



## The Role of Governance in Service-Oriented Architectures

Analyst: Anne MacFarland

### Management Summary

If the Earth's axis was tilted differently, it would still rotate on its axis as before, and orbit around the Sun, but the experience of those living on it would change. Weather patterns and agriculture would be transformed. There would be massive political ramifications due to these changes. In the enterprise data center, the move from asset-based management to service-based management re-tilts the axis of optimization from resource utilization and system uptime to business value, system resilience, and better leveraging of software and hardware assets. IT systems become less efficient – but more effective. The systems will still use the same hardware, and the same applications, though they may be deconstructed into modules more granularly than heretofore. The business processes and their priorities will not change. It is *governance* - that top-down, survey-of-the-realm approach - that demands a rethink. Traditionally enforced by applications and operating systems, at a higher level, governance has always been a considerable part of the CIO's job description, but often has been eclipsed from the view of data center operations, like the dark side of the Moon.

Today, managing an enterprise's computing services involves *more overt governance*, for governance policies must be consistent over the lifecycle of the service – from development through deployment, use in aggregations, and retirement. They must be consistent across a range of services, whose granular modifications allow IT services to evolve to meet the demand for frequent tweaks. This is more than requiring a finer granularity of management of more application components (and, in consequence, more use of automation). It is the truing of data center operations to how the business organization operates, and managing it by business results.

What this governance involves depends on the extent and character of your SOA environment. Is it internal to a business unit or to an organization? Or, is it external as well, involving partners? Will it involve commercially offered software as a service? The inherent security must be sufficient in all cases, but its nature, and that of service descriptions and requirements will be more extensive, the greater the domain that they address.

Of course, you will start small, with a pilot project where you may be able to scrimp on the full management support needed for a larger scale of implementation. Nevertheless, it is important to consider that larger scale early, and design the needed governance into your models. The basic elements are a Registry/Repository (united or as two separate components), a Mediator, and a Runtime. With these, in theory all manner of services are possible. Which possibilities are also prudent deserves closer analysis. For more details, please read on.

### IN THIS ISSUE

➤ Moving from Cliques to Teaming .....	2
➤ SOA Components .....	2
➤ Enforcing Governance .....	3
➤ Questions to Consider .....	3
➤ Conclusion .....	4

## Moving from Cliques to Teaming

A service-oriented architecture (SOA) re-orientes your software components from the cliques of application and physical location to a flexible teaming of effort across the larger data center environment. Reuse of application components, including the consistency of key processes across environments, is a very good idea. Software licensing becomes more rational and often less extravagant. More congruent processes can be more easily leveraged in combinations. Moreover, focus can be changed from *process inconsistencies* to the discovery of *sources of competitive advantage*.

Service-oriented architectures contain many tools to build competitive advantage. Modeling of business components reveals sub-processes that, with acceleration, could improve business processes. It also shows how and where processes can be re-architected for cost control and more-effective use. Strategic analysis of a business component model can also show what components can be used more innovatively to provide competitive advantage<sup>1</sup>.

The controls of a service-oriented architecture are not in the knobs and dials that control asset use (though they remain a part of infrastructure management), but in the architectural elements of the service-oriented architecture. It is in these points of control that governance must be built, as well as in the services themselves.

## SOA Components

### *Registry/System of Record*

The registry is where available services are discovered. It should be easily navigable, and its elements are, traditionally, highly standard. In the old days of paper, this meant a registry was inevitably limited in scope, just as a database, traditionally, is limited to universally-used fields. Now, with search capabilities, standardization of component description is still essential but standardization of description schema is not. This increases the flexibility of the registry.

Once a service is published to a registry, it has to be available at all times to all processes

that wish to use it. If you can't provide the ruggedness yourself, out of the considerable savings in reduction of redundant applications and environments, then you should outsource the service from someone else, and have your applications call that service, as they need it. Service-oriented architectures do not work with flimsy infrastructure.

### *Repository*

There is value in a repository separate from the registry, if only to contain service contractual details that are peculiar to different services and details that are subject to change (pricing, extensions, etc.). With search, these disparate particulars could be contained in a registry, giving one less thing to manage and one less thing to synchronize, but such unity may not be wise.

The needs of participants may differ, particularly in a SOA with multiple organizational participants. Think of a value chain with suppliers and distributors. In such a situation, aspects of the service, such as security and data use, will depend on the identity of the consumer of a service. Consumers of a service may even have their own particular specifications. This occasion for arbitration might be true even within an enterprise, if the multiple business units compete for budget dollars.

There may be a need for confidentiality. Commonly used functions are easy to share, but common analyses of data sources may be more controversial. Think about the fiefdoms of politics and culture in your environment, and whether you want to institutionalize that arrangement in your SOA.

If the registry contains services from diverse providers from different organizations, ownership and protection of ones service description becomes important. The ability to broker a service engagement from both consumer and provider side may need to be a basic part of the architecture.

It all comes down to your governance domain and your level of ambition and expectations. Is the SOA effort just to rationalize your IT assets, save money, and make permutations and recombinations of those assets easier? Alternatively, are you looking to add new capabilities, new alternate routes to market, and really integrate with your suppliers and distributors. The more ambitious (and external)

---

<sup>1</sup> Frequently, the advantage is derived by working across traditional organizational boundaries – something that is much better supported by a service-oriented architecture than by point-to-point one-off integrations.

your plans, the more that tiering your resource description particulars into a separate repository in order to control access may make sense.

### ***Broker, Enterprise Service Bus, and other Forms of Mediation***

How services are mediated, and how that mediation is managed, obviously are critical. A wide spectrum of approaches exists at present, ranging from the messaging-centric approach of the Java runtime tradition to the more traditional mediating gateways (or spokes) of *Enterprise Application Integration*. The concept of *Enterprise Service Bus* is used to describe an element that has comprehensive translation and mediation capabilities.<sup>2</sup> For the service-oriented architecture that is more organizationally heterogeneous,<sup>3</sup> the use of contracts and instrumentation to alert their violation may be a way to offload some of the policy enforcement from the service bus for the duration of the service.

A service-oriented architecture cannot be conceived only from the service provider point of view, even though many data centers have thought of themselves as service providers for years. With SOA, the data center will be both provider and consumer, and its attitudes in each persona will differ. On the consumer side, there should be the same well-thought-out requirements, though the preferences may change more frequently. As a provider, the data center must know what is negotiable – and what is not. Separation of concerns is needed.

### **Enforcing Run-Time Governance**

Governance is more than enforcing qualities of process service. The quality of service, or lack thereof, is a matter of data center operations. Management of governance issues is more a matter of the information used (its currency, its completeness, its confidentiality), and the users served (their level of trust, their range of access to assets, and their privacy concerns). Customization for localization, personalization and any other facet of service built into the mediation that is outside of service delivery becomes part of the governance realm.

SOA governance is instantiated by policies

---

<sup>2</sup> The specifics of the extent and capabilities of the Enterprise Service Bus vary by vendor.

<sup>3</sup> Say, including software as a service from an external provider.

set up as part of the development process. It can be enforced by audit-by-query or by a qualification process sometimes known as a rules engine. Built into a service, its enforcement must start at the initiation of the service, and be facilitated by instrumentation that also monitors the relevant metrics over the course of the service.

### ***Topography***

The enforcement of governance depends on the topography and geographic distribution of the systems involved. Gateways and precipitated agents both work, but they each exert different penalties on system performance, and they each have different roles in system security. Because service-oriented architecture presumes multi-tenancy, security must be built in both to protect the service from perversion or harm, and to offer users protection as part of the service.

In SOA, governance moves from an application or operating-system-bound capability to a rules-based capability. A lot more has to be explicit – but a lot more can be said – and many more options can be supported. Management of these policies should be consistent from creation, aggregation, and enforcement to reporting. What is surfaced to people in different roles can be minimized to a need-to-know to cut down on overload.

### **Questions to Consider**

#### ***How fast are your needs evolving?***

This affects sourcing. How vulnerable is your business to attack by lightweight start-ups offering immediate gratification without the back end ruggedness to back the claims up over the long term? Can you, through SOA, achieve the client-facing legerdemain that shouts *just-as-easy* to your customers?

If your industry is going through a rapid period of change (as what industry is not), factor your future industry role into your plans. You may have to be more collaborative with external forces in the future. If so, you should architect for robustness in the form of contracts, mutable service definitions, strong but non-restrictive governance policies, and for security to keep the things that you want immutable assuredly so.

### ***What is the shape and extent of your SOA domain?***

Will all the technology assets be owned by the consumer of the services? Is that strategy a supportable use of capital? Has your enterprise's IT strategy been focused on the use of consolidation to reduce the cost of IT management? SOA, though it can reduce the number of IT assets that must be supported, is also focused at more fully supporting the business. This may require more people, operating in different modes.

### ***What is the shape of your organization's application use?***

Stovepipes? Linked by a common database? Linked by workflow? How atomic do you need the services of your service-oriented architecture to be? While some (the sand castle builders) think pulverizing existing mega applications to stone dust is the best way to get the most re-use, others may see inherent efficiencies in keeping larger sub-processes (that will not need to be modified) intact. Take credit lookup – if the process of credit lookup is to be modified, it may be easier to swap in a new version of the whole process than to make it modifiable (especially given the security needs of that particular component).

While these last questions seem a part of operations, governance is needed here to keep the service structures that are built consistent and to maintain the efficiency of the system as a whole. Service-oriented approaches do lend themselves to the lure of the quick fix. The ability to act locally does not lessen the requirement to think globally – or at least across the domain of your service-oriented architecture.

If you have used Web Services tactically, it might seem that publishing a registry would be all that is needed. But, if you are trying to wring out licensing charges by having fewer copies of duplicate applications, you may find that more policies have to be set up to arbitrate use of a service by multiple parties. You want to be generous enough to avoid squabbles, but the enough must be on the lean side to control costs. Quality of service is not just a threshold but also a defined set of requirements with lower and upper boundaries. As with most new things, it is good to start with something desperately needed as a pilot, where the quality of service parameters may be simple and lenient.

### **Conclusion**

The efficiency of process reuse has been appreciated for centuries. It is the planet of technology that is being shifted, not the orb of business. This technological shift started with the development of databases that were accessed by multiple applications. With that development, governance structures had to be implemented to secure table-spaces for certain users. With service-oriented architectures, we have taken this approach from the client-server of the database structure to a more peer-to-peer approach. As the latest step towards congruence with business process, service-oriented architectures will allow business systems, and the applications that support them, to be more easily evolved, as long as governance is built in. Without proper governance, an enterprise runs the risk that the scourge of office politics will simply move to another dimension. No one wants that.



### ***About The Clipper Group, Inc.***

***The Clipper Group, Inc.***, is an independent consulting firm specializing in acquisition decisions and strategic advice regarding complex, enterprise-class information technologies. Our team of industry professionals averages more than 25 years of real-world experience. A team of staff consultants augments our capabilities, with significant experience across a broad spectrum of applications and environments.

- ***The Clipper Group can be reached at 781-235-0085 and found on the web at [www.clipper.com](http://www.clipper.com).***

### ***About the Author***

***Anne MacFarland is Director of Data Strategies and Information Solutions for The Clipper Group.*** Ms. MacFarland specializes in strategic business solutions offered by enterprise systems, software, and storage vendors, in trends in enterprise systems and networks, and in explaining these trends and the underlying technologies in simple business terms. She joined The Clipper Group after a long career in library systems, business archives, consulting, research, and freelance writing. Ms. MacFarland earned a Bachelor of Arts degree from Cornell University, where she was a College Scholar, and a Masters of Library Science from Southern Connecticut State University.

- ***Reach Anne MacFarland via e-mail at [Anne.MacFarland@clipper.com](mailto:Anne.MacFarland@clipper.com) or at 781-235-0085 Ext. 128. (Please dial “128” when you hear the automated attendant.)***

### ***Regarding Trademarks and Service Marks***

***The Clipper Group Navigator, The Clipper Group Explorer, The Clipper Group Observer, The Clipper Group Captain's Log, The Clipper Group Voyager, Clipper Notes, and “clipper.com”*** are trademarks of The Clipper Group, Inc., and the clipper ship drawings, *“Navigating Information Technology Horizons”*, and *“teraproductivity”* are service marks of The Clipper Group, Inc. The Clipper Group, Inc., reserves all rights regarding its trademarks and service marks. All other trademarks, etc., belong to their respective owners.

### ***Disclosure***

Officers and/or employees of The Clipper Group may own as individuals, directly or indirectly, shares in one or more companies discussed in this bulletin. Company policy prohibits any officer or employee from holding more than one percent of the outstanding shares of any company covered by The Clipper Group. The Clipper Group, Inc., has no such equity holdings.

### ***Regarding the Information in this Issue***

The Clipper Group believes the information included in this report to be accurate. Data has been received from a variety of sources, which we believe to be reliable, including manufacturers, distributors, or users of the products discussed herein. The Clipper Group, Inc., cannot be held responsible for any consequential damages resulting from the application of information or opinions contained in this report.