PARTNERCOMMUNITY®

BONDED for SUCCESS™

Using Asymmetry to Achieve eBonding Scalability

An overview of a proven way to make B2B integration scalable to a large number of enterprise customers and partners

August 2010

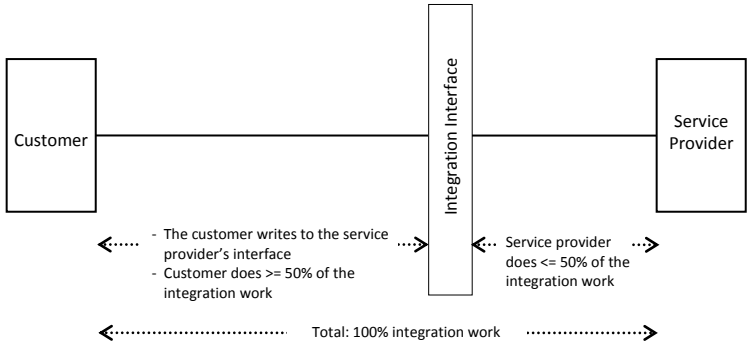
PartnerCommunity, Inc.

John Yin
(561) 376-2456
jyin@partnercommunity.com

For any service provider having a large number of enterprise customers and partners, it is important not only to be able to integrate with them, but also to do so quickly, reliably and cost-effectively. In other words, the service provider needs an integration solution that is sufficiently scalable, in terms of both implementation and ongoing maintenance.

We have been helping telecommunication service providers to electronically bond (e-bond) with their large enterprise customers since 2002. In the past eight years, we have done many dozens of integrations and learned some valuable lessons. The purpose of this white paper is to share one of these lessons relating to integration scalability.

Conventionally and intuitively, most service providers choose to define an integration interface that sets the demarcation near the center between themselves and their enterprise customers. This way, the amount of work that the two integration parties need to do would be about the same. But in reality, more often than not, the demarcation is actually closer to the service provider than to the customer, though we are not saying that the service provide is having this interface closer to itself necessarily on purpose. The reason is that as the service provider is much more knowledgeable of its own terminologies and processes, it so defines the interface in a way that is naturally more self-centric. This tendency usually means that the customer would need to do more work for the integration than the service provider, as illustrated in the diagram below.



Conventional, meet-in-the-middle integration approach is difficult to scale. To make an integration solution reusable, an asymmetric approach should be used.

Such a division of efforts renders a large portion (at least 50 percent) of the integration effort not reusable, for the simple reason that the integration work done by one enterprise customer is usually not reusable to others. An enterprise customer understandably has neither the visibility of others' requirements nor any incentive to make the work that it does reusable to others which may even include some of its competitors.

This loss of reusability may not be a big deal if this integration does not need to be repeated for more than a few enterprise customers. However, when the service provider needs to integrate with dozens or hundreds of customers and partners, it then becomes a huge deal. Every customer/partner has to repeat the "same" work, most likely make the same mistakes and, because of this, endure the same kind of frustration as every other customer or partner. In general, therefore, the closer the interface is to the service provider side, the less scalable the integration solution is. Conversely, by moving the interface towards the customer side and making it highly

asymmetric, it would help reduce the integration work that a customer has to do, and provide the service provider with an opportunity to productize its integration solution that can scale to a large number of customers. We have helped service providers to achieve up to a 10-90 asymmetry using an integration form approach, which will be described in detail later in this white paper.

This being said, for most developers, taking such an asymmetric approach is neither intuitive nor trivial. In order to design such an asymmetric interface, the service provider would need to possess a deep understanding of its enterprise customers, which includes knowing

- The customer's terminologies and processes for the type of services being integrated
- What a customer needs to do for such an integration
- Which portion of the integration is replicable with other customers
- Which parts of the replicable portion can be done by the service provider without making the integration tightly-coupled

Usually such expertise is not sufficiently accumulated until after several successful integration projects. Therefore, there goes this chicken-and-egg dilemma. By the time a service provider has finished a few integrations using a symmetric approach, its developers usually do not want to make any change to something that has already "worked."

Based on dozens of integrations that we have successfully done, we have developed a set of guidelines for selecting the optimal demarcation as well as for productizing those additional integration responsibilities belonging to the service provider because of this asymmetric integration approach.

Three keys for successful asymmetric integration:

1. *Push demarcation into customer application if appropriate*
2. *Develop integration adaptors to leading customer applications.*
3. *Also make processes and professional services repeatable.*

The first key for a successful asymmetric integration is to push the demarcation closer to the customer application being integrated with as appropriate. This minimizes not only the work that the customer has to do, but also the people and skills required to do the integration on the customer side. For example, if the customer application is Remedy®, then by pushing the demarcation into Remedy, the customer would only need a Remedy developer to carry out the integration, instead of also requiring Web Services developers.

The second key for a successful asymmetric integration is to focus on leading customer applications that are being used by many customers. Pushing demarcation into customer applications requires that the service provider be able to interact with these applications directly. By focusing on a small number of leading customer applications, the service provider can maximize the reusability with minimum amount of investment.

The third key for a successful asymmetric integration is to focus on not only productizing the integration middleware, but also "productizing" the related process and professional services. If the service provider can create highly repeatable processes and professional services, the customer will have less work to do for the integration.

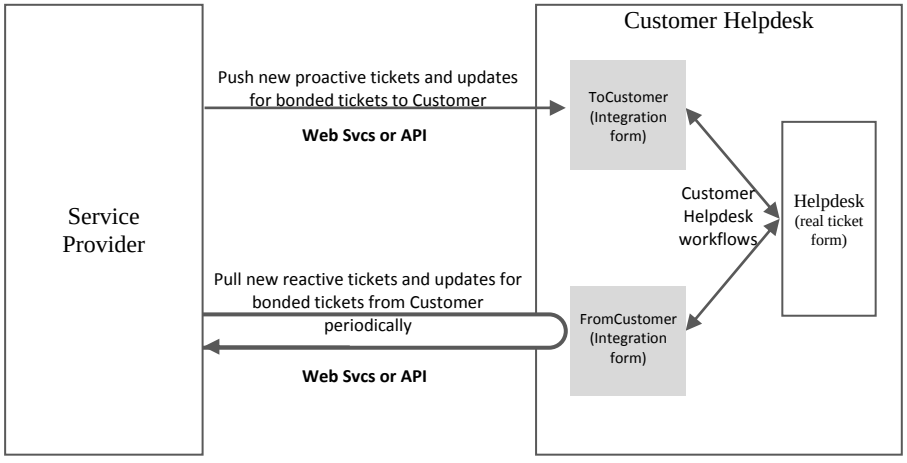
If a customer already has an external integration Web Services interface for its vendors, then it can be used to accomplish such an asymmetric, loosely-coupled integration. The customer's business rules and processes are encapsulated between

its Web Services interface and its helpdesk application. As long as the service provider can efficiently develop the required code to call such an interface, the customer does not have to do much for the integration. There are ways that a service provider can reach out to a customer's interface in a highly reusable fashion, e.g. using the eBonding appliance (EBA) from PartnerCommunity, but that is outside the scope of this white paper. However, most customers do not have such an external integration interface. Unlike a service provider, an enterprise customer usually does not have dozens of companies to integrate with. Thus, it is hard to justify the development of such an external interface financially. Even if a customer needs to integrate with more than a few suppliers and partners, the purposes and requirements for those integrations may be disparate. Therefore, it is technically difficult to define just one interface to satisfy the requirements of all the diverse integrations.

Using the proven integration forms approach, one can achieve both loose-coupled and asymmetric integration approach.

For all those customers who do not already have an external integration interface suitable for eBonding with a service provider, based on the numerous integrations that we have done, an approach that uses integration forms has been demonstrated to be the easiest and the most reliable one. In fact, the integration-form approach facilitates a loosely-coupled design as well as the asymmetric integration architecture, as explained below.

To illustrate the concept, we will use trouble ticket eBonding with Remedy as an example i.e., the integration is for incident management between a service provider and an enterprise customer whose helpdesk is Remedy. The approach can be applied equally well to other helpdesks such as HP's Service Center (HPSC, formerly Peregrine). Even for applications that do not have an API as sophisticated as Remedy or HPSC, the integration-form approach can be applied by using database tables instead of application forms. Under the integration-form approach, integration forms (usually two – one for messages in each direction) are added to the helpdesk application to exchange messages between the two parties. For readers who are familiar with the message-queue interface, the two integration forms are used as message queues but without requiring all the knowledge about message queue technologies. Such integration forms can usually be accessed via application API or Web Services that can be easily generated by the application. Some developers call the integration forms staging forms.



PARTNERCOMMUNITY®
BONDED for SUCCESS™

The service provider will insert a new record in the ToCustomer form every time it has a message for the customer. The message may be for a new proactive ticket or an update to an existing bonded ticket. The message will contain an explicit indication about what the message is intended for. The insertion of this message will trigger workflows in Remedy to take the proper action, e.g. creating or updating the real ticket in Remedy.

Similarly, the customer helpdesk (workflow) should insert a record in FromCustomer form every time it has a message for the service provider, based on the actions of its helpdesk users. The message may be for a new reactive ticket or an update to an existing bonded ticket. The message should also contain an explicit indication about what the message is intended for and what it wants the service provider to do concerning the ticket. Such messages are polled by the service provider periodically.

Using such an approach, the only developer needed on the customer side is a Remedy developer because the only development work on the customer side is to move data between the integration forms and the real ticket form, both of which are inside Remedy. The developer can utilize all existing Remedy development tools without having to learn about and write to the service provider's "symmetric" Web Services interface. It has been shown that using the integration form approach, the total amount of development time required on the customer side can be reduced by approximately 80 percent. Furthermore, as the customer application is being upgraded, most integration forms and its related workflows can be upgraded automatically without requiring any development work.

If you have been frustrated by the never-complete and/or failed integration projects, the inability to bring eBonding benefits to a large number of your customers, or your customers' refusal to migrate to your new version of interfaces, then you should look into the asymmetric integration approach. It does require that you do more work upfront, but the payoff is much greater.

For more information about the asymmetric integration approach, please contact:

John Yin, PartnerCommunity, Inc.
561-376-2456
jyin@partnercommunity.com.

PARTNERCOMMUNITY®

BONDED for SUCCESS™