Business Use Case – **Extending visibility to Business Processes participants**

Use Case Identification -

Created By:	IBM
Last Modified:	06/18/12
Version	0.6

Actors & Participants

An Actor is an individual (or sometimes role) who interacts with an application. This person has a login on the application and does work on that system.

A Participant is an individual (or sometimes role) who is directly a part of a business process that the above-mentioned application supports, but is not an actor (does not have a login to the system).

Primary Actor: **Requisitioner** – Primary party participating in this part of the Procurement process. Has access to Purchase Ordering system information and is responsible for placing orders and taking action.

Secondary Participant: **Requestor, Requestor's Manager –** Interested parties and consumer of information about a Procurement. Lacks direct access to information but needs to know about business process status or key events coming out of business applications.

Tertiary Participant: **Buyer** – An uninvolved party and consumer of Procurement information. Has potential to add value to the process.

Definitions

Procurement Process - Business process for the acquiring of appropriate goods or services at the best possible cost that meets the needs of the purchaser.

Purchase Order system – Well established business application that is used to track and execute Purchase Orders. Access to a PO system may be limited to Requisitioners and Buyers where Requestors and other interested participants must work with their Requisitioner to get procurement status.

Activity Stream – An activity stream is a list of recent activities generated around a particular work task or artifact or process. Each activity in an activity stream is represented by a standard JSON (or any mark up language) format.

OpenSocial Gadget – a user interface component that provides a small, granular view of a particular aspect of an application.

Embedded Experience – A contextual OpenSocial Gadget that is used as an alternative representation for an event (for example, an activity within an Activity Stream). This embedded application allows users to interact with applications without switching contexts to the underlying application.

Background & Business Goals

Many business applications use strict access controls where only a limited number of users are allowed access. However, any given business process (like a Procurement Process) may cross multiple application boundaries and involve any number of hand-offs. So at any given time there may be participants in a business process who lack visibility or access to information related to their procurement because of access controls or other usage barriers on a business application.

As a specific example, this use case examines a procurement process where the Requestor of a good (and other parties) lacks access to the Purchase Order system where their request is being processed. The Requestor is an

example of an interested party that needs to know some information about the status of a Purchase Order so that she can perform appropriate project planning.

Concept of Operations

1.1. Current Service or Solution

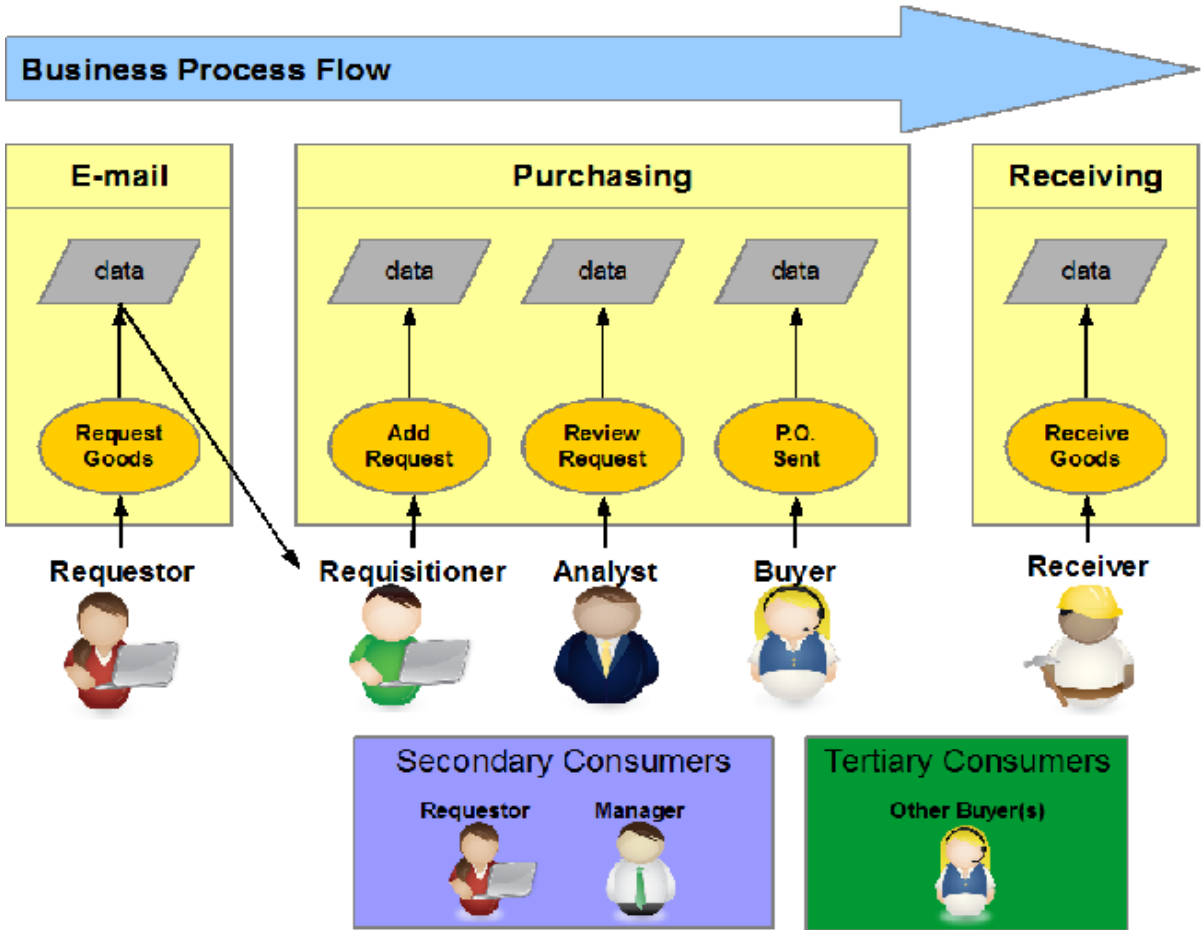
If information about participants for a particular Procurement is available a Purchase Ordering system could generate e-mail notifications on state changes. This information is often stale the moment after it is sent for fast moving processes. However, it is often impossible to identify all interested parties ahead of time because that list changes or become conditional (for example, a manager might only become involved if an order is delayed or if a project's schedule has been shifted)

Another method is to attend status meetings or contact and work directly with a Requisitioner or some other party with direct access to the information about the process that needs to be known more widely.

Also, by only allowing "interested" parties to view the status of a procurement, you miss out on potential significant value that is created when that information is made widely visible. What if somebody not directly involved knows of a lower cost or better alternative to what is being ordered? If they aren't directly involved in the original process then they would never be able to provide the suggestion that would create savings.

The business process workflow may transition through systems unfamiliar to interested participants. Simply providing participants access to these systems wouldn't help them use these systems or interpret the information that they are seeing.

1.2. Desired Social Business Implementation



Instead of forcing users to seek out information, a better solution allows information to come to them. We propose populating a user's activity stream with activities generated from different business processes that are interesting to them. The activity stream becomes the aggregator of concurrent business processes and the neutral ground on which all parties can come together and collaborate. Embedding applications into the activity stream makes it easy to provide a user experience that is more focused by using individual activities within an Activity Stream as the context for accessing backend applications.

A Requestor that contacts a Requisitioner about a procurement would get activities posted to their activity stream as a PO gets submitted and is worked through the PO System. While the Requestor has no access to the PO System, they receive this activity because they are identified as playing a role in the Business Process. This activity can include an OpenSocial Embedded Experience that is used to get dynamic status information that is scoped to the visibility appropriate to the viewer. This scoping and visibility can be managed via OAuth or through traditional access control models. For example, a Requestor may only be able to view the estimated arrival date of their order but not details like the financial information related to the order that may also exist in the PO system. The Requestor no longer has to wait for a status meeting or get time with her Requisitioner in order to get updates on her order, the information comes to her.

Tertiary parties, like a Buyer, can also tune into Procurement activities within his activity stream. They also would receive scoped visibility appropriate for them but can use the Social platform to be a neutral ground for some social collaboration. The Buyer could comment on an activity for a new PO that had been entered into the PO system, “Why don't you consider this lower cost alternative?” By opening up the visibility on the Procurement process using Social Business, time and money can be saved.

Social Business Standards

Activity Streams, OpenSocial, and potentially OAuth are involved in this use case.

1.3. Standards Gaps

Activity Streams

Some activities that are generated by a business process may require a user to take action. However, in the current Activity Streams specification there is **no way to identify an item as requiring action**. Additionally, there is no means to specify what kind of action is required, due date, etc.

Many social platforms provide different ways that could be used to filter or organize activities around a particular business process. However, there is **no known standard way for identifying how activities should be categorized** when posting them into an Activity Stream.

1.4. Standards Recommendations

OpenSocial

OpenSocial is recommended since it provides an Activity Streams API and supports OAuth and Embedded Experiences.

Use the OpenSocial Embedded Experiences within the Activity Stream. This may require some custom development but it is possible, with little effort, to create OpenSocial applications that surface only the information relevant to the user viewing the activity. This provides a focused user experience that is not distracted by the complexity of pivoting to another application. OpenSocial applications also supports OAuth as a security model.

OAuth

OAuth fits naturally with OpenSocial and provides a security model that will allow you to provide your Activity Stream application access to your backend applications without violating the principal of least privilege. OAuth allows you to scope access to resources very tightly to allow 3rd party applications (like the ActivityStream application) access to only the resources they need on behalf of the end user.

Architecture Overview Diagram