main index menu

PyBiz
The pinnacle of next generation refined search technology

Version

# Business Overview

## XDisect XML / HTTP Interface

Email eval@pybiz.com with questions or comments.



## Introduction from Joe Ellsworth, CEO of PyBiz

During the last several years the PyBiz core team worked on four major secure portal projects in both the B2B and B2C space. We drove all of them from concept through business buy-in and two all the way through product implementation and enterprise deployment.

We help businesses resolve problems affecting their secure portals, especially those that occur when businesses merge or enter new partnerships. Using extensive examples and describing several of our past projects, this overview covers the problem with merging data from different secure portals, and why XDisect is not only the solution but improves sustainable competitive advantage.

In 1999 Swatch and Hewlett-Packard publicly announced that they were going to build a secure portalcalled the "Swatch Universe" for Swatch customers. The basic idea was that Swatch had the brand pull while HP had the knowledge and technology experience to deliver and help operate such a portal. We know about this because at the time I was working at HP and was eagerly waiting to hear HP's CEO, Carly Fiorina, announce the deal -- a culmination of months of brainstorming and hard work for me and many others. We can talk about this now because Carly announced the project publicly at a major tradeshow.

"Swatch Universe" is an example of a B2C secure portal. Secure portals are valuable because they are all about owning the customer (end user) and making goods and services available to them in a nicer, easier, faster, or better way than could be experienced by going directly to the service provider. As long as a secure portal continues to provide powerful, useful and sometimes fun services, it can expect to retain its customers. The portal operator makes money as a middleman connecting those having valuable services with those wanting to buy them.

During 1998 HP announced ESN -- its own B2B secure enterprise portal where

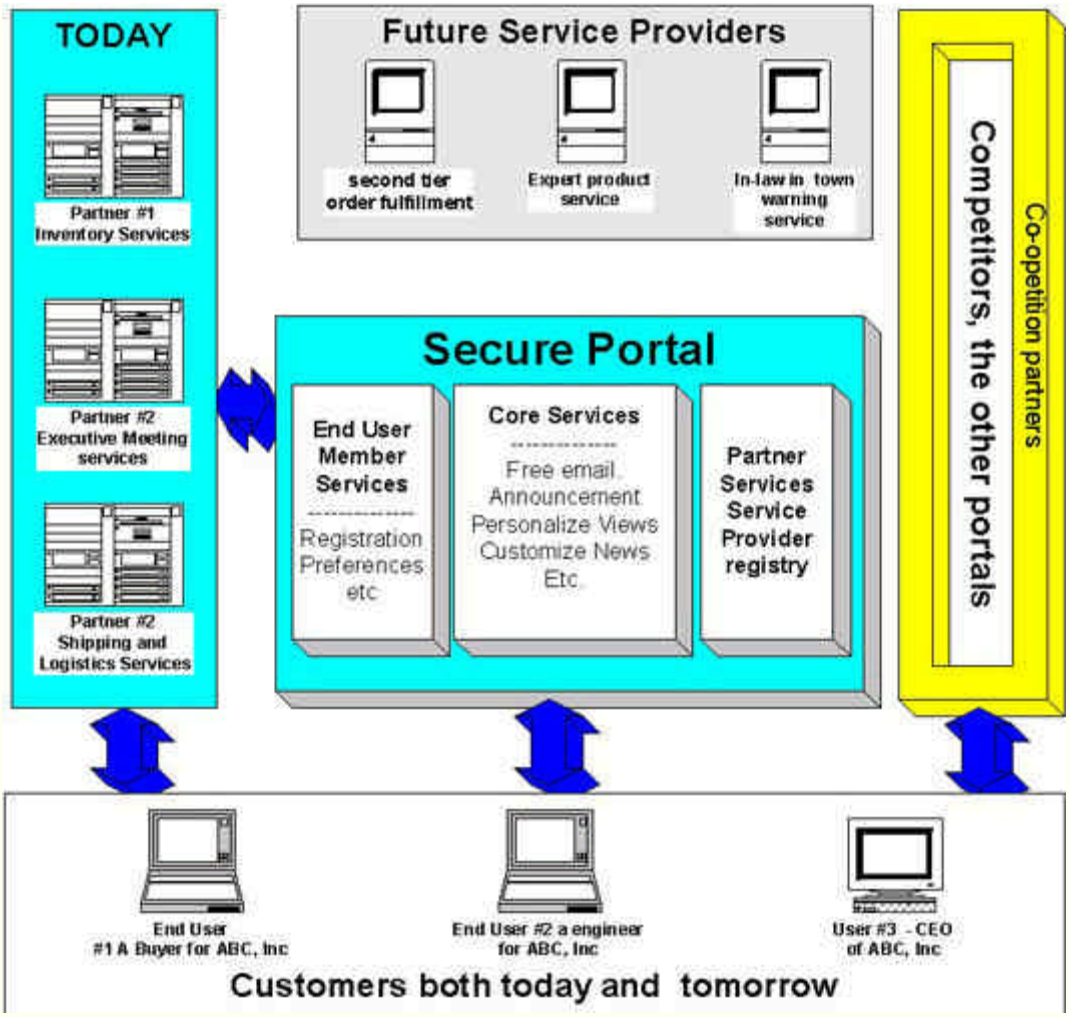
large customers and partners could log on to gain access to orders, inventory, configuration, news, and special offers. I mention this because I came up with the idea for ESN during 1996 and finally obtained the money and sponsorship to widely deploy the product during late 1997 and 1998. Of course, major projects in large companies are team efforts and there were several hundred people involved along the way. I was very pleased to have ESN, my brainchild, listed in HP's 1998 financial report as a key enabler for the emerging eBusiness centric way of doing business.

We are going to tie all this in with PyBiz and XDisect, but first let me say that we were, and are, pioneers that have consistently lead the thinking and business implementation processes in the secure portal space. We have gained unique, real world, hands-on, practical business and technological experience, and we have found some things that work really well and some that don't work at all. PyBiz solutions and services are targeted head-on at solving some of the most difficult problems we have encountered during our journey.

We have found that many of the same macro-problems occur in businesses that are not operating secure portals but have similar characteristics.

## Secure Portal Business Architecture

Before we go on, let's look briefly at the business architecture of the typical secure portal. Keeping the business architecture in mind will help you better understand some of the technological impacts on the secure portal.




## Characteristics of Secure Portals

- Secure portals generally aim to provide services to a large user base and every user is important.
- Users only have to log on and provide preference information once; the secure portal ensures that the service partners have the right information at the right time to perform services for the user.
- Sometimes the portal provider may act as a broker so the service provider doesn't know who the end customer is.
- The secure portal may collect payment for services rendered by others and then pay the actual vendors.
- Once a portal is successful, potential partners will stack up faster than L.A. traffic.

- End users expect secure portals to be responsible for what is offered and displayed. This is one of the most significant differences between the popular free portals and the emerging trusted secure portals.
- Every partner is different and will have slightly different requirements and values.
- The service-providing partners need to know who the user is and what they are supposed to be doing for that user when the user arrives at the partners' site.

### Objectives for Successful Secure Portals:

**Portal providers must continue to add new services to retain customers.**

- End users are fickle; just because they are happy today doesn't mean they will be happy tomorrow.
- Web users are harder to retain as an audience and tend to experiment on their own.
- Your existing users are the target customers for several other portals.

**Recruiting and retaining partners is essential for continued success.**

- It is impossible to build all services internally and keep pace with competition.
- Best-of-breed services are available first externally. (Just try to have an internal IT staff build Microsoft Hotmail!)
- Service vendors are hungry for any and all visibility.
- Best-of-breed services become essential as soon as your competitor has the servers available.
- Many partners require that the users come directly to their sites. This is mostly a branding issue but there are technical reasons as well.

## Business Problems with Growing Secure Portals

## New Service Partners are Mandatory

Recruiting and signing up new partners is a common business practice, but let's look at what happens after the partner is signed up and it's time for one partner to offer its services to the other's customer base.

**How do we cope with different data requirements from**

**partners?**

- This is the kicker. Imagine that when you started the portal, you had decided that you needed a user's name, company, credit card, email and home phone. In XML this would look like this:

*Hey this XML stuff is easy, right?*

Now this works fine and the portal is rolled out on time. The problem is that your fantastic BizDev department has gone out and signed up 3 new customers, each with 500 users, but they will only join your portal if you can provide access to "Wild Slugs Unlimited" and only if they can be live in less than 20 days. So now your BizDev people go out and recruit Wild Slugs.

**Here comes the problem**. Wild Slugs requires the user to supply a full address for both home and work and a list of Slug types the user is interested in. After all, if you have a live wild slug to ship, you need to get it there fast or the user will just get slimed.

So now you go back to your faithful IT staff and instruct them that you need to store and save this information. They might panic, or if you have a couple of really good people they set about:

- re-designing the database structure to store this data;
- converting the old data to the new structure;
- modifying the programs to use the new structure; and
- exhaustively testing all the changes

At the end, after a lot of work and several sleepless nights, they come up with a new structure that looks like the following in XML:

*Well maybe not quite easy but engineers like this stuff.*

This all works great ... except in the meantime your BizDev guys have went out and recruited a gaggle of new partners and these new partners have requirements of their own that aren't accommodated by the above structure. More work hours, "When it is OK to call at home?, more stress. If you can imagine it, one of your partners will probably need it. **So your IT staff probably hasn't even finished the last round of changes and here are a bunch more**.

Just imagine how fast this IT problem will mushroom as you sign up new partners. Eventually you may reach equilibrium, having all the data a new partner needs, but only because another partner needed it first. It could take a couple of years to get there and this could be achieved only if partners didn't invent new requirements.

By the way, the data you use to describe your partners and their services is likely to change even faster than your customer data.

## You have some choices.

Believe me, this is pretty much the way it goes. (We did this for two years on the HP ESN project and the rate of change never slowed down.) Your choices for handling this IT predicament:

**Limit the growth of your partners to meet your current technical capabilities.**

**Batch your changes together and wait.**

**Hire a huge IT staff and buy them all BMWs for working late.**

**Anticipate every possible need and get to market 5**

**years late.**

**Tell Wild Slugs to maintain their own extra data.**

### Use XDisect from PyBiz
*** Target the problem head-on and solve it ***

- XDisect stores any XML data structure without fussing with schema changes, so you can just create new data structures and dump them in.
- XDisect allows the old (simpler) structures to exist at the same time as the new (more complex) ones, and you don't have to migrate the old to the new unless you want to.
- XDisect provides search and retrieval functions which are important for your partners and your IT staff:
  - XDisect can return the old and new records in the same query; and
  - XDisect has special features that make it possible to discover the right records even when the structures have changed.
- XDisect efficiently searches on any word, in any attribute, no matter how complex the structures get.

I could go on and on but that is what product brochures and technical manuals are for.

**Let me summarize it this way**, CEO to CEO: Deploying a secure portal that can just *keep up with--let alone keep ahead of*--new partners' needs is really difficult. Most of the technologies today just ignore the problem, forcing the partners to give up and work with less.

In the emerging eBusiness economy, you have no choice but to deal effectively with diversity and diversity means changing data structures. The technologies and tactics from the past just do not solve the problem no matter how hard the IT staff works.

Think of it this way: Those who lead the changing business conditions go in directions that are beneficial for them and their stock holders. The rest are just along for the ride. One of the requirements of leading change is to be able to cope with the impact of that change. XDisect has been designed from the ground up to allow our customers to embrace change at a minimum of cost.

XDisect is the only technology we have found that targets the problem head-on, and that's because we encountered the problem in several major enterprise scale projects and built the product to solve the problem.

If you want to maximize your success in the secure portal space, then you need XDisect's capabilities.

If you look around your company, I bet that you will find similar problems that vex your IT and business staff. XDisect is the solution--we can break down those logjams. You might have to re-educate some IT guys along the way, but once they start seeing the benefits, they will get really excited.

## How do we securely transfer users to the service?

So now you are running a secure portal. You have signed up your users, you have all their data, and you have determined that you need to get some services from a partner. The first challenge is getting the user transferred from the portal where they logged on to the service partner--ideally without the user noticing along the way. This type of problem is called "Single Sign On" today, although when we invented the first-generation solution in 1997 we called it "Secure Server Transfer" or SST.

When we pioneered the technology at HP in 1997, this was a new and challenging task. Now there are a wide range of companies focusing on Single Sign On technologies. The SST technology eventually became known as "CEF" and HP donated it to the open source community in the spring of 2000. PyBiz has redesigned an even better version (fourth generation) which is used as an enabling tool in our solutions.

One thing to watch out for is that many of the current Single Sign On solutions require that their software runs on every server. This is OK for many of today's solutions since all of the partners are closely related, oftentimes inside the same company. But this becomes a problem when you go to your CFO and ask him to come up with an additional $90K to cover the license for your new partner "Wild Slugs Unlimited". You need Wild Slugs' service but the CEO over at Wild Slugs isn't willing to buy-in to your Single Sign On vendor because she paid for a different one last month to work with one of your competitors.

This will be a killer for a portal that needs to sign up a large number of external partners rapidly; luckily, open published protocols such as CEF can come to your rescue.

As we originally wrote it, the CEF specification conformed to an architectural concept called "Loosely Coupled Adaptive Architectures". Simply put, this means that you can build your service applications on any platform, using any computer technology and programming language, and still have them work together as if they had been designed and built to work together from the beginning.

CEF includes an open specification so participants can build their own applications to interact with CEF-enabled portals.

CEF is today's only true open standard, open specification product. We consider a standard *open* only when:

- it is possible to build an interoperable component in reasonable time with nothing but the specification (spec) document;
- it is feasible to build products conforming to the spec without any of the software supplied by the original spec author or company
- it is cost-effective to start with any standard development environment such as Python, Perl or Java, and build software components capable of interoperating with other components conforming to the spec.

## Other Questions We Can Answer

### How do we measure what the users do?

### What data do we share with each service provider?

### How do we securely share data with the service provider?

PyBiz software runs best on Linux because it has been built and tested on Linux.

www.pybiz.com
eval@pybiz.com 408-364-1741