*By examining and contrasting Web 2.0 and SOA, the authors envision a new Internet of Services that incorporates the best of both.*

**Christoph Schroth and Till Janner**

# Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services

Recently, the relationship between Web 2.0 and service-oriented architectures (SOAs) has received an enormous amount of coverage because of the notion of complexity-hiding and reuse, along with the concept of loosely coupling services (see http://blog.hbs.edu/faculty/amcafee/index.php). Some argue that Web 2.0 and SOAs have significantly different elements and thus can not be regarded as parallel philosophies (see http://edgeperspectives.typepad.com/edge_perspectives/2006/04/soa_versus_web_.html). Others, however, consider the two concepts as complementary and regard Web 2.0 as the global SOA. In this article, we investigate these two philosophies and their respective applications from both a technological and business perspective.

## DEFINITION OF TERMS AND RESEARCH APPROACH

Tim O'Reilly coined the term Web 2.0 to describe a quickly growing set of Web-based applications. SOA is considered the philosophy of encapsulating application logic in services with a uniformly defined interface and making these publicly available via discovery mechanisms (C.M. MacKenzie et al., "Oasis—Reference Model for Service Oriented Architecture 1.0," 2006; http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf and A. McAfee, "Will Web Services Really Transform Collaboration," *MIT Sloan Management Review*, vol. 46, no. 2, 2005, pp. 78-84).

O'Reilly identifies seven major characteristics inherent to the Web 2.0 philosophy: first, the Web is considered a platform for building systems that are "tied together by a set of protocols, open standards, and agreements for cooperation" (T. O'Reilly, 2005, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html). The exploitation of collective intelligence of Web users, ownership of mission-critical data, and the end of the software release cycle are quoted as central characteristics as well. The use of lightweight programming models that allow for loosely coupled applications, the use of diverse media and devices for the consumption of Internet-based applications, and the realization of rich user experiences represent further paradigms inherent to the concept of Web 2.0.

In 2006, Högg et al. conducted an in-depth investigation of 40 successful Web 2.0 applications (R. Högg et al., "Overview of Business Models for Web 2.0 Communities," *Proc. Gemeinschaften in Neuen Medien*, Technische Universität Dresden, 2006, pp. 23-37). They condensed their respective characteristics into the following statement, which works as an underlying definition for this article: "Web 2.0 is defined as the philosophy of mutually


Published by the IEEE Computer Society
1520-9202/07/$25.00 © 2007 IEEE

maximizing collective intelligence and added value for each participant by formalized and dynamic information sharing and creation."

Since the late 1990s, many definitions of SOA have been published (P. Frost and S. Frost, *Component-Based Development for Enterprise Systems: Applying the Select Perspective*, Cambridge Univ. Press, 1998 and G. Alonso et al., *Web Services Concepts, Architectures, and Applications*, Springer, 2004). The Oasis Reference Model for SOA (MacKenzie et al., 2006) defines SOA as

> … a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

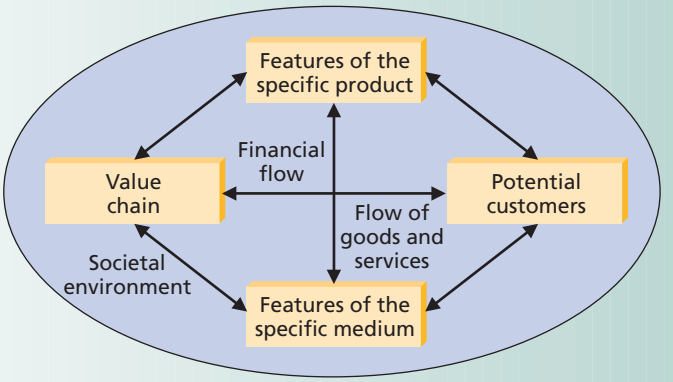
According to this model, a service provider might publish a well-defined interface on a registry that enables other stakeholders to retrieve and loosely couple the offered service with their own services.

The research approach applied for this comparative analysis is based on the examination of 40 real-world use cases of both Web 2.0 and SOA. In the case of Web 2.0, 40 popular applications (Högg et al., 2006) were selected, while case studies provided by Systeme Anwendungen und Produkte in der Datenverarbeitung (SAP), IBM, and Gartner were leveraged to investigate SOA solutions.

The Institute for Media and Communications Management (MCM) at the University of St. Gallen's business model framework (see Figure 1) was leveraged to structure the analysis along seven major components (Högg et al., 2006):

1. *Features of the specific product* comprise the design of a product or service and its value for the customer.
2. *Features of the specific medium* define the technological foundation of a product.
3. The *customers* component refers to the target groups of an offered product or service and explains their respective business needs.
4. The *value chain* is devoted to reflecting all players that are involved in the production and delivery of a product and their respective interrelationships.
5. *Financial flow* identifies revenue models and explains the roles that different stakeholders play.
6. *Flow of goods and services* describes the stakeholders' activities that are essential for the product's or service's creation.
7. Last, the *societal environment* reflects relevant outside influences on a business model (such as legal and social aspects and competitive situations).

**Figure 1. The Institute for Media Communications Management's (MCM's) business model framework.**




## WEB 2.0 VS. SOA

In comparing each product or service, it's worthwhile to consider Web 2.0's and SOA's pros and cons more thoroughly. In the following paragraphs, we analyze Web 2.0 and SOA based on the MCM business model framework's seven major components.

### Features of the specific product or service

According to Högg et al. (2006), we can classify Web 2.0 applications as follows:

- Communities that aim to unify their users by means of a common ideal such as social networking or knowledge sharing.
- Platforms or tools that help users create and share content with a broad audience (for example, Web logs and online directories). Mashup platforms let users retrieve content or functionality from arbitrary sources, mix it with other resources, and expose it for further reuse by other applications.
- Online collaboration tools support users in collaboratively performing certain tasks, such as maintaining time schedules or processing text online.

In comparison, the following are ways we can differentiate SOA use cases:

- First, SOAs allow for a cross-organizational integration of services. By adhering to common standards for the description of their service interfaces, corporations are enabled to setup loosely coupled electronic business transactions with other companies and thus automate business transactions in a quickly changeable fashion.
- Second, SOAs facilitate the intraorganizational integration of disparate services. On the basis of a central integration layer (often referred to as an Enterprise

Service Bus, or ESB), heterogeneous applications can be encapsulated and composed to a seamlessly integrated IT landscape (R.W. Schulte, "Predicts 2003: Enterprise Service Buses Emerge," Gartner Research Note, 2002; http://www.gartner.com/DisplayDocument?doc_cd=111977).
- Third, SOA-based application development significantly reduces development time thanks to the availability of reusable application building blocks.

The first major analogy between product design in the fields of Web 2.0 and SOA is the notion of reusing and composing existing resources. Both concepts let users reuse, remix, and enrich existing resources and components to new and potentially higher-level applications. The second commonness is the affinity to collaboration and coupling of remote resources or services. Both Web 2.0 and SOA applications enable the loose coupling of distant and possibly heterogeneous resources. A third apparent resemblance between Web 2.0 and SOA is the shared principle of agility and the support of permanent structural change.

**Because of their frequent implementation in the corporate context, SOAs are subject to requirements that don't exist in the case of most Web 2.0 applications.**

Web 2.0 and SOA also have divergent elements. First of all, many Web 2.0 applications incorporate a social aspect, as they facilitate human interaction and also mainly deal with human-readable content (such as text and pictures). In contrast, conventional SOAs merely aim at interconnecting dispersed business functionality and facilitating seamless machine–machine collaboration. Second, Web 2.0 is clearly about presentation and user interface integration, whereas SOA deployments are more abstract and less visible to its users. Third, and last, the degree of ex-ante determination and involved governance (McAfee, 2005) is a key differentiator between Web 2.0 and SOA. Because of their frequent implementation in the corporate context, SOAs are subject to requirements that don't exist in the case of most Web 2.0 applications and thus underlie governance mechanisms (P. Weill and J.W. Ross, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Harvard Bus. School Press, 2004).

## Features of the specific medium

As argued in Kolbitsch and Maurer, Web 2.0 doesn't stride along with a fundamental technological innovation, but is facilitated by a number of technologies (J. Kolbitsch and H. Maurer, "The Transformation of the Web: How Engineering Communities Shape the Information We Consume," *J. Universal Computer Science*, vol. 12, no. 2, 2006, pp. 187-213). First of all, the Representational State Transfer (REST) is an architectural style that enables Web clients to interact with arbitrary Web resources in a uniform way (R.T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," doctoral dissertation, Information and Computer Science Dept., Univ. of California, Irvine, 2000). The Really Simple Syndication (RSS) format supports the easy aggregation of content from arbitrary sources in the Web. AJAX represents a composite of several other technologies that together allow for rich user experiences, which is one of the key paradigms of Web 2.0 applications.

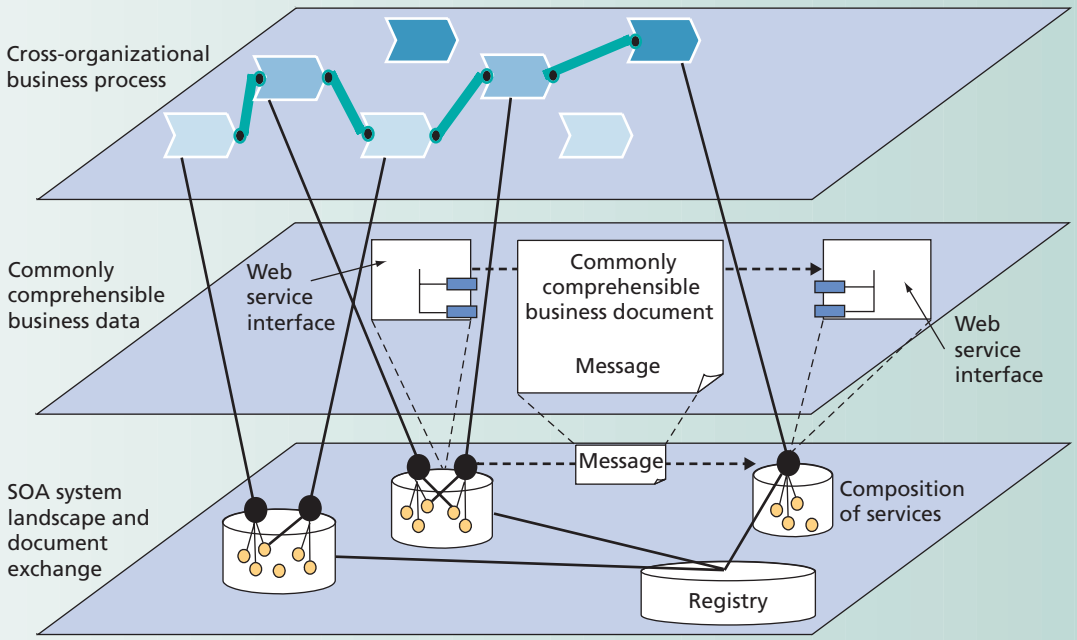
As opposed to lightweight Web 2.0 technologies (Hagel, 2006), SOAs rely on a set of more complex standards. As the Web Service Description Language (WSDL) and SOAP-based Web services are the most widely spread standards used to set up SOAs, these are in the focus of our technical analysis.

As Figure 2 shows, possibly heterogeneous applications (visualized as orange circles) are encapsulated as services and can be composed to new, aggregated functionality (black circles). WSDL is used for uniform service interface descriptions, while the Universal Description, Discovery, and Integration (UDDI) standard allows for open-service discovery. SOAP specifies a protocol for message exchange between services, while no widely accepted standard exists for data semantics (see the middle layer of Figure 2). The Business Process Execution Language is frequently used as a standard for orchestrating different services into one process choreography (see the upper layer in Figure 2).

As a first major technical similarity, standards applied in both Web 2.0 (such as RSS and REST) and SOA applications (such as WSDL) support the loose coupling of remote applications via uniform interfaces. Second, Web 2.0 and SOA technologies also share the notion of complexity hiding and reduced programming effort with the help of uniform descriptions of interfaces and data structures.

We can identify two major differences between the technological basis of Web 2.0 and SOA: First, we can make a distinction between the terms *syndication* and *coordination*. Many technologies used in the Web 2.0 context focus on static syndication of content and services, while SOAs often incorporate service coordination protocols (different services are invoked in a predefined sequence). The second difference refers to semantic interoperability. Web 2.0 applications directly involve human beings who are fault tolerant with respect to the information they're provided, whereas SOAs typically are limited to the mere interaction of machines, which aren't flexible to formal errors or semantic differences.

**Figure 2. Basic service-oriented architecture (SOA).**




## Potential customers

In the Web 2.0 context, basically every Web user can be regarded as a potential customer. Instead of heading for a small number of huge customers, Web 2.0 applications involve the bulk of private users or small businesses, also known as "the long tail" (C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*, Hyperion Books, 2006). As argued by O'Reilly, Web 2.0 is about leveraging customer self-service and thus is able to "reach out to the entire Web, to the long tail and not just to the head" (O'Reilly, 2005). In the examined SOA cases, medium-sized or larger corporations are the customers of choice. These aim to introduce SOA as a software design principle, as a possibility to streamline and harmonize internal IT landscapes or to set up cross-organizational business relationships. Because of their substantially different application domains, Web 2.0 and SOA applications serve different customer needs and requirements.

## Value chain

In the Web 2.0 context, traditional value chains are broken up to a large extent and substituted by loose networks of providers and consumers (Högg et al., 2006). Every user may publish his or her own content or functionality on the Web and thus become a platform operator, consuming resources or reusing them to compose new applications and make them publicly available. In contrast, the high technical complexity inherent to SOAs requires the existence of one or several expert players within the value chain who build and provide solutions for their customers. In contrast to this one-to-many value chain model of numerous SOA use cases (where one expert serves many clients), Web 2.0 value chains are mostly loosely coupled (many-to-many) networks of self-managed users who can offer and consume resources via the Web.

## Financial flow

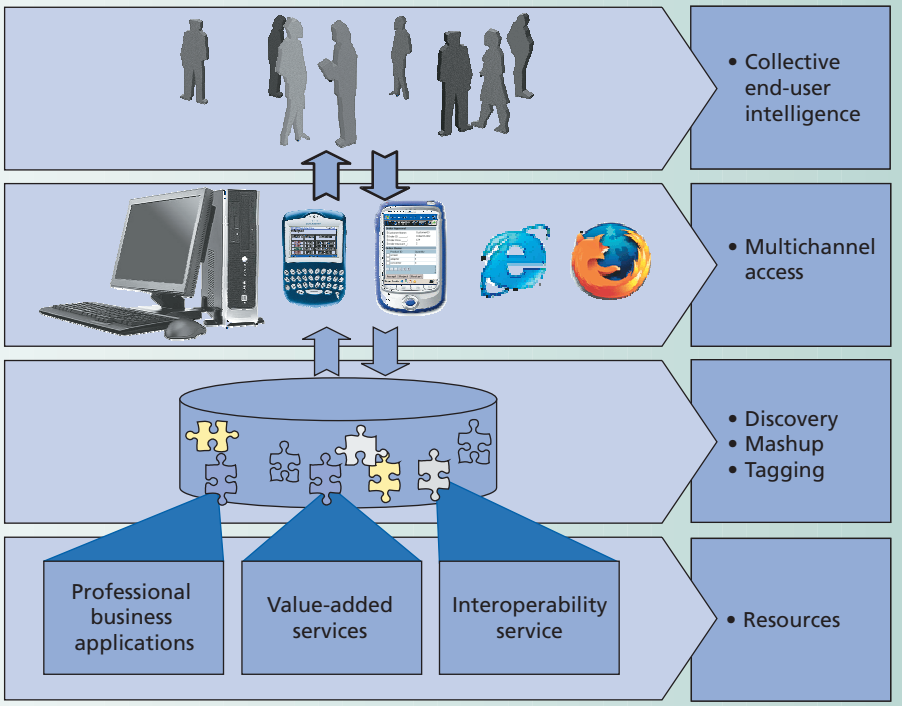
Revenue models of Web 2.0 applications strongly differ from traditional models in the software industry. First, software applications aren't packaged and sold as over-the-counter products anymore, but provided as services (or the "end of the software release cycle," Högg et al., 2006). For a significant share of Web 2.0 applications, the number of users is part of their central value, so providers generally don't introduce service fees. They take this approach to avoid limiting the amount of users, but then this approach leads to challenges with monetizing their products.

The integration of third-party advertisements represents one possible solution to this revenue model dilemma. SOA revenue models in most cases follow a more traditional approach, as license fees are charged for using the respective solutions. This could change in the next few years as enterprise services become easily retrievable and usable via the Web.

## Flow of goods and services

The decisive difference between typical product creation and provision in the fields of both Web 2.0 and SOA results

**Figure 3. Internet of Services (IoS) basic architecture.**




from the end of the software release cycle in the Web 2.0 context. In this context, services remain in a state called *the perpetual beta*, in which the product can continuously be improved (O'Reilly, 2005).

Also, numerous Web 2.0 application providers don't rely on traditional marketing and sales activities but aim at *viral marketing*—that is, recommendations autonomously propagating from one user to another. In the SOA context, large solution providers such as SAP and IBM mostly treat SOA products as software artifacts that are packaged and then sold to customers.

## Societal environment

As we previously noted, Web 2.0 use cases considerably benefit from the lack of formal guidelines and governance mechanisms with which SOAs typically must cope. However, cases exist where providers of Web-based platforms have been enforced to review the content they publish, thereby limiting the growth and dynamics of the offered solutions (Högg et al., 2006). As opposed to most Web 2.0 applications, SOAs are subject to clearly defined regulatory frameworks (such as the Sarbanes-Oxley Act), because they mostly exist in the corporate context. The design, provision, maintenance, and coupling of services must be compliant with legal frameworks and thus they do not allow for flexibility as observed within the Web 2.0 context.

## CONVERGING WEB 2.0 AND SOA CONCEPTS

In general, the philosophies of Web 2.0 and SOA serve different user needs and thus expose differences with respect to the design and used technologies of real-world applications. However, recently numerous novel use cases demonstrate the great potential of combining the technologies and principles of Web 2.0 and SOA.

One major example of the convergence of the two philosophies is the emergence of a global SOA that we refer to as Internet of Services (IoS). Up to now, normal Internet users with little IT sophistication haven't been able to easily retrieve and use certain services. This is because these services mostly reside within company boundaries and are only accessed for professional use in a corporate context. However, the provision of easily accessible services for end users might drive a novel generation of Internet use and allow for the ad-hoc setup and configuration of IT-supported business models. Without an intuitive "face" toward human users, Web-based services will remain fairly unusable by the common Web surfer, because currently these interfaces (such as those described in WSDL) are designed to be processed by machines.

Figure 3 visualizes a basic IoS architecture that's based on a combination of principles and technologies from both Web 2.0 and SOA. Resources that are accessible via the Web are registered in platforms and can thus be discovered, tagged (to collect user evaluation and to allow for folksonomies), and also mashed up (composed and interlinked with the goal of designing new resources) according to the users' requirements (O'Reilly, 2005).

Arbitrary stakeholders can provide and host services, thereby leading to a global market with decentrally organized platforms that act as brokers. The actual users can access these platforms for discovery purposes via intuitive interfaces that facilitate tagging and mashing activities without requiring any coding effort. Then we could leverage all of the different channels (such as PCs or mobile devices) to enter and use the platforms. The establishment of an open architecture that comprises possibilities for human beings to use and interact with Web-based resources has the potential to drive the development of a global mesh of services.

A combination of principles from both Web 2.0 (user self-service and collective end-user intelligence) and SOA (a composition of reusable building blocks) can facilitate the wide dissemination of many resources. Examples include professional business applications, value-added services (including location-based services), and interoperability services (for example, applications that can be leveraged by trading partners to initiate business-to-business transactions).

Enterprise mashups represent one specific use case of this type of architecture that could easily be situated at the interstice of Web 2.0 and SOA (A. Mulholland et al., *Mashup Corporations: The End of Business As Usual*, Evolved Technologist Press, 2006). Technically unsophisticated business experts would be empowered to model and deploy business models in an extremely quick and efficient fashion. The interconnection of presentation-layer-focused Web applications to internal SOA implementations could be of significant value for enterprises, as this could extend their services' reach to the Web for further use and composition by their business partners and customers.

The firm Kapow Technologies (see http://www.kapowtech.com) recently announced the release of a technical solution for these kinds of enterprise mashups. It focuses on empowering users of their Web integration platform to integrate resources available via the Web on several different levels.

First, this helps to seamlessly integrate and expose user-interface content and functionality from arbitrary sources as a new service on the Web. Second, it helps establish application mashups by composing and coupling applications that are accessible via REST or WSDL interfaces into new services.

The key goal of such platforms for creating enterprise mashups is to provide businesses with speed, flexibility, and agility in creating and changing cross-enterprise applications on the basis of a merely visual, complexity-hiding modeling interface. The envisioned result of this approach that combines Web 2.0 and SOA technologies (for example, AJAX, SOAP-based Web services, RSS, and REST) is achieving fast and business requirements-driven content and application integration.

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) provides another excellent example of the combination of Web 2.0 and SOA (T. Janner et al., "From EDI to UN/CEFACT: An Evolutionary Path Towards a Next Generation e-Business Framework," *Proc. 5th Int'l Conf. e-Business*, 2006, King Mongkut's Univ. of Technology; http://www.alexandria.unisg.ch/Publikationen/30346). It proposes a novel approach for the standardization of business processes. Instead of prescribing yet another fixed standard for the establishment of cross-organizational SOAs, the UN/CEFACT envisions establishing a publicly accessible repository featuring a basic set of modeling building blocks that can be used, extended, and tagged by the users according to their actual business requirements.

In this work, we thoroughly contrasted the two philosophies of Web 2.0 and SOA from a technical and economic perspective. We identified numerous similarities—but also clear differences—in each of the seven criteria for comparison.

Recent applications show the paramount importance of unifying the two philosophies to drive the next wave of value creation within and across enterprises. Web 2.0 incorporates a social philosophy that we consider complementary to the technology-focused SOA philosophy, as it provides techniques and design principles that strongly facilitate the active consumption of Web-based resources. By integrating the long tail of Web users (Anderson, 2006) into application design via all relevant media channels and on the basis of easily usable platforms that allow for discovering, mashing, and tagging diverse resources, we can realize a comprehensive IoS. ■

***Christoph Schroth*** *is a Research Associate and PhD candidate in information management at the University of St. Gallen and Systeme Anwendungen und Produkte in der Datenverarbeitung (SAP) Research in Switzerland. Contact him at christoph.schroth@sap.com.*

***Till Janner*** *is a Researcher Associate at SAP and is working toward his PhD in information management at the University of St. Gallen. Contact him at till.janner@sap.com.*