

The SOA Magazine

Feature Article



Commercializing Services: Web Services Distribution Channels and SOA

by Joe Labbe

Published: January 3, 2007 (SOA Magazine Issue III: January 2007, Copyright © 2007)

[Download this article as a PDF document.](#)

Abstract: Exposing web services to the outside world is much more complex than creating and maintaining services geared towards internal consumption. While internally focused projects have their technical challenges, outwardly focused web services initiatives bring to the fore a whole host of non-IT related issues such as business strategy and marketing. Those who proceed with such projects with the same mindset that made their internal projects successful run a significant risk of failing.

Web services initiatives aimed at serving the needs of non-captive customers and partners are akin in effort to that of creating a new business channel and not merely a systems integration project. In order to be successful in these efforts, you must clearly understand your organization's objectives, your customer's needs and the Web Services Distribution Ecosystem.

Introduction

For the most part, the Web services story to date has been about service creation and the technical aspects of publishing and governing internally focused projects. Lots of smart people have spent significant amounts of time and money understanding the tools, stacks and standards required to get in the game and reap the benefits associated with increased business agility, standards-based systems integration and asset reuse. While the results associated with these internal integration projects have been encouraging, only now are organizations starting to realize the real bang for the SOA buck is achieved when Web services are exposed outside the organization to their customers and partners.

By exposing Web services externally, organizations are better positioned to tap new markets for existing offerings, create new and innovative offerings, seize new partnering opportunities and expand their ability to conduct general business-to-business transactions. However, the process of exposing Web services to the outside world is not merely a matter of establishing a services DMZ or adding another layer of security. Rather, this process represents the creation of a new distribution channel that involves strategies, technologies, methods and relationships rarely considered when implementing Web services for the purpose of internal systems integration. Welcome to the concept of a distribution channel for Web services.

When standing on the shoulders of successful internally-focused Web service projects, it's easy to underestimate the effort associated with exposing similar functionality to the outside world. Why is this? For starters, most first attempts to expose services outwardly are driven by the same mindset that made those internal projects successful – an emphasis on the technical aspects of systems integration, standardization and coordination. Unfortunately, external constituents are usually not captive, have varying requirements, competing priorities and are difficult to communicate with in a uniform manner. In other words, they're customers.

From the customer's perspective, conquering the technical fetes that made your internal projects successful is a mere formality. Customers not only expect you to transparently solve these integration issues, but they also demand your services be packaged in a form and presented at the most appropriate time and place to serve their needs. Go talk to

your marketing and channel people for a preview of what's in store as you venture down this path.

Understanding the WSDC Model

Before we define what a Web Services Distribution Channel (or WSDC) is, let's first state its purpose. Regardless of whether a service is provided on a pay-per-use basis or is part of a larger, subscription-based or analog transaction, a common metric used to measure its success is usage level. The assumption being that most Web services help facilitate additional revenue generating activities so higher levels of service consumption result in greater revenues. This being the case, the purpose of creating a WSDC is to directly or indirectly increase the volume of revenue generating activities.

Now that we understand the simply stated goal of creating a WSDC, let's apply a definition. A Web Services Distribution Channel is the collection of strategies, methods, technologies and relationships required to get an organization's Web services in front of customers in an appropriate form and at the most opportune time and place to dramatically increase usage.

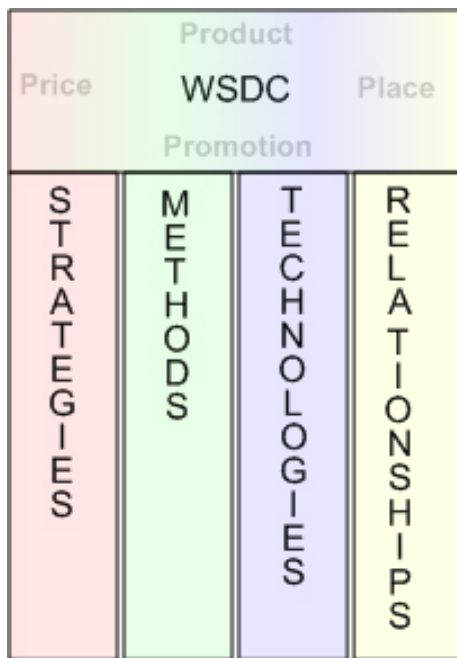


Figure 1: The fundamental WSDC pillars.

In order to better understand the WSDC concept, let's break the definition down into two bite-sized chunks and supply additional commentary.

WSDC Definition Part 1

"A Web Services Distribution Channel is the collection of strategies, methods, technologies and relationships required to get an organization's Web services in front of customers..." (Figure 1) This part of the definition hammers home the point that creating and maintaining a WSDC is much more complex than merely integrating disparate systems. No doubt the technologies involved are important. However, when services are exposed outside the organization for the purposes of creating new markets or expanding existing markets, more decisions need to be made with regard to the market (product, price, place, promotion) than technology. In fact, a well designed WSDC usually has a business plan backing it up that clearly defines the following:

- Product (service) and anticipated usage patterns
- Financial model (including pricing matrix)
- Customer profile
- Marketing plan

- Competition (existing and future competitors)
- Success metrics
- Risks (technical, brand, market and legal)

WSDC Definition Part 2

“...in an appropriate form and at the most opportune time and place to dramatically increase usage.” This part of the definition means that the services comprising a WSDC need to be packaged and promoted in a customer centric way to fuel usage growth. Services need to exist at a level of granularity and in a form that is consistent with customer needs. While many organizations now have the pre-requisite technical skills to present Web services, where they fail most often is delivering these services at a time and in a form that coincides with customer requirements. Remember the outside world wants to interact with your services on their terms and shouldn't require an understanding of how your systems are organized.

Before we dive into the details of creating the distribution channel, it's important to note your WSDC needs to be part of your organization's overall SOA strategy. Just because external parties will want to consume your services in alternative forms (in terms of granularity and technical interfaces), this does not mean you shouldn't leverage the service development, testing and maintenance expertise you've acquired while implementing internal offerings. In fact, the services you make available to the outside world should (wherever possible) rely heavily upon: the services you've already created, the governance infrastructure you've implemented and the best practices you employ within the existing SOA. Failure to serve internal and external consumers from the same SOA eliminates a lot of the reuse, scalability and agility benefits you hope to achieve.

As illustrated in Figure 2, when implementing a WSDC it is important to be cognizant of the following facts:

- internal and external consumers' requirements will vary significantly
- your organization has only one enterprise SOA that must be designed in a granular and flexible fashion so it can accommodate both sets of constituents
- the services that comprise a WSDC not only apply to service offerings that “are” the product an organization is selling (e.g. a credit report), but also to those services that facilitate normal business-to-business transactions (e.g. check order status, issue shipping notice, etc.)

One Organization - One SOA

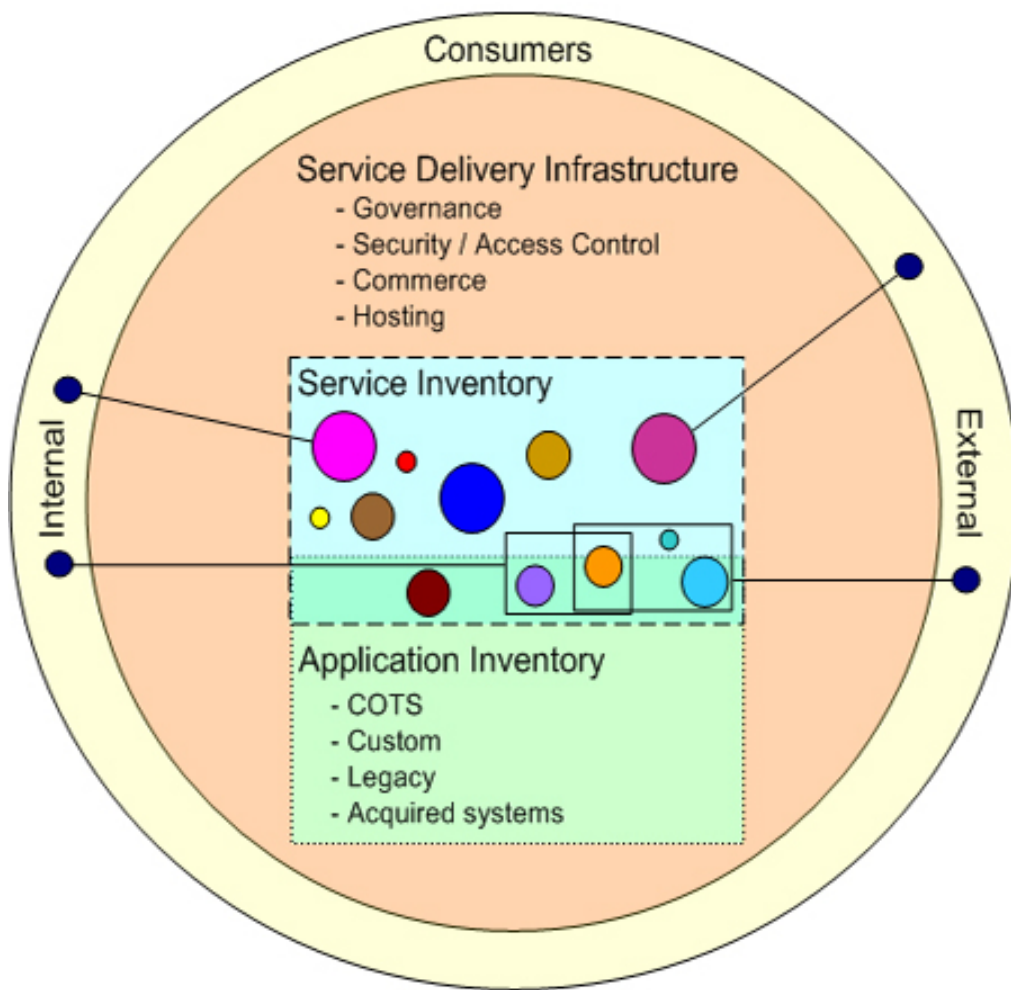


Figure 2: Internal and external services should leverage the same SOA.

A Process for Creating the Web Services Distribution Channel

Like most aspects of business, there is no one-size-fits-all strategy for achieving WSDC bliss. Since every organization is different in terms of goals, capabilities and markets served, no two WSDC implementations will look the same. However, there is a process within which an organization can engage to help ensure its WSDC is effective. At a high level, the keys to implementing an effective WSDC are:

- clearly understanding your organization's objectives (i.e. what business goal are you trying to achieve by exposing the proposed services?)
- an appreciation for the Web Service Distribution Ecosystem (functions, roles and participants)
- knowing which questions to ask and how to interpret the answers

Step 1 – Assess Service-based Offerings

The creation of a WSDC requires a frank assessment of an organization's offering in terms of:

- market applicability
- state of readiness
- sustainability

Market Applicability

Before entering a new market, it's critically important to know if customers want what you plan to sell and how they will use it. Moreover, you also need to know what alternatives will be available to your target customers. This being the case, the first part of assessing your service-based offering is determining its applicability in the marketplace by answering questions such as:

- Who is the target customer?
- What will their usage patterns look like?
- Who else is providing similar services?
- Who might enter the space if you're successful?
- What might customers pay for your services based on: the value the WSDC provides, their ability to pay, competing offerings and expectations?

State of Readiness

Once you've determined there is a viable market for your services, you then need to perform a gap analysis to ascertain the steps required to make those services actually marketable. For example, are the services ready for delivery? If yes, are they in a consumable form? Are the required hosting and commerce components in place to facilitate the service as a business? Are there relationships upon which providing this service is critically dependent? In order to firm up the financial model that supports the WSDC, you need to intimately understand the effort and cost required to bring your services to your target market.

Sustainability

Next, you need to accurately assess service maintenance requirements. During this process, you'll need to factor in questions pertaining to operational, scalability and contingency issues. Remember, this channel "is" an online business and must be highly available and reliable. Implementing a WSDC is not a fire-and-forget type of initiative. Finally, there is a growing number of service providers who can cost-effectively provide important pieces of support and maintenance infrastructure, so be sure to evaluate your options carefully.

Step 2 – Develop Financial Models and Marketing Plan

Now that you understand which services you'll be selling, who will buy them and what it's going to cost to deliver and maintain them, you're now ready to construct financial models and develop a marketing plan. Depending upon the kinds of services you'll be offering, creating financial models can be a complex process often requiring its own methodology. However, for brevity's sake, let's touch on some of the highlights below.

The first step in creating a financial model is constructing a pricing matrix -- a critically important component because the entire financial model stems from the revenue projections derived from anticipated pricing. Keep the following questions in mind when developing this part:

- What does it cost to produce these services?
- What do customers currently pay for similar services? What are their expectations?
- What are competitors charging?
- How will these services be assembled with other services that your organization or others produce?
- Do your services help customers increase revenues, reduce costs or merely facilitate existing transactions? (This question helps determine the tangible value your service provides to customers.)
- How will the customer be charged for the service - pay per use, subscription, or included in a broader analog transaction?

Marketing Plan

Before you can build a full financial model, you need to understand how your services will be marketed and the associated costs of those marketing activities. This is where a number of organizations fail either because they grossly underestimate marketing costs or merely fail to see the need to market services at all. This happens often when external Web services initiatives are looked upon as technical systems integration projects and not as business channel activities. Remember, if the goal is to increase service usage and thus derive additional revenues, you need to make sure customers know about them, understand their value and how to use them. If they don't, you need to know why.

After the marketing plan is created, you are ready to construct the financial model – tracking projected revenues to projected expenses. The completion of the financial model represents a seminal milestone because this model is the final piece of the puzzle required to make a “go/no-go” decision. If you've performed the proper due diligence and have confidence in your data, the decision to proceed or not should be coming into focus. While it's important to be passionate, be smart and honest in your appraisal. All the passion in the world is not going to make an unviable service successful.

Don't Forget The Lawyers

While exposing Web services for public consumption can open many new doors of opportunity, it also brings with it a host of new legal concerns. If you've done your job right, it's quite possible your services will be consumed by parties and used in ways you did not anticipate. Moreover, it's even possible your services will be combined with services from other publishers to create new and unintended offerings. Because of this, you need to consider the legal costs and ramifications associated with creating the WSDC. Areas to consider include: service usage rights and responsibilities, the creation of derivative works, data retention policies and service level agreements. While these issues should not scare you away from creating a WSDC, you need to seriously consider their cost and implications before proceeding.

Step 3 – Design

Now that your WSDC is backed by a sound strategic plan, it's time to get down into the technical weeds and design the actual services. Given the fact there are tomes written on the subject of designing Web services, there's no need to go into the specifics of that process here (an excellent design reference is Erl's “Service-Oriented Architecture: Concepts, Technology, and Design” [REF-1]). You should stick with methodologies that are in tune with your: development team size and style, application development tooling, application runtime and hosting environment and security policies. However, there are a couple of issues to consider that transcend environmental factors:

Service granularity

If the purpose of creating a WSDC is to increase usage of your service-based capabilities, then it's important to design your services at a level of granularity that supports your most common, and hopefully, profitable usage patterns. While there is no “one right” way to organize candidate operations and services, there clearly are wrong ways, depending upon your objectives. As a rule of thumb, your services should be atomic, isolated and have clearly defined functional boundaries. Remember that customers will often combine them with other services in ways you never anticipated so make sure the design is flexible enough to accommodate unforeseen use cases. Each new use represents a found revenue generation opportunity.

Leverage your infrastructure

WSDCs assume a business grade runtime infrastructure that handles where needed: monitoring, metering, security, compression, optimization, scalability, contingency and commerce. This being the case, don't build these functions into your services. Your services should merely contain the hooks required so the infrastructure can provide these support functions in a uniform and manageable manner to all WSDC services.

Step 4 – Implementation and Testing

No magic here. If you've designed and documented your service architecture properly, developing, testing, and delivering the services should be a fairly straightforward, though not necessarily easy, process.

Step 5 – Investigate User Distribution Acceleration Options

While the Web services story to date has been mostly about internal systems integration projects, it has also been a “programmer’s story” in the sense that developers hold the key to the users’ ability to access a service from within the applications they use. In other words, if a user wants an application to incorporate a function that exists in a Web service, a programmer must first modify the application’s source code to embed the call, re-test and re-deploy the application. The cycle of service discovery, assessing its importance (more often, assessing the importance of the person requesting the service), modifying the application at the source code level, re-testing and re-deploying is the top inhibitor to the growth of Web services deployments. That means for every new WSDC you create, there may be a significant delay before customers incorporate your service into their applications. That’s not good.

What *is* good is that there are technologies and techniques available that can help accelerate the delivery of your WSDC services into foreign applications. Think of them as packaging mechanisms that present your services when they’re needed, where they’re needed and in a form that is most applicable to the user. For example, there is a class of desktop integration software called indirect connectors that allow end users to integrate Web services with their applications without having to modify the application’s source code or requiring the assistance of a programmer. Indirect connectors allow software users to link application screen fields to the inputs and outputs of Web services so data can be extracted from application screens, processed by Web services, and have the results pasted into the same or different application fields.

In addition, the tooling for composite software and mashups is getting more sophisticated every day so it’s important to understand how these tools can help broaden the reach of your WSDC.

Step 6 – Execute Marketing Plan

The final step in implementing a WSDC is executing the marketing plan. In order for a WSDC to be successful, target customers need to be made aware of its existence and value. Awareness is created through a combination of innovative packaging techniques that are consistent with the way customers want to consume your services and partnerships with other providers who have a vested interest in your success.

This vested interest is created when: your WSDC creates additional value to the partner’s offering, a partner’s paid transaction is tied to your service, or the partner is compensated each time your service is executed. These marketing relationships come in all shapes and sizes and are more apt than any of your other channels to involve organizations with whom you’ve never considered partnering. So it’s important to understand the Web Services Distribution Ecosystem and its participants in order to maximize the channel’s value.

The Web Service Distribution Ecosystem

The Web Services Distribution Ecosystem (WSDE) is the universe of roles and participants that facilitate the delivery of publicly available Web services to consumers. Since WSDCs are merely organization-specific overlays of these roles and participants, the key to creating an effective WSDC starts with an understanding of the WSDE. Figure 3 depicts the current view of the WSDE. As the concept matures, it is inevitable new roles will emerge while others will be either consolidated or deprecated.

Web Services Distribution Ecosystem (WSDE)

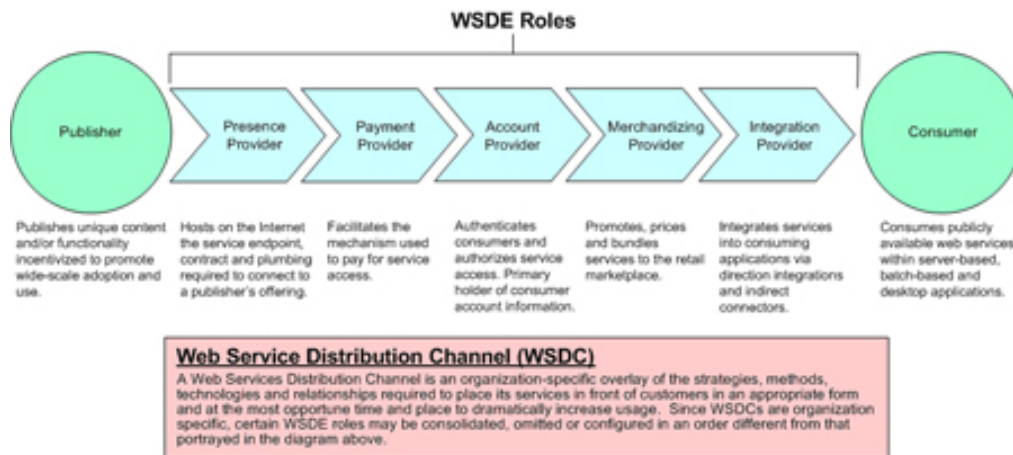


Figure 3: The components of the Web Services Distribution Ecosystem (WSDE).

[click for larger image](#)

Publisher

A publisher is a producer of unique content and/or functionality “incentivized” to promote wide-scale adoption and reuse. Publishers create or assemble the actual services consumers need to conduct business. Multiple publishers can participate in one WSDC or a publisher’s offering can be part of multiple WSDCs.

Presence Provider

The presence provider is responsible for all the technical aspects of: hosting the service endpoint on the Internet, the service contract (in form of a WSDL), and all the plumbing required to connect to the publisher’s services. The presence provider is also responsible for supplying the infrastructure and management functions required to make a service fault tolerant and scalable.

Payment Provider

The payment provider facilitates the mechanism used to pay for service access. These mechanisms may include credit card transactions, service hit counting or subscription tracking. If a WSDC is based on a free service offering, the payment provider role will most likely be omitted.

Account Provider

The account provider authenticates the consumer’s identity and authorizes service level access. It represents the primary holder of the user contact information and is a role often consolidated within the merchandizing provider role described below.

Merchandizing Provider

The merchandizing provider promotes, prices and bundles services for sale to consumers. It serves as the WSDC’s “retail” outlet and, by doing so, often “owns” the consumer relationship. This role represents the primary vehicle by which service awareness is created.

Integration Provider

The integration provider is responsible for integrating services into server-based, batch-based or desktop applications via direct integrations and/or indirect connectors.

- *Direct connections* are made when an application, server-based based process, BPM process, or mashup is modified at the source code level to contain explicit calls to a given service or proxy. While there are ways to more “loosely couple” the connections between applications and specific services, source code level modifications must be made for an application to directly integrate with a WSDC. Direct connections often use an application’s native API and usually facilitate deep integrations.

- *Indirect connections* facilitate the integration of applications and WSDCs without requiring source code level changes to the application. These connections are established by external connectors that integrate with software from the “outside-in” and do not require the knowledge or permission of the integrated application. These integrations tend to be “more shallow” in scope with regard to any one specific application but allow end users to link a broad array of applications without coding or assistance from a developer.

Like merchandizing providers, integration providers play an important role in creating awareness for a given service offering.

Conclusion

Exposing Web services to the outside world is much more complex than creating and maintaining services geared towards internal consumption. While internally focused projects have their technical challenges, outwardly focused Web services initiatives bring to the fore a whole host of non-IT related issues such as business strategy and marketing. Those who proceed with such projects with the same mindset that made their internal projects successful run a significant risk of failing.

Web services initiatives aimed at serving the needs of non-captive customers and partners are akin in effort to that of creating a new business channel and not merely a systems integration project. In order to be successful in these efforts, you must clearly understand your organization’s objectives, your customer’s needs and the Web Services Distribution Ecosystem.

Finally, an organization should have only one SOA strategy of which its WSDC is merely a part. While it’s critically important to accommodate the varying requirements of internal and external consumers, it’s just as important to make sure these accommodations are made while leveraging your existing inventory of services and infrastructure.

References

[REF-1] “Service-Oriented Architecture: Concepts, Technology, and Design”, Thomas Erl; Prentice Hall, 2005; ISBN-0-13-185858-0 (<http://www.soabooks.com>)

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL

