The need for a PubSubHub, and how social plays into this need

Position Paper for W3C & OpenSocial Workshop on Social Standards: The Future of Business

Author: Edward Krebs, Ford Motor Company

Introduction

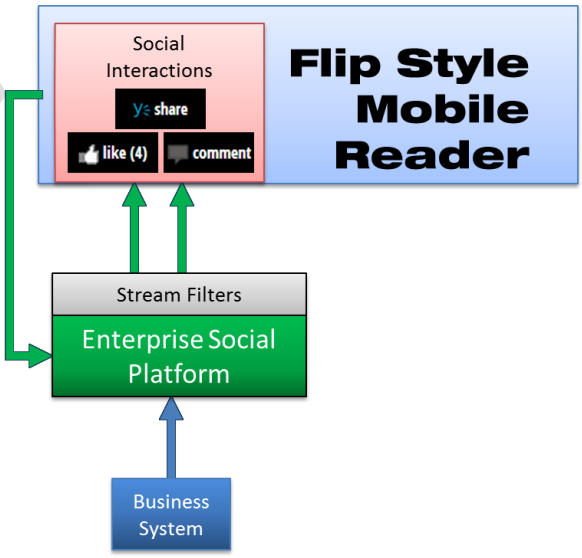
Ford has proposed in several venues the notion of Information Streams. The concept builds upon Activity Streams, but instead of system generated posts that are human-centric (e.g. Ed did this action), the posts are application centric – usually based on a workflow transition or system-to-system interface.

In order to manage all the extra posts without information overload, we also introduced the notion of prioritization (a current gap in standards) and channels – ways a user can configure the views of the information. To demonstrate how channel consumption would work in a mobile environment, we developed a proof of concept around an open source Flipboard like tool.

The idea was that like Flipboard, you should be able to do simple social interaction with the content, since it originates from a social platform. However, as we prototyped several information feed examples, we identified business systems where posting through the social platform was neither desirable nor added any value. However, there was still a need to consume these feeds from the same mobile application, which provided a great user experience by normalizing what information is presented and how to navigate through the channels.

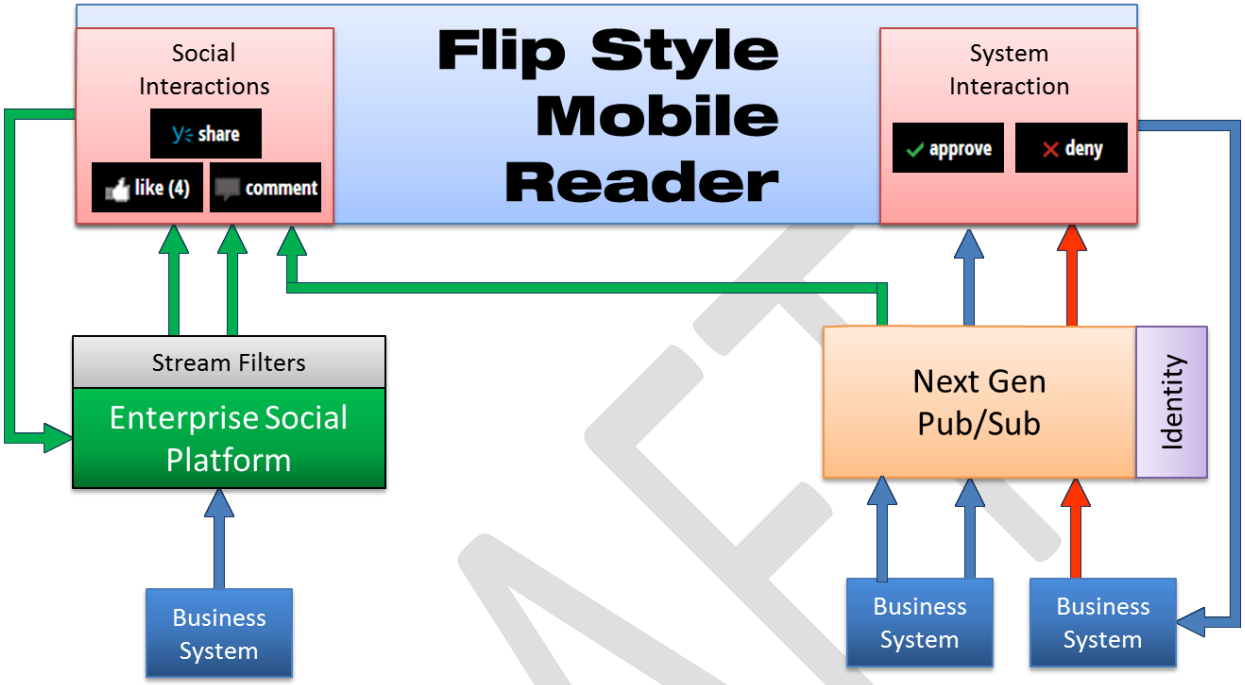
The Business Need for a PubSubHub

In order to understand the PubSubHub need, it is useful to see what the mobile app's interaction model looks like for social platform information sources. Here we see that "channels" leverage filters the user can define, which could be as simple as tags or groups. As the business system, or application, publishes into the Information Stream, the user can not only flip through and read the interesting content, but also interact with the content with social features such as sharing, liking and commenting. In this model, we assume that the content from the business system is suitable for exposure to a wide audience, since the true value of a social platform is for people that would not normally find information can now be informed and engaged.



However, not all business applications are really suited to be shared via a social platform. There are several reasons for this, which can include security needs or simply the volume of transactions that

would clog the social network. Since the focus of our exercise is user experience, we propose that the same consumption mechanism (in this case our Flip-style reader) can be used to consume, and perhaps interact with other business applications.



To do this we need some sort of publication and subscription hub. There are 4 key functions within this next generation PubSubHub.

First is the most obvious – the system is a Publisher. Methods and mechanisms exist for connecting to corporate applications, and formatting the information in such a way that various reader mechanisms can consume it. It is more than just an RSS feeder in that there will need to be a variety of connectors that can be developed or configured to get information from systems. This orchestration could be a part of the system itself, or could be a bolt-on capability. It will need to consider when and how to get the right information from a variety of systems, such as an event-driven architecture to consume the information based on key events within the application where the data is ready to be viewed.

Second is the ability for the application/data owner to configure the actions that can be taken. There are 2 types of actions that were demonstrated in the proof of concept. Social interaction is one possibility. The data owner should be able to identify what could be shared (expose the “Share” button), and also should be able to configure where the sharing is allowed. For example, many social platforms have the concept of private groups or communities, and the data owner should be able to set that destination or constraint as part of enabling the sharing function. Also, there may be simple system interactions like approving or denying request.

Third is that it needs to understand who the viewer is – their identity. Unlike the social platform model where information sharing is expected, business application owners will insist that the consumption of

information should follow the security model built into the application itself. If the additional capability mentioned above for providing system interaction is enabled, then the user identity will also be used to display or not display the action buttons based on the role from within the application. This paper does not propose a specific way to solve this, but it will be an important element to work through.

Finally is the subscription mechanism. The end user should have a variety of ways to help manage and consume information. Of course there would need to be a directory of available information systems ("publications") to subscribe to. If the system supports categories, tags or some other form of metadata, the user should be able to also filter the posts to the few that are relevant. It is also a desire to have the PubSubHub become a way for subscribers/users to create custom aggregation. For example, a manager would benefit greatly from creating a channel that contains all the "approve/deny" actions they need to take within various corporate systems, without the need to launch each business application.

Conclusion

The next generation PubSubHub should consider business use cases where the "publisher" is a business system owner, the system adheres to the originating system's security model, and the data owner can configure how and where information can be socially shared. It should also consider how next generation Flipboard-like readers can provide configurability in the subscription methods – including filtering and aggregation.

References:

https://github.com/botelho/flipboard-layout - An experimental page layout that lets you navigate pages by swiping or dragging as in a booklet, inspired by Flipboard. Licensed under the MIT License.