# CEF & XDisect use in a Community/Portal scenario

This document describes the use of CEF & XDisect in a Market Maker/Community scenario. The objective is to depict a business scenario that is easily understood and believed by a wide audience. It includes a description of the participants in the scenario, the data being exchanged by the participants and the high level flow of information

## *CEF Overview*

CEF represents a mechanism for collaboration between websites that interact with each other to provide a common set of services to end-users. It is open source and includes the base CEF protocol layered on top of http and its implementations in different languages.

### CEF Features/Benefits: -

Below are some key CEF features that will be illustrated in this scenario

- *Single sign on* - users need to login within the community or market at only the entry point. All the other sites in the community can reuse the same login & not prompt the user for any login information. It also reduces the standard user headache of password proliferation and maintenance
- *Better personalized experience due to a shared session* – allows various sites within a community or market to share and exchange information about the user's session, thus allowing for better personalization of the user experience. Of course this is governed by the security policies of the community and by the preferences of the user.
- *Potentially Consistent Look and Feel within the community* – since the transfers between the sites in the community is transparent to the user, it opens up the opportunity to provide a consistent experience to the user, so that the user feels like they are visiting one virtual site rather than 20 different sites altogether.
- *Dynamically gather information from the user as required*- this provides sites in a community the flexibility to gather only as much information as is required to personalize the experience for the user, rather than asking the user to fill out a huge form upfront during registration. In addition the information can be collected only if the user visits the site (the first time).
- *Open platform for interaction between websites* – CEF is based on industry standard web technologies like HTTP and XML. The basic protocol behind CEF is open. The source code for CEF is available in several languages and is open source. It can be found on sourceforge
- *Selective sharing of information between the portal, partners and service providers* – Typically, depending on the type of business contracts and relationships, a portal/community will trust its partners and its service providers to varying degrees/ levels. CEF allows the portal/community to share profile information with these participants according to the level of trust. This can reduce the risk of exposing too

much customer and service information to vendors that are semi-trusted or untrusted and on the flip side not trying to limit the amount of information that is shared with the trusted partner sites.

- *Lowers Adoption Resistance amongst service providers & partners* – Since CEF requires very minimal integration with service providers, which is again based on HTTP/XML, it expedites the process of getting service providers to participate in the community without forcing them to go through a long integration cycle and also without disrupting their existing site implementations
- *Allows both the central portal and service providers to concentrate on their core value add* – CEF allows tasks like user registration, linking between sites, profile management to be delegated & centralized (not mandatory) at the portal. This lets the portal focus on its core value, which is being an entry point for their customers, providing the starting set of menus, doing customer registration & profile management. It lets service providers focus on their core value add, which is to deliver the actual core service they want to without having to get side tracked into receiving users, managing their profiles, etc.

PyBiz provides support and consulting services around CEF to help customers integrate CEF into their community/ marketplace/portal architecture

## *XDisect Overview*

XDisect represents a key enabling technology for marketplaces, communities and portals.  It allows flexible storage and searching of structured data

### **Key XDisect Value: -**

- Faster time to market.
- Ability to adapt to changing business conditions.
- Minimizing IT infrastructure and planning costs.

### **XDisect Features / Benefits: -**

This white paper will highlight the benefits of building a community/portal using XDisect. Some of these feature / benefits include:

- *Flexible profile descriptions* – Allows complete flexibility in describing service provider and customer profiles without restricting them to just a flat set of static attributes.  It also allows the user profiles to evolve as the business needs of the community evolve and mature, thus allowing online communities to keep their solutions in lock step with evolving business needs
- *Personalized Catalogs* - Makes generation of personalized filtered catalogs & menus tailored for end customers easy and efficient.  In addition tailoring the content to the usage patterns of the user & presenting related services to the user could be done fairly easily using XDisect.
- *Multi-Vendor Catalogs* - Makes aggregation of catalogs from multiple vendors easy even when the service provider profiles are described using potentially different taxonomies.

- *Effective query mechanisms* - Provides effective queries for profile information across previously incompatible taxonomies.

- *Lower Adoption Resistance from partners and service providers* - Minimizes the adoption resistance for service providers by allowing them to model their product data & their profile information in a way that makes sense for them without mandating a comprehensive schema. Also it places minimal (if not none) technology demands on participating partners, thus making it easy for them to participate in the community at a relatively low cost.

- *Flexible Searches for profile and catalog information* –It provides the ability to flexibly search and retrieve information about profiles and service offerings. Search queries do not have to be pre-defined or "canned". Applications can search profiles on any attributes or elements within the profiles without having to manually specify the queries or index all the elements. It can also be useful for status queries by customers and service providers.

- *Lightweight Events based interaction* –Lightweight events allow the community participants to interact with each other in ways that cannot be pre-defined and need not be. E.g. a customer can request that they be notified when their desired service becomes available at their desired price. Thus they do not have to keep checking constantly for the best deal. On the flip side, service providers can be notified if a buyer arrives for an item they are interested in selling.

- *Fast Ramp up of participants & quick integration with partners* - XDisect makes it easier to ramp (scale) up membership in a community. XDisect allows the profile information, usage reports, etc to be maintained without a rigid schema or structure. This is extremely useful in a community where service providers and customers all have their own backend systems that store this information in different formats. Trying to get hundreds of customers and service providers to agree on a format can be a Herculean time consuming effort. This is compounded when service providers are forced to suppress what they consider key data elements in order to adhere to the market schema

- *Adapts to dynamic market requirements* - XDisect makes it easy to accommodate new participants (service providers and customers) who have special requirements. XDisect engine with its flexible data storage and retrieval also makes changing the formats of the offerings, profile documents very easy. The only place that will have to change is the front end GUI rendering logic of the community. The rest of the application layers (including XDisect) do not mandate a pre-defined structure of the XML documents and hence it would be easy to change the structure without a massive re-engineering effort.

- *Allows Data to Evolve* - XDisect makes it feasible for the community to evolve their data infrastructure as fast as their business conditions change and as the business requirements of the participants change. E.g. to start with the community might model its users using a flat profile that includes company, role, security, preferences and personal information. As more corporate customers are added, they might want to normalize the profiles into user profiles, company profiles, security profiles, etc. Later on as they get more enterprise customers, they might find that the company profiles themselves might have to be factored into geography, sites and holding company information. All of this can be accommodated without requiring massive schema changes and parallel re-engineering efforts

- *Enables Cross-Domain Information Retrieval* - Most web applications would like to do cross-domain information retrieval in a single query. E.g. show me the 401K benefits and the life insurance benefits status for a particular employee. Most systems would require splitting this request into multiple queries, running them against the backend repository and aggregating the results. With XDisect, it can be done in the same query. XDisect will search across domains for the requested information, thus providing very powerful and flexible information retrieval opportunities at a relatively low cost.

Please note that this is just a scenario and any resemblance to a live project is purely unintentional. This scenario provides only one instance of how XDisect can be used in a market maker/community application. Please contact us at chetan@pybiz.com if you have any questions about the applicability of XDisect in your problem domain

# Overview

In this section we provide an overview of the scenario and a basic flow for it

## *Scenario*

Lets take the scenario of an employee benefits portal/community called BenefitsOnline. (I hope this concocted name does not infringe on anybody. These days all “e”s and “online”s are taken :).

BenefitsOnline is an online community where users can get complete information on all of their benefits (obtained through their employers or direct). The community has newsgroups, online chat groups, etc where members can discuss benefits issues, and services & can get expert advice on the same.

BenefitsOnline also has calculators that can guide a user through the tricky process of figuring out how much their benefits are worth and what is the impact of changes in their benefits packages. It also offers similar services to employers that would like to understand the impact on their monthly payroll charge.

BenefitsOnline also offers fee-based services like 401k management, stock participation plan, medical and dental coverage, life insurance, etc.

BenefitsOnline through its partners also provides employees discounted tickets for various entertainment activities (e.g. ski trips, movies, resorts, car reservations, etc).

In addition BenefitsOnline also facilitates the online discounted purchasing of products for corporate employees. E.g. some corporate customers will offer their employees their own internal products and products from their partners at a discounted price. BenefitsOnline helps customers manage these offerings by hosting them, taking the orders online and by collecting the necessary payments from the employees and or the service providers that deal with the employees.

BenefitsOnline’s customers include end users and corporations that want to outsource their benefits management & related services.

BenefitsOnline has numerous partners that help provide this comprehensive set of benefits related services to the user community.

BenefitsOnline will provide content and services to wireless users (with PDAs, cell phones, etc) in addition to the standard desktop browser based users as well.

**Note:** since BenefitsOnline caters both to Mr. John Doe on the street and to corporate customers they could classify both as a B2B marketplace/community and as a B2C portal.

## *Basic Flow: -*

The basic flow of the community will be as follows: -

1. Service providers wanting to offer services to BenefitsOnline’s customers will register with the community. Once the registration happens, service providers only need to register new services or changes to their existing services with the community.
2. Customers that want to use the BenefitsOnline services will be allowed to browse through their offerings (a catalog of sorts) that are cross-vendor and maintained by BenefitsOnline. Of course the customers also have a suite of free services that they can use at the community.
3. If a customer wants to buy a service, BenefitsOnline will help the user find the best service provider.
4. Once a service provider has been selected, BenefitsOnline will gather information known about the user so far.
5. It will then redirect the user to the service provider’s web site securely along with the background information about the user & their current session, so that the service provider does not ask the user the same questions again.
6. If required the service provider may ask BenefitsOnline to gather more information on their behalf (in order to provide customized services to the user).
7. The service provider will then provide the service and bill the user if applicable.
8. The user will then get transferred back to the community menu for other content and related services.
9. A record of the services rendered by the service provider site will be sent back to BenefitsOnline for their financial reconciliation.
10. BenefitsOnline may use this usage report to potentially get payment from the user (or their company) and then pay the service provider.
11. Optionally BenefitsOnline may do the reverse, let the service providers take care of payments and have them provide a monthly commission payment back to BenefitsOnline.

## *Assumptions: -*

- BenefitsOnline has already gone off and signed up hundreds of service providers to provide services to its user community.
- It has hundreds of thousands of users registered for services (direct or via its corporate customers).
- BenefitsOnline makes money by charging a certain percentage per sale successfully facilitated by the portal/community. Of course they charge for advertising on their

website. Another revenue source for them is to have annual contracts with corporate customers to provide their employees whatever services they need or get paid based on their usage.

- BenefitsOnline could issue interested users a BenefitsCard. This card represents an electronic wallet of sorts that users can use to purchase and use services within the community/portal and also serves to identify the end users.

### *Notes: -*

- In general BenefitsOnline supports both a facilitator model and an intermediary interaction model. In the facilitator model the actual transaction does not necessarily go through the community. The user and the service provider may do the actual financial transaction separately. BenefitsOnline just gets the buyer and seller together. In the intermediary model, BenefitsOnline sits in the middle as a broker and isolates the buyer from the seller (for anonymity and other reasons).
- If the community has a network of sites, then some of these other sites could also be considered as service providers.
- As you might have guessed, the same above scenario also applies to an online marketplace.

## Scenario Background

In this section we provide the basic technical background for the scenario

### *Assumptions:*

1. BenefitsOnline is implemented using XDisect & CEF.
2. BenefitsOnline has a profiling system as one if its integral portal components. This system will maintain minimal profiles of customers and service providers doing business with it. Assume for simplicity that these profiles are stored in XDisect.
3. Product & service listings from service providers are also stored in XDisect (as a part of the same service provider profile).
4. All data exchanged between BenefitsOnline and its customers and service providers is in a business XML format.
5. Now lets make the most disturbing assumption, there is no comprehensive schema or strict vocabulary that service providers of BenefitsOnline need to adhere to in order to participate in the community. There may be a minimal base set of attributes that each needs to have in their messages in order to do automatic processing but no one is dictating that the attributes be exactly in a certain pre-defined structure.
6. Service providers will share only as much information with BenefitsOnline as is needed. E.g. some service providers may choose not to share price and availability information of their services.
7. BenefitsOnline trusts the service providers to varying levels and hence it will share varying amounts of information about the customer and their requests with service providers.

8. Similarly customers will also provide only the minimal set of profile that is required to do business and for personalization. We do not want to have the customer to fill out a registration form 20 pages long, just to get some basic information on benefits from the portal.

9. Both service providers and customers have already setup pre-negotiated contracts with BenefitsOnline. Here we will not discuss the actual process of customer or service provider registration.

10. BenefitsOnline has all of the state of the art security in place but we will discuss security only to the extent it applies to this scenario.

11. All customer, service provider and portal communications are implemented on top of the HTTP (S) protocol.

12. BenefitsOnline tries to be neutral to both buyers and sellers (hence it is neither buy-side nor a sell-side marketplace or community). Although BenefitsOnline will have its preferred set of vendors and partners.

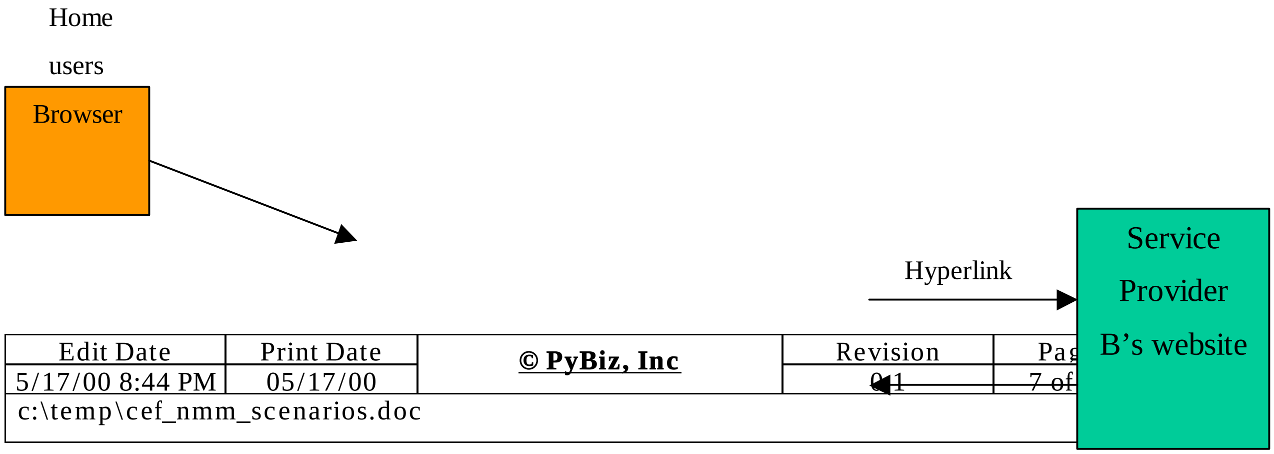
13. Service providers can interface with their legacy procuring and catalog systems using their own choice of technology. They are in no way bound by the choice of technology for BenefitsOnline.

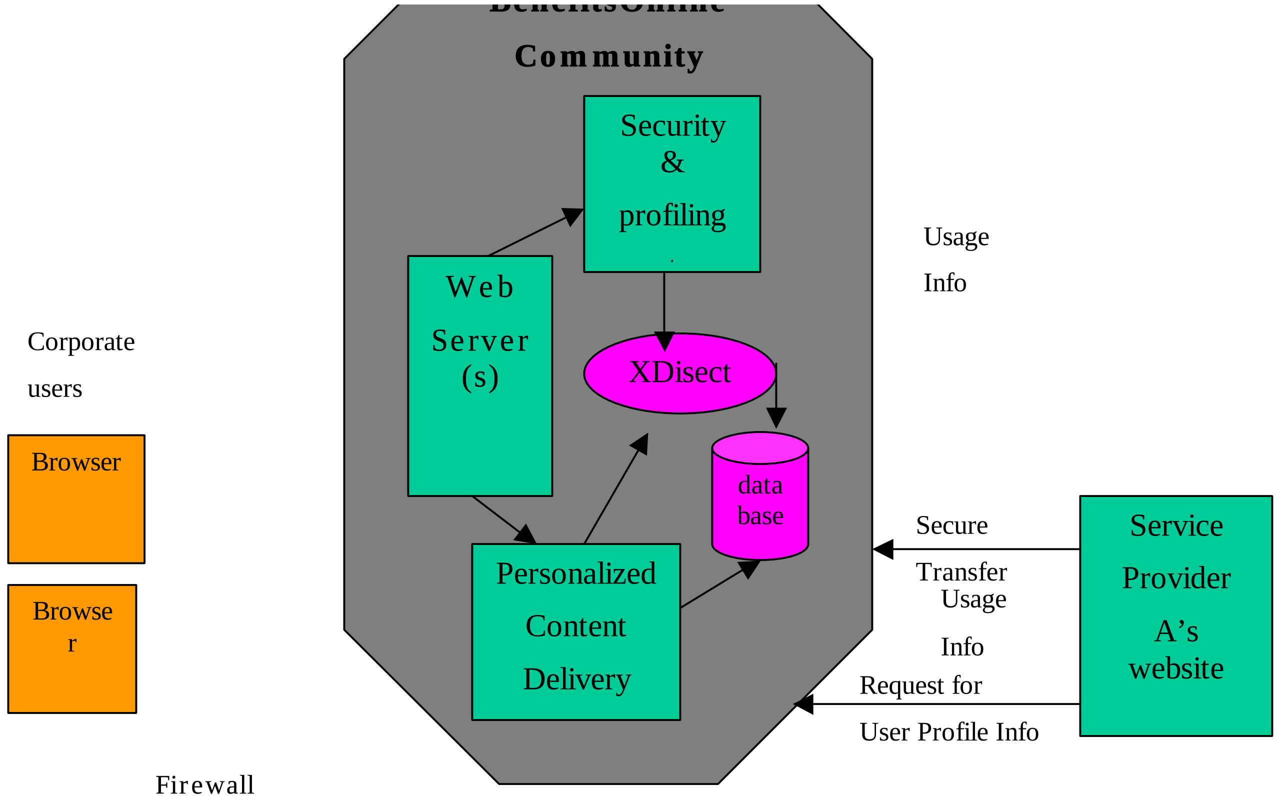
14. Service Providers can start off by just having a basic web site. BenefitsOnline will just hyperlink to their home page from its main menu. As they realize more value, service providers may choose to integrate more features like secure transfers and acceptance of the BenefitsCard (electronic wallets).

15. Customers just need a web browser to interact with BenefitsOnline. Corporate Accounts may elect to integrate their procuring and supply chain systems. Even in that case the integration will be kept simple to the extent that they just download a dump of the purchase orders in a format like XML and then integrate that data into their backend systems. The dump can be downloaded either via ftp or http, hence allowing automated tools to do this on a daily or weekly basis as is convenient for the customer.

16. The portal has means of managing the BenefitsCard accounts for the users. It keeps track of the payments made and the available currency on the user's BenefitsCard Account. If the user's BenefitsCard is nearing depletion or does not have sufficient funds to use a particular service, then the user will be prompted to add more money either via credit card or some other means to their BenefitsCard account. Thus for all practical purposes the BenefitsCard can be though of as another form of the standard pre-paid cards.

## *High Level Architecture*





## *Data Structures*

Below are basic samples of data structures that will be used by the community for exchanging information back and forth between the various participants

## Service Provider Profile

Below is a sample of a service provider's profile as maintained by BenefitsOnline
NOTE: A profile can be thought of as an arbitrarily nested data structure that can store whatever is relevant to the service provider/customer and to BenefitsOnline

```
<company>
        <id>chetaninc</id>
        <role>seller</role>
        <name>ChetanInc</name>
        <registered_address>
                <street>3386, Valley Forge Way</street>
                <city>San Jose</city>
                <state>CA</state>
                <zip_code>94087</zip_code>
                <telephone_number>408 733 8463</telephone_number>
         </registered_address>
         <billing>
              <type>credit-card</type>
              <type>BenefitsCard</type>
              <type>SAP direct</type>
         </billing>
         <service_details>
               <discount_percent>10</discount_percent>
               <service_hours>24X7</service_hours>
               <min_turnaround_days>3</min_turnaround_days>
        <service_request_url>
            http://www.chetaninc.com/benefitsonline/cgi-bin/receiveRequest.cgi
        </service_request_url>
        <required_attributes>
              <attribute_name>user_id</attribute_name>
              <attribute_name>company_id</attribute_name>
              <attribute_name>company_name</attribute_name>
              <attribute_name>return_url</attribute_name>
              <attribute_name>session_token</attribute_name>
        </required_attributes>
        <service>
          <id>bonline-401k</id>
          <title>401K management</title>
          <desc>
              Comprehensive 401k management services for corporate and end users
          </desc>
          <required_attributes>
               <attribute_name>ssn</attribute_name>
          </required_attributes>
        </service>
        <service>
          <id>bonline-gul</id>
          <title>Group Universal Life Insurance</title>
          <desc>
               Comprehensive life insurance services for corporate and end users
          </desc>
          <required_attributes>
               <attribute_name>insurance_id</attribute_name>
          </required_attributes>
        </service>
    </service_details>
    <preferences>
        <service_countries>Italy</service_countries>
        <service_countries>UK</service_countries>
        <service_countries>US</service_countries>
        <currencies>lire</currencies>
        <currencies>pound</currencies>
        <currencies>dollar</currencies>
        <notify>
           <event>order_amount GE 100000</event>
           <receive_url>http://www.chetaninc.com/notify/receiveEvent.cgi</receive_url>
           <receive_mechanism>asynchronous-send</receive_mechanism>
        </notify>
    </preferences>
</company>
```

- XDisect provides the ability to associate lightweight events with attributes within profiles as in the company structure above.
- XDisect provides the ability to store and search arbitrarily nested and complex data structures without having to pre-define any schemas, DTDs or vocabularies
- CEF allows variable sets of attributes to be shared between the portal and the service providers which can be determined dynamically on a per service provider basis and a per transfer type basis

### Notes: -

The key things to note here are: -

- the profile may have the company's role since they could be both a service provider and a customer.
- We have not shown the security privileges for this account. These could be stored as a separate XML document related by role.
- In the preferences section, the service provider has specified an event. The service provider would like to be notified if there is any potential user that is planning on an order amount more than 100,000. Perhaps they would like to give the customer special treatment (expedited delivery, etc). Since it is a large order, the service provider may want to manually verify the authenticity of the order and the customer before agreeing to fulfill it
- Of course this profile could have been normalized further so that the preferences are stored separate, etc. But we wanted to show the relationships between the various sections of the profile and hence left it all in one profile document
- We have lumped the service details along with the rest of the company profile. Some portals might choose to keep that in a separate document
- The required_attributes section under service details contains customer attributes that the service provider is interested in receiving when a user is sent to the site. In addition the profile allows the service provider to specify the required attributes on a per service basis. Thus in the above profile, the insurance service requires an additional insurance id to be passed whereas the 401k service requires the user's social security number to be passed in addition to the general user details. This information can help the service provider site customize the user experience to the user's needs. Of course the amount of data shared with the site will depend on the relationship BenefitsOnline has with that site and also the preferences of the user.

## **Customer Profile**

Below is a sample customer profile for a customer visiting the site. We would like to stress that this is just a sample. The profile will contain dynamic and static sections, the attributes of which will depend on the real business needs of the community and the willingness of the customer to divulge this information. The static portions of the profiles are likely to have been collected from the customer during the initial registration process. The dynamic sections like preferences are gathered over time as the user visits more sites and uses additional services

```
<user>
        <id>chetan</id>
        <role>hr manager</role>
        <name>Chetan Patel</name>
        <email>cpatel@juno.com</email>
        <company>
            <id>CoolTechnologies</id>
            <name> Cool Technologies Inc </name>
        </company>
```

```
<billing>
    <account>
        <type>credit-card</type>
        <number>4128 0036 5246 9245</number>
        <expiry_date>10/2000</expiry_date>
    </account>
    <account>
        <type>BenefitsCard</type>
        <number>41280936 5266 </number>
    </account>
    <default_account>BenefitsCard</default_account>
</billing>
<wireless>
   <device_type>PDA</device_type>
   <form_factor>4”x3”</form_factor>
   <color_enabled>no</color_enabled>
</wireless>
<preferred_services>
    <service>
       <id>bonline-401k</id>
       <title>401K management</title>
       <desc>
          Comprehensive 401k management services for corporate and end users
       </desc>
       <required_attributes>
          <ssn>409 445 7764</ssn>
       </required_attributes>
    </service>
    <service>
       <id>bonline-gul</id>
       <title>Group Universal Life Insurance</title>
       <desc>
          Comprehensive life insurance services for corporate and end users
       </desc>
       <required_attributes>
          <insurance_id>4094457764-80761</insurance_id>
       </required_attributes>
    </service>
 </preferred_services>
 <preferences>
    <notify>
       <event>service_provider*service.id CONTAINS “opera tickets”
               AND
               service_provider*service.price LE 25
       </event>
       <receive_mechanism>email</receive_mechanism>
    </notify>
    <localization_language>US-English</localization_language>
    <notify_new_products>yes</notify_new_products>
    <receive_weekly_newsletter>no</receive_weekly_newsletter>
    <bookmarks>
       <category>
          <name> my 401k related urls </name>
```

```
                <url>http://www.bonline-401k.com/</url>
              <category>
              <category>
                <name> my cool technology urls </name>
                <url>http://www.pybiz.com/</url>
                <url>http://www.pybiz.com/products/XDisect/</url>
              <category>
           </bookmarks>
        </preferences>
  </user>
```

- XDisect provides the ability to associate lightweight events with attributes within profiles as in the preferences section above.
- XDisect allows user profiles to grow as complex as is required to meet the business needs without requiring any changes other than to the front end GUI
- CEF allows for selective sharing of user information between the customer and the service provider

**Notes: -**

- Customer has indicated in the wireless section that he would like use his PDA to access the portal
- The preferred services section contains information about the services the customer has indicated that they would like to see. Of course this would be in addition to the standard free services and perhaps the services available via the customer's role as an HR Manager. Since some of these services will require additional information about the user, the first time the user visits the site, this information is collected on behalf of the service provider by the portal and stored in the user's profile for subsequent use
- In the preferences section the  portal allows the user to store their bookmarks organized the way they want them.
- The preferences section also contains a trigger that the user has specified. This user would like to be notified if some service provider is providing opera tickets for less than 25 dollars and the notification should be via email.

## Usage Report

The service provider sends a usage report to BenefitsOnline. It indicates what services the service provider has provided to the user. BenefitsOnline will use this information to potentially do corporate billing and pay the service provider or if the service provider is taking care of the billing, then it might use this report to keep track of how much the service provider owes it.

```
  <usage_event>
       <service_provider_id>chetaninc</service_provider_id>
       <user>
```

```
        <id>chetan</id>
        <company_id>pybiz</company_id>
    </user>
    <service>
         <start_datetime>2000-05-10 : 10-15AM</start_datetime>
         <end_datetime>2000-05-10 : 10-15AM</end_datetime>
         <id>bonline-gul</id>
    <service>
    <service>
         <date>2000-05-10</date>
         <units_consumed>45</units_consumed>
         <id>bonline-401k</id>
    <service>
</usage_event>
```

➢ XDisect allows for different taxonomies for the same class of objects. Here the two service descriptions have completely different sets of attributes, and yet both represent information about a service. XDisect will allow both of these structures to be stored and searched with no additional effort for schema definition

**Notes: -**

- We are assuming that certain services will be billed based on the amount of time they were used for and other service could be based on the units consumed. E.g. a BenefitsOnline benefits tutorial class service might want to charge the users based on the amount of slides they have viewed or based on the duration they have viewed it for.

## Detailed Use Cases

In this section we describe some detailed use cases for the above scenario

### *Service Provider A wants to post/change a service offering*

Service Providers of BenefitsOnline will periodically want to modify existing offerings for discounts, etc. In addition they will also want to add new offerings to their existing set and eventually will also want to revoke certain offerings. Below are some steps describing this process

1. Service Provider A decides they want to add a new offering or change an existing one
2. Service Provider A visits the BenefitsOnline website using a standard web browser
3. Service Provider A does a login into BenefitsOnline 's website (perhaps by filling out a user/ password form or by presenting a digital certificate)
4. BenefitsOnline (portal) forwards this request to its security & profiling system
5. Portal Security & Profiling System requests profile information for this user from XDisect (based on whatever criteria is applicable)
6. Portal Security & Profiling System authenticates Service Provider A against the profile

7. The portal then requests the personalized content delivery system to provide content to be displayed to this user
8. The content system determines service provider A's role and preferences from the profile (maintained in a session file) & uses this to personalize information to the user
9. The portal then displays this personalized content to Service Provider A
10. Service Provider A navigates through the web pages to the add/change service offer online form
11. Service Provider A then fills out the offer details and clicks the submit button
12. The portal collects the form information and automatically converts that into an XML document.
13. The portal then sends this document to the XDisect engine
14. XDisect engine stores and indexes the service offer document and acknowledges receipt. In this case it will store the service offer along with the rest of the profile
15. If any events are registered for this change or addition of the service offer, then XDisect will generate the triggers. E.g. notify a customer if a service provider is offering insurance services at a certain price
16. The portal sends back an acknowledgement with a receipt id back to Service Provider A

- XDisect provides an open HTTP/XML interface, which makes it convenient for service providers to directly upload their offer documents into the BenefitsOnline database.
- XDisect & CEF's HTTP/XML interface allows suppliers to participate in the community without having to accept and integrate any software from BenefitsOnline
- XDisect allows the service providers to post service offers with varying descriptions and yet guarantees that all of them will be considered during an offer selection
- XDisect provides the ability to do personalization of user interface by filtering content based on the user's profile

**Notes: -**

- Rather than filling out the online form, some Service Providers might prefer to directly generate the XML document representing the offer and send it over to BenefitsOnline, either via a pre-designated ftp site, an http post, a CGI form upload or email. This is useful for integration with their legacy systems, so that whenever a new product shows up on their product list database, they can automatically generate an XML file representing the product offer and upload it to BenefitsOnline. This will be extremely useful if the same service provider is working with multiple marketplaces or communities.

- If the Service Provider is not registered, then the login screen could redirect the Service Provider to a registration request form.
- It is assumed that during login session establishment happens between the service provider and BenefitsOnline.
- Since the user profile needs to be used several times, the XDisect engine will cache the profile by user, so that the next time the profiling system requests the profile, it can be presented quickly from cache.

## *User browses BenefitsOnline services*

A user visiting the BenefitsOnline site will want to browse through its offerings and may potentially elect to buy some service. This use case describes in detail the process the user would go through while browsing the BenefitsOnline site and its customized services. In this use case we skip the user authentication and personalization details since they are covered in the previous scenario: -

1. User visits the BenefitsOnline portal (portal) using a standard web browser.
2. User does a login to the portal (perhaps by filling out a user/password form or by presenting a digital certificate).
3. Portal (using its underlying security system) authenticates the user by looking up their internally maintained profile and validating their credentials. This may involve invoking the XDisect engine to get the profile.
4. Portal's profiling system then extracts the user's preferences from the profile.
5. Portal's personalized content delivery system then queries the XDisect engine for all service offerings that match the user's preferences.
6. XDisect engine searches through all the service offers of all service providers, matches the service provider preferences against the user's preferences and returns the resulting matching services. Portal or the user can specify the order or sorting sequence for the resulting offers (for presentation).
7. XDisect engine then applies the portal's business rules to the resulting offers to filter out offers or service providers that do not match.
8. XDisect engine then massages this information as per the specified sorting, ordering and transformation requirements.
9. XDisect engine then paginates this information as per the specifications of the portal (e.g. send over only 25 services at a time).
10. XDisect engine then invokes the pre-specified portal callback module with the results.
11. The portal's callback module may convert this XML information into html using XSL or some other means for presentation. This may involve determining the form factor of the device being used and formatting the content according to the device (e.g. PDA or Desktop PC Browser, etc).
12. Portal then presents this personalized information along with the other standard menu pages to the user.
13. User navigates through the offerings using the web browser.

14. User can either drill down through the offerings or may just search it (using an XDisect powered flexible search) to get to their desired service.
15. User then goes ahead and clicks on the hyperlink presented for the service. This is where a session transfer will happen to the service provider’s site.

- XDisect provides the ability to do personalization of user interface by filtering content based on the user’s profile
- XDisect provides the ability to retrieve service provider offers and match against them, even though they may be stored in different taxonomies or structures
- XDisect provides for cross domain queries, so that retrieving information about insurance services and 401K services for a user can be done in a single query even though the information may be stored in different records in the repository

## Notes: -

- Since the XDisect generated personalized content could contain large amounts of information, BenefitsOnline software may want to receive the data in specified page sizes.
- Since this personalized content generation needs to happen frequently, XDisect will cache this information on a request basis. In addition for better performance, BenefitsOnline front-end software itself may also want to cache the generated personalized HTML by user and service provider.
- For advanced users, rather than navigating through the offerings pages, they may want to just specify a search query.

## Assumptions: -

- BenefitsOnline has pre-defined the rules for presentation formatting of the data. Of course these could be specified with every request, but could turn out to be a performance bottleneck
- BenefitsOnline could pre-define the rules for sorting and ordering of the result sets. Of course these could be specified with every request or set as preferences by the user

## *BenefitsOnline Transfers the user to the partner/service provider site*

When a user elects to get more information or to order a service, BenefitsOnline will transfer the user to the actual service provider or partner site that provides this service. This use case describes the steps involved in this process

1. User clicks on the hyperlink representing the service they want to view.
2. BenefitsOnline (portal) then invokes its CEF send script to transfer the user securely to the remote site.

3. CEF send script requests relevant session information about the user from the XDisect engine (including user_id, site_id, receive url, etc) or the security system.
4. CEF send script requests the XDisect engine to determine what user information needs to be sent during the transfer.
5. XDisect engine looks up the user’s profile and the service provider’s profile and does a match to arrive at the set of attributes that need to be sent during the transfer (refer to the required_attributes section in the service provider’s profile).
6. CEF send script marshals this information.
7. CEF send script encrypts the information using either Symmetric or Asymmetric encryption (via certificates or just a shared secret) as negotiated by BenefitsOnline and the service provider.
8. CEF send script then redirects the user’s browser to the service provider site’s CEF receive script (CEF Receive) along with the above message payload.
9. CEF Receive unmarshals the received user information & decrypts the payload.
10. CEF Receive extracts session information from the payload & authenticates the user locally. NOTE: The method of authentication is not dictated by CEF. It depends on the receiving site’s implementation of authentication.
11. CEF Receive then establishes a local session for the user once the authentication passes. NOTE: CEF does not dictate what should be the method of session establishment. Whether it is setting a cookie, setting some variable in a hidden frame, using the remote_user variable in the CGI environment, etc. This will depend on the web site implementation of the remote service provider.
12. CEF Receive then transfers the user internally within their site to the relevant web page.
13. User is now ready to use the remote service provider’s service

- CEF provides single sign on. A user visiting various sites within a portal or a community does not have to login at every site
- CEF transfers from the portal to a service provider’s site are secure. This prevents eavesdropping or stealing of user session information in midstream
- CEF allows for selective sharing of user & session information between the portal and the service provider
- XDisect enables retrieval and query of profile information using flexible query mechanisms
- Since CEF transfers are compact with minimal information being transferred, it can improve the performance of the actual transfer.
- Since CEF transfers contain bare minimum information about the user, even if a particular transfer is compromised, the security risk is very minimal

## Assumptions: -

- In the above use case we are assuming that the service provider has agreed to receive a secure transfer & has integrated the CEF receive script into their infrastructure.
- The service provider has sufficient information to receive the user and service their request. The next 2 use cases explain the steps that take place if the service provider needs more information about the user to fulfill the request

## Notes: -

- As you might have already observed, this entire process of secure user transfer from one website to another is totally transparent to the end user. As far as they can see, they clicked on another hyperlink that took them to a different web page.
- Since the secure transfer to the remote site is totally transparent, it makes it easy for BenefitsOnline to maintain a consistent user experience across different portal sites.

## *Service Provider requests more user information from BenefitsOnline*

Once a user is transferred to the service provider, chances are that the information provided may not be enough to service the user. The service provider may determine that this user does not exist in their local repository. They might want to double confirm with BenefitsOnline that this is a valid user or they might want more profile information about the user.

1. Service Provider determines that they need additional user information from BenefitsOnline in order to service the request.
2. Service Provider then invokes the local CEF Portal Client script to request the desired information.
3. CEF Portal Client script then invokes the remote BenefitsOnline CEF portal script (CEF Portal script) and passes it an XML document representing the request for more profile information along with the required credentials of the service provider.
4. CEF Portal script then authenticates the service provider (using the credentials provided).
5. CEF Portal script formulates a query for the desired additional user information to the XDisect engine.
6. XDisect Engine then matches the request for additional information to the user's allowed information rules and with BenefitsOnline's own rules on information sharing with this service provider. It then returns the matching user information to the CEF Portal Script.
7. CEF Portal Script then forwards this information to the Service Provider CEF Portal Client Script.
8. Service Provider CEF Portal Client script then updates the service provider's local repository with this information.  This depends on the local caching or information sharing rules established between BenefitsOnline and the service provider

- CEF allows portals to selectively share information with service providers and vice versa.
- CEF is based on an open HTTP/XML based protocol. This enables service providers to quickly integrate it into their websites with minimal intrusion into their existing implementations
- CEF provides a convenient and secure back channel for service providers to hold additional conversations with the main portal site. This can be useful for sending requests for additional information to the portal, etc
- CEF can be easily integrated and implemented on most web technologies and platforms

**Assumptions: -**

- BenefitsOnline already has information about the user stored in the user profile. If BenefitsOnline itself does not have the request information about the user, the next use case covers how CEF collects it on behalf of the service provider

**Notes: -**

- The service provider in step 2 above does not need to invoke the local CEF Portal Client script for sending the request back to BenefitsOnline. They could just as well have adhered to the CEF protocol and sent a direct XML message over http to the CEF portal script
- As you might have already observed, this request for more user information is totally transparent to the end user as it happens over a back channel directly between the service provider and BenefitsOnline.

## *BenefitsOnline collects more information on behalf of the Service Provider*

In the previous 2 use cases we covered how a user gets transferred and how a service provider can request BenefitsOnline for more information. Now lets take the case of what happens if BenefitsOnline does not have the requested user information. E.g. BenefitsOnline transfers the user for the first time to the 401K-management web site. The 401K–management web site requires all users to have their SSN available. Since an SSN is something that could be used by other sites BenefitsOnline may want to collect this information. Also rather than each and every service provider site having to worry about popping up different types of registration forms to collect this type of information and also worry about its maintenance, they might want to delegate this responsibility to BenefitsOnline, which is already doing this kind of registration.

Another scenario where this might come in handy is if the service provider collects session information about the user via their actions. Now the service provider would like to make this available to other sites within the portal infrastructure (not just to any random site). This could be useful for other sites within the portal umbrella to personalize their experience for the user.

The use case below describes how this is done using CEF: -

1. During a user transfer, Service Provider detects they need more information about the user.
2. Service Provider holds a back channel communication with BenefitsOnline to get the additional information.
3. BenefitsOnline indicates to the Service Provider that the user should be transferred back to the portal for the additional registration.
4. Service Provider invokes the local CEF send script to transfer the user back to the portal (along with a reference to the url within the service provider that the user was wanting to visit).
5. CEF Send now does a secure transfer back to BenefitsOnline by invoking its CEF Receive Script (CEF Receive).
6. CEF Receive now locally transfers the user to BenefitsOnline's registration subsystem along with a list of attributes or additional information that needs to be collected about the user.
7. The Registration Subsystem dynamically generates a form and presents it on the user's browser requesting the user to fill it out.
8. The user fills out the additional information and submits it back to the registration subsystem.
9. Registration Subsystem requests the XDisect engine to store this information in the user profile
10. Registration Subsystem now invokes the BenefitsOnline CEF Send script to transfer the user back to the service provider along with the requested additional information. This is the regular secure transfer as before

- ➢ CEF enables service providers and the portal to dynamically collect information about the user.
- ➢ CEF makes it possible so that users need to provide only as much information as is needed to provide services.
- ➢ CEF also allows the user information to be collected if and when it is needed rather than making users fill out lengthy forms with questions that may not be relevant to all users at the time of registration.
- ➢ CEF allows the portal to collect information on behalf of the service providers. Hence it can eliminate duplication of registration and user management functionality at each service provider site reducing the service provider's overhead and letting them focus on their core value add.

## *Service Provider sends a usage report to BenefitsOnline*

Once the service provider has provided the service to the user, they are expected to send a usage report back to BenefitsOnline. This allows BenefitsOnline to take care of any backend payment reconciliation and other processes for the users. E.g. if the user is a corporate user, then BenefitsOnline will want to present the company, a monthly report of the services used by their employees at the BenefitsOnline portal. This can then be used to secure payment

from the customer company and also pay the service providers. On the other hand, if the service provider is doing the billing, then this can help both the service provider and BenefitsOnline keep track of how much they owe each other for referral fees, etc.
The use case below explains the steps in detail: -

1. Service Provider provides the necessary service to the BenefitsOnline User.
2. Once the user leaves the site, Service Provider now collects information about the service provided.
3. Service Provider invokes the local CEF Portal Client Script with the usage report.
4. CEF Portal Client Script invokes the remote CEF Portal script on the BenefitsOnline site with the usage report (using HTTP/XML).
5. BenefitsOnline’s CEF portal script receives this information and invokes XDisect to store this information in the repository associated with the user.

> ➢ CEF allows service providers and the portal to have as much interaction as is required to do business. Service Providers can start off with no interaction other than the basic user transfer the Portal. As they derive more value from the portal and are willing to invest more, they can add more support for getting and collecting user information from BenefitsOnline and sending reporting information back to the portal.
>
> ➢ CEF enables service providers to report usage information about users back to the portal.

**Notes: -**

- Since interaction between the service provider and BenefitsOnline happens over a back channel using http and XML, it is transparent to the end user.
- The above use case is similar to the use case on the Service Provider requesting more user information from BenefitsOnline. The only difference is the type of message being sent to BenefitsOnline. Hence the service provider can have such similar conversations with the Portal with very minimal effort on both sides.

## *User bookmarks the service provider site and directly visits it the next time*

Another case is that the user bookmarks the service provider site. Then the next time the user can visit the service provider directly and bypass the portal altogether. This could have problems during reconciliation, because when the service provider sends over a usage report, BenefitsOnline has no recollection of having transferred the user over. In addition now the portal’s value-add as the entry point can be challenged.
Below is one possible implementation that CEF supports to solve this problem: -

1. User visits service provider web site directly via a bookmark on their browser.
2. Service Provider web site’s SAPI (ISAPI, NSAPI, ApacheAPI) plugin (if available) will detect that the user does not have a session.
3. Optionally all of the requested modules in the website could potentially be checking for session availability via a common session detection module.

4. In either case the web site detects that the user does not have a session & can invoke the local CEF send module (CEF Send).
5. CEF Send redirects the user's browser using http to the portal's login script (along with a reference to the url where the user should be sent back).
6. Portal's login script detects that the user is not logged in and hence presents its standard credentials acceptance screen.
7. Portal authenticates the user as before.
8. Portal now detects that the user was sent over from a remote service provider site.
9. Portal directly transfers (using CEF Send) the user back to the service provider URL they had come from, instead of presenting them with their standard personalized menu pages.
10. Service Provider can now receive the authenticated user and service their requests.

**Notes: -**

- As you might have already observed, this entire user transfer process between the different web sites is almost transparent to the user. All the user sees is a login page popup and then they are sent to the web page they want to go to.

## *User visits another site from the service provider's site*

Lets say the user is using the services at the service provider's site. Then they see this link on the service provider's site for related information, which sends them off to another site. But the problem is now the user can leave the community or the portal has no track of where the user was transferred.
Below are some detailed steps on one possible solution for this use case: -

1. Service Provider detects that the user has clicked on a URL representing a different site within their site.
2. Service Provider then invokes the local CEF Send script with the URL the user was trying to visit.
3. CEF Send script transfers the user back to the BenefitsOnline Portal CEF receive script (CEF Receive).
4. CEF Receive determines that the user was trying to visit a new site and invokes the portal CEF Send script.
5. CEF Send determines whether the site being requested is a part of the community or not (by using XDisect to query its repository to determine if the service provider exists in the Community).
6. If the Service Provider is not a part of the community then: -
    a. CEF Send pops up a warning page to let the user know that they may be leaving the community.
    b. CEF Send then opens the new site in a new browser window (so that in order to come back all the user has to do is close the browser window).
7. If the Service Provider is a part of the community then CEF Send does a normal secure transfer as explained previously.

Last modified: May 9, 2000
www.pybiz.com