

---

# WebSphere Commerce V 7.0

## Social commerce architecture and programming model



This presentation discusses the social commerce architecture and programming model for WebSphere® Commerce V7.0.

## Goals

- Understand high level architecture of social commerce implementation

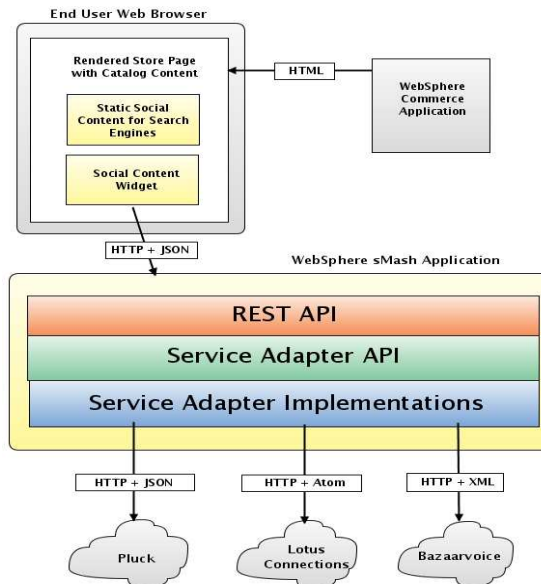
The goal of this presentation is to help you understand the high level architecture of the social commerce implementation.

## Agenda

- Architecture diagram
- Store page components
- WebSphere sMash application
- Administration
- Deployment

The agenda is to review the architecture diagram, and then go into more detail about the architecture components. The components covered are the store page components, the WebSphere sMash application, administration, and deployment.

## Social commerce architecture



4

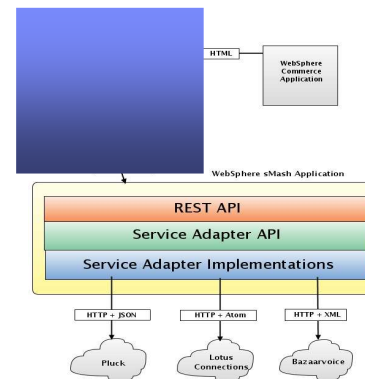
Social commerce architecture and programming model

© 2010 IBM Corporation

This slide shows a diagram of the social commerce architecture. The core of the architecture is a WebSphere sMash application that fetches and renders social content. The application interacts with the Web browser through a REST type programming interface running over HTTP and JSON protocols. The application also interacts with several external social content service providers through a corresponding set of service adapters. The remainder of the presentation will go into more detail for each of the blocks in the diagram.

## Store page components

- Social content widgets
- Static (SEO) social content
- Precision Marketing integration



This section describes the store page components, which are represented by the shaded area in the diagram. The store page components are the social content widgets, the static social content, and the Precision Marketing integration. The static social content is also known as SEO content, which stands for Search Engine Optimization. Each of these components are discussed in detail in subsequent slides.

## Social content widgets

- Dynamically render the social content
- Based on Dojo 1.3
- Highly interactive with asynchronous local refreshing
- Opaque resource IDs map social content with store objects
  - Product category
  - Product detail
- Customizable with CSS

Social content widgets dynamically render the social content served by the service providers. They are based on the Dojo 1.3 toolkit. These widgets are highly interactive components that refresh their respective page area independently of the other areas of the page. Opaque resource identifiers map the social content to store objects such as product categories and product details. The widgets are customizable with CSS.

## Sample blog widget declaration

- Sample declaration

```
<div
  dojoType="ibm.social.Blog"
  id="BlogWidget"
  title="My Sample Blog"
  resourceId="SampleResourceId">
</div>
```
- The resourceId attribute identifies a unique object in the store
- Example category resourceId:
  - `${WCParam.storeId}_${WCParam.catalogId}_${WCParam.categoryId}`
- Example product resourceId:
  - `${product.partNumber}`

The slide shows an example widget declaration for a blog widget. The resourceId attribute identifies a unique object in the store. The slide shows examples of a category and a product resourceId.

## Widget customization with CSS

- CSS class selector
  - Customize without affecting other non-widget page components
- CSS id selector
  - Set "id" attribute
  - Multiple widgets on the same page can be styled differently

Widgets are customized using CSS. Styling is accomplished with the CSS class selector which allows customizing the widget without affecting other non-widget page components. You can set the "id" attribute which allows you to do further customization with CSS "id" selectors. Using "id" selectors allows you to style multiple widgets of the same class differently.



## Static social content

- Cached social content data that is rendered using <noscript> tags
- Search engine optimized (SEO) to allow indexing by Web search engines
- Provides accessible view data for screen readers

Cached, and slightly stale, social content is rendered using noscript tags. This is done as a form of search engine optimization (SEO) to allow Web search engines to index the social content. It also allows accessible screen readers to read the same data.

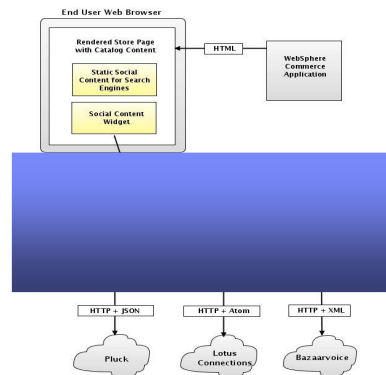
## Precision Marketing integration

- Shopper creates new social content
  - Blog post
  - Review
- Social content widget publishes activity event
  - Activates marketing engine trigger: **Customer Participates in Social Commerce**

Social commerce is integrated with Precision Marketing. When you create new social content on a store page, such as posting to a blog or writing a review, the widget publishes an activity event. This event activates a marketing engine trigger called "Customer Participates in Social Commerce". See the Information Center references slide for more information about social commerce and marketing triggers.

## WebSphere sMash application

- Overview
- User authentication
- Configuration
- REST API
- Service adapter API



This section describes the highlighted part of the diagram labeled WebSphere sMash Application and provides an overview of the application, user authentication, configuration, the REST API, and the service adapter API.

## WebSphere sMash application

- Light weight stand-alone Web server that provides
  - Dojo widgets
  - REST API
  - SEO content
- Runs in different process than WebSphere Commerce server
- IBM HTTP Server reverse proxy maintains same host name and port for store pages and service calls

The WebSphere sMash application is a stand-alone Web server that provides the Dojo widgets, REST API, and SEO content for social commerce. The application runs in a different process than the WebSphere Commerce server. The IBM HTTP Server reverse proxy configuration maintains the same host name and port for store pages and service calls.

## User authentication

- Authentication with WebSphere Lightweight Third-Party Authentication (LTPA)
- User signs in to store and receives an LTPA token containing user name in browser cookie
  - Token is encrypted
  - Cookie is transferred to sMash application through service calls
  - sMash application decrypts the token and retrieves the username
- On sMash application first start
  - LTPA key file and parameters are synchronized with application server instance

Users are authenticated using the WebSphere Lightweight Third-Party Authentication (LTPA) mechanism. You sign in to the store and receive an LTPA token in a cookie. This token is encrypted, and contains the username. This cookie is transferred to the sMash application through service calls. The sMash application decrypts the token and retrieves the username. When the sMash application is first started after being installed, the LTPA key file and parameters are automatically synchronized with the application server instance. This allows the LTPA tokens created by the application server to be decrypted by the sMash application.

## Configuration

- Flexible dependency and configuration system
- Adapter implementations plugged in with a change to dependency settings
- Configuration is provided by text files
- sMash Command Line Interface (CLI) is used to create and manage configuration tasks

The WebSphere sMash application uses a flexible dependency and configuration system. Adapter implementations can be plugged in with a simple change to the dependency settings. Configuration is provided by text files. The sMash CLI is used to create and manage configuration tasks.

## REST API

- Isolates widgets from service providers
- Provides multiple service types
  - Blogs
  - Photo galleries
  - Ratings and reviews
  - User profiles
- REST style interface with HTTP and JSON
  - Stateless server
  - Readable URLs that are cacheable

The REST API isolates the widgets from the service providers. It provides services for blogs, photo galleries, ratings and reviews, and user profiles. The API implements a REST interface using HTTP and JSON. The benefits of using a REST interface include a stateless server and readable URLs that can be cached.

## Service adapter API

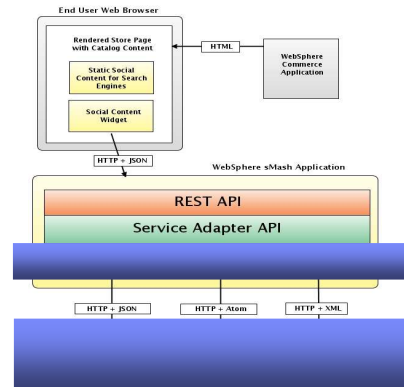
- Java™ API implemented for multiple service providers
  - Clean interface between REST API and individual service providers
  - Allows developers to plug in adapter implementations without modifying widgets or REST API
- Supports configurable user profiles
  - Unique profile implementations
  - Each adapter implementation defines which user profile fields are exposed

The service adapter API is a Java API that can be implemented for multiple service providers. It provides a clean interface between the REST API and the individual service providers. It allows developers to plug in the adapter implementation without modifying the widgets or the REST API. The API supports configurable user profiles with unique implementations. Each adapter implementation defines the user profile fields that are exposed.



## Service adapter implementations

- Service provider adapters
  - Pluck
    - Independent service provider offering Web based sales services
  - Bazaarvoice
    - Independent service provider offering Web based sales services
  - Lotus® Connections
    - IBM product for team collaboration



This section discusses the service adapter implementations and the corresponding external services. There are three choices offered for hosting social content. Pluck and Bazaarvoice are private enterprises who sell services to support Web based sales. Lotus Connections is an IBM product which provides Web based collaboration for teams.

## Service providers for social content

User social profiles	<ul style="list-style-type: none"> <li>▪Pluck SiteLife service</li> <li>▪Lotus Connections V 2.5</li> </ul>
Photo galleries	<ul style="list-style-type: none"> <li>▪Pluck SiteLife service</li> <li>▪Lotus Connections V 2.5</li> </ul>
Category blogs	<ul style="list-style-type: none"> <li>▪Pluck SiteLife service</li> <li>▪Lotus Connections V 2.5</li> </ul>
Product reviews	<ul style="list-style-type: none"> <li>▪Bazaarvoice Ratings &amp; Reviews service</li> </ul>

The social content data resides on servers external to the WebSphere Commerce implementation. If you are implementing user social profiles, photo galleries, or product category blogs, you have the choice of using Lotus Connections V 2.5 or the Pluck SiteLife service. If you are implementing product reviews, your only choice is to use the Bazaarvoice Ratings and Reviews service.

## Sample adapter configuration

```
#-----  
# The Blog adapter value can be either "pluck" or "lc" (Lotus Connections)  
#-----  
/config/soccom/adapter/blog = "pluck"  
  
#-----  
# The Photo gallery adapter value can be either "pluck" or "lc" (Lotus Connections)  
#-----  
/config/soccom/adapter/photoGallery = "pluck"  
  
#-----  
# The Profile adapter value can be either "pluck" or "lc" (Lotus Connections)  
#-----  
/config/soccom/adapter/profile = "pluck"  
  
#-----  
# The review adapter value can only be "bv" (Bazaar Voice)  
#-----  
/config/soccom/adapter/review = "bv"
```

This slide shows a fragment of the adapter configuration file. The file is customized by copying the contents of a template configuration file and editing fields to match your configuration. See the presentation "Configuring social commerce" for more details on the adapter configuration file.

## Development and deployment

- Administration
- Development environment
- Runtime environment

This section discusses methods for development and deployment. Administration tasks, the development environment, and the runtime environment are discussed.

## Administration

- Managed and updated using WebSphere Application Server management bean (MBean)
- MBean invoked with wsadmin utility
- MBean uses sMash CLI to
  - Start and stop
  - Query status
  - Generate proxy configuration

The WebSphere sMash application is managed and updated using a WebSphere Application Server management bean (MBean) which is invoked with the wsadmin utility. Under the covers, the MBean uses the sMash CLI to do the work of starting, stopping, querying status, and generating the proxy configuration.

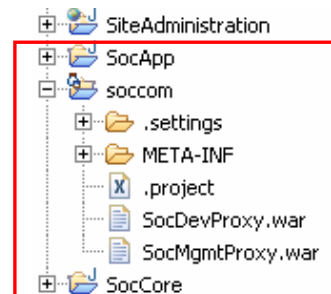
## Development environment

- WebSphere Commerce Developer
- Configuration
  - WCIntegrationWizard
  - Edit configuration file

WebSphere Commerce Developer is the preferred environment for social commerce development. You configure WebSphere Commerce Developer using WCIntegrationWizard and manual editing of configuration files. See the presentation "Configuring social commerce" for more details on performing configuration steps.

## Developer toolkit projects

- Soccom - enterprise application container
- SocDevProxy – simple proxy for routing HTTP to SocApp
- SocMgmtProxy – management MBean
- SocCore - library and sMash CLI
- SocApp - sMash application that specifies configuration and dependencies



After you import the social commerce project files into WebSphere Commerce Developer, you see three new projects. Soccom is a thin container for the enterprise application which contains deployment descriptors. Soccom contains two Web modules. The first Web module, SocDevProxy is a simple proxy for routing HTTP requests to SocApp. The second, SocMgmtProxy is the management MBean which controls starting and stopping the social commerce enterprise application. SocCore is the sMash library and CLI. SocApp is the main sMash application that specifies the configurations and dependencies.

## Runtime environment

- Configuration
  - WCIntegrationWizard or config\_ant command
  - Edit configuration file
  - IBM HTTP Server required to proxy requests to sMash application

You configure WebSphere Commerce Developer using the WCIntegrationWizard and manual editing of configuration files. The config\_ant command is a non-interactive alternative to the WCIntegrationWizard wizard. See the presentation "Configuring social commerce" for more details on performing configuration steps.



## General external references

- Pluck service provider  
<http://www.pluck.com/>
- Bazaarvoice service provider  
<http://www.bazaarvoice.com/>
- IBM Lotus Connections product  
<http://www.ibm.com/lotus/connections>
- Project Zero and sMash technology  
<http://www.projectzero.org/>
- Dojo toolkit  
<http://dojotoolkit.org/>
- Definition of REST methodology  
[http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)

This slide lists references for general background information.

## Information center references

- Precision Marketing triggers with social commerce

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.soccom.doc/tasks/tsmsocaddwidgets.htm>

This slide lists references in the WebSphere Commerce Information Center.

## Summary

- Architecture diagram
- Store page components
- WebSphere sMash application
- Administration
- Deployment

This presentation reviewed the overall architecture of the social commerce solution, the store page components, the WebSphere sMash application, administration, and deployment.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_SocialCommerceProgModel.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_SocialCommerceProgModel.ppt)

This module is also available in PDF format at: [./SocialCommerceProgModel.pdf](http://SocialCommerceProgModel.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM Lotus WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2010. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

29

© 2010 IBM Corporation