zapthink white paper

TRANSFORMATION SERVICES FOR SOA

INCREASING BUSINESS AGILITY WITH TRANSFORMATION SERVICES

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

TRANSFORMATION SERVICES FOR SOA

INCREASING BUSINESS AGILITY WITH TRANSFORMATION SERVICES

September 2006

Analysts :John Reynolds and Jason Bloomberg

Abstract

One of the age-old problems of application integration has been the mismatch between the data representation that one application provides and the data representation that another application expects. This data integration problem will limit the loose coupling between Service providers and consumers in Service-Oriented Architecture (SOA) implementations unless there is an effective way to decouple the provider and consumer's data representations.

Transformation Services convert data from one format to another, and decouple the data formats of Service providers and consumers, freeing the application developer to focus on functionality of the business process, rather than deal with possibly complex transformation logic.

Transformation Services also offload CPU-intensive transformations onto dedicated servers, and scale as necessary. Centralization of transformation logic and metadata also provides better opportunities for maintenance and monitoring of transformation activity and performance over time.

Xenos offers enterprises the transformation solutions they need to leverage data from any application, running on any system, regardless of original data format. In the context of a properly architected SOA, these solutions help to deliver on the promise of unimpeded data flow between applications.

All Contents Copyright © 2006 ZapThink, LLC. All rights reserved. The information contained herein has been obtained from sources believed to be reliable. ZapThink disclaims all warranties as to the accuracy, completeness or adequacy of such information. ZapThink shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.

I. The Business Drivers of Agility and User Empowerment

A wide range of organizations, including enterprises, government agencies, and even small and midsize firms are adopting Service-Oriented Architecture (SOA). This broad commitment to architectural change indicates that organizations across a wide spectrum of industries and geographies are starting to realize the basic business benefits of SOA: improved business agility, reduced integration expense, greater reuse of IT assets, and increased visibility into processes and information.

Yet, as organizations move forward with their SOA implementation plans, they often find themselves grappling with significant, long-lived problems of data and semantic integration among a wide variety of data sources. These data sources might be structured, as in the case of databases and enterprise applications, or semi- or unstructured such as Web pages, PDF documents, Office files, email, media content, print streams, or a wide variety of data feeds and formats. The need to access information of so many disparate types from so many disparate sources forms the semantic integration challenge that organizations must deal with today.

The need to access information of many disparate types from many disparate sources forms the semantic integration challenge that organizations must deal with today.

Companies frequently make independent decisions about technologies for different aspects of their business, and then concern themselves with tying together these disparate systems at a later date. In addition, many industries are prone to acquisition and merger activity, leaving their companies a complex checkerboard of differing technology implementations. As a result, there is rarely central decision making on technology adoption, and heterogeneity always rules the business that adapts to meet its changing market. Thus, companies continually find themselves with the challenge of meeting their data and application integration requirements in a heterogeneous environment of continuous change, which is precisely the information integration challenge facing enterprises today.

Semantic integration means going beyond integrating systems so that they can talk to another, to the point where the systems can *understand* each other. What makes semantic integration a challenge is two-fold: first, the representation of information and the information itself are often bound tightly together; and second, that information frequently lacks context. Developers often think not of the data themselves but rather the structure of those data: schemas, data types, relational database constructs, file formats, and so forth—structures that don't pertain directly to the information at hand, but rather our assumption of what the data should look like. In tightly-coupled architectures, data structures are absolutely necessary, since they provide systems a way of coping with the information they receive.

TAKE CREDIT FOR READING ZAPTHINK RESEARCH!

Thank you for reading ZapThink research! ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

Earn rewards for reading ZapThink research! Visit www.zapthink.com/credit and enter the code **XTRANS.** We'll reward you with ZapCredits that you can use to obtain free research, ZapGear, and more! For more information about ZapThink products and services, please call us at +1-781-207-0203, or drop us an email at info@zapthink.com.

How systems store and represent information interferes with the meaning of that information. To be more precise, the meaning of information and the structure of that information aren't one and the same. For example, "September 6th, 2006" is a date for sure, but whether or not it be a string, date type, or integer shouldn't matter. Yet, developers often needlessly combine the structure and meaning together inextricably. Furthermore, there isn't enough context in the structure to understand if the date is a birth date, a purchase order date, or any other date.

The Need for Seamless Service Composition

A SOA implementation's success directly relates to how quickly we can modify our Service-oriented applications to implement changes required by the business.

We can judge the relative success of a SOA implementation in many ways, but perhaps the most important is in terms of business agility: its success directly relates to how quickly we can modify our Service-oriented applications to implement changes required by the business. In many cases, the difficulty of solving the semantic integration issues limits a company's ability to respond to change quickly. While Service contracts and Service interfaces serve to dramatically reduce the barriers to application integration, exchanging information between Services and aggregating it together with other information continues to be a significant and costly challenge for most companies. Overcoming these challenges is crucial to realizing SOA's promise of continuously evolving, composite applications consisting of Services that improve business agility.

Such *Service-Oriented Business Applications* (SOBAs) are composite applications that support a company's business processes consisting not only of business logic, but also their underlying information resources. SOBAs expose business functionality and information from varied information sources, and they also include orchestration logic that ties the Services together to accomplish a higher-level task or transaction. This orchestration logic, encoded in metadata such as the Business Process Execution Language (BPEL), controls the flow of the application, passing information to Services and deciding any further course of action based on information those Services might return.

Creating a SOBA begins by defining the logic steps of the process you want to execute. The next step is to locate Services that can perform the individual functions and tasks that make up the process. If the existing library of Services is missing any of the functionality that the process needs, then it might be necessary to create new Services to support the application. The final step is to write the orchestration logic that assembles the individual Services into a cohesive application.

Semantic Integration and Loose Coupling

The business agility benefit of SOA depends upon developing continually evolving SOBAs that rely on the loose coupling of individual Services and their ease of composition. Adherence to standards insures, to some degree, the loose coupling and composition of Services and data exchange mechanisms, but there is no guarantee that the data exchanged with an individual Service will be universally comprehensible. Even when you use a metadata standard such as XML to format messages, the schema of one Service might not match the schema of another, and fields with the same meaning might use differing tags.

Dealing with semantic integration issues complicates our ability to incorporate Services into SOBAs in a way that provides business agility.

Dealing with these semantic integration issues complicates our ability to incorporate Services into SOBAs in a way that provides any business agility. To utilize a Service, orchestration logic must often extract relevant information from one data representation and transform it into another. Building and managing such data transformation logic is costly and time consuming, and serves to make the composition logic more tightly coupled and harder to modify in the future.

Copyright © 2006, ZapThink, LLC

Seamless data flow between the Services within a SOBA requires a better approach to dealing with composite data integration issues.

In order for data to flow unimpeded in a SOA implementation, therefore, we must liberate Service providers and consumers from each other's data formats. Services should be concerned with the semantic meaning of the data, not the structure of the data. The issue here is therefore one of loose coupling. While we might loosely couple application interfaces, if we deal with data the same way that we have always done—by imposing the data structures of Service providers on Service consumers, the result is every bit as tightly coupled as previous architectural approaches. In order to deliver on the promise of seamless data integration, we must transcend loosely coupling the application interface and in addition provide loose coupling at the semantic level.

In order to deliver on the promise of seamless data integration, we must provide loose coupling at the semantic level.

The Need for Right-Time Integration in a Heterogeneous Environment

Another part of making SOA a success is enabling companies to access critical, business-relevant information in a timely manner. Enterprises must often cobble together information of many disparate types from many disparate sources forms in order to make sound business decisions. However, many times this information arrives too late to actually enable the business to make those decisions in an agile fashion.

In addition, certain business operations, such as fulfilling customer purchases, billing operations, and real-time operations such as stock trading and online banking require instantaneous access to information distributed throughout the organization in a wide array of heterogeneous data sources. In order to meet their own needs, companies have resorted to making redundant copies of information just so that it is available to those critical systems when people need them. The end result is a tangled web of redundant information, dispersed throughout the organization and continually out of synch.

In addition to the requirements for the internal exchange of information, companies and government institutions frequently exchange information with external third parties, such as suppliers, partners, businesses, citizens, non-governmental organizations, and others. These interactions are complicated by the fact that it is difficult to predict the nature of the information these organizations share. While supply chain partners may trade important data relating to parts, customers, logistics, and purchases, the format of this information varies on a partner-by-partner basis, as do the means for exchanging this information. As a result, in most cases, many-to-many, business-to-business integration is a moving target, especially in a continually changing environment of heterogeneity.

II. Data Transformation Services

As we explained in the section above, the key to overcoming data integration problems in a SOBA lies in the use of data transformation Services to liberate Service providers and consumers from having to deal with data formatting issues. Transformation Services convert data from one format to another. If semantic differences exist between Service consumers and providers, some additional work needs to be done in order to overcome those differences and provide the veneer of loose coupling required in SOA. A Service-oriented approach to this problem is to implement transformation Services that act as intermediaries between consumers and providers, freeing the provider and requester to focus on the semantic meaning of the data rather than on the format of those data.

Copyright © 2006, ZapThink, LLC

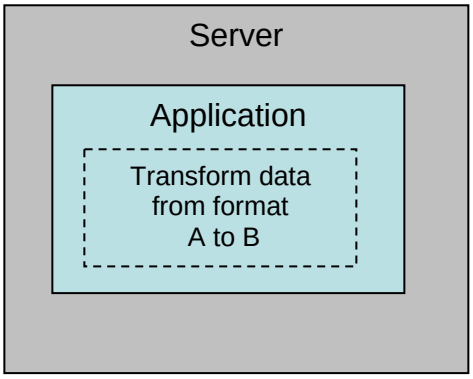
The use of data transformation Services improves loose coupling by separating the concerns of Service providers and consumers.

The use of data transformation Services improves loose coupling by separating the concerns of Service providers and consumers. The builder of a specific Service can focus on the key functionality of that Service, and the composer of a SOBA that implements a Service-oriented process can focus on process-level design concerns. The transformation Services isolate the data transformation concerns, allowing personnel skilled in transformation techniques to focus on developing the data transformations. This approach, called separation of concerns, leads to more maintainable solutions. Separation of concerns allows organizations to focus each component on what it does best, and enables the organization to utilize its resources more effectively.

The Problem of Embedded Transformation Logic

When an application or Service must transform data from one format to another in order to achieve some desired data integration goal, the most common and straightforward approach is to embed data transformation logic directly within the application, as illustrated in Figure 1 below:

Figure 1: Using an Embedded Transformation



The application internally transforms data from format A to format B

Source: ZapThink

Embedding transformation logic within applications is a natural object-oriented design technique, because in the object-oriented world, applications encapsulate internal logic, including transformation logic. Nevertheless, there are significant disadvantages to embedding transformation logic within an application:

- Other applications or Services have no practical way of reusing the embedded transformations
- It's impossible to offload the CPU overhead that the embedded data transformations cause
- The transformation logic is tightly coupled to the business logic within the application, reducing the flexibility and reusability of the applications
- The transformation logic is handled programmatically vs. declaratively through metadata, further impeding the flexibility of the application.

Copyright © 2006, ZapThink, LLC

While in many cases, data transformation logic may be so specific that it is unlikely that another Service or application will need it, and so well-built that it doesn't consume many processing resources, the real problem of embedded transformation logic is that over time, companies end up with so much purpose-built, hard-coded, and tightly-coupled transformation logic scattered across the enterprise.

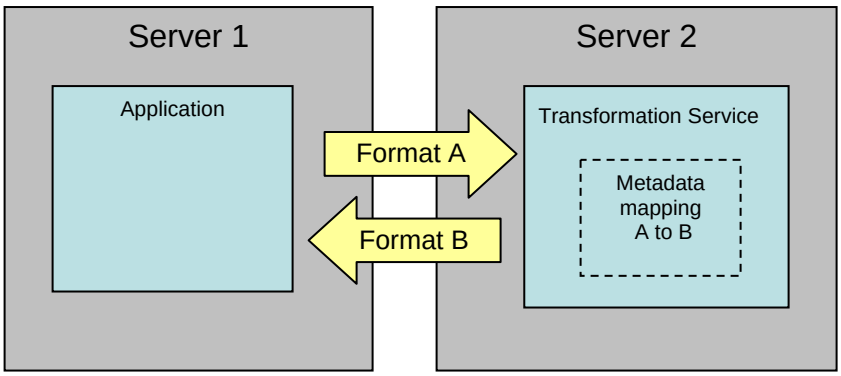
In addition, the data transformation activities that a specific application utilizes may only be applicable to one composite application scenario, but the cumulative overhead of transformations within multiple applications could have an adverse effect on the performance of other composite applications. Also note that if multiple applications within an enterprise transform an external data format into an internal data format, overall performance might improve by transforming the external data once and then using the normalized format to exchange data between the Services. This scenario points out the danger of hiding transformations from the enterprise: if we hide a transformation, there is no way to gauge its true cost.

If we hide a transformation, there is no way to gauge its true cost.

Transformation Services within Composite Applications

An alternative to embedding transformation logic within an application is to use an external transformation Service to handle the task. A transformation Service is a specialized Service that accepts incoming data in one format, and outputs data with the same semantic meaning in another format, as shown in Figure 2 below. Transformation Services are fully reusable and raise none of the visibility concerns that the use of embedded transformations raises.

Figure 2: Using a Transformation Service



The metadata-driven transformation Service on Server 2 transforms the data from format A to format B for the application on Server 1

Source: ZapThink

Use of an external transformation Service also makes it possible to offload CPU-intensive data transformations to systems optimized for such operations. Offloading transformation logic to another system allows the SOBAs to deliver value at their highest level of performance. While the overhead of using a Service hosted on a remote system can incur the overhead of passing data to and from a remote Server, the loosely-coupled, Service-oriented approach of externalizing transformation logic through Service interfaces allows organizations to

Copyright © 2006, ZapThink, LLC

independently scale the number of servers hosting the transformation Service as necessary, without impacting existing Service implementations.

Transformation Services offer an organization considerable cost savings by reducing the hodgepodge of data integration and transformation tools spread across the enterprise.

In addition to the benefits of loosely-coupled transformation logic, transformation Services offer an organization considerable cost savings by reducing the hodgepodge of data integration and transformation tools spread across the enterprise. Licensing costs for these tools add up, and reducing the number of such solutions that a business has to deal with decreases hassles and cost. In addition, through the use of transformation Services, organizations gain increased visibility of the transformation activity within the enterprise. Increased awareness of such transformation activities provides valuable insight, both for capacity planning and, if necessary, for undertaking data migration efforts. Disjoint transformation tools can't provide these insights.

Transformation Services are most useful in groups of SOBAs that share Services amongst themselves. Orchestration logic can route information through a transformation Service, both before passing it to a Service, and before processing a response from a Service. From the perspective of the SOBA's author, a transformation Service is simply another Service, and data within the application flow seamlessly from one Service to another. The transformation Services simply perform the necessary steps to make the process work properly.

Using Data Transformation Services to Leverage Legacy Assets

Transformation Services are also of great value when building Services to leverage legacy assets, easing the task of dealing with legacy data formats. There is no arguing that legacy systems perform vital, often irreplaceable functions for most companies. Tremendous business value resides on these systems—both in the form of essential data as well as critical business logic, and replacing these systems is simply not a viable business decision for most companies. Success of SOA implementations are therefore heavily dependent on effectively leveraging these legacy assets by exposing their functionality and data through Service interfaces and composing those Service assets into continually evolving SOBAs. There are three basic approaches for incorporating legacy applications and data into an SOA implementation:

- Accessing the data locked inside legacy systems directly
- Accessing business logic through APIs or other programmatic means
- Emulating user interaction through terminal emulation, a.k.a. screen-scraping

A transformation Service can be of great value when creating a Service wrapper by easing the task of dealing with legacy data formats.

The choice of which approach to take depends on the specific requirements and technical limitations at hand, but regardless of the approach, the first step to leveraging legacy within SOA is to wrap those systems with Service interfaces. A transformation Service can be of great value when creating a Service wrapper by easing the task of dealing with legacy data formats. Such a Service can convert proprietary, and perhaps unstructured, data formats to and from XML data formats, and transformation Services can also perform batch conversions of legacy system output.

Using Data Transformation Services to Support Multiple Data Formats

Support for multiple data formats in a vendor's product is dependent on customer demand. When customer demand justifies supporting an additional data representation, the vendor will modify their product to provide "built-in" support for the new format. If customer demand for a specific data format is not sufficient to justify vendor support, then the customer is left to develop (and

Copyright © 2006, ZapThink, LLC

maintain) their own transformation logic, or to engage a third party application to perform the transformations for them.

In SOA, the rationale for a Service to support for multiple data formats is much the same as the one for a vendor's products. A Service must support the data formats that most of its consumers are likely to use (encouraging the use of the Service), but rarely used data formats should not be directly supported. When a Service must support multiple data formats, relying on a third party Service to transform the diverse formats into a normalized format can be a better option than modifying the Service's logic to directly deal with all of the differing formats. Another option is to wrap a transformation Service and an existing Service into a composite Service to provide direct support for an alternate data format. Both of these approaches keep the logic of the original Service focused on the business functionality.

III. Requirements for Transformation Services

There are two basic ways to implement transformation Services: hand-coding them to provide a specific set of data transformations, or building them declaratively through the use of extensible metadata, providing the ability to add and change the transformations over time without the need for further coding. Hand-coding transformation Services is typically a first and easy choice if the required data transformations are well understood and unlikely to change. Creating hard-coded transforms requires skilled programmers that code the transformation logic using conventional programming languages. The upside to this approach is that skilled programmers can develop optimized and efficient transformations. The downside is that code optimization ties the resultant transformation logic to the intricacies of the data formats, and the code is often hard to validate, hard to maintain, and very brittle.

An extensible metadata-driven transformation Service, on the other hand, is often a better choice over the long run if it is likely that your business will need to modify existing transformations or to add new transformations in the future. Metadata-driven transformation Services use declarative definitions to bridge the semantic gap between data formats, mapping the fields of one data format to fields with the same meaning in another format. A runtime transformation engine reads these mapping definitions, and executes the desired transformations.

Metadata-based transformation definitions specify the mapping details between the fields, rather than the intricate steps necessary to perform the data transformation, enabling the use of visual tools to aid the development and testing process, resulting in transformations that are easier to develop and to validate than hand-written transformations. A comparison of hand-coded to metadata-driven transformation approaches appears in Table 1 below.

Metadata transformation definitions take into account complex data hierarchies and relationships, as well as the need to access to multiple data sources and scattered resources. The transformation engines that execute the definitions are quite sophisticated and provide features beyond those normally available to individual developers. Utilizing such a separation of concerns, creators of metadata transformation definitions can focus on the accuracy of their data maps, leaving the intricacies of efficient transformation execution to the developers of the transformation engines.

Copyright © 2006, ZapThink, LLC

Table 1: Comparison of Hand-Coded to Metadata-Driven Transformations

Transformation Service Approach	Advantages	Disadvantages
Hand-Coded (custom code)	➢ No supporting infrastructure necessary ➢ Possibility to create highest performance transformations	➢ Requires skilled programmers ➢ Possibly diverse transform implementations ➢ High cost to validate transforms. ➢ High cost of change
Metadata-Driven	➢ Tool supported transformation creation ➢ Highly tested transformation engine ➢ Transformation implementations are consistent and easier to validate ➢ Lower cost of change	➢ Higher upfront cost and learning curve for transformation engine and tools ➢ Economic benefit realized over time as transformation library takes shape.

Source: ZapThink

As an organization's library of transformation definitions grow, the ongoing costs associated with developing and utilizing metadata-driven trans-formations drop.

Metadata-driven transformation Services deliver economic benefits over the long term when compared to hand-coded transformation services. As an organization's library of transformation definitions grow, and as personnel become familiar with the supporting tools, the ongoing costs associated with developing and utilizing metadata-driven transformations drop. In contrast, hand-coded transformations may be easy and inexpensive to create in the first place, but they are remarkably expensive to modify and to maintain over the long run. Skilled programmers with detailed knowledge are required to maintain the transformation logic, and the resulting transformations are difficult and costly to modify as the need arises. Indeed, in such situations, most companies simply resort to ripping out their hard-coded transformation logic and replacing it with new, hard-coded logic, only to repeat the cycle years (or sometimes, months) later.

Transformation Service Factors to Consider

There are several factors to consider when evaluating transformation Services:

- Can the transformation Service handle a large volume of data transformations?
- Does the transformation Service support data transformations between formats the business requires, both now and in the future?
- Can the transformation Service deal with structured, semi-structured, and unstructured data sources?
- Can the transformation Service scale as needed?
- Are tools available to help create, modify, test and deploy the transformation logic in the Service?

A high-performance, high-volume transformation engine should be at the heart of any production-quality transformation Service. If the transformation engine is not efficient, data transformations may be time consuming and thus slow Service response. Organizations should be able to deploy the engine on multiple servers and provide instrumentation support for monitoring transaction activity and overhead. These capabilities allow the business to track transformation activity and scale transformation capacity as demand grows.

Copyright © 2006, ZapThink, LLC

Visual mapping tools should simplify the creation of metadata mappings between one data format and another. Data mapping can be a long and laborious process and can involve accessing diverse data sources and navigating complex data relationships. The visual mapping tools must handle non-trivial mappings to be of any real use, yet remain easy-to-use to allow business analysts and auditors to inspect and review the resultant transformations. The transformation tools should also support the running of both *ad hoc* and structured tests to insure the accuracy of a transformation, and to determine the execution speed and CPU overhead of each transformation.

Look for centralized transformation facilities for use across the enterprise.

Look for centralized transformation facilities for use across the enterprise, such as a central library to store all existing transformation logic and associated metadata, and collaborative tools to create, test, and deploy new and modified transformations. Centralized transformation facilities should also provide a common mechanism for locating and reviewing existing transformations as well as for monitoring the creation, modification and use of transformations to meet auditing and compliance requirements.

IV. Xenos Transformation Solutions

Xenos is a global provider of high-performance data and document management solutions. Their data and document management solutions enable their customers to pull data from multiple discrete sources and repurpose that information to enhance business processes. Their products support a variety of standards and streaming formats, and provide real-time information capture, transformation, transportation and presentment. Xenos products leverage existing IT investments to create applications that transform and repurpose existing data and electronic document print streams.

The Xenos Product Suite includes three core applications:

- **Xenos d2e** — An electronic document repurposing application that provides high-volume electronic document capture, indexing and transformation for print, eBusiness and enterprise content management applications.
- **Xenos infoWEB** — Provides a transport for electronic presentment of information for collaboration, business decision making, enterprise performance, and multi-channel electronic presentment.
- **Xenos terminalONE** — Captures, transforms and transports structured data formats over the Internet and across the enterprise. With XML-based data dictionaries, enterprises can capture and transform electronic transactions and transport them in any format.

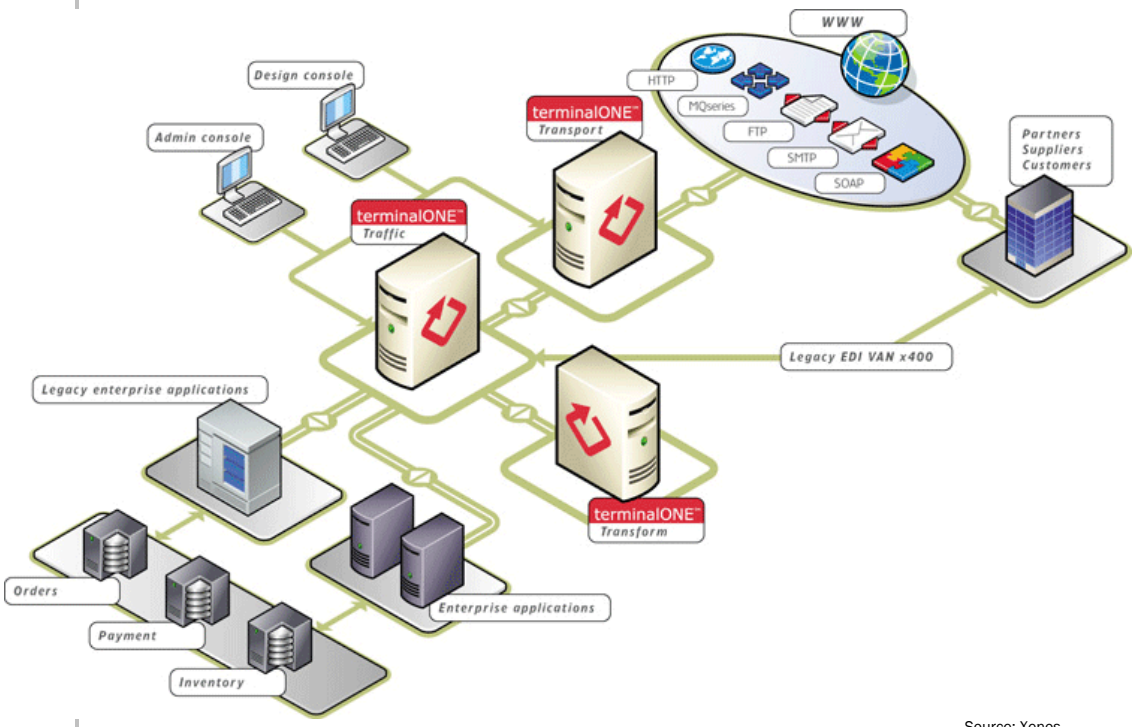
Xenos solutions include a desktop design console to ease the offline creation, validation and testing of transformations, and an admin console from which users can hot deploy new and modified transformations to high-volume transformation engines on one or more servers. The Xenos transformation hubs intercept relevant data and documents from any application, repository, or data stream in real time, transform captured data from any format to any preferred format, and assemble the data with additional content according to customized business rules. Xenos transport solutions then securely transport the results to any application, database, archive, or information consumer or routed directly into back-office systems through any available channel of delivery.

Data transformation is clearly one of Xenos' core strengths.

Data transformation is clearly one of Xenos' core strengths, and plays an important part in each of their core applications. In particular, a core component

of Xenos terminalONE is terminalONE Transform, which serves as a transformation engine that can underlie transformation Services that provide the benefits this paper discusses. TerminalONE Transform is a core component of the terminalONE transaction gateway solution. It employs a high-speed transformation engine that can convert any transaction data format into any other, including X12, EDIFACT, HIPAA, SWIFT, HL7, XML, HTML, CSV, or text on the fly. The full architecture of terminalONE, including terminalONE Transform, appears in Figure 3 below:

Figure 3: Xenos terminalONE



Source: Xenos

Xenos terminalONE is an end-to-end transaction gateway that expedites business transactions over the Internet and across disparate platforms. TerminalONE provides high-availability, high-volume data transformation, transaction security, and message routing. Companies that leverage terminalONE for B2B integration can achieve the benefits of EDI without most of the drawbacks, at a reduced cost.

Xenos' terminalONE gateway alleviates B2B integration problems by speaking multiple languages and performing bidirectional transformations on messages. It also includes a secure messaging server. TerminalONE solves problems on both sides of each integration by acting as a transformation hub that can reduce the provisioning time for new participants in B2B transactions.

TerminalONE Traffic can identify the necessary data formats before routing any data to any destination. If the data are not in the proper format, it will automatically request a transformation from terminalONE transform. Finally, the third component of the terminalONE solution is terminalONE Transport, which is a gateway solution that enables the reliable and secure point-to-point exchange of data between disparate back office systems over the Internet. TerminalONE

Copyright © 2006, ZapThink, LLC

Transport guarantees secure delivery of any kind of validated transaction data to authenticated recipients and specific back-office applications in the appropriate formats.

By leveraging the power of the entire Xenos suite, companies can address most of the challenges of data integration.

By leveraging the power of the Xenos suite, companies can address most of the challenges of data integration, without having to change their systems, protocols, or data formats. Instead, the transformation, routing, transport, and security capabilities of Xenos products provide intelligent intermediary capability that loosely couples participants, reducing the cost, risk, and time commitment necessary to respond to changing business needs.

V. The ZapThink Take

SOA has not changed the fact that companies have to contend with a wide range of diverse, disparate data sources on a daily basis in order to run their daily operations, make informed decisions, and work with their customers, partners, and suppliers. Companies continually find themselves with the challenge of meeting their data and application integration requirements in a heterogeneous environment of continuous change. The semantic integration challenge remains as an obstacle to fully realizing SOA's promise of business agility, however. Companies struggle with how to liberate the meaning of data from the structure of its representation, so that information can flow seamlessly between the Services that power the business.

By separating transformation into a separate Service, its possible to handle data transformation declaratively, increasing the flexibility of the transformations.

Data transformation Services can address the challenge of data and semantic integration. By separating transformation into a separate Service, it's possible to handle data transformation declaratively, increasing the flexibility of the transformations. Furthermore, it then becomes possible to host those transformation Services on a dedicated transformation engine like those from Xenos.

Such transformation engines provide the scalability and manageability organizations require from their transformations, while at the same time enabling the loose coupling of the transformation Services that allow such Services to address data and semantic integration issues in the context of SOA. In fact, Xenos provides essential Service-oriented building blocks for meeting the semantic challenge. The real-time high performance metadata driven transformation engine, coupled with powerful, easy-to-use tools for creating, validating, and deploying transforms enable an agile approach to data transformation.

Copyright © 2006, ZapThink, LLC

Copyright, Trademark Notice, and Statement of Opinion

All Contents Copyright © 2006 ZapThink, LLC. All rights reserved. The information contained herein has been obtained from sources believed to be reliable. ZapThink disclaims all warranties as to the accuracy, completeness or adequacy of such information. ZapThink shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.

About ZapThink, LLC

ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

ZapThink helps its customers in three ways: by helping companies understand IT products and services in the context of Service-Oriented Architecture (SOA) and the vision of Service Orientation, by providing guidance into emerging best practices for Web Services and SOA adoption, and by bringing together all our audiences into a network that provides business value and expertise to each member of the network.

ZapThink provides market intelligence to IT vendors and professional services firms that offer XML and Web Services-based products and services in order to help them understand their competitive landscape, plan their product roadmaps, and communicate their value proposition to their customers within the context of Service Orientation.

ZapThink provides guidance and expertise to professional services firms to help them grow and innovate their services as well as promote their capabilities to end-users and vendors looking to grow their businesses.

ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into the best practices for planning and implementing SOA, including how to assemble the available products and services into a coherent plan.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOA by vendors, end-users, and the press. Respected for their candid, insightful opinions, they are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry. ZapThink was founded in November 2000 and is headquartered in Baltimore, Maryland.

ZAPTHINK CONTACT:
ZapThink, LLC
108 Woodlawn Road
Baltimore, MD 21210
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com