

# Assuring Successful SOA Deployment with CoreFirst™

---

Solution White Paper  
January 2006

Copyright 2006 OpTier Ltd. All rights reserved. OpTier, CoreFirst, the tagline *Putting first things first*, the OpTier logo and all other OpTier product names and slogans are trademarks of OpTier Ltd. Other product names and/or slogans mentioned herein may be trademarks or registered trademarks of their respective companies. Specifications and product offerings are subject to change without notice.



## SOA drivers and momentum

In today's ever-changing marketplace, enterprises struggle to continuously change and adapt. Service Oriented Architecture (SOA) is designed to help enterprise IT support this business need more readily than ever before. The many definitions of SOA all encompass three core business values: flexible, changeable, continuously optimized business; rich, deep business connections; and strong control over your business. These are achieved through smaller reusable units of service that are easily combined to create new services, and by leveraging investment in existing functionality by packaging legacy (non-SOA) systems for integration with the SOA. The common benefits to the IT organization include increased reuse and lower TCO.

When using an SOA there is an exponentially growing number of interacting and changing parts to the service definition. When a user invokes a task or a service provided by the application, what we term a *transaction instance* is invoked, comprised of a complex chain of components potentially executing on multiple distributed computing tiers. SOA also mandates clear separation between the service and its implementation details. The growing number of moving parts and this distinct separation between the service and its implementation in production creates management complexity that makes the comprehensiveness and utility of management tools crucial to the success of an SOA.

SOA intends to create an environment of gradual change as opposed to the rip-and-replace attitude of previous IT architectures. It provides a flexible environment for the integration of legacy systems until they can be modularly replaced. Investments can be made on a step-by-step basis tied to individual business objectives and projects. This gradual implementation enables interest and trust in the SOA initiative to be established and ROI to accrue on an incremental basis. It also means that for the foreseeable future many enterprises will use a combination of SOA and legacy technologies together, further complicating the management challenge.

Global 2000 companies are faced with major changes, two of which are directly relevant to SOA: the rate of change, i.e., the volatility of markets, has increased substantially and will continue to do so; and businesses are increasingly faced with competition from global markets forcing them to become more active, agile, and dynamic, and requiring the same from the IT systems that support the business.

Though not essential to SOA, web services seem to be the enabling technology bringing SOA front and center on the agenda of global 2000 companies. And the trend is gaining momentum. Consider the following:

A November 2004, Forrester survey shows that large enterprises are overwhelmingly adopting service-oriented architecture (SOA): More than 70% of large enterprises report using SOA now. Beyond simply adopting SOA, 29% of large enterprises have an enterprise-level commitment to SOA, and 19% are using SOA for strategic business transformation.

A March 2005 survey of 100 CIOs by Smith Barney found that service-oriented architecture was their number one priority in the area of emerging technologies.



## SOA management

A plethora of development environments are geared toward developing IT services. Their focus has naturally moved to addressing the problem of SOA management.

An SOA environment adds a whole new dimension of complexity to the already difficult endeavor of application and systems management. With services being used by external users and applications, potentially acquired from external suppliers, and with a combination of applications that are both homegrown and packaged, the complexity is exponentially higher.

Organizations deploying or looking at deployment of SOAs may or may not be using web services and may or may not be using commercial management tools. But wherever they are on the adoption curve, the management issues that need to be addressed encompass the following areas:

**Monitoring.** The foundation of management, this includes the ability to define and monitor service capacity thresholds, faults, errors, and otherwise predictable and unpredictable events.

**Performance.** This includes the ability to manage service throughput and capacity as well as aggregating performance data and creating input for service-level agreement (SLA) reporting and for ensuring compliance with service level objectives (SLOs).

**Routing.** This includes the management of arbitration, fail-over, load balancing, dynamic routing, data and protocol transformations, and scheduling.

**Security.** Authentication, authorization, data encryption, and auditing.

**Catalog Support.** This includes the ability to manage web services as part of a discovery and integration capability including authentication, authorization, provisioning, and configuration services.

In this paper we will focus on addressing the Monitoring and Performance needs of enterprises.

## SOA monitoring & performance management challenges

By design, SOA monitoring, performance and routing management assumes that all the information necessary for the management of a service or instances of a service (i.e. transactions) are served up in the description of the service. The internal workings of the service, including any dependencies on system components and performance metrics are hidden by design. This separation obscures the transaction's invocation chain and presents the following management challenges:

**SLA Compliance is NOT Guaranteed.** Currently SLA compliance for SOA is supposed to be addressed by real-time service routing. However, routing offers a very partial solution because it is based on the unrealistic premise that sufficient separate containers are provisioned for each level of service and that contention within service containers is not an issue. Furthermore, as business priorities change, manual re-provisioning is required, SLAs are not guaranteed, and the solution's ROI is marginal at best.

**Visibility limited and lost beyond SOA borders.** Solutions based on SOA protocols share an inherent inability to monitor and manage a service as it crosses the SOA borders. Traditional management tools that look deep into SOA components or wide into legacy systems are missing the business context and therefore cannot address this gap.

**Risk of negative ROI.** SOA deployments face the risk of increased TCO and lower hardware utility as a result of crudely handling resource contention by load balancing at the service level combined with the inability to assure SLAs. These gaps lead to significant over-



provisioning and wasteful maintenance burdens diminishing the promised return on the investment in SOA.

Let's look at an Internet Banking example<sup>1</sup>. In this application the customer logs on and then browses details on different accounts. The sign-on process is time-sensitive because if the response is not timely and consistent, the customer will get bored and do something else (on a good day) or get annoyed and inquire with a competitor (on most other days)! Once the customer sees the balances, they are "hooked" and will stay to see additional account details, i.e., what posted last night that caused the balance to change? At this point the customer is already invested so they are more tolerant.

In this example there are two services: a sign-on service and an account details service. The sign-on service itself might actually invoke the account details service. But this is an implementation detail that by definition is obscured. Looking at the component level, eventually there will be database retrieval activities, querying account records. One customer may be browsing her account records while another is signing on. Traditionally, the database is set to offer the best overall performance to everybody and since the database reads have no business context associated with them, the browsing account queries will have the same priority as the sign-on account queries. This contention will not be visible to traditional SOA management tools and cannot be addressed by them.

## CoreFirst™ meets the challenges

CoreFirst™ provides an end-to-end view of every transaction (i.e., service invocation) across multiple tiers, down to the resource metrics at each tier. It also goes seamlessly beyond the SOA borders across legacy systems. CoreFirst can also actively allocate, in real-time, more of the service and legacy container resources to core business transactions, according to pre-assigned levels of priority. CoreFirst offers SOA initiatives the following benefits:

- **Assured SLA Compliance**  
CoreFirst automates the allocation of more resources to core business transactions, according to pre-assigned levels of priority, thus providing ongoing compliance with SLAs at the transaction level
- **Accelerated problem resolution and end-to-end visibility beyond SOA borders**  
CoreFirst accelerates problem area identification and resolution by maintaining the business context of transactions across and beyond the SOA borders, deep into the resource containers on the tiers and wide into legacy systems
- **Improved TCO and accelerated ROI for SOA deployment**  
CoreFirst lowers TCO, raises hardware utility and reduces resource contention by exposing and optimizing the obscured real-time deployment metrics of the service, components and legacy systems.

In the Internet Banking example above, CoreFirst provides a complete view of the invocation chain, between services, down to the database and across to legacy systems. When one service calls another, CoreFirst maintains the context for manual problem identification and resolution and the business priority for automated mitigation of contention issues. CoreFirst tracks the invocation path maintaining the context, i.e., CoreFirst knows which DB queries are for a sign-on and which are for browsing even though both queries may be accessing the exact same objects in the database. CoreFirst directs the service containers (such as J2EE

---

<sup>1</sup> As described by a Chief Architect at a Fortune 50 bank.



engines) and the databases to prioritize sign-on activities (including the DB queries) higher than other activities, virtually eliminating the occasional slowness during sign-on.

### **Assured SLA Compliance**

An SOA implies that services can be used by more than one application for more than one purpose. The definition of a service includes policies, which cover areas of security and performance and enable contracts between service providers and service users. A common contract would be for quality of service, i.e., a platinum member (service user) might require a response time of 1 second, a gold member a response time of 3 seconds, and all other users are guaranteed a response without a specific time restriction.

Existing management suites can see the user type or membership level of a service; however they can only access metrics that are exposed at the service level. The services by definition do not expose internal metrics because they are supposed to be stateless and without context. From an action perspective, existing tools can at best only allocate a new instance or reallocate an existing instance of the service to higher-level users. Traditional tools that can see contention at the component level - at the technical implementation layer below the service definition - have no indication of the business user context of the various contenders and therefore cannot offer any remedy.

Neither option provides a viable, cost-effective way to guarantee service at a required level.

CoreFirst maintains the business context down into the tiers where the components operate and across legacy systems. It views all necessary component metrics at each tier enabling CoreFirst to actively automate the allocation of more resources to core business transactions, according to pre-assigned levels of priority, thus providing ongoing compliance with SLAs at the transaction level.

### **Accelerated problem resolution and end-to-end visibility beyond SOA borders**

In a gradual transition from existing systems to SOA, parts of existing applications are often packaged (using “wrappers”) and integrated into the SOA without change. These ‘legacy’ applications, though being reused, may or may not be broken down into SOA manageable services. Though performance issues within these applications may not be new, the applications are now being used in quickly expanding and unpredicted ways in comparison to what they were originally planned for, exacerbating their performance issues and impacting service delivery. SOA management tools view only the exposed metrics up to the SOA border. Traditional tools looking at the services “from below” have no access to business context. From a management perspective, it is necessary to go beyond monitoring the performance of each service in the system and move to monitoring performance from a business viewpoint and to understanding the business impact of service disruptions. Using traditional tools to act on new performance issues is already known to be time-consuming and nowhere near as dynamic as SOA requires its solutions to be. Throwing more hardware at the problem and continuing the over-provisioning cycle exacerbates the TCO problem that SOA set out to resolve.

CoreFirst is unique in its ability to carry transactional business context across a diverse SOA and legacy architecture and look deep into the layers implementing the service on both SOA and non-SOA service delivery containers. CoreFirst therefore accelerates problem area identification and resolution by maintaining the business context beyond the SOA borders and deep into components and wide into legacy systems.

### **Improved TCO and Accelerated ROI for SOA Deployment.**

When using existing SOA management suites, they may provide integration services that include arbitration, fail-over, load balancing, dynamic routing, data transformations, protocol



conversions, and scheduling at the service level. However, the data available at the service level lacks the capability to expose the end-to-end business context and transaction flow - resulting in crude handling of resource contention causing increased TCO and lower hardware utility. Not having this context also implies that nothing can be done from a longer term capacity planning perspective to assign the right amount of resources to system components.

CoreFirst reduces resource contention by exposing the obscured operational metrics of the service, its SOA components and the legacy systems components from a business context. It then actively prioritizes execution of services and internal components when contention for resources occurs, according to defined business priorities and transaction-level SLAs. This goes far beyond the existing ability to aggregate data and report or load balance at the service level to lower TCO, raise hardware utility, and reduce resource contention. From a capacity planning perspective CoreFirst offers unprecedented depth of view into resource consumption metrics – at the transaction level – to allow for more effective capacity planning to be implemented – saving significant resources.



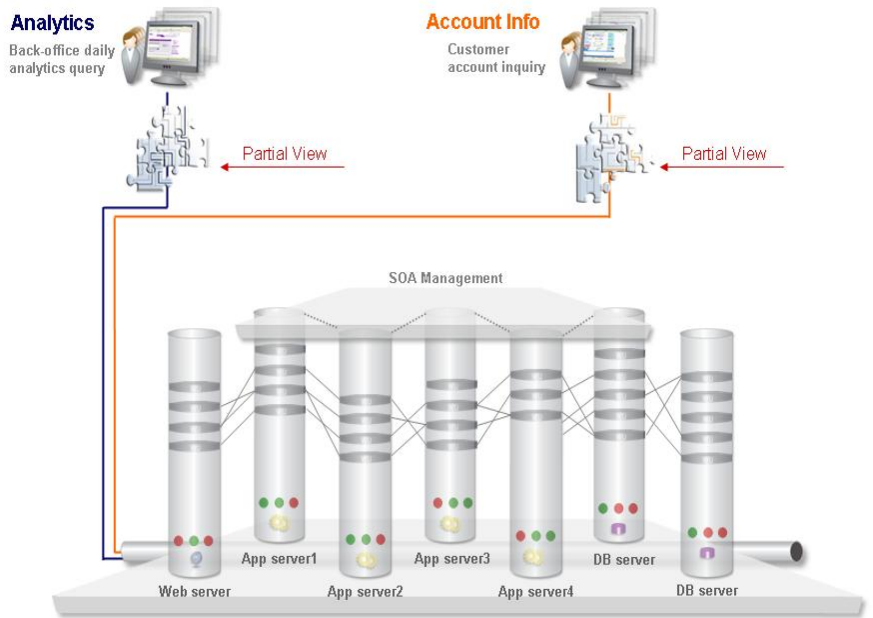
## Using CoreFirst™ to fill the gap in SOA management solutions

Business users have different priorities assigned to them for different business activities.

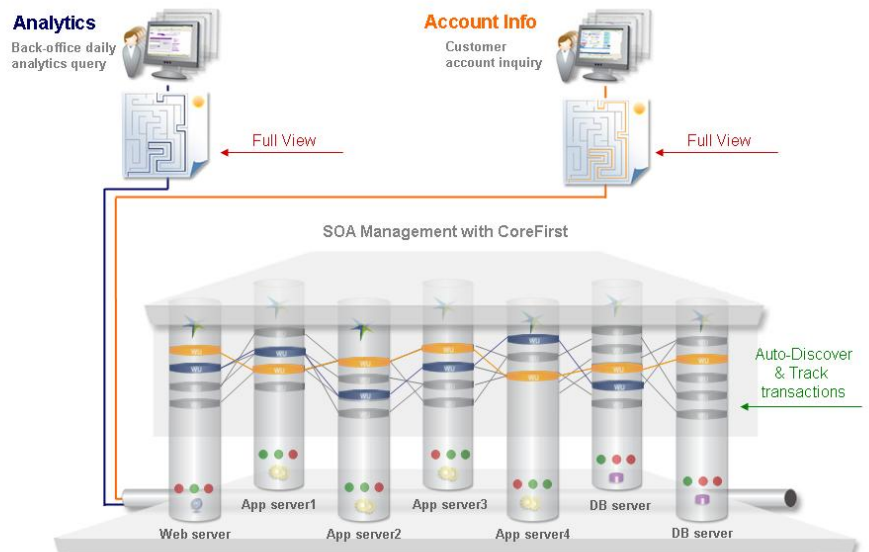
A user invokes an application creating a transaction.

Existing management tools can maintain the business context at the service level, providing a partial view.

Though component information is available to existing tools, it is devoid of business context.



**CoreFirst™** extends visibility of service delivery for problem identification and resolution and business impact analysis by maintaining the business context across and beyond the SOA borders, deep into components and wide into legacy systems. It also allows dynamic control of resource allocation schemes based on business priorities.



## Summary – CoreFirst Is For SOA

Successful deployment of SOA requires effective management of SOA. To effectively manage SOA from a quality of service as well as resource usage perspective requires a business-context-driven view of a service's components and how they function and interact among themselves in the SOA and across its borders to legacy environments. CoreFirst helps you get the most of your move to SOA by allowing you to see your business process performance, taking you beyond monitoring hardware and software application performance, and actively maintaining Service Level Agreements based on business priorities.

CoreFirst addresses the business need for flexibility and change by responding to real-time fluctuations in usage with real-time dynamic allocation of resources, assuring business priority services are met, regardless of the crunch on resources. CoreFirst can help identify new trends in usage and focus IT on problem areas by providing a cross-section view of the supporting tiers and software that caused a problem. This cross-section by business and transaction view is unique to CoreFirst.

CoreFirst helps quickly identify and resolve performance issues pre and post deployment, provides metrics for accurate planning of the architecture and capacity of the underlying systems, and increases the success of the deployment and operation of SOA based services.

CoreFirst provides the most effective SOA performance management capabilities on the market by exposing the business context of complex services and actively adapting the SOA in real-time to changing business needs.

## About OpTier

OpTier™ develops software that dynamically assures IT business service levels and optimizes resources for the enterprise. The leader in Transaction Workload Management, OpTier introduces a transactional context for IT service management.

Businesses today rely on IT transactions, made up of complex sets of components operating across multiple tiers and involving different stakeholders (IT infrastructure, applications and business). Visibility and control of these transactions are essential to the success of the business.

OpTier solutions automatically discover, track and profile all transactions – across all tiers – to clearly map IT business service topology at the transaction level. They improve collaboration among different stakeholders by providing them with a common transactional context and reduce the cost and time to resolve performance issues. They further maintain a fair share of resources and prioritize transactions based on business policies. Thus an organization can protect core services and users and deliver more from current IT infrastructure.

Established in 2002, OpTier is headquartered in New York, with sales offices throughout the US and in Europe. For further information, visit [www.optier.com](http://www.optier.com).

### Worldwide Corporate Headquarters

OpTier Inc., 130 West 42<sup>nd</sup> St., New York, NY 10036  
Tel: +1 (212) 679-2700, Fax: +1 (212) 679-4099, email: [info@optier.com](mailto:info@optier.com)

### EMEA Sales Headquarters

Foframe House, 35/37 Brent Street, London NW4 2EF, United Kingdom  
Tel: +44 (208) 201-5600, Fax: +44 (208) 201-5601

