CHAPTER 1

# Introduction to Mashups

Mashups are inspiring a new generation of technology enthusiasts and programmers. Mashups enable experienced, web savvy programmers to integrate with the giants of the Web 2.0 space. As the name implies, mashups mix and "mash" the programming interfaces from different companies' products and services to create new products and services. Yahoo, Google, Amazon, eBay, and Microsoft have published *application programming interfaces* (APIs) based on web standards that allow you to utilize their complicated functionality without being a programming expert. Dozens more companies, big and small, have followed in the same way, creating a mashup explosion of API mixing and matching. New, sometimes strange, mashup creations pop up all the time.

In this chapter, I will explain what mashups are and give you a brief history of how mashups evolved and what technologies are used to create mashups. I will also give you a small sample of the dozens of products, services, and resources that are available to make mashups.

## What Is a Mashup?

A mashup is the evolution of the way web applications are made: it allows a programmer to integrate products and services from competing companies like Microsoft, Google, Amazon, and Yahoo to create new, unique products and services, as illustrated in Figure 1-1.

These new products and services integrate APIs published by each company using web technologies that have evolved over the history of the web applications. We will look at these technologies in more detail later on in the chapter.

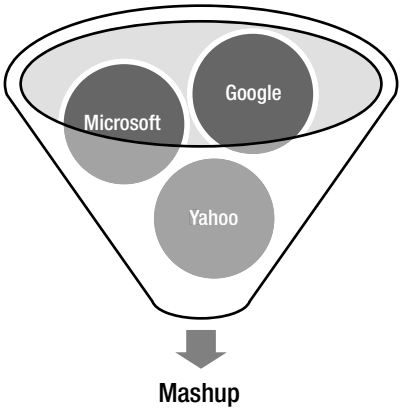


**Figure 1-1.** *Mashups can be created using APIs from competing companies.*

# A Brief History of Mashups

It will be difficult to detail the history of mashups without understanding the broader context of the history of the Web and how it has spurred the emergence of mashups.

Have you heard of Web 2.0? If you haven't, don't worry; this will be the first of many encounters with this phrase. Web 2.0 was coined in 2001 by Tim O'Reilly after the dot com crash. From a technical perspective, the word "Web" refers to the products, services, and business models that are created using the Internet as a platform. This is contrary to the PC or desktop computers being used as a platform. The "2.0" implies an upgrade from 1.0 products, services, and business models and the previous generation of technologies used to create those products and services. The new generation of technologies in Web 2.0 make web sites function and respond dynamically, like desktop applications.

Web 1.0 companies built products and services that would lock their customers in. They accomplished this by controlling the customers' data.

---

■ **Note** When I talk about customer data, I am referring to anything that the customer perceives as valuable. It doesn't have to be important to another customer. It is personal data but not necessarily sensitive like a Social Security number or address. And customers don't have to own the data. Examples of customers' data could be the weather forecasts in their city, a query on a search engine to find a consumer review web site because they want to buy a camera, a gallery of photos they want to share with family, or a subscription to a financial feed that tracks their favorite stocks. All of the examples show how data can be important to an individual, and that is the key to its value.

---

Let's look at the example of portals offered by vendors Netscape and Yahoo back in 1997. Customers logged in (using a vender-specific ID, of course) and customized his portals to

access news, weather, and sports to their liking. It saved this information for them and updated it regularly. However, the news, sports, and weather were from sources offered only by that portal alone or by the portal vendor (e.g., you only had access to Yahoo News, Yahoo Sports, and so on). And it was likely the source of data had to pay to be available to the customer, and it might be available exclusively. In this model, by controlling the data, the portal vendors would create barriers to entry by centralizing and controlling data that was valuable to the customer. To get to data, customers had to go to that vendor's products and services.

In addition to being controlled, data was isolated without the ability to be integrated with other data from other sources. There was no easy way for the customer to tie two different bits of logically groupable data in a meaningful way. For example, there was no way to merge information about an event at a local park with the weather forecast or a directory of Italian restaurants with a map of your area.

Web 1.0 API technologies were proprietary and often built with standards that were PC based and not web friendly—plug-ins, C interfaces and dynamic link libraries (DLLs). These APIs, frequently packaged in software development kits (SDKs), were about getting the programmer immersed and invested in mastering the vendor's APIs. There were developer programs and conferences touting one vendor API over another.

Web 2.0 represents how business models built on the Internet evolved from Web 1.0. After the dot com bomb in 2001, a small wave of companies emerged with a different perspective on how to leverage the Internet. That leverage came from opening up the customer data their products and services controlled. Using web standards that were common and widely adopted across traditional competitors, a new value proposition was created. This value proposition supercharged start-ups like Google, Amazon, and eBay and revitalized established Internet players like Microsoft and Yahoo.

Portals in Web 2.0 put customers in charge of their data. Let's return to the example of portals and now move forward to 2008: myYahoo, Microsoft Live, and iGoogle not only enable the customer to completely control what data they see and what source it comes from, but where on the page and how often it is displayed. Each bit of data, contained in what is called a portlet, is customizable to further give customers control over their data. If you like Google's portal but love Yahoo Sports and MSNBC News, you can get access to them inside Google's portal.

Creating new content for modern portals is based on open standards. Gone are the proprietary SDKs that are bound to vendor platform products and services. Now, a programmer can leverage the same fundamental technical knowledge to develop for Google, Yahoo, or Microsoft, as all the APIs use the same technologies, like HTML, CSS, and JavaScript.

Web 2.0 technologies play a big part in the advancement of portals. The fact is that modern portals are mashups. Mashups enable the retrieval and control of data by using the open APIs provided by service providers. In the next section, we will detail the technologies important to mashup development.

# Understanding Mashup Technology

As with all technologies, mashup technologies evolved over time. Figure 1-2 shows the progression of technologies that emerged and serve as a foundation for mashup development.

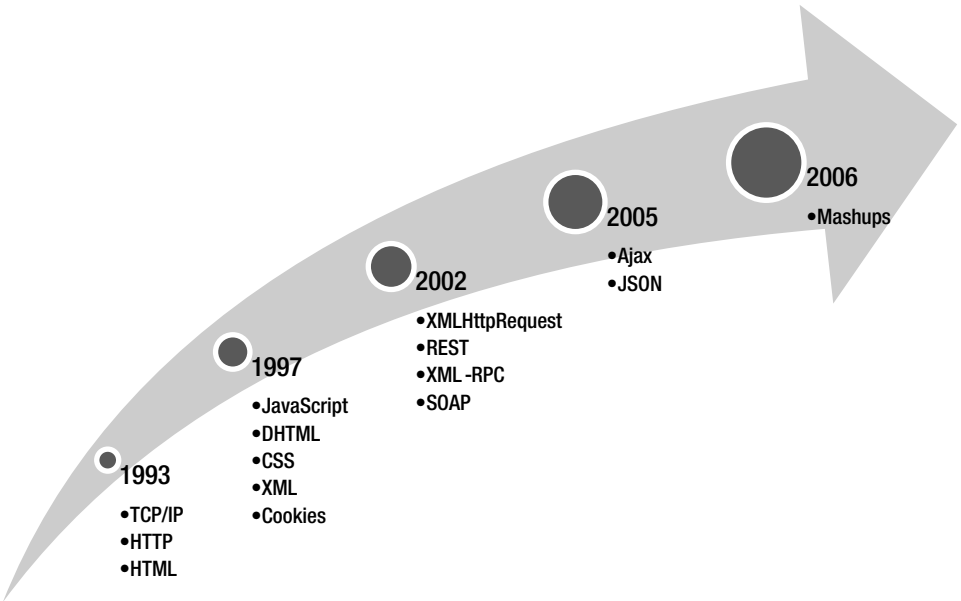


**Figure 1-2.** *Mashup technology timeline*

The most important thing to remember is not when a particular technology was created but when it became relevant to the overall community. One key factor is wide vendor adoption.

JavaScript would not be as important today if it had not been adopted by Microsoft and Netscape in their browsers. Netscape created JavaScript (originally called LiveScript) in 1995 in its browser Netscape Navigator. It wasn't until late 1996 that Microsoft shipped JScript (its own implementation of JavaScript) in its browser Internet Explorer. JavaScript became a de facto standard because it was used by virtually 100 percent of the market's browsers. Any newcomers in the browser arena had to implement JavaScript to be considered a viable alternative browser.

JSON was introduced as a JavaScript-based data format in 1999. It took over seven years to become relevant to the web development community. But as the web development community realized the need for a simple lightweight way to transfer data besides XML, JSON was rediscovered.

Before each of these technologies is explained, it is important to further categorize them into the roles they play in mashup development, which is done in Figure 1-3.

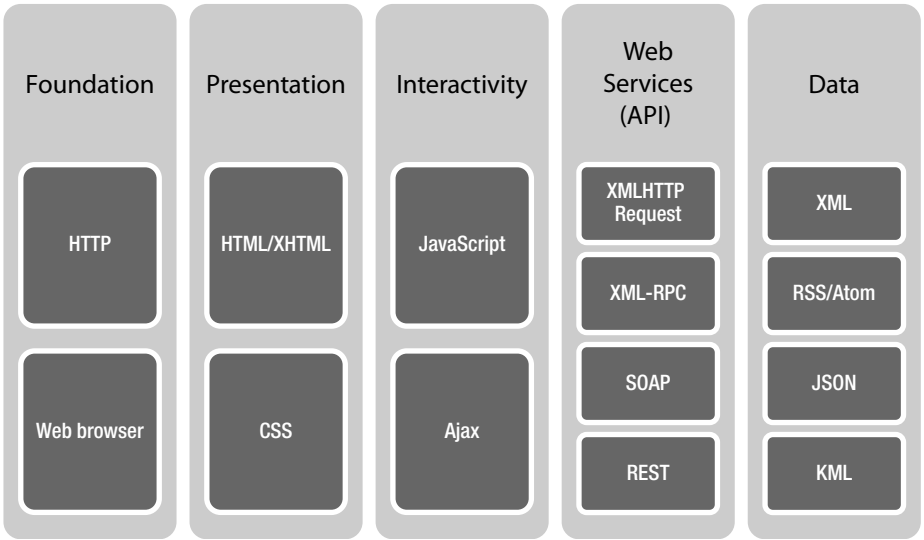


**Figure 1-3.** *Categorized technologies used by mashups*

## Foundation Technologies

You can't build a house without a foundation. The same statement holds true for creating mashups. There are two technologies that provide the foundation for all the others: HTTP and the web browser. There are dozens of books written on each of these technologies, and they are fairly well known in general, so I won't go into detail now. But it is always important to know how things work and interrelate, so I will give a broad overview here.

### HTTP

Hypertext Transfer Protocol (HTTP) is the protocol that enables us to navigate the Web. HTTP is a request/response protocol among clients and servers. An HTTP client initiates a request by establishing a TCP connection to a particular port on a remote host. An HTTP server listening on that port waits for the client to send it a request for a web page and then sends the client the web page it requests, provided the client has the relevant permissions to access that page.

### Web Browser

The delivery platform for mashups and all other web-based applications and sites is the web browser. A web browser is a software application that enables a user to display and interact with text, images, and other information. It hosts the technologies within the presentation, interactivity, web services, and data layers. The web browser uses HTTP to request web pages and data from remote servers. There are many browsers available today: Internet Explorer, Firefox, Opera, Safari, Camino (specifically on Mac OS), and Konqueror (specifically on Linux); there are even text-only browsers such as Lynx.

## Presentation Technologies

These technologies render the user interfaces that you see when you view web pages.

### HTML/XHTML

Hypertext Markup Language (HTML) is the language for the creation of web page structures. It describes the structure of text-based information in a document by denoting certain text as headings, paragraphs, and lists. It also denotes interactive forms, embedded images, and other objects. HTML is written in the form of elements called *tags*—labels surrounded by less-than (<) and greater-than signs (>).

XHTML is a reformulation of HTML in XML, therefore XHTML documents have to follow the strict rules of XML (they have to be well-formed, meaning elements need to be properly closed, element attributes need to have quotation marks around their values, etc.). It also provides new tags that have made structuring web pages easier. You can find a great overview of XML at XML.com (`http://www.xml.com/pub/a/98/10/guide0.html`).

### CSS

Cascading Style Sheets (CSS) is a language used to describe the presentation of a document written in a markup language. It is used to style web pages written in HTML and XHTML by defining rules that specify how markup should be styled and positioned.

## Interactivity Technologies

The interactive technologies are used to create custom dynamic behavior like showing or hiding content, dragging and dropping of content, and animations. Without interactivity in web pages, you are left with bland, static uninspired functionality that doesn't provide the same robust interaction that a desktop application provides.

### JavaScript

JavaScript is a scripting language used in web browsers to provide interactivity to web pages. It is compliant with a script standard called ECMAScript. JavaScript is the Mozilla Foundation's (originally created by Netscape Communications Corporation) implementation of the ECMAScript standard.

JavaScript was influenced by many languages and was designed to have a similar look to Java but be easier for nonprogrammers to work with. Contrary to its name, JavaScript is unrelated to the Java programming language. The language was renamed from LiveScript in a comarketing deal between Netscape and Sun in exchange for Netscape bundling Sun's Java runtime with its browser, which was dominant at the time.

I could spend the entire book focusing on JavaScript. Knowledge of JavaScript is not required to create basic mashups with Popfly, but it is required when you want to extend

Popfly's functionality with Popfly Blocks. There are dozens of books that can give you a deeper insight into it. I have listed some here:

- *Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional* by Christian Heilmann (Apress, 2006)
- *Practical JavaScript, DOM Scripting, and Ajax Projects* by Frank Zammetti (Apress, 2007)
- *Pro JavaScript Techniques* by John Resig (Apress, 2006)

### Ajax

Ajax is a JavaScript development technique that is so important to Web 2.0 development that it should be considered a separate technology. The name is an acronym standing for Asynchronous JavaScript and XML. It is a culmination of the trend in developing web applications that respond like desktop applications.

Ajax is asynchronous in that loading does not interfere with normal page loading and does not require that the entire web page be reloaded each time the user requests a change. For example, if you update the quantities in a shopping cart, the site could use Ajax to instantly display the new price and shipping total without having to waste time reloading the entire page again. Ajax increases a web page's interactivity, speed, functionality, and usability. JavaScript is the programming language Ajax function calls are made in. Data retrieved using the technique is commonly formatted using XML and JSON.

## Web Service Technologies: Application Programming Interfaces

When vendors want to create APIs for their products and services on the Web, web services are the means to do it. There are many options for vendors to use. Mashups use web service technologies to access a vendor's product functionality.

### XMLHttpRequest

Before we go into the various types of web services, it is important to call out the most important component of the emergence of web services as a way to create web APIs—XMLHttpRequest.

XMLHttpRequest is not a web service technology but an API that is available in JavaScript, to send data to and from a web server using HTTP, by establishing an independent communication channel between a web page's client side and server side.

The data returned from XMLHttpRequest calls are often provided by back-end databases. Besides XML, XMLHttpRequest can be used to fetch data in other formats such as HTML, JSON, or plain text.

XMLHttpRequest is an important part of the Ajax web development technique, and it is used by many web sites to implement responsive and dynamic web applications.

### XML-RPC

XML-RPC is a remote procedure call protocol that uses XML to encode its calls and HTTP as a transport mechanism.

### SOAP

Simple Object Access Protocol or Service Oriented Architecture Protocol (SOAP) is a protocol for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS. SOAP forms the foundation layer of many web services, providing a basic messaging framework those more abstract layers can build on.

### REST

Representational State Transfer (REST) is a style of web service that uses HTTP Universal Resource Locators (URLs) for function calls. When you navigate to a web page, you are using a form of REST. For example, the URL from this Yahoo search from a web browser is a REST call: `http://search.yahoo.com/search?p=ajax&fr=yfp-t-501&toggle=1&cop=mss&ei=UTF-8`. In this example the URL points to the search home URL at `search.yahoo.com/search` and sends parameters of the search after the question mark (`?`) separated by the ampersand sign (`&`).

## Data Technologies

Everything begins and ends with data. The Internet is a global tool to publish, organize, and share data. With that in mind, it is important to understand the formats commonly used in mashups to send, receive, and store data. Remember, the content of data can be endless. It can contain search results, links, photos, news, video, geographic locations, and audio. The list of content that can be formatted and delivered grows all the time.

### XML

To say that XML is an important part of how data is used in mashups is an understatement. XML seems to be the foundation for everything these days. XML is not the data itself but a way to represent the data using elements to provide structure, in the same way HTML does. The difference between HTML and XML is that with XML you can create your own custom elements to represent data in any way you want, whereas HTML is just for marking up web pages. XML's primary purpose is to facilitate the sharing of data across different information systems, particularly via the Internet.

### RSS/Atom

Really Simple Syndication (RSS) is one of the key technologies that have freed customer and vendor data. RSS is a web-feed format used to publish frequently updated content such as blog entries, news headlines, or podcasts. An RSS document, which is called a feed, web feed, or channel, contains either a summary of content from an associated web site or the full text. Users can use RSS readers on the Web or on the desktop to subscribe to the feed to receive any updates. Based on XML and transported on HTTP, RSS has become a standard for publishing text, video, and audio.

The name Atom applies to a pair of related standards. The Atom Syndication Format is an XML language used for web feeds, while the Atom Publishing Protocol (APP for short) is a simple HTTP-based protocol for creating and updating Web resources.

### JSON

JavaScript Object Notation (JSON) is a lightweight computer data interchange format. It is a text-based, human-readable format for representing objects and other data structures and is mainly used to transmit such structured data over a network connection (in a process called serialization).

JSON finds its main application in Ajax web application programming, as a simple alternative to using XML for asynchronously transmitting structured information between client and server.

### KML

Keyhole Markup Language (KML) is an XML-based language for managing the display of three-dimensional geospatial data in the mapping web applications like Google Maps and Microsoft Virtual Earth.

## Mashup Architecture

Now that we've gone through most of the dizzying array of technologies used to create mashups, it's important to know how all of them interrelate—how they fit together in an architecture. From an architectural perspective, the categories of mashup technologies can be seen as layers, as shown in Figure 1-4. A layer is a logical representation of where these technologies reside and how they are separated within a mashup.

As of this writing, mashups are primarily being hosted in web browsers. Mashups are also being hosted by operating systems like Windows Vista, in the form of Windows Sidebar Gadgets, and Mac OSX, as dashboard widgets. It is safe to assume that mashups in the future will be hosted in environments that closely emulate the web browser. Figure 1-4 details a generic mashup architecture.

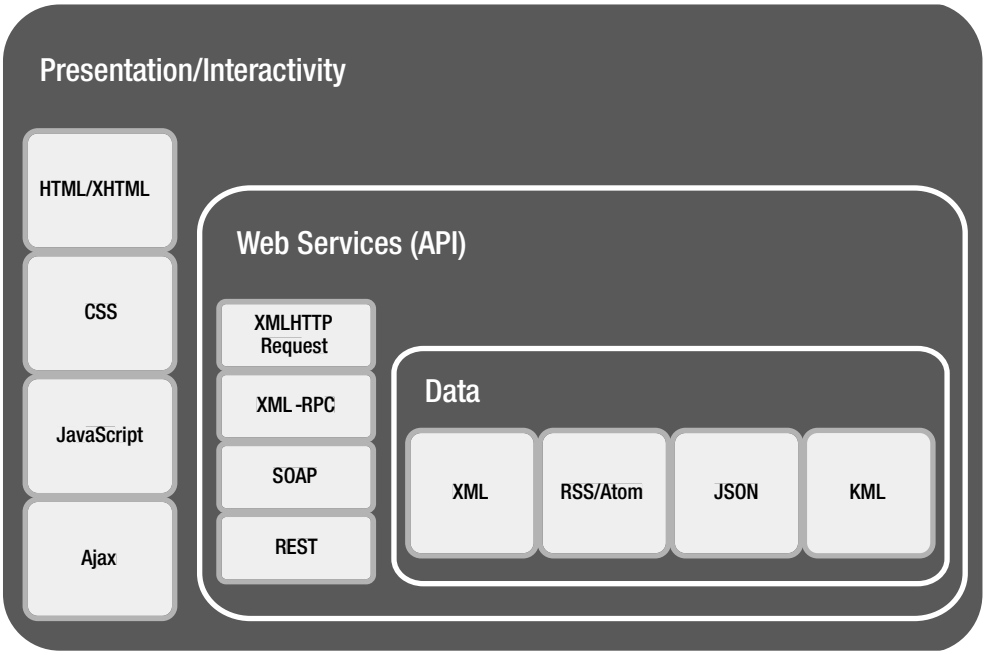


**Figure 1-4.** *Mashup architecture layers*

The presentation and interactivity layer is the user interface of the mashup. HTML, styled with CSS, displays the mashup interface to the user. JavaScript, with Ajax techniques, enhances the user's experience by making the mashup more responsive and desktop-application like. JavaScript also receives the user's input and, if necessary, initiates requests to the web services APIs used by the mashup.

For example, in a book search web application, users could search for a book they want to buy by entering the book's ISBN into the interface. Regardless if we're talking about XML-RPC, SOAP, or REST web services, XMLHTTPRequest is used to send the book's ISBN using XML or JSON to the server.

The book data (e.g., price, author, abstract and so on) stored on the server is retrieved from a database of books and formatted (usually in XML, but increasingly in JSON) before it is returned to the presentation and interactivity layer.

Once returned, JavaScript is used display the book's title, author, and cover image using HTML with CSS.

# Examples of Mashups

As of late 2007, Programmable Web has over 2,000 mashups listed. That's way too many to list here, so I picked out some representative mashups that have caught my eye and use some of the mashup APIs we'll be talking about in this book.

## AP News + Google Maps

AP News + Google Maps (`http://www.81nassau.com/apnews`) is a very interesting site that is a great example of how mashups use data and APIs from different sources to create a new product (see Figure 1-5). This site displays the Associated Press U.S. National, Sports, Business and Technology, and Strange news stories on top of Google Maps.

It retrieves a published RSS feed from the Associated Press web site and translates (using the Yahoo Geo-encoding REST API) the city and state from each story into a latitude and longitude point. Google Maps is used to plot the points on a map.

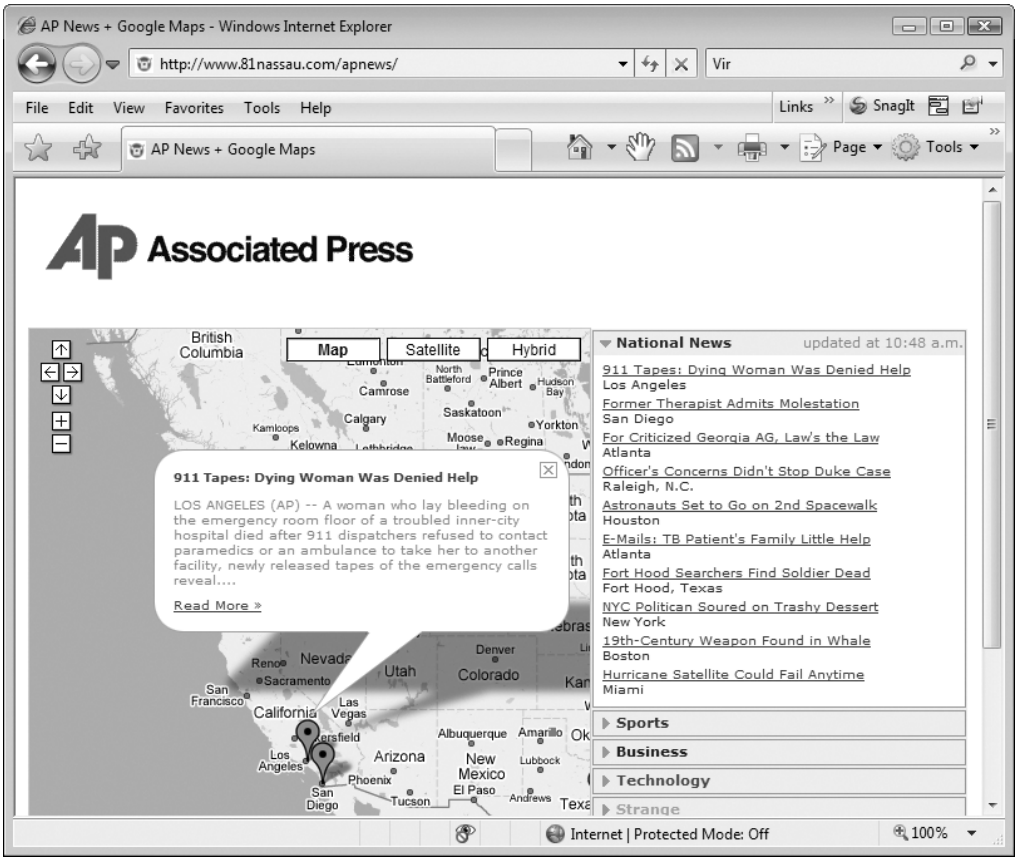


**Figure 1-5.** *API News + Google Maps: mashup of the Associated Press RSS feed, Yahoo Geo-encoding, and Google Maps*

## Markovic.com

Markovic.com (`http://www.markovic.com/markovic.com/ebay/search-virtual-earth.php`) is a mashup that displays items for sale on eBay and the seller's location using Microsoft Virtual Earth; see Figure 1-6. The mashup communicates (using SOAP and REST) with the eBay API to enable customers to search for items they are interested in buying. A list is returned in XML format that enables the developer to plot the location of the seller and display it using the Virtual Earth API.


**Figure 1-6.** *Markovic.com: mashup of Microsoft Virtual Earth and eBay APIs*

## Bubblr

The mashup called Bubblr (`http://pimpampum.net/bubblr`), shown in Figure 1-7, enables the user to search for photos on Flickr (`http://www.flickr.com`) and annotate them with speech bubbles. The mashup uses the Flickr API (using REST, SOAP, or XML-RPC) to search for tagged photos (e.g., photos labeled as "baby", "party", or "beach") or retrieve photos

using the customer's Flickr user ID. Once users find photos they are interested in, they can annotate the picture with dialog balloons.


**Figure 1-7.** *Bubblr: mashup using the Flickr API and custom picture editing*

The examples I have shown here are a very small representative sampling of the variety of mashups. The combinations of APIs are endless.

# Mashup Resources on the Web

In the following sections, I've listed some of the most useful mashup resources that exist on the Web. You should take the time to check them out!

## Web Sites

Many web sites and blogs talk about mashups; I will recommend the best ones I have found.

### Programmable Web

The best source of new and existing vendor APIs and the mashups using them is Programmable Web (`http://www.programmableweb.com`). New mashups and vendor APIs are added daily. I would start here if you want to see what's going on in the world of mashups.

### TechCrunch

TechCrunch (`http://www.techcrunch.com`) is a weblog that reports news and announcements from startups and companies that are considered a part of Web 2.0. Startups with business models that include mashup technology in their products and services are often covered.

### Mashable

Social networking web sites use mashup technology to give their users more control over creating, mixing, and sharing their data with a community of users. Mashable (`http://www.mashable.com`) covers the social networking space and has frequent news about MySpace, Facebook, Friendster, and other social networking sites.

## Mashup Web Service APIs

There seem to be new mashup web service APIs popping up all the time. Programmable Web lists over 400 APIs and counting. The explosion of APIs is directly related to how many new Web 2.0 startups come to market. As stated previously, if you are creating a Web 2.0 product or service, there is no way around *not* creating mashable APIs for it. Programmable Web and the other web sites are the best source for up-to-the-minute mashup APIs being created; I won't list all of them here.

However, the APIs from companies like Google, Microsoft, Amazon, Yahoo, and eBay have been around for a while and are considered the best places to start. When you look across the APIs offered by these companies, you begin to see how these APIs are grouped. Figure 1-8 shows how the APIs are categorized into Mapping, Photo, Search, Video, and e-Commerce.

Over time, the categories and lists of APIs is sure to grow as the Internet consumes more valuable data that customers use.

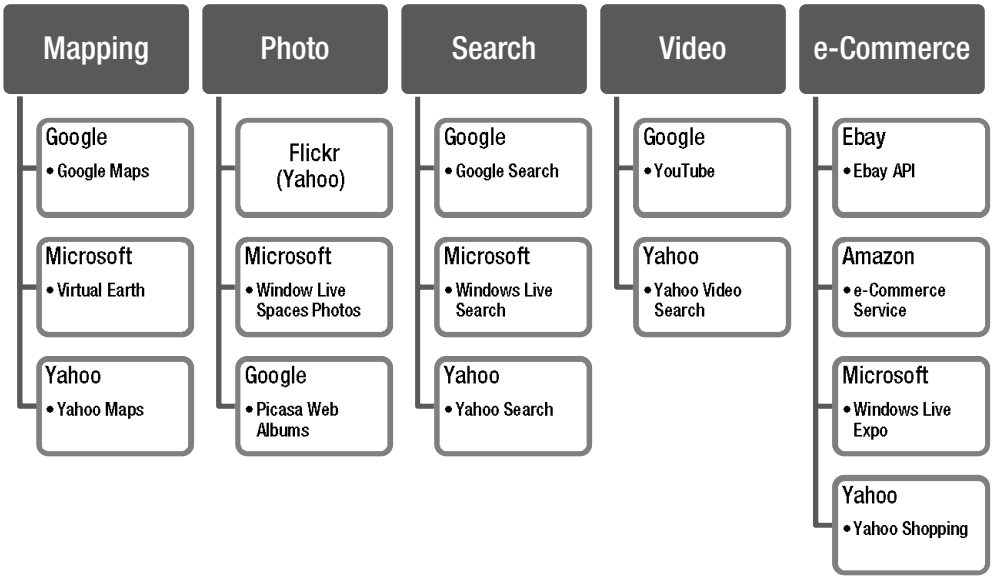


**Figure 1-8.** *APIs from Google, Microsoft, Yahoo, Amazon, and eBay categories by function*

## Mapping

Geographic coordinates (locations) can be displayed (in aerial or roadmap views) in remarkable detail. Integrating the display data with location information about business, roads, and landmarks allows directions from point to point to be determined.

Mapping is the first and most popular mashup API that is widely adopted and used by the programming community. Google Maps, which in 2002 was the first mapping API, is by far the most often used.

### Microsoft Virtual Earth

Microsoft's online mapping platform is available to web developers as well as Live Search customers. Here are the details on Virtual Earth:

- *Available at*: `http://dev.live.com/virtualearth/`
- *API used*: JavaScript interaction with embedded JavaScript control
- *Data format*: XML

### Google Maps

You can embed maps in your own web pages using the Google Maps JavaScript API and included geocoding service:

- *Available at*: `http://www.google.com/apis/maps/`
- *API used*: JavaScript interaction with embedded JavaScript control
- *Data formats*: XML and JSON

#### Yahoo Maps

Yahoo offers several mapping APIs that can integrate a store locator, view highway traffic patterns, or create custom routes:

- *Available at*: `http://developer.yahoo.com/maps/`
- *APIs used*: JavaScript with JavaScript Control, JavaScript with Flash, and REST
- *Data formats*: XML, HTML, and text

### Photo

Photo sharing among friends and family is very popular. It's an important piece of data that customers can't find enough ways to tag, categorize, comment, and share with the world.

#### Yahoo's Flickr

Flickr is the world's most popular photo sharing service:

- *Available at*: `http://www.flickr.com/services/`
- *APIs used*: REST, SOAP, and XML-RPC
- *Data formats*: XML and JSON

#### Microsoft Windows Live Photos

If you or your site visitors have a Windows Live Spaces account, you can use the Windows Live Spaces control with client-side JavaScript to let visitors use their Windows Live Spaces photos with your web site. Windows Live Spaces control is a bit complicated on the inside, but on the outside, you only have to create a `spacescontrol` element and event handlers to receive the selected photo data:

- *Available at*: `http://dev.live.com/spacescontrol/api.aspx`
- *API used*: JavaScript interacting with a JavaScript control
- *Data format*: None

#### Google Picasa Web Album

If you use Google's Picasa Web Album, you can use its web API to retrieve a list of photos, tags, comments, or user albums:

- *Available at*: `http://code.google.com/apis/picasaweb/overview.html`
- *API used*: REST
- *Data formats*: RSS and Atom

### Search

Searching is at the heart of the Internet. Without it, you wouldn't be able to navigate the billions of web pages on the Web.

#### Google Search

Google has an API that allows you to search in your web pages with JavaScript. A search box can be embedded in your pages, and you can display the search results it returns:

- *Available at*: `http://code.google.com/`
- *API used*: SOAP
- *Data format*: XML

#### Windows Live Search

This API allows developers to programmatically submit queries and retrieve results from the Windows Live search engine:

- *Available at*: `http://dev.live.com/livesearch/`
- *API used*: SOAP
- *Data format*: XML

#### Yahoo Search

There are four search types within the Yahoo Search service:

- *Contextual Web Search*: Finds related sites based on keywords
- *Related Suggestion*: Returns suggested queries.
- *Spelling Suggestion*: Spellchecks text that you send it
- *Web Search*: Searches web sites

Here are the details on the Yahoo Search service:

- *Available at*: `http://developer.yahoo.com/search/`
- *API used*: REST
- *Data format*: XML

### Video

With consumer video recording technology reaching millions, more and users are sharing, categorizing, and commenting on videos uploaded to online services, just like they do with photos.

#### Google's YouTube

YouTube is a vast repository of videos about every imaginable topic. You can use YouTube's API to integrate online videos into your web site:

- *Available at*: `http://www.youtube.com/dev`

#### Yahoo Video Search

The Video Search service allows you to search the Internet for video clips:

- *Available at*: `http://developer.yahoo.com/search/video/V1/videoSearch.html`
- *API used*: REST
- *Data format*: XML

### E-commerce

The monetization of the Internet first started with your basic banner ad at the top of web pages. Buying merchandize online was unheard of and thought to be unsafe. Today, online buying is in the billions of dollars annually, with companies like Amazon and eBay leading the way.

Note that APIs relating to search advertising are not listed here.

#### eBay API

By joining the eBay Developers Program, you can build applications using eBay web services. It provides functionality for shopping, searching, and monitoring sales of your online shop:

- *Available at*: `http://developer.ebay.com/`
- *APIs used*: SOAP and REST
- *Data format*: XML

**Amazon E-Commerce Services**

Amazon's E-Commerce Services enables developers, web site owners, and merchants to access product data and e-commerce functionality:

- *Available at*: `http://www.amazon.com/gp/browse.html/ref=sc_fe_l_2/103-1644811-9832630?%5Fencoding=UTF8&node=12738641&no=3435361`
- *APIs used*: SOAP and REST
- *Data format*: XML

**Microsoft Windows Live Expo**

Windows Live Expo, with its vast list of real estate, automobiles, jobs, and online merchandise can be accessed via a set of web services called Expo API:

- *Available at*: `http://dev.live.com/expo/`
- *APIs used*: SOAP and REST.
- *Data formats*: XML

**Yahoo Shopping**

You can display user reviews, search, and do comparison shopping using the Yahoo Shopping API:

- *Available at*: `http://developer.yahoo.com/shopping/`
- *APIs used*: SOAP and REST
- *Data formats*: XML

# Summary

The Internet has evolved from what was called Web 1.0 to Web 2.0. Web 1.0 business models locked the customer into products and services by controlling the customer's data. In Web 2.0, business models unlocked customer's data by publishing application programming interfaces (APIs) using open technology standards that anyone can use.

Mashups offer a rich world of creativity and freedom; they mix and match competing vendor APIs to create new, fun, and sometimes strange products and services. Because mashups use technologies that are familiar to web savvy programmers, mashups are popping up everywhere inspiring programmers and enthusiasts.

In the next chapter, you will see how Microsoft's Popfly is the next generation of mashup technology that will enable a wider audience of nonprogrammers to enter the mashup frenzy.