

---

# *Driving Business Value with SOA in Insurance*

A JOINT WHITE PAPER WITH  
SUN MICROSYSTEMS

*This white paper contains two sections:*

- *an independent analysis of the role of SOA in delivering business value to the insurance industry, written by Matthew Josefowicz, manager of the insurance group at Celent, and based on Celent's independent research.*
- *an overview of Sun's solution suite for SOA in insurance, provided by Sun Microsystems.*

*For more information about Sun's Insurance SOA offerings, contact Steve Quan at [steve.quan@sun.com](mailto:steve.quan@sun.com)*

---

**CELENT**



**TABLE OF CONTENTS**

1.1 WHAT IS SOA? ..... 3

1.2 UNDERSTANDING THE  
BUSINESS VALUE ..... 4

1.3 EXAMPLES ..... 7

1.4 CURRENT ADOPTION AND USAGE ... 8

1.5 SUCCESS CHECKLIST ..... 9

2.1 OVERVIEW OF SUN'S SOA  
ARCHITECTURE FOR INSURANCE ..... 11

2.2 CASE STUDY ..... 16

---

## 1.1 WHAT IS SOA?

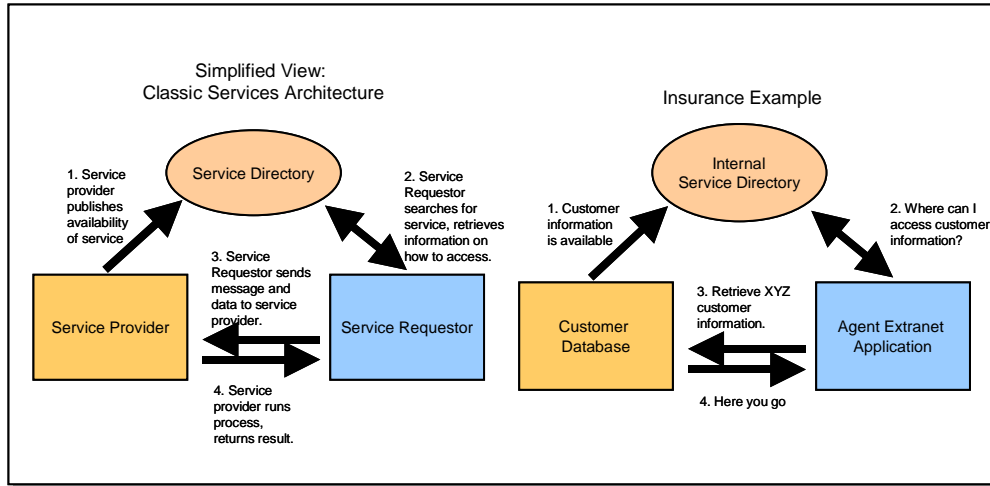
SOA, or service-oriented architecture, is a technical framework that allows faster, lower cost sharing of data and processing power across a heterogeneous IT infrastructure. SOA is platform-neutral and uses standard data types, structured text-based messages, and open transport protocols.

It is a way to expose an application or data on one computer to requests from other applications on another computer without costly and time-consuming point-to-point systems integrations or closed middleware platforms. An SOA uses text-based messages, typically written in Extensible Markup Language (XML), although UML can also be used. SOA can be used both internally, over an internal TCP/IP network, or externally, over the Internet, in which case it is often referred to as a “Web service.”

A services architecture includes at least two components: a service provider, which stands ready to receive processing requests and data; and a service requestor, that is, a system that needs to send data and processing requests and stands ready to receive a result.

A “classic” services architecture includes a service directory (also referred to as UDDI, for Uniform Description, Discovery, and Integration), so that those applications in need of a service and the providers of that service can find each other dynamically, without previously being informed of each other’s location on the network. However, many services architectures use a direct connection that does not involve a service directory.

Figure 1 illustrates a “classic” services architecture, on the left, and shows how the architecture might be applied in an insurance context.

**Figure 1: Classic and Insurance-Specific Services Architecture**

Source: Celent

A service directory adds additional flexibility to the network, but increases the initial complexity. Most insurers that have embraced SOA tested the waters with several direct connections before implementing a service directory structure.

## 1.2 UNDERSTANDING THE BUSINESS VALUE

Reducing the cost and complexity of both internal and external integration is a critical need for insurers seeking to:

- Improve time to market
- Increase IT flexibility to take advantage of changing market conditions
- Mitigate negative effects of legacy systems
- Reduce overall costs and improve profitability

By providing a simple, reusable, platform-independent systems integration methodology for both internal and external systems, services architectures give insurers significant competitive advantages. Insurers that do not embrace services risk being left behind by those that do.

### 1.2.1 INTERNAL SYSTEMS

Integrating systems and business processes is a tremendous burden for the insurance industry. Celent estimates that, in 2005, US insurers spent approximately US\$2 billion on internal and external integration for new IT projects alone, or roughly 7% of the overall spending on IT. Integration is typically the most time-consuming and difficult part of new IT projects in the industry.

But the increasing use of SOA is making it more cost-effective to make additions and changes to heterogeneous infrastructures.

SOA allows for the creation of reusable integration points. Once a system is service-enabled, it can speak to any other, similarly enabled system. This can save a tremendous amount of time and expense when compared to integrating multiple systems point-to-point with direct connections between each system.

For example, consider the problem of linking three front end systems (e.g., an agent extranet, an underwriter workflow system, and a call center management system) with four back end systems (e.g., two customer databases, a claims system, and a billing system).

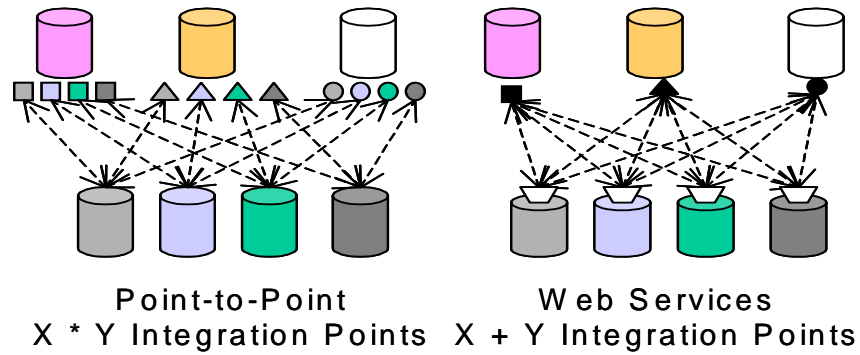
Using direct, point-to-point connections would require 12 integration points (three front ends multiplied by four back ends). Using Web services, it would only be necessary to create 7 integration points, one on each system (three front ends plus four back ends).

When dealing with a large number of systems, either disparate internal systems or a large number of partner systems, using a services architecture with reusable integration points requires only that each system be service-enabled once, rather than one time for each partner.

So, integrating X number of front end systems with Y number of back end systems would require X times Y point-to-point integrations. But it would only require X plus Y service-enablings. As X and Y get to be larger numbers, this can represent a significant, and sometimes enabling, efficiency.

This is a key issue, especially as more insurers seek to carve out outmoded components of their legacy systems and replace them with newer, more function-rich and more flexible components. By service-enabling their legacy systems, this process becomes much simpler.

**Figure 2: Web Services Creates Reusable Integration Points**

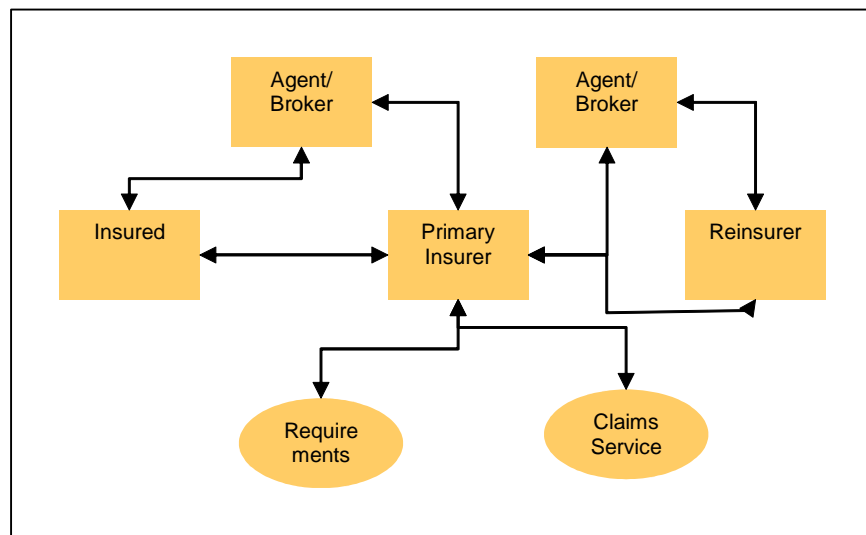


Source: Celent

### 1.2.2 EXTERNAL SYSTEMS

Insurers conduct business at the center of a complex multiparty value chain. Data, and sometimes even transactional capabilities, must be shared with agents and brokers, insureds, employers, reinsurers, third party service providers, requirements vendors, regulators, and others.

**Figure 3: Insurance External Communication**



Source: Celent

Rather than building a direct connection to a partner's system by writing to its application programming interface (API), a services connection allows a looser link that is easier to implement and more reusable.

Services also offer a number of advantages over traditional electronic data interchange (EDI), which is still the most common way for insurers and their business partners to exchange data electronically.

- Services are reusable for additional business partners. EDI formats and connections are typically customized for each business partner.
- Services may be reusable internally as well, allowing the company to leverage its investment to solve internal integration problems.
- Services can use public networks (i.e., the Internet) rather than relying on private network connections between partners, which can be expensive to maintain.
- Services are easily expanded to include more information that needs to be shared. EDI data formats are essentially "hard-coded" and difficult to change. Because XML is a flexible language that uses metadata tags to define data elements, it is easier to introduce a new data element into the message.

---

### 1.3 EXAMPLES

In addition to the detailed case study presented on page 14 of this paper, the following are some of the many examples that Celent has encountered of insurers using SOA to achieve real business results:

- A large diversified life/annuity and personal lines insurer was able to use SOA to integrate a wide variety of disparate back end systems to power its agent portal in just 31 weeks, allowing it to meet a critical competitive need. The integration points built for the project were reusable in subsequent projects, reducing the cost and time as well.
- A regional personal and commercial lines insurer used an SOA services layer to wrapper green screen mainframe applications to feed both a proprietary agent portal and upload/download to agency management systems. The SOA has significantly reduced the time and

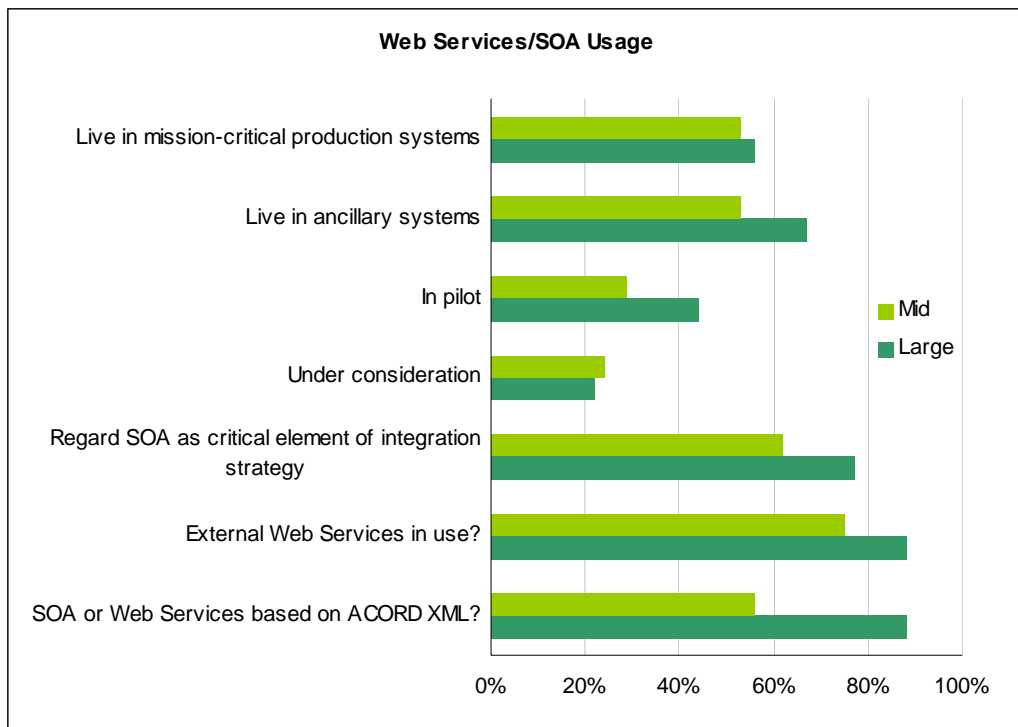
effort required to integrate the agent portal with additional lines of business.

- A multiregional auto insurer used a services architecture to share policy and coverage data on a real-time, “as needed” basis with a network of glass providers, replacing the inefficient and undesirable practice of sharing a flat file dump of the entire policy database with each partner.

## 1.4 CURRENT ADOPTION AND USAGE

SOA is in wide use in the insurance industry today. In Celent’s most recent insurance CIO/CTO survey, the majority of respondents across both the large (over US\$1 billion in annual premiums) and midsize (US\$100 million to US\$1 billion) reported using SOA in mission-critical systems today, and more than half regard SOA as a critical element of their application integration strategies.

**Figure 4: Web Services and SOA Usage**



Source: Celent 2005 US Insurance CIO/CTO Survey

While only a minority of respondents (five) reported having a UDDI or service directory structure in place today, others reported planning to implement one within two years. For many insurers, the first step is to build a few direct services, and then implement a service directory as the number of services in the enterprise grows.

Celent believes that this is the correct approach, but that insurers should be aware of the probable future need for a service directory structure and plan accordingly in the earliest days of their SOA strategies.

---

## **1.5 SUCCESS CHECKLIST**

The majority of insurers have already started to deploy SOA and are seeing significant efficiencies in both time and cost in adding to or modifying their systems once services are in place.

Celent advises insurers that are formulating or refining their SOA strategy to keep the following issues top of mind:

### **1.5.1 START LOCAL, BUT PLAN GLOBAL**

Most insurers that have approached SOA have linked their early efforts to a single, defined project with clear business value, for example integrating multiple back end systems into an agent portal. Although this is a great way to start, insurers must make sure that the architecture decisions that they make in these initial projects will allow them to achieve enterprise-level efficiencies.

### **1.5.2 INVEST IN STAFF TRAINING**

Maximizing the value of an SOA investment requires making sure that your IT staff understands and is comfortable working with the architecture. In Celent's discussions with insurers about their SOA strategies, many cited the need to upgrade staff skills as one of the challenges in successfully embracing SOA. But all reported not only better productivity but also higher satisfaction levels among staff who appreciated the chance to keep pace with modern technology.

### **1.5.3 ENSURE SOA SUPPORT FROM KEY VENDORS**

Once insurers embrace SOA, they should try to make sure that any new elements or applications that they add to their infrastructures support SOA as a primary method for integration. While building services layers for new infrastructure elements is better than doing point-to-point integration for them, insurers that have made the investment in SOA will find that most modern vendor solutions are ready, and even eager, to support that investment.

---

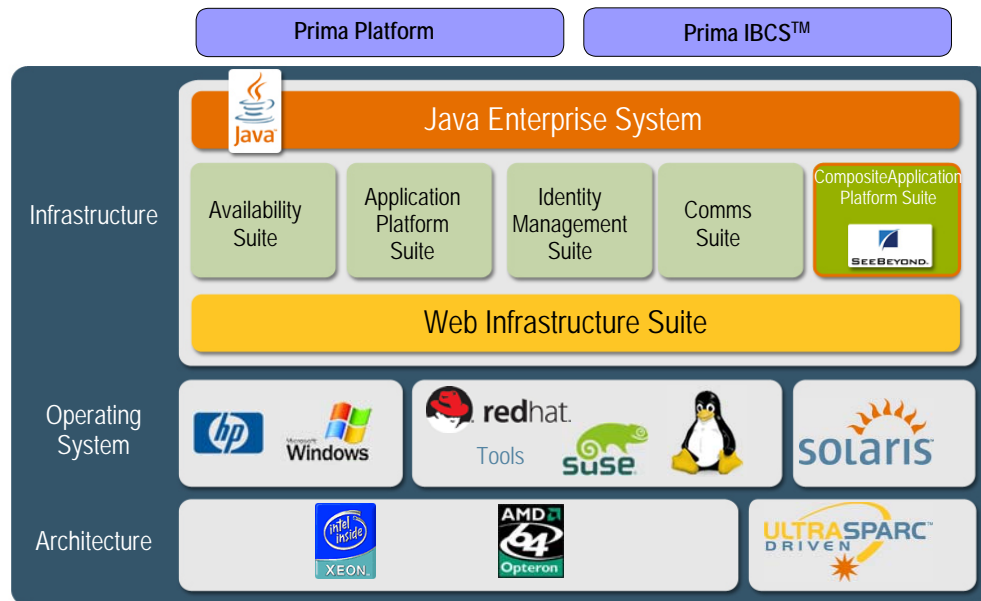
## 2.1 OVERVIEW OF SUN'S SOA ARCHITECTURE FOR INSURANCE

Sun's Insurance SOA solution includes both the Sun Java Composite Application Platform Suite (Java CAPS – Java Enterprise System and Seebeyond) with partner products like Prima Solutions. These offerings equip insurance companies with solutions built on ACORD and J2EE standards to help them wrap-and-extend the capabilities of their legacy systems rather than rip-and-recreate applications to meet current needs.

Sun and its partners provide tools and an insurance model to integrate legacy systems, implement business process management, orchestrate business services, monitor business transactions, and all tied together with a company-wide definition of business terms (which Prima calls a company-level semantic).

### 2.1.1 SOFTWARE INFRASTRUCTURE

Sun's Java Enterprise Suite (JES) is an integrated and complete middleware stack containing standards-based products that can be used together or with other standards-based 3rd party products. The products includes portal, messaging, identity, web and application servers. The components are designed to work together. They are tested and can be easily deployed together, resulting in significantly lower total cost of ownership (TCO). One of the JES components is the Java Composite Application Platform Suite (formerly SeeBeyond's ICAN). It is a pre-packaged J2EE component framework that enables rapid construction and deployment of stable and flexible insurance applications. It offers a service-oriented architecture (SOA) including wizards to create reusable services from existing systems or pre built templates, business process models that can assemble composite applications, and business activity monitoring of transactions.

**Figure 5: Sun's SOA Insurance Architecture**

Source: Sun Microsystems

### 2.1.2 NEED FOR A UNIFIED PIVOTAL DATA MODEL IN AN SOA

Legacy systems and independent projects often use different data models (with different formats, values and meanings). Building processes and services across systems, as promoted by an SOA approach, requires data transformation steps that are costly to build and maintain unless a common pivotal enterprise model is defined.

Semantics are the greatest challenge of SOA. Application development needs a common data model for your systems to call and understand each other seamlessly. Defining this common semantic is one of the main difficulties insurers face in the deployment of large scale SOA because of the complexity of the underlying business.

The insurance industry is varied in terms of functional coverage with many different business lines involved: life, non-life, and health among others. It is often possible from one company to insure your car, home, health, life, and to protect yourself against being unable to work and to plan for your retirement. These lines are valid for both personal and commercial environment and can be distributed directly by the insurance company or via intermediaries such as brokers or independent agents.

The range of data involved in insurance contracts, and the variety of relationships between those data make the creation of an insurance object-oriented model very complex. In fact, there are several hundred classes and relationships necessary to properly model the insurance industry.

### **2.1.3 A PIVOTAL MODEL BUILT FOR ADAPTATION AND CHANGE**

This variety and complexity has another impact. There is no “one size fits all” approach that can work. Complexity creates costs. “Universal” insurance models have failed in the past because they were too difficult to grasp, modify, or even adapt so that it contains only what is relevant and useful for a particular insurer. A pivotal model helps you define a data model as well as integrate the data elements with defined business workflow. This model needs to be streamlined to the specific context and product portfolio of each insurer. In fact when properly adapted, it becomes a key asset of each insurer.

The capability to start an SOA transformation from a fully documented and adaptable reference model provides a clear head start that reduces costs, risks, and time to deployment. Full benefits are achieved when a reference model is used as a starting point in this venture, with a pragmatic and “Keep it Simple” approach to adapt it. The starting point model needs to have been designed for adaptation in mind, not for universality.

Prima IBCS™ (Insurance Business Component Suite) is an extensive insurance object model covering most of the insurance domain areas. It is the result of several years of R&D by insurance experts and object oriented specialists with experience in large distributed information systems. Prima Solutions has described in UML over 100 processes and process variations, decomposed these processes into activities that describe the services the information system must provide to support the business. It contains interface definitions to dozens of business services consistently defined and documented with the data manipulated.



**Figure 7: Functional Domains in Prima IBCS 4.2**

PARAMETERS	REFERENCES	BUSINESS	FINANCE
Product	Marketing	Underwriting	Investment
Document	Subject Matter	Contract	Finance
References	Party	Claim	Reinsurance
User Management	Contact Management	WF Activity Management	
	Partner	Loss Adjustment	

Source: Sun Microsystems

Even with an extended model, it is likely that changes in the regulatory environment and the complexity of existing insurance contracts will mandate the initial and ongoing adaptation of this model to meet the individual needs of an insurance company. The model has been designed to be easily adapted and extended, leveraging best of breed design patterns, providing online documentation, and following industry standards (UML, XMI, XML, or others).

#### **2.1.4 AN APPROACH TO INDUSTRIALIZE CHANGE, PRODUCTIVITY, AND CONSISTENCY**

Prima IBCS™ V4.2 is delivered with a template-driven code generation toolset following a Model Driven Architecture (MDA) approach. This toolset automates the creation of ready-to-use implementation assets. It guarantees traceability and consistency between the model, the documentation, the implementation, and its integration within SOA platforms, BPM, and rules engines. From the model, you can generate the exact assets needed by Sun's Java Business Integration Suite and maintain them over time.

---

## 2.2 CASE STUDY

CNP is a leading European life, health, and pensions insurance company. In 2004, it had premium income of over €21 billion, with 14 million policyholders. Its distribution strategy relies heavily on partnerships, with leading partners including the French post office system (22,000 offices), over 250 savings banks, and 50 financial institutions and mutuals.

CNP needed to streamline its communications with these partners, and extend its partnership network while improving time to market and cutting overall IT expense by 30% over three years.

CNP selected Prima Platform as an integration layer to build reusable connections between its many back office and front office systems. CNP's back office included a mix of homegrown legacy applications running on OS/390 and multiple flavors of Unix, as well as some packaged applications all integrated by some homegrown middleware and an industry standard EAI solution. The front office solutions included many COBOL applications as well as newer J2EE and portal-based solutions. CNP needed to be able to integrate with a wide variety of partner systems and to decouple their front office and back office applications.

By using Prima Platform as the basis of its SOA strategy, CNP was able to complete three critical projects (two partner extranets and a claims workflow) in record time, reusing many components. With Prima's built-in standard insurance components, CNP developers were able to spend less time on research and design.

As CIO Phat Chua Lim stated, "Reducing the cost and time required to deliver new insurance applications implies leveraging existing legacy applications, favoring the re-use of components and services, and reducing complexity by enforcing a structured approach based on standards in our various projects. With Prima Platform we have acquired a strong standards-based software foundation which delivers all the key insurance and technical assets required to support service-oriented insurance applications."