

## **SOA Plug-and-Play Services**

Most large banks have a highly heterogeneous technology environment comprising several disparate applications running on a variety of platforms. With shifting economic conditions and rapidly evolving IT strategies along with mergers and acquisitions, few banks have had the appetite to untangle the morass of legacy systems running their businesses. But that is not holding up progress and innovation, thanks, in part, to the increased adoption of Web services and its conceptual cousin, the Service-Oriented Architecture (SOA).

### **Legacy Core Banking Solutions:**

To keep pace with rapidly evolving business and operational requirements along with changing customer demands, banks need to constantly upgrade their banking practices and processes. This is possible if banks regularly enhance their core systems and associated applications.

Most core banking applications at large financial institutions were developed almost two decades ago and have over time been enhanced to meet business needs, and individual applications have become complex. There are very few tools available for the outdated platforms on which the core banking applications were initially developed.

These complexities make enhancing core systems difficult, time consuming and costly.

As such a growing number of banks are considering replacing existing core systems with next-generation vendor solutions. However, replacing this complex web of applications with new core banking solution is not a straightforward task of merely switching off an old system and turning on the new one. Apart from the fundamental need to meet functional requirements, the data from old systems needs to be cleansed, transformed and then migrated to the new system. Processes driven by older applications too need to be changed and users need to be re-trained on the new application and processes.

Due to the siloed architecture within banks, where each business unit has their own systems and islands of information, core-banking replacement is a complex integration exercise. Islands of systems have to be either made redundant or integrated with the new solution based on business requirements and processes.

In such a scenario, the traditional approach of a 'big bang' replacement where all systems are replaced at one go is practically not feasible for large banks. Change management of such magnitude is not manageable and banks are evolving a phased approach to replacing legacy core banking solution. In the last few years some banks have attempted a Line of Business (LOB) - oriented migration approach. This involves the bank replacing only those modules of the incumbent core system that serve a specific line of business, like demand deposits, etc.. This approach enables the bank to experience the benefits of the new core banking solution rather early in the project. However, it also means that the new solution needs to integrate with existing modules of the incumbent core banking solution. This is where SOA can help the bank in easing the overall migration process.

### **Service-oriented Architecture**

SOA is neither a product nor a solution. It is an integration framework that binds internal and external services to create a solution. Just as a storage area network provides virtualized storage that is not dependent upon a specific storage device, and grid computing provides virtualized processing that is not dependent upon a specific computer, service-oriented architecture provides virtualized application functionality that is not dependent upon a specific block of computer code. With SOA, instead of focusing on different applications that reside on different computers, the emphasis is on business services that represent several different underlying applications.

Typically, in a non-technical context, a 'business service' is a logical unit of work that is carried out in

an enterprise and can be associated with internal operations or services offered to customers. However, from an application development perspective, 'services' has a different connotation and is used to typically define logical operations carried out by specific application components or programs. The 'service' concept is used to abstract the actual execution of the service from the requestor of the service. Hence there is a supplier consumer relationship between the requestor of the service and the supplier of the service. The architectural framework of web services brings out the usage of services in a commercial sense and in open space, essentially the Internet. The usage is based on standards defined by apex bodies like w3 and companies like Sun, Microsoft and IBM. Hence, when developers to develop applications adopted services architecture, the initial services adopted were primarily business-oriented services (like payments). Over the last few years, the scope of a 'service' has changed from business like services to execute a specific function like credit card payments to services over information architectures or networking services.

Through a web services interface, it is easy to bypass the underlying complexities and system dependencies of a given application (as in a point-to-point interface using APIs). Web services-based solutions deployed across the enterprise, especially banks, not only allow the developers to leverage plug-and-play logic across platforms quickly but also reduce the total cost of development. As more services become available, the ability to integrate with all other systems, services and companies is, in many cases, more challenging than the individual development of a particular service itself. Not only does the connection between two systems have to work as intended, but the various parties involved also have to negotiate Service-Level Agreements (SLAs), performance guarantees and contingency plans.

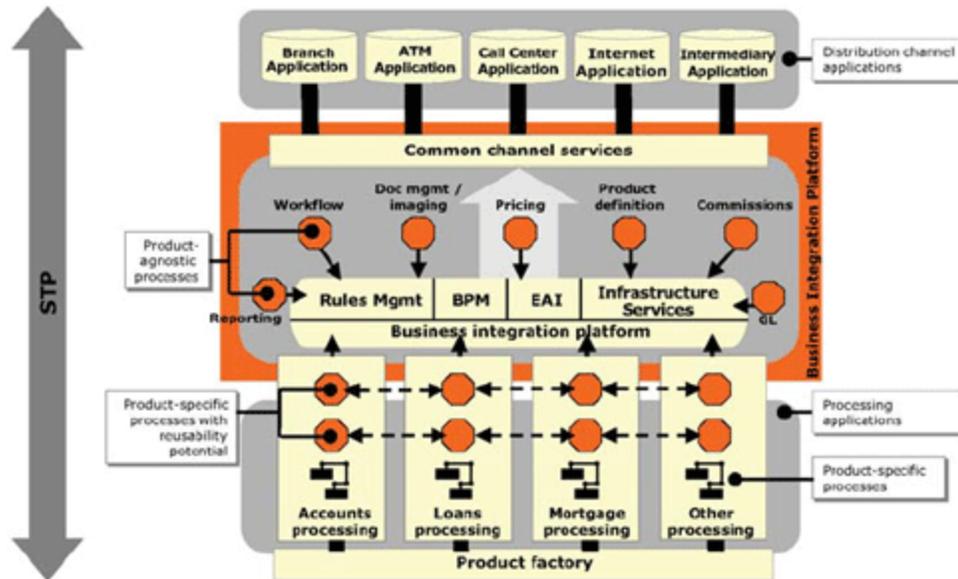
### **Adoption of Services and SOA**

A service-oriented architecture assumes an island of service identified by functionality and not by technology. It also includes process definition standards and can accommodate process changes that would take place during a core banking implementation. Since SOA is technology agnostic, not aligned to any specific technology platform, development language or technology tool, SOA can seamlessly be put into practice in existing IT environments. This quality of SOA ensures that changes in technology and processes during core banking replacements can be phased out and managed effectively. In an ideal SOA environment, services are published by applications and these services are offered for use based on various SLAs and standards-based interfaces. These services can be business services, like posting a transaction, to technical services, like data access and network communication.

In a core banking replacement project, where a bank adopts an LOB-oriented migration approach, SOA can help the bank identify business services related to specific lines of business, independent of other applications that can then be scheduled for replacements. The following diagram depicts a typical high-level banking solution framework. Here, services related to various business functions are provided by product-specific applications, whereas services independent of products such as customer relationship management or document management are provided by common infrastructure services. To achieve such a framework, a bank could start re-architecting its IT environment from any end and gradually break the monolith core system into parts.

Till the desired level of granularity is achieved, the bank could use an intermediate mechanism of 'wrapping' services around the current core to fit it into such a framework. The complexity of integration is reduced if the services are divided according to the product or category, since the number of interactions or interface points is minimal and the intermediate cost of integration is reduced. The plug-and-play nature of a SOA based solution is extremely attractive for financial institutions, and banks have cautiously started adopting web services and the vision behind services-oriented architecture. Most of the deployments thus far have been in non-core, peripheral areas such as distribution channel interfaces, or interfaces with external agencies like credit bureaus or online payment systems. Since the abstraction of business services within many legacy core systems does not exist, the adoption of SOA has been slow in legacy core systems replacement projects. Another reason for the slow adoption of SOA within banks has been the lack of standards for services internal to the organization. Standards are an important aspect of SOA, without which deployment is not possible and standards like IFX, FLX and BPEL are currently prevalent in this arena. Notwithstanding these issues, a few progressive banks

## BANKING SOLUTION FRAMEWORK: SOA ENABLES MAXIMUM AGILITY



Source: Datamonitor

have successfully adopted SOA for services external to the organization. Indeed, enabling banks to implement best-of-breed components from vendors or develop new functionality in-house without resorting to a wholesale replacement strategy, whether by open-source or proprietary tools, .NET or Java, SOA or point-to-point, at least one thing is clear, the plug-and-play benefits of SOA and Web services promises to increase the pace of innovation in financial services.

**Madhav Soundalgekar**  
Solution Architect, Finacle

### SOA Benefits -- In a Nutshell

By adopting SOA and process driven core-banking solutions, banks can achieve tremendous benefits.

- Banks can migrate functionality into a centralized middle tier environment, thus creating a centralized enterprise components' layer of business logic sitting in front of the core systems. This allows banks to remove duplication and leverage synergies across multiple product systems, thus maximizing cost savings, homogenizing operations and standardizing sales processes. The business components in this enterprise layer are product-agnostic (e.g. pricing, commissions, and workflow) and can be readily standardized across product systems with minimal customization.
- As process functionality is abstracted from the core systems, these systems can be progressively pushed back in the technology architecture, eventually functioning as pure processing engines.
- Banks earlier only had the choice of either wholesale replacement with a packaged solution or maintaining existing legacy systems. SOA provides an alternative to the complex and high risk 'big bang' replacement strategy by allowing banks to replace specific areas of functionality.
- SOA enables banks to achieve economies of scale through reuse of both technology and business components.
- Banks can overcome the problem of siloed business unit systems through the use of standardized, component based application logic.

SOA enhances flexibility and business agility so that changes can be made to individual components within the bank's technology infrastructure without major ramifications for the rest of the system.

"SOA is neither a product nor a solution. It is an integration framework that binds internal and external services to create a solution.