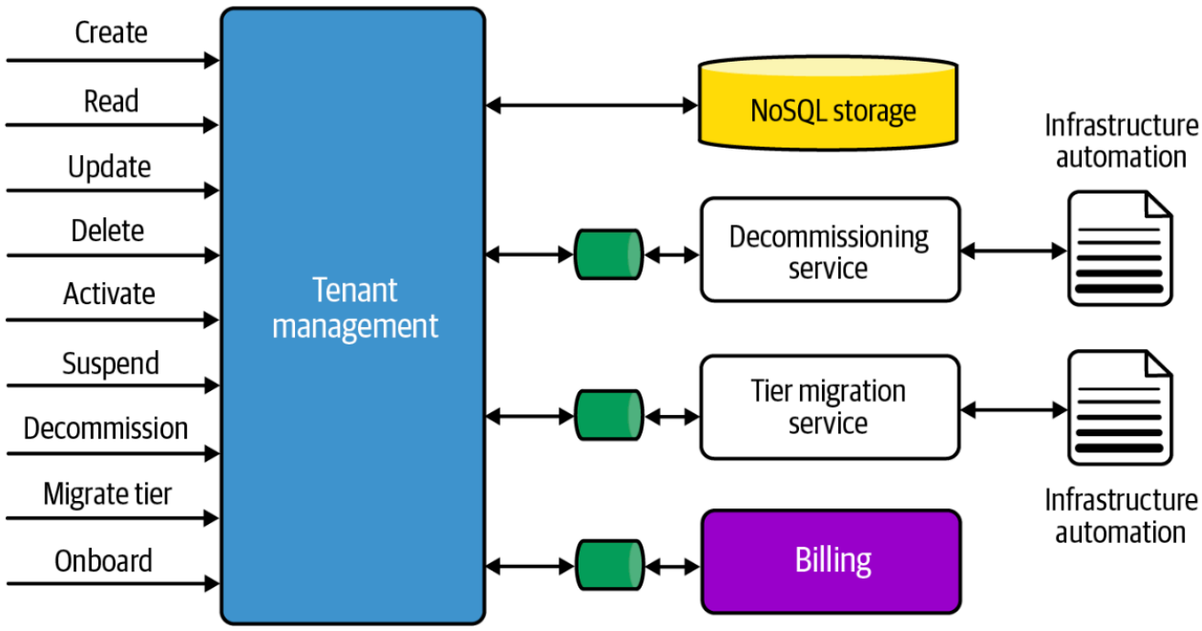

Figure 5-2. A sample tenant management implementation

El lado derecho de este diagrama destaca los diversos recursos de backend e integraciones que serían parte de tu experiencia de gestión de tenant. En la esquina superior derecha, he mostrado el almacenamiento.

Para este ejemplo, elegí usar almacenamiento NoSQL para mantener la información de configuración del tenant. **Generalmente, el tamaño y los patrones de consumo de estos datos se adaptan bien a un modelo de almacenamiento sin esquema.** Esto también facilita la aplicación de cambios en la estructura de configuración del tenant sin necesidad de actualizar esquemas o migrar datos.

Generación de un identificador de tenant

Dentro de la implementación de tu servicio de gestión de tenant, serás responsable de generar el identificador de tenant que representa la identidad única y universal de cualquier tenant en tu sistema.

El mecanismo más común utilizado para identificadores de tenant es un GUID. Proporciona una forma natural de tener un valor globalmente único sin dependencia de otros atributos del tenant.

Es importante notar que algunos entornos multi-tenant pueden incluir formas alternas y más amigables para identificar un tenant. Por ejemplo, si el sistema utiliza un subdominio o dominio personalizado para tenants individuales, entonces necesitarás alguna forma de mapear desde ese nombre de entidad al identificador de tenant. En este caso, podría tener `http://mycompany.saasprovider.com/` donde el subdominio "mycompany" representa un nombre visible externamente que se mapearía a un identificador de tenant interno.

Incluso cuando tengas algún otro nombre que estés utilizando como parte de la entrada en tu sistema, **estos nombres de entidades o referencias no deben ser considerados como tu identificador de tenant.** Hay buenas razones para mantenerlos separados.

Almacenamiento de configuración de infraestructura

Además de almacenar atributos básicos del tenant, la gestión de tenant también puede utilizarse para almacenar información de configuración de infraestructura específica del tenant.

Dependiendo del modelo de despliegue de tu aplicación multi-tenant, podrías tener configuración de identidad, patrones de enrutamiento y otras opciones de infraestructura almacenadas y gestionadas por tu servicio de gestión de tenant.

Estos datos típicamente no se gestionan directamente a través de tu experiencia de administración. En cambio, se almacenan aquí durante la configuración de tu infraestructura multi-tenant y luego se referencian por las diferentes partes de tu experiencia que necesitan esta información para configurar o resolver asignaciones de tenant que son parte de tu entorno.

Aunque el servicio de gestión de tenant es un buen hogar centralizado para el estado y la configuración del tenant, siempre debes asegurar que este servicio no se convierta en un cuello de botella de tu sistema.

Si los datos almacenados aquí están siendo accedidos frecuentemente por todas las partes móviles de tu sistema, necesitarás considerar estrategias alternas para gestionar y acceder a estos datos. **Idealmente, el estado aquí no será usado intensivamente por los servicios de aplicación de tu aplicación.**

Esta es una de las razones por las que colocamos los atributos críticos del tenant en el JWT, limitando tu necesidad de continuamente regresar a un servicio centralizado único para continuamente adquirir este contexto.

Gestión de la configuración del tenant

Es importante destacar que la gestión del tenant va más allá del onboarding. Este servicio también se utiliza para dar soporte a diferentes operaciones y casos de uso a lo largo de todo el ciclo de vida de un tenant.

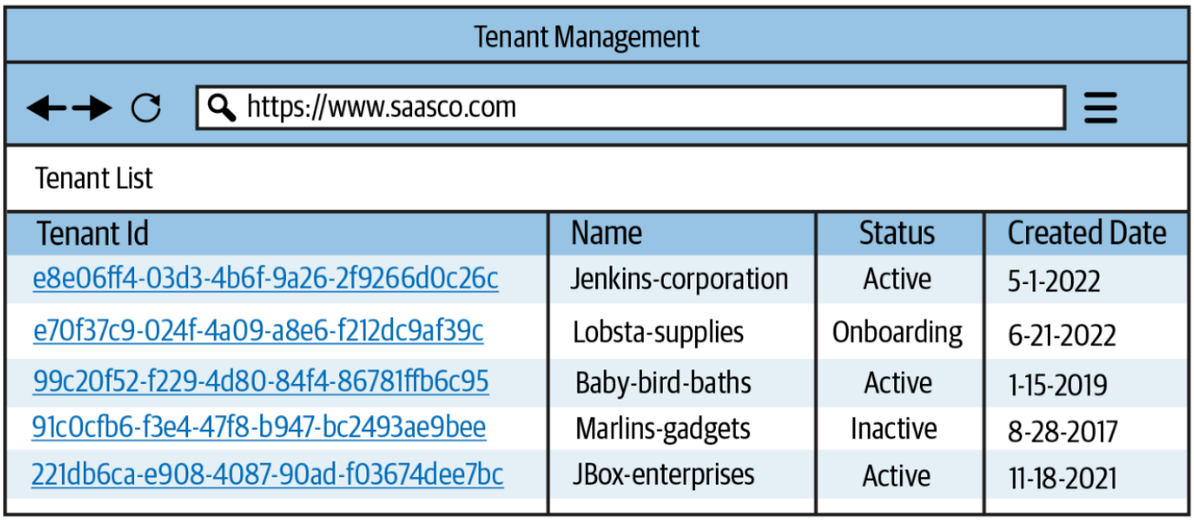

Figure 5-3. Managing tenants from the admin console

Además de mostrar información sobre el tenant, la consola también es el lugar donde se editan y gestionan las políticas para un tenant específico.

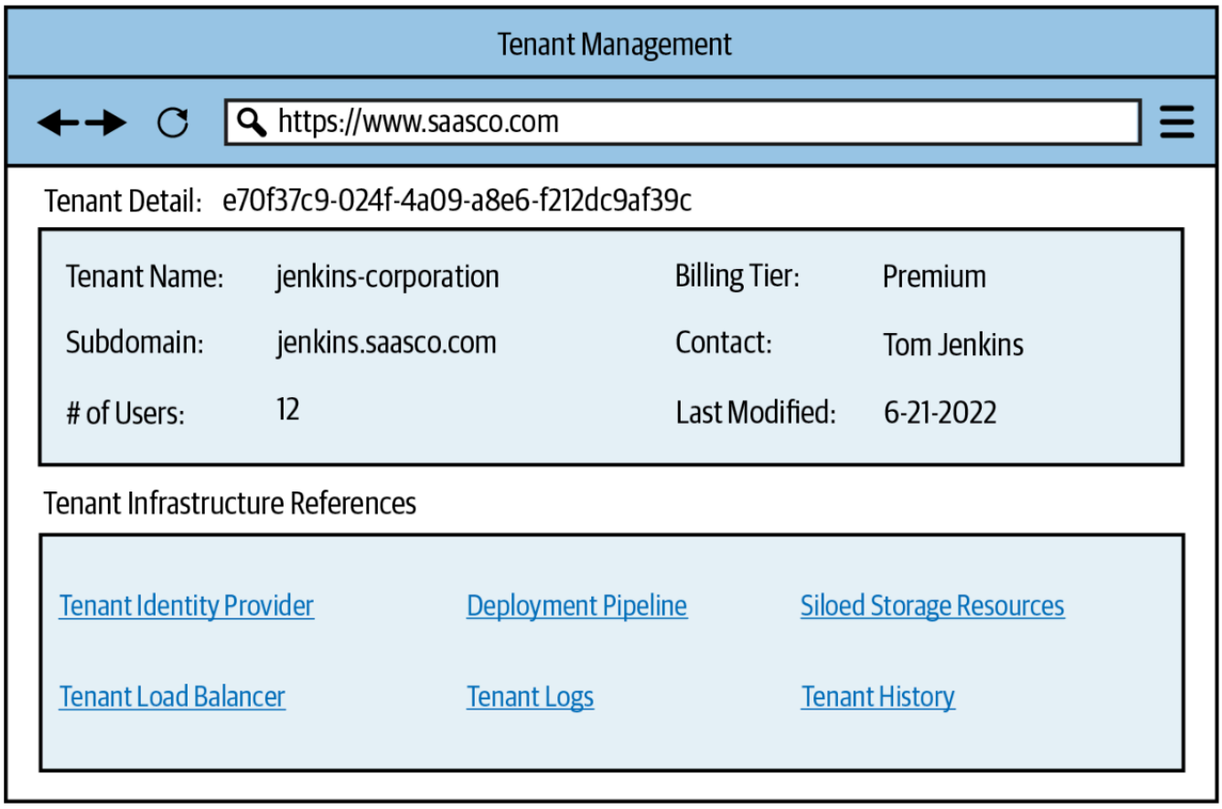

Figure 5-4. Managing tenant details from the console

En la parte inferior de la imagen, se puede ver una sección que incluye hipervínculos a recursos clave de infraestructura asociados con el tenant actual. **Estos enlaces te llevan directamente a la página de administración del proveedor de la nube o de la infraestructura para cada recurso**, lo que permite un acceso rápido a los recursos en el contexto de un tenant específico.

Cuanto más compartido (en *pool*) sea tu entorno, menos probable es que obtengas mucho valor al proporcionar este acceso contextual por tenant a los recursos de infraestructura.

Gestión del ciclo de vida del tenant

El enfoque ahora se traslada a pensar en los diversos estados por los que un tenant podría pasar durante su tiempo en tu sistema.

Activación y desactivación de un tenant

Generalmente, esta configuración se utiliza para gestionar la capacidad de un tenant para acceder a tu sistema. Con esta mentalidad, no estamos eliminando tenants del sistema. Solo estamos cambiando un interruptor para activar o desactivar su acceso.

Cuando un tenant es desactivado, tu servicio de gestión de tenant deberá asumir la responsabilidad de determinar qué acciones deben orquestarse para garantizar que se bloquee el acceso del tenant al sistema.

La gestión del estado activo de un tenant también puede estar conectada a la experiencia de facturación de tu aplicación. En algunos casos, es el sistema de facturación el que puede estar en primera línea en la gestión del estado activo de tu tenant. **Imagina un escenario en el que un tenant ha dejado de pagar. En este caso, esta información podría aparecer primero en tu sistema de facturación, que identifica a los tenants morosos.** Cuando este evento es desencadenado por el sistema de facturación, debes determinar cómo responderá tu sistema a estos eventos.

Como parte de la desactivación, también debes determinar cómo afecta esto al estado de los usuarios que están actualmente conectados al sistema. **¿Permites que continúen o terminas inmediatamente su sesión activa?** Muchos proveedores de SaaS tienden a inclinarse por un modelo más pasivo, permitiendo que los usuarios del tenant finalicen sus sesiones activas.

Por supuesto, todo lo que se puede desactivar también se puede reactivar, por lo que también querrás considerar ese camino.

La conclusión clave es que el estado del tenant debe ser gestionado centralmente por tu servicio de gestión de tenant. Debe ser visto como la única fuente de verdad para gestionar el estado de los tenants, asegurando que cualquier impacto de los cambios de estado se sincronice con las partes dependientes de tu sistema.

Desmantelamiento de un tenant

La desactivación solo tiene como objetivo suspender la cuenta de un tenant. No afecta la huella de recursos existente del entorno del tenant.

El desmantelamiento normalmente vendría después de la desactivación. Después de un cierto período de tiempo, los recursos no utilizados de este tenant pueden estar contribuyendo a costos y complejidad que no añaden ningún valor al negocio. **Ahora, tienes que considerar cómo pasar de un estado desactivado al desmantelamiento de los recursos del tenant.**

Tienes algunas opciones a la hora de elegir una estrategia de desmantelamiento.

- Podrías optar por eliminar simplemente cualquier recurso asociado con el tenant, eliminándolos esencialmente por completo del sistema, o podrías optar por archivar el estado del tenant antes de desmantelar sus recursos.
- Como parte de esto, querrás reconsiderar lo que significa rehidratar un tenant que ha sido desmantelado. Podrías, por ejemplo, dejar en su lugar elementos del tenant que tienen un impacto mínimo en tu sistema (el tenant, sus usuarios, etc.). Esto podría hacer que el proceso de reactivar al tenant sea algo más sencillo sin añadir mucho costo o complejidad a tu entorno.

En última instancia, todo esto forma parte del equilibrio que implica crear tu estrategia de desmantelamiento.

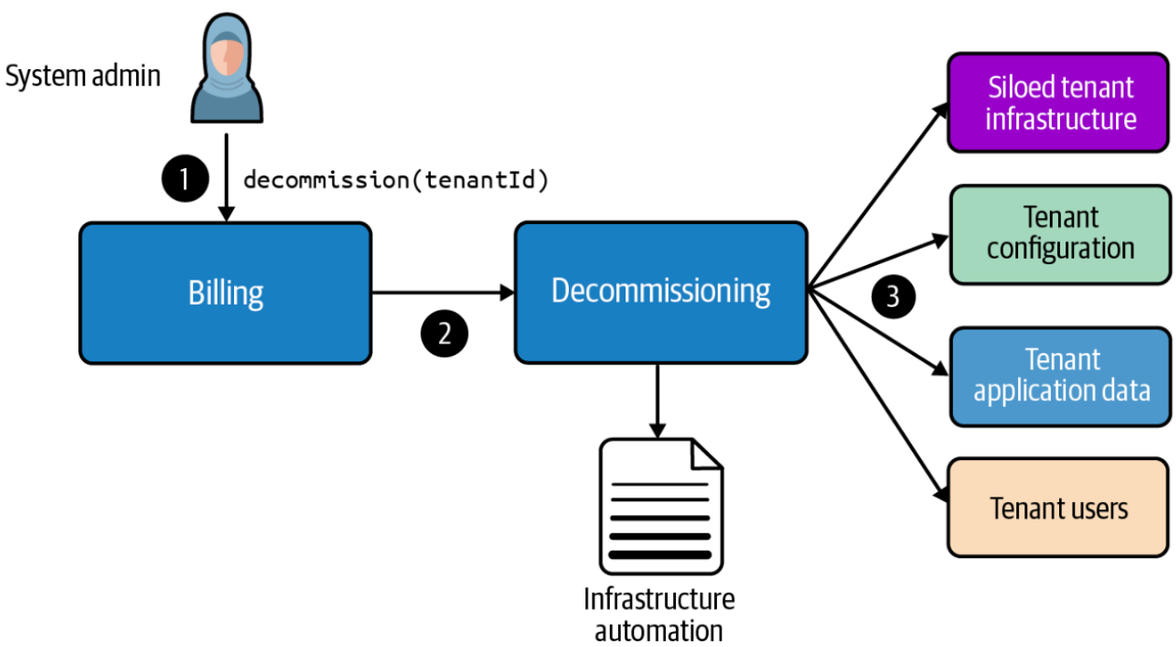

Figure 5-7. Decommissioning tenant resources

Para mí, parece más natural separar esto y permitir que el desmantelamiento exista como su propio proceso que sería desplegado, gestionado y ejecutado fuera del contexto del servicio de gestión de tenant.

El servicio de desmantelamiento tiene la tarea de iterar sobre todos los diferentes elementos del tenant y eliminar cada uno. Esto se logrará mediante una combinación de herramientas de automatización de infraestructura y scripts, así como llamadas a API. La naturaleza de cada recurso del tenant puede requerir una estrategia de desmantelamiento diferente.

Naturalmente, si tenemos alguna infraestructura de tenant en silo aquí, esos recursos en silo serán eliminados de tu sistema. Esto puede implicar eliminar todas las piezas de un despliegue de silo completo o solo los recursos individuales que podrían estar desplegados en un modelo en silo.

Cuando tienes datos en pool, eso significa que tu proceso de desmantelamiento deberá ser capaz de localizar y eliminar selectivamente los datos que se encuentran junto a los datos de otros tenants.

Cada microservicio en nuestro sistema podría gestionar datos de tenant en pool y cada uno de estos servicios puede depender de diferentes tecnologías de almacenamiento. Esto significa que tu proceso de desmantelamiento puede necesitar código separado

para eliminar datos de cada una de estas fuentes.

Automatizar este proceso puede ser abrumador, ya que requiere que los equipos estén extremadamente atentos para garantizar que su estrategia de desmantelamiento no afecte a los tenants existentes.

La última parte en torno al desmantelamiento se centra en archivar el estado y los datos del tenant. Para algunos proveedores de SaaS, esto puede permitirles continuar preservando el estado existente del tenant sin retener todas las demás partes móviles del entorno de un tenant.

Cambio de niveles (Tiers) de servicio del tenant

La última pieza de nuestra historia de gestión del ciclo de vida del tenant analiza lo que significa mover a un tenant de un nivel de servicio a otro. Para muchos, este cambio representa una de las dimensiones más desafiantes de la gestión del estado del tenant.

Moverse entre los tiers de servicio básico y premium en este ejemplo se limita a unas pocas áreas muy aisladas de este entorno multi-tenant. Aquí, se utilizan *feature flags* (indicadores de funcionalidad) dentro de nuestra aplicación para habilitar rutas, características y flujos de trabajo que están disponibles para el tenant del nivel de servicio premium.

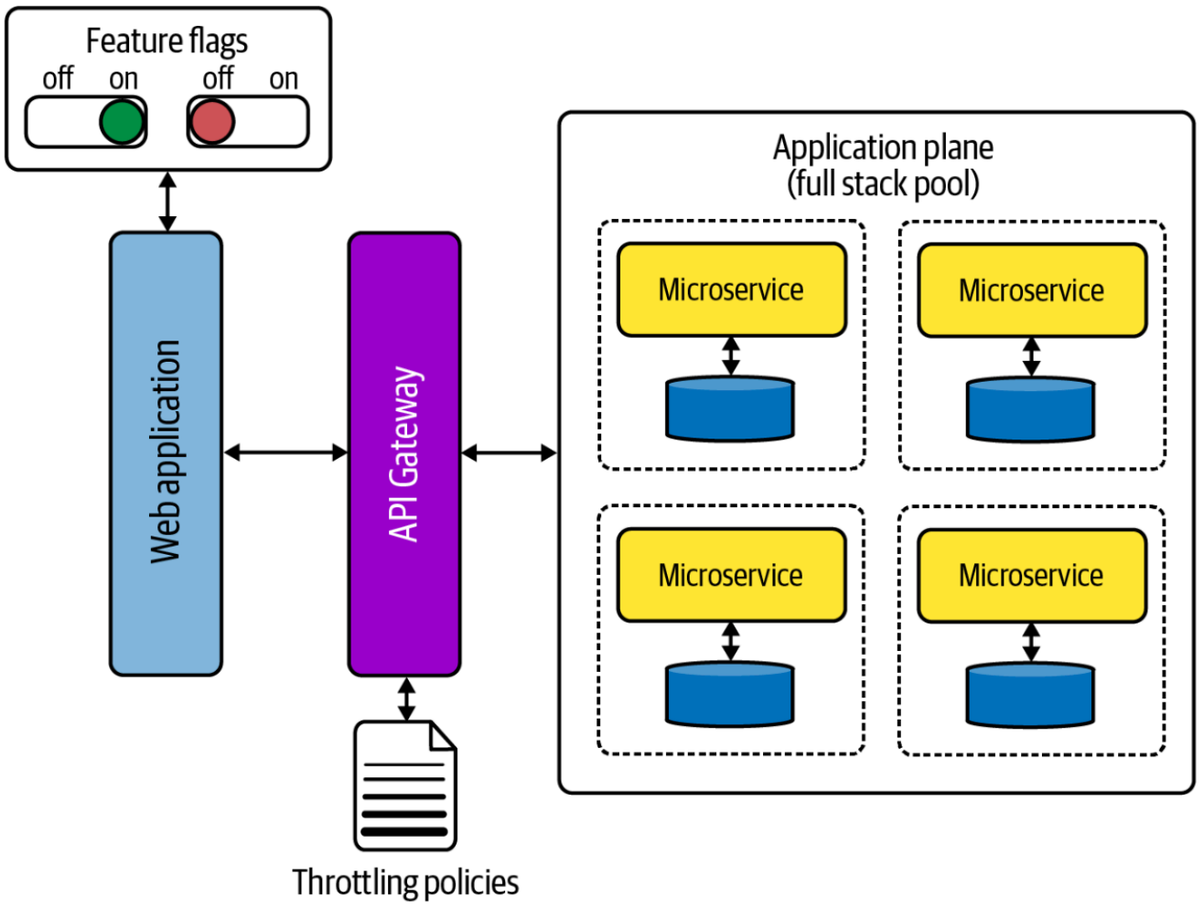

Figure 5-9. Switching tiers in a full stack pool model

Ahora, consideremos lo que significaría moverse entre niveles de servicio en un entorno con una huella de recursos más compleja.

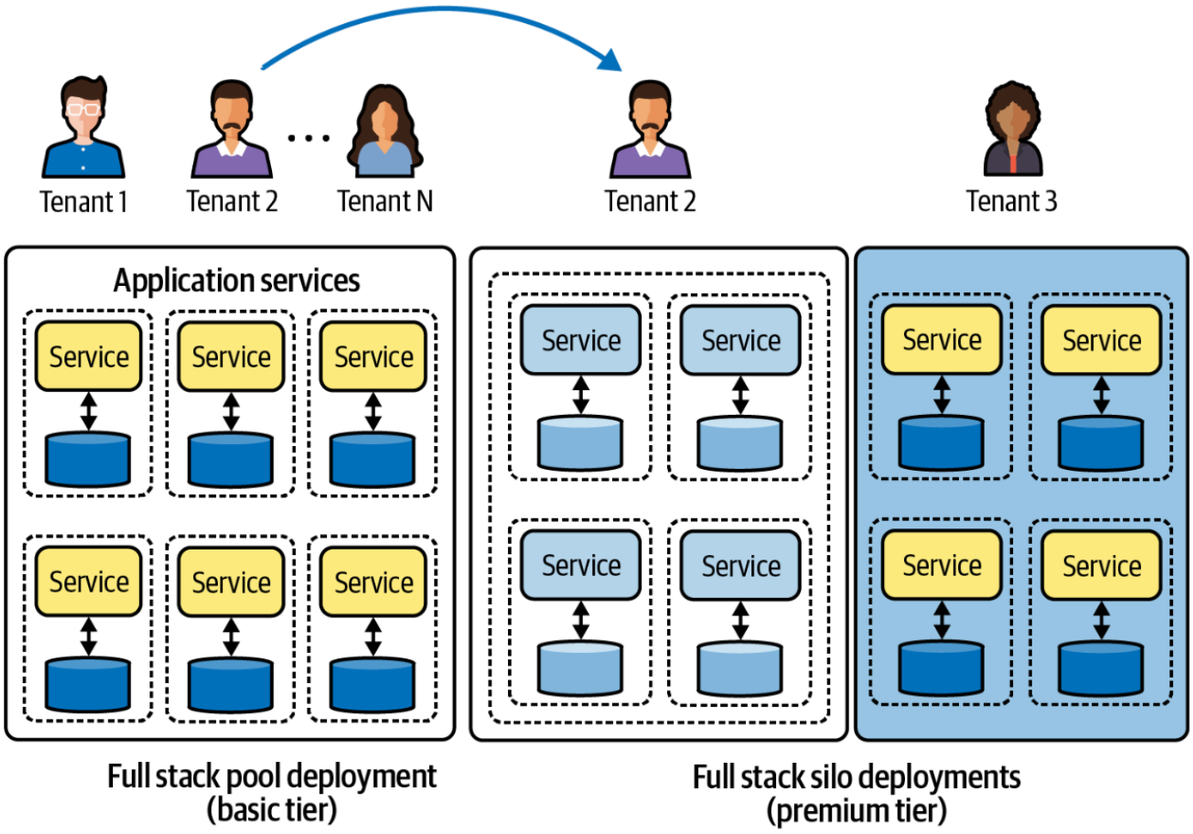

Figure 5-10. Migrating from full stack pool to full stack silo

Ahora debes pensar si esta será una migración sin tiempo de inactividad (*zero-downtime*) o si requerirás que los tenants se desactiven para pasar al nuevo nivel de servicio. También tendrás que escribir el nuevo código que mueva todos los datos y el estado del entorno en pool a tu nuevo silo de pila completa.

Muchos de los principios y estrategias que se usan comúnmente para migrar cualquier entorno de software se aplican aquí; solo que están siendo acotados a nivel de tenant.

El movimiento de datos es la parte más grande de este esfuerzo.

Para cualquier tipo de migración de nivel de servicio que consideres, también deberás tener en cuenta cómo esta migración podría afectar a los tenants que actualmente utilizan tu sistema. **Querrás asegurarte de que la extracción de datos y estado del tenant no tenga ningún impacto adverso en las cargas de trabajo existentes.**

Si te estás moviendo del nivel premium al básico y este movimiento implica que algunas partes de tu sistema pasen de una infraestructura en silo a una en pool, entonces tu migración ahora se centrará en cómo mover el cómputo y los datos a sus elementos en pool.