

WHITE PAPER



Creating Unique Financial Solutions from Existing Banking Applications

Application-Centered Building Blocks

Kirk Bailey and Jim Dempster
FIS

800.822.6758





Table of Contents

Table of Contents	1
Speed to Market: Keeping it Real.....	2
Pursuing the dream	2
Service-Oriented Architecture (SOA) Sets the Stage.....	2
Tying applications and business functions together	3
SOA Carried Forward: Application-Centered Building Blocks.....	4
Drawbacks of building blocks	5
Brokered Funds Accounting: A Case for Application-Centered Development.....	6



Speed to Market: Keeping it Real

Pursuing the dream

The American business mindset focuses keenly on finding faster, better – and in the best cases – less expensive ways to do the job.

The efficiency dream manifests daily throughout IT departments where developing new software, ensuring it performs as expected, and getting it to market can be a long, expensive proposition.

If you are a CIO or development manager in a bank that's under intense pressure to compete, you probably examine tactics reported to improve efficiency on a daily basis. Adding to the challenge is the specialized nature of the financial services market. Propositions to address the institution's technological needs are unique – understanding regulatory compliance considerations comes to mind immediately – a set of special circumstances that are difficult for software developers outside the industry to grasp quickly.

The high-tech industry has attempted to address the banking industry's need to speed product development. Accomplishing this means getting solutions to market faster, simpler and cheaper, not to mention in a more usable state. Differences between schools of thought can be subtle, and identities sometimes blur making comparative analysis confusing.

An alternative to programming fads and trends builds on the principles of service-oriented architecture (SOA). This paper will present the option of the *Application-Centered Building Block* approach. This approach advocates use of fully realized applications as components that can be placed together to form unique composite applications. These functionally capable systems are easily integrated and are quite scalable in anticipation of organizational growth. Among their benefits are process improvements, efficiency gains and a friendlier overall experience for bank employees and customers.

A working example of this development method is discussed in the case study at the end of this paper.

Service-Oriented Architecture (SOA) Sets the Stage

Service-Oriented Architecture (SOA) lays the foundation for application-centered building blocks. It does this by creating a nimble, easily adaptable scheme that helps guide and direct the placement of said building blocks to ensure they form end-user applications that fit the need. Without SOA, proposing a building blocks approach would be much less feasible.

SOA has gained prominence rapidly in the financial services and other business markets because of its increasing and evident ability to catalyze rapid, low-cost system development, improving total system quality along the way. The SOA software design framework is often delivered over the Web, making the services in question Web-based services. It separates functions into distinct units or services that developers make accessible over a network. Users, with enough technical skills, combine and reuse them in the production of business applications.¹ Services communicate by passing data to and from each another, or by coordinating an activity between two or more services, say, for a centralized, integrated organization-wide payment system.

¹ Thomas Erl, "Service-oriented Architecture: Concepts, Technology, and Design," Upper Saddle River: Prentice Hall PTR, 2005, no page cited.



When expanding upon the characteristics of SOA and its relationship to application-centered building blocks, viewing the software development process as a house-building process is more than apt. The architect develops the blueprint and calls out the materials. This “roadmap” is the role filled by the SOA software design.

Carpenters, plumbers, electricians, and heating/cooling specialists engineer the solution using the building blocks of construction that are at their disposal, except in our analogy these would be software applications. The software applications by themselves are functioning as a finished sub-system apart from its inclusion in a house. However, if there is no overall plan for how these separate components fit together, they are of diminished utility – hence the need for an overarching design, or in the case of technology, SOA principles that ensure everything fits together and functions as an integrated entity.

Opinion varies when one tries to pin down the identity of said SOA principles, but if one does enough research, a core group of particulars tends to surface again and again over time. These include:²

- Storing the “definitive” value of any piece of information in one place
- Separating functions into distinctive modular components
- Realizing business value through clearly defined functional components
- Making information easily accessible across platforms, systems and delivery channels

Tying applications and business functions together

Of course, SOA would be nothing more than a pleasing theory if it were unable to actually support different business processes within the organization. How does one set the stage to ensure that application building blocks can work together in a logical way that is transparent to end users?

According to the FIS paper, “Achieving Business Success: Why Architecture Matters,” the answer is through the use of business services. Business services can be thought of as composite applications designed to accomplish a comprehensive set of tasks necessary to complete a particular business function. Put another way, it is a set of standards governing how applications or components interface and work together to exchange information or perform specific functions. The diagram in Figure 1 depicts how composite applications can be designed to efficiently enable a business process.

² Troy Bradley and Mark Wall, “Achieving Business Success: Why Architecture Matters,” *FIS White Paper*, December 2008, p. 2

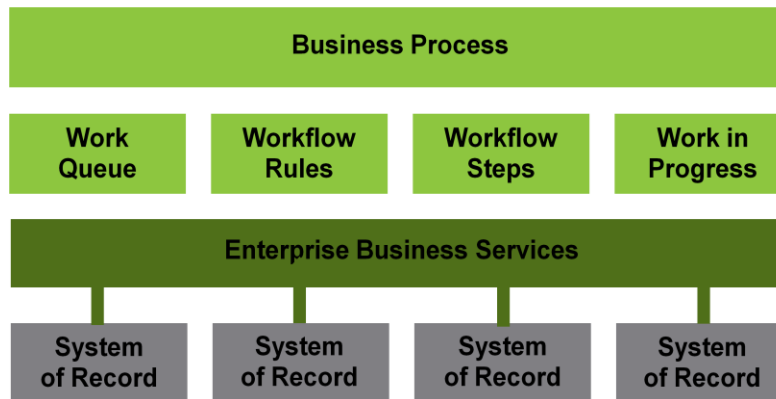


Figure 1: Combining composite applications in a design to enable a business process.

Because of its ability to tie applications and real-world business functions together, SOA has built an enthusiastic business following as of late, gaining favor in companies with outdated, often proprietary enterprise legacy systems. A good example of this trend is Nationwide Building Society, Britain's number-two mortgage lender and deposit taker. This groups plans to use a modular approach to product configuration that will end up addressing checking accounts, savings accounts, and mortgages, with an eye toward being able to configure them to their liking to get the end product they desire.³

As promising as the potential of SOA may be, in truth it only scratches the surface of what application-centered building blocks, a logical step forward from SOA, is capable of attaining.

SOA Carried Forward: Application-Centered Building Blocks

Service-Oriented Architecture is ground-breaking in that it creates the blueprinting and planning paradigm that makes it possible to combine and integrate self-contained business functions into an end application tailored to the needs of the new banking application. This is similar to how an architect's plans ensure the creation of a house tailored to the owner's requirements. SOA principles also set the stage for additional time-saving approaches to banking solution development.

This is where application-building blocks come in. They are the concrete components of construction that are placed together to make the vision of the architect a reality. As a house is composed of dozens, if not hundreds of functioning parts that come together distinctively to create a fully functioning structure, a technological system targeted to a specific banking need can be formed out of discrete but complete, fully realized applications engineered by specialists.

Application-centered building blocks are unique because the blocks in question are fully developed applications *already being used successfully*. They are brought together with only minimal modifications to create new composite applications, the best way to get to market quickly, reliably and with a minimum expense. Restricted timelines that all too often accompany high-priority projects leave little room for volume testing – making reliability

³ Steve Bills, "SAP Following New Core Strategy," *American Banker Online*, November 13, 2008, p.1



a potentially critical issue. However, this is not a problem with application building blocks that are subjected to their own rigorous testing before being called to become a component of the derived application.

Building new applications from existing ones is not a new concept. In fact, it is a time-tested approach that works especially well in banking where transaction volumes can grow quickly, which requires that application elements get as much “road-testing” as possible. Banking applications also must be regulatory compliant. Compliancy varies among many sources such as the electronic payment rules found in Reg E or the customer information required as part of the U.S. PATRIOT Act. Building from existing applications that already address regulatory requirements can save a tremendous number of development cycles.

Application-centered building blocks hold the following advantages over other rapid deployment methodologies:

- Transaction engines are proven and volume-tested
- Regulatory compliance is already taken into account
- Processing redundancies and documentation has already been established
- If so structured, compatibility with existing legacy systems will have already been proven, eliminating the need to replace them
- Not limited to Web-based applications, providing a great range of possibilities
- Is better suited to scalability and large organizations, in contrast to rapid programming methods which tend to lose effectiveness as the project grows larger
- Makes less stringent demands of the development team, because the applications are more mature and do not require as granular an approach to programming as agile programming

Since building blocks are viable commercial applications that have been used, in some cases, for years, volume testing is not an issue. Longevity breeds confidence in the managers being asked to use them. There even may be a chance that a manager has already used some of the building blocks being recombined to make the new application. Other issues that plague trendy development techniques are neutralized – it is easier to get the big picture for a new application when the modules themselves are fully developed. Bankers do not have to depend on new, untested Web services if they partner with a proven banking application developer carrying a full set of banking applications and services.

The best-of-breed concept also applies here, meaning banks can leverage applications designed specifically to excel in just one or a few functional specialties. Banks are free to select the absolute best solutions for the job, and have a strong systems integrator hook them together.

Planning is simplified because the development team is not at ground zero. Testing at a fairly advanced level has occurred for each unit. Fewer programmers have to be used.

Editing a book or paper is always a faster exercise than authoring a novel from scratch. Application development is no different than other creative processes – it is always more time consuming to start with the proverbial blank page.

Drawbacks of building blocks

The building block approach does have shortcomings that bank executives must be aware of and work to overcome when utilizing this methodology. These include:

- Existing bank applications may be built on older legacy platforms rather than on open systems
- The challenge to integrate existing applications requires a strong integration capability and middleware tools
- Training and documentation will require standardization for the new solution created



The following case study provides details that illuminate how a building block approach can add value in a real-life situation involving the financial services industry.

Brokered Funds Accounting: A Case for Application-Centered Development

For the last several years, deposit growth in the marketplace has proven a challenging prospect for financial services providers. Low-cost demand deposit accounts are disappearing as consumers look for better rates and consumer loyalty dissipates.

Brokered Funds Accounting (BFA) helps financial organizations grow their deposit base, while minimizing or promoting efficient utilization of reserve requirements. FIS Brokered Funds Accounting provides banks with the ability to offer high-dollar insured balance accounts (up to \$1 million or more), and manage allocation of these funds with multiple destination bank partners. This allows the financial institution to offer FDIC insurance on high-balance accounts, automate spreading of funds based on institution-defined criteria, and effectively manage reservable and investable funds.

Given the firm requirements and large transaction volumes of those financial services organizations that desire brokered funds solutions, FIS's technology architects created a unique approach to meet product development needs. They built a brokered funds solution from a series of banking applications – a real-life example of the building block approach to technological development. These applications needed to be volume tested, regulatory compliant, and flexible enough in design to accommodate brokerage firms, brokerage agents, banks, and end customers. As speed to market was critical, it became apparent that the more existing functionality that could be leveraged in the solution's development, the better.

The banking applications serving as the building block foundation for BFA consist of:

- A deposit system that provides interest calculations, transaction processing and back-end accounting
- A customer information system linking together and tracking sub-account relationships
- A business intelligence system creating management and operational reports
- A loan system creating billing information, tracking commitments and managing pricing
- Sub-account and sub-ledger accounting systems to allocate proper account balances and manage Reg D compliance
- An imaging system for report viewing



Figure 2 depicts the building block components FIS brought together for our BFA product:

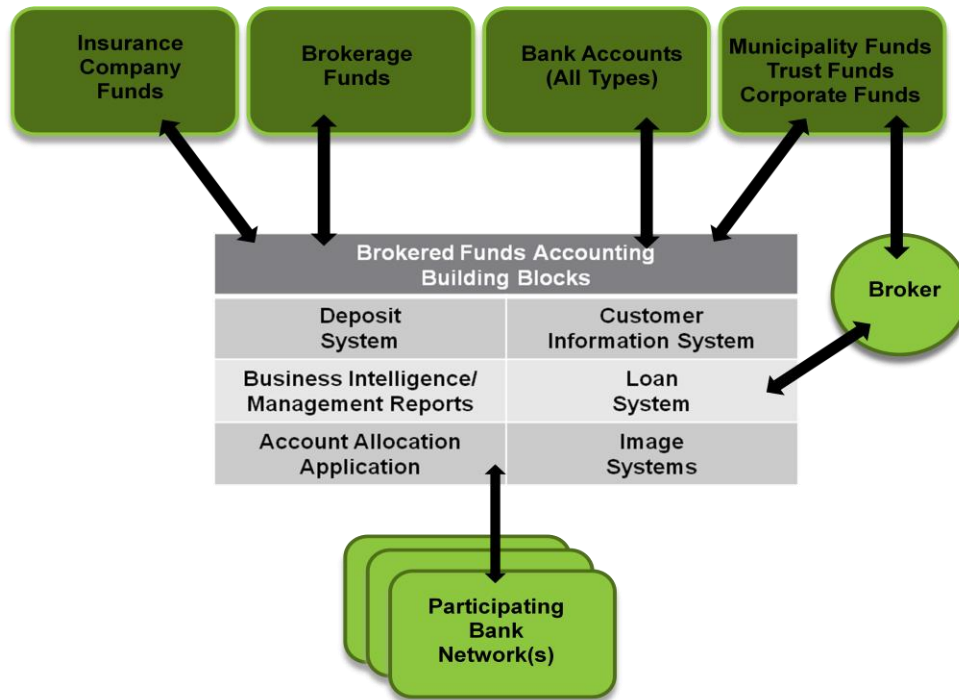


Figure 2: The building block approach as applied to Brokered Funds Accounting.

The features of the new solution formed out of this solid foundation became and remain impressive. BFA provides FDIC insurance for customer investments well beyond the current \$250,000 limit. Regulation D requirements are met. Banks can control rates and sweep thresholds by product or customer type. Financial services firms can leverage an extensive network of correspondent banks in order to maximize investment returns. Yet perhaps the proof point that attests most significantly to the value of this innovative application is that it worked right out of the gate. Transferring and reporting on the movement of millions of customer dollars on a daily basis required banking applications and expertise that were ready to go right off the shelf. The building block approach, with combined applications, delivered a flexible solution for:

- Daily processing, billing, and settlement
- Customer selection and rate setting
- Account acquisition and maintenance
- Placement of funds at a variety of banks
- Reg D compliance

The building blocks were smoothly integrated into a comprehensive solution, providing proven end-to-end capabilities with the expected workflow for automated processing. The resulting new application not only met the specific customer’s needs, but has served as a solid foundation for additional features as may be needed.