



Case Study: Mashups Interoperability and eInnovation

Citation

John Palfrey & Urs Gasser, Case Study: Mashups Interoperability and eInnovation, Berkman Publication Series (2007).

Published Version

<http://cyber.law.harvard.edu/interop/pdfs/interop-mashups.pdf>

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:2710475>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

BREAKING DOWN DIGITAL BARRIERS



CASE STUDY

Mashups Interoperability and eInnovation

by John Palfrey and Urs Gasser



Berkman

The Berkman Center for Internet & Society
at Harvard Law School

Research Center
for Information Law



University of St. Gallen

Sponsored by:

Microsoft®

CASE STUDY

Mashups Interoperability and eInnovation

John Palfrey *and* Urs Gasser

Berkman Publication Series, November 2007

<http://cyber.law.harvard.edu/interop>

The Berkman Center for Internet & Society, Harvard University
Research Center for Information Law, University of St. Gallen
Sponsored by Microsoft®

This work is licensed under the Creative Commons
Attribution-Noncommercial-Share Alike 3.0 United States License.
To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>

TABLE OF CONTENTS

Introduction	1
1.1 Definitions	1
1.2 Mashup Interoperability	2
1.3 Mashup Innovation	5
1 Mashup structure and state of play	6
1.1 Users/Consumers	6
1.2 Mashup Developers	7
1.3 Data providers	10
1.4 Pain Points in Mashups	11
2 Drivers and Inhibitors of Interoperability	14
2.1 Technology (architecture)	14
2.2 Market forces impacting mashup interoperability	16
2.3 Social Norms	19
2.4 Law regulating Web services interoperability	19
3 Benefits and drawbacks of interoperability	24
3.1 Benefits for specific stakeholders	24
3.2 Benefits for society at large	25
3.3 Drawbacks of interoperability	26
4 Maintaining Interoperability	30

INTRODUCTION

Web services have been wildly hyped for a long while now. Web services, and more specifically mashups, on which we focus here, are an area of enormous innovation. That innovation is manifested through new business models, new technologies, and clever new ways to use and share data. It's also an area where interoperability is the name of the game; the notion that people, data, and code can interact with other people, data, and code is the starting point for these services. The word “interoperable” is often in the definition of what a Web service is. The focus of this case study is the relationship between innovation in Web services applications and the interoperability (or interoperability potential) that we see. We conclude that the connection between interoperability and innovation is plain in this context. A wide variety of mashups that are useful to individuals, enterprises, and society as a whole have been enabled by interoperability in Web services, and could not exist without it. The drivers of interoperability have been market demand, private ordering, and work done in standards bodies. But the system by which it has come to pass is currently unstable, in the sense that a lawsuit or withdrawal of interoperable interfaces by a key stakeholder could set back innovation considerably. We consider several options for creating greater sustainability over time, such as license interoperability, open standards, and back-up in the form of traditional law enforcement.

1.1 Definitions

Over the past decade, computer applications have become more comprehensive, and the Internet has become more accessible – to users and, increasingly, to amateur programmers. More and more, the Web is being used not only as a portal for information but also for application-to-application communication. Web services is the general term for the interfaces available that connect different applications to each other, or any technology that supports machine-to-machine interaction over a network.¹ Web services were developed as a mechanism for connecting otherwise uninteroperable systems, and

¹ See, e.g., World Wide Web Consortium, Web Service Definition Language, <http://www.w3.org/TR/wsdl> (last updated 15 March 2001); Wikipedia, Web service, http://en.wikipedia.org/wiki/Web_service (last visited 31 October 2007).

they can make it much easier for applications to communicate,² especially with XML to code and decode the data and SOAP (Simple Object Access Protocol) to transport it using open protocols. The term “Web services” as we use it, however, encompasses more than these messaging technologies, including application-specific programming interfaces and any other software capabilities that facilitate data transfer between applications on the Web. This set of technologies provides an interoperable framework for communications between applications, but the overall interoperability depends greatly on the applications and the data exposed to them.³

1.2 Mashup Interoperability

There is no single, industry-standard definition for mashup interoperability. In order to approach a definition for mashup interoperability, we must start by defining mashups. In general, mashups exemplify Web services technology, fusing data from two or more Web applications to create an integrated experience informed by the original data sources. Mashup creators pull data dynamically from one source and integrate it with another. As a simple example, Fast Food Maps combines location information of major US fast food restaurants with Google Maps so residents of a particular city can see where they can stop for a hamburger or pizza.⁴

Interoperability specifically in the mashup context must be defined broadly enough so as to be useful in discussing all relevant applications of the term, but not so broadly that it encompasses so much as to lose meaning and value as a limiting force. Because all mashups inherently take advantage of interoperability, we take the view here that mashup interoperability is the set of conditions, including compatible technologies and willing participation by Web services and data providers, that permit developers to create mashups. Our

² W3Schools, Web Services Tutorial, <http://www.w3schools.com/webservices/default.asp> (last visited 31 October 2007).

³ Paul Anderson, What is Web 2.0? Ideas, technologies, and implications for education, <http://www.jisc.ac.uk/media/documents/techwatch/tsw0701b.pdf>, at 25; Björn Hartmann et al., Hacking, Mashing, Gluing: A Study of Opportunistic Design and Development, <http://hci.stanford.edu/publications/2006/MashUps-TR.pdf>, at 4; Tim O’Reilly, What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, at 3.

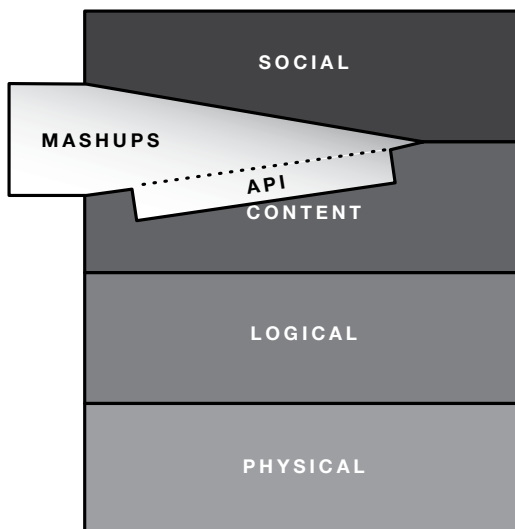
⁴ See Fast Food Maps, <http://www.fastfoodmaps.com/> (last visited 12 October 2007).

definition is thus not limited to just one protocol or set of protocols. Mashup interoperability, therefore, includes the quality that allows a mashup developer easily to convert his or her mashup from using one data source (say, a map) to another. It also includes some kind of parity in messaging technology, and further encompasses the needs for interoperability at the content level, such as data portability. However, we do not limit our discussion to the technical standards and interfaces that enable data exchange. Rather, we also discuss the broader market, legal, and social forces surrounding mashup interoperability, including the needs and focuses of the various stakeholders outlined later in this case study.

The two active ingredients of Web mashups are the data and application programming interfaces (APIs), which provide an interface with which non-programmers can gain access to a malleable form of the data. Both data and APIs can be public or private.

Mashups have recently gained attention because of the creativity involved in their development and the functionality they afford users. If the Internet is thought of in superseding layers – physical (the wires), logical (the protocols), content, and social – mashups fit between the content and social layers, changing the ways in which individuals relate to content.

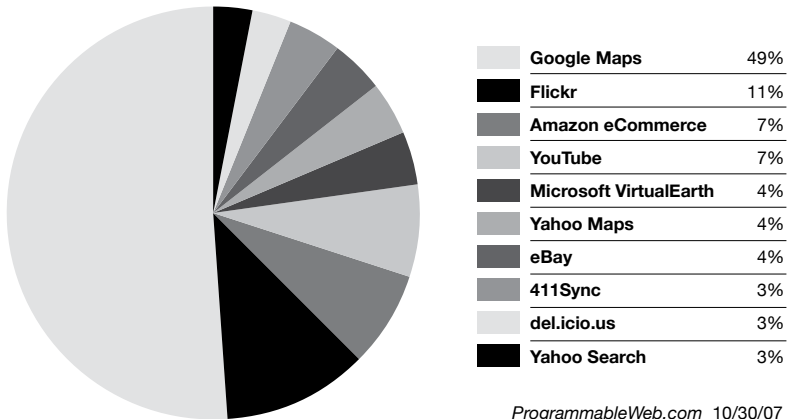
Mashups fit into the traditional stack as follows:



Mashups rely on the logical layer to support point-to-point messaging. This messaging is by definition a form of interoperability. Often, it is XML-based. Data is pulled from the content layer, greatly enabled by the API, and the final product affects the social interactions on the Internet.

Data sources come from a range of Web content, including posted APIs, statistics, maps, RSS feeds, and advertisements. Mashup content is also sourced by ‘screen scraping,’ a process where, in the absence of an API, a computer program ‘scrapes’ a site for data, using code that crawls the site and collects the information in a format the programmer can use for his or her mashup. Many people are experimenting with mashups using Microsoft, Google, eBay, Amazon, Flickr, Serena, Facebook, and Yahoo APIs; companies often post their own API so that developers can utilize it in new mashups. The result is a value-added representation of data that makes it easier for a user to synthesize information. The following chart shows the distribution of popular APIs that are used in this way:

Figure 1. Top APIs Used for Mashups⁵



The mere existence of these APIs does not lead inexorably to the development of any mashups, let alone mashups that are themselves interoperable with other APIs or mashups. The level of interoperability of a mashup depends on the endpoints into which the Web service plugs – the data on one end and the mashup on the other. The places at which data departs one point and arrives at another can be locked in such a way that they lessen the system’s in-

⁵ Programmable Web, Top APIs for Mashups, <http://www.programmableweb.com/apis> (last visited 30 October 2007).

teroperability overall. For example, a data provider could allow access to their data only through the use of their API, which may not allow for all uses, or a mashup programmer could represent his information in such a way that it is hard for someone who wants to use it as a data source to build on top of it.

1.3 Mashup Innovation

Because Web services began as a way to link large, non-interoperable systems, we observe that there is value in maintaining or enhancing the current level of interoperability from the perspective of promoting innovation in this space. Interoperability among Web services has facilitated the development of innovative mashups that are productive for society. The job of ensuring continued innovation in the mashups context becomes our focus. Challenges arise in developing an understanding of further ways to interoperate, what the stakeholders might demand, and how to achieve it if necessary.

When it comes to mashup innovation, there are three primary stakeholders:

- individual users of mashups
- programmers
- data providers

As players in the system, these stakeholders bring a variety of use cases, motivations and needs to the table.

The availability of data sources via open APIs increases the innovation potential of building on top of Web services technology. There are several types of innovation occurring above the technology layer:

- adapting existing business models, or testing new ones
- combining existing data in novel ways
- creating new content by analyzing existing data

The connection between these types of innovation and the interoperability we have observed currently crosses boundaries between layers in the stack – interoperability at the technical level enables innovation at the content level. However, the relationship may work for that step only, since that innovation may not in turn be interoperable or foster further interoperability at the logical or content levels.

1 MASHUP STRUCTURE AND STATE OF PLAY

1.1 Users/Consumers

Because a wide range of people and organizations are currently making mashups, an even broader range of people will soon be using them.

Some of the first mashups were targeted at individual users, and there continues to be a wide variety of mashups primarily useful to individuals. For instance, HousingMaps links Craigslist housing ads to Google Maps so someone looking for housing can filter the available postings and view them on a map to see what possibilities are in a given city or neighborhood.⁶ FindNearby has taken that idea a step further by bringing together listings from eBay, Craigslist, Amazon eCommerce, and other stores so consumers can search for products within a certain radius sold by both retail stores and other consumers.⁷ Nor are such mashups strictly commercial; the Center for Public Integrity's Media Tracker allows citizens not only to map the media and telecommunications companies serving a geographical area, but also links that information to those companies' campaign contributions and lobbying before the Federal Communications Commission.⁸ Thus, mashups can enable

⁶ HousingMaps, <http://www.housingmaps.com/> (last visited 31 October 2007).

⁷ FindNearby, <http://findnearby.net/> (last visited 31 October 2007).

⁸ Center for Public Integrity, Well Connected – Tracking the Players in Telecommunications, Media and Technology, <http://www.publicintegrity.org/telecom/> (last visited 31 October 2007).

individuals to find a place to live, purchase goods locally, and keep tabs on the government, for just a limited sample of possibilities. We discuss others in the following sections.

In addition, firms are using mashups to realize benefits of collaboration. For example, in their enterprise software, a mashup that a company hosts itself gives its employees the ability to integrate existing applications and new capabilities into so-called composite applications specially tuned to the needs of individual employees or small groups.⁹ IBM has its sights on this subset of the mashup market, with QEDWiki,¹⁰ a prototype browser-based development tool that allows end users to build their own composite applications in a wiki-type environment. Proto Software¹¹ offers a similar tool as a desktop application instead of in the browser.¹²

1.2 Mashup Developers

The goal of the Web services community is to encourage data mixing that will spark creativity and facilitate problem solving. We see that happening. A key advantage of the advent of open APIs is that many people can simultaneously tackle a particular problem by working on their own version of a mashup. Mashup contests are one way to get many people thinking about “cool,” useful new combinations of Web-based information.

Various companies are making it possible for individuals to create mashups without having to do any of the programming themselves. ZeeMaps.com, for example, will transpose customers’ data, submitted to ZeeMaps in an Excel spreadsheet, onto a Google map. Yahoo Pipes provides a community space where programmers can mix, match and share mashup code.¹³ On the more complex end of the spectrum, enterprise-level technologists enable mashup creation throughout organizations by centralizing APIs and making them

⁹ Kapow Technologies, Kapow Mashup Server Product Family, <http://kapowtech.com/products.html> (last visited 31 October 2007).

¹⁰ IBM Corporation, IBM Mashup Starter Kit, <http://www.alphaworks.ibm.com/tech/ibmmsk> (last visited 31 October 2007).

¹¹ Proto Software, <http://www.protosw.com/> (last visited 31 October 2007).

¹² Anne Zelenka, “Making Money in the Mashup Economy,” 21 January 2007, <http://gigaom.com/2007/01/21/making-money-in-the-mashup-economy/>.

¹³ Yahoo! Inc., Pipes, <http://pipes.yahoo.com> (last visited 31 October 2007).

available to members of the organizations. For instance, JackBe is a company whose main product, Presto, facilitates the creation and exchange of customized mashups within an enterprise.¹⁴

In a business sense, mashups have tremendous potential for companies of all sizes as a mechanism that could help pull together disparate information and make it available in a form that is most useful and relevant to the specific user base.¹⁵ While widespread and widely useful mashups are available and popular, the more easily a programmer can custom-tailor data to a certain use case, the more we observe the emergence of small, niche mashups. When individual developers can easily tweak or combine existing Web services to suit their own needs and then make the resulting mashup available to others at no additional cost, many more mashups will be made than if a significant investment of capital were necessary. And because the capital requirements are so low, a relatively modest amount of advertising revenue can suffice to make a mashup profitable even if it appeals to a limited audience. Mashup Servers, software packages designed to combine commonly used Web services technologies and make mashup creation easier for non-programmers in particular, are entering the market as a means of providing users and developers with secure and consolidated access to disparate data from a number of sources. In July 2007, the WSO2 Mashup Server team announced the release of the WSO2 Mashup Server v0.1. The WSO2 Mashup Server is a way to tailor Web-based information to the personal needs of individuals and organizations. It offers a platform for consuming data from a variety of sources including Web services, HTML pages, and feeds, and processing and combining it with other data using JavaScript with E4X XML extensions.¹⁶ Its developers envisioned the WSO2 Mashup Server becoming central to a budding ecosystem of community-developed services broadening the range of capabilities for mashups and distributed applications. Indeed, Mashup Servers are emerging

¹⁴ JackBe Corporation, Products, <http://www.jackbe.com/products/index.php> (last visited 31 October 2007).

¹⁵ In the words of JackBe's promotional material: "Imagine an enterprise with many high-value actors who need easy access to information from different data sources as well as ways to manipulate and mashup the data to reach important decisions." JackBe Corporation, Products, <http://www.jackbe.com/products/index.php> (last visited 31 October 2007).

¹⁶ E-mail from Jonathan Marsh to mashup-user mailing list, July 6, 2007, <http://wso2.org/maillarchive/mashup-user/2007-July/000001.html>.

as a new Web service with rich metadata and artifacts that can be used to quickly build bespoke user interfaces.

Mashups are created to support a range of business models and practices. First, an existing profit-seeking enterprise can incorporate mashups to facilitate or complement another business model. Second, a start-up can try to make the mashup itself the primary revenue driver of a new business model. And third, various parties have produced public service mashups to support various nonprofit goals.

Existing businesses can incorporate mashups externally (i.e., for their customers), or for their own internal purposes. For example, stock trading company E*TRADE has used mashup technology to create a value-added service, an “intelligent cash optimizer,” which shows the investor where they could put their money. It relies on E*TRADE’s own, proprietary data, but is represented in a dynamic and easy-to-understand way. Mashups are also being used in the back-end of businesses, in companies that already have applications that simply need to talk to each other. This can allow for much quicker time to market, and leverages work already done.¹⁷ Another example of an add-on mashup is a store locator with a map, which can be useful both to customers and to employees. News outlets have also turned to mashups as useful ways to show the information they are reporting, through dynamic maps and visuals. In this way, these uses of mashups represent an improvement upon the classic notion in computing of middleware.

Increasingly, the mashup is being used as the driver of the business. They work off pay-per-use, subscription, or ad-funded models. For example, sites like mapmyrun.com and povo.com are start-ups, providing value-added services to users. The use of mashups as the basis for a start-up entails greater risk than simply adding it on to an existing business because it inherently depends on one or more APIs or data sources from third parties who could change terms or cut off access entirely. This phenomenon gives rise to a series of concerns that we will explore later in this paper.

In public service mashups, non-profits, governments, or private citizens use mashups to serve the public interest without a profit motive. Some are funded through grants and other non-profit funding sources. ChicagoCrime.org is a

¹⁷ Interview with John Musser, May 18, 2007.

popular example of such a mashup. It pulls data from Google Maps and the Chicago Police Department, placing crimes and information about them onto a street-by-street map. This type of mashup is commonly opportunistic:¹⁸ the programmer finds data sources, but sees them as inaccessible to those who would use them. She remixes and represents the data so that individuals get value out of it. Experimentation with mashups by members of the non-profit and government sectors is an important reason to seek sustainability in mashup interoperability.

1.3 Data providers

Data providers fall into two main camps: companies whose data is normally gathered and held for their own purposes, and public or government organizations that collect data in the course of their service to the public.

Companies have increasingly made their data available via API, and opening data to mashup developers is quickly becoming part of the Web 2.0 movement. One proponent stated, “It’s going to be almost like a decade ago. Do you have a Web site? Check. That’s how it will be with APIs.”¹⁹

The main motivation to make data more readily available via API appears to be exposure. For a private company such as Google, allowing others to use their map functionality increases the branding of the Google name. Some say that companies create APIs in recognition of the innovative and generative possibilities afforded by allowing others to mix and mash the data.²⁰ A company’s reliance on the uptake of its API varies. Some companies tie the use of their data to payment in a pay-per-use model. For example, salesforce.com provides customers access to its data on an as-needed basis.²¹ Others provide the API (and access to the data) free, sometimes in conjunction with placing advertisements.

While public-sector information sources will always be available, it is important to note that without the proper incentives to obtain data, corporations

¹⁸ Interview with Adrian Holovaty, May 30, 2007.

¹⁹ John Musser, quoted in Jason Snyder, Digg floats API, phishing mashups to come, April 20, 2007, http://www.infoworld.com/article/07/04/20/HNdiggapi_1.html.

²⁰ Interview with Jason Callina, May 18, 2007.

²¹ Interview with John Musser, May 18, 2007.

will not collect it in the first place, let alone make it available to mashup developers. Data providers usually have to invest resources in gathering or creating data, and they expect to be compensated for that in some way. Intellectual property laws, advertising revenue, and intangible rewards in terms of visibility, goodwill, and market position currently provide incentives to provide data, but if these cease to be compelling, the Web services ecosystem will have to adjust to ensure data providers continue to get a return from their investment.

1.4 Pain Points in Mashups

Open APIs have allowed for a lot of experimentation, but they have given rise to an unstable, fragile ecosystem of law, technology, markets, users, and other players. For instance, while API providers have made them fairly open so far, they retain the right to restrict API use legally (through terms of service) or technically (by altering the API or data provided) at any time. Furthermore, companies that provide APIs free of charge (as most do) are generally under no stated obligation to mashup developers to provide reliable service. As their business models change, the reasons for a company to provide an open API may change as well. To date, many mashups have been built simply because with open APIs they are relatively easy to create and allow for powerful representations of information. This opportunistic development has occurred in spite of instability in the system. Many developers have invested time and focus on building upon the works of others through this interoperability. What is lacking is certainty that the interoperability on which they have built will persist, at least at the cost (often, at zero cost) and in the format in which it has been afforded to date. It is unclear how long it will last.

Despite the high degree of innovation in the mashup space and particularly at the messaging level, several pain points contribute to this continued instability. These include: 1) a lack of identity interoperability; 2) a lack of license interoperability, which results in confusion or instability related to terms of service and party rights and responsibilities; 3) and a lack of data portability.

Some technological inhibitors of full mashup interoperability stem from a lack of identity interoperability. Without consistent identification, the programmer must create an account with each data provider. The user must create an account with each mashup that requires registration. Security, data tampering and authentication issues discussed in the Digital ID case study

apply. According to one expert, “The biggest issue surrounding APIs is identity. What’s the standard? Is it OpenID? I don’t know. The whole area is vague. Most of the major API vendors have their own authentication APIs. Each is similar, but in the end, they’re all different.”²² Furthermore, each is tied to its own specific terms of service.

The relationship between the mashup and the data provider is generally governed by the data provider’s terms of service. These usually grant the right to use the data for purposes authorized by the data provider, delineate any responsibilities the data provider will accept (such as a certain percent uptime), and their limitation of liability. Terms of service are generally straightforward for the hobbyist mashup creator, but they can present problems to the counsel for a company considering basing its business model on mashup development.

One common problem is the frequent lack of usage terms. Put another way, the licenses themselves are not interoperable with one another. Data providers can- and do- charge for the use of their data when traffic is sufficiently high, or sometimes when the mashup starts charging their users for *their* service. Some data providers, such as Microsoft’s Windows Live Platform, have attempted to simplify their agreements by writing consistent cost information into the terms of service – a certain amount owed to Microsoft per unit of traffic.²³ However, there is no agreed-upon industry standard. Service level agreements, contracts guaranteeing availability of Web services to given developers in exchange for payment or other conditions, can introduce stability into the system, explicitly tailoring responsibilities and rights for each party, and setting terms related to the length of the agreement that establish a reasonable level of certainty on which an investment might rationally be based. In practice, terms of service and service level agreements can pose legal barriers to some people who would otherwise create mashups but do not fit squarely within the terms of service or cannot afford a service level agreement.

²² John Musser, quoted in Jason Snyder, Digg floats API, phishing mashups to come, April 20, 2007, http://www.infoworld.com/article/07/04/20/HNdiggapi_1.html. The article also discusses the possibility of phishing mashups and the privacy concerns surrounding mashups. See also John Musser, Banned Books and the Big Brother Mashup, January 8, 2006, <http://blog.programmableweb.com/2006/01/08/banned-books-and-the-big-brother-mashup/>.

²³ See Windows Live Platform Terms of Use, <http://dev.live.com/terms/> (last visited 12 September 2007).

Parties interested in Web services are concerned that the current potential for open innovation will be hampered by complicated and possibly conflicting terms of service between data sources.

Some stakeholders are also concerned that any investment they make will be dependent on the goodwill of third-party Web services providers. The system is currently unstable in that data providers may not maintain their initial level of data availability. Terms of service often release the data provider from any suggestion that their data will remain available indefinitely, or even that they will provide prior notice to the mashups that rely on it if they are removing their service. Individuals who are left with no data source for their mashup are out of luck.²⁴ This is particularly a concern in cases in which the data is not being provided willingly, but is being “scraped” from another Web site that may not take kindly to the appropriation of its data.

Data portability is a corollary of the previous two pain points. Currently, it is difficult to port settings and data from one provider to another. This could be partly solved by a persistent sense of identity and consistent terms of service across the Net. The market drives against data portability because it is often better for a data provider’s business to create lock-in, keeping a user coming back to the site instead of another.

When data is shared by multiple mashups, it gives rise to questions of ownership, rights, and control. What would happen, for example, if a provider suddenly pulled its API from the public Web, even after it had been used in several mashups that cost millions of dollars and held personal data for millions of users? Would the mashup creators have any recourse? Would their users? Barring any service level agreements, it seems that data providers can cut off access at any time, if their terms of service are to be believed. Thus, the survival of mashups depends not only on interoperable systems but also on the continued willingness of data providers to give meaning and utility to those interoperable systems.²⁵

²⁴ Interview with Phil Malone, May 3, 2007.

²⁵ Ingbert R. Floyd et al., *Web Mashups and Patchwork Prototyping: User-driven technological innovation with Web 2.0 and Open Source Software*, http://www.isrl.uiuc.edu/~twidale/pubs/mashups_patchworkPrototyping.pdf, at 9 (last visited 31 October 2007).

2 DRIVERS AND INHIBITORS OF INTEROPERABILITY

The current level of interoperability and openness of APIs and data create a rich environment for innovation, but there is instability in the system that gives reason for pause. Solutions to this instability could also have deep effects on the level of interoperability given the complexity of stakeholder needs and system requirements. For example, instability of this system is a concern for innovators, and data portability as part of achieving greater interoperability could be a mitigating force. Conversely, as programmers look for ways to increase stability, they may create one-off service level agreements and close linkages with data providers, and these relationships may reduce interoperability by increasing switching costs. The challenge is to achieve sustainability, maintaining the current level of openness to innovation, without creating too much stability -- entrenchment of the status quo -- at the expense of future extensibility.

2.1 Technology (architecture)

The messaging technology that underlies Web services is not a set-in-stone standard, but most developers appear to choose from a fairly short list of technologies. This means there can be some innovation at the technology layer, but it is dwarfed by the innovation that occurs at the content layer. Thus far, the openness of APIs and the availability of mashup programming tools seem to indicate that switching costs between messaging technologies are low.

Innovation grounded in interoperability can occur even without a single, agreed upon standard. Take the example of formats that transport data across Web 2.0²⁶ that have developed informally, yet function effectively. Really Simple Syndication (RSS)—the family of Web feed²⁷ formats used to publish frequently updated content such as blog entries, news headlines or podcasts—is one Web service that was standardized only after years of development and debate over its various iterations, and after it had already given rise to widespread innovation.

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. The basic idea of aggregating information gleaned from Web sites began in 1995, and, in the years following, RSS underwent numerous transformations. The original RSS, version 0.90, was designed by Netscape as a format for building portals of headlines to mainstream news sites. It was deemed overly complex for its goals; a simpler version, 0.91, was proposed and subsequently dropped when Netscape stopped pursuing it actively.²⁸ Version 0.91 was picked up by another company, UserLand Software, owned by Web pioneer Dave Winer, which intended to use it as the basis of its weblogging products and other Web-based writing software.²⁹

Meanwhile, a third group split off and designed a new format based on what they perceived as the original guiding principles of RSS 0.90 (before it got simplified into 0.91).³⁰ This format was called RSS 1.0. UserLand was not involved in designing this new format, and, as an advocate of simplifying 0.90, it decided to continue developing the 0.9x versions in spite of the new platform presented by RSS 1.0. UserLand created versions 0.92, 0.93, 0.94 of its original model, and finally came out with version RSS 2.0, around which most industry activity has been based.³¹ This meant that there were several re-

²⁶ O'Reilly at 2.

²⁷ Wikipedia, Web feed, http://en.wikipedia.org/wiki/Web_feed (last visited 31 October 2007).

²⁸ Mark Pilgrim, What Is RSS, December 18, 2002, <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>.

²⁹ Id.

³⁰ Id.

³¹ Id.

lated but distinct formats, all called “RSS” (not to mention another competing standard called Atom). Despite this multiplicity, RSS is a well-established part of the infrastructure of Web 2.0.

As the emerging identity infrastructure takes shape, it may serve as a lever for positive change in the mashup space as well. (See the related case study on Digital ID.) Digital ID has the potential to enable more secure and seamless transfer of personal information, thus allowing a greater degree of personalization and tailoring to make mashups even more useful and relevant than they currently are.

2.2 Market forces impacting mashup interoperability

In addition to the technical aspects of openness and lock-in, issues arising out of business concerns have taken shape. Actors on both the mashup side and the database/API side have stated their interest and have made moves towards commercializing their assets.³² This does not appear to be a symmetrical game, as the current successes of Web mashups have largely depended on players on the database/API side embracing the Web 2.0 business model and allowing free access, at least for the time being, to what they have to offer.³³ As one commentator observes, “[i]n the Internet era, one can already see a number of cases where control over the database has led to market control and outsized financial returns.”³⁴

Network effects are profoundly important in the uptake and sustainability of mashups. It follows logically that a mashup that has more users can be more sustainable than one that has a few; it can attract a wider range of funding possibilities, and its user base can serve as a voice that pushes for its maintenance. We have seen several examples of user communities vocalizing their displeasure at certain decisions by providers of online services. For example, *livejournal.com* experienced backlash from their communities after blocking community groups that were deemed to be bordering on encouraging illegal

³² Robert D. Hof, *Mix, Match, And Mutate - “Mashups” — homespun combinations of mainstream services — are altering the Net*, http://www.businessweek.com/magazine/content/05_30/b3944108_mz063.htm, Hartmann et al at 4, O’Reilly at 3.

³³ Floyd et al. at 9.

³⁴ O’Reilly at 3.

activity – but their blocks were overbroad and at times inappropriate.³⁵ Facebook.com experienced brief but wild backlash from their users after posting a new feature called ‘news feed,’ which was designed to parse the new and relevant information from a user’s profile and display it to his or her friends in condensed form. This was seen as a privacy violation at first – interestingly, it is now a main feature of Facebook. And after Digg initially bowed to legal pressure from the movie industry to take down references to a key for the encryption format used by HD-DVD, angry users completely flooded the site with submissions containing the key, after which Digg changed its mind and reinstated the postings against legal advice.³⁶ User groups can also serve to place pressure on those who threaten the services they use. At the technical level, user advocacy for Internet neutrality is another example of consumer vigilance in the Web context in favor of broadly open and interoperable systems. A powerful set of users whose mashup is being threatened by the loss of data or other part of the service could help to lay pressure to bring it back.

Another market-based mitigating factor in stack instability is that many mashups are created first as a public service, and only secondarily as a sustaining organization.³⁷ Foundations such as the Knight Foundation and the MacArthur Foundation have recently made grants that provide funding for mashups. This allows mashup developers to create valuable sites without having to go after ad revenue or the highly elusive group of users who will pay for a service. To be sure, this makes it harder for the mashup developer to justify paying for data, but it places less investment at risk than a for-profit model.

When there is a strong force asking for free data and consistently open (in the sense of accessible) interfaces, why might a data provider open an API? One reason is eyeballs: a data provider may create an open API to get their data and brand into more places. For example, an individual does not have to visit

35 See Alan Henry, *Livejournal Deletes 500 Accounts, Smarts at Backlash*, June 1, 2007, http://www.appscout.com/2007/06/livejournal_deletes_500_accoun_1.php. See also E-mail from Erica George, 31 May 2007.

36 See Wikipedia, *AACS encryption key controversy*, http://en.wikipedia.org/wiki/AACS_encryption_key_controversy#DMCA_notices_and_Digg (last visited 31 October 2007).

37 Interview with Adrian Holovaty, May 30, 2007. See, e.g., *MAPLight*, <http://www.maplight.org/map/us> (last visited 30 October 2007) (combining campaign contribution and legislative voting data), *New York City Coalition Against Hunger, Hunger Maps*, <http://www.nyccah.org/maps/index.php> (last visited 30 October 2007) (mapping soup kitchens and other hunger-related resources).

maps.google.com in order to see a Google map. This may increase market share through branding and brand availability. There is also the potential for advertising or revenue sharing.

Opening an API may also undermine competitors in strong market positions. An emerging example of this is the decision of Google to release its OpenSocial API for social networking applications with the support of every major social networking player except its fastest-growing competitor, Facebook. By tapping into the network effects not only of its own service, Orkut (which is popular in Brazil but not very much in the US), but of MySpace and others, Google seeks to undercut Facebook's advantages of being large and relatively open; OpenSocial promises to be a widely used API that is portable and based on widely used standards like HTML and JavaScript,³⁸ whereas Facebook's is not portable and based on the proprietary Facebook Markup Language.³⁹ Another example of using interoperability to undermine competitors is Google's entry into word processing and spreadsheets. While Microsoft has tried to leverage its dominance in the office software market through integration with some of its online resources, Google has leveraged its user base for Web e-mail and related services by releasing a free, online word processor and spreadsheet with open APIs to allow outside development. The fact that it is tightly integrated with Gmail (by a link next to every compatible document attached to an e-mail message) gives Google visibility as a potential Web alternative to Microsoft Office. In this endeavor, it is to Google's advantage to support Microsoft documents because doing so reduces the costs to users of switching to Google Docs, while Microsoft has no incentive to support Google Docs. Of course, in the long term this incentive structure could reduce competition once again – if a large portion of consumers adopted Google Docs in place of Office, they could become locked into Google as a vendor just as they had been to Microsoft, unless Google's APIs were so expansive that they allowed complete data export. Thus, and especially since the mashup space is too immature for patterns of competition to have become established, the long-term consequences of interoperability for competition are far from certain.

³⁸ See Google, OpenSocial Frequently Asked Questions, <http://code.google.com/apis/opensocial/faq.html> (last visited 2 November 2007).

³⁹ See Facebook Developers Wiki, FBML, <http://wiki.developers.facebook.com/index.php/FBML> (last visited 2 November 2007).

2.3 Social Norms

One of the quiet, yet powerful, forces governing activity online is social norms. The ethos of Web 2.0, at least at present, is very strong: the system on which users have come to participate in creating meaning is grounded in sharing data, code, and information. It would be very hard for a company to offer APIs for others to use and then abruptly shut them off, for fear of the consequences of violating online social norms.

This factor is yet stronger when the use of the API is for the public interest. For example, blogger Sami Ben Gharbia maintains a mashup that maps Tunisian prisons and provides information on the prisoners held in each one.⁴⁰ He has found that a mashup is a more intuitive and accessible way of presenting information than the “boring” reports and press releases common in the human rights community, effectively allowing him to reach a wider audience.⁴¹ In recent days, government agencies and news outlets also made use of mashups to disseminate information about the course of wildfires, evacuation orders, and shelter availability in Southern California.⁴² If a company providing an API took it away from these public-oriented efforts by cutting off access, or raised prices to the point at which a non-profit could no longer afford to take advantage of the technical interoperability, it could face a significant public relations backlash.

2.4 Law regulating Web services interoperability

The law that governs the mashup space primarily is private law established through a series of contracts. The provider, Company A, of the Web service offers a contract (often, but not always) to the entity that wishes to interoperate, Company B. There may also be relevant inbound and outbound contracts. Company A may have a contract with Company C for components of the service that Company A provides to others. Company B may have contracts that run to its end users, or perhaps to others who interoperate with the resulting mashup. The web of contracts can be highly complex.

⁴⁰ Tunisian Prison Map, <http://www.kitab.nl/tunisianprisonersmap/> (last visited 31 October 2007).

⁴¹ E-mail from Sami Ben Gharbia, 4 October 2007.

⁴² See, e.g., John Musser, “Track California Fires via Mashups,” <http://blog.programmableweb.com/2007/10/24/track-california-fires-via-mashups/>, 24 October 2007.

This contractual approach to governing Web services is imperfect, but it may be the best approach available. As a legal matter, depending on the particular mashup, a data provider may or may not be able to enforce its terms of use against a second-level mashup developer, i.e., one who produces a mashup based on someone else's mashup. Terms of service and service level agreements can reduce the flexibility of mashup creators either by prohibiting certain uses of data that would otherwise be feasible or by making moving to a competing data provider costly. In addition, some market participants have expressed the concern that the current potential for open innovation will be hampered by complicated and possibly conflicting terms of service between data sources. If these contracts are ambiguous or lead to lawsuits, innovation associated with mashups may be chilled. New businesses may seek assurances from data providers in the form of service level agreements, but that would only exacerbate the proliferation of potentially inconsistent or innovation-impeding contracts. Furthermore, if such one-off contracts became seen as necessary, many of the most important innovators of today – namely, individual users and nonprofits – could be deterred from participating in the mashup space.

One way that the Web services area might be coherently harmonized among private actors would be through common standards. Although we will revisit a form of this idea in the last section, standards currently play a secondary role in ensuring continued interoperability. Numerous technological standards have contributed to the ability of developers to create mashups, including XML and SOAP. These standards have made data more easily transmitted and understood between applications. However, there are no recognized standards governing the connection between applications in a broader sense, including acceptable uses and the sort of contractual terms discussed above. It is relatively easy to adhere to an extremely flexible standard like XML for organizing data because it can be adapted to virtually any use, but it is much more difficult to agree on anything like standard APIs among various applications simply because different applications do so many different things. Such standard APIs would have to be sufficiently flexible that they would potentially work in every Web service application, but specific enough that they can take advantage of the various features in each of them. While that combination is possible for data storage and transmission, it will be more difficult for applications that perform such widely divergent functions on that data. Standards are thus more likely to be applicable either within related areas (e.g., all ser-

vices creating maps from location data) or in non-technical areas such as best practices for how one should go about producing and documenting APIs or what uses or behaviors should be allowed.⁴³

Intellectual property law, mostly copyright and patent laws, could have complex effects on the mashup sphere. Although data providers expressly or implicitly waive any copyright claims against mashup developers from the use of their APIs, there is always the possibility that someone else's copyrighted information is included in that data. For instance, Amazon and Google book searching APIs reveal information about copyrighted books, and the authors of those books might have colorable copyright claims against those who make mashups with that information, depending on what portions of the books are accessible through the APIs. Perhaps more worrisomely, a developer can rarely predict when her mashup might infringe one of the thousands of poorly scrutinized, opaquely written software patents, whether held by a market participant (possibly even the data provider) or by a patent troll. This problem only compounds as mashups get more complex – if one mashup infringes a patent, every mashup based on it will likely also infringe. The potential for rent-seeking by patent holders could be another significant expense impeding the progress of mashups. Creative Commons and related movements seeking broader use of copyrighted works somewhat attenuate the copyright problem by ensuring that there is some sphere for mashups to operate safely, but there is no equivalent for patents. Patents are more expensive to obtain (such that few would obtain them only to open them up to the community) and more widely applicable (so it is much easier to infringe them without realizing it). But even within the relatively small universe of Creative Commons, there is the potential for license incompatibility if, for instance, some information is under a noncommercial license and some is not, or even if two applications use licenses that are very similar but differ on a minor point that makes it impossible to legally combine them.⁴⁴ Thus, mashup creators frequently risk

⁴³ However, at least one company, Netvibes, has attempted to create something like a cross-platform API for developing “widgets” that work on a variety of sites with a variety of data sources. See Netvibes, “What is the Universal Widget API?,” http://dev.netvibes.com/doc/universal_widget_api (last visited 18 October 2007).

⁴⁴ Larry Lessig has made a similar proposal in the context of Creative Commons to solve the problem of copyright licenses with similar goals that are incompatible because each requires that licensed content only be reused in works licensed under that specific license: Lawrence Lessig, CC in Review: Lawrence Lessig on Compatibility, November 30, 2005, <http://creativecommons.org/weblog/entry/5709>.

running afoul of various intellectual property laws and licenses, but fortunately, to date they have not been major problems.

At the moment, data providers have no obligation to provide APIs or indeed any access to their data. Even government sources often make data public reluctantly and in forms that cannot be easily used in mashups.⁴⁵ Thus, antitrust and competition law has not played a role in the evolution of Web services to date. However, should data sharing become more widespread and providing APIs commonplace, this legal area could become more relevant to mashups. It is not hard to envision any monopolist or near-monopolist attempting to shut competitors out of their Web services, and competition law could force them to keep those channels open. On the other side, too much collaboration among competitors in mashups could raise the specter of illegal collusion. For the moment, though, competition law remains distinctly in the background.

Some other areas, specifically DRM, have been the subject of specific legal regulation on both sides (in that case, laws such as the DMCA in the US, restricting DRM circumvention, and DADVSI in France, mandating DRM interoperability). In contrast, mashups, although growing quickly, are relatively immature as a technology. Many technology-related companies are still formulating a stance towards mashups, if they have given them any thought at all. Needless to say, there has not been any strong lobbying on the subject, and we have not seen much, if any, special legislation on the topics of Web services and mashups.

⁴⁵ Interview with Drew Clark, 9 October 2007.

3 BENEFITS AND DRAWBACKS OF INTEROPERABILITY

As we have seen in the other case studies, interoperability is not an end in itself. Nor is it necessarily a good thing per se. It is only desirable insofar as it leads to positive consequences, such as innovation, that outweigh the corresponding negatives, such as privacy or security issues. The previous discussion takes as almost implicit in the explosion of mashups that interoperability among Web services has led to widespread innovation. But we must consider what costs are associated with that innovation, as well as whether the current balance of costs and benefits will continue into the future.

3.1 Benefits for specific stakeholders

Those who provide Web services on which mashups can be built clearly stand to gain from that interoperability. Every mashup incorporating a Google map, for instance, displays the Google brand and interface and often includes value-added content for which Google receives advertising revenue. In essence, every for-profit mashup under the currently prevalent ad-supported model can be seen as a value chain in which each user indirectly distributes a small amount of money (which accumulates over time) from advertisers to each provider of services used in that mashup. Subscription-based, commission-based, and donation-based models are also feasible, and some or all of these are used in various contexts. The common theme to these business models is that the more a Web service is used directly or incorporated in others' mashups, the more revenue the service operator generates.

This virtuous cycle of interoperability and integration has the potential to radically reshape the landscape of users' experiences with computers and the Internet. At the extreme, we could see one or a few Web service providers creating platforms so ubiquitous and extensible that they become akin to operating systems on personal computers. Common applications like e-mail, word processing, photo storage and sharing, and instant messaging are already commonly available on Web-based platforms. Since social networking site Facebook opened itself up to outside applications, it has attracted more than 5,000 of them, with an average of 100 new applications added each day.⁴⁶ These applications give Facebook new capabilities that may attract new users and cause existing users to log in more frequently. The additional traffic generated gives rise to increased revenue for Facebook, and in some cases also directly from users or developers. Of course, developers target Facebook because its platform already has an extensive user base, so it seems likely that it will continue to grow more flexible and feature-rich for the foreseeable future, and network effects will only accelerate that growth.⁴⁷ However, it is far too early to suggest that Facebook or anything like it will eventually replace Windows, Linux, and other operating systems as the primary platform for innovation and application development on personal computers. Such predictions have been made in the past, most notably in the case of Java,⁴⁸ and so far have proven incorrect.⁴⁹

3.2 Benefits for society at large

The general benefits of innovation are shared broadly. Each Web service or mashup may have new features and capabilities that differentiate it from others previously available. Thus, individual users reap the benefits of interoperability every time they use a Web service or mashup. Moreover, public ser-

⁴⁶ Facebook Statistics, <http://www.facebook.com/press/info.php?statistics> (last visited 16 October 2007).

⁴⁷ See, e.g., Jamin Brophy-Warren, Networking Your Way to a Triple-Word Score, October 13, 2007, <http://online.wsj.com/article/SB119222790761657777.html>.

⁴⁸ In fact, Microsoft itself alleged that Java was effective competition to Windows in the operating system market when it was defending an antitrust suit in 1998. See Joel Brinkley, A New Tack Is Taken By Microsoft, December 3, 1998, <http://query.nytimes.com/gst/fullpage.html?res=9506E2DB103BF937A35751C1A96E958260>.

⁴⁹ A few attempts have been made to use Java as an operating system, but they have had relatively little success. See, e.g., Wikipedia, JavaOS, <http://en.wikipedia.org/wiki/JavaOS> (last visited 31 October 2007) ("As of 2006, Sun considers JavaOS a legacy system.").

vice mashups almost inherently create a benefit for society as a whole. Media Tracker, described above, enables a level of political transparency in the area of media and telecommunications in the US that otherwise would be extremely costly if not impossible. Sami Ben Gharbia's Tunisian prison mashup makes prominently available information about that country that otherwise would be invisible to those seeing Tunisia as only a tourist destination.

Having widespread development of new applications through interoperable Web services also fosters a more interactive culture of user-driven innovation, which Eric von Hippel and others have analyzed.⁵⁰ Individuals are not just the passive consumers of the radio-television era. They are not even just the active commenters and content generators of Web 2.0. They have the potential to tinker with the very code that governs the possibilities available on the Internet,⁵¹ and thus to push the limits of online culture. In economic terms, Web services interoperability could not only induce current market participants to innovate in ways that are open to further innovation, but also lower the barriers to entry so that the universe of potential market actors is greatly expanded to include individual users as well. By making it easier to innovate, open APIs and similar technologies could create not only innovation, but innovators.

3.3 Drawbacks of interoperability

The benefits of confronting the current instability of innovation in the Web services space through standardization should be balanced against the costs that could occur over time. While the Web services landscape currently contains many important actors, there is still the potential for vendor lock-in if developers focus on one platform. As we have seen in the other case studies, privacy and security are always concerns when determining the optimal level of openness in an interoperability ecosystem. In the Web services context in particular, having data hosted on the Web has the potential to diminish users' independence from any particular service provider. Technical standardization could lead to competition in non-technical (i.e., legal or market) dimensions, and if there is less than complete interoperability, the decision to use one data provider in a mashup may limit the possibilities available to future developers building off that mashup.

⁵⁰ See Eric von Hippel, *DEMOCRATIZING INNOVATION* (MIT Press 2005).

⁵¹ Cf. Lawrence Lessig, *CODE AND OTHER LAWS OF CYBERSPACE* (Basic Books 1999).

One problematic scenario for mashup interoperability would occur if standardization led to vendor lock-in for developers.⁵² It is possible that one platform, having become a standard through outcompeting alternatives or through some other process, could take on a gatekeeper role, restricting innovations that would be disruptive to its owner's business models or hindering growth through poor foresight in design. Even the most well-intentioned platform provider can end up impeding innovation if new applications require capabilities that the platform is not designed to support, but no alternative platform has enough users to make the applications sustainable. In other words, developers will be stuck with the shortcomings of any such standards except to the extent that, having hammered out a consensus, they are able to once again generate momentum to revise it.

This level of interoperability in Web services may also raise privacy concerns similar to those raised in the Digital ID case study. Some mashups and Web services will use or require personal information in some form. It seems likely that other mashups built on top of those services will allow an additional set of people to access that personally identifiable information. Thus, an individual mashup user may be communicating private data to an unknown number of service providers, each of whom may have different (and potentially non-user-friendly) privacy policies. The growth of Facebook applications, which can usually interact with personal data, may be an example of this phenomenon. On a more basic level, a platform provider would be able to track which user used which applications in which ways. Even if this information were kept on some sort of anonymous or pseudo-anonymous basis, there would nonetheless be the possibility that real people could be identified

52 For an example of this phenomenon from another area, consider that Microsoft Office has become a more or less universal standard in word processing, spreadsheets, databases, etc. It is also, to some degree, extensible, such that developers can create spam-fighting plugins for Outlook or custom links to a corporate document management system in Word. However, because these plugins and especially these documents have not been easily portable to other applications, there have been significant switching costs associated with adopting an alternative to Office like WordPerfect or OpenOffice. More recently, though, Microsoft has supported open-source translators to allow Office documents to be converted to non-proprietary OpenDocument files, as well as introducing its own OpenXML format. Thus, for a period of years before 2007, Office was a de facto standard that restricted flexibility to change products, but now it embraces greater interoperability.

from that data.⁵³ This has already become an issue, as some privacy advocates are concerned that Google and other online service providers already retain too much information about users, through cookies and otherwise.⁵⁴

As the continual explosion of Windows worms, viruses, and malware demonstrates, having a relatively freely extensible platform also opens the door to less desirable applications. Even as Facebook is a powerful platform because it enables programs to act on user data, some Facebook applications have abused that ability by spamming users' friends and using deceptive messages to spread virally.⁵⁵ Abuse is thus a limiting factor on the growth of features that could drive innovation when used properly. Taken to the extreme, this abuse drives the argument for locked-down "information appliances" that lack what Jonathan Zittrain calls "generativity," i.e., the ability to support user modification and innovation.⁵⁶ To the extent that Web services remain open, there will undoubtedly be security issues, whether relative annoyances like spam and pop-ups or more serious badware that could enable fraud or identity theft or disrupt the entire platform. Thus, the desirable innovation facilitated by interoperability will have to be balanced against the undesirable uses also made possible by the very same openness.

An extension of these privacy and security concerns is the potential for users' autonomy to suffer if Web service interoperability is expanded. Professor Zittrain has noted a number of reasons why consumers may be better off maintaining their data in forms and formats that they themselves control, i.e., on their own computers.⁵⁷ First, consumers may not realize the degree

53 This happened in mid-2006 after AOL released a large quantity of data containing anonymous users' search queries. See Michael Barbaro and Tom Zeller Jr., *A Face Is Exposed for AOL Searcher No. 4417749*, August 9, 2006, <http://www.nytimes.com/2006/08/09/technology/09aol.html>.

54 See, e.g., Electronic Privacy Information Center, *Gmail Privacy FAQ*, <http://www.epic.org/privacy/gmail/faq.html> (last visited 17 October 2007); Wikipedia, *Criticism of Google*, http://en.wikipedia.org/wiki/Criticism_of_Google#Privacy (last visited 17 October 2007).

55 See, e.g., Michael Arrington, *Facebook Takes Action Against "Black Hat" Apps*, August 16, 2007, <http://www.techcrunch.com/2007/08/16/facebook-takes-action-against-black-hat-apps/>; Eric Eldon, *Facebook clamps down on spam-driven application growth*, June 28, 2007, <http://venturebeat.com/2007/06/28/facebook-clamps-down-on-spam-driven-application-growth/>.

56 See Jonathan Zittrain, *THE FUTURE OF THE INTERNET – AND HOW TO STOP IT* (Yale University Press, forthcoming 2008).

57 For more detail on this discussion, see Zittrain, Chapter 8.

to which their data is or is not portable. If one's photos are hosted on Flickr, it may or may not be easy to transfer them to Facebook or Picasa. This is related to the use cost discussed below. Second, a corollary to the question of data portability is the fact that service providers, as distinct from the product providers who sell the software applications driving the non-Web-enabled computing ecosystem, can alter the set of capabilities available to users at any time. While Web service providers more often add features than take them away, they sometimes have reasons to do the latter; on the other hand, a user can always keep an old version of a program on her computer if she does not like the feature set in newer versions. If data is not portable, users may be caught in their own vendor lock-in, unable to easily demonstrate their frustration in the market by moving to a competing service. Third, at least in the United States, data held by a third party is subject to much less stringent privacy protections in the event that a government agency wishes to gain access to it. Consumers cede a lot of control by entrusting Web service providers with possession of their data, whether or not they realize the full consequences of doing so. Either way, the resulting drop in user autonomy weighs against an expansion of Web services interoperability.

If the technologies are made available according to common standards, then competition may focus on the terms of service associated with data use. While the standards decrease a technical switching cost, depending on the terms and the use case, they could instigate a corresponding legal switching cost. We see a similar example in the United States cellular phone market. Technically, a phone is designed to work on any network, but phone service providers install software that makes it extremely difficult to use the phone on any carrier. This may be better than no interoperability at all, as the technology is available in case the market and legal environment moves towards greater openness, but it does not help the current user.

There can also be a "use cost" associated with data portability. If one mashup developer uses Google Maps for their mashup, and someone layering a mashup on top of the first one wants to use Yahoo Maps instead, they may run into technical and legal issues that have not been fully understood or resolved. Could or would either Google or Yahoo prevent such a use? Would, as a practical matter, the decision by the first developer to use Google limit the ability of subsequent developers to make different choices? These issues are at present unknown territory.

4 MAINTAINING INTEROPERABILITY

The relevant policy issue in the context of this case on mashups is not how to establish interoperability (which exists by definition), but rather how to ensure that this level of interoperability is maintained so that the current rate of innovation continues. As a secondary matter, we concern ourselves with how to ensure that those who have invested in developing (i.e., the company that spends time and money to innovate on top of the interoperable system) or using (i.e., the end user who uses and may save personal data in a service) mashups do not lose their investments arbitrarily. These two concepts are related: if developers and end users cease to trust that their investments will pay off, then they are unlikely to participate in this innovative process to begin with.

Three potential approaches might help to achieve greater levels of sustainability in this context. First, we ought to consider a system of license interoperability, whereby terms of service and service level agreements are standardized.⁵⁸ Second, a standards process – ideally, an open standards process – may lead to increased sustainability. And third, governments might serve as a more or less active backstop, either applying existing law to combat threats to interoperability that happen to constitute legal violations or making a decision to safeguard interoperability as an explicit policy goal. This progression, from

⁵⁸ Again, Larry Lessig is informative on this topic: Lawrence Lessig, CC in Review: Lawrence Lessig on Compatibility, November 30, 2005, <http://creativecommons.org/weblog/entry/5709>.

informal to formal types of responses, suggests possible ways to achieve more sustainable interoperability in the context of mashups.

As discussed above, a potentially major issue for continued interoperability of Web services is the interplay among various licenses, terms of service, and service level agreements. Where these contracts are inconsistent or poorly compatible, innovation can be stifled or, perhaps worse, can continue only to be suddenly thrown off by expensive lawsuits. A useful response to this issue would be to adopt a model akin to Creative Commons of standardized terms of service and other licensing terms for mashups. Ideally, these standard contracts would be easily readable, portable across countries and legal systems, and communicable in a compact format such as a set of labels or icons. A mashup developer should be able to understand and rely on the fact that Service X is usable for only noncommercial purposes without advertising, while Service Y can be used in any application but inserts its own text ads, which cannot be stripped out without violating the license. Such standards could also be communicated through metadata, as with some Creative Commons works, or even as part of an API. If a programmer can simply check if the “useable for profit” attribute is set to “true,” she does not have to worry about license conflicts or changes at all. Even if licenses cannot be totally standardized, having a concise way of communicating the terms of use without forcing a developer to read a 10-page legal document would greatly attenuate the possible restraining effect of contracts on Web services interoperability. This could be achieved either by a movement for standards among Web service providers and developers or by an outside observer going through and classifying license agreements for major data providers.

More broadly, standards – preferably achieved through a predictable, stable, and open process, but possibly devised by some other mechanism – can ensure that mashup developers will continue to be able to make use of Web services in the future. Standard licenses are just one example of where standards can be useful. Use of existing standards for data messaging, such as XML and SOAP, are already widespread in Web services. Standard APIs, at least within particular types of Web services, would greatly reduce the learning curve for working with particular services and all but eliminate lock-in, as there would be very little cost to switch from one service to a competitor. Wherever standards can be achieved, they almost inherently enable interoperability – anything that is covered by a standard cannot be exclusively controlled or appropriated by one

actor that could shut off or delay innovation in an area.⁵⁹ While, as discussed above, it may be much more difficult to define general standards for applications than for data, they should be encouraged when feasible.

We also anticipate some role for government in maintaining Web services interoperability. Most likely, judicial or regulatory involvement will be limited to a reactive role driven by well-established legal principles. Tort and contract laws provide a familiar means for participants in this area, like any other, to police one another when appropriate. In particular, unfair competition and antitrust law can provide tools for either private actors or state regulators to force interoperability when failing to provide it constitutes unlawful monopolization or anticompetitive action. Of course, conceptions of what is “anticompetitive” could change if interoperable Web services become so widespread that not providing them would be unusual. Notwithstanding such a transformation, however, it seems likely that much can occur to hinder the level of interoperability among Web services that currently facilitates mashups without violating laws that were not drafted with them specifically in mind.

Thus, the alternative government role would be a more proactive one, passing and then enforcing regulations explicitly designed to drive innovation by promoting interoperability in Web services. Perhaps the most promising avenue for a positive, active government role would be a clearly drafted extension of unfair competition law in the Web services space to allow judges or watchdog regulators to respond on a case-by-case basis to threats to interoperability not currently considered illegal. Such a regulatory regime would have to be undertaken carefully, if at all. A promising instance of positive regulation – albeit in the direction of reducing government supervision – was the Internet tax moratorium in the United States, which has since 1998 prevented states from levying taxes on Internet access. (Such a tax might have hindered Internet penetration and reduced the potency of its network effects.) The lesson from this moratorium is that any such government action should not increase costs by imposing new burdens on users or service providers. Rather, the goal of any government regulation should be to keep the space free of unnecessary

⁵⁹ The exception here would be if a widely adopted standard turns out to violate a “submarine” patent, in which case every practitioner of the standard can be sued for infringement. A related example is the Eolas patent litigation, in which the owner of a patent apparently covering all Web browser multimedia plugins sued Microsoft in 1999; there was speculation that it would go after other Web browser producers as well.

barriers. This goal is related to the ongoing debate over Internet neutrality, but in a sense is broader, since Internet neutrality only concerns the ability of Internet service providers – the owners of the network infrastructure – to shape or restrict traffic to their customers, whereas Web services interoperability potentially relies upon all providers of Web services or mashups, ranging from individual developers to major corporations. While it is easy to imagine what a hands-off regulation might look like in the Internet neutrality context, Web services is such a broad and fast-changing area that it would be difficult for a regulator to coherently support interoperability without running the risk of either issuing regulations that are already obsolete by the time they are adopted or inadvertently hindering interoperability by creating disincentives for online service providers to make themselves open to mashups. Thus, while there is potential for government to play a positive and proactive role in facilitating mashup interoperability, it would be very challenging to construct a new set of legal rules that would have more positive effects than the scenario in which the government does not intervene.

Most clearly among our three case studies, the area of Web services demonstrates the manner in which interoperability can stimulate large-scale innovation. Web services enable the integration of diverse sources of data, making information more accessible and useful to consumers through mashups. Although contracts surrounding Web services may be inconsistent or poorly understood, standardized licenses would provide a method of facilitating continued interoperability before contract incompatibility becomes a serious problem. Standards in general already facilitate the transmission of data, and could ensure that a wide range of aspects of the current Web services space remains open to innovation. Law could remain in the background, preventing substantial threats to interoperability that cross some existing legal line, or could become a more prominent factor in the development of mashups. Finally, it is important to remember that, while the benefits of mashups and Web services so far seem to create a strong net positive, there are associated costs that may become more substantial in the future. Too much interoperability could leave the entire system open to concerns of privacy and security, and paradoxically could lead to vendor lock-in through convergence on a de facto standard. By using the tools described above and related technologies like interoperable Digital ID, society may nonetheless be able to reap the benefits of mashup innovation while keeping the associated problems to a minimum.