

# Service Oriented Architecture

## Gartner Group

According to industry analysts Gartner Group, by 2008 more than 60 percent of enterprises will use Service Oriented Architecture (SOA) as the guiding principle when creating mission-critical applications and processes.

In 2007, Gartner Group held a conference on Service-Oriented Architecture. The purpose of this white paper is to look at the key issues for the IT community in Australia/New Zealand identified by the Gartner Group and discuss some of the implications for the ERP sector.

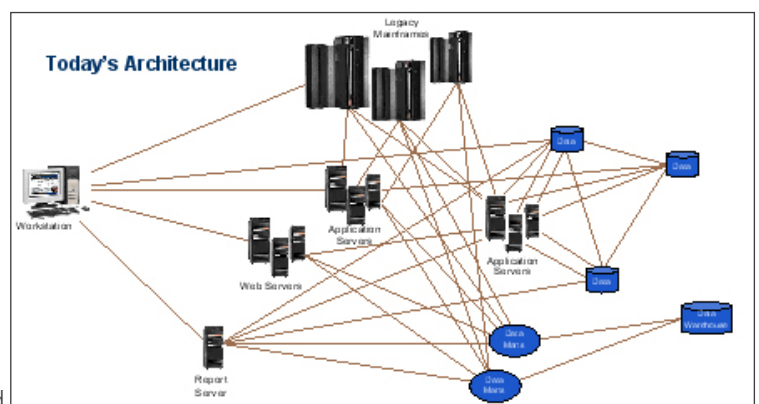
### A Historical Shift in Focus

SOA is part of a larger shift within the IT sector from implementing solutions that offer technical convenience to developing architectures that can address increasing business complexity. The drive to consolidate and rationalise at the start of the new millennium meant there was a focus on technical architecture and technical standards. As business integration became more of a driver for IT, information architecture moved to the fore with the increasing use of brokers and data warehouses.

Today, the emphasis is on business process acceleration. Companies are more willing to use SOA and web services to help address business complexity quickly and efficiently. The result has been business-contributing architecture that allows a company to implement business processes to improve their bottom line, rather than processes that suit their legacy system. The payoff for businesses has moved upstream, justifying increased spending on IT resources as the benefits and ROI become more evident to top level management.

### SOA Defined

Whether introducing a new ERP solution or adding new applications to the legacy system, disparate components need to be able to talk to each other and, increasingly, talk to the organisation's 'business partners' and customers' systems as well. Traditionally this has meant developing new code that enables different systems and applications to function together,



a complex and time consuming process that can hamper the introduction of genuine business solutions and slow the development and streamlining of business processes.

SOA splits the programming into two distinct areas - Business Logic and services which are regardless of the application, and the abstraction layer (the method of putting services together to create a business process). The theory behind SOA is that if services are created with common standards and are easily found, then reuse can occur. As a result, the programming effort behind building new business process is mainly at the abstraction layer. To simplify this, software companies are now releasing drag and drop abstraction layer software that can enable business analysts to create the abstraction layer.

While this is all good in theory, there are a number of challenges with this approach:

- **How do you start**

Firstly, how do you go from having no defined services to having all your business logic defined?

- **How do you drive the culture change:**

How do you get your programmers to move from a culture of cutting code for a specific problem to stepping back and using defined standards to cut code that is extensible for general use?

- **What is the impact on the IT team**

In general most IT teams are set up in functional/application silos, whereas SOA implies that the teams will be reorganised along business process lines.

---

## How do you start

Most organisations attempting to move down the SOA path start slowly with pilot projects. These projects allow them to begin using the new tools (repositories, middle ware, abstraction languages), understand which will become their tool set of choice and what the challenges are in using and implementing the new method.

Once these are understood, the challenge becomes one of governance. Standards must be agreed upon and services need to become visible to be reusable. It is recommended that companies introducing SOA establish a central management group to coordinate interdependencies between software projects to ensure they follow the same standards and are reusing as much as possible.

The other major problem is that the pilot project, setting up of governance and building the initial services rarely have an immediate return on investment. In fact, the first SOA pilot projects have increased cost by adding governance and building services for repeatable use, without any benefit of reduced time.

Therefore SOA must be a strategic investment that the business understands is a change to the way things are done. The problem is, when telling the business of the benefits of lower development costs and faster time to change business processes, they will agree it is strategic, yet want to see the benefits immediately.

To avoid this pitfall, several companies are trying to introduce SOA by stealth. While this does prevent unrealistic business expectations, it could make the early phases more problematic.

---

## Changing Culture in the IT Department

SOA does not mean all business processes should be built using services. All of this evolution in the IT environment is closing the divide between systematic and opportunistic IT projects. Opportunistic projects, in recent developments such as GPS, wikis, virtual worlds and mobile technology, can give companies competitive agility. Systematic projects involving security, tera-grids and data integrity, on the other hand, provide conservative reliability. The flexibility of an SOA environment allows a company to do both without having to compromise the integrity of either.

The programmers will still be needed to cut quick, one-off-use code, but just not as often. The key to moving the programmers from creating one-off code for everything to only cutting code when appropriate is governance. Just saying "We are introducing SOA, so please build and use services", won't cut the mustard.

Putting in a governance team that works across projects, deciding when and where services should be built or used, is the only way to move to that middle. In years to come when SOA becomes a standard way of doing things, then it will be the only culture everyone knows and change will be easier. Until then, good governance is critical to the successful introduction of SOA.

---

## Summary

As with any change that has huge potential, SOA is not easy. Introducing SOA takes time with no immediate return on the initial cost. But if done right, the prize of agility and speed can be achieved with significant savings in the long-term, because there is no cost for creating the next application. All of the software already exists in other applications within the business – the components simply need to be recombined in different ways to create a new application.

The question for most businesses is when to start down the road. It is not well traveled at the moment, with the leaders still cutting the path and software providers still developing the tools. Even so, if your business is not thinking about it, you are now in the minority.