

# 1. La mentalidad SaaS

## Definición de SaaS

SaaS es un modelo de negocio y de entrega de software que permite a las organizaciones ofrecer sus soluciones en un modelo centrado en el servicio, de baja fricción, que maximiza el valor para clientes y proveedores.

Se apoya en la agilidad y la eficiencia operativa como pilares de una estrategia de negocio que promueve el crecimiento, el alcance y la innovación.

## Dónde empezamos

Generalmente, los sistemas pre-SaaS se entregaban en un modelo de “software instalado” donde los clientes participaban en la instalación y configuración de su software.

En este modelo, los equipos de desarrollo de software solían estar más alejados de esos detalles de entrega y configuración. Se enfocaban en ampliar continuamente las capacidades funcionales de sus soluciones.

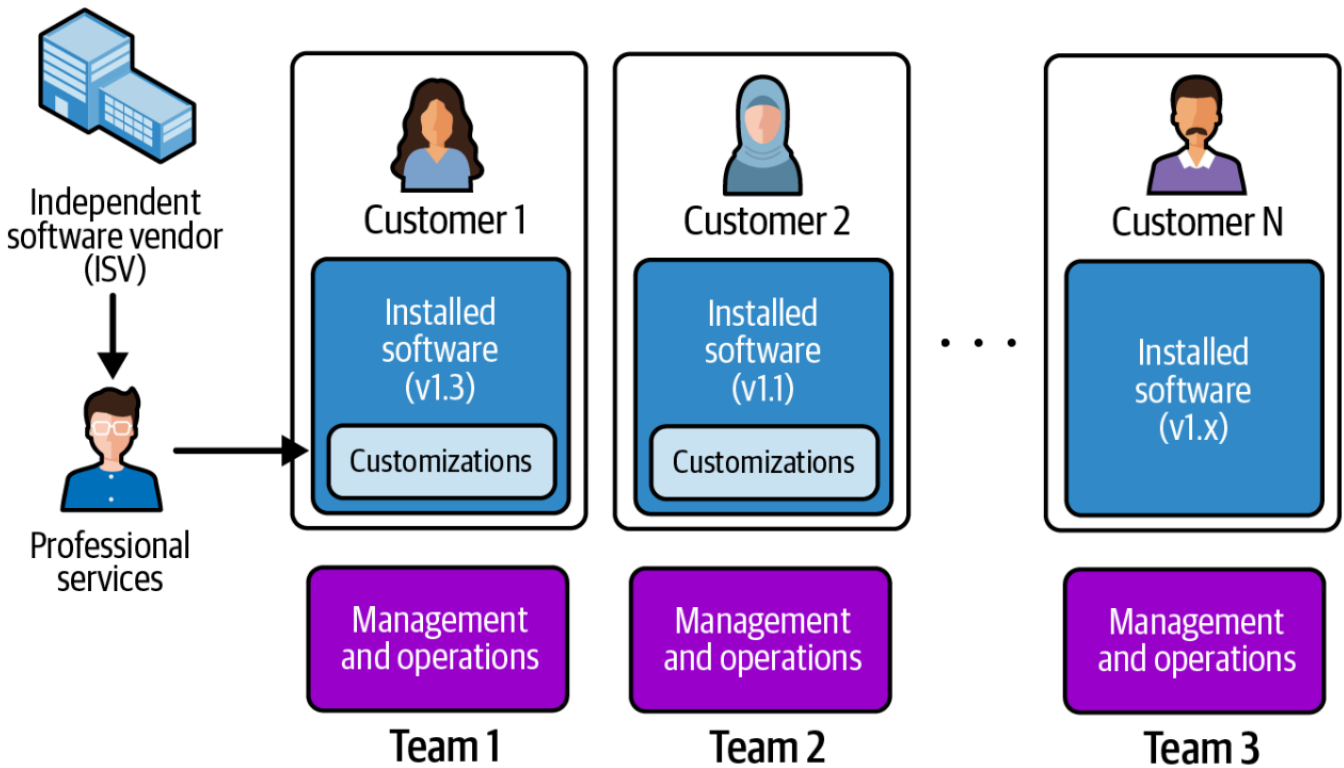


Figure 1-1. The installed software model

Este modo clásico de entrega de software es mucho más impulsado por ventas, donde el negocio se enfoca en adquirir clientes y transferirlos a los equipos de tecnología para atender las necesidades específicas de cada cliente entrante.

Siendo justos, construir y entregar software usando este modelo es y seguirá siendo un enfoque perfectamente válido para algunos negocios. Si tienes un número limitado de clientes y solo incorporas unos pocos al año, este modelo podría ser completamente adecuado. Aun tendrías ineficiencias, pero serían mucho menos prominentes.

La sobrecarga incremental de soportar cada nuevo cliente comienza a tener impactos reales en el negocio, erosionando márgenes y agregando continuamente complejidad al perfil operativo de la empresa. **El problema más grande aquí, sin embargo, es cómo este modelo impacta la agilidad, la competencia, el crecimiento y la innovación.**

Los clientes se interesan menos en mantener la capacidad de gestionar o controlar sus entornos. En su lugar, están más interesados en maximizar el valor que pueden extraer de su software.

Los clientes también se están inclinando por modelos de pricing que se alineen mejor con su perfil de valor y consumo. En algunos casos, buscan la flexibilidad de modelos de subscription y/o pay-as-you-go.

## El movimiento hacia un modelo unificado

Si eres un B2B Independent Software Vendor (ISV) con miles de clientes, por ejemplo, es poco probable que tu negocio pueda sostener un modelo donde cada cliente tenga que ser soportado, gestionado y operado de forma separada.

Aquí es donde vimos a los equipos adoptar un modelo de infraestructura compartida que les permite escalar su negocio y simplificar su modelo operativo de manera más efectiva.

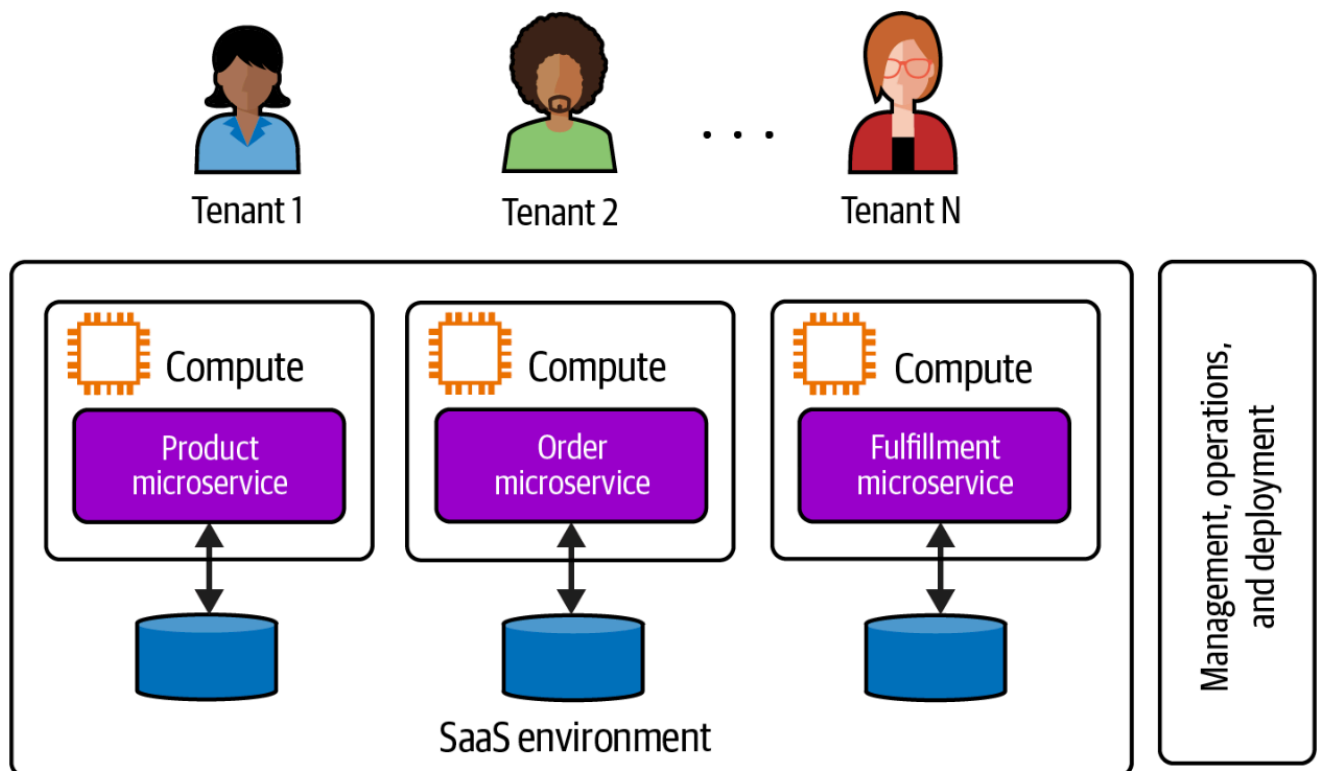


Figure 1-2. A shared infrastructure SaaS model

En su lugar, hemos pasado a una estrategia unificada donde todos los servicios de aplicación del sistema y la infraestructura son compartidos por los clientes. También verás que he reemplazado el término “cliente” por “tenant”.

La idea fundamental es que, al movernos hacia esta mentalidad unificada, miramos nuestro entorno de forma diferente. Ahora es un conjunto de recursos compartidos y ocupados por uno o más consumidores. **La idea es que esos consumidores representan clientes temporales de tu entorno, consumiendo solo los recursos que necesitan, de ahí el término “tenant”. (Inquilino)**

Con infraestructura compartida, tu automatización del despliegue simplemente desplegaría la actualización en este entorno SaaS unificado, y todos tus tenants tendrían acceso inmediato a los cambios.

Las ventajas de la infraestructura compartida se extienden a casi todos los aspectos de un negocio de software. Puede:

- Agilizar la agregación y recolección de operational telemetry.
- Simplificar la complejidad de tu automatización DevOps.
- Facilitar el onboarding de nuevos tenants.
- Correlacionar el consumo de infraestructura con la actividad real de tenants permite a los equipos maximizar márgenes y lograr economías de escala.

Además de unificar la experiencia, también aportó un nuevo nivel de agilidad a estos entornos. Bien contruidos, estos entornos crean oportunidades para lanzar nuevas funcionalidades y capacidades a un ritmo mucho más rápido, permitiendo a las organizaciones reaccionar y responder con mayor agilidad a las necesidades de clientes/mercado.

La naturaleza elástica y pay-as-you-go de la infraestructura cloud también se alinea muy bien con este modelo, soportando los modelos de pricing y escalado que encajan naturalmente con la elasticidad del cloud.

Vale la pena señalar que este movimiento hacia la infraestructura compartida también introduce una serie de nuevos desafíos. Soportar infraestructura compartida influirá directamente en el perfil de:

- Seguridad,
- Rendimiento,
- Escalabilidad,
- Disponibilidad,
- y resiliencia de tu entorno SaaS.

**Estos factores tienen un impacto claro en cómo abor das el diseño e implementación de tu entorno SaaS.**

Para crear una experiencia unificada, también debemos introducir un nuevo conjunto de componentes transversales que aporten toda la funcionalidad necesaria para gestionar, operar y desplegar una aplicación SaaS de forma centralizada.

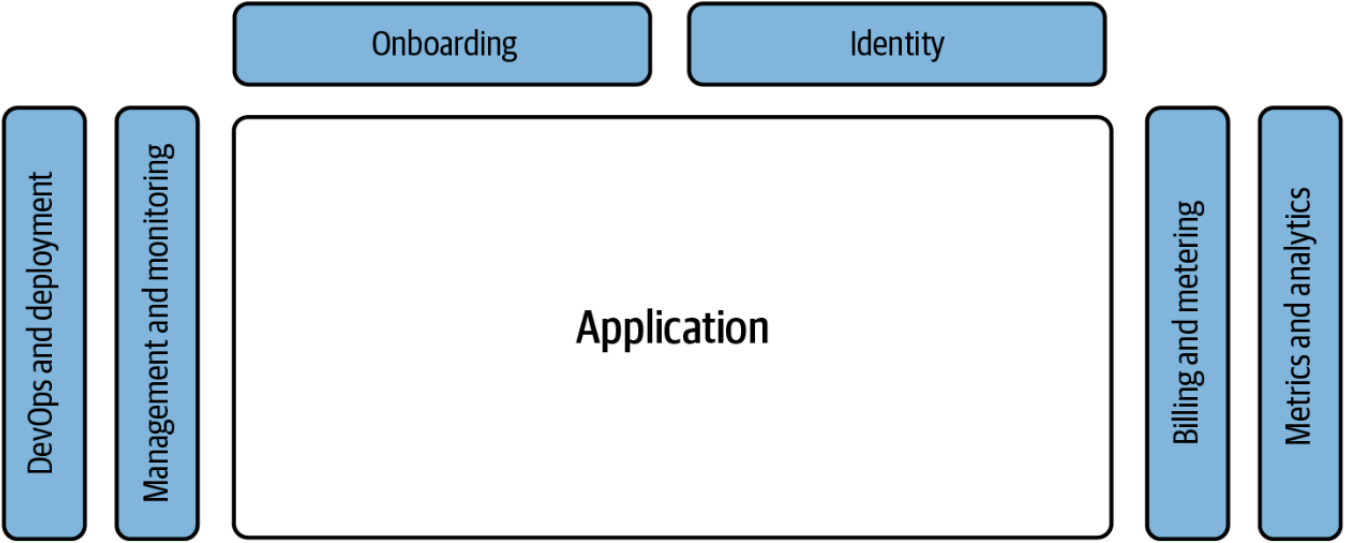


Figure 1-3. Building cross-cutting SaaS capabilities

Ahora bien, para muchos desarrolladores SaaS, es tentador ver estos componentes circundantes como elementos secundarios y menos críticos de su arquitectura SaaS.

Parte de nuestra eficiencia se logra a través de los servicios que se muestran aquí, y otra parte se consigue mediante las estrategias que aplicas a tu arquitectura de aplicación. **Si tu arquitectura de aplicación comparte infraestructura, puede añadir más eficiencia y economías de escala a tu entorno.**

**| Redefinición de multi-tenancy**

Durante años, en muchos círculos, el término “multi-tenant” se utilizó para transmitir la idea de que algún recurso era compartido por múltiples tenants. Esto crea una conexión muy estrecha entre multi-tenancy y la idea de un recurso compartido.

Los entornos SaaS sí comparten infraestructura, y ese compartir de infraestructura es ciertamente válido para etiquetarlo como multi-tenant.

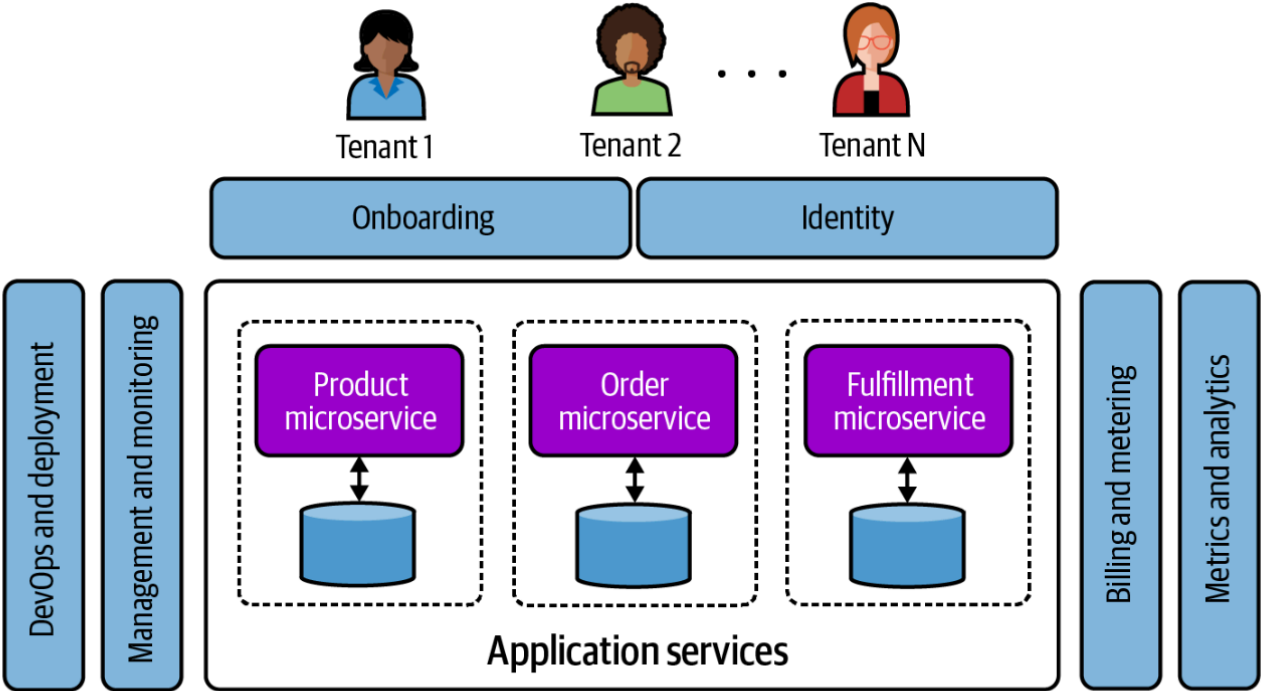


Figure 1-4. A sample multi-tenant environment

El reto es que los entornos SaaS no se ajustan exclusivamente a este modelo. Supongamos, por ejemplo, que creo un entorno SaaS que luce así

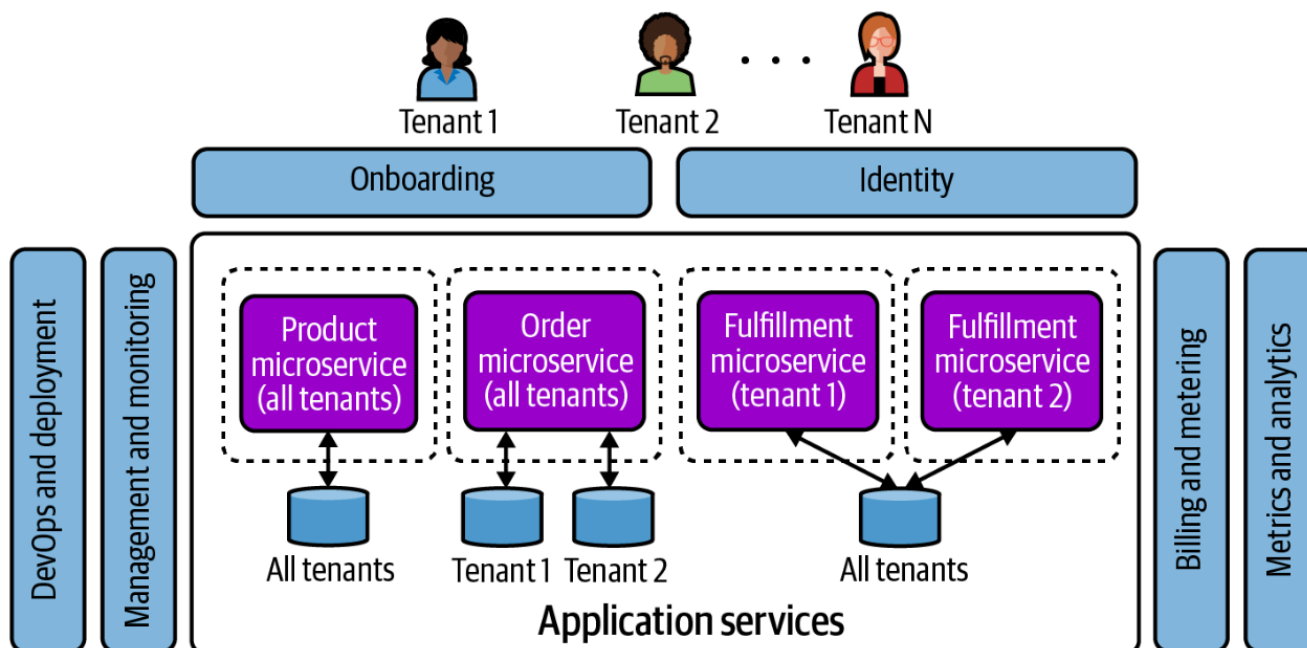


Figure 1-5. Multi-tenancy with shared and dedicated resources

Nuestros requisitos de dominio, rendimiento y/o seguridad pueden habernos exigido separar el almacenamiento para cada tenant. Así que, el cómputo de nuestro microservicio de Order sigue siendo compartido, pero tenemos bases de datos separadas para cada tenant.

**Si nos atenemos a la definición más pura de multi-tenancy, realmente no podríamos decir que todo lo que se ejecuta aquí cumple con la definición original de multi-tenancy.**

Cuando compones un entorno SaaS, no te adhieres a una única definición absoluta de multi-tenancy; eliges las combinaciones de recursos compartidos y dedicados que mejor se alinean con los requisitos de negocio y técnicos de tu sistema. **Todo esto forma parte de optimizar la huella de tu arquitectura SaaS en torno a las necesidades del negocio.**

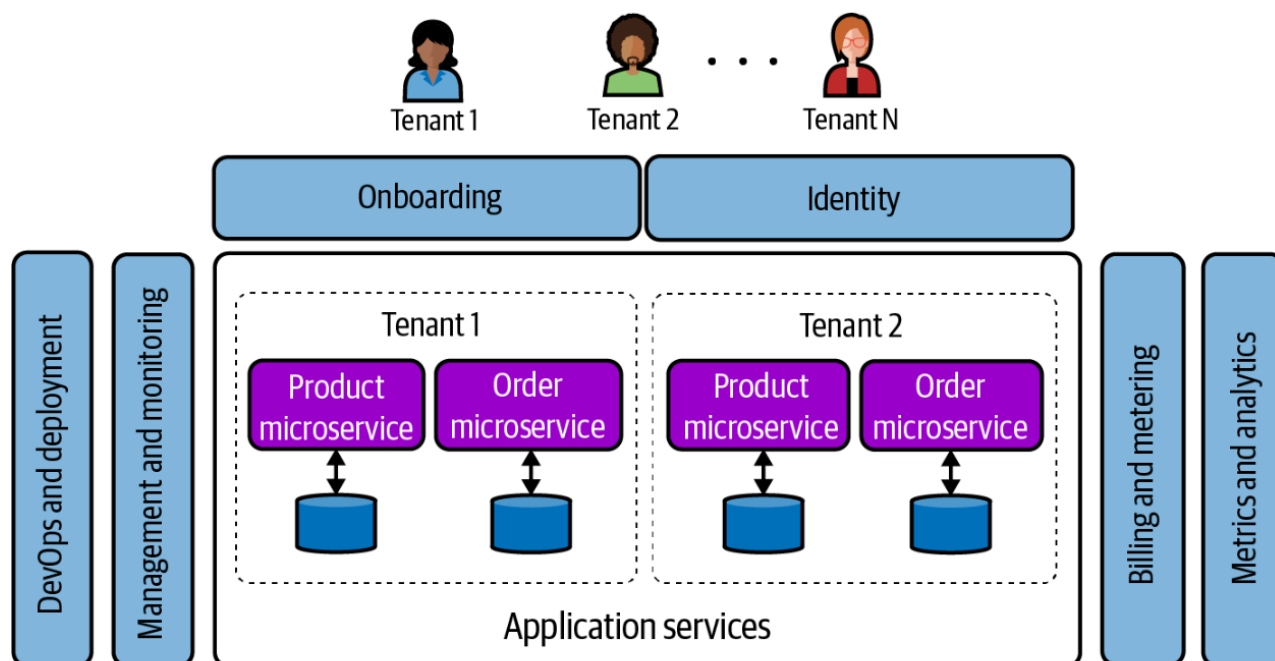


Figure 1-6. A multi-tenant environment with fully dedicated resources

En este ejemplo, los requisitos de dominio, de mercado y/o de legado nos han obligado a tener todo el cómputo y el almacenamiento ejecutándose en un modelo dedicado donde cada tenant tiene un conjunto completamente separado de recursos de infraestructura.

Aunque nuestros tenants no están compartiendo infraestructura en este modelo, verás que siguen siendo onboarded, gestionados y operados mediante el mismo conjunto de capacidades compartidas que ha abarcado todos nuestros ejemplos. **Eso significa que todos los tenants siguen ejecutando la misma versión del software y que siguen siendo gestionados y operados de forma colectiva.**

Los proveedores SaaS en proceso de migración suelen emplear este modelo como primer paso hacia SaaS. Otras industrias pueden tener requisitos de aislamiento tan extremos que no se les permite compartir infraestructura. Hay una larga lista de factores que legítimamente podrían situar a un proveedor SaaS en este modelo.

A partir de ahora, **multi-tenant se referirá a cualquier entorno que realiza el onboarding, despliega, gestiona y opera tenants a través de una experiencia única y unificada.** El carácter compartido de cualquier infraestructura no tendrá correlación con el término “multi-tenancy”.

## ¿Dónde están los límites de SaaS?

Supongamos que tu aplicación SaaS requiere que partes del sistema se desplieguen en alguna ubicación externa, o imagina escenarios donde tu aplicación tiene dependencias de soluciones de otros proveedores.

Aquí es donde no queremos ser demasiado extremos con nuestra definición de SaaS. Para mí, el límite tiene más que ver con cómo se configuran, gestionan y operan estas dependencias externas. **Si su presencia está completamente oculta para tus tenants y siguen siendo gestionadas y operadas a través de tu experiencia centralizada, para mí esto sigue siendo SaaS.**

Mi regla general aquí (con algunas excepciones) es que estamos proporcionando una experiencia de servicio. En un modelo de servicio, la visión de nuestros tenants se limita a la superficie de nuestro servicio. Las herramientas, tecnologías y recursos que se utilizan para dar vida a ese servicio deben estar completamente ocultos para nuestros tenants.

## El modelo de managed service provider (MSP)

Con MSP, intentaremos obtener eficiencias y economías de escala trasladando las operaciones a un equipo o entidad centralizada. Este es el servicio que proporcionan estos MSP. A menudo asumen la responsabilidad de instalar, gestionar y dar soporte a cada uno de estos clientes, intentando extraer algo de escala y eficiencia de las herramientas y mecanismos que utilizan para operar esos entornos de cliente.

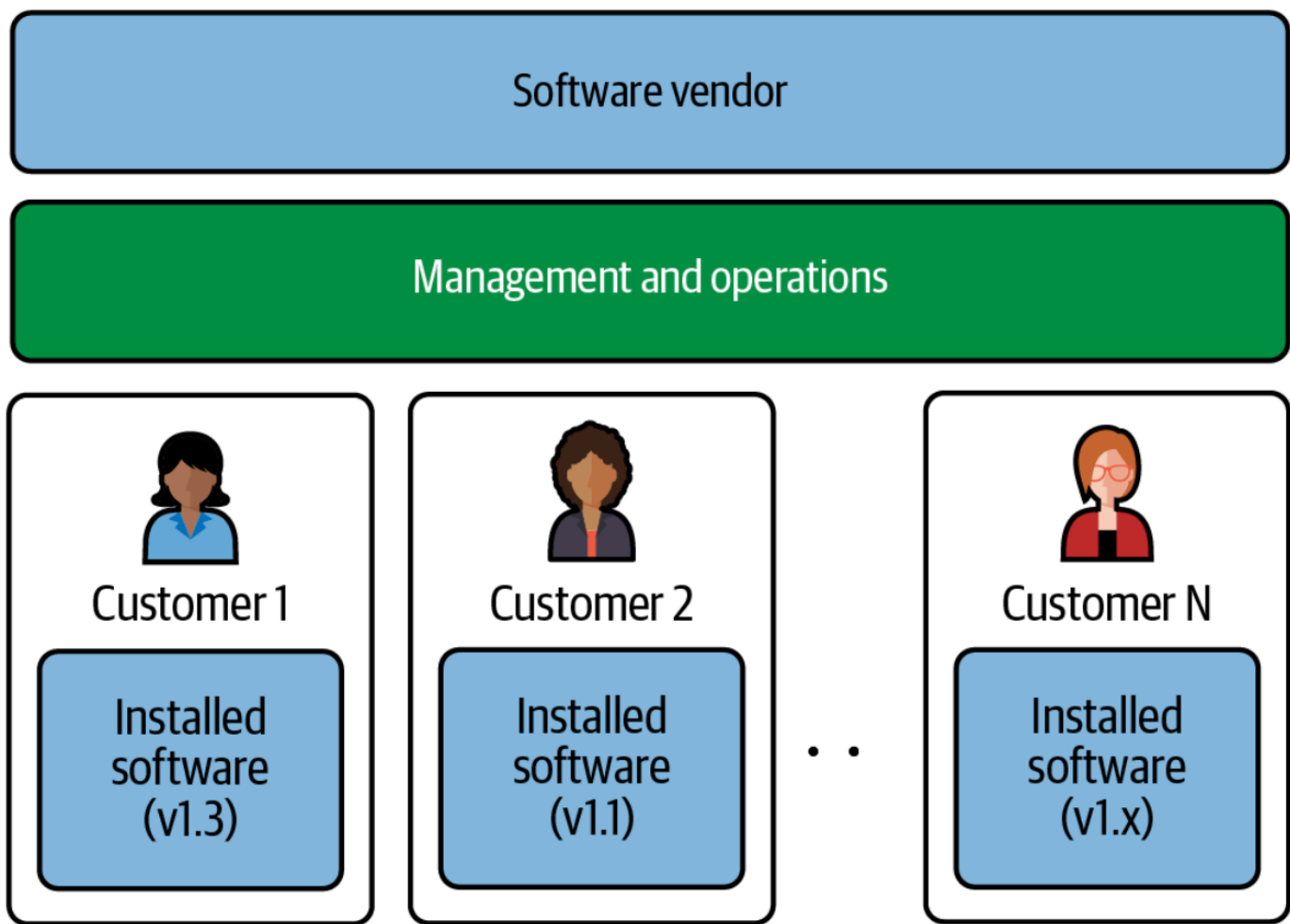


Figure 1-7. A Managed Service Provider (MSP) model

Puede verse por qué algunos pueden equiparar el modelo MSP con SaaS. Después de todo, parece buscar ofrecer una experiencia unificada de gestión y operaciones para todos los clientes. Sin embargo, si volvemos a los principios con los que describimos SaaS, podemos ver diferencias sustanciales entre el modelo MSP y SaaS.

- Una de las mayores diferencias es que se permite a los clientes ejecutar versiones separadas.
- La otra diferencia tiene más que ver con cómo se estructuran y operan los equipos SaaS. Generalmente, en una organización SaaS, intentamos evitar trazar líneas rígidas entre los equipos de operaciones y el resto de la organización. Queremos a operaciones, arquitectos, product owners y los distintos roles del equipo trabajando de cerca para evaluar y refinar continuamente la experiencia de servicio de su oferta.

La clave es comprender claramente los límites entre SaaS y MSP y evitar ver SaaS y MSP como sinónimos.

## En esencia, SaaS es un modelo de negocio

Debería quedar claro que SaaS trata de crear una cultura tecnológica, de negocio y operativa enfocada de lleno en impulsar un conjunto específico de resultados de negocio.

Con SaaS, la estrategia del negocio se centra en crear un servicio que permita a la empresa reaccionar a las necesidades de mercado actuales y emergentes sin perder impulso ni comprometer el crecimiento.

**Cuando ofreces una solución SaaS multi-tenant, las necesidades de la mayoría deben prevalecer sobre las de la minoría.** Quedaron atrás los días de perseguir oportunidades puntuales que requieren soporte dedicado y único a costa del éxito de largo plazo del servicio.

Este cambio de mentalidad influye casi todos los roles en una empresa SaaS.

- Los product owners deben ampliar su visión e incorporar atributos operativos al construir su backlog. La experiencia de onboarding, el time to value, la agilidad—todos son ejemplos de aspectos que deben estar en el radar del product owner. Deben priorizar y valorar estos atributos operativos esenciales para crear un negocio SaaS exitoso.
- Arquitectos, ingenieros y miembros de QA están igualmente influidos por este cambio. Ahora deben pensar más en cómo la solución que diseñan, construyen y prueban cumplirá con las necesidades más dinámicas de la experiencia de su servicio.
- También cambia cómo se comercializa, fija precio, vende y soporta tu oferta SaaS. Este tema de nuevas y responsabilidades solapadas es común en la mayoría de las organizaciones SaaS.

¿Cuáles son los principios centrales que dan forma y guían el modelo de negocio de las empresas SaaS?

- **Agilidad.** Adoptar SaaS es moverse hacia una cultura y mentalidad que enfatiza la velocidad y la eficiencia. Lanzar nuevas versiones, reaccionar a la dinámica del mercado, apuntar a nuevos segmentos de clientes, cambiar modelos de pricing—estas están entre una larga lista de ventajas que las empresas esperan extraer al adoptar un modelo SaaS.
- **Eficiencia operativa.** En un entorno multi-tenant, estamos altamente enfocados en hacer crecer continuamente nuestra base de clientes sin requerir recursos o equipos especializados para incorporar a estos nuevos clientes. Cada parte de la organización puede jugar un rol en construir una oferta operativamente eficiente.
- **Innovación.** La capacidad de moverse más rápido tiene muchos beneficios para las organizaciones SaaS. Las libera y les permite estar más abiertas a experimentar y ajustar su estrategia.
- **Onboarding sin fricción.** Las empresas SaaS deben considerar cuidadosamente cómo se introducen los clientes en sus entornos. Si buscas mantenerte lo más ágil y operativamente eficiente posible, también debes pensar en cómo agilizar el onboarding de clientes.
- **Crecimiento.** Las organizaciones SaaS suelen tener una noción diferente de crecimiento. Invierten en un modelo y una huella organizacional pensados para prosperar con el crecimiento.

## | Construir un servicio no un producto

Tus tenants SaaS tendrán expectativas de servicio similares. Qué tan fácilmente pueden hacer el onboarding de tu solución, cuánto tardan en obtener valor, qué tan rápido se liberan nuevas funcionalidades, qué tan fácil les resulta dar feedback, con qué frecuencia el sistema está caído.

Todas estas son dimensiones de un servicio que deben estar en la mira para los equipos SaaS. **Tener un gran producto no importará si la experiencia general para los clientes no cumple con sus expectativas.**

Esto cobra un significado adicional cuando el software se entrega en un modelo SaaS, donde la única vista que tiene el tenant de tu sistema es la superficie de tu solución SaaS.

Los tenants SaaS no tienen visibilidad de los elementos subyacentes de tu sistema. No piensan en parches, actualizaciones y configuración de infraestructura. **Solo les importa que el servicio brinde una experiencia que les permita maximizar el valor de tu solución.**