

**SUNGARD**

COMMON SERVICES ARCHITECTURE  
WHITE PAPER SERIES

# HOW FINANCIAL SERVICES ORGANIZATIONS CAN HELP ENSURE SUCCESS WITH SOA

*7 Lessons Learned Based on  
SunGard's Adoption of SOA*

---

## TABLE OF CONTENTS

---

- 2 Introduction
- 4 SunGard's Approach to SOA
- 6 7 Lessons Learned for Making SOA Work
- 12 Conclusion—What's next for CSA?

## INTRODUCTION

Like many of its financial services customers, SunGard has grown both organically and through a series of acquisitions. (In SunGard's case, more than 140). Such growth has had unique implications in managing a diverse legacy environment with very different development methodologies. In fact, the SunGard portfolio includes applications based on seven operating systems, written with more than eight programming languages, and using more than eight database management systems, six Web servers and multiple code repositories. The principal challenge in terms of development has been to unlock and leverage intellectual capital for economies of scale, while nurturing agility and focus.

The interoperability dilemma faced by SunGard is mirrored in the financial services industry, where the majority of IT budgets are now spent on developing and maintaining connectivity among specialized systems. The fragmented, heterogeneous environments commonly found in financial services organizations have been allowed to continue because of the culture and structure within the typical institution. Organizations are made up of a number of business units, each focused on its own products and services. And while the business unit is part of a larger organization, from a technology perspective, the business unit works independently to implement solutions that meet its own specific needs. Little thought is given to the ramifications of such decisions on the business as a whole.

Constrained by their own corporate cultures and fragmented IT infrastructures, financial services organizations are also faced with a variety of new challenges and outside influences fueling the demand for change. These include the need to:

- Meet more and increasingly stringent government regulations
- Collaborate more closely with other organizations
- Go to market quickly with new products and services
- Better manage risk
- Enhance data quality
- Increase efficiencies for a better competitive position

And to stand out in a crowded and fast-moving industry, institutions are forced to become more sophisticated in terms of marketing. Specifically, They need to leverage customer relationship management (CRM) techniques in order to more effectively grow their customer bases and improve retention efforts.

Meeting these challenges within the parameters of existing infrastructures is difficult and costly. As a result, many financial organizations are looking at service oriented architectures (SOAs) as a way to make their IT infrastructures better able to handle the ebb and flow of industry needs and customer demands. Plus, they are looking for ways to put an end to widespread duplication of costs and efforts and create a more efficient, uniform technology framework, from which the institution as a whole can benefit, as well as the individual business units.

For many organizations, an SOA represents a 180 degree turn in current development processes and a massive cultural shift. Therefore, the overarching key to success has more to do with people and processes than technology. Changing behavior and implementing controls to help ensure proper outcomes is by far the most important critical success factor. According to Jerry Silva, research director for the Tower Group, "The biggest challenge most banks face in deploying SOA comes from the architecture's inherent advantage of breaking down the walls between businesses in the organization, which therefore, requires a new organizational model and governance policy."

*Source: "The SOA Challenge: New People, New Models, New Ways of Thinking", Tower Group, September 2006.*

By year-end 2007, more than 75 percent of North American investment services firms will have adopted either SOA or SOA in combination with event-driven architecture (EDA) as the primary architectural approach to development of new applications (0.8 probability).

*Source: "Service-Oriented Architecture Takes Hold in the Investment Services Industry," Mary Knox, Gartner, 3 January 2006.*

## SUNGARD'S APPROACH TO SOA

Prompted by the needs of its customers, SunGard began searching for a way to connect its existing portfolio of more than 50 primary brands, not simply as a current stop-gap measure, but also to help ensure that future products will be developed with a built-in ability to integrate.

In 2002, a customer approached SunGard with a request to help integrate its portfolio of SunGard solutions. The company had a challenging technical environment with multiple legacy applications, different databases and duplicate data, the complexity of which was impacting both employee productivity and the customer experience. To help address the problem, the SunGard team focused on re-architecting the customer's development environment to help improve consistency, remove redundancy and increase efficiencies within the organization. SunGard tasked its own development teams with reconciling disparate development methodologies into one based on a services architecture and federated development model. The result was a single channel through which the company was able to provide multiple services, powered by a composite application of plug-and-play components that brought disparate functionality together in a cohesive, integrated way.

### What is CSA?

- >> An approach that helps eliminate redundancy—in both processes and architecture (human and technology) assets—and provides consistent functionality on a global scale
- >> A paradigm shift from programming to functionality assembly, with built-in integration
- >> A way to leverage existing IT investments, while moving organizations forward with new technologies

SunGard used this particular customer project as a springboard for a larger, company-wide integration initiative. It became the foundation for SunGard's Common Services Architecture (CSA), which is a type of service-oriented architecture upon which applications and common services are built. The difference between SunGard's approach and other SOAs that use Web services and platforms is that SunGard's is specifically designed to work with the wide variety of legacy and proprietary technologies commonly found in financial services organizations. In addition, CSA supports a broad range of middleware and will inherently complement an organization's existing SOA. SunGard's goal is to bring all of its products into the CSA, allowing customers to leverage a broad range of functionality and build composite applications specific to their needs in ultimate best-of-breed implementations.

## Composite Solutions: The "End Game" for Financial Services Organizations

Gartner, which is credited with coining the term service oriented architecture defines it as a style of software topology (architecture) that is modular, distributable and loosely coupled.<sup>1</sup> SunGard, focused on the "end game," or rather, what is achievable using SOA and, as a result, created its own Common Services Architecture.

An essential part of SunGard's strategy for synchronizing its software assets, the SunGard CSA is a service-oriented architecture, based on mainstream open standards. It enables discrete components from SunGard's product portfolio to be combined to form configurable, composite applications. As such, it represents an infrastructure for collaboration that allows global product development teams to share, contribute to and leverage each other's work. Other benefits include:

- >> Agility to react to market shifts and competitive threats
- >> Faster time to market for new products and services
- >> Ability to align IT with the business

## 7 LESSONS LEARNED FOR MAKING SOA WORK

With a focus on “getting it right,” CSA has been five years in the making. SunGard used a pragmatic, phased approach that began with 12 development teams and a sharp focus on building the support infrastructure. Today, there are more than 97 development teams contributing to the effort. Based on their experiences, the development teams at SunGard compiled a list of best practices and lessons learned while building CSA. These insights can be used to help financial services organizations as they begin evolving their own environments to an SOA, providing new perspectives on ways to help speed implementations and help improve project success rates.

### **#1 Lesson Learned: Governance is more important than technology.**

The very nature of SOA demands collaboration and well-defined processes that facilitate the sharing of ideas. Tower Group director Silva explains, “In these days of increased focus on risk and compliance with regulations in banking, governance must be one of the first priorities for institutions heading down the path to SOA.” The first step is to define the core team, which will be dedicated to the oversight of common requirements, standards and technologies. Then, decisions, such as where to store common code and documentation, how ideas will be shared, the process by which ideas will be adopted, etc., must be documented. Finally, agreement must be reached about how business and technical requirements will converge. These resolutions will form the basis for the governance process, which is vital to the effort’s success.

For its CSA initiative, SunGard created a model based on the methodology that has proven successful in the Open Source world. One Common Architect (technical) and one Common Product Manager (business) from each business unit make up the core team—each with voting power.

The team:

- Offers guidance and education
- Manages the common code base and infrastructure
- Provides support services
- Facilitates communication and collaboration
- Administers automation for builds, tests and deployments

Weekly global collaboration is essential and happens through a variety of avenues, including meetings, discussion forums and workgroups for proposal research.

***The bottom line: Ensure that every corner of the institution has representation and is governed. Moreover, create processes to ensure that the natural checks and balances are in place to keep the process whole.***

### **#2 Lesson Learned: Engage key stakeholders from each domain.**

Perhaps the most challenging aspect of an enterprise-wide initiative is identifying and gaining the support of key stakeholders and ensuring long-term participation. Because SOA is much more than a technology initiative, the business stakeholders are critical. And by definition an SOA is collaborative. Therefore, it’s important to get input and support from all key areas of the company, so that the organization is united, not only at the technical level, but also at the business level. These stakeholders contribute the “big picture” focus, helping ensure that the SOA continues to meet the overall needs of the institution and supports its goals.

Business stakeholders are also important, because they can allocate resources and ensure member participation. With the right stakeholders actively participating, an SOA initiative can more quickly gain momentum. And, it can help avoid common pitfalls, such as when a project runs out of steam or resources are diverted to other, more exciting initiatives. The right support can help keep the focus on meeting the needs of the business community. And to help ensure long-term support, it's critical to regularly measure and encourage participation by key individuals using objective metrics, such as attendance in collaborative discussions, training and workgroups. Tying participation to the compensation plan provides additional motivation.

***The bottom line: Identify, incent and measure participation of key stakeholders in order to help ensure that the SOA meets the needs of the business and the effort is sustained long-term.***

### **#3 Lesson Learned: Measure and report on key metrics.**

While the Open Source movement has relied on volunteerism, in a global company, measuring and quantifying results is important for all major initiatives. Without absolute quantifiable data, it's difficult to truly gauge the program's success and improve upon it. An SOA can, in fact, help deliver real bottom-line benefits as individual business units leverage reusable components.

Using a reporting tool, SunGard provides senior management with CSA adoption rates—showing how many components are being reused—and ultimately how much time and effort the company saves with an SOA. Specific examples from SunGard's early CSA projects show the time dedicated to integration dropping from multiple man-year completions to man-months; time to market reducing by 65% as a result of less duplicate code being written; and deployments costing \$1 million less than similar efforts.

Measuring individual contributions and publishing results is a natural way to leverage the basic human need for prestige and respect. Incentives will be critical to garnering long-term participation. However, in most cases, publicity and visibility regarding the individuals' efforts will provide the needed motivation. To continually improve upon its own efforts, the CSA team at SunGard routinely measures the following:

- Contributions
- Consumptions
- Participation in collaborative workgroups
- Responses to common communications
- Cost
- Revenue
- Time savings

***The bottom line: Measurement is needed in order to continually increase the value of an SOA initiative, help ensure participation by team members and garner support from executive management. As such, organizations must define areas that can and should be measured and put processes in place for regularly doing so.***

**#4 Lesson Learned: Inventory all assets in a directory or catalog.**

In today's global business environment, organizational resources rarely reside in one physical location. Therefore, when initiating an SOA project—where collaboration is the underlying framework—having a formal central repository allows intellectual property to be leveraged *no matter where it lives*. The most important repository is an asset catalog or component registry. Ideally, it should house decomposed elements of both legacy and new applications along with documentation, training modules, forum discussion threads, examples and support. Having the code and these resources in one location gives developers the ability to more rapidly create composite solutions. This registry then becomes a “marketplace” to which businesses can go for new functionality and guidance.

To leverage a registry, institutions must first deconstruct applications into defined services, interfaces and components. They must identify and segregate services of value within legacy applications. If functionality is needed that's not already included in the registry, this is the opportunity to add it and allow other groups to utilize it in their development efforts. To help ensure quality, however, the process for contributing to the registry must be clearly defined and carefully monitored. And, a process to ensure version control for the registered assets is critical. Finally, an SOA is only as good as its metadata, so measures should be put in place to ensure the accuracy and ongoing integrity of the registry's contents.

Financial services organizations need to incorporate legacy applications into ongoing development efforts, but many commercially available registries do not provide backward compatibility or support industry-specific protocols and interfaces, such as FIX. And, horizontal technologies, such as Web Services, address only some of what is being used in financial services organizations. So in order to leverage a registry that met its needs—and those of its customers—SunGard built its own. Initially it housed only components from the broad SunGard product portfolio. But today, it's expanding to include components from SunGard customers and partners for a more comprehensive offering.

***The bottom line: The right registry and processes for ongoing expansion and maintenance are critical.***

## #5 Lesson Learned: Adoption requires a maturity model.

Organizations with an SOA must change how they view applications. They need to begin by breaking them down into components of functionality and depositing these components in a registry. Moving forward, organizations will meet business goals by pulling the required pieces of functionality, or components, from the registry and assembling them to fulfill a specific need. Each new application, then, will be made from a variety of different components, each contributing a “piece of functionality” that combine to make the whole. Each component will need to follow the appropriate standards, based on the type of functionality it provides. How you identify components should be based on the following criteria:

- **What it is** – That is, a practical technical determination as to what the component physically is. For example, a non-visible, back-end, invocable service, would be defined as a WSDL, FIX or SWIFT interface. A visible, front-end, user-usable service would be defined as a GUI that follows some well-known standard.
- **What the market requires** – What types of functionality do users/customers need/want? Organizations may use formal techniques to gather this information, such as focus groups and customer surveys, or simply employ informal conversations.
- **With what can it be combined** – What solutions would require this piece of functionality? In the case of a broker, for example, it would naturally have to be understood that some of the important use cases of a rules engine would involve a trading component, or an asset allocation system, etc. In other words, what are the combined composite solutions that could possibly make use of each component?

SunGard used its own maturity model for the CSA, with which it identified components based on four levels:

- **Level 1** – *Rapid Service Orchestration*: Defined as encapsulation with various supported interface formats and protocols, this level is useful in those cases where a component can be registered as a system-consumable interface.
- **Level 2** – *User Interface Conformity*: This level streamlines user administration and provides a consistent user experience, while allowing easier deployment of additional services.
- **Level 3** – *Improved Processing*: This level is ideal for applications that require seamless data integration and lower data overhead, while benefiting from built-in access to CSA's reporting and rules engines. Within the financial services space, for example, highly shareable data appears in the following areas:
  - Normalized Structure for Financial Entities
  - Brokerage Account Model
  - Trust Account Model
  - Bank Account Model
  - Client Model
  - Transaction Model
  - Position/Lot Model
  - Security Master Model
- **Level 4** – *Architectural Guidance “Out-of-the-box”*: This level focuses on new applications, new modules for existing applications and R&D projects.

**The bottom line: A maturity model will help organizations properly classify each component and align it with the appropriate standards.**

## #6 Lesson Learned: Build integrity into the process and technology architecture.

For any large project to be successful, there must be faith throughout that the processes and technology architecture are right. As such, standards must be adhered to and there must be automated checks and balances built in to ensure that information and data going into the system are correct. For example, nobody wants to be the developer that checked in a line of code that broke the system, causing hours of additional work. Automated testing and peer reviews can help prevent these types of errors. As part of its CSA initiative, SunGard built integrity into its processes and technology through:

- Automated testing, including unit, functional and scale or stress tests
- Peer reviews
- Profiling
- Documentation
- Standards

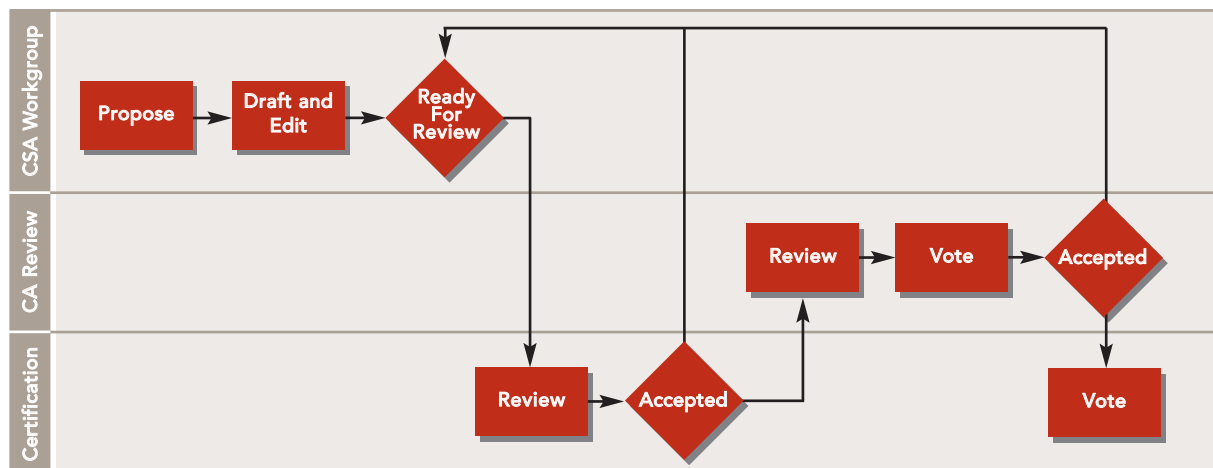
Ensuring quality is critical and it must be built into the governance process. With an SOA, quality means more than code that works. Developers must consider the ultimate composite solution.

What they contribute must not only work properly, but it must also deliver in four key areas:

- **Connectivity** – Can applications interact to form a composite solution?
- **Variability** – Can applications be customized in a consistent manner?
- **Life Cycle** – Can applications be upgraded in a predictable way?
- **Deployment** – Can applications be deployed and managed efficiently?

*The bottom line: Without integrity and quality assurance, key stakeholders will no longer have trust in the process and the SOA project will be in jeopardy.*

### Establishing Policies



**#7 Lesson Learned: Harvest the low-hanging fruit first.**

Executives and developers alike are weary of “big bang” projects that require years to implement and are costly and problematic. Often, it takes too long to realize the benefits. Excitement wanes and the project becomes more of a burden than a panacea. Therefore, the sooner an initiative can show positive results, the better.

It’s important to position the entire effort for success by choosing initial projects that will show the most value in the shortest timeframe. Where is the low-hanging fruit? The core SOA team should identify small projects already underway that require integration, meet real business needs and have a collaborative element to them. And, the team should also understand what efforts are already in process and redirect them into the SOA effort, shortening time-to-value. Also, initiatives with existing executive buy-in and support will most likely go smoother than those without. A good candidate for an initial project may be developing an interoperable user interface for multiple systems. SunGard began the CSA initiative by working on its customers’ integration needs. That effort provided proof that the approach was right and quantifiable results that helped secure and affirm executive buy-in.

***The bottom line: SOA initiatives will benefit greatly from early successes and the initial projects should have a short time-to-value.***

## CONCLUSION—WHAT'S NEXT FOR CSA?

When it comes to SOA, the development teams at SunGard have seen first-hand what works and what doesn't. They borrowed proven models from the Open Source movement to create a collaborative, vendor-agnostic environment for their own use. And now, the company is making its unique model available to the financial services community, helping institutions around the world transition to SOA. As a leading proponent of SOA within the financial services industry, SunGard has developed collaboration forums and the CSA Certification Center of Excellence to promote adoption and composite solution orchestration.

SunGard is also currently working with partners to include a wider range of product functionality into its registry, and soon will reach out to its customer community. Finally, SunGard is taking the next step in making its registry broadly accessible to the financial services industry with a software as a service (SaaS), on-demand delivery model. This approach will make hosted services available as a marketplace that allows components—from a variety of sources—to be joined to form new composite solutions.

Today, SunGard is at the forefront of SOA. With five years of practical experience developing its own SOA, SunGard is actively licensing CSA-based solutions to its customers, soliciting contributions from platform partners, hosting CSA-based solutions and helping customers build their own federated development environments. There's little doubt that SOA will play an important role in tomorrow's organizations. And, SunGard is working hard to make the transition as beneficial and seamless as possible.

**About SunGard**

With annual revenue of \$4 billion, SunGard is a global leader in software and processing solutions for financial services, higher education and the public sector. SunGard also helps information-dependent enterprises of all types to ensure the continuity of their business. SunGard serves more than 25,000 customers in more than 50 countries, including the world's 50 largest financial services companies. Visit SunGard at [www.sungard.com](http://www.sungard.com).

**Authors**

Darren Wesemann, chief technology officer, financial systems, SunGard  
[darren.wesemann@sungard.com](mailto:darren.wesemann@sungard.com)

**Managing Editor**

Pat McAnally, vice president, enterprise solutions marketing, SunGard,  
[pat.mcanally@sungard.com](mailto:pat.mcanally@sungard.com)

**References**

1. Hype Cycle for Application Integration and Platform Middleware, Gartner, 13 July 2006

[www.sungard.com/csa](http://www.sungard.com/csa)

For more information:  
1-800-825-2518  
[getinfo@sungard.com](mailto:getinfo@sungard.com)

© 2006 SunGard.

Trademark Information: SunGard and the SunGard logo are trademarks or registered trademarks of SunGard Data Systems Inc. or its subsidiaries in the U.S. and other countries. All other trade names are trademarks or registered trademarks of their respective holders.