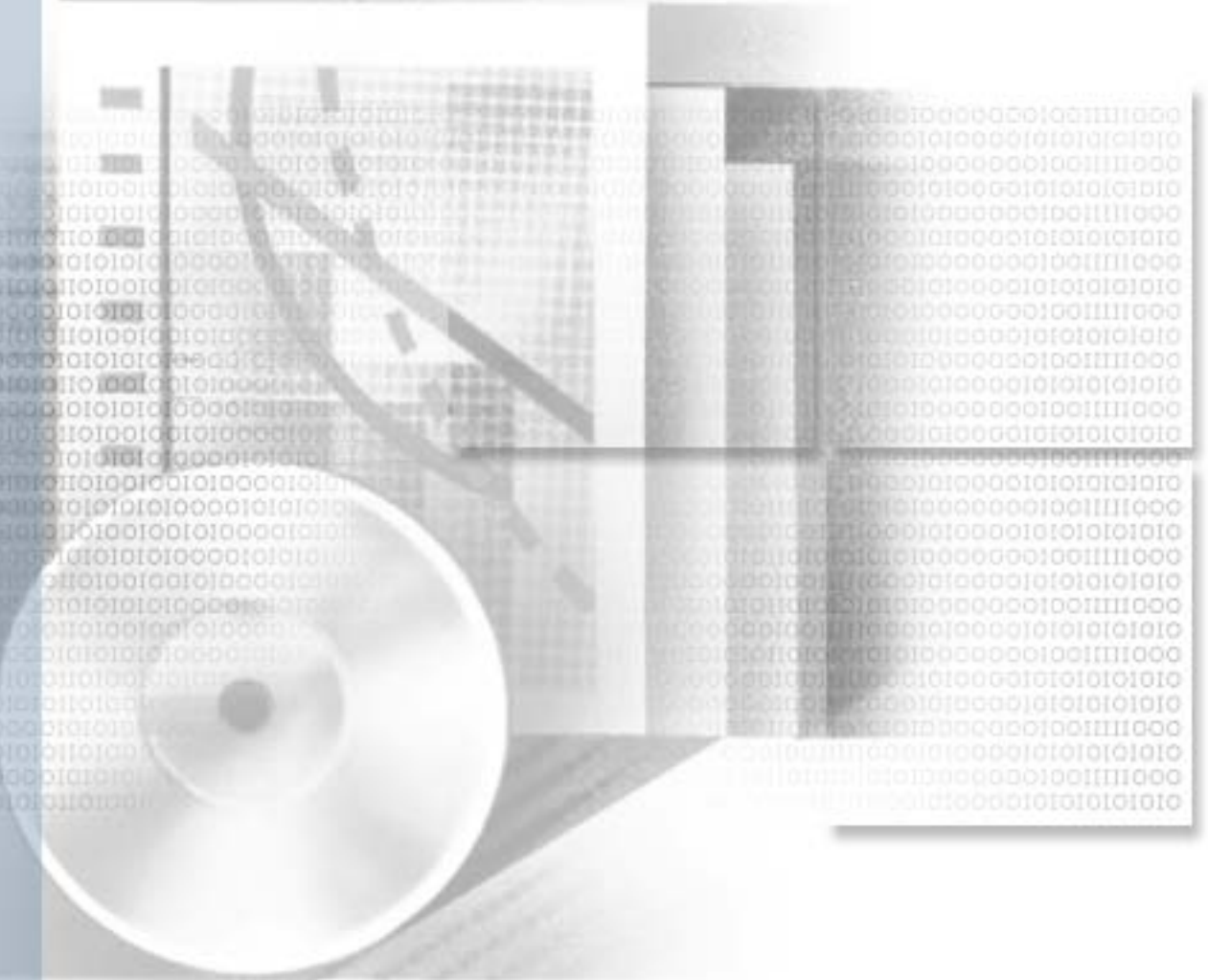


# Technology Spectrum



eXensys compliant with SOA  
(Service Oriented Architecture)

Exensys offers Web based Enterprise products based on specific verticals. Exensys provides solutions in that offer flexible usage and deployment for respective verticals such as Exensys Industry Retail Solution, Exensys Industry Manufacturing Solution, Exensys Industry Trading Solution and Exensys Industry Automotive Solution; Strategic business suite that includes Exensys BI, Exensys CRM, Exensys SCM, Exensys DMS and Exensys HCM. eXensys Enterprise Management Suite 4.0 can be deployed as vertical solutions, pre- configured or tailored to meet specific customer's requirements. Exensys 4.0 is developed on Service Oriented Architecture for more flexibility and easy integration with other systems.

While designing the eXensys product architectural framework, the technology and business leaders at Exensys did extensive knowledge storming sessions, debated and discussed on how to create higher value for global customer and key stake holders. The vision was that the customer must derive long terms benefits and the system should be flexible enough to seamlessly integrate with disparate systems, with islands of information, and significantly simplify the effort required to deploy and manage distributed applications within and across.

With a strong vision to build world class enterprise product the leaders proposed that the architectural framework for enterprise product should be SOA compliant. It must adhere all the standards of SOA framework. The product eXensys 4.0 and Managed Business Application Services uses XML message exchange using communication protocols, usually SOAP. Service description is accomplished using WSDL. Underlying and enabling all of this is metadata which is sufficient to describe not only the characteristics of these services, but also the data that drives them.

The product offering has been designed based on service and reusability aspect, it is well-defined, self contained and doesn't depend on the context or state of other services. It has followed the guiding principles of Reuse, modularity, compos ability, granulaty, componentization, Portability,and interoperability.

## **What is SOA?**

SOA is a software system structuring principle, based on the concept of services, which are offered (and described) by service providers. Service consumers read service descriptions, and using ONLY those descriptions, can interact with service provider, getting access to whatever functions it provides.

In today's world of shortened product cycles and hyper competition, IT's mission is to create a flexible environment; to ensure that the enterprise is strategically positioned to foster innovation; to respond to changing needs faster than ever before by reducing time to market for new services; and to drive down the cost of integration and total cost of ownership (TCO). This flexibility is made possible through the use of a Service-Oriented Architecture (SOA) — a design paradigm based on a loosely coupled collection of reusable services. The SOA enables agility through aligning the business and IT, providing business processes that embody core capabilities to employees, customers, suppliers, and partners. The set of infrastructure tools employed by IT

to build, configure, deploy, monitor, and manage services is called the SOA platform. As technologies that facilitate service communication and orchestration are standardized, enterprises are able to fully realize the benefits of SOA without fear of vendor lock-in.

SOA has existed in one form or another since the 1960's. The technical "perfect storm" of the Internet, TCP/IP, HTTP and XML have created the conditions for yet another incarnation of SOA to emerge. Due to the almost universal support for those technologies, this version of SOA has the potential to have a wider, longer lasting impact than any previous one. XML message exchange using web protocols, usually HTTP over TCP/IP. Service description is accomplished using WSDL.

### **WHY SOA Framework:**

A SOA-based approach breaks down applications into services and enables them to be used by other business processes. It makes incremental changes that put selection pressures on systems so they can evolve. By facilitating business services that support customers, that approach solves problems today.

To obtain agility, development teams must reduce complexity. Complexity increases the time and effort required to enhance or modify applications and services. Service-oriented patterns to reduce complexity include:

- Reuse and composition. This is particularly powerful for creating new business processes quickly and reliably.
- Achieving better alignment between business and IT
- Recomposition. The ability to alter existing business processes or other applications based on service aggregation.
- Creating more flexible and responsive IT infrastructure
- The ability to incrementally change the system. Switching service providers, extending services, modifying service providers and consumers. All of these can be done safely, due to well-controlled coupling.
- The ability to incrementally build the system. This is especially true of SOA-based integration.
- Simplifying integration implementation

### **An SOA Platform**

While designing An SOA strategy we must identifying the core platform collaboration standards it will use, and insuring that SOA platform vendors must conform and collaborate via open,

industry standards at both the protocol and services levels. SOA is the practice of sequestering the core business functions into independent services that don't change frequently. These services are glorified functions that are called by one or more presentation programs. The presentation programs are volatile bits of software that present data to, and accept data from, various users.

Proprietary collaboration dependencies between SOA elements weaken the SOA platform and in-turn will reduce flexibility and will introduce risk elements. While an SOA can include homogeneous or heterogeneous components, it is important to maintain an underlying modularity that delivers flexibility to mix and match components when designing an IT infrastructure tailored to an enterprise's requirements. This means that, while the path to an SOA may consist of incremental steps, it is essential to identify in advance the key areas needed to create a successful SOA platform.

An SOA platform provides the tools and underlying software infrastructure that enable a business to utilize the SOA. It is made up of platform services that utilize the same SOA principles and standards as the SOA business processes built with it, where each service is a point of platform collaboration. Since an SOA is an extension of Web computing, it has no central point of control — no single facility can impose its will over the platform. Instead, a diverse set of system elements collaborate with each other via shared platform services. Each element must be able to collaborate with SOA platform services such as deployment, identity, management, monitoring, and more.

At the enterprise architecture level, it is always about the business. The enterprise architecture perspective should be focused on the business needs in order ensure that IT enables business to deliver superior results.

### **SOA Platform Design Centers**

An SOA platform composed of five design centers and these are as-follows.

- ▶ **Service Composition**

Development, evolution, and governance

- ▶ **Service Control**

Policy, management, and monitoring

- ▶ **Service Delivery**

Mediate communication

- ▶ **Service Access**

End usage of services

- ▶ **Composite Application Platform**

Abstraction of business logic from multiple applications into an agile solution