Advanced API Security

OAuth 2.0 and Beyond

Second Edition

Prabath Siriwardena

Apress®

Advanced API Security

OAuth 2.0 and Beyond

Second Edition

Prabath Siriwardena

Apress®

Advanced API Security: OAuth 2.0 and Beyond

Prabath Siriwardena
San Jose, CA, USA

ISBN-13 (pbk): 978-1-4842-2049-8 ISBN-13 (electronic): 978-1-4842-2050-4
https://doi.org/10.1007/978-1-4842-2050-4

Copyright © 2020 by Prabath Siriwardena

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484220498. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

This book is dedicated to my sister Deepani,
who backed me all the time!

Table of Contents

About the Author

Prabath Siriwardena is an identity evangelist, author, blogger, and VP of Identity Management and Security at WSO2. He has more than 12 years of industry experience in designing and building critical identity and access management (IAM) infrastructure for global enterprises, including many Fortune 100/500 companies. As a technology evangelist, Prabath has published seven books. He blogs on various topics from blockchain, PSD2, GDPR, IAM to microservices security. He also runs a YouTube channel. Prabath has spoken at many conferences, including RSA Conference, KNOW Identity, Identiverse, European Identity Conference, Consumer Identity World USA, API World, API Strategy and Practice Conference, QCon, OSCON, and WSO2Con. He has traveled the world conducting workshops and meetups to evangelize IAM communities. He is the founder of the Silicon Valley IAM User Group, which is the largest IAM meetup in the San Francisco Bay Area.

Acknowledgments

I would first like to thank Jonathan Gennick, Assistant Editorial Director at Apress, for evaluating and accepting my proposal for this book. Then, I must thank Jill Balzano, Coordinating Editor at Apress, who was very patient and tolerant of me throughout the publishing process. Alp Tunc served as the technical reviewer—thanks, Alp, for your quality review comments, which were quite useful. Also I would like to thank all the external reviewers of the book, who helped to make the book better.

Dr. Sanjiva Weerawarana, the Founder and former CEO of WSO2, and Paul Fremantle, the CTO of WSO2, are two constant mentors for me. I am truly grateful to both Dr. Sanjiva and Paul for everything they have done for me.

My wife, Pavithra, and my little daughter, Dinadi, supported me throughout this process. Thank you very much, Pavithra and Dinadi.

My parents and my sister are with me all the time. I am grateful to them for everything they have done for me. Last but not least, my wife's parents—they were amazingly helpful.

Although writing a book may sound like a one-man effort, it's the entire team behind it who makes it a reality. Thank you to everyone who supported me in many different ways.

Introduction

Enterprise APIs have become the common way of exposing business functions to the outside world. Exposing functionality is convenient, but of course comes with a risk of exploitation. This book is about securing your most important business assets or APIs. As is the case with any software system design, people tend to ignore the security element during the API design phase. Only at the deployment or at the time of integration they start worrying about security. Security should never be an afterthought—it's an integral part of any software system design, and it should be well thought out from the design's inception. One objective of this book is to educate the reader about the need for security and the available options for securing APIs.

The book guides you through the process and shares best practices for designing APIs for better security. API security has evolved a lot in the last few years. The growth of standards for securing APIs has been exponential. OAuth 2.0 is the most widely adopted standard. It's more than just a standard—rather a framework that lets people build solutions on top of it. The book explains in depth how to secure APIs from traditional HTTP Basic authentication to OAuth 2.0 and the profiles built around OAuth, such as OpenID Connect, User-Managed Access (UMA), and many more.

JSON plays a major role in API communication. Most of the APIs developed today support only JSON, not XML. The book focuses on JSON security. JSON Web Encryption (JWE) and JSON Web Signature (JWS) are two increasingly popular standards for securing JSON messages. The latter part of the book covers JWE and JWS in detail.

Another major objective of the book is to not just present concepts and theories but also to explain concepts and theories with concrete examples. The book presents a comprehensive set of examples to illustrate how to apply theory in practice. You will learn about using OAuth 2.0 and related profiles to access APIs securely with web applications, single-page applications, native mobile applications and browser-less applications.

I hope this book effectively covers a much-needed subject matter for API developers, and I hope you enjoy reading it.

CHAPTER 1

APIs Rule!

Enterprise API adoption has exceeded expectations. We see the proliferation of APIs in almost all the industries. It is not an exaggeration to say a business without an API is like a computer with no Internet. APIs are also the foundation for building communication channels in the Internet of Things (IoT) domain. From motor vehicles to kitchen appliances, countless devices have started communicating with each other via APIs.

The world is more connected than ever. You share photos from Instagram in Facebook, share a location from Foursquare or Yelp in Twitter, publish tweets to the Facebook wall, connect to Google Maps via the Uber mobile app, and many more. The list of connections is limitless. All this is made possible only because of public APIs, which have proliferated in the last few years. Expedia, Salesforce, eBay, and many other companies generate a large percentage of their annual revenue via APIs. APIs have become the coolest way of exposing business functionalities to the outside world.

API Economy

According to an infographic[1] published by the ACI Information Group, at the current rate of growth, the global Internet economy is around 10 trillion US dollars. In 1984, at the time the Internet was debuted, it linked 1000 hosts at universities and corporates. In 1998, after almost 15 years, the number of Internet users, globally, reached 50 million. It took 11 years since then to reach the magic number 1 billion Internet users, in 2009. It took just three years since then to get doubled, and in 2012 it reached to 2.1 billion. In 2019, more than half of the world's population—about 4.3 billion people—use the Internet. This number could further increase as a result of the initiatives taken by the Internet giants like Facebook and Google. The Internet.org initiative by Facebook,

[1]The History of the Internet, http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic/

© Prabath Siriwardena 2020
P. Siriwardena, *Advanced API Security*, https://doi.org/10.1007/978-1-4842-2050-4_1

launched in 2013, targets to bring together technology leaders, nonprofits, and local communities to connect with the rest of the world that does not have Internet access. Google Loon is a project initiated by Google to connect people in rural and remote areas. It is based on a network of balloons traveling on the edge of space and aims to improve the connectivity of 250 million people in Southeast Asia.[2]

Not just humans, according to a report[3] on the Internet of Things by Cisco, during 2008, the number of things connected to the Internet exceeded the number of people on earth. Over 12.5 billion devices were connected to the Internet in 2012 and 25 billion devices by the end of 2015. It is estimated that by the end of 2020, 50 billion devices will be connected. Connected devices are nothing new. They've been there since the introduction of the first computer networks and consumer electronics. However, if not for the explosion of the Internet adoption, the idea of a globally connected planet would never take off. In the early 1990s, computer scientists theorized how a marriage between humans and machines could give birth to a completely new form of communication and interaction via machines. That reality is now unfolding before our eyes.

There are two key enablers behind the success of the Internet of Things. One is the APIs and the other is Big Data. According to a report[4] by Wipro Council for Industry Research, a six-hour flight on a Boeing 737 from New York to Los Angeles generates 120 terabytes of data that is collected and stored on the plane. With the explosion of sensors and devices taking over the world, there needs to be a proper way of storing, managing, and analyzing data. By 2014, an estimated 4 zettabytes of information was held globally, and it's estimated, by 2020, that number will climb up to 35 zettabytes.[5] Most interestingly, 90% of the data we have in hand today is generated just during the last couple of years. The role of APIs under the context of the Internet of Things is equally important as Big Data. APIs are the glue which connect devices to other devices and to the cloud.

[2]Google Loon, http://fortune.com/2015/10/29/google-indonesia-internet-helium-balloons/

[3]The Internet of Things: How the Next Evolution of the Internet Is Changing Everything, www.iotsworldcongress.com/documents/4643185/3e968a44-2d12-4b73-9691-17ec508ff67b

[4]Big Data: Catalyzing Performance in Manufacturing, www.wipro.com/documents/Big%20Data.pdf

[5]Big data explosion: 90% of existing data globally created in the past two years alone, http://bit.ly/1WajrG2

The API economy talks about how an organization can become profitable or successful in their corresponding business domain with APIs. IBM estimated the API economy to become a $2.2 trillion market by 2018,[6] and the IBM Redbook, *The Power of the API Economy*,[7] defines API economy as *the commercial exchange of business functions, capabilities, or competencies as services using web APIs*. It further finds five main reasons why enterprises should embrace web APIs and become an active participant in the API economy:

- Grow your customer base by attracting customers to your products and services through API ecosystems.
- Drive innovation by capitalizing on the composition of different APIs, yours and third parties.
- Improve the time-to-value and time-to-market for new products.
- Improve integration with web APIs.
- Open up more possibilities for a new era of computing and prepare for a flexible future.

Amazon

Amazon, Salesforce, Facebook, and Twitter are few very good examples for early entrants into the API economy, by building platforms for their respective capabilities. Today, all of them hugely benefit from the widespread ecosystems built around these platforms. Amazon was one of the very first enterprises to adopt APIs to expose its business functionalities to the rest. In 2006 it started to offer IT infrastructure services to businesses in the form of web APIs or web services. Amazon Web Services (AWS), which initially included EC2 (Elastic Compute Cloud) and S3 (Simple Storage Service), was a result of the thought process initiated in 2002 to lead Amazon's internal infrastructure in a service-oriented manner.

[6]IBM announces new solutions for the API economy, http://betanews.com/2015/11/05/ibm-announces-new-solutions-for-the-api-economy/

[7]*The Power of the API Economy*, www.redbooks.ibm.com/redpapers/pdfs/redp5096.pdf

The former Amazon employee, Steve Yegge, shared accidentally an Amazon internal discussion via his Google+ post, which became popular later. According to Yegge's post,[8] it all began with a letter from Jeff Bezos to the Amazon engineering team, which highlighted five key points to transform Amazon into a highly effective service-oriented infrastructure.

- All teams will henceforth expose their data and functionality through service interfaces.
- Teams must communicate with each other through these interfaces.
- There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared memory model, no backdoors whatsoever. The only communication allowed is via service interface calls over the network.
- It doesn't matter what technology is used. HTTP, Corba, Pubsub, custom protocols—doesn't matter.
- All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

This service-based approach leads Amazon to easily expand its business model from being a bookseller to a global retailer in selling IT services or cloud services. Amazon started exposing both EC2 and S3 capabilities as APIs, both in SOAP and REST (JSON over HTTP).

[8]Steve Yegge on Amazon, https://plus.google.com/+RipRowan/posts/eVeouesvaVX

Salesforce

Salesforce, which was launched in February 1999, is a leader in the software-as-a-service space today. The web API built around Salesforce capabilities and exposing it to the rest was a key success factor which took the company to the state where it is today. Salesforce kept on using platforms and APIs to fuel the innovation and to build a larger ecosystem around it.

Uber

Google exposes most of its services via APIs. The Google Maps API, which was introduced in 2005 as a free service, lets many developers consume Google Maps to create much useful mashups by integrating with other data streams. Best example is the Uber. Uber is a transportation network company based out of San Francisco, USA, which also offers its services globally in many countries outside the United States. With the Uber mobile application on iOS or Android (see Figure 1-1), its customers, who set a pickup location and request a ride, can see on Google Maps where the corresponding taxi is. Also, from the Uber driver's application, the driver can exactly pinpoint where the customer is. This is a great selling point for Uber, and Uber as a business hugely benefits from the Google Maps public API. At the same time, Google gets track of all the Uber rides. They know exactly the places of interests and the routes Uber customers take, which can be pumped into Google's ad engine. Not just Uber, according to a report[9] by Google, by 2013 more than 1 million active sites and applications were using Google Maps API.

[9]A fresh new look for the Maps API, for all one million sites, http://bit.ly/1NPH12z

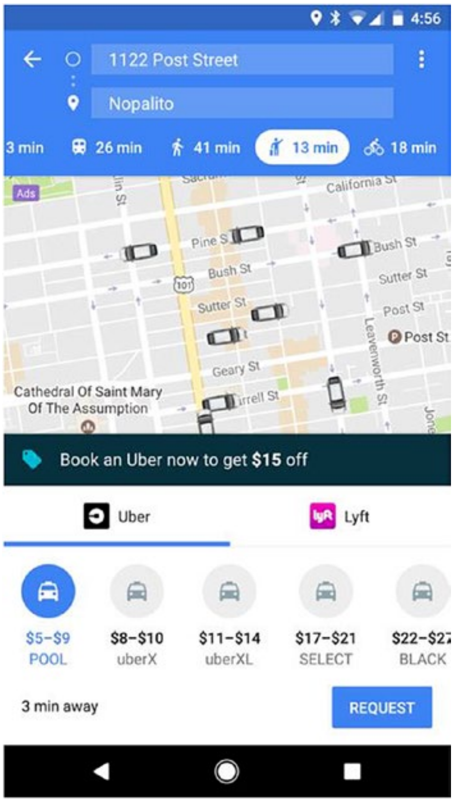

Figure 1-1. *Uber mobile app uses Google Maps*

Facebook

Facebook in 2007 launched the Facebook platform. The Facebook platform made most of the Facebook's core capabilities available publicly to the application developers. According to the builtwith.com,[10] the Facebook Graph API was used by 1 million web sites across the Internet, by October 2019. Figure 1-2 shows the Facebook Graph API usage over time. Most of the popular applications like Foursquare, Yelp, Instagram, and many more consume Facebook API to publish data to the user's Facebook wall. Both parties mutually benefit from this, by expanding the adaptation and building a strong ecosystem.

[10]Facebook Graph API Usage Statistics, http://trends.builtwith.com/javascript/Facebook-Graph-API

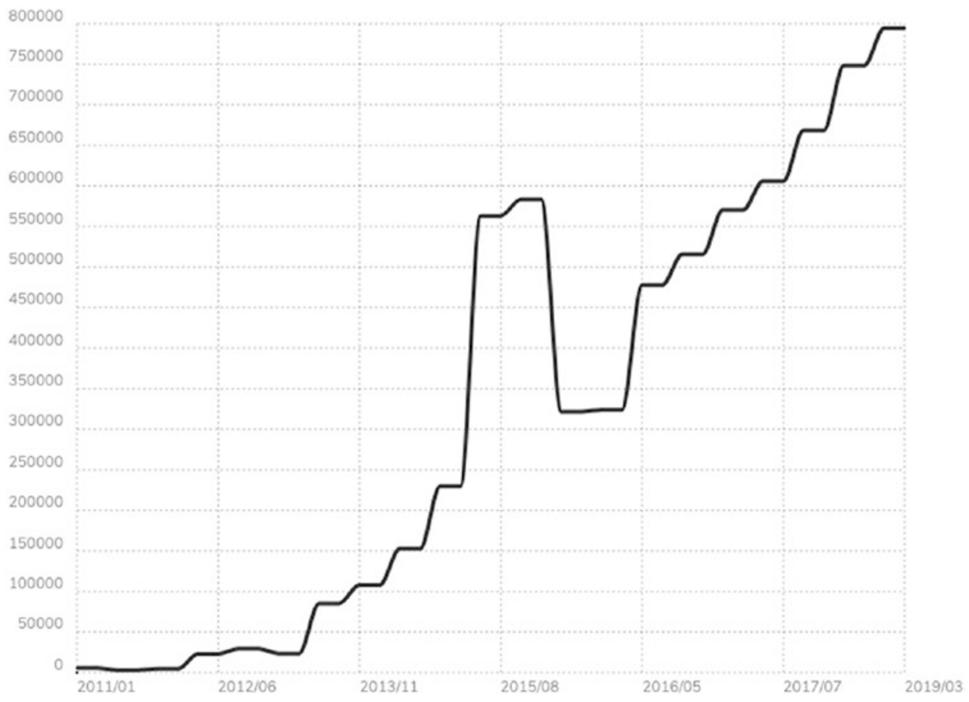

Figure 1-2. *Facebook Graph API usage statistics, the number of web sites over time*

Netflix

Netflix, a popular media streaming service in the United States with more than 150 million subscribers, announced its very first public API in 2008.[11] During the launch, Daniel Jacobson, the Vice President of Edge Engineering at Netflix, explained the role of this public API as a broker, which mediates data between internal services and public developers. Netflix has come a long way since its first public API launch, and today it has more than a thousand types of devices supporting its streaming API.[12] By mid-2014, there were 5 billion API requests generated internally (via devices used to stream the content) and 2 million public API requests daily.

[11]Netflix added record number of subscribers, www.cnn.com/2019/04/16/media/netflix-earnings-2019-first-quarter/index.html

[12]API Economy: From systems to business services, http://bit.ly/1GxmZe6

Walgreens

Walgreens, the largest drug retailing chain in the United States, opened up its photo printing and pharmacies to the public in 2012/2013, via an API.[13] They started with two APIs, a QuickPrints API and a Prescription API. This attracted many developers, and dozens of applications were developed to consume Walgreens' API. Printicular is one such application developed by MEA Labs, which can be used to print photos from Facebook, Twitter, Instagram, Google+, Picasa, and Flickr (see Figure 1-3). Once you pick your photos from any of these connected sites to be printed, you have the option to pick the printed photos from the closest Walgreens store or also can request to deliver. With the large number of applications built against its API, Walgreens was able to meet its expectations by enhancing customer engagements.

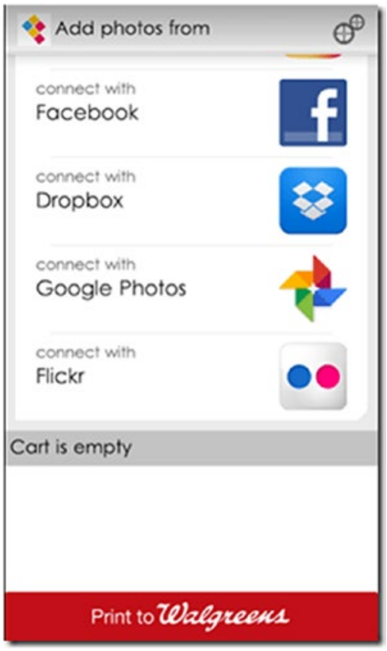

Figure 1-3. *Printicular application written against the Walgreens API*

[13]Walgreens API, `https://developer.walgreens.com/apis`

Governments

Not just the private enterprises but also governments started exposing its capabilities via APIs. On May 22, 2013, Data.gov (an initiative managed by the US General Services Administration, with the aim to improve public access to high-value, machine-readable datasets generated by the executive branch of the federal government) launched two initiatives to mark both the anniversary of the Digital Government Strategy and the fourth anniversary of Data.gov. First is a comprehensive listing of APIs that were released from across the federal government as part of the Digital Government Strategy. These APIs accelerated the development of new applications on everything from health, public safety, education, consumer protection, and many more topics of interest to Americans. This initiative also helped developers, where they can find all the government's APIs in one place (http://api.data.gov), with links to API documentation and other resources.

IBM Watson

APIs have become the key ingredients in building a successful enterprise. APIs open up the road to new business ecosystems. Opportunities that never existed before can be realized with a public API. In November 2013, for the first time, IBM Watson technology was made available as a development platform in the cloud, to enable a worldwide community of software developers to build a new generation of applications infused with Watson's cognitive computing intelligence.[14] With the API, IBM also expected to create multiple ecosystems that will open up new market places. It connected Elsevier (world-leading provider of scientific, technical, and medical information products and services) and its expansive archive of studies on oncology care to both the medical expertise of Sloan Kettering (a cancer treatment and research institution founded in 1884) and Watson's cognitive computing prowess. Through these links, IBM now provides physicians and nurse practitioners with information on symptoms, diagnosis, and treatment approaches.

[14]IBM Watson's Next Venture, www-03.ibm.com/press/us/en/pressrelease/42451.wss

Open Banking

API adaptation has gone viral across verticals: retail, healthcare, financial, government, education, and in many more verticals. In the financial sector, the Open Bank[15] project provides an open source API and app store for banks that empower financial institutions to securely and rapidly enhance their digital offerings using an ecosystem of third-party applications and services. As per Gartner,[16] by 2016, 75% of the top 50 global banks have launched an API platform, and 25% have launched a customer-facing app store. The aim of Open Bank project is to provide a uniform interface, abstracting out all the differences in each banking API. That will help application developers to build applications on top of the Open Bank API, but still would work against any of the banks that are part of the Open Bank initiative. At the moment, only four German banks are onboarded, and it is expected to grow in the future.[17] The business model behind the project is to charge an annual licensing fee from the banks which participate.

Healthcare

The healthcare industry is also benefiting from the APIs. By November 2015, there were more than 200 medical APIs registered in ProgrammableWeb.[18] One of the interesting projects among them, the Human API[19] project, provides a platform that allows users to securely share their health data with developers of health applications and systems. This data network includes activity data recorded by pedometers, blood pressure measurements captured by digital cuffs, medical records from hospitals, and more. According to a report[20] by GlobalData, the mobile health market was worth $1.2 billion in 2011, but expected to jump in value to reach $11.8 billion by 2018, climbing at an impressive compound annual growth rate (CAGR) of 39%. The research2guidance[21]

[15]Open Bank Project, www.openbankproject.com/

[16]Gartner: Hype Cycle for Open Banking, www.gartner.com/doc/2556416/hype-cycle-open-banking

[17]Open Bank Project connector status, https://api.openbankproject.com/connectors-status/

[18]Medical APIs, www.programmableweb.com/category/medical/apis?&category=19994

[19]Human API, http://hub.humanapi.co/docs/overview

[20]Healthcare Goes Mobile, http://healthcare.globaldata.com/media-center/press-releases/medical-devices/mhealth-healthcare-goes-mobile

[21]Research2guidance, http://research2guidance.com/the-market-for-mobile-health-sensors-will-grow-to-5-6-billion-by-2017/

estimated the market for mobile health sensors to grow to $5.6 billion by 2017. Aggregating all these estimated figures, it's more than obvious that the demand for medical APIs is only to grow in the near future.

Wearables

Wearable industry is another sector, which exists today due to the large proliferation of APIs. The ABI Research[22] estimates that the world will have 780 million wearables by 2019—everything from fitness trackers and smart watches to smart glasses and even heart monitors, in circulation. Most of the wearables come with low processing power and storages and talk to the APIs hosted in the cloud for processing and storage. For example, Microsoft Band, a wrist-worn wearable, which keeps track of your heart rate, steps taken, calories burned, and sleep quality, comes with the Microsoft Health mobile application. The wearable itself keeps tracks of the steps, distances, calories burned, and heart rate in its limited storage for a short period. Once it's connected to the mobile application, via Bluetooth, all the data are uploaded to the cloud through the application. The Microsoft Health Cloud API[23] allows you to enhance the experiences of your apps and services with real-time user data. These RESTful APIs provide comprehensive user fitness and health data in an easy-to-consume JSON format. This will enhance the ecosystem around Microsoft Band, as more and more developers can now develop useful applications around Microsoft Health API, hence will increase Microsoft Band adaptation. This will also let third-party application developers to develop a more useful application by mashing up their own data streams with the data that come from Microsoft Health API. RunKeeper, MyFitnessPal, MyRoundPro, and many more fitness applications have partnered with Microsoft Band in that effort, for mutual benefits.

[22]The Wearable Future Is Hackable, https://blogs.mcafee.com/consumer/hacking-wearable-devices/

[23]Microsoft Cloud Health API, https://developer.microsoftband.com/cloudAPI

Business Models

Having a proper business model is the key to the success in API economy. The IBM Redbook, *The Power of the API Economy*,[24] identifies four API business models, as explained here:

- *Free model*: This model focuses on the business adoption and the brand loyalty. Facebook, Twitter, and Google Maps APIs are few examples that fall under this model.
- *Developer pays model*: With this model, the API consumer or the developer has to pay for the usage. For example, PayPal charges a transaction fee, and Amazon lets developers pay only for what they use. This model is similar to the "Direct Revenue" model described by Wendy Bohner from Intel.[25]
- *Developer is paid directly*: This is sort of a revenue sharing model. The best example is the Google AdSense. It pays 20% to developers from revenue generated from the posted ads. Shopping.com is another example for revenue sharing business model. With Shopping.com API developers can integrate relevant product content with the deepest product catalogue available online and add millions of unique products and merchant offers to your site. It pays by the clicks.
- *Indirect*: With this model, enterprises build a larger ecosystem around it, like Salesforce, Twitter, Facebook, and many more. For example, Twitter lets developers build applications on top of its APIs. This benefits Twitter, by displaying sponsored tweets on end user's Twitter timeline, on those applications. The same applies to Salesforce. Salesforce encourages third-party developers to extend its platform by developing applications on top of its APIs.

[24] *The Power of the API Economy*, www.redbooks.ibm.com/redpapers/pdfs/redp5096.pdf

[25] Wendy Bohner's blog on API Economy: https://blogs.intel.com/api-management/2013/09/20/the-api-economy/

The API Evolution

The concept behind APIs has its roots from the beginning of computing. An API of a component defines how others would interact with it. API stands for application programming interface, and it's a technical specification for developers and architects. If you are familiar with the Unix or Linux operating system, the `man` command shouldn't be something new. It generates the technical specification for each command in the system, which defines how a user can interact with it. The output from the `man` command can be considered as the API definition of the corresponding command. It defines everything you need to know to execute it, including the synopsis, description with all the valid input parameters, examples, and many more. The following command on a Unix/Linux or even on a Mac OS X environment will generate the technical definition of the `ls` command.

```
$ man ls
NAME
     ls -- list directory contents
SYNOPSIS
     ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuwx1] [file ...]
```

Going little further from there, if you are a computer science graduate or have read about operating systems, you surely have heard of system calls. System calls provide an interface to the operating system's kernel, or a system call is how a program requests a service from the underlying operating system. Kernel is the core of the operating system, which wraps the hardware layer from the top-level applications (see Figure 1-4). If you want to print something from the browser, then the print command, which initiated from the browser, first has to pass through the kernel to reach the actual printer connected to the local machine itself, or remotely via the network. Where kernel executes its operations and provides services is known as the kernel space, while the top-level applications execute their operations and provide services in the user space. The kernel space is accessible for applications running in the user space only through system calls. In other words, system calls are the kernel API for the user space.