Gentileza de IBM

Edición Limitada IBM

APIs FOR DUMMIES®

A Wiley Brand

Aprenda cómo:

- Aplicar el poder de las APIs a los retos del negocio
- Definir la naturaleza del producto de las APIs modernas
- Administrar qué APIs proporcionar o consumir
- Construir una plataforma de tecnología API eficaz

Claus T. Jensen

APIs

FOR DUMMIES®

A Wiley Brand

Edición Limitada IBM

por Claus T. Jensen

FOR DUMMIES®

A Wiley Brand

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

APIs For Dummies®, Edición Limitada IBM

Publicada por
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com.

Copyright © 2015 by John Wiley & Sons, Inc.

Ninguna parte de esta publicación podrá reproducirse, almacenarse en un sistema de recuperación ni transmitirse de ninguna manera ni por ningún medio, ya sea electrónico, mecánico, fotocopia, grabación, escaneo o de otro tipo, excepto según lo permitido por las Secciones 107 o 108 de la Ley de Copyright de EE.UU. de 1976, sin el permiso previo escrito del Editor. Las solicitudes de permiso al Editor deberán dirigirse a Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, o al sitio http://www.wiley.com/go/permissions.

Marcas comerciales: Wiley, For Dummies, el logo Dummies Man, The Dummies Way, Dummies.com, Making Everything Easier y diseños comerciales relacionados son marcas comerciales o marcas comerciales registradas de John Wiley & Sons, Inc. y/o sus filiales en los EE.UU. y otros países, y no pueden usarse sin permiso escrito. IBM y el logo IBM son marcas comerciales registradas de International Business Machines Corporation. Todas las demás marcas comerciales son propiedad de sus respectivos propietarios. John Wiley & Sons, Inc. no está asociado con ningún producto o proveedor mencionado en este libro.

LIMITACIÓN DE RESPONSABILIDAD/EXCLUSIÓN DE GARANTÍA: LA EDITORIAL Y EL AUTOR NO OTORGAN DECLARACIONES NI GARANTÍAS RESPECTO DE LA EXACTITUD O INTEGRIDAD DE LOS CONTENIDOS DE ESTA OBRA Y ESPECÍFICAMENTE DENIEGAN TODA GARANTÍA, INCLUSO, SIN LIMITACIÓN, GARANTÍAS DE ADECUACIÓN PARA UN USO PARTICULAR. NINGUNA GARANTÍA PODRÁ SER CREADA NI EXTENDIDA POR MATERIALES PROMOCIONALES O DE VENTA. EL ASESORAMIENTO Y LAS ESTRATEGIAS CONTENIDAS EN EL PRESENTE PUEDEN NO SER ADECUADOS PARA TODAS LAS SITUACIONES. ESTA OBRA SE VENDE EN EL ENTENDIMIENTO DE QUE LA EDITORIAL NO PROPORCIONA SERVICIOS LEGALES, DE CONTABILIDAD U OTROS SERVICIOS PROFESIONALES. EN CASO DE REQUERIR ASISTENCIA PROFESIONAL, DEBERÁN PROCURARSE LOS SERVICIOS DE UN PROFESIONAL COMPETENTE. NI LA EDITORIAL NI EL AUTOR SERÁN RESPONSABLES DE DAÑOS QUE PUDIERAN SURGIR DE LA PRESENTE OBRA. LA MENCIÓN DE UNA ORGANIZACIÓN O SITIO WEB EN ESTA OBRA COMO CITA Y/O COMO POTENCIAL FUENTE DE INFORMACIÓN ADICIONAL NO SIGNIFICA QUE EL AUTOR O LA EDITORIAL APOYEN LA INFORMACIÓN QUE LA ORGANIZACIÓN O EL SITIO WEB PUEDAN PROPORCIONAR NI LAS RECOMENDACIONES QUE PUEDAN HACER. ASIMISMO, SE ADVIERTE A LOS LECTORES QUE LOS SITIOS DE INTERNET QUE FIGURAN EN ESTA OBRA PUEDEN HABER CAMBIADO O DESAPARECIDO ENTRE LA REDACCIÓN DE ESTA OBRA Y SU LECTURA.

Para ver información general sobre otros productos y servicios, o sobre cómo crear un libro *For Dummies* personalizado para su empresa u organización, contacte a nuestro Departamento de Desarrollo de Negocios en EE.UU. al 877-409-4177, o contacte a info@dummies.biz, o visite www.wiley.com/go/custompub. Para ver información sobre licencias de la marca *For Dummies* para productos o servicios, contacte a: BrandedRights&Licenses@Wiley.com.

ISBN: 978-1-119-04116-0 (pbk); ISBN: 978-1-119-04117-7 (ebk)

10 9 8 7 6 5 4 3 2 1

Agradecimientos de la Editorial

Algunas de las personas que colaboraron para llevar este libro al mercado son:

Editora del proyecto: Carrie A. Johnson
Editora de desarrollo: Kathy Simpson
Gerente editorial: Rev Mengle
Representante de desarrollo de negocios: Sue Blessing
Coordinadora de producción: Melissa Cossell

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Contenidos

Introducción 1

Sobre este libro 1
Íconos usados en este libro 2
Más allá del libro 3

Capítulo 1: La anatomía de una API 5

Lo que una API no es 5
La naturaleza de producto de las APIs 6
De API a economía API 6
Entender lo que los desarrolladores quieren 7

Capítulo 2: Administrar -y no administrar- APIs 11

¿Qué significa administrar una API?..... 11
Responsable de negocio de API 12
Operaciones de TI 13
Diseñador de API 13
La necesidad de gobernanza API 14
Decisiones de proveedor API 14
Decisiones de consumidor API 15
Argumentos a favor de las APIs no administradas 15

Capítulo 3: Descubrir la naturaleza de las buenas APIs . . . 17

Comparar APIs con autos de carrera 17
Argumentos a favor de las APIs oportunistas 18
Pensar en APIs y servicios 19
APIs versus servicios 20
APIs en colaboración con servicios 22
Reconocer el buen diseño de APIs 22

Capítulo 4: Puntos de entrada API 25

Monetice sus datos 26
Libertad para innovar 27
Movilidad en diez minutos 30
Vivir en un mundo híbrido 32
Programe su mundo 35

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 5: Middleware de API y de integración 39

El middleware de API no es simplemente “otro ESB”..... 40
La topología (repetible) de la integración 42
API y el modelo de referencia de economía de servicio 44

Capítulo 6: 10 cosas que hay que saber sobre las APIs 47

La experiencia omnicanal impulsa la necesidad de APIs ... 47
Las APIs son productos de negocio 48
El diseño de negocios es un esfuerzo principio a fin 48
Puede acceder a conocimientos mediante
la instrumentación de APIs 49
No todas las APIs son REST 49
Cada API necesita un responsable de negocios 49
No es necesario hacer versiones de APIs 50
Las APIs son fáciles de controlar con políticas 50
Las APIs tienen un lado oscuro 50

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Introducción

APIs: un tema candente, motivo de enérgicos debates entre empresarios, gerentes de TI y desarrolladores por igual. El mayor interés en el espacio público gira en torno a las APIs públicas abiertas. En cierta medida, no tener una API pública hoy es el equivalente de no tener un website a fines de la década de 1990. Sin embargo, para muchas empresas, las APIs públicas son realmente la última preocupación de negocios. Hay otras preocupaciones que ocupan el primer plano, como: construir soluciones omnicanal, innovar más rápido que la competencia, convertirse en una empresa móvil u operar en un entono de nube híbrida.

Las APIs son motores clave para todas estas iniciativas y más; por ese motivo, suscitan el interés de tantos sectores diferentes. Pero ¿qué es una API, realmente? ¿Por qué es diferente de una interfaz de programación de aplicaciones a la antigua usanza, y porqué merece nuestra atención? En principio, API es la sigla en inglés de *Interfaz de Programación de Aplicaciones*. Sin embargo, el concepto de API ha evolucionado notablemente. Las APIs hoy son bastante diferentes de las interfaces de programación de aplicaciones de antaño. Este libro explica de qué se trata ese cambio

Sobre este libro

Los ecosistemas de negocios modernos deben repensar su enfoque de la innovación e integración. Este libro sirve de guía al lector para aplicar el poder de las APIs a los retos del negocio, que van desde cambios en los modelos de negocio hasta la adopción de todo un universo de dispositivos y sensores. Experimentar el poder de las APIs significa mucho más que simplemente monetizar los datos. Tanto proveedores como consumidores de APIs necesitan tomar decisiones inteligentes, tanto en el área de negocios como de TI.
En este libro, primero se define la naturaleza básica de las APIs modernas y luego se ofrece al lector una guía por una serie de decisiones necesarias: desde qué APIs proporcionar o consumir, hasta cómo construir una plataforma de tecnología API eficaz.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

A lo largo del libro, se mencionan algunas consignas importantes, entre otras:

- ✓ **Piense en una API como un producto.** Representa algo que usted eligió compartir con un público en particular bajo ciertos términos y condiciones.
- ✓ **Pruebe temprano, aprenda rápido, escale fácilmente.** Un enfoque de experimentación de APIs es eficaz en la mayoría de los casos de uso, y las APIs son oportunistas por naturaleza.
- ✓ **Siempre use APIs como las fronteras de su dominio.** Eso le da control y visibilidad del tráfico entrante y saliente.
- ✓ **Las plataformas API se especializan en la tarea.** Se espera que la creación, operación y distribución de APIs sea muy fácil, y 100% segura.
- ✓ **Monetizar las APIs públicas no es el único punto de entrada.** Muchas empresas usan colaboración e a lo largo de sus ecosistemas de TI y de negocios.

Íconos usados en este libro

Como todos los libros Para Dummies, este tiene algunos íconos en los márgenes. He aquí sus significados.

El ícono “Tip” señala información útil.

Toda la información que aparezca acompañada por el ícono “Recuerde” es información que le resultará útil recordar.

Si no le interesan los detalles técnicos, puede pasar por alto todo lo que tenga el ícono “Información Técnica”.

Cuidado cuando vea el ícono “Advertencia”: esté alerta para evitar problemas para el negocio.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Más allá del libro

Esta breve publicación no puede ofrecer todos los detalles sobre un tema. Para ver más información, recomendamos visitar los siguientes sitios:

- ✓ developer.ibm.com/api/blog: El blog de IBM sobre APIs
- ✓ www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248188.html?Open: Material con información detallada sobre middleware de integración
- ✓ ibm.com/soa: Baje el eBook Principios de diseño SOA *For Dummies*

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 1

La anatomía de una API

En este Capítulo

- ✓ Conocer qué no es una API
- ✓ Ver cómo desarrolladores y consumidores quieren usar APIs
- ✓ Clasificar APIs
- ✓ Integrar APIs al modelo de negocio

Mediante APIs modernas, se puede acceder a formas flexibles de proyectar sus capacidades a un público que va más allá del propio equipo. Cuando están bien hechas, las APIs permiten a las empresas innovar más rápido y alcanzar nuevos públicos. Ese es el valor de las APIs, pero ¿qué son realmente las APIs y qué preguntas clave cabe plantear antes de embarcarse en la travesía API? Este Capítulo intenta responder estas preguntas.

Lo que una API no es

A veces, la mejor forma de entender un concepto consiste en explicar lo que no es. En este sentido, algunas cosas que una API no es:

- ✓ **Software:** El software no es una API (pero puede presentarse como API para facilitar el consumo de sus capacidades).
- ✓ **Interfaz de usuario:** Una interfaz de usuario no es una API (pero puede construirse sobre una API).
- ✓ **Servidor:** Un servidor no es una API (pero puede alojar una o más APIs que exponen los datos y las funciones provenientes del servidor).

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

La naturaleza de producto de las APIs

Pensemos en una API como un producto. Se la diseña con cuidado, para que sea atractiva para el consumidor objetivo, para que venda. No importa si ese consumidor realmente paga por la API, o si está afuera de la empresa o forma parte de un equipo diferente dentro de la empresa. Queremos que ese consumidor use nuestra API, porque puede crear valor tanto para el que la desarrolla como para el que la consume cuando la usa.

Comprender la naturaleza de producto de las APIs es fundamental para entender su poder. También las hace muy diferentes de las interfaces de programación de aplicaciones de la antigua escuela. Estas representaban un software que se construía e implementaba. Una API moderna, en cambio, representa un paquete de capacidades atractivas para un público, no importa cuál sea el software que se ejecute en los sistemas de back-end. Así, aunque las APIs modernas no incluyen una interfaz programática definida, están diseñadas a propósito desde la perspectiva del consumidor objetivo.

Como una API es un producto, antes de desarrollarla, usted deberá preguntarse:

- ✓ ¿Quiénes son los receptores esperados?
- ✓ ¿Cómo se puede llegar a esos consumidores?
- ✓ ¿Bajo qué términos y condiciones los consumidores pueden usar esta API?

De API a Economía API

La economía API surge cuando las APIs forman parte del modelo de negocios. Las APIs públicas y de asociados han sido habilitadores estratégicos para varios modelos de negocio, como los utilizados por Twitter y Amazon.

Las APIs de Twitter, por ejemplo, fácilmente llegan a tener diez veces más tráfico que el sitio de Twitter. El modelo de negocios de la compañía deliberadamente se enfoca en la mediación de publicaciones (tweets), y permite que la experiencia del usuario final sea proporcionada por todos aquellos que deseen hacerlo.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Desde el comienzo, Amazon eligió no ser solo un minorista de Internet sino también un portal de comerciantes omnipresente. La plataforma de comerciantes de Amazon está construida deliberadamente sobre APIs que permiten la fácil incorporación de nuevos comerciantes.

Las APIs como facilitadores de red de negocios no son nada nuevo. Durante décadas, los bancos han construido infraestructuras de pago y cámaras compensadoras basadas en APIs bien definidas. Sin embargo, las APIs modernas están construidas explícitamente para un ecosistema abierto (interno o externo), no para redes privadas cerradas. Más aún, los modelos de consumo para APIs están estandarizados, con un enfoque en la facilidad de consumo en lugar de la facilidad de creación (ver "Entender lo que los desarrolladores desean" en este Capítulo).

Algunas personas usan el término APIs de negocio para todas las APIs modernas. El término sin duda es adecuado en el sentido de que las APIs, como productos, deberían formar parte integrante de la estrategia de negocios. Pero es importante destacar que lanzar una API pública o de asociado no es la única forma de hacer que las APIs formen parte del modelo de negocio. Hay muchos casos de uso para APIs consumidas internamente; tal vez el más común de ellos sea la necesidad de proporcionar una experiencia del cliente omnicanal diferenciadora.

Ya sea que su negocio haya sido "pensado para la web" o tenga 100 años de historia, usted vive en la época de la computación en la nube, analítica, móvilidad y social, donde la modalidad omnicanal es lo mínimo que se puede esperar. Para diferenciarse de sus competidores, debe dar a los clientes una experiencia inmediatamente atractiva. Para ofrecer esa experiencia, necesita libertad para experimentar e innovar. Aproveche la oportunidad y pruébelo temprano, aprenda rápido y escale fácilmente.

Entender lo que los desarrolladores quieren

Los desarrolladores quieren usar APIs para innovar y experimentar. Para ellos, la reutilización significa acelerar el tiempo de entrega, compartir significa agilidad y encapsulación significa tener poco que aprender. No están tan interesados en cómo se crean las APIs (ni cuánto cuestan), sino más bien en cuán fáciles son de consumir.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Consumo fácil no se refiere solo a la apariencia de la API. Para el desarrollador especializado en APIs, el consumo fácil también significa que una API debe ser fácil de encontrar, registrar y usar, y debe quedar en claro cuán confiable es la API en cuanto a soluciones críticas para la misión.

Idealmente, un ecosistema API debe estar centrado en la comunidad. Una comunidad API efectiva muestra a los desarrolladores las APIs exactas que tienen a su disposición para sus actuales tareas. El registro autoservicio y las preaprobaciones ya son características reconocidas de las APIs que son visibles para la comunidad. Los atributos sociales de la comunidad permiten a las personas sentir agrado o desagrado por una API particular, y la analítica centrada en consumidores muestra el comportamiento operativo esperado de cualquier API de interés. Estas capacidades históricamente no formaron parte de la gobernanza de TI, pero son atributos centrales de las soluciones para una correcta gestión de APIs. Estas soluciones también agregan valor a los proveedores de APIs, al facilitar la creación de APIs y mejorar el control de su comportamiento en tiempo de ejecución (explicado en más detalle en el Capítulo 2).

Cuatro categorías de APIs

Al comenzar la travesía API, decidir qué APIs construir primero puede ser abrumador. Una buena API debería marcar la diferencia para el negocio. A veces, los casos de uso específicos de industria pueden impulsar la creación de APIs. Pero para entender plenamente qué es una buena API para una empresa en particular, usted debe entender la naturaleza de su situación de negocios exclusiva.

En esta sección, encontrará un enfoque que IBM considera útil en la práctica. Una buena forma de empezar es preguntarse: "¿Qué situaciones de negocio me gustaría mejorar, y cómo puedo hacerlo?" Al responder esta pregunta, probablemente termine eligiendo sus primeras APIs de una de las siguientes cuatro categorías:

- ✓ **APIs de detección:** Las APIs de detección lo ayudan a identificar oportunidades de interactuar con clientes, empleados, asociados y dispositivos. Incluyen mecanismos como detección de ubicación móvil, supervisión de sensores, analítica predictiva y observación humana.
- ✓ **APIs de enriquecimiento:** Las APIs de enriquecimiento mejoran la comprensión de la situación con datos históricos de los sistemas de gestión de relaciones con clientes (CRM), registros de cuentas, análisis demográfico, historias clínicas, etc.
- ✓ **APIs de percepción:** Las APIs de percepción ofrecen contexto dinámico para la situación actual y le permiten comprender qué

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

pasa por la mente de las personas con las que usted está tratando de interactuar. Ejemplos: APIs sociales (gente que comparte planes futuros o intereses actuales) y analítica de sensores (para el estado del sistema general, como consumo de recursos globales o congestión del tránsito). Después de todo, alguien que sale de viaje al Caribe el próximo lunes probablemente esté interesado en temas diferentes de alguien que se quedará en la fría ciudad de Chicago.

✓ **APIs de acción:** Las APIs de acción le permiten actuar casi en tiempo real. Ejemplos de interfaces de acción: notificaciones push, dispositivos instrumentados y sistemas de gestión de tareas humanas.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 2

Administrar –y No Administrar- APIs

En este Capítulo

- Revisar los roles y las tareas de administración de APIs
- Establecer gobernanza de APIs
- Ver cuándo usar APIs no administradas

Presente en muchas conversaciones sobre APIs está la noción de la administración de APIs. Aunque las APIs no son programas de software en el sentido tradicional (vea el Capítulo 1), son elementos importantes de los entornos operativos tanto del área de negocios como del área de TI, y por lo tanto deben administrarse correctamente. En este Capítulo, explicamos cómo hacerlo.

¿Qué Significa Administrar una API?

Para un consumidor de APIs, un gran portal de desarrollador lo es todo. Para un proveedor de APIs, administrar la externalización de APIs y compartir procesos es solo la punta del iceberg (ver la Figura 2-1). Debajo del nivel de agua están las preocupaciones de negocios y de TI que hacen que las APIs sean prácticas de crear, implementar y operar. Estas preocupaciones incluyen mapeo de datos, seguridad, regulación de la velocidad, supervisión y administración de versiones.

Una API administrada no solo tiene una interfaz bien definida y un público objetivo definido sino que también está bajo controles de negocios y de TI correctamente implementados. Distintos grupos tienen papeles específicos que desempeñar en la gestión de APIs, como se ve en esta sección.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Figura 2-1: Administrar APIs exige más que diseño y externalización de APIs.

Para maximizar la eficacia, los tres roles abordados en este Capítulo — responsable del negocio, operaciones de TI y diseñador de API — necesitan sus propios diseños de experiencia de usuario. Estas preocupaciones tienen bastantes diferencias entre sí, por lo cual las herramientas de uno de los roles resultan ineficientes para otro rol.

Responsable de negocio de API

El responsable de negocio de la API decide lo siguiente:

- ✓ Los planes (términos y condiciones) que rigen el consumo de APIs
- ✓ Las comunidades con las que se compartirá la API
- ✓ Si la API tiene éxito en sus objetivos (caso contrario, el modelo de negocios necesita ajustes)

Todo esto puede hacerse sin cambiar la definición ni implementación de la API en ninguno de sus aspectos. Para ver más información sobre el rol del Responsable del Negocio, consulte “La necesidad de gobernanza API” más adelante en este Capítulo.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

Operaciones de TI

El sector de Operaciones de TI debe asegurar ciertas características, todas las cuales también pueden crearse sin cambiar la definición ni la implementación de la API en ningún aspecto. Las características son:

- ✓ El tiempo de ejecución (runtime) que aloja la API puede ser operado en forma segura y robusta.
- ✓ La API está correctamente autenticada, y la autorización ha sido otorgada para quien la usa.
- ✓ El tráfico de la API está optimizado y priorizado de acuerdo con necesidades de negocio.

Hay una diferencia fundamental entre lo que el responsable de la API vende en términos de plan de acceso a la API y lo que la infraestructura informática subyacente puede ofrecer. Contar con la capacidad de reserva necesaria para poder responder a todo el potencial de tráfico asociado a los planes API vendidos puede ser muy costoso.

Para prevenir costos prohibitivos de tiempo de ejecución, asegúrese de que su runtime de API sea altamente escalable (para que la carga real importe menos) o bien aplique regulación del tráfico cuando la carga actual supere la capacidad disponible, para suavizar picos de tráfico. Estas capacidades deben estar disponibles en el runtime de su API.

Para ver más información sobre el rol de Operaciones de TI, consulte “La necesidad de gobernanza API” más adelante en este Capítulo.

Diseñador de API

La persona que tiene a su cargo diseñar APIs crea e implementa físicamente la API. Debe hacer lo siguiente:

- ✓ Definir la interfaz API
- ✓ Descubrir puntos terminales de back-end que pueden proporcionar los datos o la función requerida para implementar la API
- ✓ Configurar el mapeo entre la interfaz de la API y las fuentes de datos o funciones de back-end.

El diseñador de API debe ser capaz de realizar estas tareas sin desarrollar mucho código. Cuando la creación de una API demanda mucho código y deja de ser una configuración dinámica, el ritmo de innovación inevitablemente

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

se reduce, incluso para los equipos de desarrollo. La distinción entre configurar una API y desarrollar los datos o la función de back-end que representa esa API es fundamental para el concepto API. Como se señaló en el Capítulo 1, las APIs modernas no son software; las APIs modernas son una forma flexible de proyectar capacidades a públicos más allá del propio equipo.

La necesidad de gobernanza API

Un mito común en los círculos API es que la gobernanza hace todo más engorroso. Pero la gobernanza implica buenas decisiones: asegurarse de que las personas correctas tomen las decisiones correctas, en el momento oportuno y por los motivos correctos, sobre la base de la información correcta. Por lo tanto, si una API es importante para su organización, será conveniente tomar buenas decisiones al respecto. Como proveedor de APIs o consumidor de APIs, usted tiene varias decisiones que tomar al respecto.

Esta gobernanza difiere un poco del tipo que se aplica rutinariamente como parte de un ciclo de vida de entrega de software. Sin embargo, es importante.

Decisiones de proveedor API

Decidir quién puede usar la API, según qué términos y condiciones, es el trabajo del responsable de negocio a cargo de la API. Esta decisión operativa de negocios se aplica a todas las APIs, ya sean las APIs que el equipo de desarrollo móvil usa para construir apps móviles, las APIs usadas para integrar sistemas entre diversas líneas de negocio o las APIs usadas por consumidores externos, de modo que probablemente usted desee tomar distintas decisiones para cada uno de estos públicos en términos de qué APIs pueden usar y en qué condiciones.

El área de operaciones de TI también debe tomar las decisiones de proveedor de APIs correctas –en general, en forma de políticas de seguridad y tráfico– para proteger la infraestructura del uso indebido y la sobrecarga.

El régimen de gobernanza debe ser muy liviano, y las decisiones deben ser de tipo operativo, en lugar de decisiones típicas de ciclo de vida tomadas durante un ciclo de vida de entrega de software. Si no es posible adoptar y hacer cumplir las decisiones correctas fácilmente, la naturaleza abierta y dinámica de las APIs se verá comprometida

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

(recuerde, las buenas implementaciones API son configuraciones, no código). Las decisiones de negocio y de TI forman parte de una disciplina de buena administración de APIs y deberían estar respaldadas por la plataforma API elegida. (Para ver más información sobre middleware de API, consulte el Capítulo 5.)

Decisiones de consumidor API

Los consumidores de APIs también deben tomar buenas decisiones. En particular, deben decidir qué APIs están dispuestos a usar para qué fines y luego hacer las siguientes preguntas sobre cada API:

- ✓ ¿Cuál es el modelo de pago por uso de la API? ¿Es aceptable para los objetivos buscados?
- ✓ ¿Necesitará un proxy corporativo delante de la API para administrar licencias, pago, etc., o cada desarrollador se registrará de forma independiente?
- ✓ ¿Es segura y confiable la API para fines críticos para la misión? Los registros históricos de cuál ha sido el comportamiento de las APIs en el tiempo pueden aumentar la seguridad de los consumidores a la hora de usar las APIs.

Cuando las APIs que son consumidas son APIs propias, estas decisiones son bastante directas, y tratan sobre todo de diseño general de negocios. Cuando las APIs son APIs de terceros, las decisiones se vuelven más complejas. Al fin de cuentas, no se puede delegar la experiencia del usuario final y la responsabilidad por mantener la integridad del negocio. Usted necesita tener a alguien en su organización que sea responsable de la experiencia del usuario final y que adopte las decisiones correctas sobre qué APIs es bueno consumir dentro del modelo de entrega.

Argumentos a favor de las APIs no administradas

Todos saben que las APIs deben ser alimentadas por una solución de administración de API, ¿correcto? No tan rápido. No todas las APIs necesariamente deben ser administradas. En este Capítulo, di una definición de la buena gestión de APIs, ¿pero qué significa una API no administrada?

Las principales diferencias entre APIs administradas y no administradas se resumen a continuación:

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

- ✓ Una API no administrada puede tener un público objetivo, Pero este público objetivo rara vez está definido en forma precisa y menos aún se cumple. Si un usuario tiene acceso de red a la API, generalmente puede invocarla.
- ✓ Una API no administrada no tiene controles de negocio y de TI aplicados en forma independiente. Todo control se proporciona a través de una lógica en la implementación de APIs, generalmente en forma de código.

En otras palabras, las APIs no administradas todavía tienen una interfaz bien definida, pero no hay forma de aplicar controles en su comportamiento de tiempo de ejecución ni siquiera en quién puede usarlas. Entonces, ¿por qué querer una API no administrada?

Si una API forma parte directa de su modelo de negocio, probablemente no desee que quede sin administrar. Habiendo hecho esta aclaración, daré algunos ejemplos de situaciones en las que tener APIs no administradas es apropiado o incluso inevitable:

- ✓ Un dispositivo o sensor tiene una API definida como parte de su realidad física, como un termostato de hogar que puede ser programado en forma remota o un Fitbit (un dispositivo de vestir que monitorea la actividad física) que sincroniza datos con una computadora por una interfaz definida.
- ✓ Un sistema de software existente –tal vez un sistema estándar, como SAP o un mainframe con una interfaz REST nativa — expone una micro API.
- ✓ La API está dentro de su propio dominio y todo lo que realmente necesita es conectividad para acceder a ella.

Las APIs no administradas pueden ser recursos importantes en muchos ecosistemas, pues pueden hacer que funciones y datos clave estén disponibles en forma uniforme. El modelo de consumo uniforme es el motivo por el cual es conveniente pensar en estas interfaces como APIs. A menudo, incluso es conveniente catalogar todas las APIs no administradas que están a su disposición, para que sean fáciles de encontrar y lo más simples de usar posible en un modelo de programación particular.

Convertir todo en una API, desde el punto de vista del consumidor, es la forma más fácil y eficaz de innovar y colaborar en un entorno híbrido. Esto significa que pensar en APIs no implica solo administración de APIs y APIs administradas. Pensar en APIs debe ser parte de una estrategia de integración más amplia para convertir a su empresa en un motor de innovación.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 3

Descubrir la naturaleza de las buenas APIs

En este Capítulo

- Entender por qué las APIs son como autos de carrera
- Usar APIs oportunistas
- Combinar APIs con arquitectura orientada a servicios (SOA)
- Comprender el buen diseño API

Rapidez en la innovación es algo que se puede lograr con un buen diseño, lo cual, para una API dada, incluye su interfaz y características técnicas. Es importante destacar que el buen diseño tiene que ver con qué APIs proporcionar y cuándo hacerlo. Si todo lo que intenta hacer es ofrecer un puñado de APIs públicas estables, tal vez la pregunta de cuáles no sea tan complicada. Pero ¿qué pasa si está tratando de hacer eso, además de crear un ecosistema de asociados, y usar APIs para promover la innovación interna? Hay muchos tipos diferentes de APIs y usos para ellas. En este Capítulo, aprenderá cómo es una buena API.

Comparar APIs con autos de carrera

Se puede hacer una analogía entre las APIs y la forma en que los equipos de autos Fórmula 1 construyen y actualizan los autos de carrera (vea la Figura 3-1). En la Fórmula 1, cada auto que corre una carrera es un prototipo. Ningún equipo hace el mismo auto para dos carreras consecutivas. Un auto de carrera se construye con componentes rápidamente reemplazables con interfaces bien definidas, y el auto mismo está instrumentado con controles y analítica incorporada.

Aunque partes del auto pueden mantenerse estables durante la temporada, algún componente siempre se optimiza según las lecciones aprendidas de la carrera anterior.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

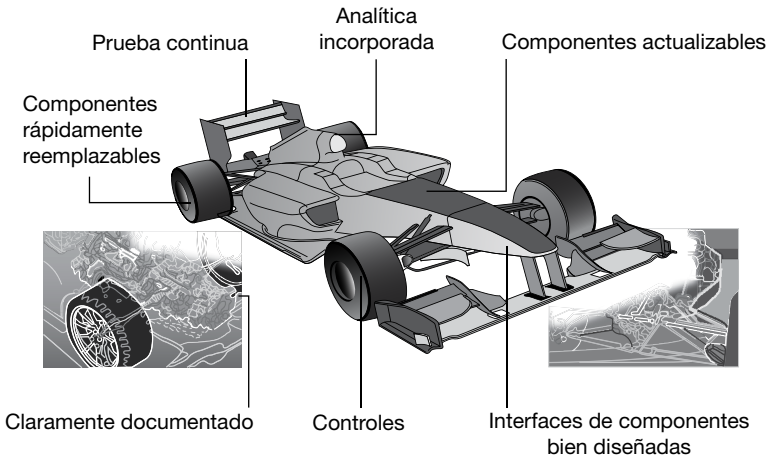

Figura 3-1: Los automóviles de carrera y las APIs deben construirse según los mismos principios de diseño.

Las empresas modernas se parecen bastante a los equipos de Fórmula 1, siempre tratando de optimizar el modelo de negocio y buscando el equilibrio justo entre cambio y estabilidad. Las APIs son una forma de aprovechar la experimentación para ventaja de la empresa. “Pruebe temprano, aprenda rápido y escale fácilmente” es un buen principio para aplicar al mundo de las APIs.

Argumentos a favor de las APIs oportunistas

¿Las APIs siempre deben diseñarse para ser reutilizables? La reusabilidad implica la estabilidad en un período relativamente largo, y la estabilidad es apropiada si una API se utilizará para la integración de asociados o se expondrá externamente como parte de su modelo de negocio. Pero si la API se crea simplemente para mejorar la colaboración entre un equipo de desarrollo móvil y el equipo que mantiene un sistema de registro de back-end, la reusabilidad puede no ser deseable ni apropiada.

Una API debe ser atractiva de usar y, para un desarrollador, debe ser más rápido y ágil usar esa API que crear el código de una solución diferente. Si las necesidades de un desarrollador móvil cambian rápidamente, las APIs deben cambiar igual de rápido para seguir el ritmo.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Esta situación plantea el caso de las APIs oportunistas: APIs que se crean y cambian rápidamente, definidas para satisfacer una necesidad específica del consumidor. Para ver más información sobre el punto de vista del desarrollador, vea el Capítulo 1.

Nada en el concepto de una API requiere que necesariamente sea reutilizable o estable en el tiempo. La importancia de la reusabilidad y estabilidad dependen únicamente de cuál es su objetivo de negocio para tener la API. Si ese objetivo implica el cambio rápido, las APIs oportunistas son apropiadas.

Para proporcionar APIs oportunistas, crear y mantener las APIs tiene que ser fácil y barato. De lo contrario, el costo del cambio oportunista se vuelve inviablemente alto. De modo que el software de gestión de APIs se concentra en este reto.

Las buenas soluciones de administración de APIs generan APIs a través de la configuración y no del código, y la tarea de crear o cambiar una API suele completarse en minutos. La naturaleza de una API de fácil administración es simplemente que es definida y controlada por la configuración. Independientemente del costo de creación y mantenimiento, hay un valor significativo en administrar correctamente incluso las APIs oportunistas (vea el Capítulo 2).

Administrar APIs oportunistas se traduce en los siguientes beneficios:

- ✓ Definición y aplicación de controles de negocio y TI
- ✓ Conocimiento global del desempeño de la API desde la perspectiva de negocios
- ✓ Flexibilidad de operaciones de TI para mover y escalar cargas de trabajo API

Estos beneficios son aspectos importantes de la estrategia de probar temprano, aprender rápido y escalar fácilmente, y resultan críticos para optimizar el cambio en un mundo de oportunidades e innovación.

Pensar en APIs y servicios

La arquitectura orientada a servicios (SOA) es una tecnología vigente desde hace una década. Las APIs modernas son más recientes. Ambos enfoques de la integración tienen sus proponentes y abordan tanto problemas de negocio como de TI. ¿Cuál es la diferencia real entre ambos enfoques? ¿Se debe elegir entre ellos?

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

APIs versus servicios

El concepto central de SOA es la noción de servicio. El Open Group, por ejemplo, define un servicio como “una representación lógica de una actividad repetible con un resultado especificado”. Los servicios son autocontenidos y opacos para sus consumidores. Tienen contratos de interacción bien definidos. Desde el punto de vista técnico, estas características también se aplican a cualquier API bien definida, de modo que técnicamente, una API es también un servicio.

En ese caso, ¿las APIs son otra forma de referirse a los servicios? Bien, hay una diferencia importante entre los servicios y las APIs, y ese es el objetivo de su diseño (consulte la Figura 3-2). Las APIs siempre están diseñadas para ser atractivas para el consumidor al que apuntan, y cambian a medida que cambian las necesidades del consumidor. Los servicios, por el contrario, están generalmente diseñados con el costo global y la estabilidad como principales preocupaciones. En la analogía del automovilismo, la API es el auto de carrera diseñado para tener buena apariencia y consumo, mientras que el servicio es el auto corriente, diseñado teniendo en cuenta el costo y la producción masiva.

Figura 3-2: APIs y servicios abordan cuestiones diferentes

El Capítulo 1 describe la naturaleza de producto de las APIs y explica que deben destinarse a las necesidades de un grupo particular de consumidores. Para un consumidor, usar una API es una cuestión de velocidad, agilidad y tener poco que aprender. Estos criterios de diseño son las diferencias fundamentales entre APIs y la noción clásica de servicios, por lo menos desde la perspectiva del proveedor de servicio:

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

- ✓ Para un proveedor de servicios, la reutilización tiene que ver con el esfuerzo que implica la entrega de la API. Para un consumidor de APIs, la reutilización tiene que ver con el tiempo de entrega de su software, independientemente de cuál sea el costo para proporcionar APIs consumidas como parte de ese software.
- ✓ Para un proveedor de servicios, compartir tiene que ver con eficacia. Para un consumidor de APIs, compartir tiene que ver con agilidad (si no es ágil, la API no será utilizada).
- ✓ Para un proveedor de servicios, la encapsulación significa tener poco que cambiar. Para un consumidor de APIs, la encapsulación significa tener poco que aprender (si es compleja, la API no será utilizada).

¿Cuán a menudo ha visto una iniciativa SOA demorada por conflictos entre proveedores de servicio y consumidores de servicio acerca de lo que constituye una buena interfaz de servicios? Por un lado, un desarrollador móvil solo desea que sea simple para su app particular. Por otro lado, el equipo de back-end desea que todos usen el mismo servicio estandarizado y modelo de datos. En lugar de forzar una resolución de este conflicto, ¿existe la forma de satisfacer ambas necesidades sin incurrir en costos prohibitivos?

Existe una analogía histórica en la evolución de bases de datos. Las primeras generaciones de bases de datos se enfocaban exclusivamente en lo interno: las tablas, esquemas y procedimientos de datos. Sin embargo, rápidamente fue obvio que existía la necesidad de exponer subconjuntos de datos controlados en una forma en particular, a la medida de un grupo particular de consumidores de datos, y surgió el concepto de la visualización de datos como capacidad clave en la mayoría de las bases de datos modernas, un proxy liviano sobre el dominio de datos representado por los aspectos internos de la base.

Las APIs son visualizaciones (proxy) controladas de los datos y las capacidades de un dominio, optimizadas para consumidores de APIs. En tanto que sea barato crear y mantener APIs proxy, puede usarlas para entregar un dominio de múltiples maneras, optimizado para cada grupo de consumidores de API. (Después de todo, probablemente quiera dar a asociados externos una visión diferente de sus capacidades que la visión que tienen sus desarrolladores internos.)

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

APIs en colaboración con servicios

La arquitectura SOA surgió como forma de proteger a los consumidores de servicios de los cambios en el back-end. Pero, ¿quién protege a los proveedores de servicio de la merma en necesidades cambiantes de las soluciones de front-end omnicanal? Aplicar APIs y servicios en forma conjunta genera un remanso de calma en un huracán de cambios. Los servicios son la forma por la cual los proveedores codifican las capacidades base de sus dominios. Las APIs son la forma en que dichas capacidades (servicios) se reempaquetan, se convierten en productos y comparten en forma fácil de usar. Las APIs y los servicios son complementarios y no contradictorios, y cuando se aplican juntos, aumentan sustancialmente la eficacia general de la innovación empresarial.

Reconocer el Buen Diseño API

No hay duda de que el diseño técnico de las APIs es importante, pero también varía ampliamente con las opciones de tecnología que se hacen para el diseño y la implementación de una API en particular. Por ejemplo, lo que constituye una interfaz REST bien diseñada es muy distinto de lo que constituye una interfaz SOA bien diseñada. Se han escrito libros enteros sobre diseño de interfaces, con un nivel de detalle que va más allá de lo que este libro puede ofrecer. Basta con decir que el diseño de interfaz de APIs en general es bien comprendido, como muestran estos ejemplos:

- ✓ Las interfaces REST se basan en recursos. El aspecto más importante del diseño de interfaz es la estructura URI que permite a un consumidor navegar el gráfico de objeto representado por la API
- ✓ Las interfaces SOAP están basadas en métodos. Los aspectos más importantes del diseño de interfaz son el conjunto de métodos con soporte y las estructuras de datos de cada método
- ✓ Las interfaces MQTT se basan en eventos. Los aspectos más importantes del diseño de interfaz son el conjunto de eventos (emitidos o recibidos) y los mensajes de evento asociados

No todas las APIs son REST. En general, las interfaces REST son excelentes para consumo humano y son la preferencia actual de los desarrolladores móviles. Pero las interfaces REST tienden a intercambiar mucha información y aunque son extensibles, no llevan estructuras de datos complejas con mucho texto.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Las interfaces SOAP son excelentes para la integración sistema a sistema, y los equipos de operaciones de TI las prefieren debido a que no intercambian tanta información y sus estructuras de datos son más precisas. Las interfaces MQTT se prefieren por comunicarse con la Internet de las Cosas, en la cual el ancho de banda y la vida de batería son preocupaciones clave, y la entrega garantizada puede ser la diferencia entre prevenir accidentes y dejarlos suceder inadvertidamente.

Para entender el valor de una API se necesita algo más que la denominada cadena de valor de API

Gran parte de la popularidad que tienen las APIs se centra en la cadena de valor de la API ilustrada en esta figura; la forma en que las APIs extienden un modelo de negocio en un ecosistema abierto.

Lamentablemente, la mayoría de los ejemplos analizados en la industria en su conjunto se centran exclusivamente en APIs públicas, y este no es el único caso de uso para las APIs. Para peor, la imagen de cadena de valor genérica no es interesante desde la perspectiva de qué APIs proporcionar y por qué.

El motivo es que no dice nada sobre los distintos tipos de objetivos de negocio que llevarían a uno a siquiera considerar una API. El Capítulo 4 aborda esa brecha conceptual, al definir puntos de entrada API típicos con los criterios de decisión relevantes.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 4

Puntos de entrada API

En este Capítulo

- Monetizar los datos
- Aprovechar la innovación
- Acelerar la transición a móvil
- Volverse híbrido
- Programar todo

¿Qué significa, realmente "pensar en APIs"? Algunos piensan simplemente en la denominada cadena de valor API (vea el Capítulo 3) pero, ¿eso es todo? ¿Eso explica tanto interés? Creo que hay otros elementos a considerar para comprender las APIs. Como se comentó en el Capítulo 2, deben tomarse decisiones sobre el equilibrio entre APIs empresariales y oportunistas. Hay decisiones que tomar respecto de los términos y condiciones por los cuales se compartirán las APIs, y hay decisiones respecto de cómo mapear las APIs requeridas con los activos existentes durante la implementación de APIs. Todas estas decisiones dependen del resultado de negocio deseado.

La parte clave de la frase "pensar en APIs" es la primera palabra: Pensar. Pensar en lo que uno está tratando de lograr para el negocio, qué público quiere atraer, qué tipos de APIs se necesitan para atraer a ese público, y cómo curar (o adecuar) sus datos y funciones como activos de aplicaciones (servicios) para dar soporte a esas APIs que se deben proporcionar. En el proceso, no olvide pensar en qué APIs va a consumir usted mismo y de quién. Pensar en las APIs no es solo ser un proveedor de APIs; muchas organizaciones consumen varias veces la cantidad de APIs que proporcionan. Estas consideraciones son los elementos clave de una estrategia API eficaz.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Este Capítulo define cinco puntos de entrada API que, en mi experiencia, son representativos de las agendas de negocio y de TI que impulsan el pensamiento API. Una empresa puede tener múltiples agendas al mismo tiempo, pero cada una lleva a distintos criterios de decisiones para la adopción de APIs. Puede leer el Capítulo de principio a fin o elegir el punto de entrada que más se ajusta a las necesidades de negocio.

Monetice sus datos

Monetizar sus datos tiene que ver con externalizar conocimientos o funciones de manera de atraer a terceros a usar esos conocimientos o funciones. La monetización puede adquirir distintas formas. El ejemplo más obvio es cuando el tercero paga para usar la API desarrollada por usted. En otros casos, debe ser usted quien en realidad paga al consumidor de la API a cambio de un mayor alcance de negocios y un ecosistema más fuerte. O puede realizar la incorporación de asociados a través de APIs sin ningún pago directo involucrado. Los objetivos de negocio clave para la monetización son hacer pivote en el negocio, cambiar la cadena de valor y aumentar su alcance e influencia.

El éxito con este punto de entrada probablemente requiera una planificación cuidadosa. Si bien usted puede y debe hacer alguna experimentación, “estilo oportunista”, el producto final debe ser un conjunto de APIs empresariales estables que un tercero pueda usar durante un período prolongado.

Pensando en las APIs en el contexto de monetizar los datos, he aquí algunas pautas:

- ✓ **Objetivo:** El resultado deseado es monetario o se basa en un valor no monetario, como aumentar su influencia.
- ✓ **Público:** El público es inevitablemente un tercero. Los casos típicos son asociados y desarrolladores externos (no desarrolladores de planta ni contratados en su organización). Tratar una línea de negocio diferente dentro de su empresa como un asociado o tercero no es infrecuente, en particular cuando las distintas líneas de negocio se tratan como entidades económicamente independientes.
- ✓ **APIs a proporcionar:** Las APIs necesarias para interesar al público deben proporcionar el valor que usted quiere vender. Su elección debe ser cuidadosamente pensada, no solo en términos del valor raíz brindado, sino también en términos de la forma que hace el consumo atractivo.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

- ✓ **Términos y condiciones de APIs:** No olvide incluir en sus condiciones los términos y condiciones que habilitarán el consumo de APIs, tales como freemium, pago por consumo o contrato prepago.
- ✓ **Implementación de APIs:** La forma de adecuar (curar) los datos y funciones para implementar las APIs se traduce en calidad y confiabilidad. Algunos dicen que el costo de la implementación es el factor más importante, pero el costo de implementación no define la adecuación de la estrategia de monetización de datos. Lo que decide la viabilidad a largo plazo es si los consumidores finales experimentan tanto valor como confiabilidad.
- ✓ **Consumir las APIs de otro:** En algunos casos, también debe considerar qué APIs consumir. Aunque usted genere valor principalmente proporcionando APIs para que otros consuman, implementar esas APIs puede implicar la creación de compuestos de mayor nivel de APIs existentes, la mayoría de las veces combinando esas APIs existentes con algo que sea exclusivamente suyo.

El punto de entrada de monetización de datos puede ser sobre el que más oyó hablar, pero no es el más común. En la actualidad, una gran mayoría de iniciativas APIs son para casos de uso internos, y esto puede ser válido cuando los intercambios de APIs públicas sean totalmente maduros.

Libertad para innovar

La libertad para innovar es el imperativo más importante para muchas empresas hoy. Pruebe temprano, aprenda rápido, escale fácilmente: características clave de una empresa dinámica e interactiva. El enfoque de este Punto de entrada API es perseguir oportunidades de negocio agresivamente y hacer de la innovación un proceso de aprendizaje mediante el siguiente modelo:

- ✓ Todo es un prototipo hasta que se prueba en la práctica.
- ✓ Descubrir que algo no funcionó no equivale a un fracaso; es aprendizaje.
- ✓ Quedarse quieto equivale al fracaso seguro.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

Tal como se comentó en el Capítulo 3, el papel de las APIs es proporcionar un remanso en la tormenta del cambio. Esta función tiene dos aspectos:

- ✓ Entregar rápidamente lo que el consumidor de API que experimenta necesita y eliminar lo que ya no es necesario
- ✓ Proteger al proveedor de la merma de usuarios (vea en el Capítulo 3 porqué definir e implementar nuevas "visualizaciones" de APIs sobre activos existentes debería ser fácil y barato)

El punto de entrada de 'Libertad para Innovar' tal vez no sea tan glamoroso como 'Monetice sus Datos' (sección anterior) pero es por lejos el caso de uso de APIs que IBM ve con más frecuencia en las grandes empresas. Poder componer nuevas capacidades innovadoras, internas y/o externas, sin excederse en los costos, es algo que todos se esfuerzan por lograr. Para acelerar la innovación, se necesita una combinación de planificación cuidadosa y reacción oportunista.

Pensando en las APIs en el contexto de libertad para innovar, aquí hay algunas guías:

- ✓ **Objetivo:** El resultado deseado es descubrir rápidamente qué funciona y qué se diferencia en el mercado, y luego ampliar la escala de los éxitos.

  Incluso cuando crea que sabe lo que el mercado necesita, una actitud de aprendizaje puede ofrecer una contrastación saludable con la realidad. Es fácil confundir síntoma con causa cuando uno se enfrenta a dinámicas complejas del mercado.
- ✓ **Público:** El público comprende principalmente a desarrolladores internos, de planta o tercerizados. A veces se contratan agencias externas para que colaboren en un esfuerzo de innovación. En estos casos, se celebran acuerdos más formales sobre qué APIs proveer y cuándo hacerlo, pero aun así es mejor mantener algún comportamiento oportunista para mantener la velocidad de aprendizaje. Las negociaciones contractuales son el anatema de la rapidez en la innovación.
- ✓ **APIs a proporcionar:** Las APIs que se necesitan para atraer al público son una combinación de APIs empresariales predefinidas y APIs oportunistas (vea el Capítulo 3) como consecuencia de la necesidad de una app innovadora. Incorporar algunas APIs empresariales para exponer datos estratégicos en forma accesible puede ser la plataforma de despegue no solo para el desarrollo sino también para la generación de ideas.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Mantenga las APIs predefinidas reducidas y simples. La exposición de toda la estructura de datos de un sistema de back-end de clientes probablemente no sea fácil de consumir para un desarrollador de canales innovadores.

- ✓ **Términos y condiciones de APIs:** Los términos y las condiciones que rigen el consumo de APIs siguen siendo importantes, no en términos de pago, sino en cuanto a la protección de la seguridad y estabilidad de los sistemas de back-end. Después de todo, la innovación es impredecible.
- ✓ **Implementación de APIs:** La forma en que adecua los datos y las funciones requeridas para implementar APIs difiere para las APIs preplaneadas y para las APIs oportunistas impulsadas por la demanda:
    - Para APIs preplaneadas, debe decidir cuidadosamente qué segmentos de datos exponer a nivel de la organización. Además, su implementación (en general servicios empresariales con proxy) debe tener en cuenta el costo final del tiempo de ejecución. Las APIs preplaneadas serán utilizadas en formas no predecibles, de modo que el costo de tiempo de ejecución no debe ser un factor que impida ese uso.
    - Para APIs oportunistas, las consideraciones más importantes son la velocidad y el costo de desarrollo. Si hay que tomar algún atajo para lanzar la API hoy en lugar de la semana próxima, hágalo siempre que tenga un plan viable para emprolijar la implementación de la API cuando esta se convierta en un éxito.

      Muchas APIs oportunistas pueden no vivir mucho tiempo. Si resulta que necesita algo diferente, descarte la API oportunista y comience de nuevo en la próxima iteración del proceso de aprendizaje.

- ✓ **Consumir las APIs de otro:** Considerar qué APIs de terceros consumir es importante para este punto de entrada. Como ejemplo simple, construir apps móviles sociales es difícil sin acceder a APIs de servicios sociales públicos conocidos, como Twitter, Facebook y LinkedIn. Estas APIs de terceros deben ser parte del catálogo de APIs que usted proporciona a sus desarrolladores internos; no deberían tener que ir a un sitio externo para encontrar las cosas ellos mismos. Incluso puede ser conveniente adecuar (curar) las APIs de terceros para crear su propia versión, más simple, ya que algunas de las APIs públicas son bastante complejas en su versión nativa.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

La innovación nunca es fácil, pero puede ser asistida de diversas maneras. El uso correcto de APIs puede hacer que el back-end corporativo se integre al motor de innovación en lugar de ser una carga demasiado pesada. Las empresas con historia tienen la ventaja de tener más activos para exponer como APIs. Pero incluso para las nuevas empresas, el enfoque impulsado por APIs para la implementación de soluciones innovadoras ofrece más flexibilidad en cuanto a la extracción de datos y función. El uso de APIs libera a los desarrolladores de canales para que puedan concentrarse en la experiencia del usuario en lugar de la integración. El uso de APIs también promueve una experiencia omnicanal, ya que los datos y las funciones detrás de la API por definición son remotos y pueden accederse desde cualquier canal o aplicación en la empresa.

Movilidad en diez minutos

Este punto de entrada API se relaciona estrechamente con el punto de entrada libertad para innovar (vea la sección anterior) y puede ser considerado una variante extrema. La diferencia es que para este punto de entrada, todo tiene que ver con dar soporte oportuno, brindando lo que su equipo de desarrollo móvil necesita aquí y ahora.

Las empresas hoy necesitan crear muchas apps pequeñas y autocontenidas, en lugar de los tradicionales portales amplios. Creo firmemente que los consumidores móviles prefieren orquestar sus propias experiencias omnicanal en lugar de tener a alguien que predefina el proceso. Lo veo en mi propio comportamiento, donde rara vez dedico más de un par de minutos en una app móvil antes de pasar a otra cosa.

El punto de entrada ‘movilidad en diez minutos’ implica la colaboración entre los responsables de back-end y los equipos de desarrollo móvil. Esos equipos de desarrollo móvil pueden ser internos o externos, pero en todo caso necesitan apalancar datos y funciones existentes fácilmente, para proporcionar una experiencia atractiva. Poder ver precios de acciones es útil, por ejemplo, pero poder hacer algo con ellos en el contexto de la propia cartera de inversión es el verdadero diferenciador.

El enfoque de este punto de entrada es la simplicidad controlada. Oculte la complejidad, simplifique lo que el desarrollador móvil ve y consume, y al mismo tiempo proporcione control operativo de TI y de negocios apropiado. Además, el proceso debe ser rápido, para no bajar el ritmo de la innovación móvil.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Por diversión, algunos desarrolladores de IBM probaron si era posible tomar datos del mainframe y ponerlos en un dispositivo móvil, usando el enfoque API, en 10 a 15 minutos. De hecho, ¡fue posible! No tenían un diseño bonito, pero comprobó que la complejidad de la lógica de integración puede eliminarse de la ecuación con el uso correcto de una API y tecnología de integración.

'Movilidad en diez minutos' es un concepto de innovación oportunista. Pensando en las APIs en ese contexto, he aquí algunas guías:

- ✓ **Objetivo:** El resultado deseado es el soporte inmediato para las necesidades de sus equipos de desarrollo móvil. El equipo de desarrollo móvil es el responsable de definir qué datos requiere la experiencia del usuario final, y el responsable de back-end construye las APIs que ofrecen exactamente esos datos. Mapear con los datos y las funciones (servicios) existentes es parte de la implementación de la API, no la responsabilidad del equipo de desarrollo móvil, así de simple.
- ✓ **Público:** El público para las APIs es su equipo de desarrollo móvil.
- ✓ **APIs a proporcionar:** Las APIs necesarias para interesar al público son prácticamente todas oportunistas. Puede ser que tenga suerte de poder aplicar APIs existentes a una app móvil nueva, pero es conveniente evitar crear demasiadas dependencias entre las apps en contratos particulares de APIs.
- ✓ **Términos y condiciones de APIs:** Los términos y condiciones que rigen el consumo APIs tienen que ver con proteger la seguridad y estabilidad de sistemas back-end. La misma afirmación es válida para la 'libertad para innovar' (consulte la sección anterior).
- ✓ **Implementación de APIs:** Como las APIs producidas son oportunistas y rara vez viven mucho tiempo, las consideraciones más importantes son la velocidad y el costo de desarrollo. Si hay que tomar algún atajo para lanzar la API hoy en lugar de la semana próxima, hágalo siempre que tenga un plan viable para emprolijar la implementación de la API cuando esta se convierta en un éxito.
- ✓ **Consumir las APIs de otro:** Las aplicaciones móviles atractivas en general necesitan un elemento social, de modo que es crucial considerar qué APIs sociales públicas (como Twitter, Facebook, y LinkedIn) consumir y cómo controlar ese consumo. Existe la opción de adaptar las APIs sociales de terceros en una versión propia, más simple (consulte "Libertad para innovar" más arriba en este Capítulo).

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

> Como mínimo, debe considerar el lado oscuro de las APIs (vea el Capítulo 6) y tratar de minimizar el impacto en su reputación de negocios si surgen problemas de estabilidad o ética de las APIs públicas que usted decide usar.

El enfoque oportunista de este punto de entrada API puede sonar inmanejable, debido a la cantidad de APIs involucradas y su reutilización inevitablemente limitada, y de hecho con la tecnología disponible hasta hace tres o cuatro años, hubiera sido prácticamente imposible para muchas grandes empresas. ¡Ya no lo es hoy! El software de administración de APIs cambia la ecuación, con su capacidad de delinear fácilmente una API liviana y centrada en el cliente, basada en los datos de la cartera de software y servicios empresariales. Si es demasiado fácil crear una nueva API, y si es simple administrar y compartir grandes cantidades de APIs con diversas comunidades, la reutilización y el rigor arquitectónico en las interfaces API son temas mucho menos preocupantes. En cambio, la principal preocupación es qué hará que el consumidor de API sea exitoso.

Vivir en un mundo híbrido

Ningún libro sobre APIs estaría completo si no hablara de cómo las APIs se relacionan con la nube. El cuarto punto de entrada API se centra en usar las APIs como el modelo de consumo uniforme en un ecosistema híbrido de sistemas on-premise y entornos de nube pública y privada. El mantra de negocios es la “libertad de elección”: la libertad de elegir cómo obtener funciones o datos y libertad para implementar soluciones de la forma deseada.

Un enfoque de la integración centrado en API descansa en los siguientes principios:

- ✓ Una empresa desconectada no es competitiva, de modo que una solución de nube equivale a una solución híbrida, con partes Cloud y partes on-premise, y la integración debe administrarse a escala.
- ✓ La nube tiene que ver con capacidades (de negocio y de TI), no ubicación, de modo que cualquier modelo de consumo ubicuo debe ser independiente de la ubicación.
- ✓ En un mundo que es un “sistema de sistemas” no hay un perímetro de red tradicional para aplicar, de modo que es preciso que el control de las interacciones se realice al nivel de las aplicaciones.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Si para un consumidor todo es una API (con capacidad remota), el consumidor no necesita conocer nada sobre dónde y cómo se aloja la API. Los catálogos de APIs sindicadas pueden y deben proporcionar visibilidad entre límites de dominios y proveedores. En este punto de entrada API, uno de los apoyos más importantes para un desarrollador es el catálogo de APIs que están fácilmente disponibles para el consumo. No muestre cada API que existe (hay demasiadas); muestre solo las que son relevantes para el desarrollador en cuestión. Ese desarrollador no debería tener que preocuparse sobre cómo y por qué se obtiene la API; debería enfocarse únicamente en lo que puede hacer con la API una vez que la tiene a su disposición.

En un mundo híbrido, la estructura de dominios es inevitablemente compleja. Los mercados públicos, catálogos de APIs privadas, portales de asociados y otros forman parte del entramado API. Consiguientemente, una estructura de comunidad bien definida es más importante que nunca, con correlación directa entre decisiones de diseño de comunidad y decisiones de diseño de API.

La Figura 4-1 ilustra cómo un desarrollador puede usar APIs para acceder en forma segura cualquier parte de un entorno híbrido. En el momento del desarrollo, el mercado de APIs proporciona información sobre las APIs que están disponibles para la comunidad de ese desarrollador. En runtime, las gateways de nube aseguran la comunicación entre el entorno del consumidor de la API y cualquier punto terminal de API, independientemente de su ubicación.

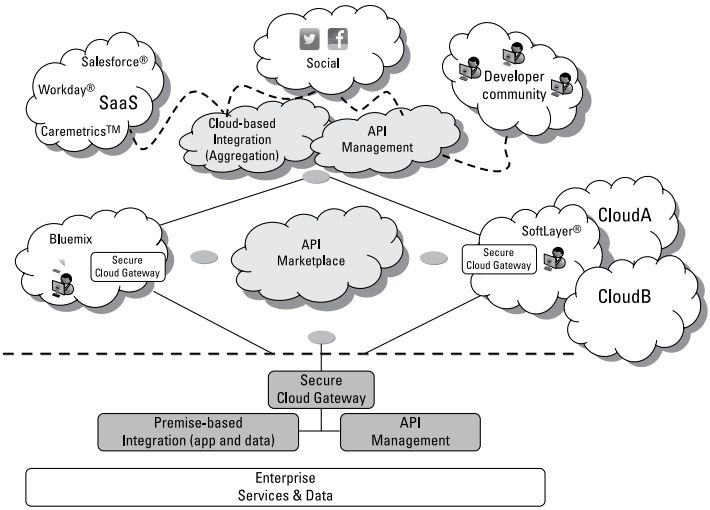

Figura 4-1: Uso de APIs como lingua franca para un entorno híbrido.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Pensando en las APIs en el contexto de un mundo híbrido, algunos consejos son:

- ✓ **Objetivo:** El resultado deseado es empoderar tanto a desarrolladores como a operaciones de TI. Deben introducirse controles cuidadosos en las APIs disponibles, y toda la comunicación en una red abierta debe ser segura y administrada correctamente.
  El motivo citado con más frecuencia para no adoptar un enfoque de nube híbrida es la preocupación por la seguridad. El segundo más citado es el temor a perder el control operativo.
- ✓ **Público:** El público puede ser cualquier combinación de desarrolladores internos y externos que forman parte del ecosistema híbrido, de modo que diseñar estructuras comunitarias apropiadas es muy importante. Tomar decisiones a nivel de la persona individual es muy poco práctico; necesita una estructura que le permita tomar y hacer cumplir decisiones para compartir APIs a nivel de la comunidad, tratando a todos los desarrolladores de una misma comunidad de la misma manera.
- ✓ **APIs a proporcionar:** Las APIs que necesita para interesar al público dependen de la estructura de la comunidad de su público. Para públicos externos, probablemente necesite un buen conjunto de APIs empresariales predefinidas; para públicos internos, necesita algunas APIs oportunistas para dar soporte a la rápida creación de apps innovadoras. No hay reglas generales que se apliquen a todos los casos híbridos; cada caso híbrido tiene sus particularidades.
- ✓ **Términos y condiciones de APIs:** Los términos y condiciones que rigen el consumo de APIs pueden ser complicados en un mundo híbrido, tanto para el área de negocios como de TI. Esto es válido ya sea que las APIs sean administradas o no administradas (vea el Capítulo 2), pero los mecanismos de implementación suelen ser muy diferentes para cada una:
  - Para APIs administradas, se aplican políticas de negocios y de TI en forma normal en la plataforma API: esta es una de las ventajas de usar APIs como el modelo de consumo uniforme. Asegúrese de que su plataforma API sea consciente de la comunidad para que pueda tomar las decisiones en el nivel de comunidad adecuado.
  - Para APIs no administradas, el único vehículo para aplicar términos y condiciones de negocio es la visibilidad de la comunidad, junto con acuerdos formales que se establecen a través de canales laterales. En lo que respecta a las APIs no administradas, el área de operaciones de TI no tiene una forma integrada de hacer cumplir los controles de seguridad y tráfico al nivel de la API. En cambio, las APIs no administradas deben invocarse a través de túneles seguros establecidos al nivel de la red.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

- ✓ **Implementación de APIs:** La forma de adecuar o curar datos y funciones para implementar APIs difiere para APIs empresariales preplanificadas y APIs oportunistas, impulsadas por la demanda. Tal como se describe en "Libertad para innovar" más arriba en este Capítulo, las principales diferencias son la robustez y el costo de runtime versus el tiempo y costo de desarrollo.
- ✓ **Consumir las APIs de otro:** Considerando qué APIs de Terceros consumir es más difícil para este punto de entrada que para los demás porque con el tiempo la cantidad y variedad de APIs disponibles aumenta sustancialmente.

  El mejor consejo es comenzar a consumir APIs de manera simple. Elija una pequeña cantidad de APIs importantes que desea consumir: APIs sociales, analíticas, móviles de back-end o de otro tipo, según su necesidad de negocios más inmediata. También tiene la opción de adecuar APIs de terceros y convertirlas en una versión propia más simple o controlada; tenga en cuenta esta opción al decidir.

Los entornos híbridos son intrínsecamente complejos. El uso de APIs como lenguaje universal puede hacer que esa complejidad sea mucho más manejable desde la perspectiva del desarrollador. Desde la perspectiva de operaciones de TI, la complejidad puede volverse manejable debido a la evolución en plataformas de nube híbrida.

Programe su mundo

El último punto de entrada API se concentra en el mundo de dispositivos y maquinaria, sobre la base de dos creencias: la revolución móvil impulsada por consumidores es más que teléfonos, y la fabricación y logística impulsarán la Internet de las Cosas corporativa inteligente.

La atención de salud, los servicios públicos, las ciudades, la industria… prácticamente para donde miremos, veremos la necesidad de un enfoque más inteligente para combinar personas, software y máquinas. No hay duda, la idea de programar todo en una única experiencia inteligente todavía es una idea emergente. Pero todos los días se ven signos de avance hacia ese objetivo en diversas áreas, como autos inteligentes, experiencias minoristas interactivas y edificios ecoamigables.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Una diferencia importante separa este punto de entrada de los demás: En general no se puede cambiar un dispositivo después de que existe en su forma física. Ese hecho a su vez significa que las APIs de dispositivos siguen siendo lo que son. Por lo tanto, el enfoque cambia para proporcionar una experiencia enriquecida con ejecución optimizada, de la siguiente manera:

- ✓ Extienda el software al campo físico.
- ✓ Controle todos los componentes, software o dispositivos a través de APIs programables.
- ✓ Use feedback y conocimientos obtenidos de los dispositivos para optimizar el comportamiento de todo el sistema, no solo un componente.
- ✓ Cuando corresponda, monetice sus capacidades de control general como su propio conjunto de APIs (consulte “Monetice sus Datos” en este Capítulo).

El punto de entrada ‘Programe su mundo’ en general se refiere más al consumo de APIs que a la provisión de nuevas APIs. Pensando en las APIs en el contexto de programar el mundo que nos rodea, las pautas son:

- ✓ **Objetivo:** El resultado deseado es un entorno totalmente programable. Busque infraestructura física programable y construya su propio software para que también sea controlable a través de APIs.
- ✓ **Público:** El público es principalmente interno. Sus propios desarrolladores son los que construirán sistemas inteligentes sobre APIs de dispositivos y software. Si usted es un productor de dispositivos, debe asegurarse de que sus dispositivos tengan APIs destinadas al consumo de otros, según lo requiera su modelo de negocio.
- ✓ **APIs a proporcionar:** Las APIs que necesita para interesar al público están definidas principalmente por las interfaces diseñadas en sus dispositivos y máquinas físicas. No tiene muchas opciones, ya que la mayoría de las APIs que utilizará están predefinidas.
- ✓ **Términos y condiciones de APIs:** Si tiene acceso de red a un dispositivo, probablemente pueda comunicarse con él porque sus APIs son por definición no administradas. Los términos y las condiciones no son tan importantes para este punto de entrada como para los demás. Mucho más importante es tener una buena biblioteca de APIs de dispositivos disponibles.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

En ciertos casos es conveniente controlar el acceso a las APIs a nivel del dispositivo. Esto se puede hacer agregando frente a las APIs de dispositivo una capa de APIs proxy administradas con controles de seguridad integrados.

- ✓ **Implementación de APIs:** Para APIs de dispositivos, los métodos usados para adaptar los datos y funciones para implementar las APIs no son una preocupación suya (salvo que, claro está, usted sea un productor de dispositivos físicos).

- ✓ **Consumir las APIs de otro:** Considerar qué APIs de terceros es muy importante para este punto de entrada API. Tener los contratos correctos celebrados con proveedores de dispositivos y maquinaria es un aspecto crítico. Si no tiene documentación API actualizada, simplemente no podrá comunicarse con el dispositivo.

Conforme el mundo se vuelve más inteligente e instrumentado, la necesidad de programar sus capacidades seguirá en aumento. Como este entorno difiere mucho de un entorno clásico exclusivamente de software, puede ser necesario empezar cuanto antes. Embarcarse en esta travesía exige a los desarrolladores y a las personas del negocio por igual obtener experiencia en lo que significa programar sin problemas interacciones entre dispositivos, software y personas.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 5

Middleware de API y de Integración

En este Capítulo

- Entender el middleware de API
- Elegir una topología de dominio

Así como se mencionó en capítulos anteriores, hay distinciones clave entre APIs administradas y no administradas. Una API administrada tiene controles de negocio y de TI propietarios aplicados, lo cual a su vez suscita la pregunta de dónde y cómo se aplican esos controles en tiempo de ejecución.

Aplicar los controles en código en general es una mala idea. Por un lado, es sabido que no es confiable, según la calidad del código de cada implementación de API. Además, una organización realmente debe poder cambiar los controles sin tener que cambiar y reimplementar la API misma. En otras palabras, los controles deben basarse en políticas, con independencia de las políticas administradas que definen la intención operativa (de negocios y de TI) y el tiempo de ejecución de API que aplica esa intención. Ejemplos de políticas comunes son el nivel de seguridad requerido y la cantidad de tráfico permitido.

Un middleware que aloja APIs y aplica políticas API comúnmente se denomina gateway API. No olvidemos que las plataformas API no son solo el tiempo de ejecución (runtime); necesitan también capacidades de tiempo de desarrollo.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

El middleware de API no es "simplemente otro ESB"

¿Alguna vez oyó decir: "Ya tengo un *enterprise service bus* (ESB), así que estoy listo para las APIs"? La implicancia es que el middleware de API es "simplemente otro ESB." De hecho, hay diferencias significativas:

- ✓ Idealmente, las APIs son definidas por configuración en lugar de codificación. Definir la API por configuración preserva la naturaleza de peso liviano del proxy API y da soporte a un tiempo de entrega rápido para APIs nuevas y modificadas. En contraste, las herramientas ESB de fines generales están centradas en flujo o en código, y las implementaciones de servicio forman parte del ciclo de vida de entrega de software empresarial. Incluso con entrega continua, un enfoque centrado en código es difícil de mantener, tratando de satisfacer la necesidad de APIs oportunistas que cambian rápidamente (vea el Capítulo 2).
- ✓ Los tiempos de ejecución API deben tener la velocidad del rayo, ser completamente seguros y muy escalables. Idealmente, la gateway API tiene características de red de tipo router, de modo que agregar un proxy API como parte de una interacción principio a fin nunca genere problemas con el tiempo de respuesta o la latencia. Las gateways API se basan en lenguajes de configuración específicos de dominio y de peso liviano con ejecución del tipo stateless. Si bien varios tiempos de ejecución ESB de propósitos generales también deben ser rápidos, robustos y escalables, los motores de ejecución deben poder dar soporte a la composición a toda escala y cierta cantidad de statefulness. Por consiguiente, los tiempos de ejecución ESB de propósitos generales no pueden estar tan optimizados en cuanto a la productividad como los tiempos de ejecución API.
- ✓ Las APIs tienen controles de negocio y TI administrados por políticas que incluyen desde autenticación hasta controles de tráfico y los términos y condiciones de negocio por los que se puede consumir la API (el plan API). Si bien algunos ESBs integrales incluyen políticas de gestión de tráfico, pocos tienen las capacidades de políticas de seguridad y certificación de una gateway API a gran escala, y ninguno tiene los controles de negocio separados incorporados a un plan API. Más aún, la naturaleza de software de los

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

ejecutables ESB en general deja menos espacio para que las operaciones puedan controlar los objetos de tiempo de ejecución en forma independiente después de la implementación.

- ✓ Las APIs deben estar disponibles para los desarrolladores en modalidad autoservicio. Cualquier tipo de proceso de aprobación posimplementación desacelera el ritmo de adopción y finalmente, a escala, genera costos significativos para la organización.

  El mecanismo preferido para compartir —que ha demostrado gran eficacia— consiste en publicar APIs en entornos de portal de desarrollador. En particular para el uso interno de APIs, compartir es una función preferentemente administrada en forma comunitaria, asegurándose de que ciertos desarrolladores vean solo las APIs que se supone que su comunidad debe usar.

  Los ESBs de finalidad general no incluyen ninguna de estas características, e incluso las soluciones de gestión de servicios (incorporadas o separadas) están destinadas más a controles de desarrollo de servicio y controles operativos que a optimizar el proceso que permite a los desarrolladores compartir.

- ✓ Finalmente, los responsables de API necesitan estadísticas de negocio para saber quiénes usan sus APIs y cuánto las usan. Estas estadísticas miden el éxito respecto de los objetivos de negocio para la cartera de APIs en lugar de enfocarse en cuestiones de tecnología informática, como los dashboards operativos típicamente incluidos en plataformas ESB.

Algunos argumentan que se necesita un solo bus, que puede reutilizarse para responder a necesidades de integración clásica, arquitectura orientada a servicios (SOA) y también administración de APIs. Sin embargo, sin proporcionar experiencias dedicadas para el desarrollador de la API, el propietario de la API y el consumidor de la API, se arriesga a realizar la ingeniería del middleware para el mínimo denominador común. Y no hace falta mencionar que los desarrolladores de integración clásicos también necesitan su experiencia optimizada por separado. Más aún, si usted necesita solo administración de APIs o capacidades de integración para fines generales, ¿por qué pagar por ambos?

Las empresas modernas necesitan la gestión de APIs así como la integración empresarial, pero como dos plataformas separadas destinadas a distintos públicos. Recuerde, no obstante, que si usted ya está haciendo SOA, tiene un buen conjunto de activos que puede descubrir rápidamente, reunir en APIs (proxy), y exhibir a través de una plataforma de administración de APIs.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohibe terminantemente su difusión, distribución o utilización no autorizada.

La topología (repetible) de la integración

Una gateway API controla el tráfico por el límite entre el consumidor de la API y la implementación de la API; también aplica los términos y condiciones para el consumo de la API definidos por el propietario de la API. Históricamente, tendimos a pensar en distintos tipos de gateways –gateways web, seguridad, mensajería o API – como "cajas" diferentes en una topología de implementación. Pero en un mundo híbrido de dispositivos, personas y software, la percepción de distintas cajas de gateways rápidamente desaparece. A medida que la escala y variedad de la interacción crecen en órdenes de magnitud, es ventajoso desde el punto de vista operacional tener una estrategia de gateway unificada con un único centro de comando y control para todo el tráfico que cruza los límites internos o externos.

Como buena práctica, el tráfico en un límite de dominio siempre debería pasar por una API. Este enfoque de la integración crea un nivel deseado de visibilidad y control. Además, proporciona máxima flexibilidad en términos de equilibrar implementaciones empresariales y oportunistas en una forma que es completamente invisible para el consumidor API.

La elección de una topología de dominio de granularidad gruesa o fina depende de cada empresa. Los límites típicos entre dominios están marcados por:

- ✓ Organización (por ejemplo, desarrollador móvil versus equipo de back-end)
- ✓ Propiedad (por ejemplo, límites entre líneas de negocio)
- ✓ Seguridad (por ejemplo, zona desmilitarizada versus red interna)
- ✓ Calidad de servicio (por ejemplo, mainframe versus distribuido)
- ✓ Ecosistema (por ejemplo, público o de un asociado)

Es importante que este modelo de topología de integración sugerido sea repetible y componible. Para cada dominio dentro de un ecosistema, las buenas prácticas dictan la misma estructura de topología, con interacciones entre dominios en runtime siempre a través de una gateway (API) como se ilustra

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

en la Figura 5-1. En la figura, cada dominio tiene la misma estructura de círculos concéntricos con distintos tipos de capacidades de integración. La topología de integración estándar para cada dominio puede estar compuesta por una topología de sistema de sistemas entre múltiples dominios simplemente repitiendo el patrón. En otras palabras, la estructura de dominio puede ser definida por las necesidades de una empresa, pero los mecanismos para compartir y consumir APIs siguen siendo los mismos. Esta estructura es un abordaje fresco a la naturaleza de los entornos de integración híbridos como sistemas de sistemas, y es una buena opción para cualquier punto de entrada API definido en el Capítulo 4.

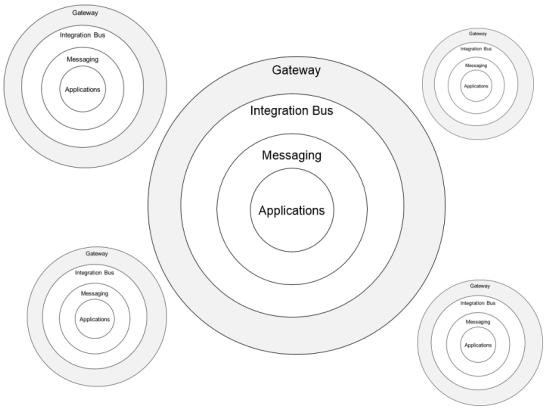

Figura 5-1: La topología de la integración es un patrón repetible.

Algunos argumentan que la latencia o el tiempo de respuesta es un problema en un patrón que siempre pasa por una API para la interacción entre dominios. Las buenas gateways API hacen que la latencia y el tiempo de respuesta no sean un problema, tal como se describe en “El Middleware de API no es otro ESB” en este Capítulo. Como analogía, ¿con cuánta frecuencia considera el número de routers entre un browser y un servidor web? Casi nunca. Simplemente supone que la infraestructura de routers es lo suficientemente rápida y escalable como para que la cantidad de routers no sea un problema. Garantizado, la distancia de red real entre dos puntos puede agregar y de hecho agregará algo de latencia, pero esta latencia se debe a la distancia de red en lugar de la cantidad de routers de red involucrados. Las buenas gateways API son como routers en este sentido: agregar una de estas gateways en el camino del consumidor de APIs al proveedor de APIs no implica una diferencia notable.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Utilice ESB de finalidad general en el rol de bus de integración, adecuando datos y funciones en servicios bien formados. Luego defina APIs (proxy) para controlar la productización de esos servicios en el ecosistema más allá de su propio dominio. Finalmente, ejecute todo el tráfico API a través de una gateway API muy escalable, robusta y segura. De esa manera, cada parte de la plataforma de integración de principio a fin se utiliza en el área de su fortaleza principal (consulte la Figura 5-1).

Si le interesa conocer más detalles sobre las capacidades de no superposición de una topología de integración de principio a fin, consulte el IBM Redbook Integration Throughout and Beyond the Enterprise, en www.redbooks.ibm.com/Redbooks.nsf/ RedbookAbstracts/sg248188.html?Open.

Modelo de referencia API y economía de servicio

En los años 2000, IBM definió un modelo de referencia SOA que delineaba las capacidades de middleware requeridas para crear, implementar y administrar servicios en forma eficaz. En la economía API, esas capacidades siguen siendo importantes, ya que los servicios son la base sobre la que se pueden definir las APIs, pero ya no son suficientes. Adoptar un ecosistema abierto, entregar APIs como parte de un producto de negocio y promover y administrar cuidadosamente su “persona” de negocio entre dominios requiere algo más que middleware de integración clásico.

El modelo de referencia para esta plataforma de middleware extendida se define en la Figura 5.2. Es una guía práctica para el conjunto de capacidades de middleware a buscar entre sus plataformas de integración y API.

Los elementos centrados en integración del modelo de referencia IBM SOA original se entregan en forma comprimida como la capa inferior del modelo de referencia de economía de servicio y API. Además, el nuevo modelo de referencia tiene capacidades para definir y administrar APIs, para portales de desarrollador y mercados, así como capacidades que aceleran el consumo de APIs.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Es importante no tomar decisiones sobre las partes del modelo de referencia en forma aislada. En cambio, use la estrategia de middleware integrado para convertir los activos en ventajas de negocio.

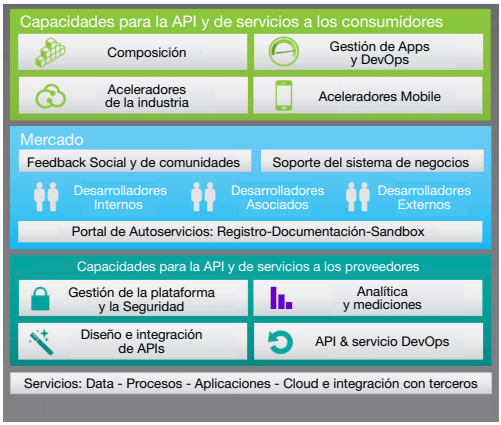

Figura 5-2: El modelo de referencia API y economía de servicio.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Capítulo 6

10 Cosas que hay que Saber Sobre las APIs

En este Capítulo

- Ver qué APIs son productos
- Reconocer que el diseño nunca se detiene
- Comprender que cada API necesita un responsable

Parte de la travesía API tiene que ver con la mentalidad que permite a una organización pensar en las APIs como una moda eficaz. Este Capítulo contiene algunas de las cosas que aprendí en mi propia travesía por el mundo de las APIs.

La experiencia omnicanal impulsa la necesidad de APIs

El concepto básico de APIs no es ninguna novedad. La diferencia ahora es que los usuarios modernos (consumidores y empresas) esperan una experiencia multicanal que sea tanto social como personal. Para ser verdaderamente personal, la experiencia debe ser autoorquestada, por lo menos en cierta medida. Una empresa ya no puede definir un proceso de canal del tipo “talla única”.

La autoorquestación inevitablemente señala en la dirección de las micro apps, que a su vez llevan a la necesidad de APIs construidas a medida. Una experiencia omnicanal implica un ecosistema que incluye personas, software y dispositivos, lo cual nuevamente conduce a la necesidad de APIs especialmente diseñadas para una finalidad particular o a medida.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Las APIs son productos de negocio

Pensando en las APIs como productos de negocio, uno puede ver más fácilmente la diferencia entre un enfoque centrado en APIs y el clásico enfoque de entrega de software. Para productos, hay diversas preguntas clave que hacer:

- ✓ ¿Quién es el público?
- ✓ ¿Qué quiere comprar?
- ✓ ¿Cuáles son los términos y condiciones bajo los que estoy dispuesto a vender?

Los términos comprar y vender se usan deliberadamente aunque los modelos económicos que respaldan las APIs pueden variar ampliamente. Ya sea que el "precio" sea dinero efectivo o influencia, o que el modelo sea pagado por el consumidor o por el proveedor, la naturaleza de la API como producto persiste.

El diseño de negocios es un esfuerzo principio a fin

Las APIs ya no se restringen al ámbito de la TI. Las APIs deberían formar parte de su diseño de negocios de principio a fin.

Consideremos a una viajera que comparte sus experiencias en canales sociales. Un día, publica en Twitter una experiencia muy mala con la aerolínea A. Diez minutos más tarde, recibe un email de la aerolínea: "Lamentamos que haya tenido una mala experiencia. Esto es lo que podemos hacer por usted." La semana siguiente, tiene una excelente experiencia de viaje con la aerolínea B y, como de costumbre, lo publica en Twitter. Cinco minutos más tarde, esa aerolínea replica su comentario con texto agregado: "Felices de que haya tenido una buena experiencia. Nos vemos la próxima."

Ambas aerolíneas consideraron por adelantado cómo incorporar los canales sociales, a través del uso de APIs, en el entramado de sus modelos operativos de negocio más amplios.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

Puede acceder a conocimientos mediante la instrumentación de APIs

Pruebe temprano, aprenda rápido, aumente la escala fácilmente: un ingrediente de esta receta es la capacidad de aprender rápido y la mejor forma de hacerlo es aprovechar la información que ya fluye por el sistema de operaciones del negocio. Usted puede acceder a esta información fácilmente a través de la instrumentación de APIs y usar analítica de negocios asociada, es decir, capacidades que deberían formar parte de una plataforma de middleware API totalmente funcional.

No todas las APIs son REST

Un mito común es “SOAP dejó de existir; las APIs son siempre REST.” Aunque la mayoría de las APIs modernas actualmente se basan en REST/ JSON en lugar de SOAP, esto no significa que no se encuentre utilidad al formalismo de SOAP; sino más bien que dicho formalismo no es necesario para el consumo humano de APIs (el enfoque de la mayoría de los debates APIs actuales). Todavía hay muchos usos de SOAP en comunicación máquina a máquina y herramientas de composición formal. Hay buenas razones por las que la industria ha inventado más de un protocolo vinculante; los buenos diseñadores eligen el correcto para cada propósito.

Cada API necesita un responsable de negocios

En otras palabras, ningún responsable de API en el área de negocio está exento de responsabilidad y poder de decisión. Es parte de la naturaleza humana resistirse a ser el “dueño” de cosas que no se controlan por completo (“solo esta parte es mi responsabilidad”). Sin embargo, es necesario asignar un responsable de negocios a cada API. Esa persona luego se convierte en el punto focal de decisiones sobre cómo convertir a la API en un producto y compartirla.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

No es necesario hacer versiones de APIs

Decir que las APIs no son versionadas es como decir que no hay necesidad de cambiar los pañales de un bebé. Obviamente, los cambios disruptivos suceden y cuando se producen hay que manejarlos. Decir que el versionado no es necesario simplemente significa dejar a los consumidores de una API descubrir el versionado ellos mismos.

Como las APIs son productos de negocio, las versiones deben hacerse con sabiduría. Reduzca el abandono por versiones acuñando una nueva versión oficial solo cuando una actualización no sea compatible hacia atrás.

Las APIs son fáciles de controlar con políticas

Las políticas son las formas tradicionales en que se codifican las intenciones de las áreas operativas de TI y del negocio. Por lo tanto, son vehículos para la consistencia e integridad de negocios. Los términos y condiciones que rigen el consumo y la administración de APIs deben codificarse en forma de políticas, que a su vez son aplicadas por medio de plataformas API.

Es importante asegurarse de que las políticas puedan ser cambiadas en forma independiente de la lógica que representa a la API misma (como interfaz y mapeo de datos). Así se podrá dar soporte al cambio dinámico del comportamiento operativo.

Las APIs tienen un lado oscuro

La innovación sin integridad de negocios es una propuesta de valor frágil. Poco importa que la violación de la integridad sea una promesa incumplida, la exposición de información sensible o un comportamiento inadecuado, el resultado es en gran medida el mismo. Bastará una sola mala experiencia para que la gente pierda la confianza en uno.

Cuando usted usa una API provista por un tercero, cuide que la integridad de su propio negocio no se vea perjudicada. El vehículo utilizado –acuerdos formales con penalidades, mecanismos de compensación o evaluación justificada de robustez y seguridad – no es tan importante como el hecho de haber tomado las precauciones correspondientes. Recuerde incluir en el análisis las cuestiones éticas.

Materiales protegidos por derechos de autor © 2015 John Wiley & Sons, Inc. Se prohíbe terminantemente su difusión, distribución o utilización no autorizada.

ACUERDO DE LICENCIA PARA EL USUARIO FINAL DE WILEY

Visite www.wiley.com/go/eula para acceder al ebook EULA de Wiley.