

---

**The Evolution of UDDI**  
**UDDI.org White Paper**

---



The Stencil Group, Inc.  
41 Freelon Street  
San Francisco, California 94107

*[www.stencilgroup.com](http://www.stencilgroup.com)*

July 19, 2002

---

## Table of Contents

---

INTRODUCTION: THE EVOLUTION OF UDDI .....	3
Introduction.....	3
A Brief Background on UDDI's History.....	3
The Universal Business Registry and Private Implementations.....	4
UDDI VERSION 3: A FOCUS ON REGISTRY INTERACTION .....	6
What's New in this Version of UDDI?.....	6
A Closer Look at Registry Interaction.....	6
Key Functional Concepts.....	7
WEB SERVICES SCENARIOS WITH REGISTRY INTERACTION.....	10
Scenario 1: Private Test Registry.....	10
Scenario 2: Supporting Collaboration among Trading Partners.....	12
CONCLUSION .....	14
REFERENCES AND RESOURCE LIST.....	15

---

## Introduction: The Evolution of UDDI

---

### Introduction

Almost two years ago, the Universal Description, Discovery, and Integration (UDDI) project began amid a flurry of sweeping industry initiatives. Champions of the new technology economy envisioned a day in which automated discovery and execution of e-commerce transactions would result in an exceedingly liquid and frictionless environment for business. Speculative businesses like high-flying net markets promised to change the fundamental underpinnings of traditional supply chains.

In today's more sober business environment, however, it is clear that pragmatic and incremental technologies—like those that comprise the web services concept—quietly have succeeded in ways that the dot-com boom's self-consciously disruptive models of doing business could not. The technology industry's understanding of the role of UDDI, too, has evolved to reflect these pragmatic imperatives.

Indeed, far from the naïve assumption that UDDI represents the radical reengineering of business processes, the consortium of software companies and their customers that is developing the specification has focused on the very real challenges of interoperability and interaction facing IT organizations as they begin to incorporate web services concepts into their software systems. This evolution of UDDI from “e-business directory” to “web services infrastructure” directly reflects the lessons and requirements of today's IT mandates.

Today's business operating environment requires that IT organizations identify and plan for an architecture that can not only provide for scalability, but also “agility”—the ability to add new offerings or reorient existing functions in a highly flexible manner. This agility requires that we think of IT in terms of services, not physical assets.

This concept is not necessarily a new one. Indeed, progressive enterprise software architects have long advocated a service-oriented architectures (SOA) in which applications are designed with modular, loosely coupled interfaces that hide the complexity of the underlying systems. For a variety of reasons, however, such an approach was not a practical approach for solving broad-based enterprise software needs until now.

### A Brief Background on UDDI's History

UDDI is one important enabling element of this emerging infrastructure. UDDI itself represents both a specification of a proposed standard and a consortium of backers. The sponsoring organization, UDDI.org, is comprised of more than 200 major software developers and e-business leaders who hope to catalyze the

development of UDDI and related technologies.

As shown in the table below, UDDI.org recently has released the third version of its eponymous specification. With this release, the consortium's stated goal is to turn over the UDDI project to an independent standards organization in the near future. UDDI.org currently is negotiating with the W3C, OASIS, and others to shepherd the standard forward.

**Figure 1: History of UDDI Specification**

VERSION	DATE	KEY OBJECTIVE
1.0	September 2000	Create foundation for registry of Internet-based business services
2.0	June 2001	Align specification with emerging web services architectures and provide more flexible taxonomy
3.0	July 2002	Support wide interaction of private and public implementations

This paper takes a closer look at two aspects of this evolution of UDDI:

- The key architectural changes in the recent Version 3 specification, and
- The business role for UDDI in the context of the business applications of today's rapidly emerging web services concepts.

First, however, we briefly call out an important, conceptual distinction between two facets of what UDDI represents.

### **The Universal Business Registry and Private Implementations**

When UDDI was first launched, much of the initial attention was focused on the concept of a "Universal Business Registry" (UBR) that represented a master directory of publicly available e-commerce services. The commonly used metaphor for the UBR is that of a telephone directory, because, the information provided in a listing consists of three conceptual components: "white pages" of company contact information; "yellow pages" that categorize businesses by standard taxonomies; and "green pages" that document the technical information about services that are exposed.

Four companies (IBM, Microsoft, NTT Com, and SAP) currently operate the UBR, a public instance UDDI, much as organizations like VeriSign's Network Solutions

unit operate root nodes in the domain name system (DNS) database. As the most visible element of the UDDI infrastructure, critics often single out the relatively sparse population of services listed in these UBR databases as evidence that the standard (and web services in general) are missing in action.

However, this criticism is specious, for it overlooks the reality that most of today's web service applications are not intended for public use, but rather inside organizations or among existing, trusted business partners. It is for this reason that the focus of the UDDI specification has evolved to support a variety of implementations of the standard, including public registries such as the UBR and private registries that may be implemented in software products like application servers or other stand-alone products that can be deployed within a company's own network boundaries.

**Figure 2: Several "Flavors" of UDDI Registries**

REGISTRY TYPE	DESCRIPTION	WEB ANALOGY	EXAMPLE APPLICATION
Public	From an end-user's perspective, a public registry appears to be a service in the cloud. Although administrative functions may be secured, access to the registry data itself is essentially open and public. Data may be shared or transferred among other registries.	Web Site	Universal Business Registry (UBR)
Private	An internal registry, behind a firewall, that is isolated from the public network. Access to both administrative features and registry data is secured. Data is not shared with other registries.	Intranet	Internal Test Environment
Shared/ Semi-Private	A registry deployed within a controlled environment, but with controlled access to the outside world and shared with trusted outside partners. Administrative features may be delegated to trusted parties. Data may be shared with other registries in a controlled way.	Extranet	Trading Partner Network

Indeed, as we discuss in the remainder of this paper, it is in supporting the interaction among a variety of private implementations, as well as the public UBR nodes, that the UDDI specification takes a pragmatic turn.

---

## UDDI Version 3: A Focus on Registry Interaction

---

### What's New in this Version of UDDI?

Although many aspects throughout the UDDI specification have matured in the version 3.0 release, the chief architectural change is the concept of “registry interaction.” This shift reflects the increasing recognition that UDDI is one element of a larger set of web services technologies that support the design and operations of myriad software applications within and among business organizations. In short, just as each enterprise application embodies the specific characteristics of the business process it supports, so should the enabling technologies like UDDI support a variety of infrastructural permutations.

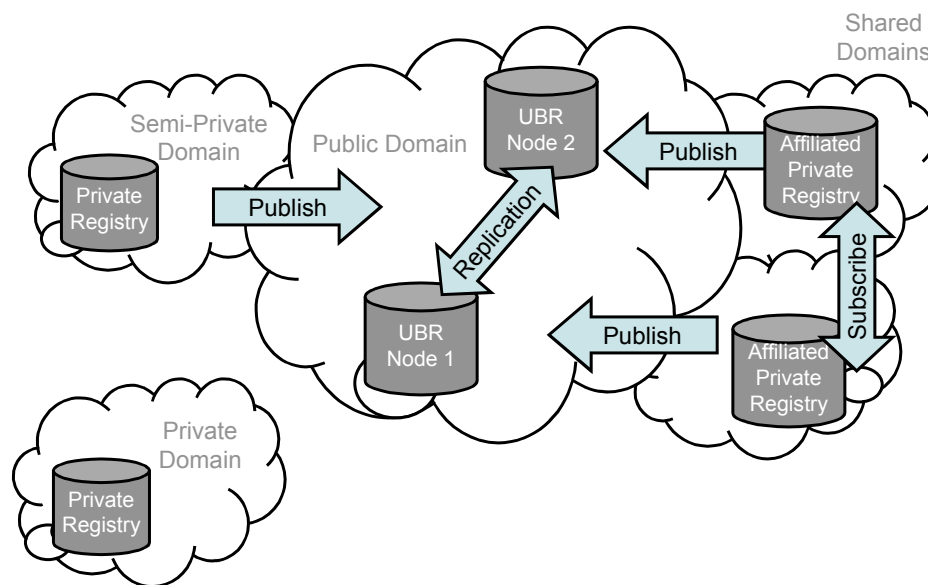
For UDDI, this business requirement dictated an increased emphasis on providing a means to define the relationships among a variety of UDDI registries, not simply access to one, public registry of business services, the UBR. Although the UDDI specification included from the start concepts like replication and distribution among server peers, earlier definitions of the standard did not fully address the nuts-and-bolts required for the more sophisticated, hierarchical model now dictated.

### A Closer Look at Registry Interaction

While the specification enables a technical interoperability of registries, it does not dictate the nature of or policies for such interaction. Rather, it leaves those issues to be decided upon by the registry operators. Obviously, the establishment of these policies, as well as a key management infrastructure, will become a critical element to successful distribution of registry responsibilities on not just a technical level, but also on a business process plane.

Elsewhere in this report, we suggest two specific business scenarios that UDDI and other web services infrastructure are well suited to help address, but it is worth examining what is meant by the concept of “registry interaction” in the Version 3 UDDI specification. Simply put, registry interaction refers to using UDDI to support a variety of network/infrastructure topologies. The possibilities have expanded from a stand-alone, single-registry approach to include hierarchical, peer-based, delegated, and others. In short, the structure of a UDDI registry (or registries) can now reflect the realities and relationships of the underlying business processes that it supports.

**Figure 3: Conceptual Illustration of Registry Interaction**



Comment: This diagram illustrates several models of registry interaction enabled by Version 3 of the UDDI specification. Through mechanisms like publish/subscribe and replication among peer nodes of a registry, the information in UDDI servers can be fully public (like the UBR), semi-private (such as the affiliated registries shown here), or even fully private and isolated from the public network (as depicted in the “Private Domain” above).

Managing multiple versions of registry entries presents a challenge to any implementation of the registry, but it is a critical element of managing this sort of distributed infrastructure. The specification itself provides guidance to help facilitate the maintenance and mapping of keys and records across registries, but the technical objectives of the recent UDDI specification are intended to do just that—facilitate, but not define, a wide range of business scenarios. It will be the registry operators, users, and software developers who design and implement a wide range of business policies and constructs on top of the basic UDDI infrastructure.

### Key Functional Concepts

The Version 3 specification addresses several features that support this emphasis on registry interaction. While relatively little of the existing features have changed, a handful of key functional concepts have been added or expanded to accommodate the variety of new taxonomies. Some of the most important issues addressed in the Version 3 specification include:

- Registration key generation and management

- Registration subscription API set
- XML digital signatures

Each of these major categories of functionality is discussed in turn, below.

### ***Creating and Managing Registration Keys***

Although fairly esoteric to a lay reader, the mechanics of defining, creating, and managing the registration “key” is an important concept in any sort of data management system. It becomes critical in a distributed system—like UDDI—that involves multiple entities creating records that may be replicated or versioned across multiple versions of the database.

The UDDI specification defines syntax for the development of UDDI keys that will preserve the integrity of each registry. Essentially, the intent is to provide a structure for creating unique keys for publication in multiple UDDI registries.

#### **Figure 4: Example UDDI Registry Keys**

Example 1 (domain-based key):  
`uddi:somecompany.com:serviceName`

Example 2 (globally unique identifier):  
`uddi:4CD7E4BC-648B-426D-9936-443EAA8AE23`

With the new flexibility granted to generate and to use and to replicate keys in private and public namespaces, information about services can be published using a single key to any number of registries. This is a significant change; previously, key creation and service publication was managed only at an individual registry level.

However, the responsibility to implement key management functions remains incumbent on either:

- The publisher to maintain the uniqueness of keys they publish
- The operator of the UDDI registry to develop a UDDI compliant key creation process on behalf of the all publishers to maintain uniqueness

Vendors implementing the UDDI specification into their products likely will provide proprietary tools to manage the creation and maintenance of keys as part of their overall web services management solutions.

### ***Publishing and Subscribing to Services***

Using terminology commonly used in scenarios that define the interactions among systems that create and consume information in a structured way, the UDDI specification describes roles for “publishing” and “subscribing” to services. The former refers to systems that register services in the directory, while the



latter refers to systems that need to be notified of changes to a particular record.

### Publishing Services

The Publication API defined in the specification allows a user (or system) to publish information to a UDDI compatible registry and, in the process, generate and assign a key. Several important new facets of the publication process reflect new registry interaction concepts. These include:

- Generation and assignment of registry keys
- Rules and namespaces for managing unique and non-unique record keys
- Defining the roles of root and affiliate registries
- Updating or deleting an existing entity

### Subscribing to Services

The Subscription API defined in the specification allows a user (or system) to monitor the creation, deletion, and changes made to services in a registry. Several new features defined in this version of the specification help support the “peering” or sharing of records among registries. These include:

- Notification of newly registered businesses or services
- Changes to existing businesses or services
- Obtaining registry data for use in a private UDDI registry
- Obtaining registry data for use by a registrar

### ***Ensuring Data Integrity***

In opening up registries more widely to publishers, subscribers, and peers, the question of authorizing access to records—be it for creating, modifying, or simply reading records—becomes an important one. Like many questions of business policy, the UDDI specification leaves details of implementation are left to registry operators.

However, the specification does provide a means for implementing features that help validate the integrity of data in the registry through the use of XML Digital Signatures (DSIG). Most elements of a registry record optionally may be signed using the DSIG specification maintained by the W3C. Thus, while the specification does not define specific policies around security and authorization, it does provide the means for specific implementations to provide for these needs.

The primary benefit of digital signatures is to ensure that:

- Data has not been altered since it was signed and published
- Ownership of a particular registry entity can be validated
- Confidence that data transferred among registries can be assured

---

## Web Services Scenarios with Registry Interaction

---

Many practical applications of general web services and specific UDDI concepts exist, and it is not our objective to document them exhaustively here. Instead, we outline two scenarios that are representative of the sort of applications that would benefit—or even depend upon—the registry interaction features enabled in Version 3 of the UDDI specification.

### Scenario 1: Private Test Registry

#### *Business Scenario*

For the past year, the IT organization of a major corporation had begun to explore the possibilities of web services approaches to application development and integration. Using the technology first in pilot projects and other piecemeal efforts, the IT team had skirted around the question of how to deploy and manage its web services applications. As it begins to plan for using web services in the company's mission-critical business processes and to create services that will be available to the rest of the organization, the IT team realizes that it will require a more controlled and systematic approach.

#### *Overview of Issues*

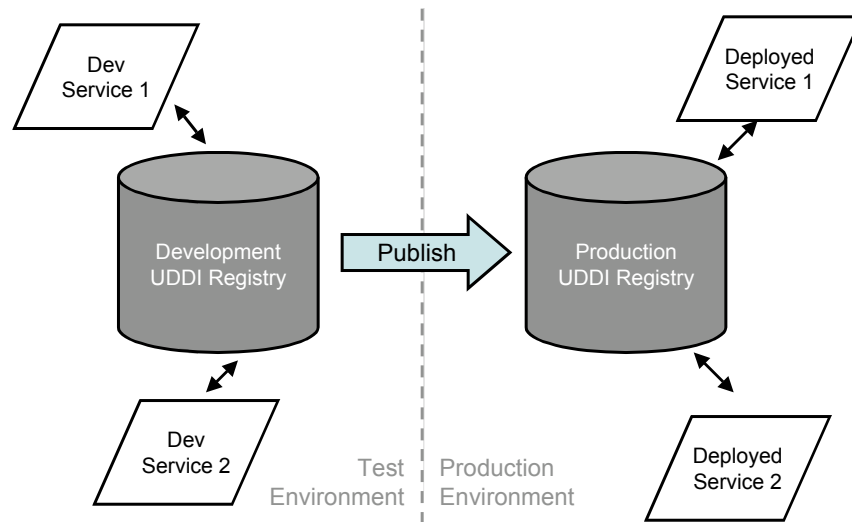
- **Need to test real-world conditions.** As software is developed, testing and debugging must occur under conditions as close to real-world production environment as possible and, in fact, incorporate several external, functioning services in the test scenarios. Additionally, it is desirable that as few modifications as possible be made to the component software to switch from “test” to “production” mode.
- **Clear separation between production and test systems.** At the same time, development versions of software must not interfere with actual production systems. Because services can be highly distributed and are loosely coupled, maintaining this distinction is paramount to ensure that dependencies are managed systematically.
- **Requirement to support distributed developer base.** Developers using the system may be based world-wide and, in fact, use different platforms and technologies from group to group. As a result, interoperability and support for a variety of network connections is an important functional requirement.

#### *Description of Solution*

The IT organization develops a test environment that utilizes a “one-way peering” model of registry interaction to create two overlapping domains for services. Those in the “private” domain can interact with the outside world, but not the other way around. When development versions of software have been fully

tested and certified, they are promoted to the production sphere, using the expanded publishing features of the UDDI Version 3 specification.

**Figure 5: Illustration of Private Test Registry**



Comment: As services are certified and promoted to the production environment, the associated UDDI entities are published from the development registry to the production registry using new features enabled in Version 3 of the UDDI specification.

---

## Scenario 2: Supporting Collaboration among Trading Partners

### ***Business Scenario***

A large manufacturer has built a business based on providing “specialty” and custom fabricated plastics components on a spot and contract basis. Its role in the middle of the supply chain—between commodity suppliers like refiners and the plants of manufacturers like consumer packaged goods concerns—requires that the company manage relationships with multiple business partners and even act as an intermediary between its suppliers and customers. In order to increase its value to partners by providing visibility into supply and demand, as well as reduce its own costs of managing inventory and logistics, the company has embarked upon a program of automating a largely manual process of communicating with its suppliers using web services-based interfaces to the key applications.

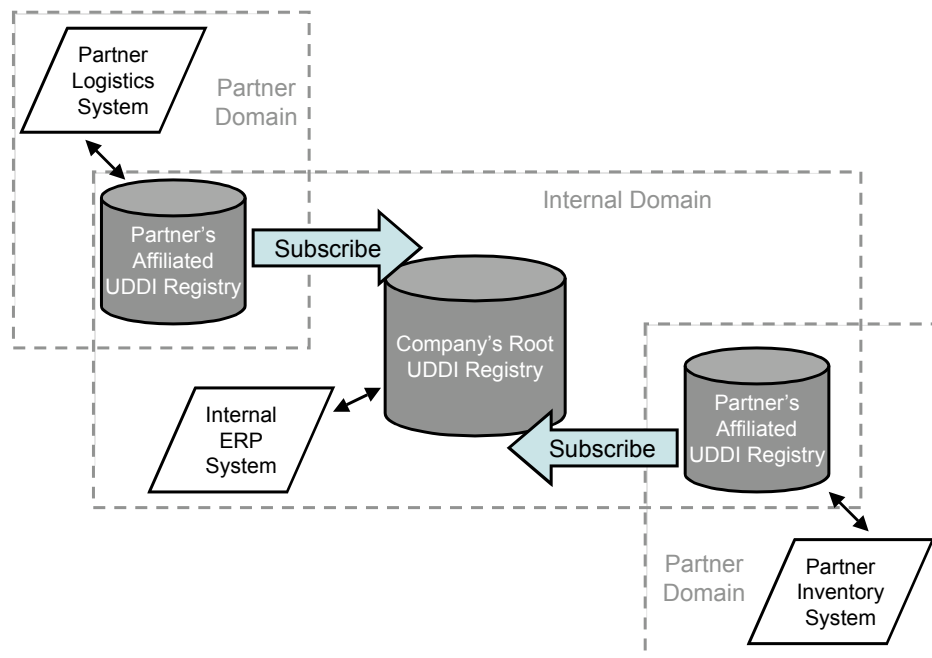
### ***Overview of Issues***

- **Interoperability.** The sources of data for the new system range from internal systems like ERP applications to third-party services like inventory and logistics tracking. Because all of these applications are established, long-running systems, standardizing on one particular platform is not an option.
- **Decentralization and collaboration.** The company’s business relationships are highly customized, and as a result, the integration infrastructure must be significantly decentralized. In fact, many of the business processes in question cannot be controlled by any single organization but, rather, require the cooperation of all parties involved.
- **Security.** Many of the systems in question are highly strategic, and information about these systems—even where they exist—may be highly sensitive and should not be shared with other companies in the network.

### ***Description of Solution***

As part of an overall web services solution, the company implements a service broker using a UDDI registry as a central element. By deploying it within the boundaries of a “DMZ” trusted environment, the company can both isolate interactions from its internal network, as well as limit the exposure of the registry to the outside world. In addition, by establishing subscription-based relationships with partners’ registries in the trading network, the company can ensure that information is fully, but safely, distributed among trading partners. The registry also implements the XML Digital Signatures support in Version 3 of the UDDI specification to ensure that only authorized parties are allowed to modify a record or entity in the registry.

**Figure 6: Illustration of Trading Partner Collaboration**



Comment: Partners use UDDI Version 3's new subscription features to monitor the company's root registry. They gain visibility to only a desired subset of all of the services available, as defined in the company's business policies.

---

## Conclusion

---

The emerging web services architecture has become familiar to most IT and e-business executives, and the model's business benefits—increased flexibility of IT assets, better integration and coordination among systems, and reduced development costs—are becoming increasingly understood.

UDDI, too, has evolved to reflect today's pragmatic business requirements. By emphasizing the interaction of private and public registries, Version 3 of the specification helps to bring the vision of wide deployment of web services closer to fruition. Indeed, by reflecting real-world use cases, this evolution supports the promise and reality of today's web services applications.

---

## References and Resource List

---

To learn more about the UDDI specification and organization, please visit the following resources.

- UDDI.org  
<http://www.uddi.org/>
- UDDI Specification V3  
<http://www.uddi.org/specification.html>
- UDDI Use Case Scenarios  
<http://www.uddi.org/>
- The Stencil Group, "Why UDDI Will Succeed, Quietly"  
[http://www.stencilgroup.com/ideas\\_scope\\_200104uddi.html](http://www.stencilgroup.com/ideas_scope_200104uddi.html)