Service-Enabling Your Mainframe

Taking Evolutionary Approaches that Don’t Break the Bank
TABLE OF CONTENTS

1 EVOLUTION TO SOA AND THE REAL-TIME ENTERPRISE .................................................. 3
2 CONTINUUM OF OPTIONS TO SERVICE-ENABLE THE MAINFRAME ................................. 8
3 TIBCO MAINFRAME SERVICE SUITE ........................................................................ 11
4 CHOOSE THE MOST COST EFFECTIVE STRATEGY .................................................... 14
5 TIBCO – THE COMPLETE MAINFRAME SOLUTION PROVIDER ..................................... 16
6 ABOUT TIBCO ........................................................................................................ 16
The mainframe is an important strategic asset for many companies. Most mainframes have been in use for decades and are home to valuable legacy data. But the mainframe is not just a data store. In many industries, including financial services, government, healthcare, and insurance, mainframes are much more than data stores: they are strategic assets, used for complex, robust processing, and they are not going away.

From integration to process management to real-time business, companies are challenged to cost effectively incorporate their mainframes into their new applications and architecture approaches. There are a variety of solutions to better integrate your mainframe. From off-host solutions that work for simple data or transaction integration to best-of-breed on-host integration approaches, they all offer advantages and disadvantages. So, what is the best approach to integrate your mainframe assets into a flexible, dynamic service-oriented architecture (SOA)?

Which approach will provide the right balance of flexibility, adaptability, and total cost of ownership? How can a CIO successfully integrate or service-enable mainframe applications? How can companies build flexibility into their SOA strategies to accommodate somewhat static mainframe assets? What alternatives do organizations have to minimize cost and maximize flexibility? Is it possible to move beyond SOA to event-driven architectures (EDA) on the mainframe?

This paper will discuss the pros and cons of various approaches and introduce TIBCO’s market-leading solutions for mainframe integration. From simple data access integration to more complex service enablement, the TIBCO Mainframe Service Suite™ allows companies to start anywhere on the SOA continuum and migrate or service-enable their mainframe solutions as time and benefits dictate. By working with a single vendor, no work is lost from stage to stage.

1. Evolution to SOA and the Real-Time Enterprise

SERVICE-ORIENTED ARCHITECTURE
Enterprise IT is made up of applications and information sources that perform many functions and store countless pieces of information. These applications and information assets reside within different departments, have been built using different technologies, and run in different execution environments.
Service-oriented architecture (SOA) is an architectural paradigm for creating and managing business services that can access these functions, assets, and pieces of information with a common interface, regardless of the location or technical makeup of the function or piece of data. This interface must be agreed upon within the environment of systems that are expected to access or invoke that service.

SOA generally exhibits the following characteristics:

- **Loosely Coupled:** SOA enables independently developed service consumers and providers to interact regardless of technical makeup or location and enables consumers to identify and discover services they are interested in with minimal knowledge of who is consuming or providing the service for the exchange to take place.

- **Request/Reply:** SOA primarily supports interactions in which one specific system requests a piece of information or the execution of a function from one service provider, and the service provider subsequently sends a reply that provides the requested service.

- **Synchronous:** SOA primarily supports the synchronous invocation and execution of services. This means that when a consumer requests a piece of information or invokes a function, a connection between the two systems must be maintained until a response is received.

The primary value of SOA is that it enables the reuse of existing services and information, either standalone or as part of composite applications that perform more complex functions by orchestrating numerous services and pieces of information. For example, at a shipping and distribution company, a tracking service allows a retail customer to check on the status of the shipment. When coupled with other services, the tracking service allows retail customers to check on their shipment over the internet, by phone, or by walking into a distribution and receiving center. The same tracking service can be used by the shipping and distribution company’s business partners to provide tracking information to their own customers. Thus, a simple service is reused in different ways and combined with other services to perform a specific business function.

The ability to compose services that can be invoked on a stand-alone basis can dramatically reduce the cost and complexity of integrating incompatible applications. The ability of developers across an enterprise to quickly create, reconfigure, and repurpose composite applications enables IT departments to
satisfy new and changing business requirements with existing assets in a timely manner.

However, an SOA infrastructure does not address all the capabilities needed in a real-time enterprise architecture. It does not have the ability to monitor, filter, analyze, correlate, and respond in real time to events. In addition, an SOA cannot handle long-running processes or the guaranteed delivery of information across disconnected systems. These limitations are addressed with an event-driven architecture (EDA).

**EVENT-DRIVEN ARCHITECTURE**

As the pace of business increases, companies are being forced to more quickly sense and respond to changes, threats, and opportunities in the markets they serve, their supply chain, and their internal operations. Each of these changes, threats, and opportunities manifests itself as an event—a record of something that has happened. An event may be the placement of an order, a change in the price of some raw material, or the acknowledgement that an airplane will arrive 30 minutes later than expected. Events can be scheduled (e.g., a confirmation message will be sent every time an order leaves the warehouse) or unscheduled (e.g., bad weather forces an airplane to take a longer route around a storm and will therefore be late).

Consider the shipping and distribution company mentioned earlier. It cannot deliver a package if the address provided is incorrect. As a result, it needs to either contact the intended recipient to obtain the correct address or request an address correction from an internal system that might be offline for routine maintenance. If the correct address cannot be located in a reasonable timeframe, the package needs to be returned to the sender. Finally, information about the success or failure of this transaction needs to be transmitted to various departments for follow-up actions such as billing and reporting.

To address such needs, businesses require the ability to send, receive, and respond to unpredictable business information and events asynchronously. An EDA provides that ability and, combined with SOA, creates a real-time enterprise architecture that enables real-time business.

An EDA is an architectural paradigm based on using events as triggers that initiate the immediate delivery of a message that informs numerous recipients about the event so they can take appropriate action. The message is delivered using
a message bus, taking into account the specific requirements of the situation in terms of importance, urgency, and security.

EDA generally exhibits the following characteristics:

• **Asynchronous**: EDA primarily supports asynchronous interactions in which information is sent without the expectation of an immediate response or the requirement to maintain a live connection between the two systems while waiting for a response.

• **Publish/Subscribe**: EDA primarily supports many-to-many interactions in which systems publish information about an event to the network so that numerous other systems, which have subscribed and are authorized to receive such messages, can receive that information and act on it accordingly.

• **Decoupled**: EDA enables interactions between systems in which the publisher of a message does not know who the subscribers are and vice-versa – the interaction is entirely about the information being sent and received, not about a relationship between the two systems.

The primary value of EDA is that it allows companies to identify and respond to events that need to be addressed by one or more systems through events management. The events, collected via an EDA, can be analyzed and correlated to identify relevant patterns (or lack thereof) and then aggregated to build up information that is needed to solve the problems. With this process, companies can proactively address and respond to real-world scenarios in real time.

**Evolving Toward the Real-Time Enterprise**

TIBCO believes that SOA and EDA must be delivered within a unified architectural framework to achieve true business agility. Many vendors are marketing SOA solutions without addressing the notion of asynchronous messaging and events, but organizations need to be able to send, receive, and respond to real-time business information and events. There are many benefits to SOA, but it does not support decoupled asynchronous message exchange and therefore cannot support the delivery of real-time business information and events, nor the subsequent analysis and processing of these events – a requirement for almost every company today.

Here is one example that illustrates how a combination of events and services is necessary for a real-time enterprise architecture: A company’s stock splits two
for one. As a result, the depository trust company sends notification to every subscribing firm. The company must reliably transmit this event across all its front office, reconciliation, and back-office systems; the event may also need to be transmitted to partners and customers. Each system can then generate its own related events and perform services such as doubling the number of shares and halving the cost basis for the company’s shares being traded and held across the firm. The services can then be shared across different departments/systems or by a particular department/system.

Real-time enterprise architecture combines SOA and EDA to deliver an agile architecture that is the foundation for real-time business. A real-time enterprise architecture enables companies to create flexible, re-configurable, standards-based services as well as detect, monitor, filter, analyze, and correlate real-time events – both scheduled and unscheduled.

Mainframe systems can create a wide variety of challenges on the path to service enablement and the real-time enterprise.

- **Assets Unavailable** – Many mainframe systems have data or transactions that have not been created to be easily and directly called by other programs.

  - **Data resides in antiquated DBMSs** – For many companies, data resides in less prominent database management systems (e.g., CA-IDMS, CA-Datacom, Adabas, Model 204) that cannot be called from prominent open systems adapters.

  - **Screen-based transaction design** – Older transaction systems are screen oriented and business logic is embedded within screen interaction. As a result, it can be difficult to access the transactions and use their business logic as services or events.

  - **Complex data designs** – Embedded in COBOL copybooks within programs, there can be complicated “Redefines” statements that reveal multiple definitions for underlying data structures and record formats.

- **High volume, complex data extraction, cleansing, and enrichment** – Mainframe systems that are in place today typically support high volumes of transactions and/or large data sets. When a service requires large amounts of data to be extracted, cleansed, and enriched before the result set can be found, performance loads can significantly degrade performance if data is being
shipped back and forth from the mainframe to open systems environments for manipulation. Traditional off-host adapter strategies don’t work for these challenges.

2. Continuum of Options to Service-Enable the Mainframe

There are a wide variety of strategies that can be used to service-enable mainframe assets. The following diagram illustrates options on a continuum that is characterized primarily by the complexity of the service being created. At the left end, a new service may simply require access to mainframe data that is a well bounded and accessible asset. At the right side of the diagram are solutions that require the integration of multiple mainframe data sources, data scrubbing of large data sets, and, perhaps, some business logic to be executed against the result set to arrive at the answer. These approaches require different techniques to minimize total cost of ownership (TCO) while meeting performance and flexibility requirements.

Figure 1. Continuum of mainframe integration requirements

DATA ACCESS AND TRANSACTION ACCESS

Data access and transaction access solutions are best suited for scenarios that have three simple characteristics:

- **Accessibility** – The data required or the transactions being called are directly available through simple call routines without requiring complicated coding. The
information required for the off-host service is readily available through a simple call and return solution.

- **Completeness** – The data required or transactions required for the off-host service provide a complete answer as is. In this case, simple calls to the mainframe asset produce exactly the proper result. This also means that mainframe transactions are defined in such a way that they may be called directly, expect to be called by an “outside” application, and function entirely as-is without any unexpected errors or return codes.

- **Simplicity** – The data or transaction set to execute typically requires only one data set (or call pattern), and new transactions do not need to be created in the mainframe environment to create a complete solution and ensure referential integrity.

For these instances, a simple off-host adapter strategy typically provides the fastest time to benefit, meets the business needs, and provides the lowest TCO.

**COMPLEX, MULTI-SOURCE INTEGRATION**

Often times, customers’ mainframe environments have grown quite complex. Mergers and acquisitions have added new complexity into the technical environment, and economic pressures precluded re-writing acquired systems. Multiple mainframe solutions may have evolved to support dramatically different lines of business at a time when divisional autonomy and flexibility were more important than lowest cost of operation. Whatever the reason, IT executives may be faced with disparate customer or product data that must be integrated, scrubbed, and sorted to support business applications.

In this case, off-host adapter strategies are too complicated and will not scale to perform. Complexities are created by the need to write 3GL programs in the off-host environment that combine data from various sources, scrub data dynamically, and combine the data into a meaningful format. Perhaps another third party data scrubbing solution is needed to scrub the data. Then, the data has to be mapped into a meaningful format. If updates are needed back to the mainframe data sets, referential integrity and appropriate transaction scope have to be carefully designed and tested so the interface is not brittle. These complexities are better handled by a more robust solution. Further, if large volumes of data are being passed back and forth, performance suffers as the LAN/WAN connections are taxed beyond their capabilities. In real time, this just does not work.
A more robust, on-host integration tool is often required to support this situation. By using an on-host strategy, customers avoid shipping large amounts of data back and forth. The power of the mainframe processor is better leveraged to maximize real-time performance. Ideally, the tool would provide a near zero coding environment, leveraging table-based and graphical interfaces to integrate data from multiple sources.

**SOLUTION EXTENSION AND SOA**

Typically mainframe systems host not only valuable legacy data but also rich business logic accumulated over decades. Data-centric integration strategies do not provide the means to make use of the business logic on the mainframes.

A solution extension strategy provides a way to leverage business logic present on the mainframe and extend the life of mainframe assets. The functionality of the mainframe application can be enhanced by adding logic on either the mainframe or on the distributed system or both. However, given the increasing difficulty in finding mainframe programming talent, deploying enhancement in a distributed environment is more practical. An integration broker is required either on host or outside the host environment to manage the interaction between enhancement logic and parent logic. An integration broker that has a development environment to support the addition of processing logic is very desirable.

Graduating to SOA – at the far end of the mainframe integration continuum – requires creation of services from business logic available on the mainframe. The integration mechanism for accessing business logic is much more complex than it is for mainframe data access. The mechanism may heavily rely on the host environment’s transaction processing capabilities and inter-program communication interfaces. This requires transaction processing on the mainframe to be fast, reliable, and highly scalable.

 Successfully incorporating the mainframe into SOA requires an on-host solution that can provide robust transaction-processing capabilities. The solution must also provide the means for configuration management, governance, and task control. Finally, the solution must also have provision to interact with an orchestration
engine in order to compose applications using mainframe and non-mainframe-based services, and have the ability to support a high volume trigger mechanism to support the event-driven paradigm.

3. TIBCO Mainframe Service Suite

TIBCO supports the full spectrum of SOA approaches. The diagram below illustrates how the TIBCO Mainframe Service Suite™ covers the breadth of solutions required to service-enable your mainframe. The product suite consists of off-host TIBCO Adapters™ to support simple bidirectional data and transaction access, mainframe resident TIBCO® Object Service Broker to address complex multi-source integration scenarios, and TIBCO Substation ES™ to provide transaction-processing capabilities for cost effectively supporting very complex and high volume SOA requirements.

TIBCO ADAPTER FOR z/OS

For simple data access or transaction access solutions, customers merely use off-host adapters available from TIBCO. A service is created in the open systems environment that includes the appropriate calls to get mainframe data. Under this approach, the service is fully enabled without the need for any mainframe knowledge and without the need to modify any mainframe assets. If the mainframe assets are constructed in such a way that they are directly callable and return a useful set of information, then the TIBCO Adapters will meet the requirements with little to no impact on the existing mainframe applications. The adapters are configurable, so simple data access and transaction access designs can be implemented with a minimum of 3GL code.

TIBCO Adapter™ for z/OS is designed to meet even the most demanding performance and scalability requirements. First, it makes direct use of the interface...
products that are already provided by host vendor’s system software, minimizing overhead and maximizing performance. Second, in the open systems environment, the adapter takes advantage of clustering capabilities to scale as transaction volumes increase.

TIBCO OBJECT SERVICE BROKER

The TIBCO Object Service Broker provides capabilities to address more complicated mainframe integration scenarios that involve accessing data from multiple sources. Object Service Broker is a mainframe resident integration broker that acts as intermediary between mainframe and non-mainframe databases and applications.

Instead of relying on ODBC/JDBC to access data sources, and low level custom code to access legacy applications that result in overhead that reduces functionality and degrade performance, Object Service Broker takes a different approach. It integrates with legacy data sources through a set of native database adapters. The on-host strategy leverages the power of the mainframe to process complicated transactions and return the minimum accurate result set to the open systems environment. This provides high performance access to a wide variety of data sources including sequential files and databases such as Adabas, IMS, DB2, CA-IDMS, CA-Datacom, Oracle, and others.

Sometimes, additional logic is required on host during the integration exercise. This might be required to integrate two separate data sets or to add processing logic that is needed but does not exist in the current mainframe applications. Object Service Broker allows processing logic to be added to result sets on host through a simple table driven metaphor. Programmers do not need to learn mainframe tool sets to add processing logic that uses the power of the mainframe environment.

Further, using the Object Service Broker Integration Gateway product, which includes and XML parser, legacy data structures can be mapped directly to an XML document without the need to write any code. Similarly, incoming XML documents can automatically populate mainframe data structures without the need for complex coding on the mainframe.

Object Service Broker’s integrated application development capabilities allow companies to consider mainframe-based orchestration. This is an especially attractive proposition when the majority of assets to be integrated and
orchestrated reside on the mainframe and the scenario doesn’t call for high transaction processing performance.

**TIBCO SUBSTATION ES**

More challenging initiatives, such as utilizing the legacy business logic in an SOA, require transaction processing on the mainframe to be fast, reliable, and highly scalable. TIBCO Substation ES provides this level of processing. Substation ES is a mainframe-resident server that communicates simultaneously with multiple transaction regions of the host.

Substation ES supports interfaces in two primary mainframe transaction-processing environments: CICS and IMS. The CICS interface exploits External CICS Call Interface (EXCI) and provides for multiple methods of invocation such as Call by Program Name, Call by Transaction Code, and Transfer Control (XCTL). In addition, the CICS interface uses Transient Data Queueing facilities of CICS to support “triggers.” Trigger transactions provide a means of publishing events and data-based events within the domain of the CICS region to promote EDA. Similarly, the IMS interface embeds a full featured implementation of the IBM Open Transaction Manager Access (OTMA) architecture. It supports recoverable and non-recoverable
transactions in addition to a very high volume trigger mechanism capable of sustaining throughput for EDA and complex event processing (CEP).

Organizations can capture events and route them as required to mainframe-resident applications, to applications on distributed platforms, or to TIBCO BusinessEvents™ CEP software, depending on business requirements.

Substation ES supports asynchronous processing and has bidirectional capabilities. A standards-based implementation provides the mechanism for mainframe transactions to participate in loosely coupled SOA deployments. Common services within Substation ES provide for task control, dispatching, interface and configuration management, data transformation and logging, tracing, and audit facilities.

**TIBCO ACTIVEMATRIX BUSINESSWORKS ADVANTAGE**

TIBCO Mainframe Service Suite is an integral part of the TIBCO SOA product family that also includes TIBCO ActiveMatrix BusinessWorks™, a standards-based web services platform used to connect disparate applications and data with little to no programming. It provides an integrated services environment (ISE) for creating web services and orchestrating process flows to improve the consistency and adaptability of both IT and business operations. This platform plays an integral role in the deployment of an SOA.

Organizations, irrespective of their place on the mainframe integration continuum, can tap into the rich functionality of ActiveMatrix BusinessWorks by virtue of its association with components of the TIBCO Mainframe Service Suite, namely, Adapters, Object Service Broker, and Substation ES. To illustrate, users can select Adapter services from a graphical palette of resources available in ActiveMatrix Business activities, greatly enhancing developer productivity and ensuring consistency across the organization.

**4. Choose the Most Cost Effective Strategy**

TIBCO provides a full range of mainframe integration solutions for the simplest to the most complex environments, allowing organizations to select a strategy that meets their business needs, optimizes technical performance, and provides lowest total cost of ownership. The diagram below illustrates typical trade-offs that CIOs
Some IT organizations are aggressively migrating all applications off of the host. For those organizations, the TIBCO Adapter strategy can be pursued for all but the highest volume large-result-set applications. To minimize costs to scale the adapter strategy, it is necessary to carefully analyze the cost of adding more processors off host versus using the Object Service Broker as a lower cost migration tool. If transaction volumes are not excessive but mainframe integration needs are very complicated, it becomes necessary to consider carefully the trade-off of using the on-host strategy as a highly productive alternative that avoids expensive 3GL development and maintenance that may be required to create the required service through an off-host adapter strategy. In cases where there is a mix of simple data access needs and more complex integration, TIBCO Mainframe Service Suite offers the flexibility to pick the combination of components that best suit the integration scenario. TIBCO Mainframe Service Suite allows decision-makers to choose the pace at which they service-enable mainframes and also gives them the means to balance costs.
5. TIBCO – The Complete Mainframe Solution Provider

The mainframe is one of the most challenging assets to integrate into a modern enterprise architecture because of its longevity in the enterprise and the breadth of technologies involved. Longevity challenges can include size and complexity of code base, lack of documentation and staff knowledge, and the extent of one-off, point-to-point interfaces. The state of the art for mainframe application development has changed dramatically over the years. The variety of technologies brings its own set of challenges. Today's mainframe applications are typically a combination of transaction system, data stores, and best practices from the last 20 years.

TIBCO understands that mainframe integration, SOA, and EDA are not a "one size fits all" domain. TIBCO offers the widest product suite for addressing all parts of the integration spectrum. With services and support to back up its products, TIBCO is your one-stop mainframe integration partner.

6. About TIBCO

TIBCO Software Inc. (NASDAQ: TIBX) is a provider of infrastructure software for companies to use on-premise or as part of cloud computing environments. Whether it’s optimizing claims, processing trades, cross-selling products based on real-time customer behavior, or averting a crisis before it happens, TIBCO provides companies the two-second advantage™ – the ability to capture the right information at the right time and act on it preemptively for a competitive advantage. More than 4,000 customers worldwide rely on TIBCO to manage information, decisions, processes and applications in real time. Learn more at www.tibco.com