

Compliments of **IBM**

IBM Limited Edition

# SOA Design Principles

FOR  
**DUMMIES**<sup>®</sup>  
A Wiley Brand

## **Learn:**

- To apply the design principles of SOA
- The three fundamental aspects of SOA
- Why SOA is simply good design
- How to make your enterprise engaging



**Claus T. Jensen**



***SOA Design  
Principles***  
**FOR  
DUMMIES<sup>®</sup>**  
A Wiley Brand  
IBM LIMITED EDITION

**by Claus T. Jensen**

**FOR  
DUMMIES<sup>®</sup>**  
A Wiley Brand

## SOA Design Principles For Dummies®, IBM Limited Edition

Published by  
**John Wiley & Sons, Inc.**  
111 River St.  
Hoboken, NJ 07030-5774  
[www.wiley.com](http://www.wiley.com)

Copyright © 2013 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published by John Wiley & Sons, Inc., Hoboken, New Jersey

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, the Wiley logo, For Dummies, the Dummies Man logo, A Reference for the Rest of Us!, The Dummies Way, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. IBM and the IBM logo are registered trademarks of IBM. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

**LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.**

For general information on our other products and services, or how to create a custom book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact [info@dummies.biz](mailto:info@dummies.biz), or visit [www.wiley.com/go/custompub](http://www.wiley.com/go/custompub). For information about licensing the brand for products or services, contact [BrandedRights&Licenses@Wiley.com](mailto:BrandedRights&Licenses@Wiley.com).

ISBN 978-1-118-63957-3 (pbk); ISBN 978-1-118-64023-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# Table of Contents

.....

<b>Introduction</b> .....	<b>1</b>
About This Book .....	1
How This Book Is Organized .....	1
Icons Used in This Book.....	2
<b>Chapter 1: Designing the Future with SOA</b> .....	<b>3</b>
Seeing Where SOA Has Been — and Where It’s Going .....	3
Then: Controlled environments.....	4
Now: Loss of centralized IT control .....	4
Making enterprises engaging .....	5
Seeing (and seizing) opportunities .....	7
Catching the Enterprise Service Bus .....	7
Understanding Three Fundamental Aspects of SOA.....	8
<b>Chapter 2: The Elements of Good SOA Design</b> .....	<b>11</b>
Service Orientation at the Core.....	12
Process Integrity at an Internet Scale .....	13
Integration with Enterprise Capabilities and Back-end Systems .....	14
Mediating and integrating with an ESB.....	15
Integrating at the edge .....	15
A Basis in Industry Standards .....	16
Leveraging and Extending Open-Source Technologies.....	17
Providing the Platform for a Growing Ecosystem .....	18
<b>Chapter 3: Going Mobile with SOA</b> .....	<b>19</b>
Why Mobile Matters .....	19
The Fundamentals of Mobile SOA.....	21
APIs.....	21
Managing business APIs .....	22
Connecting business APIs via edge gateways.....	22
Apps.....	23
Enabling apps.....	24
Creating apps .....	24
App stores .....	24

<b>Chapter 4: Socializing with SOA</b> .....	<b>27</b>
Interacting via Social Media.....	27
Connecting People and Knowledge .....	29
<b>Chapter 5: Putting Your Head in the Cloud with SOA</b> .....	<b>31</b>
Taking Cloud to the Business .....	32
Transitioning to Cloud Computing .....	32
<b>Chapter 6: Tapping Big Data Insight with SOA</b> .....	<b>35</b>
Tapping into Existing Data Streams .....	35
Putting data to work.....	36
Predicting the future .....	37
Using Data from Disparate Sources .....	38
Providing Business Insight .....	39

# Introduction

---

**T**hese days, many people refer to mobile communications, social business, cloud computing, and/or Big Data as “the next cool thing.” These “cool things” don’t come out of thin air, however. Like all good new business initiatives, they’re based on a foundation of tried-and-true engineering and design principles, which is where service-oriented architecture (SOA) comes in.

What you may call basic SOA or information technology (IT)–centric SOA applies SOA principles to the flexible integration of an enterprise’s IT portfolio. But controlling interactions through well-defined services and interfaces certainly applies to mobile, social, cloud, and Big Data solutions as well. SOA is simply good design for IT integration and for the business solutions of an ever-more-connected and ever-more-distributed world.

## *About This Book*

I believe in the power of the good design principles of SOA and the need to enable new solutions on a stable, well-trying design base. This book is my guide to applying the design principles of SOA to the business problems of today, not just the IT problems of yesterday. It tells you how to use service-oriented thinking and standards-based integration for process integrity and business insight at Internet scale.

## *How This Book Is Organized*

You can read this book cover to cover, if you like, or start with a topic that interests you. Here’s what you’ll find in each chapter:

- ✓ **Chapter 1**, “Designing the Future with SOA,” briefly reviews past uses of SOA and introduces new ones.

- ✓ **Chapter 2**, “The Elements of Good SOA Design,” presents the design principles of SOA.
- ✓ **Chapter 3**, “Going Mobile with SOA,” is the first of four chapters on new business uses for SOA.
- ✓ **Chapter 4**, “Socializing with SOA,” gives you insight into applying SOA in a social business context.
- ✓ **Chapter 5**, “Putting Your Head in the Cloud with SOA,” discusses how cloud computing is built on SOA design principles.
- ✓ **Chapter 6**, “Tapping Big Data Insight with SOA,” shows you how to gain business insight from the stream of interactions flowing through your business.

## Icons Used in This Book

You’ll find the following icons in the margins of this book:



A Tip icon provides pointers that you may find useful in your own work.



The Remember icon points out some important things to keep in mind.



Although I hope that you’ll read every paragraph of this small book, feel free to skip the advanced material marked with a Technical Stuff icon.



Pay close attention whenever you see a Warning icon, which marks information that you need to know for legal or other reasons.



# Chapter 1

---

# Designing the Future with SOA

.....

## *In This Chapter*

- ▶ Looking at the IT-centric past
  - ▶ Understanding the business-centric future with SOA
  - ▶ Defining an engaging enterprise
  - ▶ Discovering three aspects of SOA
- .....

**M**any wise people in the industry seem to find service-oriented architecture (SOA) uninteresting simply because it has become mainstream. But the very fact that SOA has become mainstream is what allows you to apply SOA capabilities to new mission-critical challenges. In fact, new business and information technology (IT) initiatives, such as mobile communications, social business, cloud computing, and Big Data, require more (and more complex) integration than ever before.

## *Seeing Where SOA Has Been — and Where It's Going*

In this section, I show you where SOA has been so you can see where it's going.

## *Then: Controlled environments*

Historically (with or without SOA), IT organizations have attempted to control integration by providing a managed development environment and deploying solutions onto approved middleware and runtimes. SOA design principles were applied primarily to the flexible integration of the IT portfolio of the enterprise within a tightly controlled environment.

The problem is that this approach no longer works.

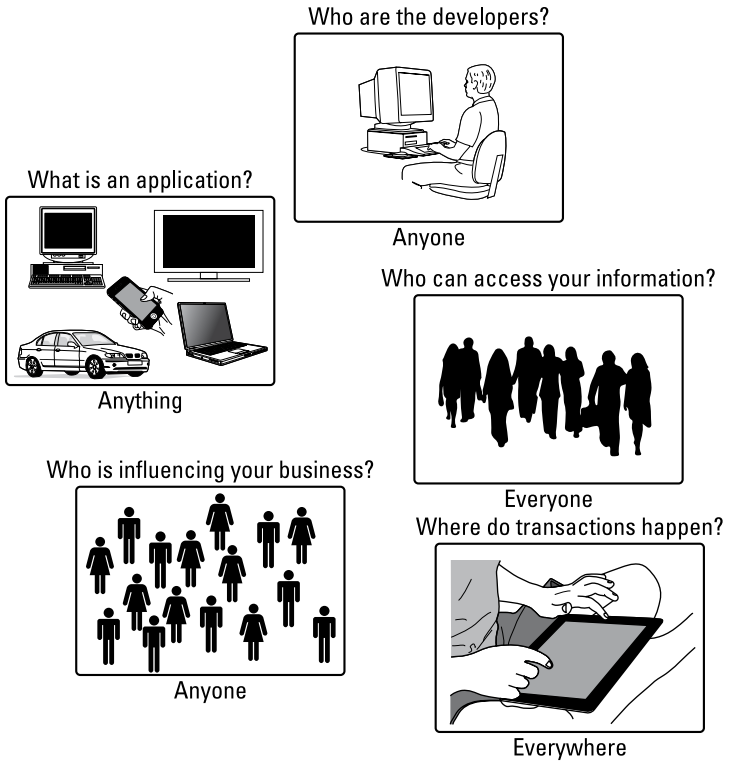
## *Now: Loss of centralized IT control*

Today, developers (unless they're prevented from doing so) download and use any library or open-source tool that's recommended by their social network peers. Fundamentally, centralized IT has lost control of both the development environment and the runtime environment. Here are a few examples:

- ✓ Businesses outsource the creation of mobile apps to agencies.
- ✓ Partners provide more and more of the backbone for business operations.
- ✓ Teenagers want to build their own mashups on cellphones.
- ✓ Businesses want to sell information and insight as marketable commodities.

Figure 1-1 illustrates that anyone can be a developer; applications can run on anything from a cellphone to a refrigerator; transactions can happen anywhere and anytime, and it's impossible to predict who may influence your business or access your information.

Because a controlled, centralized IT environment no longer works, applying SOA only to IT problems is no longer sufficient. Just because the world is changing, however, doesn't mean you should sit down and cry about the past. These changes give you the opportunity to act in a bigger and more important context, as you see in the next section.



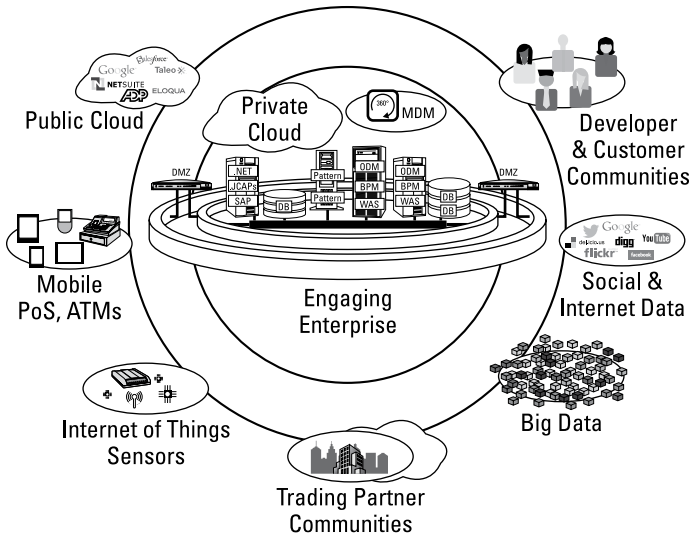
**Figure 1-1:** Development and control aren't just for IT departments anymore.

## *Making enterprises engaging*

SOA was never only about enterprise application integration. The idea that you can control interactions through well-defined services and interfaces certainly applies to mobile, social, cloud, and Big Data — even, or perhaps especially, in an environment in which neither development nor operations is fully under the enterprise's control. You should apply the power and value of SOA's good design principles to bigger, more challenging business problems. (I discuss those principles in detail in Chapter 2.)

Becoming an engaging enterprise is at the top of the priority list for most businesses today. An *engaging enterprise* has

transformed itself from being transaction-centric to being interaction-centric (see Figure 1-2).



**Figure 1-2:** An engaging enterprise is interaction-centric.

In an interaction-centric business,

- Two or more participants in the enterprise's network conduct business interactions that are of interest to the enterprise through systems that may not be fully under the enterprise's control. The interaction is often not predefined; it happens as it occurs.
- Participants in these interactions range across IT systems, devices, and people, which may be interacting inside the enterprise or remotely (via mobile devices or in the cloud). Often, the participants interact simultaneously over many channels.
- Many interactions aren't transactional in nature and don't represent any kind of contract or commitment.
- Finally, the enterprise itself doesn't need to take part in all business interactions of interest. The enterprise may need to know and understand what's being said about it in social media, for example, yet that need doesn't make the enterprise a participant in such interactions — merely an interested observer.

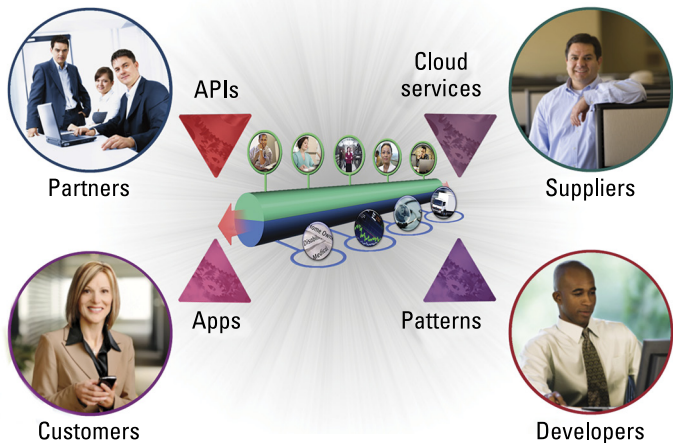
### *Seeing (and seizing) opportunities*

Becoming an engaging enterprise isn't just about innovating internally with developers on staff. You must find ways to profit from the innovation that exists within external communities — communities that you don't control and can only influence. You need to mediate loosely coupled participants, provide location and time transparency, and transform unplanned interactions into structured conversation and business insight, thereby supporting the dynamic evolution of smart transactions.

In fact, you can enable engagement in your enterprise by applying the most basic SOA pattern, which I cover in the next section.

### *Catching the Enterprise Service Bus*

The Enterprise Service Bus (ESB) pattern (see Figure 1-3) is the foundation of the loosely coupled nature of SOA-based systems. Mediation is the key to this pattern.



**Figure 1-3:** The ESB pattern mediates between consumers and providers.



Most people remember the consumer and provider parts of the pattern but often forget the mediation that occurs between them. In the context of the engaging enterprise, however, mediation may be the most important part.

Classically, ESBs have mediated message formats, protocols, and other IT characteristics in an IT transactional context. In an engaging enterprise, however, mediation occurs based on skills, availability, and location. An engaging enterprise connects and mediates people, devices, cloud environments, and so on and is still based on the ESB pattern.

## *Understanding Three Fundamental Aspects of SOA*

Depending on your viewpoint, you may be interested in the services that are part of SOA solutions; you may be interested in the architecture of those services and how they interact with the business processes that consume them; or you may be interested in a service-oriented way of thinking about your business. The following three fundamental aspects are part and parcel of SOA:

- ✔ **Service:** A repeatable business task (such as checking customer credit or opening a new account)
- ✔ **Service orientation:** A way of thinking about your business through linked services and the outcomes they provide
- ✔ **SOA:** A business-centric architectural approach based on service-oriented principles

Classically, certain design principles have been associated with SOA:

- ✔ Standardized service contracts
- ✔ Loose coupling
- ✔ Service abstraction
- ✔ Service reusability
- ✔ Service autonomy
- ✔ Service statelessness
- ✔ Service discoverability
- ✔ Service composability

These principles largely address the characteristics of services rather than the larger systems within which those services operate. The design principles for services remain important, but you also need to consider the system-wide design aspects that I discuss in Chapter 2.





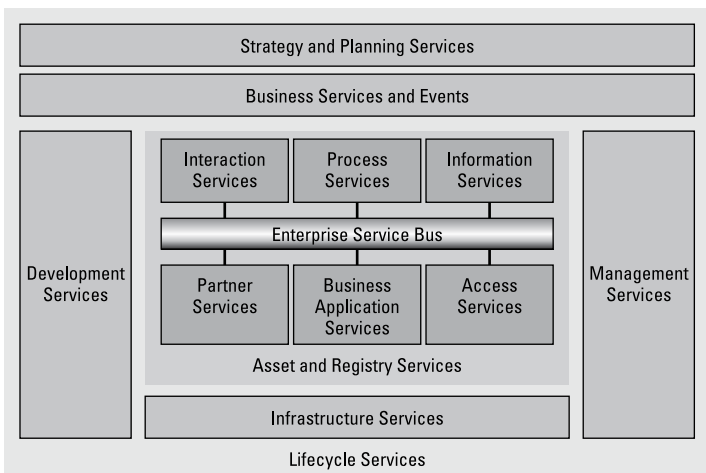
## Chapter 2

# The Elements of Good SOA Design

### *In This Chapter*

- ▶ Recognizing good design
- ▶ Discovering the design principles of SOA

What makes an engaging enterprise good, and what key design principles lead you there? While reference models such as the IBM SOA Reference Model (see Figure 2-1) outline the technology components required to build, run, and manage SOA solutions, they don't help you engineer good business solutions. To engineer good business solutions, thinking in terms of services is a start, but service orientation isn't all there is to good SOA design, as you see in this chapter.



**Figure 2-1:** The IBM SOA Reference Model defines a good SOA platform.

From the perspective of an engaging enterprise (see Chapter 1), what you most likely want to engineer and control are the business interactions that are important to the enterprise. In that context, saying that SOA is simply good design is a powerful way of asserting that SOA is the way you do business when you design solutions that integrate systems of engagement with systems of record.

The good design principles of SOA guide and enable the establishment of an engaging enterprise. These principles are

- ✔ Service orientation at the core
- ✔ Process integrity at an Internet scale
- ✔ Integration with enterprise capabilities and back-end systems
- ✔ A basis in industry standards
- ✔ Leveraging and extending open-source technologies
- ✔ Providing the platform for a growing ecosystem

In this chapter, I give you a closer look at these principles and discuss how they apply to an engaging enterprise.

## *Service Orientation at the Core*

Service orientation doesn't begin with technology; it begins with the mindset of thinking about your business and the world around you in terms of functional components. Thinking in terms of services and processes transcends any particular channel or business unit and provides a uniform mediated architecture that can connect the key stakeholders inside and outside the enterprise.

As you can no longer expect to control the development environment or the runtime environment in its entirety (all of which I discuss in Chapter 1), the best you can do is control the interfaces between the systems of engagement and your core enterprise capabilities.



Service orientation is all about providing well-defined interfaces (services) with clear business semantics and runtime-enforced security and workload policies.



Whether you call these well-defined interfaces *services* or *business APIs* depends mainly on context and audience; both concepts are derived from the principles of service orientation.

## Process Integrity at an Internet Scale

Have you ever been in one of these situations?

- ✔ You call your local utility to get help with some problem. The utility's automated voice response system asks you all kinds of questions before you finally get through to a human being — and the first thing that human being does is ask you the same questions all over again.
- ✔ You're standing in a check-in line at the airport and discover that the airline forgot to tell you that your flight has been canceled.
- ✔ You didn't get an item that you ordered via e-mail because someone at the other end of the transaction forgot to enter your order into the vendor's order system.

Few things are more annoying than finding that a business forgot about its past interactions with you or failed to deliver on the promises it made. Integrity — in all things — is very important to your customers, and it shapes their perception of your business.

Business processes are everywhere — not just in the form of automated process flows, but also in any structured sequence of business interactions that's part of the operational fabric of the enterprise. Yet flawless execution of mission-critical processes is difficult. Process integrity has three main elements:

- ✔ **Transaction integrity:** Ensures that individual updates of business and IT resources are linked and processed as a single unit of work, all completing successfully or being rolled back
- ✔ **Interaction integrity:** Ensures that elements of people's interactions with business and IT systems are intact and

remembered wherever and whenever those interactions occur

- ✓ **Information integrity:** Helps deliver trusted, secured information to business processes, regardless of delivery channel, operational platform (IT or people), and information lineage

Each element is enabled by a robust SOA infrastructure.

Classically, the notion of transaction integrity has been limited to a single runtime container and a time span measured in microseconds. The advent of Business Process Management (BPM) extended the notion of transaction processing to horizontal business processes. An engaging enterprise needs Internet-scale process integrity.

Internet scale isn't just about wild swings in transactional volume; it's as much (or even more) about extending integrity across time, space, locations, and channels. Thus, process integrity at Internet scale requires a transition from database-centric transaction principles (such as two-phase commit) to more loosely coupled transaction models that include compensation and recovery, which are better suited for long-running or asynchronous business interactions.



In an engaging enterprise, business processes aren't limited to what happens inside the enterprise's own walls: The capability to change and integrate those changes both internally and externally is important to success.

## *Integration with Enterprise Capabilities and Back-end Systems*

Does your bank's mobile banking service show a different account balance from the one you see when you go to an automated teller machine? Have you ever been unable to complete an online transaction and wound up on a page that says you need to fax or call someone? Those scenarios (and others like them) are signs of lack of integration between an enterprise's systems of engagement and its systems of record.

## Mediating and integrating with an ESB

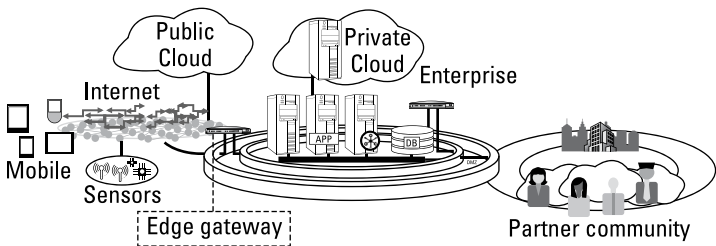
Integration with the enterprise's transactional backbone is very important for developers who want to support new business models and extended ecosystems. The basic ESB pattern is an excellent basis for loosely coupling external and internal participants. As I mention in Chapter 1, the mediation part of the ESB pattern is often overlooked and undervalued, yet this part of the pattern enables mediation between external business APIs and internal services in a way that doesn't require recoding or extension of the existing transactional backbone.



For details on ESB, flip back to Chapter 1.

## Integrating at the edge

It's tempting to say, "Of course you need to integrate with the back-end system. What else are services about?" But exactly what needs to be integrated, and how do you control that integration when you aren't in control of the way that the service consumer is being developed? In the world of the engaging enterprise (see Figure 2-2), the control point isn't a user interface framework but an *edge gateway* — a point at the edge of the enterprise where the uncontrolled meets the somewhat controlled.



**Figure 2-2:** Engaging the world through an edge gateway.

The notion of using an edge gateway to control the perimeter of the enterprise isn't new in terms of security and access control, but in an engaging enterprise, an edge gateway needs to do much more:

- ✔ **The workload needs to be controlled at the edge.** This control includes monitoring and enforcing business service-level agreements (SLAs) with partners and consumers.
- ✔ **Accounting at the edge is new.** Not only do SLAs need to be enforced, but also in many cases, consumers of a service or API need to pay for that use as part of an indirect business model.
- ✔ **Analytics are important at the edge.** The engaging enterprise needs to understand, analyze, and learn from the totality of its business interactions, many of which are visible only at the edge, given that making intrusive changes in all back-end systems is rarely practical.

## *A Basis in Industry Standards*

You may have wanted to use your process model in another cool tool but known that it wasn't possible. Or perhaps you've wanted to do a mashup with another service, only to find that the other service isn't REST-full. Maybe you've been annoyed that your event-based third-party system didn't integrate with your ESB? In short, you may have been frustrated by an application's lack of industry standards compliance.



Enterprises have had their own standards for application integration for decades, of course. For SOA, however, an enterprise standard isn't sufficient: Integration across an engaging enterprise must be based on industry standards.

The need for industry standards is clear. As you lose control of parts of the end-to-end environment, you can't demand that everyone else follow some proprietary enterprise standard. You can't be the global top dog — and neither can anyone else. Varying relationships and interactions have varying structures of influence.

But what capabilities need standards the most? Here are a few:

- ✔ **Interaction styles:** Different interaction styles need to be supported. Some people have associated SOA very closely with the WS\* standards, but not all services are web services. In fact, an engaging enterprise must support all message-based, interface-based, resource-based (such as REST), and event-based interactions.

- ✓ **Edge capabilities:** Firm standards are needed for policy, accounting, and all the other additional capabilities required at the edge (see “Integrating at the edge,” earlier in this chapter).
- ✓ **Information collection and processing:** Standards are needed to specify how to collect and process information about business interactions in search of business insight.
- ✓ **Channel-specific standards:** Finally, channel-specific standards are needed. Just think of mobile devices and the Internet where many different providers and manufacturers need to participate and interact on the same network.

## *Leveraging and Extending Open-Source Technologies*

Most chief information officers will tell you that open-source technologies are part of the strategic evolution of tools and middleware. Although it's not always on par with vendor-provided capabilities, open-source technology is often good enough for the most standardized aspects of the IT infrastructure. Furthermore, in the context of an engaging enterprise, apps may be created by third-party stakeholders (such as customers and partners) who use open-source technology to create those apps.

Open-source technology is here to stay. You can't contain it or constrain it to a separate, disconnected environment; instead, you need to embrace and extend it. You can't prevent new age developers (you know, those folks who are self taught to some degree, grew up with the Internet, live through their networks, and such) from downloading any open-source library or tool that they like or that their peers recommend; instead, you provide infrastructure that makes it easy to deploy and manage their solutions. Set standards on how to sense and respond to what happens in the operating environment. Enforce the need for consolidated insight with the support of the business.



Creating solutions is easy, but managing what you've created is hard. Understanding this fact and actively seeking to engage developer communities will make your life easier.

## *Providing the Platform for a Growing Ecosystem*

Have you ever wondered who your talented new developer takes advice from, because it sure isn't you, the "graybeard"? Or conversely been a talented new developer and thinking that your organization has no clue as it is preventing you from interacting with your peers and keeping up with the competition?

As I mention in the preceding section, peer communities are very important to new age developers as sources of inspiration and advice. Consequently, it's important for an engaging enterprise to understand and interact with those communities. You need to know, for example,

- ✔ What the developer community is saying about your enterprise
- ✔ How to entice developers to use your services when they build solutions
- ✔ How to make it easy for them to interact with you, making you the preferred partner and supplier, instead of your competitors

The notion of APIs and API management — the idea that external business interfaces can be codified and published — is critical for enabling an extended and growing ecosystem. APIs are business services that provide a managed interface for interaction through the corporate boundary, so you should apply the full power of SOA to creating and managing business APIs, including the notion of API registries (often called API catalogs) for publication and advertising of the externalized enterprise.



## Chapter 3

# Going Mobile with SOA

---

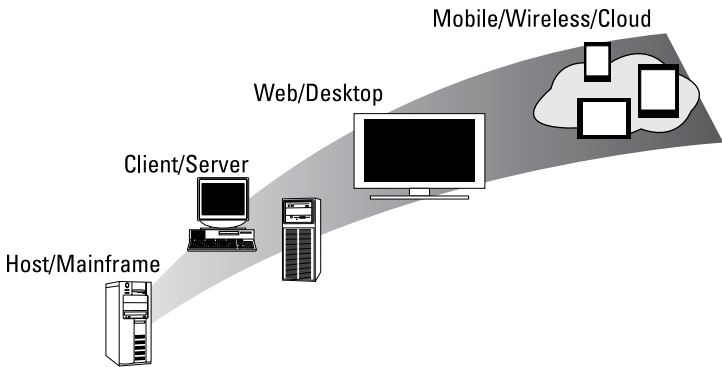
### *In This Chapter*

- ▶ Seeing why mobile communication is important
  - ▶ Getting down to mobile SOA fundamentals
- 

**T**he young woman checking her bank balance while on the train to the city, the savvy shopper using a price comparison service while in the supermarket, and the child purchasing movie-maker software for her iPad — these scenes and many more have become everyday occurrences in modern society. In this chapter, I show you why going mobile is important and how you can use service-oriented architecture (SOA) design principles to improve your mobile solutions.

## *Why Mobile Matters*

From a technology perspective, as illustrated in Figure 3-1, mobile solutions are a natural evolution of enterprise computing, with each successive generation of solutions being more loosely coupled to the enterprise systems of record. But mobile technology affects more than just IT departments. Terms such as *mashup* and *app* have become part of everyday conversation. The volume of on-the-move information retrieval and action is growing so fast that in ten years it may overtake the volume of more traditional electronic transactions. In fact, mobile changes the business model, not just the channel.



**Figure 3-1:** Mobile computing is a significant evolution of enterprise computing.

Mobile interactions occur in different contexts from classical transactions:

- ✔ You take a mobile device with you everywhere and have it at hand wherever you are.
- ✔ You can access any information or contact any friend on the fly.
- ✔ Mobile interaction can be asynchronous. Messages can be pushed, not just pulled.
- ✔ The growth of social media has created consumers who expect their opinions to matter and who choose mobile devices as their way of interacting with the world. (I discuss social business in Chapter 4.)

Clearly, these characteristics change the ways in which interactions need to be orchestrated, managed, and monitored. If you're not present here and now, you're not present at all. Mobile consumers have little patience and no tolerance for not being able to do what they want, where they want it, and when they want it.

# The Fundamentals of Mobile SOA

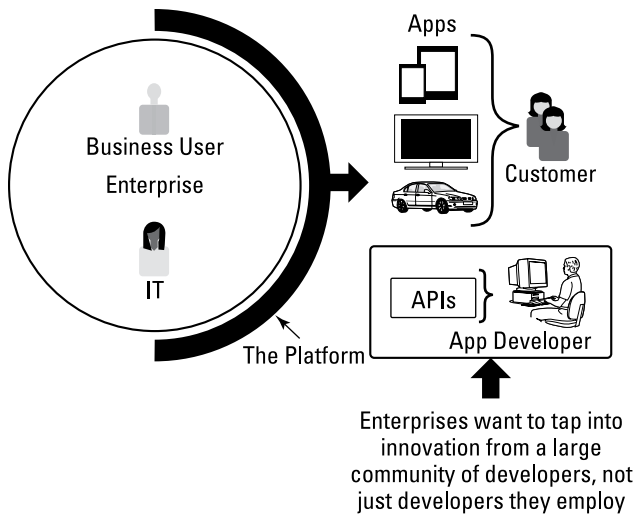
In this section, I look at three of the fundamental tenets of mobile — APIs, apps, and app stores — through the lens of SOA design principles.

## APIs

*API* (see Chapter 2) is a recycled term that's currently being used to represent a public persona for your business — a persona that can be consumed by customers, Independent Software Vendors (ISVs), and business partners. Typically, a successful public persona is

- ✔ Simple in scope (such as a small number of unique APIs)
- ✔ Pervasive throughout multiple architectures (such as supports multiple protocols and programming models for service and data access)
- ✔ Presented as a simple data model (such as JavaScript Object Notation, JSON)
- ✔ Provides controls in the form of policies (such as a quota, which limits the number of calls to this API from a given user during a specific period)

APIs are the externalized aspects of services. APIs shouldn't be viewed as alternatives to SOA, but as parts of a well-architected service-oriented enterprise. As Figure 3-2 shows, APIs are simply the way that an engaging enterprise chooses to expose business services to a broader audience in a controlled fashion.



**Figure 3-2:** Business APIs are the external persona of an engaging enterprise.

APIs, however, are a specific genre of services with a life cycle that focuses on external consumption. This aspect is more than just a nuance: It drives focus on simplicity, security, and compatibility with standards-based external systems.

### *Managing business APIs*

Managing business APIs is key to extending an enterprise's reach to the new channels presented by mobile and Software As a Service (SaaS) solutions. In essence, this transformation replicates that of e-business in the late 1990s. E-business placed web platforms in front of mainframe-based applications as a way to extend the mainframe applications to the web without destabilizing core business transactions. Now enterprises are expanding beyond the walls of the enterprise by using APIs to responsibly externalize internal services.

### *Connecting business APIs via edge gateways*

Connecting business APIs to systems of record involves multiple challenges, including security, authentication, and scalability:

- ✔ Business APIs connect elastic clients to not-so-elastic systems of record, leading to the need for traffic management and throttling.
- ✔ Security models have to cope with different degrees of authority for private APIs, protected partner APIs, and open public APIs.
- ✔ Traffic must be monitored for accounting and service-level agreement purposes.

These concerns aren't new per se, but they take on a whole new level of urgency in the API business. As organizations mobilize more lines of business, controlling multiple applications in the wild and managing their ongoing performance can bring any IT department to its knees. That's why a good edge-gateway strategy is needed. When a good edge gateway is in place, from an integration perspective, that gateway becomes just another channel to your ESB (see Chapter 1). That channel is self-regulating and relieves the ESB of security, caching, and traffic management concerns for interactions across the boundary between systems of engagement and systems of record.



You also need to think about adding new bindings (think REST) and applying change management consistently across all kinds of services and APIs, because many APIs are simply external renderings of an internal service.

## Apps

APIs enable apps. In this context, an app is a component of a distributed application that can reside outside the boundaries of your enterprise. A common example is a mobile phone app that resides in an external app store (see “App stores,” later in this chapter).

Apps can be much more, however. They can run on any device (such as a car or a set-top box), transmit data gathered from a sensor network (such as a smart electric meter or a pacemaker), or represent an application running in an external SaaS provider environment. Ultimately, an app is any piece of external code, developed by anyone from corporate developers to mobile customers, that interacts with your published APIs.

### *Enabling apps*

When you enable apps, you enable innovation. Although you're the sole creator of your own public persona (refer to "APIs," earlier in this chapter), you want to encourage the world at large to extend this persona by building apps and making it possible for you to reach markets that you wouldn't be able to reach on your own. Thus, composite apps are key elements of extending enterprise services to external stakeholders.

### *Creating apps*

Although published business APIs need to be relatively stable, apps vary much more rapidly and significantly. The rapid life cycle of apps is exacerbated by the fact that apps typically are fine-grained, built to purpose, and updated frequently for multiple target environments (such as devices and SaaS providers).

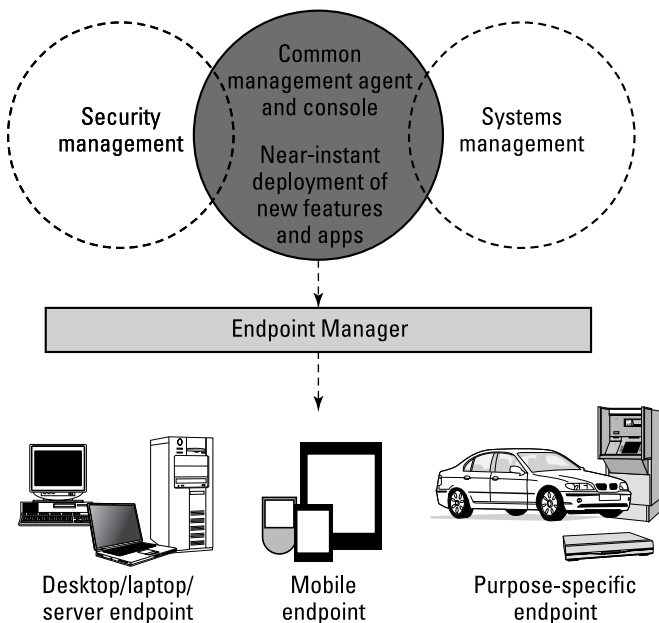
To foster the proper creation of apps, you need a first-class app software development kit that enables you to create a variety of external applications (especially mobile) for a variety of device platforms. You also need an endpoint management solution that enables you to extend automation and deployment to the mobile environment (see Figure 3-3).

## *App stores*

Like APIs, apps need to be published, either to your own app store or to someone else's. Decisions about how to handle app stores usually are an important part of an enterprise's business model for mobile platforms. You have several options:

- ✔ **Provide your apps for free.** Choose this option when the apps provide value to the consumer but are used to simply promote the enterprise, strengthen brand and customer relationships, and so on.

- ✔ **Charge for the apps but use broker app stores to deploy them and handle the accounting.** Choose this option when you want to monetize the apps but not have the hassle of running an app store (maybe you don't have the community reach either).
- ✔ **Provide your own app store.** Here, you remain in full control and can create additional value through community creation (control point). This strategy is tough to execute without community recognition.
- ✔ **Serve as the broker for everybody else and make the app store part of your enterprise's core revenue stream.** Choose this option when running app stores is part of your business, not just something you do to enable your business.



**Figure 3-3:** Extending visibility, control, and automation to mobile devices.

## Publishing apps and APIs

Having great APIs and apps won't help you if no one knows about them. Responsibly publishing your enterprise apps and APIs to key places outside your enterprise makes it possible for these assets to be discovered and used properly.

Whether you publish to your own API portal or someone else's, the publishing

process involves an element of marketing. You need to be able to alert, track, rate, and monitor assets within communities of interest. After you update an API, for example, you can go on Twitter and post a tweet with a hashtag like #ACME-API to inform developers that a new version of the Acme API or app is available.



## Chapter 4

# Socializing with SOA

---

### *In This Chapter*

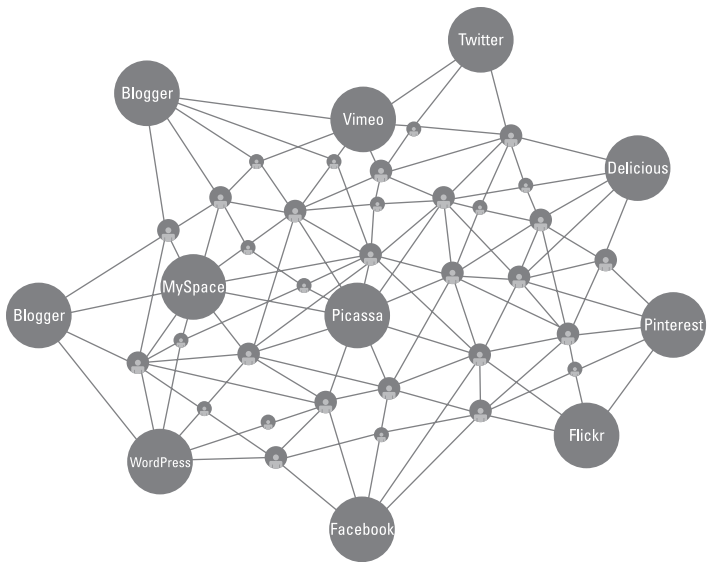
- ▶ Networking with social media
  - ▶ Making connections with people and knowledge
- 

**T**witter, Facebook, and LinkedIn are all part of the social fabric that connects individuals and professionals — part of the way people and enterprises engage in the Internet age. In this chapter, I illustrate how an engaging enterprise can socialize with SOA.

## *Interacting via Social Media*

If you've been reading chapter by chapter up to this point, you may ask, "Didn't you just talk about the community aspects of SOA, and isn't that it as far as social business and SOA go?" Not quite. In fact, not at all.

The reality is that systems of people collaborate, communicate, and use social data, analytics, and networks to get the daily job done. Integration of collaboration, communication, and social analytics makes human roles more instrumented, orchestrated, and even governed. The social media revolution is changing the way people interact and creates new relationships that leverage what I call the *social graph* (see Figure 4-1).



**Figure 4-1:** Interacting through the social graph.

The social graph represents the way that people connect and interact through various social media not only through direct interactions but also often through any number of intermediaries. For SOA to support such a social network, the set of endpoints in the SOA fabric must expand vastly to include people and knowledge as addressable resources. People interacting in a social context aren't interacting through messages; they're interacting through knowledge, questions, and answers. Mediation is no longer about protocols and formats, but about whom to contact and where to find the best information that answers a particular question.

Becoming a social partner is an important ingredient in strengthening customer relationships and broadening community reach. Leveraging people as addressable endpoints — making them part of the SOA fabric — enables an engaging enterprise to route human interactions along the most optimal channel, provide the most optimal information in context, and let customers interact not only with IT systems but also directly with employees and partners. But doing so requires a significant expansion of the notion of mediation as part of an ESB.

## Connecting People and Knowledge

The boundaries between human and information technology (IT) systems will continue to blur. You may soon be interacting in a connected knowledge fabric that permeates every aspect of society. Most of that interaction will be through social media, and business designs will include people and systems as endpoints connected to the same knowledge bus. A *knowledge bus* is a natural evolution of the ESB (see Chapter 1). Rather than mediating messages and functionality, which is the key capability of an ESB, the knowledge bus mediates knowledge.

Social business is about people, and people are looking for answers to questions — not just responses to requests. They're looking for answers that they can understand, so human semantics in interaction is a necessity. People are also looking for mediation across time, location, and community simply as part of their social outreach. When I look at the way my 11-year-old daughter is already communicating with her friends, blending audio, video, text, and even gameplay, it seems obvious to me that before long, people and knowledge will be connected in a social fabric built on the design principles of SOA.



Through SOA, you can describe and connect people, their available knowledge, and the way they can interact whether synchronously or asynchronously. Someone who can provide an answer is, from an SOA design perspective, providing a valuable business service. This doesn't mean that people will be treated as robots; instead SOA-based interactions need to become more humanlike, including the notion that humans are rarely 100 percent certain of the right answer to a given question, nor do they require such absolute certainty in order to have received real value.

What was so cool about watching IBM's Watson supercomputer compete in the quiz show *Jeopardy!* in 2011, for example? To me, it was the notion that Watson didn't just provide a reply to a request; it also showed how certain it was of its answers by assessing the clues that it had found as evidence.

This made the interaction with Watson much more human-like. The notion of probabilistic computing is a key enabler of human-centric collaboration (so are rich context and semantics for the information flowing through the social fabric). This applies whether the interaction is human to human or human to IT system.



From a SOA design principle perspective, to be practically possible, a social network needs

- ✓ Standards for channels and knowledge
- ✓ Service oriented thinking to codify what business services people offer and require
- ✓ To incorporate ecosystems inside and outside the enterprise
- ✓ To maintain integrity in all interactions or people will quickly lose faith in a particular source of information or service to the detriment of future business interactions

## Chapter 5

---

# Putting Your Head in the Cloud with SOA

---

### *In This Chapter*

- ▶ Seeing how the cloud affects the business model
  - ▶ Adopting cloud computing
- 

**M**uch has already been written about the relationship between cloud and SOA. Cloud environments provide services defined with SOA design principles and are themselves built on a SOA foundation.

Service orientation allows the standardization of the offerings delivered via a cloud computing model. In fact, those offerings are simply services with well-defined functionality and interfaces. The virtualization of dynamic infrastructure enables those cloud services to be delivered in an elastic and flexible fashion — virtualization that itself is based on treating IT infrastructure capabilities as services that can be provisioned and deprovisioned as needed. Thus, SOA and dynamic infrastructure together enable cloud computing. Those are the basic IT aspects of the cloud.

In this chapter, I cover the business impact of cloud-based solutions and the way that SOA design principles apply to them.

## *Taking Cloud to the Business*

A cloud computing model has a profound effect on the IT business model. Specifically, IT spending changes from a capital-cost model to an operating-cost model, which forces external and internal cloud providers to think in terms of cash flow, profit, and everything else required to run a business—in this case, the business of IT.

Furthermore, from a business organization perspective, the business must decide matters such as who's allowed to use a self-service interface that results in immediate provisioning and chargeback of additional IT resources. Economic impact that used to take weeks or months now happens in seconds, and there's no "regret" button after the order for more resources has been executed.

What's in play across business and IT likely requires changes in existing integrated service management processes such as provisioning, discovery, security, and monitoring. Planning and executing that transition is important to success in the cloud.

## *Transitioning to Cloud Computing*

The transition to cloud computing gives rise to several practical questions:

- ✔ How do you deploy solutions into the cloud by using a continuous delivery model?
- ✔ How do you provision test environments needed for hybrid solutions that are partly on premise and partly in the cloud?
- ✔ Who is responsible for security in a cloud environment, and who pays the cost if security is breached?
- ✔ How do you monitor whether your service-level agreements are being met?
- ✔ How do you discover services across on-premises, cloud, and mobile environments?

You can answer all these questions, and more like them, in a relatively straightforward fashion simply by applying the good design principles of SOA, as follows:

- ✔ **Service orientation at the core** leads to well-defined consumable services that can have policies applied to them and whose consumption can be monitored and managed.
- ✔ **Process integrity at Internet scale** leads to well-defined requirements for elasticity and transactionality across an end-to-end environment.
- ✔ **Integration with enterprise capabilities and back-end systems** enables the creation of hybrid solutions that mix cloud and on-premises capabilities.
- ✔ **A basis in industry standards** enabled cloud computing in the first place; it also plays an important role in the creation and integrated service management of hybrid solutions.
- ✔ **Leveraging and extending open-source technologies** comes into play because many cloud environments are based on an open-source core. Educational and government institutions in particular can be expected to continue to pursue open-source technologies as core elements of their runtime infrastructures.
- ✔ **Providing the platform for a growing ecosystem** applies to cloud computing because the cloud itself is a prime example of an ecosystem-centric environment that enables goods and services to be exchanged way outside the traditional boundaries of the enterprise.



For more details on these and other SOA design principles, turn back to Chapter 2.





## Chapter 6

# Tapping Big Data Insight with SOA

---

### *In This Chapter*

- ▶ Leveraging existing data streams
  - ▶ Using disparate data
  - ▶ Enabling insight
- 

**M**aybe you think that I've lost it — there's no direct correlation between Big Data and SOA. Don't be so sure of that. I claim that Big Data has everything to do with SOA design principles. Three questions highlight the close relationship between Big Data and SOA. These questions are

- ✓ How do you tap existing data streams?
- ✓ How do you access and manage data from a disparate set of information sources?
- ✓ How do you provide business insight for higher-level business solutions?

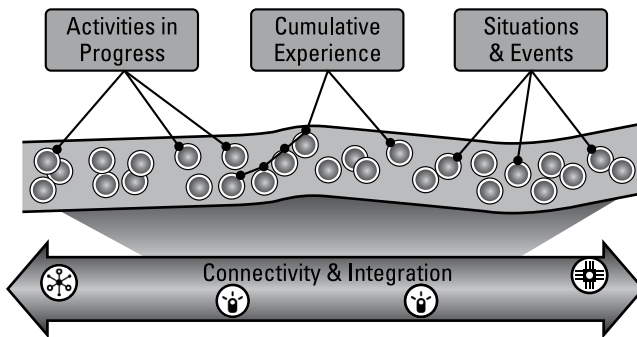
In this chapter, I answer those questions.

## *Tapping into Existing Data Streams*

Consider the question of how you tap into the data already flowing through the business — in particular, business interactions of interest to an engaging enterprise. How do you collect and process insight by instrumenting applications to

provide a “stream of consciousness” in a nonintrusive fashion when, after all, many of those applications are developed outside your control?

The answer is simply that you instrument the ESB middleware that carries all interactions between systems of engagement and systems of record, siphoning off information from any interaction that has an endpoint inside the enterprise. Figure 6-1 illustrates the process.



**Figure 6-1:** Instrumenting the ESB middleware.

Although this solution doesn’t cover all relevant information, such as everything happening in a social community (see Chapter 4), it does address most of the visibility challenge. Following this approach, though, results in a Big Data challenge.

## *Putting data to work*

A great deal of data comes off the instrumentation of the ESB, and you’ll likely want to use that data in many ways, such as the following:

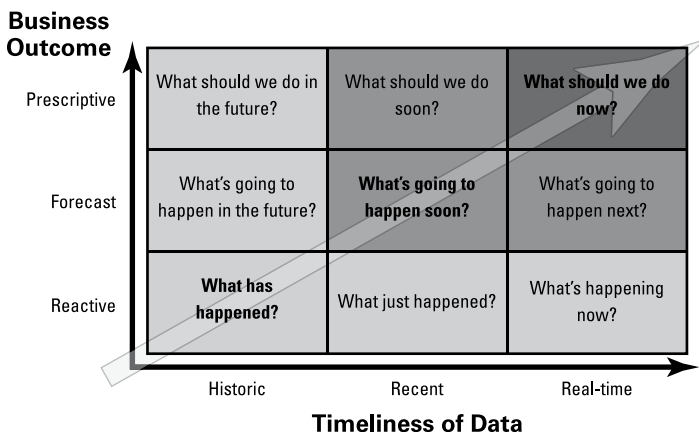
- ✓ **Knowing your client better:** Know the client’s location, reach the client now, and use authoritative information as part of any interaction. Customers expect you to know who and where they are. (They don’t want to repeat who they are and why they’re calling.)
- ✓ **Determining the next best action:** Determine the right thing to do here and now by leveraging strategic insight

from information collected over time. Customers expect you to be smart about what you suggest as the next interaction; if you're not, they won't give you the time of day. (They won't like it if you propose a solution that they rejected last week.)

- ✔ **Adapting to situation and context:** Make deployed solutions adaptive, and instrument processes to react to the current situation and location. Customers have no patience with processes that clearly weren't created with their circumstances in mind. (They don't want to be on the phone with someone who's following a script that includes five steps they've already completed.)
- ✔ **Improving compliance:** Check qualifications and policies on the fly, handle exceptions in a structured fashion, and apply risk management to the information collected. Authorities, internal as well as external, expect you to comply with policies and regulations at all times and be able to prove it. (Enterprises don't want to be in a situation in which they can't provide documented evidence that they're in compliance with relevant legislation.)

## Predicting the future

A common thread through all the data uses is the desire to become progressively more predictive (see Figure 6-2), which in turn requires giving up the notion of absolute truth.



**Figure 6-2:** Progressively becoming more predictive.

## Big Data and the four Vs

An engaging enterprise also has to deal with the four Vs of Big Data:

- ✔ **Volume:** The amount of information to process is growing exponentially due to the Internet of things (such as sensors, mobile devices, appliances, and cars).
- ✔ **Velocity:** Much information of interest to an enterprise is never persisted in a data store. Storing this information would be practically impossible anyway, so it has to be processed on the fly. Consider RFID sensors that tell you when a very important customer has walked into the store, or consider data from the Big Hadron Collider, which is searching for the most elusive atomic particles within a millionth of a second.
- ✔ **Variety:** Information of interest rarely conforms to a centrally defined model. As much as 80 percent of the world's data — e-mails, social interactions, historical journals, and so on — is unstructured, with no predefined schema.
- ✔ **Veracity:** Not all information can be trusted equally. Trustworthiness must be quantified by source and type of information (such as opinions posted on a blog, tweets, or information from partners who have their own agendas).



An engaging enterprise has to deal in probability and risk as part of doing business.

## Using Data from Disparate Sources

How do you access and control disparate information sources? You apply the good design principles of SOA, standardizing access to in-flight information, describing such access in terms of well-defined services, integrating with information in the systems of record, and maintaining information integrity across the fabric of the engaging enterprise.



By exposing both information streams and analytical capabilities as well-defined services, you create desirable loose couplings among the information source, the analytical engine, and the business application that needs the business

insight. Look for technology capabilities that mix Big Data and SOA, instead of always relying on traditional data warehouse approaches.

Chapter 4 describes the need for an engaging enterprise to connect and mediate people and devices. That same enterprise in an information context needs to move from focusing on transactional payload to focusing on deriving and leveraging business insight. Without understanding people, their motives, and their needs, it's hard to optimize business interactions that involve them.

## *Providing Business Insight*



How do you compose different kinds of insight for strategic advantage? SOA doesn't create insight; but it helps you apply insight in the right way, at the right time, and in the right place. SOA also helps you collect evidence on what's going on in the operational environment and can help capture business outcomes.

In addition to what SOA can supply, most business analytics use cases need to be able to collate and reconcile information — a capability provided by Master Data Management solutions. Beyond Master Data Management, the most advanced business insight solutions may need specialist information providers; analysis is only as good as the information that it's based on, after all. SOA isn't a substitute for information management capabilities, but the blend of SOA and information management is what unlocks the full potential of Big Data.

# Notes



# Notes

A series of 20 horizontal lines for writing notes, starting below a dotted line.





## Use service-oriented thinking for a more engaging enterprise

Believe in the power of the good design principles of SOA! The need exists to enable new solutions on a stable, well-tried design base. This book guides you in applying the design principles of SOA to the business problems of today. SOA is simply good design for the business solutions of a connected and distributed world.

- *See where SOA is going — development and control aren't just for IT departments anymore*
- *Leverage and extend open-source technologies — embrace them and extend them*
- *Learn the role of mobile — mobile changes the business model*
- *Interact via social media — add human intelligence to the business network*



Open the book and find:

- The past uses and new functions of SOA
- The elements of good SOA design
- How to go mobile with SOA
- Insight into applying SOA in a social business context
- How cloud computing is built on SOA design principles

**Making Everything Easier!™**

Go to **Dummies.com**<sup>®</sup>  
for videos, step-by-step examples,  
how-to articles, or to shop!